AAE Freshman Project Report

# Group 9

Groupmates:

Tsz Kit CHAN

Yui Tin CHAN

Tsz Tung CHIU

Zhendi FAN

Chengcheng ZHANG

# 

# 1. Background of Path Planning to Aviation Engineering

Path Planning, in other words motion planning, refers to the calculation of the shortest path between two different specific points. Between two of the points, there can be different obstacles or even different areas with different limitations. Path Planning is widely used in different areas, including computing and gaming, with examples of League of Legends and AI.[1] Path planning surrounds us, but how does this technology really work in the Aviation industry?

As we all know, one of the most important things to consider in aviation is safety. By using the Path Planning technology, airports can simulate and plan different scenarios for them so that they will have different backup plans for different situations for example wind speeds and directions, amount of aircrafts etc. to maximize the capacity and reduce the chance of accidents. Besides the airport, airplanes also use Path Planning technology. Planes need fuel to fly. Carrying too much fuel brings external load to the plane while the plane cannot reach the destination if it carries an exact amount of fuels. Path planning is useful in this situation. By entering data and coordinates in the computer system, the system can calculate the shortest path available with the lowest cost. This helps pilots to estimate more accurately about the amount of fuel to carry hence can reduce operating cost. By using Path Planning, the plane can also fly by shortest route and to avoid those areas that are not able to fly.

## 2. Theory of Path Planning Algorithm

We have introduced why path planning is important to aviation engineering. So let us discuss how path planning actually works on the computer.

In this project, the algorithm we used is called A star. A star algorithm is a method with a long history. It uses the search strategy called best-fit. Best-fit strategy means that the algorithm can explore the most likely node and also eliminate all the interfering solutions.[2] The algorithm finds the shortest node next to the starting node continuously until it reaches the goal node. As the algorithm calculates the route in a fixed pattern, the result for every single data is guaranteed to be the same. In the algorithm, we can set obstacles, borders and different limitations. Both obstacles and borders cannot be touched by the final route and those factors will either increase or reduce the movement of the search of neighbouring nodes to simulate the situation we have entered. Also, it is found that more time is needed for a larger or more complicated task.[2]

## 3. Introduction of the Engineering Tools

## a. Python

Python is an interpreted, high-level and general-purpose programming language, combined with dynamic typing and dynamic binding, which is widely used in computer coding. It was created by Guido van Rossum in 1991, a dutch programmer. Python aims for providing language constructs and object-oriented approach to increase the readability, helping users to code in a clear and logical order.[3] Also, Python supports modules and packages, which encourages program modularity and code reuse.[4] Due to its comprehensive standard library, for examples: programming paradigms, object-oriented, and functional programming, Python was very popular when it was released.[5] After that, there are Python 2.0 and Python 3.0. More different operating systems can be supported by Python, such as Linux or MacOS.

## b. GitHub

Github is a website host for software development and version control, providing a platform for people to exchange their ideas about code. Moreover it provides source code management service and several collaboration features, such as bug tracking and feature request. The professional and enterprise version of Github are usually used for business purposes.[6] On the other hand, the free version of Github is used for open source projects.[7] What makes Github powerful are three features – fork, pull request and merge. It makes the process of coding much more easy and efficient, users can use other modules and change by their needs.[8]

## 

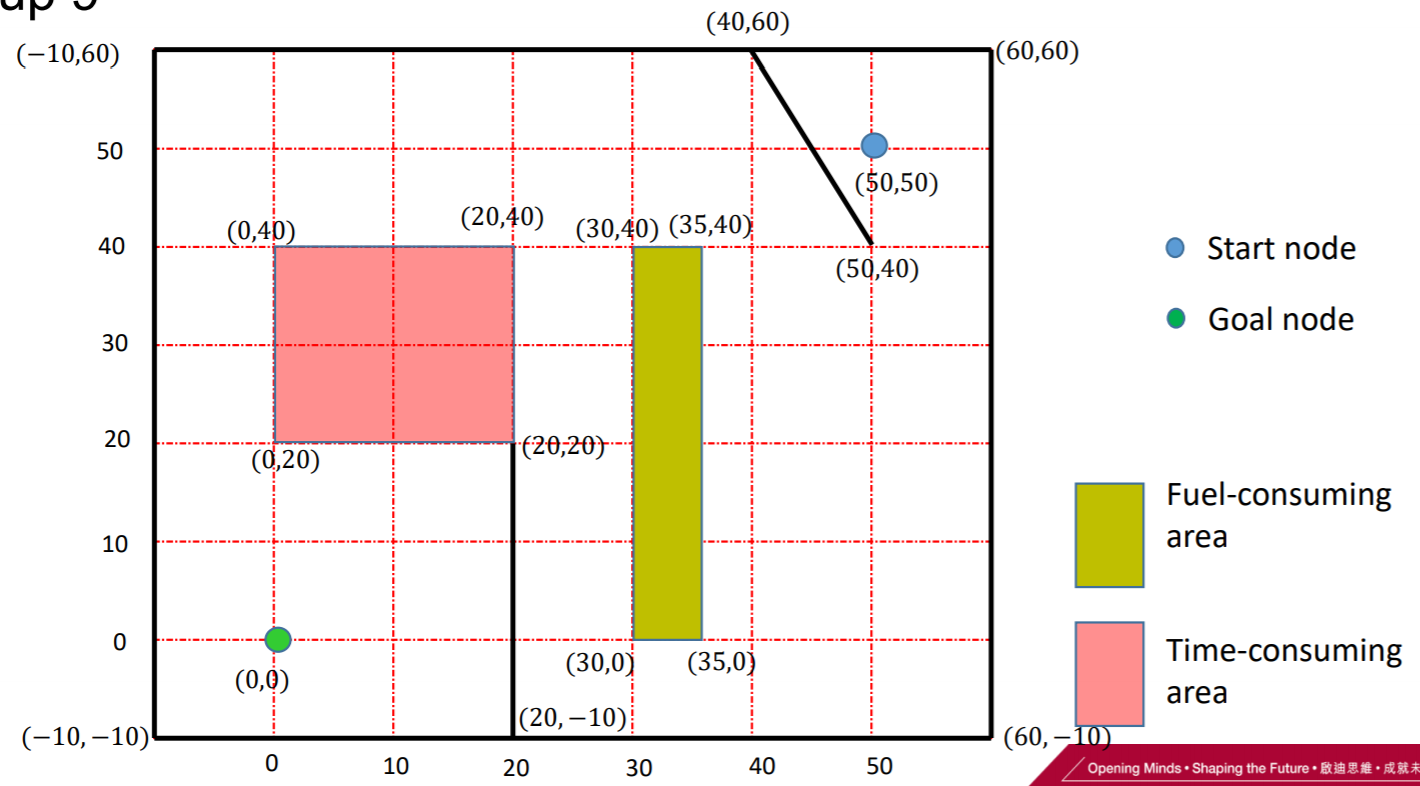
## 

## 4. Task 1: Methodology, Results and Discussion

## a. Methodology

In task 1, we have to find out which Polyu aircraft models that achieve the minimum cost.

Our group assigned the graph:

Fig.1

Initially, we got the graph:

