Algorithmic Aspects of Trades[1]

Ziba Eslami*
Department of Computer Sciences,
Shahid Beheshti University, G.C.
Tehran, IRAN

1. Introduction

The notion of a *trade* in design theory was first formally introduced in 1977 by Hedayat [11]. However, trades have been used in the literature quite earlier under different disguise. Back in 1916, "quadrangular and hexagonal transformations", appeared in constructing Steiner triple systems of order 15 [28]. In the 1970s, names such as "t-pods" [14], "null designs" [13], "functions with strength t" [3] and "fragments" [12] carried the concept of a combinatorial trade. The original use of these objects as defined by Hedayat was to avoid some undesirable blocks in an experimental design while maintaining the same parameter-variety set. But the idea behind them was later extended to solve several diverse problems concerning graphs, t-designs, signed designs, latin squares, and large sets.

There are some review papers on these objects available [1, 21, 15, 27]. In [1, 21] the structure and properties of these objects as well as their interaction with other combinatorial structures are surveyed. More recently, some algorithms which exploit trades and their rich algebraic properties were developed and utilized to solve several classification and existence problems in design theory. Since this algorithmic approach is not touched in available reviews, this paper intends to provide a brief update on this algorithmic approach and its applications. The paper is organized as follows: in Section 2, introductory concepts are presented; Section 3 describes an algorithm for the classification of trades while in Section 4 applications of this algorithm for the case of t-designs and large sets are explained. Section 5 is about a local search algorithm for constructing signed designs.

2. Definitions and Preliminaries

Let v, k, t and λ be positive integers. Let X be a set of cardinality v and let $P_i(X)$ denote the set of all i-subsets of X. The elements of X and $P_k(X)$ are called *points* and *blocks*, respectively. We define a t-(v,k,λ) *design*, or briefly a *t-design* to be a pair $(X,\mathscr{B})$, where $\mathscr{B}$ is a collection of blocks of X with the property that every element of $P_t(X)$ lies in exactly λ blocks of $\mathscr{B}$. We also note that if $\lambda > 0$, then we must have $0 \le t \le k \le v$, for the definition to make sense. An *isomorphism* between $(X,\mathscr{B})$ and $(X',\mathscr{B}')$ is a one-one mapping from X to X' such that the blocks of $\mathscr{B}$ are mapped onto the blocks of $\mathscr{B}'$. If no such mapping exists, then the designs are said to be *non-isomorphic*. The set of *automorphisms* of a design

[1]This research was supported by a grant from Shahid Beheshti University.

*Corresponding author. Tel.: +982129903005; fax: +982122431655 Tehran, Iran; Email: z_eslami@sbu.ac.ir

(that is, isomorphisms from a design to itself) forms a group which acts in a natural way as a permutation group on the points of the design and consequently on its blocks. A design is called *rigid* if its automorphism group is the trivial group.

A t-(v,k) *trade* $T = \{T_1, T_2\}$ consists of two disjoint collections of blocks of X, T_1 and T_2, such that for every $A \in P_t(X)$, the number of blocks containing A is the same in both T_1 and T_2. The covering property of the elements of $P_t(X)$ in a t-(v,k) trade forces T_1 and T_2 to have the same number of blocks. This number, which we refer to as the *volume* of the trade is an important characteristic of the trade and is denoted by $vol(T)$. A trade is *void* if $\mathrm{vol}(T) = 0$. Clearly, both T_1 and T_2 must cover the same set of points which is called the *foundation* of the trade and is denoted by $found(T)$. For each point $x \in \mathrm{found}(T)$, we consider the set of all blocks containing it. By omitting x from these blocks, we obtain a $(t-1)$-$(v-1,k-1)$ trade and we call it the *derived* trade with respect to x. Two trades $T = \{T_1, T_2\}$ and $T' = \{T_1', T_2'\}$ are called *isomorphic* if there exists a bijection $\sigma : \mathrm{found}(T) \to \mathrm{found}(T')$ such that $\sigma(T) = \{\sigma(T_1), \sigma(T_2)\} = \{T_1', T_2'\} = T'$. An isomorphism σ such that $\sigma(T) = T$ is called an *automorphism* of T. Clearly, the set of all automorphisms of T forms a group. T is called *rigid* if its automorphism group is trivial.

