ISSN 2959-6157

Application of Gradient Descent Optimization in Optimal Location Selection for Fire Stations in Newly Built Cities

Junjie Wu[1,*,&], Yiyang Zhang[2,&], Yuqing Zhuang[3,&], Zhuoer Huang[4,&], Ziyu Wang[5,&]

[1]International Department Jining Confucius School Jining, 272000 China, franciswujj@gmail.com
[2]BASIS International School Hangzhou Hangzhou 310016 China, flyfish7hg@gmail.com
[3]International Department, Quanzhou NO.5 Middle school Quanzhou 362002 China, 3459034574@qq.com
[4]Wuhan Britain-China School, Wuhan, 430000 China, 1707969728@qq.com
[5]Chongqing Bashu Ivy International School, Chongqing, 401147 China, franciswujj@gmail.com
[&]These authors contributed equally to this work and should be considered co-first authors.

Abstract:

Fires have been one of the most common disasters threatening human lives. The location of a fire station is critical to firefighting and saving people as well as properties. This paper discusses the situation of finding an optimal location for a fire station in a newly built city. In addition, this paper aims to explore the applications of gradient descent optimization in practical problems. Two gradient descent algorithms are used to minimize the average response time for a fire call. The model for the average response time is built step by step. The first part considers the fire frequency distribution in the area only. The second part adds a weight to each box based on the population and wealth distribution. In the third part, the spread of fires is taken into consideration, and we use the penalty method to convert this constrained problem into an unconstrained one.

Keywords: Gradient descent, Penalty method, Optimal location.

1 Introduction

Gradient Descent is one of the most commonly used methods for solving unconstrained optimization problems. It can be used to solve model parameters in machine learning algorithms by minimizing the loss function. It involves repeatedly moving a point on a curve in the opposite direction to the gradient at that point for a chosen step size. Conversely, if we need to solve for the maximum value of a function, then we can add a negative sign to the function and try to minimize the new function. In machine learning, two gradient descent methods have been developed based on basic gradient descent methods, namely the random gradient descent method and the batch gradient descent method.

Gradient Descent Optimization (GDO), also known as the steepest descent method, was proposed by French mathematician Augustine Louis Cauchy in 1847. It is one of the most classic and simple first-order methods in optimization methods. In the first decade of the 20th century, optimization algorithms such as stochastic gradient descent showed amazing performance for large-scale problems. Specifically, second-order stochastic gradient and averaged stochastic gradient are asymptotically efficient after a single pass on the training set [1]. A simple warm restart technique for stochastic gradient descent is proposed to improve its anytime performance when training deep neural networks in 2016 [2]. In the research in 2020, stochastic gradient descent was applied with a linear regression algorithm in a machine-learning environment for predicting biomass higher heating value [3].

The purpose of this paper is to discuss the real-world applications of optimization algorithms and apply gradient descent to a real-world problem. Considering the hazards of fire accidents, which often happen without warning, our group think of the situation where city planners plan to build a fire station in a newly built city. This paper introduces the gradient descent algorithm as a numerical method to determine the optimal location of the fire station in order to minimize the average time for fire trucks to travel from the fire station to a fire location.

2 Unweighted

2.1 Dataset

To find out the optimal location for a fire station, our group created a 12*12 map for a virtual city based on the layout of a district in Beijing and a fire distribution chart considering seven types of land. As shown in Figure 1, in the map, r stands for residential areas, i for industrial areas, w for woods, f for farmlands, c for crowded areas (commercial areas and transport), b for landfill sites, and o for others.

