Strategic Cell Placement in EDA: Harnessing the Power of Simulated Annealing

Mohamed Noureldin - 900203758
Yaseen Ahmed - 900201235
Ziad Miniesy - 900202283

Dr. Mohamed Shalan

Dec 11, 2023

1. Introduction

This project is aimed at implementing a placement optimization algorithm for electronic design automation (EDA). The primary objective is to optimally place cells on a grid to minimize the Half Perimeter Wire Length (HPWL). This is achieved through a simulated annealing algorithm, a probabilistic technique to approximate the global optimum in a large search space.

The report covers detailed explanations of the project's components, including file parsing, data structure initialization, grid representation, the simulated annealing process, and GIF generation for visualization.

2. Project Overview

Core Components

1. **Data Structures:**
    a. **Cells:** Represent electronic components with a unique ID and their grid coordinates.
    b. **Nets:** Define connections between cells, holding information about connected cells and their spatial bounding box.
2. **Grid Representation:**
    a. A two-dimensional grid where each cell can be placed. The grid dimensions are determined from the input file.
3. **File Parsing:**
    a. Extracts vital information from a netlist file about the number of cells, nets, and grid dimensions, setting the stage for the optimization process.
4. **Placement Strategy:**
    a. Initially, cells are placed randomly on the grid. This placement serves as the starting point for the optimization process.
5. **Optimization Algorithm - Simulated Annealing:**
    a. Employed to iteratively improve the placement. It involves making swaps between cells and evaluating the impact on the overall wiring length (HPWL).
6. **Visualization:**
    a. The CImg library is used for generating images of the grid at different stages, offering a visual representation of the placement optimization process.

Key Functions

1. **Initial Random Placement:**
    a. Cells are initially placed at random positions on the grid.
2. **Swap Mechanism:**
    a. Cells are swapped to explore different configurations, seeking a layout that minimizes the wiring length.
3. **HPWL Calculation:**
    a. The Half Perimeter Wire Length (HPWL) for each net is computed to quantify the wiring length and guide the optimization.
4. **Simulated Annealing:**

    a. A probabilistic technique that intelligently explores the placement space, balancing between exploring new configurations and improving current ones.

5. **Visualization and Output:**
    a. Generation of grid images and a final GIF to visualize the progression of the placement optimization.

3. File Parsing

Overview

The file parsing process reads a netlist file to extract information about cells, nets, and grid dimensions, vital for initializing the grid and data structures.

Pseudocode

Parse Netlist File:

```
parseNetListFile(fileName):
   Open fileName
   Read firstLine
   parseNetListFirstLine(firstLine)
   For each line in file:
       parseNetListNetLine(line)
   Close file
```

Parse the First Line:

```
parseNetListFirstLine(firstLine):
   Split firstLine into components, nets, rows, columns
   Initialize grid with rows and columns
   Initialize cells and nets
```

Parse Net Lines:

```
parseNetListNetLine(line):
   Split line into net components
   Update net and cells with connections
```

Example Explanation

Given a sample netlist file:

```
3322 3012 220 212
```

1. **Line 1 ("3322 3012 220 212"):**
    a. Sets 3 cells, 3 nets, and a 2x2 grid.
    b. Initializes structures for cells, nets, and grid.
2. **Subsequent Lines:**
    a. Line 2 ("220"): Connects cells 0, 1, 2 to the first net.
    b. Line 3 ("212"): Connects cells 2, 0 to the second net.
    c. Line 4 ("212"): Connects cells 1, 2 to the third net.
3. **Error Handling:**
    a. Checks for file opening issues and incorrect line formats.
    b. Exits the program if errors are detected, with an error message.

4. Initial Placement Strategy

Overview

The initial placement of cells on the grid is done randomly. This randomization provides a starting point for the optimization process.

Process

Random Placement Algorithm:

```
placeInitiallyRandom():
   For each cell:
       Generate random coordinates within grid dimensions
       If the chosen position is empty:
           Place cell at these coordinates
           Mark position as occupied
       Else:
           Find another position
```

5. Swap Mechanism

The swap mechanism is a core part of the simulated annealing process, allowing the exploration of different cell configurations on the grid.

Types of Swaps

1. **Swapping Two Cells:**
    a. Cells at different positions are exchanged.
    b. Adjustments are made in the grid to reflect the new positions.
2. **Swapping Cell with Empty Space:**
    a. A cell is moved to an empty position, leaving its former position vacant.
    b. Useful for exploring configurations where cells are more spread out.

Swap Algorithm

```
swapCells(cell1, cell2):
    Save original positions of cell1 and cell2
    Exchange positions of cell1 and cell2
    Update grid to reflect new positions
    Recalculate HPWL for affected nets
```

Importance

- The swap mechanism enables the simulated annealing algorithm to effectively navigate the solution space, moving towards configurations with lower HPWL.
- It balances between making small, local changes and more significant, global rearrangements of the cells.

6. Simulated Annealing Algorithm

Function Explanation

1. **Initial Setup:**
    a. **Initial Temperature:** Set high to allow exploration of the solution space.
    b. **Final Temperature:** Set low to gradually focus on refinement rather than exploration.
    c. **Temperature Reduction:** The cooling rate determines how quickly the temperature decreases.
2. **Temperature's Role:**
    a. High temperatures allow the algorithm to accept worse solutions (increased HPWL) for exploration.
    b. As the temperature decreases, the algorithm becomes more selective, favoring solutions that improve or maintain HPWL.
3. **Iteration Process:**
    a. At each temperature level, a set number of moves (swaps) are tried.
    b. Each move's impact on HPWL is calculated.
4. **Move Acceptance:**
    a. If a move reduces HPWL, it's accepted.

    b. If HPWL increases, the move is accepted with a probability depending on the current temperature and the HPWL change. This probability decreases as the temperature drops.

5. **Cooling Down:**
    a. After a set number of moves at a given temperature, the temperature is reduced.
    b. The process continues until the final temperature is reached.

Simulated Annealing Algorithm in Pseudocode

```
simulateAnnealing(initialHPWL, coolingRate):
    Set currentTemperature to a high value
    Set finalTemperature to a low value
    While currentTemperature > finalTemperature:
        For each move in a predefined number of moves:
            Perform a swap
            Calculate new HPWL
            If HPWL is improved:
                Accept the swap
            Else:
                Calculate acceptance probability
                Accept swap based on probability
        Reduce currentTemperature by coolingRate
```

7. Output Analysis of the Simulated Annealing Process

Initial and Final Grid States

- **Initial Grid:** Shows a random placement of cells ('numbered') and empty spaces ('----'), reflecting the starting point of the optimization process.
- **Final Grid:** Demonstrates a more optimized arrangement with cells often clustered and more empty spaces along the edges.

