SAT符号化を用いた釣合い型不完備ブロック計画の構成

松中，春樹
丹生，智也
番原，睦則
田村，直之

(Citation)
人工知能学会論文誌,27(2):10-15

(Issue Date)
2012

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90002837

2011 年大会速報論文特集

Short Paper

SAT符号化を用いた釣合い型不完備ブロック計画の構成

Generating Balanced Incomplete Block Designs by SAT Encoding

松中 春樹 神戸大学 大学院システム情報学研究科
Haruki Matsunaka Graduate School of System Informatics, Kobe University
matsunaka@stu.kobe-u.ac.jp

丹生 智也 神戸大学 大学院工学研究科
Tomoya Tanjo Graduate School of Engineering, Kobe University
tanjo@stu.kobe-u.ac.jp

番原 睦則 神戸大学 情報基盤センター
Mutsunori Banbara Information Science and Technology Center, Kobe University
banbara@kobe-u.ac.jp,http://kaminari.istc.kobe-u.ac.jp/banbara-jp.html

田村 直之 （同 上）
Naoyuki Tamura
tamura@kobe-u.ac.jp,http://bach.istc.kobe-u.ac.jp/tamura-jp.html

keywords: propositional satisfiability (SAT), SAT encoding, balanced incomplete block design (BIBD)

Summary

Propositional Satisfiability (SAT) is fundamental in solving many application problems in Artificial Intelligence and Computer Science. Remarkable improvements in the efficiency of SAT solvers have been made over the last decade. Such improvements encourage researchers to solve constraint satisfaction problems by encoding them into SAT (i.e. “SAT encoding”). Balanced Incomplete Block Design (BIBD) is one of the most typical block designs. BIBDs have been applied in several fields such as design experiments, coding theory, and cryptography. In this paper, we consider the problem of generating BIBDs by SAT encoding. We present a new SAT encoding that is an enhancement of order encoding with the idea of binary tree. It is designed to reduce the number of clauses required for cardinality constraints, compared with order encoding. In our experiments, our encoding was able to give a greater number of solutions than order encoding and state-of-the-art constraint solvers Mistral and choco.

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 は人工知能および計算機科学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，スケジューリング，定理証明，システム検証，制約充足問題 (Constraint Satisfaction Problem; CSP) など，様々な分野に応用されている．

近年，大規模な SAT 問題を高速に解くことが可能な SAT ソルバが実現され，これらの分野への実用的応用が急速に拡大している [Biere 09, 井上 10]．特に，CSP を SAT 問題に符号化して，高速な SAT ソルバを用いて求解する **SAT 符号化** [田村 10] 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，支持符号化法，対数符号化法，順序符号化法.

なかでも，順序符号化法 [Tamura 06] は，SAT ソルバの基本動作である単位伝播が，元の CSP における範囲伝播に対応しており，効率の良い求解が可能である．また，その有効性は，順序符号化法を実装した制約ソルバ Sugar[*1] が 2008–2009 年国際制約ソルバ競技会において 2 年連続優勝 (GLOBAL 部門) したことからも伺える．

釣合い型不完備ブロック計画 (Balanced Incomplete Block Design; $BIBD$**)** [Colbourn 07] はラテン方格と並ぶ，組合せデザイン分野の代表的な問題である．$BIBD$ の応用分野としては，実験計画，符号理論，暗号理論などがある．

本論文では，$BIBD$ を構成する問題に対して，新しい SAT 符号化法を提案する．この符号化法は，二分木を用いて順序符号化法を改良したものである．順序符号化法と比較して，基数制約 (cardinality constraint) の符号化に必要な節数が少なくてよい点が特長である．提案する SAT 符号化法の有効性を示すために，小中規模の問題に対する実行実験を行った．その結果，提案手法は従来の順序符号化法および最新の高速な制約ソルバである Mistral, choco と比較して，より多くの問題を解くことができた．

*1 http://bach.istc.kobe-u.ac.jp/sugar/

	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$
1	1	1	1	0
2	1	1	0	1
3	1	0	1	1
4	0	1	1	1

