

Literal Block Distanceに基づく学習節共有を行う 並列 SAT ソルバ Glucans

Glucans: A parallel SAT solver with LBD-based sharing of learnt clauses

徐 曉雋*¹
Xiaojuan XU

松本 翔太*¹
Shota MATSUMOTO

上田 和紀*²
Kazunori UEDA

*¹早稲田大学大学院基幹理工学研究科

Graduate School of Fundamental Science and Engineering, Waseda University

*²早稲田大学理工学術院

Faculty of Science and Engineering, Waseda University

We developed a parallel SAT solver called Glucans which runs multiple threads based on some fast SAT solvers in SAT Competition 2011, namely Glucose, GLUEMINISAT, Contrasat, and CIRMiniSat. In Glucans, each thread shares learnt-clauses selected based on LBD in the conflict analysis and propagation phases of the CDCL algorithm. To evaluate its performance, we submitted Glucans to the Parallel Track of SAT Challenge 2012. Glucans solved 521 of a total of 600 problems and ranked 4th at SAT Challenge 2012. Because Glucans was the fastest at 179 problems, which was more than twice of PeneLoPe that was the second fastest solver, we made another experiment to evaluate Glucans and PeneLoPe, by letting them solve more difficult problems including problems from other competitions and by extending the time limit. This experimental result shows that Glucans is faster than PeneLoPe, and sharing learnt clauses in propagation phases contributes much to the performance.

1. はじめに

SAT 問題を解く高速な並列 SAT ソルバとして学習節の共有を行うさまざまなものが研究されてきた。多くの SAT ソルバで学習節の管理は共通の課題だが、特に並列 SAT ソルバではその量が増え、効果的な学習節を保持しそれ以外を削除することが難しくなる。本研究では、長期的な学習節の有用性を表す尺度である LBD (Literal Block Distance^[1]) に基づいて学習節を共有し、より積極的な学習節の削除を行う並列 SAT ソルバ Glucans を作成し、その評価を行った。

2. SAT

充足可能性問題 (SATisfiability: SAT) とは、与えられた命題論理式全体が真となるような変数の割当が存在するか (SATISFIABLE), もしくはどのように割当ててもそのような割当は存在しない (UNSATISFIABLE) ことを示す問題である。一般に NP 完全問題であると知られており、命題論理式が巨大になると現実的な時間内では解を探索できなくなる。しかしながら、人工知能やソフトウェア検証に関する実問題には、ランダムに生成される問題とは異なり一般的に問題に構造が存在し、それらをうまく利用することで高速な求解ができるため、これらの実問題を解くエンジンとして SAT ソルバの研究が進んでいる。

2.1 SAT ソルバ

SAT ソルバには、系統的に解を探索することで、充足可能であれば変数の割当の組み合わせを、充足不能であればその判定ができる系統的 SAT ソルバと、充足可能な解のみを確率的に探索する確率的 SAT ソルバが存在する。本研究では前者の系統的 SAT ソルバを扱う。

連絡先: 徐 曉雋, 早稲田大学大学院基幹理工学研究科情報理工学専攻, 〒169-8555 東京都新宿区大久保 3-4-1 63 号館 5 階 02 号, 03-5286-3340, xxj(at)ueda.info.waseda.ac.jp, 現在株式会社ジーニーに所属

現在多くの SAT ソルバは、Davis, Putnam, Logemann, Loveland によって提案された DPLL アルゴリズムを基にし、衝突時の原因を解析して学習節を生成する機能がある Conflict Driven Clause Learning (CDCL) アルゴリズムで実装されている。

