A backtracking algorithm for finding t-designs*

M. M-Noori[a,b] B. Tayfeh-Rezaie[a,1]

[a] *Institute for Studies in Theoretical Physics and Mathematics (IPM), P.O. Box* 19395-5746*, Tehran, Iran*

[b] *Department of Mathematics, University of Tehran, Tehran, Iran*

Abstract

A detailed description of an improved version of backtracking algorithms for finding t-designs proposed by G. B. Khosrovshahi and the authors of this paper [J. Combin. Des. 10 (2002), 180-194] is presented. The algorithm is then used to determine all 5-(14,6,3) designs admitting an automorphism of order 13, 11 or 7. It is concluded that a 5-(14,6,3) design with an automorphism of prime order p exists if and only if $p = 2, 3, 7, 13$.

1. Introduction

One of the central topics in design theory is the classification of different kind of designs. Most of these classifications are usually carried out using computational methods. These methods have been improved over the years

*This research was partially supported by a grant from IPM.

[1] Corresponding author, Email: tayfeh-r@ipm.ir.

by different authors. Moreover, the increasing speed of computers have made it now possible to solve more difficult problems of classifications of designs.

There are some known results on the classifications of t-designs for $t \geq 3$, see for example [2,4,6,7,8,9]. The problem of search for a t-design can be formulated as a solution to a linear Diophantine system of equations. Backtracking algorithms are usually used to solve the system. For a brief description of backtracking algorithms and their applications the reader is referred to [3]. In [4], an improved version of the classical backtracking algorithm is presented for finding and classifying t-designs. The algorithm uses a systematic method to obtain new redundant equations from the initial equations which are helpful in speeding up the classical backtracking algorithm. It also takes advantage of the so-called *preset* technique to prune the search space. The improved algorithm was used in [4] to classify all 6-(14,7,4) designs with a nontrivial automorphism group. In this paper, we first present this algorithm in details and then use it to determine all 5-(14,6,3) designs admitting automorphisms of order 13, 11 and 7.

Let t, k, v, and λ be integers such that $0 \leq t \leq k \leq v$ and $\lambda > 0$. Let V be a v-set and $P_k(V)$ the set of all k-subsets (called *blocks*) of V. A t-(v, k, λ) *design* on V is a collection $\mathcal{D}$ of the blocks of V such that every t-subset of V occurs exactly λ times in $\mathcal{D}$. If no blocks are identical, then $\mathcal{D}$ is called *simple*. If $\mathcal{D}$ is simple, then $\overline{\mathcal{D}} = P_k(V) \setminus \mathcal{D}$ is obviously a simple t-$(v, k, \binom{v-t}{k-t} - \lambda)$ design which is called the *supplement* of $\mathcal{D}$. Here, we are concerned only with simple designs. Let $W \subseteq V$ such that $|W| = i \leq t$. Then $\mathcal{D}_W = \{B \setminus W : B \in \mathcal{D}, W \subseteq B\}$ is a $(t-i)$-$(v-i, k-i, \lambda)$ design which is called the *derived design* of $\mathcal{D}$ with respect to W. Two t-(v, k, λ) designs $\mathcal{D}_1$ and $\mathcal{D}_2$ are called *isomorphic* if there is a permutation σ on V such that $\sigma\mathcal{D}_1 = \mathcal{D}_2$ (note that σ induces a permutation $\bar{\sigma}$ on the blocks, for simplicity, we write $\sigma\mathcal{D}$ instead of $\bar{\sigma}\mathcal{D}$). An *automorphism* of $\mathcal{D}$ is a permutation σ such that $\sigma\mathcal{D} = \mathcal{D}$. The group generated by some of the automorphisms of $\mathcal{D}$ is called an *automorphism group* of $\mathcal{D}$ and the group of all its automorphisms is called the *full automorphism group* of $\mathcal{D}$, denoted by $Aut\mathcal{D}$.