Suppose the sets $P_k(X)$ and $P_t(X)$ are ordered with some legitimate ordering. Now, we define a $\binom{v}{t} \times \binom{v}{k}$ *inclusion* matrix, $P^v_{t,k} = (p_{AB})$, whose rows and columns are indexed by the elements of $P_t(X)$ and $P_k(X)$, respectively and is defined as follows:

$$p_{AB} = \begin{cases} 1 & A \subseteq B \\ 0, & \text{otherwise.} \end{cases}$$

For $t < k < v-t$, it is known that that the rank of $P^v_{t,k}$ is $\binom{v}{t}$ [11] and hence its kernel, denoted by $N^v_{t,k}$, is a $\mathbb{Z}$-module of dimension $\binom{v}{k} - \binom{v}{t}$.

3. An algorithm for the classification of trades

There are different bases for $N^v_{t,k}$ in the literature. For a brief description, the reader is referred to [19] where the authors also introduce a new basis which is called *the standard basis* by them. In [9], it is shown how this basis can be used to classify t-(v,k) trades as follows.

The $\binom{v}{k} - \binom{v}{t}$ trades of the standard basis constitute the columns of a matrix $M^v_{t,k}$ which has the following block structure:

$$M^v_{t,k} = \begin{bmatrix} I \\ \bar{M}^v_{t,k} \end{bmatrix}$$

, where I is the identity matrix of order $\binom{v}{k} - \binom{v}{t}$. The rows of $M^v_{t,k}$ corresponding to I are indexed by the so-called *starting blocks* and the remaining rows by the *non-starting blocks* [23]. In [8], the following recursive block structure is obtained for $M^v_{t,k}$:

$$M^v_{t,k} = \begin{bmatrix} M^{v-1}_{t-1,k-1} & 0 \\ N & M^{v-1}_{t,k} \end{bmatrix}$$

A direct way to produce and classify all t-(v,k) trades is to compute linear combinations of the columns of $M^v_{t,k}$ with coefficients $0, 1$ and -1, and then to decide whether the resulting trade is simple. Except for a few small values of the parameters, the dimension of $N^v_{t,k}$ makes it impractical to deal with all the columns of $M^v_{t,k}$. However, considering the recursive structure of $M^v_{t,k}$, the problem turns into classifying $(t-1)$-$(v-1,k-1)$ trades. Suppose $(t-1)$-$(v-1,k-1)$ trades have been classified so that we have one representative for each isomorphism class. Let T be a t-(v,k) trade and let D_1 be its derived trade with respect to the point 1. D_1 is clearly isomorphic to one of the representative $(t-1)$-$(v-1,k-1)$ trades, say D'_1. So, there exists a permutation π such that $D'_1 = \pi D_1$. Therefore, πT (an isomorphic copy of T) will be the extension of D'_1. Hence, to classify t-(v,k) trades, up to isomorphism, it suffices to extend only the representatives of the isomorphism classes of $(t-1)$-$(v-1,k-1)$ trades. The structure of $M^v_{t,k}$ helps us in determining t-(v,k) trades by extending $(t-1)$-$(v-1,k-1)$ trades. Let T' be a $(t-1)$-$(v-1,k-1)$ trade. Then the coefficients of the first $\binom{v-1}{k-1} - \binom{v-1}{t-1}$ columns of $M^v_{t,k}$ are specified by the blocks of T'. To extend T', it suffices to determine the coefficients of the remaining columns of $M^v_{t,k}$ in such a way that the result would be a simple trade. Finally, we have to check for isomorphism among all extensions to obtain a classification of all t-(v,k) trades. The pseudo code for the algorithm is as follows:

Algorithm. A recursive algorithm to classify trades
Procedure ClassifyTrads (t,v,k)
begin

```
        if (t=1) then
            Compute (t, v, k);
        else begin
            ClassifyTrades (t-1, v-1, k-1);
            Extend (t-1, v-1, k-1);
        end
    end
```

In the pseudo code, **Compute** produces all simple trades which can be obtained as a linear combination of the columns of $M^v_{t,k}$ with coefficients 1 and -1.

