Kobe University Repository : Kernel

PDF issue: 2025-06-17

パッキング配列問題の制約モデリングとSAT符号化

則武，治樹
番原，睦則
宋，剛秀
田村，直之
井上，克巳

(Citation)
コンピュータソフトウェア,31(1):116-130

(Issue Date)
2014

(Resource Type)
journal article

(Version)
Version of Record

(Rights)
ここに掲載した著作物の利用に関する注意 本著作物の著作権は日本ソフトウェア科学会に帰属します．本著作物は著作権者である日本ソフトウェア科学会の許可のもとに掲載するものです．ご利用に当たっては「著作権法」に従うことをお願いいたします．
Notice for the use of this material: The copyright of this material is retained by t…

(URL)
https://hdl.handle.net/20.500.14094/90002836

KOBE UNIVERSITY

パッキング配列問題の制約モデリングとSAT符号化

則武 治樹　番原 睦則　宋 剛秀　田村 直之　井上 克巳

制約モデリングは与えられた問題を効率よく解く上で重要な役割を果たすことが知られている．近年，大規模な命題論理の充足可能性判定 (SAT) 問題を高速に解くことが可能な SAT ソルバーが実現され，制約充足問題 (CSP; Constraint Satisfaction Problem) を SAT 問題に符号化して，高速な SAT ソルバーを用いて求解するアプローチが成功を収めている．本論文では，組合せデザイン分野の一問題であるパッキング配列 (PA) 問題を例にとり，SAT に基づく制約モデリングについて考察する．PA は別名で相互直交的な部分ラテン方陣系とも呼ばれる困難な組合せ問題であり，データベースのディスク最適配置などへの応用が知られている．まず最初に，PA 問題を異なる観点から捉えた 4 つの制約モデル (CSP 表現) を提案する．次に，これらの制約モデルを順序符号化法を用いて SAT 問題に符号化する方法を示す．なかでも，基本・相違モデルは，与えられた PA のパッキング制約について，その符号化後の節数を小さく抑えられる点が特長である．最後に，組合せデザイン・ハンドブックにある PA 問題を用いて，提案する制約モデルの比較・評価を行った．実験の結果，最適値が未知であった 2 問について既知の上限が最適値であることを示し，5 問について新しい下限を得ることに成功した．

Constraint modeling is known to play an important role in solving problems efficiently. Remarkable improvements of SAT solvers have been made over the last decade. Such improvements encourage researchers to solve Constraint Satisfaction Problems (CSPs) by encoding them into SAT. In this paper, we study SAT-based constraint modeling of the Packing Array (PA) problem in Combinatorial Designs. PA is also called mutually orthogonal partial latin squares and has been applied to optimal disk allocation in Databases. We first present four constraint models from different viewpoints of the PA problem, and then present their SAT encodings based on the order encoding. Particularly, *basic alldiff model* is designed to reduce the number of clauses for the packing constraints of a given PA. To evaluate the efficiency of our proposed models, we carried out experiments on the PA problems in Handbook of Combinatorial Designs. We succeeded in proving the optimality of previously known upper bounds for two arrays and in obtaining improved lower bounds for five arrays.

1　はじめに

パッキング配列 (*PA*; Packing Arrays) は，組合せデザイン分野の代表的な問題である直交表 (Orthogonal Arrays) の拡張である [13]．PA は別名で相互直交的な部分ラテン方陣系 (Mutually Orthogonal Partial Latin Squares) †1とも呼ばれる求解困難な組合せ問題であり，データベースのディスク最適配置などへの応用が知られている [2]．直交表のもう 1 つの拡張として，カバリング配列 (CA; Covering Arrays)

Constraint Modeling and SAT Encoding of the Packing Array Problem.

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

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

Katsumi Inoue, 国立情報学研究所情報学プリンシプル研究系, Principles of Informatics Research Division, National Institute of Informatics.

コンピュータソフトウェア, Vol.31, No.1 (2014), pp.116–130.
[研究論文] 2013 年 7 月 18 日受付.
大会同時投稿論文

†1 本論文では Mutually Orthogonal Partial Latin Squares の和訳として “相互直交的な部分ラテン方陣系” を用いる．

がある．CA はソフトウェアテストの分野に広く応用され，群論等の数学的手法，メタヒューリスティクス，線形計画法，制約充足問題 (CSP; Constraint Satisfaction Problem)，命題論理の充足可能性判定 (SAT) など，これまで数多くの CA 構成手法が提案されている [4] [8] [9] [10] [11] [12] [21] [22] [23] [29] [30] [33] [35] [45] [49]．これらの研究の解説論文としては [44] がある．しかしながら，PA 構成手法に対する既存研究は，グラフ理論等を用いた数学的手法 [1] [39]，メタヒューリスティクス [38]，線形計画法 [8] のみであり，CSP および SAT を用いた手法は筆者らの知る限り存在しない．

制約モデリング [36] は，与えられた問題を効率よく解く上で重要な役割を果たすことが知られており，主として CSP の分野で研究が進められてきた．しかし，近年 SAT 技術 (特に，SAT 符号化と SAT ソルバー) が求解困難な問題を解くための基盤技術として整備され，SAT に基づく制約モデリングの重要性が高まっている．なぜならば，与えられた問題の求解効率は使用する符号化法とソルバーだけでなく，制約モデリングにも大きく依存するからである．さらに，どのような制約モデルが適しているかは自明ではない．

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り，人工知能および計算機工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，ハードウェア検証，ソフトウェア検証，定理証明，CSP など，様々な分野に応用されている [7] [24]．

近年，大規模な SAT 問題を高速に解くことが可能な SAT ソルバー [31] †2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．特に，CSP を SAT 符号化 [42] した後，高速な SAT ソルバーを用いて (元の CSP の) 解を求めるアプローチが成功を収め，これまで数多くの SAT 符号化法が提案されている [6] [14] [18] [25] [26] [34] [40] [41] [43] [46] [48]．

順序符号化法 [40] [41] は，整数有限領域上の CSP

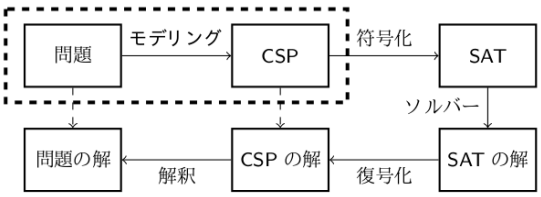


図 1 SAT を用いた問題解法

を SAT 問題に符号化する比較的新しい SAT 符号化法である (4 章参照)．順序符号化法は，求解困難な問題として知られるショップ・スケジューリング問題，二次元矩形パッキング問題，組合せテストのテストケース生成問題に対して，未知の最適値を決定するなど優れた性能を示している [4] [5] [27] [28] [37] [40]．また，順序符号化法の有効性は，SAT 型 CSP ソルバー Sugar †3 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) したことからも窺える．

図 1 にモデリング，符号化，ソルバーの関係を示す．本論文では，パッキング配列を例にとり，特に順序符号化法と CDCL ソルバーに適した制約モデリングについて考察する (図 1 の破線部分)．

本論文ではまず最初に，PA を構成する問題に対して，**基本行列モデル**，**拡張行列モデル**，**拡張・相異モデル**，**基本・相異モデル**と呼ぶ 4 つの制約モデル (CSP 表現) を提案する．特に拡張・相異モデルは，与えられた PA のパッキング制約を CSP における代表的なグローバル制約 [47] である alldifferent 制約を用いて簡潔に表現できる．

