Minimum Time Coverage Planning for Floor Cleaning Robot Using Line Sweep Motion

Myoung Hwan Choi
Dept. of Electronic and Communication Eng.
Kangwon National University
192-1 Hyoza 2 dong, Chuncheon, Kangwondo, Korea
mhchoi@kangwon.ac.kr

***Abstract*—A coverage problem for a floor cleaning robot is studied where the coverage region consists of polygonal cells, and line sweep motion is used for coverage. Since curved paths or making turns inside the coverage area can lead to uncovered region, it is desirable to employ line sweep cleaning path. This work studies the optimal line sweep coverage where the sweep paths of the cells consist of straight lines and no turn is allowed inside the cell. An optimal line sweep coverage solution is presented when the line sweep path is parallel to an edge of the cell boundary. The total time to complete the coverage task is minimized. The optimal sequence of cell visits is computed in addition to the optimal line sweep paths and the optimal cell decomposition.**

Keywords*—line sweep motion; optimal coverage; group traveling salesman problem; floor cleaning robot

I. INTRODUCTION

The floor cleaning robot is being used more frequently as its price is lowered to an affordable price range. Since the floor cleaning robot is often operated by battery power, it is crucial to find the optimal cleaning path that uses the minimum energy or minimum time. The coverage problem in the field of robotics is the problem of moving a sensor or a cleaning tool over a given region. It has been utilized in floor cleaning [1-2], lawn mowing [3], mine hunting [4], automated harvesters [5], and window cleaners [6]. Subsea application of the coverage problem includes autonomous underwater coverage [7], mine detection and classification in mine countermeasure tasks [8-9].

This work addresses the coverage problem that arises in floor cleaning applications, and focuses on the offline algorithms that computes an optimal line sweep path for planar coverage. This work is inspired by the applications in which the robot must travel in a straight line, such as in the floor cleaning. Since curved paths or making turns inside the coverage area can lead to uncovered region, it is desirable to employ line sweep cleaning path, and the coverage algorithms that allow frequent turning maneuvers inside the cells cannot be used. This work builds upon [10] and [27] which computes the optimal sweep direction of cells and the optimal cell decomposition for the coverage region consisting of polygonal cells.

Extensive research works have been published on the topic of robotic coverage in a known environment. For example, Moravec and Elfes [11] proposed an approximate cellular decomposition model, where the workspace is decomposed into cells with the same size and shape. Arkin and Hassin [12] proposed an approach based on covering salesman problem. Hert et al. [7] and Jung [14] describes an algorithm for nonpolygonal region. Choset and Pignon [13] describes an offline coverage algorithm for polygonal region which performs line sweep decomposition called boustrophedon decomposition and creates a sequence of subregions using a heuristic Traveling Salesman algorithm. A survey of papers on coverage for robotics can be found in [15]. Yang and Luo [16] presents a neural network approach for coverage with obstacle avoidance. Fang and Anstee [17] decomposes the surveyable area into subareas using an approximation to the generalized Voronoi diagram and calculate the sub-area paths to obtain a mission plan. Works that address the optimal coverage have also been published extensively. For example, Arkin et al. [18] shows that coverage problems that minimize the number of turns executed and the travel distance on planar rectilinear regions are NP complete in general. Gabriely and Rimon [19] considers the problem of covering a continuous planar area by a square-shaped tool attached to a mobile robot. Jinemez [20] presents a coverage planning based on genetic algorithms. Unfortunately, these optimal coverage algorithms that allow frequent turning motions inside the cells cannot be used in the underwater coverage mission using sidescan sonar, because the data obtained from the sidescan sonar is useful only in a straight line motion.

Of the approaches to coverage planning, line sweep based coverage algorithm such as the boustrophedon cellular decomposition ([13], [21]) is the most closely related with the line sweep based coverage. Huang [10] uses the boustrophedon approach to achieve the optimal coverage. Huang's approach seeks to minimize the number of turns required to cover all cells to minimize the mission time. Huang shows that the optimal line sweep decomposition must use a sweep path that is parallel to an edge of the cell boundary, and computes the optimal cell decomposition of the coverage region and the optimal sweep directions of the cells. Choi [27] proposed the optimal underwater coverage for autonomous underwater vehicles when the sweep path of a cell is parallel to an edge of the cell boundary. The total time to complete the coverage task is minimized which is the sum of the sweep times of the cells and the travel times between the cells. This work extends Choi's work to the floor cleaning robot.

