特 集

SAT ソルバーの基礎

Foundations of SAT Solvers

井上 克巳
Katsumi Inoue
国立情報学研究所 / 総合研究大学院大学情報学専攻
National Institute of Informatics / Department of Informatics, The Graduate University for Advanced Studies
ki@nii.ac.jp

田村 直之
Naoyuki Tamura
神戸大学 学術情報基盤センター
Information Science and Technology Center, Kobe University
tamura@kobe-u.ac.jp

keywords: Boolean satisfiability, SAT solvers, DPLL, incomplete solvers, SAT competition.

1. はじめに

ブール式または命題論理式の充足可能性判定 (Boolean (or propositional) satisfiability (testing); **SAT**) は，最初に NP 完全性が証明された問題であり [Cook 71]，計算理論にとって中心的である [Garey 79]．SAT はまた，論理合成，ハードウェア・ソフトウェアの検証，制約充足問題，プランニング，自動推論・定理証明など，計算機工学や人工知能における数多くの問題において基本となる．このため，SAT を解くアルゴリズムおよびそれを実装したソルバーに関して，半世紀前からこれまでに膨大な量の研究がなされてきた．こうした多くの努力が結実し，この十数年で SAT ソルバーの性能は飛躍的に向上した．

一般に SAT ソルバーの実行時間は最悪で指数時間オーダーかかるが，実際の応用問題においては平均的によい性能を示すことがある．例えば，SAT の有効性がハードウェア検証において認められており [Biere 99]，10^6 以上の変数を含む大規模な検証問題が現実的な時間で解けることから，SAT ソルバーがこの分野では日常的に使われるようになっている．また，“International Conference on Theory and Applications of Satisfiability Testing” (SAT 国際会議) が毎年開催されており，その中で SAT 競技会も開催され SAT ソルバーの実行時間を競い合うようになっている．これは，あたかも SAT の計算量の壁をプログラミング技術で乗り越えようとしているようでもあり，SAT ソルバーに関するインプリメンテーション技術は工学であると同時に，工芸の域にまで達している．

本稿では，SAT 問題を解くための代表的なソルバーについて解説し，SAT の研究分野全体の概観も行う．とくに，1960〜62 年に考案された DPLL アルゴリズムに基づく系統的ソルバー，1990 年代に発展した局所探索に基づく確率的ソルバーの基本技術について述べる．なお SAT に関する研究についてより深く知りたければ，2009 年に出版された “Handbook of Satisfiability” [Biere 09] に，これまでの SAT 技術について幅広く説明されているので参照していただきたい．また，SAT ソルバーに関する最近の解説・レビューには [Gomes 08, Malik 09] などもある．

2. SAT に関する諸定義

まず，**(ブール) 変数** (Boolean variables) の集合 V を考える．V に含まれる要素は**命題変数**または **(命題) アトム** (propositional atom) とも呼ばれる．V の各命題変数は，その値として 1 または 0 を取り，これらの値はそれぞれ**真** ($true$) または**偽** ($false$) に対応する．V に対する **(部分)(真偽値) 割り当て** ((partial) truth assignment) は (部分) 関数 $I : V \to \{1, 0\}$ で与えられる．**ブール式** (Boolean formula) または **(命題) 論理式** (propositional formula) は，V の要素に対して論理結合子 (operator) である**否定** (NOT, $\neg$)，**論理積** (AND, $\wedge$)，**論理和** (OR, $\vee$) を再帰的に適用して構成される．

V に対する真偽値割り当て I と命題論理式 ψ が与えられたとき，ψ の真偽値を次のように再帰的に定義する．ここで，I により ψ に $true$ が割り当てられるとき，I は ψ を**充足する** (satisfy) といい，$I \models \psi$ と表す．また，$I \models \psi$ が成り立たないことを $I \not\models \psi$ と表す．以下，ψ, ψ_1, ψ_2 を任意の論理式とする．

- 命題アトム x に対して，$I \models x \Leftrightarrow I(x) = 1$.
- $I \models \neg\psi \Leftrightarrow I \not\models \psi$.
- $I \models \psi_1 \vee \psi_2 \Leftrightarrow I \models \psi_1$ または $I \models \psi_2$.
- $I \models \psi_1 \wedge \psi_2 \Leftrightarrow I \models \psi_1$ かつ $I \models \psi_2$.

割り当て I が ψ を充足するとき，I は ψ の**モデル** (model) であるという．命題論理式 ψ が**充足可能** (satisfiable) であるとは，ψ を充足する割り当て，すなわち ψ のモデルが存在することを意味する．ψ を充足する割り当てが存在しないとき，ψ は**充足不能** (unsatisfiable) であるという．例えば，$x \wedge \neg x$ は充足不能な式である．

命題論理式の充足可能性判定 (SAT) は，命題論理式が

充足可能であるかどうかを決定する問題である．SAT の各インスタンスを**SAT 問題** (SAT problem (instance)) と呼び，SAT 問題を解く手続き (またはプログラム) を**SAT ソルバー** (SAT solver) と呼ぶ．SAT ソルバーの役割は，与えられた SAT 問題が充足可能である (SAT) か充足不能である (UNSAT) かの判定を行うことにあるが，多くの SAT ソルバーでは，充足可能であればモデルを発見し，充足不能であればモデルが存在しないことを証明するという構成的な手続きとなっている．

SAT においては，以下で定義する**連言標準形** (conjunctive normal form; **CNF**) で表される命題論理式 (CNF 式) を考えることが一般的である．いま，変数 x またはその否定 $\neg x$ を**リテラル** (literal) といい，前者を**正リテラル** (positive literal)，後者を**負リテラル** (negative literal) と呼ぶ．リテラル l に対して，$\overline{l}$ により，l の**補リテラル** (complement) を表す．すなわち，x がアトムのとき，$\overline{x} = \neg x$ であり，$\overline{\neg x} = x$ である．リテラルの選言 (disjunction)，すなわち，リテラルを論理和 $\vee$ で結合した論理式を**節** (clause) という．節 C を構成するリテラルの個数を C の**長さ** (length) または**サイズ** (size) という．長さ 0 の節を**空節** (empty clause) と呼び $\square$ で表す．空節は $false$ と同一視され，どんな割り当てにも充足されない．また，長さ 1 の節を**単位節** (unit clause) と呼ぶ．節の連言 (conjunction)，すなわち，節を論理積 $\wedge$ で結合した論理式を**CNF 式** (CNF formula) という．CNF 式は式を構成する節の集合と同一視され，**節理論** (clausal theory) または単に**節集合**とも呼ばれる．これらの定義より，CNF 式または節集合が充足可能であるためには，それを構成するすべての節が真となるような割り当てが存在しなければならず，各節が真となるためにはその節に含まれるリテラルの少なくとも一つが真とならなければならない．

〔例 1〕 CNF 式 ψ が次で与えられるとする:

$$(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (x_3 \vee x_4) \wedge$$
$$(\neg x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4)$$

このとき，割り当て $(x_1, x_2, x_3, x_4) = (1,1,1,0)$ は ψ を充足し，よって ψ は充足可能である (SAT). 他の ψ のモデルには $(1,0,0,1)$ と $(1,0,1,0)$ が存在する．一方，CNF 式 $\psi' = \psi \wedge (\neg x_1)$ は充足不能である (UNSAT). なぜなら，ψ の任意のモデル I において $I(x_1) = 1$ であり，$I \not\models \neg x_1$ であるため，$I \not\models \psi'$ となるからである．

