



SATソルバーの最新動向と利用技術

宋, 剛秀
番原, 睦則
田村, 直之
鍋島, 英知

(Citation)

コンピュータ ソフトウェア, 35(4):72-92

(Issue Date)

2018

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

All Rights Reserved, Copyright (C) Information Processing Society of Japan. Notice for the use of this material: The copyright of this material is retained by the Japan Society for Software Science and Technology(JSSST). This material is published on this web site with the agreement of the JSSST. Please comply with Copyright Law of Japan ...

(URL)

<https://hdl.handle.net/20.500.14094/90006800>



解 説

SAT ソルバーの最新動向と利用技術

宋 剛秀 番原 睦則 田村 直之 鍋島 英知

命題論理式の充足可能性判定 (SAT) 問題を解くプログラムである SAT ソルバーは、2000 年以降その性能面において飛躍的に進化した。それに伴い、解きたい問題を SAT 符号化により SAT 問題へと変換し、SAT ソルバーを用いて解く SAT 型システムが、プランニング、ソフトウェア・ハードウェア検証、スケジューリング問題など様々な分野で成功を収めるようになった。本稿では、まず SAT ソルバーの最新動向として、性能面と機能面における進歩をその要因の 1 つである SAT ソルバーの国際競技会の視点から説明を行う。次に SAT ソルバーの利用技術の視点から、SAT ソルバーの機能面の進歩と符号化技術を組み合わせることで、複雑な問題を解くことが可能になることの説明を行う。そのような例として多目的最適化問題のパレート解を SAT ソルバーを利用して求める方法を説明する。

Since 2000, SAT solvers that are programs solving SAT instances has enormously progressed in performance. Due to the performance improvement, SAT-based systems that encode problems into SAT instances and solve them by SAT solvers have succeeded in various research fields such as planning, software/hardware verification, scheduling problems, etc. In this paper, as recent advances in SAT solvers, we first explain the progress of their performance and functions from the viewpoint of the international competition of SAT solvers which is a factor of their progress. Then, from the viewpoint of utilization technologies, we explain that we can solve more complex problems by combining progressive functions of SAT solvers with encoding methods. As an example, we explain a solving method for the multiobjective optimization problems by using SAT solvers.

1 はじめに

命題論理式の充足可能性を判定する充足可能性判定 (SAT) 問題は、1960 年代から多くの研究が蓄積されてきた。特に 2000 年以降 SAT 問題を解くプログラムである SAT ソルバーが性能面において飛躍的に進歩し、SAT 技術は様々な応用分野における推論の基盤技術として注目を集めている。

SAT に関する最新の話題としては、2015 年 12 月に

Recent Advances in SAT Solvers and their Utilization Technologies.

Takehide Soh, Mutsunori Banbara, Naoyuki Tamura, 神戸大学情報基盤センター, Information Science and Technology Center, Kobe University.

Hidetomo Nabeshima, 山梨大学大学院医学工学総合研究部, Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi.

コンピュータソフトウェア, Vol.35, No.4(2018), pp.72–92.

[解説論文] 2017 年 6 月 19 日受付。

チューリング賞受賞者 Donald E. Knuth による著名な教科書 “The Art of Computer Programming” の 4 巻の最新分冊 [45] が出版されたことが挙げられる。この分冊では 300 ページ以上もの分量が SAT ソルバーとその応用の説明に割かれ、序文の一節には “*The SAT problem is evidently a killer app, because it is key to the solution of so many other problems.*” と SAT の重要性が killer app という言葉を用いて述べられている。2016 年 5 月には、30 年以上未解決だった Boolean Pythagorean Triples 問題が SAT ソルバーを用いて証明 [33] されたことも Nature で取り上げられ話題になった [50]。また 2017 年 2 月には、Google とオランダ国立数学・計算機科学研究センター (CWI) のチームで、同一の SHA-1 (Security Hash Algorithm 1) のハッシュ値をもつ 2 つのファイルが実際に作成されたことが報告された^{†1}。このハッシュ値の作成の過程で必要な near-collision ブロックペア

を見つける計算の一部でも SAT ソルバーが利用されている。

SAT 型システム とは与えられた問題を **SAT** 符号化により SAT 問題へと変換し, SAT ソルバーを用いて解くシステムである。SAT ソルバーの求解性能の飛躍的な進歩を背景に, 以下のような分野において成功をおさめている。

- 自動テストパターン生成 [51]
- プランニング (SATPLAN, Blackbox) [44]^{†2}
- ジョブショッパスケジューリング [18] [49]
- 有界モデル検査 [16]
- ソフトウェア検証 (Alloy) [40]^{†3}
- 書換えシステム (AProVE) [27]^{†4}
- インテル社の i7 プロセッサの検証 [42] (SMT ソルバーによる検証)
- Eclipse のコンポーネント間の依存解析 [52]^{†5}
- 解集合プログラミング (clasp) [25]^{†6}
- Linux のパッケージマネージャである DNF の依存性解決^{†7}
- 制約充足問題 (Sugar) [85] [88]^{†8}
- パッキング配列問題 [68]

この他にも, ペトリネットの検証 [69], システム生物学 [82], グラフ理論の問題 [81], 多目的最適化問題 [80] などにも応用されている。

本稿では, SAT ソルバーの最新動向として, 国際競技会における評価部門の変遷過程の視点から SAT ソルバーの性能面と機能面における進化について説明を行う。次に SAT ソルバーの利用技術の視点から, SAT ソルバーの機能面の進化と符号化技術を組み合わせることで, 複雑な問題を解くことが可能になることの説明を行う。そのような例として多目的最適化問題のパレート解を SAT ソルバーを利用して求める方

法を説明する。

SAT ソルバーや周辺技術については, 人工知能学会・情報処理学会の 2 つの特集号やコンピュータ・ソフトウェアでもいくつかの解説がある (7 節を参照)。特に, 解説論文 [11] [64] では, SAT ソルバーの性能面における進化が説明されており, 解説論文 [79] では, 制約プログラミングシステムへの応用の視点で SAT 技術が説明されている。

しかし, SAT ソルバー, 特にその機能と利用技術については, その後も様々な発展と進化が遂げられている。本稿の特に新しい点は上述した視点から, 以下の点について説明を行っていることである。

- SAT ソルバーの最新動向について
 - － 国際競技会が SAT ソルバーの性能面と機能面の進化におよぼした影響
 - － Certified UNSAT : SAT 問題が与えられ充足不能と判定された時に証明を出力する機能
 - － 極小 UNSAT サブセット (Minimal Unsatisfiable Subset; MUS) : 充足不能である SAT 問題における, 充足不能である部分節集合の中で極小のものを 1 つ出力 (もしくは列挙) する機能
- SAT ソルバーの利用技術について
 - － ハイブリッド符号化 : それぞれ異なる性質を持つ順序符号化と対数符号化を統合した符号化
 - － 多目的最適化の解法 : 順序符号化・ハイブリッド符号化などの符号化技術とインクリメンタル SAT・MUS などの SAT ソルバーの進化した機能を組み合わせることによる解法

以降では, まず 2 節で SAT 問題と SAT ソルバーの定義を説明する。次に SAT ソルバーの最新動向を 3, 4, 5 節で説明する。3 節では, 性能面における進化を取り上げ, 初期から最新の SAT ソルバーに至るまでの高速化技術を説明する。4 節では, SAT ソルバーの継続的な進化を支えている SAT ソルバーの国際競技会の説明を行い, 最新の SAT ソルバーの性能面や機能面の進化に与えた影響について説明する。5 節では, SAT ソルバーの機能面における進化について説明する。SAT ソルバーの利用技術については, 6 節

^{†1} <https://shattered.io>

^{†2} <https://www.cs.rochester.edu/u/kautz/satplan/blackbox/>

^{†3} <http://alloy.mit.edu/alloy/>

^{†4} <http://aprove.informatik.rwth-aachen.de/index.asp?subform=home.html>

^{†5} https://wiki.eclipse.org/Equinox_P2_Resolution

^{†6} <https://potassco.org>

^{†7} <https://fedoraproject.org/wiki/Features/DNF>

^{†8} <http://bach.istc.kobe-u.ac.jp/sugar/>

で様々な問題を解く際に必要となる技術の 1 つである符号化技術について説明する。さらに符号化技術と SAT ソルバーの進化した機能を組み合わせることで、複雑な問題を解くことが可能になることを多目的最適化問題を例にとって説明する。最後に 7 節で、SAT に関する教科書および解説論文をまとめて紹介する。

2 SAT 問題と SAT ソルバー

SAT 問題 (SAT problems) は、命題論理式の充足可能性を判定する問題である。通常、問題の入力は連言標準形 (Conjunctive Normal Form; CNF) の命題論理式で与えられる。この CNF 式は 1 つ以上の節の論理積 (連言) であり、節 (clause) は 1 つ以上のリテラルの論理和 (選言) である。リテラル (literal) は、命題変数かあるいはその否定である。命題変数は真または偽の値をとる。例を以下に示す。

$$(\neg p_1 \vee \neg p_2) \wedge (\neg p_1 \vee \neg p_3) \wedge (\neg p_2 \vee \neg p_3) \wedge (p_1 \vee p_2 \vee p_3)$$

この例では、 p_1 に真、 p_2 と p_3 に偽が割当てられた時、与えられた CNF 式が真になる。SAT 問題が充足可能であるとは、このように与えられた CNF 式を真にする値割当てが存在することを意味する。一方、そのような値割当てが存在しない時に、SAT 問題は充足不能と呼ばれる。

SAT ソルバー は、与えられた SAT 問題が充足可能か充足不能かを判定するプログラムである。通常、充足可能であればその値割当てを解として出力する。充足可能あるいは充足不能の両方を判定可能である系統的な探索を行う SAT ソルバーは、DPLL (提案者である Davis, Putnam, Logemann, Loveland の頭文字からなる) [19] と呼ばれるアルゴリズムを実装したソルバー、および、DPLL に節学習を組合せた CDCL (Conflict Driven Clause Learning) [14] [57] と呼ばれるアルゴリズムを実装したソルバーに大別される。また並列ソルバーも活発に開発されている。次節から、その進化について説明を行う。

3 SAT ソルバーの性能面における進化

本節では SAT ソルバーの性能面における進化について説明する。本節の内容は既存の解説論文 [11] [64]

にも詳しく記載されているので、興味のある読者はこれらの論文も参照されたい。また、本稿とこれらの解説論文との差分は 7 節に記載されているので、既に上記の解説論文を読んでいる読者は、差分を確認してから読まれると幸いである。

まず 2000 年以降 SAT ソルバーがどのように進化したのか、2 つの図を用いて説明する。

図 1 は、SAT ソルバーの進化を表した系統樹である。左側の逐次ソルバーの系統樹をみると、DPLL ソルバーから始まり、CDCL ソルバー Chaff, MiniSat, Glucose を経て多岐にわたる SAT ソルバーが開発されていることが分かる。また右側の並列ソルバーの多くは、逐次ソルバーから派生していることがわかる。

星印 (★) は 2016 年に開催された SAT ソルバーの国際競技会 (以下、SAT 競技会と呼ぶ) の Main トラック (逐次) あるいは Parallel トラック (並列) で入賞したソルバーであることを表している。なお図に含まれない他のソルバーについても、多くは MiniSat, Glucose を基に開発が行われている。

