

### OUTPUT REPORT FOR QUESTION 3:

Average time and Standard deviation for k=20 for 5 runs

avg\_time =

18.4625

std\_deviation =

1.8321

Average time and Standard deviation for k=100 for 5 runs

avg\_time =

22.5094

std\_deviation =

1.7921

Average time and standard deviation for k=200 for 5 runs

avg\_time =

33.8062

std\_deviation =

1.1930

Average time and standard deviation for k=2000 for 5 runs (run in parallel with 2 processes so not very accurate).

avg\_time =

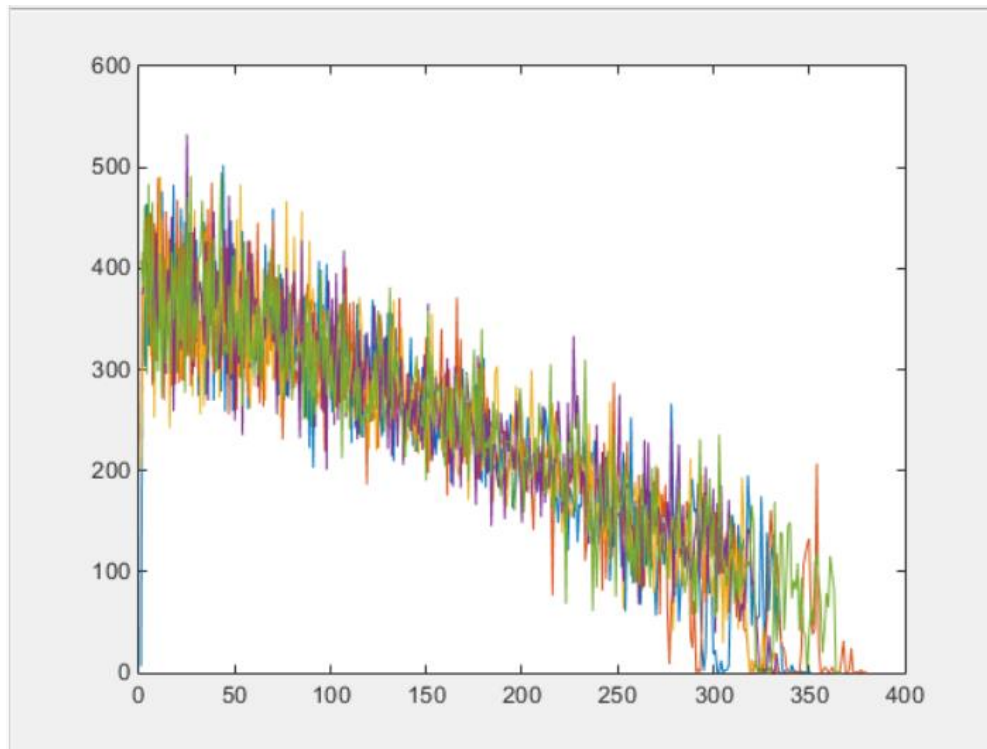
1.6550e+03

std\_deviation =

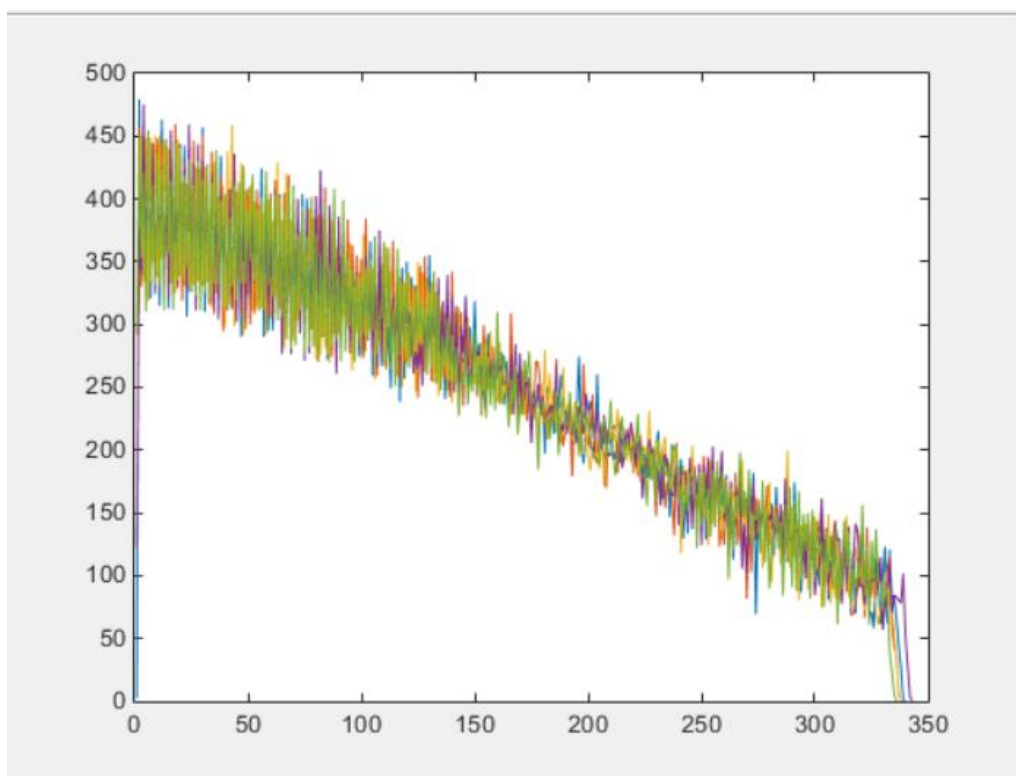
228.7532

GRAPHS FOR QUESTION 3:

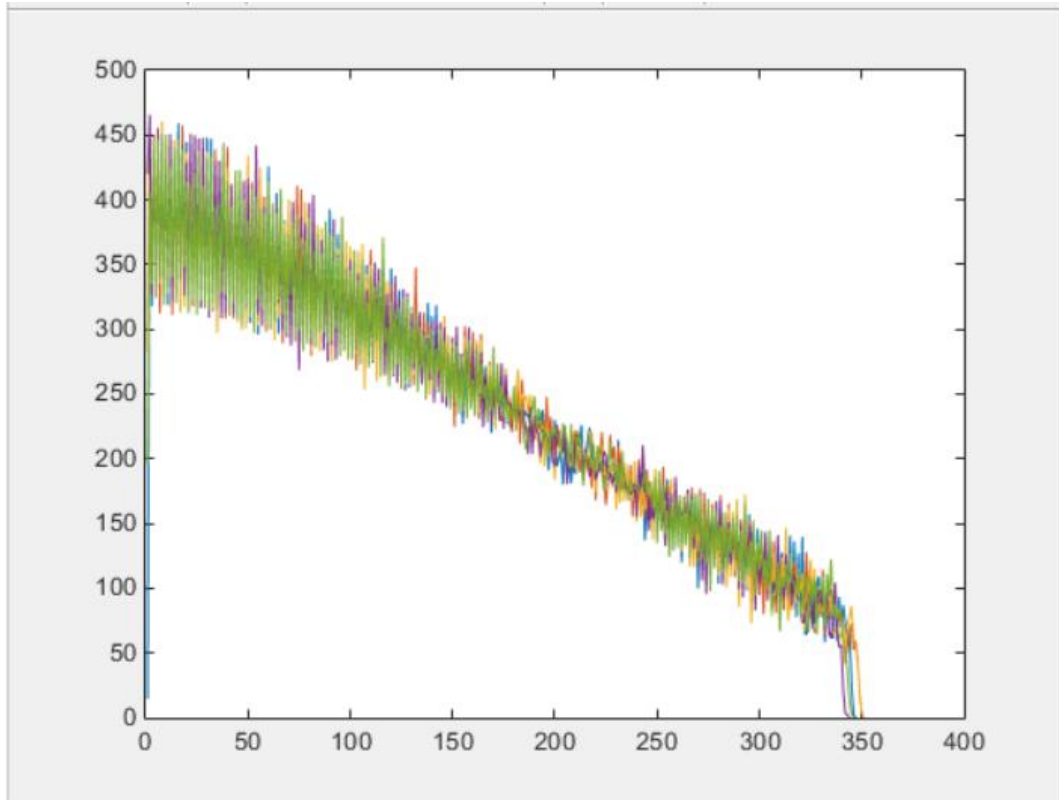
1. FOR  $K=20$  AND NUMBER OF ITERATIONS IS 5



