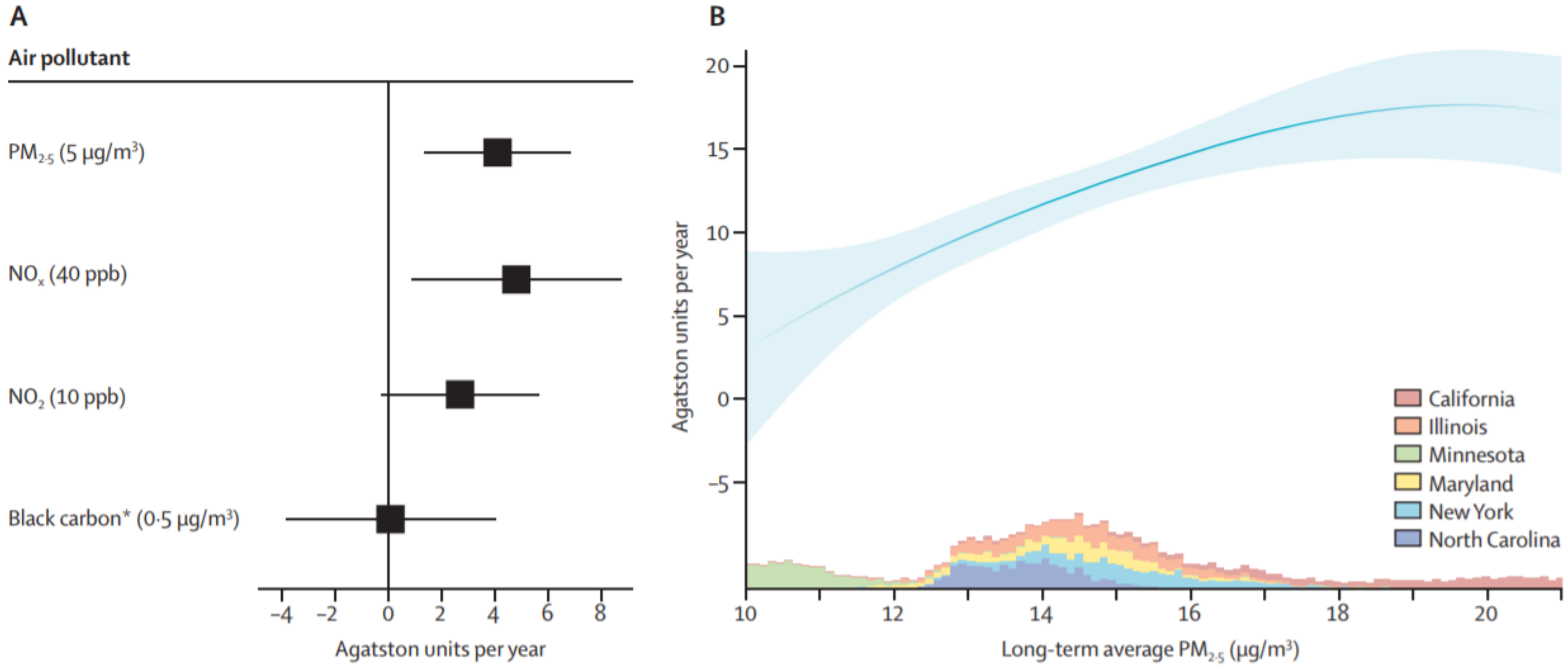
Solving the vehicle routing problem in air quality sampling

