Lower Bounds for q-ary Codes with Large Covering Radius

Wolfgang Haas

Immanuel Halupczok*

Jan-Christoph Schlage-Puchta

Albert-Ludwigs-Universität
Mathematisches Institut
Eckerstr. 1
79104 Freiburg, Germany
wolfgang_haas@gmx.net
math@karimmi.de
jcsp@cage.ugent.be

Submitted: Jan 12, 2009; Accepted: Oct 27, 2009; Published: Nov 7, 2009
Mathematics Subject Classification: 94B65

Abstract

Let $K_q(n,R)$ denote the minimal cardinality of a q-ary code of length n and covering radius R. Recently the authors gave a new proof of a classical lower bound of Rodemich on $K_q(n,n-2)$ by the use of partition matrices and their transversals. In this paper we show that, in contrast to Rodemich's original proof, the method generalizes to lower-bound $K_q(n,n-k)$ for any $k>2$. The approach is best-understood in terms of a game where a winning strategy for one of the players implies the non-existence of a code. This proves to be by far the most efficient method presently known to lower-bound $K_q(n,R)$ for large R (i.e. small k). One instance: the trivial sphere-covering bound $K_{12}(7,3) \geqslant 729$, the previously best bound $K_{12}(7,3) \geqslant 732$ and the new bound $K_{12}(7,3) \geqslant 878$.

Keywords: covering codes, lower bounds, partition matrices.

*The second author was supported by the Fondation Sciences mathématiques de Paris.

1 Introduction

Let $q \geqslant 2$ and $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. The Hamming distance $\mathrm{d}(\mathbf{x}, \mathbf{y})$ between $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_q^n$ and $\mathbf{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{Z}_q^n$ is defined by

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) = |\{i \in \{0, \ldots, n-1\} : x_i \neq y_i\}|.$$

We say $C \subset \mathbb{Z}_q^n$ is a q-ary code of length n and covering radius (at most) R, if

$$\forall \mathbf{x} \in \mathbb{Z}_q^n \ \exists \mathbf{y} \in C \quad \text{with} \quad \mathrm{d}(\mathbf{x}, \mathbf{y}) \leqslant R \tag{1}$$

is satisfied. Let $K_q(n, R)$ denote the minimal cardinality of a q-ary code of length n and covering radius R.

For a monograph on covering codes see [2]. An updated table of bounds on $K_q(n, R)$ is published online by Kéri [5]. An easy lower bound on $K_q(n, R)$ is the sphere-covering bound

$$K_q(n, R) \geqslant \frac{q^n}{V_q(n, R)}, \tag{2}$$

where

$$V_q(n, R) = |\{\mathbf{y} \in \mathbb{Z}_q^n : d(\mathbf{y}, \mathbf{x}) \leqslant R\}| = \sum_{0 \leqslant i \leqslant R} \binom{n}{i} (q-1)^i$$

denotes the cardinality of a Hamming-sphere with radius R centered on an arbitrary word $\mathbf{x} \in \mathbb{Z}_q^n$.

The following classical lower bound due to Rodemich [7] improves on the sphere-covering bound in case of $R = n - 2$:

$$K_q(n, n-2) \geqslant \frac{q^2}{n-1}. \tag{3}$$

In a previous paper the first and the third author together with Jörn Quistorff [3] presented a new proof of Rodemich's bound by the use of partition matrices and their transversals. In this paper we show that, in contrast to Rodemich's original proof, the method generalizes to lower-bound $K_q(n, n-k)$ for any $k > 2$. We present the method by considering the following game between player P (the "partition searcher") and player T (the "transversal searcher"). Player P successively offers n partitions of $\mathbb{Z}_M$ into q subsets, while player T chooses one set per partition. Player T wins if each element of $\mathbb{Z}_M$ occurs in less than k of the sets he has chosen. It turns out that if player T has a winning strategy for that game, then $K_q(n, n-k) > M$ holds true.

We make extensive computer calculations to compute winning strategies for player T with various values of n, k and q to lower-bound $K_q(n, n-k)$. Although the computing time grows rapidly with increasing k, the method yields almost 150 new lower bounds for $K_q(n, R)$ in Kéri's tables, most of them with substantial improvements.

This paper is organized as follows. In Section 2 we define partition matrices, their (generalized) transversals and the connection to covering codes. In Section 3 we consider the game from above and describe how to compute winning strategies for player T. In

Section 4 we study the computational algorithms in detail. In Section 5 we present the results of these computations.

We always assume that k, n, q, m, M are integers with $3 \leqslant k \leqslant n$, $q \geqslant 2$ and $0 \leqslant m \leqslant M$.

Added in Proof. After the submission of the article we recognized, that by a simple modification of the method we are able to produce new lower bounds not only for $K_q(n, n-k)$, but also for $K_q(n,k)$, where k is small compared to n. The details will appear elsewhere.

2 Partition Matrices and Covering Codes

The following definition generalizes the one given in [3].

Definition 1. *A $q \times n$-matrix $\mathcal{M}$, whose entries are subsets $P \subset \mathbb{Z}_M$ with $|P| \geqslant m$ is called an $(n, M, q; m)$-partition matrix, if the sets in each of the n columns of $\mathcal{M}$ form a partition of $\mathbb{Z}_M$ into q subsets. In case of $m = 0$ we simply speak of a (n, M, q)-partition matrix.*

A sequence $\mathcal{T} = (P_1, \ldots, P_d)$ of d subsets $P_i \subset \mathbb{Z}_M$, $1 \leqslant i \leqslant d$ is called a d-transversal. If $P_1, \ldots, P_d$ stem from pairwise different columns of a partition matrix $\mathcal{M}$, we speak of a d-transversal in $\mathcal{M}$.

For $z \in \mathbb{Z}_M$ and $i \geqslant 0$ we set

$$\mathrm{mult}(z) = \mathrm{mult}(z, \mathcal{T}) = |\{1 \leqslant i \leqslant d \mid z \in P_i\}|,$$

$$Z_i = Z_i(\mathcal{T}) = \{z \in \mathbb{Z}_M \mid \mathrm{mult}(z) = i\} \subset \mathbb{Z}_M,$$

$$Z_{\geqslant i} = Z_{\geqslant i}(\mathcal{T}) = \{z \in \mathbb{Z}_M \mid \mathrm{mult}(z) \geqslant i\} \subset \mathbb{Z}_M.$$

If $l \geqslant 0$, then a (d, l)-transversal is a d-transversal with $Z_{\geqslant l} = \emptyset$. For any given k, we say that a transversal $\mathcal{T}$ is of type $(a_1, \ldots, a_k; d)$ if it is a d-transversal and $|Z_{\geqslant j}(\mathcal{T})| = a_j$ holds for $1 \leqslant j \leqslant k$.

Clearly transversals of type $(a_1, \ldots, a_k; d)$ exists if and only if $M \geqslant a_1 \geqslant a_2 \geqslant \ldots \geqslant a_k \geqslant 0$ and moreover $a_i = 0$ for $i > d$. We will call such tuples $(a_1, \ldots, a_k; d)$ *admissible.*

Theorem 2. *If $R \leqslant n-2$ then the following two statements are equivalent:*

(i) *Every (n, M, q)-partition matrix has an $(n, n-R)$-transversal.*

(ii) $K_q(n, R) > M$.

Proof. (i) $\Rightarrow$ (ii): Let $C \subset \mathbb{Z}_q^n$ be a code of cardinality M. Let $\mathcal{C} = (c_{jk})$ $(j \in \mathbb{Z}_M, k \in \mathbb{Z}_n)$ be an $M \times n$-matrix whose rows are the codewords of C (in an arbitrary order). For $i \in \mathbb{Z}_q, k \in \mathbb{Z}_n$ set $P_{ik} = \{j \in \mathbb{Z}_M \mid c_{jk} = i\}$. Then $\mathcal{M} = (P_{ik})$ is an (n, M, q)-partition matrix. By assumption, it has an $(n, n-R)$-transversal $(P_{x_i,i})_{0 \leqslant i \leqslant n-1}$ with $x_i \in \mathbb{Z}_q$ $(0 \leqslant i \leqslant n-1)$. Then for every $j \in \mathbb{Z}_M$ the equation $c_{ji} = x_i$ holds for less than $n - R$ values of $i \in \mathbb{Z}_n$. Hence, C has covering radius $> R$.