図 1 $BIBD(4,4,3,3,2)$ の結合行列

2. 釣合い型不完備ブロック計画と関連研究

2·1 釣合い型不完備ブロック計画

v，b，r，k，λ を正の整数とし，v 個の元から成る集合 $\mathbb{P}$ と，$\mathbb{P}$ のいくつかの k 点部分集合からなる集合 $\mathbb{B}$ の組 $(\mathbb{P}, \mathbb{B})$ を**結合構造**と呼ぶ．$BIBD(v,b,r,k,\lambda)$ は，以下のように定義される．

【定義 1】 $BIBD(v,b,r,k,\lambda)$ は，以下の条件を満たす結合構造である．

- $\mathbb{P}$ の異なる 2 点を含むブロックの数が λ である．
- $\mathbb{P}$ の任意の点を含むブロックの数が r である．
- ブロックの総数は b である．
- $2 < k < v$ で，$\mathbb{B}$ が $\mathbb{P}$ の k 点部分集合全体の真部分集合となる．

〔例 1〕 $BIBD(4,4,3,3,2)$ の例を示す．

$$\begin{aligned}\mathbb{P} &= \{1,2,3,4\}\\ \mathbb{B} &= \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}\end{aligned}$$

$BIBD(v,b,r,k,\lambda)$ の点とブロックを $\mathbb{P} = \{p_1,\ldots,p_v\}$，$\mathbb{B} = \{B_1,\ldots,B_b\}$ とし，$p_i \in B_j$ のとき，$a_{i,j} = 1$，そうでないとき，$a_{i,j} = 0$ と定義する．この時，$v \times b$ 行列 $A = (a_{i,j})$ を $BIBD(v,b,r,k,\lambda)$ の**結合行列**と呼ぶ．図 1 に例 1 の $BIBD(4,4,3,3,2)$ の結合行列を示す．これは $v = 4$，$b = 4$ の 4×4 行列であり，各行の和が $r = 3$，各列の和が $k = 3$，相異なる 2 つの行の内積が $\lambda = 2$ であることがわかる．

***BIBD* 構成問題**とは，与えられた v，b，r，k，λ に対し，$BIBD(v,b,r,k,\lambda)$ が存在するかどうかを判定し，存在する場合は $BIBD(v,b,r,k,\lambda)$ を構成する問題である．$BIBD$ 構成問題は求解困難な組合せ問題であり，数多くの未解決問題が残されている [Colbourn 07]．最近では $BIBD(22,33,12,8,4)$ が存在しないことが，計算機による探索で示された．

2·2 $BIBD$ 構成問題の CSP 表現

$BIBD$ 構成問題の CSP 表現について述べる．この CSP 表現は $BIBD(v,b,r,k,\lambda)$ の結合行列に基づいており，1 つの行列と 3 種類の制約から構成される．

基本行列 は二値変数を要素とする $v \times b$ 行列である．各要素 $x_{i,j}$ $(1 \le i \le v,\ 1 \le j \le b)$ は，結合行列の各要素を表し，そのドメインは $x_{i,j} \in \{0,1\}$ である．

行制約 は "基本行列の各行の和が r" を表す制約である．

$$\sum_{j=1}^{b} x_{i,j} = r \quad (1 \le i \le v) \tag{1}$$

列制約 は "基本行列の各列の和が k" を表す制約である．

$$\sum_{i=1}^{v} x_{i,j} = k \quad (1 \le j \le b) \tag{2}$$

内積制約 は "基本行列の相異なる 2 つの行の内積が λ" を表す制約である．

$$\sum_{j=1}^{b} x_{i,j} \cdot x_{i',j} = \lambda \quad (1 \le i < i' \le v) \tag{3}$$