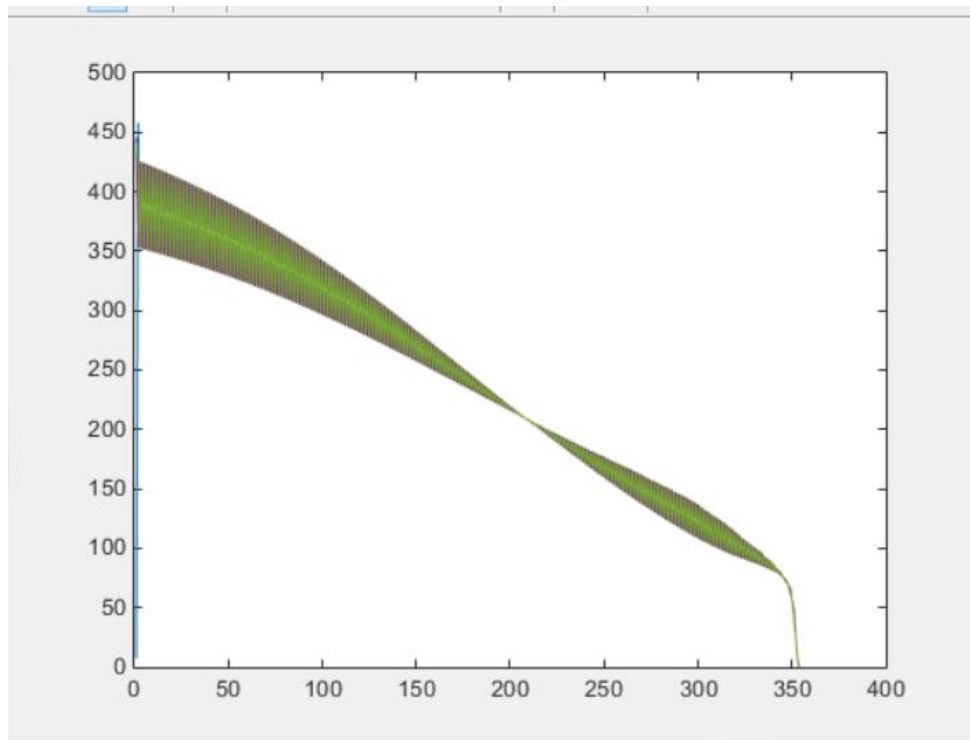
2. FOR  $K=100$  AND NUMBER OF ITERATION IS 5



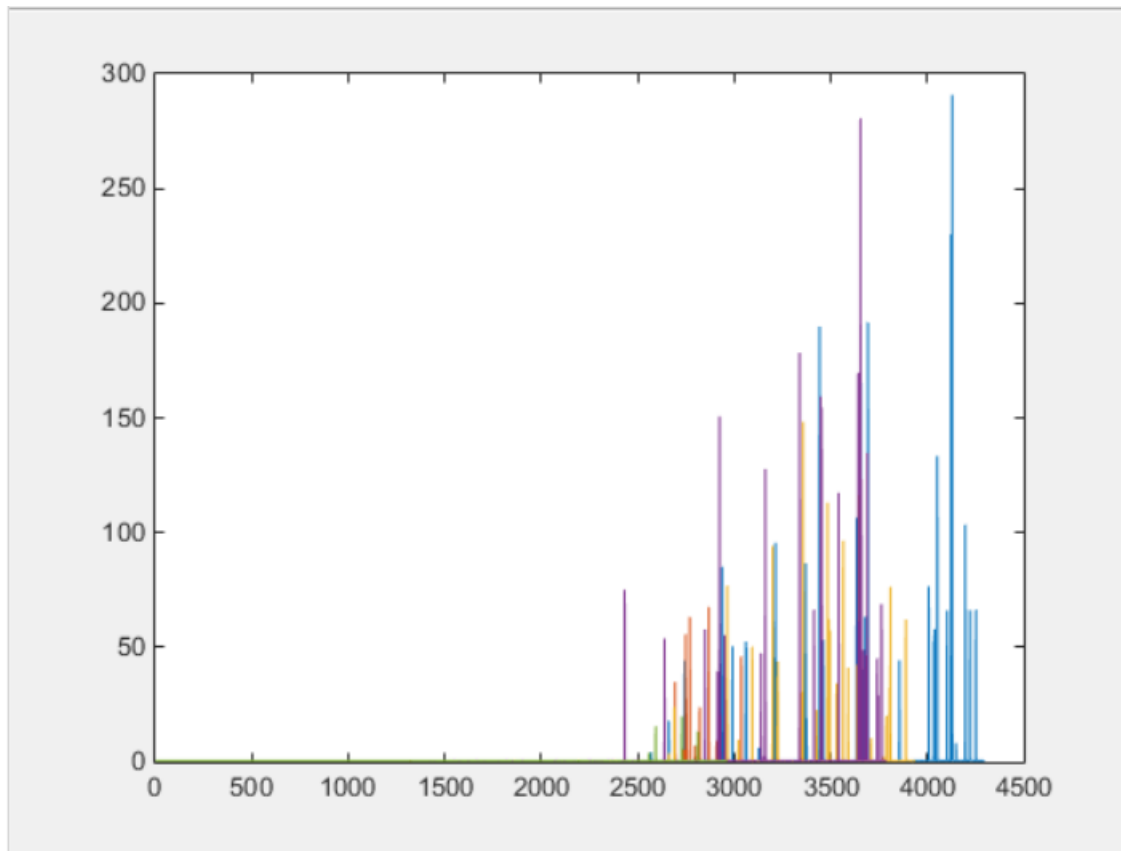
3. FOR  $K=200$  AND NUMBER OF ITERATIONS IS 5