(ii) $\Rightarrow$ (i): Let $\mathcal{M} = (P_{ik})$, $(i \in \mathbb{Z}_q, k \in \mathbb{Z}_n)$ be a (n, M, q)-partition matrix. For every $j \in \mathbb{Z}_M$ and every $k \in \mathbb{Z}_n$ there exists exactly one $c_{jk} := i \in \mathbb{Z}_q$ with $j \in P_{ik}$. Then $C := \{(c_{j,0}, \ldots, c_{j,n-1}) \in \mathbb{Z}_q^n \mid j \in \mathbb{Z}_M\}$ is a code of cardinality $|C| \leqslant M$, which by our assumption has covering radius $> R$. Hence, there exists a tuple $(x_0, \ldots, x_{n-1}) \in \mathbb{Z}_q^n$ such that for every $j \in \mathbb{Z}_M$ the equation $c_{ji} = x_i$ holds for less than $n - R$ values of $i \in \mathbb{Z}_n$. Consequently, $(P_{x_i,i})_{0 \leqslant i \leqslant n-1}$ is the desired $(n, n-R)$-transversal. □

Very recently Kéri [5] announced a computer-aided proof of the new bound $K_4(6, 3) \geqslant 10$, improving on the bound $K_4(6, 3) \geqslant 8$ due to Chen and Honkala [1]. As an example application of Theorem 2, we give a non-computational proof of this result.

Lemma 3. *Every* $(4, 9, 4)$*-partition matrix contains a* $(3, 2)$*-transversal.*

Proof. Let $\mathcal{M}$ be a $(4, 9, 4)$-partition matrix with the columns $0, 1, 2, 3$, whose entries are subsets $P \subset \mathbb{Z}_9$. Note that a $(3, 2)$-transversal in $\mathcal{M}$ simply is a 3-transversal in $\mathcal{M}$ consisting of pairwise disjoint sets.

A set $P \subset \mathbb{Z}_9$ in $\mathcal{M}$ with $|P| \leqslant 1$ can easily be extended to a $(3, 2)$-transversal, so without loss each column of $\mathcal{M}$ consists of three 2-(element-)sets and one 3-set.

Moreover we may assume that any two 2-sets in different columns are distinct. Otherwise without loss let $\{0, 1\}, \{0, 1\}, \{2, 3\}$ occur in the columns $0, 1, 2$. Let $R \neq \{0, 1\}$ be a set in column 1 disjoint to $\{2, 3\}$. Then $\{0, 1\}, R, \{2, 3\}$ is a $(3, 2)$-transversal.

We may assume that there is a 2-set and an intersecting 3-set, say in columns $0, 1$, since otherwise two disjoint 2-set in any two columns and the 3-set in another were a $(3, 2)$-transversal. So let $P_0 = \{0, 1\}$ occur in column 0 and $Q_1 = \{5, 6\}$ as well as $\{7, 8\}$ in column 1. Now all four sets $P_{0i}, P_{1i}, P_{2i}, P_{3i}$ in column i, $i = 2, 3$ intersect $\{0, 1, 5, 6\}$ as well as $\{0, 1, 7, 8\}$, since $\{0, 1\}$, $\{5, 6\}$ or $\{7, 8\}$ together with a counterexample would be a $(3, 2)$-transversal. Without loss assume that P_{2i} and P_{3i} do not intersect $\{0, 1\}$, $i = 2, 3$. Then P_{2i} and P_{3i} intersect $\{5, 6\}$ as well as $\{7, 8\}$, implying $\{5, 6, 7, 8\} \subset P_{2i} \cup P_{3i}$. Therefore in column i $(i = 2, 3)$ one of the two sets P_{2i} and P_{3i} must be a 2-set Q_i satisfying $Q_i \subset \{5, 6, 7, 8\}$ and one of the two sets P_{0i} and P_{1i} is a 2-set P_i intersecting $\{0, 1\}$ with $P_i \subset \{0, 1, 2, 3, 4\}$.

There exists a 2-set Q_0 in column 0 intersecting $\{5, 6, 7, 8\}$. Now $Q_0 \subset \{5, 6, 7, 8\}$ is impossible, since then at least two of the four pairwise distinct sets $Q_0, \ldots, Q_3 \subset \{5, 6, 7, 8\}$ are disjoint and would give a $(3, 2)$-transversal together with P_0, P_2 or P_3; so without loss $Q_0 = \{4, 5\}$. Then $Q_0, \{7, 8\}$ and P_2 are a $(3, 2)$-transversal except when $P_2 = \{y, 4\}$ with $y \in \{0, 1\}$. Now in the first column there exists a 2-set Q different from Q_0 and $\{0, 1\}$. Again $Q \subset \{5, 6, 7, 8\}$ is impossible. Then Q, $\{5, 6\}$ or $\{7, 8\}$, P_2 is a $(3, 2)$-transversal. □

Theorem 4 (Kéri [5])**.** $K_4(6, 3) \geqslant 10$.

Proof. By Theorem 2 it suffices to show that every $(6, 9, 4)$-partition matrix $\mathcal{M}$ has a $(6, 3)$-transversal. Let $\mathcal{M}_{i,l}$ (resp. $\mathcal{M}_{i,r}$) denote the sub-matrix of $\mathcal{M}$ consisting of the first (resp. last) i columns of $\mathcal{M}$.

(i) Without loss each set in $\mathcal{M}$ has cardinality $\geqslant 2$ and no 2-set occurs twice.

Otherwise we may assume that in $\mathcal{M}_{2,r}$ there occurs a $\leqslant 1$-set or the same 2-set twice. By Lemma 3, $\mathcal{M}_{4,l}$ has a $(3,2)$-transversal $\mathcal{T}_1$, contained in, say $\mathcal{M}_{3,l}$. Now one easily sees that $\mathcal{M}_{3,r}$ has a $(3,2)$-transversal $\mathcal{T}_2$, which together with $\mathcal{T}_1$ forms a $(6,3)$-transversal in $\mathcal{M}$.

We now know that each column of $\mathcal{M}$ consists of three 2-sets and one 3-set. We denote the 3-set in column i by Q_i, $0 \leqslant i \leqslant 5$.

(ii) Without loss $\mathcal{M}_{3,l}$ has a $(3,2)$-transversal $\mathcal{T}_1 = (P_0, P_1, P_2)$ with $P_0 \cup P_1 \cup P_2 = \mathbb{Z}_6$.

It suffices to show that there exists a $(3,2)$-transversal in $\mathcal{M}$ consisting of 2-sets. We may assume that in two different columns there exists a 2-set (say $\{0,1\}$ in column 0) and an intersecting 3-set (say in column 1), since otherwise two disjoint 2-sets in $\mathcal{M}_{2,l}$, two disjoint 2-sets in columns 2 and 3 and Q_4, Q_5 in $\mathcal{M}_{2,r}$ would form a $(6,3)$-transversal. Now there must be two 2-sets in column 1 disjoint to $\{0,1\}$, say the sets $\{2,3\}$ and $\{4,5\}$. In each of the columns $i = 2,3,4,5$ there exists a 2-set R_i disjoint to $\{0,1\}$. We may assume $R_i \subset \{2,3,4,5\}$, since otherwise $\{0,1\}$, $\{2,3\}$ or $\{4,5\}$ and R_i would be a $(3,2)$-transversal consisting of 2-sets. Now at least two of the four sets $R_2 \ldots, R_5$ are equal or disjoint, say R_2 and R_3. The first case is impossible by (i) and the latter case gives us the $(3,2)$-transversal $\{0,1\}$, R_2 and R_3.

The proof of Theorem 4 is complete, if we are able to show that there exists a $(3,3)$-transversal $\mathcal{T}_2$ in $\mathcal{M}_{3,r}$ with the additional property that

(*) if $x \in \mathbb{Z}_9$ occurs in two sets of $\mathcal{T}_2$ then $x \in \{6,7,8\}$,

since then $\mathcal{T}_1$ together with $\mathcal{T}_2$ forms the desired $(6,3)$-transversal in $\mathcal{M}$.