図 2 は、図 1 の SAT ソルバーの中で Satz (2001 年まで開発)、Chaff (2007 年まで開発)、MiniSat (2012 年まで開発)、MapleCOMSPS (2016 年に公開) の求解性能を比較したグラフである。比較は、2016 年に開催された SAT 競技会の Main トラックで使われたベンチマーク 500 問を用いて、著者の計算機上で行った。

グラフはカクタスプロットと呼ばれるもので、縦軸がベンチマーク 1 問あたりの制限時間を表し、横軸はその 1 問あたりの制限時間が与えられた時に 500 問中何問の問題が解けたかを表している。例えば、1 問あたりの制限時間を 100 秒とすると、Satz は約 20 問、Chaff は約 40 問、MiniSat は約 60 問、MapleCOMSPS は約 80 問解けていることになる。どの制限時間で見ても、古い SAT ソルバーより新しい SAT ソルバーが、多くの問題を解いており、SAT ソルバーが性能面において継続して進化していることがわかる。

以降の節では、SAT ソルバーの性能面における進化がどのような技術によって実現されたかについて、Satz 等の歴史的な DPLL ソルバーから MiniSat まで

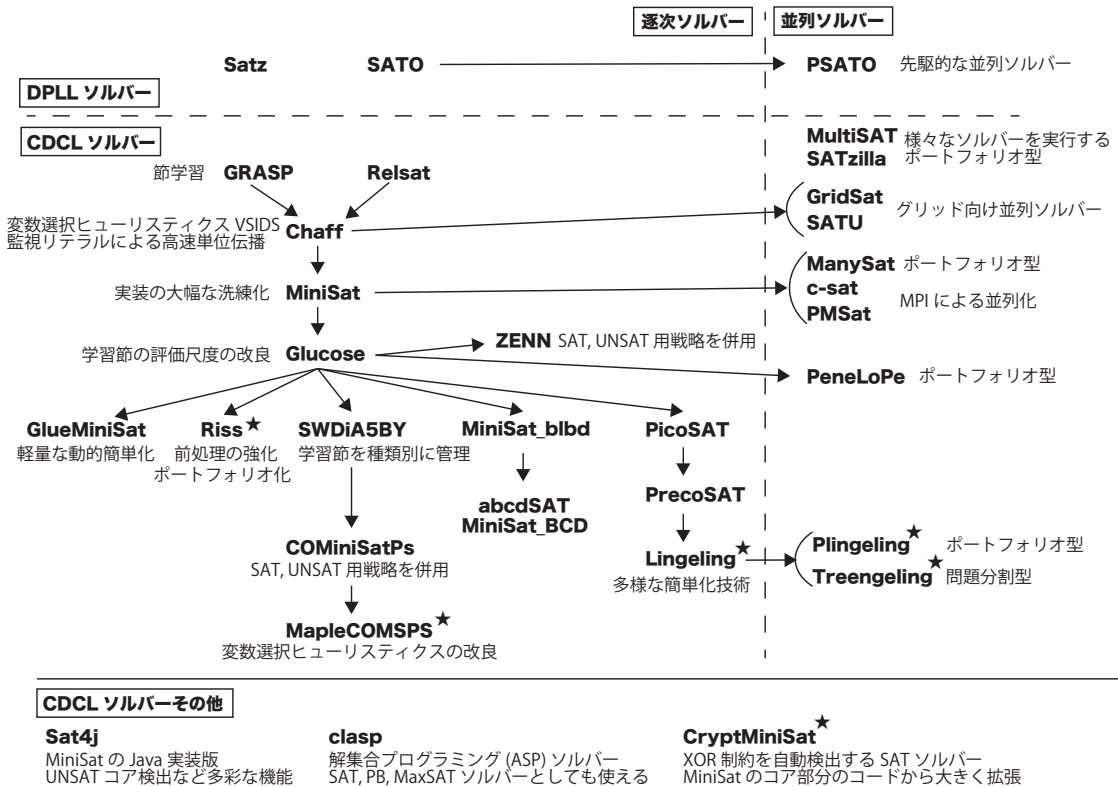


図 1 代表的な SAT ソルバーの系統樹。

星印 (★) は 2016 年に開催された SAT 競技会 の Main/Parallel トラック (逐次, 並列) で入賞したソルバー。

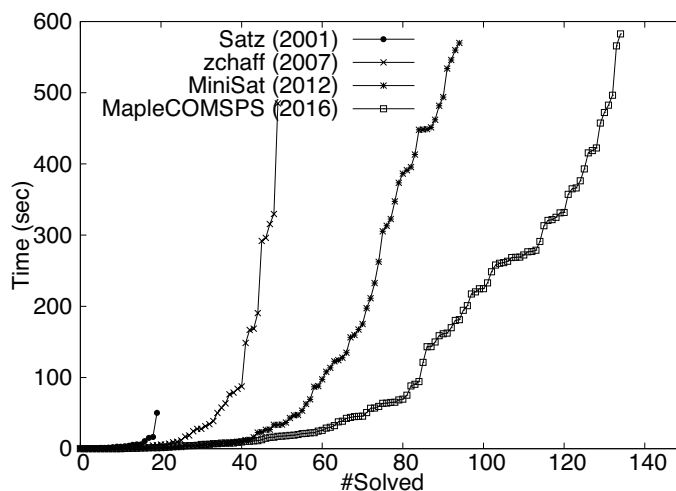


図 2 代表的な SAT ソルバーの求解性能の比較。

2016 年に開催された SAT 競技会 の Main トラックのベンチマーク 500 問を使用。

(3.1 節) とそれ以降 (3.2 節) に分けて説明する。

3.1 DPLL ソルバーから MiniSat までの技術

DPLL [19] と呼ばれる 1962 年に提案されたアルゴリズムが現在の系統的探索を行う SAT ソルバーの基礎になっている。その後、以下のアルゴリズム、実装技術、ヒューリスティクスが DPLL と組合わされ最新 SAT ソルバーの基となっている **MiniSat** [21] が登場した。

- **CDCL** [14] [57]. 探索過程で矛盾が発生した際にその原因を解析・記録することで同様の矛盾の発生を防ぐ節学習とそれに基づくバックジャンプ法を DPLL と組合せたアルゴリズム。
- 2 リテラルウォッチ [59]. 効率の良い単位節検出のためのアルゴリズムおよびデータ構造。CPU データキャッシュのミスヒット率を低減する効果もある。
- 変数選択ヒューリスティック **VSIDS** (Variable State Independent Decaying Sum) [59]. 変数 (もしくはリテラル) の学習節への出現回数を基に優先度を定める低計算コストのヒューリスティクス。
- リスタート [54] [76] [21]. 矛盾の発生回数がある閾値を超えた場合に、全ての部分値割当てをキャンセルし、探索を最初からやり直すことで、局所的な探索に陥ることを防ぎ、安定した性能を獲得する手法。
- 値割当てのキャッシュ (Phase Saving) [73]. 変数へ割当てられた値を保存しておき、次回値割当てされる時の初期値として利用する手法。

上に述べた 5 つの技術の効果について説明する。図 3 の上下 2 つのグラフは、論文 [43] の発表スライド資料に記載されたグラフを著者の許可を得て 2 つに分けて記述したものである。図 3 の上のグラフは、もっとも基礎的な DPLL アルゴリズムに、上述の各々の技術を 1 つ追加したソルバーの求解性能を示している。もっとも大きく性能向上が見られるのは、DPLL に CDCL を追加したソルバーであり、次に DPLL に VSIDS を追加したソルバーの性能向上が大きいことが分かる。

図 3 の下のグラフは、上述の技術を全て実装した代

表的な SAT ソルバーである **MiniSat** から、逆に各々の技術を 1 つ削除したソルバーの求解性能を示している。このグラフからも CDCL を削除した時に性能劣化がもっとも顕著であり、CDCL が現在の SAT ソルバーのコア技術となっていることがわかる。

3.2 最新 SAT ソルバーの高速化技術

MiniSat 以降の最新の SAT ソルバーに実装されている高速化技術について説明する。

学習節の評価尺度. CDCL は、SAT ソルバーで矛盾が発生した際に、その矛盾の原因を解析し、学習節として元の CNF 式に追加することで以後同様の矛盾が起こることを防ぐ。しかし、学習節の過度の増加は、探索速度の低下をまねくため、“良い” 学習節のみを保持していくことが望ましい。そのため学習節の良さを測るための評価尺度が研究されており、中でも **LBD** (literal block distance) と呼ばれる評価尺度が **Glucose** 以降の SAT ソルバーに取り入れられている [3] [62]。この他に、学習節がどれぐらい探索に使われたかというカバレッジを用いた評価尺度も 2017 年に提案されている [61]。

簡単化. 問題として与えられた CNF 式には、冗長な変数や節が含まれることがあり、簡単化を行うことでより少ない変数や節をもつ充足同値な CNF 式に変換できることがある。SAT ソルバーが問題を解く前に行う静的な簡単化技術として、Subsumption elimination, Variable elimination, Probing 等の技術が研究されている [55] [84] [20]。例えば、Subsumption elimination では、2 つの節 c と c' があり、 $c \rightarrow c'$ が成り立つならば節 c' を削除するという簡単化を行う。また SAT ソルバーが SAT 問題を求解中に行う動的な簡単化技術として、Hyper binary resolution [5], Clause elimination [31] [41], Equivalent literal substitution [15] [6], Unhiding [32], Self-subsumption resolution [29] [28], Lazy clause simplification [63] などが提案されている。

変数選択ヒューリスティクス. ほとんどのソルバーは、前述した VSIDS (Variable State Independent Decaying Sum) [59] を変数選択ヒューリスティクスとして採用している。2016 年の SAT 競技会の Main

