

Complete Coverage Path Planning Based on Improved Area Division

Lihuan Ma¹, Zhuo Sun^{1*}, Yuan Gao²

¹Transportation Engineering College, Dalian Maritime University, Dalian, China

²School of Maritime Economics and Management, Dalian Maritime University, Dalian, China

Email: mlh1120210426@dlnu.edu.cn, *mixwind@gmail.com

How to cite this paper: Ma, L.H., Sun, Z. and Gao, Y. (2023) Complete Coverage Path Planning Based on Improved Area Division. *World Journal of Engineering and Technology*, 11, 965-975.

<https://doi.org/10.4236/wjet.2023.114063>

Received: April 5, 2023

Accepted: November 27, 2023

Published: November 30, 2023

Abstract

It is difficult to solve complete coverage path planning directly in the obstructed area. Therefore, in this paper, we propose a method of complete coverage path planning with improved area division. Firstly, the boustrophedon cell decomposition method is used to partition the map into sub-regions. The complete coverage paths within each sub-region are obtained by the Boustrophedon back-and-forth motions, and the order of traversal of the sub-regions is then described as a generalised traveling salesman problem with pickup and delivery based on the relative positions of the vertices of each sub-region. An adaptive large neighbourhood algorithm is proposed to quickly obtain solution results in traversal order. The effectiveness of the improved algorithm on traversal cost reduction is verified in this paper through multiple sets of experiments.

Keywords

Generalized Traveling Salesman Problem with Pickup and Delivery, Complete Coverage Path Planning, Boustrophedon Cellular Decomposition, Adaptive Large-Neighborhood Search Algorithm, Mobile Robot

1. Introduction

With the development of the field of artificial intelligence in recent years, robotics as its representative product has been applied to various industries. Complete coverage path planning (CCPP) [1] is an important part of robot path planning research, which is the design of a path that traverses all areas of the environment, based on a priori information obtained from a map of the coverage area. It has a wide range of applications, such as maritime waste cleaning and patrolling, and path planning for floor sweeping robots.

There are three main 3 types of complete coverage path planning algorithms: traditional method, grid method and cell decomposition method. Among them, the boustrophedon cellular decomposition method (BCD) proposed by Choset [2] effectively reduces the complexity of the complete coverage planning problem. However, the selection of access order between subregions is not considered. For the selection of neighboring paths between subregions, Viet HH *et al.* [3] treat each subregion as a point and solve the inter-subregions path planning by solving the travelling salesman problem Zhou Lina *et al.* [4] traversed the adjacency graph with the DFS algorithm to find an exhaustive coverage path available for traveling salesman. Huang Jiahao *et al.* [5] established a sub-region adjacency graph based on the ant colony algorithm to solve the problem, which can quickly and efficiently obtain the visiting sequence of each sub-region. The existing literatures are solved by establishing the adjacency graph through the relative location relationship of each sub-region to obtain the access order of sub-regions, without considering the impact of inter-subregions path planning on the cost of intra-subregions path planning.

Unlike the existing literature, in this paper, we first partition the map into subregions according to BCD, and then treat the vertices of these subregions as entry and exit points. The robot can enter or leave the subregion from any vertex, so the problem of obtaining the visiting sequence of each subregion can be converted into a problem of the route where the vertices are visited. The robot can enter or leave the subregion from any vertex, so the problem of obtaining the order in which the robot visits the subregion can be converted into a problem of the order in which it visits these vertices. Then this problem can be described by the generalized traveling salesman problem with pickup and delivery (GTSPPD).

Both generalized traveling salesman problem and traveling salesman problem with pickup and delivery are NP-hard problems. The solving algorithm mainly consists of two parts: the exact algorithm and the heuristic algorithm. In terms of exact solutions, VD Šarić *et al.* converted GTSP into a traveling salesman problem for solving [6], but the existing solvers for TSP as an NP-Hard problem are equally difficult to solve for large-scale cases [7]. Exact algorithms for directly solving the GTSP include branch-and-bound algorithms [8], and branch-and-cut algorithms [9]. In terms of heuristic algorithm solving, specific local search algorithms [10], adaptive large neighbourhood algorithms [11], genetic algorithms [12].