任意の命題論理式は，論理的同値である CNF 式に指数時間オーダーで変換でき，さらに充足可能性が同値であるような CNF 式には線形時間で変換可能である (詳細は [田村 10] を参照のこと)．このことから SAT 問題を CNF 式に限定しても一般性を失うことはない．さらに CNF 式への制限は，SAT を解くアルゴリズムの単純化を可能とする[*1]ことから，SAT ソルバーの設計にとっても本質的である．ちなみに，ベンチマーク問題における SAT ソルバーへの CNF 式の入力は **DIMACS CNF 形式**[*2] と呼ばれる整数列で簡潔に記述されることが多い．

Cook は，SAT が非決定性チューリングマシンを用いて多項式時間で解ける (クラス NP に属する) ことと，クラス NP に属するすべての問題が SAT 問題に多項式時間で還元できることを示し，SAT の計算量が **NP 完全**であることを証明した [Cook 71]．これより，P=NP でない限り，すべての SAT 問題を入力サイズの多項式時間で解けるようなアルゴリズムは存在しない．また CNF 式に対する SAT において，すべての節の長さが k で与えられる問題は k-**SAT** と呼ばれる．$k \geq 3$ において k-SAT は NP 完全であり，2-SAT は多項式時間で解ける [Cook 71]．とくに，**3-SAT** は代表的な NP 完全問題の一つであり，多くの問題で NP 困難性を示すのに使われている [Garey 79].

3. 系統的 SAT ソルバー

SAT は典型的な組合せ問題であり，SAT を解くための多くのアルゴリズムとソルバーが開発されてきた．これらの SAT ソルバーを大別すると，**系統的ソルバー** (systematic solver) と，**確率的ソルバー** (stochastic solver) の二種類が存在する．系統的ソルバーは，**系統的探索** (systematic search) を行う**完全な** (complete) アルゴリズムに基づいており，SAT 問題の充足可能性と充足不能性のいずれも判定可能である．SAT 問題が充足不能であるとき，系統的ソルバーはすべての可能な真偽値割り当てに相当する探索を行った後に UNSAT と判定して終了する．これに対し確率的ソルバーは，**確率的局所探索** (stochastic local search; **SLS**) を行う**不完全な** (incomplete) アルゴリズムに基づいており，充足可能性は判定可能であるが充足不能性は一般に判定できない．

以下，本章で系統的ソルバーについて解説し，次章で確率的ソルバーを扱う．

3·1 DPLL

系統的 SAT ソルバーの基礎となっているアルゴリズムは，1960〜62 年に Davis らにより開発された **DPLL** アルゴリズムである．DPLL は，Davis と Putnum による最初の手続き [Davis 60] に対して，Davis, Logemann, Loveland が改良を加えたもの [Davis 62][*3]を指している．DPLL はシンプルではあるが，SAT のための決定手続きとして長年もっとも使われてきたのみならず，現在の高速ソルバーでも最適化が施されて使われ続けており，脅威

*1 否定標準形 (negation normal form; NNF) の論理式を入力とする高速 SAT ソルバーも開発されている [Jain 09].

*2 ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/

*3 DPLL は，Davis-Putnam-Logemann-Loveland の頭文字を取ったものであるが，[Davis 62] の著者だけの頭文字を取って DLL とも呼ばれることがある．なお以前のテキストや文献で，“Davis-Putnum 手続き” と呼ばれていたものも [Davis 60] ではなく [Davis 62] のアルゴリズムを指すことが多いが (例えば [Chang 71])，最近では DPLL または DLL と呼ぶ方が一般的である．

的な長寿性を誇っている．DPLL は完全なアルゴリズムであり，系統的な二分探索 (binary search) を行う．一方，最初の SAT アルゴリズムである **DP** [Davis 60] は，**融合** (導出，resolution) 操作を繰り返し適用して空節を導くことで充足不能性を示すものである．DPLL は DP における融合を分割規則 (後述) に置き換えただけのものであるが，この変更が後年になって予想以上の効果[*4]をもたらすことになる．なお，DP については，Robinson により一階述語節理論における融合原理に拡張され [Robinson 65]，その後の定理証明研究につながっていく [Chang 71]．

以下，DPLL アルゴリズムを図 1 に沿って説明する[*5]．DPLL は，CNF 式 ψ を入力として受け取る．このとき ψ が空であれば，SAT が返される．さもなければ，最初に DPLL は ψ に出現するすべての単位節に含まれる変数に対し，それらを真にするための適切な割り当てを行う．すなわち，正リテラル x からなる単位節に対しては $I(x) = 1$ とし，負リテラル $\neg y$ からなる単位節に対しては $I(y) = 0$ とする．こうしていくつかの変数に値が割り当てられると，DPLL はこの部分割り当てを使って**単位伝播** (unit propagation) と呼ばれる ψ の簡単化を行う．これは例えば $I(x) = 1$ と割り当てられたとすると，リテラル x を含むすべての節は I によって充足されるためこれらの節を消去し，残った中で $\neg x$ を含む節では $I(\neg x) = 0$ なので $\neg x$ を除去する．同様に，y に 0 が割り当てられた場合，$\neg y$ を含む節をすべて消去し，残った節から y をすべて除去する．ここでもし簡単化された ψ が空節 $\square$ を含めば，UNSAT が返される．単位節が無くなった時点で，DPLL は適当な**変数選択ヒューリスティックス**に基づいて，ψ 中で未割り当ての変数 x を一つ選択し，x に 1 (または 0) を割り当てた場合を再帰的に試みる (**分割規則**，splitting rule)．x に 1 を割り当てる場合，$\psi \wedge x$ に対する DPLL が SAT を返せば，ψ のすべての節は充足されているため元の DPLL も SAT を返す．さもなければ x に 0 を割り当てる場合を調べ，$\psi \wedge \neg x$ に対する DPLL の結果を返す．

単位伝播の手続き (図 2) では，現在の部分割り当てを使って ψ を充足するために可能な割り当てをすべて行う．この結果，もしある節が未割り当ての変数 x をただ一つだけ含んでおり，他のすべての変数がすでに 0 と割り当てられたとすると，必然的に x に対する割り当ては一意に決定される．こうして新たな単位節が生まれれば単位伝播を繰り返し，この作業を単位節がなくなるか，**矛盾** (conflict) が検出されるまで繰り返す．ここで矛盾は，伝播によって同じ変数に同時に 1 と 0 が割り当てられなければならない状況において発生する．矛盾が発生すると，DPLL アルゴリズム上では空節 $\square$ を生成して UNSAT を返すことになるが，これを再帰的に呼び出していたのは分割規則を適用した時点である．実際，選択された変数 x に対して分割規則により最初に値を割り当てた時点では，真と偽のいずれの値を割り当てるのが正しいかは不明であったため，この時点まで戻って (すなわち，**バックトラック**して) x の値を変更する．このように，DPLL は再帰呼び出しの中で，単位節伝搬によって真偽値が確定しない変数に対しては真偽値割り当ての組合せを網羅的に試みることになる．その結果，もし充足可能な割り当てが見つからなければ最終的に UNSAT を返す．

DPLL(CNF ψ)
begin
  if ψ is empty **then** return SAT;
  ψ := **UnitPropagation**(ψ);
  if $\square$ exists in ψ **then** return UNSAT;
  Choose variable x in ψ by some hueristic;
  if DPLL($\psi \wedge x$) returns SAT **then** return SAT;
  else return the result of **DPLL**($\psi \wedge \overline{x}$)
end