This recursive construction which ignores isomorphic copies of the derived trades, results in a considerable reduction in the number of extensions that are to be checked further to distinguish isomorphism classes. In [20], Khosrovshahi *et al.*, utilized mathematical reasoning to classify 2-$(v,3)$ trades for $v=6$ and 7. They proved that there exist 3 trades with foundation size 6 and 12 trades with foundation size 7. For $v=8$, there are over 560,000,000 distinct trades and we definitely need an algorithm to accomplish classification. Application of the above algorithm to this case results in about 300,000 extensions and reports (see [9]) the total number of 2-$(8,3)$ trades as 15,011. The algorithm has also been applied to some classes of 2-$(9,3)$ trades [9].

4. Combinatorial Trades in Classification of t-Designs and Large Sets

In general, any kind of complete classification of combinatorial objects is a challenging task. There are several common algorithmic approaches which are used to search for configurations with particular properties. Indeed, the use of clever computational techniques in enumeration theory, has not only enabled many existence and enumeration questions to be settled, but has also allowed larger classes of designs to be analyzed, often leading to the formulation of conjectures whose truth have been established for infinite families of designs. In [2, 4, 26, 25] some interesting instances are provided.

Since special classes of some combinatorial objects can be translated into the framework of trades, we can employ any algorithm for the classification of trades on these objects as well. Therefore, in this section, we demonstrate how the algorithm of Section 3 can be used on structures other than trades. We shall describe two such instances here.

4.1. Trades and t-designs. Let $T_1=(X,\mathscr{B})$ be a t-$(v,k,\binom{v-t}{k-t}/2)$ design. Then, $T=\{T_1,T_2\}$ is a trade with $\mathrm{vol}(T)=\binom{v}{k}/2$, where $T_2=(X,P_k(X)\backslash\mathscr{B})$. This means that, with some modification, the algorithm of the previous section can be employed for the classification of t-$(v,k,\binom{v-t}{k-t}/2)$ designs. In practical situations, however, the number of designs grows so rapidly that we may consider classifying a subset of such designs. A good choice is usually imposing in addition a non-trivial automorphism π, and then classify non-rigid designs. However, this does not affect the applicability of the algorithm for the case of non-rigid designs. To see this, note that if the design T_1 has a non-trivial automorphism π, then π is an automorphism of the trade T as well. Furthermore, any automorphism σ of the trade T such that $\sigma T_1=T_2$ corresponds to an isomorphism between the two designs T_1 and T_2. Indeed, for a non-rigid trade, we have the following simple lemma.

Lemma 4.1.1. *Let $T=\{T_1,T_2\}$ be a simple t-(v,k) trade of volume $\binom{v}{k}/2$. Then T is non-rigid if and only if one of the following occurs:*

(i) T_1 and T_2 are non-rigid designs.
(ii) T_1 and T_2 are isomorphic rigid designs.

Therefore, If we manage to classify all t-(v,k) trades of volume $\binom{v}{k}/2$ with a non-trivial automorphism, then we can also obtain the classification of all non-rigid t-$(v,k,\lambda_{tr}/2)$ designs, where $\lambda_{tr} = \binom{v-t}{k-t}$. In addition to this, we may construct some rigid designs as well. To see this, note that each of these trades, say $T = \{T_1, T_2\}$, consists of two t-$(v,k,\lambda_{tr}/2)$ designs T_1 and T_2. According to the above Lemma, if T admits exactly two automorphisms and T_1 is rigid, then T_2 is also a rigid design isomorphic to T_1. These trades produce t-$(v,k,\lambda_{tr}/2)$ rigid designs and the rest of the trades produced by the algorithm, add to the collection of non-rigid designs.

