3C-01

CDCL SATソルバーにおけるバックトラック条件の緩和

楢崎　修二†

長崎大学工学研究科†

はじめに

命題論理式の充足可能性判定問題を解く SAT ソルバーは種々の制約問題に応用できることから重要な技術として現在広く使われている[1,2]。

SAT ソルバーは充足不可能な問題に対して停止しない確率的ソルバーと停止する系統的ソルバーに分類されるが系統的ソルバーの基本となる探索アルゴリズムは CDCL アルゴリズムであり、高速化に関する多くの研究がなされている。

本稿ではこの CDCL アルゴリズムの重要な 1 ステップである矛盾の検出とそれによるバックトラックの実行に関する変更の可能性について考察する。

1章　CDCL アルゴリズム

CDCL アルゴリズムは以下のように説明できる。

1. 未割当リテラルを選ぶ。なければ式の充足性を判定して終了する
2. 選んだリテラルにより矛盾が発生していないかどうかを調べる。これは節毎に矛盾が発生しているかどうかを判定する手続きに帰着する（これを論理制約伝播という）。
3. 矛盾が発生したならば、論理制約伝播をやめ、矛盾の原因となる割当を調べて学習節を構成し、それに基づきバックトラックを行い、1 に戻る
4. 無矛盾であれば、1 に戻る

このアルゴリズムの特徴は、制約を満足しない解候補を選んだ場合に、原因を調べて再び同じ過ちを起こさないこと（すなわち問題からの知識抽出）、得られた知識（禁止されるべき割当）が論理式（正確には節）として表されるため、解くべき問題と得られる知識がどちらも論理式として統一的に表現・処理できることにある。

近年の CDCL アルゴリズムの高速化は、アルゴリズム自体の並列化、未割当リテラルの選択に関するヒューリスティクスの導入、適切なデータ構造の利用による論理制約伝播の高速化（監視リテラルリストなど）、得られた学習節の有効性の検証（学習節データベース管理）といった観点から行われている。

2章　提案手法

探索においては与えられた問題から適切な知識を効率よく獲得することが重要である。CDCL アルゴリズムにおいては矛盾からよい学習節を抽出することに対応する。

そのため、学習節の獲得を効率よく行えるならば処理全体の高速化につながるであろう。現在の CDCL アルゴリズムでは再帰的に論理制約伝播を繰り返し、矛盾が 1 回発生すると学習節を 1 つ生成し、バックトラックを行っている。もし矛盾が発生しても論理制約伝播を止めずに続けると、さらなる矛盾を見つけることができ、別の学習節が生成できるはずである。

ここで、あるバックトラック後に次の矛盾を見つけるまでに必要な伝播回数よりも矛盾発生後直ちにバックトラックを行わずに伝播を進め、次に矛盾を検出するまでに必要な伝播回数が小さければ、これは効率のよい知識獲得が可能になると思われる。この提案手法は以下のように表せる。

1. 未割当リテラルを選ぶ。なければ式の充足性を判定して終了する
2. 十分に論理制約伝播を進める
3. 矛盾が発生していたならば、矛盾の原因となる割当を調べて複数の学習節を構成し、それに基づきバックトラックを行い、1 に戻る
4. 無矛盾であれば、1 に戻る

この変更による実装に関する影響について以下にまとめる。

新たな値の導入：矛盾が検出された変数を含む節からの新たな割当を避ける必要がある。これは割当値として現在の **T**, **F**, **未割当**の 3 値に加え、矛盾発生中を意味する第 4 の値を追加することで可能になる。この値は最初の矛盾が発生した時に初めて使われ（割当テーブルに格納され）、論理制約伝播直後のバックトラックの処理中に全てテーブルから削除することができる。この間に必要な処理は矛盾理由の解析であるが、この処理は変数の割当を参照するこ