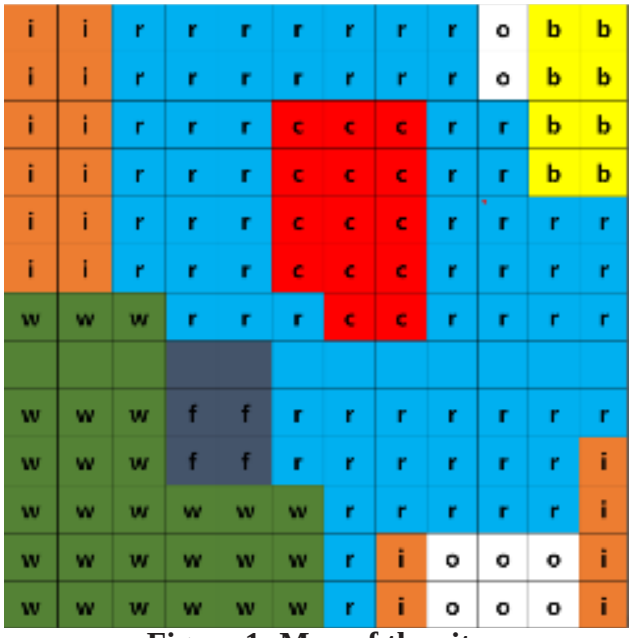

Figure 1: Map of the city

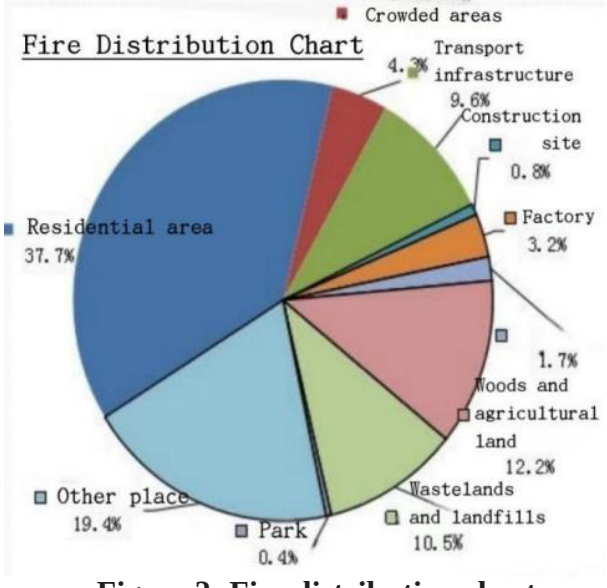

Figure 2: Fire distribution chart

Then, we generate the fire frequency distribution in the city. We assume that the total number of fires in the region per year is 300. According to the proportion of fires (see Figure 2), happening in different places, we calculated the number of fires in each type of land. After this, seven randomized attempts are used to determine the count of fire frequency for each grid based on regions by selecting random grids. The total number selected is the fire frequency. Figure 3 shows the final results of this portion.

0	3	2	0	0	1	2	2	3	2	2	4
0	1	4	2	1	2	3	1	2	0	6	0
1	1	3	1	2	6	1	2	2	1	2	0
0	1	2	3	1	3	2	4	3	3	3	0
2	1	4	1	2	4	5	1	2	2	7	2
1	2	1	4	3	3	2	5	4	2	1	1
2	0	0	2	4	1	0	0	2	3	1	3
1	1	1	2	2	4	6	2	1	3	4	0
1	1	0	0	1	2	1	4	6	2	1	2
0	3	0	0	1	0	4	0	5	3	6	2

Figure 3: Fire frequency distribution

2.2 Modelling

To optimize the position of the fire station, the total time for the fire station to reach each grid should be minimized. We assume the speed of the fire trucks is constant.

The function to calculate the response time of a fire truck for a fire is given by

$$T = 3.2 + 1.7r^{0.91},$$

where T is the response time and r is the distance between the station and the fire.[4] This formula can be understood intuitively. 3.2 is the time the fire truck gets ready to set out. 1 / 1.7 is the speed of the fire truck, and the exponent 0.91 indicates that the distance is nearly a linear function of time.

To simplify calculations, we divide the 12*!2 grid into 36 2*2 smaller grids. We sum up the number of fires in each smaller grid and calculate the distance from the center of each grid to the fire station. The origin (0, 0) is at the bottom left corner.

Therefore, the average response time for a fire in the area is

$$T = 3.2 + 1.7\frac{\sum_{i=1,j=1}^{i=12,j=12} n_i \times \sqrt{\left(x - x_i^2\right)^2 + \left(y - y_i^2\right)^2}^{0.91}}{s},$$

where n_i is the number of fires in each smaller grid, (x, y) is the position of the fire station, (x_i, y_i) is the coordinate of the center of a 2*2 grid and s is the total number of fires in the whole city. Note that this function is a strong

convex function. In the next two sections, we will use two gradient descent algorithms to minimize this function.