Empty Cells on the Edges

- This pattern arises as the algorithm moves cells closer together to minimize the total wire length, especially for connected cells, leaving the peripheries less occupied.

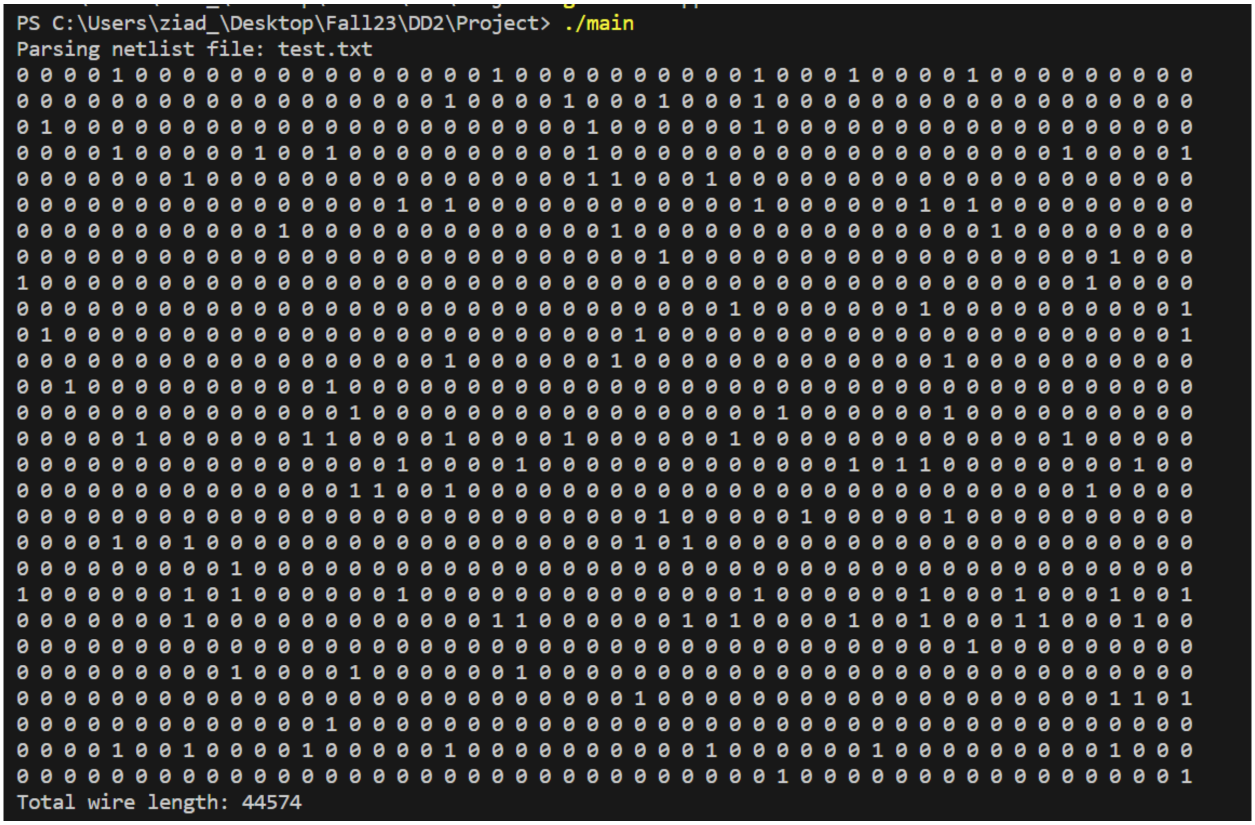
Changes in Wire Length

- The output displays a significant reduction in wire length from the initial to the final state, showcasing the effectiveness of the simulated annealing in optimizing the grid layout for shorter wiring.

Execution Time and -O3 Flag

- The 8-second execution time illustrates the efficiency of the algorithm, enhanced by the -O3 optimization flag in g++, which accelerates computation-intensive tasks in the annealing process.

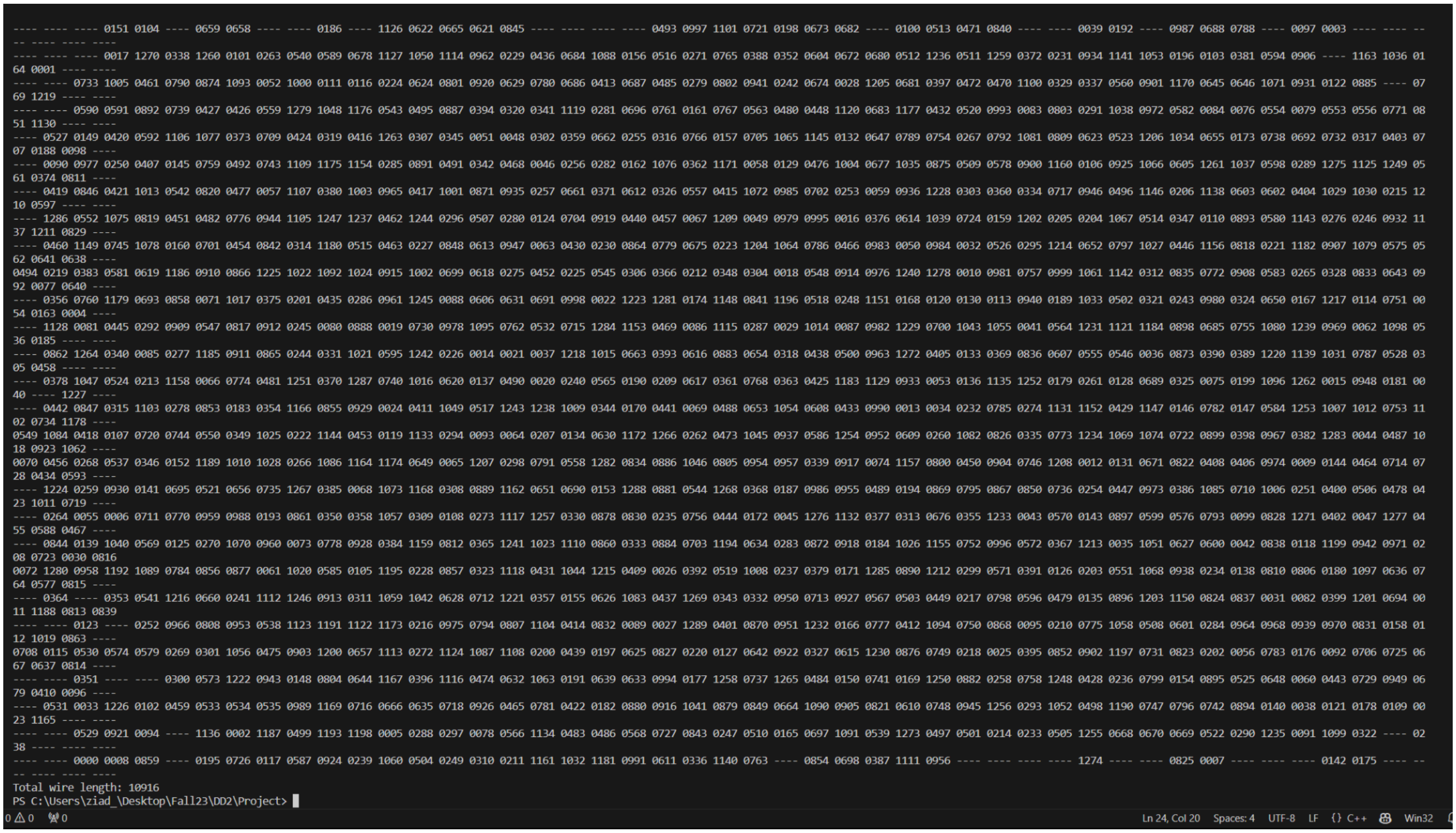
Output Example



```
PS C:\Users\ziad_\Desktop\Fall23\DD2\Project> ./main
Parsing netlist file: test.txt
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Total wire length: 44574
```

