

# 拡張融合法に基づく次世代SATソルバーの試作

Towards Next-Generation SAT Solvers based on Extended Resolution

森 淳\*<sup>1</sup>      鍋島 英知\*<sup>2</sup>  
Jun Mori      Hidetomo Nabeshima

\*<sup>1</sup>Growth xPartners    グロースエキスパートナーズ  
Growth xPartners

\*<sup>2</sup>山梨大学大学院医学工学総合研究部

Department of Research Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

In this paper, we propose a new SAT solver based on extended resolution. Extended resolution is expected as a basis of next-generation SAT solvers, since it is more powerful than general resolution. There are some solvers based on extended resolution. Although those solvers can show great improvement for a certain kind of instances such as pigeonhole problems, it is difficult to get improvement for various kinds of instances. We propose a new applied condition of extension rule which has a key role in extended resolution. The condition is based on the cause of successive conflicts. The experimental results show that our proposed SAT solver can improve the performance for a various kinds of instances.

## 1. はじめに

命題論理の充足可能性問題 (propositional satisfiability testing; SAT 問題) は, ハードウェアやソフトウェアの検証問題やプランニング問題, スケジューリング問題など実世界の様々な分野に応用されている. そのため, SAT 問題を高速に解く SAT ソルバーの研究が盛んに行われている. その SAT ソルバーにおいて現在主流となっている手法が矛盾からの節学習 (conflict-driven clause learning; CDCL) [3, 8] に基づくソルバーであり, CDCL ソルバーと呼ばれる. CDCL ソルバーは, 証明システムとして融合法に相当する証明能力を持つことが知られている [7].

この融合法よりも強力な証明手法の 1 つに拡張融合法 [9] がある. 拡張融合法は, 融合法に拡張規則を追加した比較的単純な枠組みにもかかわらず, 融合法では証明が困難な鳩ノ巣問題のような問題に対して短い証明を構成可能であることが知られており [4], 次世代 SAT の基盤アルゴリズムとして注目されている. 拡張融合法に基づく SAT ソルバーはいくつか存在し [1, 6], 特定の問題群において CDCL ソルバーよりも良い性能を示すことが実証されているが, 多様な問題群において安定した性能向上を果たしているとはいえない. そこで本稿では, 連続した矛盾から得られる学習節群に基づき拡張規則を適用する SAT ソルバーを提案する. そして評価実験の結果, SAT 競技会で使われる多様な問題集において性能向上が得られることを示し, 今後の次世代の SAT ソルバー開発の指針としたい.

## 2. CDCL 手続き

命題変数またはその否定をリテラルと呼び, リテラルの選言を節と呼ぶ. SAT 問題は通常, 節の連言で表現され, これを連言標準形 (conjunctive normal form; CNF) と呼ぶ. 例えば  $(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee x_4)$  のように表記される. こ

で  $x_i$  は命題変数である.

現在多くの SAT ソルバーの基本アルゴリズムである CDCL 手続き [3, 8] について概説する. CDCL 手続きでは, まず (1) 単位節 (真偽値が未割り当てのリテラルを 1 つだけ含み, 残りのリテラルが偽の節) があれば, その節を充足するため, 未割り当てリテラルに真を割り当てる単位伝搬を繰り返す. もし, 単位節が存在しなければ, (2) 変数選択ヒューリスティクスに従い適当な変数を選択し, それに真もしくは偽を割り当て, (1) に戻る. ここで, 変数選択ヒューリスティクスにより選択された変数を決定変数 (decision variable), 決定変数への値割り当てを決定 (decision), 決定変数の数を決定レベル (decision level) と呼ぶ. もし (1) の単位伝搬の過程で矛盾が発生 (ある変数  $x$  に対し,  $x$  を含む単位節と  $\neg x$  を含む単位節が同時に出現) した場合, 矛盾発生の原因を解析し, その矛盾発生を防ぐ節を学習した後, 適切な決定レベルまでバックジャンプする. 例えば, 矛盾の原因が変数  $a, b, c$  に真を割り当てたことであった場合 (すなわち  $a \wedge b \wedge c$ ), この原因を否定した節  $\neg a \vee \neg b \vee \neg c$  を学習する. この学習節により, 再び  $a, b, c$  に同時に真が割り当てられることを防ぐ. その後, 学習節が単位節となる決定レベルまでバックジャンプする. これが CDCL ソルバーにおける基本的な動作の流れである.

