

SAT ソルバーの学習節を考慮した新高速化手法

A New Search Method for SAT Solver Dealing with Learnt Clauses

早田 翔*¹ 安本 猛*¹ 越村 三幸*² 藤田 博*² 長谷川 隆三*²
 Sho Hayata Takeru Yasumoto Miyuki Koshimura Hiroshi Fujita Ryuzo Hasegawa

*¹九州大学大学院システム情報科学府
 Graduate School of Information Science and Electrical Engineering

*²九州大学大学院システム情報科学研究所
 Faculty of Information Science and Electrical Engineering

SAT problem is a fundamental problem in computer science and has many application areas such as planning, scheduling and hardware verification. Therefore it is important to make a faster solver. In this paper, we propose a new search method, which is called PHASE SHIFT. It improve the performance with integrating different search methods. Our experimental results show the effectiveness of this approach.

1. はじめに

充足可能性問題 (Boolean Satisfiability Problem : SAT 問題) とは、「与えられた論理式に対し、式全体を真とする変数割り当てが存在するか否かを判定する」という問題である。回路検証やネットワーク検証、スケジューリングなどの計算機科学における様々な問題が SAT に還元可能であるため、古くから重要視され、研究されている。SAT は NP 完全問題であることが知られており、変数の数を n とした場合、 $O(2^n)$ の計算量が必要であるため、これらをより高速に解くアルゴリズムの開発が期待されている。

SAT 問題を解くプログラムを SAT ソルバーと呼ぶ。毎年 SAT ソルバーの性能を競う競技会が開催されており、SAT 問題を高速に解く高性能な SAT ソルバーが次々に開発されている。

本稿では、これらの高性能な SAT ソルバーに搭載されている様々な高速化手法を統合する手法である **Phase Shift** を提案する。また、既存ソルバーとの比較実験を行い、提案手法の有効性について評価する。

2. Phase Shift

本章では、提案手法である Phase Shift の概要について述べる。Phase Shift は一つの SAT ソルバー内で特徴の異なる様々な探索手法 (Phase) を入れ替え (Shift) ながら探索する手法である。情報を共有しながら探索手法のみを切り替えることで、相乗効果を生み出し高速化を図る。Phase Shift のイメージを図 1 に示す。

Phase Shift と同様に様々な探索手法を用いて推論を行う既存のソルバーに pppfolio[1] がある。ppfolio の探索方法のイメージを図 2 に示す。ppfolio では、様々な SAT ソルバーを呼び出して実行するのに対して、Phase Shift では一つのソルバー内で部分的に探索手法を切り替える。具体的には、学習節の削除方法とリスタート戦略のみが変更される。これにより、探索手法の切り替えに大きなコストが必要なく、学習節や変数アクティビティといった情報の共有にも特別な処置が必要無いという利点がある。

連絡先: 早田 翔, 九州大学大学院 システム情報科学府 情報学専攻, 〒819-0395 福岡県福岡市西区大字元岡 744 番地ウエスト二号館 914 室, 092-802-3599, 2IE13035M@s.kyushu-u.ac.jp