(iii) No set P in $\mathcal{M}_{3,r}$ satisfies $P \subset \{6,7,8\}$.

If P in $\mathcal{M}_{3,r}$ satisfies $P \subset \{6,7,8\}$, then P and any two disjoint sets from the other two columns of $\mathcal{M}_{3,r}$ form a $(3,3)$-transversal in $\mathcal{M}_{3,r}$ with the property (*) and we are done.

(iv) No element of $\{6,7,8\}$ lies in two 2-sets of $\mathcal{M}_{3,r}$.

If an element of $\{6,7,8\}$ lies in two 2-sets R_1 and R_2 of $\mathcal{M}_{3,r}$, then by (i) we have $|R_1 \cup R_2| = 3$, and R_1, R_2 and any set R disjoint to R_1 and R_2 in the remaining column of $\mathcal{M}_{3,r}$ form a $(3,3)$-transversal in $\mathcal{M}_{3,r}$ with the property (*).

Now in each of the columns i with $i = 3,4,5$ there occurs a 2-set P_i in $\mathcal{M}_{3,r}$ intersecting $\{6,7,8\}$, since otherwise Q_i must be the set $\{6,7,8\}$ itself, contradicting (iii).

(v) For each $i = 3,4,5$ we have $\{6,7,8\} \subset P_i \cup Q_i$. P_i as well as Q_i have exactly one element in common with $\mathbb{Z}_6$.

By (iii) P_i has exactly one element in common with $\mathbb{Z}_6$. By (iv) no 2-set intersecting $\{6,7,8\}$ can occur in $\mathcal{M}_{3,r}$ besides P_3, P_4 and P_5. Thus both elements of $\{6,7,8\} \setminus P_i$ lie in Q_i and (v) follows.

We now complete the proof of Theorem 4. At least two of the three sets P_3, P_4 and P_5, say P_3 and P_4, have nonempty intersection, since otherwise P_3, P_4 and P_5 are a $(3,2)$-transversal in $\mathcal{M}_{3,r}$. By (i) and (iv) $P_3 \cap P_4 = \{x\}$ with $x \in \mathbb{Z}_6$. Let R denote a set among the two sets P_5 and Q_5 not containing x. By (v) $R \cap \mathbb{Z}_6 = \{y\}$ with $x \neq y$. Again by (v) there occur two 2-sets $\subset \mathbb{Z}_6$ in column 4. Let S denote the one from this two sets not containing y. Then P_3, R and S form a $(3,3)$-transversal in $\mathcal{M}_{3,r}$ with the property (*), since S neither contains x by $P_3 \cap P_4 = \{x\}$ and $S \neq P_4$. $\square$

It appears, that Theorem 2 leads to many similar improvements for lower-bounding $K_q(n, n-3)$ with some small values of q and n by the use of individual considerations. For instance in a forthcoming paper [4] we further improve Theorem 4 to $K_4(6,3) \geqslant 11$ and settle $K_4(5,2) = 16$. In the next section we develop a mechanical approach for the use of Theorem 2 which is well suitable for computer calculations.

3 The Game

We consider a game between player P, who tries to find a certain covering code and player T, who tries to prove that no such code exists. More precisely, player P wants to show $K_q(n, n-k) \leqslant M$ and player T wants to show $K_q(n, n-k) > M$. We describe the game in the following definition.

Definition 5. *The game $G(n, M, q, k; m)$ between player P and player T goes as follows. Player P chooses a partition of $\mathbb{Z}_M$ consisting of q subsets $B_1, \ldots, B_q \subset \mathbb{Z}_M$ satisfying $|B_i| \geqslant m$ for $1 \leqslant i \leqslant q$ and player T chooses one of the sets. Then player P chooses a second partition with the same properties and again player T chooses one of the sets. This goes on until player P has chosen n such partitions and player T has chosen an n-transversal $\mathcal{T}$. Player T wins, if $\mathcal{T}$ is an (n, k)-transversal, otherwise he loses.*

In case of $m = 0$ we simply speak of the game $G(n, M, q, k)$.

If $K_q(n, n-k) \leqslant M$, then by Theorem 2 there exists an (n, M, q)-partition matrix $\mathcal{M}$ without an (n, k)-transversal. Then player P has a winning strategy for the game $G(n, M, q, k)$: he simply chooses the n partitions from the columns of $\mathcal{M}$. Thus we get:

Theorem 6. *If player T has a winning strategy for the game $G(n, M, q, k)$, then $K_q(n, n-k) > M$.*

Note that conversely $K_q(n, n-k) > M$ does not imply that player T has a winning strategy for the game $G(n, M, q, k)$. In fact, the existence of a covering code is equivalent to the statement that player P has a winning strategy which does not depend on the choices of player T.

By Theorem 6 it is our task to compute a winning strategy for player T for the game $G(n, M, q, k)$, if we want to lower-bound $K_q(n, n-k)$. We begin by the definition of winning tuples. Note that, if after d steps of the game $G(n, M, q, k; m)$ player T has chosen a d-transversal $\mathcal{T}$, the winner of the game depends only on the type $(a_1, \ldots, a_k; d)$ of $\mathcal{T}$, provided both players finish the game playing optimal.

Definition 7. *Suppose that in the game $G(n, M, q, k; m)$, after d steps $(0 \leqslant d \leqslant n)$ player T has chosen a d-transversal $\mathcal{T}$. We then call $\mathcal{T}$ a winning transversal, if player T* *has a winning strategy in this situation; otherwise we call $\mathcal{T}$ losing. An admissible tuple $(a_1, \ldots, a_{k-1}; d)$ is a winning tuple if a transversal of type $(a_1, \ldots, a_{k-1}, 0; d)$ is winning; otherwise it is a loosing tuple.*

Note: a non-admissible tuple is neither winning nor loosing. Some admissible tuples can not occur in the game $G(n, M, q, k; m)$ because the corresponding transversal would have to contain sets of cardinality less than m. Of course it is unnecessary to consider such tuples; however, for simplicity we still call them winning or losing, depending on the winner when the remainder of the game is played with sets of cardinality at least m.

The following lemma is evident.

Lemma 8. *Let a'_i, $1 \leqslant i \leqslant k-1$ be integers with $a'_i \leqslant a_i$ $(1 \leqslant i \leqslant k-1)$, such that $(a'_1, \ldots, a'_{k-1}; d)$ is admissible. If $(a_1, \ldots, a_{k-1}; d)$ is a winning tuple for the game $G(n, M, q, k; m)$, then so is $(a'_1, \ldots, a'_{k-1}; d)$.*

The idea is now to recursively determine the winning tuples for the game $G(n, M, q, k)$ with decreasing d by starting with $d = n$; if $(0, \ldots, 0; 0)$ turns out to be a winning tuple, then player T has a winning strategy for the game $G(n, M, q, k)$ and $K_q(n, n-k) > M$ follows by Theorem 6. However, we can do better. It may happen that $(0, \ldots, 0; 0)$ is not a winning tuple, but $(a_1, 0, \ldots, 0; 1)$ is winning for some $a_1 \geqslant 0$ (i.e. player T can win the game if the first partition contains a set with at most a_1 elements by choosing that set). In that case, we may still be able to prove $K_q(n, n-k) > M$ using the following theorem, as winning the game $G(n, M, q, k; m)$ for $m > 0$ is often easier than winning the game $G(n, M, q, k)$.

Theorem 9. *Suppose that $0 = m_0 < m_1 < \cdots < m_l$ are integers such that $(m_{i+1} - 1, 0, \ldots, 0; 1)$ is a winning tuple for the game $G(n, M, q, k; m_i)$ for $0 \leqslant i \leqslant l-1$ and such that player T has a winning strategy for the game $G(n, M, q, k; m_l)$. Then $K_q(n, n-k) > M$ holds true.*

Proof. By Theorem 2 it suffices to show that under the assumptions of the theorem, every (n, M, q)-partition matrix $\mathcal{M}$ has an (n, k)-transversal, so suppose that $\mathcal{M}$ is given. Denote by m the cardinality of the smallest set P_0 occuring in $\mathcal{M}$, and let i be maximal with $m_i \leqslant m$. Without loss, we may suppose that P_0 occurs in the first column of $\mathcal{M}$.

