The Discovery of Simple 7-Designs with Automorphism Group $P\Gamma L(2,32)$

Anton Betten, Adalbert Kerber, Axel Kohnert, Reinhard Laue, Alfred Wassermann

Universität Bayreuth,
Lehrstuhl II für Mathematik
D-95440 Bayreuth

Abstract. A computer package is being developed at Bayreuth for the generation and investigation of discrete structures. The package is a C and C++ class library of powerful algorithms endowed with graphical interface modules. Standard applications can be run automatically whereas research projects mostly require small C or C++ programs. The basic philosophy behind the system is to transform problems into standard problems of e.g. group theory, graph theory, linear algebra, graphics, or databases and then to use highly specialized routines from that field to tackle the problems. The transformations required often follow the same principles especially in the case of generation and isomorphism testing. We therefore explain some of this background.

We relate orbit problems to double cosets and we offer a way to solve double coset problems in many important cases. Since the graph isomorphism problem is equivalent to a certain double coset problem, no polynomial algorithm can be expected to work in the general case. But the reduction techniques used still allow to solve problems of an interesting size. As an example we explain how the 7-designs in the title were found. The two simple 7-designs with parameters 7-$(33,8,10)$ and 7-$(33,8,16)$ are presented in this paper. To the best of our knowledge they are the first 7-designs with small λ and small number of blocks ever found. Teirlinck [19] had shown previously that non trivial t-designs without repeated blocks exist for all t. The smallest parameters for the case $t=7$ are 7-$(40320^{15}+7,8,40320^{15})$.

The designs have $P\Gamma L(2,32)$ as automorphism group, and they are constructed from the Kramer-Mesner method [7]. This group had previously been used by [13] in order to find simple 6-designs. The presentation of our results is compatible with that earlier publication.

The Kramer-Mesner method requires to solve a system of linear diophantine equations by a $\{0,1\}$-vector. We used the recent improvements by Schnorr of the LLL-algorithm for finding the two solutions to the 32×97 system.

1 Introduction

A library of C and C++ routines arose from a project for the constructive handling of discrete structures on a computer. The routines were written as part of diploma theses, doctoral theses and other research projects over several years. Already now the library of DISCRETA is powerful enough to support ambitious research activities. We describe here those parts of the package which were used in order to find 7-designs. Other aspects are mentioned in order to give an impression of the interfaces to users.

To begin with, we recall that a t-(v,k,λ)-design is defined to be a pair (V, B), consisting of a set V of vertices and a set B of blocks, where V is of order v and where each block $b \in B$ is a subset of order k in V. The other two parameters t and λ mean that each t-subset T of V is contained in exactly λ blocks.

A direct approach to the evaluation of all the t-(v,k,λ)-designs on V is easy to formulate: Consider the matrix

$$M_{t,k}^{v} := (m_{T,K}^{v}),$$

the rows of which are indexed by the t-subsets $T \subseteq V$ and the columns of which are indexed by the k-subsets $K \subseteq V$ while the entries themselves are defined to be

$$m_{T,K}^{v} := \begin{cases} 1 & \text{if } T \subseteq K \\ 0 & \text{otherwise.} \end{cases}$$

It is obvious that the set of all the t-(v,k,λ)-designs on V bijectively corresponds to the 0-1-solutions x of the system of linear equations

$$M_{t,k}^{v} \cdot x = \begin{pmatrix} \lambda \\ \vdots \\ \lambda \end{pmatrix}.$$

This is easy to see but difficult to solve. In the case of the quite moderate parameters $t := 7$, $v := 33$, $k := 8$ and $\lambda := 10$ the matrix $M_{7,8}^{33}$ has about $6 \cdot 10^{13}$ entries, so that there is no hope systematically to find a solution.

But there exists in fact a way of attacking this problem, namely by imposing a further condition. (This is, of course, risky, since the set of designs fulfilling this additional condition might be empty!) We impose the condition that a given subgroup A of the symmetric group S_V on the set of vertices is contained in the automorphism group $Aut(V, B)$ of the design in question:

$$A \leq Aut(V, B).$$

(The automorphism group $Aut(V, B)$ consists of the permutations $\pi \in S_V$ that induce permutations of the set B of blocks!) An interesting case of such a group is a certain subgroup of the symmetric group S_{33} on a set of 33 points which is usually denoted in geometry by $P\Gamma L(2, 32)$. It can be described as follows. Take the 32-dimensional vector space over $GF(2)$ and consider the set of its onedimensional subspaces, there are exactly 33 of such subspaces. The general linear group $GL(2, 32)$ induces a permutation group on this set, which is denoted by $PGL(2, 32)$. This group together with the permutation coming from the Frobenius automorphism $\kappa \mapsto \kappa^2$ (when applied to the coordinates of the vectors) generates the group $P\Gamma L(2, 32)$.

As soon as we have imposed this condition $A \leq Aut(V,B)$ we can considerably reduce our numerical problem: $M^{v}_{t,k}$ can be replaced by the matrix

$$M^A_{t,k} := (m^A_{T,K}),$$

the rows of which are indexed by the elements of an (arbitrary) transversal $\mathcal{T}$ of the set of orbits of A on the set $\binom{V}{t}$ of t-subsets, while the columns are indexed by the elements of an (arbitrary) transversal $\mathcal{T}'$ of the set of orbits of A on the set $\binom{V}{k}$ of k-subsets of V :

$$T \in \mathcal{T}(Orb(A, \binom{V}{t})),\ K \in \mathcal{T}'(Orb(A, \binom{V}{k})).$$

The matrix $M^A_{t,k}$ is therefore of size

$$|Orb(A, \binom{V}{t})| \times |Orb(A, \binom{V}{k})|,$$

which is in fact 32×97 in the above mentioned particular example, and so the *data reduction is enormous*, it is in fact by the *factor* $2 \cdot 10^{10}$ in our example. The entries of the matrix are defined by

$$m^A_{T,K} := |\{K' \in Orb(K) \mid T \subseteq K'\}|.$$

($Orb(K)$ means the orbit of K under the action of A on V.) This matrix is called the *Kramer-Mesner* [7] matrix, since their theorem says that the set of t-(v,k,λ)-designs on V is bijective to the set of 0-1-solutions x of

$$M^A_{t,k} \cdot x = \begin{pmatrix} \lambda \\ \vdots \\ \lambda \end{pmatrix}.$$

It therefore remains to evaluate the Kramer-Mesner matrix and to find a 0-1-solution of this system of linear equations.