次に，これらの制約モデルを順序符号化法を用いて SAT 問題に符号化する方法を示す．しかし，単純に順序符号化法を適用した場合，パッキング制約に対する符号化後の節数が大きくなる問題が生じる．実際に，与えられた $PA(b; k, g)$ に対して，基本・相異モデルを除く 3 つの制約モデルでは，パッキング制約の SAT 符号化に $O(g^2)$ の節数が必要になる．これに対して，基本・相異モデルは拡張・相異モデルの alldifferent 制約を分解することにより，符号化後の節

†2 本論文では特に断らない限り，現在高速 SAT ソルバーの主流となっている CDCL (Conflict-Driven Clause Learning) ソルバー [31] を意味する．

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

表 1 既知の $PAN(k,g)$ [13]

$g\backslash k$	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3	+	9	6	4	-	-	-	-	-	-	-	-	-	-	-	-
4	+	+	16	9	8	5	-	-	-	-	-	-	-	-	-	-
5	+	+	+	25	15	10	10	7	-	-	-	-	-	-	-	-
6	36	34	30–34	30–34	16–34	12–19	12–14	12	12	9	-	-	-	-	-	-
7	+	+	+	+	+	49	28	21	14–15	14	14	10				
8	+	+	+	+	+	+	64	22–34	16–25	16–19	16–17	16	16	12	-	-
9	+	+	+	+	+	+	+	81	27–45	27–30	27	18–21	18–19	18	18	14

数を $O(g)$ に抑えることができる点が特長である．

最後に，提案モデルの有効性を比較・評価するために，2007 年に出版された組合せデザイン・ハンドブック [13] にある最大の PA を構成する問題 (全 46 問) を用いて実行実験を行った．その結果，提案モデルは最適値が未知であった 2 問について既知の上限が最適値であることを示し，5 問について新しい下限を得ることに成功した．さらに，与えられた $PA(b;k,g)$ の $\frac{k}{b}$ の値に着目し，提案モデルの比較考察，対称性除去の効果について述べた．

以降，次の節 2 では，パッキング配列の定義と関連研究について述べる．提案する制約モデルを節 3 で説明した後，それらの SAT 符号化を節 4 で示す．節 5 ではパッキング配列の対称性の除去について簡単に触れる．節 6 で提案モデルの比較・評価を行い，最後に節 7 で本論文をまとめる．

2 パッキング配列と関連研究

以下の定義は，論文 [39] に基づいている．

定義 1 パッキング配列 $PA(b;k,g)$ とは以下の条件を満たす $b \times k$ 配列 $A = (a_{ij})$ である．

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

(2) 任意の異なる 2 つの列 $1 \le c_1 < c_2 \le k$，および任意の値の組 $(x_1,x_2) \in {\mathbb{Z}_g}^2$ に対して，$x_1 = a_{rc_1}$ かつ $x_2 = a_{rc_2}$ を満たす行 r が高々1 つ存在する (**パッキング制約**)．

定義 2 パッキング配列数 $PAN(k,g)$ とは，$PA(b;k,g)$ が存在する最大の b である．

定義 3 $PAN(k,g) = b$ のとき，パッキング配列 $PA(b;k,g)$ は最適であるという．

1	2	3	4
0	**0**	0	0
0	**1**	2	2
0	**2**	1	1
1	**0**	1	2
1	**1**	0	1
1	**2**	2	0
2	**0**	2	1
2	**1**	1	0
2	**2**	0	2

図 2 最適な $PA(9;4,3)$ の例

図 2 に，最適な $PA(9;4,3)$ の例を示す．最初の 2 列について，$(0,0)$ から $(2,2)$ までの全部で $3^2 = 9$ 通りあるそれらの値の組合せすべてが，高々1 つだけ存在することがわかる (ボールド体で表示)．他のどの 2 列の組合せについても同様の性質を満たす．

表 1 に組合せデザイン・ハンドブック [13] に記載されている，$PAN(k,g)$ に対する既知の上下限を示す．表の (k,g) 成分が整数 m の場合は，$PAN(k,g) = m$ を表し，ℓ–u の場合は，$\ell \le PAN(k,g) \le u$ を表している．表 1 中の “+” の欄は，右の欄と同一の値であることを表し，“–” の欄は，Abdel-Ghaffar の論文 [1] で決定された値を表す．$k \le 2g$ の場合しか記載されていないのは，$k > 2g$ については，Abdel-Ghaffar によって既に決定されているためである [1]．なお，次の 6 つについては，ハンドブック出版以前に，Stardom [38] によって表 1 より良い結果が得られている: $31 \le PAN(5,6)$, $31 \le PAN(6,6)$, $20 \le PAN(7,6)$, $17 \le PAN(8,6)$, $PAN(9,6) = 14$, $PAN(11,7) = 15$.

以下，SAT を用いたアプローチをよりわかりやすくするために，2 つの問題を定義する．***PA*問題**とは，

与えられた $\langle k, g, b\rangle$ に対して，$PA(b; k, g)$ が存在するかどうかを判定し，存在する場合，$PA(b; k, g)$ を構成する問題である．**PA最適化問題**とは，与えられた $\langle k, g\rangle$ に対して，$PAN(k, g) = b$ となる最大の $PA(b; k, g)$ を構成する問題である．

本論文は，PA 問題に対する制約モデルおよびその SAT 符号化について述べる．PA 最適化問題の解は，b を変化させた複数の PA 問題を解くことによって求めることができる．すなわち，SAT 符号化された PA 問題群は，充足可能 (SAT) と充足不能 (UNSAT) の両方の問題を含む可能性がある．最適値 (最大の b) は，SAT と UNSAT の境界に位置する．なお，既知の上限に対する PA 問題が SAT の場合は，その上限が最適値となる．

本研究と関連が深い研究として，Hnich らの CA 問題に対する制約モデルおよびその SAT 符号化がある [22] [23]．CA は，“高々1 つ”という PA のパッキング制約 (定義 1 の 2) を，“少なくとも 1 つ”というカバレッジ制約に置き換えたものである．Hnich らは，CA 問題に対して，従来の非 SAT 型 CSP ソルバーの利用を前提とした制約モデル，確率的 SAT ソルバーの利用を前提とした支持符号化法 [18] [26] ベースの CA の符号化を提案している．これに対して，本研究は PA 問題に対して，SAT 型 CSP ソルバー (特に，順序符号化法と CDCL ソルバーの利用) に適した制約モデルを提案する点が大きく異なる．Hnich らの制約モデルとの具体的な違いは，提案するモデルが alldifferent 制約を使っているのに対して，Hnich らは global_cardinality 制約を使っている点である．global_cardinality 制約は，変数に割り当てられる値の出現回数をカウントする必要があり，一般に alldifferent 制約より計算コストが大きい．また，CA 問題に対して，CDCL ソルバーと順序符号化法の組合せが，Hnich らの確率的 SAT ソルバーに適した SAT 符号化よりも優れた性能を示すことは，先行研究 [4] で確認されている．

3 制約モデル

本節では，PA 問題に対して，基本行列モデル，拡張行列モデル，拡張・相異モデル，基本・相異モデルと呼ぶ，4 つの新しい制約モデルを提案する．これらの制約モデルは，CSP 分野で広く用いられる行列モデル [16] に基づいている．行列モデルは，変数を要素にもつ 1 つ以上の行列 (変数の集合) と，その行列上および行列間の制約式から構成される．

3.1 基本行列モデル

基本行列モデルは，与えられた $PA(b; k, g)$ に対して，1 つの行列と 1 種類の制約から構成される．

- **基本行列**とは，整数変数を要素にもつ $b \times k$ 行列である．各要素 $x_{r,i}$ $(1 \le r \le b,\ 1 \le i \le k)$ は PA の各要素を表し，そのドメインは $x_{r,i} \in \{0, 1, 2, \ldots, g-1\}$ である．基本行列は PA そのものを表す行列である．
- パッキング制約は以下のように表される．

$$\neg(x_{r,i} = m) \quad \vee \quad \neg(x_{r,j} = n) \quad \vee \quad \neg(x_{r',i} = m) \quad \vee \quad \neg(x_{r',j} = n) \qquad (1)$$

ただし，$1 \le r < r' \le b,\ 1 \le i < j \le k,\ 0 \le m, n \le g-1$.

