The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

SAT ソルバー Glucose を用いた MCS 列挙

MCSes Enumeration with the Glucose SAT Solver

越村 三幸 *1　佐藤 健 *2
Miyuki Koshimura　Ken Satoh

*1九州大学 大学院システム情報科学研究院
Faculty of Information and Electrical Engineering, Kyushu University

*2国立情報学研究所
National Institute of Informatics

Enumerating all Maximal Satisfiable Subsets (MSSes) or all Minimal Correction Subsets (MCSes) of an unsatisfiable CNF Boolean formula is a cornerstone task in various AI domains. This paper considers MCSes enumeration with a SAT solver. We aim to develop a procedure which outperforms several MCSes enumerators proposed so far. The paper presents a basic enumeration procedure and compares it with a state-of-the-art enumerator `Enum-ELS-RMR-Cache`. The experimental results show that the proposed procedure is more efficient than `Enum-ELS-RMR-Cache` to solve Partial-MaxSAT instances but it is inefficient than `Enum-ELS-RMR-Cache` to solve plain MaxSAT instances.

1. はじめに

解を一つも持たない制約集合の MSS（Maximal Satisfiable Subset）あるいは MCS（Minimal Correction Subset）を求めることは，人工知能の様々な分野で重要とされている [Grégoire 18]．本稿では，SAT ソルバーを利用した MCS の列挙を論ずる．同様の研究は多くあるが，我々は，効率性の点でそれらを上回る列挙手続きの提案と実装を目的としている．本稿ではその基本手続きを示し，その実装の性能を現時点で最高性能を示していると思われる `Enum-ELS-RMR-Cache`[Grégoire 18] と比較する．

2. 準備

本稿では，問題は節集合で与えられるものとする．節（clause）はリテラル（literal）の選言（論理和 $\vee$）でリテラルはブール変数あるいはその否定（$\neg$）である．ブール変数の集合を Var とする．変数の値割当て μ は，Var から $\{0,1\}$ の写像である．節集合 Σ に現れる変数の値割当て μ で Σ の全ての節を 1 にするものがある時，Σ は充足可能（satisfiable），そのような μ がない時，充足不能（unsatisfiable）であるという．

本稿では二種類の節，ハード節とソフト節を扱う．ハード節は必ず成り立って欲しい制約，ソフト節は出来るだけ成り立って欲しい制約を表す．ハード節の集合 Σ_1 とソフト節の集合 Σ_2 からなる節集合 $\Sigma\ (= \Sigma_1 \cup \Sigma_2)$ に対して，MSS と MCS は次のように定義される．

定義 1 (MSS) Σ の *MSS*（*maximal satisfiable subset*）Φ は，Σ_1 を含む Σ の部分集合（$\Sigma_1 \subseteq \Phi \subseteq \Sigma$）で，充足可能，かつ，$\forall\alpha \in \Sigma \setminus \Phi, \Phi \cup \{\alpha\}$ は充足不能，を満たすものである．

MSS は充足可能性を保ちつつ，Σ_1 を Σ_2 のソフト節でぎりぎりまで拡大した節集合である．

定義 2 (MCS) Σ の *MCS*（*minimal correction subset*）Ψ は，Σ の部分集合（$\Psi \subseteq \Sigma$）で，その補集合，つまり $\Sigma \setminus \Psi$ が Σ の *MSS* であるものである．

連絡先: 越村三幸，九州大学大学院システム情報科学研究院，〒 819-0395 福岡市西区元岡 744，092-802-3599，koshi@inf.kyushu-u.ac.jp

MSS は Σ_1 を含むので，MCS は Σ_2 の部分集合である．MCS は Σ_2 の部分集合で，それを取り除くと Σ が充足可能になる，ぎりぎりまで小さい節集合である．

SAT ソルバーを利用した MCS 列挙プログラムではしばしば，ソフト節に対し節選択変数（clause selector）と呼ばれる変数が次のように導入される．それぞれのソフト節 $\alpha \in \Sigma_2$ に対し，新変数 s_α を導入し，その否定を元のソフト節に加えた $\alpha \vee \neg s_\alpha$ を作る．これにより，新しい節集合 $\Sigma_2^S = \{\alpha \vee \neg s_\alpha \mid \alpha \in \Sigma_2\}$ が得られる．そして，SAT ソルバーを，Σ そのものでなく，$\Sigma_1 \cup \Sigma_2^S$ の充足可能性判定に利用する．