The evaluation of the Kramer-Mesner matrix can be done by application of two basic principles of Algebraic Combinatorics which we should like to describe here. The first of the basic principles that come in makes use of the fact that a transversal of orbits can be obtained from a transversal of double cosets as soon as we have a transitive group at hand. This fact is described in the following lemma (which is old, but we do not know where exactly it appeared for the first time):

The Split Lemma. *Let G be a group acting transitively on a set Ω. Then the orbits of a subgroup U of G on Ω correspond bijectively to the double cosets $N_G(\omega)\backslash G/U$ by the mapping $\omega^{gU} \mapsto N_G(\omega)gU$, where $N_G(\omega)$ is the stabilizer of a fixed $\omega \in \Omega$ under the G-action.*

This lemma is known in special applications, for example coding theory [18] and theoretical chemistry. In the case of designs we can apply it, since the symmetric group S_V forms a single orbit on $\binom{V}{t}$ as well as on $\binom{V}{k}$. We shall give details in the following section.

There are also more general situations where this lemma can be applied, namely in each case when we distinguish *labelled and unlabelled structures*. Discrete structures are represented by a data structure which in general is not unique for the object presented. For example a graph has to be labelled, which means the vertices must be numbered before the computer can handle it. But for n vertices there are $n!$ different labellings with labels $1, \ldots, n$. Analoguous ambiguities arise with t-designs, groups, codes and other kinds of discrete structures, the unlabelled structure is defined to be an equivalence class of the labelled one, or, in other terms, an isomorphism class of labelled structures. Therefore we consider isomorphism problems with highest priority. Usually, the set of labelled structures is very big, and many of them will be isomorphic. Then one has to find a group acting on the set of objects such that the isomorphism types are just the orbits of that particular group. Algorithms for finding a full set of orbit representatives will finally give the desired isomorphism types. For example the set of labelled graphs on v vertices is of order $2^{\binom{v}{2}}$, and the acting group is the symmetric group on the vertices again. Since this group acts transitively on the set of labelled graphs with v vertices and given number of edges, the split lemma in fact shows that these graphs can be obtained from double coset representatives in a symmetric group. We can explain here, in addition, the application to coding theory. A linear code is a subspace of some dimension k, say, of a vector space V of a dimension n over a finite field $GF(q)$ for some prime power q. The code vectors are n-tuples with entries from $GF(q)$. We consider two codes as equivalent if there exists a permutation of the positions of all entries transforming one code into the other or we can in addition multiply all entries at fixed positions by the same constant different from 0. This means that the group $GF(q)^* \wr S_n$ presented as the subgroup U of all monomial matrices in $G = GL(n,q)$ acts on the set of subspaces. Since $GL(n,q)$ is transitive on the set of all subspaces of a fixed dimension k, by the split lemma the orbits of U on the set of these subspaces correspond to the double cosets $N_{GL(n,q)}(K)\backslash GL(n,q)/GF(q)^* \wr S_n$, where K is a fixed subspace of dimension k of V.

Thus we have demonstrated, how double coset transversals help to evaluate designs, graphs and linear codes by suitable applications of the split lemma mentioned above.

It remains to tell something about the evaluation of double coset transversals. Here the second basic principle comes in which we would like to mention here.

The basic algebraic tool is that of homomorphism, which means compatible mapping. It serves very well in a stepwise simplification of group actions and corresponding constructive methods in algebraic combinatorics, to. Here is the corresponding lemma:

The Homomorphism Principle. *Let a group G act on a set Ω_1 and on a set Ω_2. Let $\sigma : \Omega_1 \to \Omega_2$ be a mapping that is compatible with both group actions.*

Then, for each $\omega \in \Omega_2$ and each $g \in G$ the sets $\sigma^{-1}(\omega)$ and $\sigma^{-1}(\omega^g)$ intersect the same orbits of G on Ω_1. If $\omega_1, \omega_2 \in \sigma^{-1}(\omega)$, for some $\omega \in \Omega_2$, and $\omega_1^g = \omega_2$, for some $g \in G$, then $g \in N_G(\omega)$.

The proof is obvious.

We apply the homomorphism principle in two different ways. Firstly, we assume that a solution of the orbit problem is already known in the image domain of σ. Then only the preimage sets $\sigma^{-1}(\omega)$ of representatives ω and as acting group on $\sigma^{-1}(\omega)$ only the stabilizer $N_G(\omega)$ have to be considered. The size of the full set of all preimages of one orbit is reduced to a fraction and the order of the acting group is reduced by the same factor, that is by the length of the orbit in the image domain. Therefore using a series of systematic simplifications by homomorphisms reduces the overall complexity about logarithmically.

The second way we use the homomorphism principle is to deduce a solution in the image domain of σ from a solution of the orbit problem in the preimage domain. We call this application a *fusion*.

A combination of both principles can be used to find double coset representatives [14].

Theorem 1. *Let A_2, A_1, B be subgroups of a group G and $A_2 < A_1$. Then the following mapping between the respective sets $A_i \backslash G$ of right cosets,*

$$\sigma : A_2\backslash G \to A_1\backslash G \ ,$$

sending the coset $A_2 g$ onto the coset $A_1 g$ is compatible with the action of B on $A_2\backslash G$ and $A_1\backslash G$ by multiplication from the right. If $A_1 = \bigcup_{x\in X} A_2 x$ then $\sigma^{-1}(A_1 g) = \bigcup_{x\in X} A_2 x g$. A set of double coset representatives for $A_2\backslash G/B$ is obtained from a set T of double coset representatives for $A_1\backslash G/B$ by computing representatives from the orbits of $t^{-1}A_1 t \cap B$ on $\sigma^{-1}(A_1 t)$, for each $t \in T$.