制約 (1) は，“基本行列のどの 2 つの列に対しても，g^2 通りある値の組合せすべてが高々1 つ存在する”というパッキング制約を，基本行列上の at_most_one 制約を使って定義通りに表している．この制約モデルの欠点は，パッキング制約に対して生成される at_most_one 制約の個数が $\binom{b}{2}\binom{k}{2}g^2$ と，非常に多くなる点である．

3.2 拡張行列モデル

拡張行列モデルでは，CSP 分野で用いられる拡張行列 [23] の概念を導入する．拡張行列モデルは，与えられた $PA(b; k, g)$ に対して，2 つの行列と 2 種類の制約から構成される．

- 基本行列 (基本行列モデルと同じ)
- **拡張行列**とは，整数変数を要素にもつ $b \times \binom{k}{2}$ 行列である．各要素 $y_{r,(i,j)}$ $(1 \le r \le b,\ 1 \le i < j \le k)$ は基本行列の変数の組 $(x_{r,i}, x_{r,j})$ に対応し，そのドメインは $y_{r,(i,j)} \in \{0, 1, 2, \ldots, g^2-1\}$ である．
- パッキング制約は以下のように表される．

$$\neg\left(y_{r,(i,j)} = \ell\right) \quad \vee \quad \neg\left(y_{r',(i,j)} = \ell\right) \qquad (2)$$

	$(1,2)$	$(1,3)$	$(1,4)$	$(2,3)$	$(2,4)$	$(3,4)$
0	0	0	0	0	0	0
1	2	2	5	5	8	
2	1	1	7	7	4	
3	4	5	1	2	5	
4	3	4	3	4	1	
5	5	3	8	6	6	
6	8	7	2	1	7	
7	7	6	4	3	3	
8	6	8	6	8	2	

図 3 図 2 の $PA(9;4,3)$ に対する拡張行列

ただし，$1 \le r < r' \le b$, $1 \le i < j \le k$, $0 \le \ell \le g^2 - 1$.

- **チャネリング制約**とは，拡張行列の各変数 $y_{r,(i,j)}$ と基本行列の各変数 $x_{r,i}$ と $x_{r,j}$ を関係づける制約であり，以下のように表される．

$$y_{r,(i,j)} = g x_{r,i} + x_{r,j} \tag{3}$$

ただし，$1 \le r \le b$, $1 \le i < j \le k$.

拡張行列モデルの特長は，“どの 2 つの列に対しても，g^2 通りある値の組合せすべてが高々1 つ存在する” という PA のパッキング制約が，“拡張行列の各列について，0 から $g^2 - 1$ までの値が高々1 つ存在する” という制約に置き換えられる点である．すなわち，パッキング制約は拡張列上の at_most_one 制約を使って表される．図 3 に，図 2 の $PA(9;4,3)$ に対応する拡張行列を示す．拡張行列の各列に 0 から 8 までの値が高々1 つだけ存在していることがわかる．

拡張行列の導入は一見冗長に思われるが，PA 問題に類似した CA 問題に対する有効性が実験的に示されている [4][23]．この拡張行列モデルの欠点は，基本行列モデルと同様に，パッキング制約に対して生成される at_most_one 制約の個数が $\binom{b}{2}\binom{k}{2}g^2$ と非常に多くなる点である．

3.3 拡張・相異モデル

拡張・相異モデルでは，CSP における最も代表的な**グローバル制約** [47] である alldifferent 制約を用いる．グローバル制約とは，引数として任意の数の変数をとり，それらの間に成り立つ関係を表現するものである．alldifferent$(X_1, X_2, \ldots, X_n)$ は，各変数 X_i に割り当てられる値がすべて異なることを表す制約である．拡張・相異モデルは，与えられた $PA(b;k,g)$ に対して，2 つの行列と 2 種類の制約から構成される．

- 基本行列 (基本行列モデルと同じ)
- 拡張行列 (拡張行列モデルと同じ)
- パッキング制約は以下のように表される．

$$\mathsf{alldifferent}\left(y_{1,(i,j)}, y_{2,(i,j)}, \ldots, y_{b,(i,j)}\right) \tag{4}$$

ただし，$1 \le i < j \le k$.

- チャネリング制約 (拡張行列モデルと同じ)

拡張・相異モデルの特長は，“どの 2 つの列に対しても，g^2 通りある値の組合せすべてが高々1 つ存在する” という PA のパッキング制約が，“拡張行列の各列に出現する値がすべて異なる” という制約に置き換えられ，$\binom{k}{2}$ 個の alldifferent 制約によって簡潔に表現される点である．図 3 から，拡張行列の各列に出現する値がすべて異なっていることがわかる．

3.4 基本・相異モデル

拡張・相異モデルは，基本行列モデル，拡張行列モデルと比較して，パッキング制約を alldifferent 制約を用いて簡潔に表現することができる．しかし，SAT 符号化を用いる場合，符号化後の節数が大きくなるという問題は依然として残る．この問題を解決するために，拡張行列上の alldifferent 制約を基本行列上の $\neq$ 制約に置き換えることにより，符号化後の節数を小さく抑える基本・相異モデルを提案する．

基本・相異モデルは，与えられた $PA(b;k,g)$ に対して，1 つの行列と 1 種類の制約から構成される．

- 基本行列 (基本行列モデルと同じ)
- パッキング制約は以下のように表される．

$$(x_{r,i} \neq x_{r',i}) \quad \vee \quad (x_{r,j} \neq x_{r',j}) \tag{5}$$

ただし，$1 \le r < r' \le b$, $1 \le i < j \le k$.

制約 (5) は，拡張・相異モデルのパッキング制約 (4) を以下のように分解することで得られる．

$$\begin{aligned} & \mathsf{alldifferent}\left(y_{1,(i,j)}, y_{2,(i,j)}, \ldots, y_{b,(i,j)}\right) \\ \iff & \bigwedge_{1 \le r < r' \le b} \left(y_{r,(i,j)} \neq y_{r',(i,j)}\right) \\ \iff & \bigwedge_{1 \le r < r' \le b} \left((x_{r,i} \neq x_{r',i}) \vee (x_{r,j} \neq x_{r',j})\right) \end{aligned}$$

表 2 制約モデルの分類

	パッキング制約	
	at_most_one 制約	alldifferent 制約
基本行列	基本行列モデル	基本・相異モデル
拡張行列	拡張行列モデル	拡張・相異モデル

このように，パッキング制約を基本行列上の $\neq$ 制約で表すことにより，拡張行列およびチャネリング制約は不要となり，生成される節数を小さく抑えることができる．節数に関する詳細な比較は節 4.2 で述べる．

最後に，これまで述べた 4 つの制約モデルについて，PA のパッキング制約をどの行列上でどのような制約を使って表現するか，によって分類した表を示す (表 2 参照)．

4 SAT 符号化

4.1 順序符号化法の概要

順序符号化法 [40][41] 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 $\{0,1,2,3\}$ の場合，3 つの命題変数 $p(z \le 0)$, $p(z \le 1)$, $p(z \le 2)$ を用い，以下の節を追加する．

$$\neg p(z \le 0) \vee p(z \le 1) \quad \neg p(z \le 1) \vee p(z \le 2)$$

この時，上記の節を充足可能にする真理値割り当ては 4 通りあり，それぞれ $z=0$, $z=1$, $z=2$, $z=3$ に対応する．

$p(z \le 0)$	$p(z \le 1)$	$p(z \le 2)$	解釈
1	1	1	$z=0$
0	1	1	$z=1$
0	0	1	$z=2$
0	0	0	$z=3$