$s_\alpha = 1$ の仮定の下では，$\alpha \vee \neg s_\alpha$ と元の節 α は充足可能性が一致するので，$s_\alpha = 1$ とすることを $\alpha \vee \neg s_\alpha$ を活性化（activate）する，という．逆に $s_\alpha = 0$ とすることを非活性化（deactivate）する，という．MSS を求めることは，$\Sigma_1 \cup \Sigma_2^S$ の充足可能性を保ちつつ，活性化された選択変数をぎりぎりまで増やす，ことに対応する．言い換えると，MCS を求めることは，充足可能性を保ちつつ，非活性化された選択変数をぎりぎりまで減らす，ことに対応する．このことを利用して，次節のアルゴリズムは MCS を列挙する．

3. MCS 列挙アルゴリズム

Algorithm 1 に提案アルゴリズムの概観を示す．6 行目の $SAT(\Sigma_1 \cup \Sigma_2^S, A)$ が SAT オラクルを示し，A に含まれるリテラルの値が全て 1 と仮定した時の $\Sigma_1 \cup \Sigma_2^S$ の充足可能性を判定する．充足可能であれば st が $TRUE$ となり，見つけた変数の値割当て μ が得られる．充足不能であれば，st が $FALSE$ となる．

本アルゴリズムは，SAT オラクルが見つけた変数の値割当てを出発点として MCS を探していく．アルゴリズム中の A と B は選択変数の集合であり，A に対応するソフト節の集合 $\{\alpha \mid \mu(s_\alpha) = 1\}$（以降 A_{MSS}）が MSS の候補であり，B に対応するソフト節の集合 $\{\alpha \mid \mu(s_\alpha) = 0\}$（以降 B_{MCS}）が MCS の候補である．このアルゴリズムでは，SAT オラクルが充足可能を返す限り，6〜10 行目が繰り返され，B は段々と小さくなり B_{MCS} は MCS に近づいていく．逆に A は段々と大きくなり A_{MSS} は MSS に近づいていく．10 行目の $\bigvee_{s_\alpha \in B} s_\alpha$ は，B_{MCS} より大きな集合を以降の MCS の探索から除外す

Algorithm 1 Enum-MCS (Σ の全ての MCS を列挙する)

入力： $\Sigma(=\Sigma_1 \cup \Sigma_2)$（充足不可能な節集合，$\Sigma_1$ はハード節の集合，Σ_2 はソフト節の集合）

出力： Σ の全ての MCS

1: $\Sigma_2^S \leftarrow \{\alpha \vee \neg s_\alpha \mid \alpha \in \Sigma_2\}$;　// s_α は α の選択変数
2: $S \leftarrow \{s_\alpha \mid \alpha \in \Sigma_2\}$;　　// 選択変数の集合
3: $A \leftarrow \emptyset$;　// MSS 候補に対応する選択変数の集合
4: $B \leftarrow \emptyset$;　// MCS 候補に対応する選択変数の集合
5: **while true do**
6: 　$(st, \mu) = SAT(\Sigma_1 \cup \Sigma_2^S, A)$　// SAT オラクル
7: 　**if** $st = TRUE$ **then**
8: 　　$A \leftarrow \{s_\alpha \mid \mu(s_\alpha) = 1\}$
9: 　　$B \leftarrow \{s_\alpha \mid \mu(s_\alpha) = 0\}$
10: 　　$\Sigma_1 \Leftarrow \Sigma_1 \cup (\bigvee_{s_\alpha \in B} s_\alpha)$　// 阻止節
11: 　**else if** $A = \emptyset$ **then**
12: 　　**return**
13: 　**else**
14: 　　$output(B)$; // B_{MCS} が MCS
15: 　　$A \leftarrow \emptyset$;
16: 　**end if**
17: **end while**

表 1: 変数と節の個数の平均

	変数の数	節の数	
		ハード節	ソフト節
MS	156,844	0	496,827
PMS	16,032	100,134	10,759

るための節である．

SAT オラクルの答えが充足不能なら，その時の B_{MCS} より小さな MCS 候補はない，ことになり，この B_{MCS} が MCS であることわかる（14 行目）．15 行目の $A \leftarrow \emptyset$ は，MSS 候補がない状態から再び探索を始めることを表す．SAT オラクルの答えが充足不能で A が空なら，もうこれ以上 MCS はないことが分かり，手続きを終了する（11，12 行目）．