This paper is organized as follows. In Section II, the formulation of the optimal coverage problem as a Group TSP problem and its solution procedures are described for a given cell decomposition. The optimal line sweep path and the cell visit sequence for a given cell decomposition is computed. In Section III, the optimal line sweep path, the cell visit sequence and the optimal cell decomposition of the coverage region is computed. In Section IV, the proposed solution procedure is applied to a cell configuration example.

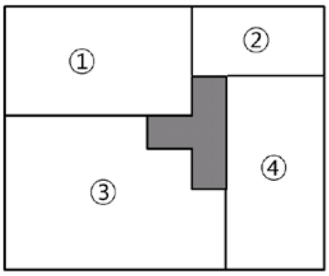


Fig.1. Cell configuration model example

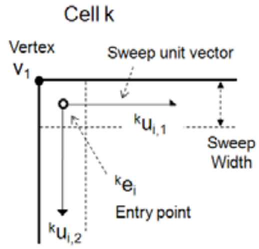
II. MINIMUM TIME LINE SWEEP PATH PROBLEM FORMULATED AS A GROUP TSP

A. Entry point and Exit point of a cell

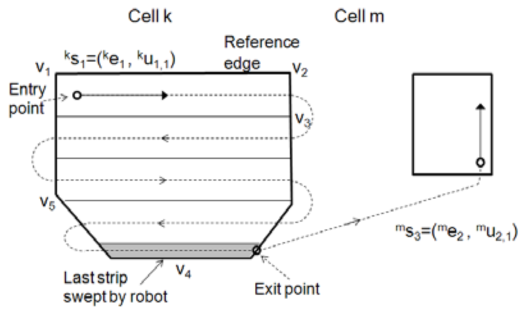
The coverage region consisting of four cells and an island shown in Fig.1 is used to illustrate the algorithm. The sweep path that covers each cell is composed of a sequence of parallel straight lines. The robot executes a back and forth line sweep motion within each cell. This work considers the case where the line sweep path inside each cell is parallel to an edge of the cell boundary. Each cell is covered in a single continuous sweep as shown in Fig.2(b), in a series of non-overlapping adjacent strips of equal width except for the last strip of the cell as shown by the shaded triangular region.

The cell configuration model of the coverage region in Fig.1 can result in a number of cell decompositions, varying from a decomposition consisting of four individual cells as shown in Fig.1 to a decomposition consisting of a single cell with all four cells merged together. This work differentiates the two optimal line sweep paths: 1) the optimal line sweep path of a given cell decomposition and 2) the optimal line sweep path of the coverage region. The former is computed when each cell is swept individually. The latter is computed from all possible cell decompositions that can result from merging the cells in the cell configuration. A two step approach is used to solve the optimal line sweep coverage problem. In step 1, the optimal line sweep path of a given cell decomposition is computed. Objective cost is the total coverage time which is the sum of cell sweep times and travel times between cells. In step 2, the optimal line sweep path of the coverage region is computed. Since merging cells into a larger cell can result in a shorter mission time, all possible cell merge possibilities, i.e. all possible cell decompositions, are investigated. The coverage time and the optimal sweep path of each cell decomposition is computed using the solution procedure in step 1. The cell decomposition resulting in the minimum mission time is selected as the optimal cell decomposition of the coverage region and the corresponding sweep path as the optimal sweep path. The procedure for the step 1 is described in the rest of this section, while the step 2 is illustrated in Section III.