We play the game $G(n, M, q, k; m_i)$, and we let player P choose the n columns of $\mathcal{M}$; as $m_i \leqslant m$, he is allowed to do so. Suppose first $i = l$. Then by assumption player T has a winning strategy, and thus there exists an (n, k)-transversal in $\mathcal{M}$. Now suppose $i < l$. Then player T can choose P_0 in the first step of the game. Since $m \leqslant m_{i+1} - 1$, by assumption and Lemma 8 we get that $(m, 0, \ldots, 0; 1)$ is a winning tuple, so (P_0) is a winning 1-transversal, and again there exists an (n, k)-transversal in $\mathcal{M}$. $\square$

So the enhanced strategy to prove $K_q(n, n-k) > M$ is now the following. Determine the winning tuples for the game $G(n, M, q, k)$. If player T has a winning strategy, we are done. Otherwise, check whether there exists a winning tuple of the form $(a_1, 0, \ldots, 0; 1)$. If not, give up. If yes, then repeat the procedure for the game $G(n, M, q, k; a_1 + 1)$. If we obtain a winning tuple $(a_2, 0, \ldots, 0; 1)$ with $a_2 > a_1$, then repeat again; go on until either

$a_{i+1} = a_i$ (in that case, give up) or player T has a winning strategy for one of the games (then Theorem 9 implies $K_q(n, n-k) > M$).

In the remainder of this section, we show how winning tuples $(a_1, \ldots, a_{k-1}; d)$ can be determined manually. Clearly, for $d = n$ any tuple is a winning tuple. For $d \geqslant n-2$, we will get explicit formulas (Lemmas 10 and 11), and for smaller d, we will prove the general Lemma 12, which is essentially a method of checking every possible partition, but adapted to manual computations. It can be applied with a reasonable amount of work when $k = 3$ and n and q are not too big (say, $n, q \leqslant 12$). As an example how this works in practice, we will prove Theorem 13.

Lemma 10. *If $a_{k-1} \leqslant q-1$, then $(a_1, \ldots, a_{k-1}; n-1)$ is a winning tuple for the game $G(n, M, q, k; m)$, provided it is admissible.*

Proof. Assume that in the game $G(n, M, q, k; m)$, after $n-1$ steps player T has chosen a transversal $\mathcal{T}$ of type $(a_1, \ldots, a_{k-1}, 0; n-1)$ with $a_{k-1} \leqslant q-1$ and player P has chosen the partition $B_1, \ldots, B_q$ of $\mathbb{Z}_M$ in step n. Then as $|Z_{\geqslant k-1}| \leqslant q-1$, at least one of the sets $B_1, \ldots, B_q$ does not intersect $Z_{\geqslant k-1}$ and this set can be taken by player T to complete $\mathcal{T}$ to an (n, k)-transversal. $\square$

Lemma 11. *Assume $r := q - a_{k-1} > 0$. If $a_{k-2} \leqslant q + r^2 - r - 1$ or $M \leqslant (q-r)m + r^2 - 1$, then $(a_1, \ldots, a_{k-1}; n-2)$ is a winning tuple for the game $G(n, M, q, k; m)$, provided it is admissible.*

Proof. Assume that in the game $G(n, M, q, k; m)$, player T has chosen a transversal $\mathcal{T}$ of type $(a_1, \ldots, a_{k-2}, q-r, 0; n-2)$ after $n-2$ steps and player P has chosen the partition $B_1, \ldots, B_q$ of $\mathbb{Z}_M$ in step $n-1$. By $|Z_{\geqslant k-1}| = q - r$, at least $r = q - |Z_{\geqslant k-1}|$ sets, say $B_1, \ldots, B_r$, do not intersect $\mathbb{Z}_{\geqslant k-1}$ and thus all extend $\mathcal{T}$ to an $(n-1, k)$-transversal.

If $a_{k-2} \leqslant q + r^2 - r - 1$, then we have $|Z_{k-2} \cap (B_1 \cup \ldots \cup B_r)| \leqslant |Z_{k-2}| = |Z_{\geqslant k-2}| - |Z_{\geqslant k-1}| \leqslant q + r^2 - r - 1 - (q-r) = r^2 - 1 < r^2$. This implies that at least one of the sets $B_1, \ldots, B_r$, say B_1, intersects Z_{k-2} in at most $r-1$ elements, which means that B_1 extends $\mathcal{T}$ to an $(n-1, k)$-transversal $\mathcal{T}'$ satisfying $|Z_{\geqslant k-1}(\mathcal{T}')| = |Z_{k-1}(\mathcal{T})| + |Z_{k-2}(\mathcal{T}) \cap B_1| \leqslant (q-r) + (r-1) = q-1$. Thus the (admissible) type of $\mathcal{T}'$ satisfies the prerequisites of Lemma 10, and thus $\mathcal{T}'$ and also $\mathcal{T}$ are winning transversals. If $M \leqslant (q-r)m + r^2 - 1$, then

$$|Z_{k-2} \cap (B_1 \cup \ldots \cup B_r)| \leqslant M - |B_{r+1} \cup \ldots \cup B_q| \leqslant M - (q-r)m \leqslant r^2 - 1$$

and the lemma follows again. $\square$

We now show how the winning tuples of step $d+1$ can be used to determine the winning tuples of step d. The statement of the lemma is long and complicated, but it will become clear in the proof.

Lemma 12. *Suppose that $l := q - a_{k-1} > 0$ and that for any integers $b_{ij} \geqslant 0$ $(1 \leqslant i \leqslant l$, $1 \leqslant j \leqslant k-2)$ satisfying*

$$\sum_{1 \leqslant i \leqslant l} b_{ij} \leqslant a_j - a_{j+1} \qquad (1 \leqslant j \leqslant k-2) \tag{4}$$

there exist integers $b_i \geqslant 0$ $(1 \leqslant i \leqslant l)$ with the properties

(i) *if $1 \leqslant i \leqslant l$ then $(a_1 + b_i, a_2 + b_{i1}, \ldots, a_{k-1} + b_{i,k-2}; d+1)$ is a winning tuple for the game $G(n, M, q, k; m)$,*

(ii) $\sum_{1\leqslant i\leqslant l} b_i + \sum_{1\leqslant i\leqslant l} \sum_{1\leqslant j\leqslant k-2} b_{ij} + \max\{1, m\} \cdot a_{k-1} + l > M$.

Then $(a_1, \ldots, a_{k-1}; d)$ is a winning tuple for the game $G(n, M, q, k; m)$.

Proof. Assume in the game $G(n, M, q, k; m)$ player T has chosen a transversal $\mathcal{T}$ of type $(a_1, \ldots, a_{k-1}, 0; d)$ after d steps and player P the partition $B_1, \ldots, B_q$ of $\mathbb{Z}_M$ in step $d+1$. We have

$$|B_i| \geqslant m \qquad (1 \leqslant i \leqslant q), \tag{5}$$

since we are playing the game $G(n, M, q, k; m)$. Without loss suppose that $B_1, \ldots, B_l$ do not intersect $Z_{\geqslant k-1}$; this is possible as $|Z_{\geqslant k-1}| = a_{k-1} = q - l$. Moreover, if some of the sets B_i are empty, then suppose that B_1 is one of the empty sets. For $1 \leqslant i \leqslant l$, $0 \leqslant j \leqslant k-2$ we set

$$b_{ij} = |Z_j \cap B_i|.$$

Clearly for $1 \leqslant j \leqslant k-2$ we have $\sum_{1\leqslant i\leqslant l} b_{ij} \leqslant |Z_j| = |Z_{\geqslant j} \setminus Z_{\geqslant j+1}| = a_j - a_{j+1}$, so that (4) is satisfied. By the prerequisites of Lemma 12 there exist integers $b_i \geqslant 0$ for $1 \leqslant i \leqslant l$ satisfying (i) and (ii). We now verify that there is an integer i_0 with $1 \leqslant i_0 \leqslant l$ and

$$|B_{i_0}| \leqslant b_{i_0} + \sum_{1\leqslant j\leqslant k-2} b_{i_0 j}. \tag{6}$$

If there exist empty sets B_i, then B_1 is empty and $i_0 = 1$ does the job. Otherwise, suppose that (6) is false. Then for $1 \leqslant i \leqslant l$ we have $|B_i| \geqslant b_i + 1 + \sum_{1\leqslant j\leqslant k-2} b_{ij}$. For $i > l$, we use (5) to get $|B_i| \geqslant \max\{1, m\}$; hence

$$M = \sum_{1\leqslant i\leqslant l} |B_i| + \sum_{l+1\leqslant i\leqslant q} |B_i| \geqslant \sum_{1\leqslant i\leqslant l} b_i + l + \sum_{1\leqslant i\leqslant l} \sum_{1\leqslant j\leqslant k-2} b_{ij} + \max\{1, m\} \cdot a_{k-1}$$

contradicting (ii).