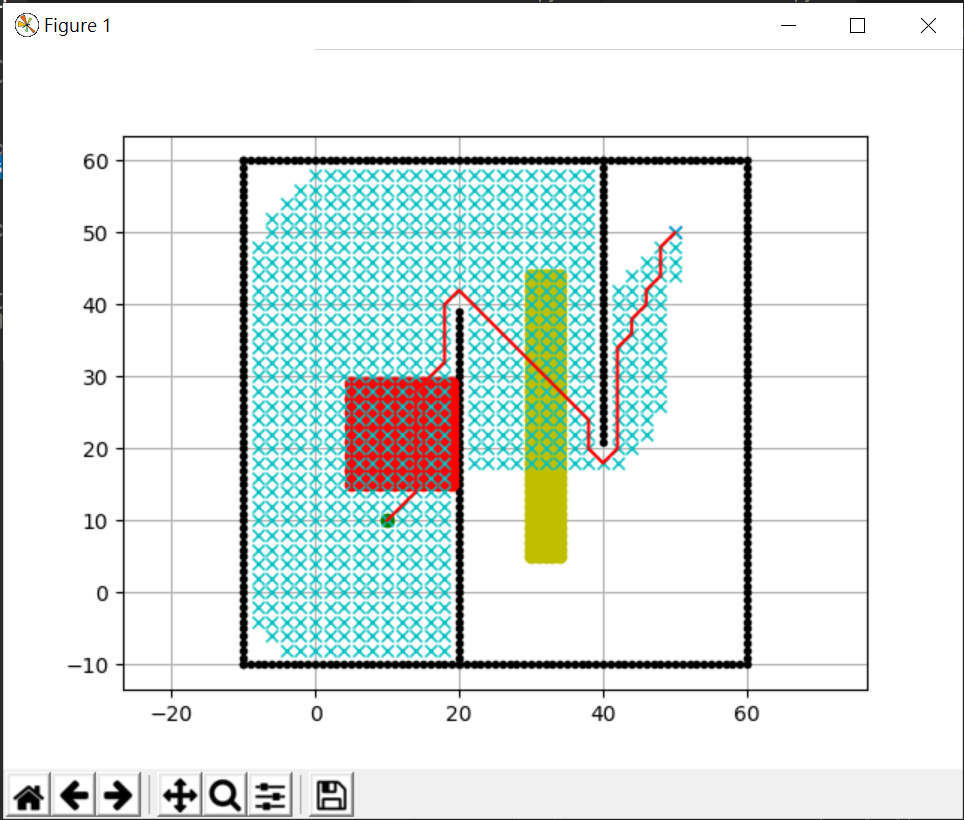


Fig.2

So, we have to change the starting and ending point, the position and size of the fuel-consuming area and time-consuming area, the length and position of the borders.

For the start and goal position, we have to change the data from

**[10,10],[50,50] → [50,50], [0,0]**

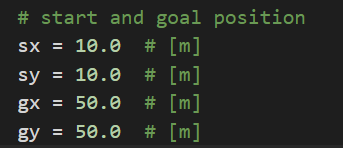


Fig.3 The original start and goal position

To renew the positions, we will need to change the data of **[sx,sy] and**

**[gx,gy].**Therefore, the code will change to **[50,50] and [0,0]**.

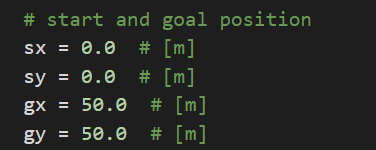
****

Fig.4 The assigned start and goal position

For the position and size of the fuel-consuming area, the original fuel-consuming area is a **square (15x15)** and place between **[5,5] and [3,30]**

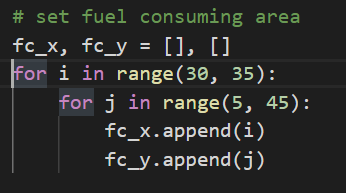


Fig.5 The original fuel-consuming area

So, we have to change the code to

**for i in range(30, 35):**

**for j in range(0, 40):**

**fc\_x.append(i)**

**fc\_y.append(j)**

The time-consuming area is a **rectangle** **(5x40)** and place between **[25,5]**

**and [35,45]**

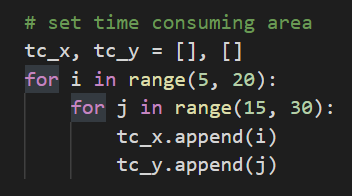
****

Fig.6 The original of time-consuming area

So, we have to change it to

**for i in range(0,20):**

**for j in range(20, 40):**

**tc\_x.append(i)**

**tc\_y.append(j)**

For the border, there are two borders place at **[20,0] with 50 units** longand

**[40,60] with 40 units** long.

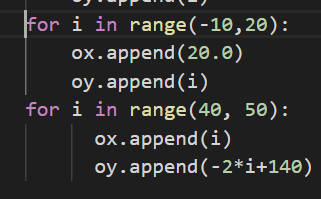
****

Fig.7 The original of the two borders

So, we have to change them to

**for i in range(-10,20):**

**ox.append(20)**

**oy.append(i)**

**for i in range(40,50):**

**ox.append(i)**

**oy.append((100-i)/2+20)**

Finally, we combine all changes and get this graph:

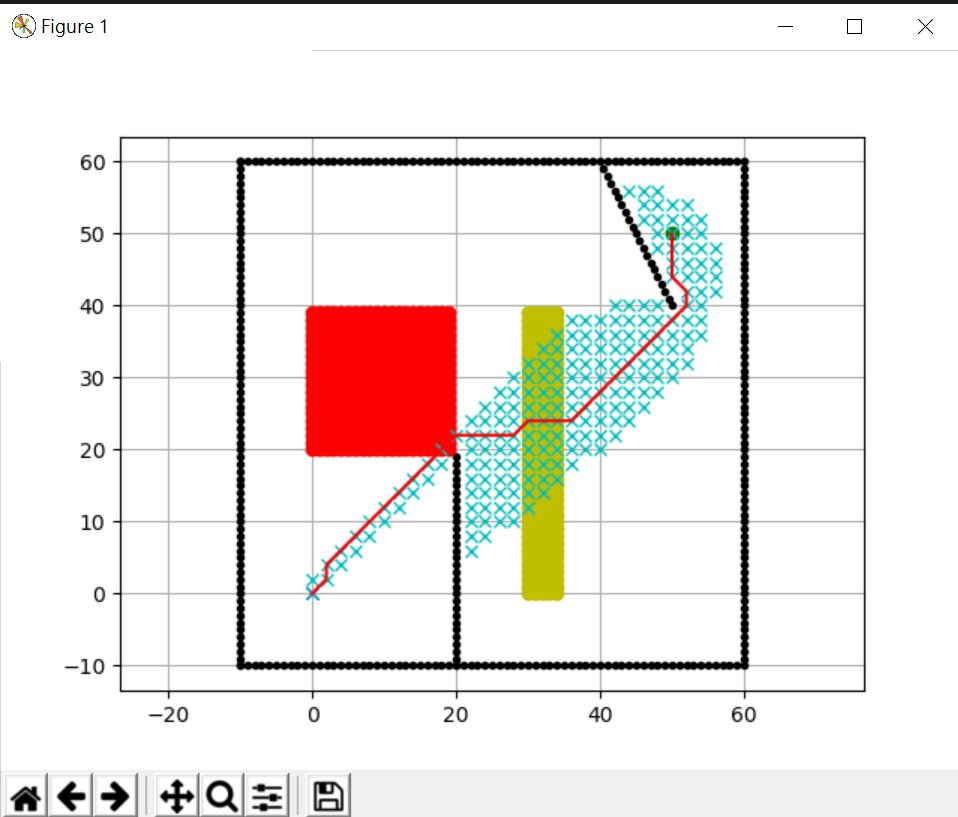
****

Fig.8 The final graph

## b. Results

The task assigned us to find which Polyu aircraft models that

achieve the minimum cost.

The equation to find the trip cost:

**C = 𝐶𝐹 ∙ (∆𝐹 + ∆𝐹𝑎 (𝑥, 𝑦) ) + 𝐶𝑇 ∙ (∆𝑇 + ∆𝑇𝑎 (𝑥, 𝑦) ) + 𝐶c**

𝐶𝐹=cost of fuel per kg

𝐶𝑇=time related cost per minute of flight

𝐶𝑐=fixed cost independent of time

𝐶𝑇=time related cost per minute of flight

∆𝐹=trip fuel

∆𝑇=trip Time

The trip cost equation provided in python:

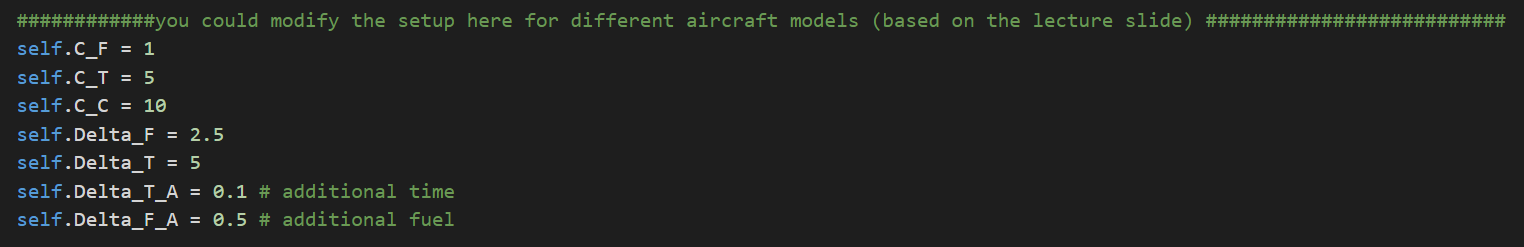


Fig.9 The trip cost code

The data of each PolyU Flight model is important to calculate the trip cost.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Aircraft Model | C\_F | Delta\_F | C\_T | Delta\_T | C\_C | Delta\_Fa | Delta\_Ta |
| PolyU-A380 | 1 | 1 | 2 | 5 | 10 | 0.2 | 0.2 |
| PolyU-A381 | 1 | 1.5 | 3 | 5 | 10 | 0.3 | 0.4 |
| PolyU-A382 | 1 | 2.0 | 4 | 5 | 10 | 0.4 | 0.5 |
| PolyU-A383 | 1 | 2.5 | 5 | 5 | 10 | 0.5 | 0.1 |

We can enter each data of a different PolyU-A38X model to find out the cheapest trip cost.

|  |  |
| --- | --- |
| Model | Result |
| PolyU-A380 | 834.468181006535 |
| PolyU-A381 | 1053.309847460627 |
| PolyU-A382 | 1272.0515139147199 |
| PolyU-A383 | 1490.2931803688125 |

If we compare with these four PolyU models, we can see the PolyU-A380 costs $834 and has the cheapest trip cost.

