

Problem 1

a)

- i) Since, the cost of transport is directly proportional to Euclidean distance between the factories, the weights should be equal to each other. For simplicity, let them all be 1.

$$w_i = w = 1$$

- ii) Defining Notation:

Anchor Points $\bar{a}_1, \dots, \bar{a}_{\cancel{4}}$ Positions of factories

Weight $w_i = w = 1$

Minimal point $\bar{x} \in \mathbb{R}^2$

\bar{x} is the optimal point for building the assembly line.

Minimization Problem:

$$\min_{\bar{x} \in \mathbb{R}^2} \left\{ f(\bar{x}) = \sum_{i=1}^4 \|\bar{x} - \bar{a}_i\|_2 \right\}$$

We are minimizing the Euclidean distance from the assembly line to the factories (anchor points)

Let $\bar{x} = (x, y)$

let $f = f_1 + f_2 + f_3 + f_4$

$$\min_{\bar{x} \in \mathbb{R}^2} \left\{ \sum_{i=1}^4 \|\bar{x} - \bar{a}_i\|_2 \right\} =$$

Note that
 \bar{x} is assumed
 to be not an
 anchor point

$$\min_{\bar{x} \in \mathbb{R}^2} \left\{ \begin{array}{l} f_1 \\ f_2 \\ f_3 \\ f_4 \end{array} \right\} = \left\{ \begin{array}{l} \sqrt{(x-1)^2 + (y-1)^2} + \sqrt{(x-1)^2 + (y-3)^2} \\ + \sqrt{(x-2)^2 + (y-5)^2} + \sqrt{(x-3)^2 + (y-1)^2} \end{array} \right\}$$

To find the minimal point we need

$$\text{to take } \nabla f = \bar{0} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial x} + \dots \\ \frac{\partial f_1}{\partial y} + \dots + \frac{\partial f_4}{\partial y} \end{bmatrix}$$

$$\frac{\partial f_1}{\partial x} = \frac{x-1}{\sqrt{(x-1)^2 + (y-1)^2}} \quad \frac{\partial f_2}{\partial x} = \frac{x-1}{\sqrt{(x-1)^2 + (y-3)^2}}$$

$$\frac{\partial f_3}{\partial x} = \frac{x-2}{\sqrt{(x-2)^2 + (y-5)^2}} \quad \frac{\partial f_4}{\partial x} = \frac{x-3}{\sqrt{(x-3)^2 + (y-1)^2}}$$

$$\frac{\partial f_1}{\partial y} = \frac{y-1}{\sqrt{(x-1)^2 + (y-1)^2}} \quad \frac{\partial f_2}{\partial y} = \frac{y-3}{\sqrt{(x-1)^2 + (y-3)^2}}$$

$$\frac{\partial f_3}{\partial y} = \frac{y-5}{\sqrt{(x-2)^2 + (y-5)^2}} \quad \frac{\partial f_4}{\partial y} = \frac{y-1}{\sqrt{(x-3)^2 + (y-1)^2}}$$

~~using the fact that \bar{x} .~~

using the fact that \bar{x} is the point such that $\|\bar{x} - \bar{a}_i\|_2$ is minimal.

$$\sum_{i=1}^n \frac{\|\bar{x} - \bar{a}_i\|_2}{\|\bar{x} - \bar{a}_i\|} = 0$$

using the fact that $\|\bar{x}\|_2 = \frac{\|\bar{x}\|}{\|\bar{x}\|_2}$

$$\left(\sum_{i=1}^n \frac{1}{\|\bar{x} - \bar{a}_i\|} \right) \bar{x} = \sum_{i=1}^n \frac{\bar{a}_i}{\|\bar{x} - \bar{a}_i\|}$$

$$\bar{x} = \frac{1}{\sum_{i=1}^n \frac{1}{\|\bar{x} - \bar{a}_i\|}} \sum_{i=1}^n \frac{\bar{a}_i}{\|\bar{x} - \bar{a}_i\|}$$

~~Rewriting~~ Rewriting $\bar{x} = T(\bar{x})$

$$T(\bar{x}) = \frac{1}{\sum_{i=1}^n \frac{1}{\|\bar{x} - \bar{a}_i\|}} \sum_{i=1}^n \frac{\bar{a}_i}{\|\bar{x} - \bar{a}_i\|}$$

Finding the minimal point \bar{x} is equivalent to solving the fixed point iteration problem

$$\bar{x}^{k+1} = T(\bar{x}^k) \quad \text{Weiszfeld Method.}$$

iii)

$\bar{x}^{k+1} = \tau(\bar{x}^k)$ can also

be written as:

$$\bar{x}^{k+1} = \bar{x}^k - \frac{1}{\sum_{i=1}^4 \frac{1}{\|\bar{x}^k - \bar{a}_i\|}} \sum_{i=1}^4 \frac{\bar{x}^k - \bar{a}_i}{\|\bar{x}^k - \bar{a}_i\|}$$

Looks like gradient descent. It has a unique solution iff $f(x)$ has strong convexity.