学習節は同様な探索失敗の繰り返しを避けるための有用な知識であるが, 矛盾のたびに獲得されるため, その数は膨大なものになる. すべての学習節を保持することはメモリ不足をもたらす. 単位伝搬速度を低下させるため困難である. よって多くの CDCL ソルバーでは, 今後の探索に有用と考えられる学習節を残し, そうでない学習節を定期的に削除する. Audemard らにより提案された学習節評価尺度 LBD (Literal blocks distance) [2] は, 優れた評価尺度の 1 つであり, LBD を用いたソルバーは, SAT 競技会においても優秀な成績を示している. ある学習節の LBD は, その節に含まれるリテラルブロックの数で与えられる. リテラルブロックとは, ある決定レベルにおいて値が割り当てられたリテラル集合を指す. ブロックに含まれるリテラル群は, 今後の探索においてもある決定レベルにおいて一斉に値が割り当てられることが期待できる. すなわち, ブロックをあたかも 1 つのリテラルとみなすこ

連絡先: 山梨大学大学院医学工学総合教育コンピュータ・メディア工学専攻, 〒400-8511 山梨県甲府市武田 4-3-11, E-mail: mori@nabelab.org

とができ、LBD の小さい学習節ほど強い制約として働く有用な節と考えることができる。

また多くの CDCL ソルバーでは、特定の条件を満たしたときに探索を途中で打ち切り、初めからやり直すリスタート戦略 [5] も併用している。ただし、リスタート前に獲得した学習節は保有し続けるため、リスタート後に同様の矛盾発生を繰り返すことはない。リスタート戦略は、ソルバーが探索する空間に変化を与えることでソルバーの性能を安定させる優れた手法である。

### 3. 拡張融合法 [9]

拡張融合法は、融合法における融合規則に加え、拡張規則を用いる証明手法である。まず融合規則の定義を以下に示す。

$$\frac{x \vee y_1 \vee \dots \vee y_n \quad \neg x \vee z_1 \vee \dots \vee z_m}{y_1 \vee \dots \vee y_n \vee z_1 \vee \dots \vee z_m}$$

ここで  $x, y_i, z_j$  はリテラルである。

拡張規則とは、任意の命題論理式と等価な新しい命題変数を導入する規則であり、以下で定義される。

$$\overline{x \leftrightarrow F}$$

ここで  $F$  は任意の論理式、 $x$  は原問題に含まれない新規変数である。上記推論規則は、証明過程の任意の時点で、任意の式に別名を与えることができることを意味する。本稿では、 $F$  としてリテラルの選言のみ考慮する。例えば、 $F = y_1 \vee y_2 \vee y_3$  とすると、上記拡張規則は、次の 4 つの節

$$\begin{aligned} &(\neg x \vee y_1 \vee y_2 \vee y_3) \\ &(\neg y_1 \vee x) \\ &(\neg y_2 \vee x) \\ &(\neg y_3 \vee x) \end{aligned} \quad (1)$$

を追加する。

拡張融合法は、融合規則を含む点で、少なくとも融合法と同等の証明能力を持ち、さらに拡張規則によって、融合法よりも短い証明を与えることが可能である [4]。

拡張規則は任意の時点で任意の式と等価な新規変数を導入する規則であるため、際限なく適用することは、変数と節数の増大をもたらす現実的ではない。よって、拡張規則を適用する場合は、何らかの条件を満たした場合に制限する必要がある。どのような式に対してどのタイミングで別名を与えるかは、拡張融合法を機械的に自動処理する上で重要な問題となる。