The main contributions of this study are as follows: Firstly, the GTSPPD model is developed and solved to improve the complete coverage algorithm based on area division. Secondly, the adaptive large neighborhood search (ALNS) algorithm is designed for solving the GTSPPD model with respect to its characteristics. Finally, the efficiency of the improved complete coverage path planning algorithm based on GTSPPD and the superiority of the ALNS algorithm is demonstrated by the comparison of arithmetic examples.

2. Problem Description and Mathematical Model

2.1. Description of the Problem

After decomposing the environment map into multiple accessible sub-regions by BCD, the complete coverage path includes the coverage path within the sub-region and the transfer path between the different sub-regions. The transfer path costs of the vertices between different sub-regions are different, while the different vertices of the sub-region start and end the coverage of the sub-region will also produce different complete coverage paths, so two different paths need to be jointly considered for optimization. Thus all vertices of each region are treated as a cluster. For each sub-region $c \in C$, its internal CCPP can go from any point of the corresponding cluster to any point of that cluster. The set of start points is denoted as $P_c = \{p_1, p_2, \dots, p_c\}$ and the set of end points is denoted as $D_c = \{d_1, d_2, \dots, d_c\}$. The set of vertices of all subregions in the network is denoted as $V' = \bigcup_{c \in C} (P_c \cup D_c)$. The set of vertices, including the starting point, is denoted $V = \{0\} \cup V'$. Let $E = \{(i, j), i, j \in V\}$ be the set of all edges in the network, if an edge (i, j) , $i \in P_{c_1}$, $j \in D_{c_2}$, $c_1 = c_2$, $c_1, c_2 \in C$, The edge is a complete coverage path within a subregion and the cost is obtained by the Boustrophedon back-and-forth motions; if an edge (i, j) , $i \in D_{c_1}$, $j \in P_{c_2}$, $c_1 \neq c_2$, the edge is the transfer edge between two subregions and the cost can be calculated by the A^{*} algorithm [13]. The transfer cost for both cases is denoted as d_{ij} if an edge does not satisfy the above two cases, it is a redundant edge added for modelling convenience considerations and d_{ij} is an extreme value M .

At this point the CCPP requires finding the transfer order that minimizes the total path cost between the scan start point and the scan end point for all sub-regions, and satisfies the following conditions:

- The robot starts from the depot 0 and eventually returns to the depot;
- When visiting a polygon c , one of the start vertices $p \in P_c$ is visited and then one of the end points $d \in D_c$ is visited afterwards;
- Each sub-region needs to be accessed once and only once.

2.2. Mathematical Model

This model belongs to the generalized traveling salesman problem with pickup and delivery, where the vertices of each subregion are treated as a cluster, the start vertex is the pickup point, and the end vertex is the delivery point, assuming infinite carrier capacity and cargo size of 1. With x_{ij} as the decision variable indicating whether arc (i, j) is selected and y_i as a variable indicating whether vertex i is visited, the GTSPPD model is built as follows:

$$\text{minimize } \sum_{i, j \in V} d_{ij} x_{ij} \quad (1)$$

s.t.

$$\sum_{i \in V'} x_{ij} = \sum_{i \in V'} x_{ji} = y_i \quad \forall j \in V \quad (2)$$

$$\sum_{i \in P_c} y_i = 1 \quad \forall c \in C \quad (3)$$

$$\sum_{i \in D_c} y_i = 1 \quad \forall c \in C \quad (4)$$

$$\sum_{j \in V'} x_{0j} = \sum_{j \in V'} x_{j0} = 1 \quad (5)$$

$$\mu_j \geq \mu_i + 1 - (1 - x_{ij}) * M \quad \forall i, j \in V', i \neq j \quad (6)$$

$$\sum_{i \in P_c} \sum_{j \in D_c} x_{ij} = 1 \quad \forall c \in C \quad (7)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (8)$$

