Simulated Annealing Results

Phase 2's goal was to implement simulated annealing to reduce the wire length, HPWL of the input circuit's chip. To achieve this, my annealing process makes up to 167 passes over the nodes where at each pass, the algorithm makes n, the number of gates, swaps. For each new solution, the cost is calculated using the function:

$$cost = 0.7 * actual\_chip\_width * height + 0.3 * total\_HPWL$$

This cost function takes into account both the chip area and HPWL to minimize. This is because initial implementations of the cost function without chip area caused some rows of the chip to expand more than expected. By incorporating area into the cost, row width still goes out of the initial set bounds but not by much.

To try and combat the long runtime of this simulated annealing algorithm, I implemented an early termination condition to exit annealing if more than 10 passes occur without accepting a swap. The runtimes were still long but the idea with this approach was that, by continuing to run, we may find a few more good swaps but the solution is good where continuation is not worth the extra runtime. My initial implementation's runtime was still slow because the algorithm recalculated the total HPWL for each solution by iterating over every gate. This was inefficient as most of the gates would not require updating of their HPWL. Thus, the current solution takes this into account, only updating the gates swapped, gates who's x position changed during the swap, and all those gate's fanin gates.

The results from annealing are shown in the table below. We can see the runtime of this algorithm increases quickly and may not be viable for very large net lists. If you look at the initial HPWL in the output files compared to the final HPWL reported below, smaller net lists can see over a 50% reduction in wire length but as we increase the number of gates, that gain is still great but not as large as 50%.

Circuit Name	Chip Width	Chip Height	Starting HPWL	Final HPWL	Runtime
c17	5	4	36	24	0.128 seconds
c17_dummy	5	5	42	23	0.117 seconds
c423	23	22	3798.5	1670	17.71 seconds
c499	31	30	6152.5	2653.5	28.84 seconds
c880	35	33	12238.5	6338	81.23 seconds
c1355	38	35	17635.5	8323.5	143.21 seconds
c1908	44.5	43	32826.5	14872	5.00 minutes
c2670	60.5	57	64682	26629.5	12.50 minutes
c3540	68.5	64	90477	40609.5	16.70 minutes
c5315	88	81	167129	128367	39.53 minutes
c6288	88	85	191188	136356	32.56 minutes
c7552	100.5	93	286154	203848	1.36 hours