図 1 DPLL algorithm

UnitPropagation(CNF ψ)
begin
  while $\square$ doesn't exist and a unit clause l exists in ψ **do**
    Assign 1 to l and simplify ψ;
  return ψ
end

図 2 Unit propagation

〔例 2〕 CNF 式 ψ が次で与えられているとする．

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2) \wedge (x_1 \vee x_2) \\ \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_3).$$

最初に DPLL は単位節 x_2 に 1 を割り当てる．単位伝播により，ψ は次式に簡単化される:

$$\psi' : \quad (\neg x_1 \vee \neg x_3) \wedge (x_1 \vee x_3) \wedge (\neg x_1 \vee x_3).$$

この時点で単位節は存在せず，変数 x_1 を選択し 1 を割り当てたとする．これは $\mathbf{DPLL}(\psi' \wedge x_1)$ において，単位節 $\neg x_3$ と単位節 x_3 を導くために矛盾が生じる．よってバックトラックして，x_1 に 0 を割り当てると，単位節 x_3 を導き，これに 1 を割り当てることですべての節が充足される．こうして $\mathbf{DPLL}(\psi' \wedge \neg x_1)$ が SAT となるため，元の DPLL も SAT となって終了する．

以上の DPLL は系統的 SAT アルゴリズムの基本であるが，SAT ソルバーを実装する上ではいくつか選択の余地

---

*4 CNF 式 ψ が $\psi = \psi' \wedge (x \vee C_1) \wedge (\neg x \vee C_2)$ (ψ' は CNF 式，x は変数，C_1, C_2 は節) と書けるとすると，DP が ψ を $\psi' \wedge (C_1 \vee C_2)$ に置き換えるのに対し，DPLL では $\psi' \wedge x$ と $\psi' \wedge \neg x$ の 2 通りに分割してその結果の OR を取る．ψ' を簡単化する際に，$C_1 \vee C_2$ で包摂される節を取り除くよりも，単位節 x または $\neg x$ からの伝播を行なう方がはるかに効果が大きい．

*5 ここでは DPLL を SAT 問題を解くための探索手続きと捉えている．これに対し，[Nieuwenhuis 06] では DPLL を抽象化した推論システムとして論理的に再定義しており，これにより他の論理体系との融合を容易にしている．この点に関しては，本特集の解説 [岩沼 10] を参照されたい．

があり，分離規則における変数選択ヒューリスティックスもその一つである．よく知られたものに，**MOMS ヒューリスティックス**があり，これは最小サイズの節集合に最大個数出現している変数を次に選択する．系統的ソルバーの一つである Satz では，MOMS ヒューリスティックスに加えて，変数 x に対して単位節 x と単位節 $\neg x$ を ψ にそれぞれ加えて 2 つの単位伝播を独立に行なった結果に基づき比較するという **UP ヒューリスティックス** (unit propagation heuristics) を改良して使い，単位伝播の効果がより発揮できる工夫を施している [Li 97]．なお，Chaff [Moskewicz 01] など近年の最適化されたソルバーでは，独自に改良したヒューリスティックスが使われている．

3·2 系統的ソルバーの効率化

系統的 SAT ソルバーは，Chaff [Moskewicz 01] の出現によって，高速化が一段と推進された．Chaff は DPLL に対して，**監視リテラル** (watched literals) と**矛盾解析** (conflict analysis) という技術を使って効率化している．

単位伝播は SAT 問題を解く実行時間の中で最も多くの時間を消費していることから，監視リテラルの概念が単位伝播を効率化するための技術として Chaff に導入された [Moskewicz 01]．この方法では，節中で 0 が割り当てられていない任意の 2 つのリテラルを監視しておく．もし監視リテラルの一つが 0 に割り当てられれば，別の未割り当てリテラルがあればそのリテラルを新たに監視し，無ければ節を単位伝播の対象とする．監視リテラルが 2 つとも未割り当てのままだと伝播の対象にならず何もしない．この方式により，ある変数 x に真偽値が新たに割り当てられるたびに x を持つすべての節の状態を更新するという手間が不要になり，x を監視している節のみを更新すればよい．さらに，バックトラッキングの際に，各節の監視リテラルを更新する必要もない．

Chaff における矛盾解析 [Moskewicz 01, Zhang 01] は，Relsat [Bayardo 97] や GRASP [Marques-Silva 99] における矛盾からの**節学習** (clause learning) を進化させたもので，現在の割り当てが矛盾を引き起こしたときに，この矛盾に関係している直接の原因を解析し，そのコアとなる部分割り当てを節 (**補題** (lemma) または **nogood** ともいう)[*6]の形で記録する．例えば，$(x_1, x_2, x_3) = (1, 1, 0)$ が矛盾の原因であるとき，節 $\neg(x_1 \wedge x_2 \wedge \neg x_3) = \neg x_1 \vee \neg x_2 \vee x_3$ が生成される．このような学習節は新たな制約としてはたらくため，以後同じ矛盾の発生を防ぐ効果が期待される．ただし，学習節の追加によるオーバーヘッドもあるため，矛盾が起きるたびに学習節を毎回追加することは避けるべきで，**バックジャンプ法** (backjumping) と呼ばれるバックトラックポイントの選択と合わせて，実現方法には工夫を要する [Zhang 01]．

このように，矛盾解析と監視リテラルの利用は，Chaff を始めとする近年の SAT ソルバーが基本 DPLL アルゴリズムと比べて格段に高速化された最大の要因ではあるが，他にも多くの最適化手法が存在している．たとえば，変数選択ヒューリスティックスや変数の選択順序による影響を緩和するための**ランダム・リスタート戦略** (random restarts) [Gomes 98] などの技術も高速化には欠かせない．Eén と Sörensson により開発された MiniSat [Eén 03a] は，これらの高速化技術を効率良く実装することで，劇的な性能向上と普及に貢献した代表的な SAT ソルバーである．これらの高速 SAT ソルバーの技術については [鍋島 10b] で詳細が述べられている．

4. 確率的 SAT ソルバー

系統的 SAT ソルバーとは異なり，確率的 SAT ソルバーは不完全な確率的局所探索 (SLS) に基づいているため充足不能性の判定ができず，与えられた制限時間内で解 (モデル) が発見できなければ失敗して終了する．このため，確率的ソルバーの主な使用目的は，充足可能な場合に解を発見することにある[*7]．

確率的 SAT ソルバーにおける SLS は，**MaxSAT** 問題の近似解法としてもそのまま利用できる．MaxSAT は，与えられた CNF 式に含まれる節の数を最大に充足する割り当てを求めるものであり，SLS における局所解 (local minima) を近似解とすることができる．なお，MaxSAT 問題の厳密解を求める手法は [平山 10] を参照のこと．

4·1 Walksat

確率的 SAT ソルバーは，GSAT [Selman 92] の提案以来 AI 分野で研究が進み，これを改良した Walksat [Selman 96a] が代表的である．GSAT は，多くの最適化問題に適用されてきた SLS が，CNF 式のモデルを見つけるためにも有効であることを示した．Walksat は，**混合ランダムウォーク戦略** (mixed random walk strategy) を使って局所解から脱出することで，GSAT を改良している．