Now in step $d+1$ of the game $G(n, M, q, k; m)$ let player T choose the set B_{i_0}. Consider the $(d+1)$-transversal $\mathcal{T}' = (\mathcal{T}, B_{i_0})$. Since B_{i_0} does not intersect $Z_{\geqslant k-1}(\mathcal{T})$, we know that $\mathcal{T}'$ is a $(d+1, k)$-transversal. Furthermore $B_{i_0} = \bigcup_{0\leqslant j\leqslant k-2} (Z_j(\mathcal{T}) \cap B_{i_0})$, implying $|B_{i_0}| = \sum_{0\leqslant j\leqslant k-2} b_{i_0 j}$ and thus

$$b_{i_0,0} \leqslant b_{i_0} \tag{7}$$

by (6). Since for $j \geqslant 1$ we have $Z_{\geqslant j}(\mathcal{T}') = Z_{\geqslant j}(\mathcal{T}) \cup (Z_{j-1}(\mathcal{T}) \cap B_{i_0})$, we find $|Z_{\geqslant j}(\mathcal{T}')| = a_j + b_{i_0,j-1}$ $(1 \leqslant j \leqslant k-1)$. Therefore $\mathcal{T}'$ is of the admissible type $(a_1 + b_{i_0,0}, a_2 + b_{i_0,1}, \ldots, a_{k-1} + b_{i_0,k-2}, 0; d+1)$, which by (i), (7) and Lemma 8 is a winning tuple. Therefore $\mathcal{T}'$ and, consequently $\mathcal{T}$ are winning transversals, implying that $(a_1, a_2, \ldots, a_{k-1}; d)$ is a winning tuple and Lemma 12 follows. $\square$

Using Lemmas 10, 11 and 12, we obtain winning tuples for smaller and smaller d, until we (hopefully) prove that player T wins the game, or at least that $(a_1, 0, \dots, 0; 1)$ is a winning tuple for some a_1. As an example let us use the previous lemmas to improve on the presently best bound $K_9(7, 4) \geqslant 35$ due to Lang, Quistorff and Schneider [6].

Theorem 13. $K_9(7, 4) \geqslant 45$.

Proof. By Theorem 6 it suffices to show that player T has a winning strategy for the game $G(7, 44, 9, 3)$. By Lemma 10 and 11 we find the winning tuples $(44, 8; 6)$ and $(44, 2; 5)$, $(38, 3; 5)$, $(28, 4; 5)$, $(20, 5; 5)$, $(14, 6; 5)$, $(10, 7; 5)$, $(8, 8; 5)$. We next want to show that $(7, 5; 4)$ is also a winning tuple.

We apply Lemma 12 with $n = 7$, $M = 44$, $q = 9$, $k = 3$, $m = 0$, $a_1 = 7$, $a_2 = 5$ and $d = 4$; we have $l = q - a_2 = 4$. Assume that the integers $b_{i1} \geqslant 0$ $(1 \leqslant i \leqslant 4)$ satisfy (4), i.e. $\sum_{1 \leqslant i \leqslant 4} b_{i1} \leqslant 2$. For $1 \leqslant i \leqslant 4$, we choose the integers $b_i \geqslant 0$ as a function of b_{i1} according to the following table:

b_{i1}	0	1	2
b_i	13	7	3

Clearly (i) from Lemma 12 is satisfied by the winning tuples stated above. To verify (ii) from Lemma 12, it suffices to consider the case when the sum $b := \sum_{1 \leqslant i \leqslant 4} b_i + \sum_{1 \leqslant i \leqslant 4} b_{i1}$ is minimal. As the sequence $13, 7, 3$ is convex, b is minimal if $b_1, \dots, b_4$ are as equal as possible, i.e. if two of them equal one and two values equals zero. Then $b = 42$ and (ii) from Lemma 12 is satisfied, too. An application of Lemma 12 now yields that indeed $(7, 5; 4)$ is a winning tuple for the game $G(7, 44, 9, 3)$. In a similar way we proceed to get the following winning tuples

$$
\begin{array}{l}
(33, 0; 4)(26, 1; 4)(20, 2; 4)(15, 3; 4)(11, 4; 4)(7, 5; 4) \\
(18, 0; 3)(14, 1; 3)(10, 2; 3)(7, 3; 3) \\
(10, 0; 2)(7, 1; 2)(4, 2; 2) \\
(4, 0; 1)(2, 1; 1).
\end{array}
$$

In the final stage we see that $(4, 0; 1)$ is a winning tuple. Since by $44/9 < 5$ player T may choose a transversal of type $(a, 0; 1)$ with $a \leqslant 4$ in step 1. This means that indeed player T has a winning strategy for the game $G(7, 44, 9, 3)$. $\square$

The bound $K_9(7, 4) \geqslant 45$ may be further improved to $K_9(7, 4) \geqslant 51$ (see the tables in Section 5) with the help of Theorem 9. In the same way as above one shows that $(3, 0; 1)$ is a winning tuple for the game $G(7, 50, 9, 3; 0)$, that $(4, 0; 1)$ is a winning tuple for $G(7, 50, 9, 3; 4)$ and that $(5, 0; 1)$ is a winning tuple for $G(7, 50, 9, 3; 5)$. Since $50/9 < 6$ we see that player T has a winning strategy for $G(7, 50, 9, 3; 5)$ and the bound $K_9(7, 4) \geqslant 51$ follows from Theorem 9.

4 The implementation

To prove a statement of the form $K_q(n, n-k) > M$ using a computer, we use the strategy described after Theorem 9. We now describe in detail the algorithm to compute all winning tuples of a game $G(n, M, q, k; m)$. It will be slightly more handy to work with losing tuples than with winning tuples (Lemma 16 would be more complicated otherwise). Given the set of losing tuples for step $d+1$, we have to determine the set of losing tuples for step d. In principle, this means that we iterate through all admissible tuples $(a_1, \ldots, a_{k-1}; d)$ and check whether player P can find a partition, such that no matter which of its sets player T chooses, he gets a losing $(d+1)$-transversal.

Let us now make this precise. Fix an admissible tuple $(a_1, \ldots, a_{k-1}; d)$ and a corresponding transversal $\mathcal{T}_d = (P_1, \ldots, P_d)$ of type $(a_1, \ldots, a_{k-1}, 0; d)$.

Definition 14. *A subset $B \subset \mathbb{Z}_M$ is losing if $|B| \geqslant m$, and when added to $\mathcal{T}_d$, it yields a losing $(d+1)$-transversal. We call a partition of a subset $B \subset \mathbb{Z}_M$ losing if it consists only of losing sets.*

Clearly the tuple $(a_1, \ldots, a_{k-1}; d)$ is losing iff there is a losing partition of $\mathbb{Z}_M$ consisting of q sets. If $B \subset \mathbb{Z}_M$, then for $0 \leqslant j \leqslant k-1$ we set

$$c_j = |Z_j(\mathcal{T}_d) \cap B|. \tag{8}$$

Adding a set B to $\mathcal{T}_d$ yields a transversal of type $(a_1 + c_0, \ldots, a_{k-1} + c_{k-2}, c_{k-1}; d+1)$. Thus the following lemma holds true.