Problem1: Compute the optimal line sweep path when a cell decomposition consisting of polygonal cells is given and each cell is swept individually. Objective cost is the sum of cell sweep times and travel times between cells



(a) Definition of entry point and sweep unit vector

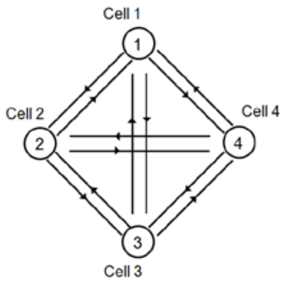


(b) Definition of exit point and an example of line sweep

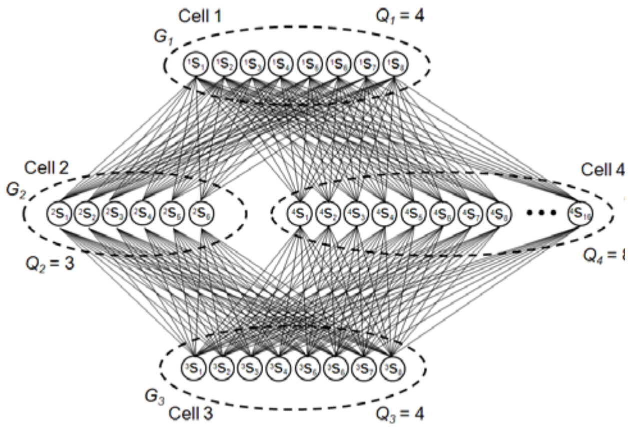
Fig.2. Entry point, exit point and an example of cell sweep

The solution procedure is illustrated using the cell decomposition shown in Fig.1. In order to compute the travel time between cells, the concepts of entry point and exit point of a cell are introduced. Let the vertices of the cell k with Q_k vertices be denoted as v_m, m=1,…, Q_k. Sweep path of a cell begins near a vertex of the polygon. There exist two possible direction vectors for the sweep path originating from each vertex, and the initial sweep paths are located inside the polygon boundary by one half of the sweep width as shown in Fig.2(a). The intersection of the two possible sweep paths is defined as the *entry point* of the vertex, and the entry points of cell k with Qk vertices are denoted as ^ke_i, i=1,…,Q_k. Let the edge of the cell boundary to which the sweep path is adjacent and parallel be called a reference edge. In Fig.2(b), the reference edge is the line segment v_1-v_2. Let the unit sweep direction vector be called sweep unit vector and the two possible sweep unit vectors for the i-th entry point of cell k, ^ke_i, be denoted as $^ku_{i,1}$ and $^ku_{i,2}$. Sweep motion is defined as a pair of an entry point and a sweep unit vector such that the set of sweep motions for cell k can be written as $\{^ks_n$, n=1, …, $2Q_k\}$ = $\{(^ke_i, {}^ku_{i,j})$, i=1,…,Q_k and j=1,2$\}$. The sweep path moves towards the vertex that is farthest from the reference edge. The farthest vertex in the example of Fig.2(b) is v_4. A sweep motion,

${}^{k}s_n$, thus completely describes a sweep of a cell by specifying the entry point and the sweep unit vector. A cell with Q_k vertices needs to be covered with only one of $2Q_k$ possible sweep motions. The intersection of the sweep path of the last strip and the cell boundary is defined as an *exit point*, as shown in Fig.2(b). Let us assume that the next cell to sweep is cell m. After the sweep of cell k, robot travels from the exit point of cell k to an entry point of cell m.



(a) Connection graph



(b) Coverage graph

Fig.3. For ease of readability, two directed edges between a pair of nodes are drawn as an undirected edge between the two nodes. The edges between cell 1 and cell 3, and the edges between cell 2 and cell 4 are not shown in the graph (b) and (c).