Walksat はまず始めに，すべての変数に対しランダムに真偽値を割り当てる．もしこの割り当てにより，与えられた命題論理式 ψ が充足されれば，SAT が返される．さもなければ，次の混合ランダムウォーク戦略にしたがう (ここで確率パラメータ p は，$0.5 \sim 0.6$ がよいという報告がある):

- 確率 p で，充足されていない節の中に出現する変数を一つ取り出す;
- 確率 $1 - p$ で，最適な局所移動を行う (これは GSAT と同様)．すなわち，ψ 中の変数の中で，その値を反

∗6 すべての学習節は与えられた入力節集合の論理的帰結であるため，これらを加えても SAT の結果は保存される．補題は定理証明で，nogood は制約充足で使われてきた用語であり，探索木の枝刈りのために同様に用いられる．

∗7 SAT に対する不完全な解法として，SLS 以外にも線形計画法や遺伝的アルゴリズムに基づくものもある．

```
Walksat(CNF ψ, Maxtries, Maxflips)
begin
  for i := 1 to Maxtries do
   Assign all variables randomly; Let A be the assignment;
    for j := 1 to Maxflips do
    if all clauses are satisfied by A then return SAT;
     Choose variable x by Mixed Random Walk;
      Update A by filpping x;
  return UNKNOWN
end
```

図 3 Walksat algorithm

転することにより，充足されていない節の数が最も多く減るようなものを選ぶ．

ここで選択された変数は，いずれもその真偽値が**反転され** (flipped)，これを取り出してきた節を新たに充足するが，その代わりこれまで充足されていた節が一般には充足されなくなる．こうして値の反転は，ψ のモデルが見つかるまで，固定回数繰り返し行なわれる．この方法で解を発見できなければ，ランダム初期割り当てをやり直し，以上の過程を固定回数繰り返す．もしすべての試行が失敗し解に到達しなければ，UNKNOWN が返される．

図 3 に Walksat のアルゴリズムを示す．ここで，パラメータ “Maxtries” はランダム割り当て回数の上限，また “Maxflips” は変数の反転回数の上限として与えられる．

〔例 3〕 節集合が $\psi = \{C_1, C_2, C_3, C_4, C_5\}$ であるとし，各節が次で与えられているとする:

$$C_1 = (x_1 \vee x_2 \vee x_3), \quad C_2 = (\neg x_1 \vee x_4 \vee x_5),$$
$$C_3 = (\neg x_3 \vee \neg x_4 \vee \neg x_5), \quad C_4 = (\neg x_1 \vee \neg x_3),$$
$$C_5 = (\neg x_3 \vee \neg x_5).$$

まず，Walksat がランダムに割り当て $(x_1, x_2, x_3, x_4, x_5) = (1,0,1,0,1)$ を与えたとすると，C_4, C_5 が充足されない．そこで確率 p で，これらの節の中からランダムに x_5 を選んで反転させた割り当て $(x_1, x_2, x_3, x_4, x_5) = (1,0,1,0,0)$ を考えると，C_2, C_4 が充足されない．今度は確率 $1-p$ で，充足されない節数が最も減るように変数 x_1 を選んで反転させると，$(x_1, x_2, x_3, x_4, x_5) = (0,0,1,0,0)$ はすべての節を充足する．よって，Walksat は SAT を返す．

4·2 ランダム k-SAT と相転移

ランダム k-SAT は，与えられた変数集合 V と各要素の否定から，ランダムに長さ k の節集合を生成して (同一リテラルや補リテラルを含む節は除く) 作られる．この種の問題に対して系統的ソルバーを用いた場合，実応用問題から作られる SAT 問題の場合と比べて，充足可能な問題については DPLL の性能が経験的に悪く[*8]，確率的ソルバーの研究が進むきっかけにもなった．

ランダム k-SAT に関しては，**相転移** (phase transition) という実に興味深い現象が生じる [Selman 96b, Crawford 96]．ランダム 3-SAT 問題の生成に際して，変数の個数 n を固定したまま節数 m を変化させると，物理現象の状態変化 (水が沸点で水蒸気に変化するなど) のごとく，解の存在確率がある点で劇的に変化する．具体的には，十分大きな n に対し，相転移は $m/n = 4.26$ 付近で起こり，小さな n では m/n がやや大きい値で起こる．また，n を増加させるにつれ，相転移付近での DPLL の呼び出し回数のピークはより先鋭になっていき，SAT 問題がこの付近で劇的に難しくなる．しかも，n を固定したときに，小さな m に対してはほとんどの SAT 問題が SAT となり，大きな m ではほぼ UNSAT となる．これらの比 $\alpha = m/n$ が 4.26 となる相転移付近の問題では解の存在確率がちょうど 50%くらいになって解き難いことを特徴付けている．直感的には，$\alpha = 4.26$ という点は，制約が弱いためにモデルが存在する問題領域と制約が強過ぎてモデルが存在しない問題領域の境界を特徴付けており，問題の難易度は easy-hard-easy という変化パターンになる．

ランダム 3-SAT における相転移現象により，解くことが非常に難しいランダム SAT 問題を作るために相転移における α の値を参照することができ，SAT ソルバーのベンチマーク問題の作成においても利用されている．

4·3 サーベイ伝播

2002 年に Mézard らは，これまでとはまったく異なる視点から**サーベイ伝播** (survey propagation, **SP**) と呼ばれる局所探索アルゴリズムを提案した [Mézard 02]．ここでは，与えられた論理式 ψ に対し，ψ に出現する変数と節 (の ID) をノード集合とし，節 ID とその節に含まれる変数間をリテラルの正負に対応した 2 種類のアークで結んだ**因子グラフ** (factor graph) を考える．SP は統計物理におけるスピングラスの研究から産み出されたもので，変数の真偽値がスピン ± 1 に対応し，論理式に対応する因子グラフをスピングラス系とみなしたときに，節が充足されることがそのエネルギーがゼロになることに対応づけられる．ψ のモデルを求めるために，因子グラフのノード間で「サーベイ」と呼ばれるメッセージを伝播する．このメッセージ伝播はモデル集合に相当する周辺分布を更新する形で局所的に行なわれ，確率値が最大の変数に値を割り当てて式を簡単化するという操作を繰り返す．DPLL と異なり，この変数割り当て過程ではほとんどバックトラックが起こらない．このため，SP は 10^6 変数以上のランダム 3-SAT 問題のうち相転移付近の最も難しい問題でもほぼ線形時間で解くことができる．

SP は AI における確率推論で提案された**信念伝播** (belief propagation [Pearl 88], **BP**, 確率伝播ともいう) との関係が指摘されている [Braunstein 05]．これまでのところ，サーベイ伝播の適用はランダム SAT 問題に限定されており，現実的な応用問題への適用は今後の課題である．こ

*8 理論的にも，ランダム k-SAT を解く最悪計算量については，確率的ソルバーの方が系統的ソルバーよりもよい結果が得られている．この点に関する解説は少し前であるが [岩間 01] に詳しく，最近の結果については [Dantsin 09] を参照されたい．

の分野は，物理学と計算機科学が密接に関係しており，解明されていないことも多いため今後の発展が期待される．

4·4 系統的ソルバーとの併用

確率的ソルバーと系統的ソルバーは，それぞれの設計思想から異なっているため，両者の特徴を併せもつソルバーの開発は容易ではない．これまでに多くの試みがなされてきているが，近年の [Prestwich 07, Pham 07] の手法は問題の構造を反映することで，高速な系統的ソルバーと比較し得る性能を持っている．