制約については，制約に違反する範囲を符号化すればよい．一般の線形制約については，以下のようになる．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．

$$\bigwedge_{\Sigma_{i=1}^{n} b_i = c-n+1} \bigvee_i (a_i z_i \le b_i)^{\#}$$

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 $\{0,1,2,3\}$ の時，制約 $w - z \le -1$ は以下の 4 つの節に符号化される．

$$\neg p(z \le 0) \qquad p(w \le 0) \vee \neg p(z \le 1)$$
$$p(w \le 1) \vee \neg p(z \le 2) \qquad p(w \le 2)$$

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．図 4 に $w - z \le -1$ の充足領域を表す．制約を満たす点は ●，違反点は × でプロットされている．

$w \ne z$ の場合，一旦 $w - z \le -1 \vee z - w \le -1$ の

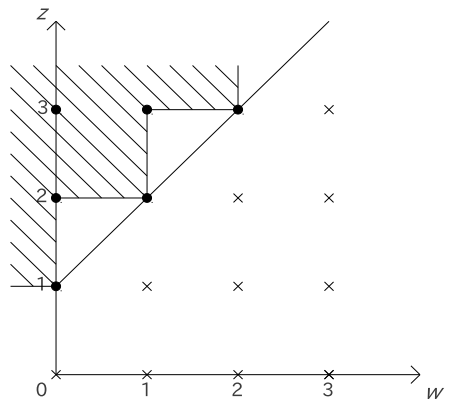


図 4 $w - z \le -1$ の充足領域

ように線形制約に置き換え，さらに Tseitin 変換を用いて $(p \vee q) \wedge (\neg p \vee w - z \leq -1) \wedge (\neg q \vee z - w \leq -1)$ と変換してから符号化する (p, q は新しい命題変数). したがって $w \neq z$ は以下の 9 個の節に符号化される.

$$
\begin{gathered}
p \vee q \\
\neg p \vee \neg p(z \leq 0) \\
\neg p \vee p(w \leq 0) \vee \neg p(z \leq 1) \\
\neg p \vee p(w \leq 1) \vee \neg p(z \leq 2) \\
\neg p \vee p(w \leq 2) \\
\neg q \vee \neg p(w \leq 0) \\
\neg q \vee p(z \leq 0) \vee \neg p(w \leq 1) \\
\neg q \vee p(z \leq 1) \vee \neg p(w \leq 2) \\
\neg q \vee p(z \leq 2)
\end{gathered}
$$

順序符号化法では，直接符号化 [14] [34] [48]，支持符号化 [18] [26]，対数符号化 [25] [46] 等と比較し，より短い節が生成される．また，順序符号化法で生成した節に対しては，CDCL 型 SAT ソルバーにおける単位伝播が整数制約上での範囲伝播に対応することが知られている [42].

4.2 *PA*問題の順序符号化

節 3 で提案した 4 つの制約モデルを，順序符号化法を用いて SAT に符号化する方法を述べる．基本行列の整数変数 $x_{r,i}$，拡張行列の整数変数 $y_{r,(i,j)}$，拡張行列のチャネリング制約 (3) については，節 4.1 で述べたように符号化される．よって以下では，各制約モデルのパッキング制約の SAT 符号化について述べる.

基本行列モデル: パッキング制約 (1) は，以下のように符号化される.

$$
\begin{aligned}
& p(x_{r,i} \leq m-1) \quad \vee \quad \neg p(x_{r,i} \leq m) \quad \vee \\
& p(x_{r,j} \leq n-1) \quad \vee \quad \neg p(x_{r,j} \leq n) \quad \vee \\
& p(x_{r',i} \leq m-1) \quad \vee \quad \neg p(x_{r',i} \leq m) \quad \vee \\
& p(x_{r',j} \leq n-1) \quad \vee \quad \neg p(x_{r',j} \leq n)
\end{aligned}
$$

ただし，$1 \leq r < r' \leq b$, $1 \leq i < j \leq k$, $0 \leq m, n \leq g-1$. また，$p(x_{r,i} \leq -1)$ と $p(x_{r',j} \leq -1)$ は偽とする．例えば，最初の 2 行は $\neg(x_{r,i} = m) \vee \neg(x_{r,j} = n)$ を表している.

拡張行列モデル: パッキング制約 (2) は，以下のように符号化される.

$$
\begin{aligned}
& p(y_{r,(i,j)} \leq \ell - 1) \quad \vee \quad \neg p(y_{r,(i,j)} \leq \ell) \quad \vee \\
& p(y_{r',(i,j)} \leq \ell - 1) \quad \vee \quad \neg p(y_{r',(i,j)} \leq \ell)
\end{aligned}
$$

ただし，$1 \leq r < r' \leq b$, $1 \leq i < j \leq k$, $0 \leq \ell \leq g^2 - 1$. また，$p(y_{r,(i,j)} \leq -1)$ と $p(y_{r',(i,j)} \leq -1)$ は偽とする.

拡張・相異モデル: パッキング制約 (4) については，まず alldifferent 制約を以下のように分解する.

$$
\begin{aligned}
& \mathsf{alldifferent}\left(y_{1,(i,j)}, y_{2,(i,j)}, \ldots, y_{b,(i,j)}\right) \\
\Longleftrightarrow \quad & \bigwedge_{1 \leq r < r' \leq b} \left(y_{r,(i,j)} \neq y_{r',(i,j)}\right)
\end{aligned}
$$

ただし，$1 \leq i < j \leq k$. その後，各等号否定 $y_{r,(i,j)} \neq y_{r',(i,j)}$ は，節 4.1 で述べたように符号化される.

基本・相異モデル: パッキング制約 (5) については，新しい命題変数 p, q を導入し，まず以下のように置き換える.

$$
\begin{aligned}
& (x_{r,i} \neq x_{r',i}) \vee (x_{r,j} \neq x_{r',j}) \\
\Longleftrightarrow \quad & (p \vee q) \qquad\qquad\qquad\quad \wedge \\
& (\neg p \vee (x_{r,i} \neq x_{r',i})) \quad \wedge \\
& (\neg q \vee (x_{r,j} \neq x_{r',j}))
\end{aligned}
$$

ただし，$1 \leq r < r' \leq b$, $1 \leq i < j \leq k$. その後，各等号否定 $x_{r,i} \neq x_{r',i}$ は，節 4.1 で述べたように符号化される.

表 3 に各制約モデルについて，SAT 符号化後の節数を示す．与えられた $PA(b; k, g)$ に対して，基本・相異モデルを除く 3 つの制約モデルでは，パッキング制約の符号化に $O(g^2)$ の節数が必要になる．これに対して，基本・相異モデルは符号化後の節数を $O(g)$ に抑えることができる.