## c. Discussion

We can figure out which PolyU-A38X models achieve the cheapest cost through python code. We found that the PolyU-A380 had the lowest trip cost compared with other PolyU aircraft models. All the aircraft flew in the same route but why did they perform different trip costs?

As we can see the table in the Result section, each aircraft model has a numerical difference. ∆𝐹, C\_T, ∆𝐹a, ∆Ta are different between each model and these data play a key role on the trip cost. The model, PolyU-A380 has the lowest of these four data and after the calculation, we find it costs only 834.

## 

## 5. Task 2.1: Methodology, Results and Discussion

## a. Methodology

## In task 2.1, we need to find the PolyU Aircraft Model(4 constraints with 2 variables) that achieve minimum cost for the challenge assigned to our group.

To solve this task, we need to define the values of Cf and Ct which meet the constraints. Meanwhile the values should make the Aircraft Model of our group achieve minimum cost for the target assigned to our group.

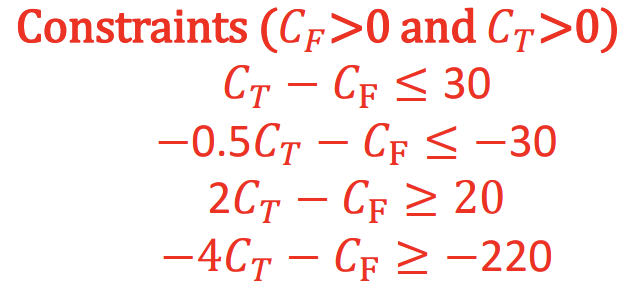


Fig. constraints of variables

From the lecture, we can know the function of trip cost:

**C = 𝐶𝐹 ∙ (∆𝐹 + ∆𝐹𝑎 (𝑥, 𝑦) ) + 𝐶𝑇 ∙ (∆𝑇 + ∆𝑇𝑎 (𝑥, 𝑦) ) + 𝐶c**

Since **∆𝐹=∆𝑇,** and the values of **∆𝐹𝑎 (𝑥, 𝑦)** and **∆𝑇𝑎 (𝑥, 𝑦)** are negligibly small when compare to **∆𝐹** and **∆𝑇**. We can nearly regard the function as:

**C = 𝐶𝐹 ∙ ∆𝐹 + 𝐶𝑇 ∙ ∆𝑇 + 𝐶c**

According to the given table of Aircraft Model, we can continuously simplify the function:

**C = 𝐶𝐹 \*5 + 𝐶𝑇 ∙\*5+ 10**

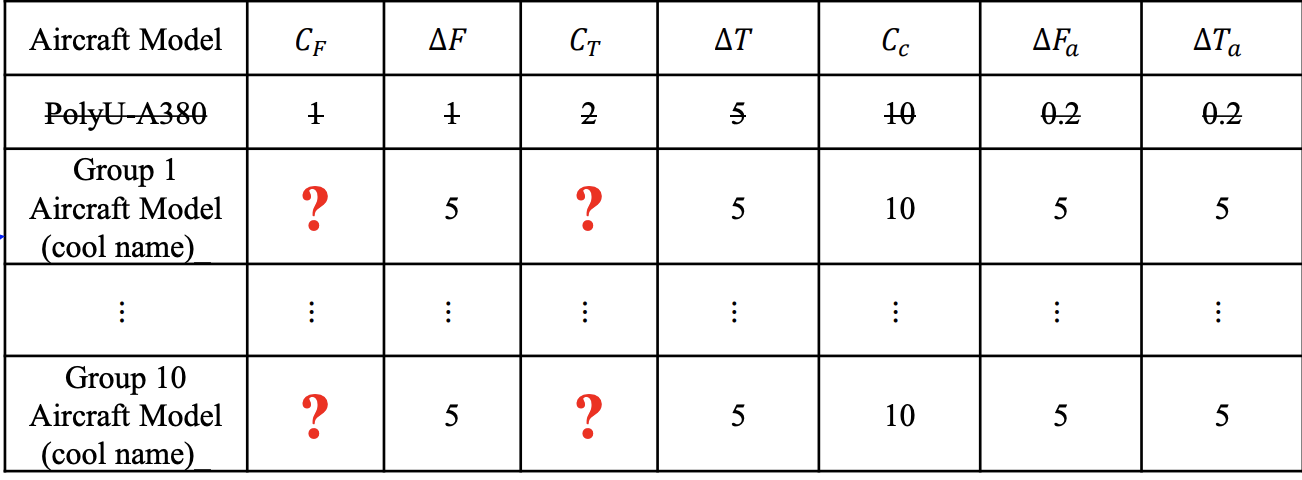


fig. Given table of task 2.1

We can observe that when (**𝐶𝐹**+**𝐶𝑇**) achieve the minimum value, cost of the flight will achieve to the minimum in the same time.Using linear programming, we can find the values should be **𝐶F**=**𝐶𝑇=20**

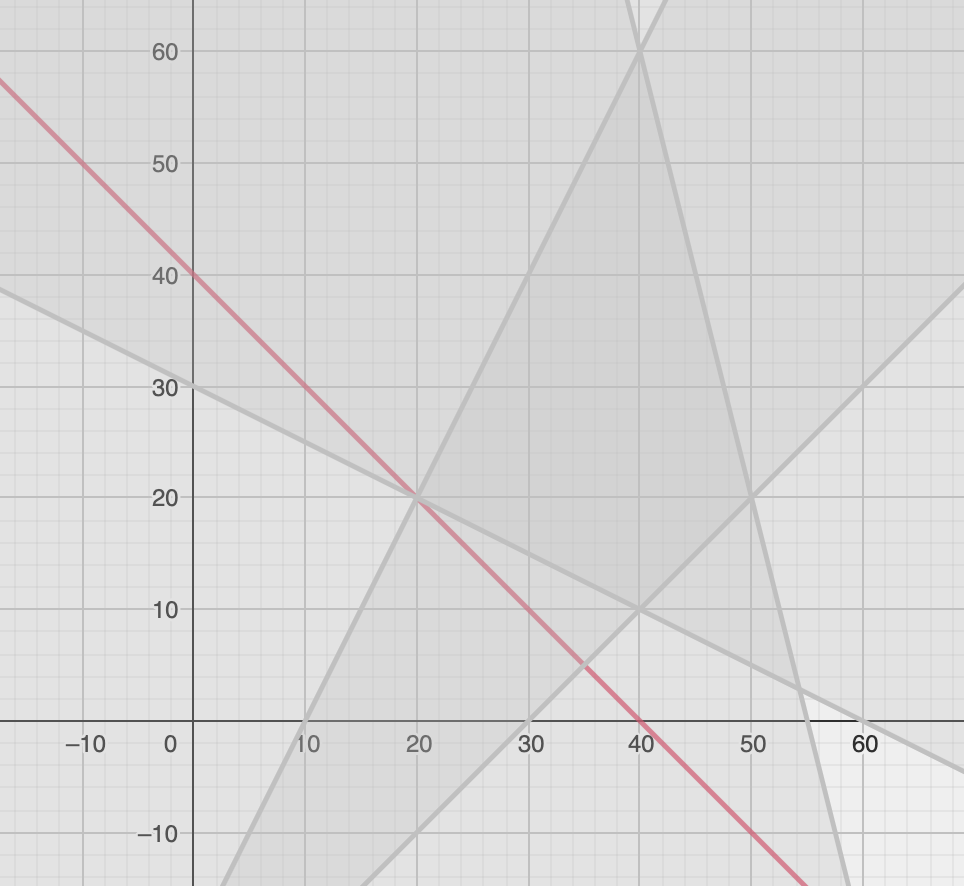


fig. Linear programming result

According to the linear programming result, we build our Aircraft Model

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Aircraft Model | 𝐶𝐹 | ∆𝐹 | 𝐶𝑇 | ∆𝑇 | 𝐶𝑐 | ∆𝐹 𝑎 | ∆𝑇 𝑎 |
| Group 9  Aircraft Model | 20 | 5 | 20 | 5 | 10 | 5 | 5 |

fig. Aircraft Model of group9 for task 2.1

Similarly as task1, we input our Aircraft Model to the code.