本稿では、上記  $F$  としてリテラルの選言のみを考慮するが、その利点を式 (1) を例にとりながら以下に示す。まず 1 つめは、節長の短縮である。もし  $y_1 \vee y_2 \vee y_3$  という式が、節集合中に頻出する場合、それらを  $x$  で置き換えることは節長の短縮をもたらす。もし十分な頻度で出現しているならば、これは単位伝搬速度の低下を防ぐ効果が期待できる。もう 1 つの重要な利点は、AND 関係が内在した節のコンパクトな学習が可能になることである。例えば

$$(\neg x \vee z_1 \vee \dots \vee z_n)$$

という節を学習した場合、 $\neg x \leftrightarrow \neg y_1 \wedge \neg y_2 \wedge \neg y_3$  であるので、これは以下の 3 つの節を 1 つの節でコンパクトに表現す

表 1: Glucose と GlucosER の求解数比較

ソルバー	求解数 (全 123 問)
Glucose	62 問
GlucosER	61 問

ることに相当する。リテラル  $\neg x$  には、AND 関係が内在していると考えられる。

$$\begin{aligned} &(\neg y_1 \vee z_1 \vee \dots \vee z_n) \\ &(\neg y_2 \vee z_1 \vee \dots \vee z_n) \\ &(\neg y_3 \vee z_1 \vee \dots \vee z_n) \end{aligned}$$

CDCL ソルバーが、それまでの DPLL ソルバーと比較して大きく性能を伸ばした最大の要因は、節学習にある。拡張規則は、その学習節集合をよりコンパクトにする可能性があり、また OR 制約だけでなく、AND 関係を内在した制約の学習を可能にする。この点からも拡張融合法は、CDCL ソルバーの探索能力を押し上げる可能性がある。

### 4. 関連研究

拡張融合法を導入したソルバーに Audemard らが開発した GlucosER [1] がある。GlucosER は、SAT 競技会で優秀な成績を示した Glucose[2] を基に開発され、学習節評価尺度として LBD が用いられている。

GlucosER での拡張規則の適用条件を示す。GlucosER は、 $(l_1 \vee \alpha), (l_2 \vee \alpha)$  ( $\alpha$  はリテラルの選言) のような、1 つのリテラルだけが異なり、残りのリテラル群が等しい学習節を連続して獲得したとき、式  $l_1 \vee l_2$  に別名を与える。拡張融合法は、別名を与えたりテラル群が矛盾の原因となった場合に新規変数を用いて学習節を圧縮する効果を発揮する。そのため矛盾の原因となるリテラルの組み合わせに対し別名を与えなければならない。GlucosER では、連続で似た学習節が得られた場合に、そこに含まれているリテラル群は矛盾の原因として再登場する可能性が高いと判断していると考えられる。

GlucosER の性能を評価するため、その基となった Glucose 1.0 との比較検証実験を行った。実験に使用した問題は、SAT 競技会 2009 の Application 部門における充足不能な問題 123 問を用いた。実験環境は Core Duo 1.66Hz, 2GB RAM である。1 問あたりの制限時間は 600 秒とし、求解数を評価した。各ソルバにおける求解数を表 1 に、求解時間の分布を図 1 に示す。これらの結果から、Glucose の性能は、拡張融合法の導入によって向上しておらず、若干低下していることが分かる。その原因としては、まず拡張規則で選ばれたリテラル群が拡張後に矛盾の原因として再登場することが少なく、拡張を行うためのコストが上回ってしまう場合が考えられる。また、新規変数は学習節評価尺度 LBD において、その性質から新規変数を含んだ節のブロック数を増やしてしまう傾向にある。そのため、新規変数を含んだコンパクトな学習節に悪い評価を与え、結果として探索に役立てる前に破棄されてしまった可能性も考えられる。

