**Mini Project 1**

The most important parameters of simulated annealing are

1)Cost Function

I considered the initial cost function as the area of the first polish string generated using polish\_final.cpp file of my program.

2)Initial Temperature

After trying a lot of cases I discovered that an Initial Temperature of 40000 could be a good guess

3)Final Temperature

After trying a lot of cases I discovered that a Final Temperature of 0.001 could be a good guess

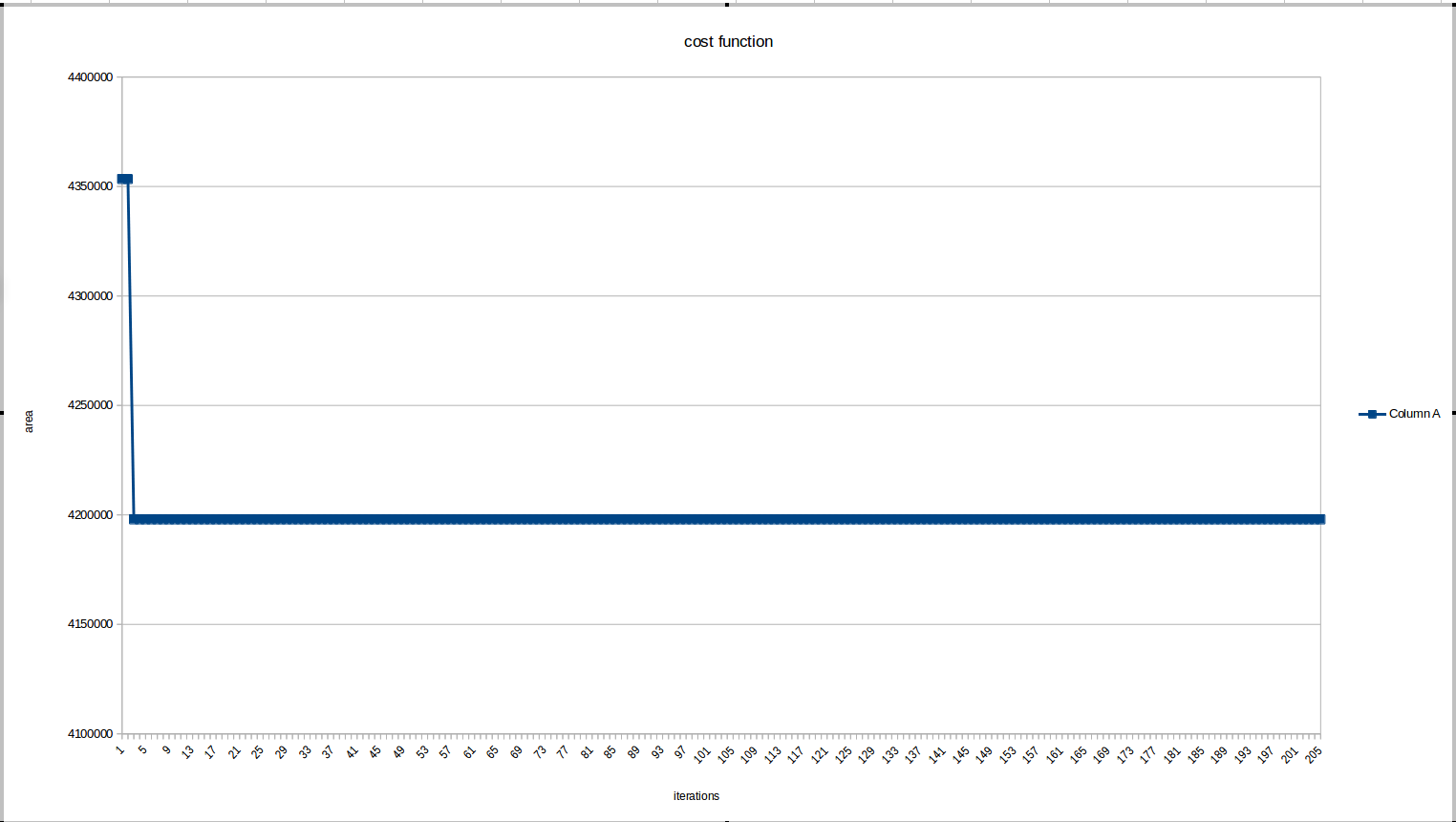
4)Number of steps per temperature and the cooling schedule

This is the "speed parameter" of the code! I initially started with constant number and then found out that using exponential functions considerably reduces the compiling time.