B. Formulation as a Group TSP

Connection graph of the cells is a graph where a node represents a cell, and a directed edge from cell i to cell j represents a visit from cell i to cell j. Tour of the cells is often represented by a connection graph. The connection graph for the cell model is shown in Fig.3(a), where visits from a cell to all other cells are possible. In order to deal with the coverage of the cells in terms of entry points and sweep unit vectors, the concept of coverage graph is introduced, which is defined to be a graph where a node represents a sweep motion ${}^{k}s_i$, and a directed edge from node ${}^{k}s_i$ to node ${}^{m}s_j$ represents the combination of the sweep of cell k with the sweep motion ${}^{k}s_i$ and the travel from the exit point of cell k to the entry point of ${}^{m}s_j$. The coverage graph is constructed from the connection graph by replacing the cell k of the connection graph with a group of sweep motions $\{{}^{k}s_1, {}^{k}s_2, \ldots, {}^{k}s_{2Q_k}\}$ of the cell k as shown in Fig.3(b). The sweep motions are the nodes of the coverage graph. This reflects the fact that visiting one of the sweep motions in the group for a cell is equivalent to visiting the cell. A directed edge from cell i to cell j in the connection graph is replaced by all possible directed edges from the sweep motions for cell i to the sweep motions for cell j in the coverage graph. The edge cost from node ${}^{k}s_i$ to node ${}^{m}s_j$ is defined to be the sum of the sweep cost of cell k with the sweep motion ${}^{k}s_i$, and the travel cost from the exit point of cell k to the entry point of ${}^{m}s_j$ arriving with the vehicle heading direction aligned with the sweep unit vector of ${}^{m}s_j$. By definition, the edge cost from node ${}^{k}s_i$ to node ${}^{m}s_j$ includes the sweep cost of cell k but does not include the sweep cost of cell m, and thus is not equal to the reverse edge cost, i.e. the edge cost from node ${}^{m}s_j$ to node ${}^{k}s_i$.The coverage graph is thus a directed graph. Edges between the nodes in the same group are not allowed since visiting two nodes in the same group would mean sweeping the same cell twice. For ease of readability, two directed edges between a pair of nodes are drawn as a single undirected edge in Fig.3(b). For example, the undirected edge from ${}^{1}s_1$ to ${}^{2}s_1$ in fact represents two directed edges: a directed edge from ${}^{1}s_1$ to ${}^{2}s_1$ and a directed edge from ${}^{2}s_1$ to ${}^{1}s_1$. Also, the edges between nodes of cell 1 and nodes of cell 3, and the edges between nodes of cell 2 and nodes of cell 4 must be present but are not shown in Fig.3(b) for ease of readability.

Let the number of cells be denoted as M, and the groups of nodes G_i corresponding to cell i be written as $G_i = \{{}^{i}s_1, {}^{i}s_2, \ldots, {}^{i}s_{2Q_i}\}$. Let V denote a set of all nodes of G_i, i = 1, …, M , such that

$$V = \{{}^{1}s_1, {}^{1}s_2, \ldots, {}^{1}s_{2Q_1}, {}^{2}s_1, {}^{2}s_2, \ldots, {}^{2}s_{2Q_2}, \ldots, {}^{M}s_1, {}^{M}s_2, \ldots, {}^{M}s_{2Q_M}\}$$

The indices of the nodes in V need to be rearranged in a single sequence of numbers, so the set U is defined using V so that the elements of U and V are exactly the same, but the indices of U begins from 1 and ends at N, where N is the total number of nodes given by $N = \sum_{i=1}^{M} 2Q_i$.

$$U = \{u_1, u_2, \cdots, u_N\} \quad \text{where} \quad u_1 = {}^{1}s_1,\ u_2 = {}^{1}s_2,\ \cdots\ u_N = {}^{M}s_{2Q_M}$$

The element u_i is a node of the coverage graph. The node u_i describes a sweep motion of a cell by specifying the entry point and the sweep unit vector. The set of node indices belonging to cell k can be computed as follows.

$$I_k = \{n \mid a_k \le n \le b_k\}, \text{ where } a_k = 1 + \sum_{i=1}^{k-1} 2Q_i,\ b_k = \sum_{i=1}^{k} 2Q_i$$

Then, the nodes u_i , $a_k \le i \le b_k$ in the set U belong to cell k. Let c_{ij} denote the edge cost from node u_i to node u_j representing the visit from node u_i to node u_j. All possible visits between the nodes are included in the coverage graph. Hence, the coverage graph represents all combinations of the cell sweep paths possible from the cell decomposition.