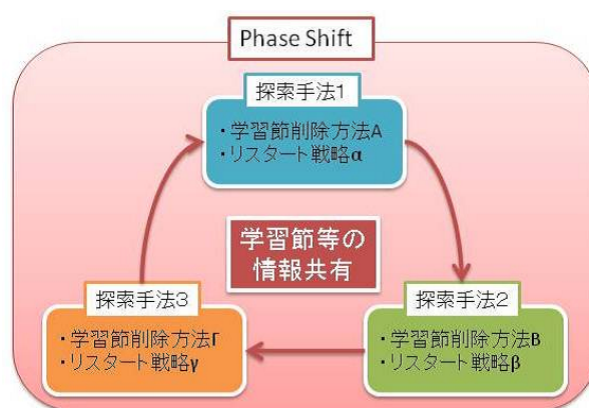


図 1: Phase Shift のイメージ図

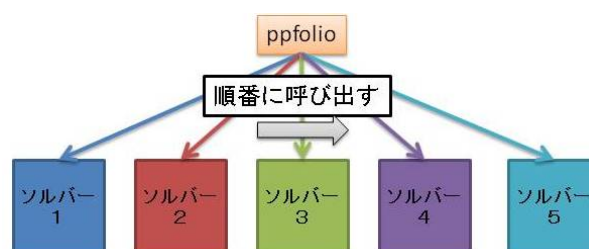


図 2: pppfolio のイメージ図

3. Phase Shift の実装

本研究では MiniSat[2] を基に開発した探索手法と、GlueMiniSat[4] を基に開発した探索手法の 2 つを用いる。今回、2 つだけを用いるのは、あまり多くの手法を切り替えて探索を行うよりも、制限時間内に 2 つの手法を何度も切り替えて探索を行うほうが、相乗効果が高まるのではないかと考えたからである。以降では、MiniSat を基に開発した探索手法による探索を **Sat Phase**、GlueMiniSat を基に開発した探索手法による探索を **Glue Phase** と呼ぶこととする。

探索手法 (Phase) の切り替えにはリスタートの回数を用いる。Phase ごとにリスタート回数の上限をあらかじめ決めてお

き、リスタート回数が上限に達するともう一方の Phase に切り替える。このリスタートの上限回数を **Phase 幅** と呼ぶこととする。例えば、Sat Phase の Phase 幅を 32 と設定すると、Sat Phase に切り替えてから 32 回リスタートを行うと Glue Phase に切り替える。

前章でも述べたように、Phase Shift における探索手法の切り替えとは、リスタート戦略と学習節の削除方法が変更されるということである。つまり、Sat Phase と Glue Phase ではリスタート戦略と学習節削除方法が異なる。ただし、学習節削除の実行間隔は両 Phase とも共通である。

次に、両 Phase で使用される LBD という学習節評価尺度についての説明を行った後、各 Phase のリスタート戦略と学習節削除方法について説明する。また、学習節削除の実行間隔についても述べる。

3.1 LBD

Literals Blocks Distance(LBD) とは、現在様々な SAT ソルバで使用されている学習節評価尺度である [5]。LBD は、節の各リテラルが割り当てられた時の決定レベルを基に節を評価する。同じ決定レベルで割り当てられたリテラル群をリテラルブロックと定義すると、LBD は以下のように定義される。

定義 ある節 C が n 個のリテラルブロックで構成されているとき、節 C の LBD の値は n である。

LBD の例を図 3 に示す。図中の括弧内の数字はリテラルが値を割り当てられた時の決定レベルを表す。同じ色のリテラル群が一つのリテラルブロックであり、この例では 3 つのリテラルブロックがあるため、LBD の値は 3 である。

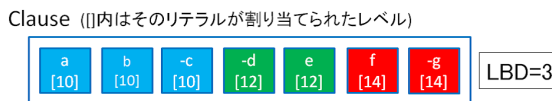


図 3: LBD 値算出例

ブロックに含まれる変数は、今後の探索においてもある決定レベルで再び同じ塊となって現れることが期待できる。LBD の値は小さい節ほど重要度は高いとされている。ブロックの数が少なければ、少ない変数決定で節が伝播する可能性が高いからである。

LBD の値は学習節が生成されたときにリテラルブロック数がカウントされ値が決定する。そして、単位伝播発生時、もしくは矛盾節となった時に再カウントされ更新する。一般的にはカウントした値が、元の LBD 値よりも低ければ更新するという更新法を採用しているが、本研究ではカウントした値に必ず更新し、常に最新の値を保持する更新法を用いる [6]。

また、リテラルブロック数のカウント時は決定レベル 0 のリテラルを無視するという TrueLBD の手法を用いる [6]。

3.2 Sat Phase

Sat Phase は、MiniSat を基に開発しており、リスタート戦略には Luby リスタート [3] を用いている。Luby リスタートでは Luby 数列と呼ばれる以下のような数列を利用する。

$$Luby = \{1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 1, 1, 2, 4, 8, \dots\} \quad (1)$$

n 回目の探索で発生した矛盾の回数が、Luby 数列の n 番目の数値の 100 倍に達するとリスタートを行う。本研究では Sat

Phase のみ Luby リスタートを利用するので、 n の値には Glue Phase における探索回数を含まない。

学習節の削除方法は MiniSat と異なり、学習節を LBD 値の大きさによって並べ替え、値の大きいものから半分を削除する。ただし、LBD 値が 2 以下の学習節については削除しない。