### 5. 拡張規則の適用条件の提案

GlucosER では、拡張規則により 2 つのリテラルに別名を与えるのみである。本研究では、より柔軟な拡張を目指し、別

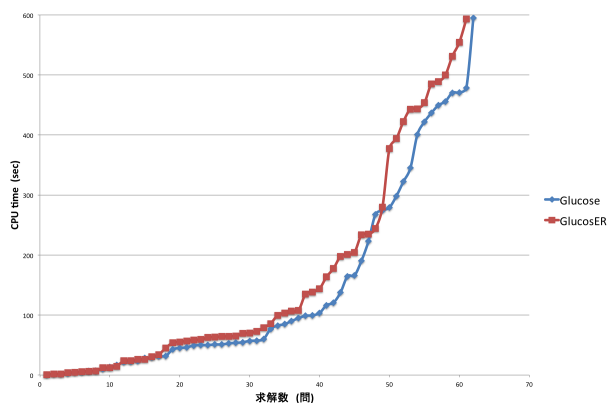


図 1: Glucose と GlucosER における求解時間の分布

名を与えるリテラル集合のサイズを制限しない方針をとる。また別名を与えるリテラル集合は、GlucosER と同様に、拡張後に矛盾の原因として再登場する可能性が高いリテラルの組み合わせを選択したい。よって GlucosER 同様に学習節群から別名を与えるリテラル群を選出する手法をとる。

本研究ではバックジャンプ直後の矛盾発生に注目する。多くのソルバーでは、矛盾が発生したとき、その矛盾の原因が複数あったとしても、そのうちの 1 つのみを抽出する。この場合、バックジャンプ直後の単位伝搬において、しばしば残りの矛盾の原因が解消されず、再びすぐに矛盾が発生することになる。従って、矛盾の原因が複数ある場合は、1 つずつ矛盾を解消しながら、その都度学習節を獲得しバックジャンプすることを繰り返すことになる。いくつかの問題を選択して評価したところ、連続して矛盾が発生する場合は全体の約 75% を占めた。また、この過程から得られる学習節群には、しばしば共通のリテラル群が含まれている。先の問題群において 5 つの矛盾が連続発生した場合、平均で 17 個のリテラルが共通要素として存在した。これらの共通のリテラル群は、複数の矛盾の原因の共通基盤と考えることができ、今後の探索においても、矛盾を引き起こす要因となることが期待できる。

そこで我々は、バックジャンプ直後に発生する矛盾に注目し、それらの学習節に含まれる共通のリテラル群に対して別名を与えるものとする。

しかし、この条件はしばしば満たされるため、頻繁に拡張が行われてしまう。過度な拡張規則の適用は、学習節の保持と同様にソルバーの性能低下を招くためより強い制限が必須である。我々は、バックジャンプ直後の矛盾の連続発生回数や共通するリテラル数に関して制限を設けて評価を行ったが、改善結果を得ることができなかった。これは、問題によって矛盾の原因となったりテラル数や、その出現頻度等の傾向が異なるため、特定の閾値によって適用条件を厳しくしたとしても、問題によっては過度な拡張規則の適用が行われることがあり、このような制限では全体としての性能向上を達成することが難しいためと考えられる。そこで、別名を与える対象のリテラル群には、検出後に別の評価尺度により評価し、拡張を制限するものとする。

我々は、学習節の評価尺度 LBD に着目し、LBD に基づき今後の探索に役立つようなリテラル群を良いとする尺度を用いる。大きな LBD 値をもつ学習節は、学習節の削除時に有用でないと判断され捨てられてしまうが、そのような学習節のなかには、大きなブロックを有しているながらもその他の小さなブ

