Classes of Simplified NP-complete Satisfiability Problem *)

一些简化的 NP 完全满足性问题类*)

GONG Ping(龚 平) XIAO Hua(肖华) XU Daoyun(许道云)
(贵州大学计算机科学系 贵阳 550025)

Abstract (k,s)-SAT is the propositional satisfiable problem restricted to instances where each clause has exactly k distinct literal and every variable occurs at most s times. It is known that there exits an exponential function f such that for $s\leqslant f(k)$, all (k,s)-SAT instances are satisfiable, but $(k,f(k)+1)$-SAT is already. NP-$complete(k>3)$. Exact values of f are only known for $k=3$ and $k=4$, and it's open whether f is computable. In[2], S. Hoory and S. Sezider obtain a computable upper bound function for $f(k)(k\geqslant 3)$. The approach is to create some instance in(k,s)-SAT by calculation stairways, which are corresponded to constructing some formulas in $MU(1)$. However, the calculation for stairways is nondeterministic. It is dificult to determine the upper bounds of $f(k)$ for larger.

In this paper, a tree rule is introduced by reducing the steps of calculating stairways, and a deterministic calculation for stairways is presented to get the upper bounds for $f(k)(k\geqslant 3)$. The deterministic algorithm is practical, and the upper bounds are near the bounds S. Hoory and S. Sezider got.
Keywords $(k,s)-CNF$, $NP-complete$, $MU(1)$, $Tree\ rule$

1 Introduction

A literal is a propositional variable or a negated propositional variable. A clause C is a disjunction of literals, $C=(L_1\vee\cdots\vee L_m)$. A formula F in conjunctive normal form (CNF) is a conjunction of clauses, $F=(C_1\wedge\cdots\wedge C_n)$. We can view a clause$(L_1\vee\cdots\vee L_m)$as a set of literals$\{L_1,\cdots,L_m\}$, a CNF-formula $(C_1\wedge\cdots\wedge C_n)$ *as a set of clauses* $\{C_1,\cdots,C_n\}$ or a list of clauses $[C_1,\cdots,C_n]$. For CNF-formulas F_1 and $F2$, F_1+F_2 is conjunction of F_1 and F_2. The size of CNF formula F is defined as $size(F)=\sum_{c\in F}|C|$. $var(F)$ is the set of variables occurring in the formula F and $lit(F)$ is the set of literals over the variables of F. We denote $\#cl(F)$ as the number of clauses of F and $\#var(F)$ as the number of variables occurring in F. $CNF(n,m)$ is the class of CNF formulas with n variables and m clauses. The deficiency of a formula F is defined as $\#cl(F)-\#val(F)$, denoted by d(F). Let F be a CNF formula. We denote by $pos(x,F)$ $(resp.\ neg(x,F))$ the number of occurrences of positive (resp. negative) literal x in F, and define $occ(x,F)=pos(x,F)+neg(x,F)$. For fixed positive integers k and s, we denote by $(k,s)-CNT$ the set of formulas $F\in CNF$, and$(k,s)-SAT$ is the propositional satisfiability problem restricted to instances in $(k,s)-CNF$.

It was observed by Tovey[1] that all formulas in $(3,3)-CNF$ are satisfiable, and the satisfiability problem restricted to $(3,4)-CNF$ is already NP-complete. This was generalized in Kratochvil's work, where it is shown that for each $k\geq 3$, there is some integer $s=f(k)$, such that
(1) all formulas in $(k,s)-CNF$ are satisfiable, and
(2) $(k,s+1)-SAT$, the satisfiability problem restricted to $(k,s+1)-CNF$, is already $NP-$complete.

Therefore the function f can be defined by the equation

$$f(k)=\max\{s:(k,s)-CNF\cap UNSAT=\varnothing\}.$$

From[1], it follows that $f(3)=3$ and $f(k)\geq k$ for $k>3$. However it is open whether f is computable.

A formula F is minimal unsatisfiable(MU) if F is unsatisfiable and $F-\{C\}$ is satisfiable for any clause C of F. We denote $MU(k)$ as the set of minimal unsatisfiable formulas with deficiency $k\geq 1$. especially, $MU(1)$ is the set of minimal unsatisfiable formulas with deficiency one. It is known that an unsatisfiable formula contains a minimal unsatifiable subformula. Therefore, $f(k)=\max\{s:(k,s)-CNF\cap MU=\varnothing\}$.