元来，確率的ソルバーは充足可能な SAT 問題に対してそのモデルを求めるのに長けている代わりに，充足不能な問題に対して UNSAT を判定することが系統的ソルバーのようにはできなかった．そこで，系統的ソルバーと確率的ソルバーを同時に持つことで，充足可能・充足不能いずれの問題に対しても対処できるようにすることが考えられる．この試みは，Multisat [Inoue 06] で実装され，SATLIB[*9] からの異なる種類の SAT 問題を平均的によい性能で解いたほか，プランニングやスケジューリングなど最適解を求める問題において，上下限からそれぞれ SAT と UNSAT を繰り返し判定することで解の範囲を絞っていくインクリメンタル探索において有効性が示された．ここでは，充足不能な部分問題においては Chaff が問題を最初に解き，充足可能な問題では Walksat が解くというように，各ソルバーの特徴を引き出している．

このアプローチは一般に，**ポートフォリオ戦略** (portfolio strategy) [Huberman 97] として，複数のアルゴリズムを組み合わせて難しい組合せ問題を解く一般的な解法に対する一つの実装になっている．経済学からのアイデアであるポートフォリオ戦略を用いることで，解を見つけられないリスクと探索の効率化のトレードオフを最適化することができる．Multisat が系統的ソルバーと確率的ソルバーの両者を用いているのに対し，Gomes と Selman は複数の確率的ソルバーを組み合わせたポートフォリオ戦略について解析しており [Gomes 01]，SATzilla では高速な系統的ソルバーのパラメータを複数変えた形でポートフォリオ戦略を用いている [Xu 08].

5. SAT 競 技 会

SAT ソルバーの性能を理論面ではなく実用面で評価する場として，第 5 回の SAT 国際会議以降に，**SAT 競技会** (SAT competition) が併設開催されるようになった．それ以前にも，第 2 回 DIMACS Challenge など 1990 年代前半にいくつかの催しがあったものの，SAT ソルバーの高速化に対応した競技会の本格的な幕開けはこの SAT2002 競技会からである．SAT を解くためのヒューリスティックスに関しては，膨大な量の研究論文が出版されているものの，アルゴリズム相互の比較は理論面だけでは難しく論文からは実用面における指針が得られないことが多い．競技会は SAT ソルバーを工学レベルでサポートし，各ソルバーの経験的評価を容易にし，各々の長短所と限界を明らかにしてくれる．さらに，このような会に参加することで開発者の研究意欲も刺激し，人材育成にもつながっている．まさに，アルゴリズム研究者や優秀なプログラマーの腕の見せ所でもある．また，多くの実応用問題を解けるように SAT ソルバーをチューニングすることは重要であるが，SAT 競技会向けに厳選される問題が，競技終了後もソルバー開発のためのベンチマーク問題として SATLIB 問題とともに使用され続けている．

SAT 競技会は 2002〜2005 年まで毎年開催され，以降は隔年開催となったものの，開催されなかった年には **SAT レース** (SAT race) が行われている．競技会では，ベンチマーク問題のタイプによって 3 部門に分かれて競い合う:

- **Applications (aka Industrial) 部門:** モデル検査，プランニング，暗号，バイオインフォマティクスなど，多様な応用問題のインスタンスを集めたもの．現在の SAT ソルバーの実問題における威力を見せることを意図しており，最も重視されている部門である．
- **Random uniform k-SAT 部門:** ランダム k-SAT 問題のインスタンスだけを集めたもの．
- **Crafted (aka Handmade) 部門:** ソルバーが解くのに手間取ったその他の問題すべてを扱う．解けなかった問題の数が最小なソルバーが表彰される．

それぞれの部門で，問題の充足可能性を示す SAT，問題の充足不能性を示す UNSAT，どちらのタイプの問題もあってモデルの有無を判定する SAT+UNSAT，の 3 種類に分かれて競い合う．UNSAT であることを示す問題では完全な探索を必要とするため，SAT であることを示す問題よりも一般に難しいとされる．

表 1 Gold Medalists of SAT 2009 Competition

Category	SAT+UNSAT	SAT	UNSAT
Application	precosat	SATzilla	glucose
Random	SATzilla2009_R	TNM	March_hi
Crafted	clasp	clasp	SATzilla2009_C

2009 年の SAT 競技会で優勝したソルバーを表 1 に記す．Application 部門において，precosat は Minisat 系と呼ばれる MiniSat [Eén 03a] を改良した系統的ソルバー，SATzilla はポートフォリオ型 [Xu 08]，glucose は Minisat 系 [Audemard 09] である．Crafted 部門において，clasp は解集合ソルバでもある系統的ソルバー [Gebser 07]，SATzilla2009_C はポートフォリオ型である．Random 部門の SATzilla2009_R はポートフォリオ型，TNM は確率的ソルバー，March_hi は系統的ソルバーである．ほかに特別部門として，並列ソルバー[*10]の Application 部門で ManySAT (マルチコアに対応したポートフォリ

*9 http://www.satlib.org/

*10 並列 SAT ソルバーや分散 SAT ソルバーについては [平山 10] を参照のこと．

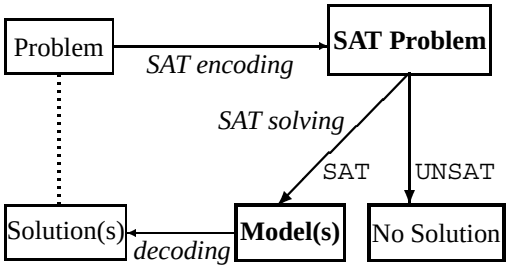

図 4 Problem Solving with SAT Solvers

オ型) が，並列ソルバーの Random 部門で gNovelty2+ (確率的ソルバー) が，Best Minisat Hack 部門で Minisat 09z がそれぞれ優勝した．この結果から，問題のタイプによって得意・不得意が出ており，優勝者もほぼ分散している．ポートフォリオ型の SATzilla は 3 部門で優勝しており，平均的な強みがある万能型のソルバーである．また，SLS に基づく確率的ソルバーの進歩が，とくにランダム SAT において見られた．なお，競技会に参加したソルバーはすべて競技会のページ[*11] からダウンロード可能であり，原則としてソースコードも公開されている．

6. SAT ソルバーをどう使うか?

これまでに SAT を解くアルゴリズムと主要なソルバーについて概観してきたが，いずれも変数集合と CNF 式が与えられたときに，CNF 式を充足する割り当てが存在するか否かを判定することが主目的であった．この問題は構造が明確で単純であるが，現実の問題を直接表現するための枠組みとは言えない．実際，論理パズルや論理回路などブール制約で自然に書ける問題やランダム SAT 問題を除くと，SAT 競技会で用いられる問題やベンチマーク問題のほとんどは，より複雑な応用問題から SAT 問題に変換されて作られたものである．そこで本章では，こうした SAT 問題への変換を含め，実際の問題が SAT とどのように関わっているかについて述べる．

図 4 に，ある問題を SAT ソルバーを利用して解く様子を示す．実際に SAT ソルバーを利用する際には，解きたい問題をいかにして SAT 問題に変換するかという **SAT 符号化** (SAT encoding) 手法が重要になる．このとき，問題 P が解けることと，SAT 符号化によりできる問題 $\tau(P)$ が充足可能 (または充足不能) であることが 1 対 1 対応していなければならない[*12]．さらに，多くの問題では解の存在だけを問うのではなく，実際に解を得たい場合がある．この場合，得られたモデルに含まれる情報を抽出・加工することにより，元の問題の解を得るような**復号化** (decoding) も必要である．