ロックが多数存在するために削除の判定を受けたものが存在する。提案手法では、このような小さなブロック群に対し別名を与えることで、本来削除される学習節の LBD による評価を改善し求解に役立てることを期待する。検出した学習節群の平均 LBD を  $P$ 、別名を与えようとしているリテラル群の LBD を  $T$  とし、以下の条件を満たす場合、拡張規則を適用する。

$$P/T > 0.73 \quad (2)$$

式 (2) により、別名を与える対象のリテラル群の LBD が、検出した学習節群の LBD の大部分を有している関係性を確認できる。係数 0.73 は予備実験により決定した。新規変数によって置換することで、学習節の LBD は改善されることになる。ただし別名を与えるリテラル群のサイズが 2 の場合には、制限を緩め、優先的に拡張を行うようにした。これは GlucosER が注視したのと同様に、今後複数の節をまとめ上げる効果が比較的に得やすいと考えたためである。

また拡張規則により導入された新規変数が効果を発揮するのは、学習節にその新規変数が含まれた場合である。そのためには、最初のきっかけとして変数選択ヒューリスティクスにおいて新規変数が選ばれる必要がある。そこで我々は、新規変数が、別名を与えられたリテラル群よりも選ばれやすくなるようなバイアスをかけることにした。

上述のように拡張規則の適用条件を絞り込み、変数選択ヒューリスティクスにバイアスをかけたとしても、拡張規則により導入された節集合がほとんど使われない場合がある。そのような節は求解コストの増加をまねくため、学習節集合の定期的削減と同様に、拡張により導入された節集合の削減を行う。具体的には、任意の新規変数が変数選択ヒューリスティクスにより選ばれた回数に注目し、一定回数の変数選択が行われた際に、すべての新規変数に対して、学習に使用されたかどうかを確認する。ここで学習に一度も使われていない新規変数を不要な拡張であったと判断し、これに関わるすべての節を削除する。

一方で、提案手法では新規変数が学習に使用された場合には、その新規変数を有用と判断し今後保持しつづける。ここで、その新規変数を表すリテラル群のを含んでいた検出元の学習節群に対して、リテラル群を新規変数に置き換える。これにより学習節群の LBD の値が変化し、より有益な学習節に変わる効果が期待できる。また節を短く表現できるため、メモリの節約にも貢献する。

## 6. 評価実験

上述の提案手法を SAT ソルバー GlueMiniSat [10] に実装し評価実験を行った。GlueMiniSat のバージョンは 2.2.5a である。GlueMiniSat は Glucose と同様に学習節評価尺度 LBD に基づくソルバーであり、良い学習節を獲得するため非常に積極的にリスタートする特徴を持つ。特に充足不能な問題の証明に強く、SAT 競技会でも優秀な成績を納めている。本研究では、GlueMiniSat に上述の拡張融合法を導入した GlueMiniSatER を試作し、評価を行った。評価用問題集と実験環境は先と同様である。Glucose, GlucosER, GlueMiniSat, GlueMiniSatER における求解数と求解時間の分布をそれぞれ表 2 と図 2 に示す。

まず表 2 から、GlueMiniSatER が最も良い性能を示していることがわかる。また図 2 より、GlucosER では Glucose と比較して、全体的に求解時間が増大傾向にあるものの、GlueMiniSatER ではほぼ同等か若干改善していることが分かる。これは、提案手法における拡張規則とその制限が有効に働き、過度な拡張を抑えることができたためと考えられる。

表 2: 各ソルバーにおける求解数比較

ソルバー	求解数 (全 123 問)
Glucose	62 問
GlucosER	61 問
GlueMiniSat	89 問
GlueMiniSatER	92 問