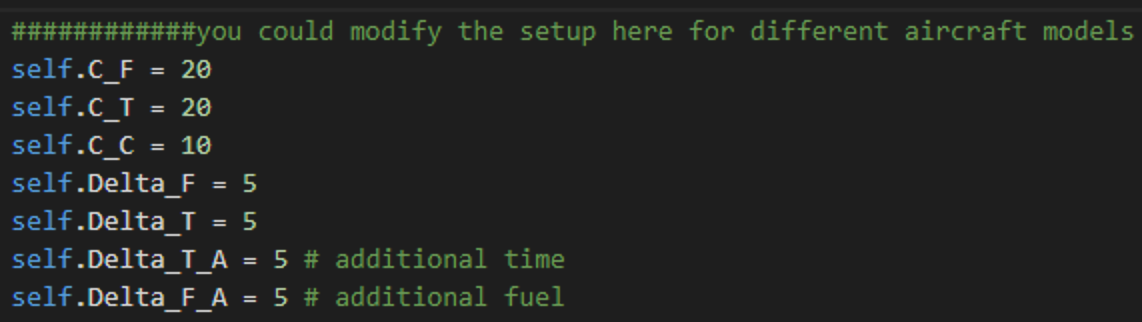


fig. The trip cost code of task2.1

Finally, we run the python code and record the results.

## b. Results

Using linear programming, we can easily find the cheapest Aircraft Model for our assigned task.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Aircraft Model | 𝐶𝐹 | ∆𝐹 | 𝐶𝑇 | ∆𝑇 | 𝐶𝑐 | ∆𝐹 𝑎 | ∆𝑇 𝑎 |
| Group 9  Aircraft Model | 20 | 5 | 20 | 5 | 10 | 5 | 5 |

fig. Aircraft Model of group9 for task 2.1

Running this model, we can get the path figure of task2.1

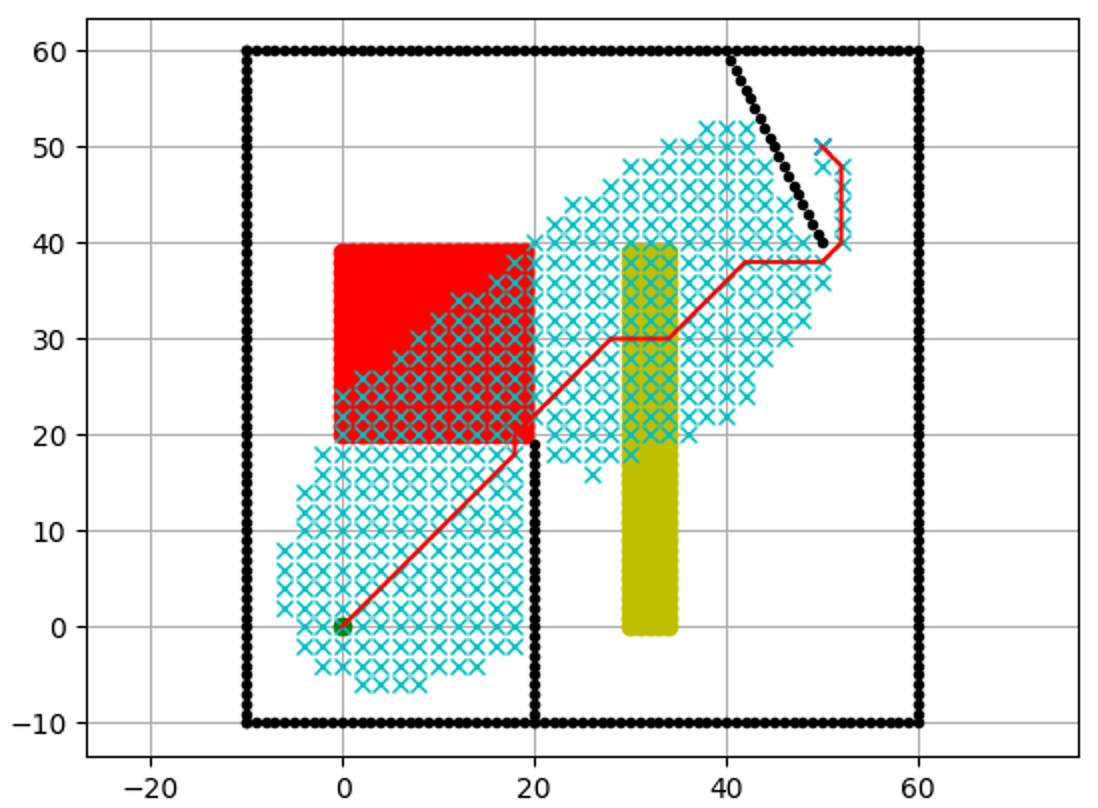


fig. Task 2.1 result figure

According to the printed running result, we know this Model will cost 8479.70. This model can achieve minimum cost for the challenge assigned to our group.

## c. Discussion

To figure out this task, we use linear programming. We find the Aircraft Model should be:

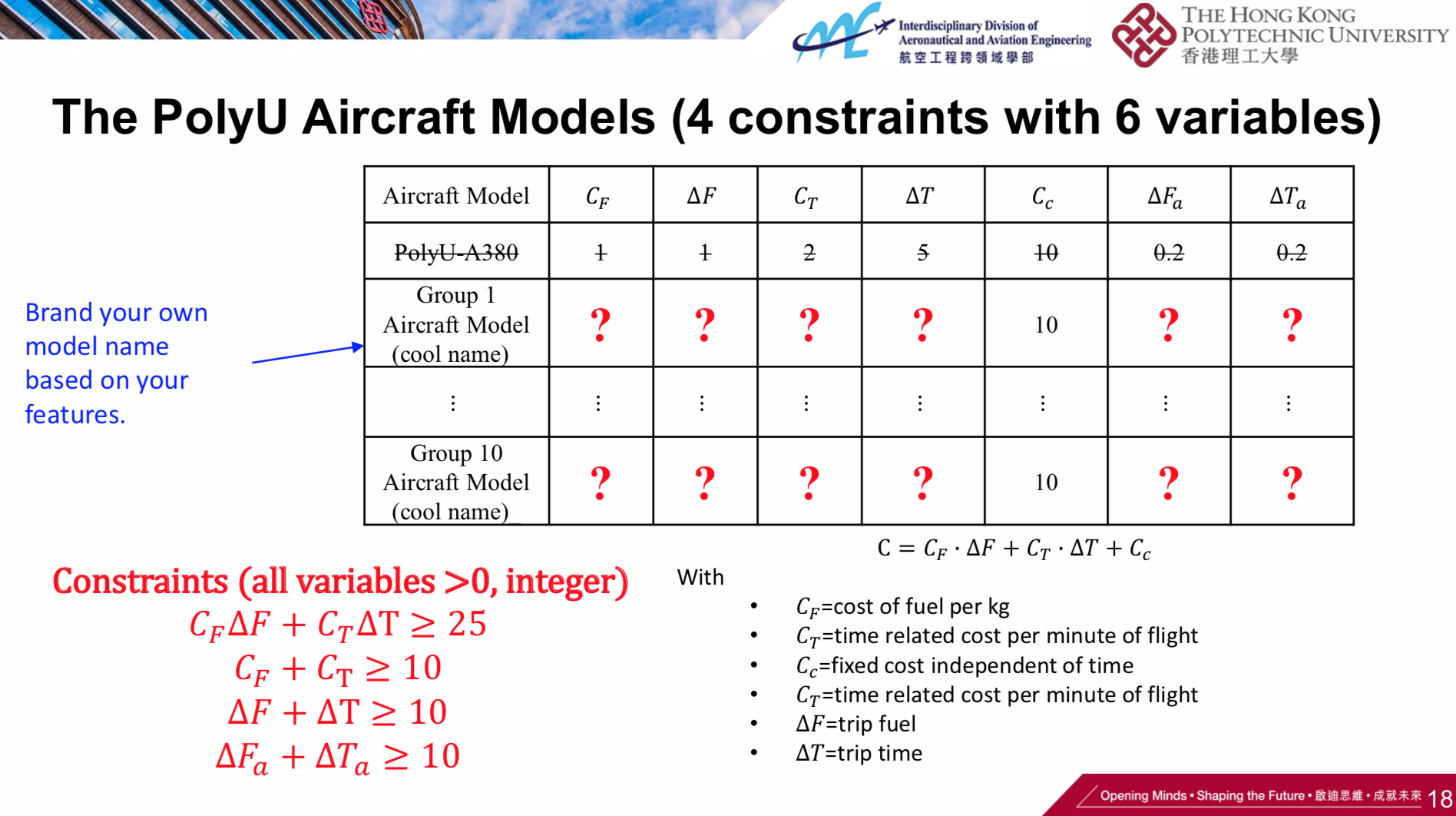
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Aircraft Model | 𝐶𝐹 | ∆𝐹 | 𝐶𝑇 | ∆𝑇 | 𝐶𝑐 | ∆𝐹 𝑎 | ∆𝑇 𝑎 |
| Group 9  Aircraft Model | 20 | 5 | 20 | 5 | 10 | 5 | 5 |

