Task 1: Logic and SAT Encoding

Question 1: Express each of the above-mentioned constraints in FOL and transform into clauses in CNF format. How many clauses in total are there to encode a wedding seating arrangement in terms of <M>, <N>, <E> and <F>?

There are 4 constraints for which CNF sentences needs to be generated.
For each guest 1<= i <= M
N = Number of Tables

1. Each guest can sit at at-least 1 table
FOL: ¬Xi1 v ¬Xi2 ... v ¬Xi(n-1) v ¬Xi(n+1) ... v ¬ XiN => Xin (for each n that 1 <= n <= N)
CNF: X11 v X12 v ... v X1N
# of clauses: M

2. Each guest can sit at at-most 1 table.

X11 => ¬X12	¬X11 v ¬X12
X11 => ¬X13	¬X11 v ¬X13
...	
X1n => ¬X1j	¬X1n v ¬X1j

(where 1<= j != n <= N)
# of clauses: M*N*(N-1)/2

3. Any 2 friends should be seated at the same table

X11 <=> X21	(X11 => X21) ^ (X21 => X11)	(¬X11 v X21) ^ (¬X21 v X11)
X1n <=> X2n	(X1n => X2n) ^ (X2n => X1n)	(¬X1n v X2n) ^ (¬X2n v X1n)

Xij <=> Xkj (where k != i)
# of clauses: N*F*2

4. Any 2 enemies should be seated at different table
Xin => ¬Xjn and Xjn => ¬Xin
(¬Xin v ¬Xjn) ^ (¬Xjn v ¬Xin) [only 1 of the 2 CNF clauses is required as both give the same result]
# of clauses: N*E

Total # of clauses: M + (M*N*(N-1)/2) + (N*F*2) + (N*E)

Task 2: Instance Generator

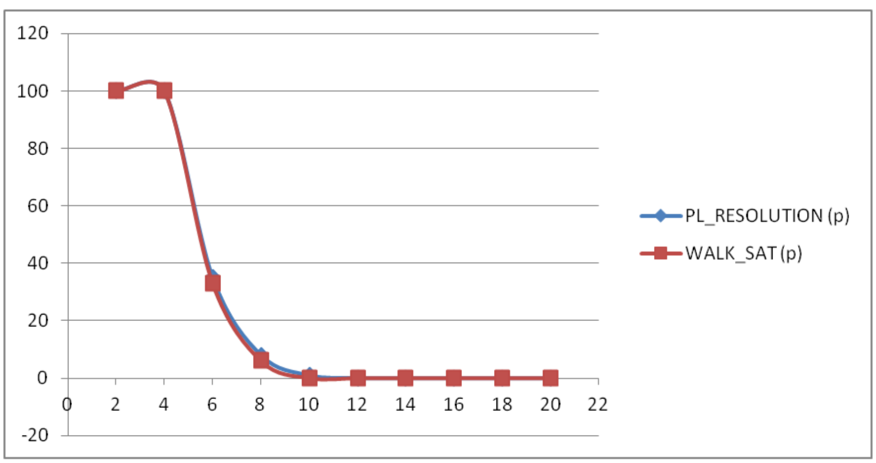
Implemented the generation of Relationship matrix R and the CNF sentences.

Task 3: SAT Solvers

Implement PL_RESOLUTION and WALK_SAT.

Task 4: Experiment 1

e	PL_RESOLUTION	WALK_SAT
2	100	100
4	100	100
6	35	33
8	8	6
10	1	0
12	0	0
14	0	0
16	0	0
18	0	0
20	0	0



Question 2: Compare the curves that result from running this experiment with both algorithms. Are they the same? Why, or why not?

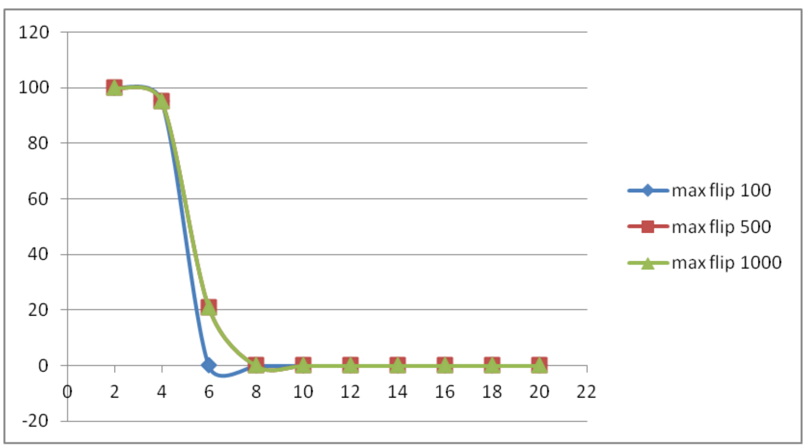
No. The Although both the curves look to be the same, the PL_RESOLUTION curve has the edge at point 6 through 8.

