日本ソフトウェア科学会第 27 回大会 (2010 年度) 講演論文集

# SAT 技術を用いた組合せテストケース生成

番原 睦則　松中 春樹　田村 直之　井上 克巳

本稿では，SAT 符号化を用いた組合せテストのテストケース生成について述べる．命題論理の充足可能性判定問題 (SAT 問題) を解く SAT ソルバーの性能が飛躍的に向上したことをうけて，SAT 技術を多分野に応用する研究が急速に拡大している．SAT 符号化とは，元の問題を SAT 問題に変換し，SAT ソルバーを用いて求解する方法である．組合せテストはソフトウェア／ハードウェアのテスト手法の一つである．この手法の特長は，欠陥の多くは少数のパラメータの組み合わせによって発生するという観測を元に，テストケースの増大を回避し，現実的かつ効果的なテストを行える点である．本研究では，順序符号化法と Hnich 符号化法の 2 つの SAT 符号化の長所を取り入れることにより，組合せテストケース生成の性能を大きく向上できることを示す．

## 1　はじめに

命題論理の充足可能性判定 (SAT) は，与えられた命題論理式の充足可能性を判定する問題である．SAT は Cook により最初に NP 完全性が証明された問題である．SAT は計算機科学および人工知能における最も基本的な問題として，論理合成，プランニング問題，スケジューリング問題，制約充足問題 (Constraint Satisfaction Problems, CSP)，制約最適化問題，定理証明など，さまざまな分野に応用されている．近年，$10^6$ 〜 $10^7$ 個の変数をもつ大規模な SAT 問題を，非常に高速に解くことが可能な SAT ソルバーが実現され，これらの分野への実用的応用が急速に拡大している [18]．特に，CSP を SAT 問題に変換して，高速な SAT ソルバーを用いて求解する **SAT 符号化**の研究が注目を集め，これまでに数多くの符号化法が提案されている：直接符号化法 (direct encoding) [9][34]，支持符号化法 (support encoding) [20][13]，多値符号化法 (multivalued encoding) [26]，対数符号化法 (log encoding) [19][12]，順序符号化法 (order encoding) [31][32]．

ソフトウェア／ハードウェアのテストは，製品を開発する過程において重要な役割を果たしている．しかし，たとえ小規模な製品であっても，網羅的テストはテストケースの増大を招き現実的に実行不可能である．また，ソフトウェア／ハードウェアの大規模化・複雑化に伴い，必要となるテストケースが増える一方，そのテスト期間については短期化が求められている．

**組合せテスト**(Combinatorial Testing) はソフトウェア／ハードウェアのテスト手法の一つである．この手法の特長は，欠陥の多くは少数のパラメータの組み合わせによって発生するという観測を元に，テストケースの増大を回避し，現実的かつ効果的なテストを行える点である．組合せテストのテストケース生成は，組合せデザイン分野の**被覆配列** (Covering Array, $CA$) †1を生成する問題に帰着できる．組合せテストに対する被覆配列の有用性が示

Generating Combinatorial Test Cases by SAT Techniques

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

Haruki Matsunaka, 神戸大学大学院システム情報学研究科, Graduate School of System Informatics, Kobe University.

Katsumi Inoue, 国立情報学研究所, National Institute of Informatics.

†1　本稿では covering array の和訳として “被覆配列” を用いる．

されて以来，数多くの $CA$ 生成手法が提案されている [35] [2] [14] [3] [22] [24] [5] [28] [1] [15] [16]．これらの研究のサーベイ論文としては [6] [33] がある．

Hnich らは，確率的 SAT ソルバーに適した $CA$ の SAT 符号化法を提案している [15] [16]．Myra B. Cohen らは，SAT と貪欲法を組合せたアルゴリズムを提案している [4]．しかしながら，Conflict-Driven Clause Learning (CDCL) SAT ソルバーに適した SAT 符号化の研究は非常に少ない．

本稿では，最小 $CA$ を生成する問題に対して，CDCL SAT ソルバーに適した新しい SAT 符号化法を二つ提案する．一つは，順序符号化法をベースにしたものである．順序符号化法では，SAT ソルバーの基本動作である単位伝播 (unit propagation) が，元の CSP における範囲伝播 (bounds propagation) に対応しており，効率の良い求解が可能である．もう一つは，Hnich 符号化法と順序符号化法を融合したものである．順序符号化法と比較して，カバレッジ制約の符号化に必要な節数が少なくてよい点が特長である．

CDCL SAT ソルバーは，近年の SAT 技術において重要な役割を果たしている．この CDCL SAT ソルバーを用いることは，効率の良いテストケース生成を実現するために重要である．提案する二つの符号化法は，どちらも全体的あるいは部分的に順序符号化法を用いている．順序符号化法と CDCL SAT ソルバーとの相性の良さは，順序符号化法を実装した CSP ソルバー Sugar[†2] が 2008–2009 年国際 CSP ソルバー競技会において 2 年連続優勝 (GLOBAL 部門) したことからも伺える．

提案する SAT 符号化法の有効性を示すために，強さ 2〜6 の小中規模の問題に対する実行実験を行った．その結果，提案手法は様々な既存手法 (群論等の数学的手法，貪欲法，局所探索法など) と比較して，ほぼ同等の結果を得ることができた．さらに，2009 年に出版された Handbook of Satisfiability [36] 中に記載されている未解決問題に対して，既知の最良値が最適値であることを証明した．加えて，いくつかの問題に対して，既知の下限 [2] を更新することにも成功した．

†2 http://bach.istc.kobe-u.ac.jp/sugar/

| | | | | |
|---|---|---|---|---|
| **0** | **0** | **0** | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| **0** | **0** | **1** | 0 | 0 |
| **0** | **1** | **0** | 0 | 0 |
| **0** | **1** | **1** | 1 | 1 |
| **1** | **0** | **0** | 0 | 0 |
| **1** | **0** | **1** | 1 | 1 |
| **1** | **1** | **0** | 1 | 1 |
| **1** | **1** | **1** | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

**図 1** $CA(10;3,5,2)$．**どの 3 個の列についても，全部で $2^3$ 通りあるそれらの値の組合せすべてが出現している．**

以下，第 2 節で被覆配列の定義・関連研究について概観する．第 3 節では既存の SAT 符号化法である Hnich 符号化法を紹介する．提案手法は第 4–5 節で述べる．第 6 節では実行実験について述べ，最後に第 7 節で結論を述べる．

## 2 被覆配列と関連研究

### 2.1 被覆配列

被覆配列 $CA(b;t,k,g)$ とは，$b \times k$ 配列 ($b$ 行 $k$ 列) であり，各要素は $\{0,1,2,\ldots,g-1\}$ のいずれかの値を取る．そして，どの $t$ 個の列 ($1 \leq t \leq k$) についても，全部で $g^t$ 通りあるそれらの値の組合せすべてが少なくとも一つ出現する．ここでは，$t$ を**強さ**，$k$ を**因子**，$g$ を**水準**と呼ぶことにする．目的は $CA(b;t,k,g)$ が存在する最小の $b$ を見つけることである．

**例 1** 組合せテストにおける一つのテストケースが，$CA(b;t,k,g)$ の各行に対応しており，テストケース数が $b$ の値に対応する．例として，強さ $t=3$，因子 $k=5$，水準 $g=2$ に対するテストケースの生成，すなわち $CA(b;3,5,2)$ を生成する問題を考える．図 1 に，この問題の最適解 ($b=10$) を示す．最初の 3 列について，異なる 0 と 1 の組合せをボールド体で記している．全部で $2^3$ 通りあるそれらの値の組合せすべてが，少なくとも一つ出現していることがわかる．他のどの 3 列の組合せについても同様の性質を満たす．