Quanyang Lu
05/01/2018

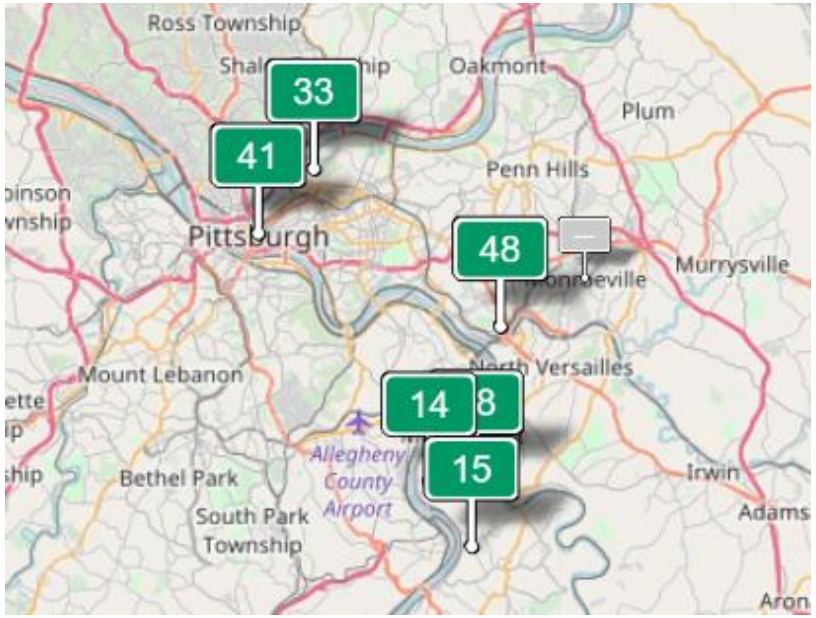
Background

- Air pollution has been proved to link to cardio-vascular diseases and pre-mature death (>3 million per year globally)

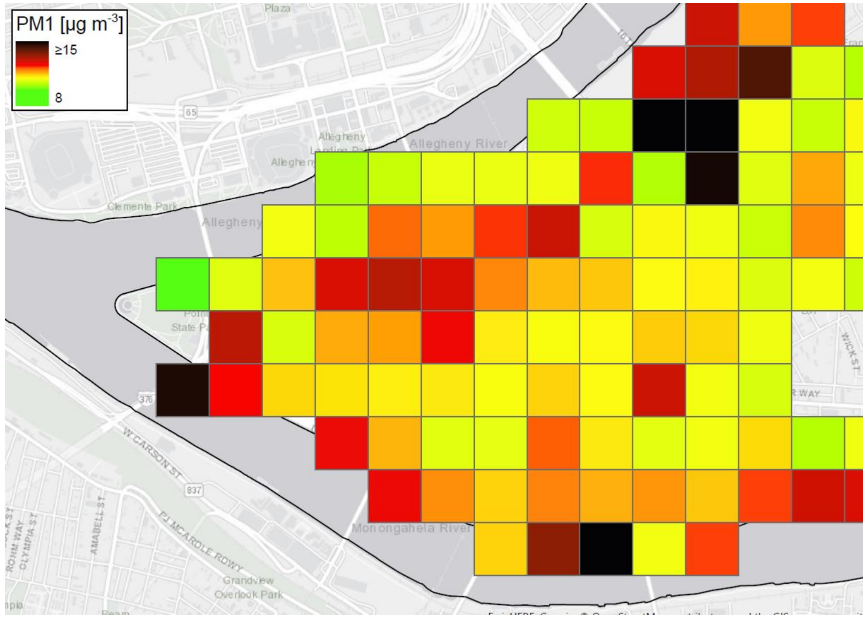


Why mobile sampling?

- Traditional monitoring sites are stationary and sparse
- Mobile sampling are now deployed for high-resolution sampling