SAT 符号化においては，命題変数の組 V と命題 CNF 式 ψ を生成しなければならないが，解きたい元の問題はより高度な表現形式で記述されることが多い．以下では，より複雑な表現形式から命題論理表現に帰着させるための SAT 符号化および拡張に関する取り組みの例を挙げる．これらの多くは本 SAT 特集における他の解説で詳細が論じられているので合わせて参照していただきたい．

i. **多値変数間の制約**

制約充足問題 (constraint satisfaction problem; **CSP**) における領域変数は一般に離散的な多値変数である．例えば，グラフ彩色問題において，ノード i の色を表す変数 v_i は赤，青，黄，などの値を取る．この場合例えば，ブール変数 $v_{i,j}$ を導入して,「v_i の色は j である」という命題を表現し，これらの命題を変数としてブール制約を書き表すことができる．CSP の SAT 符号化に関しては他にもさまざまな符号化法が考えられており，[田村 10] でそれらの詳細と比較が論じられている．

ii. **最適化問題**

CSP が与えら得た制約を充足する変数の値の組を求めるのに対し，**制約最適化問題** (constraint optimization problem; COP) では，そのように制約を充足する解の中で，ある目的関数の値を最小または最大にするような変数の値の組を求める問題である．この場合は，CSP を複数回解くことで最適解を求めることができ，解説 [田村 10, 鍋島 10a] ではそのような例が**インクリメンタル探索** (incremental search) [Eén 03b] として取り上げられている．すなわち，最小化 (または最大化) すべきパラメーターの値を固定した判定問題を解き，その結果によって上限または下限を更新し，次々にパラメーターを変えて問題を解いていく方法で SAT ソルバーが利用可能である [Inoue 06]．例えば，矩形パッキング問題がこの方式で解かれている [Soh 10].

iii. **順列の問題**

一般に COP は組み合わせ最適化問題であり，取り得る要素の組み合わせが解になる．これに対して，アクションやジョブなど要素の順序が重要になる問題がある．このうち，**プランニング** (planning) は目標を達成するアクション列を作成する問題であり，AI における SAT 応用の代表例として 90 年代から盛んに SAT 符号化に関する研究がなされてきた (例えば [Kautz 96])．一方で，**スケジューリング** (scheduling) はオペレーションズリサーチで研究されてきた歴史が長く SAT 符号化の研究は多くないものの，[Crawford 94] で提案された SAT 符号化手法の拡張が一般の CSP でも有効に利用できることがわかっている [Tamura 09, 田村 10]．これらの問題における SAT 符号化の詳細に関しては [鍋島 10a] を参照されたい．

なお，プランニングとスケジューリングにおいては，ある所定ステップや時間で解けるかどうかを判定する問題に加えて，最短長のプランやスケジュールを求める最適化問題として与えられる場合が多い．この場合は，COP

*11 http://www.satcompetition.org/
*12 この性質が成立し，かつ SAT 符号化が多項式時間で行われるためには，問題 P の計算量が NP 完全を超えてはならない．

と同様に SAT ソルバーを用いたインクリメンタル探索により問題を解いていくことができ，高速ソルバーを使えば学習節も再利用可能である [Nabeshima 06].

iv. **様相論理式**

有界モデル検査 (bounded model checking; **BMC**) は，SAT ソルバーの工学的応用の中では最も成功したものの一つである．BMC ではある時点で性質が成り立つかどうかといった様相を扱うが，[Biere 99] により時相論理式の SAT 符号化が示された．BMC では，元の問題の否定を取った命題を SAT 符号化し，SAT であればそのモデルは元の問題の性質を成り立たせない反例に対応し，UNSAT ならば元の性質が常に成り立つことを示すことになる．なお，BMC を含むハードウェア・ソフトウェアの検証問題では一般に，スケジューリング問題と同様に最適化問題を扱っている [Eén 03b]．BMC を含むシステム検証の詳細は [番原 10] を参照されたい．

v. **擬似ブール制約**

ブール制約の一般化として，**擬似ブール制約** (pseudo Boolean constraints) がある．擬似ブール制約は変数が 0 か 1 のブール値だけを取る制約充足問題であるが，とくに $\sum_i a_i x_i \geq b$ の形の線型制約を主たる対象としている．ここで a_i と b は整数，x_i はリテラルであり，真を 1，偽を 0 と解釈する．擬似ブール制約は，真となるリテラルの個数に関する条件である**基数制約** (cardinality constraint) をそのまま記述でき，テストケース自動生成 [番原 10] などの重要な応用がある．なお，$x_1 + x_2 + \cdots + x_n \geq 1$ により節 $x_1 \vee x_2 \vee \cdots x_n$ を表せることから分かるように，擬似ブール制約は SAT を含んでいる．したがって，擬似ブール制約の解を探索する擬似ブールソルバーを SAT ソルバーとして利用することも可能である．しかし，現状では SAT ソルバーに匹敵する性能は得られておらず，今後の発展が期待される．逆に，擬似ブール制約を SAT に符号化し通常の SAT ソルバーで解を探索する方法も研究されている [Eén 06, Bailleux 06]．特に，MiniSat+ [Eén 06] は，2005 年の擬似ブールソルバー競技会で優れた成績を収めている点が注目に値する．

vi. **一階述語論理式**

一階述語論理式の充足可能性判定は，**反駁法** (refutation) による自動定理証明で用いられる．エルブランの定理より，ある一階述語節理論 P が充足不能であるためには，P の節のすべての変数を定数で置き換えた基礎例の有限集合が存在して充足不能である．基礎節集合には SAT ソルバーが適用できるため，ここでの SAT 符号化は基礎化に等しい．元々DP と DPLL は定理証明研究の中で産み出されたが，適切な基礎例の集合を生成する方法が開発されていなかったため，融合原理による方法が研究の主流であった．ところが 90 年代半ばより基礎例の生成手法の改善が Plaisted らによって開始され [Plaisted 00]，最近では最速の定理証明プログラムに匹敵する性能の証明器が出現してきた．これらの手法は現在**例化に基づく定理証明** (instantiation-based theorem proving) と呼ばれており，基礎例の生成と SAT による解法のフェーズが明確に分かれている手法と，分かれていない手法に大別される．前者の代表は Ganzinger によって考案された Inst-Gen 法 [Ganzinger 03] であり，後者は Baumgarter らの Model Evolution 法 (ME) [Baumgartner 08] などがある．DPLL の研究論文 [Davis 60, Davis 62] の手法に近いのは前者であるが，後者の ME は DPLL を一階論理に直接持ち上げようとしており，この意味で SAT に深く関連している．