内積制約 (3) は乗算 $x_{i,j} \cdot x_{i',j}$ を含むが，新しい二値変数 $y_{i,i',j} \in \{0,1\}$ $(1 \le i < i' \le v,\ 1 \le j \le b)$ を導入することにより，以下の制約に置き換えることができる．

$$y_{i,i',j} = 1 \Leftrightarrow (x_{i,j} = 1 \wedge x_{i',j} = 1) \tag{4}$$

$$\sum_{j=1}^{b} y_{i,i',j} = \lambda \quad (1 \le i < i' \le v) \tag{5}$$

本論文では $BIBD$ 構成問題の CSP 表現として，制約 (1) (2) (4) (5) を用いる．この CSP 表現の主要な制約 (1) (2) (5) は，**基数制約** $\sum_{i=1}^{n} X_i = c$ ($X_i \in \{0,1\}$, c は整数定数) で表される．したがって，SAT 符号化を用いて $BIBD$ 構成問題を効率よく解くためには，基数制約の SAT 符号化が重要な位置を占める．

本節で述べた CSP 表現は制約ソルバの性能を競う国際制約ソルバ競技会のベンチマークにも使用されてる一般的なものであり，特定のソルバに対して有利な表現ではない．他の制約モデルとしては，Meseguer と Torras のモデル [Meseguer 01]，Prestwich のモデル [Prestwich 01] などがある．

3. 順序符号化法

3·1 順序符号化法の概要

順序符号化法 [Tamura 06] は，整数有限領域上の CSP を SAT 問題に符号化する手法の一つである．順序符号化法は，ショップ・スケジューリング問題 [Tamura 06]，組合せテストケース生成問題 [Banbara 10] に対して，未解決問題の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．

順序符号化法では，各整数変数 z について，そのドメインが $\{a_1, a_2, \ldots, a_n\}$ の時 (ただし $a_1 < a_2 < \ldots < a_n$)，$z \le a_i$ を表す $n-1$ 個の命題変数 $p(z \le a_1)$, $p(z \le a_2)$, $\ldots$, $p(z \le a_{n-1})$ を用いる．なお，$z \le a_n$ は常に真であるため，命題変数 $p(z \le a_n)$ 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．

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

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

$$\neg p(z \le 0) \vee p(z \le 1)$$

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 $z = 0, z = 1, z = 2$ に対応する．

$p(z \le 0)$	$p(z \le 1)$	解釈
1	1	$z = 0$
0	1	$z = 1$
0	0	$z = 2$

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 $a_1 < z_1 \le b_1$，…，$a_n < z_n \le b_n$ 中のすべての点 $(z_1, \ldots, z_n)$ が制約に違反する時，以下の節を追加する．

$$p(z_1 \le a_1) \vee \neg p(z_1 \le b_1) \vee \cdots \vee p(z_n \le a_n) \vee \neg p(z_n \le b_n)$$

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．今，a_i を非零の整数定数，c を整数定数，z_i を相異なる整数変数とする．この時，制約 $\sum_{i=1}^{n} a_i z_i \le c$ は以下のように符号化できる．

$$\bigwedge_{b_i} \bigvee_{i} (a_i z_i \le b_i)^{\#}$$

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．

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

ただし，z の取り得る最小値未満の a については $p(z \le a)$ 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 w, z のドメインが $\{0,1,2\}$ の時，制約 $w - z \le -1$ は以下の 3 つの節に符号化される．

$$\neg p(z \le 0) \quad p(w \le 0) \vee \neg p(z \le 1) \quad p(w \le 1)$$

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

3·2 $BIBD$ の順序符号化

$BIBD$ 構成問題の CSP 表現を順序符号化法を用いて SAT 問題に符号化する方法を述べる．二値変数 $x_{i,j}$，$y_{i,i',j}$ に対し，命題変数 $p(x_{i,j} \le 0)$，$p(y_{i,i',j} \le 0)$ を導入する (ただし，$1 \le i < i' \le v, \quad 1 \le j \le b$)．

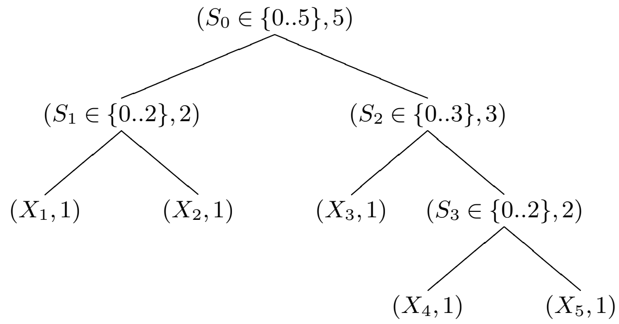
制約 (4) は以下の節に符号化される．

$$p(y_{i,i',j} \le 0) \vee \neg p(x_{i,j} \le 0) \tag{6}$$

$$p(y_{i,i',j} \le 0) \vee \neg p(x_{i',j} \le 0) \tag{7}$$

$$\neg p(y_{i,i',j} \le 0) \vee p(x_{i,j} \le 0) \vee p(x_{i',j} \le 0) \tag{8}$$