2.3 Solving the model using gradient descent with momentum

This model uses momentum and self-adjusted learning rate to calculate the optimal location of the fire station using gradient descent. The gradient is calculated as

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right).$$

Then,

$$\frac{\partial f}{\partial x} = \frac{f(x+h, y) - f(x, y)}{h}.$$

The same also goes for $\frac{\partial f}{\partial y}$. To use momentum, a velocity vector is added to keep track of momentum of the "object". The learning rate also changes according to previous gradients in the algorithm. The overall function for the algorithm is:

$$vx_{i+1} = \frac{\alpha}{\sqrt{k_x} + ?} \times \frac{\partial f}{\partial x} + \rho v x_i,$$

$$vy_{i+1} = \frac{\alpha}{\sqrt{k_y} + ?} \times \frac{\partial f}{\partial y} + \rho v y_i,$$

where $x_{i+1} = x_i - vx$, $y_{i+1} = y_i - vy$, $k_x = \sum_{i=0}^{n} \frac{\partial f}{\partial x}^2$ and $k_y = \sum_{i=0}^{n} \frac{\partial f}{\partial y}^2$.

Rho is 0.9 in the program and epsilon is 10^{-8}. The learning rate is changed by dividing the initial learning rate with the accumulated sum of previous gradients' square value. The algorithm uses iterations to find the optimal solution, and the iteration ends when the $f(x, y)$ value changes by less than 10^{-8} compared to the previous one. The use of momentum can bring advantages to the function by making the function converge faster as the function has a previously increased amount of change as velocity. However, this algorithm is more suited to functions that have a long semi-major axis. The use of learning rate adjustment can also make the function converge faster since the learning rate determines how far and fast the current position goes.

2.4 Solving the model using gradient descent

This method uses self-adjusted and solved step-length to make convergence to solution faster. To calculate the best step length, during every iteration, we differentiate $f(x - m\frac{\partial f}{\partial x}, y - m\frac{\partial f}{\partial y})$, where m is the step length, with respect to m:

$$h = \frac{d}{dm} f\left(x - m\frac{\partial f}{\partial x}, y - m\frac{\partial f}{\partial y}\right).$$

We let h equal zero and solve the equation using python with an initial guess of 0.4. The algorithm then finds a numerical solution for the variable m that satisfies the equation represented by h.

$$m = float(nsolve(h, m, 0.4, verify = False)).$$

The iterations end when the magnitude of the gradient is small enough, e.g., $m < 10^{-9}$. Overall, this algorithm is also aimed at making the function converge faster to the optimal solution. The algorithm's use of numerical solve can decrease the iterations required to reach the solution, sometimes smaller than the first model. Both models give an answer (6.906,6.238) for the coordinate of the fire station.

3 Weighted

In this part of the problem, we consider another two factors critical to the location of a fire station, population and wealth distributions. For an area with high density of population and wealth, we multiply the fire frequency by a large weight, meaning that we give higher priority for that area.

We generate the weight distribution diagram in the following steps:

Step 1: Rate the population and wealth densities of each box on a scale of 1-5 and 1-6, respectively, based on the relative position of the box in the whole area. For both population and wealth, 1 indicates the highest density, as shown by Figure 4 and Figure 5.

	1	2	3	4	5	6	7	8	9	10	11	12
1	5	4	4	4	4	4	4	4	5	5		
2	4	3	4	4	3	3	3	3	3	4		
3	4	4	4	4	3	2	2	2	3	4		
4	5	4	4	4	3	1	1	1	3	4		
5	4	3	3	4	3	1	1	1	3	4	4	5
6	5	5	5	5	3	1	1	1	3	4	4	5
7				5	3	2	2	2	3	4	4	5
8				4	3	3	3	3	3	4	3	4
9				5	5	5	4	4	4	4	4	4
10							5	3	4	4	4	5
11							5	4				5
12							5	4				5

Figure 4: Wealth density ratings

	1	2	3	4	5	6	7	8	9	10	11	12
1	4	3	6	5	5	5	5	5	5	5		
2	4	3	6	5	4	4	4	4	4	5		
3	3	3	6	5	4	3	3	3	4	5		
4	4	4	6	5	4	3	2	3	4	5		
5	3	3	6	5	4	3	1	3	4	5	6	6
6	5	5	5	4	4	3	2	3	4	5	6	6
7				4	4	4	3	3	4	5	6	6
8				3	4	4	4	4	4	5	6	6
9				4	5	5	5	5	5	5	6	3
10							4	6	6	6	6	3
11							5	2	6	6	6	4
12							5	2	6	6	6	4

Figure 5: Population density ratings

Step 2: Use a computer to randomly generate the corresponding population and wealth density of each box according to the following standards (Table 1):

Table 1: population and wealth densities standard

Population scale	Population density /1000 persons	Wealth scale	Wealth density /billion RMB
1	9-11	1	9-11
2	7-9	2	7-9
3	4-7	3	5-7
4	2-4	4	3-5
5	0-2	5	1.5-3
/	/	6	0-1.5

Step 3: Add the population and wealth densities of each box together, and multiply the sum by 1000, as shown by Table 2.

