Assignment 5 Report

EE23B039 A Lekhasree

Introduction

This report outlines the approach taken to analyze keyboard layouts for typing efficiency by calculating finger travel distances using a simulated annealing algorithm. The primary goal is to optimize the layout of keys to minimize the distance fingers travel while typing.

Keyboard Layout Representation

The QWERTY keyboard layout given in updated assignment 4 is used here, where each key is represented by its character and its position (x, y).

Assumption and Home Row Keys Explanation

I considered that the finger returns back to home row keys after typing each letter, so the distance is multiplied by 2. For this, I fixed the coordinates of home row keys, and the distance of a specific key is measured as the distance from the nearest home key position. A function is written to find the coordinates of the nearest home row key and the minimum distance.

Distance Calculation

Everywhere in this code, distance is calculated using the Euclidean distance formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the keys.

Input Processing

The user inputs a string, and the frequency of each character is counted. This frequency information is used for calculating the total distance traveled when typing the string. A dictionary is created that shows the number of times a particular key is pressed on the keyboard.

Simulated Annealing Algorithm

The simulated annealing method is employed to find an optimized keyboard layout:

- Copies the initial layout to current_layout.
- Calculates the initial distance using calculate_total_distance().
- Sets best_layout and best_distance to track the best found layout.
- Initializes distances list to store distances for each iteration.
- Sets the initial temperature to initial_temp.
- Runs for max_iterations.
- Creates a new layout by randomly swapping two keys from the current layout.
- Uses random.sample() to select two different keys to swap.

- Calculates new_distance for the new layout.
- Checks if new_distance is less than current_distance.
- If yes, accepts the new layout.
- If no, keeps the previous layout only
- Updates best_layout and best_distance if the new distance is the best found.
- Appends current_distance to the distances list.
- Decreases temperature by multiplying it with cooling_rate.
- Returns best_layout, best_distance, and the list of distances for analysis.

Parameters

The following parameters are used for the simulated annealing:

- Initial temperature: 1000
- Cooling rate: 0.99
- Maximum iterations: 2000

Results

The algorithm generates a log of distances over iterations, allowing us to visualize the optimization process.

Sample text input:

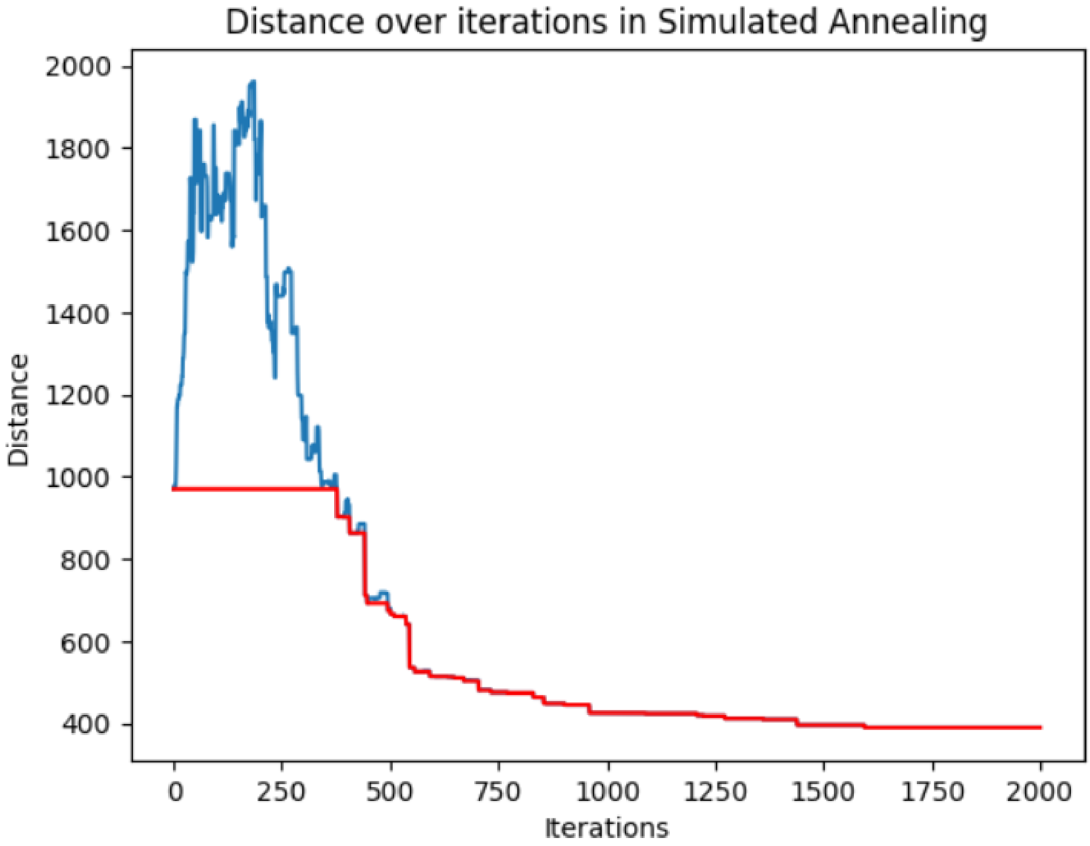
The environment is the basic life support system for all living things on planet Earth. It is a combination of natural and human-made components. Natural components include air, water, land and living organisms. Roads, industries, buildings, etc., are human-made components. The natural environment can be differentiated into four main components – Biosphere, Lithosphere, Hydrosphere and Atmosphere. The topmost layer of the Earth is called the Lithosphere, which is a thin layer of soil made of rocks and minerals. The hydrosphere consists of various types of water bodies like seas, oceans, rivers, lakes, ponds, etc. Atmosphere, consisting of water vapour, gases and dust particles, is the layer of air that surrounds the Earth. The living world consisting of human beings, plants and animals constitute the biosphere.