SAT 符号化の良し悪しについては，利用する SAT ソルバーの特性 (CDCL ソルバーか確率的ソルバーか) 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 2 つの基準がよく用いられる．(a) 符号化後の SAT 問題に対する SAT ソルバーの単位伝播と元の CSP に対する制約伝播との関連，(b) 符号化後の節数 (少ない方が良いとされる). (a) について，本論文で用いた順序符号化法は，SAT ソルバーの単位伝播が元の CSP に対する範囲伝播

表 3 $PA(b;k,g)$ 問題に対する SAT 符号化後の節数の比較

	基本行列モデル	拡張行列モデル	拡張・相異モデル	基本・相異モデル
基本行列	$bk(g-1)$	$bk(g-1)$	$bk(g-1)$	$bk(g-1)$
拡張行列	–	$b\binom{k}{2}(g^2-1)$	$b\binom{k}{2}(g^2-1)$	–
パッキング制約	$\binom{b}{2}\binom{k}{2}g^2$	$\binom{b}{2}\binom{k}{2}g^2$	$\binom{b}{2}\binom{k}{2}(2g^2+1)$	$\binom{b}{2}\binom{k}{2}(4g+3)$
チャネリング制約	–	$O\left(bk^2g^2\right)$	$O\left(bk^2g^2\right)$	–

に対応しており，効率の良い求解が可能である [42]. (b) については，表 3 に示すように，基本・相異モデルの節数が最も少なく，他のモデルよりも優れていると言える.

5 対称性の除去

パッキング配列は対称性が高く，対称解が多く存在する．パッキング配列には以下の対称性が存在することが知られている [39].

- 行と列に関する対称性
  与えられたパッキング配列に対して，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．例えば，図 5 は図 2 に挙げた $PA(9;4,3)$ の 1 列目と 2 列目を入れ替えることで得られる対称解である.
- 値に関する対称性
  与えられたパッキング配列に対して，任意の列中の値をスワップしたものも解となる．例えば，図 6 は図 2 に挙げた $PA(9;4,3)$ の 1 列目の値を (0 を 2, 1 を 0, 2 を 1 に) スワップすることで得られる対称解である.

CSP 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 [19]．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.

行と列に関する対称性を除去するためには，以下の 2 つの手法のいずれかを用いる．なお，これらは同時に使用することはできない.

- Double Lex [16]：基本行列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 (Lexicographic Order Constraint) を適用する.
- Snake Lex [20]：Double Lex の改良として提案されたスネーク順序 (Snake Ordering) に基づく対称性除去手法である．釣合い型不完備ブロック計画などに対する有効性が示されている.

1	2	3	4
0	**0**	0	0
1	**0**	2	2
2	**0**	1	1
0	**1**	1	2
1	**1**	0	1
2	**1**	2	0
0	**2**	2	1
1	**2**	1	0
2	**2**	0	2

図 5 列に関する対称解

1	2	3	4
2	0	0	0
2	1	2	2
2	2	1	1
0	0	1	2
0	1	0	1
0	2	2	0
1	0	2	1
1	1	1	0
1	2	0	2

図 6 値に関する対称解

値に関する対称性を除去するためには，与えられた $PA(b;k,g)$ に対して，新しく整数変数 $f_{i,v}(1 \le i \le k, 0 \le v \le g-1)$ を導入する．$f_{i,v}$ は列 i に出現する値 v の個数を表す．その上で，すべての列 i に対して制約 $f_{i,0} \le f_{i,1} \le \cdots \le f_{i,g-1}$ を適用する．以下，この制約を出現個数制約と呼ぶ．なお，出現個数制約は Double Lex および Snake Lex のいずれかと同時に使用することができる．これらの手法はパッキング配列の対称解を除去するものであり，解の存在性は保たれる.

6 実行実験

提案する制約モデルの有効性を比較・評価するために，組合せデザイン・ハンドブック [13] にある PA 最適化問題 (全 46 問，表 1 参照) の最適値 $PAN(k,g)$ を求める実験を行った.

6.1 実験概要

実験に用いたベンチマーク問題は以下の通りである．

ベンチマーク問題 (79 問)： 各 $PAN(k,g)$ の既知の下限 ℓ および上限 u に対して，b を $\ell \leq b \leq \min(\ell+5, u)$ のように変化させた $PA(b;k,g)$ 問題全 79 問．既知の下限 ℓ は筆者らの先行研究 [32] で得られた値を反映した値である．既知の上限 u は表 1 と同じである．b の上限を $\min(\ell+5, u)$ に制限しているのは，既知の上限 u は Plotkin bounds と呼ばれる符号理論の定理 [39] より得られた弱い上限であり，$\ell+5$ を超える大きな b をもつ PA 問題は現実的な時間内では解けないことが先行研究 [32] で確認されているためである．

次に，上記ベンチマーク問題 (PA 問題，79 問) に対して，制約モデル (4 通り)，対称性除去の有無 (3 通り) の計 $4 \times 3 = 12$ 通りの CSP 表現を生成した．

制約モデル (4 通り)： 基本行列モデル，拡張行列モデル，拡張・相異モデル，基本・相異モデルの 4 つの制約モデル．

対称性除去の有無 (3 通り)： Double Lex と出現個数制約の組合せ (以降，“D” で表示)，Snake Lex と出現個数制約の組合せ (“S” で表示)，対称性除去なし (“無” で表示) の 3 通り．

以降，これら 12 通りの組合せを，“制約モデル名 (SB)”，$SB \in \{\mathrm{D}, \mathrm{S}, \text{無}\}$ で表す．例えば，基本行列モデル (D) は，基本行列モデルに Double Lex と出現個数制約を適用したモデルを表す．

最後に，ベンチマーク問題 (79 問) に対して，上記の組合せ (12 通り) を用いて生成された計 $79 \times 12 = 948$ 通りの CSP 表現を，順序符号化法を用いて SAT に符号化し，以下の 3 種類の SAT ソルバーを用いて求解した．順序符号化を行うプログラムには，SAT 型 CSP ソルバー Sugar を使用した．

SAT ソルバー (3 種類)： glucose 2.0 [3], MiniSat 2.2.0 (core) [15], clasp 2.0.4 [17] の 3 種類の CDCL ソルバー．glucose と clasp は SAT 競技会 2011 および SAT チャレンジ 2012 と呼ばれる SAT ソルバー競技会の優勝ソルバー．

SAT ソルバーの CPU 時間は，Mac OS X (Xeon 2.66GHz，24GB メモリ) 上で計測し，各ベンチマーク問題に対する SAT ソルバーのタイムアウト時間 ($T.O$) は 3600 秒とした．

6.2 実験結果

表 4 に，制約モデルと対称性除去の組合せ 12 通りのいずれかで解けた問題 (42 問) について，求解に要した glucose の CPU 時間を示す．3 つのソルバーのうち，glucose の結果を示した理由は，glucose が全 79 問中最多の 42 問を解き，clasp のみで解けた 1 問を除いて，MiniSat が解いた 41 問，clasp が解いた 34 問をすべて含んでいるからである．3 列目の b の値に “$*$” が付いている場合は，その値が最適値であることを示している．各問題の一番良い CPU 時間をボールド体で記している．また，最後の行は，各モデルで解けた問題の合計数を示している．なお，表 4 中の PA 問題はすべて SAT であった．

解けた問題数については，基本・相異モデル (S) が全 79 問中最多の 31 問を，次いで基本・相異モデル (D) が 30 問，拡張・相異モデル (無) が 28 問，拡張行列モデル (無) が 26 問，基本行列モデル (無) が 25 問を解いており，問題によって有効なモデルは異なっているが，全体的には基本・相異モデルが優れた性能を示していることがわかる．また，対称性の除去は基本・相異モデルに対して効果があるが，他のモデルにおいては逆に解ける問題数が減る結果となった．

CPU 時間を比較するために，図 7 に表 4 のカクタスプロットを示す．横軸は解けた問題数を，縦軸は昇順に並び替えた SAT ソルバーの CPU 時間を示している．すなわち，この表はグラフが右にあるほど多くの問題を解き，下にあるほど高速に求解していることを示す．図 7 より，基本・相異モデル (S) と基本・相異モデル (D) が，他と比較して，高速に問題を解いていることがわかる．また，この 2 つを比較すると，2,200 秒付近までは基本・相異モデル (D) が良いが，それを超えると，基本・相異モデル (S) が逆転していることがわかる．

