Answer 3

1 Introduction

It is also known that 3-SAT exhibits an easy-hard-easy computational pattern. Determining the satisifiability of sets of clauses that are small in relation to the total number of distinct propositional variables in the set is usually easy because there are fewer constraints in assigning truth values to the propositional variables. Determining the satisifiability of sets of clauses that are large in relation to the total number of distinct propositional variables in the set is usually easy because there are too many constraints to assign truth values to the propositional variables and the set is unsatisfiable. Somewhere in between these two extremes the satisfiability problem becomes hard.

2 Generating Test Cases

I wrote a short program *generator.py* and used it to generate 13 cases for this question from *file0.cnf* to *file12.cnf* in the current folder.

Example: *file0.cnf*

c example CNF file with 1000 propositional variables and 100 clauses

p cnf 1000 100

-405 -89 -266 0

-780 58 202 0

223 -663 -487 0

-149 570 936 0

…

3 Testing

I tested these 13 cnf files by **minisat** on the CSE machines. I recorded the details in the txt format files from *file0Statistics.txt* to *file12Statistics.txt* in the current folder. In these two extremes, it’s easy to solve since some of them have zero or almost zero CPU time. Even some of them are unsatisfiable, but the CPU time is very low. Some of cases are as follows.

**Case 0**:

Number of variables: 990

Number of clauses: 100

restarts: 1

conflicts: 0 (-nan /sec)

decisions: 1 (0.00 % random) (inf /sec)

propagations: 0 (-nan /sec)

conflict literals: 0 (-nan % deleted)

Memory used: 5.00 MB

CPU time: 0 s

SATISFIABLE

**Case 1**:

Number of variables: 100

Number of clauses:100

restarts: 1

conflicts: 0 (-nan /sec)

decisions: 1 (0.00 % random) (inf /sec)

propagations: 0 (-nan /sec)

conflict literals: 0 (-nan % deleted)

Memory used: 5.00 MB

CPU time: 0 s

SATISFIABLE

**Case 2**:

Number of variables: 400

Number of clauses: 1200

restarts: 1

conflicts: 18 (inf /sec)

decisions: 128 (0.00 % random) (inf /sec)

propagations: 1390 (inf /sec)

conflict literals: 325 (6.88 % deleted)

Memory used: 5.00 MB

CPU time: 0 s

SATISFIABLE

**Case 6**:

Number of variables: 100

Number of clauses: 986

restarts: 1

conflicts: 3 (inf /sec)

decisions: 2 (0.00 % random) (inf /sec)

propagations: 50 (inf /sec)

conflict literals: 3 (0.00 % deleted)

Memory used: 5.00 MB

CPU time: 0 s

UNSATISFIABLE

**Case 7**:

Number of variables:100

Number of clauses:1486

restarts: 1conflicts: 3 (inf /sec)decisions: 3 (0.00 % random) (inf /sec)propagations: 36 (inf /sec)conflict literals: 2 (0.00 % deleted)Memory used: 5.00 MBCPU time: 0 sUNSATISFIABLE

**Case 8**:

Number of variables:100

Number of clauses:2000

restarts: 0

conflicts: 0 (-nan /sec)

decisions: 0 (-nan % random) (-nan /sec)

propagations: 8 (inf /sec)

conflict literals: 0 (-nan % deleted)

Memory used: 5.00 MB

CPU time: 0 s

UNSATISFIABLE

At the beginning, I only generated 8 cases. However, most of them are easy to solve except two cases which C is 4 and 5. Therefore, I guess that C value may between 4 and 5 and then generated other cases for testing. Based on the 13 cases, I found that when C approaches 4.25, the CPU time gradually becomes very high and the number of conflicts is very large. It means that the problem becomes very hard. When C equals to 4.25, CPU time is 102.248s. It also has many conflicts and unsatisfiable after a very long time. The details are as follows. Except it, when C equals 4.5, the CPU time is 52.1s. When C equals 4.75, the CPU time is 13.416s.

**Case 12**:

Number of variables: 400

Number of clauses:1700

restarts: 8190

conflicts: 4784225 (46790 /sec)

decisions: 5807724 (0.00 % random) (56800 /sec)

propagations: 283229619 (2770026 /sec)

conflict literals: 74493486 (25.87 % deleted)

Memory used: 6.00 MB

CPU time: 102.248 s

UNSATISFIABLE

**Case 10**:

Number of variables: 400

Number of clauses: 1800

restarts: 4350

conflicts: 2552014(48983 /sec)

decisions: 3110877 (0.00 % random) (59710 /sec)

propagations: 145426024 (2791286 /sec)

conflict literals: 39410100 (24.78 % deleted)

Memory used: 5.00 MB

CPU time: 52.1 s

UNSATISFIABLE

**Case 11**:

Number of variables: 400

Number of clauses: 1900

restarts: 1534

conflicts: 729867 (54403 /sec)

decisions: 883583(0.00 % random) (65860 /sec)

propagations: 41083843 (3062302 /sec)

conflict literals: 9998794 (26.75 % deleted)

Memory used: 5.00 MB

CPU time: 13.416 s

4 Conclusion

According to the test, I came up with the constant value C≈4.25 empirically. The number of variables is 400 and the number of clauses is 1700. I also made a bar chart(*barchat.jpg*) by *barchat.py* in the folder in order to show the conditions of all cases. The 13 cases show the point where the problem is hard to determine.图表, 形状

描述已自动生成