04/30/2018, Pittsburgh, PA

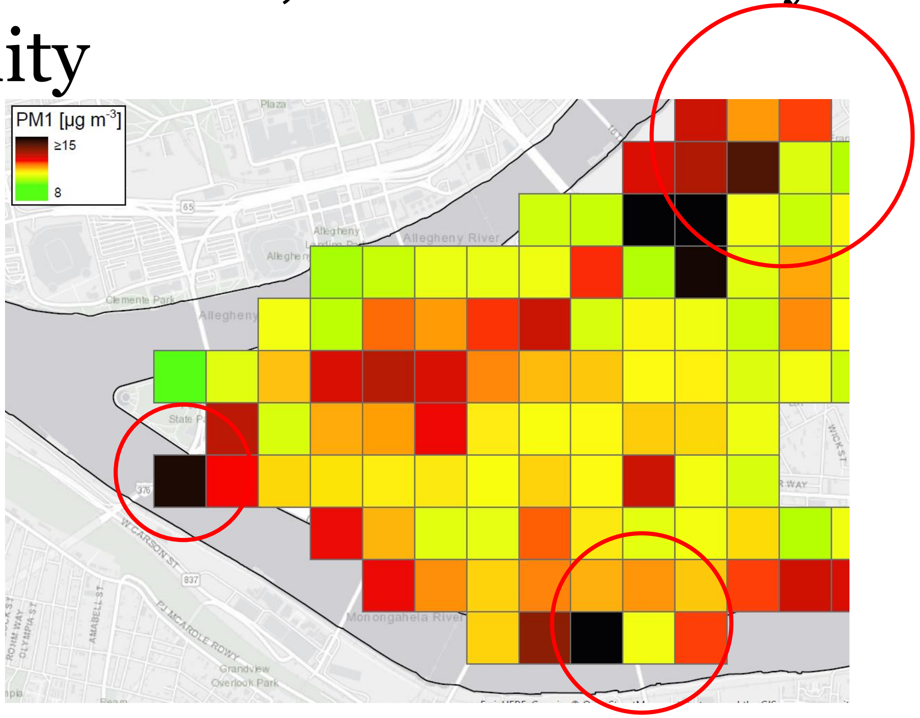


PM_1 concentration of downtown Pittsburgh

Route planning for mobile sampling

- We want to plan the route to maximize information gain
- The driving and sampling hours are limited, we need to stay at each node for 15 mins for data quality

Photo of sampling van of CMU CAPS lab



PM_1 concentration of downtown Pittsburgh

Problem formulation (1)

- Objective: navigate the sampling van to maximize information gain
- The reward decrease with the visited time ***vt***,

$$R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$$

- We assume every nodes has been at least visited once, so that we have some prior knowledge

Problem formulation (2)

- Maximize $\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{N} y_{i,j,t,vti,vtj} = \sum_{k=1}^{N}\sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Problem formulation (2)

Objective:
Rewards collected by all visited nodes at all time t, and all possible visited times vt.

- Maximize $\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N} \sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Problem formulation (2)

- Maximize $\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N} \sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Integer constrain:
0/1 whether visit or not

Problem formulation (2)

- Maximize $\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N} \sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Pipeline constrain:
At each node, sum of all outgoing flow cannot exceed T

Problem formulation (2)

- Maximize $\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N}\sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Single presence constrain:
At each time step, one and only one node could be visited.

Problem formulation (2)

- Maximize $\sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N} \sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N} \sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Heraclitus constrain: One cannot visit a node twice with same vt

Problem formulation (2)

- Maximize $\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{vt=1}^{T} R_{i,t,vt}\, x_{i,t,vt}$, subject to:
  - $x_{i,t,vt} \in \{0,1\}$ (1)
  - $y_{i,j,t,vt,vtj} \leq x_{i,t,vt} * T$ (2)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{T} x_{i,t,vt} = 1$ (3)
  - $\sum_{t=1}^{T} x_{i,t,vt} \leq 1$ (4)
  - $\sum_{i=1}^{N}\sum_{vt=1}^{N} y_{i,j,t,vtj,vtk} = \sum_{k=1}^{N}\sum_{vt=1}^{N} y_{j,k,t+1,vtj,vtk} + x_{j,t,vt}$ (5)
- Given $R_{i,t,vt} = R_{i,t,1} * \frac{1}{vt}$

Flow conserve constrain: at each node, the incoming flow = 0/1 (visited or not) + outgoing flow

Solving with MATLAB

- Basically, we are solving a MILP, with dimension of (N^2T^3)
- However, MATLAB does not scale the problem very well, ending up getting 20000*20000 matrix (memory out).

linprog

Solve linear programming problems

Linear programming solver

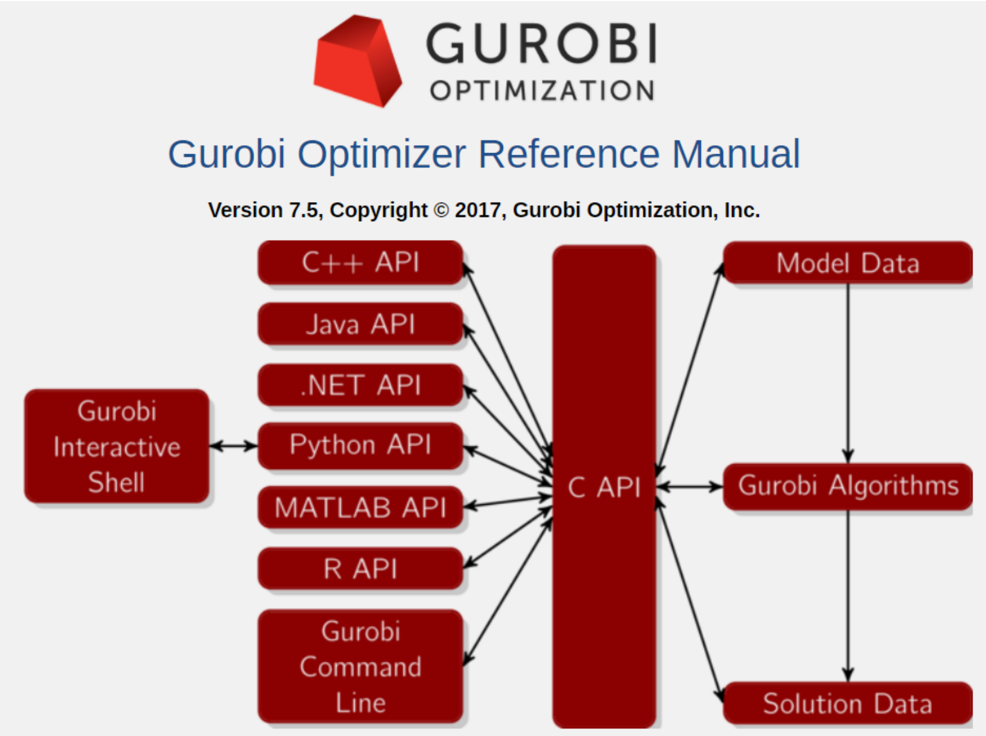
Finds the minimum of a problem specified by