以下の定義は，論文 [16] に基づいている．

**定義 1 (被覆配列)** 被覆配列 $CA(b;t,k,g)$ とは，以下の性質を満たす $b \times k$ 配列 $A=(a_{ij})$ である．

- $a_{ij} \in \mathbb{Z}_g = \{0,1,2,\ldots,g-1\}$

- 任意の異なる $t$ 個の列 $1 \leq c_1 \leq c_2 \leq \ldots \leq c_t \leq k$，および任意の値の組 $(x_1, x_2, \ldots, x_t) \in \mathbb{Z}_g^t$ に対して，$x_i = a_{rc_i} (\forall i; 1 \leq i \leq t)$ を満たす行 $r$ が少なくとも一つ存在する．

**定義 2 (被覆配列数)** 被覆配列数 $CAN(t,k,g)$ とは，$CA(b;t,k,g)$ が存在する最小の $b$ である．

$$CAN(t,k,g) = \min\{b \mid CA(b;t,k,g)\}$$

**定理 1 ( [2] [36])** $CAN(t,k,g)$ について，以下の性質が成り立つ．

1. $g^t \leq CAN(t,k,g) \leq g^k$
2. $CAN(t,k-1,g) \leq CAN(t,k,g)$
3. $CAN(t,k,g-1) \leq CAN(t,k,g)$
4. $g \cdot CAN(t-1,k-1,g) \leq CAN(t,k,g)$

一般に，$CAN(t,k,g)$ の決定は NP 完全である [27]. 強さ $2 \leq t \leq 6$ に対する $CAN(t,k,g)$ の最新の上限は，Charles Colbourn によって Web 上で管理・公開されている [7]．表 1 に，$t = 3$ および $k, g$ の小さな値に対して，これまでに知られている $CAN(t,k,g)$ の最新の上下限 [2] [7] を示す．表の $(k,g)$ 成分が整数 $m$ の場合は $CAN(t,k,g) = m$ を表し，$m,n$ の場合は $m \leq CAN(t,k,g) \leq n$ を表している．

本稿では，SAT(および CSP) を用いたアプローチをより分かりやすくするために，2 つの問題を定義する．***CA* 判定問題**とは， 与えられた $\langle t,k,g,b \rangle$ に対して，$CA(b;t,k,g)$ が存在するかどうかを判定し，存在する場合，$CA(b;t,k,g)$ を生成する問題である．***CA* 最適化問題**とは，与えられた $\langle t,k,g \rangle$ に対して，$CAN(t,k,g) = b$ となる最小の $CA(b;t,k,g)$ を生成する問題である．

### 2.2 Hnich の CSP 表現

Hnich らは，$CA$ 判定問題を CSP で表現する方法を提案している [15] [16]．この場合，$CA(b;t,k,g)$ に対する CSP 問題は，サイズ $b$ に対する判定問題となる．つまり，CSP 問題が

- 充足可能 (SAT) であれば「サイズ $b$ の被覆配列が存在」
- 充足不能 (UNSAT) であれば「サイズ $b$ の被覆配列は存在しない」

と判定できる．この方法を $CA$ 最適化問題へ応用するには，$b$ の値を変化させた CSP 問題群を生成し，二分探索等を用いて，最小の $b$ の値 (UNSAT 直後の SAT) を求めればよい．

Hnich の CSP 表現は，2 つの行列と 2 種類の制約から構成される．

**基本行列** は整数変数を要素とする $b \times k$ 行列である．各整数変数 $x_{r,i}$ $(1 \leq r \leq b, 1 \leq i \leq k)$ は，被覆配列の各要素の値を表し，そのドメインは $x_{r,i} \in \{0,1,2,\ldots,g-1\}$ である．すなわち $x_{r,i} = m$ は $CA(b;t,k,g)$ の $(r,i)$ 成分が $m$ であることを表す．

**拡張行列** は整数変数を要素とする $b \times \binom{k}{t}$ 行列である．各列は $t$ 個の因子の可能な組合せの一つを表す．各整数変数 $y_{r,i'}$ $(1 \leq r \leq b, 1 \leq i' \leq \binom{k}{t})$ は，基本行列における $t$ 個の変数の組を表し，そのドメインは $y_{r,i'} \in \{0,1,2,\ldots,g^t-1\}$ である．

**カバレッジ制約** はどの $t$ 個の列 $(1 \leq t \leq k)$ についても，全部で $g^t$ 通りあるそれらの値の組合せすべてが少なくとも一つ出現するという制約である．このカバレッジ制約は，拡張行列の各列中に 0〜$g^t-1$ の値が少なくとも一つ出現するという**基数制約**に置換えることができる．Hnich らは，基数制約として大域基数制約 (Global Cardinality Constraint, $GCC$) [25] を採用している†3.

**チャネリング制約** は拡張行列の変数と，対応する基本行列の $t$ 個の変数とを結びつける制約である．$1 \leq c_1^{i'} \leq c_2^{i'} \leq \cdots \leq c_t^{i'} \leq k$ を拡張行列の列 $i'$ に対応する基本行列の相異なる $t$ 個の列とする．この場合，チャネリング制約の内包的表現は以下のようになる (ただし，$1 \leq r \leq b, 1 \leq i' \leq \binom{k}{t}$)．

$$y_{r,i'} = \sum_{\ell=1}^{t} g^{t-\ell} x_{r,c_\ell^{i'}}$$

チャネリング制約については，内包的表現と外延的表現の 2 通りが可能である．例 1 の場合，その内包的表現は

$$y_{r,(i,j,\ell)} = 4x_{r,i} + 2x_{r,j} + x_{r,\ell}$$

また，外延的表現を用いる場合は以下のようになる．

†3 $GCC$ を用いれば，各列中に 0〜$g^t-1$ の値が高々 $b - g^t + 1$ 個出現するというカバレッジ制約の上限も表現できる (ただし，この条件は省略可能である)．

**表 1 既知の $CAN(3,k,g)$**

| $k\backslash g$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 4 | 8 | 27 | 64 | 125 | 216 | 343 | 512 |
| 5 | 10 | 28,33 | 64 | 125 | 222,240 | 343 | 512 |
| 6 | 12 | 33 | 64 | 125 | 222,258 | 343 | 512 |
| 7 | 12 | 36,40 | 76,88 | 125,180 | 222,293 | 343 | 512 |
| 8 | 12 | 36,42 | 76,88 | 145,185 | 222,304 | 343 | 512 |
| 9 | 12 | 36,45 | 76,112 | 145,185 | 234,379 | 343,472 | 512 |
| 10 | 12 | 36,45 | 76,112 | 145,185 | 234,393 | 364,479 | 512 |
| 11 | 12 | 36,45 | 76,121 | 145,225 | 234,463 | 364,637 | 536,960 |
| 12 | 14,15 | 36,45 | 76,121 | 145,225 | 234,463 | 364,637 | 536,960 |
| 13 | 14,16 | 36,51 | 76,124 | 145,245 | 234,503 | 364,637 | 536,960 |
| 14 | 14,16 | 36,51 | 76,124 | 145,245 | 234,503 | 364,637 | 536,960 |
| 15 | 14,17 | 36,57 | 76,124 | 145,245 | 234,514 | 364,637 | 536,960 |
| 16 | 14,17 | 36,60 | 76,124 | 145,245 | 234,514 | 364,637 | 536,960 |

| (1,2,3) | (1,2,4) | (1,2,5) | (1,3,4) | (1,3,5) | (1,4,5) | (2,3,4) | (2,3,5) | (2,4,5) | (3,4,5) |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 2 | 1 | 0 | 2 | 2 |
| **1** | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 |
| **2** | 2 | 2 | 0 | 0 | 0 | 4 | 4 | 4 | 0 |
| **3** | 3 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 |
| **4** | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 |
| **5** | 5 | 5 | 7 | 7 | 7 | 3 | 3 | 3 | 7 |
| **6** | 7 | 7 | 5 | 5 | 7 | 5 | 5 | 7 | 3 |
| **7** | 6 | 7 | 6 | 7 | 5 | 6 | 7 | 5 | 5 |
| 7 | 7 | 6 | 7 | 6 | 6 | 7 | 6 | 6 | 6 |