2.1.1 CDCL アルゴリズム

CDCL アルゴリズムでは、変数割当と呼ばれるフェーズでヒューリスティックに従って、適当な変数に真偽値の割当を行う。このとき問題中の節によって他の変数の割当が決まることがあるので、推論フェーズで連鎖的に割当を行っていく (伝播)。誤った変数の割当を行った場合、伝播中に矛盾が生じるので、その原因の解析を行って、適度な決定レベルまで割当を取り消すというバックトラック操作を行う。このとき同時に学習 (clause learning) が行われ、バックトラックした決定レベルから同じ衝突を繰り返し発生させないための学習節を、問題の節集合に追加していく。最終的に推論の途中で充足可能 (SATISFIABLE) と判定されるか、最初の割当までバックトラックを行ってもまだ衝突が発生する充足不可能 (UNSATISFIABLE) な状態になるまでこれを繰り返す。

2.1.2 活性度

多くの CDCL の SAT ソルバでは衝突時に生成される学習節に含まれる変数群の使用頻度を表す活性度 (変数の活性度) と呼ばれる指標を増加させ、変数割当時の選択基準とすることで、衝突に関連する変数から割当を行うようにしている。また、新規の学習節生成にあたって矛盾と関係する既存の学習節があった場合には、その使用頻度を表す活性度 (学習節の活性度) を増加させる。この活性度について学習節をソートした場合に、活性度の低い学習節はあまり有用でないと考えられることから、学習節の増大による問題の肥大化を防ぐための冗長な学習節の削除基準としている。

2.1.3 LBD

LBD は、現在の探索域によらず将来にわたって有効な学習節の評価尺度であり、定義 2.1.1 のように定義される。特に LBD

の値が 2 であるものは長さが 2 の節と同程度に伝播が発生しやすいと考えることができ、推論フェーズで有用な学習節として削除しないようにする。

定義 2.1.1 (LBD (Literal Block Distance)) [1] ある節 C が与えられたときに、節 C 中に含まれるすべてのリテラルを、その現在の割当の決定レベルが同じものを 1 ブロックとして、 n 個のブロックに分割する。このとき、節 C の LBD を n とする。

2.1.4 strict LBD

Strict LBD は LBD を拡張した評価尺度であり LBD よりも細かく学習節を区別できる。次のように定義される。

定義 2.1.2 (strict LBD) [5] C が LBD が n の節であるとする。もし節 C がブロック中にリテラルがひとつしか存在しないユニットリテラルブロックを 1 つでも含むなら、節 C の strict LBD を n とする。1 つも含まない場合は strict LBD は未定義とする。

3. 並列 SAT ソルバ Glucans

並列 SAT ソルバ Glucans は逐次 SAT ソルバをベースに Pthreads でマルチスレッド化を行い、各スレッドに学習節の共有機能を実装することによって作成した。

3.1 ベースソルバの選定

本研究で作成した並列 SAT ソルバ Glucans は、作成時点で最後に開催された SAT ソルバの競技会である SAT Competition 2011 へ投稿された SAT ソルバを参考にした。SAT Competition 2011 において充足可能な問題と充足不可能な問題の分野でのそれぞれ優勝 (Gold) と準優勝 (Silver) した SAT ソルバである表 1 の左側 2 列の SAT ソルバ、Glucose, GLUEMINISAT, Contrasat, CIRMiniSat を基にしている。これらは、代表的な SAT ソルバである MiniSat [7] を拡張したものである。Glucose や GLUEMINISAT [5] は LBD に基づいて学習節を管理する逐次 SAT ソルバであり、本研究の LBD に基づく学習節共有による並列化と親和性が高い。特に、優勝した SAT ソルバである Glucose [3] をベースに、GLUEMINISAT と同じ MiniSat を拡張した SAT ソルバである Contrasat [4], CIRMiniSat の機能も組み込んだ。後者の 2 つの SAT ソルバは充足可能な問題における高速化が期待できる。

表 1: SAT Competition 2011 の結果

	Gold	Silver	Bronze
ALL	Glucose2.0	GLUEMINISAT	Lingeling
SAT	Contrasat	CIRMiniSat	mphasesat64
UNSAT	GLUEMINISAT	Glucose2.0	QuteRSat