Let $W_{tk}(v)$ be a $\binom{v}{t} \times \binom{v}{k}$ $(0,1)$-matrix whose rows and columns are indexed by the t-subsets and k-subsets of V, respectively, and for a t-subset T and a k-subset K, $W_{tk}(v)(T,K) = 1$ if and only if $T \subseteq K$. $W_{tk}(v)$ is called the *incidence* matrix of t-subsets vs. k-subsets of V. We simply write W_{tk} instead of $W_{tk}(v)$ if there is no risk of confusion. If D is the (0,1)-column vector representation of $\mathcal{D}$ (with the blocks ordered in the same order of the indices of columns of W_{tk} and $D(B) = 1$ if and only if $B \in \mathcal{D}$), then we have

$$W_{tk}D = \lambda J, \tag{1}$$

where J is the all-one column vector. The equation (1) is used to find t-designs computationally.

2. The Algorithm

In this section, we expose and formulate the algorithm presented in [4] in details. Indeed this improved version of classical backtracking algorithm enables us to find new t-designs. There are two main new ideas in this improved algorithm: the first being a utilization of the so-called preset technique and the second being the addition of some redundant equations to system (1) above. This modification prunes the search space and thus substantially increases the speed of the classical backtracking algorithm. We illustrate the ideas involved by a simple example.

Example 1. Suppose that we want to find all 1-(4,2,2) designs by a backtracking algorithm. By (1), we need to solve the system of equations

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}. \tag{2}$$

First we set $x_1 = 1$. At this stage, we know nothing about $x_i, i = 2, \ldots, 6$. In the second step, set $x_2 = 1$. From the first equation of (2), it follows that $x_3 = 0$. In this case we say that x_3 is *preset* to 0 and in the next step x_4 is dealt with. The idea of presetting a variable is crucial for improving the backtracking algorithm. On the other hand, by adding new suitable equations to (2) which are not linearly independent from the equations in (2) , one can make use of the presetting technique more efficiently. For example, in (2) by subtracting the last two equations from the sum of the first two equations, we have $x_1 = x_6$. Hence, in the first step of algorithm, x_6 can be preset to 1. By adding further similar equations to (2), we obtain the following equivalent system of equations:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Later on this section we will explain a systematic method to extract proper equations from the incidence matrices.

Now, we need some more notation to formulate the *preset* concept.

Notation. All matrices are assumed to be integer valued. The ith entry of a column vector P is denoted by P_i. Let $A = [A_{ij}]$ and $B = [B_{ij}]$ be two $h \times k$ and $k \times l$ matrices, respectively. By $|A|$, we refer to the matrix whose (i, j) entry is $|A_{ij}|$. The *positive product* $A \odot_p B$ is an $h \times l$ matrix defined by

$$(A \odot_p B)_{ij} = \sum_{e=1}^{k} \max(a_{ie}b_{ej}, 0).$$

The *negative product* is defined by $A \odot_n B = A \odot_p (-B)$. It is easy to see that

$$A \odot_p B = \frac{1}{2}(|A||B| + AB), \tag{3}$$

$$A \odot_n B = \frac{1}{2}(|A||B| - AB). \tag{4}$$

Suppose that we want to find all (0,1)-solutions X of the equation

$$AX = C, \tag{5}$$

where X and C are column vectors. To do this, equivalently, we can assume that X is a (-1,1)-vector (if X is a (0,1)-solution of (5), then $X' = 2X - J$ is a (-1,1)-solution of $AX' = 2C - AJ$). By (3) and (4), we have

$$A \odot_p X = P, \tag{6}$$

$$A \odot_n X = N, \tag{7}$$

where $P = 1/2(|A|J + C)$ and $N = 1/2(|A|J - C)$. Initially, we let $X = 0$. Now, suppose that in a step of the algorithm we have a partial (0,±1)-solution for X. By (6) and (7), if for some i, we have $(A \odot_p X)_i > P_i$, or $(A \odot_n X)_i > N_i$, then the algorithm should backtrack. On the other hand, if for any i, we have $(A \odot_p X)_i = P_i$, then for each j for which $A_{ij} > 0$ and $x_j = 0$, x_j should be preset to -1 and for each j for which $A_{ij} < 0$ and $x_j = 0$, x_j should be preset to 1. A similar argument is used when $(A \odot_n X)_i = N_i$ for any i. When some x_i is preset to $+1$ or -1, it can possibly enforce presetting some other x_j. We continue the process of presetting until it can not continue anymore, then we go to the next step of the algorithm.

