6A-03

最大充足可能性問題の疎な例題に対する厳密アルゴリズムの改良

酒井　隆行　　　　　　　玉置　卓

京都大学情報学研究科

1　はじめに

最大充足可能性問題 (Max SAT) とは，入力として与えられる節集合に対して，充足する節の数が最大となる変数割当を求める問題である．節とはリテラルの論理和であり，リテラルとはブール変数とその否定である．Max SAT は NP 困難な問題の 1 つである．Max ℓ-SAT では，入力インスタンスに対して各節が高々ℓ 個のリテラルしか含まないという制限を課す．

n 変数，$m(= cn)$ 節からなる Max SAT のインスタンスが与えられたとき，自明に $O(m2^n)$ 時間で解ける．ある絶対定数 $\mu > 0$ が存在して，$O\left(\mathrm{poly}(m)2^{(1-\mu)n}\right)$ 時間で解けるかどうかは未解決問題である．現時点では Max SAT の特別な場合である充足可能性問題 (SAT) に対してさえ最良の上界は $O\left(\mathrm{poly}(m)2^{(1-\frac{1}{O(\log c)}n)}\right)$ 時間である [1, 6]．本研究でも，ある定数 $\mu(c) > 0$ に対して，$O\left(\mathrm{poly}(m)2^{(1-\mu(c))n}\right)$ 時間で Max SAT を解くアルゴリズムを与えることを目指す．既に，Dantsin と Wolpert [3] によって $\mu(c) = \Omega\left(\frac{1}{c\log c}\right)$ の指数領域決定性アルゴリズムが，Sakai と Seto と Tamaki [4] によって $\mu(c) = \Omega\left(\frac{1}{c^2\log^2 c}\right)$ の多項式領域決定性アルゴリズムと $\mu(c) = \Omega\left(\frac{1}{c\log^3 c}\right)$ の多項式領域乱択アルゴリズムが示されている．本研究では，以下の結果を得た．

定理 1. n 変数，cn 節のインスタンスが与えられたとき，*Max SAT* を $O\left(2^{(1-\mu(c))n}\right)$ 時間，多項式領域で解く決定性アルゴリズムが存在する．ここで，$\mu(c) = \Omega\left(\frac{1}{c\log c}\right)$ である．

このアルゴリズムは [4] の決定性アルゴリズムと同様に 2 つの技法を用いている．1 つは Schuler の width reduction [1, 6] であり，もう 1 つは Santhanam の greedy restriction [2, 5] である．本研究が [4] と異なる点は以下の通りである．[4] では始めに Max Formula SAT と呼ばれる問題に対するアルゴリズムを与え，それを Max ℓ-SAT を解くためのサブルーチンとして用いている．本研究では直接 Max ℓ-SAT を解くためのサブルーチンを設計する．本稿ではアルゴリズムの記述のみを与える．計算時間の解析は紙数が限られているため省略する．

2　準備

$V = \{x_1, \ldots, x_n\}$ をブール変数集合とする．ブール値が 'true' であるとき 1，'false' であるとき 0 で表現する．変数 $x \in V$ の否定を $\overline{x}$ と表記する．リテラルは変数またはその否定である．ℓ-*constraint* はブール関数 $\phi : \{0,1\}^n \to \{0,1\}$ であり，V の ℓ 変数に依存する．0-constraint は '0' または '1' である (定数関数)．Max SAT のインスタンス Φ は制約と重み関数の組の集合である．つまり， $\Phi = \{(\phi_1, w_1), \ldots, (\phi_m, w_m)\}$，ただし各 ϕ_i は ℓ_i-constraint で $w_i : \{0,1\} \to \{-\infty\} \cup \mathbb{Z}$ である．各制約 ϕ_i はリテラルの論理和の形をとる．重み関数 w に対して重み関数 $\widetilde{w}$ を $\widetilde{w}(1) = w(1), \widetilde{w}(0) = -\infty$ と定める．Φ の幅を $\max_i \ell_i$ とする．Max ℓ-SAT ではインスタンスの幅は高々ℓ に制限される．$\mathrm{Val}(\Phi, a) := \sum_{i=1}^{m} w_i \cdot \phi_i(a)$ とし，$\mathrm{Opt}(\Phi) := \max_{a\in\{0,1\}^n} \mathrm{Val}(\Phi, a)$ とする．

3　Max SAT アルゴリズム

まず，Max SAT アルゴリズムのサブルーチンとなる Max ℓ-SAT に対するアルゴリズムを説明する．

$\mathrm{var}(\phi_i)$ は ϕ_i に現れる変数の集合，$L(\phi_i)$ は ϕ_i に含まれるリテラル数，$\mathrm{freq}_{\phi_i}(x)$ は ϕ_i に変数 x が含まれるかどうかの特性関数を表す．$\mathrm{var}(\Phi) := \cup_{i=1}^{m}\mathrm{var}(\phi_i)$，$\widetilde{L}(\Phi) := \sum_{i:L(\phi_i)\geq 2} L(\phi_i)$，$\widetilde{\mathrm{freq}}_{\Phi}(x) := \sum_{i:L(\phi_i)\geq 2} \mathrm{freq}_{\phi_i}(x)$ と定義する．任意のインスタンス Φ，任意の変数集合 $\{x_{i_1}, \ldots, x_{i_k}\}$，任意の割当 $a_1, \ldots, a_k \in \{0,1\}$ に対して，$\Phi[x_{i_1} = a_1, \ldots, x_{i_k} = a_k] := \{(\phi'_1, w_1), \ldots, (\phi'_m, w_m)\}$ と定義する．ここで，$\phi'_i = \phi_i[x_{i_1} = a_1, \ldots, x_{i_k} = a_k]$ は制約 ϕ_i の依存する変数の一部に値を代入して簡単化することで得られる

