3J1-OS7-2

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 Encodings

松中 春樹[*1] 丹生 智也[*2] 番原 睦則[*3] 田村 直之[*3]
Haruki Matsunaka Tomoya Tanjo Mutsunori Banbara Naoyuki Tamura

[*1] 神戸大学 大学院システム情報学研究科
Graduate School of System Informatics, Kobe University
[*2] 神戸大学 大学院工学研究科
Graduate School of Engineering, Kobe University
[*3] 神戸大学 情報基盤センター
Information Science and Technology Center, Kobe University

In this paper, we consider the problem of generating Balanced Incomplete Block Designs (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 boolean 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，プランニング問題，スケジューリング問題，制約充足問題 (Constraint Satisfaction Problems, CSP)，定理証明など，さまざまな分野に応用されている．

近年，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 [井上 10]．特に，CSP を SAT 問題に変換して，高速な SAT ソルバを用いて求解する **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，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*) [Mathon 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性能評価を行った．その結果，提案手法は従来の順序符号化法と比較して，より多くの問題を解くことができた．

連絡先: 松中春樹，神戸大学 大学院システム情報学研究科，兵庫県神戸市灘区六甲台町 1-1 神戸大学情報基盤センター田村研究室，078-803-5364，matsunaka@stu.kobe-u.ac.jp

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

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っている．

例 1 $BIBD(4,4,3,3,2)$ の例を示す．

$$\mathbb{P} = \{1,2,3,4\}$$

$$\mathbb{B} = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}$$

$BIBD(v,b,r,k,\lambda)$ の点とブロックを $\mathbb{P} = \{p_1,\dots,p_v\}$，$\mathbb{B} = \{B_1,\dots,B_b\}$ 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．例 1 の $BIBD(4,4,3,3,2)$ の結合行列を図 1 に示す．これは $v = 4$，$b = 4$ の 4×4 の行列であり，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．

	{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

$BIBD$ 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 [Mathon 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 [Bilous 03]．

2.2 $BIBD$ 構成問題の CSP 表現

$BIBD(v,b,r,k,\lambda)$ 構成問題の古典的な CSP 表現について述べる．この CSP 表現は，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 になる制約である．

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

列制約 は基本行列の各列の和が k になる制約である．

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

内積制約 は基本行列の相異なる行の内積が λ になる制約である．

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

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 $BIBD$ 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．従って，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．

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 問題に変換する方法の一つである．順序符号化法は，ショップ・スケジューリング問題，パッキング問題，組合せテストケース生成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 [Tamura 06, Koshimura 10, Soh 10, Banbara 10]．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 $p(z\le a_1)$, $p(z\le a_2)$, …, $p(z\le a_{n-1})$ 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，二つの命題変数 $p(z\le 0)$, $p(z\le 1)$ 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 を互いに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．

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

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 $w-z\le -1$ は以下の三つの節に符号化される．

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

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

$BIBD(v, b, r, k, \lambda)$ の CPS 表現を順序符号化法を用いて SAT 問題に変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する．

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

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

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

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

制約 (1) は，まず以下の線形比較の連言に置換えられる．

$$\left(\sum_{j=1}^{b} x_{i,j} \le r\right) \land \left(\sum_{j=1}^{b} x_{i,j} \ge r\right)$$

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．また，制約 (2)(5) も同様に符号化する．

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，制約 (1) の符号化には $O(b2^{v-1})$ 個の節が必要となる．この問題を回避するために，新しい基数制約の順序符号化法を提案する．

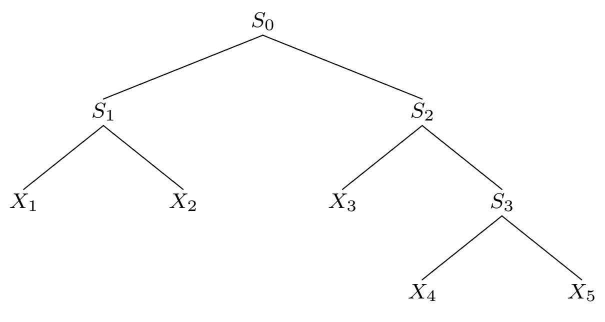
4. 二分木を用いた基数制約の順序符号化

本節では，基数制約の $\sum_{i=1}^{n} X_i$ を二分木を用いて分解し，部分和を順序符号化法を用いて符号化する手法を提案する．従来の順序符号化法では，基数制約の符号化に $O(2^{n-1})$ 個の節が必要である．一方で，提案手法は節数を $O(n^2 + n \log n)$ 個に抑えることができる．

基数制約 $\sum_{i=1}^{n} X_i = c$ に対して，以下の手法で二分木を生成する．二分木の根に整数変数 $S_0 \in \{0, 1, \ldots, n\}$ を割り当てる．$S_p \in \{0, 1, \ldots, m\}$ $(2 \le m)$ が割り当てられた節点に対し，2 つの子節点を導入し，それぞれに，整数変数 $S_{c1} \in \{0, 1, \ldots, \lfloor m/2 \rfloor\}$，$S_{c2} \in \{0, 1, \ldots, m - \lfloor m/2 \rfloor\}$ を割り当てる．ただし，子節点の変数のドメインが $\{0, 1\}$ になる場合は，X_i を割り当てる．

親節点 S_p と 2 つの子節点 S_{c1}，S_{c2} は $S_p = S_{c1} + S_{c2}$ の関係を満たす．そして，根 S_0 は $S_0 = c$ となる．