In order to find new equations from the initial ones and add them to the system, we define some special intersection matrices for t-designs (note that these matrices were derived in [4] by a different method). Let $\mathcal{D}$ be a t-(v, k, λ) design. For nonnegative integer s, let

$$M_{tk}^{s}(v) = \sum_{i=0}^{t} (-1)^{t-i} W_{is}^{T} W_{ik}. \tag{8}$$

The rows and columns of $M_{tk}^{s}(v)$ are indexed by the s-subsets and k-subsets of V in the same orders of indices of rows and columns of $W_{is}^{T} W_{ik}$, respectively. Usually we write M_{tk}^{s} instead of $M_{tk}^{s}(v)$ if there is no risk of confusion. It is easy to see that

$$M_{tk}^{s} D = b_{tk}^{s} J, \tag{9}$$

where $b_{tk}^{s} = \sum_{i=0}^{t} (-1)^{t-i} \binom{s}{i} \lambda_i$. Using the well known identity

$$\binom{l-1}{t} = \sum_{i=0}^{t} (-1)^{t-i} \binom{l}{i},$$

we have

$$M_{tk}^{s}(S, K) = \binom{|S \cap K| - 1}{t},$$

for any s-subset S and k-subset K of V. It can be seen from (8) that the equations in (9) are not linearly independent from the equations in (1). However, appending some of them to (1), sometimes improves the speed of the algorithm. The useful equations from (9) to be added to (1) very highly depend on the parameters of the underlying design and are chosen by hand.

Example 2. Let $\mathcal{D}$ be a 1-(4,2,2) design. Then

$$M_{1,2}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and $M_{1,2}^2 D = 0$. These are exactly the appended equations discussed in Example 1.

Example 3. Let $\mathcal{D}$ be a 5-(14,6,3) design. Consider the following matrices:

$$M_{5,6}^6(S,K) = \begin{cases} 1 & \text{if } S = K \ , \\ -1 & \text{if } S \cap K = \varnothing, \\ 0 & \text{otherwise,} \end{cases}$$

$$M_{5,6}^7(S,K) = \begin{cases} 1 & \text{if } K \subset S \ , \\ -1 & \text{if } S \cap K = \varnothing, \\ 0 & \text{otherwise,} \end{cases}$$

which yield the useful equations $M_{5,6}^6 D = -9J$ and $M_{5,6}^7 D = 0$.

In [5], Kramer and Mesner suggested an approach, based on W_{tk}, to find t-designs with a prescribed automorphism group. In their approach the size of W_{tk} is reduced by applying a suitable group on the indices of rows and columns of W_{tk}. Suppose that $\{\Delta_i\}$ and $\{\Gamma_j\}$ are the orbits of t-subsets and k-subsets of V under the action of the group G, respectively. Let $W_{tk}[G]$ be a matrix whose rows and columns are indexed by $\{\Delta_i\}$ and $\{\Gamma_j\}$, respectively and define $W_{tk}[G](\Delta_i, \Gamma_j) = |\{K \in \Gamma_j : T \subseteq K\}|$, for some fixed $T \in \Delta_i$. Then it is easily seen that $W_{tk}[G]$ is a well defined matrix and every $(0,1)$-solution D of the equation

$$W_{tk}[G]D = \lambda J,$$

is a t-(v,k,λ) design with G as its automorphism group [5]. $M^s_{tk}[G]$ is defined similarly. Let $\{\Lambda_i\}$ be the orbits of s-subsets of V under the action of G. Let $M^s_{tk}[G]$ be a matrix whose rows and columns are indexed by $\{\Lambda_i\}$ and $\{\Gamma_j\}$, respectively and define

$$M^s_{tk}[G](\Lambda_i,\Gamma_j) = \sum_{K\in\Gamma_j} \binom{|S\cap K|-1}{t}, \quad \text{for some fixed } S \in \Lambda_i.$$