**図 2 図 1 の $CA(10;3,5,2)$ を拡張行列で表現．各列は横線の上に示した因子の組を表している．**

$$(y_{r,(i,j,\ell)}, x_{r,i}, x_{r,j}, x_{r,\ell}) \in \{(0,0,0,0),(1,0,0,1),$$
$$(2,0,1,0),(3,0,1,1),(4,1,0,0),$$
$$(5,1,0,1),(6,1,1,0),(7,1,1,1)\}$$

図 2 に図 1 を拡張行列で表現したものを示す．

本稿では，Hnich の CSP 表現を SAT 問題に変換する符号化法を提案する．提案手法を述べる前に，既存の Hnich 符号化法について述べる．

## 3 Hnich 符号化法

SAT 符号化を行うには，CSP 表現を連言標準形 (Conjunctive Normal Form, CNF) で表す必要がある．

Hnich 符号化法 [15] [16], では，命題変数 $p(x_{r,i}=v)$ と $p(y_{r,i'}=w)$ を導入する (ただし，$1 \le r \le b$, $1 \le i \le k$, $0 \le v \le g-1$, $1 \le i' \le \binom{k}{t}$, $0 \le w \le g^t-1$)．命題変数 $p(x_{r,i}=v)$ は，基本行列で $x_{r,i}=v$ が成り立つことを意味する．同様に，命題変数 $p(y_{r,i'}=w)$ は，拡張行列で $y_{r,i'}=w$ が成り立つことを意味する．

被覆配列 $CA(b;t,k,g)$ は，以下の節に変換される (ただし，$1 \le r \le b$, $1 \le i \le k$, $0 \le v < v' \le g-1$, $1 \le i' \le \binom{k}{t}$, $0 \le w < w' \le g^t-1$．節 (6) は，$y_{r,i'}=w$ と $x_{r,i}=v$ が整合する $r$,$i$,$i'$,$v$,$w$ に対してのみ生成される)．

$$\bigvee_v p(x_{r,i}=v) \tag{1}$$
$$\neg p(x_{r,i}=v) \vee \neg p(x_{r,i}=v') \tag{2}$$
$$\bigvee_w p(y_{r,i'}=w) \tag{3}$$
$$\neg p(y_{r,i'}=w) \vee \neg p(y_{r,i'}=w') \tag{4}$$
$$\bigvee_r p(y_{r,i'}=w) \tag{5}$$
$$\neg p(y_{r,i'}=w) \vee p(x_{r,i}=v) \tag{6}$$

節 (1,3) は，それぞれ $x_{r,i}$ と $y_{r,i'}$ が少なくとも一つの値をとることを表す (at-least-one 節)．節 (2,4) は，それぞれ $x_{r,i}$ と $y_{r,i'}$ が同時に二つ以上の値をとらないことを表す (at-most-one 節)．節 (5) は，拡張行列の各列中に $0$〜$g^t-1$ の値が少なくとも一つ現れ

ること (カバレッジ制約) を表す．節 (6) は，チャネリング制約を表す．

Hnich 符号化法は，支持符号化法に基づいている．第 2.2 節の CSP 表現と異なる点は，カバレッジ制約の上限を表す節がないことである．この制約は SAT で表現することは可能だが困難であるという点から省略されている．また，節 (1,3,4) は省略可能である (詳しくは論文 [15] [16] を参照)．

Hnich らは，本節で述べた符号化法と確率的 SAT ソルバーを組合せることによって，強さ $2 \leq t \leq 4$ および $k, g$ の小さな値に対する $CA$ 最適化問題に対して，様々な既存手法とほぼ同等の結果を得ることに成功している．彼らが見つけた上限 $CAN(3,7,3) \leq 40$ は現在でも最良値である．

しかしながら，確率的 SAT ソルバーに適した符号化法が，CDCL SAT ソルバーに適しているとは限らない．本稿では，CDCL SAT ソルバーに適した SAT 符号化法を二つ提案する．

## 4 順序符号化法

### 4.1 順序符号化法の概要

順序符号化法は，整数有限領域上の CSP を SAT 問題に変換する方法の一つである．順序符号化法は，Crawford と Baker によりジョブショップ・スケジューリング問題に適用された方法 [8] [17] [23] を CSP に適用可能なように一般化したものである．ショップ・スケジューリング問題および 2 次元ストリップパッキング問題で未知だった最適値決定に成功する等，有望な手法であることが示されている [30] [21] [29]．

順序符号化法では，各整数変数 $x$ について，そのドメインが $\{a_1, a_2, \ldots, a_n\}$ の時 (ただし $a_1 < a_2 < \ldots < a_n$)，$x \leq a_i$ を表す $n-1$ 個の命題変数 $p(x \leq a_1)$, $p(x \leq a_2)$, $\ldots$, $p(x \leq a_{n-1})$ を用いる．なお，$x \leq a_n$ は常に真であるため，命題変数 $p(x \leq a_n)$ は不要である．また，これらの命題変数間の関係を表す以下の節を用いる．

$$\neg p(x \leq a_i) \vee p(x \leq a_{i+1}) \quad (1 \leq i \leq n-2)$$

例えば，変数 $x$ のドメインが $\{0,1,2\}$ の場合，二つの命題変数 $p(x \leq 0)$, $p(x \leq 1)$ を用い，以下の節を追加する．

$$\neg p(x \leq 0) \vee p(x \leq 1)$$

この時，上記の節を充足可能にする真理値割り当ては 3 通りあり，それぞれ $x = 0$, $x = 1$, $x = 2$ に対応する．

| $p(x \leq 0)$ | $p(x \leq 1)$ | 解釈 |
|---|---|---|
| 1 | 1 | $x = 0$ |
| 0 | 1 | $x = 1$ |
| 0 | 0 | $x = 2$ |

制約については，制約に違反する点ではなく違反する範囲を符号化する．すなわち，範囲 $a_1 < x_1 \leq b_1$, $\ldots$, $a_n < x_n \leq b_n$ 中のすべての点 $(x_1, \ldots, x_n)$ が制約に違反する時，以下の節を追加する．

$$p(x_1 \leq a_1) \vee \neg p(x_1 \leq b_1) \vee$$
$$\cdots \vee p(x_n \leq a_n) \vee \neg p(x_n \leq b_n)$$

線形式を用いた線形制約については，より簡潔な符号化が可能である [32]．今，$a_i$ を非零の整数定数，$c$ を整数定数，$x_i$ を互いに異なる整数変数とする．この時，制約 $\sum_{i=1}^{n} a_i x_i \leq c$ は以下のように符号化できる．

$$\bigwedge_{b_i} \bigvee_{i} (a_i x_i \leq b_i)^{\#}$$

ここで $b_i$ は，$\sum_{i=1}^{n} b_i = c - n + 1$ を満たすように動くとし，変換 $()^{\#}$ は以下のように定義する．

$$(a\,x \leq b)^{\#} \quad \equiv \quad \begin{cases} p(x \leq \lfloor b/a \rfloor) & (a > 0) \\ \neg p(x \leq \lceil b/a \rceil - 1) & (a < 0) \end{cases}$$

ただし，$x$ の取り得る最小値未満の $a$ については $p(x \leq a)$ を偽に変換し，最大値以上については真に変換する．

例えば，整数変数 $x, y$ のドメインが $\{0,1,2\}$ の時，制約 $x - y \leq -1$ は以下の三つの節に符号化される．

$$\neg p(y \leq 0)$$
$$p(x \leq 0) \vee \neg p(y \leq 1)$$
$$p(x \leq 1)$$

ここで，$p(x \leq 0) \vee \neg p(y \leq 1)$ は，「$x \leq 0$ または $y > 1$」であること，すなわち「$x \geq 1$ かつ $y \leq 1$」が制約に違反する領域であることを表している．

### 4.2 $CA$ の順序符号化

Hnich の CSP 表現を，順序符号化法を用いて SAT 問題に変換する方法について述べる．

順序符号化法では，命題変数 $p(x_{r,i} \le v)$ と $p(y_{r,i'} \le w)$ を導入する[†4](ただし，$1 \le r \le b$, $1 \le i \le k$, $-1 \le v \le g-1$, $1 \le i' \le \binom{k}{t}$, $-1 \le w \le g^t - 1$).