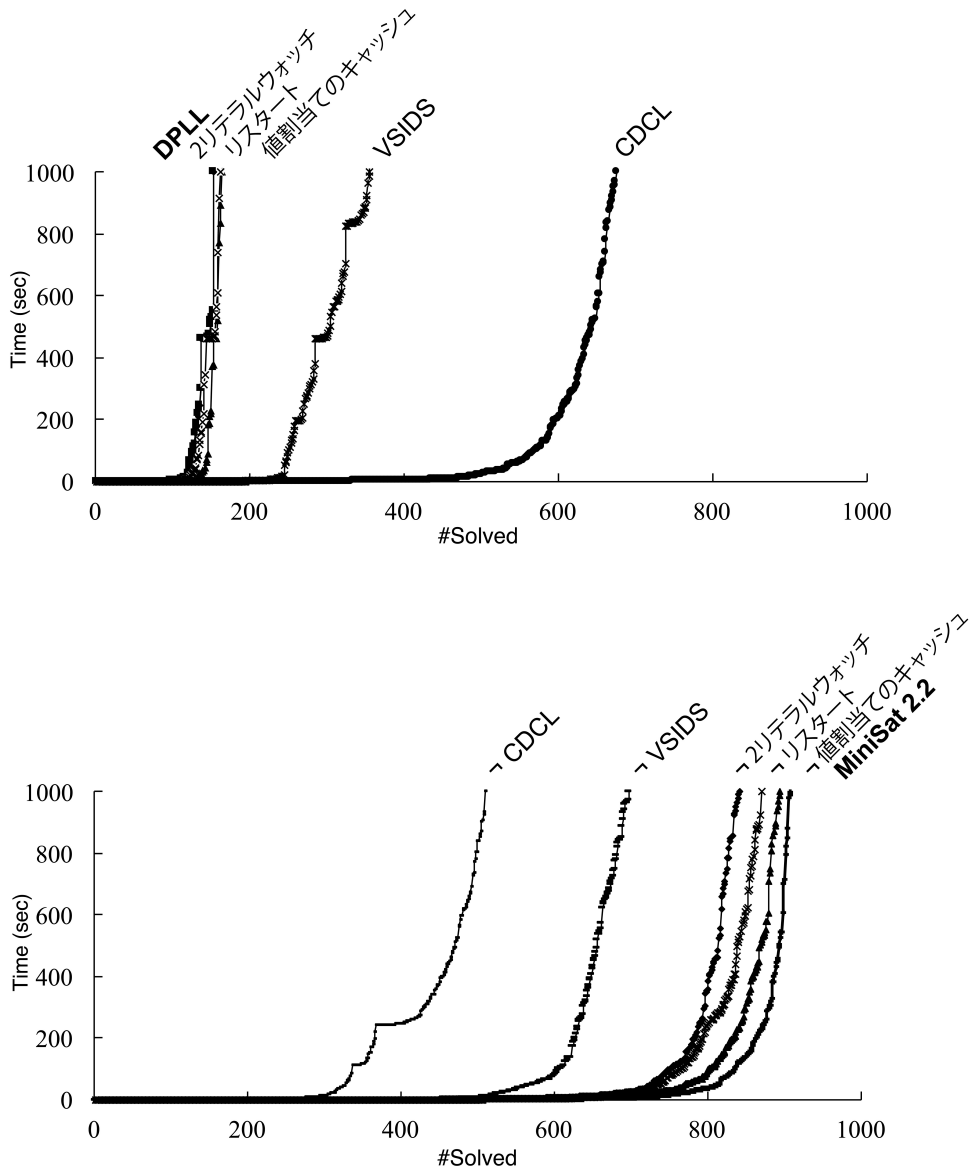


図3 2リテラルウォッチ, リスタート, 値割当てのキャッシュ, VSIDS, CDCL を DPLL に追加したソルバー (図上) と代表的な CDCL ソルバー MiniSat から削除したソルバー (図下) の性能比較. 論文[43]の発表スライド資料に掲載されたグラフを著者の許可を得て上下2つに分けて記載.

トラックで優勝した MapleCOMSPS では Learning Rate Based Branching Heuristic (LRB)[53] と呼ばれる強化学習を使ったヒューリスティクスが提案され, 逐次的に VSIDS と切り替えて使用されている.

並列化. SAT ソルバーの実行時間の大部分 (80%~90%) は, 単位伝播と呼ばれる値割当ての伝播処理で

占められることが知られている. この処理は逐次性の高い処理で並列化が難しい. そのため, 複数の逐次 SAT ソルバーを並列実行する並列 SAT ソルバーがこれまで提案されている. この並列 SAT ソルバーは主に2種類に大別される. 探索空間分割型は, 一部の命題変数の真値を仮定することで複数の部分問題を生

成し、それぞれの部分問題を複数の SAT ソルバーで解く並列ソルバーである [56] [2]. 与えられた SAT 問題が充足可能であることは、1 つの部分問題で充足解が発見されれば良い。一方、充足不能であることは、全ての部分問題が充足不能であることを示さなければならない。ポートフォリオ型は、同一の問題を複数の異なる SAT ソルバーで並行して解く並列ソルバーである [9] [4]. SAT ソルバーは、複数の異なる種類のソルバーを選択することもあれば、同一の SAT ソルバーの探索パラメータを変更して、並列実行することもある。どれか 1 つの SAT ソルバーが充足可能もしくは充足不能を返せば、直ちに計算が完了する。

4 SAT 競技会

SAT ソルバーの高速化技術の継続的な発展を支えた 1 つの要因は、SAT ソルバーの国際競技会 (SAT 競技会) の存在である。本節では、SAT 競技会の説明を通して、SAT ソルバーの進化に与えた影響を明らかにする。また 2016 年の SAT 競技会の結果を通して、最新の動向を説明する。

4.1 競技会の概要

SAT ソルバーの競技会 (SAT 競技会) は、古くは 1990 年代に数回開催され、2002 年からは継続して毎年開催されている。

SAT 競技会は基本的に、“参加者が自分のソルバーを指定されたサーバにアップロードし、参加者によるテスト期間後、最終版がオーガナイザ側で実行されて、SAT 国際会議の時に結果が公表される”という流れで進められる。

SAT 競技会の大きな特長は、参加ソルバーのシステム記述の提出や、ソースコードの公開が義務付けられていることである。つまり、最新の技術・実装を誰でも入手でき、自分のアイデアを上乗せして試すことができる。このことは、新規参入の敷居を低くし、ソルバー開発競争を激しくし、ほぼ毎年優勝ソルバーが変わり、何らかの新しいアイデアが優勝ソルバーに入っている、という循環を促し、SAT ソルバーの性能面における継続した進化に大きく貢献していると考えられる。

4.2 競技会の変遷と SAT ソルバーに与えた影響

SAT 競技会で出題されるベンチマーク問題や評価の方法の変遷を辿ると、SAT ソルバーにどういった性能や機能が求められてきて、どのように影響したのかが分かってくる。以下に特に大きな変化を説明する。

コアソルバーの定義。ポートフォリオ型の SAT ソルバーは、複数種類の既存 SAT ソルバーを問題に応じて使分ける (もしくは一定時間で切替える) ことで、1 つの CPU 上であっても比較的容易に競技会での成績を向上することができる。しかし、このような傾向はポートフォリオ型の計算能力の源泉でもある逐次ソルバー (単一戦略のソルバー) の進化を阻害することになる。このことから 2013 年の競技会から、逐次ソルバーの競技部門に出場できるソルバー (コアソルバーと呼ばれる) が、“高々 2 つまでの SAT ソルバーで構成されるソルバー”と定義された。ポートフォリオ型に頼らない逐次ソルバーの高速化技術が引き続き求められていることが分かる。

特別トラックの追加。いくつかの SAT 競技会では、CNF 式の充足可能性判定以外の機能の性能を競う競技部門が設けられており、充足可能性判定のみを計算するという単一機能の SAT ソルバーからの進化を促してきた。特に Certified UNSAT トラック、極小 UNSAT サブセット (Minimal Unsatisfiable Subset; MUS) トラック、インクリメンタル SAT トラックで必要となる機能は、SAT ソルバーの応用で使われることが多く、これらの機能が SAT ソルバーに求められていることがわかる。SAT ソルバーの機能面における進化については、5 節でもう少し詳しく解説する。

Agile トラックの追加。2016 年に Agile トラックと呼ばれる競技部門が新たに設定された。Agile トラックでは、比較的簡単に解けるベンチマーク問題を 1 問あたり制限時間 1 分で何問解けるかを競う。問題は 5000 問用意されており、Main トラックが 500 問のベンチマークを 1 問あたり 5000 秒の制限時間で解く設定になっていることを考慮すると、Agile トラックは立ち上がり早い、低いオーバーヘッドを持つ SAT ソルバーを評価するために設けられた部門だといえる。CEGAR (Counterexample Guided

Abstraction Refinement), SMT ソルバー, 最適化問題などの SAT ソルバーの応用・拡張では, SAT ソルバーが数千から何十万回も呼び出されることがあるため, このような使い方に適したソルバーの開発が求められていることが分かる. 実際に 2016 年は SMT ソルバーの探索から得られた CNF 式が問題として使用された.

MiniSat/Glucose Hack トラックの追加. 上述した定義やトラックの他に, SAT ソルバー開発を促したものととして 2009 年に追加された Minisat Hack (2016 年からは Glucose Hack) トラックがある. このトラックは, 当時 SAT ソルバーの代表的な実装になりつつあった MiniSat に 125 行以内の変更のみを行った SAT ソルバーが参加できるトラックである. 主催者の意図は, SAT 競技会参加の敷居を低くすることと, このような小さな変更でどれくらい大きく性能を向上出来るのかを確認することであった. MiniSat Hack トラックには毎年多くの SAT ソルバーが参加しており, また (制限なしの) Main トラックで入賞するソルバーも珍しくない. 例えば 2011 年には, MiniSat Hack トラックで優勝した Cir Minisat Hack というソルバーが Main トラックの SAT 部門で準優勝を果たしている. このような事実は, 若い研究者の参入をさらに促すことになり, 結果として SAT ソルバーの性能面での継続的な進化を支えていると考えられる.

4.3 2016 年の SAT 競技会の結果

表 1 は, 2016 年 SAT 競技会の Main トラック, Parallel トラックの結果である. 両トラックは, ソフトウェア・ハードウェア検証, プランニング・スケジューリング等の人工知能分野の問題, 定理証明などの問題で構成されており, 実問題における各 SAT ソルバーの性能が分かるもっとも重要な部門である.

Main トラックは逐次ソルバーのみ参加可能であり, 逐次ソルバーの性能は並列ソルバーの土台となるため, 毎年もっとも重要視されている. 2016 年に 1 位だった MapleCOMSPS は, MiniSat から派生した SAT ソルバーであり (図 1 参照), 強化学習を取り入れた変数選択ヒューリスティック [53] と既存の VSIDS を両方使うソルバーである. 図 2 に示した 2001 年か

らの SAT ソルバーと MapleCOMSPS の比較からもわかるように, 性能が年を経るごとに向上しており, SAT 競技会の Main トラックが支えた逐次ソルバーの性能面における進化を確認することができる. この他に入賞した Riss と Lingeling も MiniSat から派生したソルバーであり, SAT ソルバー開発における MiniSat の影響の大きさをうかがうことができる.

Parallel トラックでは 24 コア (48 スレッド) とメモリ 64GB (逐次は 24GB) の共有メモリ環境上で, Main トラックの逐次ソルバーと同じベンチマークを使って並列 SAT ソルバーの性能が評価される. Main トラックで 3 位だった Lingeling を使った探索空間分割型の並列 SAT ソルバーである Treengeling と, 同じく Lingeling を使ったポートフォリオ型の並列 SAT ソルバーである Plingeling が 1 位と 2 位を獲得した. 逐次ソルバーの結果と比較すると約 1.5 倍の数の問題が解けるようになっており, 大幅に性能が向上していることが分かる. Parallel トラックには毎年 10 以上のソルバーが参加しており (2016 年も 11 のソルバーが参加), 継続してその性能が競われている.

図 4 は, 逐次ソルバーである Lingeling とその並列版である Treengeling と Plingeling の性能を比較したカクタスプロットである. パラメータが異なる複数の逐次ソルバーを並列実行する素朴なポートフォリオ型でも大幅に性能が向上している. 問題を部分問題に分割して解く探索空間分割型ではさらに求解性能の伸びが見られる.

SAT 競技会に参加したソルバーのシステム記述や結果の詳細は, 2016 年度の競技会ウェブページから誰でも取得することができる^{†9}. もしくは, 競技会のオーガナイザが 3 ページのサマリ版を公開しているので, 概要を知りたい方はこちらを読むことをお勧めする^{†10}.

