Unique Solution Instance Generation for the 3-Satisfiability (3SAT) Problem

Mitsuo Motoki*   Ryuhei Uehara†

Abstract

For the 3-Satisfiability Problem (3SAT), we consider a problem of generating its positive instance (and its solution) randomly. For the problem, it is desirable if an instance generation algorithm produces only (and all) "unique solution instances", i.e., instances with only one solution. In this paper, we consider two simple instance generation algorithms, and estimate the number of clauses required by the algorithms to generate unique solution instances with high probability.

*`mmotoki@is.titech.ac.jp`. Tokyo Institute of Technology.
†`uehara@komazawa-u.ac.jp`. Komazawa University.

1 Introduction

A positive instance generation problem for NP problem X is to generate $x \in X$ and a witness of x (or solution of x) randomly. For 3SAT, this problem is to generating satisfiable 3CNF f and truth assignment t satisfying f.

Generally, instance generation problem is often used in testing or checking efficiency of algorithms which solve NP problem X. Furthermore it is useful to estimate various properties of the NP problem experimentally, for example, cross over point of 3SAT (the point at which half of the randomly generated 3SAT instances are satisfiable).

It is eventually desirable to generate instances according to given probabilistic distribution. But it seems too hard to construct such an algorithm. On the other hand, it is often required that to generate instances which is according to characteristic distribution or have some properties.

In this paper, we consider a problem to generate unique solution instances. In the case of 3SAT, though unique solution instances are positive instances, by adding only one clause which suppresses the solution, we can generate negative instances. Since it is impossible to generate all negative instances unless NP = co-NP, this negative instance generation algorithm is effective. Furthermore this negative instances are useful for MAX-3SAT problem, optimization version of 3-SAT, because the maximum number of satisfied clauses is $m-1$, where m is the number of generated clauses.

We propose two algorithms for generating unique solution instance of 3SAT, obtain theoretical bounds of the number of clauses to generate unique solution instance with high probability, and evaluate them experimentally. The theoretical bound of our first algorithm is $\Theta(n \ln n)$, and that of second one is $O(n)$. Furthermore the point at which half of instances generated by the second algorithm have unique solution is about 4.3n. This point is nearby the crossover point in random 3SAT, estimated at $4.258n + 58.26n^{1/3}$ [CA96].

2 Preliminaries

We consider 3-conjunctive normal form formulae (3CNF) over n variables, $x_1, x_2, \ldots, x_n$. A clause is a set of exactly 3 literals, variable x or its complement $\overline{x}$. We only consider the ordinary clause, i.e, any variable appears at most one time in a clause. 3CNF f of m clauses is a sequence of clauses, $c_1, c_2, \ldots, c_m$. The choice of the clauses will be made with replacement, i.e., the algorithms may choose a clause more than once.

Throughout this paper, we use T_k to denote the set of assignments whose Humming distance from $t_0 = 1^n$ is equal to k.

For any truth assignment t of n variables, we denote by C_t the set of clauses satisfied by t. Let N denote $\|C_t\|$ and N_k denote $\|C_{t_0} \cap C_t\|$ where $t \in T_k$ for each k with $1 \le k \le n$.

Proposition 2.1 *$N = 7\binom{n}{3}$ and $N_k = 6\binom{n}{3} + \binom{n-k}{3}$.*

If an assignment t does not satisfy a clause c, then we say c *suppresses* t. If a clause c of 3CNF f is the first clause in f which suppresses an assignment t, then we say c *hits* t.

3 Naive Instance Generation Algorithm G

In this section, we consider a simple algorithm G.

Algorithm G
input: the number of variables n, the number of clauses m;
begin
  choose truth assignment $\hat{t} \in \{0,1\}^n$ randomly;
  for $i = 1$ **to** m
    do choose a clause from $C_{\hat{t}}$ randomly as ith clause c_i of f ;
  output $(f_{G(n,m)}, \hat{t})$;
end.

Clearly G can output all positive instances. And intuitively $f_{G(n,m)}$, output formula of $G(n,m)$, has few solutions if m is large. Hence, our interests is a condition of m that f has only one solution with high probability.

Without loss of generality, we can assume that $t_0 = 1^n$ is chosen as $\hat{t}$, solution of $f_{G(n,m)}$.

3.1 Upper Bound

Theorem 3.1 *For any positive constant $\delta < 1$, if $m \geq \frac{7}{3}n\ln n - \frac{7}{3}n\ln\frac{\delta}{2}$, then $f_{G(n,m)}$ has only one solution with probability at least $1-\delta$.*

Proof. We consider a condition that $f_{G(n,m)}$ has an another solution $t_k \in T_k$. In this case, $f_{G(n,m)}$ consists of only elements of $C_{t_0} \cap C_{t_k}$. Thus, probability that $f_{G(n,m)}$ has another solution $t \in T_k$ is $\left\{\frac{N_k}{N}\right\}^m$.