被覆配列 $CA(b;t,k,g)$ は，チャネリング制約を除いて，以下の節に変換される (ただし，$1 \le r \le b$, $1 \le i \le k$, $0 \le v \le g-1$, $1 \le i' \le \binom{k}{t}$, $0 \le w \le g^t - 1$).

$$\neg p(x_{r,i} \le -1) \tag{7}$$
$$p(x_{r,i} \le g-1) \tag{8}$$
$$\neg p(x_{r,i} \le v-1) \vee p(x_{r,i} \le v) \tag{9}$$
$$\neg p(y_{r,i'} \le -1) \tag{10}$$
$$p(y_{r,i'} \le g^t - 1) \tag{11}$$
$$\neg p(y_{r,i'} \le w-1) \vee p(y_{r,i'} \le w) \tag{12}$$
$$\bigvee_r (\neg p(y_{r,i'} \le w-1) \wedge p(y_{r,i'} \le w)) \tag{13}$$

節 (7,8,9) と節 (10,11,12) は，それぞれ基本行列と拡張行列における整数変数の上下限・順序関係を表す．明らかに式 (13) は CNF 式ではない．そのため，この式は **Tseitin 変換**を用いて充足同値 (equi-satisfiable) な CNF 式に変換される．

チャネリング制約 $y_{r,i'} = \sum_{\ell=1}^{t} g^{t-\ell} x_{r,c_\ell^{i'}}$ は，まず以下の線形比較の連言に置換えられる．

$$\left( y_{r,i'} \le \sum_{\ell=1}^{t} g^{t-\ell} x_{r,c_\ell^{i'}} \right) \wedge \left( y_{r,i'} \ge \sum_{\ell=1}^{t} g^{t-\ell} x_{r,c_\ell^{i'}} \right).$$

その後，各線形比較は前節で述べた $\sum_{i=1}^{n} a_i\, x_i \le c$ と同じ方法で変換される．

この符号化の欠点は，カバレッジ制約に必要な節の数が大きくなる点である．すなわち，式 (13) に対して，Tseitin 変換によって新しく $O(b)$ の命題変数が導入され，全体として $O(b\binom{k}{t}g^t)$ の節が必要となる．この問題を回避するために，新しく混合符号化法を提案する．

†4 $p(x_{r,i} \le -1)$, $p(y_{r,i'} \le -1)$, $p(x_{r,i} \le g-1)$, $p(y_{r,i'} \le g^t - 1)$ は冗長である．すなわち，最初の二つは常に偽，後の二つは常に真である．しかし，符号化の説明を簡潔にするためにそのまま用いる．

## 5 混合符号化法

混合符号化法は，順序符号化法と Hnich 符号化法の融合である．基本的アイデアは，基本行列には順序符号化法を用い，拡張行列には Hnich 符号化法を用いる点にある．これにより，Tseitin 変換による節数の増加を回避できる．

混合符号化法では，命題変数 $p(x_{r,i} \le v)$ と $p(y_{r,i'} = w)$ を導入する (ただし，$1 \le r \le b$, $1 \le i \le k$, $-1 \le v \le g-1$, $1 \le i' \le \binom{k}{t}$, $0 \le w \le g^t - 1$).

被覆配列 $CA(b;t,k,g)$ は，以下のように変換される (ただし，$1 \le r \le b$, $1 \le i \le k$, $0 \le v \le g-1$, $1 \le i' \le \binom{k}{t}$, $0 \le w < w' \le g^t - 1$. 節 (20,21) は，$y_{r,i'} = w$ と $x_{r,i} = v$ (すなわち，$x_{r,i} \ge v \wedge x_{r,i} \le v$) が整合する $r$,$i$,$i'$,$v$,$w$ に対してのみ生成される).

$$\neg p(x_{r,i} \le -1) \tag{14}$$
$$p(x_{r,i} \le g-1) \tag{15}$$
$$\neg p(x_{r,i} \le v-1) \vee p(x_{r,i} \le v) \tag{16}$$
$$\bigvee_w p(y_{r,i'} = w) \tag{17}$$
$$\neg p(y_{r,i'} = w) \vee \neg p(y_{r,i'} = w') \tag{18}$$
$$\bigvee_r p(y_{r,i'} = w) \tag{19}$$
$$\neg p(y_{r,i'} = w) \vee \neg p(x_{r,i} \le v-1) \tag{20}$$
$$\neg p(y_{r,i'} = w) \vee p(x_{r,i} \le v) \tag{21}$$

節 (14,15,16) は順序符号化法の節 (7,8,9) と同じである．節 (17,18,19) は Hnich 符号化法の節 (3,4,5) と同じである．チャネリング制約は節 (20,21) によって表される．これらは，Hnich 符号化法の節 (6) を順序符号化法の変数に適合するように変更したものである．

上記の節 (14)〜(21) のうち，節 (17,18) は省略可能である．節 (18) は導出原理を用いて節 (16,20,21) から導出可能である．例えば，$CA(11;2,5,3)$ の場合，$\neg p(y_{r,(i,j)} = 0) \vee \neg p(y_{r,(i,j)} = 2)$ は，$\neg p(x_{r,j} \le 0) \vee p(x_{r,j} \le 1)$, $\neg p(y_{r,(i,j)} = 0) \vee p(x_{r,j} \le 0)$, $\neg p(y_{r,(i,j)} = 2) \vee \neg p(x_{r,j} \le 1)$ から導出される．

節 (17,18) を省略しても，SAT の解をデコードすることにより，$CA$ の解を得ることができる．なぜならば，任意の SAT の解に対して，節 (19) は，拡張行

**表 2** $CA(b;t,k,g)$ **に対する節数の比較**

| | Hnich 符号化法 | 順序符号化法 | 混合符号化法 |
|---|---|---|---|
| 基本行列 | $\{bk\}+bk\binom{g}{2}$ | $bk(g-1)$ | $bk(g-1)$ |
| 拡張行列 | $\left\{b\binom{k}{t}+b\binom{k}{t}\binom{g^t}{2}\right\}$ | $b\binom{k}{t}(g^t-1)$ | $\left\{b\binom{k}{t}+b\binom{k}{t}\binom{g^t}{2}\right\}$ |
| カバレッジ制約 | $\binom{k}{t}g^t$ | $O(b\binom{k}{t}g^t)$ | $\binom{k}{t}g^t$ |
| チャネリング制約 | $b\binom{k}{t}g^t t$ | $O(b\binom{k}{t}g^t)$ | $O(b\binom{k}{t}g^t t)$ |

列の各列において $0$〜$g^t-1$ の値が少なくとも一回出現することを保証する (カバレッジ制約)．$0$〜$g^t-1$ の各出現に対して，対応する基本行列の $t$ 個の値が節 (20,21) から導かれる．基本行列の各要素がただ一つの値を取ることは，節 (14,15,16) によって保証される．

最後に，Hnich 符号化法，順序符号化法，混合符号化法によって生成される $CA(b;t,k,g)$ の節数を表 2 †5に示す．括弧 “{}” で囲まれた数字は，省略可能な節であることを示している．