The 1st and 2nd order conditions for convexity:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{1st Order}$$

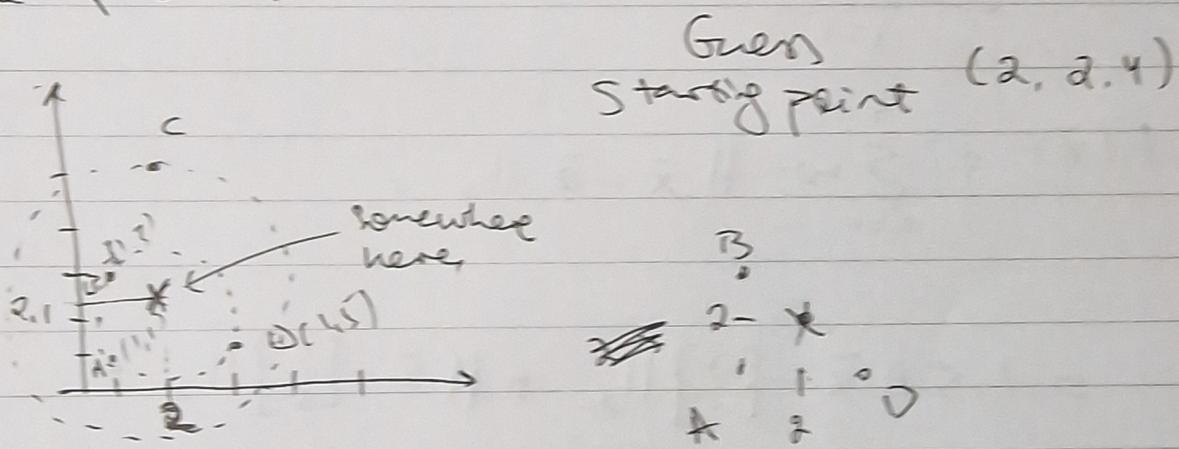
Hessian of f is positive semidefinite
for $\forall \bar{x}$ 2nd order

$$\bar{x}^T \nabla^2 f(x) \bar{x} \geq 0$$

iv) $[1.40000548, 2.59999102]$, 50 iterations

v) No, \bar{x} is assumed to be not equal to the anchor points

vi) Choose a starting point in the middle of one factories:

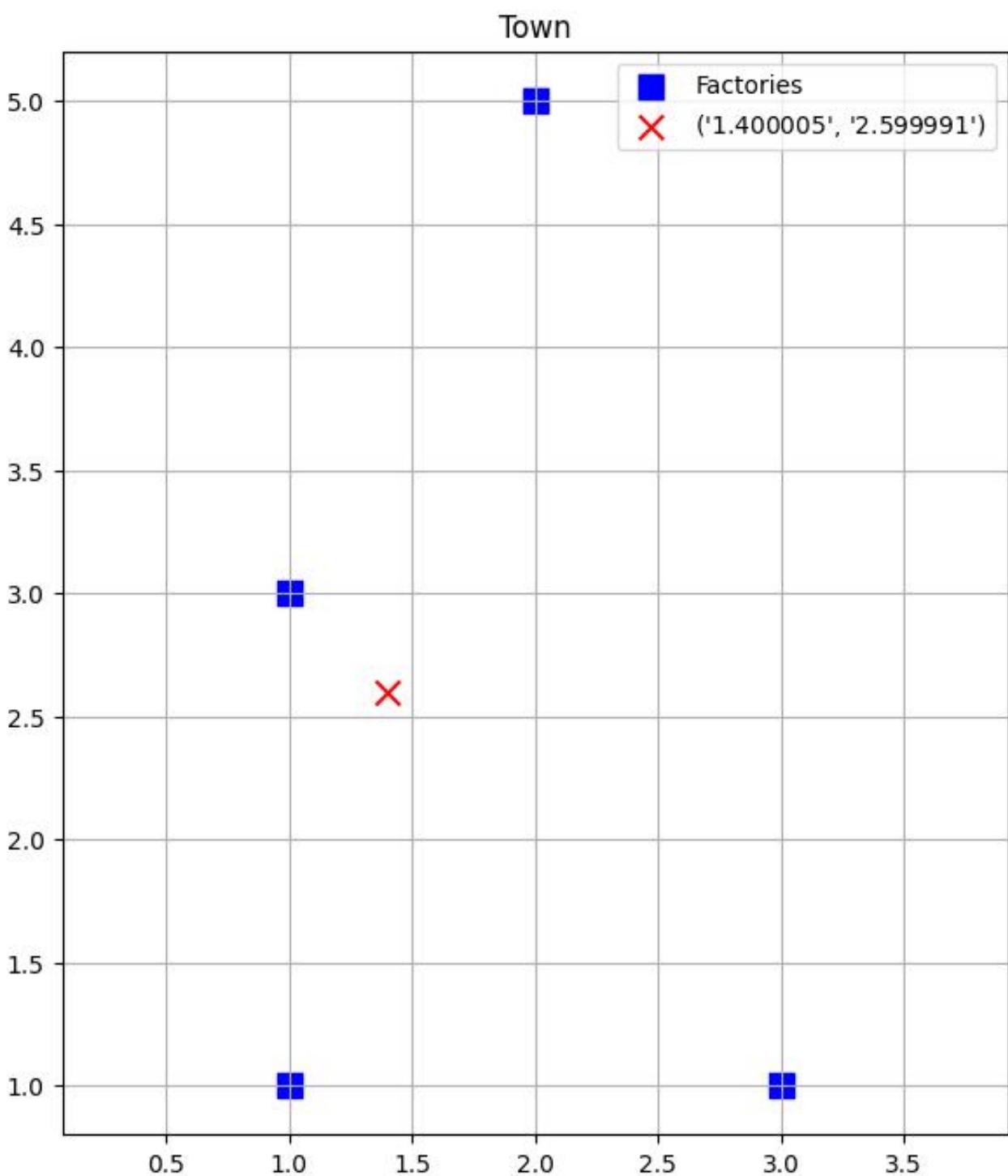


Plugging into ~~the~~ the algorithm:

$(2, 2.4)$ (and ~ 47 iterations)

$(2, 2)$ ~ 49

Original ~ 50 iterations



b)

The new weight:

$$w_1 = 1.5, w_2 = w_3 = w_4 = 1.0$$

$$h(\bar{x}) = \sum_{i=1}^4 w_i \cdot \text{distance}$$

$$h(\bar{x}) = \sum_{i=1}^4 w_i \|\bar{x} - \bar{a}_i\|_2 =$$

$$= 1.5 \|\bar{x} - [1] \|_2 + \|\bar{x} - [3] \|_2 +$$

$$+ \|\bar{x} - [2] \|_2 + \|\bar{x} - [1] \|_2$$

The Algorithm: Weiszfeld Method

```
def Weiszfeld(x_k, w, a):
```

```
grad_k, summa = grad_h(x_k, w, a)
```

```
num = 0
```

```
while np.linalg.norm(grad_k) > 1e-5: # stopping criterion
```

```
grad_k, summa_k = grad_h(x_k, w, a)
```

```
x_k += (-1 / summa_k) * grad_k
```