5 SAT ソルバーの機能面における進化

通常の SAT ソルバーは, 2 節で述べたように, 与えられた SAT 問題の充足可能性を判定する単一機能

^{†9} <http://baldur.itl.kit.edu/sat-competition-2016/>

^{†10} <http://www.cs.utexas.edu/~marijn/publications/hot2017.pdf>

表 1 2016 年 SAT 競技会の結果 (逐次, 並列)

トラック名	1 位	2 位	3 位
Main トラック (500)	MapleCOMSPS (203)	Riss (202)	Lingeling (201)
Parellel トラック (500)	Treengeling (315)	Plingeling (302)	CryptoMiniSat (297)

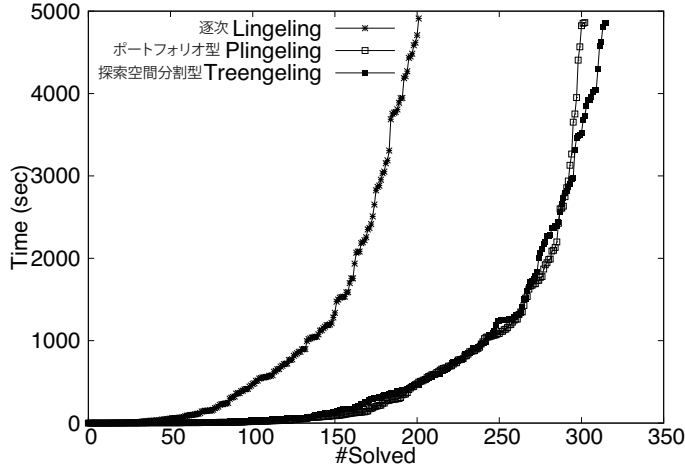


図 4 逐次ソルバー Lingeling と並列ソルバー Plingeling (ポートフォリオ型), Treengeling (探索空間分割型) の比較.

のプログラムであった. しかし, SAT ソルバーが性能面において進化し, 応用の需要が増えるのに従って, SAT ソルバーは単一ではなく複数の機能をもつように進化してきた. 本節では, SAT ソルバーの充足可能性判定以外の機能を紹介する.

5.1 Certified UNSAT

通常の SAT ソルバーの場合, 与えられた問題が充足不能なら単に充足不能と回答するだけで, 利用者にはなぜ充足不能になったのかの理由はわからない. Certified UNSAT 機能は, SAT 問題が与えられ充足不能と判定された時に証明を出力する機能である. 通常, 証明を出力するだけでなく, 検証ツールで証明の検証も行われるため, 検証が容易になるよう簡潔な証明を出力することが要求される. その原理は反駁に至る導出木が正しいか検証することであるが, 効率的な検証のために, ソルバーが探索過程で獲得する各学習節が正しいかを順に調べる手法が利用されている [94].

Certified UNSAT 機能は現在多くの SAT ソルバー

で利用可能である. これらのソルバーでは, 問題が充足不能の時に判定結果だけでなく, その証明を出力し, 検証することができる [94].

実際に SAT ソルバーが使われた Erdős discrepancy 予想の $C = 2$ の場合の解決 [46] や Boolean Pythagorean Triple 問題の解決 [33] といった未解決問題の解決については, Certified UNSAT 機能を使うことで証明が正しいことが確認されている. 特に後者は証明が 200 テラバイト^{†11} になったことから Nature [50] に “Two-hundred-terabyte maths proof is largest ever” と題して取り上げられて話題になった.

5.2 極小 UNSAT サブセット

与えられた問題が充足不能であるときに, 具体的に問題のどの部分が充足不能なのか分かると, SAT ソルバーを使って最適化問題などを解く際に効果的に探

^{†11} ただし, 200 テラバイトは最大サイズで, 論文の著者らは証明の圧縮技術を使うことで 64 ギガバイトまで小さくしている [33].

索を行えることがある。UNSAT サブセット (Unsatisfiable Subset) 機能とは、充足不能である SAT 問題 Ψ における、充足不能である部分節集合 $\Psi' \subseteq \Psi$ を 1 つ出力 (もしくは列挙) する機能のことをいう。極小 UNSAT サブセット (Minimal Unsatisfiable Subset; MUS) 機能とは、任意の節 $c \in \Psi'$ について $\Psi' \setminus \{c\}$ が充足可能であるような UNSAT サブセット Ψ' を 1 つ出力 (もしくは列挙) する機能のことをいう。

UNSAT サブセットを計算するためには、反駁に至る導出木を利用し、求められた充足不能性が元の問題のどの節から導かれたのかを解析する。UNSAT サブセットから極小 UNSAT サブセットを求める素朴な方法は UNSAT サブセットの節集合から節を 1 つずつ削除して、充足不能かどうかを確認する。充足不能であれば、その節は不要である。UNSAT サブセットの節集合からどの節を削除しても、充足可能になるとき、その UNSAT サブセットは極小 UNSAT サブセットである。

極小 UNSAT サブセットを計算する機能は、**Sat4j**^{†12} や **HaifaHLMUC**^{†13} などの SAT ソルバーで利用することができる。なお SAT 競技会の特別トラックでは、極小 UNSAT サブセットが課題として取り上げられたが、計算コストが高いため応用では UNSAT サブセットが使われることが多い。

UNSAT サブセットは、SAT ソルバーを利用した MaxSAT 問題の解法 [58] など実際に最適化問題の解法に利用され成功している。実用的な問題への応用としては解集合プログラミング (ASP) ソルバーが UNSAT サブセットを用いた最適化により時間割問題の既存手法を大きく上回る性能を示すことも報告されている [12]。

5.3 インクリメンタル SAT

インクリメンタル SAT とは、SAT 問題に節を追加しながら求解を繰り返す解法 (インクリメンタル SAT 解法) [22] であり、CEGAR や最適化問題などに SAT ソルバーを応用する際に必要となる。高速なインクリメンタル SAT ソルバーを実現するには、探索情報の

再利用が鍵になる。学習節の再利用は典型的な方法であり、その効果がジョブショップスケジューリング問題や、プランニング問題で確認されている [66]。SAT ソルバーのインクリメンタル SAT 機能は、SAT 問題を連続的に解くための API を提供し、学習節などの探索情報を再利用可能にするものである。このようなインクリメンタル SAT 機能は、**MiniSat** が対応し、その後継ソルバーのほとんどで利用可能である (図 1 参照)。

学習節の他にも変数や節数の活性度など再利用できる情報はいくつか考えられる。前節で説明した SAT 競技会の特別トラックにおけるインクリメンタル SAT トラックは 2015 年の SAT 競技会から始まった比較的新しいトラックであり、これからの研究の発展が期待される。またこのトラックでは、標準 API である IPASIR (Re-entrant Incremental Satisfiability Application Program Interface) の頭文字を逆に並べたもの^{†14}の仕様が公開されており、機能の標準化が進んでいる。

インクリメンタル SAT の応用は、最適化問題や列挙問題など多岐にわたる。近年ではインクリメンタル SAT により実現した CEGAR 解法をハミルトン閉路問題へと適用した研究がある [81]。

6 SAT ソルバーの利用技術

SAT ソルバーの入力は素朴な CNF 式であるため、利用して問題を解きたくても、問題と CNF 式のギャップが大きく、つまづいてしまうことがある。このような問題を解決するために、SAT ソルバーを利用するための技術が研究されてきた。

SAT ソルバーを拡張して利用する技術。代表的なものとして、背景理論付き SAT (Satisfiability Modulo Theory; SMT) ソルバーがある [39] [91] [37]。SMT ソルバーは、配列などのデータ構造や算術演算など様々な背景理論付きの一階理論を扱えるソルバーである。現在 SMT ソルバーで主流となっている遅延処理型 SMT 技術では、背景理論を取扱う既存の専用ソルバーと連携できるように SAT ソルバーが拡張されて

^{†12} <http://www.sat4j.org>

^{†13} <http://strichman.net.technion.ac.il/>

^{†14} <http://baldur.iti.kit.edu/sat-race-2015/index.php?cat=rules>

いる。既存の背景理論ソルバーが利用できるという点で柔軟性があり、SAT ソルバーの求解性能を比較的幅広い問題に応用することができるのが利点である。SMT ソルバーの他にも、非単調論理プログラミングが制約プログラミングの概念と融合した解集合プログラミング (Answer Set Programming; ASP) における ASP ソルバー [25] が、SAT ソルバーを拡張する方法で実現され、成功を収めている。

SAT ソルバーを直接利用する技術。 解きたい問題を CNF 式へと変換する符号化技術 [89] も活発に研究されている。この方法では、最新の SAT ソルバーをそのまま利用できるというメリットがある。また、符号化をうまく工夫することが出来れば、問題によっては SMT ソルバーや既存の専用ソルバーの性能を上回ることもしばしばある。

以降、本節では著者らの専門分野でもある符号化技術や、符号化とインクリメンタル SAT、UNSAT サブセットを組合せた多目的最適化問題の解法を通して、SAT ソルバーを利用する技術を紹介する。

6.1 符号化技術

SAT 型システムは、与えられた問題を SAT 符号化により SAT 問題へと変換し、SAT ソルバーを用いて解くシステムである (図 5 参照)^{†15}。SAT 符号化とは、与えられた問題を SAT ソルバーで扱えるようにするため、与えられた問題 (もしくは部分問題) を CNF 式に変換 (符号化) することである。問題を CNF 式に符号化できれば何でも良いかというのではなく、符号化方法の違いで SAT 型システムの性能が大きく異なることも珍しくない。そのため、SAT 符号化は SAT ソルバーと同様に重要な研究分野である。The Art of Computer Programming 4 巻分冊 6 [45] の 97 ページでも以下のように符号化の重要性が SAT ソルバーと比較して述べられている。 “*Thus the art of problem encoding turns out to be just as important as the art of devising algorithms for satisfiability.*”

^{†15} 最適化問題などインクリメンタル SAT を使って問題を解く場合には SAT ソルバーで複数回部分問題を解くため、枠組みにループ構造が必要になる場合があるが、ここでは省略する。

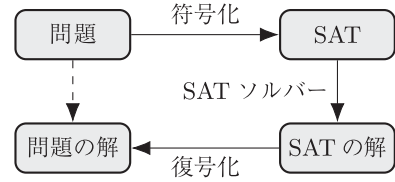


図 5 SAT 型システムの枠組み。

例えば、以下のような整数有限領域上の線形の不等式で表される制約の SAT 符号化を考えてみよう。

$$\sum_{i=1}^n a_i x_i \leq k \quad (a_i > 0)$$

ここで、全ての a_i が 1、 x_i が 0-1 変数の場合、この基数制約と呼ばれる単純な場合でも、数多くの SAT 符号化が研究されている [93] [7] [77]。 a_i が 1 以外の整数値を取る場合、擬似ブール制約と呼ばれる。この場合も同じく数多くの研究がある [23] [8] [70] [86] [72] [74]。さらに、 x_i が整数有限領域上のドメインを持つ場合の SAT 符号化の研究も活発に行われている [38] [92] [26] [88] [90] [78]。このように SAT 符号化の研究は SAT 分野における重要なテーマの 1 つであることが分かる。