Clearly $M^s_{tk}[G]$ is a well defined matrix and by (9), we have

$$M^s_{tk}[G]D = b^s_{tk}J.$$

An important technique to limit search space in backtracking algorithms is the *isomorphic rejection* technique. The idea is simply to remove the isomorphic copies of a partial solution X from the set of partial solutions in any step of the algorithm because they all lead to isomorphic solutions. This technique is usually used in the initial steps of the algorithm where many copies of each partial solution are found.

3. Isomorphism test

As mentioned in Section 2, we use the isomorphic rejection technique only in the early steps of the backtracking algorithm. However, there may be some isomorphic copies among the final solutions too and of course we have to remove those. Suppose that we have found every possible design with a prescribed automorphism group G. The isomorphism test is carried out in two phases. In the first phase, we apply the normalizer of G in S_v to the set of solutions. This rapidly rejects many of the isomorphic copies. In the second phase, we deal with the remaining designs regardless of G. There are many algorithms to test isomorphism between designs, see [3] for details. Here we use a simple method to check isomorphism between two given t-(v,k,λ) designs $\mathcal{D}_1$ and $\mathcal{D}_2$. Let $\overline{\mathcal{D}}_1$ and $\overline{\mathcal{D}}_2$ be the supplements of $\mathcal{D}_1$ and

$\mathcal{D}_2$, respectively. Let $V_i = \{1, \ldots, i\}$ for $k \leq i \leq v$. Let Σ denote the set of all one-to-one functions from V_i to V_v for all $k \leq i \leq v$. If $\sigma \in \Sigma$ and σ is defined on V_i, we write σ_i instead of σ to show its domain. Σ can be ordered lexicographically. Then, Σ is searched to find a possible permutation σ_v such that $\sigma_v \mathcal{D}_1 = \mathcal{D}_2$. In order to test a σ_i, we proceed as follows. If there is a k-subset B of V_i containing i such that $B \in \mathcal{D}_1, \sigma_i(B) \notin \mathcal{D}_2$ or $B \in \overline{\mathcal{D}}_1, \sigma_i(B) \notin \overline{\mathcal{D}}_2$, then we skip to the next element σ_j of Σ with $j \leq i$. If there is no such B, then either $i = v$ and $\sigma_v \mathcal{D}_1 = \mathcal{D}_2$, or $i < v$ and we keep testing the next element of Σ.

4. 5-(14,6,3) designs

In the literature there is only one 5-(14,6,3) design found in [1]. The full automorphism group of this design is of order 39. In this section, we use our algorithm to find new designs with these parameters admitting prescribed automorphism groups. We focus only on the cyclic groups of prime orders. An automorphism of prime order p which contains exactly r cycles of length p is said to be of type p^r. First we need the following result.

Lemma. *There are no* 5-(14, 6, 3) *designs with an automorphism of type* $7^1, 5^1$ *or* 5^2.

Proof. Let $\mathcal{D}$ be a 5-(14,6,3) design on the point set $V = \{1, \ldots, 14\}$ with an automorphism σ. First suppose that $\sigma = (1\ 2\ \cdots\ 7)$. Then the blocks $\{i, 8, 9, 10, 11, 12\}$ $(1 \leq i \leq 7)$ must be all in $\mathcal{D}$ or in its supplement, which is impossible. Now let $\sigma = (1\ \cdots\ 5)$. Assume that $\mathcal{D}'$ is the derived design of $\mathcal{D}$ with respect to the point set $\{11, 12, 13, 14\}$. Then there are no blocks $\{i, j\}$ in $\mathcal{D}'$ such that $1 \leq i \leq 5$ and $6 \leq j \leq 10$. Therefore, $\mathcal{D}'$ must contain a 1-(5,2,3) design. However, there is no such design. Finally, let $\sigma = (1\ \cdots\ 5)(6\ \cdots\ 10)$. Let $\mathcal{D}'$ be the derived design of $\mathcal{D}$ with respect to 14. The block orbits of $\mathcal{D}'$ are of lengths 1 or 5. $\mathcal{D}'$ has 429 blocks so it should contain at least 4 orbits of length 1. But there are exactly 2 orbits of length 1 on the 5-subsets of V. This completes the proof. □