4. FOR  $K=2000$  AND NUMBER OF ITERATIONS IS 5



5. FOR K=1 AND NUMBER OF ITERATIONS IS 5



QUESTION 2:

ASSUMPTIONS :

1 . It was observed from the graph that the objective function stabilizes after a certain value of l iterations it was noted that the values were almost stable after 500 iterations. And hence the objective functions and the number of iterations are plotted only for 500 itearations.

The objective function increases the time of computation or the time taken to run the program. The main aim of the problem is to understand the fast working of the algorithm. By calculating the objective function for 500 values we can appreciate the working of the algorithm and at the same time observe that the objective function is increasing and hence we can know that we are approaching the maximum value that the function can attain.

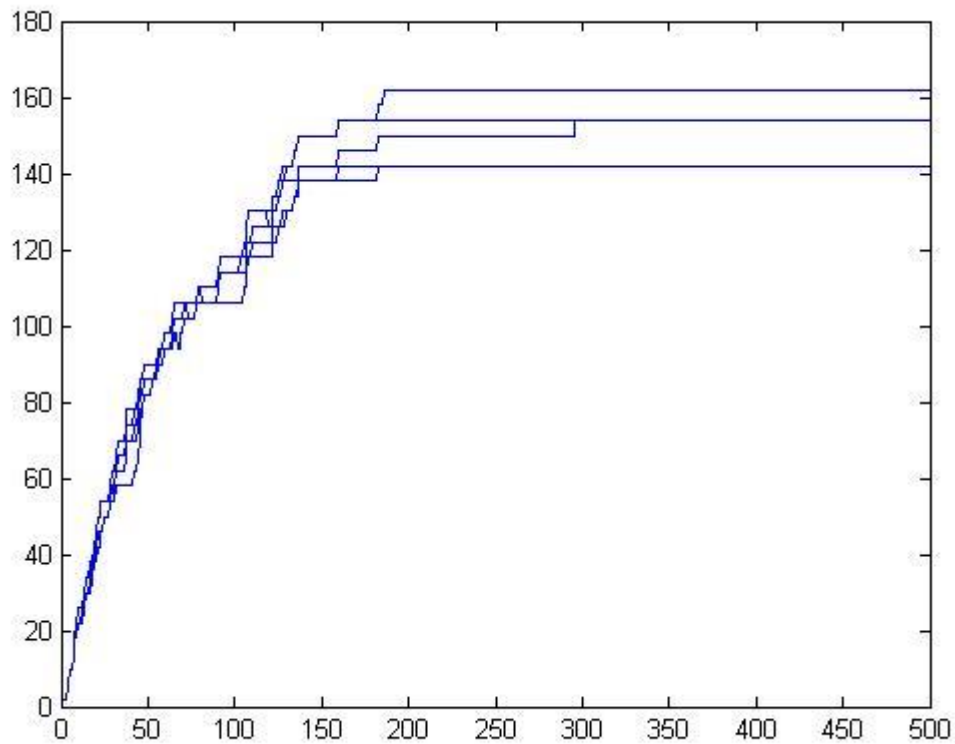
Thus,the assumption made to run the program is a plausible assumption.

2. The kernel calculations ar seperated and stored as a matrix. This also decreases the run time required.

3.The multiplication for 'y' is also done seperately and then it is used for the calculation of objective function.

4. Since, these values also contribute to the runtime they are also included as a part of the runtime and since we run them only once in the loop the time taken to run the first iteration is comparatively greater than all the other iterations and **this explains the standard deviation value obtained.**

GRAPH:



avg\_time =

1.9758e+03

std\_deviation =

72.6073