We now proceed to illustrate this with an example. We consider the class of 2-$(10,3,4)$ designs. The 60 blocks of these designs constitute exactly half of all $\binom{10}{3}$ blocks. let $T_1 = (X, \mathscr{B})$ correspond to a 2-$(10,3)$ design with 60 blocks and define T_2 as the remaining 60 blocks of $(X, P_k(X)\backslash\mathscr{B})$. Clearly, T_2 is a 2-$(10,3,4)$ design disjoint from T_1 and hence $T = \{T_1, T_2\}$ is a trade of volume 60. Therefore, we can run *ClassifyTrades* (2, 10, 3) to classify these trades.

In *ClassifyTrades* (2, 10, 3), 1-$(9,2)$ trades of volume 36 are first classified. Up to isomorphism, there exist exactly 10 non-isomorphic 1-$(9,2)$ trades, $S_1, \ldots, S_{10}$ [8]. The direct extensions of these derived trades result in over 200,000,000 solutions for which isomorphism testing would be clearly hard to carry out. To overcome this difficulty, as described before, we may consider a subset of this class consisting of non-rigid trades. We classify all 2-$(10,3)$ trades of volume 60 with a non-trivial automorphism, then we can also obtain the classification of all non-rigid 2-$(10,3,4)$ designs and at the same time construct some rigid designs. For a complete description and computational results, the interested reader is referred to [8], where this classification is used to enumerate (but not construct) the exact number of rigid trades and consequently the exact number of rigid 2-$(10,3,4)$ designs. Some other parameter sets on which this algorithm has successfully been applied are 3-$(11,4,4)$ designs [6], 4-$(12,5,4)$ designs 5-$(13,6,4)$ designs, and 6-$(14,7,4)$ designs [7].

4.2. **Trades and large sets.** A *large set* of t-(v,k,λ) designs, denoted by $LS[N](t,k,v)$, is a partition of $P_K(X)$ into N disjoint t-(v,k,λ) designs, where $N = \binom{v-t}{k-t}/\lambda$. We first consider $N = 2$ in which the large set will consist of two disjoint t-(v,k,λ) designs, where $\lambda = \binom{v}{k}/2$. The explanation of the previous section shows that an $LS[2](t,k,v)$ is exactly a t-(v,k) trade $T = \{T_1, T_2\}$ of volume $\binom{v}{k}/2$ where T_1 and T_2 are t-$(v,k,\binom{v-t}{k-t}/2)$ designs. This translation of large sets in terms of trades can again be exploited to obtain classification results about large sets in the same way that we did about designs. We therefore omit the details and mention that non-rigid 2-$(10,3)$ trades of volume 60, which we obtained in previous section, are exactly all non-rigid $LS[2](2,3,10)$ and the same is true about other parameter sets.

For $LS[N](t,k,v)$ with $N > 2$ (i.e., when the large set is not directly a trade), an algorithm based on trades has been devised to generate a large set whose existence question was open. The following simple observation, is the basis of the method used. Let $\mathbf{B} = \{\mathscr{B}_i\}_{i=1}^{N}$ be an $LS[N](t,k,v)$ with a non-trivial automorphism π of type $1^n p^m$ (in other words, π consists of m cycles of (prime) length p and n fixed points). Then if $\mathscr{B}_i^{\pi} \neq \mathscr{B}_i$ for some $1 \leq i \leq N$, $\mathbf{B}$ contains also $\mathscr{B}_i^{\pi^j}, j = 1, \ldots, (p-1)$. This means that if the large set contains one such design, i.e. $\mathscr{B}^{\pi} \neq \mathscr{B}$, then it also contains $\mathscr{B}^{\pi}, \mathscr{B}^{\pi^2}, \ldots, \mathscr{B}^{\pi^{p-1}}$. So, if we can produce all candidates for $\mathscr{B}$ in a large set, we construct at the same time p components of it. Now, if $N-2$ designs of large set are produced this way, then the remaining 2 (disjoint) designs will naturally form a trade and hence the standard basis for trades can be employed to find them.