To find all 5-(14,6,3) designs $\mathcal{D}$ with a prescribed automorphism group G, we consider the equations

$$W_{5,6}[G]D = 3J, \quad (10)$$

$$M^6_{5,6}[G]D = -9J, \quad (11)$$

$$M^7_{5,6}[G]D = 0. \quad (12)$$

These equations are solved using our modified backtracking algorithm described in Section 2. The computations have been carried out on a PC with a 533 MHz Pentium II CPU and 128 MB RAM running a C program. We have been able to solve the system above for the cyclic groups of orders $p = 13, 11, 7$. For the case $p = 13$, $W_{5,6}[C_{13}], M^6_{5,6}[C_{13}]$ and $M^7_{5,6}[C_{13}]$ have 154, 231 and 264 rows, respectively and 231 columns. The results are as follows: if we use only the equation (10) in our algorithm, then the computational time is about 30 minutes. By incorporating the equations (11) and (12) to the system, the computation is reduced to 15 minutes. We can do even better. Assume that the point 1 is fixed by the automorphism of order 13. Therefore, we can add the equation

$$M^6_{4,5}(13)[C_{13}]D' = 9J,$$

to our system, where $\mathcal{D}'$ is the derived design of $\mathcal{D}$ with respect to 1. The computational time in this situation is around 30 seconds. This is an instance which demonstrates that by adding an appropriately chosen set of equations to the system the speed of backtracking algorithms improves substantially. The case $p = 11$ takes about a minute and the appended equations do not show a remarkable improvement. For the case $p = 7$, we used the equations (10), (11) and (12) in our algorithm and it took a few days to find all solutions. The isomorphism test was done to remove the isomorphic copies among the solutions using the method described in Section 3 which took only a few minutes. The results are summarized in the following theorem.

Theorem. *Let $\mathcal{D}$ be a* 5-(14, 6, 3) *design.*

(i) There is a unique design $\mathcal{D}$ with an automorphism of order 13*. The full automorphism group is of order* 39*.*

(ii) There are no designs $\mathcal{D}$ with an automorphism of order 11*.*

(iii) There exist exactly 4 *designs $\mathcal{D}$ with an automorphism of order* 7*. Three of these having full automorphism group of order* 42 *and one of order* 14*.*

Corollary. *There are* 5-(14, 6, 3) *designs with an automorphism of prime order p if and only if $p = 2, 3, 7, 13$.*

It is not known that if it is possible to partition the 6-subsets of a 14-set into three 5-(14,6,3) designs. According to our computations, these designs do not permit such partitioning.

The 5-(14,6,3) designs with an automorphism of order 7 are new and are presented in the Appendix.

Acknowledgments. The authors thank the referees for their helpful suggestions which have improved the presentation of the paper.

References

1. A. E. Brouwer, *Table of t-designs without repeated blocks,* $2 \leq t \leq k \leq v/2, \lambda \leq \lambda^+/2$, Math. Centrum. Report ZN76, Amsterdam, 1977; unpublished update, 1986.

2. Y. M. Chee, C. J. Colbourn, and D. L. Kreher, *Simple t-designs with $v \leq 30$*, Ars Combin. **29** (1990), 193–258.

3. P. B. Gibbons, *Computational methods in design theory*, in: The CRC handbook of combinatorial designs (C. J. Colbourn and J. H. Dinitz, eds.), CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton (1996), pp. 718–740.