The optimal line sweep path of the given cell decomposition can now be solved by searching for the optimal tour of the coverage graph such that the tour passes through all groups G_1, G_2, …,G_M, passing through only one node in each group, and return to the starting node. This is a problem referred to as a group traveling salesman problem (Group TSP). Since the Group TSP problem is known to be NP-hard ([22]), and it is an active area of research in algorithm research community (see for example,[23]), this work does not attempt to propose a general solution algorithm. Instead, the Group TSP problem is formulated as a binary integer programming problem, and the optimal solution is computed. Since the feasible solution space is well defined in the binary integer programming, the solution is found by exhaustive searching of the feasible solution space, and it is shown that in practice, a problem of a moderate scale can be solved quickly.

C. Solution of Group TSP by exhaustive search of the coverage graph

The Group TSP is solved by searching the coverage graph. Let

c_{ij} : cost of the edge from node ui to node u_j

x_{ij} : binary integer variable corresponding to the edge from node u_i to node u_j

such that

$x_{ij} = 1$ if the edge is included in the tour

$x_{ij} = 0$ if the edge is not included in the tour

Even though the size of the solution space of Group TSP is extremely large, the size of the feasible solution space of x_{ij} is relatively small, and can be used to solve the Group TSP problem. Let the set {1,2,3,4} represent the set of group numbers. The number of sequences by which the groups can be visited is P(M,M)= M! = 4! = 24, where P(M,M) denotes the number of M-permutations of a set of M elements. The group visit sequences in this example are given by 4-permutations of the set {1,2,3,4}. For each of the group visit sequences, there are $\prod_{i=1}^{M} 2Q_i$ = 8*8*16*6 = 6144 different ways the nodes can be visited. The total number of node visit sequences in the coverage graph is thus Ne = 24*6144 = 147,456. The expression for this number in general is given by $N_e = M!\prod_{i=1}^{M} 2Q_i$. For each of these node visit sequences, a vector $X_k=\{x_{ij}\}$ is assigned, so that element x_{ij} that has a value 1 implies the corresponding edge is included in the node visit. Since X_k, k=1,…, N_e is the set of all possible tours of the coverage graph, the remaining task in solving the Group TSP problem is to search for the X_k that results in the minimum cost J in (1). This method requires exhaustive search of Ne=147,456 feasible solution vectors and computationally feasible while direct application of binary integer programming to Group TSP problem requires binary search of 21444 vectors.

The Problem 1 is formalized below for the convenience of future reference.

(J*, X*) = Problem1 (D)

where the optimal sweep path X*={x*ij} results in the minimum mission time $J^* = \sum_{i,j} c_{ij} x_{ij}^*$ given the cell decomposition D = {Cell 1,Cell 2, …,Cell M}

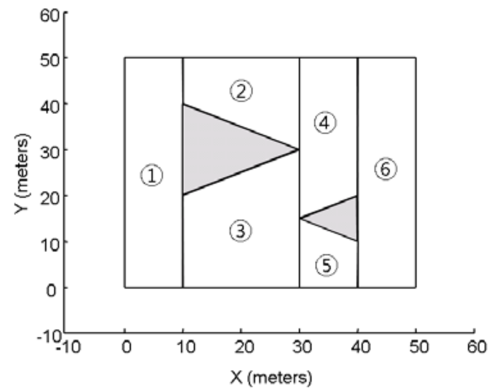


Fig. 4. Cell configuration model of the coverage region

III. OPTIMAL CELL DECOMPOSITION OF THE COVERAGE REGION

In Section II, the optimal line sweep path of the given cell decomposition and the mission time were computed. Since merging cells into a larger cell can result in a shorter mission time, the mission times and the sweep paths of all possible cell merges, i.e. all cell decompositions, are computed using Problem1. Huang [10] used merging multiple cells into a single cell using adjacency graph and dynamic programming. This paper examines the partitions resulting from the given cell configuration.