Lemma 15. *B is losing if $|B| \geqslant m$ and if moreover $c_{k-1} > 0$ or $(a_1 + c_0, \ldots, a_{k-1} + c_{k-2}; d+1)$ is a losing tuple.*

Set $a_0 := M$, $a_k := 0$ and $\tilde{c}_j := Z_j(\mathcal{T}_d) = a_j - a_{j+1}$ for $0 \leqslant j \leqslant k-1$. Then the integers c_j $(0 \leqslant j \leqslant k-1)$ defined in (8) satisfy $0 \leqslant c_j \leqslant \tilde{c}_j$. We say that the tuple $[c_0, \ldots, c_{k-1}; r]$ $(1 \leqslant r \leqslant q)$ is losing if there is a losing partition $B_1 \cup \ldots \cup B_r$ of B. Our goal is then to determine whether $[\tilde{c}_0, \ldots, \tilde{c}_{k-1}; q]$ is losing, since this just means that $(a_1, \ldots, a_{k-1}; d)$ is losing.

For $r = 1$, the tuple $[c_0, \ldots, c_{k-1}; 1]$ is losing if and only if a set $B \subset \mathbb{Z}_M$ satisfying (8) is losing. We then proceed by recursion on r. The following lemma is evident.

Lemma 16. *Suppose $r > 1$ and $r', r'' \geqslant 1$ with $r' + r'' = r$. Then the tuple $[c_0, \ldots, c_{k-1}; r]$ is losing if and only if there exist non-negative integers c'_j, c''_j $(0 \leqslant j \leqslant k-1)$ satisfying $c'_j + c''_j = c_j$, such that $[c'_0, \ldots, c'_{k-1}; r']$ and $[c''_0, \ldots, c''_{k-1}; r'']$ are losing.*

Using this lemma we can determine whether $[c_0, \ldots, c_{k-1}; r]$ is a losing tuple by first choosing a fixed value r' (with $1 \leqslant r' < r$) and then iterating through all tuples $(c'_j)_{0 \leqslant j \leqslant k-1}$ with $0 \leqslant c'_j \leqslant c_j$. Finally we find out whether $[\tilde{c}_0, \ldots, \tilde{c}_{k-1}; q]$, and thus $(a_1, \ldots, a_{k-1}; d)$ is losing.

Now let us take a break and compute the running time of the algorithm described so far. A game consists of n steps. In each step, we have to check roughly M^{k-1} tuples $(a_1, \ldots, a_{k-1}; d)$. To compute the losing tuples for partial partitions with r sets out of those with r' and r'' sets $(r' + r'' = r)$, we have to run through M^k c'_i-tuples. By choosing r' carefully, roughly $\log q$ of these steps are necessary. Thus the total running time (for one game) is $n \cdot M^{2k-1} \cdot \log q$, which is a lot. Moreover, the memory needed to store all losing tuples is roughly M^k, which is a lot, too. We will now describe some significant improvements; however the precise running time will be difficult to determine. For one of the improvements, we need the following lemma.

Lemma 17. *1. Suppose $(a_1, \ldots, a_{k-1}; d)$ is a losing tuple and $1 \leqslant \mu < \nu \leqslant k-1$. For $1 \leqslant j \leqslant k-1$ define $a'_j = a_j - \delta_{j\mu} + \delta_{j\nu}$ (where δ_{ij} is the Dirac function). Then $(a'_1, \ldots, a'_{k-1}; d)$ is losing, too (provided it is admissible).*

2. Fix a tuple $(a_1, \ldots, a_{k-1}; d)$ and suppose that a corresponding tuple $[c_0, \ldots, c_{k-1}; r]$ is losing. Let $0 \leqslant \mu < \nu \leqslant k-1$. Define $c'_j = c_j - \delta_{j\mu} + \delta_{j\nu}$ and suppose that for $0 \leqslant j \leqslant k-1$ we still have $0 \leqslant c'_j \leqslant \tilde{c}_j$. Then $[c'_0, \ldots, c'_{k-1}; r]$ is losing, too.

In other words, if $\nu > \mu$ then augmenting a_ν is worse than augmenting a_μ, and augmenting c_ν is worse than augmenting c_μ.

Proof. (1) We use induction over d, starting at $d = n$. For $d = n$, this is clear, since there are no losing tuples $(a_1, \ldots, a_{k-1}; n)$. Now let $1 \leqslant d \leqslant n-1$ and suppose that the lemma is true for $d+1$; we want to prove it for d.

Fix any transversal $\mathcal{T}_d$ of type $(a_1, \ldots, a_{k-1}, 0; d)$. By $a'_\mu \geqslant a'_{\mu+1}$ we have $a_\mu > a_{\mu+1}$. Therefore there exists $x \in \mathbb{Z}_M$ satisfying $\mathrm{mult}(x, \mathcal{T}_d) = \mu$. Similarly, $a_{\nu-1} > a_\nu$ implies the existence of $y \in \mathbb{Z}_M$ satisfying $\mathrm{mult}(y, \mathcal{T}_d) = \nu - 1$.

Note that we may assume $x \neq y$. This is clear if $\mu < \nu - 1$. If $\mu = \nu - 1$, this follows from $a_\mu \geqslant a_\nu + 2$ by $a'_\mu \geqslant a'_\nu$.

We now build another d-transversal $\mathcal{T}'_d$ from $\mathcal{T}_d$ by deleting x from a set P from $\mathcal{T}_d$ with $x \in P$ and adding y to a set P' from $\mathcal{T}_d$ with $y \notin P'$. Then clearly $\mathcal{T}'_d$ satisfies $\mathrm{mult}(x, \mathcal{T}'_d) = \mu - 1$, $\mathrm{mult}(y, \mathcal{T}'_d) = \nu$ and $\mathrm{mult}(z, \mathcal{T}'_d) = \mathrm{mult}(z, \mathcal{T}_d)$ whenever $z \in \mathbb{Z}_M \setminus \{x, y\}$. Therefore $\mathcal{T}'_d$ is of type $(a'_1, \ldots, a'_{k-1}, 0; d)$ and it remains to show that $\mathcal{T}'_d$ is a losing transversal. To prove this, it suffices to check that whenever $B \subset \mathbb{Z}_M$ is a set, such that the $(d+1)$-transversal $\mathcal{T}_{d+1} := (\mathcal{T}_d, B)$ is losing, then $\mathcal{T}'_{d+1} := (\mathcal{T}'_d, B)$ loses, too, since then a losing partition of $\mathbb{Z}_M$ for $\mathcal{T}_d$ is also a losing partition for $\mathcal{T}'_d$.

Now for $z \in \mathbb{Z}_M \setminus \{x, y\}$ we still have $\mathrm{mult}(z, \mathcal{T}'_{d+1}) = \mathrm{mult}(z, \mathcal{T}_{d+1})$. Moreover we have $\mathrm{mult}(x, \mathcal{T}_{d+1}) = \mu$, $\mathrm{mult}(x, \mathcal{T}'_{d+1}) = \mu - 1$ if $x \notin B$ and $\mathrm{mult}(x, \mathcal{T}_{d+1}) = \mu + 1$, $\mathrm{mult}(x, \mathcal{T}'_{d+1}) = \mu$ if $x \in B$. Similarly we have $\mathrm{mult}(y, \mathcal{T}_{d+1}) = \nu - 1$, $\mathrm{mult}(y, \mathcal{T}'_{d+1}) = \nu$ if $y \notin B$ and $\mathrm{mult}(y, \mathcal{T}_{d+1}) = \nu$, $\mathrm{mult}(y, \mathcal{T}'_{d+1}) = \nu + 1$ if $y \in B$.

