1E3-OS-4-1

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

Constructing Packing Arrays by SAT Encoding

則武 治樹[*1] 番原 睦則[*2] 田村 直之[*2] 井上 克巳[*3]
Haruki Noritake Mutsunori Banbara Naoyuki Tamura Katsumi Inoue

[*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 情報基盤センター
Information Science and Technology Center, Kobe University

[*3] 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

Remarkable improvements in the efficiency 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 consider the problem of finding optimal packing arrays (PAs) by SAT encoding. PAs have been applied to optimal disk allocation in Databases. We first present two CSP representations of packing arrays: the basic matrix model and the extended matrix model. Particularly, in the extended matrix model, the packing constraints of the given PA can be concisely expressed by using alldifferent constraints. We then present a new SAT encoding based on the order encoding. In our experiments, we succeeded either in proving the optimality of known bounds or in improving known lower bounds for many arrays listed in Handbook of Combinatorial Designs.

1. はじめに

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，様々な分野への実用的応用が急速に拡大している [3, 7]．特に，制約充足問題 (Constraint Satisfaction Problem; CSP) を SAT 問題に符号化 (**SAT 符号化** [10]) して，高速な SAT ソルバーを用いて求解する研究が注目を集め，これまでに数多くの符号化法が提案されている．

なかでも，順序符号化法 [9] は，整数有限領域上の CSP を SAT に変換する方法の 1 つであり，ショップ・スケジューリング問題，組合せテストケース生成問題など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 [10]．また，順序符号化法の有効性は，その符号化法を実装した制約ソルバー Sugar が 2008–2009 年国際制約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．

パッキング配列 (Packing arrays; PA) は，組合せデザイン分野の問題の 1 つである [1, 5, 8]．PA は直交表 (Orthogonal arrays) の拡張であり，応用分野としては，データベースのディスク最適配置などがある [2]．

本稿では，まず PA を構成する問題に対して，基本行列モデルと拡張行列モデルの 2 つの CSP 表現を提案する．特に，拡張行列モデルは，与えられた PA のパッキング制約をグローバル制約を用いて簡潔に表現することができる点が特長である．次に，拡張行列モデルに対する新しい SAT 符号化法を提案する．この符号化法は順序符号化法をベースに，符号化に必要な節数が少なくなるように最適化されている点が特長である．

提案手法の有効性を評価するために，組合せデザイン・ハンドブック [5] 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，提案手法は最適値が未知であった 3 問について既知の上限値が最適値であることを示し，さらに，7 問について新しい下限値を得ることに成功した．

連絡先: 則武 治樹，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，noritake@stu.kobe-u.ac.jp

2. パッキング配列と関連研究

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 が高々一つ存在する (**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．

図 1 に，最適な $PA(9; 4, 3)$ の例を示す．最初の 2 列について、全部で $3^2 = 9$ 通りあるそれらの値の組み合わせすべてが，高々一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 列の組み合わせについても同様の性質を満たす．

本稿では，SAT 符号化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* 判定問題**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 (最適な)$PA(b; k, g)$ を構成する問題である．PA 最適化問題に対する既存研究としては，グラフ理論等を用いた数学的手法 [1, 8]，整数線形計画法を用いた手法 [4] がある．

パッキング配列と類似したものとして，被覆配列 (Covering Array; CA) がある．CA は PA のパッキング制約 (定義 1 の 2) 中の，“ 高々一つ ” を “ 少なくとも一つ ” に置き換えたものである．CA はソフトウェアテスト等の分野に応用されている．最小の CA を構成する問題に対しては，これまで群論などを用いた数学的手法，貪欲法，局所探索法，CSP および SAT 符号化 [6]，を用いた解法が提案されている．

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 1: 最適な $PA(9;4,3)$ の例

(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
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 2: 図 1 の $PA(9;4,3)$ に対応する拡張行列

3. 制約充足問題としての表現

3.1 基本行列モデル

与えられた PA 判定問題 $PA(b;k,g)$ に対して，基本行列モデルは，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．
- パッキング制約は以下のように表される．ただし，$1 \le r < r' \le b,\ 1 \le i < j \le k,\ 0 \le m,n \le g-1$.

