数理解析研究所講究録
第1636巻 2009年 177-184

MIPソルバーを用いたBIBDの構成法

Construction of BIBDs using MIP solvers

防衛大学校情報工学科
横谷 大輔, 山田 武夫

Yokoya Daisuke, Yamada Takeo
{g47090, yamada}@nda.ac.jp
Department of Computer Science, The National Defense Academy
Yokosuka, Kanagawa 239-8686, Japan

1 はじめに

BIBD（balanced incomplete block design：**つり合い不完備ブロック計画** [1], [2]）は v 行 b 列の 0-1 行列で，(i) 各行の 1 の数は r 個，(ii) 各列の 1 の数は k 個，(iii) 異なる任意 2 行の交差数は λ，を満たすものをさす．ここに，2 つの行の**交差数**とは，それら 2 行のベクトルとしての内積を意味する．これをパラメータ v, b, r, k, λ を有する BIBD と呼び，BIBD(v, b, r, k, λ) と記す．BIBD(v, b, r, k, λ) を $X = (x_{ij})$ と表記すると，(i)〜(iii) より以下が成立する．

$$\sum_{l=1}^{b} x_{il} = r\,,\ \ i = 1, \cdots, v, \tag{1}$$

$$\sum_{i=1}^{v} x_{il} = k\,,\ \ l = 1, \cdots, b, \tag{2}$$

$$\sum_{l=1}^{b} x_{il}x_{jl} = \lambda\,,\ \ i, j = 1, \cdots, v,\ i < j, \tag{3}$$

$$x_{il} \in \{0, 1\}\,, i = 1, \cdots, v,\ \ l = 1, \cdots, b. \tag{4}$$

X が BIBD であるとき，その行，列を任意に並べ換えて得られる行列 Y も明らかに BIBD である．このような場合，X と Y は**同型**（isomorphic）であるという．パラメータ (v, b, r, k, λ) に対して，一般には同型でない複数の BIBD が存在するが，本稿では（1）〜（4）を満たす 1 つの行列 X を見い出すことのみを問題とし，同型でない BIBD をすべて列挙する問題等は対象としない．

BIBD は統計学において，**実験計画法** [3] の基本的なツールであり，薬品の効果の判定，農作物の品種改良，工業製品の品質改善等においてデータの取得・分析の効率化に大きく貢献している．また，同様の考え方を**符号理論**，**暗号理論**などに適用する試みや欠陥品の検査，宝くじ（ロト 6）の買い方に利用しようとする提案もなされている [4].

2 BIBDの存在条件と構成法

（1）〜（4）より，直ちに次の必要条件が導かれる．

定理 1 [5] BIBD(v, b, r, k, λ) において，次が成り立つ．

$$vr = bk, \tag{5}$$

$$r(k-1) = \lambda(v-1). \tag{6}$$

また，次は **Fisher の不等式**と呼ばれる．

定理 2 [5] BIBD(v, b, r, k, λ) においては

$$b \geq v,\ r \geq k \tag{7}$$

特に $b = v$, $r = k$ である BIBD を**対称 BIBD** という．

BIBD の存在のための一般性のある十分条件は知られていないが，特別なケースとして次が知られている．

定理 3 [5] パラメータ v, λ と $k = 3$ の BIBD$(v, b, r, 3, \lambda)$ が存在するための必要十分条件は

$$b = \frac{v(v-1)}{6}\lambda,\ r = \frac{\lambda(v-1)}{2} \tag{8}$$

が自然数であることである．

系 1 [5] パラメータ v と $k = 3$, $\lambda = 1$ の BIBD$(v, b, r, 3, 1)$ が存在するための必要十分条件は $b = v(v-1)/6$, $r = (v-1)/2$ で，

$$v \equiv 1 \text{ or } 3 \mod 6 \tag{9}$$

が成り立つことである．

定理 3 の BIBD，すなわち $k = 3$ の場合を **3 重システム**（triple system: TS）といい，系 1 の BIBD は**シュタイナー 3 重システム** (Steiner triple system) といって，STS(v) と記す．

（5）～（7）を満たすパラメータ (v, b, r, k, λ) に対し，BIBD(v, b, r, k, λ) を構成する一般的な方法は知られていないが，シュタイナー 3 重システムについては，19 世紀に Kirkman [6] が，さらに $k = 3, 4$ の場合には Hanani [7] が BIBD の存在のための必要十分条件と構成法を示している．これら以外の場合については**有限射影幾何** (finite projective geometry: FPG) や**有限アフィン幾何** (finite affine geometry: FAG), **差集合** (difference set: DS) 等を用いた代数学的手法で，個別に BIBD(v, b, r, k, λ) を構成する試みがなされてきた [3].