Now one easily sees that the tuples $(\ldots; d+1)$ describing $\mathcal{T}_{d+1}$ and $\mathcal{T}'_{d+1}$ again satisfy the prerequisites of the lemma, with μ and (or) ν possibly increased by 1, unless the new values of μ and ν become equal, or the new value of ν is equal to k. In the first case, the tuples describing $\mathcal{T}_{d+1}$ and $\mathcal{T}'_{d+1}$ are the same; in the second case $\mathrm{mult}(y, \mathcal{T}'_{d+1}) = k$. In all cases indeed $\mathcal{T}'_{d+1}$ is losing (using the induction hypothesis).

(2) Let $\mathcal{T}_d$ be a transversal of type $(a_1, \ldots, a_{k-1}, 0; d)$ and let $B \subset \mathbb{Z}_M$ be a subset corresponding to $[c_0, \ldots, c_{k-1}; r]$, i.e. $|Z_j(\mathcal{T}_d) \cap B| = c_j$ for $0 \leqslant j \leqslant k-1$. By $0 \leqslant c'_\mu = c_\mu - 1$ we get $c_\mu \geqslant 1$, implying $Z_\mu(\mathcal{T}_d) \cap B \neq \emptyset$. Thus there exists $x \in B$ with $\mathrm{mult}(x, \mathcal{T}_d) = \mu$. Similarly $c'_\nu \leqslant a_\nu - a_{\nu+1}$ implies $c_\nu + 1 \leqslant a_\nu - a_{\nu+1}$ and thus $c_\nu = |Z_\nu(\mathcal{T}_d) \cap B| < a_\nu - a_{\nu+1}$. Therefore there exists $y \in \mathbb{Z}_M \setminus B$ with $\mathrm{mult}(y, \mathcal{T}_d) = \nu$. Then $B' = (B \setminus \{x\}) \cup \{y\} \subset \mathbb{Z}_M$

corresponds to $[c'_0, \ldots, c'_{k-1}; r]$ and we have to show that there exists a partition of B' consisting of r losing sets.

We claim that a losing partition of $B = B_1 \cup \ldots \cup B_r$ yields a losing partition of B' by replacing x by y. For the sets not containing x, this doesn't change anything. If, say $x \in B_1 \subset B$ and $B'_1 := (B_1 \setminus \{x\}) \cup \{y\} \subset B'$, then either the tuples describing $(\mathcal{T}_d, B_1)$ and $(\mathcal{T}_d, B'_1)$ satisfy the prerequisite of part (1), or $\nu = k-1$ and $\operatorname{mult}(y, (\mathcal{T}_d, B'_1)) = k$. In both cases we find that $(\mathcal{T}_d, B'_1)$ is a losing transversal, hence indeed B'_1 is a losing set. $\square$

Now let us get back to the algorithm. First of all, it is not necessary to iterate through all tuples $(a_1, \ldots, a_{k-1}; d)$: increasing an a_i only makes the tuple more likely to be a losing one (see Lemma 8), so it suffices to walk along the hypersurface between losing and winning. Let us make this precise.

Suppose that $a_{i+1}, \ldots, a_{k-1}$ are fixed. Then we determine the tuples $(a_1, \ldots, a_i)$ such that $(a_1, \ldots, a_{k-1}; d)$ loses in the following way. Denote by $A := A(a_{i+1}, \ldots, a_{k-1})$ the smallest integer such that any $a_1, \ldots, a_i$ with $a_i \geqslant A$ loses. Equivalently by Lemma 8, A is the smallest integer such that $(A, \ldots, A, a_{i+1}, \ldots, a_{k-1}; d)$ loses, so A can be determined quickly using binary search or something similar. After that, it remains to iterate through all $a_i < A$ and apply the same algorithm recursively with $a_i, \ldots, a_{k-1}$ fixed. This method is also used to save memory: for fixed $a_{i+1}, \ldots, a_{k-1}$, store the corresponding value $A = A(a_{i+1}, \ldots, a_{k-1})$, and store further details concerning $a_1, \ldots, a_i$ only if $a_i < A$.

It turned out (experimentally) that for any fixed $a_{i+1}, \ldots, a_{k-1}$, almost all losing $a_1, \ldots, a_i$ satisfy $a_i \geqslant A$, so the above method yields a huge gain in efficiency. We can gain some more time by using the observation that usually, $A(a_{i+1}, \ldots, a_{k-1})$ is not much bigger than $A(a_{i+1}+1, \ldots, a_{k-1})$: use $A(a_{i+1}+1, \ldots, a_{k-1})$ as a first estimate to find $A(a_{i+1}, \ldots, a_{k-1})$ more quickly. (This means that when iterating through values for a_{i+1}, we have to start from the big values.)

Concerning the losing tuples $[c_0, \ldots, c_{k-1}; r]$, a similar strategy can be used. For each given partial tuple $c_i, \ldots, c_{k-1}$ let $C := C(c_i, \ldots, c_{k-1}; r)$ be the smallest integer such that whenever $c_{i-1} \geqslant C$, the total tuple loses; equivalently, C is the smallest integer such that the tuple $[0, \ldots, 0, C, c_i, \ldots, c_{k-1}; r]$ loses. By Lemma 16, this can be computed using the formula

$$C(c_i, \ldots, c_{k-1}; r) = \min_{\substack{(c'_j)_{j \geqslant i}, (c''_j)_{j \geqslant i} \\ c'_j + c''_j = c_j}} \left(C(c'_i, \ldots, c'_{k-1}; r') + C(c''_i, \ldots, c''_{k-1}; r'') \right), \tag{*}$$

where $r' + r'' = r$. Now whenever we are interested in such a value $C(c_i, \ldots, c_{k-1}; r)$, before we apply (*) to compute it, we first compute $C(c_{i+1}, \ldots, c_{k-1}; r)$ (which takes much less time); if this turns out to be less or equal to c_i, then $C(c_i, \ldots, c_{k-1}; r) = 0$ anyway. To determine whether $[\tilde{c}_0, \ldots, \tilde{c}_{k-1}; q]$ is losing (the only thing we are really interested in), we compute $C(\tilde{c}_1, \ldots, \tilde{c}_{k-1}; q)$ in the above way and compare it with $\tilde{c}_0$.

Many values $C(c_i, \ldots, c_{k-1}; r)$ will be needed several times in the computation, so each time we compute such a value, we store it for later re-use.

When computing (*), we may save a lot of time by computing $C(c''_i, \ldots, c''_{k-1}; r'')$ only when this is really necessary. Suppose that we already know that the minimum will be at most C_0, that we now want to treat the pair of tuples $(c'_i)_i$, $(c''_i)_i$, and that somehow we got an estimate $C(c''_i, \ldots, c''_{k-1}; r'') \geqslant e$. We first compute $C(c'_i, \ldots, c'_{k-1}; r')$; if it turns out that $C(c'_i, \ldots, c'_{k-1}; r') + e \geqslant C_0$, then we skip the computation of $C(c''_i, \ldots, c''_{k-1}; r'')$. Of course, we might always use the trivial estimate $e = 0$, but we can do better.

We iterate through the tuples $(c''_j)_{j \geqslant i}$ in an order such that tuples with larger entries come first. Fix $\mu \geqslant i$ and consider the tuple $(d''_j)_{j \geqslant i}$, where $d''_j = c''_j + \delta_{j\mu}$. We get a first estimate from the obvious inequality $C(c''_i, \ldots, c''_{k-1}; r'') \geqslant C(d''_i, \ldots, d''_{k-1}; r'')$. If $C(c''_i, \ldots, c''_{k-1}; r'') > 0$, then indeed $C(c''_i, \ldots, c''_{k-1}; r'') \geqslant C(d''_i, \ldots, d''_{k-1}; r'') + 1$ by Lemma 17. Of course we do not know whether $C(c''_i, \ldots, c''_{k-1}; r'') > 0$ or not as we didn't compute it yet but $C(d''_i, \ldots, d''_{k-1}; r'') > 0$ is a sufficient condition for $C(c''_i, \ldots, c''_{k-1}; r'') > 0$. These are the estimates which we use. (When $m > 0$, it happens very often that we have exactly $C(c'_i, \ldots, c'_{k-1}; r') + C(d''_i, \ldots, d''_{k-1}; r'') + 1 = C_0$; therefore, the additional "+1" obtained using Lemma 17 reduces computation time extremly.)