3.3 Glue Phase

Glue Phase は、GlueMinisat を基に開発した手法である。リスタート戦略には、LBD と決定レベルの移動平均を利用した動的リスタートを利用する。矛盾発生の際に、その時生成した学習節の LBD 値とその時点での決定レベルを記録する。リスタートするか否かを判定する時に、全体平均（判定時までの記録全体の平均）と移動平均（最近 50 個の記録の平均）とを比較し、移動平均の方が大きければリスタートを起こす。判定は以下の式に従う。式中の $rate$ は実行時に与える定数である。

$$\text{移動平均} \times rate > \text{全体平均} \quad (2)$$

リスタート判定時に、LBD もしくは決定レベルのどちらかが式 2 を満たせばリスタートを行う。本研究では、 $rate$ の値を LBD では 0.8、決定レベルでは 1.0 に設定している。

学習節の削除方法は、学習節を LBD 値の大きさによって並び替え、値の大きいほうから 4 分の 3 を削除する。ただし、LBD 値が 3 以下の学習節については削除はしない。なお、上記の削除後に元の学習節数の半分以上が削除されず残っている場合は、残った学習節の半分を LBD 値の大きいものから削除する。

3.4 学習節削除の実行間隔

学習節削除を行うタイミングは、Phase に関わらず共通であり、Glucose[7] が採用していた **Aggressive Reduce Database** を利用する。Aggressive Reduce Database では以下の式を満たした場合に学習節削除が行われる。

$$\text{実行開始からの矛盾発生回数} \geq \text{閾値} \quad (3)$$

閾値は初期値として定数 $BASE$ を持ち、学習節削除が行われる度に定数 INC と学習節削除が行われた回数 num を用いて以下のように更新される。

$$\text{閾値} + = BASE + INC \times num \quad (4)$$

本研究では以下のように設定している。

- $BASE = 30000$
- $INC = 10000$

4. 予備実験 1

本章では、Phase 幅の最適値を見つけるために、Phase 幅を様々な値に変えて実験を行った。比較のために一般的な SAT ソルバーである MiniSat2.2 と、競技会で優秀な成績を収めた高速 SAT ソルバー、GlueMinisat2.2.5 と Glucose2.0 の実験結果も載せる。

4.1 実験環境

本実験では全て以下の環境で行っている。

CPU Core Duo 3.33GHz

メモリ 8GB

OS LINUX

また、使用する問題セットは以下のものを用いた。

- SAT 2009 competition Application 部門にて使用された 292 問
- SAT 2005 competition から 176 問 (SAT05)

SAT 2009 competition で使用された 292 問は以下のように分けられている。

- SAT 2009 competition 用に作成された 165 問 (SAT09)
- SAT 2007 competition で使用された 74 問 (SAT07)
- SAT-Race 2008 で使用された 27 問 (SATRACE08)
- SAT-Race 2006 で使用された 26 問 (SATRACE06)

問題 1 つに対する実行時間は 1200 秒で、その時間内でソルバーが答えを返せなかった場合は問題を解けなかったことになる。

4.2 実験結果

実験結果を表 1 に示す。表中の括弧内の数字はその Phase の Phase 幅を表す。

表 1: Phase 幅別実験結果

	SATISFIABLE	UNSATISFIABLE	TOTAL
MiniSat2.2	236	109	345
GlueMiniSat2.2.5	230	123	353
Glucose2.0	233	121	354
Sat(32) Glue(64)	237	122	359
Sat(32) Glue(96)	237	121	358
Sat(32) Glue(128)	233	100	333
Sat(48) Glue(128)	238	122	360
Sat(64) Glue(128)	237	123	360
Sat(64) Glue(192)	238	123	361
Sat(112) Glue(112)	239	124	363