- The following output is the output of simulated annealing on the netlist file t3.

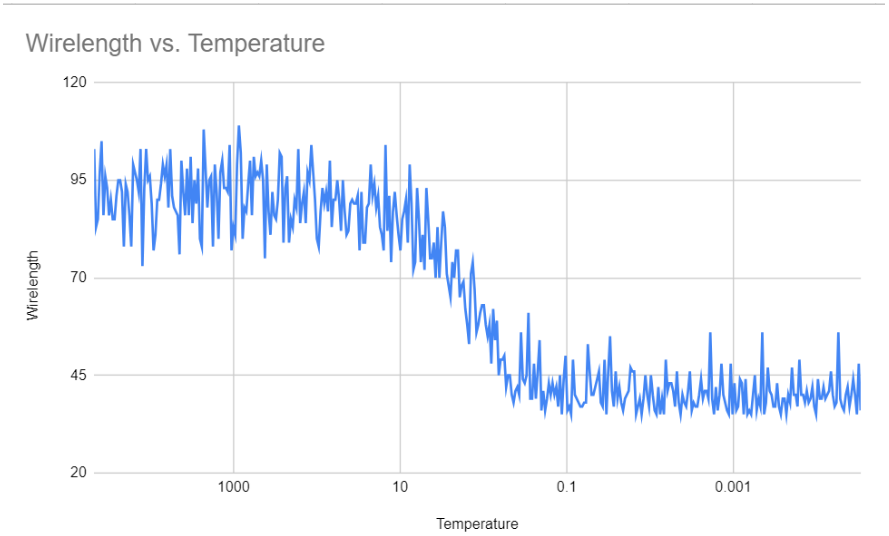
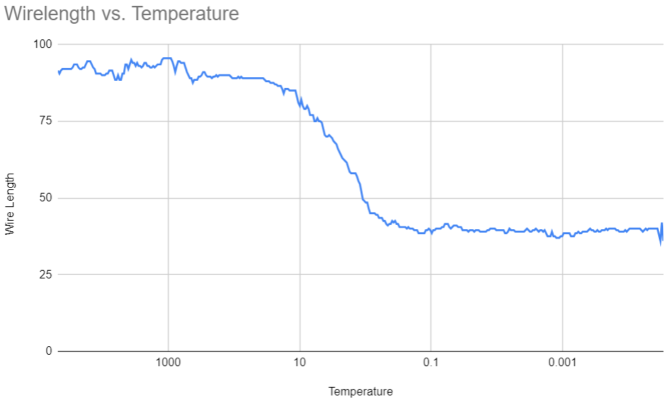
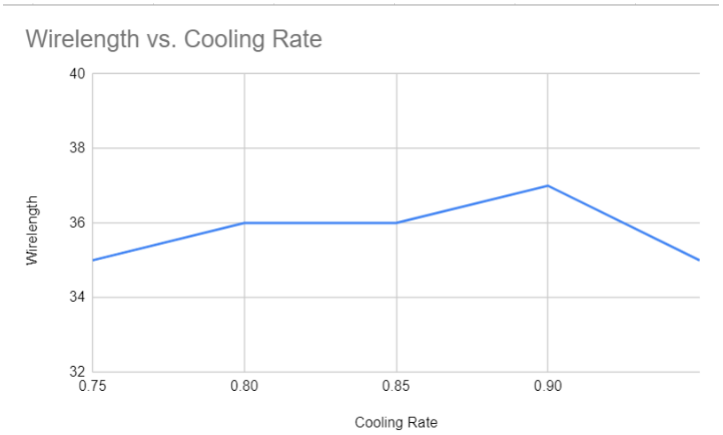
  This is the initial random placement of the grid. 1 represents a cell containing a block and 0 represents an empty cell. The initial total wire length is 44,574

