

Train Line Solution Slides

Problem Boss: Bryce Sandlund
Antonio Molina (y0105w49)
Nick Wu (xiawuc1)
Derek Kisman (SnapDragon)
David Van Brackle (vanb)
Matt Fontaine (Tehqin)
Nalin Bhardwaj
Finn Lidbetter
Rongqi Qiu
Tomas Rokicki
Arnav Sastry (arknave)

Problem

Given n points on a line representing metropolitan areas, determine the maximum utility of placing at most k train stations, where the utility of the stations to a given area is $p \cdot 2^{-d}$, where p is the population of the area and d is the distance from the closest train station to the area.

Consider graphing utility of placing a single station given populations.

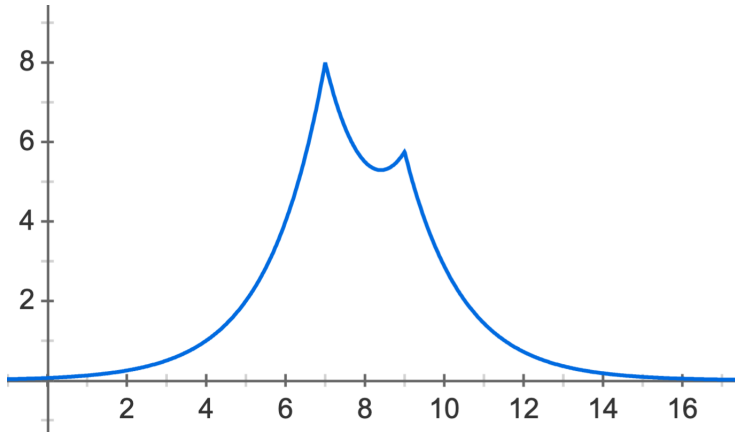


Figure: Graph of $f(x) = 7 \cdot 2^{-|x-7|} + 4 \cdot 2^{-|x-9|}$, the utility of placing a single station at x given a population of size 7 at position 7 and a population of size 4 at position 9.

Two observations can be proved formally:

- 1 It is only ever worth it to place stations directly on populations.
 - Proof: The utility function of a single area between any two populations is convex, thus the sum of the utility functions for all areas between two populations is also convex. Thus it is maximized at the boundary.
 - Alternatively, take derivatives, or apply Jensen's inequality.

Two observations can be proved formally:

- 1 It is only ever worth it to place stations directly on populations.
 - Proof: The utility function of a single area between any two populations is convex, thus the sum of the utility functions for all areas between two populations is also convex. Thus it is maximized at the boundary.
 - Alternatively, take derivatives, or apply Jensen's inequality.
- 2 Utility becomes negligible at a certain distance.
 - With each increasing unit of distance, utility halves, so to achieve a utility f such that $|f - f^*| < \epsilon$, where $0 < \epsilon < 1$ and f^* is the correct answer, we need only consider stations distance $O(\log(1/\epsilon))$ from a given population.
 - To achieve relative error $\leq 10^{-6}$, it can be proven we need only consider stations at distance at most 28 from a given population.

Solution Idea 1

Define:

- $DP(i, j) :=$ maximum utility possible in the first i populations placing at most j stations.
- $Place(i, j) :=$ maximum utility possible in the first i populations placing a station at i and placing at most j stations.

Then:

- $DP(i, j) = \max \left(\begin{array}{l} DP(i-1, j) \\ \max_{\ell=1}^{30} Place(i-\ell, j-1) + utility(i-\ell, i) \end{array} \right).$
- $Place(i, j) = \max_{\ell=1}^{30} DP(i-\ell, j-1) + utility(i-\ell, i).$

Solution Idea 1

Define:

- $DP(i, j) :=$ maximum utility possible in the first i populations placing at most j stations.
- $Place(i, j) :=$ maximum utility possible in the first i populations placing a station at i and placing at most j stations.

Then:

- $DP(i, j) = \max \left(\begin{array}{l} DP(i-1, j) \\ \max_{\ell=1}^{30} Place(i-\ell, j-1) + utility(i-\ell, i) \end{array} \right).$
- $Place(i, j) = \max_{\ell=1}^{30} DP(i-\ell, j-1) + utility(i-\ell, i).$

Pitfall: Forgetting $DP(i-1, j)$ as part of $DP(i, j)$ calculation.