以降では、もっとも一般的な形である x_i が整数有限領域上のドメインを持つ場合を取り上げて SAT 符号化を説明する。基数制約と擬似ブール制約も MaxSAT 問題などへの応用があり重要であるが、興味のある読者は上述の文献を参照いただきたい。特に著者らの擬似ブール制約符号化に関する論文 [86] では、以下で説明する順序符号化と BDD を利用した擬似ブール制約の符号化 [8] および有名な基数制約符号化である Sequential Counter [77] との関係が述べられているので、一読いただければ幸いである。

例として線形不等式 $x + y \leq 7$ (ただし、 x と y は整数変数で $x, y \in \{2, 3, 4, 5, 6\}$ とする) で表される制約を用いて、代表的な 2 つの符号化である直接符号化 [92] と順序符号化 [88] にどのような違いがあるのかを説明する。

直接符号化 [92] では、整数変数 x とそのドメインの各値 i に対して、 $x = i$ を表す命題変数 $p_{x=i}$ を用いる。整数変数 x に対して用いる命題変数は以下の 5 つになる。

$$p_{x=2}, \quad p_{x=3}, \quad p_{x=4}, \quad p_{x=5}, \quad p_{x=6}$$

さらに命題変数 $p_{x=i}$ が $x = i$ の時かつその時に限って真となるように、以下の at-least-one 節と at-most-one 節を追加する。

$$p_{x=2} \vee p_{x=3} \vee p_{x=4} \vee p_{x=5} \vee p_{x=6}$$

$$\neg p_{x=i} \vee \neg p_{x=j} \quad (2 \leq i < j \leq 6)$$

以上の命題変数と節を整数変数 y に対しても同様に導入する。

最終的に制約 $x + y \leq 7$ ($x, y \in \{2, 3, 4, 5, 6\}$) は、図 6 に示すように制約に違反する点 (図 6 右) を 1 つずつ節で表すことで 15 節 (図 6 左) に符号化される。例えば $\neg p_{x=2} \vee \neg p_{y=6}$ は $\neg(p_{x=2} \wedge p_{y=6})$ を意味しており、 $x = 2$ かつ $y = 6$ の時に、制約に違反することを表している。

順序符号化[88]では、整数変数 x とそのドメインの各値 i に対して、 $x \geq i$ を表す命題変数 $p_{x \geq i}$ を用いる。整数変数 x に対して用いる命題変数は以下の 4 つになる。ここで $p_{x \geq 2}$ は常に真となるため出力されないことに注意されたい。

$$p_{x \geq 3}, p_{x \geq 4}, p_{x \geq 5}, p_{x \geq 6}$$

さらに命題変数 $p_{x \geq i}$ が $x \geq i$ の時かつその時に限って真となるように、以下の節を追加する。

$$\neg p_{x \geq i} \vee p_{x \geq i-1} \quad (2 < i \leq 6)$$

最終的に制約 $x + y \leq 7$ ($x, y \in \{2, 3, 4, 5, 6\}$) は、図 7 に示すように制約に違反する領域 (図 7 右) を節で表すことで 5 節 (図 7 左) に符号化される。例えば、 $\neg p_{x \geq 4} \vee \neg p_{y \geq 4}$ は $\neg(p_{x \geq 4} \wedge p_{y \geq 4})$ を意味しており、 $x \geq 4$ かつ $y \geq 4$ の時に、制約に違反することを表している (図 7 右の灰色の領域)。

順序符号化を実装した制約充足問題を解くための SAT 型システム Sugar[85] は 2008 年、2009 年に国際 CSP ソルバー競技会のグローバル部門で優勝し、様々な制約最適化問題を解くことに用いられるなど [10][83][49]、性能が高く評価されている。この他に順序符号化は、現在知られている符号化の中で、効率的に解けるクラスに属する制約充足問題 (Tractable CSP) を同じく効率的に解けるクラスに属する SAT 問題 (Tractable SAT) に符号化できる唯一の方法として知られている [71]。解説論文 [79] では、順序符号化が SAT 型制約プログラミングシステムにおける実装の視点から説明されているので、興味のある読者は

参考にされたい。

ハイブリッド符号化 [78] は、比較的近年提案された SAT 符号化であり、順序符号化と対数符号化を統合した符号化である。順序符号化は上述したように良い性質や性能を持っているが、与えられた制約の変数のドメインが大きくなると符号化節数が大きくなりすぎて問題が解けなくなってしまうことがある。対数符号化は、各整数変数 x の 2 進表現に着目し、 x の i 桁目 (LSB を 1 桁目とする) が 1 に等しいことを表す命題変数 $p_{x,i}$ を用いる SAT 符号化であり、制約を比較的少ない節数に符号化できる [38]。ハイブリッド符号化は、順序符号化の短所を対数符号化で補うことで、性能を向上することに成功している [78]。

図 8 は、順序符号化、対数符号化、ハイブリッド符号化を、国際制約充足問題 (CSP) ソルバー競技会のベンチマーク 1458 問で比較評価したものである [78]。図より、順序符号化、対数符号化と比較してハイブリッド符号化の性能が向上していることが分かる。またハイブリッド符号化を実装した SAT 型システム Diet-Sugar を同じベンチマークを用いて SAT 型でない他のシステムと比較したものが、図 9 である。比較に用いたソルバーは、Mistral (version 1.550), Opturion CPX (version 1.0.2), Yices (version 2.4.2, QF_LIA), z3 (version 4.3.2, QF_LIA) である。Mistral, Opturion CPX はそれぞれ国際 CSP ソルバー競技会の 2009 年、2015 年の優勝ソルバーであり、Yices と z3 は 2015 年の国際 SMT ソルバー競技会の優勝、準優勝ソルバーである。図 9 から分かるように、このベンチマークではハイブリッド符号化を実装した SAT 型システムが、より良い性能を示していることが分かる。

6.2 多目的最適化への SAT ソルバーの利用

前節では、整数有限領域上の線形の不等式で表される制約の SAT 符号化の説明を行った。本節では、複雑な問題に対する利用技術の例として、論文 [80] で提案された SAT ソルバーを多目的最適化問題に応用する方法を説明する。

多目的最適化問題 (Multiobjective optimization problem) とは、複数の目的関数をもつ最適化問題で

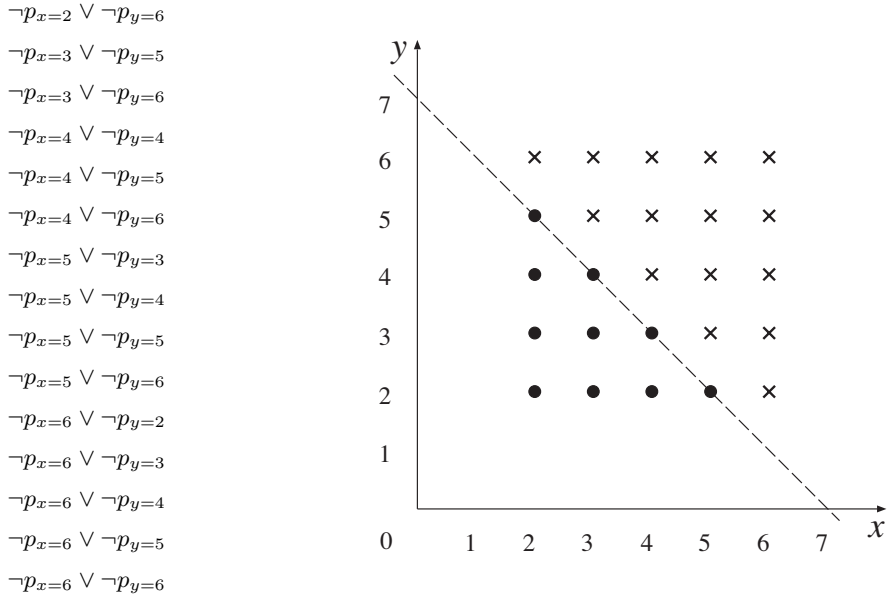


図 6 直接符号化. 右図の違反点を表す × 印が左図の節に対応している.

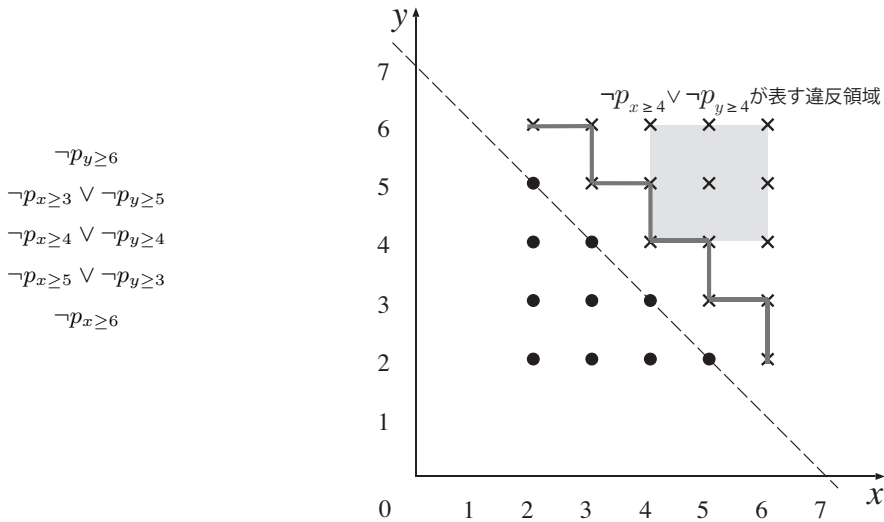


図 7 順序符号化. 右図の違反領域を表す線が左図の節に対応している.

ある. 最適解はパレート最適性 (Pareto optimality) によって定義される^{†16}. 多目的最適化問題は, 目的関数間のトレードオフを自然に表現可能であり, 人工知能分野のホットピックの 1 つになっている.

^{†16} 多目的最小化問題の場合, パレート最適とは, いずれかの目的関数の値を大きくしなければ, 他のどの目的関数の値も小さくできない状態のことである.

例として, 2つの目的関数 f_1 と f_2 を持つ以下の整数有限領域上の多目的最適化 (最小化) 問題を考える.

- 整数変数: $x, y \in \{0, 1, 2, 3, 4\}, z \in \{0, 1\}$

- 制約: $(x + y \geq 3) \wedge$

$$(x \neq 1 \vee y \neq 2 \vee z = 1) \wedge$$

$$(x \neq 2 \vee y \neq 1 \vee z = 1)$$

- 目的関数: $f_1 = x + 1, f_2 = y + 2z + 1$

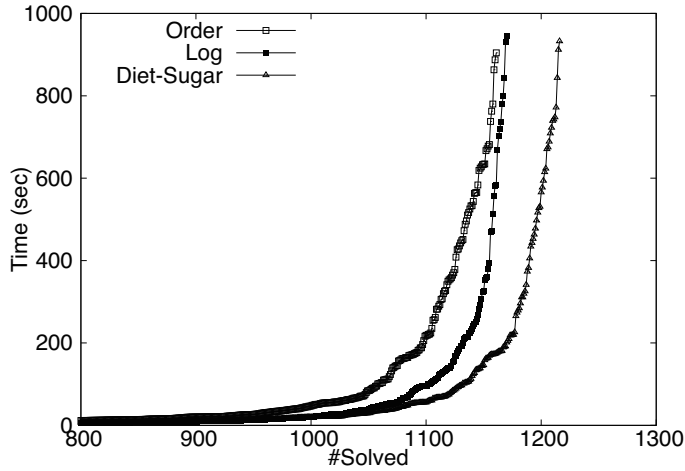


図 8 順序符号化 (Order), 対数符号化 (Log), ハイブリッド符号化 (Diet-Sugar) の比較 [78]. 国際 CSP ソルバー競技会のベンチマーク 1458 問で評価.