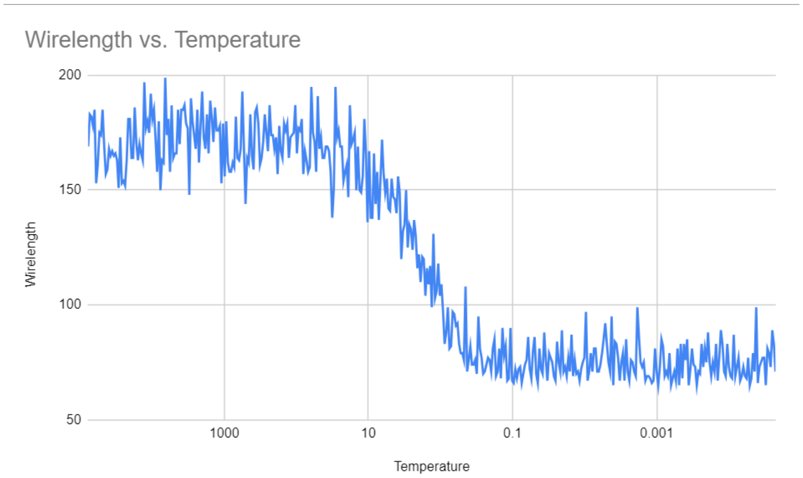


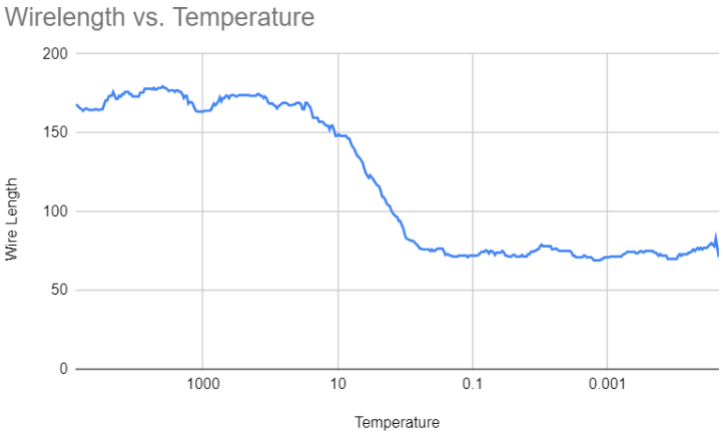
```
---- ---- ---- 0151 0104 ---- 0659 0658 ---- ---- 0186 ---- 1126 0622 0665 0621 0845 ---- ---- ---- ---- 0493 0997 1101 0721 0198 0673 0682 ---- 0100 0513 0471 0840 ---- ---- 0039 0192 ---- 0987 0688 0788 ---- 0097 0003 ---- ---- --
-- ---- ---- ----
---- ---- ---- 0017 1270 0338 1260 0101 0263 0540 0589 0678 1127 1050 1114 0962 0229 0436 0684 1088 0156 0516 0271 0765 0388 0352 0604 0672 0680 0512 1236 0511 1259 0372 0231 0934 1141 1053 0196 0103 0381 0594 0906 ---- 1163 1036 01
64 0001 ---- ----
---- ---- 0733 1005 0461 0790 0874 1093 0052 1000 0111 0116 0224 0624 0801 0920 0629 0780 0686 0413 0687 0485 0279 0802 0941 0242 0674 0028 1205 0681 0397 0472 0470 1100 0329 0337 0560 0901 1170 0645 0646 1071 0931 0122 0885 ---- 07
69 1219 ---- ----
---- ---- 0590 0591 0892 0739 0427 0426 0559 1279 1048 1176 0543 0495 0887 0394 0320 0341 1119 0281 0696 0761 0161 0767 0563 0480 0448 1120 0683 1177 0432 0520 0993 0083 0803 0291 1038 0972 0582 0084 0076 0554 0079 0553 0556 0771 08
51 1130 ---- ----
---- 0527 0149 0420 0592 1106 1077 0373 0709 0424 0319 0416 1263 0307 0345 0051 0048 0302 0359 0662 0255 0316 0766 0157 0705 1065 1145 0132 0647 0789 0754 0267 0792 1081 0809 0623 0523 1206 1034 0655 0173 0738 0692 0732 0317 0403 07
07 0188 0098 ----
---- 0090 0977 0250 0407 0145 0759 0492 0743 1109 1175 1154 0285 0891 0491 0342 0468 0046 0256 0282 0162 1076 0362 1171 0058 0129 0476 1004 0677 1035 0875 0509 0578 0900 1160 0106 0925 1066 0605 1261 1037 0598 0289 1275 1125 1249 05
61 0374 0811 ----
---- 0419 0846 0421 1013 0542 0820 0477 0057 1107 0380 1003 0965 0417 1001 0871 0935 0257 0661 0371 0612 0326 0557 0415 1072 0985 0702 0253 0059 0936 1228 0303 0360 0334 0717 0946 0496 1146 0206 1138 0603 0602 0404 1029 1030 0215 12
10 0597 ---- ----
---- 1286 0552 1075 0819 0451 0482 0776 0944 1105 1247 1237 0462 1244 0296 0507 0280 0124 0704 0919 0440 0457 0067 1209 0049 0979 0995 0016 0376 0614 1039 0724 0159 1202 0205 0204 1067 0514 0347 0110 0893 0580 1143 0276 0246 0932 11
37 1211 0829 ----
---- 0460 1149 0745 1078 0160 0701 0454 0842 0314 1180 0515 0463 0227 0848 0613 0947 0063 0430 0230 0864 0779 0675 0223 1204 1064 0786 0466 0983 0050 0984 0032 0526 0295 1214 0652 0797 1027 0446 1156 0818 0221 1182 0907 1079 0575 05
62 0641 0638 ----
0494 0219 0383 0581 0619 1186 0910 0866 1225 1022 1092 1024 0915 1002 0699 0618 0275 0452 0225 0545 0306 0366 0212 0348 0304 0018 0548 0914 0976 1240 1278 0010 0981 0757 0999 1061 1142 0312 0835 0772 0908 0583 0265 0328 0833 0643 09
92 0077 0640 ----
---- 0356 0760 1179 0693 0858 0071 1017 0375 0201 0435 0286 0961 1245 0088 0606 0631 0691 0998 0022 1223 1281 0174 1148 0841 1196 0518 0248 1151 0168 0120 0130 0113 0940 0189 1033 0502 0321 0243 0980 0324 0650 0167 1217 0114 0751 00
54 0163 0004 ----
---- 1128 0081 0445 0292 0909 0547 0817 0912 0245 0080 0888 0019 0730 0978 1095 0762 0532 0715 1284 1153 0469 0086 1115 0287 0029 1014 0087 0982 1229 0700 1043 1055 0041 0564 1231 1121 1184 0898 0685 0755 1080 1239 0969 0062 1098 05
36 0185 ---- ----
---- 0862 1264 0340 0085 0277 1185 0911 0865 0244 0331 1021 0595 1242 0226 0014 0021 0037 1218 1015 0663 0393 0616 0883 0654 0318 0438 0500 0963 1272 0405 0133 0369 0836 0607 0555 0546 0036 0873 0390 0389 1220 1139 1031 0787 0528 03
05 0458 ---- ----
---- 0378 1047 0524 0213 1158 0066 0774 0481 1251 0370 1287 0740 1016 0620 0137 0490 0020 0240 0565 0190 0209 0617 0361 0768 0363 0425 1183 1129 0933 0053 0136 1135 1252 0179 0261 0128 0689 0325 0075 0199 1096 1262 0015 0948 0181 00
40 ---- 1227 ----
---- 0442 0847 0315 1103 0278 0853 0183 0354 1166 0855 0929 0024 0411 1049 0517 1243 1238 1009 0344 0170 0441 0069 0488 0653 1054 0608 0433 0990 0013 0034 0232 0785 0274 1131 1152 0429 1147 0146 0782 0147 0584 1253 1007 1012 0753 11
02 0734 1178 ----
0549 1084 0418 0107 0720 0744 0550 0349 1025 0222 1144 0453 0119 1133 0294 0093 0064 0207 0134 0630 1172 1266 0262 0473 1045 0937 0586 1254 0952 0609 0260 1082 0826 0335 0773 1234 1069 1074 0722 0899 0398 0967 0382 1283 0044 0487 10
18 0923 1062 ----
0070 0456 0268 0537 0346 0152 1189 1010 1028 0266 1086 1164 1174 0649 0065 1207 0298 0791 0558 1282 0834 0886 1046 0805 0954 0957 0339 0917 0074 1157 0800 0450 0904 0746 1208 0012 0131 0671 0822 0408 0406 0974 0009 0144 0464 0714 07
28 0434 0593 ----
---- 1224 0259 0930 0141 0695 0521 0656 0735 1267 0385 0068 1073 1168 0308 0889 1162 0651 0690 0153 1288 0881 0544 1268 0368 0187 0986 0955 0489 0194 0869 0795 0867 0850 0736 0254 0447 0973 0386 1085 0710 1006 0251 0400 0506 0478 04
23 1011 0719 ----
---- 0264 0055 0006 0711 0770 0959 0988 0193 0861 0350 0358 1057 0309 0108 0273 1117 1257 0330 0878 0830 0235 0756 0444 0172 0045 1276 1132 0377 0313 0676 0355 1233 0043 0570 0143 0897 0599 0576 0793 0099 0828 1271 0402 0047 1277 04
55 0588 0467 ----
---- 0844 0139 1040 0569 0125 0270 1070 0960 0073 0778 0928 0384 1159 0812 0365 1241 1023 1110 0860 0333 0884 0703 1194 0634 0283 0872 0918 0184 1026 1155 0752 0996 0572 0367 1213 0035 1051 0627 0600 0042 0838 0118 1199 0942 0971 02
08 0723 0030 0816
0072 1280 0958 1192 1089 0784 0856 0877 0061 1020 0585 0105 1195 0228 0857 0323 1118 0431 1044 1215 0409 0026 0392 0519 1008 0237 0379 0171 1285 0890 1212 0299 0571 0391 0126 0203 0551 1068 0938 0234 0138 0810 0806 0180 1097 0636 07
64 0577 0815 ----
---- 0364 ---- 0353 0541 1216 0660 0241 1112 1246 0913 0311 1059 1042 0628 0712 1221 0357 0155 0626 1083 0437 1269 0343 0332 0950 0713 0927 0567 0503 0449 0217 0798 0596 0479 0135 0896 1203 1150 0824 0837 0031 0082 0399 1201 0694 00
11 1188 0813 0839
---- ---- 0123 ---- 0252 0966 0808 0953 0538 1123 1191 1122 1173 0216 0975 0794 0807 1104 0414 0832 0089 0027 1289 0401 0870 0951 1232 0166 0777 0412 1094 0750 0868 0095 0210 0775 1058 0508 0601 0284 0964 0968 0939 0970 0831 0158 01
12 1019 0863 ----
0708 0115 0530 0574 0579 0269 0301 1056 0475 0903 1200 0657 1113 0272 1124 1087 1108 0200 0439 0197 0625 0827 0220 0127 0642 0922 0327 0615 1230 0876 0749 0218 0025 0395 0852 0902 1197 0731 0823 0202 0056 0783 0176 0092 0706 0725 06
67 0637 0814 ----
---- ---- 0351 ---- ---- 0300 0573 1222 0943 0148 0804 0644 1167 0396 1116 0474 0632 1063 0191 0639 0633 0994 0177 1258 0737 1265 0484 0150 0741 0169 1250 0882 0258 0758 1248 0428 0236 0799 0154 0895 0525 0648 0060 0443 0729 0949 06
79 0410 0096 ----
---- 0531 0033 1226 0102 0459 0533 0534 0535 0989 1169 0716 0666 0635 0718 0926 0465 0781 0422 0182 0880 0916 1041 0879 0849 0664 1090 0905 0821 0610 0748 0945 1256 0293 1052 0498 1190 0747 0796 0742 0894 0140 0038 0121 0178 0109 00
23 1165 ---- ----
---- ---- 0529 0921 0094 ---- 1136 0002 1187 0499 1193 1198 0005 0288 0297 0078 0566 1134 0483 0486 0568 0727 0843 0247 0510 0165 0697 1091 0539 1273 0497 0501 0214 0233 0505 1255 0668 0670 0669 0522 0290 1235 0091 1099 0322 ---- 02
38 ---- ---- ----
---- ---- 0000 0008 0859 ---- 0195 0726 0117 0587 0924 0239 1060 0504 0249 0310 0211 1161 1032 1181 0991 0611 0336 1140 0763 ---- 0854 0698 0387 1111 0956 ---- ---- ---- ---- 1274 ---- ---- 0825 0007 ---- ---- ---- 0142 0175 ---- --
-- ---- ---- ----
Total wire length: 10916
PS C:\Users\ziad_\Desktop\Fall23\DD2\Project>
```