以上，ベンチマーク問題全 79 問に関する評価では，解けた問題数，求解に要した CPU 時間ともに，基本・相異モデルと対称性除去の組合せが最も効果的であった．この結果は，節 4.2 で述べた節数の比較 (図

表 4 ベンチマーク結果：CPU 時間 (秒)

	基本行列モデル			拡張行列モデル			拡張・相異モデル			基本・相異モデル		
k g b	無	D	S	無	D	S	無	D	S	無	D	S
4 3 9*	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**
5 3 6*	**<0.00**	**<0.00**	0.01	**<0.00**	**<0.00**	0.01	0.01	**<0.00**	0.01	0.01	**<0.00**	**<0.00**
6 3 4*	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**	**<0.00**
5 4 16*	**0.01**	**0.01**	0.02	**0.01**	0.02	0.02	0.02	0.03	0.07	0.09	0.04	0.02
6 4 9*	0.03	0.20	0.03	0.05	0.66	0.23	0.62	1.47	0.09	7.41	1.16	**0.02**
7 4 8*	5.12	96.72	0.33	37.83	88.74	**0.05**	0.27	*T.O*	*T.O*	0.41	0.18	2.34
8 4 5*	**<0.00**	**<0.00**	**<0.00**	**<0.00**	0.01	0.01	0.17	0.01	0.01	**<0.00**	**<0.00**	**<0.00**
6 5 25*	0.04	0.22	0.08	*T.O*	0.08	0.13	0.48	0.14	15.80	**0.03**	0.16	0.44
7 5 15*	**6.46**	*T.O*	*T.O*	15.27	*T.O*	*T.O*	70.95	*T.O*	*T.O*	62.62	31.54	17.35
8 5 10*	0.05	0.07	3.02	**0.03**	0.07	477.43	1.89	1309.21	1399.11	0.31	0.06	0.11
9 5 10*	*T.O*	14.93	*T.O*	*T.O*	*T.O*	*T.O*	27.56	*T.O*	592.72	5.83	1.09	**0.93**
10 5 7*	**0.02**	0.46	**0.02**	0.03	0.16	0.05	401.60	0.03	5.96	0.06	0.04	0.04
3 6 36*	0.03	0.10	0.07	0.10	0.10	0.07	0.05	0.11	0.10	**0.02**	0.07	0.05
4 6 34*	*T.O*	**0.36**	4.22	*T.O*	1.95	4.55	471.42	3.12	1.86	1.23	26.41	31.54
5 6 31	*T.O*	*T.O*	185.56	*T.O*	**116.62**	209.01	*T.O*	951.94	*T.O*	*T.O*	625.76	3500.12
6 6 31	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	**20.50**	*T.O*	*T.O*	*T.O*	*T.O*
8 6 17	24.35	*T.O*	*T.O*	**17.42**	*T.O*	*T.O*	243.48	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
9 6 14*	**0.93**	*T.O*	*T.O*	10.17	*T.O*	*T.O*	20.99	*T.O*	*T.O*	26.77	*T.O*	*T.O*
10 6 12*	**0.07**	*T.O*	0.13	0.11	0.45	9.35	2.87	30.09	0.68	0.36	0.28	0.32
11 6 12*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	3.21	*T.O*	2209.84	**0.35**	5.90
12 6 9*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	**0.14**	0.36	0.57
8 7 49*	*T.O*	*T.O*	*T.O*	*T.O*	**2440.90**	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
11 7 15*	1.96	142.95	17.99	9.05	181.68	369.41	**1.59**	34.88	449.68	3.62	25.22	17.81
12 7 14*	2382.49	2.45	37.97	462.93	9.39	6.90	1632.34	*T.O*	114.61	405.01	**0.17**	0.27
13 7 14*	*T.O*	16.66	*T.O*	*T.O*	*T.O*	989.76	*T.O*	*T.O*	*T.O*	*T.O*	2.50	**0.96**
14 7 10*	0.17	3.55	0.14	0.24	*T.O*	1.23	0.66	4.99	0.65	**0.05**	0.46	0.34
10 8 25	1058.90	*T.O*	*T.O*	**791.83**	*T.O*	*T.O*	2048.01	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
11 8 21	14.07	*T.O*	*T.O*	**6.62**	*T.O*	*T.O*	104.01	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
11 8 22	**781.94**	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	1013.51	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
12 8 18	**8.73**	*T.O*	*T.O*	14.69	*T.O*	*T.O*	44.59	*T.O*	*T.O*	50.07	3467.89	2283.27
12 8 19*	*T.O*	*T.O*	*T.O*	**562.61**	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
13 8 17*	*T.O*	*T.O*	218.30	**134.17**	*T.O*	*T.O*	2536.77	*T.O*	*T.O*	*T.O*	*T.O*	1802.67
14 8 16*	*T.O*	30.94	52.43	*T.O*	28.96	9.97	*T.O*	223.42	160.32	*T.O*	6.10	**2.49**
15 8 16*	*T.O*	*T.O*	*T.O*	*T.O*	880.94	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	8.81	**7.66**
16 8 12*	1404.82	*T.O*	17.75	2657.45	297.48	*T.O*	1265.67	*T.O*	*T.O*	**0.18**	8.57	0.29
11 9 28	30.01	*T.O*	*T.O*	**17.42**	*T.O*	*T.O*	79.05	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
11 9 29	**56.26**	*T.O*	*T.O*	392.92	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
14 9 20	*T.O*	*T.O*	*T.O*	1815.91	*T.O*	*T.O*	**670.59**	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*
15 9 18	*T.O*	**7.30**	21.44	*T.O*	106.00	24.97	*T.O*	18.74	25.49	50.68	10.83	8.83
16 9 18*	*T.O*	318.82	221.87	*T.O*	324.66	*T.O*	*T.O*	585.65	2773.84	*T.O*	39.60	**19.79**
17 9 18*	*T.O*	3339.23	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	**99.14**	386.88
18 9 14*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	*T.O*	**7.27**	828.76	2057.81
合計	25	21	22	26	22	20	28	20	19	25	30	**31**

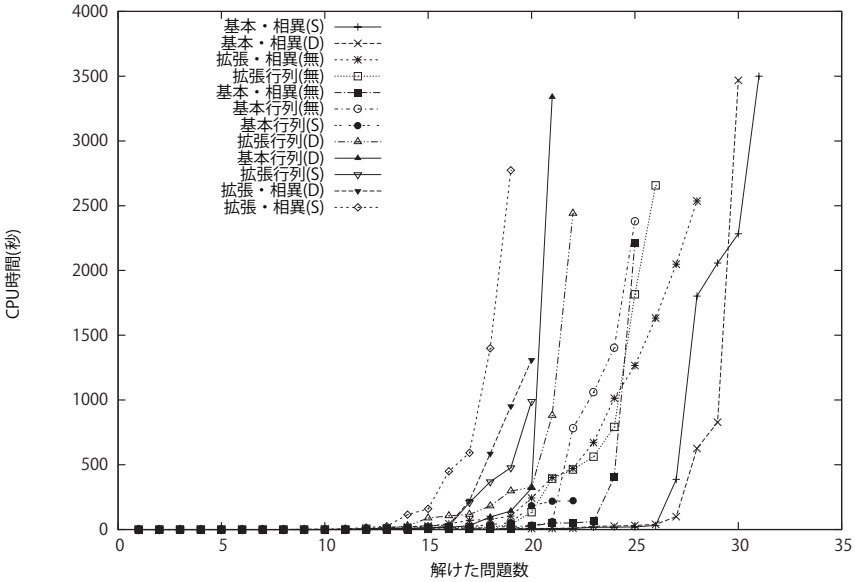


図 7　表 4 のカクタスプロット

3 参照) に関する評価結果とも一致している．

6.3　新しく得られた結果

表 5 に，今回の実験で新しく得られた結果を示す．これら 7 問の内訳は，これまで最適値が未知であった問題に対して，既知の上限が最適値であることを新しく決定した 2 問と，新しい下限を見つけることに成功した 5 問である．例えば，$PAN(12,8)$ の既知の上下限は，表 1 より $16 \leq PAN(12,8) \leq 19$ であった．しかし，表 4 から $PA(19;12,8)$ は SAT であるため，$PAN(12,8) = 19$ であることが決定できた．