節 (6)(7) は $y_{i,i',j} = 1 \Rightarrow (x_{i,j} = 1 \wedge x_{i',j} = 1)$ を，節 (8) は $y_{i,i',j} = 1 \Leftarrow (x_{i,j} = 1 \wedge x_{i',j} = 1)$ を符号化したものである．



$S_0 = 3 \quad S_0 = S_1 + S_2 \quad S_1 = X_1 + X_2$
$S_2 = X_3 + S_3 \quad S_3 = X_4 + X_5$

図 2 $\sum_{i=1}^{5} X_i = 3 \; (X_i \in \{0,1\})$ に対する二分木と制約の例

行制約 (1) は，まず線形比較の連言 $\left(\sum_{j=1}^{b} x_{i,j} \le r\right) \wedge \left(\sum_{j=1}^{b} x_{i,j} \ge r\right)$ に置き換えられる．その後，各線形比較は前節で述べた $\sum_{i=1}^{n} a_i z_i \le c$ と同じ方法で符号化される．列制約 (2)，内積制約 (5) も同様に符号化される．

この符号化の欠点は，基数制約 (1)(2)(5) の符号化に必要な節数が巨大になることである．例えば，行制約 (1) の符号化には $O(v 2^{b-1})$ 個の節が必要となる．この問題を回避するために，基数制約に対する新しい順序符号化法を提案する．

4. 二分木を用いた基数制約の順序符号化

基数制約 $\sum_{i=1}^{n} X_i = c \; (X_i \in \{0,1\}$, c は整数定数) を二分木を用いて分解した後，順序符号化法を用いて符号化する手法を提案する．

基数制約 $\sum_{i=1}^{n} X_i = c$ に対して，以下のように二分木を生成する．ラベル値 n の孤立点から始めて，ラベル値 $m \ge 2$ の各終端点に対して，ラベル値が $\lfloor m/2 \rfloor$ と $m - \lfloor m/2 \rfloor$ の 2 つの子節点を接続する操作を再帰的に繰り返す．この操作により，ラベル値 1 の葉を n 個もつ二分木が生成される．続いて，この二分木の葉に二値変数 $X_i (1 \le i \le n)$ を割り当てる (全単射)．根にはドメイン $\{0..n\}$ の新しい整数変数を割り当てる．ラベル値 m の各内点にドメイン $\{0..m\}$ の新しい整数変数を割り当てる．次に，二分木を基に新しい制約を生成する．根に割り当てられた変数 S_0 に対して，制約 $S_0 = c$ を生成する．親節点の変数 S_p とその 2 つの子節点の変数 S_q，S_r に対して，制約 $S_p = S_q + S_r$ を生成する．最後に，生成された変数および制約を順序符号化法を用いて符号化する．

図 2 に，基数制約 $\sum_{i=1}^{5} X_i = 3 \; (X_i \in \{0,1\})$ に対して生成される二分木と制約の例を示す．各節点の左側が割り当てられた整数変数，右側がラベル値を表している．根に割り当てられた変数 S_0 は基数制約の左辺 $\sum_{i=1}^{5} X_i$ に対応し，各内点に割り当てられた変数 $S_j (1 \le j \le 3)$