But since we used an approximation, the result of our task might not be very reliable. So that we slightly change the volume of **𝐶F** and **𝐶𝑇**. By repeatedly doing that, we find if **𝐶F** + **𝐶𝑇** equal to a same constrain, the cost of the trip will not change. This result is contradicted with the given function: “**C = 𝐶𝐹 ∙ (∆𝐹 + ∆𝐹𝑎 (𝑥, 𝑦) ) + 𝐶𝑇 ∙ (∆𝑇 + ∆𝑇𝑎 (𝑥, 𝑦) ) + 𝐶c**'' because the values of **∆𝐹𝑎 (𝑥, 𝑦)** and **∆𝑇𝑎 (𝑥, 𝑦)** are apparently not the same. We infer that there might be some code errors that lead to this result.

However, doing this task strengthens our ability of using linear programming to figure out problems. Meanwhile we have a deeper understanding of the AStarPlanner code.

## 6. Task 2.2: Methodology, Results and Discussion

## a. Methodology



For the convenience of expression,

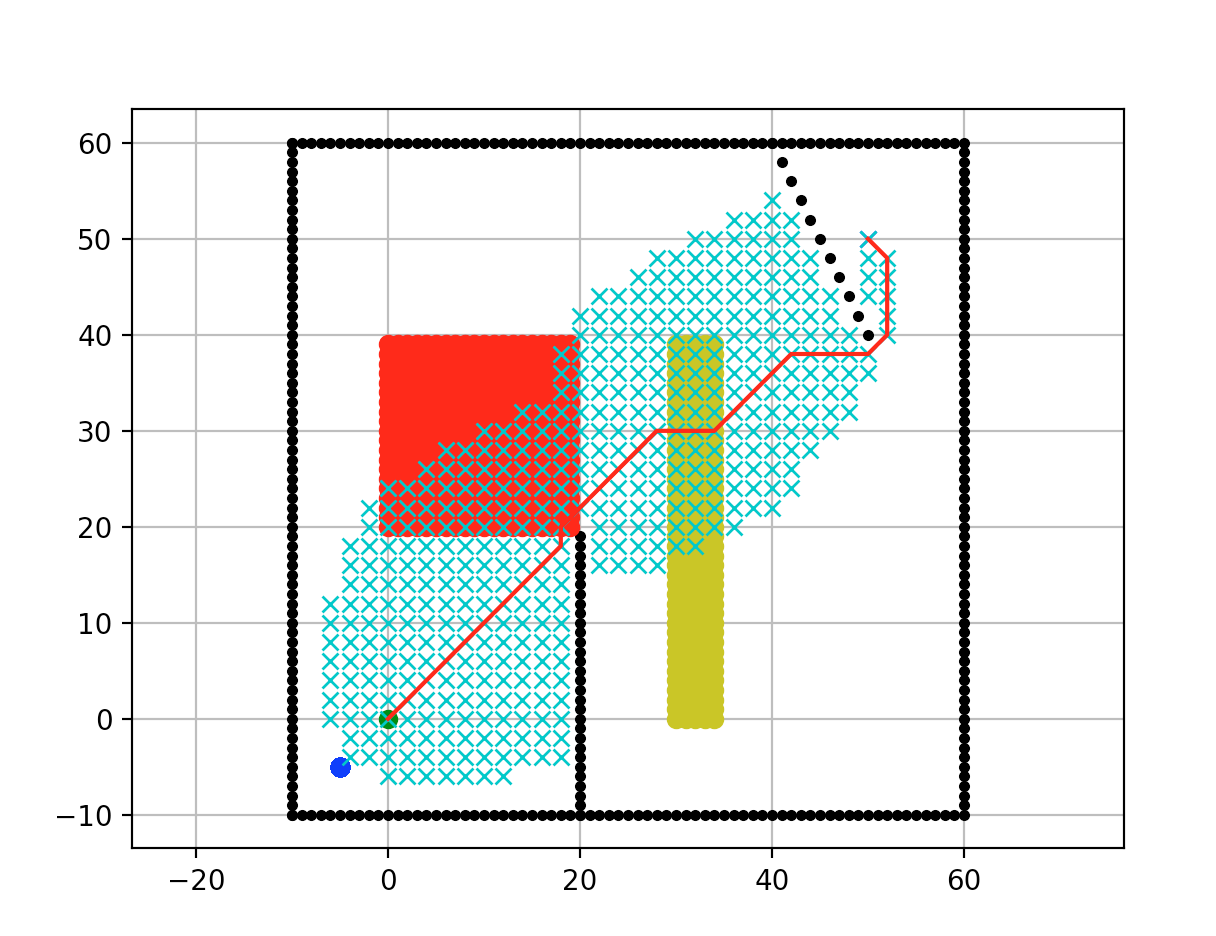
let CF=a; CT=b; △F=c; △T=d; △Fα=e; △Tα=f

so the formula become C=a(c+e)+b(d+f)+Cc

It is obviously that wherever it go, the formula have a part “ac+bd+Cc”(Cc is a constant, we don’t need to consider it)

While our first constraint just contains “ac+bd” and it is at least 25. So we must let it equal to 25 to make sure our result is cheapest.

And consider constraint 2 and 3 together with 1. We know abcd’s value is 1,1,15,10. So far we don’t know the order, but we know there must be value 1 between a and b, and 1 between c and d. We consider in consuming area, a or b will multiply e or f accordingly, so they must be as small as possible. Thus, we let them equal to 1 and 10(we still don’t know the order now)



Then observing the route, we find it goes through a long distance in fuel consuming area than in time consuming area. So we need to let them both as small as possible we could and give the smaller one to fuel consuming. 10 must be compared with 1, so the other one 9 compared with 1 that is the pair given to fuel consuming.

To see the result clearly:

1(15+9)+10(1+1)

(Note: All the analyzation is base on the original formula, not the formula in our original code)

## b. Results

That is:

CF=1; CT=10; △F=15; △T=1; △Fα=9; △Tα=1

## 

## 

## 7. Task 3: Methodology, Results and Discussion

To understand the whole program may be not easy, but to add a special part for the program is much more simple. And that is what we do.

Our aim is to add a special area just like the consuming area. The only difference is the number and the color what it shows. So the things we need to do is find the code relation to the original and copy and make a little change at the variables, and it would become the things we want.