Table 2: sums of population and wealth densities

524	290	269	496	532	402	480	558	264	350	0	0
805	550	534	515	772	708	837	861	1037	466	0	0
360	320	281	536	932	1413	1300	1415	908	553	0	0
180	370	436	517	817	1641	1800	1614	946	586	0	0
230	440	514	393	838	1643	2115	1550	729	605	340	236
170	190	30	40	882	1571	1918	1597	913	555	296	251
0	0	0	40	1025	1238	1440	1332	933	417	451	140
0	0	0	360	430	979	879	775	947	426	672	474
0	0	0	40	0	349	473	409	528	461	372	370
0	0	0	0	0	0	120	650	457	353	361	180
0	0	0	0	0	0	110	1074	0	0	0	180
0	0	0	0	0	0	120	1061	0	0	0	100

Step 4: Notice that we set both the population and property densities of boxes in the woods to zero, as there is generally few people and wealth in the woods. However, fires are likely to spread from the edge of the forest to the surrounding farmlands or industrial areas. We divide the forest into three layers, the outermost layer, the middle layer and the innermost layer, as shown in Figure 6.

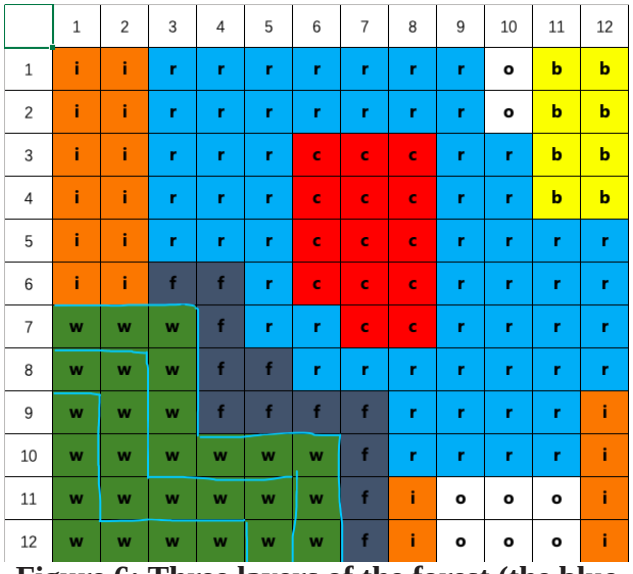

Figure 6: Three layers of the forest (the blue lines)

We assume that if fires burn in the outermost or middle layer, then they are bound to spread to the surrounding agricultural or industrial land. However, if fires burn in the innermost layer, it is likely that firefighters can prevent them from affecting the surroundings if they respond promptly enough. Therefore, for boxes in the outermost and middle layers of the forest, we add the population and wealth densities of the eight boxes around them. But for boxes in the innermost layer, we keep the population and wealth densities zero.

Step 5: Generate the weight distribution by comparing the sum in each box with the following standards (Table 3):

Table 3: Weight standards

Sum	Weight
> 1500	3.0
1200-1500	2.6
800-1200	2.2
500-800	1.8
200-500	1.4
< 200	1.0

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.8	1.4	1.4	1.8	1.4	1.8	1.4	1.4	1.4	1.4	1.0	1.0
2	1.8	1.8	1.8	1.8	1.8	2.2	2.2	2.2	1.8	1.4	1.0	1.0
3	1.4	1.4	1.4	1.4	2.2	2.6	2.6	2.6	2.2	1.8	1.0	1.0
4	1.0	1.4	1.4	1.4	2.2	3.0	3.0	3.0	2.2	1.4	1.0	1.0
5	1.4	1.4	1.8	1.8	2.2	2.6	3.0	3.0	2.2	1.8	1.4	1.0
6	1.0	1.0	1.0	1.4	2.2	2.6	3.0	3.0	2.2	1.8	1.4	1.4
7	1.0	1.4	2.2	1.0	2.2	2.2	2.6	2.6	2.2	1.8	1.4	1.0
8	1.4	2.2	1.8	1.4	1.8	2.2	2.2	2.2	2.2	1.4	1.4	1.4
9	1.0	1.8	1.4	1.0	1.0	1.4	1.8	1.8	1.4	1.4	1.4	1.4
10	1.0	1.4	1.0	1.4	1.4	2.2	1.0	1.8	1.4	1.4	1.4	1.4
11	1.0	1.0	1.4	1.4	2.6	1.4	1.0	2.2	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.4	1.4	1.0	2.2	1.0	1.0	1.0	1.0

