

Keyboard Optimisation

Madhav Soorya Tadepalli ee23b040

October 2024

Note:

Any given file for the layout should be named 'layout.py' and be present in the directory that the main script('keyboardOptimisation.py') is run in.

Objective

Develop a Python program that analyzes keyboard usage patterns for a given text input and a keyboard layout file.

The program should optimise the keyboard layout using Simulated Annealing. Then, generate a Heatmap key usage and calculate the total distance traveled by fingers while the layout is being optimised.

Procedure

This assignment is a follow-up of Assignment 4, so the following procedure is the modifications/add-ons to the code from the previous assignment.

I first converted all the blocks of code into functions (most were already packaged in function, but some of it wasn't) and had to add the parameters 'keys' and 'characters' to each of them as they could no longer be global variables.

I had to make a function called 'get_neighbours' which swapped any 2 random keys to create a neighbouring keyboard layout. In doing so, I had to make sure:

1. All the special characters associated with that key were also swapped.
2. If the swapped key was on the home row, the corresponding home row key of all other keys must be updated

Then, I made the main 'simulated_annealing' function which was for the most part similar to the example code given in moodle for TSP. I made changes to suit this problem.

The choice for 'initial_temp', 'cooling_rate' and 'num_iterations' were purely trial and error, I used what seems to work well for mid-length strings. I noticed that different values worked better for different length strings.

Output

The following is for a random example string I chose, this can be changed by changing the variable 'para' in the file 'keyboardOptimisation.ipynb'

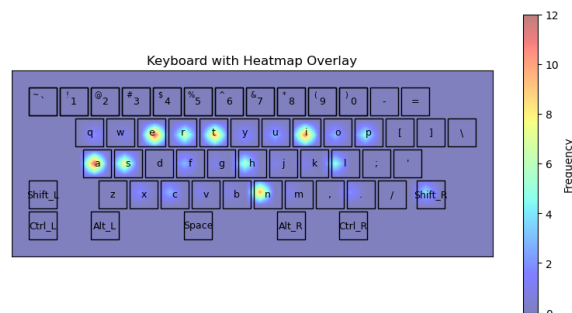


Figure 1: qwerty Layout Heatmap

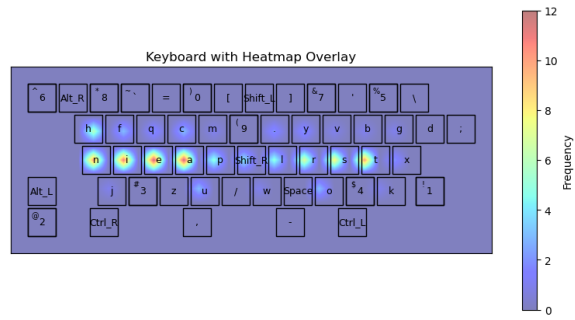


Figure 2: Optimised Layout Heatmap

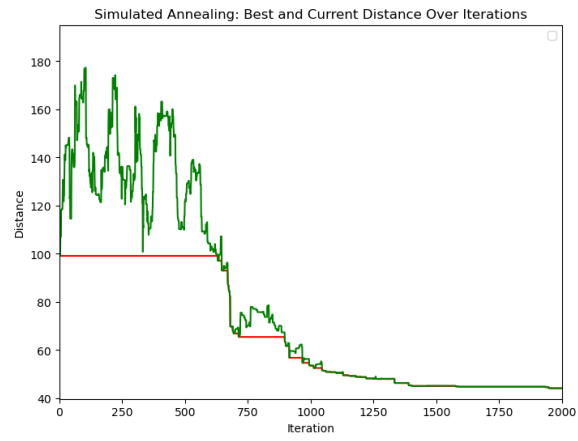


Figure 3: Distance Plot