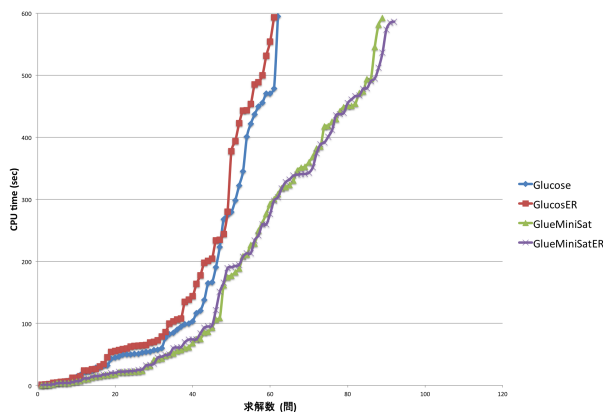


図 2: 各ソルバーにおける求解時間の分布

ここで、ソルバーが求解までに至るまでに発生した総矛盾回数と総単位伝搬数を表 3 に示す。これは Glucose と GlucosER, GlueMiniSat, GlueMiniSatER が共通して解けた問題群における合計値を示している。提案手法は拡張融合法を導入したときに、導入以前より矛盾回数と単位伝搬数がともに値が小さくなっており、拡張融合法をうまく適用できた効果が現れているといえる。

## 7. まとめと今後の課題

本研究では、バックジャンプ直後に発生する矛盾に着目し、複数の矛盾に共通する原因に対して拡張規則を適用することで、拡張融合法が多様な問題においても性能の向上を果たせることを示した。その一方で、我々の手法は、Audemard らの手法が得意とする問題群に対して、未だ有効でないことがわかっている。特定の問題に特化することと、様々な問題において安定した性能を示すことの間にはトレードオフが存在するが、できる限りそれらを両立させるような条件を探ることが今後の重要な課題である。

表 3: 共通解答部分における比較

ソルバー	総矛盾回数	総単位伝搬数
Glucose	12,461,084	$13,215 \times 10^6$
GlucosER	13,042,644	$13,294 \times 10^6$
GlueMiniSat	25,420,103	$33,572 \times 10^6$
GlueMiniSatER	24,760,981	$32,605 \times 10^6$

## 謝辞

本研究の一部は日本学術振興会科学研究費基盤研究 (B) No. 24300007 ならびに人工知能研究振興財団の研究助成によるものである。

## 参考文献

- [1] Gilles Audemard, George Katsirelos, and Laurent Simon. A restriction of extended resolution for clause learning sat solvers. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.
- [2] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern sat solvers. In Craig Boutilier, editor, *IJCAI*, pp. 399–404, 2009.
- [3] Roberto J. Bayardo and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In Benjamin Kuipers and Bonnie L. Webber, editors, *AAAI/IAAI*, pp. 203–208. AAAI Press / The MIT Press, 1997.
- [4] Stephen A. Cook. A short proof of the pigeon hole principle using extended resolution. *SIGACT News*, Vol. 8, No. 4, pp. 28–32, October 1976.
- [5] Carla P. Gomes, Bart Selman, and Nuno Crato. Heavy-tailed distributions in combinatorial search. In Gert Smolka, editor, *CP*, Vol. 1330 of *Lecture Notes in Computer Science*, pp. 121–135. Springer, 1997.
- [6] Jinbo Huang. Extended clause learning. *Artif. Intell.*, Vol. 174, No. 15, pp. 1277–1284, 2010.
- [7] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning sat solvers with restarts. In Ian P. Gent, editor, *CP*, Vol. 5732 of *Lecture Notes in Computer Science*, pp. 654–668. Springer, 2009.
- [8] João P. Marques Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Computers*, Vol. 48, No. 5, pp. 506–521, 1999.
- [9] G. S. Tseitin. On the complexity of derivation in the propositional calculus. *Zapiski nauchnykh seminarov LOMI*, Vol. 8, pp. 234–259, 1968. English translation of this volume: Consultants Bureau, N.Y., 1970, pp. 115–125.
- [10] 鍋島英知, 岩沼宏治, 井上克巳. Glueminisat 2.2.5: 単位伝搬を促す学習節の積極的獲得戦略に基づく高速 sat ソルバー. *コンピュータ ソフトウェア*, Vol. 29, No. 4, pp. 146–160, 2012.