表 1 基数制約 $\sum_{i=1}^{n} X_i = c\,(X_i \in \{0,1\}, c$ は整数定数) の符号化に必要な節数と補助変数の数

手法	節数	補助変数の数
順序符号化法	$O(2^{n-1})$	0
提案手法	$O(n^2 + n\log n)$	$O(n\log n)$

はその部分和を表している．

表 1 に基数制約 $\sum_{i=1}^{n} X_i = c$ の SAT 符号化に必要な節数と補助変数の数を示す．順序符号化法は $O(2^{n-1})$ 個の節が必要であるのに対し，提案手法では $O(n^2 + n\log n)$ 個と少ない節数に抑えることができる*2．補助変数とは，提案手法において二分木の内点に割り当てた整数変数を符号化するために必要な命題変数である．順序符号化法には必要ないが，提案手法ではこれらの整数変数の符号化に $O(n\log n)$ 個の補助変数が必要となる．

提案手法の有効性 (特に，求解の効率性と拡張性) について述べる．符号化法の効率性については，解く問題の性質および利用する SAT ソルバの特性 (系統的ソルバか確率的ソルバか) によって異なるため明確な基準はないが，SAT 符号化の研究者の間では以下の二つの基準がよく用いられる．

(1) 符号化後の SAT 問題に対する SAT ソルバの単位伝播と元の CSP に対する制約伝播との関連

(2) 符号化に必要な節数

(1) について，提案手法のベースとなる順序符号化法は，SAT ソルバの単位伝播が元の CSP に対する範囲伝播に対応しており，直接符号化法，支持符号化法，対数符号化法等の他の符号化法と比較して，効率の良い求解が可能である [田村 10]．(2) については，表 1 に示したように，提案手法は従来の順序符号化法と比較して少ない節数に抑えることができる．さらに，提案手法は基数制約だけでなく擬似ブール制約 $\sum_{i=1}^{n} a_i X_i = c$ (a_i は非零の整数定数, $X_i \in \{0,1\}$, c は整数定数)，線形制約 $\sum_{i=1}^{n} a_i z_i = c$ (a_i は非零の整数定数，z_i は相異なる整数変数，c は整数定数) にも適用可能であり拡張性が高い．

基数制約は古くから研究されているが，ここ数年新しい SAT 符号化法がいくつか提案されている．Bailleux らの符号化法 [Bailleux 03] は，単位伝播がアーク整合性維持法に対応し，基数制約 $\sum_{i=1}^{n} X_i \le c$ の符号化に $O(n^2)$ の節数を必要とする．Sinz の符号化法 [Sinz 05] と Codish らの符号化法 [Codish 10] は，必要な節数が各々$O(nc)$, $O(n\log^2 c)$ に抑えられる．Bailleux らの符号化法は，二分木を用いる点で提案手法と類似しているが，順序符号化法とは異なる．また，これらの符号化は基数制約専用であるため，擬似ブール制約および線形制約に直接適用することはできない．Eén らによる擬似ブール制約の符号化法 [Eén 06] は，高速な擬似ブール制約ソルバ MiniSat+ に実装され広く用いられている．提案手法と既存の符号化法との比較実験および考察に関しては今後の課題とする．なお，5 章の $BIBD$ 構成問題 (全 237 問) を用いて，提案手法と Bailleux らの符号化法を比較した結果，解けた問題数は提案手法が 1 問多かった．

表 2 解けた問題数の比較

$v \times b$	問題数	提案手法	順序符号化法	Mistral	choco
49-100	3	**3**	**3**	**3**	**3**
101-200	11	**11**	**11**	**11**	**11**
201-300	12	**12**	**12**	**12**	**12**
301-400	17	**16**	**16**	**16**	15
401-500	14	**13**	12	**13**	7
501-600	19	**17**	**17**	**17**	7
601-700	19	**17**	**17**	16	6
701-800	20	**16**	15	10	4
801-900	19	**16**	15	10	5
901-1000	16	**11**	10	9	4
1001-1100	27	**22**	**22**	6	3
1101-1200	20	**16**	15	4	4
1201-1300	25	**17**	16	4	3
1301-1380	15	**14**	13	2	4
合計	237	**201**	194	133	88

5. 実　行　実　験

提案手法の有効性を評価するため，$BIBD(v,b,r,k,\lambda)$ 構成問題 ($v \times b \le 1400$, $3 \le k \le v/2$, 全 237 問) を用いて比較実験を行った．CSP 表現には 2·2 節で述べた制約 (1) (2) (4) (5) を使用した．比較に用いたシステムは，以下の通りである．

- 提案手法を用いて SAT 符号化し，高速 SAT ソルバ MiniSat 2.2 (core) *3を用いて求解.
- 順序符号化法を用いて SAT 符号化し，MiniSat 2.2 (core) を用いて求解．節数を抑えるため，Sugar のヒューリスティックスを用いて基数制約を分解.
- 高速制約ソルバ Mistral 1.550 *4 を用いて求解.
- 高速制約ソルバ choco (choco2_impwdeg) *5 を用いて求解.

