The Data Gathering Problem for Sensors on a Line or a Tree

Jiemin Zeng* Joe Mitchell† Jie Gao*

1 Problem Definition

Suppose there are n sensors at points $\{p_1, p_2, \ldots, p_n\}$ and a robot traveling at a constant speed s is to collect data from these sensors. Let $d(p_i, p_j)$ be the distance between points p_i and p_j. A sensor at p_i generates data at a fixed rate of r_i and has a storage capacity of c_i (i.e., bucket size). When a robot visits a sensor, all current data stored in the sensor is collected. We assume that data collection at each sensor happens instantaneously, i.e., we ignore the time of data transmission, which is typically much smaller than the time taken by the robot to move between the sensors. If the amount of data generated at a sensor goes beyond its capacity (i.e., its bucket is full), additional data generated is lost. The question we ask is: Can we determine a route for the robot to collect data from the sensors that minimizes the amount of data that is lost? Equivalently, can we maximize the amount of data collected? Notice that data collection, and robot movement, is assumed to continue ad infinitum.

This problem belongs to the general family of traveling salesman problems or vehicular routing problems with constraints [4, 3, 2, 1]. It is naturally motivated by the scenario of a static sensor network monitoring an environment with a mobile data mule to collect sensor data. It can also be used to model the case of battery recharging and energy management in a sensor network. In that case, each sensor sensor i uses its battery with capacity c_i at a rate of r_i. When the battery at a sensor is depleted the sensor becomes ineffective. Thus one would like to minimize the total amount of time of ineffectiveness, over all sensors.

2 Algorithms for the Line Case

We first look at the case when the points are on a line. Without loss of generality, we assume that the minimum data rate is 1. In fact, we can further assume that all sensors have a data rate of 1 – if a sensor has data rate $r_i > 1$, we can replicate this sensor with r_i copies, each with unit data rate and capacity c_i/r_i. Thus, in the following discussion we focus on the case of all sensors having unit data rates. In addition, we assume that the input data is integral; specifically, the sensors p_i are located at integer coordinates and the values c_i for all i are integers. With this assumption, the optimal schedule can be shown to be periodic.

Lemma 1. *The optimal schedule that minimizes data loss is periodic, assuming integral input data.*

Proof. If the sensors are located at integral positions, the distances between any two of them are integers as well. Thus, all states of the problem can be encoded by the position of the robot and the current amount of data at each sensor i. All of these values are integers. Thus, the total number of possible states is finite; after a state reappears we realize that the robot must follow the same schedule, making the schedule periodic. □

Theorem 1. *Assume that the capacities and rates of all sensors are the same: $c_i = c$ and $r_i = 1$, for $1 \leq i \leq n$. Then there exists an optimal path that minimizes data loss with the following properties: (1) its leftmost and rightmost points are at sensors, (2) it is a path making U-turns only at the leftmost and rightmost points.*

Proof. The first claim is obvious since one can remove the portion of the path beyond the leftmost sensor and shorten the time for one period of the trip, which results in all covered sensors being visited more frequently than before. Now, let there be an optimal path P with leftmost sensor at x_l and rightmost sensor at x_r.

Let us assume that P contains U-turns that do not occur at the extreme points x_l, x_r. A U-turn is called a left U-turn if the robot was traveling from right to left before the turn, and a right U-turn otherwise. Note that one cycle of P can be split into two sub-paths, a path P_l from x_l to x_r and a path P_r from x_r to x_l. These sub-paths may have U-turns at the extreme points x_l, x_r. Without loss of generality, assume that there are U-turns in the sub-path from x_l to x_r. Consider the closest left U-turn to x_l and name the point it occurs at z_l. Since P is periodic, we can assume that in one full period, P starts and ends at x_l.

Now, we will modify the trip P. Notice that for the robot to make a left U-turn at z_l, it must make a right U-turn in the path from x_l to z_l. Let the rightmost such U-turn be z_r. We remove the path that begins at the first time P_l reaches z_l to the last time P_l reaches z_l. See Figure 1 for an example.

After the surgery, we get a schedule without the round trip beginning at z_l. Suppose the trip eliminated has length

*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. Email: {jiezeng, jgao}@cs.stonybrook.edu.

†Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794. Email: jsbm@ams.stonybrook.edu.

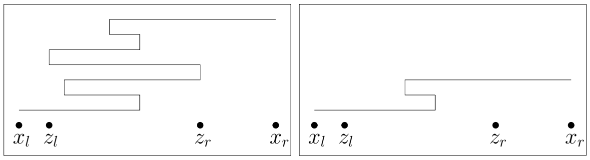

Figure 1: Remove a segment of the path to eliminate a U-turn at z_l.

