Sweep Coverage with Multiple Mobile Sensors

Project for Algorithm: Analysis and Theory

Xiaofeng Gao
Department of Computer Science,
Shanghai Jiao Tong University, Shanghai, China

Abstract. This project introduces the concept of sweep coverage with multiple mobile sensors. It includes the brief introduction of sweep coverage, the concepts and the definitions regarding min-max individual trajectory problem. Finally, it lists all the requirements and rules for each group. Please read this document carefully and complete the corresponding tasks.

Keywords: Sweep Coverage, Mobile Sensor, Trajectory

1 Sweep Coverage Problem

Assume there are \( n \) targets in an objective region. Each target \( t_i \) has its coordinate \((x_i, y_i)\) and cannot move during the detection period. A target is said to be detected by a mobile sensor \( s_j \) if \( s_j \) moves to the position of \( t_i \). The trajectory of a mobile sensor is a cycle which starts from a target, travels across every target once, and returns back to the starting target. Define \( \text{Len}(s_i) \) as the trajectory length of the sensor \( s_i \). The sweep coverage is to collect the information of all the targets by a mobile sensor, such that the trajectory length is minimized. If we consider the distance between targets as the Euclidean distance, then this problem is exactly the Euclidean Travelling Salesman Problem (ETSP), which is NP-complete in the plane.

Now we hope to improve the efficiency of sensor detection process. Instead of using only one mobile sensor, assume we have \( m \) sensors to work cooperatively. Traditionally researchers have two ways to deal with the sweep coverage with multiple sensors:

1. Compute a single TSP cycle, and split it into \( m \) segments, each with equal length.
2. Partition the target set into \( m \) clusters, and compute the TSP cycle for each cluster respectively.

Unfortunately, both of the above methods fail to pursue an optimal solution. Fig. 1 shows two counter examples for both scenarios. In Fig. 1 (a), two mobile sensors work together to detect four targets along
a rectangular path. Their individual trajectory length should be
\[ \text{Len}(s_1) = \text{Len}(s_2) = \frac{(1+2) \cdot 2}{2} = 3. \]
However, a better way is to detect targets separately as shown in Fig. 1 (b), with trajectory length of 2 each. Similarly, in Fig. 1 (c), two mobile sensors work separately for six targets located on the vertices of a hexagon. Each of them has a trajectory length of \( 2 + \sqrt{3} \approx 3.732 \), while a better schedule should be assigning them together along the hexagon, and the average trajectory length will be \( 6/2 = 3 \) instead.

We may implement these two ideas together and try to determine a best schedule. For example, in Fig. 2, there are 6 distinct routes with 10 mobile sensors. Some of the cycles are scheduled for single sensor, while some others are assigned for multiple sensors together. If all the sensors start detection process simultaneously, we hope to complete the procedure as soon as possible and reduce the energy consumption. Thus, we are aiming to minimize the longest trajectory length and guarantee that each target is visited by some sensor once.

**Fig. 2. An Example of Cooperative Sweep Coverage**

## 2 Tasks and Requirements

In this project, you are required to finish the following tasks.

### 2.1 Problem Formulation

We define our problem as *Cooperative Sweep Coverage* problem (CSC). Given \( n \) targets and \( m \) mobile sensors, design a trajectory schedule for mobile sensors such that each target is detected once and the longest trajectory length is minimized. Please formulate this problem formally as a mathematical programming. Your goal is \( \min \max_{1 \leq j \leq m} \text{Len}(s_j) \), and you need to complete the description of every constraint. Moreover, can you convert your programming as an ILP (Integer Linear Programming)?

### 2.2 CSC in One-Dimensional Space

If we restrict the domain of our discussion to one-dimensional space, say, every target lays along a line, then CSC problem is actually polynomial-time solvable. Please design an efficient algorithm to solve this problem in polynomial time with respect to the input size. You need to complete the following steps:
1. Describe your design first, introduce the necessary concepts, symbols, definitions, etc., and write the pseudo code of your design.
2. Analyze the time complexity of your algorithm.
3. Prove the correctness of your design.
4. Draw a numerical example (with coordinate information) with more than 20 targets and 3 sensors to illustrate your design.
5. Compare your method with the above mentioned two traditional methods (you need to explain how to cluster the targets).

2.3 CSC in Two-Dimensional Space

If we consider CSC problem in two-dimensional space, then the problem becomes NP-complete and we could hardly propose polynomial-time optimal algorithms. Correspondingly, try your best to solve this problem. Similarly, you need to consider the following questions:

1. For small-scale instance, can you propose an algorithm to compute an optimal solution? (Now the time complexity of your design may not be polynomial any more.) Prove the correctness of your design.
2. For large-scale instance, design efficient polynomial-time algorithm to solve CSC problem. Discuss the time complexity of your design.
3. Try to analyze the approximation ratio of your algorithm and prove it.

2.4 Numerical Experiments

Test the efficiency of your design by simulations, where $n$ is set from 20 to 1000, and $m$ is set from $\frac{n}{10}$ to $\frac{n}{5}$. You need to draw two sample trajectory schedule with $m = 50$, $n = 5$ and $m = 100$, $n = 10$. Next, compare your method numerically with the above mentioned two traditional methods, which can be viewed as the baselines. You also need to introduce what algorithms you will use to compute the two baselines. Plot figures for comparisons, where the value of each point should be the average value of at least 50 different calculations. The more instances you calculate, the more accurate your results will be. Note that you may also try some heuristics for CSC problem with faster executing time.

2.5 Report Requirements

You need to submit a report for this project, with the following requirements:

1. Your report should have the title, the author names, IDs, email addresses, the page header, the page numbers, figure for your simulations, tables for discussions and comparisons, with the corresponding figure titles and table titles.
2. Your report is English only, with a clear structure, divided by sections, and may contain organizational architecture like itemizations, definitions, or theorems and proofs.
3. Please define your variables clearly. If needed, a symbol table is strongly recommended to help readers catch your design.
4. Please also include your latex source and simulation codes upon submission.