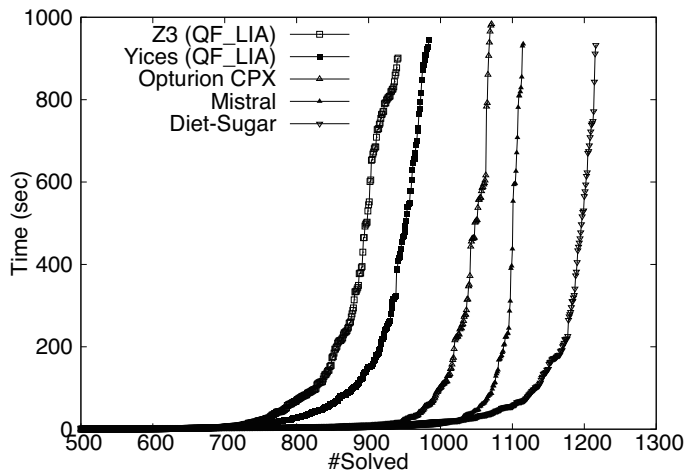


図 9 SMT ソルバー (z3, Yices), CSP ソルバー (Mistral, Opturion CPX) とハイブリッド符号化を実装した SAT 型システム Diet-Sugar との比較 [78]. 国際 CSP ソルバー競技会のベンチマーク 1458 問で評価.

与えられた整数変数 x, y, z 上の制約を満たしつつ、パレート最適性を満たす目的関数の値の全ての組合せが問題の解となる (パレートフロントと呼ばれる). この例の目的関数を取りうる値の組合せを表したのが、図 10 である. これら (f_1, f_2) の値の組合せの中で、パレート最適性を満たすものは、左下に他の値の組合せが存在しない点であり、二重丸で表された $(1, 4), (3, 3), (4, 1)$ の 3 つである. 最終的にこの問題の解、パレートフロントは $\{(1, 4), (3, 3), (4, 1)\}$ と

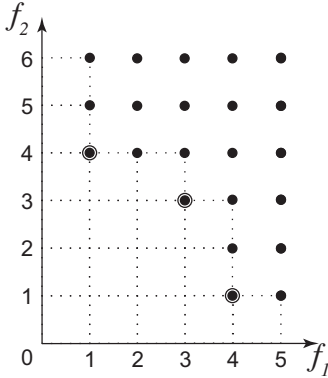
なる.

この問題を SAT ソルバーを利用して解くための鍵となるのは、順序符号化と後述する P -極小モデル [67][48] である. SAT ソルバーを利用したパレートフロントの計算は、次の手順で行うことができる.

1. 目的関数の値を表す目的変数を導入する.
2. 目的変数を順序符号化を使って CNF 式に符号化する. 他の制約については、任意の SAT 符号化で符号化する.

表 2 P -極小モデルと対応する目的関数の値の組

$p_{o_1 \geq 2}$	$p_{o_1 \geq 3}$	$p_{o_1 \geq 4}$	$p_{o_1 \geq 5}$	$p_{o_2 \geq 2}$	$p_{o_2 \geq 3}$	$p_{o_2 \geq 4}$	$p_{o_2 \geq 5}$	$p_{o_2 \geq 6}$	$p_{o_2 \geq 7}$	(f_1, f_2)
0	0	0	0	1	1	1	0	0	0	(1,4)
1	1	0	0	1	1	0	0	0	0	(3,3)
1	1	1	0	0	0	0	0	0	0	(4,1)

図 10 問題例の目的関数を取りうる値の組合せ (黒点).
二重丸の 3 つの点がパレート最適.

3. 目的変数から符号化された命題変数の集合を P とし, 符号化された CNF 式の P -極小モデルを全て列挙する.

この 3 つの手順を上述の例を使ってもう少し詳しく説明する.

最初に, 目的関数 f_1 と f_2 に対応する目的変数 o_1 と o_2 を導入する. この時に, o_1 と o_2 のドメインは x, y, z のドメインより, それぞれ $\{1, 2, 3, 4, 5\}$ と $\{1, 2, 3, 4, 5, 6, 7\}$ となる. また元の制約に加えて目的変数を定義する制約 $o_1 = x + 1$ と $o_2 = y + 2z + 1$ も併せて導入する.

次に目的変数 o_1 と o_2 を順序符号化で符号化する. 目的変数 o_1 に対しては, 4 つの命題変数 $p_{o_1 \geq 2}$, $p_{o_1 \geq 3}$, $p_{o_1 \geq 4}$, $p_{o_1 \geq 5}$ と次の 3 つの節が得られる.

$$p_{o_1 \geq 2} \vee \neg p_{o_1 \geq 3}$$

$$p_{o_1 \geq 3} \vee \neg p_{o_1 \geq 4}$$

$$p_{o_1 \geq 4} \vee \neg p_{o_1 \geq 5}$$

この 3 節からなる CNF 式を充足する値割当ては, 次のようになる (真を 1, 偽を 0 で表す).

$p_{o_1 \geq 2}$	$p_{o_1 \geq 3}$	$p_{o_1 \geq 4}$	$p_{o_1 \geq 5}$	o_1 の値
0	0	0	0	1
1	0	0	0	2
1	1	0	0	3
1	1	1	0	4
1	1	1	1	5

ここでモデルを CNF 式を充足する値割当てのなかで 1 に割当てられた変数の集合と定義する. すると順序符号化においては, モデルの間の包含関係が o_1 の値の大小関係と一致していることが分かる. 例えばモデル $\{p_{o_1 \geq 2}, p_{o_1 \geq 3}\}$ はモデル $\{p_{o_1 \geq 2}, p_{o_1 \geq 3}, p_{o_1 \geq 4}\}$ に包含されるが, 対応する o_1 の値は 3 と 4 であり, 包含関係と大小関係が対応することが分かる. ここでは o_1 の符号化を例に挙げたが, o_2 も同様に符号化される.

最後に, 目的変数を順序符号化し, 他の制約を任意に符号化して得られた CNF 式に対して P -極小モデルを全て列挙すればパレートフロントを得ることができる. ここで P -極小モデルとは, モデルと命題変数の集合 P の共通集合で包含関係が極小となるものである^{†17}. 本節の例では, P は目的変数を符号化して得られる以下の命題変数の集合となる.

$$P = \{p_{o_1 \geq 2}, p_{o_1 \geq 3}, p_{o_1 \geq 4}, p_{o_1 \geq 5}\} \cup$$

$$\{p_{o_2 \geq 2}, p_{o_2 \geq 3}, p_{o_2 \geq 4}, p_{o_2 \geq 5}, p_{o_2 \geq 6}, p_{o_2 \geq 7}\}$$

SAT ソルバーを使った P -極小モデルの計算の説明はここでは省略するが, インクリメンタル SAT を使った方法が論文[48]で, UNSAT サブセットを使った方法が論文[1]で提案されているので, 興味のある読者は参照されたい.

^{†17} 例えば $P = \{p, q, r\}$ でモデルが $\{p, q, r, s\}, \{p, r, s\}, \{q, s\}, \{p, r\}$ の場合, P -極小モデルは $\{p, r\}$ と $\{q\}$ となる.

ここまで説明した方法で例から得られる P -極小モデルを表 2 に示す. それぞれの P -極小モデルに対応する f_1 と f_2 の値 (o_1 と o_2 の値) は, それぞれパレート最適であることが分かる. 実際, 順序符号化を用いることで P -極小モデルとパレート最適である目的関数の値の組が一对一に対応することが証明されている [80]. また目的変数を順序符号化すれば, 他の制約は任意に符号化できるので前節で説明したハイブリッド符号化を用いることが可能である. 実際に, 先の論文 [80] ではハイブリッド符号化を用いることで, 従来手法と比較して競争的な結果を得ることに成功している.

このように, 多目的最適化問題のような一見複雑な問題も, 符号化, インクリメンタル SAT, UNSAT サブセットなどを組み合わせることで, SAT ソルバーを利用して解くことができる.

7 SAT の教科書と解説論文

SAT の代表的な教科書として 2009 年と 2015 年に出版された以下の 2 つが挙げられる.

- Handbook of Satisfiability [17]
- The Art of Computer Programming
4 巻分冊 6 [45]

日本語の解説論文は 2 つの特集号でまとめて出版されている. 1 つめは 2010 年 1 月に人工知能学会の学会誌「人工知能」25 巻 1 号で特集された「最近の SAT 技術の発展」であり以下の解説論文で構成されている. SAT と周辺分野の基本的な内容が詳しく解説されている.

- SAT ソルバーの基礎 [36]
- 高速 SAT ソルバーの原理 [65]
- 制約最適化問題と SAT 符号化 [89]
- SMT: 個別理論を取り扱う SAT 技術 [39]
- モデル列挙とモデル計数 [30]
- *—SAT: SAT の拡張 [34]
- SAT によるプランニングとスケジューリング [60]
- SAT によるシステム検証 [13]

2 つめは 2016 年 8 月に情報処理学会の学会誌「情報処理」57 巻 8 号で特集された「SAT 技術の進化と応

用 〜パズルからプログラム検証まで〜」であり以下の解説論文で構成されている. 一般の読者に向けて分かりやすく SAT と周辺分野について解説されている.

- SAT 技術の進化 [11]
- SAT とパズル
〜問題をいかに SAT ソルバーで解くか〜 [87]
- SAT とラムゼー数
〜数学の未解決問題への挑戦〜 [24]
- SAT と AI [35]
- SAT ソルバーの最近の進展 [64]
- MaxSAT: SAT の最適化問題への拡張
〜MaxSAT ソルバーの活用法〜 [47]
- SMT ソルバーによるプログラム検証 [37]

特集号ではないが, 日本ソフトウェア科学会の学会誌「コンピュータソフトウェア」でも以下の解説論文が出版されているので参照されたい.

- SAT ソルバ・SMT ソルバの技術と応用 [91]
- SAT 問題と他の制約問題との相互発展 [75]
- SAT 型制約プログラミングシステムと周辺技術 [79]

これらの解説論文の中で特に関わりが深いものについて以下に説明する. 3 節に記載した, SAT ソルバーの性能面における進化の内容は論文 [11] [64] にも詳しく説明されている. 「SAT 技術の進化」[11] では, SAT ソルバーの主要技術の説明, 2013 年から 2015 年の SAT 競技会の逐次ソルバーの結果, SAT の拡張問題, どのソルバーを使うべきかなどの内容が一般の読者向けに分かりやすく説明されている. 「SAT ソルバーの最近の進展」[64] では, 主要な逐次ソルバーの求解可能な問題群の違いや, 並列ソルバーの仕組み, 2013 年から 2015 年の SAT 競技会の並列ソルバーの結果などの内容が一般の読者向けに分かりやすく説明されている. 本稿の 3 節における主な差分は, 2016 年の SAT ソルバーおよび並列ソルバーを加えた拡張を行った SAT ソルバーの系統樹 (図 1) の説明と, 2000 年からの SAT ソルバーの性能面における進化が分かるように, Satz (2001), zchaff (2007), MiniSat (2012), MapleCOMSPS (2016) と各年代の SAT ソルバーの性能を執筆当時最新であった 2016 年の SAT 競技会のベンチマークで改めて比較実験したことな

どである (図 2). また 3.1 節では, 現在の SAT ソルバーにおける主要技術の性能への貢献度合いを示す重要な資料として図 3 に論文 [43] の発表スライド資料に掲載されたグラフの説明を追加している.

