A Very Interesting Title

Jared Coleman[1]

[1]Loyola Marymount University

January 14, 2025

Abstract

Abstract goes here.

1 Discrete Case - Start after failure

An agent starts at time 0 at $S = (0, 0)$ carrying an object and is moving in a straight line at speed 1 towards $T = (1, 0)$ to deliver it. At time 1, the agent has not arrived at T. A priori, we know there are n positions, $(a_i, 0)$ for $i = 1, \ldots n$, where the agent may have failed. We assume $0 < a_1 < a_2 < \ldots < a_n < 1$ and the probability that the agent failed at a_i is p_i where $\sum_{i=1}^{n} p_i = 1$. Due to the failure, at time 1, we deploy a second agent (called the *finisher*), starting at position $H = (x, y)$, that can pick up the object and finish the delivery to T. We are interested in minimizing the expected time of delivery of the object by the finisher.

In the following we will use $d(i, j)$ to mean the Euclidean distance between positions a_i and a_j and $d(i, x)$ for $x \in \{S, T, H\}$ to mean the distance between position i and S, T, H, respectively. We define $C[i, j, k]$ to equal the minimum expected delivery time assuming that positions $a_1, \ldots, a_i$ and positions $a_j, \ldots, a_n$ have not been visited, positions $a_{i+1}, \ldots, a_{j-1}$ have been visited (and the object not found), and the finisher is at position k, where $k = i + 1$ or $k = j - 1$. We assume $j > i + 1$, i.e., at least one position between a_i and a_j has been visited. We define $C[0, i, k]$ to be the case where positions $a_1, \ldots, a_{i-1}$ have been visited (and the object not found), positions $a_i, \ldots, a_n$ have not been visited and the finisher is at position $k = 1$ or $k = i$. We define $C[i, n + 1, k]$ to be the case where positions $a_{i+1}, \ldots, a_n$ have been visited (and the object not found), positions $a_1, \ldots, a_i$ have not been visited and the finisher is at position $k = n$ or $k = i + 1$. Define $p_k^{i,j} = \frac{p_i}{1-\sum_{k=i+1}^{j-1} p_k}$ to be the probability the object is at position k given that it is known to not be at positions $i + 1, \ldots, j - 1$. (Note: this includes the case where $i = 0$ and $j = n + 1$.)

Observe that in the minimum expected time trajectory, the finisher must first go directly to the first position i it visits in time $d(H, i)$. If not, a better trajectory can be created by going directly to the first position. Observe further that once the finisher has visited a position on the line, the remainder of its trajectory must stay on the line and visit positions that are immediately to the left or to the right of the most recent position it visited since clearly it makes no sense to skip a position by leaving the line. Thus at any time, the finishers optimal trajectory consists of a contiguous interval of visited positions along with an interval to its left and an interval to its right (either of which could be empty but not both) containing unvisited positions. If the left interval is empty, the optimal trajectory consists of going to the right and visiting all remaining positions (picking up the object along the way) until T is reached. If the interval to the right is empty, the optimal trajectory consists of visiting all positions to the left until the object is found and then taking the object directly to T.

These observations lead to the following recursive formula for the cost of the optimal trajectory:

$$MinExpCost = \min_{i \in [1,n]} \{d(H, i) + C[i - 1, i + 1, i]\},$$

$$C[0, i, k] = d(k, T), 1 \leq i \leq n, k \in \{1, i - 1\}$$

$$C[1, n+1, k] = d(k, 1) + d(1, T), k \in \{2, n\}$$

$$C[i, n+1, k] = d(k, i) + p_i^{i,n+1} * d(i, T) + (1 - p_i^{i,n+1}) * C[i-1, n+1, i], 2 \leq i \leq n-1, k \in \{i+1, n\}$$

$$C[i, j, k] = \min\{d(k, i) + p_i^{i,j} * d(i, T) + (1 - p_s^{i,j}) * C(i-1, j, i), d(k, j) + p_j^{i,j} * d(j, T) + (1 - p_j^{i,j}) * C(i, j+1, j)\}.$$

Using dynamic programming the optimal cost can be calculated in $O(n^2)$ time and the trajectory can be recovered by remembering whether the best choice was to go the right or left at each expansion of the search interval. This leads to

Theorem 1.1. *The minimum expected cost trajectory for recovery and delivery on a line with* n *discrete potential failure points can be computed in* $O(n^2)$ *time.*

DK: Open question 1: points in general position. NP-hard? Approximation?

DK: Open question 2: can above be done using a greedy approach?

DK: Open question 3: what if $p_i = 1/n$? Can you do better than above?

DK: Open question 4: what if $a_i = i/(n+1)$? Can you do better than above?

DK: Open question 5: can you use the answer to question 3 or 4 to approximate an arbitrary continuous distribution on [0,1]?