これに対して，最近ではコンピュータの計算パワーを利用して BIBD を探索するアプローチも試みられている．これについては，**制約プログラミング** (constraint programming: CP) [8], **ニューラルネットワーク** (ANN) [9], **シミュレーテッド・アニーリング** (SA) [10], **局所探索法** (LS) [11] などの方法がある．これらの方法のなかでは，今までのところ局所探索法が最も良い成績を上げているが，これらのアプローチによって（代数学的方法で見つけることが出来なかった）新しい BIBD が発見されたという例はない．

3 バックトラック法

（1）～（4）を満たす行列 $X = (x_{ij})$ を求める問題は一種の**制約充足問題**であるが，これを次の**非線形 0-1 計画問題** $\overline{P}$ に変換する．

$$\overline{P}: \quad \max \sum_{i=1}^{v}\Big(\sum_{l=1}^{b} x_{il}\Big)+\sum_{l=1}^{b}\Big(\sum_{i=1}^{v} x_{il}\Big)+\sum_{i=1}^{v-1}\sum_{j=i+1}^{v}\Big(\sum_{l=1}^{b} x_{il}x_{jl}\Big) \tag{10}$$

$$\text{s. t.} \quad \sum_{l=1}^{b} x_{il} \le r\,,\ i=1\cdots v, \tag{11}$$

$$\sum_{i=1}^{v} x_{il} \le k\,,\ l=1,\cdots,b, \tag{12}$$

$$\sum_{l=1}^{b} x_{il}x_{jl} \le \lambda\,,\ i,j=1,\cdots,v,\ i<j, \tag{13}$$

$$x_{il} \in \{0,1\}\,,\ i=1,\cdots,v,\ l=1,\cdots,b. \tag{14}$$

ここで，制約式が不等号に緩和されていることと，目的関数が各制約式の左辺の総和であることに注意する．このことから，次が言える．

定理 4

(i) $\overline{P}$ は常に実行可能である．

(ii) $\overline{P}$ の任意の実行可能解 $X=(x_{ij})$ に対し，目的関数値を $z(X)$ とすると，

$$z(X) \le vr+bk+\frac{v(v-1)\lambda}{2} \tag{15}$$

(iii) (15) 式が等号で成立する場合，(1) ～ (4) 式が満足される．すなわち，X は BIBD である．

従って，問題 $\overline{P}$ を解き (15) 式で等号が成立した場合は X が BIBD(v,b,r,k,λ) を与え，(15) 式が不等号の場合は BIBD が存在しないことが結論される．しかし，$\overline{P}$ は非線形の数理計画問題であるので，一般には厳密解を得ることが困難である．そこで，以下では X を上から順に 1 行ずつ決定していくアプローチを試みる．

今，X が j 行目まで求まっているとし，これを $\overline{X}_j=(\overline{x}_{il})$ $(i=1,2,\cdots,j)$ と記して第 j ステップにおける**部分解**と呼ぶ．(1) ～ (4) より，これは

$$\sum_{l=1}^{b} \overline{x}_{il} = r\,,\ i=1,2\cdots,j, \tag{16}$$

$$\sum_{i=1}^{j} \overline{x}_{il} \le k\,,\ l=1,2\cdots,b, \tag{17}$$

$$\sum_{l=1}^{b} \overline{x}_{hl}\overline{x}_{il} = \lambda\,,\ h,i=1,\cdots,j,\ h<i. \tag{18}$$

を満たす $j\times b$ の 0-1 行列でなければならない．$\overline{X}_j$ をもとに，第 $j+1$ ステップの部分解を得るには，

$$\sum_{l=1}^{b} x_l = r \tag{19}$$

$$x_l \le k-\sum_{i=1}^{j} \overline{x}_{il}\,,\ l=1,2,\cdots,b \tag{20}$$

$$\sum_{l=1}^{b} \overline{x}_{il}x_l = \lambda\,,\ i=1,2,\cdots,j \tag{21}$$

を満たすベクトル $x=(x_l)$ を求め，これを $\overline{X}_j$ の最下行に付加して

$$\overline{X}_{j+1}=\begin{pmatrix}\overline{X}_j\\ x\end{pmatrix} \tag{22}$$

とすればよい．このために，問題 $\overline{P}$ と同様に（19），(21）の左辺の総和を目的関数とする次の問題を考える．

$$P_j(X_j): \quad \max \sum_{l=1}^{b} x_l+\sum_{l=1}^{b}\Big(\sum_{i=1}^{j}\bar{x}_{il}\Big)x_l \tag{23}$$