Then, probability that $f_{G(n,m)}$ has at least two solution is bounded by follows.

$$\begin{aligned}\Pr\left[\, f_{G(n,m)} \text{ has at least two solutions}\right] &\leq \sum_{\hat{t}'\neq\hat{t}} \Pr\left[f_{G(n,m)} \text{ has another solution } \hat{t}'\right] \\ &= \sum_{k=1}^{n}\binom{n}{k}\left\{\frac{N_k}{N}\right\}^m\end{aligned}$$

If this value is less than δ, then probability that $f_{G(n,m)}$ has only one solution is larger than $1-\delta$. Let $p(k)$ be $\binom{n}{k}\left\{\frac{N_k}{N}\right\}^m$. It is sufficient to show that $p(1) \leq \delta/2$ and $\frac{p(k)}{p(k+1)} \geq 2$ for $m > \frac{7}{3}n\ln n - \frac{7}{3}n\ln\frac{\delta}{2} = \frac{7}{3}n\ln\frac{2n}{\delta}$.

Here, since $m \geq \frac{7}{3}n\ln\frac{2n}{\delta}$, we get

$$\begin{aligned}p(1) &= n\left\{\frac{N_1}{N}\right\}^m = n\left\{1-\frac{3}{7n}\right\}^m \\ &\leq ne^{-\frac{3}{7n}m} \leq \frac{\delta}{2}.\end{aligned}$$

Moreover, since $m \geq \frac{7}{3}n\ln\frac{2n}{\delta} \geq \frac{7}{3}n\ln 2n$, we have

$$\begin{aligned}\frac{p(k)}{p(k+1)} &= \frac{k+1}{n-k}\left\{\frac{N_k}{N_{k+1}}\right\}^m \\ &\geq \frac{1}{n}\left\{\frac{N}{N_1}\right\}^m \geq \frac{1}{n}e^{\frac{3}{7n}m} \geq 2.\end{aligned}$$

This completes the proof of the theorem. □

3.2 Lower Bound

Intuitively, if m is too small, then $f_{G(n,m)}$ shall have some solutions near $\hat{t}$. This observation leads us to the lower bound of number of clauses.

Theorem 3.2 *For any positive constant $\varepsilon < 1$, if $m < \frac{7}{3}n \ln n + \left(\frac{7}{3}\gamma - \alpha\right) n$, then $f_{G(n,m)}$ has at least two solutions with probability at least $1 - \varepsilon$, where γ is the Euler's constant and $\alpha > \frac{7\pi}{3\sqrt{6\varepsilon}}$.*

Proof. It is easy to see that $f_{G(n,m)}$ has unique solution if and only if all assignments except $\hat{t}$ are hit.

Let $t_1^i \in T_1$ be an assignment whose ith bit x_i is 0. We remark that when $f_{G(n,m)}$ has unique solution, all possible t_1^is should be hit. We consider a clause c_i hitting t_1^i.

Then, since c_i hits t_1^i and does not hit t_0, $c_i = (x_i \vee \overline{x_{a_i}} \vee \overline{x_{b_i}})$ for some a_i and b_i. Thus, c_i does not suppress suppress t_1^j for each $j \neq i$, and the number of possible c_i is $\binom{n-1}{2}$.

Let X be a random variable defined to be the number of randomly chosen clauses from C_{t_0} required to hit all assignments in T_1. We first determine the expected value of X. Let $c_1, c_2, \ldots, c_X$ denote the clauses, where $c_i \in C_{t_0}$ denotes ith clause of $f_{G(n,m)}$. We divide the sequence into *epochs*, where epoch i begins with the clause following clause which hits a remaining assignment in T_1 and ends with the clause which hits another remaining assignment in T_1. Define the random variable X_i, for $0 \leq i \leq n-1$, to be the number of clauses in the ith epoch, so that $X = \sum_{i=0}^{n-1} X_i$.

Further, let p_i denote the probability of one of remaining assignments in T_1 is hit on any clause of the ith epoch. This is the probability of drawing a clause hitting one of the $n-i$ remaining assignments and so,

$$p_i = \frac{3(n-i)}{7n}.$$

The random variable X_i is geometrically distribution with parameter p_i. Thus, the expected value of X_i is $\frac{1}{p_i}$ and its variance is $\frac{(1-p_i)}{p_i^2}$.