The total number of partitions of an n-element set is given by the Bell number B_n ([25]) which is defined by

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k , \ B_0 = 1, B_1 = 1$$

The first several Bell numbers are $B_0 = 1$, $B_1 = 1$, $B_2 = 2$, $B_3 = 5$, $B_4 = 15$. The list of all partitions, P_i ,i = 1,…,B_M, of a set of M cells can be computed by the script SetPartition.m obtained from the MATLAB Central webpage ([26]). All partitions of the set of four cells {1,2,3,4} are given by

P_1={1,2,3,4}, P_2={1,2,3}+{4},
P_3={1,2,4}+{3}, P_4={1,2}+{3,4},
P_5={1,2}+{3}+{4}, P_6={1,3,4}+{2},
P_7={1,3}+{2,4}, P_8={1,3}+{2}+{4},
P_9={1,4}+{2,3}, P_{10}={1}+{2,3,4},
P_{11}={1}+{2,3}+{4}, P_{12}={1,4}+{2}+{3},
P_{13}={1}+{2,4}+{3}, P_{14}={1}+{2}+{3,4},
P_{15}={1}+{2}+{3}+{4}.

For example, the partition P_2 = {1,2,3} + {4} consists of a large cell formed by merging cells 1,2 and 3, and a second cell which is the cell 4. The partition P_1 = {1,2,3,4} consists of a single cell obtained by merging the four cells 1,2,3 and 4, and the partition P_{15} ={1}+{2}+{3} +{4} is the partition consisting of the four individual cells. The partitions can be used as the cell decomposition input of Problem1. For each partition P_i, the optimal mission time J_i^* and sweep path X_i^* are computed, such that (J_i^*, X_i^*) = Problem1(P_i), i = 1,…, ,B_M. The partition resulting in the minimum mission time is chosen as the optimal cell decomposition $\hat{P}^*$ and the corresponding optimal sweep path is selected as the optimal sweep path $\hat{X}^*$ of the coverage region. The solution algorithm for Problem2 is summarized below.

Problem2: Compute the optimal cell decomposition $\hat{P}^*$ and the optimal sweep path $\hat{X}^*$ of the coverage region when the cell configuration is given.

Step 1: Compute all possible partitions of the cells,

P_i, i=1,… ,B_M.

Step 2: Solve Problem1 using each of the partition P_i as the cell decomposition input.

$$(J_i^*, X_i^*) = \text{Problem1}(P_i),\ \ i = 1,\dots, B_M$$

Step 3: Find the optimal solution

$$(\hat{X}^*, \hat{P}^*) = (X_m^*, P_m) \quad \text{such that}$$

$$m = \arg_i \min\{J_i^*\}, i = 1, \cdots, B_M$$

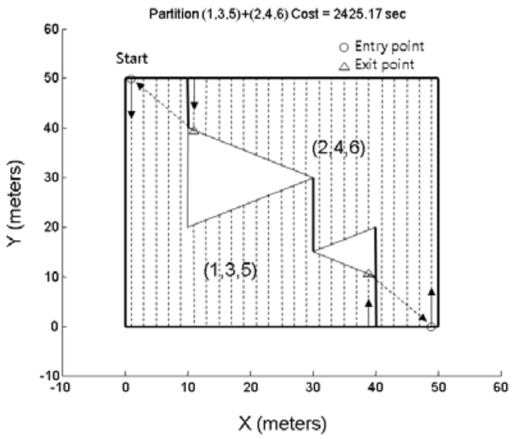


Fig. 5. The minimum time cleaning path of the coverage region and the optimal cell partition.

IV. SIMULATION RESULT

The cell configuration used in the computation of the minimum time coverage solution is shown in Fig.4, where the coverage region consists of six cells and two islands. MATLAB was used to implement the proposed algorithm on a computer with Windows 7 on Intel Core i5 CPU, 2.8GHz machine with 4GB memory. The linear traveling speed of the robot was 0.5 meter/sec, acceleration and deceleration were 0.5 meter/sec^2, and the time for turning was 0.5 sec. Total possible number of cell partitions, B_6, is 206, and the valid number of cell partitions was computed to be 52. The minimum time coverage path for these 52 cell partitions were computed, and the coverage path with the smallest coverage time of the 52 cell partition was chosen as the minimum time coverage path of the coverage area, and shown in Fig. 5. The optimal cell partition was found to be (1,3,5) + (2,4,6) where the cells 1,3,5 were merged into a single large cell, and cells 2,4,6 were merged into a second large cell. The entry points of the merged cells are marked with a small circle, and the exit points are marked with a small triangle. The total computation time was 4 hours and 51 minutes.