なお，本アルゴリズムは，各選択変数に否定をつけたリテラル集合 $S^- = \{\neg s_\alpha \mid \alpha \in \Sigma_2\}$ に関する極小モデルを列挙している，と見なすことができ，[Koshimura 09] の極小モデル生成と手続き的には同じである．

4.　計算機実験

前節のアルゴリズムを SAT ソルバー Glucose 3.0[Audemard 09, Eén 03] を用いて実装した．評価には，[Grégoire 18] と同じ 1090 個のインスタンスを用いた．その内訳は，ソフト節のみからなるインスタンス（以降 MS）が 493 個，ソフト節とハード節からなるインスタンス（以降 PMS）が 597 個である．

実験には，メモリ 32Gb を有する Intel Xeon E3-1246v6(3.70GHz) プロセッサ上の Ubuntu 18.04 を用いた．1 インスタンス当たりの制限時間と制限メモリは，[Grégoire 18] と同じく，それぞれ，30 分，8Gb とした．

表 1 は，MS，PMS，それぞれのインスタンス当たりの変数の数と節の数の平均を示している．MS の方が，変数の数，節の数のいずれも平均的には多いことが分かる．表 2 に，上記の制限時間及び制限メモリで列挙した MCS の総和を MS，PMS 毎に示した．比較のため [Grégoire 18] で提案された MCS 列挙プログラム `Enum-ELS-RMR-Cache`[*1] の列挙数も示した．これは，現段階で最も高速に MCS を列挙するプログラムの一つである．

表 2: MCS 総列挙数（単位：千個）

	`Enum-ELS-RMR-Cache`	提案手法
MS	349,155	60,173
PMS	494,308	637,800

表から，MS に対しては `Enum-ELS-RMR-Cache` が，PMS に対しては提案手法が優位であるのが分かる．現実的な問題はハード節とソフト節の双方を用いて表現されることが多いと思われるので，この結果から，提案手法の方が現実的な問題に適用するには優位である，と言ってもいいだろう．

提案手法の性能が MS に対して芳しくない原因の一つとして，MS のソフト節が多い，ことが考えられる．提案方式では，1 つのソフト節に対して 1 つの選択変数が導入される．現実装では，選択変数の連言を Glucose の assumption に設定し，MCS を求める．ソフト節が増えると連言，つまり assumption も多くなるが，これが多くなると SAT ソルバーの性能が落ちていくことが知られている．MS に対する性能を上げるには，これに何らかの対応が必要だと思われる．先駆的な研究 [Audemard 13] を足がかりとしたい．

5.　おわりに

SAT オラクルを利用して MCS を列挙する手続きを提案し，SAT ソルバー Glucose を用いて実装した．`Enum-ELS-RMR-Cache` との比較実験では，提案手法は PMS に対しては優位であったが，MS に対しては劣っていた．本提案手法は，`Enum-ELS-RMR-Cache` の手続きと比べると簡潔である．それにも関わらず，PMS に対して優位性を示せたことは本手続きの将来性の高さを示していると思われる．今後は，実験結果を詳細に分析し，手続きの改良手法を考えていきたい．

謝辞：本研究は JSPS 科研費 JP16K00304，JP17K00307 の助成を受けたものです．

参考文献

[Audemard 09] G. Audemard, L. Simon: Predicting Learnt Clauses Quality in Modern SAT Solver, IJCAI-09, pp.399-404, 2009.

[Audemard 13] G. Audemard, J.-M. Lagniez, L. Simon: Improving Glucose for Incremental SAT Solving with Assumption: Application to MUS Extraction, SAT 2013, pp.309-317, 2013.

[Eén 03] N. Eén, N. Sörensson: An Extensible SAT-solver, SAT-03, pp.502-518, 2003.

[Grégoire 18] É. Grégoire, Y. Izza, J.-M. Lagniez: Boosting MCSes Enumeration, IJCAI-18, pp.1309-1315, 2018.

[Koshimura 09] M. Koshimura, H. Nabeshima, H. Fujita, R. Hasegawa: Minimal Model Generation with Respect to an Atom Set, FTP 2009, pp.49-59, 2009.

*1 http://www.cril.fr/enumcs から入手した．なお，実験で用いた 1090 個のインスタンスも同サイトから入手した．