In order to obtain a set Γ_1 of double coset representatives for $A_1\backslash G/B$ from such a set Γ_2 for $A_2\backslash G/B$ let γ run through Γ_2, put $\rho = \sigma(\gamma)$ into Γ_1, and for each element in $\sigma^{-1}(\rho)$ remove the representative of its double coset from Γ_2.

Proof. In order to prove this we only need to interpret an orbit $\{Agb_1, Agb_2, \ldots, Agb_r\}$ of B on the set of right cosets of a subgroup A of G as the set of those cosets which lie in the same double coset AgB. The homomorphism principle yields the assertion, since $t^{-1}A_1 t \cap B$ is just the stabilizer of $A_1 t$ in B. □

This may suffice as a description of two basic principles of Algebraic Combinatorics, we should like now to give a detailed description of their application in order to find the first 7-designs with moderate parameters, to be more precise: to find a 7-$(33, 8, 10)$-design via an evaluation of the Kramer-Mesner matrix of $P\Gamma L(2, 32)$ and then finding a 0-1-solution of the corresponding system of linear equations.

2 Computation of the Kramer-Mesner Matrix

Recall from above that we have to evaluate *two transversals of double cosets* in the symmetric group S_{33}. On the left hand side there is in the first case the stabilizer of a 7-subset of the set of 33 vertices, and in the second case it is the stabilizer of an 8-subset. On the right hand side we have, in both cases, the group $P\Gamma L(2,32)$. We shall describe a way of solving these two problems in one wash by using a so-called *ladder* of subgroups, which first meets the stabilizer of a 7-subset and ends up in a stabilizer of an 8-subset. But let us describe that slightly more general in order to make the generality quite clear. Let us discuss a way of construction of a double coset transversal in an arbitrary finite group G.

Since in many cases we cannot find chains of subgroups with small indices leading from G downwards to a prescribed subgroup A, we use some deviations instead of a direct way. In fact, we may proceed going along a sequence of subgroups A_i where *either* $A_i \leq A_{i-1}$ *or* $A_i \geq A_{i-1}$. The key to this method is to consider also cases $A_i \leq A_{i-1}$, where representatives for double cosets $A_i\backslash G/B$ are known and then, *by fusion*, reduce the set to double coset representatives for $A_{i-1}\backslash G/B$. The discussion above leads directly to an algorithm, see [10, 14]. For a recent object oriented version see [20].

An example indicates how one can obtain a set of double coset representatives in S_{33} where on one side the group A is a Young subgroup being the normalizer of a set $K = \{1,\ldots,k\}$ for some $k < 33$. In the application to the construction of a 7-design we choose as B the group $P\Gamma L(2,32)$. Of course S_{33} is transitive on the set of all subsets of the same cardinality k. Therefore, by the split lemma, the orbits of B on the set of these subsets correspond to the double cosets of the stabilizer A of K in S_{33} and B. We indicate the sequence of subgroups leading from S_{33} to A, which can be used for a determination of the double cosets.

If $\mathcal{B} = (B_1, B_2, \ldots, B_k)$ is a partition of $\{1,\ldots,n\}$ into blocks B_i the corresponding Young-subgroup of S_n is the normalizer $N_{S_n}(B_1,\ldots,B_k)$ of all these blocks. Then our sequence of subgroups is as in Fig.1.

All orbit problems in this example deal with very small sets of points only. In contrast to this, the index of a Young subgroup in S_n is a usually very big multinomial coefficient. Of course the set of orbit representatives will be also very large, since the multinomial coefficient can be reduced at most by the factor $|B|$.

A similar chain of subgroups exists in General Linear Groups. There one can take the normalizers of subspaces instead of Young subgroups. If $(T_1, T_2, \ldots, T_n)$ is an ascending chain of subspaces of a vector space $V(n,q)$ of dimension n then we use the subgroup relation

$$N_{GL(n,q)}(T_i) \geq N_{GL(n,q)}(T_i) \cap N_{GL(n,q)}(T_{i-1}) \leq N_{GL(n,q)}(T_{i-1})$$

for all i in order to construct a sequence along which we compute representatives for the double cosets with the monomial group. Again the full General Linear Group is transitive on the set of all subspaces of a fixed dimension such that the

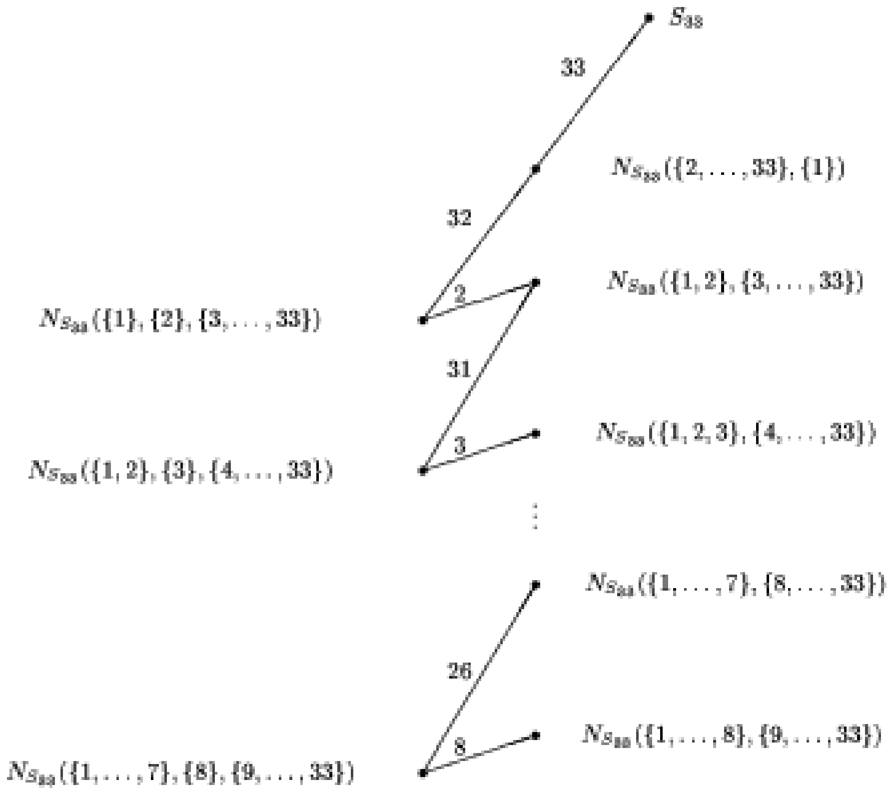

Fig. 1.

split lemma applies. Therefore one can use the same algorithm with groups of a different kind to solve the problem of code construction [1].