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

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 つ出現することを表している．この CSP 表現の欠点は，パッキング制約に対して生成される制約式の個数が $\binom{b}{2}\binom{k}{2}g^2$ 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，制約プログラミングにおける代表的な**グローバル制約** [11] の 1 つである 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$(X_1, X_2 \ldots X_\ell)$ は，任意の 2 つの変数に割り当てられる値が相異なることを表す制約である．

与えられた PA 判定問題 $PA(b;k,g)$ に対して，拡張行列モデルは，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．ただし，$1 \le i < j \le k$.

$$\text{alldifferent}\left(y_{1,(i,j)} \cdots, y_{b,(i,j)}\right) \quad (2)$$

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

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

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 つ出現する” という PA のパッキング制約は，“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 2 に，図 1 の $PA(9;4,3)$ に対応する拡張行列を示す．拡張行列の各列に出現する値がすべて異なっていることがわかる．

Hnich らの CA 判定問題に対する CSP 表現 [6] との違いは，提案する CSP 表現が 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．

4. SAT 符号化

4.1 順序符号化法の概要

順序符号化法 [9] 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，制約に違反する範囲を符号化する．例えば，整数変数 w，z のドメインが $\{0,1,2\}$ の場合，算術比較 $w \le z$ は以下の 2 つの節に符号化される．

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

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 $\sum_{i=1}^{n} a_i\, z_i \leq c$ は以下のように符号化できる．

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

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．

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

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

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

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

$w \neq z$ の場合，一旦 $w - z \leq -1 \vee z - w \leq -1$ の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 7 個の節に符号化される．

$$\begin{array}{cc} & p \vee q \\ \neg p \vee \neg p(z \leq 0) & \neg q \vee \neg p(w \leq 0) \\ \neg p \vee p(w \leq 0) \vee \neg p(z \leq 1) & \neg q \vee p(z \leq 0) \vee \neg p(w \leq 1) \\ \neg p \vee p(w \leq 1) & \neg q \vee p(z \leq 1) \end{array}$$

4.2 PAの順序符号化

基本行列モデル，拡張行列モデルについて，順序符号化法を用いた SAT 符号化について述べる．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，4.1 節で述べたように符号化される．

基本行列モデルのパッキング制約 (1) は，以下のように符号化される．ただし，$1 \leq r < r' \leq b$, $1 \leq i < j \leq k$, $0 \leq m, n \leq g - 1$.

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

例えば，1 行目は $\neg(x_{r,i} = m)$ を表している．

拡張行列モデルのパッキング制約 (2) については，まず alldifferent 制約を以下のように分解する．ただし，$1 \leq i < j \leq k$.

$$\begin{array}{cl} & \mathsf{alldifferent}\left(y_{1,(i,j)} \cdots, y_{b,(i,j)}\right) \\ \Longleftrightarrow & \bigwedge_{1 \leq r < r' \leq b} \left(y_{r,(i,j)} \neq y_{r',(i,j)}\right) \end{array}$$

その後，各等号否定 $y_{r,(i,j)} \neq y_{r',(i,j)}$ を，4.1 節で述べたように符号化する．

表 1: $PA(b; k, g)$ に対する節数の比較

	基本行列モデル 順序符号化法	拡張行列モデル 順序符号化法	拡張行列モデル 最適化手法
基本行列	$bk(g-1)$	$bk(g-1)$	$bk(g-1)$
拡張行列	–	$b\binom{k}{2}(g^2-1)$	–
パッキング	$\binom{b}{2}\binom{k}{2}g^2$	$\binom{b}{2}\binom{k}{2}(2g^2+1)$	$\binom{b}{2}\binom{k}{2}(4g+1)$
チャネリング	–	$O\left(b\binom{k}{2}g^2\right)$	–

4.3 順序符号化の最適化

拡張行列モデルは，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．しかしその反面，SAT 符号化に必要な節数が非常に大きくなるという問題がある．この問題を解決するために，拡張行列上のパッキング制約を基本行列上の制約に置き換えることにより，符号化に必要な節数を小さく抑える最適化手法を提案する．