3.2 学習節の共有

Glucans では各スレッドで衝突の発生時に学習した学習節を自スレッド以外のスレッドと共有する。LBD が 5 以下の節についてのみ共有し、非同期的に共有が出来るようにリンクリストで実装されたキューへ、学習節とその LBD 値を入れる。キューはスレッドで 1 つずつ持つ。共有のタイミングは学習節が生成される度であり、学習節の重複のチェックは行わない。

共有時に学習節をほかのスレッドへ書き出すには各スレッドが非同期的に以下の手順に従って行う。

1. 衝突が発生し、衝突解析で得られた学習節の LBD が 5 以下ならば共有するためにコピーを作成する。
2. コピーとその使用回数を表すカウンタを束ねる構造体 C を作成する。
3. 新しいリストの要素 E を作成し、E に C へのポインタを持たせる。
4. 共有していないスレッドのリストの Tail とその要素 T をロックする。リストが空なら Head も同時にロックする。
5. Tail の次の要素を末端を表す NULL から E に書き換える。
6. E の次の要素を NULL とし、Tail を E に書き換えて Tail と元の要素 T をアンロックする。
7. 3 から 6 を残りのスレッドの数だけ繰り返す。

また反対に他のスレッドからの学習節を取り込むために以下の手順を書き出し後に行う。

1. 自分のリストの Head をロックする。空を表す NULL なら終了する。
2. Head から要素 E を取り出す。
3. E をロックして E に含まれるポインタから学習節の構造体 C を参照する。
4. C から学習節をコピーし、自スレッドへ追加する。
5. C の使用回数を表すカウンタをインクリメントし、スレッドの数 -1 と等しくなれば C を削除する。
6. E の次の要素 N を読み出して E を削除する。
7. N が NULL でなければ次の要素を E として 3 へ戻る。
8. N が NULL なら Tail をロックして Tail を NULL へ書き換えて Tail をアンロックする。

3.2.1 推論フェーズ中での共有

異なる探索域において、学習節の LBD は推論中に変化する。Glucans では推論中の LBD 更新によって LBD の小さくなった学習節を共有することで、より多くの効果的な学習節の獲得を可能にしたバージョンを作成した。特に有用と考えられる LBD が 2 の節を共有するようにしたが、単に各スレッドが推論中に共有するだけでは、自スレッドでの生成分と他スレッドからの受信分が大量に重複する恐れがある。これは学習節の共有タイミングが受信側では衝突時のみであり、推論フェーズで先に LBD が 2 の全く同じ学習節が発見される可能性があるためである。そこで Glucans では、重複率を減らすため推論中の学習節共有について次の 2 つの制約を設けた。2 つ目の制約からこのバージョンを便宜的に Strict 版と呼ぶ。

- 学習節に共有されたかどうかのフラグをつけ、衝突解析によって生成された時に共有済みのものを再度共有しない。
- LBD を基準とする代わりに、strict LBD が 2 の学習節に限定し、推論フェーズでの共有量自体を抑制する。

3.3 学習節の削除

学習節は本来冗長な節であるために、推論速度を維持するために削除することを迫られる。Glucans では他の SAT ソルバと同じように学習節を削除するために、まず直近の探索に必要な学習節は活性度で、比較的遠い未来に必要となると考えられる節は LBD で評価し、その結果に基づいて学習節を削除することを考えた。ここで、LBD と活性度の両方で評価しようとする、削除を行おうとするたびに全学習節に対して 2 回ソートする必要があるため、1 回のソートで行えるように LBD が 2 のものを残し、LBD > 2 の学習節については活性度の高いものを保持する戦略をとる Activity 版を作成した。また、Glucose と同じ LBD のみに基づいた学習節の削除を行う Basic 版を作成した。

3.4 関連研究

代表的な並列 SAT ソルバとして Plingeling[?] が挙げられる。Plingeling は 2011 年の競技会である SAT Competition 2011 が行われた時点で最も高速な SAT ソルバである。Pthreads で実装されたマルチスレッドソルバで、スレッド間では長さ 1 の学習節である自明な変数の割当を共有し、Lingeling で実装されている同値な変数の共有を Boss-Worker モデルを採用して素集合データ構造を用いて行う。