By linearity of expectation,

$$E[X] = \sum_{i=0}^{n-1} E[X_i] = \frac{7n}{3} \sum_{i=0}^{n-1} \frac{1}{n-i} = \frac{7n}{3} \sum_{i=1}^{n} \frac{1}{i} = \frac{7n}{3} H_n,$$

where H_n is the nth Harmonic number that is asymptotically equal to $\ln n + \gamma + O\left(\frac{1}{n}\right)$.

Since the X_i's are independent, we can determine the variance of X as follows.

$$\begin{aligned}
\sigma_X^2 &= \sum_{i=0}^{n-1} \sigma_{X_i}^2 = \frac{49n^2}{9} \sum_{i=0}^{n-1} \frac{1}{(n-i)^2} - \frac{7n}{3} \sum_{i=0}^{n-1} \frac{1}{n-i} \\
&\leq \frac{49n^2}{9} \frac{\pi^2}{6} - \frac{7n}{3} H_n < \frac{49\pi^2}{54} n^2.
\end{aligned}$$

Hence $\sigma_X < \frac{7\pi}{3\sqrt{6}} n$. Now, we have

$$\begin{aligned}
\Pr[\text{when adding } m \text{ clauses, } f \text{ has unique solution}] &\leq \Pr[\forall t_1^i \in T_1 \text{ are hit}] \\
&= \Pr[X \leq m] \\
&= \Pr[|X - E[X]| \geq E[X] - m].
\end{aligned}$$

Since $m < \frac{7}{3} n \ln n + \left(\frac{7}{3}\gamma - \alpha\right) n$, $E[x] - m = \alpha n \geq \sqrt{\frac{1}{\varepsilon}} \sigma_X$. Thus, using the Chebyshev's inequality, we have

$$\Pr[|X - E[X]| \geq E[X] - m] \leq \Pr\left[|X - E[X]| \geq \sqrt{\frac{1}{\varepsilon}} \sigma_X\right] \leq \varepsilon.$$

This completes the proof of the theorem. $\square$

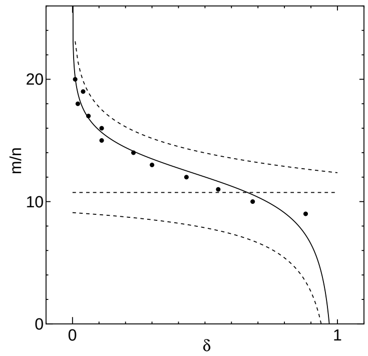

Figure 1: the ratio of m/n versus the ratio δ

3.3 Experimental Results

In theoretical analysis of the algorithm G, there is a gap between the upper and lower bounds of the number of clauses. In this section, we compare these theoretical bounds with the number of clauses that we obtained through experiments.

First we describe the setup used in our experiments. Our algorithm to check the unique solution formulae is based on a local search algorithm "Random Walk Strategy" by Selman et al.[SKC96]. We tune their algorithm to find another solution efficiency by setting strategy of choosing initial assignments. If $f_{G(n,m)}$ has another solution, we anticipate that it has tendency to be close to t_0. Hence we choose initial assignments with the following strategy.

1. all assignments of T_1.
2. t_n ($= 0^n$).
3. 20 randomly chosen assignments.

We fixed the number of variables is 100, and we selected the numbers of clauses being 900 to 2000 at intervals of 100. For each point, we checked the ratio of unique solution formulae of 100 formulae for chosen at uniformly random.

In Figure 1, along the horizontal axis we give the ratio of not unique solution formula δ, and along the vertical axis the ratio m/n, where m is the number of the clauses and n is the number of variables. Top and bottom dotted curves show theoretical bounds, center dotted line shows $\frac{7}{3}\ln n$. With this graph we assume that the relation of the ratio of number of clauses and number of variables with the ratio of not unique solution formulae is like $m/n = \frac{7}{3}\ln n - a\ln\delta - b\frac{1}{\sqrt{1-\delta}} + c$, and get $m/n = \frac{7}{3}\ln n - 1.82\ln\delta - 2.53\frac{1}{\sqrt{1-\delta}} + 2.59$ (solid curve).

4 Modifying the Algorithm

In the previous section , we show that the algorithm G generates unique solution instances with high probability when $m = \Theta(n\ln n)$. But $\Theta(n\ln n)$ clauses is too many compared with the cross over point, i.e., the number of clauses where ratio of satisfiable formulae is just half. For example, in the case of 3SAT, the cross over point is about $m = 4.2n$ [CA96].

Thus, we modify the algorithm to generate unique solution instances with less number of clauses. Since assignments in T_1 are likely to be other solutions, we add clauses which suppress all assignments in T_1.

