Optimizing Keyboard Layout Using Simulated Annealing

Your Name

October 13, 2024

Abstract

This report details the optimization of a QWERTY keyboard layout using simulated annealing. The goal of the optimization process is to minimize the total distance a user's fingers must travel while typing a given text. By iteratively swapping keys and employing a probabilistic acceptance of suboptimal solutions, a more efficient layout is obtained, resulting in a significant reduction in typing distance.

1 Simulated Annealing Algorithm

Simulated annealing is a probabilistic technique used to approximate the global optimum of a given function. The algorithm is inspired by the physical process of annealing, where materials are heated and then cooled slowly to remove defects and optimize their structure. In optimization, simulated annealing is useful for exploring a large search space and avoiding being trapped in local minima.

The key parameters of the algorithm are:

- **Initial Temperature:** A high starting temperature that allows exploration of the solution space.
- **Cooling Rate:** The rate at which the temperature decreases. A slower rate allows more extensive exploration but increases computational cost.
- **Iterations:** The number of swaps and evaluations performed during the optimization.
- **Acceptance Criterion:** Solutions that reduce the objective cost are always accepted. However, worse solutions are also accepted probabilistically based on the temperature, allowing the algorithm to escape local minima.

1.1 Objective Function: Distance Calculation

The objective function in this problem is the total distance a user's fingers must travel while typing the input text. The distance between two consecutive key presses is calculated using the Euclidean distance formula:

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

where (x_1, y_1) and (x_2, y_2) are the coordinates of two consecutive keys on the keyboard layout.

For each character in the input text, the distance between the corresponding keys is calculated, and these distances are summed to give the total typing distance. The aim is to reduce this total distance through the iterative swapping of keys.

2 Optimization Process

The optimization process begins with the QWERTY layout as the initial solution. In each iteration, two keys are randomly selected and their positions are swapped. The new layout is evaluated by calculating the total typing distance for the given text.

If the new layout improves the objective function (i.e., reduces the total typing distance), the swap is accepted. If the new layout is worse, it may still be accepted with a probability proportional to the current temperature and the increase in cost. This acceptance criterion allows the algorithm to escape local minima and explore a wider range of solutions.

The temperature is gradually reduced according to the cooling schedule:

$$T = T_0 \times \alpha^t$$

where T_0 is the initial temperature, α is the cooling rate, and t is the current iteration number.

3 Results

3.1 Initial and Optimized Layouts

As seen in the optimized layout, certain commonly used characters in the text have been moved closer together as compared to the qwerty layout, resulting in a reduction in the total finger travel distance.

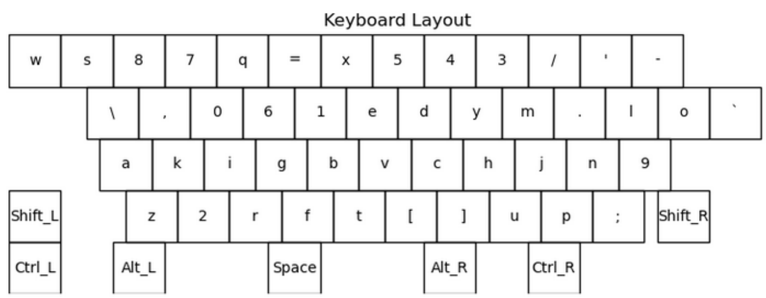

Figure 1: Optimized layout after simulated annealing.

3.2 Cost and Temperature Progression

Figure 2 shows the progression of the total cost (typing distance) and temperature over the course of the optimization. As expected, the cost gradually decreases as the layout improves, while the temperature decreases according to the cooling schedule.

3.3 Final Distance Reduction

The initial total typing distance for the input text was found to be 193.23 units. After 10,000 iterations of simulated annealing, the total distance was reduced to 146.46 units, representing a 24.20% improvement in typing efficiency. This demonstrates the effectiveness of simulated annealing in optimizing keyboard layouts for specific input texts.