4.3 考察

実験結果を見ると、Sat Phase に対して Glue Phase の Phase 幅を大きくし過ぎると性能が低下しやすい傾向がある。しかしながら、ほとんどの場合、SATISFIABLE の結果は MiniSat と同等の性能を示し、UNSATISFIABLE の結果は GlueMiniSat と同等の性能を示している。全体の解けた問題数も大部分が MiniSat と Glueminisat, Glucose を上回っており、Phase Shift の有効性を示した。

5. 予備実験 2

先ほどの実験では、Glue Phase のリスタート戦略に LBD と決定レベルの移動平均を利用した動的なリスタート戦略を利用した。この動的なリスタート戦略は、Luby リスタート戦略と比較するとリスタートしやすい戦略ではあるが、時に長い探索を行ってしまうこともあり、Sat Phase の Luby リスタート戦略と似たような動きになる場合も考えられる。そこで、Glue Phase のリスタート戦略を安定して頻繁にリスタートさせる静的なリスタート戦略に変更し、Glue Phase と Sat Phase の違

いを更に際立たせることで、相乗効果を向上させることができると考えた。静的なリスタートに変更した Glue Phase を元の Glue Phase と区別するために、**Unsat Phase** と以降は呼ぶこととする。

本章では Glue Phase から Unsat Phase に変更し実験を行った。本実験でも、Phase 幅の最適値を見つけるために、Phase 幅を様々な値に変更して実験を行った。実験結果から、Sat Phase の Phase 幅を 32、Unsat Phase の Phase 幅を 192 に設定した場合に最も多くの問題を解くことができた。このときの結果を以下の表に示す。

また、比較のために、前章の実験において最も良かった結果と、MiniSat2.2, Glueminisat2.2.5, Glucose2.0 の実験結果を示す。さらに、Sat Phase 単体で実行した場合の結果と、Unsat Phase 単体で実行した場合の結果を示し、Phase Shift によってどのような変化がもたらされているかを見る。

5.1 実験結果

実験結果を表 2 に示す。

表 2: Unsat Phase に変更した場合の実験結果

	SATISFIABLE	UNSATISFIABLE	TOTAL
MiniSat2.2	236	109	345
GlueMiniSat2.2.5	230	123	353
Glucose2.0	233	121	354
Sat(112) Glue(112)	239	124	363
Sat(32) Unsat(192)	246	123	369
Sat Phase 単体	245	115	360
Unsat Phase 単体	232	116	348

5.2 考察

実験結果から、Glue Phase を Unsat Phase に変更することで、UNSATISFIABLE に対する性能を落とすことなく SATISFIABLE に対する性能を上げることができたと言える。

次に、各 Phase 単体の結果と比較する。

Sat Phase 単体では、解が SATISFIABLE となる問題に対して非常に性能が高いが、その一方で UNSATISFIABLE となる問題に対する結果は他の SAT ソルバーと比較すると、あまり良くはない。しかし、Phase Shift に組み込むことで、UNSATISFIABLE となる問題を解いた数が増加し、弱点が補強されている。加えて、SATISFIABLE においても解けた問題数が上昇しており、Phase Shift の相乗効果の影響によるものと思われる。

Unsat Phase 単体では、他の SAT ソルバーの結果と比較すると、あまり良い結果ではなかった。しかし、Phase Shift に組み込むことで格段に性能が向上した。このことから、単体では高い性能をもっていない手法でも Phase Shift を利用することで良くなる可能性があることがわかった。

6. 実験

予備実験では SAT Competition 2005 と 2009 の問題を用いて Phase 幅等の調整を行ってきた。本章では異なる問題セットを用いて実験評価することで、提案手法が一般的な問題に対しても効果があるかを検証した。

6.1 実験結果

本実験では、問題セットとして SAT Competition 2011 Application 部門 300 問を用いた。実験環境、制限時間は予備実

験と同じである。実験結果を表 6.1 に示す。また、その結果を図 4 に示す。グラフは、あるソルバーにおいて縦軸に示す時間までに解けた問題の問題数を示すものである。よってグラフが右にあるソルバーほど、性能が良いことを表している。