A careful analysis of the fusion of the step from the normalizer $N_{S_{33}}(\{1,\dots,7\},\{8,\dots,33\})$ to the normalizer $N_{S_{33}}(\{1,\dots,8\},\{9,\dots,33\})$ shows that for each representative T of a 7-orbit of A and for each 8-orbit O_j one gets the number $m(T,j)$ of 8-subsets in O_j that contain T. This is the information needed to form the Kramer-Mesner matrix M which allows to find a 7-design. The number $m(T,j)$ is just the entry $m^A_{T,K}$ for some $K \in O_j$. It is easy to see that this number is independent of the choice of the representative T.

We look for 7-$(33,8,\lambda)$ designs having the group $B = P\Gamma L(2,32)$ as an automorphism group. Such a design then consists of full orbits O_j. One has to choose appropriate columns of M to get the desired design. Each column selected stands for selecting all 8-subsets of the corresponding orbit for the design. The condition for a simple design says that in each row of the Kramer-Mesner matrix the entries of the selected columns must sum up to λ.

Since the designs constructed in this way have at least $P\Gamma L(2,32)$ as its automorphism group, one should ask for the full automorphism group. While such a question is hard to answer in general, in this case we only have to notice that by [12] $P\Gamma L(2,32)$ is a maximal subgroup of S_{33}. Thus, the only possibilities for the full automorphism group could be $P\Gamma L(2,32)$ or S_{32}, the latter case being impossible since it would require all 8-subsets to be included into the design because of the transitivity of S_{33} on this set. We therefore conclude that any incomplete design having $P\Gamma L(2,32)$ as an automorphism group must have this group as the full automorphism group.

We have included the Kramer-Mesner matrix for this problem for convenience of the reader at the end of the article. Actually this matrix had appeared already in [13] together with a description of simple 6-designs. To make our results comparable to that paper we decided to use the representation of the matrix there. Our own result differed only by some permutation of the 97 columns and 32 rows.

3 {0, 1}-Solutions of Linear Diophantine Systems

Now it remains to solve for the Kramer-Mesner matrix M, an $l \times s$-matrix, the equation

$$M \cdot v = \lambda(1,\ldots,1)^t \text{ for a } \{0,1\}\text{-vector } v \ . \tag{1}$$

This is a special instance of the multi-dimensional *subset sum* problem which is known to be NP-complete [4]. Our approach therefore uses an algorithm which generally solves only a weaker problem, but often also gives a solution to (1). In fact, we could find such a solution for a difficult problem in this way as shown below.

As in [2, 3, 8] we reduce the problem to that of finding short vectors in a lattice. At the moment a polynomial method to find short vectors in a lattice is not known. But the algorithm of Lenstra, Lenstra and Lovász [11] guarantees to find a nontrivial vector in an m-dimensional lattice that has at most $2^{m/2}$ the length of the shortest nontrivial vector in this lattice. This does not look very promising, but in practice the so called LLL-algorithm performs much better than is guaranteed by its worst case bounds.

Meanwhile there were several improvements of the original algorithm and lattices have been found which are better suited to the subset sum problem, [8, 15, 16, 17]. So the performance of the algorithm dramatically improved.

Let $\mathbb{R}^n$ be the n-dimensional $\mathbb{R}$-vector space with the ordinary inner product $\langle .,. \rangle$. A discrete, additive subgroup $L \subset \mathbb{R}^n$ is called a *lattice*.

Every lattice L is generated by a set of linearly independent vectors $b_1,\ldots,b_m \in L$, the *basis* of L:

$$L = L(b_1,\ldots,b_m) = \{x_1 b_1 + \cdots + x_m b_m \mid x_1,\ldots,x_m \in \mathbb{Z}\} \ .$$

m is called the *rank* of the lattice L.

For a sequence of linear independent vectors $b_1,\ldots,b_m \in \mathbb{R}^n$ we let $b_1^*,\ldots,b_m^*$ be the *Gram-Schmidt orthogonalized* sequence. We thus have

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \quad \text{for } i = 1,\ldots,m, \text{ where } \mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \ .$$

The vectors $b_1^*, \ldots, b_m^*$ are linearly independent, but in general they are not in the lattice spanned by $b_1, \ldots, b_m$. Note that the orthogonalized vectors $b_1^*, \ldots, b_m^*$ depend on the order of the basis vectors $b_1, \ldots, b_m$.

Definition 2. A basis $b_1, \ldots, b_m$ of the lattice L is called *LLL-reduced with* δ if

$$|\mu_{i,j}| \leq 1/2 \quad \text{for } 1 \leq j < i \leq m \ , \tag{2}$$

$$\delta \|b_k^*\|^2 \leq \|b_{k+1}^* + \mu_{k+1,k} b_k^*\|^2 \quad \text{for } k = 1, \ldots, m-1 \ , \tag{3}$$

where δ is a constant with $1/4 < \delta \leq 1$.

In order to find a lattice basis which fullfils (2) and (3) a finite number of two kinds of linear transformation are applied:

Algorithm (LLL-algorithm, see [11]). Set $k := 1$. Now do until $k = m - 1$:

1. For $i = 1, \ldots, k-1$ replace b_k by $b_k - rb_j$, where $r = \lceil \mu_{k,j} \rfloor$ is the nearest integer to $\mu_{k,j}$.
2. if $\delta \|b_k^*\|^2 > \|b_{k+1}^* + \mu_{k+1,k} b_k^*\|^2$ then interchange b_{k+1} and b_k and set $k := \max(k-1, 1)$,
   otherwise set $k := k + 1$.