The formulas in $MU(1)$ have a simple structure and can be constructed in a recursive way[4], it is easer to search for an unsatisfiable formula in$(k,s)-CNF\cap MU(1)$ than in$(k,s)-CNF\cap UNSAT$. In[2], S. Hoory and S. Szeider defined a new function:

$$f_1(k)=\max\{s:(k,s)-CNF\cap MU(1)=\varnothing\}.$$

Evidently, $f_1(k)\geq f(k)$. To construction required $MU(1)-formula$, they introduced a new concept of stairway, namely, a finite integer sequence of $\sigma=(a_1,a_2,\cdots,a_n)$, with $a_1\geq a_2\geq\cdots\geq a_n\geq 1$. For each $F\in(k,s)-CNF\cap MU(1)$, there is a calculatioin for stairways under some rules. However, the calculation stairways generating required $MU(1)$-formula is nondeterministic.

In this paper, we introduce tree rule in the calculating stairways, and present a deterministic calculation for stairways to get the upper bounds for $f(k)(k\geqslant 3)$. The deterministic algorithm is practical, and the upper bounds are near the bounds S. Hoory and S. Szeider got (see Table 1).

Table 1

$f(k)$	Sze[2]	this paper
$3\leq f(3)\leq$	3	3
$4\leq f(4)\leq$	4	4
$5\leq f(5)\leq$	7	7
$7\leq f(6)\leq$	11	11
$13\leq f(7)\leq$	17	19
$24\leq f(8)\leq$	29	33
$41\leq f(9)\leq$	51	58
$f(10)\leq$	no	101
$f(11)\leq$	no	189

*)项目来源:国家自然科学基金(批准号:60463001),贵州省自然科学基金,贵州省省长专项基金。龚 平 研究生,研究方向:计算复杂性,可计算分析。肖 华 研究生,研究方向:计算复杂性,可计算分析。许道云 教授,博士,研究方向:计算复杂性,可计算分析,非单调逻辑。

2 Preliminaries

In this section, we present some basic results about $MU(1)$ and calculation for stairways. Let F be a CNF formula with n variables $x_1,\cdots,x_n$ and m clauses $C_1,\cdots,C_m$. Let L be a literal and F a CNF formula, we define $L\vee_d F=\{L\vee f\mid f\in F\}$. It is easy to show the following lemma.

Lemma 1 For a pair of $MU(1)$-formulas F_1 and F_2 with $var(F_1)\cap var(F_2)=\emptyset$, let x be a new variable, then $F=x\vee_d F'_{11}+F'_{12}+\neg x\vee_d F'_{21}+F'_{22}$ is a formula in $MU(1)$, where $F'_{i1}+F'_{i2}=F, F'_{i1}\cap_d F'_{i2}=\emptyset$, and $F'_{ij}\neq\emptyset(i,j=1,2)$.

Clearly, if $t_1=occs(y,F_1)\leqslant s$ for any $y\in var(F_1)$ and $t_2=occs(z,F_2)\leq s$ for any $z\in var(F_2)$, and $t_1+t_2\leq s$, then $occs(x,F)\leq s$ for any $w\in var(F)$.

Furthermore, if F can be represented as $\{(x\vee f_1),\cdots,(x\vee f_{t_1}),f_{t_1+1},\cdots,f_{n_1},(\neg x\vee g_1),\cdots,(\neg x\vee g_{t_2}),g_{t_2+1},\cdots,g_{n_2}\}$, where $t_1,t_2\geqslant 1$, such that both $F_1=\{f_1,\cdots,f_{t_1},f_{t_1+1},\cdots,f_{n_1}\}$ and $F_2=\{g_1,\cdots,g_{t_1},g_{t_2+1},\cdots,g_{n_2}\}$ are $MU(1)$-formulas, and $var(F_1)\cap var(F_2)=\emptyset$, and for fixed $k>1$, if $|C|=k$ for each clause C in F, then $F\in(k,s)$-$CNF\cap MU(1)$.

Based on the above principle, we can construct formulas in (k,s)-$CNF\cap MU(1)$ by calculation of finite integer sequences.

We call a finite integer sequence $\sigma=(a_1,\cdots,a_n)$ *stairway*, if $a_1\geq\cdots\geq a_n>0$. ε is the empty sequence. For $1\leq i\leq n$, a_i is an *lentry* of σ, n is the length of σ, denoted as $|\sigma|=n$. For a finite sequence of non-negative integers σ, let σ^{ord} denote a stairway obtained from σ by removing $0'$s and reordering the entries nonincreasingly.