4 Discussion and Conclusion

The simulated annealing approach successfully reduced the total typing distance by re-arranging the positions of the keys. By allowing suboptimal moves early in the process and gradually narrowing the search, the algorithm was able to escape local optima and find a globally better solution.

It should be noted that the optimized layout is specific to the input text. Different texts with different character frequencies may result in different optimized layouts. This suggests that custom keyboard layouts could be designed for individual users based on their most commonly typed words or phrases, potentially improving typing speed and reducing finger strain.

The program should be run using the command: `python3 ee23b008.py`

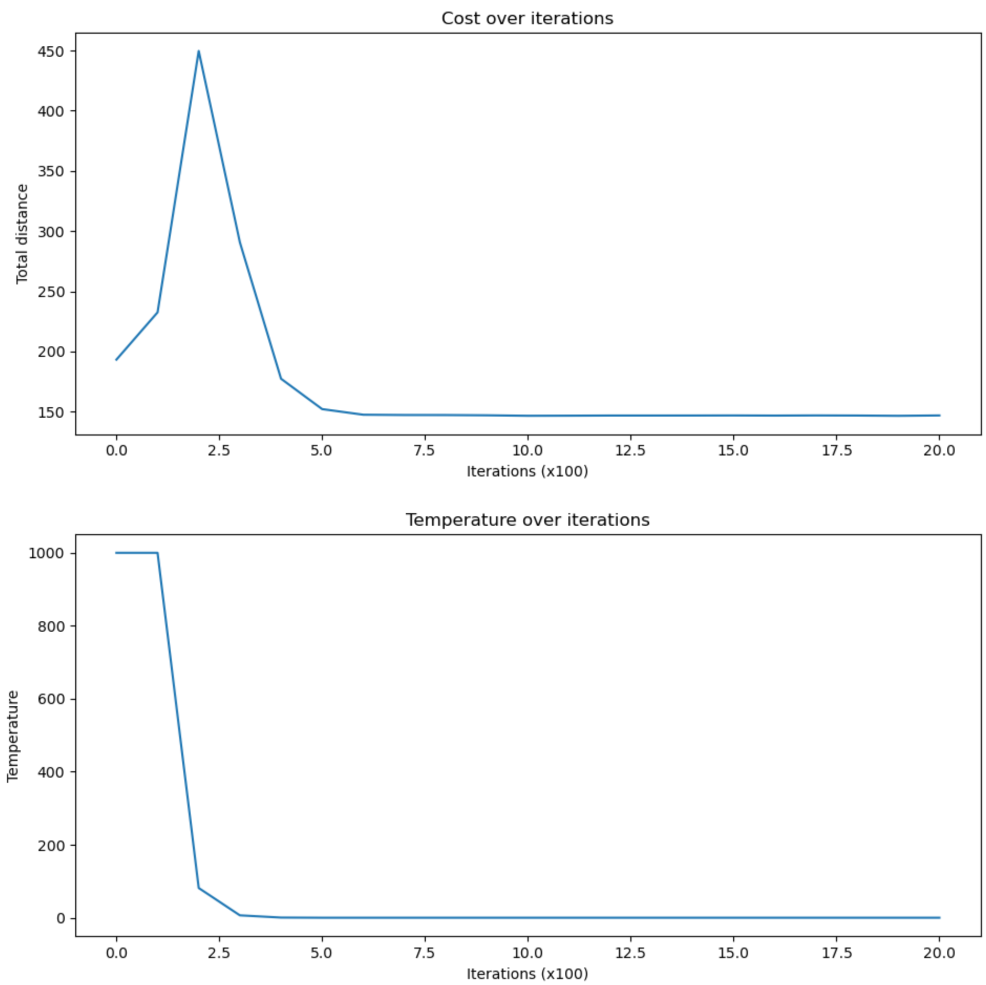

Figure 2: Cost and temperature progression during optimization.