これら 7 問を求めた制約モデルの内訳は，基本行列モデル (無) が 7 問，拡張行列モデル (無) と拡張・相異モデル (無) が各 5 問，基本行列モデル (S)，基本・相異モデル (無)，基本・相異モデル (S)，基本・相異モデル (D) が各 1 問となっている．また，SAT ソルバーは glucose と MiniSat が比較的良いものの，7 問全てを解くソルバーはない．例えば，$PA(23;7,6)$ は基本行列モデル (無) と clasp の組合せのみが求解に成功している．

以上，新しく得られた結果に関する評価では，基本行列モデル (無) が他のモデルより良い結果を示している．しかし，この結果は，前節で述べたベンチマーク問題全 79 問に関する結果とは大きく異なるものとなった．次節では，“何故このような違いが生じたのか” ついて，より深く考察を行う．

表 5　新しい結果

これまでの結果 [13] [38]	新しく得られた結果
$20 \leq PAN(7,6)$	$\mathbf{23} \leq PAN(7,6)$
$22 \leq PAN(10,8)$	$\mathbf{25} \leq PAN(10,8)$
$16 \leq PAN(11,8)$	$\mathbf{22} \leq PAN(11,8)$
$16 \leq PAN(12,8) \leq 19$	$PAN(12,8) = \mathbf{19}$
$16 \leq PAN(13,8) \leq 17$	$PAN(13,8) = \mathbf{17}$
$27 \leq PAN(11,9)$	$\mathbf{29} \leq PAN(11,9)$
$18 \leq PAN(14,9)$	$\mathbf{20} \leq PAN(14,9)$

6.4　考察

与えられた $PA(b;k,g)$ に対して，その縦横比 $\frac{k}{b}$ を考える．すなわち，$\frac{k}{b} < 1$ であれば，与えられた $PA(b;k,g)$ は縦長，$\frac{k}{b} > 1$ であれば横長である．本節では，この $\frac{k}{b}$ の値が各制約モデルの求解性能に関係しているのではないかという仮説を基に，基本・相異モデルと基本行列モデルについて比較考察を行う．

図 8 は，横軸が昇順に並び替えた $\frac{k}{b}$ の値，縦軸が解けた問題の累積数を表している．図より，$\frac{k}{b} = 0.9$

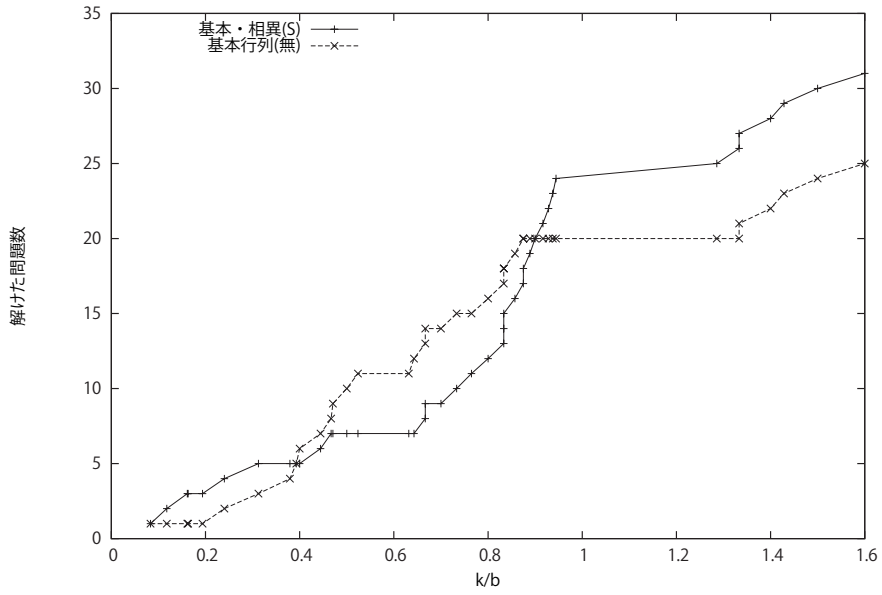


図 8　基本・相異モデル (S) と基本行列モデル (無) の比較

表 6　基本・相異モデルと基本行列モデルの比較

		基本行列モデル			基本・相異モデル		
範囲	問題数	無	D	S	無	D	S
$\frac{k}{b} \leq 0.7$	56	**14**	6	7	9	9	9
$\frac{k}{b} > 0.7$	23	11	15	15	16	21	**22**
合計	79	25	21	22	25	30	**31**

付近で 2 つのモデルの累積求解数が逆転し，最終的に基本・相異モデル (S) が 31 問，基本行列モデル (無) が 25 問を解いていることがわかる．また，$\frac{k}{b} = 0.7$ 付近から，基本・相異モデル (S) の求解性能が右上がりに向上している点が特徴的である．

表 6 は，解けた問題数を $\frac{k}{b} = 0.7$ の前後で 2 つに分けて示したものである．$\frac{k}{b} \leq 0.7$ では，基本行列モデル (無) が最も多い 14 問を解き，$\frac{k}{b} > 0.7$ では，基本・相異モデル (S) が最も多い 22 問を解いている．このように，$\frac{k}{b} = 0.7$ の前後で，2 つのモデルの求解性能に大きな違いがあることが確認できた．

この $\frac{k}{b}$ を指標に用いた定量的な観察結果を基に，新しく得られた結果 (表 5 の 7 問) を見ると，全てのモデルで解けた $PA(17; 13, 8)$ を除いて，それらの $\frac{k}{b}$ の値は 0.7 以下であり，基本行列モデル (無) が得意とする問題であった．$\frac{k}{b} = 0.7$ の定性的な意味に関する考察は，今後の重要な課題である．

最後に，対称性除去の効果について考察する．$\frac{k}{b} > 0.7$ では，どちらのモデルでも求解数が増えており効果が認められる．$\frac{k}{b} \leq 0.7$ の場合，基本行列モデルで求解数が減っている原因としては，値に関する対称性を除去する制約の SAT 符号化に $O(b^2kg)$ の節数が必要となり，b の大きな縦長の PA に対して符号化後の節数増大をまねく点が挙げられる．

7　まとめ

本論文では，組合せデザイン分野の PA 問題を取り上げ，順序符号化法と CDCL ソルバーに適した制約モデリングについて考察を行った．提案した 4 つの制約モデルのうち，特に，基本・相異モデルは，与えられた PA のパッキング制約について，符号化後の節数を小さく抑えることができる点が特長である．

提案モデルの有効性を比較・評価するために，組合せデザイン・ハンドブック [13] にある PA 最適化問題 (全 46 問) を用いて実行実験を行った結果，最適値がこれまで未知であった 2 問について，既知の上限が

最適値であることを示し，5 問について新しい下限を得ることに成功した．与えられた $PA(b;k,g)$ の $\frac{k}{b}$ の値を指標に用いて，提案モデルの比較考察を行った．その結果，新しい結果が得られた 7 問については，全てのモデルで解けた 1 問を除いて，$\frac{k}{b} \leq 0.7$ であり，基本行列モデル (無) が得意とする問題であった．また，$\frac{k}{b} > 0.7$ の問題については，基本・相異モデルと対称性除去の組合せが最も効果的であることが確認できた．

本論文で提案した制約モデルの基本部分は，CSP 分野のモデリング技法である行列モデル [16] とグローバル制約 [47] の効果的な利用に基づいている．その上に，符号化後の節数を小さく抑える等の CSP を SAT で効率よく解くための工夫を加えることにより，PA 問題に対して優れた結果を得ることができた．本研究では，基本の制約モデルおよびそれを変形したモデルを問題に対して適用し，各手法の特徴を解析・比較した．このようなアプローチは，これまで CSP 分野で研究および蓄積されてきた様々な組み合わせ問題・組合せ最適化問題のモデリングにおいても有用な手法であると考えている．ただし，対称性除去が有効に働く問題の構造に関する考察等は今後の重要な課題である．

参 考 文 献

[1] Abdel-Ghaffar, K. A. S.: On the Number of Mutually Orthogonal Partial Latin Squares, *Ars Combinatoria*, Vol. 42(1996), pp. 259–286.