$$y_i \in \{0, 1\} \quad \forall i, j \in V \quad (9)$$

$$\mu_i \geq 0 \quad \forall i \in V' \quad (10)$$

The objective function (1) is to minimize the routing cost. Constraints (2) to (4) indicate that each subregion needs to have one point accessed for both the start and end point clusters species and only one point can be accessed. Constraint (5) indicates that the mobile robots all start from the starting point and return to the starting point. The purpose of constraint (6) is to eliminate sub-tour. Constraint (7) ensures the continuity of those start and end points that are visited within the same cluster. Constraints (8) to (10) define the domains of the decision variables.

3. Adaptive Large-Neighbourhood Search

The generalized traveling salesman problem with pickup and delivery belongs to the NP-hard problem, and the running time of taking the exact algorithm to solve the large-scale cases is too long, so an Adaptive Large Neighborhood Search algorithm is proposed in this paper for fast solution of this problem [14].

3.1. ALNS Framework

The ALNS algorithm flow is shown in **Figure 1**.

A high quality initial solution can speed up the algorithm search efficiency, so that the optimization results can be obtained faster. Therefore, a heuristic algorithm is designed to construct the initial solution in this paper.

The algorithmic procedure of constructing the initial solution by greedy algorithm is as follows:

Step 1: Initialize an empty path S0.

Step 2: Add the depot points to S0. Randomly select a cluster and choose the two points with the smallest distance to be added to S0. Repeat the process until all clusters are visited.

Step 3: Add the depot points at the end of S0 to get the initial solution.

Step 4: End.

3.2. Destroy Operators and Repair Operators

In this paper, three destroy operators and four repair operators are designed.