Definition 2 (D(s)-rule) For $s\geq 2$, let $\sigma_1=(a_1,\cdots,a_j)$ and $\sigma_2=(a_{j+1},\cdots,a_n)$ be two stairways, and let $(a_2,a_j,a_{j+2},\cdots,a_n)^{ord}=(b_1,\cdots,b_{n-2})$. *We put* $s'=\min(s,n)-2$, and define a stairway $\sigma_1\oplus_s\sigma_2=(a_1-1,a_{j+1}-1,b_1-1,\cdots,b_{s'}-1,b_{s'+1},\cdots,b_{n-2})^{ord}$. We say that $\sigma_1\oplus_s\sigma_2$ is a -saturated inference of σ_1 and σ_2 by the rule.

For a given integer $s\geq 2$ and a set Γ of stairways, $D(\Gamma,s)$ is the minimal set S satisfying the following conditions:
(1) $\Gamma\subseteq S$.
(2) if $\sigma_1,\sigma_2\in S$, then $\sigma_1\oplus_s\sigma_2\in S$.

We say that a stairway σ can be inferred from stairways set Γ by $D(s)$-rule, written as $\Gamma a_{D(s)}\sigma$, if only if $\sigma\in D(\Gamma,s)$. Especially, when stairways set is a singleton $\{\sigma_0\}$, then denote $\sigma_0 a_{D(s)}\sigma$ if $\sigma\in D(\{\sigma_0\},s)$.

For given stairways set Γ and a stairway σ, if $\Gamma a_{D(s)}\sigma$, then there exists a finite sequences of $\sigma_0,\sigma_1,\cdots,\sigma_n$ such that
(1) $\sigma_n=\sigma$, and
(2) for each $1\leqslant i\leqslant n$, there exist $h,l(0\leqslant h,l\leqslant i-1)$ such that $\sigma_i=\sigma_h\oplus_s\sigma_l$.

We call this sequence the D(s)-derivation sequence of σ from Γ.

Let a formula $F=\{C_1,\cdots,C_m\}\neq\emptyset$, with $1\leqslant|C_1|\leqslant\cdots\leqslant|C_m|\leqslant k$, and let n be the largest integer in $\{i,\cdots,m\}$ with $|C_n|<k$, we associate with F a stairway

$$\textstyle\sum_k(F)=(k-|C_1|,\cdots,k-|C_n|).$$

We have a general result.

Lemma 2 Let σ be a stairway, and $k>0, s>1$. If $(k)a_{D(s)}\sigma$, then there is a formula $F\in MU(1)$ with $\sum_k(F)=\sigma$ and $occs(x,F)\leq s$ for any $x\in var(F)$.

By lemma1and 2, we have

Lemma 3 Let F be a $MU(1)$-formula with $occs(x,F)\leq s$ for any $x\in var(F)$, and let $|C|\leq k$ for any $C\in F$. Then, $(k)a_{D(s)}\sigma$, where $\sum_k(F)=\sigma$.

By Lemma 2 and Lemma 3, we have

Corollary 1 $f_1(k)=\min\{s:(k)a_{D(s)}\varepsilon\}-1$.

It is known that for some, if $s>f_1(k)$, then the satisfiable problem restricted to (k,s)-CNF is *NP-complete*. We have easily a lower bound and an upper bound.

Lemma 4[1] For $k>1$, each formula $F\in(k,k)-CNF$ is satisfiable, and the satisfiable problem restricted to $(k,2^{k-1})-CNF$ is *NP-complete*.

In[2], it is known that $f_1(k)$ is a computable function.
Algorithm for $f_1(k)$
Input: k;
Output: s;
begin
$s_l=k; s_r=2^{k-1}$;
while $s_l\neq s_r-1$ do
{ $s_0=s_l+\lfloor\frac{s_r-s_l}{2}\rfloor$;
Computing $D(k,s)$
if $\varepsilon\in D(k,s)$ then $s_r=s_0$ else $s_l=s_0$; }
$s=s_l$;
output $s-1$;
end;
where $D(k,s)=D(\{(k)\},s)$.

In the above algorithm, the key of which is Computing $D(k,s)$. To end the computation of $D(k,s)$, the authors of[2] introduce a partial order of stairways, called domination. In fact, the purpose is to avoid loops of derivations and removing some worse derivations of stairways. (In following contents, we will use the word "domination" to mean that there is a loop or a worse stairway in the derivation.) However, the computation is still nondeterministic.