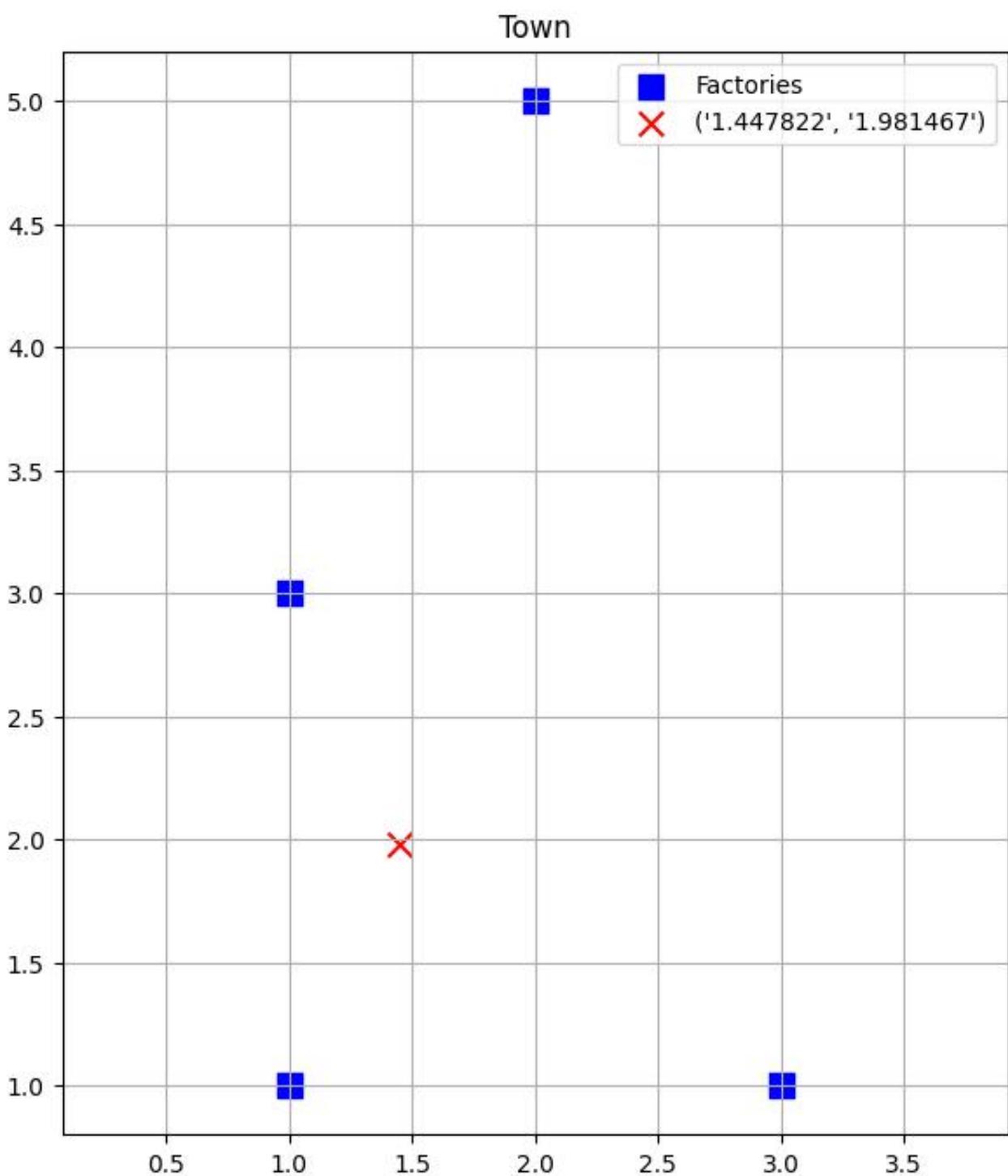
```
num += 1
```

```
return x_k, num
```

where grad_h is the function of ∇h .

This gives

$[1.44782162, 1.98146687]$, 37 iterations



2.

$$\frac{\partial f}{\partial x} = -200(w' - x) + 20.2(x - 1) + 19.8(z - 1)$$

$$\frac{\partial f}{\partial y} = 2(y - 1) + 360y(y' - z)$$

$$\frac{\partial f}{\partial z} = -180(y^2 - z) + 20.2(z - 1) + 19.8(x - 1)$$

$$\frac{\partial f}{\partial w} = 400w(w^2 - x) + 2(w - 1)$$

$$\nabla^2 f = \begin{pmatrix} 220.2 & 0 & 19.8 & -400w \\ 0 & 2y + 1080y^2 - 360z & -360z & 0 \\ 19.8 & -360y & 200.2 & 0 \\ -400w & 0 & 0 & 1200w^2 - 400x^2 \end{pmatrix}$$

Putting this into python (NEXT PAGE)

Solution

a) [1, 1, 1, 1]

b) 12 iterations

```

1 import numpy as np
2
3
4 # Defining The hessian
5 def Hessian(x,y,z,w): *Output: 4x4 np array
6 :
7
8 # defining The grad
9 def grad_f(x,y,z,w): * Output 4d np array
10 :
11
12 # defining d
13 def d_k(x,y,z,w):
14     grad = grad_f(x,y,z,w)
15     Hess = Hessian(x,y,z,w)
16     return np.linalg.solve(Hess, -grad), grad
17
18
19
20 # Pure Newton Algorithm
21 def Pure_Newton(x,y,z,w):
22     xk = np.array([x,y,z,w])
23     d, gradf = d_k(xk[0], xk[1], xk[2], xk[3])
24     num = 0
25     while np.linalg.norm(gradf) > 1e-5:
26         num += 1
27         xk += d
28         d, gradf = d_k(xk[0], xk[1], xk[2], xk[3])
29
30
31
32     return xk, num
33
34
35
36
37
38
39
40
41
42 print(Pure_Newton(0.0, 1.0, 2.0, 3.0))

```