According to the characteristics of GTSPDP, the designed destruction operator

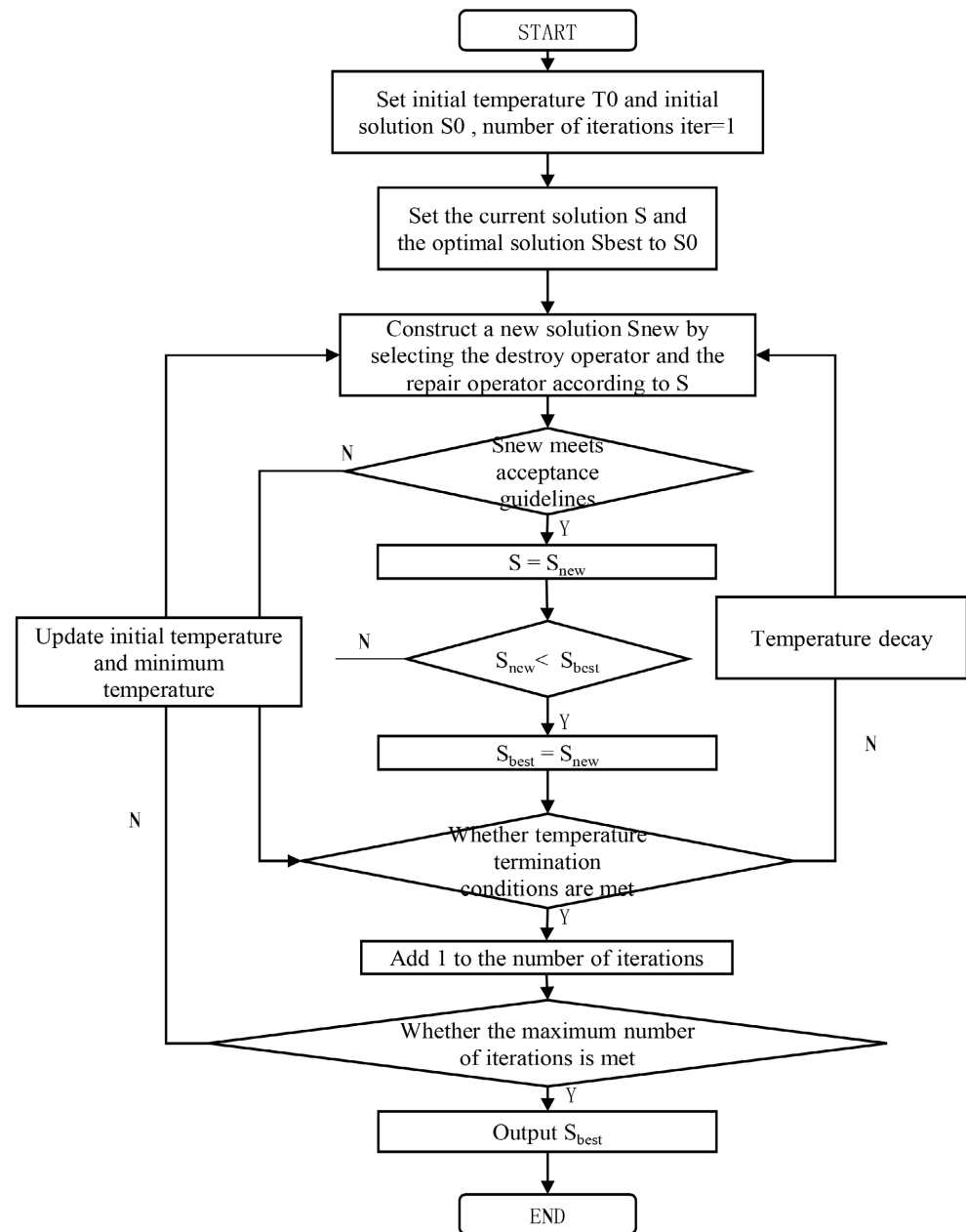


Figure 1. Flow chart of ALNS algorithm.

returns the customer point to its corresponding cluster after removing it according to the specific destruction rules, and the cluster becomes the cluster to be selected as the object selected by the repair operator. The destroying process is shown in **Figure 2**, the first path in the figure indicates the path before destroying, the red dashed line indicates the cluster selected by the destroying operator, and the cluster enters into the area to be selected after removal (the ellipse area in the figure). Each destroy operator is described as follows:

Random destroy operator: randomly select a certain number of clusters and remove both the pickup and delivery points belonging to the cluster in the path.

Worst transfer destroy operator: Remove all the two pickup points and two

delivery points with the highest transfer cost between the two sub-areas.

Complete destroy operator: destroys all vertices in a path.

Depending on the rules, the repair operator can select a cluster from the removed cluster, select the pickup and delivery points in it, and then repair it to the destroyed path. The repair process is shown in **Figure 3**. According to the repair operator, select a cluster from among the set which has the clusters need to be selected, select two points from the cluster as pickup and delivery points, and then insert them into the damaged path to form a new complete coverage path. Each repair operator is described as follows:

Random repair operator: randomly selects the start and end points in the removed cluster to be repaired to a random position in the path.

Best repair operator: selects the start and end points of the removed cluster with the lowest internal search cost and inserts them as a whole into the location with the lowest total path cost change.

Lowest cost repair operator: Selects the lowest cost start and end points of the path from the removed cluster and inserts these two points into the path at random. This operator and the best path repair operator can be seen as weaker versions of the best repair operator, retaining more randomness and enhancing the ability of the ALNS algorithm to jump out of local optima.

Best path repair operator: randomly selects the start and end points from the removed clusters and inserts them at the location where the total cost of the path

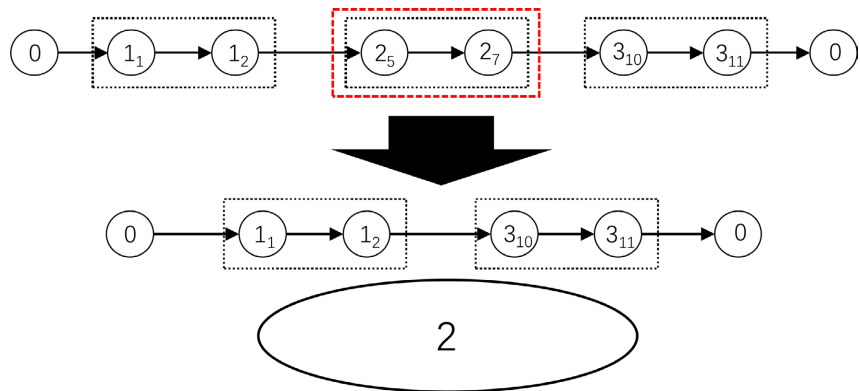


Figure 2. The destroying process.