制約を表す．この定義はリテラルの集合とそれらへ割当に対しても自然に拡張できる．

以下の補題が必要となる．

補題 1 (Lemma 1 [4])**.** *Max ℓ-SAT* のインスタンス Φ が与えられたとき，$\mathrm{Opt}(\Phi)$ は $\mathrm{poly}(m)2^{\widetilde{L}(\Phi)}$ 時間で計算できる．

図 1 に示したアルゴリズムについて以下が成り立つ．

定理 2. n 変数で $\widetilde{L}(\Phi) = cn$ の *Max ℓ-SAT* のインスタンス Φ が与えられたとき，**EvalFormula** は，$\mathrm{Opt}(\Phi)$ を $\mathrm{poly}(n)2^{\left(1-\Omega\left(\frac{1}{\ell c}\right)\right)n}$ 時間で計算できる．

[4] の Corollary 1 では $\mathrm{poly}(n)2^{\left(1-\Omega\left(\frac{1}{\ell^2 c^2}\right)\right)n}$ 時間となっていたことに注意されたい．

次に，Max ℓ-SAT に対するアルゴリズムと width reduction の組み合わせに基づいた，Max SAT の疎なインスタンスに対するアルゴリズムを図 2 に示す．**MaxSAT** は定理 1 で述べた計算時間を達成する．アルゴリズムの正しさは次の主張により保証される．

主張 1. **MaxSAT** において，もし *3* 行目の **else** の条件を満たすなら，$\mathrm{Opt}(\Phi) = \max\{K_L, K_R\}$ である．

参考文献

[1] C. Calabro, R. Impagliazzo, and R. Paturi. A duality between clause width and clause density for SAT. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity (CCC)*, pages 252–260, 2006.

[2] R. Chen, V. Kabanets, A. Kolokolova, R. Shaltiel, and D. Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, TR13-057, 2013. Also In: Proceedings of the 29th Conference on Computational Complexity (CCC), pp. 262–273, (2014).

[3] E. Dantsin and A. Wolpert. MAX-SAT for formulas with constant clause density can be solved faster than in $O(2^n)$ time. In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 4121 of *Lecture Notes in Computer Science*, pages 266–276. Springer, 2006.

[4] T. Sakai, K. Seto, and S. Tamaki. Solving sparse instances of max SAT via width reduction and greedy restriction. *Theory of Computing Systems.* To appear.

[5] R. Santhanam. Fighting perebor: New and improved algorithms for formula and QBF satisfiability. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 183–192, 2010.

[6] R. Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005.

EvalFormula($\Phi = \{(\phi_1, w_1), \ldots, (\phi_m, w_m)\}$: instance, n: integer)
01: **if** $\widetilde{L}(\Phi) = cn < 3n/4$,
02: compute $\mathrm{Opt}(\Phi)$ by Lemma 1 and return $\mathrm{Opt}(\Phi)$.
03: **else**
04: $x = \arg\max_{x \in \mathrm{var}(\Phi)} \widetilde{\mathrm{freq}}_\Phi(x)$.
05: $K_0 \leftarrow$ **EvalFormula**$(\Phi[x=0], n-1)$.
06: $K_1 \leftarrow$ **EvalFormula**$(\Phi[x=1], n-1)$.
07: return $\max\{K_0, K_1\}$.

図 1: Max ℓ-SAT アルゴリズム

MaxSAT($\Phi = \{(\phi_1, w_1), \ldots, (\phi_m, w_m)\}$: instance, ℓ, n: integer)
01: **if** $\forall \phi_i \in \Phi, |\mathrm{var}(\phi_i)| \le \ell$,
02: return **EvalFormula**(Φ, n).
03: **else**
04: Pick arbitrary $\phi_i = (l_1 \vee \cdots \vee l_{\ell'})$ such that $\ell' > \ell$.
05: $\Phi_L \leftarrow \{\Phi \setminus \{(\phi_i, w_i)\}\} \cup \{(l_1 \vee \cdots \vee l_\ell, \widetilde{w}_i)\}$.
06: $K_L \leftarrow$ **MaxSAT**(Φ_L, ℓ, n).
07: $\Phi_R \leftarrow \Phi[l_1 = \cdots = l_\ell = 0]$.
08: $K_R \leftarrow$ **MaxSAT**$(\Phi_R, \ell, n-\ell)$.
09: return $\max\{K_L, K_R\}$.

図 2: Max SAT アルゴリズム