δ and the trip after this surgery, denoted by P', has round-trip time length T. That means, the trip P has length $T+\delta$. Since

Since the rate of data accumulation and the bucket size are the same for all sensors, all sensors take the same amount of time to overflow. As a result, when a robot travels from right to left and encounters an overflowing sensor b, then all sensors to the left of b are also overflowing – since the last time they were visited was definitely before the last time that b was visited.

In the shortened path P', the sensors to the left of z_l are reached δ time sooner than in P. In the paths P and P', all sensors to the left of z_l overflow in both paths or there exists a sensor to the left of z_l that does not overflow in either P or P'. Let the leftmost such sensor be y_l. We define b_l to be the leftmost of either x_l or y_l. Let n_l denote the number of sensors to the left of b_l. Since all sensors to the left of b_l overflow in both P and P', the amount of data lost in those sensors is δn_l. Also, note that there exists a path in either P or P' from y_l to x_r to y_l where none of the sensors from x_r to y_l overflow. The terms b_r, y_r, and n_r are defined similarly in the right direction.

Let m denote the amount of overflow for sensors covered by P' in one full period and let n'' be the number of sensors not covered by the path P' (i.e., those that are either to the left of x_l or the right of x_r). The overflow rate is $m/T + n''$ in path P' and for P is $\frac{m+\delta(n_l+n_r)+e}{T+\delta} + n''$ in path P where e is any extra overflow in P that is not accounted for.

Since P is an optimal path, then $\frac{m+\delta(n_l+n_r)+e}{T+\delta} + n'' \leq m/T + n''$. We simplify this relationship and get

$$n_l + n_r + \frac{e}{\delta} \leq \frac{m}{T}. \tag{1}$$

Consider a simple periodic path P'' only making U-turns at b_l and b_r. If b_l or b_r is defined by a non-overflowing sensor, then there is no overflow between b_l and b_r since there exists a longer path between b_l and x_r (or b_r and x_l) in P or P' that does not cause any sensor to overflow. Thus the overflow rate of the path is just $n_l + n_r + n''$. If b_l is z_l and b_r is z_r then the overflow rate of the path is $n_l + n_r + \frac{e'}{\delta} + n''$ where e' is the extra overflow. Note that by definition, $e' \leq e$ since a cycle of P'' is shorter than the path cut out in P. For either case, the overflow rate of the direct periodic path between b_l and b_r is bound by $n_l + n_r + \frac{e'}{\delta} + n''$. By Equation 1,

$$n_l + n_r + \frac{e'}{\delta} + n'' \leq \frac{m}{T} + n''$$

$$\frac{(T+\delta)(n_l + n_r + \frac{e'}{\delta})}{T+\delta} \leq \frac{m}{T+\delta} + \frac{\delta(n_l + n_r + (\frac{e}{\delta}))}{T+\delta}$$

$$n_l + n_r + \frac{e'}{\delta} + n'' \leq \frac{m + \delta(n_l+n_r) + e}{T+\delta} + n''$$

Since P is optimal, the direct path must be optimal as well. Therefore, there exists an optimal path that is direct with only U-turns at the extreme points. □

The immediate consequence from the above theorem is that one can find the optimal schedule in $O(n^2)$ time, enumerating all possible pairs of extreme points.

3 Algorithms for the Tree Case

We extend our results to a tree topology, with the sensors placed on a tree network embedded in the plane. Then, we show that the structure of an optimal schedule for the robot is to follow (repeatedly) a simple cycle (a doubling of a subtree). Again we assume that all sensors have the same capacity c and the same rate, 1, of data accumulation. We also assume that the input is integral, i.e., c is an integer and the distance between any two sensors on the tree network is an integer. The proof of the following theorem is omitted from this abstract.

Theorem 2. *Let there be n sensors, p_1,p_2,…,p_n on a tree G. For all p_i, $1 \leq i \leq n$, let $c_i = c$ and $r_i = 1$, i.e. let the capacity and rates of all sensors be the same. There exists an optimal motion of the robot that minimizes data loss with the following properties: (1) the robot only changes direction at sensors, (2) the robot moves in a cycle around a fixed subtree of G.*

A consequence of the above structural result is that we can compute an optimal robot motion (we can identify an optimal subtree of G) in time that is pseudo-polynomial, using a dynamic programming algorithm. We can show that the problem is weakly NP-hard.

References

[1] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 166–174, New York, NY, USA, 2004. ACM.

[2] S. Coene and F. C. R. Spieksma. Profit-based latency problems on the line. *Oper. Res. Lett.*, 36(3):333–337, May 2008.

[3] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1 – 10, 2011.

[4] W. Yu and Z. Liu. Vehicle routing problems with regular objective functions on a path. *Naval Research Logistics (NRL)*, 61(1):34–43, 2014.