Solution Idea 1

Define:

- $DP(i, j) :=$ maximum utility possible in the first i populations placing at most j stations.
- $Place(i, j) :=$ maximum utility possible in the first i populations placing a station at i and placing at most j stations.

Then:

- $DP(i, j) = \max \left(\begin{array}{l} DP(i-1, j) \\ \max_{\ell=1}^{30} Place(i-\ell, j-1) + utility(i-\ell, i) \end{array} \right)$.
- $Place(i, j) = \max_{\ell=1}^{30} DP(i-\ell, j-1) + utility(i-\ell, i)$.

Pitfall: Forgetting $DP(i-1, j)$ as part of $DP(i, j)$ calculation.

Complexity: $O(kn \log(1/\epsilon))$.

Solution Idea 1

Define:

- $DP(i, j) :=$ maximum utility possible in the first i populations placing at most j stations.
- $Place(i, j) :=$ maximum utility possible in the first i populations placing a station at i and placing at most j stations.

Then:

- $DP(i, j) = \max \left(\begin{array}{l} DP(i-1, j) \\ \max_{\ell=1}^{30} Place(i-\ell, j-1) + utility(i-\ell, i) \end{array} \right)$.
- $Place(i, j) = \max_{\ell=1}^{30} DP(i-\ell, j-1) + utility(i-\ell, i)$.

Pitfall: Forgetting $DP(i-1, j)$ as part of $DP(i, j)$ calculation.

Complexity: $O(kn \log(1/\epsilon))$. **TLE!** We must use the **aliens trick** to reduce the number of DP states.

- Consider $f(k) :=$ maximum utility placing at most k stations.
- Then $f(k)$ is concave: $f(k + 1) - f(k) \leq f(k) - f(k - 1)$.
- Intuitively, with each additional station added, $f(k)$ increases progressively less.

- Consider $f(k) :=$ maximum utility placing at most k stations.
- Then $f(k)$ is concave: $f(k + 1) - f(k) \leq f(k) - f(k - 1)$.
- Intuitively, with each additional station added, $f(k)$ increases progressively less.

Solution Idea 2

Define:

- $DP(i, \lambda) :=$ maximum utility possible in the first i populations where each station costs λ .
- $Place(i, \lambda) :=$ maximum utility possible in the first i populations placing a station at i where each station costs λ .

Then:

- $DP(i, \lambda) = \max \left(\begin{array}{l} DP(i - 1, \lambda) \\ \max_{\ell=1}^{30} Place(i - \ell, \lambda) + utility(i - \ell, i) \end{array} \right)$.
- $Place(i, \lambda) = \max_{\ell=1}^{30} DP(i - \ell, \lambda) + utility(i - \ell, i) - \lambda$.

Final Algorithm

- Binary search to find the correct λ^* so that about k stations are placed.
- Answer is $DP(n, \lambda^*) + \lambda^* \cdot k$.

Final Algorithm

- Binary search to find the correct λ^* so that about k stations are placed.
- Answer is $DP(n, \lambda^*) + \lambda^* \cdot k$.

Pitfall: Using $\text{answer} = DP(n, \lambda^*) + \lambda^* \cdot \text{count}$, where count is the number of stations placed.

Final Algorithm

- Binary search to find the correct λ^* so that about k stations are placed.
- Answer is $DP(n, \lambda^*) + \lambda^* \cdot k$.

Pitfall: Using $\text{answer} = DP(n, \lambda^*) + \lambda^* \cdot \text{count}$, where count is the number of stations placed.

Complexity: $O(n \log(1/\epsilon) \log \text{range})$, where $\log \text{range}$ is the total number of λ 's tried.

Final Algorithm

- Binary search to find the correct λ^* so that about k stations are placed.
- Answer is $DP(n, \lambda^*) + \lambda^* \cdot k$.

Pitfall: Using $\text{answer} = DP(n, \lambda^*) + \lambda^* \cdot \text{count}$, where count is the number of stations placed.

Complexity: $O(n \log(1/\epsilon) \log \text{range})$, where $\log \text{range}$ is the total number of λ 's tried.

Statistics: 70 submissions, 3 accepted.

Questions? Comments? Concerns? Email Bryce Sandlund:
bcsandlund@gmail.com.