[2] Abdel-Ghaffar, K. A. S. and Abbadi, A. E.: Optimal Disk Allocation for Partial Match Queries, *ACM Transactions on Database Systems*, Vol. 18, No. 1(1993), pp. 132–156.

[3] 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 2009), 2009, pp. 399–404.

[4] 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), 2010, pp. 112–126.

[5] Banbara, M., Tamura, N. and Inoue, K.: Generating Event-Sequence Test Cases by Answer Set Programming with the Incidence Matrix, in *Technial Communications of the 28th International Conference on Logic Programming* (ICLP 2012), Dovier, A. and Costa, V. S.(eds.), LIPIcs, Vol. 17, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012, pp. 86–97.

[6] Béjar, R., Hähnle, R. and Manyà, F.: A Modular Reduction of Regular Logic to Classical Logic, in *Proceedings of the 31st IEEE International Symposium on Multiple-Valued Logic* (ISMVL 2001), 2001, pp. 221–226.

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

[8] Bulutoglu, D. and Margot, F.: Classification of Orthogonal Arrays by Integer Programming, *Journal of Statistical Planning and Inference*, Vol. 138(2008), pp. 654–666.

[9] 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.

[10] 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.

[11] 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.

[12] Cohen, M. B., Gibbons, P. B., Mugridge, W. B. and Colbourn, C. J.: Constructing Test Suites for Interaction Testing, in *Proceedings of the 25th International Conference on Software Engineering* (ICSE 2003), 2003, pp. 38–48.

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

[14] de Kleer, J.: A Comparison of ATMS and CSP Techniques, in *Proceedings of the 11th International Joint Conference on Artificial Intelligence* (IJCAI 1989), 1989, pp. 290–296.

[15] 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, Springer, 2003, pp. 502–518.

[16] 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, Springer, 2002, pp. 462–476.

[17] 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 2007), MIT Press, 2007, pp. 386–392.

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

[19] Gent, I. P., Petrie, K. E. and Puget, J.-F.: Symmetry in Constraint Programming, *Handbook of Constraint Programming*, Rossi, F., van Beek, P. and Walsh, T.(eds.), Elsevier, 2006, pp. 329–376.

[20] 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, Springer, 2009, pp. 391–399.

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

[22] Hnich, B., Prestwich, S. D. and Selensky, E.: Constraint-Based Approaches to the Covering Test Problem, in *Proceedings of the International Workshop on Constraint Solving and Constraint Logic Programming* (CSCLP 2004), LNCS 3419, Springer, 2004, pp. 172–186.

[23] 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.

[24] 井上克巳, 田村直之: SAT ソルバーの基礎, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 57–67.

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

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

[27] 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.

[28] 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(2010), pp. 2316–2318.

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

[30] Meagher, K. and Stevens, B.: Group Construction of Covering Arrays, *Journal of Combinatorial Designs*, Vol. 13, No. 1(2005), pp. 70–77.

[31] 鍋島英知, 宋剛秀: 高速 SAT ソルバーの原理, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 68–76.

[32] 則武治樹, 番原睦則, 田村直之, 井上克巳: 制約充足問題の SAT 符号化を用いたパッキング配列の構成, 2012 年度人工知能学会全国大会 (第 26 回) 論文集 1E3-OS-4-1, 人工知能学会, 2012.

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

[34] 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 1992), 1992, pp. 440–446.

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

[36] Smith, B. M.: Modelling, *Handbook of Constraint Programming*, Rossi, F., van Beek, P. and Walsh, T.(eds.), Elsevier, 2006, pp. 377–406.

[37] Soh, T., Inoue, K., Tamura, N., Banbara, M. and Nabeshima, H.: A SAT-based Method for Solving the Two-dimensional Strip Packing Problem, *Fundamenta Informaticae*, Vol. 102, No. 3–4(2010), pp. 467–487.

[38] Stardom, J.: Metaheuristics and the Search for Covering and Packing Arrays, Master's thesis, Simon Fraser University, Burnaby, Canada, 2001.

[39] Stevens, B. and Mendelsohn, E.: Packing arrays, *Theoretical Computer Science*, Vol. 321, No. 1(2004), pp. 125–148.

[40] 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, Springer, 2006, pp. 590–603.

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

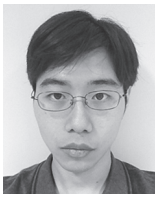
[42] 田村直之, 丹生智也, 番原睦則: 制約最適化問題と SAT 符号化, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 77–85.

[43] Tanjo, T., Tamura, N. and Banbara, M.: A Compact and Efficient SAT-Encoding of Finite Domain CSP, in *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing* (SAT 2011), Sakallah, K. A. and Simon, L.(eds.), LNCS 6695, Springer, 2011, pp. 375–376.

[44] 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.

[45] Tung, Y.-W. and Aldiwan, W. S.: Automating Test Case Generation for the New Generation Mission Software System, in *Proceedings of IEEE Aerospace Conference*, 2000, pp. 431–437.

[46] Van Gelder, A.: Another Look at Graph Coloring via Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 156, No. 2(2008), pp. 230–243.
[47] van Hoeve, W. J. and Katriel, I.: Global Constraints, *Handbook of Constraint Programming*, Rossi, F., van Beek, P. and Walsh, T.(eds.), Elsevier, 2006, pp. 169–208.
[48] Walsh, T.: SAT v CSP, in *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming* (CP 2000), 2000, pp. 441–456.
[49] Williams, A. W.: Determination of Test Configurations for Pair-Wise Interaction Coverage, in *Proceedings of 13th International Conference on Testing Communicating Systems* (TestCom 2000), 2000, pp. 59–74.

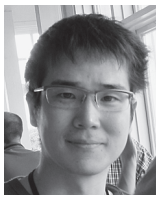


則武治樹

2012 年神戸大学情報知能工学科卒業. 2012 年神戸大学大学院システム情報学研究科情報科学専攻博士課程前期課程入学. 2014 年同大学大学院修了予定. 制約プログラミング, SAT 技術の応用に興味をもつ.

番原睦則

1994 年神戸大学理学部数学科卒業. 1996 年同大学大学院自然科学研究科博士課程前期数学専攻修了. 1996 年国立奈良工業高等専門学校助手. 1998 年同校講師. 2003 年より神戸大学情報基盤センター講師. 2007 年同校准教授. 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.



宋　剛秀

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

田村直之

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

井上克巳

1982 年京都大学工学部数理工学科卒業. 1984 年同大学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. 京都大学博士 (工学). 人工知能, システム生物学, システムズ・レジリエンスなどの研究に従事.