$$\min_{x} f^{T}x \text{ such that } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub. \end{cases}$$

f, x, b, beq, lb, and ub are vectors, and A and Aeq are matrices.

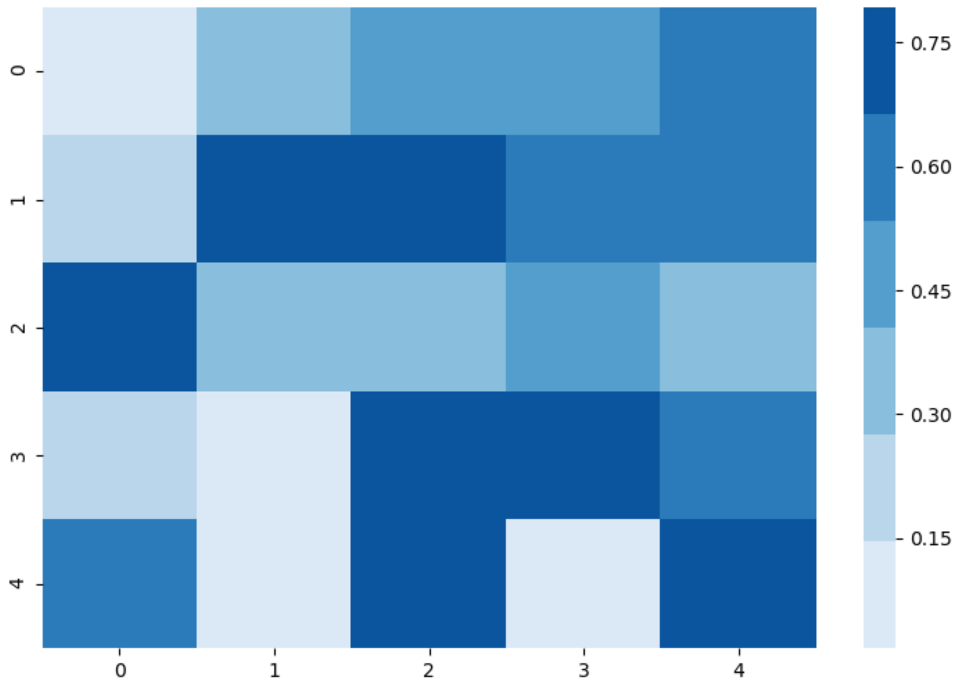
Switching to Gurobi

- Professional large-scale optimization problem solver
- Able to handle problems of millions of variables