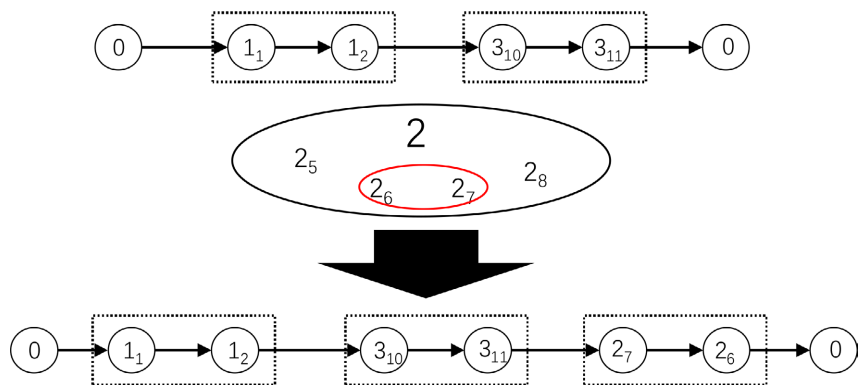


Figure 3. The repairing process.

changes the least.

3.3. Adaptive Strategy

Adaptive weights: After a new path is obtained in each iteration using the destroy and repair operators, the formula for calculating the score of the removal and placement operator used is as follows:

$$score = \max\left\{\frac{distance(S) - distance(S_{new})}{distance(S)}, 0\right\} \quad (11)$$

The formula calculates the latest score of the used operator by the degree of improvement of the newly generated paths over the old ones, using a selection of 0 with the maximum of the new scores to ensure that no negative values are generated. A new weight value is then calculated at the end of that round of iterations based on the score, with the new weight value being the weight value of b ($b \in (0,1)$) plus the score of the operator of $(1-b)$. In this paper the initial weights and the initial score are both 1. The choice of which destroy and repair operator to use is based on the weights in the form of a standard roulette wheel.

Acceptance function: In addition to directly retaining the result that is better than the current optimal solution, the result can also be retained and used as the initial solution for the next operation when it satisfies the acceptance function. The probability of a new path being accepted is $\min\{\exp((distance(S) - distance(S_{new})/T), 1)\}$, T is the current temperature T in simulated annealing.

4. Case Studies and Sensitivity Analysis

In this section, different sized arithmetic cases are generated for solving and analyzing the results according to the different working environments that the mobile robot may encounter. Each algorithm is named by serial number—number of clusters—number of vertices—grid precision. All algorithms in this section are programmed and implemented in Python 3.9, with Win10 operating system, 2.90 GHz CPU and 16 GB RAM.

4.1. Case Analysis

In order to verify the effectiveness of the improved algorithm, first the results obtained by the improved algorithm were compared with the Boustrophedon complete coverage algorithm [5] with the generated paths in a small-scale case. After that, in order to verify the effectiveness of the model with the ALNS algorithm, experiments with different scale cases were conducted. In this paper, the Gurobi algorithm is used to solve the GTSPPD model and the ALNS algorithm to solve the GTSPPD model with the Boustrophedon complete coverage algorithm for different scale comparison experiments, respectively.

4.1.1. Analysis of Results

Firstly, experiments are conducted in a small-scale algorithm. In this paper, a 20

× 20 grid map with obstacles is randomly generated, and each grid has a length and width of 1, where black is the obstacle. The improved area decomposition algorithm based on the GTSPPD model with Boustrophedon complete coverage algorithm is based on the environment map after cell decomposition to solve the complete coverage path, and the map is decomposed by the swept string partitioning method to obtain 11 subregions. The solution results of the improved algorithm with Boustrophedon complete coverage algorithm are shown in **Figure 4**.