各システムにおいて，SAT ソルバ／制約ソルバの解が充足可能 (`SAT`) の場合は，その解から $BIBD$ を構成することができる．また，解が充足不能 (`UNSAT`) の場合は，$BIBD$ を構成できないことを意味する．Mistral と choco については，制約 (4) を $y_{i,i',j} = x_{i,j} \cdot x_{i',j}$ で置き換え，各基数制約の記述にグローバル制約の一つである weightedSum を用いた．実験環境は Linux マシン (Intel Xeon 3.00GHz，メモリ 8GB) であり，各ソルバのタイムアウトは 1800 秒とした．

まず最初に，表 2 に各システムで解けた問題数を示す．左から順に，$v \times b$ の値の範囲，問題数，解けた問題数を表している．各 $v \times b$ について解けた問題数が最も多いシステムの値をボールド体で示す．提案手法は全 237

*2 順序符号化に必要な節数については，論文 [Tamura 06] および解説論文 [田村 10] を参照いただきたい．

*3 `http://minisat.se/Main.html`

*4 `http://homepages.laas.fr/ehebrard/Software.html`

*5 `http://www.emn.fr/z-info/choco-solver/`

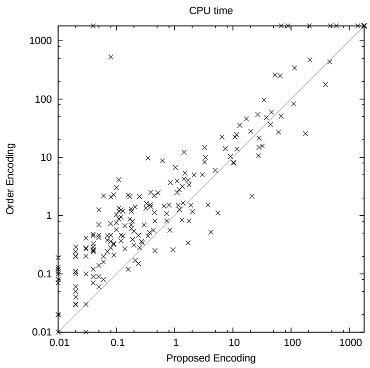


図 3 CPU 時間の比較

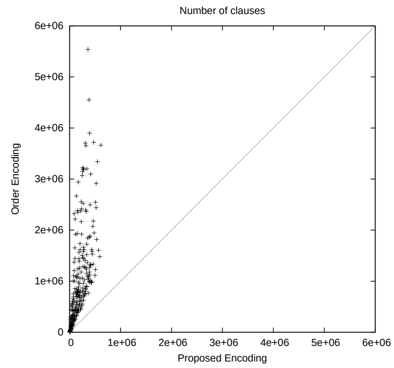


図 4 節数の比較

問中 201 問と最も多くの問題を解いている．提案手法で解けた問題数は順序符号化法より 7 問多く，順序符号化法で解けた問題は全て提案手法でも解けている．また，$v \times b$ の値が大きくなると，Mistral と choco は解ける問題数が大きく減るのに対して，提案手法は安定的に問題を解いている．なお，各システムとも解けた問題はすべて SAT，すなわち，$BIBD$ を構成できる場合であった．

次に，提案手法と順序符号化法について，MiniSat が求解に要した CPU 時間の比較結果を図 3 に示す．横軸が提案手法，縦軸が順序符号化法の CPU 時間を表している (横軸，縦軸ともに対数目盛)．図 3 より，多くの問題に対して提案手法が順序符号化法より高速に求解していることがわかる．より詳細には，提案手法が順序符号化法よりも高速に求解した問題数は 165 問であり，その逆は 30 問であった．また，解けた問題の CPU 時間の幾何平均は，提案手法が 0.28 秒，順序符号化法が 0.76 秒であり，提案手法が約 2.7 倍高速に求解している．