Algorithm $G2$

input: number of variables n, number of clauses m;
begin
  choose a truth assignment $\hat{t} \in \{0,1\}^n$ randomly;
  let $\hat{t}_1^i$ $(1 \le i \le n)$ be an assignment different from $\hat{t}$ on only ith bit, x_i;
  for $i = 1$ **to** n
    do choose a clause which suppresses $\hat{t}_1^i$ randomly as ith clause c_i of f;
  for $i = n+1$ **to** m
    do choose a clause satisfied by $\hat{t}$ randomly as ith clause c_i of f;
  output $(f_{G2(n,m)}, \hat{t})$;
end.

Remind that we can assume that $\hat{t} = t_0 = 1^n$ without loss of generality.

4.1 Upper Bound

Theorem 4.1 *For any given fixed δ with $0 < \delta < 1$ and any fixed real ε with $0 < \varepsilon$, when $m \ge \max\left\{\frac{7}{6}(n+3)\ln\frac{2(1+\varepsilon)}{\varepsilon\delta}, \frac{7}{3}n\ln\frac{16(1+\varepsilon)}{3}\right\}$, $f_{G2(n,m+n)}$ has only one solution with probability at least $1-\delta$.*

Remark. We can choose *any* ε independent of δ. For example, setting $\varepsilon = 1$, we have the following linear upper bound:

$$\begin{cases} m \ge \frac{7}{6}(n+3)\ln\frac{4}{\delta} & \delta < \frac{4}{(32/3)^{\frac{2n}{n+3}}} \\ m \ge \frac{7}{3}n\ln\frac{32}{3} & \text{otherwise.} \end{cases}$$

Proof. We first consider the probability that any fixed t in T_k is not suppressed by the first n clauses. For each $c_1, c_2, \cdots, c_n$, we assume that c_i is the form $(x_i \vee \overline{x_{a_i}} \vee \overline{x_{b_i}})$. We denote bye $t(x_i)$ the ith value of the assignment t. Then two cases arise. In the case $t(x_i) = 1$, c_i does not suppress t. On the other hand, when $t(x_i) = 0$, c_i suppresses t with probability $\frac{n-k}{n-1}\frac{n-k-1}{n-2}$ since t has (n-k) 1s. Hence c_i does not suppress t with probability $1 - \frac{n-k}{n-1}\frac{n-k-1}{n-2} = \frac{(k-1)(2n-k-2)}{(n-1)(n-2}$. The first case occurs $(n-k)$ times, and the second case occurs k times. Thus, t is not suppressed by $c_1, c_2, \cdots, c_n$ with probability $\left\{\frac{(k-1)(2n-k-2)}{(n-1)(n-2}\right\}^k$. Since $\|T_k\|$ is $\binom{n}{k}$, the expected value of the number of the assignments in T_k which does not suppressed by the first n clauses is $\binom{n}{k}\left\{\frac{(k-1)(2n-k-2)}{(n-1)(n-2)}\right\}^k$. Let $E(n, n+m)$ be the expected value of the number of the assignments which does not suppressed by the first $n+m$ clauses. Then

$$E(n, n+m) = \sum_{k=2}^{n} \binom{n}{k} \left\{\frac{(k-1)(2n-k-2)}{(n-1)(n-2)}\right\}^k \left\{\frac{N_k}{N}\right\}^m.$$

Let p_i be the probability that the number of assignment which does not suppressed by the first $n+m$ clauses is i. Then, $E(n, n+m) = 0p_0 + 1p_1 + 2p_2 + \cdots > p_1 + p_2 + \cdots = 1 - p_0$. This implies that $p_0 > 1 - E(n, n+m)$. Thus it is sufficient to show that $E(n, n+m) < \delta$.

For short, we let $q(k) = \binom{n}{k}\left\{\frac{(k-1)(2n-k-2)}{(n-1)(n-2)}\right\}^k \left\{\frac{N_k}{N}\right\}^m$. It is sufficient to show that $\frac{q(k)}{q(k+1)} \ge 1 + \varepsilon$ and $q(2) \le \frac{\varepsilon\delta}{(1-\varepsilon)}$.

It is not difficult to see that, by a brute force, $\left\{\frac{q(k+1)}{q(k)}\right\} / \left\{\frac{q(k+2)}{q(k+1)}\right\} > 1$ for any positive integers n and k with $2 \le k \le n-2$. Hence our goal is now to show that $\frac{q(2)}{q(3)} \ge 1 + \varepsilon$ and $q(2) \le \frac{\varepsilon\delta}{(1-\varepsilon)}$.