Figure 7: Weight distribution

After creating the map of weight distribution (see Figure 7), we multiply the fire frequency in each box by the weight of that box. We still aim to minimize the time function

$$T = 3.2 + 1.7\frac{\sum_{i=1,j=1}^{i=12,j=12} n_i \times \sqrt{\left(x - x_i^2\right)^2 + \left(y - y_i^2\right)^2}^{0.91}}{s}.$$

and the only variable that has been changed is n_i. The optimal position we get from our program is (6.994, 6.997). This is close to the result in Part 1, which is probably because in Part 1 we focus on fire frequencies and in Part 2 we take population and wealth densities into account. Both the fire frequency and the population and wealth densities are the highest in the crowded area, so out fire station should be built around there.

4 Constrained

4.1 Constraint added and justifications

Now, we take the spread of fire into consideration. Given the speed of the fire, the fire trucks need to arrive at the fire area before the fire reaches its maximum coverage capacity, which we assume to be 12 square kilometers. After calculation, it is known that the fire station needs to be built within 6 kilometers of where the fire starts. In order to simplify the model, we assume that the fire occurs in the box (1,1).

Now, from that assumption, we put the constraints in the form of an algebraic expression,

$$(x-1)^2 + (y-1)^2 \leqslant 6^2,$$

into our unconstrained optimal fire station location, and we find that the results of Parts 1 and 2 do not satisfy our inequality. Now, solving this inequality without the help of the penalty function is undoubtedly difficult, but given that the three-dimensional function is strongly convex, we can think of the inequality constraint as an equality constraint, and the simple proof is as follows, on this function, the value of the function will gradually increase from the lowest point to the point spreading in the direction of any x-y plane. This means that we can convert the inequality conditions to the equality restriction

$$(x-1)^2 + (y-1)^2 = 6^2,$$

The literal meaning of this is that under the constraint conditions, the optimal location must satisfy its equality constraints.

The simplified problem will be much simpler than before so that it can still be solved with the ordinary optimization algorithm with the aid of the no-penalty function. The following will first introduce the steps and ideas of our optimization algorithm in general, and then explain the specific operating steps and principles in detail.

4.2 Solution to Part 3

4.2.1 An original method

Now we observe the ordinary gradient descent algorithm formula:

$$\theta - a\nabla f(\theta),$$

In the ordinary gradient descent, the size and direction of the gradient cannot be predicted in advance, so the optimization route of the machine, that is, the iterative process, cannot be predicted in advance. However, after adding the restriction of equality constraint, our optimization route is determined; that is, we need to carry out an optimization algorithm similar to the gradient descent algorithm along the circular trajectory formed by the equality constraint. Similar to the normal gradient descent algorithm, we will use the Z-direction descent rate of its position after each iteration to replace the gradient size. Moreover, the position x=0 in the equation constraint is selected as the initial iteration position. The reason for selecting this position is that after observing the function of the three-dimensional function under the equation constraint in MATLAB, it is found that the position of the two endpoints under the circular image of the equation constraint is the point with the larger function value. The learning rate of general gradient descent plays the same role in this optimization algorithm, and human debugging is also necessary to obtain a better iterative learning rate. We set the learning rate in the above formula as 0.5 degrees. We take (1,1) as the center of the circle, similar to the principle of polar coordinates, rotate an Angle of $\theta * \Delta z$ on the circular trajectory from the initial position, and calculate the next position (x, y) after iteration by analogy and iteration repeatedly. Finally, there will be an effect similar to the gradient descent algorithm, that is, when the decline rate of z becomes smaller. At this time, it is also close to the lowest point of the function value, and the change Angle of its next iteration will become smaller, so as to have the effect of not missing the lowest point. When the decline rate of z is large, the change Angle will also become larger so that it can iterate faster and get close to the minimum value.