PeneLoPe[2] は本研究の Glucans とほぼ同時期に開発され、同様に LBD に基づいた学習節の共有を行う並列 SAT ソルバであり、2012 年の競技会である SAT Challenge 2012 の結果が 2 位であるなど非常に高速な SAT ソルバである。PeneLoPe では学習節を他のスレッドから受け取る際に、現在の探索状態における割当から、その節の単位伝播の発生頻度を充足しやすさで推測する PSM[2] と呼ばれる評価基準を用いて不要と思われる学習節を探索対象として追加せずに必要になるまで退避する。本研究の新規性は特に、学習節の保有戦略が異なる点やその共有量にある。SAT Challenge 2012 に出場した Glucans では LBD > 2 のものはその使用頻度の少ないものから削除する Activity 版を作成した。また、比較のためすべて LBD 順に評価する Basic 版を作成した (3.3 節)。さらに、異なる共有タイミングを有する推論中の LBD 更新で strict LBD が 2 となった学習節も共有する Strict 版も作成した (3.2.1 節)。

4. 評価実験

初めに評価実験として、最初に作成した Glucans の Activity 版で、SAT 問題に対する解の探索時間を競う競技会である SAT Challenge 2012 (600 題) へ参加した。競技会の結果と、それを踏まえて Basic 版, Strict 版を追加して、2010 年の競技会から 2012 年の SAT Challenge 2012 までの問題すべて (約 800 題) を用いてローカル環境で行った追加実験の結果も述べる。

4.1 実験環境

ローカル環境での評価は、OS が CentOS 6.3 (kernel 2.6.32), CPU が 2 × Eight-Core Intel Xeon E5-2660 2.20GHz, メモリ容量が 32GB のサーバー上で GCC 4.4.6 でコンパイルしたバイナリを使用して行った。

4.2 SAT Challenge 2012 の結果

SAT Challenge 2012 の制限時間 900 秒以内で解けた問題数を競うルールで、Parallel Track: Application SAT+UNSAT に投稿した Glucans の最終的な結果は、全 600 題中 521 題解き 4 位であったが、1 位の pfolioUZK との差は 10 問であった。

表 2 に SAT Challenge 2012 の結果を解けた問題数の総数と SAT, UNSAT 別を集計したものを示す。

表 2: SAT Challenge 2012 の結果

ソルバ名	SAT	UNSAT	SAT+UNSAT	順位
pfolioUZK	254	277	531	1
PeneLoPe	240	290	530	2
ppfolio2012	242	283	525	3
Glucans Activity	234	287	521	4

また、図 1 に SAT Challenge 2012 の結果を上位 4 つのソルバについて示す。このグラフは縦軸が探索時間で横軸は解けた問題を探索時間について昇順にソートした問題数となっている。系列が右によるほど解ける問題数が多いことを示し、下によるほど高速であると示す。なお、横軸は ppfolio2012 が約 100 秒かかる問題以上を拡大したものとした。