$$\text{s. t.} \quad \sum_{l=1}^{b} x_l \le r, \tag{24}$$

$$x_l \le k-\sum_{i=1}^{j}\bar{x}_{il}\,,\ l=1,2,\cdots,b, \tag{25}$$

$$\sum_{l=1}^{b}\bar{x}_{il}x_l \le \lambda\,,\ i=1,2,\cdots,j, \tag{26}$$

$$x_l\in\{0,1\}\,. \tag{27}$$

$\overline{P}$ と異なり，この問題は**線形**の 0-1 計画問題なので，CPLEX [12] などの MIP ソルバーで厳密に解ける可能性が高い．このときの最適目的関数値を $z_j(X_j)$ とすると，定理 4 と同様に，

$$z_j(X_j)=r+j\lambda \tag{28}$$

の場合，そのときのみに（19）～（21）が満たされる．そこで，(28）が成立する場合には前述のように (22）式より $\overline{X}_{j+1}$ を得て第 $j+1$ ステップに移行する．

X の最初の 2 行は

$$X_2=\Big(\overbrace{11\cdots11\cdots10\cdots00\cdots0}^{r}\ \underbrace{11\cdots1}_{\lambda}0\cdots0\underbrace{1\cdots1}_{r-\lambda}0\cdots0\Big) \tag{29}$$

として一般性を失わないので，これから出発し，$X_2\to X_3\to\cdots\to X_v$ と v 段まで移行すれば BIBD が得られたことになる．しかし，一般には途中で（28）が成立しない場合が生じるので，このような場合も含め解を得るための方法を以下に述べる．

最初に注意すべき点は，問題 $P_j(X_j)$ の最適解は一般に複数個存在するので，たとえ（28）式が成立しても X の第 $j+1$ 行目にそれらのうちのどれを取り込むのが適当かは事前には分からないという事実である．そこで部分解 X_j に対し（28）式を満たす $P_j(X_j)$ の最適解を全列挙し，これらを第 $j+1$ 行目に代入して得られる部分解のそれぞれを X_j の**子部分解**と呼ぶ．これより，X_2 からスタートして部分解とその子部分解から成る**木**が得られ，BIBD が存在する場合には X_2 からその BIBD までの**パス**が必ず存在する．部分解 X_j で（28）式が成立しない場合，すなわち $z_j(X_j)<r+j\lambda$ のときは，X_j 以降で BIBD が得られる可能性はないので X_j を**終端**する．BIBD が存在しない場合には部分解が第 v ステップに到達することなく，すべて途中で終端されてしまう．

以上より，X_2 を根とする部分解の木を**深さ優先法**などによりくまなく走査すれば BIBD が得られるか，またはその非存在が証明される．このためには，標準的な**バックトラック法**を用いる．すなわち，部分解 X_j が終端された場合，1 ステップ上の X_{j-1} へ戻り，j ステップ目の部分解でまだ走査されていないものへ移って走査を続行する．j ステップ目の部分解がすべて終端され，未走査のものが残っていない場合には X_{j-1} も終端し，さらに 1 ステップ上に逆上る．

4 分枝限定法

前節のバックトラック法を実装するには，i) $P_j(X_j)$ の最適解の**全列挙法**，ii) 部分解の効率的な**枝払い法**，iii) 部分解の**探索戦略**等を考慮する必要がある．以下これらについて検討を加える．

まず，CPLEX [12] 等を用いて $P_j(X_j)$ を解き，最適解 x^* と最適値 $z_j(X_j)$ を得たとする．このような最適解は x^* 以外にも存在しうるので，それらを全列挙するために，以下では $P_j(X_j)$ にさらに制約条件を付加した次の子問題を導入する．

$$
\begin{aligned}
P_j(X_j, F, R): \quad & \max \ (23) \\
& \text{s. t.} \ (24)\sim(27), \\
& \quad x_i = 1, \quad i \in F \qquad (30) \\
& \quad x_i = 0, \quad i \in R. \qquad (31)
\end{aligned}
$$

ここに，F，R はそれぞれ 1 または 0 に固定する変数の添え字の集合で，これをうまく設定することによって $P_j(X_j)$ の最適解を全列挙するアルゴリズムを再帰的に構成することができる．これを BIBD を見出すための分枝限定法の枠組みに組み込んだものが次ページの BAB_BIBD である．