## 6 実行実験

提案手法の有効性を評価するために，Hnich 符号化法，順序符号化法，混合符号化法の比較実験を行った．実験内容および実験環境は，以下の通りである．

- ベンチマーク問題として，強さ $2 \le t \le 6$，および $k, g$ の小さな値に対する $CA$ 最適化問題 (97 問) を用いる．
- 各最適化問題に対して，$CA(b;t,k,g)$ の $b$ を変換させた複数の $CA$ 判定問題群を生成し，SAT 問題に符号化する．
- 符号化された $CA$ 判定問題群は，充足可能 (SAT) と充足不能 (UNSAT) の両方の問題を含む．最適値 (最小の $b$) は UNSAT と SAT の境界に位置する．
- どの符号化法に対しても，“行と列に関する対称性 [11]” と “値に関する対称性 [15] [16]” を取り除くための制約を追加する．
- 混合符号化法については節 (17,18) を，Hnich 符号化法については節 (3,4) を省略する．
- CDCL SAT ソルバーとしては，MiniSat [10] を用いる．
- すべての CPU 時間は，Linux マシン (Intel Xeon 3.00GHz，8GB メモリ) 上で計測し，各 $CA$ 判定問題に対する MiniSat のタイムアウト ($T.O$) は 1800 秒である．ただし，$CA(14;3,12,2)$ のタイムアウトのみ 12 時間に設定している．

まず最初に，符号化された $CA$ 判定問題を解くのに MiniSat が要した CPU 時間を示す (表 3–4 参照，単位は秒)．サイズ $b$ に関しては，得られた最良の上限 (あるいは下限) のみ記載する．記号 “$*$” は，その値が最適値であることを示す．また，各問題に対して，一番良い CPU 時間をボールド体で記している．

順序符号化法と混合符号化法は，47 問に対して既知の最適値を再生成することができた．それに対して，Hnich 符号化法は 29 問の最適値を得るにとどまった．さらに，順序符号化法と混合符号化法は，$CAN(3,12,2)$ と $CAN(6,8,2)$ の 2 問に対して，既知の最良値の最適性を証明することに成功した．これまで，$CAN(3,12,2)$ の上下限は，$14 \le CAN(3,12,2) \le 15$ であった．しかし，表 4 より，$CA(14;3,12,2)$ が UNSAT，つまり解がないことがわかる．よって，$CAN(3,12,2)=15$ であることが証明できた．この $CAN(3,12,2)=15$ は，2009 年に出版された Handbook of Satisfiability [36] において未解決問題として挙げられている問題の解である．加えて，順序符号化法と混合符号化法は，いくつかの $CA$ 最適化問題に対して，既知の下限 [2] を更新することに成功した．表 5 に提案手法によって得られた新しい結果を示す．

†5 順序符号化法と混合符号化法について，$p(x_{r,i} \le -1)$, $p(x_{r,i} \le g-1)$, $p(y_{r,i'} \le -1)$, $p(y_{r,i'} \le g^t-1)$ を含む冗長な節の数は無視する．

**表 5 新しく得られた結果：$80 \leq CAN(3,8,4)$ と $15 \leq CAN(3,k,2)$ は，実行実験で得られた結果 $20 \leq CAN(2,7,4)$ と $CAN(3,12,2) = 15$ に対して，定理 1 の (4) 及び (2) を適用した結果である．**

| 新しい結果 | これまでの結果 |
|---|---|
| $\mathbf{20} \leq CAN(2,7,4) \leq 21$ | $19 \leq CAN(2,7,4) \leq 21$ |
| $\mathbf{80} \leq CAN(3,8,4) \leq 88$ | $76 \leq CAN(3,8,4) \leq 88$ |
| $CAN(3,12,2) = \mathbf{15}$ | $14 \leq CAN(3,12,2) \leq 15$ |
| $\mathbf{15} \leq CAN(3,k,2)$ $(k \geq 13)$ | $14 \leq CAN(3,k,2)$ $(k \geq 13)$ |
| $\mathbf{50} \leq CAN(5,9,2) \leq 54$ | $48 \leq CAN(5,9,2) \leq 54$ |
| $CAN(6,8,2) = \mathbf{85}$ | $84 \leq CAN(6,8,2) \leq 85$ |

順序符号化法と混合符号化法が最適値を決定した 49 問は，すべて同じ問題である．最適値を決定できなかった $97 - 49 = 48$ 問に対して，最良値を求めた問題数を比較すると，順序符号化法が 35 問，混合符号化法が 39 問となり，混合符号化法は，順序符号化法より 4 問多く最良値を生成することができた．

次に，提案手法で得られた $CA$ 最適化問題の最適値および最良値を，以下の様々な既存手法と比較する．

- CSP+ILOG [16]
  第 2.2 節で述べた CSP 表現を，高速な CSP ソルバー ILOG で求解．
- Hnich 符号化法 +walksat [16]
  第 3 節で述べた Hnich 符号化法を，確率的 SAT ソルバー walksat の改良版で求解．
- Colbourn の $CA$ 表 [7]
  直交表，群論等の数学的手法，貪欲法，局所探索法など様々な手法を用いて得られた結果．

表 6–7 に比較結果を示す．各 $CAN(t,k,g)$ に対して，最も良い値をボールド体で記している．記号 "-" は，その結果が論文等で入手できなかったことを示す．

提案手法は，Colbourn の $CA$ 表と比較して，強さ $2 \leq t \leq 6$ および $k, g$ の小さな値の $CAN(t,k,g)$ に対して，ほぼ同等の結果を得た．また，$CAN(2,k,g)$ と $CAN(3,k,3)$ のいくつかの問題について，Hnich 符号化法の結果に劣っているものの，総合的には多くの問題に対してより良い結果を得た．

最後に，実際に生成された節数について簡単に触れる．第 5 節において，混合符号化法では節 (17,18) が省略可能であることを述べた．例として $CA(15;3,12,2)$ では，節を省略した場合の節数は 77,442 であり，省略しない場合は 175,826 であった．

## 7 まとめ

本稿では，$CA$ 最適化問題に対して，CDCL SAT ソルバーに適した二つの新しい SAT 符号化方法を提案した．強さ $2 \leq t \leq 6$ の小中規模の問題に対する実行実験を行った結果，提案手法は様々な既存手法 (群論等の数学的手法，貪欲法，局所探索法など) と比較して，ほぼ同等の結果を得ることができた．さらに，2009 年に出版された Handbook of Satisfiability [36] 中に記載されている未解決問題に対して，既知の最良値が最適値であることを証明した．加えて，いくつかの問題に対して，既知の下限 [2] を更新することにも成功した (表 5 参照)．

## 参 考 文 献

[1] Bulutoglu, D. and Margot, F.: Classification of orthogonal arrays by integer programming, *Journal of Statistical Planning and Inference*, Vol. 138(2008), pp. 654–666.

[2] Chateauneuf, M. A. and Kreher, D. L.: On the State of Strength-Three Covering Arrays, *Journal of Combinatorial Designs*, Vol. 10, No. 4(2002), pp. 217–238.

[3] Cohen, D. M., Dalal, S. R., Fredman, M. L., and Patton, G. C.: The AETG System: An Approach to Testing Based on Combinatiorial Design, *IEEE Transactions on Software Engineering*, Vol. 23, No. 7(1997), pp. 437–444.

[4] Cohen, M. B., Dwyer, M. B., and Shi, J.: Constructing Interaction Test Suites for Highly-Configurable Systems in the Presence of Constraints: A Greedy Approach, *IEEE Trans. Software Eng.*, Vol. 34, No. 5(2008), pp. 633–650.