定理証明プログラムとしては，iProver が Inst-Gen に基づいたもので，Darwin が ME をもとにしたものであるが，双方とも優秀な証明器であり，CASC2007 の EPR 部門で 1 位と 2 位を取っている．

vii. **算術式・データ構造・背景理論**

ソフトウェアの形式的検証において，配列やリストなどのデータ構造や算術演算まで考えると，そのための背景理論が必要になり，コンパクトな記述のためには述語表現が用いられる．述語表現に加えて，こうしたデータ構造や算術式を記述できるように拡張した一階理論については，**背景理論付き SAT** (Satisfiability Modulo Theories; **SMT**) として定式化することにより，SAT ソルバーを拡張した SMT ソルバーが利用できる．SMT は現在かなり活発に研究されており，その詳細に関しては [岩沼 10] を参照されたい．なお，等式を扱う述語論理の定理証明においても，等号の扱いに関する理論を背景で用いるため，例化に基づく定理証明の [Ganzinger 03] も SMT ソルバーを使っていると言える．

viii. **解集合プログラミング**

システム検証に限らず，一般に知識表現として述語表現を許すものとしては，**解集合プログラミング** (answer set programming; **ASP**) がある．ASP は制約プログラミングの一種でもあり，問題を論理プログラムとして記述し，解集合ソルバーによって求まる解集合が元の問題の解に対応する．ASP においても基数制約が記述できるため検証問題に使われるほか，非単調推論を含む AI 問題や，セマンティックウェブやバイオインフォマティクスなどへの応用がある．解集合ソルバーと SAT ソルバーは多くの類似点があり，clasp [Gebser 07] のように高速 SAT ソルバーとしても使えるものもある．ASP の解説については [井上 08] を参照されたい．

ix. **解の列挙**

システム検証における反例をすべて見つけたり，推論において論理的帰結の検査を行う場合などで，ある論理式集合のすべてのモデルを求めたい問題もある．その場合，通常の SAT ソルバーのように一つモデルを求めて終了するのではなく，バックトラックして他のモデルも計算する必要がある (**モデル列挙**，model enumeration)．なお，ASP でも通常は解集合を列挙している．さらに，確率推論では，ある論理式を充足するモデルが全体のモデル集合の中でどれくらいあるかを知りたいために，**モデ**

ル計数 (model counting, **#SAT**) の技術が必要になる問題もある．これらの詳細は [長谷川 10] を参照されたい．

x. **確からしさの付与**

SAT 符号化において，知識の確からしさや変更可能性の尺度に応じて，節に重みをつけることがある．こうしてできる問題は**重み付き MaxSAT** であり，ソルバーも開発されている．このうち，必ず満たして欲しいハード制約とできれば満たして欲しいソフト制約の 2 種類の重みだけを持つ問題が MaxSAT であり，すべてがハード制約である問題が SAT に相当する．(重み付き)MaxSAT に関する詳細は [平山 10] を参照されたい．

xi. **限量ブール式**

全称 (∀) または存在限量子 (∃) 付きのブール制約式 (Quantified Boolean formula; **QBF**) の充足可能性判定問題は **QSAT** (quantified SAT) とも呼ばれている．通常の SAT は変数がすべて存在限量されている QSAT である．一般の QSAT の計算量は PSPACE 完全であるため SAT 符号化は望めないが，SAT ソルバーを拡張した **QBF ソルバー**で解くことができる．また，**多項式階層** (polynomial hierarchy) に含まれる AI 問題を QSAT 問題に変換し QBF ソルバーで解く研究も増えてきている [Egly 00]．QSAT に関する詳細は [平山 10] を参照されたい．

7. お わ り に

本解説では SAT ソルバーの基礎として，系統的ソルバーと確率的ソルバーの基礎とそれぞれの特徴，さらに SAT ソルバーの拡張と応用の概要について説明した．前章で述べたように，SAT ソルバーは多くの問題で盛んに利用されている．高速 SAT ソルバーが解の有無を判定することを利用して，SAT 符号化をアセンブリ言語記述のように使ったり，数式パッケージツールや制約ソルバーのように必要に応じて SAT ソルバーを呼び出したり，より高次の推論システムを実現するための基本ソフトのような役割を果たしたりと，使われ方もさまざまである．まさに,「問題を速くスケーラブルに解くために，バックエンドとして高速 SAT ソルバーを使う」という問題解決の方法論が確立されつつあると言える．また，SAT ソルバーの開発で培われた高速化技術が他のソフトウェア開発でも応用されれば，他分野においても高速なシステムが今後出現してくることになるかもしれない．

謝　　辞

本研究の一部は，平成 20∼23 年度科学研究費補助金基盤研究 (A)「制約最適化問題の SAT 変換による解法とその並列分散処理に関する研究」(No. 20240003，代表: 田村直之) および平成 21 年度 NII 共同研究 (公募型)「SAT 変換技術の拡張による求解困難な制約最適化問題の解法に関する研究」による．本解説ではとくに，山梨大学・岩沼宏治教授に SAT と SMT と定理証明の関連について貴重なコメントをいただいた．

◇ 参　考　文　献 ◇

[Audemard 09] Audemard, G. and Simon, L.: Predicting learnt clauses quality in modern SAT solvers, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)* (2009)

[Bailleux 06] Bailleux, O., Boufkhad, Y., and Roussel, O.: A translation of pseudo Boolean constraints to SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, pp. 191–200 (2006)

[番原 10] 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)

[Baumgartner 08] Baumgartner, P. and Tinelli, C.: The model evolution calculus as a first-order DPLL method, *Artificial Intelligence*, Vol. 172, No. 4–5, pp. 591–632 (2008)

[Bayardo 97] Bayardo, R. J. and Schrag, R. C.: Using CSP look-back techniques to solve real-world SAT instances, in *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, pp. 203–208 (1997)

[Biere 99] Biere, A., Cimatti, A., Clarke, E., and Zhu, Y.: Symbolic model checking without BDDs, in *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS '99)*, Vol. 1579 of *Lecture Notes in Computer Science*, pp. 193–207, Springer (1999)

[Biere 09] Biere, A., Heule, M. J. H., Maaren, van H., and Walsh, T. eds.: *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications*, IOS Press (2009)

[Braunstein 05] Braunstein, A., Mézard, M., and Zecchina, R.: Survey propagation: an algorithm for satisfiability, *Random Structures & Algorithms*, Vol. 27, No. 2, pp. 201–226 (2005)

[Chang 71] Chang, C.-L. and Lee, R. C.-T.: *Symbolic Logic and Mechanical Theorem Proving*, Academic Press (1971)

[Cook 71] Cook, S. A.: The complexity of theorem-proving procedures, in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pp. 151–158, ACM (1971)

[Crawford 94] Crawford, J. M. and Baker, A. B.: Experimental results on the application of satisfiability algorithms to scheduling problems, in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pp. 1092–1097, AAAI Press (1994)

[Crawford 96] Crawford, J. M. and Auton, L. D.: Experimental results on the crossover point in random 3-SAT, *Artificial Intelligence*, Vol. 81, No. 1–2, pp. 31–57 (1996)

[Dantsin 09] Dantsin, E. and Hirsch, E. A.: Worst-case upper bounds, in Biere, et al. [Biere 09], pp. 403–424

[Davis 60] Davis, M. and Putnam, H.: A computing procedure for quantification theory, *Journal of the ACM*, Vol. 7, No. 3, pp. 201–215 (1960)

[Davis 62] Davis, M., Logemann, G., and Loveland, D.: A machine program for theorem proving, *Communications of the ACM*, Vol. 5, No. 7, pp. 394–397 (1962)

[Eén 03a] Eén, N. and Sörensson, N.: An extensible SAT-solver, in *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2007)*, pp. 502–518 (2003)

[Eén 03b] Eén, N. and Sörensson, N.: Temporal induction by incremental SAT solving, *Electronic Notes in Theoretical Computer Science*, Vol. 89, No. 4 (2003)

[Eén 06] Eén, N. and Sörensson, N.: Translating pseudo-Boolean constraints into SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, pp. 1–26 (2006)