アルゴリズム中で，$P_j(X_j, F, R)$ は 0-1 計画問題（IP 問題）なのでこれを厳密に解くことは必ずしも容易ではない．この部分を連続緩和し，LP 問題を解いた場合にも（28）式が不成立であればこの子問題を終端することが出来るが，アルゴリズム中では Step 2 (i) がこれに相当する．もちろん，LP 問題の解が 0-1 解でない場合には Step 2 (ii) のように改めて IP 問題を解く必要がある．以下では Step 2 (i)，(ii) をこの順に実行する方法を BAB_BIBD(LP)，(i) を省略して (ii) のみを実行する方法を BAB_BIBD(IP) と記す．

最後に，分枝木の走査法であるが，Step 4 において添字集合 U を昇順に整列した場合と逆順の場合では（Step 5 で再帰呼び出しされる部分解が異なるため）生成される分枝木とその走査順序が異なってくる．本稿で前者を BAB_BIBD(FORWARD)，後者を BAB_BIBD(BACKWARD) と記し，IP/LP と併せて BAB_BIBD(FORWARD, LP) など 4 種類の戦略を検討する．

5 計算例

前節までのアルゴリズムを ANSI C 言語で実装し，DELL Precision 670 (CPU : Xeon 3.8GHz, メモリー : 512MB) 上で MIP ソルバー ILOG CPLEX 10.100 [12] を用いて実験を行った．表 1 は他の文献（[10], [11] など）でとりあげられている 63 例について，BAB_BIBD(FORWARD, LP) の計算結果を示したもので，BAB の欄が今回の方法による計算時間を秒単位で表しており，Time over とあるのは 3600 秒以内に解けなかったことを示している．他の欄は制約プログラミング（CP, [8]），ニューラル・ネットワーク（ANN, [9]），シミュレーテッド・アニーリング（SA, [10]），局所探索法（LS, [11]）による結果を示したもので，このうち LS は DEC Alphaserver 1000A 5/300 (300MHz) 上での CPU 時間（秒）を示しているが，ANN，SA については計算時間は不明である．これらの欄の空欄は計算が行われていないことを，ハイフン (–) は解が得られなかったことを示している．

一部の問題について，本稿の方法は非常に長い計算時間を要することがあるが，全般的には CP，ANN，SA よりもはるかに優れ，LS とは（使用コンピュータの能力を勘案しても）同等以上であるように思われる．

アルゴリズム　**BAB_BIBD**
input: j, X, F, R

Step 1. $j = v$ ならば X が BIBD なのでこれを出力して終了.
Step 2. (i) LP 問題 $\overline{P}_j(X, F, R)$ を解く.
(a) $\bar{z}_j(X, F, R) < r + \lambda$ なら return.
(b) 解 $x_j(X, F, R)$ が整数解なら step 3 へ，そうでなければ (ii) へ進む.
(ii) IP 問題 $P_j(X, F, R)$ を解く.
(a) $z(X, F, R) < r + \lambda$ なら return.
(b) そうでなければ step 3 へ.
Step 3. (22) 式により，X に $x^*(X, F, R)$ を加え $j+1$ 行の行列に更新する．ついで BAB_BIBD$(j+1, X, \emptyset, \emptyset)$ を再帰呼び出しする.
Step 4. 整数解 $x^*(X, F, R)$ の添字集合 $U := \{ l \mid x^*_l = 1, l \notin F\}$ を求め，昇順（または，降順）に整列して $U = \{u_1, u_2, \cdots, u_p\}$ とする.
Step 5. $l = 1, 2, \cdots, p$ について以下を行う.
- $F' := F \cup \{u_1, u_2, \cdots, u_{l-1}\}$, $R' := R \cup \{u_l\}$ とする.
- BAB_BIBD(j, X, F', R') を再帰呼び出しする.

Step 6. (j, X) 以降の BIBD は存在しないので終了.

6 むすび

本報告では，BIBD を求める新しいアプローチとして MIP ソルバーを利用した分枝限定法アルゴリズムを提示した．　数値実験では少数個の例外を除き $v \leq 20$ 程度の BIBD を短時間に求めることが出来たが，より大規模な問題を解いたり，未知の BIBD を発見するにはさらに効率的な枝払い法や部分解に含まれる不適切な 0-1 ベクトルを上手に見分けて部分解から排除する方法を研究してアルゴリズムを改善して行く必要がある.

参考文献

[1] C. J. Colbourn, J. H. Dinitz: *Handbook of Combinatorial Designs 2nd ed.*, Chapman & Hall/CRC, New York, 2007.

[2] D. R. Stinson: *Combinatorial Designs Construction and Analysis*, Springer-Verlag, New York, 2004.