Remark. With step 1 of the algorithm we achieve condition (2) which assures that the LLL-reduced basis vectors are "as orthogonal as possible".

In condition (3) the vector $b_{k+1}^* + \mu_{k+1,k} b_k^*$ is the orthogonal projection of the vector b_{k+1} on the orthogonal complement of the subspace generated by $b_1, \ldots, b_{k-1}$. In other words to fullfil condition (3) step 2. of the algorithm does the following: if for some $k \in \{1, \ldots, m-1\}$ the last vector of the Gram-Schmidt orthogonalized sequence $b_1^*, \ldots, b_{k-1}^*, b_{k+1}^*$ is shorter than the last vector of the Gram-Schmidt orthogonalized sequence $b_1^*, \ldots, b_{k-1}^*, b_k^*$ by at least a factor $\delta < 1$ the two vectors are swaped, i.e. b_{k+1} is the new vector b_k.

This is the natural generalization of an algorithm by Gauss [5, Art. 171, 183, 272] to reduce binary, respectively ternary quadratic forms.

For a lattice $L \subset \mathbb{R}^n$ of rank m the successive minima $\lambda_1, \ldots, \lambda_m$ of L are defined through: $\lambda_i = \lambda_i(L)$ is the smallest radius r of a ball centered at the origin which contains exactly i linearly independent lattice vectors. It follows that $\lambda_1(L)$ is the euclidean length of the shortest nonzero lattice vector of L.

The following theorem from [11] states that an LLL-reduced basis contains relatively short vectors.

Theorem 3. *Every basis* $b_1, \ldots, b_m$ *that is LLL-reduced with* $1/4 < \delta \leq 1$ *satisfies*

$$\|b_i\| \leq \left(\frac{4}{4\delta - 1}\right)^{(m-1)/2} \lambda_i(L) \ . \tag{4}$$

In [11] the authors also give the following running time:

Theorem 4. *Let $b_1, \ldots, b_m$ be an ordered basis for an integer lattice L such that $\|b_i\|^2 \leq B$ for $1 \leq i \leq m$. Then the LLL-algorithm computes a LLL-reduced basis for L using at most $O(m^4 \log_2 B)$ arithmetic operations and the integers on which these operations are performed have length at most $O(m \log_2 B)$.*

Several speedups of the algorithm have been proposed. Schnorr [16, 17] introduces variants which use floating point arithmetic to circumvent the time consuming use of long integer arithmetic.

In [17] the authors use the so called *deep insertions*: Instead of (3) – where the LLL-algorithm behaves like the bubble sort method – they interchange b_k not just with b_{k-1} but with the leftmost vector b_i, $1 \leq i < k$, for which $\|b_i^*\|^2$ is at least decreased by a factor δ.

There are other kinds of lattice basis reduction beside of LLL-reduction. One classical definition of lattice basis reduction is *Korkine-Zolotarev reduction* [6]: Let $b_1, \ldots, b_m$ be an ordered basis of the lattice L. We define L_i as the orthogonal projection of L in $\langle b_1, \ldots, b_{i-1}\rangle^\perp$. Then L_i is a lattice of rank $m-i+1$. Further we denote with $L_i(b_i, \ldots, b_k)$ with $i \leq k \leq m$ as the orthogonal projection of the lattice spanned by the vectors $b_1, \ldots, b_k$ in $\langle b_1, \ldots, b_{i-1}\rangle^\perp$.

Denote with $\pi_i : \mathbb{R}^n \rightarrow \langle b_1, \ldots, b_{i-1}\rangle^\perp$ the orthogonal projection so that $b - \pi_i(b) \in \langle b_1, \ldots, b_{i-1}\rangle$.

Definition 5. An ordered basis $b_1, \ldots, b_m$ of a lattice L is called *Korkine-Zolotarev reduced* [6] if it fulfills (2) and if

$$\|b_i^*\| = \lambda_1(L_i) \quad \text{for } i = 1, \ldots, m \ .$$

The following theorem from [15] reveals that Korkine-Zolotarev reduction is stronger than LLL-reduction.

Theorem 6. *A Korkine-Zolotarev reduced basis $b_1, \ldots, b_m$ satisfies*

$$\sqrt{\frac{4}{i+3}}\lambda_i(L) \leq \|b_i\| \leq \sqrt{\frac{i+3}{4}}\lambda_i(L) \quad \text{for } i = 1, \ldots, m \ .$$

The bad news are there is no polynomial time algorithm for Korkine-Zolotarev reduction known. In [15, 17] the authors define a weakened version of Korkine-Zolotarev reduction:

Definition 7. Let β be an integer with $2 \leq \beta < m$. A basis $b_1, \ldots, b_m$ is called *β-reduced* if it satisfies (2) and if for $i = 2, \ldots, m-\beta+1$ the orthogonal projections of $b_i, \ldots, b_{i+\beta-1}$ in $\langle b_1, \ldots, b_{i-1}\rangle^\perp$ form a Korkine-Zolotarev reduced basis of the lattice $\pi_i(L(b_i, \ldots, b_{i+\beta-1}))$.

A basis $b_1, \ldots, b_m$ is called *β-reduced with δ* if (2) is satisfied and if

$$\delta\|b_i^*\| \leq \lambda_1(L_i(b_i, \ldots, b_{i+\beta-1})) \quad \text{for } i = 1, \ldots, m-\beta+1 \ .$$

Remark. Note that a LLL-reduced basis with δ is 2-reduced with δ. Actually in case of $\beta > 2$ step 2 of the LLL-algorithm is generalized in β-reduction with δ to the following:

Instead of looking whether a swap of the vectors b_{k+1} and b_k would give a shorter new b_k^* we are searching for the linear combination of the vectors $b_k, \ldots, b_{k+\beta-1}$ as new vector b_k which produces the shortest vector b_k^*.

In [15, 17] the length of the basis after β-reduction is bounded as follows:

Theorem 8. *Every β-reduced basis $b_1, \ldots, b_m$ of a a lattice L satisfies*

$$\|b_1\|^2 \leq \alpha_\beta^{(m-1)/(\beta-1)} \lambda_1(L)^2$$

provided that $\beta - 1$ divides $m - 1$.