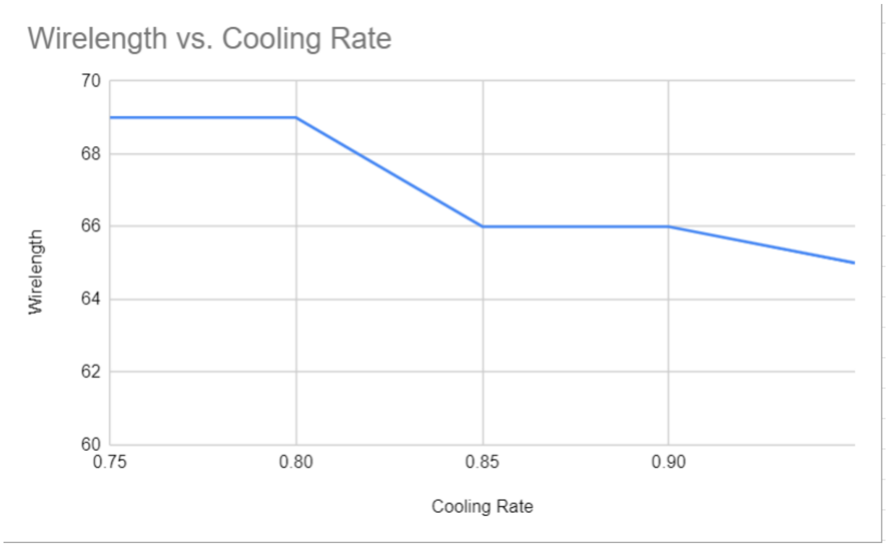
This is the output after running SA. The corners are empty, the wire length is 10,916.