例 2 基数制約 $\sum_{i=1}^{5} X_i = 3$ の提案手法で生成される二分木と変数，制約を示す．



$X_1, X_2, X_3, X_4, X_5 \in \{0, 1\}$

$S_0 \in \{0, 1, 2, 3, 4, 5\}$ $\quad S_1, S_3 \in \{0, 1, 2\}$ $\quad S_2 \in \{0, 1, 2, 3\}$

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

$S_3 = X_4 + X_5$

手法	新しい命題変数の数	節数
従来の順序符号化法	0	$O(2^{n-1})$
提案手法	$O(n \log n)$	$O(n^2 + n \log n)$

表 1: 基数制約に対する命題変数・節数の比較

表 1 に基数制約 $\sum_{i=1}^{n} X_i = c$ の符号化に必要な節数を示す．従来の順序符号科法は $O(2^{n-1})$ の節数が必要であるのに対し，提案手法は $O(n^2 + n \log n)$ と大幅に少なくなっている．提案手法では，節点に整数変数を割り当てるため $O(n \log n)$ 個の新しい命題変数が必要である．

提案手法は二分木を用いる点で，Bailleux らの基数制約の符号化 [Bailleux 03] と類似している．しかし，提案手法は線形和制約 $\sum_{i=1}^{n} a_i z_i = c$ の符号化にも適用可能な点が大きく異なる．

5. 性能評価

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 ($v \times b \le 1400$, 全 237 問) を用いて比較実験を行った．比較に用いたシステムは，以下の通りである．

- 提案手法
  $BIBD$ 構成問題の CSP 表現 (1, 2, 4, 5) を提案手法 (4. 節) を用いて分解した後，順序符号化法を用いて SAT 問題に変換し，高速 SAT ソルバ MiniSat 2.2 (core) [Eén 03] で求解．

- 順序符号化法
  $BIBD$ 構成問題の CSP 表現 (1, 2, 4, 5) を順序符号化法を用いて SAT 問題に変換し，高速 SAT ソルバ MiniSat 2.2 (core) で求解．節数を抑えるため，基数制約を Sugar のヒューリスティックスを用いて分解．

- Mistral 1.550 *2
  $BIBD$ 構成問題の CSP 表現 (1–3) を高速制約ソルバ Mistral を用いて求解．

- choco (choco2_impwdeg) *3
  $BIBD$ 構成問題の CSP 表現 (1–3) を高速制約ソルバ choco を用いて求解．

提案手法と順序符号化法については，MiniSat のタイムアウトは 1800 秒とした．Mistral と choco については，乗算を二値変数に置換え，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，タイムアウトは 1800 秒とした．実験環境は，Linux マシン (Intel Xeon 3.00GHz，メモリ 8GB) である．

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．また，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 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．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

*2 `http://4c.ucc.ie/~ehebrard/mistral/doxygen/html/main.html`
*3 `http://www.emn.fr/z-info/choco-solver/`

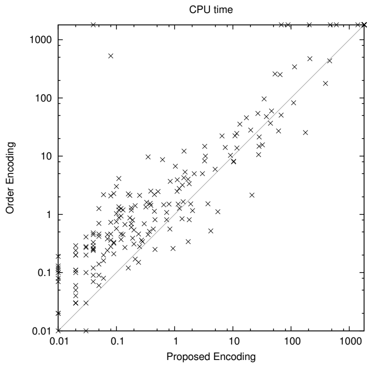


図 2: CPU 時間の比較

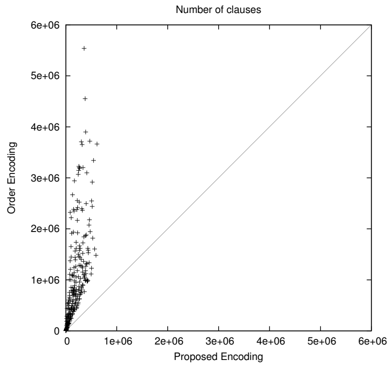


図 3: 節数の比較

$v \times b$	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

表 2: 解けた問題数の比較

結果を図 2 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り)．図 2 から，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．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 3 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 3 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．

6. まとめ

本論文では，$BIBD$ 構成問題に対して，新しい SAT 符号化方法を提案した．$BIBD(v, b, r, k, \lambda)$ 構成問題 ($v \times b \leq 1400$, 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，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．

参考文献

[Bailleux 03] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, *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)*, pp. 112–126 (2010)

[Bilous 03] Bilous, R. T., Running, S., Codes, H., Rees, G. H. J. V., and Rees, G. H. J. V.: Self-Dual Codes and the (22,8,4) Balanced Incomplete Block Design, *J. of Combin. Designs*, Vol. 13, pp. 363–376 (2003)

[Eén 03] Eén, N. and Sörensson, N.: An Extensible SAT-solver, in *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003), LNCS 2919*, pp. 502–518 (2003)

[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 (2010)

[Koshimura 10] Koshimura, M., Nabeshima, H., Fujita, H., and Hasegawa, R.: Solving Open Job-Shop Scheduling Problems by SAT Encoding, *IEICE TRANSACTIONS on Information and Systems*, Vol. E93-D, No. 8, pp. 2316–2318 (2010)

[Mathon 07] Mathon, R. and Rosa, A.: $2-(v, k, \lambda)$ designs of small order, in Colbourn, C. J. and H.Dinitz, J. eds., *The CRC Handbook of Combinatorial Designs*, pp. 25–58, CRC Press (2007)

[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, *Fundamenta Informaticae*, Vol. 102, No. 3–4, pp. 467–487 (2010)

[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)