The constant α_β is the maximum of $\|b_1\|/\|b_\beta^*\|$ taken over all Korkine-Zolotarev reduced bases $b_1, \ldots, b_\beta$. ¿From [15] we know that $\alpha_2 = \frac{4}{3}$, $\alpha_3 = \frac{3}{2}$ and $\alpha_\beta \leq \beta^{1+\ln \beta}$. With β increasing $\alpha_\beta^{1/(\beta-1)}$ converges to 1.

Often the vectors of a reduced lattice basis still are not short enough to solve the linear diophantine systems. Since a reduced lattice basis depends on the order of the initial lattice basis, we shuffle the basis vectors after β-reducing the lattice and repeat this process several times. Kreher and Radziszowski [8] gave the following improvement of the algorithm: After each β-reduction step we test if there are pairs (i, j) with $1 \leq i < j \leq m$ so that $\|b_i \pm b_j\| < \|b_i\|$. If this is the case we set b_i to $b_i \pm b_j$. Then we start again with shuffling and β-reduction.

To solve (1) we combine the approach of Kreher and Radziszowski [8] with the new ideas of Schnorr et al. [3, 15, 16, 17].

This means that we apply lattice basis reduction to the following lattice basis L to get a reduced lattice basis L':

$$L := \left(\begin{array}{ccc|cc} & & & c_0 1 & 0 \\ & c_0 M & & \vdots & \vdots \\ & & & c_0 1 & 0 \\ \hline c_1 2 & & 0 & 0 & c_1 1 \\ & \ddots & & \vdots & \vdots \\ 0 & & c_1 2 & 0 & c_1 1 \\ \hline 0 & \ldots & 0 & 1 & 0 \\ 0 & \ldots & 0 & 0 & 1 \end{array}\right) ,$$

where M is a $l \times s$-matrix and c_0 and c_1 are constants which control the behaviour of the algorithm. The choice of c_0 should force an exact solution over the integers whereas a good choice of c_1 will yield a $\{0, 1\}$-solution:

Suppose c_0 is large. Then by the reduction the whole upper block of about the first $s - l$ columns and l rows will be transformed to 0, because each nonzero entry would be divisible by c_0 which means that the euclidean length of the whole correspondending column would be large. Since the rank of the Kramer-Mesner matrix M is about l only $s - l$ vectors of the reduced basis can consist only of zeros in the first l rows. c_1 should be approximately the expected value of λ.

The algorithm has found a solution if L' contains a vector $(b_{i,1},\ldots,b_{i,z})^t$ with the following form:

$$|b_{i,z}| = 1, \quad b_{i,1} = \ldots = b_{i,l} = 0, \quad |b_{i,l+1}| = \ldots = |b_{i,z-2}| = c_1 \;\; ,$$

where l is the number of rows of the Kramer-Mesner matrix M and $z = l + s + 2$.

The Kramer-Mesner matrix M of the 7-$(33, 8, 10)$ and 7-$(33, 8, 16)$ designs has 32 rows and 97 columns which result in a lattice L with 131 rows and 99 columns. We used β-reduction with deep insertions, where we chose $\beta = 40$, $\delta = 0.999999999$, $c_0 = 30$ and $c_1 = 10$. We found the following solutions after one iteration which took about 9 minutes on a PC 486 with 66 MHz and 16 MB RAM:

The 32×97 Kramer-Mesner matrix M:

```
2222222222222200000000000000000000000000000000000000000000000000000000000000000000000000000000000
2101110000000211112111111111111100000000000000000000000000000000000000000000000000000000000000000
1110000001000100200000000000010001111121121111111100000000000000000000000000000000000000000000000
0011000000000021001000000000000010001100000010001213111111111110000000000000000000000000000000000
0001200000000000000010000010011100000000100100100110001000011211111111100000000000000000000000000
0000110000000010000001000000000100100000110011100000010100100000011002111111110000000000000000000
0000018000110100100000101001001000100000100010010101001000000100010001001000100000000000000000000
0000004200000000500000200520000000002000000200000000000200000500000000000200005000000000000000000
0000002300000000000000002000000200000050000020000000000000000002000000500000020050000000000000000
0001000121000001100100000101000000001000100010100100001000000010011100000101001011100000000000000
0000000012000100110110100001100000011010001100100000100000000010000000010000101001100000000000000
0000000200002000000000210002000002000005200000002000000000000000002000000000200000000000000000000
0000000010120000000010100002500000000000000011100000011001011000010010010001050010100000000000000
0200000000002210020000101000001111000000001000000000000000000100011001010000000100001011100000000
0010100000001001001200100000010001000000200000001001011011000100000000000010100000001001110000000
0000110011000010100001010000102000000000000010100000010100000001010010000000001000000001021100111
0011020000000010010000000100100000010001000000000000101000010000010001100001000101001200000001001
0000002200000000024002000000000000000000000000000000000000000002002200000002200000000002000000000
0000010010011000010110010011001110000100000000100000101000100000000100000010110001000000001000110
0200000200000200000200000200002000000000002000000000000000002000000000000000020000000000000200000
0001000001010011000000000000100000110001000100001000010000001000011000000100000000001000310101001
0010100000011000100200101000000000100011000010110010011000000000020000001000000000000010000010000
0200000000000001000001000000000010220100000110000000000001001001001100000010001110100210000100001
0011000000000000100000100000001010001101000000100010000001010000000201000001010000000101002000011
0000000200000000000000200200000000000000200000000200000000200000000020202200000000000020000000200
0010000000000000000110000101000000000001000000010010100002001000001101111001100000000111000000021
0000000000100000100000110100000000000010001000000000002001100000200000010010000111000100110001001
0000000000000000000010010100010110000100000100001010111000000010001010000201001100000000000001100
0000000010000001000000000001010001101000100100001110010001000000100100011000100000000011100101100
0000000050000000000000000000000000000000500000000000000050000000000000000000000000000000050000001
0000000000000000000000000000000000000000000000050000000000000000000000500000000000000000000000001
0000000000000000000000000000000000000000000000050000000000000000000005000000000000000005000050001
```

with the solution vectors v for $\lambda = 10$ and $\lambda = 16$ respectively:

```
0011100010100100110001101000010010000000101101100010111110000000010010010100001101110110010101110000
1100011101011011001110010111101101111111010010011101000011111110110110101111001000100110101000111
```

To make the paper self - contained we include from [13] the permutation representation of $P\Gamma L(2,32)$ and representatives from the orbits on all 7- and 8-subsets that correspond to the Kramer-Mesner matrix.