提案手法では，拡張行列モデルのパッキング制約 (2) を以下のように分解した後，各等号否定 $x_{r,i} \neq x_{r',i}$ と $x_{r,j} \neq x_{r',j}$ を 4.1 節で述べたように符号化する．

$$\begin{array}{cl} & \mathsf{alldifferent}\left(y_{1,(i,j)} \cdots, y_{b,(i,j)}\right) \\ \Longleftrightarrow & \bigwedge_{1 \leq r < r' \leq b} \left(y_{r,(i,j)} \neq y_{r',(i,j)}\right) \\ \Longleftrightarrow & \bigwedge_{1 \leq r < r' \leq b} \left((x_{r,i} \neq x_{r',i}) \vee (x_{r,j} \neq x_{r',j})\right) \end{array}$$

このように，パッキング制約を基本行列上の等号否定で表すことができるため，拡張行列およびチャネリング制約の符号化は不要となり，生成される節数を小さく抑えることができる．また見方を変えると，分解後の CSP 表現は基本行列モデルの変形になっており，基本行列モデルの最適化手法とも呼べる．

符号化法の有効性については，解く問題の性質および利用する SAT ソルバーの特性 (系統的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) 符号化に必要な節数．(a) について，提案手法のベースとなる順序符号化法は，SAT ソルバーの単位伝播が元の CSP に対す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 [10]．(b) については，表 1 に示すように，提案する最適化手法は 4.2 節で述べた拡張行列モデルの順序符号化と比較して，非常に少ない節数に抑えることができる．また，基本行列モデルの順序符号化と比較しても，大きな $g \geq 5$ に対しては少ない節に抑えることができる．

5. 実行実験

提案手法の有効性を評価するために，組合せデザイン・ハンドブック [5] にある PA 最適化問題 (全 46 問) を用いて実験を行った．比較に用いた提案システムは，以下の通りである．

基本システム： 順序符号化法を用いて，基本行列モデルを SAT に符号化し，SAT ソルバーを用いて求解．

拡張システム： 順序符号化法を用いて，拡張行列モデルを SAT に符号化し，SAT ソルバーを用いて求解．

最適化システム： 順序符号化法に最適化手法を適用した新しい符号化法 (4.3 節) を用いて，拡張行列モデルを SAT に符号化し，SAT ソルバーを用いて求解．

表 2: ベンチマーク結果：$PA(b;k,g)$

k	g	b	基本システム	拡張システム	最適化システム
4	3	9*	**<0.00**	**<0.00**	**<0.00**
5	3	6*	**<0.00**	**<0.00**	**<0.00**
6	3	4*	**<0.00**	**<0.00**	**<0.00**
5	4	16*	**<0.00**	0.03	0.03
6	4	9*	**0.10**	0.25	0.12
7	4	8*	0.16	0.27	**0.11**
8	4	4*	**<0.00**	**<0.00**	**<0.00**
6	5	25*	**0.06**	0.20	0.14
7	5	15*	121.30	**6.94**	474.48
8	5	10*	**0.08**	0.12	0.13
9	5	10*	144.83	**1.59**	1.66
10	5	7*	0.27	0.04	**0.01**
3	6	36*	0.07	0.09	**0.04**
4	6	34*	**0.08**	0.98	2.02
5	6	30	16.95	**12.74**	649.14
6	6	31	*T.O*	*T.O*	**5367.57**
7	6	23	**3284.32**	*T.O*	*T.O*
8	6	17	**64.84**	420.02	922.15
9	6	14*	**22.32**	1186.76	101.68
10	6	12*	5.04	**0.46**	0.56
11	6	12*	*T.O*	*T.O*	**52.47**
12	6	9*	66.36	1.24	**0.11**
11	7	15*	41.70	7.99	**6.21**
12	7	14*	**2.22**	2935.23	45.80
14	7	10*	2.73	2.52	**0.06**
10	8	25	*M.O*	**2884.9**	*T.O*
11	8	21	*M.O*	**226.29**	721.59
13	8	17*	*M.O*	**265.56**	1130.99
16	8	12*	895.25	3832.76	**0.19**
11	9	28	*M.O*	**146.00**	*T.O*
14	9	20	*M.O*	3006.43	**707.49**
15	9	18*	*M.O*	1304.15	**59.92**
18	9	14*	*M.O*	*T.O*	**16.21**