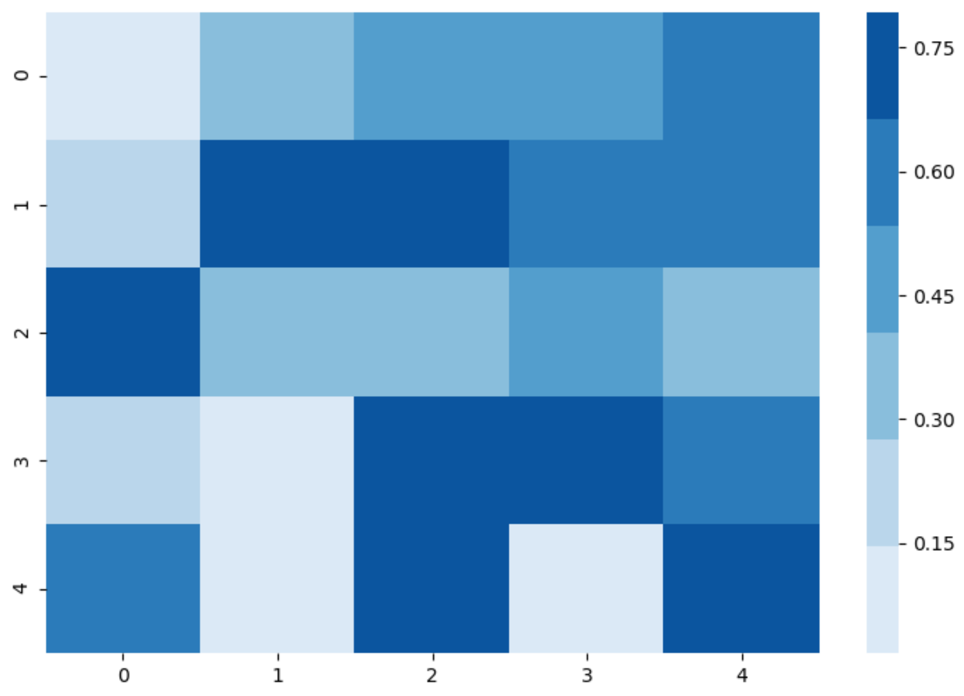
Sampling downtown area with 5*5 blocks

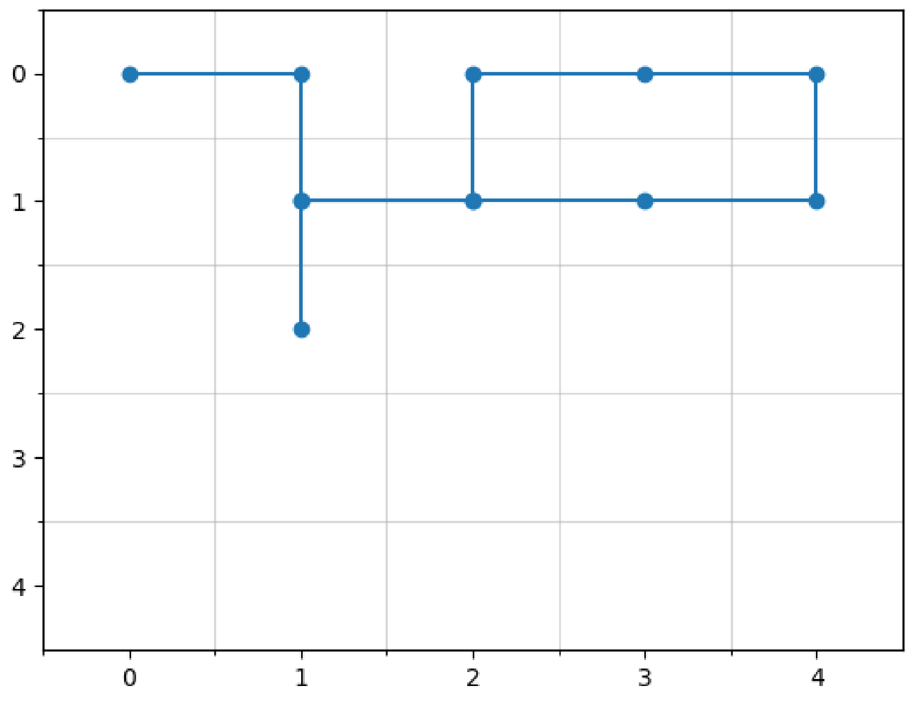
- Consider a 3-hour sampling period (N = 25, T = 12)
- Prior knowledge expressed in ranging in (0,1)