A Relaxation of Backtrack Condition in CDCL SAT Solvers

†Shuji Narazaki, Nagasaki University

とはない。従って、このような変更を加えても対応が必要な処理は論理制約伝播のみであり、全体に与える影響は少ない。

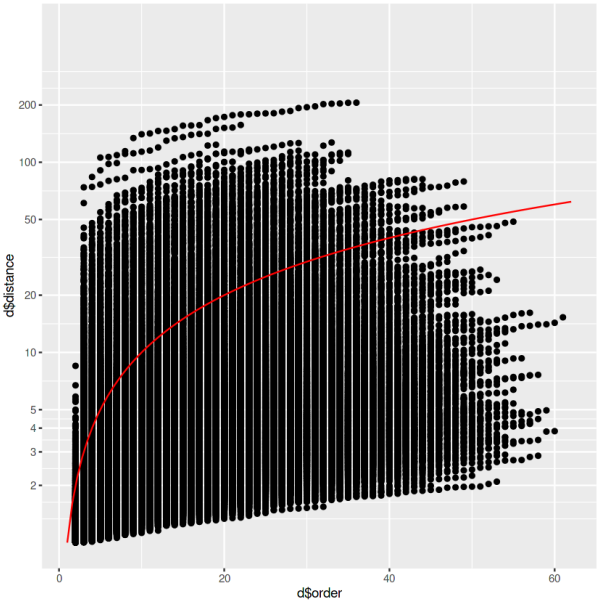
監視リテラルデータ構造：CDCL アルゴリズムでは論理制約伝播を高速に行うために、多くの SAT ソルバーは監視リテラルというデータ構造を用いている[2]。これはバックトラック時になにもしなくてよいという特徴を持つ。従って、複数の矛盾を検出するように変更したとしても、バックトラック時に監視リテラルに関する追加の処理は必要とされない。

これらのことから、提案手法の導入は論理制約伝播の回数が増加する事以外に実装上の大きなペナルティを持つものではないため、高速化が期待できると考えられる。

3章　妥当性の検討

提案手法の妥当性を検証するために具体的な問題において基礎的な検討を行う。

本提案手法の有効性は矛盾の分布に依存するため、ある問題における矛盾変数の分布を調査した。結果を下図に示す。



横軸は矛盾変数の出現数であり、縦軸はその矛盾変数を見つけるまでに掛かった伝播回数である。ただし、最初の矛盾変数を見つけるまでに必要とした伝播回数を 1 として正規化したものである。矛盾変数は多く存在し、早いものでは 2 つ目の矛盾変数を見つけるに要する追加の手間は 0.1 程度である一方、100 倍を超える手間を要する場合もある。2 つ目の矛盾変数での伝播回数の平均値は 2 倍を超えないため、最初の矛盾変数の検出直後にバックトラックを行うことが最良とは言えないことがわかった。また変数の分布は大きな分散を持つため、この手法を用いる際には

- 探索する矛盾変数の個数の上限(P1)
- 追加探索のために許される伝播回数の上限(P2)

などをパラメータとして用いつつ「十分に論理制約伝播を進める」ことが必要になると考えられる。

次に、矛盾変数を複数回見つけたからといってそれら全てが有効というわけではない。以下のような可能性が考えられる。

- 同じ変数である、
- 学習節やバックトラック点が共通となる。

そこで我々が開発中の SAT ソルバー[3]を基に学習節の解析を行った。先に挙げたパラメータ P1, P2 をソルバーに導入して探索範囲を制御し、得られた矛盾からそれぞれ解析を行って得られた学習節の等価性（および包含性）を確認した。下表は P2 を 10（最初に矛盾を見つけるまでに要した伝播回数の 10 倍を上限とする）に固定し、P1 に対する実際に得られる有効な学習節（重複するもの、無駄なものを除いた節）の平均個数である（250 変数の 3SAT 問題 100 問の平均値）。

P1	2	4	8	16
平均節数	1.248	2.104	3.554	5.978

このように検出された矛盾変数には重複も存在するが、真に異なる複数の学習節が生成されうる。逆に言えば、複数の学習節からどこへバックトラックをすればよいか、またそれらからの節や変数の優先度に対する調整をどのようにすればよいかを検討する必要がある。

まとめ

本発表では、矛盾の発生で停止しない CDCL アルゴリズムの改良案を提案し、実装に関する検討や妥当性の根拠となる実験結果を示した。以上のことから、提案手法は有効であると思われる。今後はこの結果に基づき実装を完成させて評価を行う予定である。

参考文献

[1] Armin Biere et. al, Handbook of Satisfiability, IOS Press, 2009.
[2] 鍋島英知，栄剛秀，高速 SAT ソルバーの原理，人工知能学会誌，25 巻 1 号, pp.68--76, 2010-01.
[3] 楢崎修二，Minisat との比較による Haskell SAT ソルバーの高速化，IPSJ2016, 情報処理学会，4A-05,