4. G. B. Khosrovshahi, M. Mohammad-Noori, and B. Tayfeh-Rezaie, *Classification of* 6-$(14, 7, 4)$ *designs with nontrivial automorphism groups*, J. Combin. Des. **10** (2002), 180-194.

5. E. S. Kramer and D. M. Mesner, *t-Designs on hypergraphs*, Discrete Math. **15** (1976), 263–296.

6. D. L. Kreher and S. P. Radziszowski, *The existence of simple* 6-$(14, 7, 4)$ *designs*, J. Combin. Theory Ser. A **43** (1986), 237–243.

7. S. P.Radziszowski, *Enumeration of all simple t-$(t+7, t+1, 2)$ designs*, J. Combin. Math. Combin. Comput. **12** (1992), 175–178.

8. B. Schmalz, *The t-designs with prescribed automorphism group, new simple* 6-*designs*, J. Combin. Des. **1** (1993), 125–170.

Appendix

The orbit representations of new 5-(14,6,3) designs admitting an automorphism of order 7 are given below. The point set is $V = \{1, \ldots, 9, A, \ldots, E\}$. Design $\#i$ $(1 \leq i \leq 4)$ has G_i as the full automorphism group where G_i are as follows:

$$G_1 =< \sigma_1, \sigma_2 >,\ G_2 =< \sigma_1, \sigma_2, \sigma_3 >,\ G_3 =< \sigma_1, \sigma_4, \sigma_5 >,\ G_4 = G_2,$$
$$\sigma_1 = (1\ \cdots\ 7)(8\ \cdots\ E),$$
$$\sigma_2 = (1\ 8)(2\ E)(3\ D)(4\ C)(5\ B)(6\ A)(7\ 9),$$
$$\sigma_3 = (2\ 3\ 5)(4\ 7\ 6)(8\ E\ C)(A\ B\ D),$$
$$\sigma_4 = (2\ 3\ 5)(4\ 7\ 6)(8\ D\ 9)(B\ C\ E),$$
$$\sigma_5 = (1\ A)(2\ B)(3\ C)(4\ D)(5\ E)(6\ 8)(7\ 9).$$

Note that $|G_1| = 14$ and $|G_2| = |G_3| = |G_4| = 42$.

Design #1.

123458	123459	12345B	123468	123469	12346C	12349A
1234AD	1234BC	1234BD	1234CE	1234DE	12356A	12356B
12356D	12358C	12359C	12359E	1235AD	1235AE	1235CD
123689	12368D	1236AC	1236AE	1236BC	1236BE	12389D
1238AB	1238AC	1238AE	1238BC	1238BE	1238DE	1239AB
1239AD	1239BD	1239BE	1239CE	123CDE	12458A	12458E
1245AC	1245BD	1245BE	1245CD	12468A	12468C	12469B
1246AB	1246BE	1246CD	12489B	12489D	1248AE	1248BC
1248CD	1249AC	1249CE	1249DE	124ABD	124ACE	124ADE
12589B	12589C	1258BD	1258CD	1259AD	1259DE	125ABE
125ACE	1269BD	1269CE	126ABD	1358AC	1359BD	135ACE

Design #2.

123458	123459	12345A	12349D	1234AD	1234BC	1234BD
1234BE	1234CE	12358B	12358D	12368A	12369B	12389A
12389B	12389E	1238AD	1238BE	1238CD	1238CE	1239AB
1239AE	1239DE	123BCD	12489B	1249AC	1249AE	

Design #3.

123458	123459	12345C	12349D	1234AB	1234AC	1234AE
1234BD	1234CD	12358A	1235BC	12368A	12369C	12389B
12389C	12389E	1238AD	1238BD	1238CE	1239AE	123ABC
123BDE	12489B	12489D	1248CE			

Design #4.

123458	12345A	12345C	12348D	12349C	1234AB	1234AE
1234BD	1234DE	12358A	12359A	123689	12369B	12389C
12389E	1238AC	1238AE	1238BC	1238DE	1239AD	123ABC
123BCD	12489B	12489D	1249AC	1249AE	1249DE	1268AE