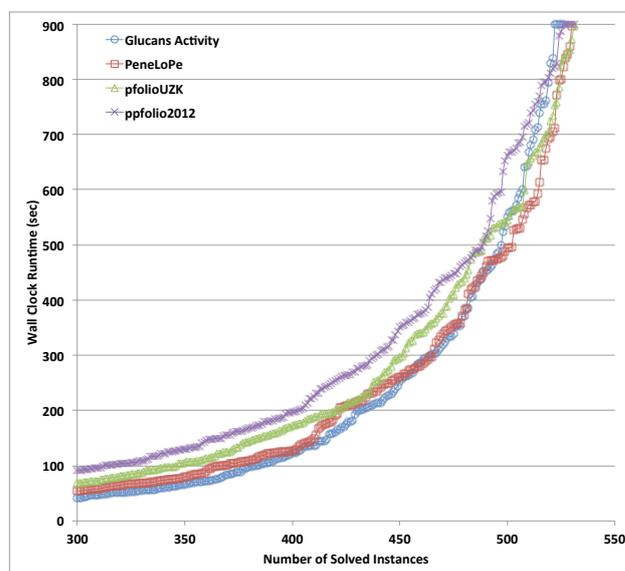


図 1: SAT Challenge 2012 の Cactus プロット

SAT Challenge 2012 ではグラフから分かるように突出して速いソルバがあるという結果ではない。しかしながら途中まで Glucans が最も下によっているように、全問題の約 30% (179 題) において Glucans が最も高速である。これは 2 番目に多い PeneLoPe の 78 題 (約 13%) の 2 倍以上であり、特徴的な結果となっている。

4.3 追加評価実験の結果

2010 年から 2012 年までの競技会の問題すべて (800 題) を使用し、SAT Challenge 2012 に参加した Activity 版に加えて、Basic 版, Strict 版についても評価したローカル環境で行った実験の結果を図 2 に示す。横軸が解けた問題数を、縦軸が探索時間を示し、各問題に対する結果を探索時間について昇順に並べている。なお、横軸は Glucose2.0 が 200 秒かかる問題を最小値として拡大し、それぞれの問題について 2 回の実験結果を別々に解けたものとして集計している。短時間で解ける問題数については各ソルバ共に有意な差がないものの、Glucans と PeneLoPe と比較した場合には、特に長時間必要な難しい問題において、Glucans の方が高速であることが分かる。

解けた問題数の比較を表3に載せる。制限時間を SAT Challenge 2012 に比べて延ばし、問題数を増やした場合には Glucans のどのバージョンでも PeneLoPe より多くの問題を解くことが出来ることがわかる。特に最も多くの問題を解くことが出来た Glucans の Strict 版と PeneLoPe を比較すると Glucans の方が 14 題多い。

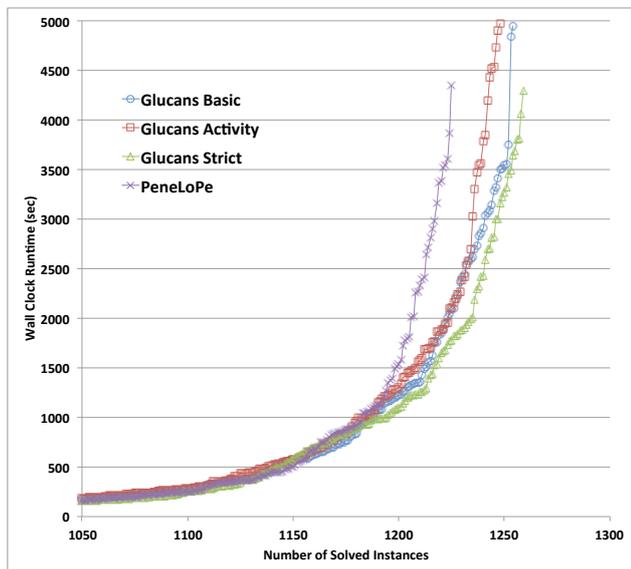


図 2: 各ソルバの実行時間 (Cactus Plot)

表 3: 解けた問題数

ソルバ名	SAT	UNSAT	SAT+UNSAT	順位
Glucans Strict	253	277	630	1
Glucans Basic	249	279	628	2
Glucans Activity	249	276	625	3
PeneLoPe	250	266	616	4

最後にそれぞれの問題について PeneLoPe と Glucans の Strict 版のより詳しい比較を行う。図3に Glucans を横軸に、PeneLoPe を縦軸に同じ問題の実行時間をプロットして示す。

この結果 Glucans Strict の探索時間が PeneLoPe より短い左上のプロットは 467 題で、これは全体の約 63%の問題について Glucans Strict 版の方が高速であることを示している。

この実験結果から、特に、PeneLoPe のように現在の探索状態における割当から、その節の単位伝播の発生頻度を充足しやすさで推測し、充足しやすい節を一時的に退避するよりは、Glucans のより多くの学習節を削除する方法のほうが、実行時間が長くなっていくにつれて多くのメモリを必要とされるときに有効であることがわかる。また、推論中の共有もその解を求める速度に寄与していることがわかる。