The improved algorithm gives a total path distance of 393.1 and the Boustrophedon complete coverage algorithm gives a total path distance of 439.4. In complete coverage path planning, transfer paths used for other than coverage are often generated due to factors such as obstacle avoidance, and this part of the path includes both sub-region transfer paths and the part of repeated coverage within sub-regions, so this part of the path can be regarded as non-working paths. Compared with the paths obtained by Boustrophedon complete coverage algorithm, the improved algorithm results in 28.1 non-working paths transferred between different regions, while the non-working paths obtained by Boustrophedon complete coverage algorithm is 74.4, and the non-working path distance is reduced by 62.2%.

4.1.2. Model and Algorithm Performance Validation

In order to verify the effectiveness of the model and algorithm at different scales, this subsection generates 10 arithmetic cases of different scales from small to large on grid maps with different accuracies. And according to these 10 cases, experiments are conducted, in which the results obtained by the improved algorithm of solving the GTSPPD model by Gurobi solver, the improved algorithm of solving the GTSPPD model by ALNS algorithm and Boustrophedon complete

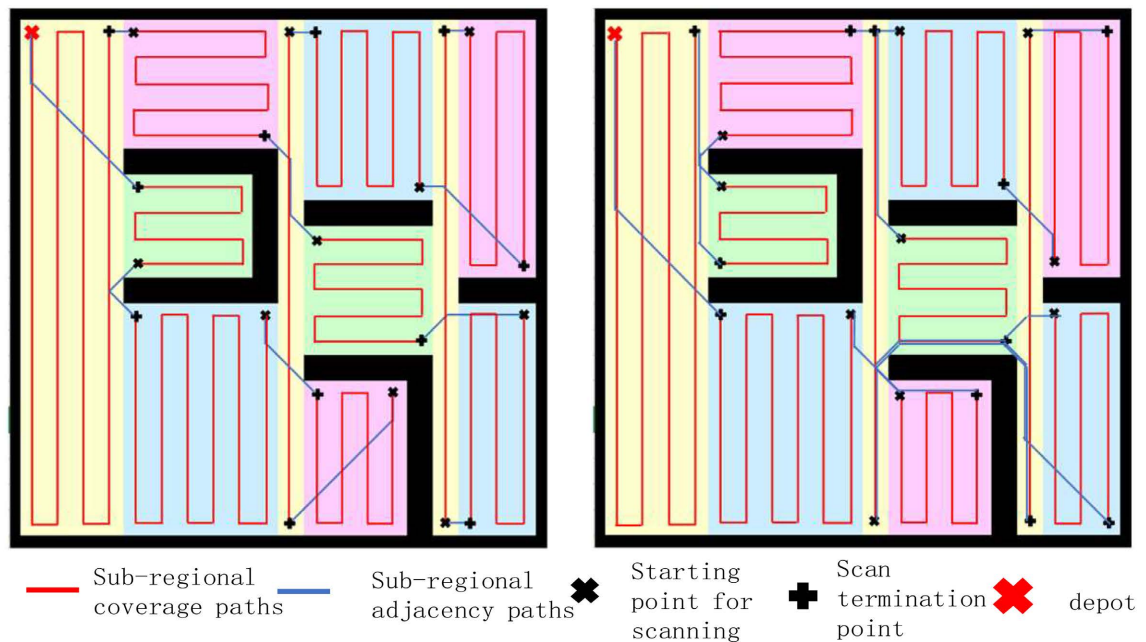


Figure 4. Comparison of Complete coverage path planning result.

coverage algorithm are compared for each case. The solution results are shown in **Table 1**. Time in the table indicates the time required for the algorithm to solve out the subinterval access order, and the solver's solution time is limited to 1800 s, and bold indicates the minimum cost as well as the shortest solution time obtained by different solving methods. The ALNS algorithm in the table solves the algorithm when it is a large-scale algorithm, and the solution time is all within 3 minutes, and the computation time is acceptable in the heuristic algorithm.