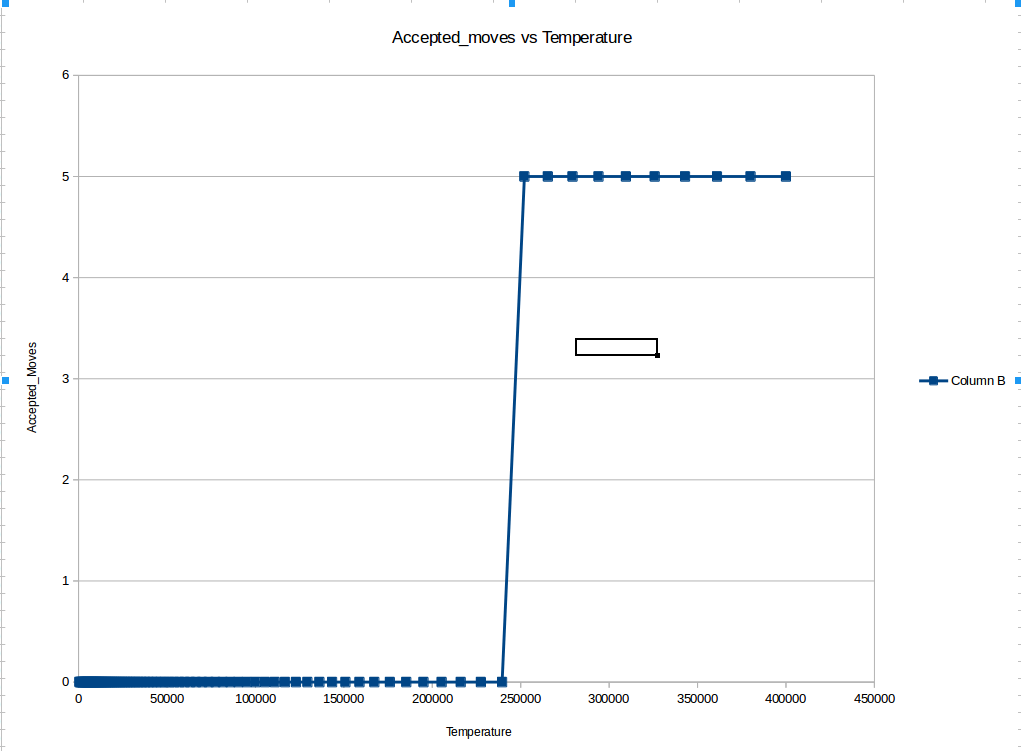
**Explanation:**

I tried various ranges of temperatures and decided to go with 40,000, with step size 30. Ideally step size should be a mathematical function, initially I took the value as (logT)(k) /(T). But with trying to fit the correct answer I ended up with this value. I did start soft blocks and have added the code I started with, but I realize there is still a long way to go. I have tried creating a modular structure for the code as I wanted to learn modular coding and wanted to add both hard and soft blocks together. My code crashed a couple of times for 100. I think that my move 3 still needs a lot of refining. I have plotted the mentioned graphs when the code worked. There are still many loop holes in my code which I intend to fix soon.