6.1 節に説明した順序符号化は, 「SAT 型制約プログラミングシステムと周辺技術」[79] では SAT 型制約プログラミングシステムにおける実装の視点から説明されているので, 興味のある読者は参考にされたい.

8 おわりに

本稿では, SAT ソルバーの最新動向として, その性能面と機能面における進化を SAT 競技会の視点から述べた. SAT 競技会は単なるソルバーの評価の場に留まらず, 積極的に逐次・並列などのソルバー種類の定義, 機能の標準化, ベンチマークの整理と追加, 若手育成のためのトラックの設立を行っている. このような取組みは, 分野への参加者を増やし結果的に 15 年以上に渡る継続的な SAT ソルバーの進化を支えてきた大きな要因になっていると考えられる.

また利用技術の視点から見ると, 性能面の進化と併せて機能面の進化が重要であることが分かる. 本稿で説明したように符号化技術とインクリメンタル SAT, MUS などの進化した機能を組合せることで, 多目的最適化のような複雑な問題を解くことが可能になる. このように応用が広がることも, ベンチマークや参加者を増やす要因になり, 今後 SAT ソルバーの進化に良い影響を与えると考えられる.

SAT ソルバーを使った問題解法は単純なソルバーまかせではないが, 問題のモデリング, SAT 符号化, SAT ソルバーの機能の活用などを検討すると, 既存解法の性能を大きく上回ることも珍しくない. 本稿が, SAT ソルバーやその利用技術について少しでも理解する一助になれば幸いである.

謝辞 本研究は JSPS 科研費 JP16K16036, JP16H0280, JP17K00300 の助成を受けたものである.

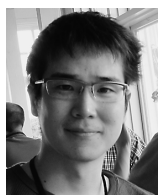
参考文献

- [1] ALVIANO, M.: Model Enumeration in Propositional Circumscription via Unsatisfiable Core Analysis, *Theory and Practice of Logic Programming*, Vol. 17, No. 5-6(2017), pp.708725.
- [2] Audemard, G., Lagniez, J., Szczepanski, N., and Tabary, S.: An Adaptive Parallel SAT Solver, *Proceedings of the 22nd International Joint Conference on Principles and Practice of Constraint Programming (CP 2016)*, LNCS 9892, 2016, pp.30–48.
- [3] Audemard, G. and Simon, L.: Predicting Learnt Clauses Quality in Modern SAT Solvers, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 2009, pp.399–404.
- [4] Audemard, G. and Simon, L.: Extreme Cases in SAT Problems, *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT 2016)*, LNCS 9710, 2016, pp.87–103.
- [5] Bacchus, F.: Enhancing Davis Putnam with Extended Binary Clause Reasoning, *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, 2002, pp.613–619.
- [6] Bacchus, F. and Winter, J.: Effective Preprocessing with Hyper-Resolution and Equality Reduction, *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, LNCS 2919, 2003, pp.341–355.
- [7] Bailleux, O. and Bouffkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP 2003)*, LNCS 2833, 2003, pp.108–122.
- [8] Bailleux, O., Bouffkhad, Y., and Roussel, O.: A Translation of Pseudo Boolean Constraints to SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, No. 1-4(2006), pp.191–200.
- [9] Balyo, T., Sanders, P., and Sinz, C.: HordeSat: A Massively Parallel Portfolio SAT Solver, 2015, pp.156–172.
- [10] Banbara, M., Matsunaka, H., Tamura, N., and Inoue, K.: Generating Combinatorial Test Cases by Efficient SAT Encodings Suitable for CDCL SAT Solvers, *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17)*, LNCS 6397, 2010, pp.112–126.
- [11] 番原睦則, 鍋島英知: SAT 技術の進化, 情報処理, Vol. 57, No. 8(2016), pp.704–709.
- [12] Banbara, M., Soh, T., Tamura, N., Inoue, K., and Schaub, T.: Answer Set Programming as a Modeling Language for Course Timetabling, *TPLP*, Vol. 13, No. 4-5(2013), pp.783–798.
- [13] 番原睦則, 田村直之: SAT によるシステム検証, 人工知能学会誌, Vol. 25, No. 1(2010), pp.122–129.
- [14] Bayardo Jr., R. J. and Schrag, R.: Using CSP

- Look-Back Techniques to Solve Real-World SAT Instances, *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 1997)*, 1997, pp. 203–208.
- [15] Berre, D. L.: Exploiting the Real Power of Unit Propagation Lookahead, *Electronic Notes in Discrete Mathematics*, Vol. 9(2001), pp. 59–80.
- [16] Biere, A.: Bounded Model Checking, *Handbook of Satisfiability*, IOS Press, 2009, pp. 457–481.
- [17] Biere, A., Heule, M., van Maaren, H., and Walsh, T.(eds.): *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications (FAIA), Vol. 185, IOS Press, 2009.
- [18] Crawford, J. M. and Baker, A. B.: Experimental Results on the Application of Satisfiability Algorithms to Scheduling Problems, *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, 1994, pp. 1092–1097.
- [19] Davis, M., Logemann, G., and Loveland, D. W.: A Machine Program for Theorem-Proving, *Communications of the ACM*, Vol. 5, No. 7(1962), pp. 394–397.
- [20] Eén, N. and Biere, A.: Effective Preprocessing in SAT Through Variable and Clause Elimination, *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT 2005)*, LNCS 3569, 2005, pp. 61–75.
- [21] Eén, N. and Sörensson, N.: An Extensible SAT-solver, *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, LNCS 2919, 2003, pp. 502–518.
- [22] Eén, N. and Sörensson, N.: Temporal Induction by Incremental SAT Solving, *Electronic Notes in Theoretical Computer Science*, Vol. 89, No. 4(2003).
- [23] Eén, N. and Sörensson, N.: Translating Pseudo-Boolean Constraints into SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, No. 1-4(2006), pp. 1–26.
- [24] 藤田博, 越村三幸: SAT とラムゼー数～数学の未解決問題への挑戦～, *情報処理*, Vol. 57, No. 8(2016), pp. 716–719.
- [25] Gebser, M., Kaufmann, B., and Schaub, T.: Conflict-driven Answer Set Solving: From Theory to Practice, *Artif. Intell.*, Vol. 187(2012), pp. 52–89.
- [26] Gent, I. P.: Arc Consistency in SAT, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002)*, 2002, pp. 121–125.
- [27] Giesl, J., Thiemann, R., Schneider-Kamp, P., and Falke, S.: Automated Termination Proofs with AProVE, *Rewriting Techniques and Applications, 15th International Conference, RTA 2004, Aachen, Germany, June 3-5, 2004, Proceedings*, 2004, pp. 210–220.
- [28] Hamadi, Y., Jabbour, S., and Sais, L.: Learning for Dynamic Subsumption, *International Journal on Artificial Intelligence Tools*, Vol. 19, No. 4(2010), pp. 511–529.
- [29] Han, H. and Somenzi, F.: On-the-Fly Clause Improvement, *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009)*, LNCS 5584, 2009, pp. 209–222.
- [30] 長谷川隆三, 藤田博, 越村三幸: モデル列挙とモデル計数, *人工知能学会誌*, Vol. 25, No. 1(2010), pp. 96–104.
- [31] Heule, M., Järvisalo, M., and Biere, A.: Clause Elimination Procedures for CNF Formulas, *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17)*, LNCS 6397, 2010, pp. 357–371.
- [32] Heule, M., Järvisalo, M., and Biere, A.: Efficient CNF Simplification Based on Binary Implication Graphs, *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011)*, LNCS 6695, 2011, pp. 201–215.
- [33] Heule, M. J. H., Kullmann, O., and Marek, V. W.: Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer, *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT 2016)*, LNCS 9710, 2016, pp. 228–245.
- [34] 平山勝敏, 横尾真: *-SAT: SAT の拡張, *人工知能学会誌*, Vol. 25, No. 1(2010), pp. 105–113.
- [35] 井上克巳: SAT と AI, *情報処理*, Vol. 57, No. 8(2016), pp. 720–723.
- [36] 井上克巳, 田村直之: SAT ソルバーの基礎, *人工知能学会誌*, Vol. 25, No. 1(2010), pp. 57–67.
- [37] 石井大輔, 上田和紀: SMT ソルバーによるプログラム検証, *情報処理*, Vol. 57, No. 8(2016), pp. 734–737.
- [38] Iwama, K. and Miyazaki, S.: SAT-Variable Complexity of Hard Combinatorial Problems, *Proceedings of the IFIP 13th World Computer Congress*, 1994, pp. 253–258.
- [39] 岩沼宏治, 鍋島英知: SMT: 個別理論を取り扱う SAT 技術, *人工知能学会誌*, Vol. 25, No. 1(2010), pp. 86–95.
- [40] Jackson, D.: *Software Abstractions – Logic, Language, and Analysis*, MIT Press, 2006.
- [41] Järvisalo, M., Biere, A., and Heule, M.: Blocked Clause Elimination, *Proceedings of the 16th International Conference of Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010)*, 2010, pp. 129–144.
- [42] Kaivola, R., Ghughal, R., Narasimhan, N., Telfer, A., Whitemore, J., Pandav, S., Slobodová, A., Taylor, C., Frolov, V. A., Reeber, E., and Naik, A.: Replacing Testing with Formal Verification in Intel Core™ i7 Processor Execution Engine Validation, *Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009)*, LNCS 5643, 2009, pp. 414–429.
- [43] Katebi, H., Sakallah, K. A., and Silva, J. P. M.: Empirical Study of the Anatomy of Modern Sat Solvers, *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011)*, LNCS 6695, 2011, pp. 343–

