

	P=0.2		P=0.8	
	Runtime	Success	Runtime	Success
5 max flips	~.0007 seconds	4/5	~.0009 seconds	2/5
100 max flips	~.0008 seconds	5/5	~.0007 seconds	5/5
1000 max flips	~.0007 seconds	5/5	~.0007 seconds	5/5

We were surprised to find that our algorithm performed similarly well for each p and max_flip value. As expected, our algorithm performed best with lowest p , because this meant that our algorithm would more frequently choose the best Boolean to flip rather than randomly picking one. However, we rarely required more than 10 flips in any of our ~100 automated trials due to the small domain of the diagnosis problem. This surprised us but considering the size of each clause, picking a random choice in this clause leaves only 1 of 2 flip options. Therefore if 1 of these 2 satisfies our clauses, we should expect only a few flips to occur before the correct of the 2 Booleans is chosen.

To improve the WalkSAT algorithm, we flipped the literal from all clauses that contradicted the most clauses rather than just a contradicting literal from a random clause. Choosing to flip a contradicting literal from a random clause doesn't provide as much of an effect since we are limiting to the randomly selected literal. If we flip the literal that is causing the most issues, we should become much closer to the solved state. If we think about this using a heuristic, we could define the distance from the solution state as the number of clauses that evaluate to false. Applying this to flipping the ideal literal, we would see that the greatest step we could take toward the solution state is if we flip the literal that will validate the most clauses as opposed to the just the random clause we pick in the original WalkSAT algorithm.