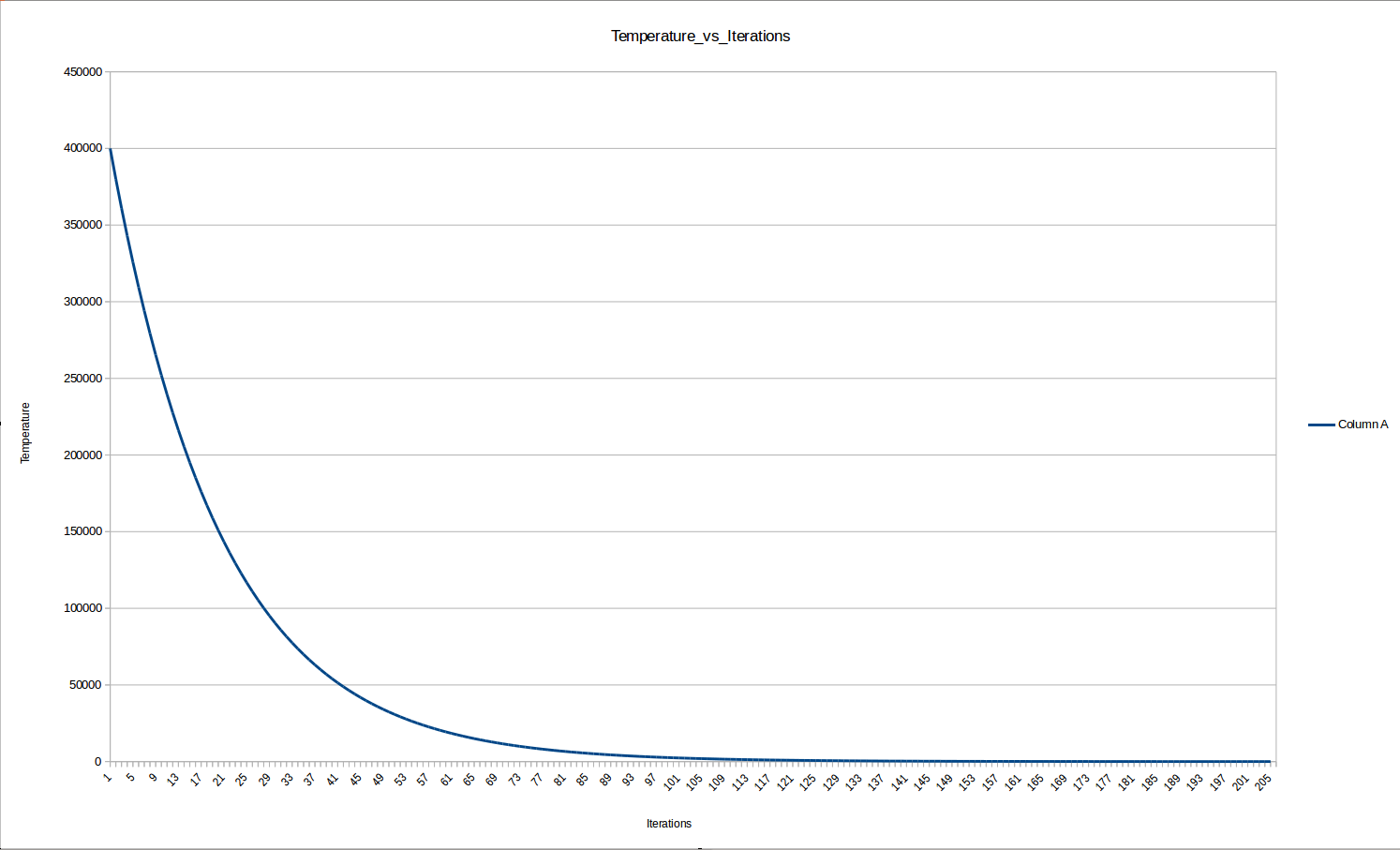
**Analysis of the graphs**



As seen in the graph it is evident that with the iterations the cost function starts becoming constant. The reason for this I believe could be that as temperature decreases , Simulated annealing starts becoming picky P(higher cost function) reduces.



I have considered 40,000 as my T and it is evident from the graph that as temperature decreases the probability of getting a move accepted reduces. Initially all moves , wrong or right are accepted but later a strict condition is applied. And only when the cost function is negative a move is accepted.



As seen in the graph, no. of iterations are a function of temperature. As temperature decreases, no. of iterations increase, implying to get your cost accepted more trials are necessary.