- 356.
- [44] Kautz, H. A. and Selman, B.: Planning as Satisfiability, *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 1992)*, 1992, pp. 359–363.
 - [45] Knuth, D. E.: *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability*, Addison-Wesley Professional, 2015.
 - [46] Konev, B. and Lisitsa, A.: A SAT Attack on the Erdős Discrepancy Conjecture, *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT 2014)*, LNCS 8561, 2014, pp. 219–226.
 - [47] 越村三幸, 藤田博: MaxSAT: SAT の最適化問題への拡張～MaxSAT ソルバーの活用法～, 情報処理, Vol. 57, No. 8(2016), pp. 730–733.
 - [48] Koshimura, M., Nabeshima, H., Fujita, H., and Hasegawa, R.: Minimal model generation with respect to an atom set, *Proceedings of the 7th International Workshop on First-Order Theorem Proving (FTP '09)*, 2009, pp. 49–59.
 - [49] Koshimura, M., Nabeshima, H., Fujita, H., and Hasegawa, R.: Solving Open Job-Shop Scheduling Problems by SAT Encoding, *IEICE Transactions*, Vol. 93-D, No. 8(2010), pp. 2316–2318.
 - [50] Lamb, E.: Two-hundred-terabyte maths proof is largest ever, *Nature*, Vol. 534(2016), pp. 17–18.
 - [51] Larrabee, T.: Test Pattern Generation using Boolean Satisfiability, *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 11, No. 1(1992), pp. 4–15.
 - [52] Le Berre, D. and Rapicault, P.: Dependency Management for the Eclipse Ecosystem: Eclipse P2, Metadata and Resolution, *Proceedings of the 1st International Workshop on Open Component Ecosystems, IWOCE '09*, 2009, pp. 21–30.
 - [53] Liang, J. H., Ganesh, V., Poupart, P., and Czarnecki, K.: Learning Rate Based Branching Heuristic for SAT Solvers, *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT 2016)*, LNCS 9710, 2016, pp. 123–140.
 - [54] Luby, M., Sinclair, A., and Zuckerman, D.: Optimal Speedup of Las Vegas Algorithms, *Information Processing Letters*, Vol. 47, No. 4(1993), pp. 173–180.
 - [55] Lynce, I. and Silva, J. P. M.: Probing-Based Preprocessing Techniques for Propositional Satisfiability, *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2003)*, 2003, pp. 105.
 - [56] Manthey, N., Lanti, D., and Irfan, A.: Modern Cooperative Parallel SAT Solving, *POS-13. Fourth Pragmatics of SAT workshop, a workshop of the SAT 2013 conference, July 7, 2013, Helsinki, Finland*, 2013, pp. 41–54.
 - [57] Marques-Silva, J. P. and Sakallah, K. A.: GRASP: A Search Algorithm for Propositional Satisfiability, *IEEE Transactions on Computers*, Vol. 48, No. 5(1999), pp. 506–521.
 - [58] Morgado, A., Heras, F., Liffiton, M. H., Planes, J., and Marques-Silva, J.: Iterative and Core-guided MaxSAT Solving: A Survey and Assessment, *Constraints*, Vol. 18, No. 4(2013), pp. 478–534.
 - [59] Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S.: Chaff: Engineering an Efficient SAT Solver, *Proceedings of the 38th Design Automation Conference (DAC 2001)*, 2001, pp. 530–535.
 - [60] 鍋島英知: SAT によるプランニングとスケジューリング, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 114–121.
 - [61] Nabeshima, H. and Inoue, K.: Coverage-Based Clause Reduction Heuristics for CDCL Solvers, *Proceedings of the 20th International Conference on Theory and Applications of Satisfiability Testing (SAT 2017)*, LNCS 10491, 2017, pp. 136–144.
 - [62] 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(2012), pp. 211–230.
 - [63] Nabeshima, H., Iwanuma, K., and Inoue, K.: On-the-Fly Lazy Clause Simplification Based on Binary Resolvents, *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, IEEE, 2013, pp. 987–995.
 - [64] 鍋島英知, 岩沼宏治, 井上克巳: SAT ソルバーの最近の進展, 情報処理, Vol. 57, No. 8(2016), pp. 724–729.
 - [65] 鍋島英知, 宋剛秀: 高速 SAT ソルバーの原理, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 68–76.
 - [66] Nabeshima, H., Soh, T., Inoue, K., and Iwanuma, K.: Lemma Reusing for SAT based Planning and Scheduling, *Proceedings of the International Conference on Automated Planning and Scheduling 2006 (ICAPS 2006)*, 2006, pp. 103–112.
 - [67] Niemelä, I.: A Tableau Calculus for Minimal Model Reasoning, *Proceedings of the 5th International Workshop on Theorem Proving with Analytic Tableaux and Related Methods (TABLEAUX 1996)*, LNCS 1071, 1996, pp. 278–294.
 - [68] 則武治樹, 番原睦則, 宋剛秀, 田村直之, 井上克巳: パッキング配列問題の制約モデリングと SAT 符号化, コンピュータソフトウェア, Vol. 31, No. 1(2013), pp. 116–130.
 - [69] Ogata, S., Tsuchiya, T., and Kikuno, T.: SAT-Based Verification of Safe Petri Nets, *Proceedings of Automated Technology for Verification and Analysis: Second International Conference (ATVA 2004)*, 2004, pp. 79–92.
 - [70] Ogawa, T., Liu, Y., Hasegawa, R., Koshimura, M., and Fujita, H.: Modulo Based CNF Encoding of Cardinality Constraints and Its Application to MaxSAT Solvers, *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, IEEE, 2013, pp. 9–17.
 - [71] Petke, J. and Jeavons, P.: The Order Encoding: From Tractable CSP to Tractable SAT, *Proceedings*

- of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011), LNCS 6695, 2011, pp. 371–372.
- [72] Philipp, T. and Steinke, P.: PBLib – A Library for Encoding Pseudo-Boolean Constraints into CNF, *Theory and Applications of Satisfiability Testing – SAT 2015*, Heule, M. and Weaver, S.(eds.), Lecture Notes in Computer Science, Vol. 9340, Springer International Publishing, 2015, pp. 9–16.
- [73] Pipatsrisawat, K. and Darwiche, A.: A Lightweight Component Caching Scheme for Satisfiability Solvers, *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT 2007)*, LNCS 4501, 2007, pp. 294–299.
- [74] Sakai, M. and Nabeshima, H.: Construction of an ROBDD for a PB-Constraint in Band Form and Related Techniques for PB-Solvers, *IEICE Transactions*, Vol. 98-D, No. 6(2015), pp. 1121–1127.
- [75] 酒井政裕, 今井健男: SAT 問題と他の制約問題との相互発展, コンピュータソフトウェア, Vol. 32, No. 1(2015), pp. 103–119.
- [76] Selman, B., Kautz, H., and Cohen, B.: Local Search Strategies for Satisfiability Testing, *Cliques, Coloring, and Satisfiability: the Second DIMACS Implementation Challenge*, Johnson, D. J. and Trick, M. A.(eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, 1996, pp. 521–532.
- [77] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP 2005)*, LNCS 3709, 2005, pp. 827–831.
- [78] Soh, T., Banbara, M., and Tamura, N.: Proposal and Evaluation of Hybrid Encoding of CSP to SAT Integrating Order and Log Encodings, *International Journal on Artificial Intelligence Tools*, Vol. 26, No. 1(2017), pp. 1–29.
- [79] 宋剛秀, 番原睦則, 田村直之: SAT 型制約プログラミングシステムと周辺技術, コンピュータソフトウェア, Vol. 34, No. 1(2017), pp. 67–80.
- [80] Soh, T., Banbara, M., Tamura, N., and Berre, D. L.: Solving Multiobjective Discrete Optimization Problems with Propositional Minimal Model Generation, *Proceedings of the 23rd International Joint Conference on Principles and Practice of Constraint Programming (CP 2017)*, LNCS 10416, 2017, pp. 596–614.
- [81] Soh, T., Berre, D. L., Roussel, S., Banbara, M., and Tamura, N.: Incremental SAT-Based Method with Native Boolean Cardinality Handling for the Hamiltonian Cycle Problem, *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA 2014)*, LNAI 8761, 2014, pp. 684–693.
- [82] Soh, T. and Inoue, K.: Identifying Necessary Reactions in Metabolic Pathways by Minimal Model Generation, *Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010)*, 2010, pp. 277–282.
- [83] Soh, T., Inoue, K., Tamura, N., Banbara, M., and Nabeshima, H.: A SAT-based Method for Solving the Two-dimensional Strip Packing Problem, *Fundamenta Informaticae*, Vol. 102, No. 3-4(2010), pp. 467–487.
- [84] Subbarayan, S. and Pradhan, D. K.: NiVER: Non Increasing Variable Elimination Resolution for Preprocessing SAT instances, *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, LNCS 3542, 2004.
- [85] Tamura, N. and Banbara, M.: Sugar: a CSP to SAT Translator Based on Order Encoding, *Proceedings of the 2nd International CSP Solver Competition*, 2008, pp. 65–69.
- [86] Tamura, N., Banbara, M., and Soh, T.: PB-Sugar: Compiling Pseudo-Boolean Constraints to SAT with Order Encoding, *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, IEEE, November 2013, pp. 1020–1027.
- [87] 田村直之, 宋剛秀, 番原睦則: SAT とパズル～問題をいかに SAT ソルバーで解くか～, 情報処理, Vol. 57, No. 8(2016), pp. 710–715.
- [88] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Constraints*, Vol. 14, No. 2(2009), pp. 254–272.
- [89] 田村直之, 丹生智也, 番原睦則: 制約最適化問題と SAT 符号化, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 77–85.
- [90] 丹生智也, 田村直之, 番原睦則: 位取り記数法に基づく整数有限領域上の制約充足問題のコンパクトかつ効率的な SAT 符号化, コンピュータソフトウェア, Vol. 30, No. 1(2013), pp. 211–230.
- [91] 梅村晃広: SAT ソルバ・SMT ソルバの技術と応用, コンピュータソフトウェア, Vol. 27, No. 3(2010), pp. 24–35.
- [92] Walsh, T.: SAT v CSP, *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP 2000)*, 2000, pp. 441–456.
- [93] Warners, J. P.: A Linear-Time Transformation of Linear Inequalities into Conjunctive Normal Form, *Inf. Process. Lett.*, Vol. 68, No. 2(1998), pp. 63–69.
- [94] Wetzler, N., Heule, M., and Jr., W. A. H.: DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs, *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT 2014)*, LNCS 8561, 2014, pp. 422–429.

**宋 剛秀**

2004 年神戸大学工学部電気電子工学科卒業。2006 年同大学大学院自然科学研究科電気電子工学専攻修了。サントリー (株) を経て、2011 年総合研究大学院大学複合科学研究科情報学専攻博士後期課程修了。博士 (情報学)。2012 年より神戸大学情報基盤センター助教。制約プログラミング, SAT 技術およびそれらの応用に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。

**番原睦則**

1994 年神戸大学理学部数学科卒業。1996 年同大学大学院自然科学研究科博士課程前期数学専攻修了。1996 年国立奈良工業高等専門学校助手。1998 年同校講師。2003 年より神戸大学情報基盤センター講師。2007 年同准教授。博士 (工学)。論理プログラミング, 制約プログラミング, SAT 技術等に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。

**田村直之**

1980 年神戸大学理学部物理学科卒業。1985 年同大学大学院自然科学研究科博士課程システム科学専攻修了。学術博士。1985 年日本 IBM 東京基礎研究所入社。1988 年神戸大学工学部勤務。2003 年より神戸大学情報基盤センター教授。論理プログラミング, 制約プログラミング, SAT 技術等に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。

**鍋島英知**

2001 年神戸大学大学院・自然科学研究科情報メディア科学専攻博士課程修了。同年山梨大学工学部コンピュータ・メディア工学科助手。現在, 同大学大学院・医学工学総合研究部准教授。博士 (工学)。SAT, プランニング, 定理自動証明, WEB 知的処理等の研究に従事。2004 年 FIT 優秀論文賞受賞, 2012 年日本ソフトウェア科学会高橋奨励賞受賞, 2014 年日本ソフトウェア科学会ソフトウェア論文賞受賞。