最後に，提案手法と順序符号化法について，生成される SAT 問題の節数の比較結果を図 4 に示す．横軸が提案手法，縦軸が順序符号化法によって生成された SAT 問題の節数を表している．図 4 から，提案手法の節数は，順序符号化法と比較して，圧倒的に少ないことがわかる．例えば，比較的規模の大きな $BIBD(27, 27, 13, 13, 6)$ の符号化後の節数は，順序符号化法が $2,418,228$ に対して，提案手法は $232,443$ と約 10 分の 1 に抑えられる．

なお，他の制約モデル (および SAT 符号化) として，Prestwich のモデル [Prestwich 01] の性能も測定したが，解ける問題数が非常に少ないため，実験結果は省略した．

6. 対称解の除去

$BIBD$ 構成問題は対称性が高く，対称解が多く存在する．例えば，$BIBD(4, 4, 3, 3, 2)$ の結合行列 (図 1) を見ると，任意の相異なる 2 つの行 (あるいは列) を入れ替えたものも解となることがわかる．制約プログラミングの分野において，対称解を除去することは**対称性除去 (symmetry breaking)** と呼ばれ，重要な研究課題となっている．一般に対称性除去は解探索のコストを減らす効果があるが，SAT 符号化に対する有効性については十分に研究されているとはいえない．

本章では，提案手法に対する対称性除去の有効性を評価するために，以下の手法を用いて実験を行った．これらの手法は結合行列の行と列に関する対称解を除去するものであり，存在すべき解を失うものではない．実験環境，ベンチマーク，タイムアウトは，5 章と同じである．

- Double Lex [Flener 02]：基本行列 (2·2 節) の連続する相異なる 2 つの行，および連続する相異なる 2 つの列に対して辞書式順序制約を適用する．
- Frisch の手法 [Frisch 04]：Double Lex に加え，基本行列の 1,2 行目を固定する．3 行目以降の各行に対して，固定した 1,2 行目との内積制約から導かれる制約を追加する．
- Snake Lex [Grayland 09]：Double Lex の改良として提案されたスネーク順序に基づく手法である．

表 3 提案手法に対称性除去を適用した結果：解けた問題数

問題数	なし	Double	Frisch	Snake
237	201	202	**203**	200

表 4 提案手法に対称性除去を適用し UNSAT の問題を解いた結果：CPU 時間 (単位：秒)

v	b	r	k	λ	なし	Double	Frisch	Snake
15	21	7	5	2	$> 24h$	8,314	6,042	**2,541**
21	28	8	6	2	$> 24h$	$> 24h$	$> 24h$	$> 24h$
22	22	7	7	2	$> 24h$	$> 24h$	62,104	**4,917**

表 3 に提案手法に対称性除去手法を適用して解けた問題数を示す．問題数に続いて，対称性除去なし，Double Lex，Frisch の手法，Snake Lex を適用して解けた問題数を表している．Double Lex の拡張である Frisch の手法が 203 問と最も多くの問題を解いた (対称性除去なしと比べて 3 問増・1 問減)．新たに解けた 3 問は，$v \times b$ の値が各々578, 1083, 1224 と比較的大きく，MiniSat が要した CPU 時間は 1208.67 秒，1002.29 秒，512.93 秒であった．これは，追加制約が解の探索空間を削減し，解の発見に有効に作用したと考えられる．(どの対称性除去手法を用いても) 解けなくなった 1 問は，MiniSat がバックトラックなしで 0.04 秒と短時間で解いていることから，非

常に特殊なケースといえる．追加制約により MiniSat の変数選択順序および学習節に変化が生じ，解けなくなったものと考えられる．

5 章の実験結果と同様，解けた問題はすべて SAT であり，UNSAT の問題は 1 問も解くことができなかった．そこでベンチマーク中から $BIBD$ が構成できない (UNSAT となる) ことが理論的に証明されている問題を 3 問選び，タイムアウトを 1800 秒から 24 時間に延ばして同様の実験を行った．表 4 に CPU 時間を示す．“$> 24h$” は 24 時間以内に求解できなかったことを表す．Frisch の手法と Snake Lex が 3 問中 2 問の UNSAT を示すことができた．