表 3: 新しく得られた結果

これまでの結果 [5]	新しい結果
$30 \leq PAN(6,6) \leq 34$	$\mathbf{31} \leq PAN(6,6) \leq 34$
$16 \leq PAN(7,6) \leq 34$	$\mathbf{23} \leq PAN(7,6) \leq 34$
$12 \leq PAN(8,6) \leq 19$	$\mathbf{17} \leq PAN(8,6) \leq 19$
$12 \leq PAN(9,6) \leq 14$	$PAN(9,6) = \mathbf{14}$
$14 \leq PAN(11,7) \leq 15$	$PAN(11,7) = \mathbf{15}$
$22 \leq PAN(10,8) \leq 34$	$\mathbf{25} \leq PAN(10,8) \leq 34$
$16 \leq PAN(11,8) \leq 25$	$\mathbf{21} \leq PAN(11,8) \leq 25$
$16 \leq PAN(13,8) \leq 17$	$PAN(13,8) = \mathbf{17}$
$27 \leq PAN(11,9) \leq 45$	$\mathbf{28} \leq PAN(11,9) \leq 45$
$18 \leq PAN(14,9) \leq 21$	$\mathbf{20} \leq PAN(14,9) \leq 21$

各最適化問題 $PAN(k,g)$ に対して，$PA(b;k,g)$ の b を (既知の下限から上限まで) 変化させた PA 判定問題群を生成し，それらを上記の 3 つのシステムを用いて求解した．符号化された PA 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，既知の上限値に対する PA 判定問題が SAT の場合は，その上限値が最適値となる．

順序符号化を行うプログラムには，制約ソルバー Sugar[*1] を使用し，バックエンドの SAT ソルバーとしては，MiniSat 2.2 (core) [*2] を使用した．CPU 時間は，Linux マシン (Intel Core i5 2.40GHz，3.7GB メモリ) 上で計測し，各 PA 判定問題に対する Sugar のタイムアウト ($T.O$) は 5400 秒とした．$M.O$ は Sugar がメモリオーバーのため符号化に失敗したことを示す．

最初に，符号化された PA 判定問題を解くのに MiniSat が要した CPU 時間を示す (表 2 参照，単位は秒)．サイズ b に関しては，得られた最良の下限 (あるいは上限) 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．なお，表中の PA 判定問題はすべて SAT であった．

提案システムは，22 問に対して既知の最適値を再構成することができ，さらに，3 問に対して既知の上限が最適値であることを決定した．例えば，これまで $PAN(9,6)$ の上下限は，$12 \leq PAN(9,6) \leq 14$ であった．しかし，表 2 から $PA(14;9,6)$ は SAT であるため，$PAN(9,6) = 14$ でることが決定できた．表 3 に，提案システムよって得られた新しい結果のサマリを示す．提案システムは，組合せデザイン・ハンドブック [5] にある PA 最適化問題 (全 46 問) に対して，最適値が未知であった 3 問について，既知の上限値が最適値であることを示すことに成功し，さらに，7 問について，新しい下限値を見つけることに成功した．

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

*2 http://minisat.se/Main.html

表 4: 最適値を求めた問題数

基本システム	拡張システム	最適化システム
21	23	**25**

最後に，表 4 に，最適値を求めることができた 25 問に対して，各システムが求解できた問題数を示す．その結果，最適化システムが 25 問と，最も多くの最適値を求めることに成功している．しかし，基本システム，拡張システムで解けるが，最適化システムでは解けない問題が 3 問あった．これに関する調査・考察については，今後の課題としたい．

6. まとめ

本稿では，PA 判定問題に対する 2 つの CSP 表現を提案した．特に，拡張行列モデルは，与えられた PA のパッキング制約を alldifferent 制約を用いて簡潔に表現することができる．また，拡張行列モデルに対する新しい SAT 符号化法を提案した．この符号化法は順序符号化法をベースに，符号化に必要な節数が少なくなるように最適化されている．組合せデザイン・ハンドブック [5] 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，提案システムは，10 問について最適値の決定，あるいは下限値の更新に成功した (表 3 参照)．

参考文献

[1] Abdel-Ghaffar, K. A. S.: On the number of mutually orthogonal partial latin squares, *Ars Comb.*, Vol. 42(1996), pp. 259–286.

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

[3] 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.

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

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

[6] 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.

[7] 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.

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

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

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

[11] 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.