[3] 石井吾郎: 実験計画法／配置の数理, 培風館, 東京, 1972.

[4] T. Beth et al.: *Design theory, volume II*, Cambridge University Press, 1999.

[5] M. Hall, Jr.: *Combinatorial Theory second Edition*, JOHN WILEY & SONS, INC, 1998.

[6] T. Kirkman: "On a problem in combinations," *Cambridge and Dublin Math. J*, **2**: 191-204, 1847.

表 1: 分枝限定法　計算結果と他の方法との比較.

v	b	r	k	λ	CP	ANN	SA	LS	BAB
6	10	5	3	2	15.10		○	0.00	0.00
6	30	15	3	6			○	0.00	0.00
6	40	20	3	8			○	0.21	0.00
7	7	3	3	1	6.63	○	○	0.00	0.00
7	14	6	3	2	264.00		○	0.00	0.00
7	21	9	3	3			○	0.00	0.00
7	28	12	3	4			○	0.01	0.00
7	42	18	3	6			○		0.00
7	70	30	3	10				0.08	0.01
7	77	33	3	11				0.10	0.01
7	84	36	3	12				0.11	0.01
7	91	39	3	13				0.16	0.01
8	14	7	4	3	—		○	0.00	0.00
9	12	4	3	1	140.00	○	○	0.00	0.00
9	18	8	4	3			○	0.05	0.01
9	24	8	3	2			○	0.02	0.00
9	60	20	3	5				0.13	0.01
9	72	24	3	6				0.20	0.01
9	84	28	3	7				0.22	0.01
9	96	32	3	8				0.32	0.01
9	108	36	3	9				0.50	0.02
9	120	40	3	10				0.65	0.02
10	30	9	3	2			○	0.05	0.01
10	15	6	4	2			○	0.04	2073.94
10	18	9	5	4			○	0.12	0.02
10	30	12	4	4			○	0.12	0.05
10	60	18	3	4				0.20	0.01
10	90	27	3	6				0.50	0.01
10	120	36	3	8				0.89	0.02
11	22	10	5	4			○	1.20	0.01

[7] H. Hanani: "Balanced incomplete block designs and related designs," *Discrete Mathematics*, **11**: 255-369, 1975.

[8] S. Prestwich: "Balanced incomplete block design as satisfiability," *Irish Conf. on AI and Cognitive Science*, **12**: 189-198, 2001.

[9] T. Kurokawa et al.: "Neural network parallel computing for BIBD problems," *IEEE Trans on Circuits and Systems, II*, **39**: 243-247, 1992.

[10] P. Bofill et al.: "Comparison of simulated annealing and mean field annealing as applied to the generation of block designs," *Neural Networks*, **16**: 1421-1428, 2003.

[11] S. Prestwich: "A local search algorithm for balanced incomplete block designs," *Lecture Notes in AI*, 2002.

[12] CPLEX Ver. 10.0 ILOG, 2006. (http://www.ilog.com)

表 1: (つづき)

v	b	r	k	λ	CP	ANN	SA	LS	BAB
11	55	15	3	3				0.23	0.01
11	55	20	4	6				0.80	0.03
11	11	5	5	2			○	0.04	0.02
11	110	30	3	6				0.91	0.02
12	44	11	3	2				0.25	0.01
12	33	11	4	3				0.33	Time over
12	22	11	6	5			○	1.23	0.03
12	88	22	3	4				0.94	0.03
13	26	6	3	1		○		0.15	0.01
13	13	4	4	1		○	○	0.01	0.00
13	39	15	5	5				11.20	0.05
13	26	12	6	5				7.93	17.60
13	52	12	3	2				0.31	0.02
13	78	18	3	3				1.09	0.02
13	104	24	3	4				1.81	0.04
14	26	13	7	6				—	0.38
15	35	7	3	1		○		0.54	0.02
15	42	14	5	4				42.90	0.14
15	35	14	6	5				—	3.29
15	15	7	7	3			○	0.61	0.01
15	70	14	3	2				1.50	0.02
16	80	15	3	2				2.81	0.04
16	20	5	4	1				0.16	0.02
16	48	15	5	4				43.20	0.11
16	16	6	6	2			○	0.54	Time over
16	24	9	6	3				9.80	0.03
16	40	15	6	5				—	2.75
18	51	17	6	5				—	0.79
19	57	9	3	1				3.91	0.04
19	19	9	9	4				9.73	0.21
21	21	5	5	1				0.22	0.31
25	30	6	5	1				14.30	0.05
31	31	6	6	1				2.04	0.06