3 A deterministic computation

For statement convenience, for a stairway σ, we denote by $|\sigma|^-$ the number of non 1's integers in stairway.

Fact For stairways σ_1 and σ_2, if $|\sigma_1|^-+|\sigma_2|^->s$, then $|\sigma_1\oplus_s\sigma_2|=|\sigma_1|+|\sigma_2|>s$, and if $|\sigma_1|^-+|\sigma_2|^-<s$, then $|\sigma_1\oplus_s\sigma_2|=|\sigma_1|^-+|\sigma_2|^-<s$, then $|\sigma_1\oplus_s\sigma_2|=|\sigma_1|^-+|\sigma_2|^-+sign(|\sigma_1|+|\sigma_2|-s)$ (Where $sign(a)=a$, if $a>0$; otherwise, $sign(a)=0$).
The fact is mainly to determine the length of a new stairway after application D(s) rule to two given ones. It will be useful later in the Tree rule definition.

Definition 3 For stairways σ,σ' and integers k,s (where $|\sigma|\leqslant s-1$), we say σ' is inferred from σ and (k) by tree rule, if only if there exits a sequence of pairs $\langle\alpha_0,\beta_0\rangle,\cdots,\langle\alpha_n,\beta_n\rangle$ and $\alpha_{n+1}=\sigma'$, such that
1. $\alpha_0=\sigma$, and each β_i *is* (k) or α_j *with* $j\leqslant i$.
2. for each $i, 0\leqslant i\leqslant n, \alpha_{n+1}=\alpha_i\oplus_s\beta_i$.
3. for each pair $\langle\alpha_i,\beta_i\rangle, 0\leqslant i\leqslant n$, such that
- if α_i is dominated by α_j with $j<i$, then σ' is the result of α_j by tree rule.
- Otherwise:

(a) if $|\alpha_i|^-+|\sigma|^-+|\alpha_i|+|\sigma|<2s$ (namely, $|\alpha_i\oplus_s\sigma|<s$), then $\alpha_{i+1}=\alpha_i\oplus_s\sigma$.
(b) else $\beta_i=(k)$ and $\sigma'=\alpha_{i+1}=\alpha_i\oplus_s\beta_i$.

For statement conveniently, we denote by $\sigma'=Tree(\sigma,k,s)$

the procedure of calculation stairways σ and k by Tree rule. We show an example to specify Tree rule.

For integers k, s, we call a stairways sequence $\{\sigma_i\}_{i=0}^{n}$ the Tree(s)-derivation sequence of σ' from σ, if satisfies following conditions:
(1) $\sigma_0=\sigma$ and $\sigma_n=\sigma'$;
(2) $\sigma_{i+1}=Tree(\sigma_i, k, s)$, for $0\leqslant i<n$;
Clearly, Tree rule is a special case basing on D(s) rule. Therefore, we get a small lemma.

Lemma 5 For stairway σ and integers k, s, if there is a Tree(s)-derivation sequence of ϵ from σ, then σ $a_{D(s)}$ ϵ.

To get the bound as small as possible, it is necessary to add following two rules to Tree(s)-derivation to deal with dominations.

For a Tree(s)-derivation sequence $\{\sigma_i\}_{i=0}^{n}$: if there is a stairway σ_j dominates σ_l, with $0\leqslant j\leqslant l$, then the domination has happened. To go on derivation, we backtrack to some stairway and set σ_{l+1} as follows:

First, search a minimal stairway σ_j of this kind in lexicographic order, when there are several stairways dominating σ_l in the Tree(s)-derivation sequence. Then: if $\sigma_l=\sigma_j$, then set σ_{l+1} by one of following rules in which setting following $\sigma_x=\sigma_l$ (where deal with the loop); if σ_l is strictly dominated by σ_j, then set σ_{l+1} by one of following rules where set following $\sigma_x=\sigma_j$ (where deal with the situation of worse stairway).

Branching 1 rule: In procedure of $\sigma_x=Tree(\sigma_{x-1}, k, s)$, search a minimal stairway α_p in lexicographic order, which is not used by Branching 1 rule before and $|a_p\oplus_s\alpha_p|<s$, then $\sigma_{l+1}=Tree(\alpha_p, k, s)$; If no such stairway, $\sigma_{l+1}=$"$NULL$".

