Experiment 1 (25 points): n=4, k=3, m=4 to 36

Compare the curves that result from running this experiment with both algorithms. Are they the same? Why, or why not?

They are nearly the same. There are cases in which WalkSAT does not satisfy the sentence but PL resolution does (since it is complete), this is because the max_flips is limited to 250, thus there might be a model possible if more flips are performed, but for the current experiment it does not satisfy. This happens only for one or two test cases, otherwise the probability of satisfiability is nearly the same for both algorithms.
At what ratio do random sentences switch from being easy \((p \sim 1)\) to being very hard \((p \sim 0)\)? Is this precisely the same place as the critical ratio of \(m/n = 4.3\) suggested by Russell & Norvig? Why, or why not?

At ratio \(m/n=17/4=4.25\), the graph goes from being easy to hard. This can be seen in the following experiment screenshot:

![Screenshot](image)

It is nearly the same suggested in the book. They have taken a larger variable set of \(n=50\) and check for large number of clauses, hence it is more accurate. In this experiment, \(n\) and \(m\) are lower, but the threshold is nearly the same.

After the \(m/n = 26/4=6.5\) ratio, the probability becomes nearly zero for both the algorithms. This is shown in the following screenshot:
Experiment 2 (25 points): n=4, k=2 to 4, m=4 to 28

What seems to happen to the satisfiability threshold for this choice of parameters? Has it moved, or stayed the same? Give an explanation as to why this might be the case.

Ans: The satisfiability threshold is directly proportional to the length of symbols in clause. If k is less, the ratio of m/n at which the threshold is attained is less.

For given inputs, the thresholds are as follows:
For k=2, threshold is attained at ~2.25
For k=3, threshold is attained at ~4.25
For k=4, threshold is attained at ~7 to 7.2

We can note that the satisfiability threshold increases as and when k increases. This can be explained because as the number of symbols in a clause increase, the chance of the clause of becoming True increases and thus satisfiability increases. If one of the symbol in a clause is True, that clause becomes true.
Below are the graphs generated by plotting $m/n$ ratio versus $P($satisfiability$)$ for $k=2,3,4$. Since the permutation limit for $n=4$ reaches at $m/n$ ratio = 4 (ie. When $k=4$, and $m/n=4$, permutation limit is reached and thus I have considered for $k=4,n=5$)
Do you have confidence in the results of this experiment? Why, or why not?

Yes, I feel that the results are quite correct. This is because when \( k \) increases, the probability of the clause becoming true increases (Since there is OR in between the literals) and so the satisfiability threshold also increases. As seen from graph above, for \( k=5 \), satisfiability threshold is as high as 7.20.

Experiment 3 (25 points): \( n=4 \), \( k=3 \), \( m=4 \) through 32

Runtime for WalkSAT vs \( m/n \) ratio graph:
How long does it take each to finish? In what regions is the advantage of WalkSAT seen to be the greatest? What kinds of problems reside in this region (i.e., easy or difficult)?

It takes much faster for a WalkSAT algorithm to finish because WalkSAT searches locally whereas PL Resolution compares the two clauses and the function is called after each iteration. Advantage of WalkSAT in terms of satisfiability can be seen at a lower m/n ratio where it can find a solution within max_flips. Here, the problems are said to be easy. However, if we remove the limit on max_flips, we can have advantage of a fast WalkSAT algorithm at very high m/n ratios, at which PL resolution would take a long time to resolve. The advantage here in terms of runtime and these problems are said to be difficult since there are too many clauses to check.

What was the best runtime, and at which clause/symbol ratio was it achieved? What was the worst, and at which clause/symbol ratio was it achieved? How much of a discrepancy was there between these?

The best runtime is at m/n = 1, it takes an average of 1.62 iterations to find satisfiability. The worst runtime is at m/n=6.25, it takes an average of 47 iterations to find satisfiability. There is a discrepancy of 45.38 between the best and worst case times.

Extra Credit (10 points):

What is the possible effect of treating sentences that WalkSAT cannot find models for as unsatisfiable? How will this bias the results?

Ans: In our experiment we are taking max_flips just as high as 250 after which WalkSAT declares the sentence as unsatisfied. However, this might not be the case. The sentence may be satisfied if more than 250 flips are performed, a case which would change the entire result set. Thus, if WalkSAT cannot find models for a particular sentence, it doesn’t particularly mean that the sentence is unsatisfied. This creates biased results because PL Resolution gives satisfiability with certainty but to check the satisfiability with WalkSAT, we need to remove the constraints from max_flips.