[5] Cohen, M. B., Gibbons, P. B., Mugridge, W. B., and Colbourn, C. J.: Constructing Test Suites

**表 3　ベンチマーク結果：**$CA(b;2,k,g)$

| $t$ | $k$ | $g$ | $b$ | SAT/UNSAT | Hnich | 順序 | 混合 |
|---|---|---|---|---|---|---|---|
| 2 | 3 | 3 | **9*** | SAT | **0.00** | 0.01 | **0.00** |
| 2 | 4 | 3 | **9*** | SAT | 0.01 | **0.00** | 0.01 |
| 2 | 5 | 3 | 10 | UNSAT | 0.59 | **0.02** | **0.02** |
| 2 | 5 | 3 | **11*** | SAT | 0.03 | 0.03 | **0.01** |
| 2 | 6 | 3 | 11 | UNSAT | 7.55 | **0.04** | 0.05 |
| 2 | 6 | 3 | **12*** | SAT | 0.03 | **0.02** | 0.03 |
| 2 | 7 | 3 | **12*** | SAT | **0.05** | 0.38 | 0.32 |
| 2 | 8 | 3 | 13 | SAT | **4.22** | 1263.58 | 263.76 |
| 2 | 9 | 3 | 13 | SAT | 760.05 | **228.39** | *T.O* |
| 2 | 10 | 3 | 14 | SAT | 518.18 | 79.75 | **14.60** |
| 2 | 11 | 3 | 15 | SAT | 0.15 | 0.77 | **0.13** |
| 2 | 12 | 3 | 15 | SAT | 1.31 | 0.49 | **0.21** |
| 2 | 13 | 3 | 15 | SAT | **16.40** | 18.53 | 30.60 |
| 2 | 14 | 3 | 15 | SAT | 372.95 | **192.93** | 373.64 |
| 2 | 15 | 3 | 16 | SAT | 155.21 | 31.71 | **30.80** |
| 2 | 16 | 3 | 16 | SAT | 743.74 | 1674.69 | **390.72** |
| 2 | 3 | 4 | **16*** | SAT | 0.02 | **0.01** | **0.01** |
| 2 | 4 | 4 | **16*** | SAT | 0.04 | 0.04 | **0.02** |
| 2 | 5 | 4 | **16*** | SAT | 0.05 | 0.05 | **0.04** |
| 2 | 6 | 4 | 18 | UNSAT | *T.O* | **4.31** | 9.90 |
| 2 | 6 | 4 | **19*** | SAT | 21.28 | **0.87** | 3.14 |
| 2 | 7 | 4 | 19 | UNSAT | *T.O* | 177.23 | **174.89** |
| 2 | 7 | 4 | 22 | SAT | 71.18 | 20.71 | **4.66** |
| 2 | 8 | 4 | 23 | SAT | 550.61 | 22.89 | **3.39** |
| 2 | 9 | 4 | 24 | SAT | *T.O* | **230.44** | 469.66 |
| 2 | 10 | 4 | 25 | SAT | *T.O* | 440.30 | **254.51** |
| 2 | 11 | 4 | 26 | SAT | *T.O* | *T.O* | **884.22** |
| 2 | 3 | 5 | **25*** | SAT | 0.08 | **0.05** | 0.06 |
| 2 | 4 | 5 | **25*** | SAT | 0.26 | 0.18 | **0.14** |
| 2 | 5 | 5 | **25*** | SAT | 0.47 | 0.46 | **0.15** |
| 2 | 6 | 5 | **25*** | SAT | 3.21 | **0.51** | 0.76 |
| 2 | 7 | 5 | 29 | SAT | *T.O* | 394.38 | **90.50** |
| 2 | 8 | 5 | 36 | SAT | *T.O* | **654.95** | *T.O* |
| 2 | 9 | 5 | 38 | SAT | *T.O* | 914.78 | **279.84** |
| 2 | 10 | 5 | 41 | SAT | *T.O* | *T.O* | **386.87** |
| 2 | 11 | 5 | 42 | SAT | *T.O* | **1330.56** | *T.O* |
| 2 | 3 | 6 | **36*** | SAT | 0.46 | 0.16 | **0.14** |
| 2 | 4 | 6 | 37 | SAT | 22.72 | **4.19** | 4.24 |
| 2 | 5 | 6 | 40 | SAT | *T.O* | **627.67** | *T.O* |
| 2 | 6 | 6 | 45 | SAT | *T.O* | **614.53** | *T.O* |
| 2 | 7 | 6 | 50 | SAT | *T.O* | 1765.78 | **1110.20** |
| 2 | 8 | 6 | 52 | SAT | *T.O* | *T.O* | **1236.90** |
| 2 | 9 | 6 | 57 | SAT | *T.O* | *T.O* | **773.33** |
| 2 | 10 | 6 | 62 | SAT | *T.O* | *T.O* | **407.40** |
| 2 | 11 | 6 | 63 | SAT | *T.O* | *T.O* | **1287.25** |
| 2 | 3 | 7 | **49*** | SAT | 1.67 | **0.64** | 0.66 |
| 2 | 4 | 7 | **49*** | SAT | 460.07 | **98.10** | 171.55 |
| 2 | 5 | 7 | 57 | SAT | *T.O* | *T.O* | **570.96** |
| 2 | 6 | 7 | 65 | SAT | *T.O* | **1513.01** | *T.O* |
| 2 | 7 | 7 | 68 | SAT | *T.O* | *T.O* | **1446.27** |
| 2 | 8 | 7 | 72 | SAT | *T.O* | *T.O* | **1362.44** |
| 2 | 9 | 7 | 81 | SAT | *T.O* | *T.O* | **1098.09** |
| 2 | 10 | 7 | 88 | SAT | *T.O* | *T.O* | **442.81** |
| 2 | 11 | 7 | 93 | SAT | *T.O* | *T.O* | **449.59** |