Graphs

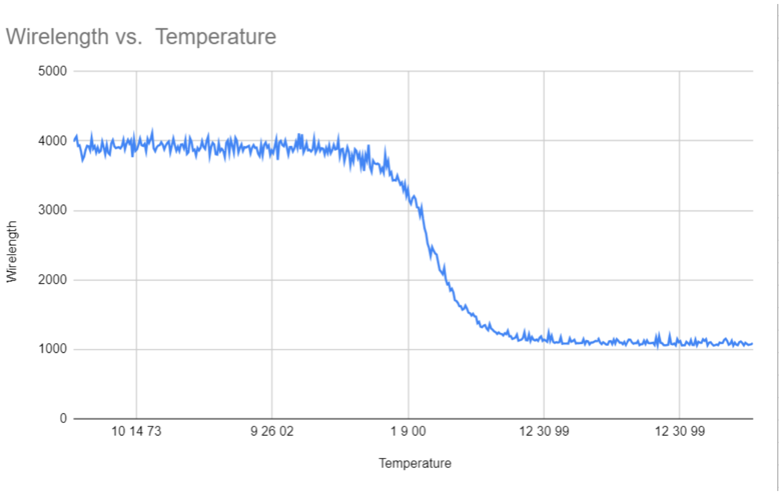
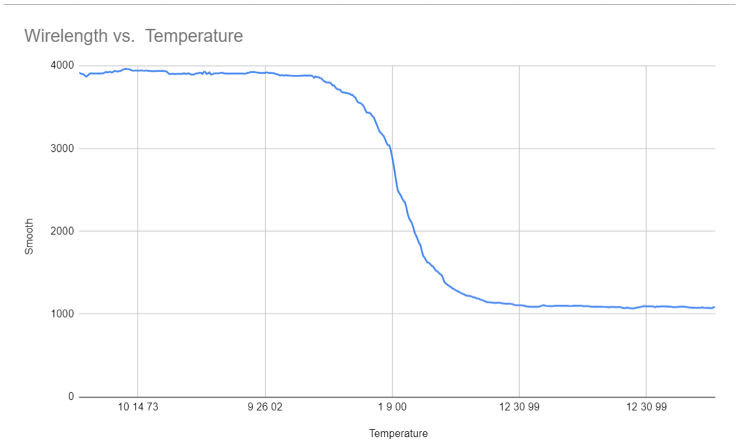
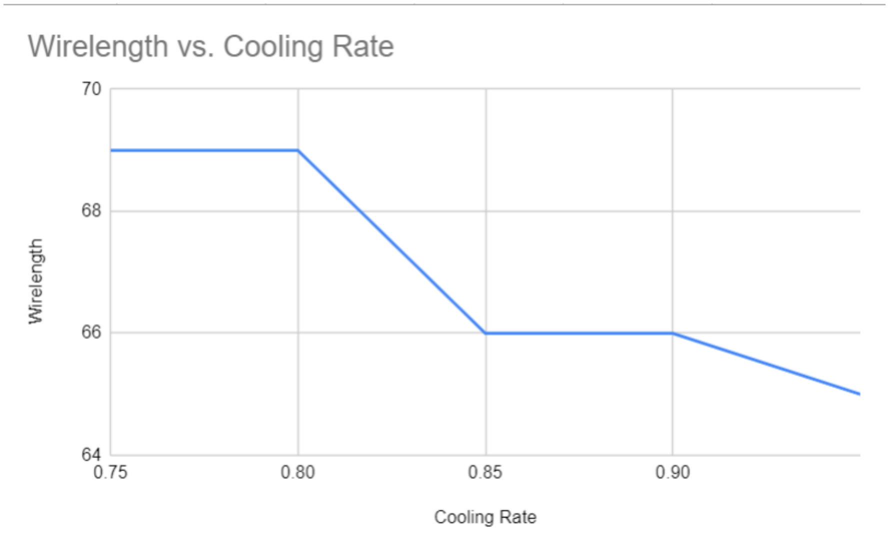
d0	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

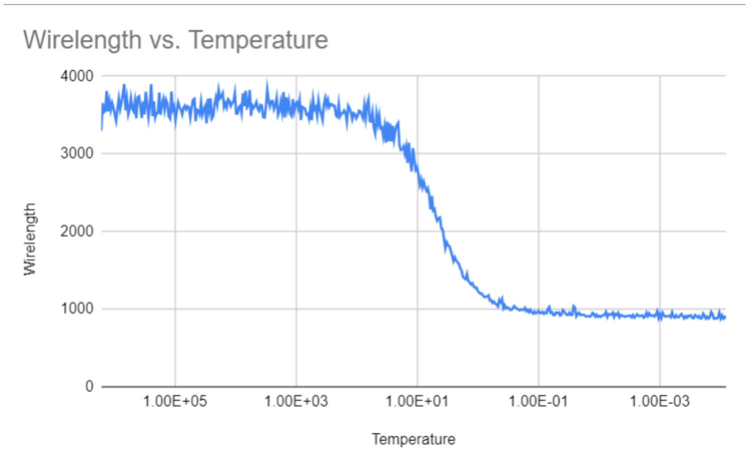
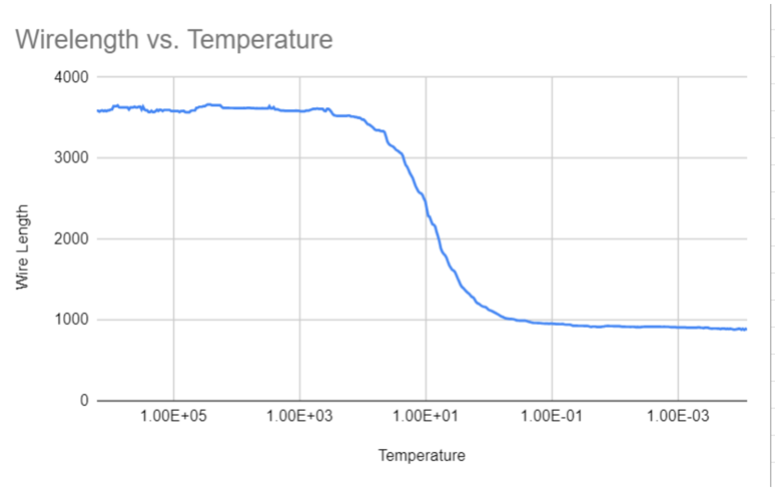
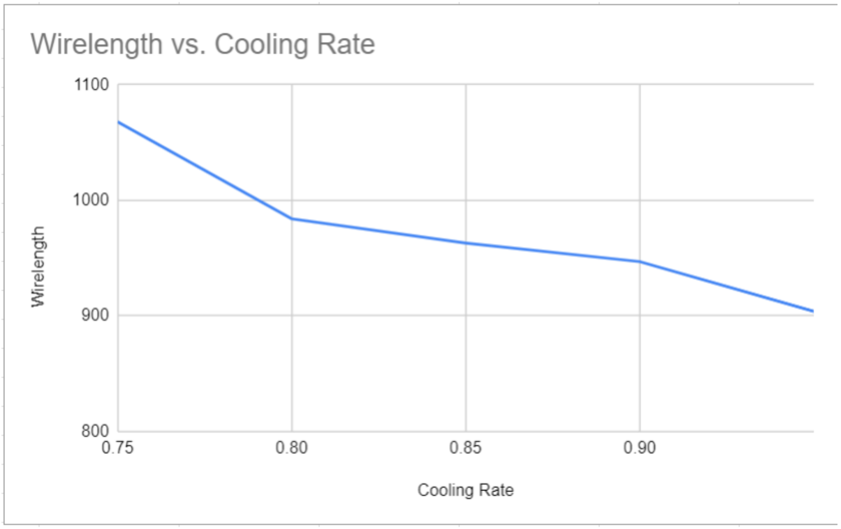
<table>
<tr><td>d1</td><td></td></tr>
<tr><td>WL vs Temp</td><td>