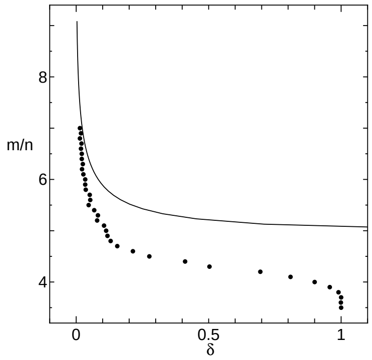

Figure 2: the ratio m/n versus the ratio δ

Here, since $m \geq \frac{7}{3} n \ln \frac{16(1+\varepsilon)}{3} \geq \frac{7}{3} \ln \frac{2(1+\varepsilon)(2n-5)^3}{3(n-1)(n-2)^2}$, and $\frac{N_3}{N_2} = 1 - \frac{3-21/n+36/n^2}{7n-30+59/n-60/n^2} \geq 1 - \frac{3}{7n}$, we have

$$
\begin{aligned}
\frac{q(2)}{q(3)} &= \frac{3(n-1)(n-2)^2}{2(2n-5)^3} \left\{ \frac{N_2}{N_3} \right\}^m \geq \frac{3(n-1)(n-2)^2}{2(2n-5)^3} \left(1 - \frac{3}{7n} \right)^m \\
&\geq \frac{3(n-1)(n-2)^2}{2(2n-5)^3} e^{\frac{3m}{7n}} \geq 1 + \varepsilon.
\end{aligned}
$$

On the other hand, since $m \geq \frac{7}{6}(n+3) \ln \frac{2(1+\varepsilon)}{\varepsilon\delta} \geq \frac{7n(n-1)}{6(n-2)} \ln \frac{2(n-2)(1+\varepsilon)}{(n-1)\varepsilon\delta}$, we get

$$
\begin{aligned}
q(2) &= \frac{2(n-2)}{(n-1)} \left\{ \frac{N_2}{N} \right\}^m = \frac{2(n-2)}{(n-1)} \left\{ \frac{7n^2 - 13n + 12}{7n^2 - 7n} \right\}^m \\
&= \frac{2(n-2)}{(n-1)} \left\{ 1 - \frac{6(n-2)}{7n(n-1)} \right\}^m \leq \frac{2(n-2)}{(n-1)} e^{-\frac{6(n-2)}{7n(n-1)} m} \leq \frac{\varepsilon\delta}{(1+\varepsilon)}.
\end{aligned}
$$

Combining all together, $E(n, n+m) \geq \delta$, which completes the proof. □

4.2 Experimental Results

In theoretical analysis of the algorithm $G2$, we can show only the upper bound of number of clauses. The same as G, we compare this theoretical upper bound with the number of clauses that we obtained through experiments.

The experiments are the same as the algorithm G except the choice of the initial assignments. Because $f_{G2(n,m)}$ has no solution in T_1, we change the strategy as follows.

1. $2n$ assignments in T_2 such that $x_i = 0$ and it is not suppressed by c_i.
2. t_n.
3. 20 randomly chosen assignments.

We fixed the number of variables is 100, and we selected the numbers of clauses being 350 to 700 at intervals of 10. For each point, we checked ratio of unique solution formulae of 100 formulae chosen at uniformly random. In Figure 2, along the horizontal axis we give the ratio of not unique solution formula δ , and along the vertical axis we give the ratio m/n, where m is the number of the clauses and n is the number of variables. Solid curve is points theoretical upper bound such that $m/n = \frac{7}{3}\ln\frac{16(1+\varepsilon)}{3} + 1$ and $\delta = \frac{2(1+\varepsilon)}{\varepsilon\left\{\frac{16}{3}(1+\varepsilon)\right\}^{\frac{2n}{n+3}}}$ (these values come from $\frac{7}{6}(n+3)\ln\frac{2(1+\varepsilon)}{\varepsilon\delta} = \frac{7}{3}n\ln\frac{16(1+\varepsilon)}{3}$).

With this graph, we can see that G2 can output unique solution formulae with high probability when number of clauses is larger than cross over point ($\approx 4.3n$).

References

[MR95] R. Motowani and P. Raghavan. *Randomized Algorithms.* Cambridge, 1995.

[SKC96] Bart Selman, Henry A. Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In David S. Johnson and Michael A. Trick, editors, *Cliques, Coloring, and Satisfiability*, Vol. 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 521–531 1996.

[CA96] James M. Crawford and Larry D. Auton. Experimental results on the crossover point in random 3-SAT. Artificial Intelligence, Vol.81, pp. 31–57, 1996.