The group $P\Gamma L(2,32)$ can be presented as generated by the following two permutations of $\{1,\cdots,33\}$:

$\alpha=$ (1 2 4 8 16)(3 6 12 24 17)(5 10 20 9 18)(7 14 28 25 19) (11 22 13 26 21)(15 30 29 27 23)(31)(32)(33)

$\beta=$ (1 18 30)(2 21 12)(3 10 28)(4 31 32)(5 24 14)(6 7 17)(8 25 27) (9 19 20)(11 15 13)(16 23 29)(22 33 26).

There are 32 orbits on the set of all 7-subsets and 97 orbits on the set of all 8-subsets.

orbits on 7-subsets of V

Nr	representative	length
1.	1 2 3 4 5 6 7	81840
2.	1 2 3 4 5 6 8	163680
3.	1 2 3 4 5 6 9	163680
4.	1 2 3 4 5 6 10	163680
5.	1 2 3 4 5 6 11	163680
6.	1 2 3 4 5 6 12	163680
7.	1 2 3 4 5 6 13	163680
8.	1 2 3 4 5 6 14	81840
9.	1 2 3 4 5 6 15	81840
10.	1 2 3 4 5 6 16	163680
11.	1 2 3 4 5 6 17	163680
12.	1 2 3 4 5 6 19	81840
13.	1 2 3 4 5 6 22	163680
14.	1 2 3 4 5 7 9	163680
15.	1 2 3 4 5 7 10	163680
16.	1 2 3 4 5 7 12	163680
17.	1 2 3 4 5 7 13	163680
18.	1 2 3 4 5 7 15	81840
19.	1 2 3 4 5 7 20	163680
20.	1 2 3 4 5 7 24	81840
21.	1 2 3 4 5 8 10	163680
22.	1 2 3 4 5 8 11	163680
23.	1 2 3 4 5 8 12	163680
24.	1 2 3 4 5 8 13	163680
25.	1 2 3 4 5 8 17	81840
26.	1 2 3 4 5 8 24	163680
27.	1 2 3 4 5 8 26	163680
28.	1 2 3 4 5 9 11	163680
29.	1 2 3 4 5 9 12	163680
30.	1 2 3 4 5 9 17	32736
31.	1 2 3 4 5 10 12	32736
32.	1 2 3 4 5 11 16	32736

orbits on 8-subsets of V

Nr	representative	length
1.	1 2 3 4 5 6 7 8	81840
2.	1 2 3 4 5 6 7 9	163680
3.	1 2 3 4 5 6 7 10	163680
4.	1 2 3 4 5 6 7 11	163680
5.	1 2 3 4 5 6 7 12	163680
6.	1 2 3 4 5 6 7 13	163680
7.	1 2 3 4 5 6 7 14	163680
8.	1 2 3 4 5 6 7 15	163680
9.	1 2 3 4 5 6 7 16	163680
10.	1 2 3 4 5 6 7 17	163680
11.	1 2 3 4 5 6 7 18	81840
12.	1 2 3 4 5 6 7 19	163680
13.	1 2 3 4 5 6 7 22	163680
14.	1 2 3 4 5 6 8 9	163680
15.	1 2 3 4 5 6 8 10	163680
16.	1 2 3 4 5 6 8 12	163680
17.	1 2 3 4 5 6 8 13	163680
18.	1 2 3 4 5 6 8 14	163680
19.	1 2 3 4 5 6 8 15	163680
20.	1 2 3 4 5 6 8 16	163680
21.	1 2 3 4 5 6 8 17	163680
22.	1 2 3 4 5 6 8 19	163680
23.	1 2 3 4 5 6 8 20	163680
24.	1 2 3 4 5 6 8 21	163680
25.	1 2 3 4 5 6 8 23	163680
26.	1 2 3 4 5 6 8 24	163680
27.	1 2 3 4 5 6 8 26	163680
28.	1 2 3 4 5 6 8 27	163680
29.	1 2 3 4 5 6 8 30	81840
30.	1 2 3 4 5 6 8 32	163680
31.	1 2 3 4 5 6 8 33	163680
32.	1 2 3 4 5 6 9 10	163680
33.	1 2 3 4 5 6 9 11	163680
34.	1 2 3 4 5 6 9 12	163680
35.	1 2 3 4 5 6 9 13	81840
36.	1 2 3 4 5 6 9 14	163680
37.	1 2 3 4 5 6 9 15	163680
38.	1 2 3 4 5 6 9 17	163680
39.	1 2 3 4 5 6 9 18	163680
40.	1 2 3 4 5 6 9 19	163680
41.	1 2 3 4 5 6 9 22	81840
42.	1 2 3 4 5 6 9 23	81840
43.	1 2 3 4 5 6 9 24	163680
44.	1 2 3 4 5 6 9 26	163680
45.	1 2 3 4 5 6 9 27	163680
46.	1 2 3 4 5 6 9 29	81840
47.	1 2 3 4 5 6 9 33	163680
48.	1 2 3 4 5 6 10 11	163680
49.	1 2 3 4 5 6 10 12	163680
50.	1 2 3 4 5 6 10 13	163680
51.	1 2 3 4 5 6 10 15	163680
52.	1 2 3 4 5 6 10 18	163680
53.	1 2 3 4 5 6 10 19	163680
54.	1 2 3 4 5 6 10 20	163680
55.	1 2 3 4 5 6 10 22	81840
56.	1 2 3 4 5 6 10 24	163680
57.	1 2 3 4 5 6 10 25	163680
58.	1 2 3 4 5 6 10 26	163680
59.	1 2 3 4 5 6 10 28	81840
60.	1 2 3 4 5 6 10 32	81840
61.	1 2 3 4 5 6 11 12	81840
62.	1 2 3 4 5 6 11 14	163680
63.	1 2 3 4 5 6 11 16	163680
64.	1 2 3 4 5 6 11 20	81840
65.	1 2 3 4 5 6 11 21	163680
66.	1 2 3 4 5 6 11 22	163680
67.	1 2 3 4 5 6 11 23	163680
68.	1 2 3 4 5 6 11 25	163680
69.	1 2 3 4 5 6 11 26	163680
70.	1 2 3 4 5 6 11 27	81840
71.	1 2 3 4 5 6 11 33	163680
72.	1 2 3 4 5 6 12 13	163680
73.	1 2 3 4 5 6 12 15	81840
74.	1 2 3 4 5 6 12 17	163680
75.	1 2 3 4 5 6 12 20	163680
76.	1 2 3 4 5 6 12 24	163680
77.	1 2 3 4 5 6 12 26	81840
78.	1 2 3 4 5 6 12 32	163680
79.	1 2 3 4 5 6 13 16	163680
80.	1 2 3 4 5 6 14 24	81840
81.	1 2 3 4 5 6 16 17	163680
82.	1 2 3 4 5 6 16 22	20460
83.	1 2 3 4 5 6 16 33	163680
84.	1 2 3 4 5 6 17 19	163680
85.	1 2 3 4 5 6 17 33	163680
86.	1 2 3 4 5 7 9 13	163680
87.	1 2 3 4 5 7 9 17	163680
88.	1 2 3 4 5 7 9 32	163680
89.	1 2 3 4 5 7 10 20	81840
90.	1 2 3 4 5 7 10 32	81840
91.	1 2 3 4 5 7 12 15	163680
92.	1 2 3 4 5 7 12 17	81840
93.	1 2 3 4 5 7 12 24	81840
94.	1 2 3 4 5 7 13 26	163680
95.	1 2 3 4 5 8 10 13	163680
96.	1 2 3 4 5 8 13 19	81840
97.	1 2 3 4 5 9 13 24	81840