The following describes the detailed principles and operations

First, we introduce how to calculate the z mentioned here, that is, the rate of change of the height or function value. First, we clarify that this algorithm is only effective for such special problems and it has certain limitations

We make the tangent line along the constraint function from the (x, y) position obtained after each iteration and get the function expression of the tangent line. In this case, the decline of its own function along the tangential direction can be approximately the decline of (x, y) in the

space curve, which is assumed because it is strongly convex and the shape is similar to a three-dimensional image of a bowl. Now that we have a linear relationship between x and y, we just need to replace y with x to get a function of z with respect to x, and the derivative of that is the rate of change of z with respect to x. But notice that this is in three dimensions, so when x changes the minimum, the linear relationship between y and x is the same minimum in the tangent direction of the transformation in the mathematical sense, so the result of the derivation divided by the square root of 2 is the actual rate of change along the tangent direction. Now that we have a linear relationship between y, z and x, let's say y=ax, z=cx and let's make the sum of x squared plus y squared plus z squared equal to 1, to help us see the actual change in z along the tangent. At this point, the whole step of our algorithm, namely the principle, has been roughly sorted out.

If the reader finds that the error is large when applied to other similar problems, it is recommended to calculate the rate of change of z carefully. Calculate the change value of the space curve function corresponding to the change minimum x after the change minimum x. Instead of calculating the change value along its tangent direction, strictly calculate the relationship between the y change minimum and the X minimum to determine whether it is a linear or nonlinear relationship, and calculate the change value of z in the space curve function according to the above-introduced method, that is, the method of square summation and then root.

4.2.2 The penalty method

In addition to the algorithm, we came up with on our own, we also tried the penalty method. Its idea is to convert a constrained problem into one without constraints.[5]

Specifically, we used the quadratic penalty method. The steps are as follows:

Step 1: Construct the quadratic penalty function in the form:

$$P(x,\sigma) = f(x) + \frac{1}{2}\sigma\left[\sum_{i\in\mathcal{E}} c_i^2(x)?+\sum_{i\in\mathcal{I}} c_i^{\,2}(x)\right],$$

where $x \in \mathbb{R}^n and f(x)$ is the objective function, σ is the penalty coefficient, and $c_i(x) = 0, i \in \mathcal{E}$ and $c_i(x)?0, i \in \mathcal{I}$ are the equality and inequality constraints, respectively. Notice that the term including the inequality constraints is $c_i^{\,2}(x)$, which is defined as $\left(0, c_i^2(x)\right)$. This means the penalty is zero when the inequality is satisfied.

Step 2: Choose an initial value for σ and x and minimize the function $P(x,\sigma)$ for this value of σ. This is an unconstrained optimization problem and can be solved using gradient descent or other methods.

Step 3: Increase σ by a factor ρ and minimize $P(x,\sigma)$ for the new value of σ. Keep doing this until a certain condition is met, i.e., the function passes the convergence test. This works because as σ increases, the penalty term $\frac{1}{2}\sigma\left[\sum_{i\in\mathcal{E}} c_i^2(x)?+\sum_{i\in\mathcal{I}} c_i^{\,2}(x)\right]$ will also increase, which can force our point to get closer to the feasible domain. As σ approaches the infinity, x will asymptotically approach the optimal solution.

The penalty function of this specific problem is

$$P(x,\sigma) = T(x) + \frac{1}{2}\sigma\left[(x-1)^2 + (y-1)^2 - 6^2\right]^2.$$

According to our program, the optimal location of the fire station should be at (4.244, 6.220.)