Fortunately we found what we need



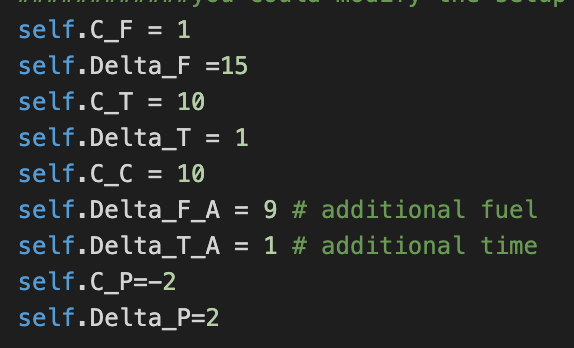
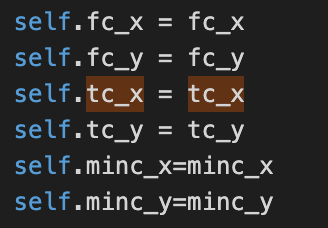
and add two variable at the end of it

then find a crucial variable in the whole program

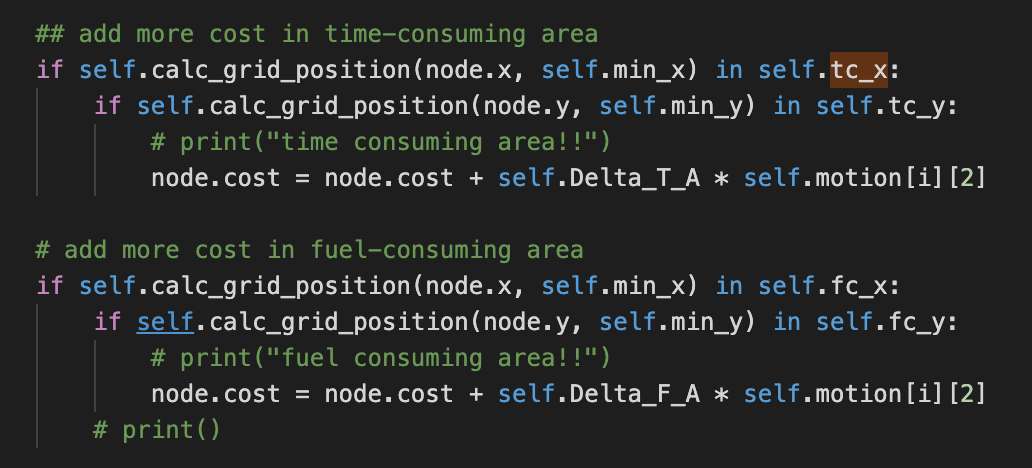


we find all the calculation part and the sand table showing part

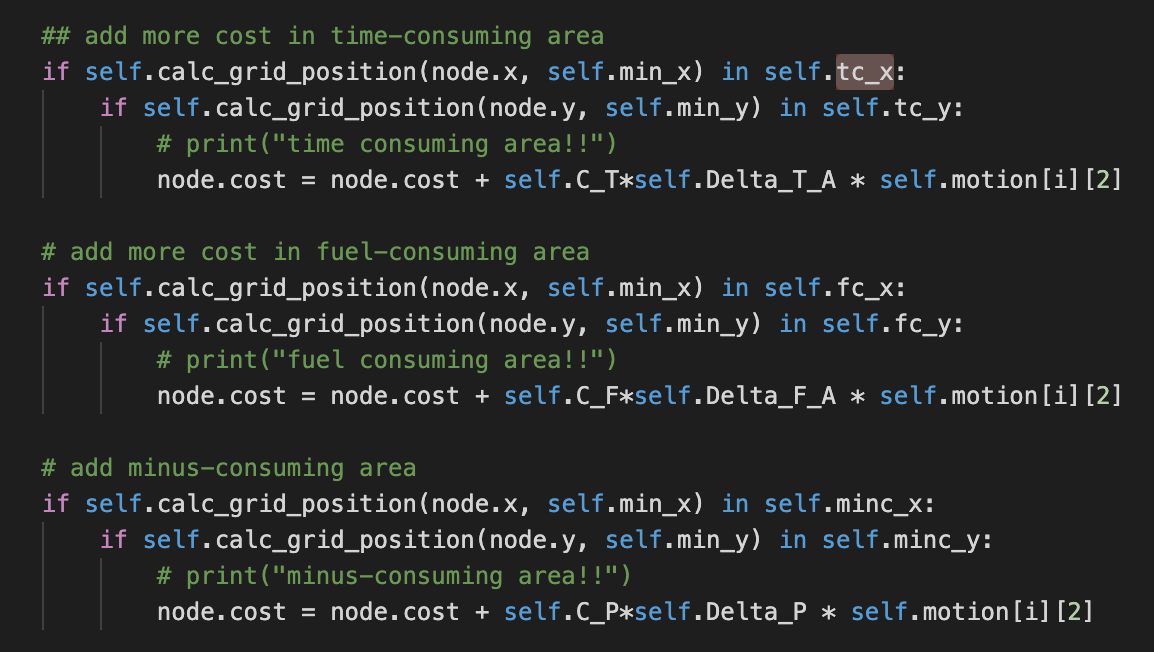
like:



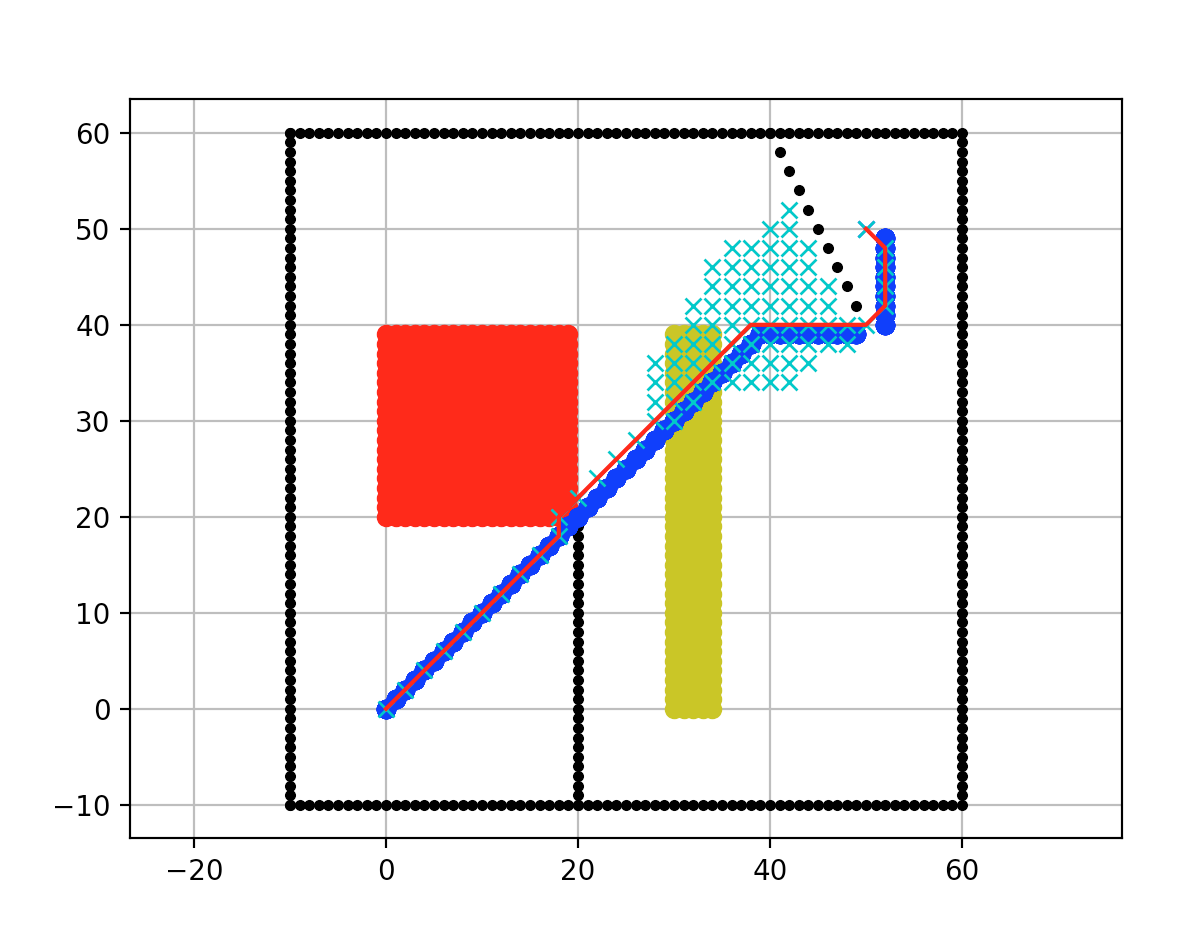
But when we edit the calculation part, we find some error in it:



It does not multiply CF or CT to △Fα or △Tα accordingly. We confirmed it with our TA and he said l am right. So we revise the original one and add our new part.



Finally we made it. The minus cost area can be any shape we want.

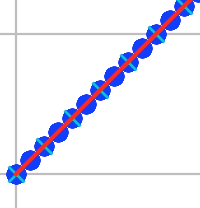


Then we can consider where the 16 m2 should be placed.

A funny thing happened. The professor in the class said a 1\*1 square is considered as 1

m2 and squares must continue.

But actually our special areas in the sand table are composed of circles with radius 1. Later we found the actual area is not what we see on screen!!!!



To analyze the situation, we can suppose the special can be in exactly squares. There is no doubt that we need to best use each space of the minus cost area, so each1\*1space must have route crossing. And the longest cross is diagonal.