**表 4　ベンチマーク結果：**$CA(b;t,k,g)\quad(3 \le t \le 6)$

| $t$ | $k$ | $g$ | $b$ | SAT/UNSAT | Hnich | 順序 | 混合 |
|---|---|---|---|---|---|---|---|
| 3 | 4 | 2 | **8*** | SAT | **0.00** | **0.00** | **0.00** |
| 3 | 5 | 2 | 9 | UNSAT | **0.01** | **0.01** | **0.01** |
| 3 | 5 | 2 | **10*** | SAT | **0.00** | **0.00** | **0.00** |
| 3 | 6 | 2 | 11 | UNSAT | 0.30 | 0.02 | **0.01** |
| 3 | 6 | 2 | **12*** | SAT | 0.02 | 0.01 | **0.00** |
| 3 | 7 | 2 | **12*** | SAT | 0.02 | 0.03 | **0.01** |
| 3 | 8 | 2 | **12*** | SAT | 0.03 | 0.04 | **0.01** |
| 3 | 9 | 2 | **12*** | SAT | 0.05 | 0.07 | **0.03** |
| 3 | 10 | 2 | **12*** | SAT | 0.08 | 0.12 | **0.05** |
| 3 | 11 | 2 | **12*** | SAT | 0.13 | 0.15 | **0.05** |
| 3 | 12 | 2 | 14 | UNSAT | *T.O* | **5607.25** | 6228.16 |
| 3 | 12 | 2 | **15*** | SAT | 0.61 | 0.89 | **0.44** |
| 3 | 13 | 2 | 16 | SAT | 24.91 | 7.57 | **4.24** |
| 3 | 14 | 2 | 16 | SAT | *T.O* | **15.09** | 23.68 |
| 3 | 15 | 2 | 17 | SAT | *T.O* | 435.35 | **1.97** |
| 3 | 16 | 2 | 17 | SAT | *T.O* | 86.07 | **14.93** |
| 3 | 17 | 2 | 20 | SAT | *T.O* | **62.40** | 125.27 |
| 3 | 18 | 2 | 21 | SAT | **2.46** | 44.99 | 35.62 |
| 3 | 19 | 2 | 22 | SAT | **1.54** | 176.79 | 4.16 |
| 3 | 4 | 3 | **27*** | SAT | 0.07 | 0.05 | **0.04** |
| 3 | 5 | 3 | 32 | UNSAT | *T.O* | **16.07** | 27.85 |
| 3 | 5 | 3 | **33*** | SAT | 632.27 | **2.81** | 16.85 |
| 3 | 6 | 3 | **33*** | SAT | *T.O* | 15.46 | **12.44** |
| 3 | 7 | 3 | 46 | SAT | *T.O* | **1180.95** | *T.O* |
| 3 | 8 | 3 | 52 | SAT | **407.37** | 1148.09 | *T.O* |
| 3 | 9 | 3 | 56 | SAT | *T.O* | *T.O* | **202.78** |
| 3 | 4 | 4 | **64*** | SAT | 9.00 | 0.68 | **0.56** |
| 3 | 5 | 4 | **64*** | SAT | *T.O* | 2.01 | **1.35** |
| 3 | 6 | 4 | **64*** | SAT | *T.O* | 3.13 | **1.54** |
| 3 | 4 | 5 | **125*** | SAT | *T.O* | **6.13** | 6.99 |
| 3 | 5 | 5 | **125*** | SAT | *T.O* | 86.51 | **54.96** |
| 3 | 6 | 5 | **125*** | SAT | *T.O* | 177.13 | **59.09** |
| 4 | 5 | 2 | **16*** | SAT | 0.02 | **0.01** | **0.01** |
| 4 | 6 | 2 | 20 | UNSAT | *T.O* | 0.07 | **0.05** |
| 4 | 6 | 2 | **21*** | SAT | 0.36 | 0.08 | **0.03** |
| 4 | 7 | 2 | 23 | UNSAT | *T.O* | 0.35 | **0.32** |
| 4 | 7 | 2 | **24*** | SAT | 529.12 | 0.68 | **0.37** |
| 4 | 8 | 2 | **24*** | SAT | 107.13 | 0.79 | **0.63** |
| 4 | 9 | 2 | **24*** | SAT | 399.55 | **1.13** | 1.20 |
| 4 | 10 | 2 | **24*** | SAT | 279.46 | 6.47 | **0.96** |
| 4 | 11 | 2 | **24*** | SAT | 26.20 | 4.66 | **1.92** |
| 4 | 12 | 2 | **24*** | SAT | 1573.40 | 11.28 | **2.12** |
| 4 | 13 | 2 | 36 | SAT | *T.O* | **372.87** | 734.72 |
| 4 | 5 | 3 | **81*** | SAT | 840.28 | **1.13** | 1.52 |
| 4 | 5 | 4 | **256*** | SAT | *T.O* | 105.37 | **104.00** |
| 5 | 6 | 2 | **32*** | SAT | 4.66 | 0.13 | **0.11** |
| 5 | 7 | 2 | 41 | UNSAT | *T.O* | 1.53 | **1.10** |
| 5 | 7 | 2 | **42*** | SAT | 849.79 | 1.41 | **1.24** |
| 5 | 8 | 2 | 52 | SAT | *T.O* | **95.01** | 134.37 |
| 5 | 9 | 2 | 49 | UNSAT | *T.O* | **344.65** | 521.59 |
| 5 | 9 | 2 | 54 | SAT | *T.O* | 1431.35 | **1097.27** |
| 5 | 10 | 2 | 60 | SAT | *T.O* | **550.67** | *T.O* |
| 5 | 6 | 3 | **243*** | SAT | *T.O* | **156.64** | 197.40 |
| 6 | 7 | 2 | **64*** | SAT | 922.29 | **2.97** | 3.42 |
| 6 | 8 | 2 | 84 | UNSAT | *T.O* | 86.91 | **52.94** |
| 6 | 8 | 2 | **85*** | SAT | *T.O* | 76.28 | **48.49** |

for Interaction Testing, *Proceedings of the 25th International Conference on Software Engineering (ICSE 2003)*, 2003, pp. 38–48.

[ 6 ] Colbourn, C. J.: Combinatorial Aspects of Covering Arrays, *Le Matematiche (Catania)*, Vol. 58(2004), pp. 121–167.

[ 7 ] Colbourn, C. J.: Covering Array Tables, `http://www.public.asu.edu/~ccolbou/src/tabby/catable.html`, 2010. Last Accessed on April 26th 2010.

[ 8 ] 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.

[ 9 ] de Kleer, J.: A Comparison of ATMS and CSP Techniques, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI*

**表 6　比較結果：**$CAN(2,k,g)$

| $t$ | $k$ | $g$ | 提案手法 + MiniSat | Hnich+ ILOG [16] | Colbourn $CA$ 表 [7] |
|---|---|---|---|---|---|
| 2 | 3 | 3 | **9** | **9** | **9** |
| 2 | 4 | 3 | **9** | **9** | **9** |
| 2 | 5 | 3 | **11** | **11** | **11** |
| 2 | 6 | 3 | **12** | **12** | **12** |
| 2 | 7 | 3 | **12** | **12** | **12** |
| 2 | 8 | 3 | **13** | 14 | **13** |
| 2 | 9 | 3 | **13** | **13** | **13** |
| 2 | 10 | 3 | **14** | **14** | **14** |
| 2 | 11 | 3 | **15** | **15** | **15** |
| 2 | 12 | 3 | **15** | – | **15** |
| 2 | 13 | 3 | **15** | – | **15** |
| 2 | 14 | 3 | **15** | – | **15** |
| 2 | 15 | 3 | 16 | – | **15** |
| 2 | 16 | 3 | 16 | – | **15** |
| 2 | 3 | 4 | **16** | **16** | **16** |
| 2 | 4 | 4 | **16** | **16** | **16** |
| 2 | 5 | 4 | **16** | **16** | **16** |
| 2 | 6 | 4 | **19** | **19** | **19** |
| 2 | 7 | 4 | 22 | **21** | **21** |
| 2 | 8 | 4 | 23 | 23 | **22** |
| 2 | 9 | 4 | 24 | 24 | **23** |
| 2 | 10 | 4 | 25 | 25 | **24** |
| 2 | 11 | 4 | 26 | 25 | **24** |
| 2 | 3 | 5 | **25** | **25** | **25** |
| 2 | 4 | 5 | **25** | **25** | **25** |
| 2 | 5 | 5 | **25** | **25** | **25** |
| 2 | 6 | 5 | **25** | **25** | **25** |
| 2 | 7 | 5 | **29** | **29** | **29** |
| 2 | 8 | 5 | 36 | 34 | **33** |
| 2 | 9 | 5 | 38 | **35** | **35** |
| 2 | 10 | 5 | 41 | 38 | **36** |
| 2 | 11 | 5 | 42 | 39 | **38** |
| 2 | 3 | 6 | **36** | **36** | **36** |
| 2 | 4 | 6 | **37** | **37** | **37** |
| 2 | 5 | 6 | 40 | **39** | **39** |
| 2 | 6 | 6 | 45 | 42 | **41** |
| 2 | 7 | 6 | 50 | 45 | **42** |
| 2 | 8 | 6 | 52 | 48 | **42** |
| 2 | 9 | 6 | 57 | 51 | **46** |
| 2 | 10 | 6 | 62 | 53 | **49** |
| 2 | 11 | 6 | 63 | 55 | **52** |
| 2 | 3 | 7 | **49** | **49** | **49** |
| 2 | 4 | 7 | **49** | **49** | **49** |
| 2 | 5 | 7 | 57 | 52 | **49** |
| 2 | 6 | 7 | 65 | 58 | **49** |
| 2 | 7 | 7 | 68 | 61 | **49** |
| 2 | 8 | 7 | 72 | 63 | **49** |
| 2 | 9 | 7 | 81 | 66 | **59** |
| 2 | 10 | 7 | 88 | 71 | **61** |
| 2 | 11 | 7 | 93 | 73 | **67** |