Branching 2 rule: In procedure of $\sigma_x=Tree(\sigma_{x-1}, k, s)$, search a minimal stairway α_m in lexicographic order, which is not used by Branching 2 rule before and $|\alpha_m\oplus_s\sigma_x|<2s$, if no such stairway, $\sigma_{l+1}=$"$NULL$"; Otherwise, set σ_{l+1} by following procedure:

```
begin
β0 = αm ⊕s σx; i = 1;
if |β0| ≥ |αm| then σl+1 = "NULL";
else
  while i > 0;
{   βi+1 = βi ⊕s σx;
    if |βi| ≤ |βi+1| then σl+1 = βi; and stop;
    else
      if |βi+1| ≤ ⌊ s/2 ⌋ then σl+1 = βi+1; and stop;
        else i = i+1; }
   end.
```

Please note that the above procedure is deterministic. For expression conveniently, we denote above two rules by $\sigma_{l+1}=B_1(\sigma_j, \sigma_l, k, s)$ and $\sigma_{l+1}=B_2(\sigma_j, \sigma_l, k, s)$, respectively.

Now, we make an extension of the definition of Tree(s)-derivation sequence.

For integers k, s, we call a stairways sequence $\{\sigma_i\}_{i=0}^{n}$ the Tree(s)-derivation sequence of σ' from σ, if satisfies following conditions:
(1) $\sigma_0=\sigma$ and $\sigma_n=\sigma'$;
(2) for each $0\leq i<n$, $\sigma_{i+1}=Tree(\sigma_i, k, s)$, or $\sigma_{i+1}=B_1(\sigma_l, \sigma_i, k, s)$ or $\sigma_{i+1}=B_2(\sigma_l, \sigma_i, k, s)$ for some $0\leqslant l\leqslant i$.

For integers k, s, when $\sigma=(k)$, we denote by $TDS(k, s)$ the Tree(s)-derivation sequence from (k). Therefore for integer k, we can define a new function:

$$f_2(k)=\min\{s: \epsilon\in TDS(k, s)\}-1.$$

Evidently, $f_2(k)\geqslant f(k)$. However, it can be computed by a deterministic algorithm.

```
Algorithm for f2(k)
Input: k.
Output: s.
begin
  sl = k; sr = 2^(k-1);
  while sl ≠ sr − 1 do
        s0 = sl + ⌊(sr − sl)/2⌋; Computing TDS(k, s0);
        if ε ∈ TDS(k, s0) then sr = s0 else sl = s0;
        output sl − 1
end;
TDS(k, s0)
begin
  σ0 = (k); i = 1;
  while σi−1 ≠ ε and σi−1 ≠ "NULL" do
  {σi = Tree(σi−1, k, s0);
     if the domination happens, then search a minimal stairways σj dominates σi in lexicographic order with j < i, and
σi+1 = B1(σj, σi, k, s);
      if σi+1 = "NULL" then;
σi+1 = B2(σj, σi, k, s); i = i+2;
         else i = i+2;
      else i = i+1; }
return();
end.
```

where the core of above algorithm is computing $TDS(k, s)$, which is deterministic, because it can stop for any inputs and return the Tree-derivation sequence the last stairway of which is ϵ or "$NULL$". By applying it, we have got some values of $f_2(k)$, which are near the values S. Hoory and S. Sezider got. (see table 1). Also, we have found some classes of NP-complete problems.

Theorem 1 The following classes contain Minimal unsatisfiable formulas: $(10, 102)-CNF$, $(11, 190)-CNF$. Hence, the satisfiability problem restricted to any of these classes is NP—complete.

Conclusion For determining the upper bound of $f(k)$ for integer k, S. Hoory and S. Szeider in[2] introduce an undeterministic algorithm. Basing on the work of them, in this paper, we have shown a deterministic algorithm to determine the upper bounds which are near the ones S. Hoory and S. Szeider got. Also, by applying it, some classes of NP-complete problems have been found.

Refeacter

1 Tovey C R. A simplified NP-complete satisfiability problem, Discrete Applied Mathematics, 1984. 85～89
2 Hoory S, Szeider S. Computing unsatisfiable k-SAT instances with few occurrences per varible, Conference SAT2004, Vancouver, BC, Canada
3 Aharoni R. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. Journal of Combinatorial Theory. Series A 43, 1996. 196～204
4 Davydov G, Davydova I, Kleine Büning H. An efficient algorithm for the minimal unsatisfiability problem for a subclass of C-. 1emN-. 1emF. Annals of Mathematics and Artificial Intelligence, 1998. 229～245
5 Kleine Büning H. On subclasses of minimal unsatisfiable formulas, Discrete Applied Mathematics, 2000. 83～98