Assignment 4: Keyboard Optimization
REPORT

Dhruv Prasad

October 2024

1 Introduction

Python program to optimize the keyboard layout for a given string of text by reducing the total finger travel distance during typing using *simulated annealing*.

2 Usage

- Extract **ee23b130.py**, **kbd_layout.py** and **my_kbd_plotting.py** from my .zip submission and save them to the same directory
- Run the script with **python ee23b130.py**
- A file **ee23b130_a5.png** will be created, open it to see the result if it doesn't display automatically. Please view it in full screen!

Note: To test my code with a different layout, either modify my layout file or replace it with another in the same format with the same file name.

Note: To test my code with a different text input, modify the sample_text string in my script.

2.1 Assumptions

- The positions of the keys must have integral y-coordinates for the keys to not overlap, as I have assumed key height to be 1. On the other hand x-coordinates can be anything. *Note: The image will still be generated*

- The string may have any character in it but unknown characters will be ignored.
  Note: I have modified the default layout to have an 'Enter' key and I have handled newlines in the sample text

3 Methodology

3.1 Optimization Technique: Simulated Annealing

The basic steps for simulated annealing in this problem are:

- Start with the initial QWERTY keyboard layout.
- Generate a new keyboard layout by swapping two randomly selected keys.
- Calculate the total finger travel distance for the new layout.
- If the new layout gives a lesser typing distance, accept it.
- If the new layout gives a lesser typing distance, accept the new layout with a probability proportional to the current temperature.
- Gradually reduce the temperature by multiplying by a cooling rate and repeat the process for a predefined number of iterations.

3.2 Plotting and Visualization

- I use my functions for drawing keyboard and heatmaps from the previous assignments to plot the intial and final keyboards. These functions are defined in **my_kbd_plotting.py**
- I plot the distances of keyboard layouts tested in each of the iterations along with the lowest distance encountered until then. This helps to show how the total distance cost decreases as I optimize.

4 Results and Analysis

The simulated annealing algorithm was run for 500 iterations, with the initial temperature set to 1000 and a cooling rate of 0.998. The layout was optimized for the following input text:

"Heat maps originated in 2D displays of the values in a data matrix. Larger values were

represented by small dark gray or black squares (pixels) and smaller values by lighter squares. Sneath (1957)

displayed the results of a cluster analysis by permuting the rows and the columns of a matrix to place similar

values near each other according to the clustering. Jacques Bertin used a similar representation to display data

that conformed to a Guttman scale. The idea for joining cluster trees to the rows and columns of the data matrix

originated with Robert Ling in 1973. Ling used overstruck printer characters to represent different shades of gray,

one character-width per pixel."

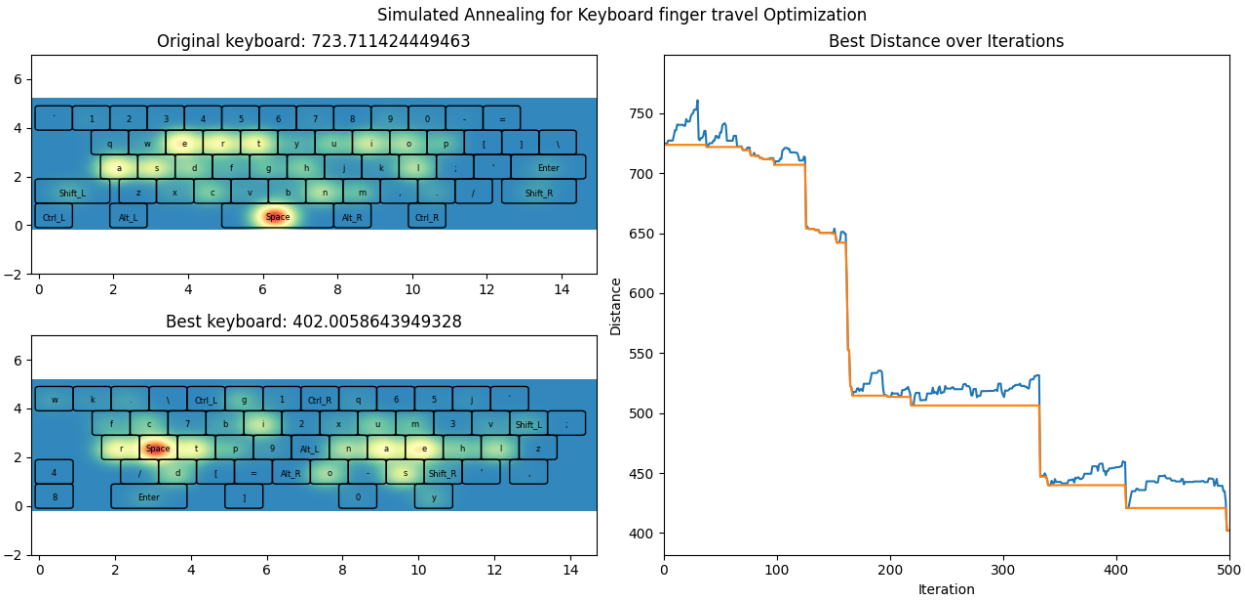

Figure 1: Example Output

5 Conclusion

In this assignment, we successfully applied simulated annealing to optimize a keyboard layout for a specific input text. The algorithm explored various layouts by making small random swaps and effectively minimized the total finger travel distance over several iterations.

This approach can be extended to optimize layouts for different input texts.