The solution vectors have an entry 1 in the i-th place if and only if the i-th orbit on 8-subsets is part of the design. Thus, for $\lambda = 10$ we have

$$b \ = \ 27 \times 163\,680 + 11 \times 81\,840 + 1 \times 20\,460 \ = \ 5\,340\,060$$

blocks in the 7-(33, 8, 10) design. The same number of blocks can also be obtained from the following well known formula:

$$b \ = \ \frac{\binom{v}{t}}{\binom{k}{t}} \cdot \lambda \ = \ \frac{\binom{33}{7}}{\binom{8}{7}} \cdot 10 \ = \ 5\,340\,060 \ .$$

The authors thank C. Praeger for pointing out reference [12] to us. We also thank the referees for helpful suggestions for a detailed presentation of the designs.

References

1. E. Arnold: Äquivalenzklassen linearer Codes, Zulassungsarbeit Bayreuth 1993.
2. E. F. Brickell: Solving low density knapsacks. *Advances in Cryptology, Proceedings of Crypto '83*, Plenum Press, New York (1984), 25–37.
3. M. J. Coster, B. A. LaMacchia, A. M. Odlyzko, C. P. Schnorr: An improved low-density subset sum algorithm. *Proceedings EUROCRYPT '91, Brighton, May 1991* in *Springer Lecture Notes in Computer Science* **547** (1991), 54–67.
4. M. R. Garey, D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company (1979).
5. C. F. Gauss: *Disquisitiones Arithmeticae*, German edition by H. Maser, Chelsea Pub., New York (1965).
6. A. Korkine, G. Zolotareff, Sur les forms quadratiques. *Math. Ann.* **6** (1873), 366–389.
7. E. S. Kramer, D. M. Mesner: t-designs on hypergraphs. *Discrete Math.* **15** (1976), 263–296.
8. D. L. Kreher, S. P. Radziszowski: Finding Simple t-Designs by Using Basis Reduction. *Congressus Numerantium* **55** (1986), 235–244.
9. J. C. Lagarias, A. M. Odlyzko: Solving low-density subset sum problems. *J. Assoc. Comp. Mach.* **32** (1985), 229–246.
10. R. Laue: Construction of combinatorial objects – A tutorial. *Bayreuther Math. Schr.* **43** (1993), 53–96.
11. A. K. Lenstra, H. W. Lenstra Jr., L. Lovász: Factoring Polynomials with Rational Coefficients, *Math. Ann.* **261** (1982), 515–534.
12. M. W. Liebeck, C. E. Praeger, J. Saxl: The maximal factorizations of the finite simple groups and their automorphism groups, *Memoirs of the Amer. Math. Soc.* **432**(1990), Chapter 9.
13. S. Magliveras, D. W. Leavitt: Simple 6-(33, 8, 36) designs from $P\Gamma L_2(32)$. *Computational Group Theory*, M. D. Atkinson ed., Academic Press 1984, 337–352.

14. B. Schmalz: The t-designs with prescribed automorphism group, new simple 6-designs. *J. Combinatorial Designs* **1** (1993), 125–170.
15. C. P. Schnorr: A hierachy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science* **53** (1987), 201–224.
16. C. P. Schnorr: A More Efficient Algorithm for Lattice Basis Reduction. *J. Algorithms* **9** (1988), 47–62.
17. C. P. Schnorr, M. Euchner: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Proceedings of Fundamentals of Computation Theory 91* in *Lecture Notes in Computer Science* **529** (1991), 68–85.
18. D. Slepian: Some further theory of group codes. In I. F. Blake: *Algebraic Coding Theory: History and Development (Benchmark papers in electrical engineering and computer science)*, Stroudsburg, Dowden, Hutchinson & Ross Inc. (1973), 118–151.
19. L. Teirlinck: Non trivial t-designs without repeated blocks exist for all t. *Discrete Mathematics* **65** (1987), 301–311.
20. S. Weinrich: Konstruktionsalgorithmen für diskrete Strukturen und ihre Implementierung, Diplomarbeit Bayreuth (1993), 274 pp.

This article was processed using the LaTeX macro package with LLNCS style