To illustrate, the smallest open case on $LS[7](3,k,v)$, i.e. $LS[7](3,5,11)$, was resolved with this approach. Let $\mathbf{B} = \{\mathscr{B}_i\}_{i=1}^{7}$ be an instance of a $LS[7](3,5,11)$ which is invariant under $G =< \sigma >$ where σ is of type $1^2 3^3$. For this class, computational results show that $\mathbf{B}$ can not consist of G-invariant designs [24]. Therefore, the only possible structure for a G-invariant $LS[7](3,5,11)$, would be $(\mathscr{B}_1, \mathscr{B}_1^{\sigma}, \mathscr{B}_1^{\sigma^2})\mathscr{B}_2(\mathscr{B}_3, \mathscr{B}_3^{\sigma}, \mathscr{B}_3^{\sigma^2})$, where $\mathscr{B}_2^{\sigma} = \mathscr{B}_2$. We now propose the following algorithm to construct $\mathbf{B}$: take $\mathscr{B}_1$ to be any of 24 3-(11,5,4) designs invariant under σ. The choices for these designs can be obtained as in [10]. Next, determine all permutations π of type $1^2 3^3$ such that $\mathscr{B}_1, \mathscr{B}_1^{\pi}$, and $\mathscr{B}_1^{\pi^2}$ are disjoint. Relabel the points so that the designs are disjoint under the action of σ (i.e, $\pi = (1)(2)(345)(678)(9AB)$ for every solution). Now, we have a set of 3 disjoint isomorphic designs. Removing orbits of these designs from columns of the Kramer-Mesner matrix, obtain (if possible) all candidates for $\mathscr{B}_2$ (i.e. $\mathscr{B}_2^{\sigma} = \mathscr{B}_2$). Now, we are left with 198 blocks from among which we need to construct 2 more disjoint (isomorphic under σ) 3-(11,5,4) designs to complete the large set. These designs form a 3-(11,5) trade $T = \{T_1, T_2\}$ with $vol(T) = 66$ such that $T_1^{\sigma} = T_2$. So, we can implement backtracking on the standard basis of trades to produce such trades [5].

5. A LOCAL SEARCH ALGORITHM FOR THE CONSTRUCTION OF SIGNED t-DESIGNS

The search for a combinatorial structure may be formulated in the form of an optimization problem in which the "goodness" of an approximate solution is measured in terms of an objective or cost function. The goal is to find a configuration of minimum cost through applying a series of transformations. A new configuration is accepted if its cost does not exceed the cost of its pre-image. In this section, we present an algorithm of this type for the construction of t-designs which is based on trades. But first we prefer to scrutinize designs and trades from an algebraic point of view.

In the definitions for designs and trades, we used the term collection deliberately to reflect the fact that any block B is allowed to occur a positive number of times, say $m(B)$, in the structure. The integer $m(B)$ is called the multiplicity of B. Our formal definition of a t-(v,k,λ) design is then a vector of non-negative integers $\mathbf{m} = (m(B) : B \in P_k(X))$ with the property that $\sum_{B \supset A} m(B) = \lambda$, for all $A \in P_t(X)$. A t-(v,k) trade $T = \{T_1, T_2\}$ corresponds similarly to an integer vector $\mathbf{m} = (m(B) : B \in P_k(X))$ with the property that $\sum_{B \supset A} m(B) = 0$ for all $A \in P_t(X)$, provided that we negate the multiplicities of blocks of the T_1 (or T_2) part of the trade. A t-design or trade is *simple* if no two blocks are identical.

Another closely related configuration to the aforementioned structures, is the notion of a *signed* design. A signed design with positive integer parameters t, k, v, λ, is an integer vector $\mathbf{m} = (m(B) : B \in P_k(X))$ with the property that $\sum_{B \supset A} m(B) = \lambda$ for all $A \in P_t(X)$. Clearly a design is just a signed design with non-negative multiplicity for each block.

It will also be convenient for us to consider these structures as solutions to a matrix equation. We consider the system of equations $P^v_{t,k}\mathbf{m} = \lambda j$ (*), where j is the $\binom{v}{t}$-dimensional all-one vector. Any integral solution of (*) is called a t-(v,k,λ) signed design; any non-negative integral solution of (*), is called a t-(v,k,λ) design; and any integral solution of (*) for $\lambda = 0$, is called a t-(v,k) trade. Therefore, the set of all t-(v,k) trades is the kernel of $P^v_{t,k}$.