The coverage path of the merged cell {1,3,5} begins from the entry point marked with a small circle at the top left corner, and ends at the exit point at the lower right. The robot then moves to the entry point of the merged cell {2,4,6} at the lower right corner and completes the path at the top left corner. For comparison, the coverage path where each of the six cells was cleaned individually without the cell merge is shown in Fig. 6. It can be seen that the coverage time is reduced by merging the adjacent cells into a single large cell before coverage.

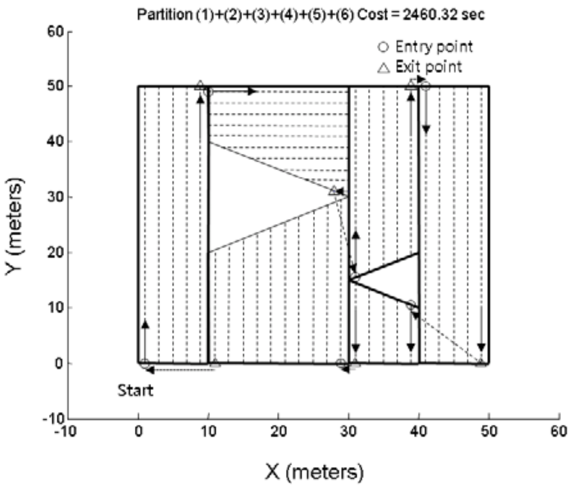


Fig. 6. The minimum time cleaning path of the 6 individual cells.

V. CONCLUSIONS

In this paper, a procedure that computes the minimum time line sweep path of a given cell decomposition is proposed that computes the cell sweep sequence in addition to the sweep path inside the cells. The total time to complete the coverage task is minimized which is the sum of the travel times between the cells and the sweep times of the cells. The optimal coverage problem is formulated as a group traveling salesman problem (Group TSP) by introducing the concepts of entry point, exit point, and coverage graph. The Group TSP problem is converted to a binary integer programming problem, which is then solved using exhaustive search of the feasible solution space. The partitions of the cells are examined to compute the optimal cell decompositions and the optimal line sweep path of the given coverage region. The proposed solution algorithm is applied to a cell configuration example, and it is shown that a problem of a moderate scale can be solved quickly.

REFERENCES

[1] Yasutomi, F., Takaoka, D., Yamada,M., and Tsukamoto, K., "Cleaning robot control", IEEE Int. Conf. Robotics Automation, Philadelphia, PA, pp.1839-1841, 1988..

[2] [2] Oh, J.S., Choi, Y.H., Park, J.B., and Zheng, Y.F., "Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map", IEEE Trans. Industrial Electronics, vol.51, no.3, pp.718-726, 2004.

[3] Huang, Y.Y., Cao, Z.L. and Hall, E.L., "Region filling operations for mobile robot using computer graphics", Proc. of the IEEE Conference on Robotics and Automation, pp.1607-1614, 1986.

[4] Najjaran, H. and Kircanski, N., "Path planning for a terrain scanner robot", Proc. 31st Int. Symp. Robotics, Montreal, QC, Canada, pp.132-137, 2000.

[5] Ollis, M. and Stentz, A., "First results in vision-based crop line tracking", Proc. IEEE Int. Conf. Robotics Automation, Minneapolis, MN, pp.951-956, 1996.

[6] Farsi, M., Ratcliff, K., Johnson, P.J., Allen, C.R., Karam, K.Z., and Pawson, R., "Robot control system for window cleaning", Proc. 11th Int. Symp. Automation Robotics Construction, Brighton, U.K., pp.617-623, 1994.

[7] Hert, S., Tiwari, S., and Lumelsky, V., "A terrain-covering algorithm for an AUV", Autonomous Robots, vol.3, pp.91-119, 1996.