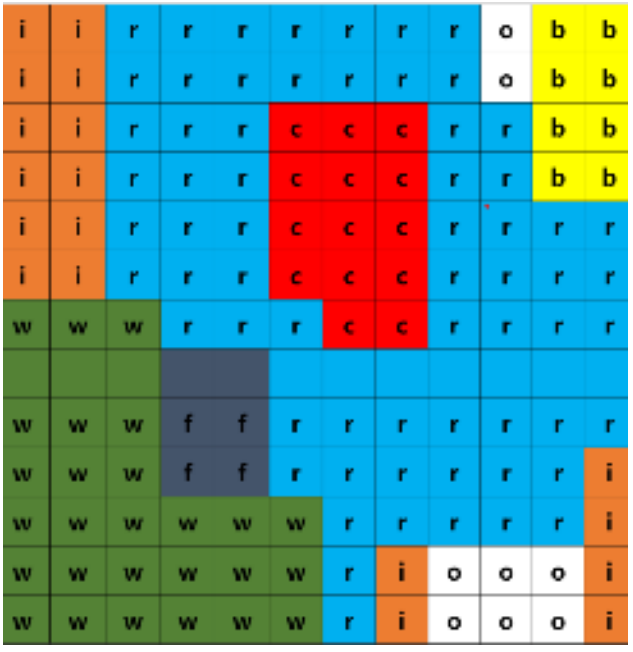

Figure 8: Results of Parts 1, 2 and 3

In Figure 8, the blue dot is the position of the station in Part 1, the yellow dot is for Part 2 and the orange dot is for Part 3. The point in Part 3 is noticeably closer to the forest.

5 Conclusion

5.1 Conclusions on the location of the fire station

Our model predicts that the best place for the fire station is typically towards the city's center, presumably because this area is the busiest and has the greatest potential for property loss. Positions are around (6.91, 6.24) and (6.99, 6.99) for the situation before and after adding weight, respectively. The position after adding the constraint is

around (4.24, 6.22).

5.2 Conclusions on the Gradient Descent Algorithms

The gradient descent algorithm with momentum (in 2.3) took more than a hundred iterations, whereas the algorithm in 2.4 typically took around eight iterations. This is most likely a result of the long axis in this model being less obvious than the short axis, which makes the momentum algorithm iterate more frequently. Therefore, the momentum method might not be the best approach to tackle our problem.

5.3 Future prospect

We must admit that our model is a simplified one. There are a few things we can do, if time permits, to improve it. There are also some research directions that we may explore in the future.

5.3.1 . Consider the situation of building more than one fire station

Currently, the location of fire stations is frequently chosen using the shortest distance theory. But in the future, we can think about choosing several ideal places for fire stations. For handling increasingly complicated and changing fire incidents and emergencies, this strategy guarantees better coverage and reaction times within metropolitan regions.

5.3.2 . Consider a more realistic path or model

The shortest way through a straight line is frequently used in our present path design. Future planning should, however, take into account more practical routes or models. For instance, it is important to consider the geography, the state of the roads, and other potential difficulties. This can entail choosing sloped or twisting roads that more accurately represent real-world situations.

5.3.3 . Consider restrictions or barriers

It is essential to include a number of restrictions or obstacles in the fire station optimization procedure. The response time and efficiency of fire stations may be impacted by these obstacles, which may include traffic congestion, congested areas, or difficult-to-reach locations. In order to provide effective emergency response, the placement and distribution of fire stations can be improved.

5.3.4 . Consider factors like weather and regional impacts

Future fire station designs should take into account elements like local impacts and weather patterns. Different regions may experience different weather patterns as well as particular regional difficulties like natural catastrophes or distinctive physical factors. The ability to situate fire stations and allocate resources more precisely is made possible by taking these considerations into account.

5.3.5 . Investigate more optimization algorithms

Exploring and assessing various algorithms is crucial to improving fire station optimization. For instance, randomized GDO or Nadam algorithms might be used to decide on the best sites for fire stations and how to allocate resources. Planning and administration of fire stations might become more effective and efficient with ongoing study and innovation in algorithm development.

Acknowledgment

Junjie Wu, Yiyang Zhang, Yuqing Zhuang, Zhuoer Huang and Ziyu Wang contributed equally to this work and should be considered co-first authors.

References

[1] Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. *International Conference on Computational Statistics*.

[2] Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv: Learning*

[3] Ighalo, J.O., Adeniyi, A.G., & Marques, G. (2020). Application of linear regression algorithm and stochastic gradient descent in a machine-learning environment for predicting biomass higher heating value. *Biofuels, 14*.

[4] Meerschaert, M. M. (2013a). 3.2 Multivariable Optimization. In *Mathematical modeling* (pp. 66–67). Amsterdam: Elsevier.

[5] Liu, H., Hu, J., Li, Y., & Wen, Z. (2020). 7.1 The Penalty Methods. In *Zui You Hua: Jian Mo, Suan Fa Yu Li Lun = optimization: Modeling, algorithm and Theory* (pp. 297–304). Beijing: Gao deng jiao yu chu ban she.