Let $\mathbf{m}$ be a t-(v,k,λ) signed design and let $\mathbf{m}_T$ be an arbitrary t-(v,k) trade. Clearly, $\mathbf{m} + \mathbf{m}_T$ is also a signed design. This augmentation of a signed design by a trade is referred to as *trade-off*. In this section we show how trade-off can be employed in the context of combinatorial optimization to devise a local search algorithm for constructing signed designs.

The basic idea is to choose a trade T in such a way that its augmentation to the signed design can reduce the number of blocks with negative multiplicities. If this can be accomplished, then we might hope to produce signed designs which are "close" to designs. For a set of parameters, the set (S) of feasible solution is taken to be the solutions of (*). Define the neighborhood of $\mathbf{m} \in S$ as elements of $S_{\mathbf{m}}$, where $S_{\mathbf{m}} = \{\mathbf{m}' \in S$ such that $\mathbf{m} - \mathbf{m}'$ is a minimal trade$\}$. In this model, the following can be considered as cost functions:

$$\varphi_1(\mathbf{m}) = \sum_{i=1}^{\binom{v}{k}} |m_i|,$$

$$\varphi_2(\mathbf{m}) = -\sum_{i=1, m_i<0}^{\binom{v}{k}} m_i,$$

$$\varphi_3(\mathbf{m}) = \varphi_1(\mathbf{m}) + \varphi_2(\mathbf{m}),$$

$$\varphi_4(\mathbf{m}) = \varphi_1(\mathbf{m}) - |\{m_i | m_i \neq 0\}|.$$

For example. the cost function φ_2 shows the number of blocks with negative multiplicities in the signed design. Clearly, $\varphi_2(\mathbf{m}) = 0$ if and only if $\mathbf{m} = 0$ or equivalently, when $\mathbf{m}$ is a t-design. We can now present the local search algorithm as follows.

Algorithm. A local search algorithm to construct signed t-designs.
Procedure ConstructSignedDesign (t, v, k, λ)
begin

```
    m = FindSignedDesign (t,v,k,λ);
    repeat
        m' = ChooseNeighbor (m,S_m);
        if φ(m') < φ(m) then m = m';
    until φ(m') ≥ φ(m) for all m' ∈ S_m;
end
```

In the pseudo code, **FindSignedDesign**, finds a t-(v, k, λ) signed design $\mathbf{m}$. An algorithm to do this can be found in [16]. **ChooseNeighbor** selects a new signed design from $S_{\mathbf{m}}$ by choosing first a set of blocks with undesirable multiplicity in $\mathbf{m}$ and then proceeds to construct all minimal trades passing through those blocks. For each such trade T, its augmentation to $\mathbf{m}$ is computed and tested with the appropriate cost function. If the test fails, another set of blocks is tested and processed until all the signed designs in the neighborhood are examined.

Examples of signed designs produced using this approach are 2-$(15,5,4)$, 4-$(12,5,4)$, and 4-$(15,5,5)$ designs [22, 17, 18].