[8] Reed, S., Petillot, Y., and Bell, J., "An Automatic Approach to the Detection and Extraction of Mine Features in Sidescan Sonar", IEEE J. of Oceanic Eng., vol.28, no.1, pp.90-105, 2003.

[9] Sariel, S., Balch, T. and Erdogan, N., "Naval Mine Countermeasure Missions", IEEE Robotics and Automation Magazine, March, pp.45-52, 2008.

[10] Huang, W.H., "Optimal line-sweep-based decompositions for coverage algorithms", 2001 IEEE Conf. on Robotics and Automation, pp.27-32, 2001.

[11] Moravec, H.P. and Elfes, A., "High resolution maps from wide angle sonar", IEEE Int. Conf. Robotics Automation, St. Louis, MO, pp.116-121, 1985.

[12] Arkin, E.M. and Hassin, R., "Approximation algorithms for the geometric covering salesman problem", Discrete Appl. Math., 55, pp.197-218, 1994.

[13] Choset, H. and Pignon, P., "Coverage path planning: The boustrophedon cellular decomposition", Int. Conf. on Field and Service Robotics, Canberra, Australia, 1997.

[14] Y.S. Jung, B.H. Lee, M.H. Choi, S.Y. Lee, K.W. Lee, "An Efficient Underwater Coverage Method for Multi-AUV with Sea Current Disturbance", Int. Journal of Control, Automation and Systems, Vol.7, No.4, pp.615-629, Aug., 2009

[15] Choset, H., "Coverage for robotics - A survey of recent results", Annals of Mathematics and Artificial Intelligence, vol.31, pp.113-126, 2001.

[16] Yang, S. and Luo, C., "A Neural Network Approach to Complete Coverage Path Planning", IEEE Trans. on Systems, Man, and Cybernetics - PART B, vol.34, no.1, pp.718-725, 2004.

[17] Fang, C. and Anstee, S., "Coverage Path Planning for Harbour Seabed Surveys using an Autonomous Underwater Vehicle", OCEANS 2010, pp.1–8, 2010.

[18] Arkin, E.M., Bender, M.A., Demaine, E.D., Fekete, S.P., Mitchell, J.S.B. and Sethia,S., "Optimal covering tours with turn costs", Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, Washington. D,C., United States. pp.138-147, 2001.

[19] Gabriely, Y. and Rimon, E., "Spanning-tree based coverage of continuous areas by a mobile robot", Annals of Mathematics and Artificial Intelligence vol.31, pp.77-98, 2001.

[20] Jimenez, P.A., Shirinzadeh, B., Nicholson, A., Alici, G., "Optimal Area Covering using Genetic Algorithms", Proceedings of 2007 IEEE/ASME int. conf. on Advanced Intelligent Mechatronics, pp.1-5, 2007.

[21] Garcia, E. and Gonzalez, P. de Santos, "Mobile-robot navigation with complete coverage of unstructured environments", Robotics and Autonomous Systems, vol.46, pp.195-204, 2004.

[22] Safra, S and Schwartz, O., "On the Complexity of Approximating TSP with Neighborhoods and Related Problems", Computational Complexity vol.14, pp.281- 307, 2005

[23] Elbassioni, K., Fishkin, A.V., Mustafa, N.H., and Sitters, R., "Approximation Algorithms for Euclidean Group TSP", 32nd International Colloquium, L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp.1115-1126, 2005.

[24] Laporte, G.," The traveling salesman problem: An overview of exact and approximate algorithms", European Journal of Operational Research vol.59, pp.231-247, 1992

[25] Rota, G., "The Number of Partitions of a Set", American Mathematical Monthly, vol.71, no.5: pp.498-504, 1964.

[26] Luong, B., "MATLAB Script for computing all partitions of a set of n elements", [webpage] http://www.mathworks.cn/matlabcentral/fileexchange/24133-set-partition/content/SetPartFolder/SetPartition. m., May, 2012

[27] Myoung Hwan Choi, "Optimal Underwater Coverage of a Cellular Region by Autonomous Underwater Vehicle Using Line Sweep Motion", Journal of Electrical Engineering & Technology, vol.7, no.6, pp. 1023-1033, 2012.11.