The difference between the results obtained by the ALNS algorithm and those obtained by Gurobi is very small, and in some cases the results are even the same. As the size of the solution increases, the GTSPPD model solution is not guaranteed to be optimal in the required time, and even in the largest cases the results are inferior to those obtained by the ALNS algorithm. In terms of solution time, except for the smallest accuracy cases, the solution time of the ALNS algorithm is much smaller than that of the GTSPPD model. This proves the effectiveness of the ALNS algorithm: results with higher accuracy can be obtained in a shorter time.

In all cases, the improved algorithm outperforms the Boustrophedon complete coverage algorithm, except for the 1-4-16-10 cases where the results are the same as the Boustrophedon complete coverage algorithm, and the difference in results becomes more pronounced as the size of the cases increases. Although the time is longer compared to the Boustrophedon algorithm, this computational time difference is negligible compared to the actual working time.

The above results show the total path cost and the comparison results of the non-working path cost obtained by the improved algorithm with Boustrophedon complete coverage algorithm are shown in **Table 2**.

The results for the non-working paths in the above 10 cases are shown in **Table 2**, and the GAP calculation formulae in the table are as follows:

$$GAP = (D_{\text{Boustrophedon}} - D_{\text{GTSP}}) / D_{\text{Boustrophedon}} \cdot \text{where } D_{\text{Boustrophedon}} \text{ is the non-working}$$

Table 1. Comparison of total path distances of different algorithms.

Name	Gurobi	Time (s)	ALNS	Time (s)	Boustrophedon	Time (s)
1-4-16-10	70.8	0.09	70.8	1.41	70.8	0.5
2-7-26-20	348.2	1.13	348.2	2.58	367.4	0.39
3-9-36-40	1417	29.79	1417	3.58	1446.9	0.75
4-10-40-50	2163.8	20.25	2163.8	4.05	2216.6	0.76
5-11-44-50	2077.6	1357.53	2077.6	5.89	2144.4	1.67
6-12-48-100	10,117.3	1800	10,117.3	6.16	10,236.1	2.26
7-15-60-100	8135	1800	8134.6	8.47	8312.8	2.13
8-16-64-100	8299.5	1800	8299.5	9.59	8488	2.24
9-27-108-100	8342.5	1800	8373.4	28.89	8730.2	3.80
11-52-208-100	10,223.8	1800	10,143	169.89	10,755	11.37

distance of the complete coverage path obtained by Boustrophedon complete coverage algorithm. From the table, it can be seen that the non-working paths obtained by the improved algorithm are effectively reduced compared to the Boustrophedon complete coverage algorithm, and in general this trend becomes more pronounced as the case size increases, with a maximum reduction of 63.95%. The results in **Table 1** and **Table 2** show that the results obtained by the GTSPPD model are significantly better than those obtained by Boustrophedon. These arithmetic examples fully demonstrate the superiority of the improved algorithm in this paper.

Table 2. Comparison of non-working path distance results.

Name	GTSPPD	Boustrophedon	GAP (%)
1-4-16-10	2.8	2.8	0
2-7-26-20	22.2	41.4	46.38
3-9-36-40	70	99	29.29
4-10-40-50	92.8	145.6	36.26
5-11-44-50	127.6	194.4	34.36
6-12-48-100	203.30	322.1	36.88
7-15-60-100	265.60	443.8	40.15
8-16-64-100	250.50	439	42.94
9-27-108-100	223.50	611.2	63.43
11-52-208-100	345	957	63.95