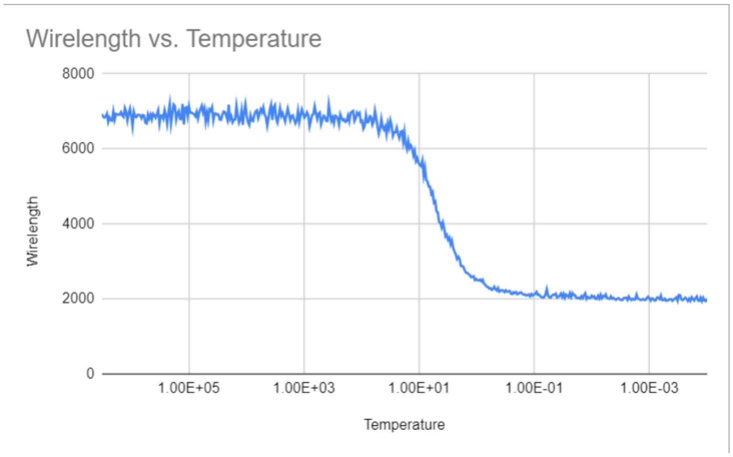
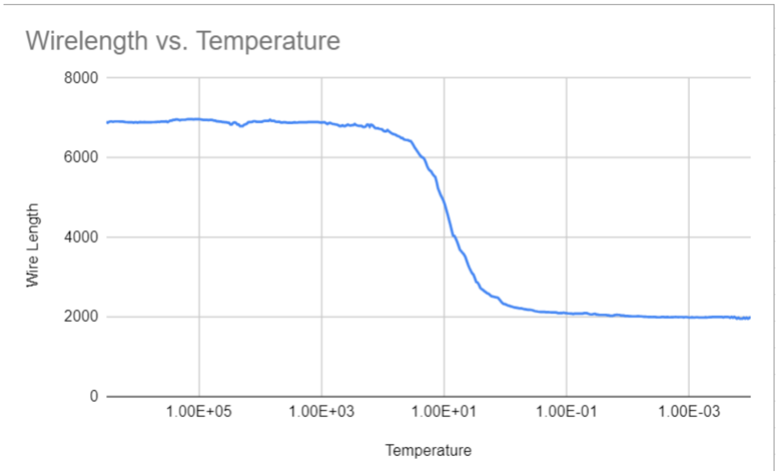
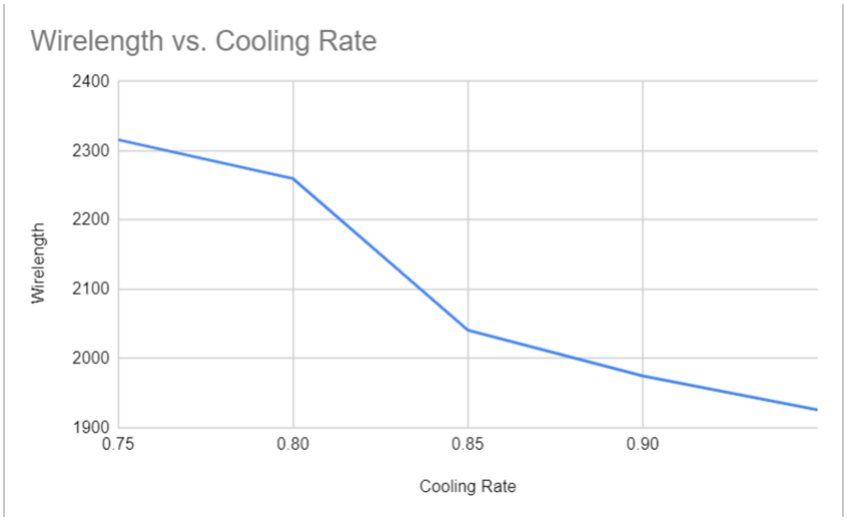
</td></tr>
<tr><td>Smoothed WL vs Temp</td><td>