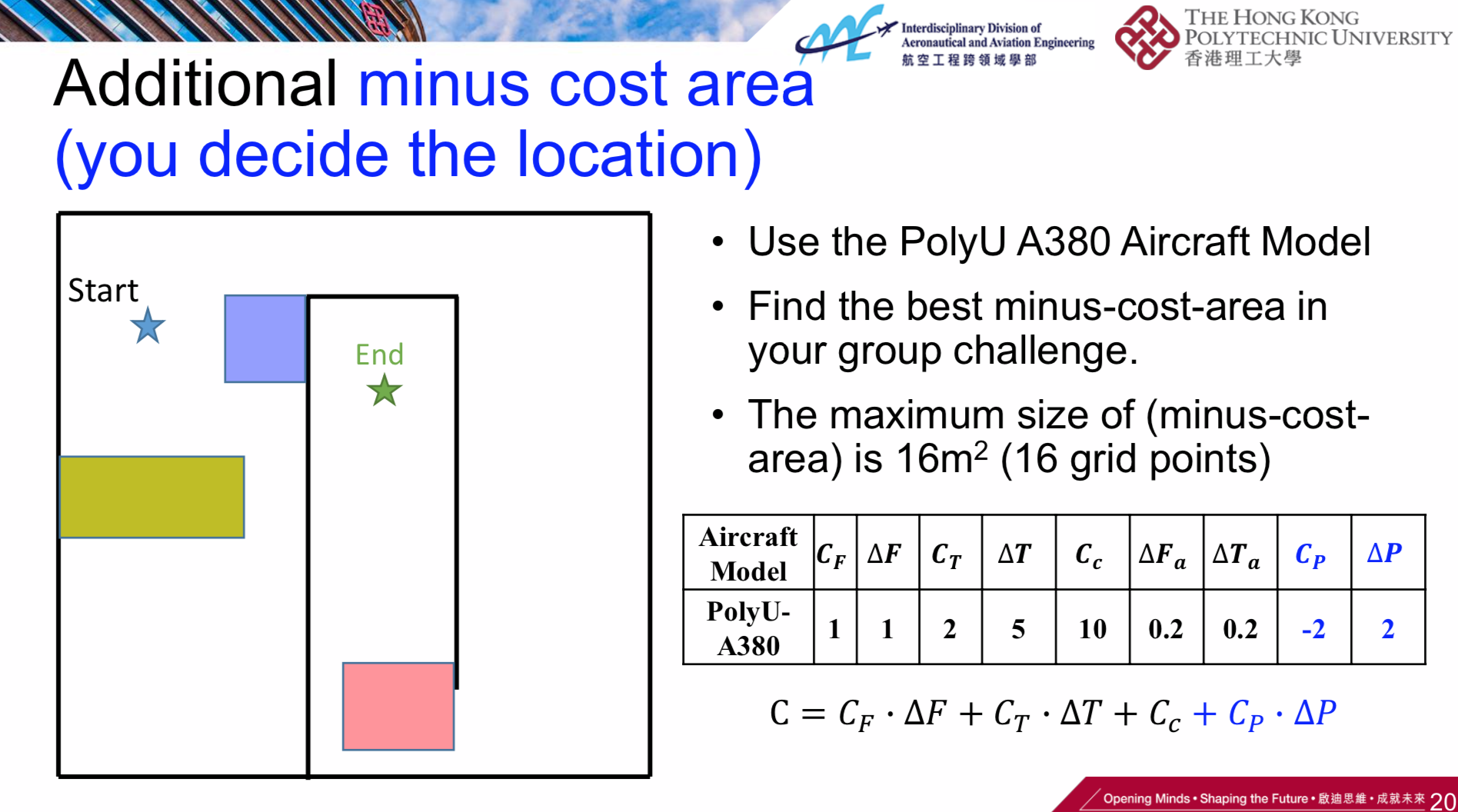
Of course we can’t change the original route.

So we have to find 16 continuous diagonal-line-segments in the original route. If not, find as much as possible part and place the minus cost area.

Therefor if we can find 16 continuous diagonal-line-segments in the original route, we can simple add (-2)\*2\*16\*2^(1/2) to our original result. And as for our case, it can find 16 continuous diagonal-line-segments, so this method is suitable.

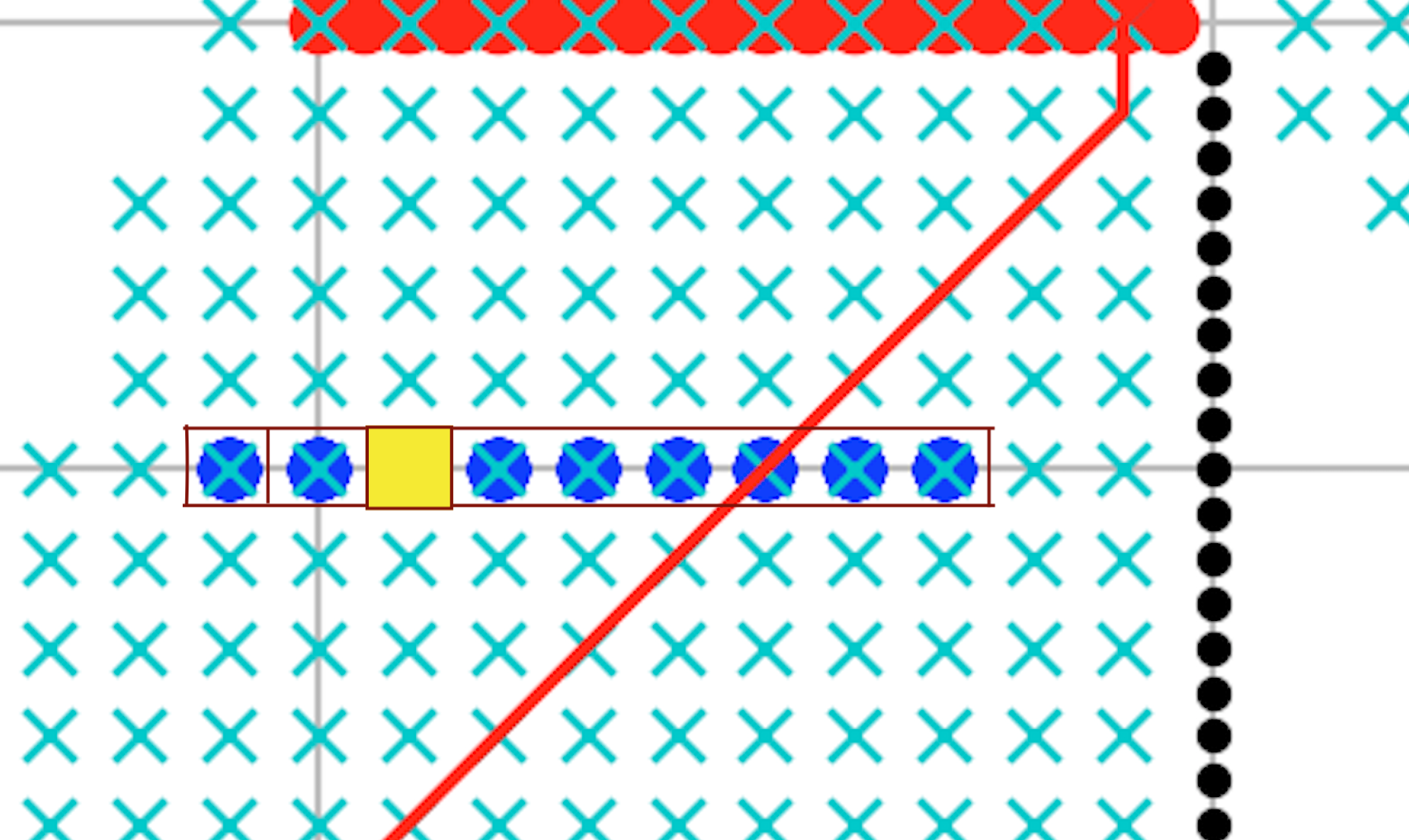
As for other cases we need to find out how many is diagonal and how many is not. But the calculation ideas are the same.

But I noticed another requirement version in our assessment slide.



It said 16 m2 is 16 grid points.

After a series of experiments, we found the actual space is composed by the yellow square shown below, that is using cross as center, two grids long on each side. So we can use the solution that we showed before directly.



## 

## 

## 

## 

## 

## 

## 

## 8. Reflective Essay (no more than one-page for each member)

## a. Member 1 (Eden)

After having few lectures about path planning by using Python and Github, I realise that coding isn't that difficult. With the help of open source software, a lot of code has been provided through the internet in different areas and I don't need to learn all the code and code it by myself. The main subject is only changing the data or the calculated equation. The code provided by the people can greatly reduce the time for understanding the code and work in an effective way. Also, I can self-learn from the code given by others.