Greedy driving schedule

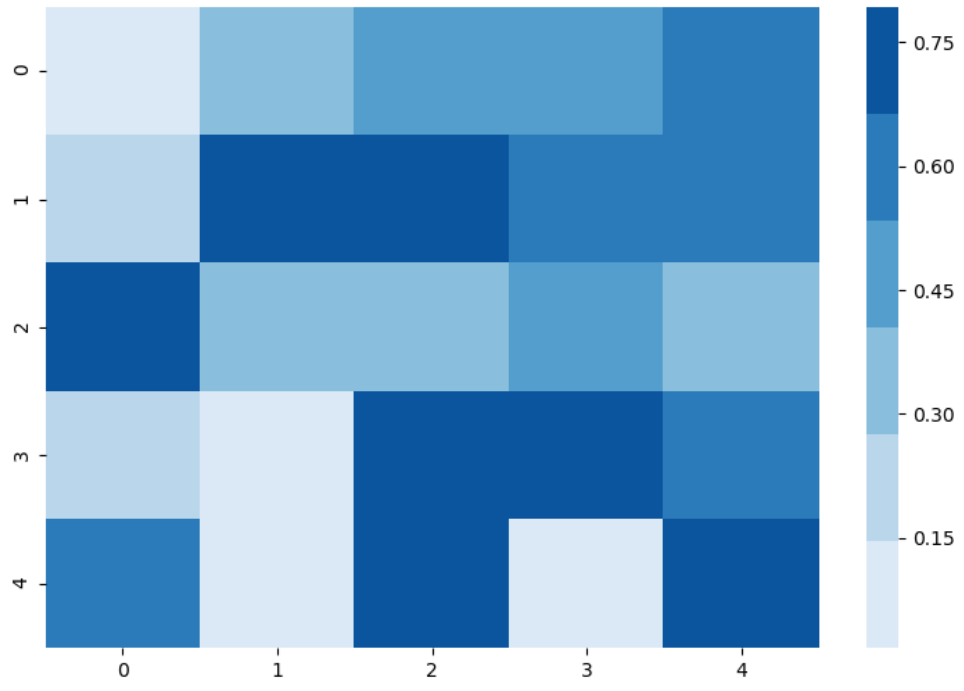
- Always go to the neighboring node with highest reward
- Total reward = 5.596

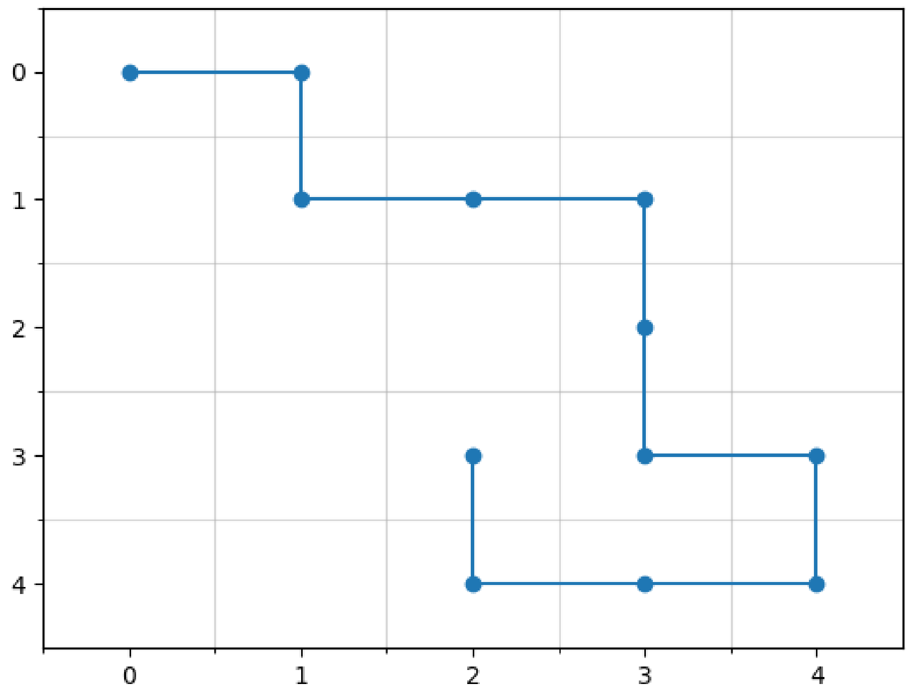




Optimal driving schedule

- Number of variables: N^2T^3 = 1,080,000 (computing time: 30 min)
- Total reward: 6.80663 (21.6% increase)





Conclusion

- Our algorithm provides an exact solution to the discounted-reward vehicle routing problem
- The problem is NP-hard and its size scale to N^2T^3, which is very time-consuming to solve
- Approximation is needed, such as
  - Each node could be visited at most 3 times
  - The dimension of this problem reduce to $9N^2 T$

Acknowledgement

- The author would like to thank Prof. Fei Fang for her advisement and helpful discussion on this course project.
- The author would like to thank his academic advisor Prof. Allen Robinson for support on taking this course.
- The author would also like to thank Zhongju Li and Peishi Gu for the sharing of their PM_1 map data.