Issues:

It gives different layouts for the same input text when run different times. This is because sampling is done randomly and for each random pick the layout is different

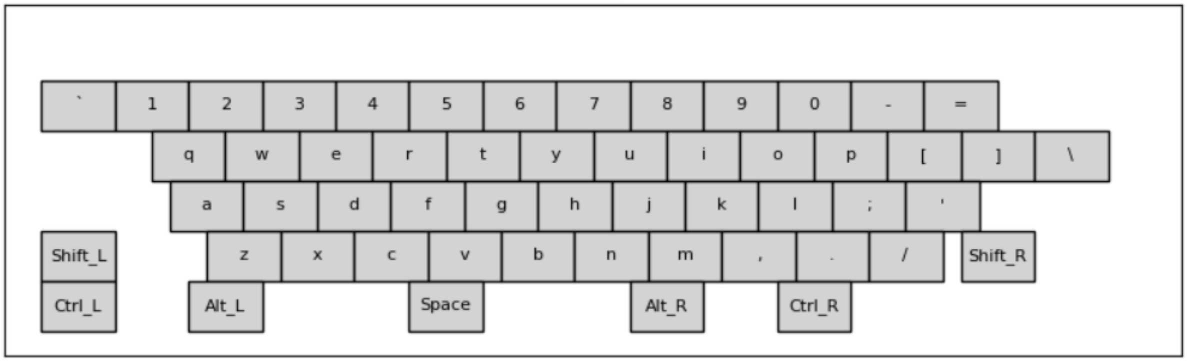
Output Screenshots

```
Enter a string: The environment is the basic life support system for all living thin
Total finger travel distance: 970.3263954683597
Best layout found with distance: 389.98540446292265
```

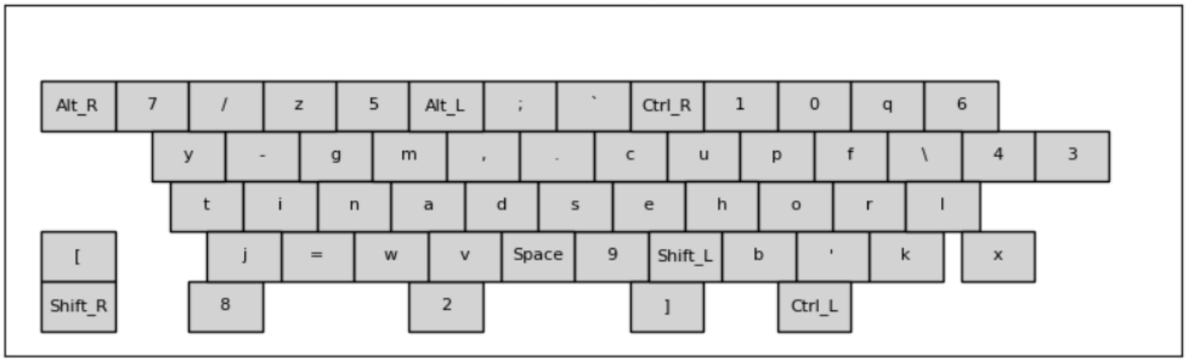


```
Initial and Optimized Layouts:
```

Initial Layout


` 1 2 3 4 5 6 7 8 9 0 - =
q w e r t y u i o p [ ] \
a s d f g h j k l ; '
Shift_L z x c v b n m , . / Shift_R
Ctrl_L Alt_L Space Alt_R Ctrl_R

Optimized Layout


Alt_R 7 / z 5 Alt_L ; ` Ctrl_R 1 0 q 6
y - g m , . c u p f \ 4 3
t i n a d s e h o r l
[j = w v Space 9 Shift_L b ' k x
Shift_R 8 2] Ctrl_L