REFERENCES

1. E. J. Billington, 2003, Combinatorial trades: a survey of recent results, in: W.D. Wallis (Ed), Designs 2002:Further Computational and Constructive Design Theory, Kluwer Academic Publishers, pp. 47-67. Book Series: Mathematics and its Applications, Vol. 563.
2. C. J. Colbourn, S. S. Magliveras, and D. R. Stinson, *Steiner triple systems of order 19 with nontrivial automorphism group*, Math. Comp. 59 (1992), 283–295.
3. M. Deza, P. Frankl, and M. N. Singhi, *On functions of stength t*, Combinatorica 3 (1983), no. 3-4, 331–339.
4. J. H. Dinitz, D. K. Garnick, and B. D. McKay, *There are* 526,915,620 *non-isomorphic 1-factorizations of* k_{12}, J. Combin. Designs 2 (1994), 273–285.
5. Z. Eslami, *LS [7](3,5,11) exists*, J. Combin. Des. 11 (2003), 312–316.
6. Z. Eslami and G.B.Khosrovshahi, *A complete classification of 3-(11,4,4) designs with a nontrivial automorphism group*, J. Combin. Designs 8 (2000), 419–425.
7. ______, *Some new 6-(14,7,4) designs*, J. Combin. Theory Ser. A. 93 (2001), 141–152.
8. Z. Eslami, G.B.Khosrovshahi, and B.Tayfeh-Rezaie, *On halvings of the 2-(10,3,8) design*, J. Statist. Plann. Inference 86 (2000), 411–419.
9. ______, *On classification of* 2-(8,3) *and* 2-(9,3) *trades*, J. Combin. Math. Combin. Comput. (2001), no. 38, 231–242.

10. Z. Eslami, G.B.Khosrovshahi, and M.M.Noori, *Enumeration of t-designs through intersection matrices*, Designs, Codes and Cryptography **32** (2004), 185–191.
11. W. Foody and A. Hedayat, *On the theory and applications of bib designs with repeated blocks*, Ann. Statist. **5** (1977), 932–945.
12. P. B. Gibbons, *computating techniques for the construction and analysis of block designs*, Technical Report 76-92, Computer Science Department, University of Toronto, 1976.
13. R. L. Graham, S.-Y.R.Li, , and W.-C.W.Li, *On the structure of t-designs*, SIAM J.Algebraic Discrete Methods **1** (1980), 8–14.
14. J. E. Graver and W. B. Jurkat, *The module structure of integral designs*, J. Combin. Theory Ser. A **15** (1973), 75–90.
15. A. S. Hedayat, *The theory of trade-off for t-designs*, Coding Theory and Design Theory, Part II, IMA Vol. Math. Appl. **21** (1990), 101–126, Springer, New York,.
16. A. S. Hedayat, G. B. Khosrovshahi, and D. Majumdar, *A prospect for a general method of constructing t-designs*, Disc. Appl. Math. **42** (1993), 31–50.
17. G. B. Khosrovshahi, A.Nowzari-Dalini, and R.Torabi, *some simple and automorphism-free* 2-$(15,5,4)$ *design*, J. Combin. Math. Des. **37** (1995), 285–292.
18. ———, *trading signed designs and some new* 4-$(12,5,4)$ *design*, Des. Codes Cryptography **11** (1997), 279–288.
19. G. B. Khosrovshahi and Ch.Maysoori, *On the bases for trades*, Linear Algebra Appl. **226-228** (1990), 731–748.
20. G. B. Khosrovshahi, H.R.Maimani, and R.Torabi, *Classification of* 2-$(6,3)$ *and* 2-$(7,3)$ *trades*, Aust. J. Combin. **19** (1999), no. 1, 55–72.
21. ———, *On trades: an update*, Discrete Appl. Math. **95** (1999), no. 1-3, 361–376.
22. ———, *An automorphism-free* 4-$(15,5,5)$ *design*, J. Combin. Math. Combin. Comput. **37** (2001), 187–192. MR 1 834 442
23. G. B. Khosrovshahi and S.Ajoodani-Namini, *A new basis for trades*, SIAM J. Discrete Math. (1990), no. 3, 364–372.
24. R. Laue, *private communications*, 2002.
25. E. Spence, *Classification of hadamard matrices of order* 24 *and* 28, Discrete Math. **140** (1995), 185–243.
26. ———, *The complete classification of steiner systems* $s(2,4,25)$, J. Combin. Designs **4** (1996), 285–300.
27. A. P. Street, 1996, Trades and defining sets, in: C.J. Colbourn and J. H. Dinitz (Eds), The CRC Handbook of Combinatorial Designs,CRC Press, New York.
28. H. S. White, F. N. Cole and L. D. Cummings,*Complete classification of the triad systems on fifteen elements*, Memoirs of the National Academy of Sciences **14** (1916), 1-189.