**表 7　比較結果：**$CAN(t,k,g)$　$(3 \le t \le 6)$

| $t$ | $k$ | $g$ | 提案手法 + MiniSat | CSP+ ILOG [16] | Hnich+ walksat [16] | Colbourn $CA$ 表 [7] |
|---|---|---|---|---|---|---|
| 3 | 4 | 2 | **8** | **8** | – | **8** |
| 3 | 5 | 2 | **10** | **10** | – | **10** |
| 3 | 6 | 2 | **12** | **12** | – | **12** |
| 3 | 7 | 2 | **12** | **12** | – | **12** |
| 3 | 8 | 2 | **12** | **12** | – | **12** |
| 3 | 9 | 2 | **12** | **12** | **12** | **12** |
| 3 | 10 | 2 | **12** | **12** | **12** | **12** |
| 3 | 11 | 2 | **12** | **12** | **12** | **12** |
| 3 | 12 | 2 | **15** | – | **15** | **15** |
| 3 | 13 | 2 | **16** | – | **16** | **16** |
| 3 | 14 | 2 | **16** | – | 17 | **16** |
| 3 | 15 | 2 | **17** | – | 18 | **17** |
| 3 | 16 | 2 | **17** | – | 18 | **17** |
| 3 | 17 | 2 | 20 | – | **18** | **18** |
| 3 | 18 | 2 | 21 | – | 20 | **18** |
| 3 | 19 | 2 | 22 | – | – | **18** |
| 3 | 4 | 3 | **27** | – | – | **27** |
| 3 | 5 | 3 | **33** | – | **33** | **33** |
| 3 | 6 | 3 | **33** | – | **33** | **33** |
| 3 | 7 | 3 | 46 | – | **40** | **40** |
| 3 | 8 | 3 | 52 | – | 46 | **42** |
| 3 | 9 | 3 | 56 | – | 51 | **45** |
| 3 | 4 | 4 | **64** | – | – | **64** |
| 3 | 5 | 4 | **64** | – | – | **64** |
| 3 | 6 | 4 | **64** | – | – | **64** |
| 3 | 4 | 5 | **125** | – | – | **125** |
| 3 | 5 | 5 | **125** | – | – | **125** |
| 3 | 6 | 5 | **125** | – | – | **125** |
| 4 | 5 | 2 | **16** | **16** | – | **16** |
| 4 | 6 | 2 | **21** | **21** | – | **21** |
| 4 | 7 | 2 | **24** | – | **24** | **24** |
| 4 | 8 | 2 | **24** | – | **24** | **24** |
| 4 | 9 | 2 | **24** | – | **24** | **24** |
| 4 | 10 | 2 | **24** | – | **24** | **24** |
| 4 | 11 | 2 | **24** | – | – | **24** |
| 4 | 12 | 2 | **24** | – | – | **24** |
| 4 | 13 | 2 | 36 | – | – | **32** |
| 4 | 5 | 3 | **81** | – | **81** | **81** |
| 4 | 5 | 4 | **256** | – | – | **256** |
| 5 | 6 | 2 | **32** | – | – | **32** |
| 5 | 7 | 2 | **42** | – | – | **42** |
| 5 | 8 | 2 | **52** | – | – | **52** |
| 5 | 9 | 2 | **54** | – | – | **54** |
| 5 | 10 | 2 | 60 | – | – | **56** |
| 5 | 6 | 3 | **243** | – | – | **243** |
| 6 | 7 | 2 | **64** | – | – | **64** |
| 6 | 8 | 2 | **85** | – | – | **85** |

*1989)*, 1989, pp. 290–296.

[10] 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.

[11] Flener, P., Frisch, A. M., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., and Walsh, T.: Breaking Row and Column Symmetries in Matrix Models, *Proceedings of the 8th International Joint Conference on Principles and Practice of Constraint Programming (CP 2002), LNCS 2470*, 2002, pp. 462–476.

[12] Gelder, A. V.: Another Look at Graph Coloring via Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 156, No. 2(2008), pp. 230–243.

[13] Gent, I. P.: Arc Consistency in SAT, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002)*, 2002, pp. 121–125.

[14] Hartman, A. and Raskin, L.: Problems and Algorithms for Covering Arrays, *Discrete Mathematics*, Vol. 284, No. 1-3(2004), pp. 149–156.

[15] Hnich, B., Prestwich, S. D., and Selensky, E.: Constraint-Based Approaches to the Covering Test Problem, *Proceedings of the International Work-*

*shop on Constraint Solving and Constraint Logic Programming (CSCLP 2004), LNCS 3419*, 2004, pp. 172–186.

[16] Hnich, B., Prestwich, S. D., Selensky, E., and Smith, B. M.: Constraint Models for the Covering Test Problem, *Constraints*, Vol. 11, No. 2-3(2006), pp. 199–219.

[17] 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(2006), pp. 2291–2306.

[18] 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).

[19] Iwama, K. and Miyazaki, S.: SAT-Variable Complexity of Hard Combinatorial Problems, *Proceedings of the IFIP 13th World Computer Congress*, 1994, pp. 253–258.

[20] Kasif, S.: On the Parallel Complexity of Discrete Relaxation in Constraint Satisfaction Networks, *Artificial Intelligence*, Vol. 45, No. 3(1990), pp. 275–286.

[21] 越村三幸, 鍋島英知, 藤田博, 長谷川隆三: SAT 変換による未解決ジョブショップスケジューリング問題への挑戦, スケジューリング・シンポジウム *2009* 講演論文集, 2009, pp. 209–213.

[22] Lei, Y. and Tai, K.-C.: In-Parameter-Order: A Test Generation Strategy for Pairwise Testing, *Proceedings of 3rd IEEE International Symposium on High-Assurance Systems Engineering (HASE 1998)*, 1998, pp. 254–261.

[23] 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.

[24] Nurmela, K. J.: Upper Bounds for Covering Arrays by Tabu Search, *Discrete Applied Mathematics*, Vol. 138, No. 1-2(2004), pp. 143–152.

[25] Régin, J.-C.: Generalized Arc Consistency for Global Cardinality Constraint, *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI 1996)*, Vol. 1, 1996, pp. 209–215.

[26] Selman, B., Levesque, H. J., and Mitchell, D. G.: A New Method for Solving Hard Satisfiability Problems, *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI 1992)*, 1992, pp. 440–446.

[27] Seroussi, G. and Bshouty, N. H.: Vector Sets for Exhaustive Testing of Logic Circuits, *IEEE Transactions on Information Theory*, Vol. 34, No. 3(1988), pp. 513–522.

[28] Shiba, T., Tsuchiya, T., and Kikuno, T.: Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing, *Proceedings of 28th International Computer Software and Applications Conference (COMPSAC 2004)*, 2004, pp. 72–77.

[29] 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*, (2009). to appear.

[30] 田村直之, 多賀明子, 番原睦則, 宋剛秀, 鍋島英知, 井上克巳: ショップ・スケジューリング問題の SAT 変換による解法, スケジューリング・シンポジウム *2007* 講演論文集, 2007, pp. 97–102.

[31] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Proceedings of the 12th International Joint Conference on Principles and Practice of Constraint Programming (CP 2006), LNCS 4204*, 2006, pp. 590–603.

[32] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Constraints*, Vol. 14, No. 2(2009), pp. 254–272.

[33] 土屋達弘, 菊野亨: ペアワイズテスト - ソフトウェアテストの効率化を求めて -, 電子情報通信学会論文誌 *D*, Vol. J90-D, No. 10(2007), pp. 2663–2674.

[34] Walsh, T.: SAT v CSP, *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP 2000)*, 2000, pp. 441–456.

[35] Williams, A. W.: Determination of Test Configurations for Pair-Wise Interaction Coverage, *Proceedings of 13th International Conference on Testing Communicating Systems (TestCom 2000)*, 2000, pp. 59–74.

[36] Zhang, H.: Combinatorial Designs by SAT Solvers, *Handbook of Satisfiability*, IOS Press, 2009, pp. 533–568.