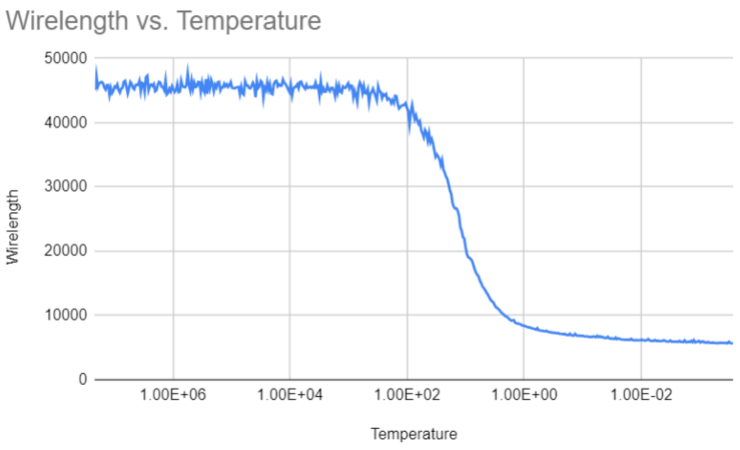
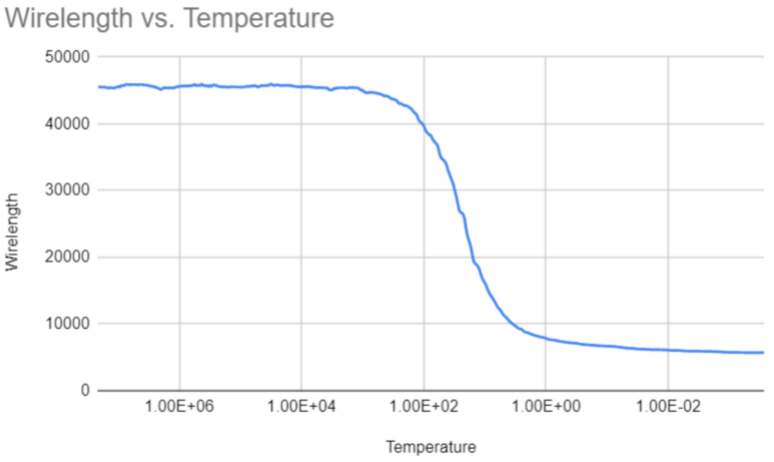
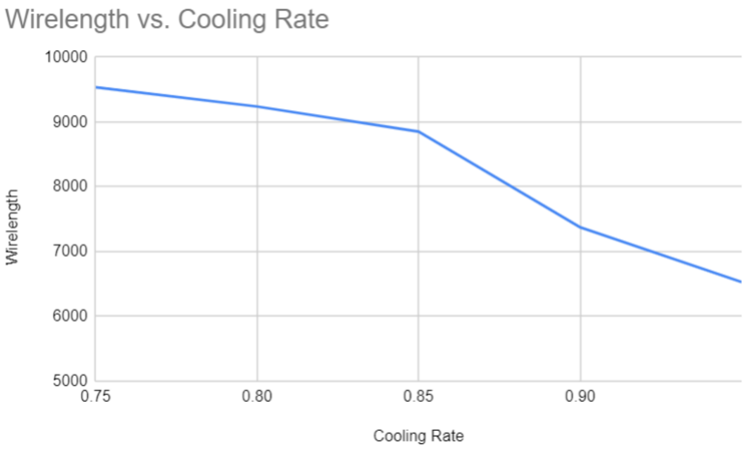
</td></tr>
<tr><td>WL vs Cooling Rate</td><td>


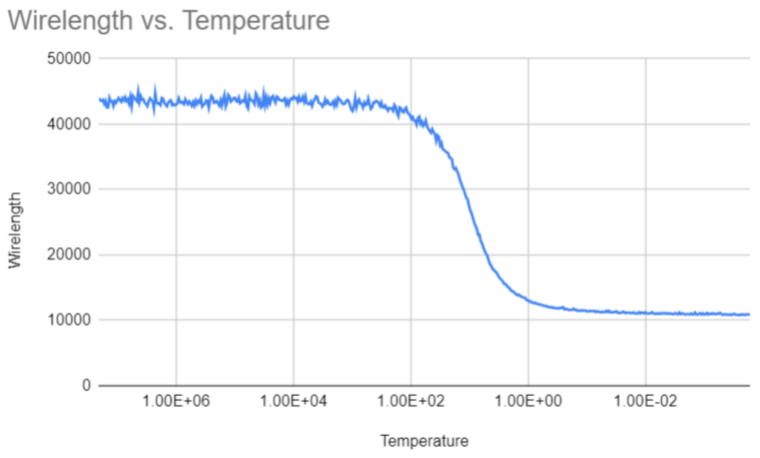
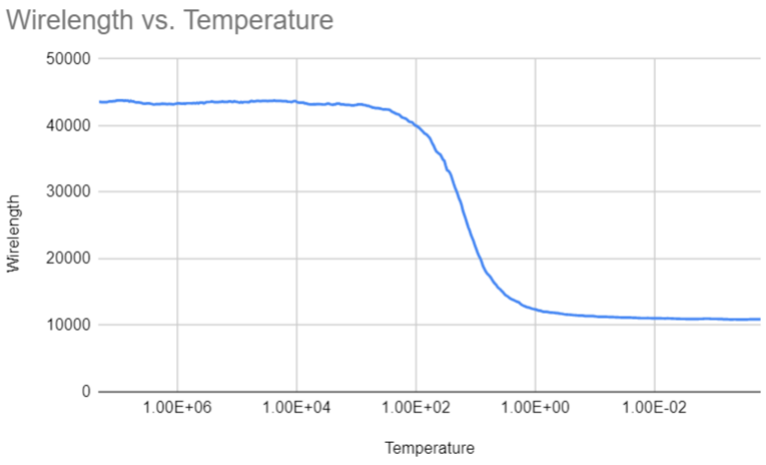
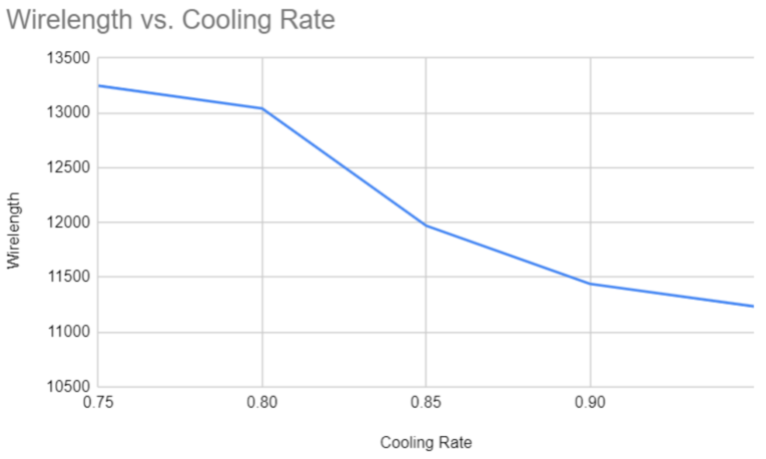
</td></tr>
</table>

d2	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

d3	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

t1	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

t2	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

t3	
WL vs Temp	
Smoothed WL vs Temp	
WL vs Cooling Rate	

Conclusion

As we can see from the graphs, as the temperature decreases, the wire length decreases. This is expected because each iteration the temperature is multiplied by the cooling rate, and each iteration the wire length is decreased. Moreover, as we increase the cooling rate, the wire length decreases. This is because the temperature converges to the final temperature slower when the cooling rate is higher.