The reason is PL_RESOLUTION is a sound and complete algorithm, and it can always tell with certainty satisfiability (or unsatisfiability) if one exits. But WALK _SAT is limited by the max_flips that is allowed. If the solution is within the max_flips, then WALK _SAT would find it, but if the solution is well above the max_flips, WALK_SAT cannot find the satisfiability.

i.e, if we keep max_flips = infinity, then WALK_SAT will find the satisfiablility ONLY if there is a solution, else it will keep going forever.

Task 5: Experiment 2

f	max_flips = 100	max_flips = 500	max_flips = 1000
2	100	100	100
4	95	95	95
6	0	21	21
8	0	0	0
10	0	0	0
12	0	0	0
14	0	0	0
16	0	0	0
18	0	0	0
20	0	0	0



Question 3: What seems to happen to the satisfiability as <f> increases? Give an explanation as to why this might be the case.

We can observe that the satisfiability decreases as we increase <f>.
The reason behind this is that, the increase in <f> gives rise to more number of clauses (while symbols remain as is). This in turn might increase the number of max_flips required and thus taking the solution out of the reach of the WALK_SAT algorithm.

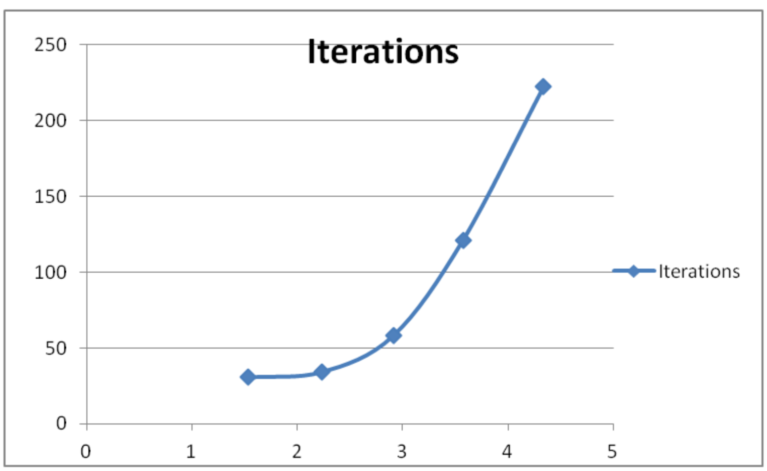
Question 4: How does the result vary with different <max_flips>? Why, or why not?

The higher the max_flips, the higher the value of p <satisfiability>. This is because, a satisfiable model might be available at 125th flip, which cannot be found by a WALK_SAT algorithm running with max_flips = 100.

e.g, p = 5 for max_flips = 100
and p = 6 for max_flips = 127

Task 6: Experiment 3

M	N	Clauses/Symbol Ratio	Iterations
16	2	1.535	30.54
24	3	2.23333	34
32	4	2.91094	57.88
40	5	3.575	121.14
48	6	4.33792	222.08
<f=2%>	<e=2%>		



Question 5: Is the average ratio of clause/symbol in the sentences consistent with that you theoretically derive from the result of Question 1? Why or why not? You need to consider the probability setting <f=2%, e=2%> in this case.

Yes. As we can see from the above data table, the increase in M and N clearly has significance in the clause/symbol ratio and the number of iterations. This can mean only that, the number of clauses and symbols required for the CNF representation increases with increase in M and N.

Question 6: How does average runtime change with regard to the average ratio of clause/symbol in this experiment? Is the curve consistent with that of AIMA Figure 7.19(b)? Why or why not?

- From the graph above, we can see that the average runtime curve increases with increase in number of clauses/symbols ratio.

- The Above graph is consistent with the AIMA Figure 7.19(b). This is because, although there are many versions of WALK_SAT, both the graphs contain data points that were generated using same algorithm. AIMA 7.18