5. Concluding Remarks

In this paper, to solve the complete coverage path planning problem, a generalized traveling salesman problem model with pickup and delivery is proposed to jointly optimize the intra-traversal and inter-regional transfer paths for each sub-region according to the total cost. An ALNS algorithm for solving the problem is also developed for fast solution of the model. The effectiveness of the improved algorithm for solving the complete coverage problem is demonstrated through case studies.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Mac, T.T., Copot, C. Tran, D.T. and De Keyser, R. (2016) Heuristic Approaches in Robot Path Planning: A Survey. *Robotics and Autonomous Systems*, **86**, 13-28. <https://doi.org/10.1016/j.robot.2016.08.001>
- [2] Choset, H (2000) Coverage of Known Spaces: The Boustrophedon Cellular Decom-

- position. *Autonomous Robots*, **9**, 247-253. <https://doi.org/10.1023/A:1008958800904>
- [3] Viet, H.H., Dang, V.-H., Laskar, Md.N.U. and Chung, T.C. (2013) BA*: An Online Complete Coverage Algorithm for Cleaning Robots. *Applied Intelligence*, **39**, 217-235. <https://doi.org/10.1007/s10489-012-0406-4>
- [4] Zhou, L.N., Wang, Y., Zhang, X. and Yang, C.Y. (2020) Complete Coverage Path Planning of Mobile Robot on Abandoned Mine Land. *Chinese Journal of Engineering*, **42**, 1220-1228.
- [5] Huang, J.H., Hao, R.K. and Lv, G.Z. (2020) Map Full Traversal Path Planning Based on Ant Colony System Algorithm. *Software Guide*, **19**, 41-45.
- [6] Šarić, V.D. (1997) An Efficient Transformation of the Generalized Traveling Salesman Problem into the Traveling Salesman Problem on Digraphs. *Information Sciences*, **102**, 105-110. [https://doi.org/10.1016/S0020-0255\(96\)00084-9](https://doi.org/10.1016/S0020-0255(96)00084-9)
- [7] Karapetyan, D. and Gutin, G. (2012) Efficient Local Search Algorithms for Known and New Neighborhoods for the Generalized Traveling Salesman Problem. *European Journal of Operational Research*, **219**, 234-251. <https://doi.org/10.1016/j.ejor.2012.01.011>
- [8] Noon, C.E. and Bean, J.C. (1991) A Lagrangian-Based Approach for the Asymmetric Generalized Traveling Salesman Problem. *Operations Research*, **39**, 623-632. <https://doi.org/10.1287/opre.39.4.623>
- [9] Yuan, Y., Cattaruzza, D., Ogier, M. and Semet, F. (2020) A Branch-and-Cut Algorithm for the Generalized Traveling Salesman Problem with Time Windows. *European Journal of Operational Research*, **286**, 849-866. <https://doi.org/10.1016/j.ejor.2020.04.024>
- [10] Gutin, G. and Karapetyan, D. (2010) A Memetic Algorithm for the Generalized Traveling Salesman Problem. *Natural Computing*, **9**, 47-60. <https://doi.org/10.1007/s11047-009-9111-6>
- [11] Smith, S.L. and Imeson, F. (2017) GLNS: An Effective Large Neighborhood Search Heuristic for the Generalized Traveling Salesman Problem. *Computers & Operations Research*, **87**, 1-19. <https://doi.org/10.1016/j.cor.2017.05.010>
- [12] Nar, V., Ncan, T. and Süral, H. (2010) A Genetic Algorithm for the Traveling Salesman Problem with Pickup and Delivery Using Depot Removal and Insertion Moves. *Proceedings of the 2010 International Conference on Applications of Evolutionary Computation- Volume Part II*, Springer-Verlag, Berlin, 431-440. https://doi.org/10.1007/978-3-642-12242-2_44
- [13] Xu, Z., Liu, X. and Chen, Q. (2019) Application of Improved Astar Algorithm in Global Path Planning of Unmanned Vehicles. 2019 *Chinese Automation Congress (CAC)*, Hangzhou, 22-24 November 2019, 2075-2080. <https://doi.org/10.1109/CAC48633.2019.8996720>
- [14] Ropke, S. and Pisinger, D. (2006) An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, **40**, 455-472. <https://doi.org/10.1287/trsc.1050.0135>