[Egly 00] Egly, U., Eiter, T., Tompits, H., and Woltran, S.: Solving advanced reasoning tasks using quantified boolean formulas, in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 417–422, AAAI Press (2000)

[Ganzinger 03] Ganzinger, H. and Korovin, K.: New directions in instantiation-based theorem proving, in *Proceedings of the 18th IEEE Symposium on Logic in Computer Science (LICS '03)*, pp. 55–64, IEEE Computer Society (2003)

[Garey 79] Garey, M. R. and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Free-

man and Company, New York (1979)
[Gebser 07] Gebser, M., Kaufmann, B., Neumann, A., and Schaub, T.: Conflict-driven answer set solving, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pp. 386–392 (2007)
[Gomes 98] Gomes, C. P., Selman, B., and Kautz, H.: Boosting combinatorial search through randomization, in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pp. 431–437, AAAI Press (1998)
[Gomes 01] Gomes, C. P. and Selman, B.: Algorithm portfolios, *Artificial Intelligence*, Vol. 126, No. 1–2, pp. 43–62 (2001)
[Gomes 08] Gomes, C. P., Kautz, H., Sabharwal, A., and Selman, B.: Satisfiability solvers, in *Handbook of Knowledge Representation*, Vol. 3 of *Foundations of Artificial Intelligence*, pp. 89–134, Elsevier (2008)
[長谷川 10] 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)
[平山 10] 平山 勝敏, 横尾 真：∗-SAT: SAT の拡張, 人工知能学会誌, Vol. 25, No. 1 (2010)
[Huberman 97] Huberman, B. A., Lukose, R. M., and Hogg, T.: An economics approach to hard computational problems, *Science*, Vol. 275, pp. 51–54 (1997)
[Inoue 06] Inoue, K., Soh, T., Ueda, S., Sasaura, Y., Banbara, M., and Tamura, N.: A competitive and cooperative approach to propositional satisfiability, *Discrete Applied Mathematics*, Vol. 154, No. 16, pp. 2291–2306 (2006)
[井上 08] 井上 克巳, 坂間 千秋：論理プログラミングから解集合プログラミングへ, コンピュータソフトウェア, Vol. 25, No. 3, pp. 20–32 (2008)
[岩間 01] 岩間 一雄：CNF 充足可能性判定問題の計算複雑さ, 人工知能学会誌, Vol. 16, No. 5, pp. 636–647 (2001)
[岩沼 10] 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)
[Jain 09] Jain, H. and Clarke, E. M.: Efficient SAT solving for non-clausal formulas using DPLL, graphs, and watched cuts, in *46th Design Automation Conference (DAC)* (2009)
[Kautz 96] Kautz, H. and Selman, B.: Pushing the envelope: planning, propositional logic, and stochastic search, in *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pp. 1194–1201, AAAI Press (1996)
[Li 97] Li, C. M. and Anbulagan, : Heuristics based on unit propagation for satisfiability problems, in *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 366–371, Morgan Kaufmann (1997)
[Malik 09] Malik, S. and Zhang, L.: Boolean satisfiability—from theoretical hardness to practical success, *Communications of the ACM*, Vol. 52, No. 8, pp. 76–82 (2009)
[Marques-Silva 99] Marques-Silva, J. P. and Sakallah, K. A.: GRASP: a search algorithm for propositional satisfiability, *IEEE Transactions on Computers*, Vol. 48, pp. 506–521 (1999)
[Mézard 02] Mézard, M., Parisi, G., and Zecchina, R.: Analytic and algorithmic solution of random satisfiability problems, *Science*, Vol. 297, pp. 812–815 (2002)
[Moskewicz 01] Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S.: Chaff: engineering an efficient SAT solver, in *Proceedings of the 38th Design Automation Conference*, pp. 530–535, ACM (2001)
[Nabeshima 06] Nabeshima, H., Soh, T., Inoue, K., and Iwanuma, K.: Lemma reusing for SAT based planning and scheduling, in *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS 2006)*, pp. 103–113, AAAI (2006)
[鍋島 10a] 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)
[鍋島 10b] 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)
[Nieuwenhuis 06] Nieuwenhuis, R., Oliveras, A., and Tinelli, C.: Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T), *Journal of the ACM*, Vol. 53, No. 6, pp. 937–977 (2006)
[Pearl 88] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann (1988)
[Pham 07] Pham, D. N., Thornton, J., and Sattar, A.: Building structure into local search for SAT, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pp. 2359–2364 (2007)
[Plaisted 00] Plaisted, D. A. and Zhu, Y.: Ordered semantic hyperlinking, *Journal of Automated Reasoning*, Vol. 25, No. 3, pp. 167–217 (2000)
[Prestwich 07] Prestwich, S. D.: Variable dependency in local search: prevention is better than cure, in *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT 2007)*, pp. 107–120 (2007)
[Robinson 65] Robinson, J. A.: A machine-oriented logic based on the resolution principle, *Journal of the ACM*, Vol. 12, No. 1, pp. 23–41 (1965)
[Selman 92] Selman, B., Levesque, H. J., and Mitchell, D. G.: A new method for solving hard satisfiability problems, in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, pp. 440–446, AAAI Press (1992)
[Selman 96a] Selman, B., Kautz, H., and Cohen, B.: Local search strategies for satisfiability testing, in Johnson, D. J. and Trick, M. A. eds., *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 26, pp. 521–532, American Mathematical Society (1996)
[Selman 96b] Selman, B., Mitchell, D. G., and Levesque, H. J.: Generating hard satisfiability problems, *Artificial Intelligence*, Vol. 81, No. 1–2, pp. 17–29 (1996)
[Soh 10] Soh, T., Inoue, K., Tamura, N., Banbara, M., and Nabeshima, H.: A SAT-based method for solving the two-dimensional strip packing problem, *Journal of Algorithms in Cognition, Informatics and Logic*, p. to appear (2010)
[Tamura 09] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling finite linear CSP into SAT, *Constraints*, Vol. 14, No. 2, pp. 254–272 (2009)
[田村 10] 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)
[Xu 08] Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K.: SATzilla: portfolio-based algorithm selection for SAT, *Journal of Artifificial Intelligence Research*, Vol. 32, pp. 565–606 (2008)
[Zhang 01] Zhang, L., Madigan, C. F., Moskewicz, M. W., and Malik, S.: Efficient conflict driven learning in boolean satisfiability solver, in *Proceedings of ICCAD-01*, pp. 279–285 (2001)

〔担当委員：××○○〕

19YY 年 MM 月 DD 日 受理

著 者 紹 介

井上 克巳（正会員）

1982 年京都大学工学部数理工学科卒業．1984 年同大学院工学研究科数理工学専攻修士課程修了．松下電器産業 (株), (財) 新世代コンピュータ技術開発機構，豊橋技術科学大学，神戸大学を経て，2004 年より国立情報学研究所．現在，同情報学プリンシプル研究系教授および総合研究大学院大学複合科学研究科情報学専攻教授．2008 年より東京工業大学大学院情報理工学研究科計算工学専攻連携教授．京都大学博士 (工学)．人工知能，システム生物学などの研究に従事．

田村 直之（正会員）

1980 年神戸大学理学部物理学科卒業．1985 年同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 年より神戸大学学術情報基盤センター学術情報処理研究部門教授．制約プログラミング，SAT，線形論理，論理プログラミングなどに興味をもつ．日本ソフトウェア科学会，情報処理学会会員．