7. ま と め

本論文では，$BIBD$ 構成問題に対する新しい SAT 符号化法を提案した．提案手法は二分木を用いて従来の順序符号化法を改良したものであり，基数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．$BIBD(v, b, r, k, \lambda)$ 構成問題 ($v \times b \leq 1400$, $3 \leq k \leq v/2$, 全 237 問) を用いて比較実験を行った結果，提案手法は順序符号化法，最新の高速制約ソルバ Mistral，choco と比較して，より多くの問題を解くことができた．また，提案手法に対する対称性除去の有効性についても評価実験を行い，SAT，UNSAT の両方の問題に対して有効であることを確認した．基数制約は，$BIBD$ の他にも時間割問題，SAT の拡張である MaxSAT および擬似ブール制約，CSP の拡張である MaxCSP などの重要な応用がある．提案手法を制約ソルバ Sugar に実装し，これらの応用に対する有効性を評価することが今後の課題である．

◇ 参 考 文 献 ◇

[Bailleux 03] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 9th International Joint Conference on Principles and Practice of Constraint Programming (CP 2003), LNCS 2833*, pp. 108–122 (2003)

[Banbara 10] Banbara, M., Matsunaka, H., Tamura, N., and Inoue, K.: Generating Combinatorial Test Cases by Efficient SAT Encodings Suitable for CDCL SAT Solvers, in *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17), LNCS 6397*, pp. 112–126 (2010)

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

[Codish 10] Codish, M. and Zazon-Ivry, M.: Pairwise Cardinality Networks, in *Proceedings of the 16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, Vol. 6355 of *Lecture Notes in Computer Science*, pp. 154–172, Springer (2010)

[Colbourn 07] Colbourn, C. J. and Dinitz, J. H.: *Handbook of Combinatorial Designs*, Chapman & Hall/CRC (2007)

[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)

[Flener 02] Flener, P., Frisch, A. M., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., and Walsh, T.: Breaking Row and Column Symmetries in Matrix Models, in *Proceedings of the 8th International Joint Conference on Principles and Practice of Constraint Programming (CP 2002), LNCS 2470*, pp. 462–476 (2002)

[Frisch 04] Frisch, A. M., Jefferson, C., and Miguel, I.: Symmetry Breaking as a Prelude to Implied Constraints: A Constraint Modelling Pattern, in *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI'2004)*, pp. 171–175 (2004)

[Grayland 09] Grayland, A., Miguel, I., and Roney-Dougal, C. M.: Snake Lex: An Alternative to Double Lex, in *Proceedings of the 15th International Joint Conference on Principles and Practice of Constraint Programming (CP 2009), LNCS 5732*, pp. 391–399 (2009)

[井上 10] 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, pp. 57–67 (2010)

[Meseguer 01] Meseguer, P. and Torras, C.: Exploiting symmetries within constraint satisfaction search, *Artificial Intelligence*, Vol. 129, No. 1-2, pp. 133–163 (2001)

[Prestwich 01] Prestwich, S.: Balanced Incomplete Block Design as Satisfiability, in *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'01)*, pp. 189–198 (2001)

[Sinz 05] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 11th International Joint Conference on Principles and Practice of Constraint Programming (CP 2005), LNCS 3709*, pp. 827–831 (2005)

[Tamura 06] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, in *Proceedings of the 12th International Joint Conference on Principles and Practice of Constraint Programming (CP 2006), LNCS 4204*, pp. 590–603 (2006)

[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, pp. 77–85 (2010)

〔担当委員：阿部 明典〕

2011 年 8 月 20 日　受理

著 者 紹 介

松中　春樹

2010 年神戸大学工学部情報知能工学科卒業．2010 年同大学大学院システム情報学研究科博士課程前期課程情報科学専攻進学．制約プログラミング，組合せテスト，SAT 技術等に興味をもつ．

丹生　智也（学生会員）

2007 年神戸大学工学部情報知能工学科卒業．2009 年同大学大学院工学研究科博士課程前期課程情報知能学専攻修了．2009 年同研究科博士課程後期課程情報知能学専攻進学．制約プログラミング，論理プログラミング，SAT 技術，擬似ブール制約等に興味をもつ．日本ソフトウェア科学会会員．

番原　睦則（正会員）

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 年同校准教授．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，情報処理学会会員．

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，情報処理学会会員．