5. まとめと今後の課題

本研究の Glucans では、推論フェーズでの学習節の共有する Strict 版が最も高速で、ほぼ同時期に研究されていた PeneLoPe よりも特に大規模な問題において高速であることが分かった。このような評価実験の結果から LBD が 2 の学習節が SAT ソルバにとって非常に重要であり、特に strict LBD が 2 の学習

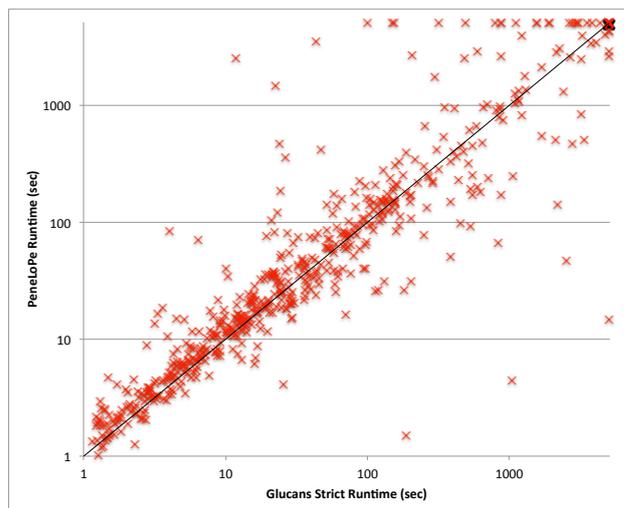


図 3: Glucans Strict と PeneLoPe の比較 (両対数)

節を積極的に共有することが探索の高速化につながるということが分かった。Glucans では学習節の共有を LBD が 5 以下の学習節に制限しているが、この値についての最適値の算出や動的な調整方法について、特に学習節の共有基準を削除されない学習節の LBD 上限から推測することを今後検討する必要がある。

参考文献

- [1] Gilles Audemard, Laurent Simon: Predicting Learnt Clauses Quality in Modern SAT Solver, In Twenty-first International Joint Conference on Artificial Intelligence, pp. 399–404, 2009.
- [2] Gilles Audemard, Benoît Hoessen, Saïd Jabbour, Jean-Marie Lagniez and Cédric Piette: Revisiting Clause Exchange in Parallel SAT Solving, In Fifteenth International Conference on Theory and Applications of Satisfiability Testing, pp. 200–213, 2012.
- [3] Gilles Audemard and Laurent Simon: GLUCOSE: a solver that predicts learnt clauses quality, SAT Competition 2009 Solver Description, <http://www.cril.univ-artois.fr/SAT09/solvers/booklet.pdf>, 2009.
- [4] Allen Van Gelder: Contrasat – A Contrarian Sat Solver, Extended System Description, Journal on Satisfiability, Boolean Modeling and Computation 8, pp. 117–122, 2012.
- [5] 鍋島英知, 岩沼宏治, 井上克巳: GlueMiniSat2.2.5: 単位伝播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.
- [6] Xiaojuan Xu, Yuichi Shimizu, Kazunori Ueda: Glucans System Description, in Proceedings of SAT Challenge 2012 : Solver and Benchmark Descriptions, pp. 23–24, Department of Computer Science Series of Publications B, vol.B-2012-2, University of Helsinki, Helsinki.
- [7] Niklas Een, Niklas Sörensson: An Extensible SAT-solver, Theory and Applications of Satisfiability Testing Lecture Notes in Computer Science, Volume 2919, pp. 502–518, 2004.
- [8] Armin Biere: Lingeling and Friends Entering the SAT Challenge 2012, In Proc of SAT Challenge 2012 : Solver and Benchmark Descriptions, vol. B-2012-2 of Department of Computer Science Series of Publications B, pp. 33–34, University of Helsinki, 2012.