表 3: SAT2011 での実験結果

	SATISFIABLE	UNSATISFIABLE	TOTAL
MiniSat2.2	83	72	155
Glucose2.0	80	91	171
GlueMiniSat2.2.5	72	93	165
提案手法	91	88	179

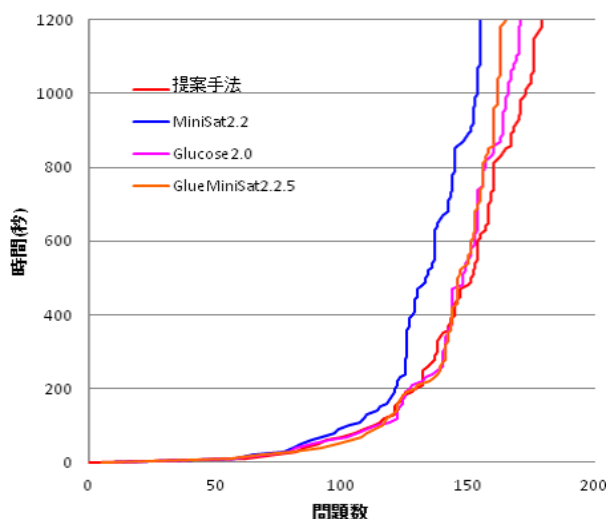


図 4: SAT2011 での実験結果グラフ

6.2 考察

実験結果から、SATISFIABLE の解けた問題数は他の高速ソルバーを上回っており、UNSATISFIABLE の結果も同等の性能を示すことができた。解けた問題数の総合では、他のソルバーを大きく上回り、SAT Competition 2011 の問題でも提案手法の有効性を示すことができた。

グラフを見ると、600 秒付近から提案手法が他のソルバーとの差を広げており、1200 秒以上では更に差を広げる可能性がある。

7. まとめ

本研究では、一つの逐次ソルバーにおいて複数の手法を切り替えながら探索する、Phase Shift を提案した。情報を共有しながら探索手法のみを切り替えることで、相乗効果を生み出し、高速化させることに成功した。実験により、既存の高速ソルバーを上回る結果を示し、Phase Shift の有効性を示すことができた。

本研究では、数種類の Phase 幅の場合でしか実験できていないため、今後の課題としては、更なる実験にて最適値を探し出すということがあげられる。

また本研究では、Glue Phase の改良し Unsat Phase にしたことで、より性能を向上させることができた。このことから、Sat Phase に関しても手を加えることで性能を向上させる可能性がある。今後は Phase Shift に適した Sat Phase の改良を

考えていきたい。

謝辞 本研究は科研費 (25330085) の助成を受けたものである。

参考文献

- [1] Oliver Roussel. : Description of ppfolio, SAT 2011 competition System Description. <http://www.cril.univ-artois.fr/~roussel/ppfolio/solver1.pdf>
- [2] Eén, Niklas and Sörensson, Niklas : An Extensible SAT-solver, SAT 2003, pp. 502-518 (2003)
- [3] Luby, M., Sinclair, A. and Zuckerman, D. : Optimal speedup of Las Vegas Algorithms, Inform. Process Lett., Vol.47, No.4, pp.173-180 (1993)
- [4] 鍋島英知, 岩沼宏治, 井上克巳: GlueMiniSat 2.2.5 : 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, 日本ソフトウェア科学会第 28 回大会, 6E-1 (2011)
- [5] Gilles Audemard and Laurent Simon.: Predicting Learnt Clauses Quality in Modern SAT Solvers, IJCAI 2009, pp. 399-404 (2009)
- [6] 奥川巧, 安本猛, 越村三幸, 藤田博, 長谷川隆三 : SAT ソルバの学習節に対する新しい評価手法の提案, 人工知能学会第 26 回全国大会, 1E3-OS-4-3 (2012)
- [7] Audemard, G. Simon, L.: GLUCOSE: a solver that predicts learnt clauses quality, SAT 2009 competitive events booklet, preliminary version, pp.7-8 (2009)