The programming usage makes me understand that programming and coding are very crucial and useful in nowadays society. Programs store different formulas and allow people to input data. After that, the computer can output the result in a few seconds, which is much faster than manual calculation. This significantly elevates the working efficiency and can provide a better service to people. Moreover, the machine calculation can minimize the human errors .

## b. Member 2 (Tommy, Chan Tsz Kit)

This is my second time learning the programming language. The first language I have learnt is Pascal in secondary school, the concept is not a big difference but it brings me a new concept. However, when I have just received the python code, it took me some time to figure out the concept of the code and some functions. I can learn the python code via the internet.

Besides, this project is not hard for students who have not learnt the programming before. It is because we are provided with the code with some notes. We can just follow the script and change the data we want. We are not assigned to learn the language from the very beginning but we learn from the script that we do not understand. We cna check via the internet to become familiar with the key concept.

Moveover, the “path planning” term is unfamiliar to me. This lecture not only introduces this new term to me, but also teaches how important path planning is. Plan planning plays an important role in city development in the future. As the UAV technology is becoming well-developed, more drones can be used in delivering or filming. A good path planning increases their efficiency, reliability and avoids the incident.

## c. Member 3 (Jeremy Chan Yui Tin)

I have never learnt programming before having this class, so it was quite difficult for me to pick up on the first few lessons. I even find it difficult to upload the images to Github in the first lecture, which is the first very step in this course. But luckily, with my help of group mates and lecturers, I was able to figure out the problem quickly. I also learned how path planning is important to our daily lives and how convenience it brings to us.

After receiving our tasks, I think they are more complicated than what I think. I expect that what we need to do is that we need to find out the shortest distance between two points only. However, there are several tasks in the requirements. It takes quite some time for me to change the original obstacles to ours, since there are a few hundred lines in the codes. But after I find those codes, I find that those tasks are not actually as hard as I thought.

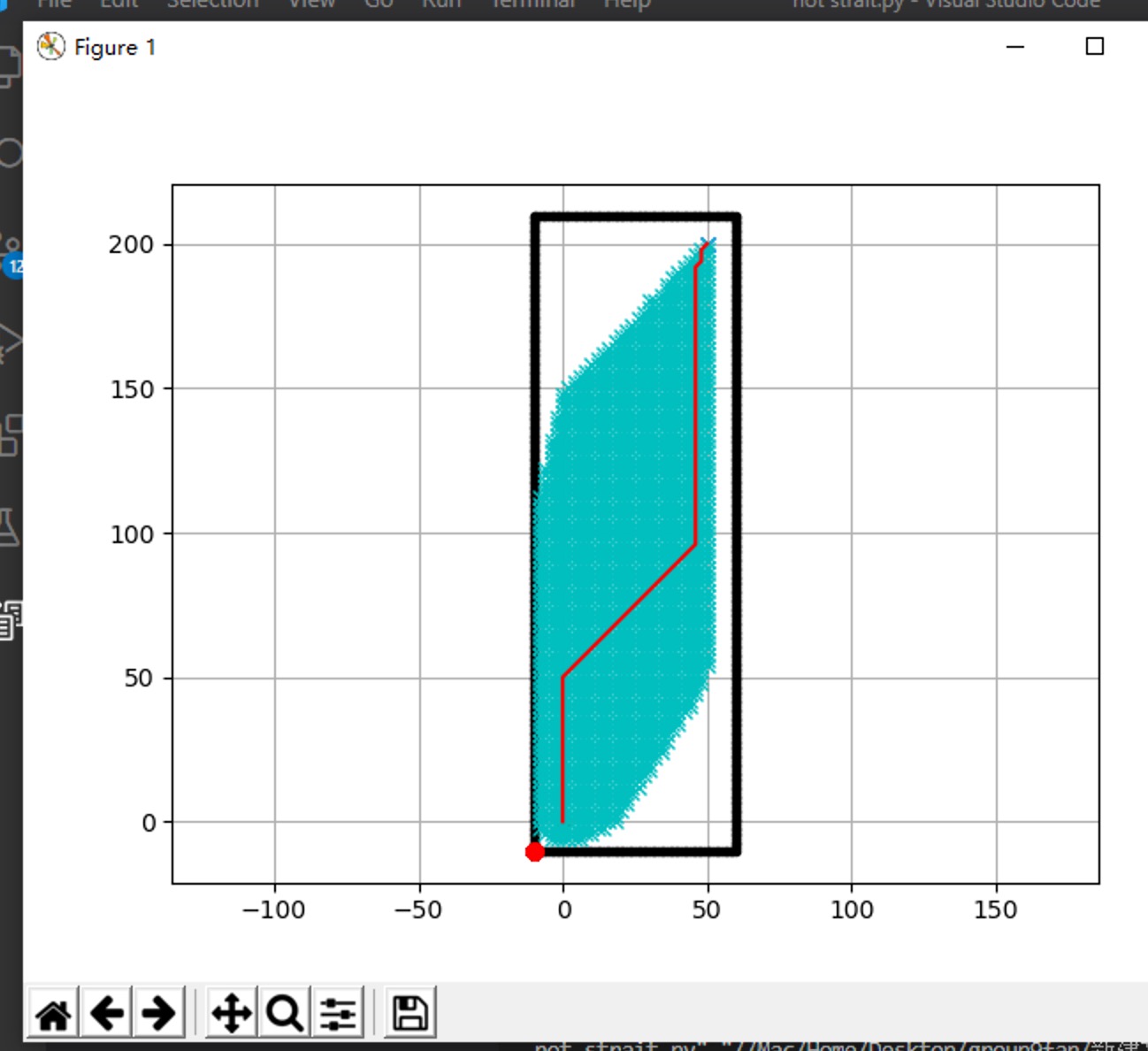
This programme is good for those who have no experience in programming, just like me, and after participating in this project, I find that I do have interest in programming and as the world relies on computing and technology more and more heavily, the importance of programming increases. We need to prepare ourselves with different skills to follow-up the changes of the world. Thanks to this class and in my free time, I may try to discover some programming instead of just playing League of Legends, one of the games which uses a path planning algorithm.

## d. Member 4 (Fan)

This is the first time I've used Python to solve practical problems. Although I had studied Python before, I realized that it was very difficult to solve practical problems. Since I am using a macbook, I meet lots of difficulties related to Programming environment. However, another two groupmates met the same situation like I did. We searched the solve methods together, discussed the problems we met. Finally, we fixed the problems we met.

This project strengthened my sense of operation. As I started to really work on the given tasks, I found it was getting pretty difficult after task2.1. I can’t clearly understand the theories behind the code. But thankfully, with the help of Burton, I get to know how the code really works and which way I should use to figure out task2.2. After working together with my groupmates, we finished all the three tasks. From this project I learned that cooperation can make the difficulties easier to solve.

In addition, we find the AStar algorithm does have some defects. For example, AStar only has eight directions but in the real world we have infinite directions to go. This fact leads to a problem that the shortest path figured by AStar is usually longer than the real shortest path. If we increase the amount of girds, this problem will still exist. Here is an example, we can easily infer that the shortest path should be a straight line but the AStar path takes a detour. Maybe the designer can deal with this problem by adding more directions as the point moves.



## e. Member 5(Burton Zhang ChengCheng)

As for me, l think the most challenging part in this project is to upload local documents to GitHub and to import matplotlib.pyplot successfully. l think it is because my laptop is mac so the original python version is too old, and it misses a lot of plug in. Anyway, I find all the solutions online.

Fortunately, after that, everything became simple. l found it not so hard to deal with the task if we get some tips. Finally, I finish all the tasks smoothly and revise a small mistake in our original code.

In this class, l am really glad that I know such a useful code platform---GitHub.

## 9. References

1. W. (2020, October 27). Motion planning. Retrieved November 10, 2020, from https://en.wikipedia.org/wiki/Motion\_planning
2. Yap, P. (2002, May). Grid-based path-finding. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 44-55). Springer, Berlin, Heidelberg.
3. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
4. Python What is Python? Executive Summary, from https://www.python.org/doc/essays/blurb/
5. "About Python". Python Software Foundation. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files."
6. "Why GitHub's pricing model stinks (for us)". *LosTechies*. November 7, 2012. Archived from the original on June 29, 2015. Retrieved June 29, 2015.
7. "The Problem With Putting All the World's Code in GitHub". *Wired*. June 29, 2015. Archived from the original on June 29, 2015. Retrieved June 29, 2015.
8. Techcrunch What exactly is Github anyway? Retrieved July 14,2012, from https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/