Finally, there is one thing to note concerning the decompositions $r = r' + r''$. Of course we may choose and fix those decompositions once and for all in the beginning in an optimal way. However, due to the fact that all $C(c'_i, \ldots, c'_{k-1}; r')$ are computed but not all $C(c''_i, \ldots, c''_{k-1}; r'')$, it is sometimes preferable to keep r' small and instead need a few more steps. One could probably write an algorithm which chooses a good decomposition sequence, but as anyway the maximal number q of sets in a partition we are considering is about 20, we just chose some (experimentally) good decompositions by hand.

The program implementing the above algorithm (in C++) can be obtained from the authors. Maybe it could be further improved in the following way. In the manual computations in the example $K_9(7, 4) > 44$, we significantly reduced the work by exploiting the convexity of the sequence b_i. In general, this sequence does not need to be convex, but it seems to be close to convex most of the time. If this could be exploited by the computer, this should yield a big gain of running time.

5 The results

In this section we list the results we obtained by applying our algorithm. All values in the tables below are the best ones one can obtain by using the strategy given after Theorem 9. The computations were performed on a cluster of standard PC's. The range was limited by both time and memory. In the simplest case $k = 3$ a very short program suffices to calculate the new bounds for the usually tabulated range $q \leqslant 21$ in total time of roughly a minute. In case of $k > 3$ we had to implement the ideas mentioned in Section 4 and we were restricted to $q \leqslant 16$ in the case $k = 4$ and to $q \leqslant 9$ in the case $k = 5$. For $k > 5$ the computational effort became too large for a systematical treatment. Here we got only a few sporadic new bounds: six bounds for $k = 6$, two bounds for $k = 7$ and still a new bound for a ternary code in the case $k = 8$. In the most difficult case $K_9(10, 5)$, checking a single game took about one week, and it takes about four games (with increasing m) to verify $K_q(n, R) > M$ when M is close to the best we can do. In several cases the memory used almost reached the physical limit of 1 GB.

The results of the computations are compiled in the following tables. We included the sphere-covering bound (2) and the upper bound for comparison as well as a column for k, since this is the most important parameter for the computation time.

As can be seen from the tables, the bigger q and n are, the better our approach works. If n is big but q is small, then the values of $K_q(n, n-k)$ are known for small k, so our method doesn't help here, as for large k, it is too slow.

Table 1. $q = 3$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_3(10,4)$	6	14	16	17	36
$K_3(11,4)$	7	26	28	30	81
$K_3(11,5)$	6	9	10	11	27
$K_3(13,5)$	8	29	32	33	108
$K_3(13,6)$	7	10	11	13	36

Table 2. $q = 4$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_4(7,2)$	5	78	80	84	128
$K_4(7,3)$	4	15	16	19	32
$K_4(8,3)$	5	37	39	44	96
$K_4(9,3)$	6	101	108	110	256
$K_4(8,4)$	4	9	10	13*	28
$K_4(9,4)$	5	21	23	26	64
$K_4(10,4)$	6	51	55	59	208
$K_4(9,5)$	4	7	8	10	16
$K_4(10,5)$	5	13	15	18	54
$K_4(11,5)$	6	30	31	36	128
$K_4(11,6)$	5	9	11	14	32

*The bound $K_4(8,4) \geqslant 13$ was very recently also obtained by Kéri

Table 3. $q = 5$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_5(5,2)$	3	18	21	22	35
$K_5(6,2)$	4	59	65	71	125
$K_5(7,2)$	5	215	222	225	525
$K_5(6,3)$	3	11	13	16	25
$K_5(7,3)$	4	30	34	38	100
$K_5(8,3)$	5	97	99	109	325
$K_5(7,4)$	3	7	11	12	21
$K_5(8,4)$	4	18	21	25	65
$K_5(9,4)$	5	52	55	64	255
$K_5(9,5)$	4	12	13	19	55
$K_5(10,5)$	5	31	34	41	175
$K_5(11,5)$	6	86	90	103	625
$K_5(10,6)$	4	9	11	16	45
$K_5(11,6)$	5	20	21	29	125
$K_5(11,7)$	4	7	11	12	25

Table 4. $q = 6$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_6(5,2)$	3	29	33	36	66
$K_6(6,2)$	4	115	120	133	274
$K_6(6,3)$	3	17	19	24	41
$K_6(7,3)$	4	57	62	70	246
$K_6(8,3)$	5	217	234	246	1080
$K_6(7,4)$	3	11	13	18	36
$K_6(8,4)$	4	33	36	46	216
$K_6(9,4)$	5	112	114	136	738
$K_6(8,5)$	3	8	13	15	30†
$K_6(9,5)$	4	21	24	32	144
$K_6(10,5)$	5	65	71	83	615
$K_6(10,6)$	4	15	16	25	72

†A recent update in [5] from September 16, 2009

Table 5. $q = 7$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_7(5,2)$	3	43	47	55	97
$K_7(6,2)$	4	204	210	233	343
$K_7(6,3)$	3	25	28	36	77
$K_7(7,3)$	4	99	101	127	343
$K_7(7,4)$	3	16	19	25	49
$K_7(8,4)$	4	56	58	76	343
$K_7(9,4)$	5	221	227	264	1843
$K_7(8,5)$	3	11	17	20	49
$K_7(9,5)$	4	35	39	52	323
$K_7(10,5)$	5	126	131	160	1225
$K_7(9,6)$	3	8	14	17	37
$K_7(10,6)$	4	24	27	39	175

Table 6. $q = 8$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_8(5,2)$	3	63	64	83	128
$K_8(6,2)$	4	337	354	382	512
$K_8(6,3)$	3	35	40	52	107
$K_8(7,3)$	4	161	171	196	512
$K_8(7,4)$	3	22	24	37	92
$K_8(8,4)$	4	89	96	118	512
$K_8(8,5)$	3	15	22	29	90[‡]
$K_8(9,5)$	4	55	58	80	384
$K_8(10,5)$	5	225	232	287	2461
$K_8(9,6)$	3	11	17	22	48
$K_8(10,6)$	4	37	40	58	342
$K_8(10,7)$	3	9	17	20	44[‡]

[‡]A recent update in [5] from September 16, 2009

Table 7. $q = 9$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_9(5,2)$	3	87	90	113	189
$K_9(6,2)$	4	527	546	585	729
$K_9(6,3)$	3	48	57	71	147
$K_9(7,3)$	4	248	252	308	729
$K_9(7,4)$	3	30	35	51	120
$K_9(8,4)$	4	136	144	181	729
$K_9(8,5)$	3	21	27	39	120
$K_9(9,5)$	4	83	88	120	729
$K_9(10,5)$	5	380	390	481	3969
$K_9(9,6)$	3	15	21	31	81
$K_9(10,6)$	4	55	60	87	477
$K_9(10,7)$	3	12	21	25	81

Table 8. $q = 10$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{10}(5,2)$	3	117	128	149	250
$K_{10}(6,2)$	4	788	800	890	1350
$K_{10}(6,3)$	3	64	74	92	209
$K_{10}(7,3)$	4	367	382	451	1350
$K_{10}(7,4)$	3	39	46	66	168
$K_{10}(8,4)$	4	200	210	265	1156
$K_{10}(8,5)$	3	27	34	48	168
$K_{10}(9,5)$	4	121	128	174	1088
$K_{10}(9,6)$	3	19	26	39	114
$K_{10}(10,6)$	4	79	85	122	826
$K_{10}(10,7)$	3	15	26	33	114

Table 9. $q = 11$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{11}(5,2)$	3	154	165	208	365
$K_{11}(6,2)$	4	1135	1174	1276	2343
$K_{11}(6,3)$	3	83	94	119	275
$K_{11}(7,3)$	4	525	539	636	2343
$K_{11}(7,4)$	3	51	59	84	216
$K_{11}(8,4)$	4	283	293	374	1681
$K_{11}(8,5)$	3	34	41	64	216

Table 10. $q = 12$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{12}(5,2)$	3	197	216	256	468
$K_{12}(6,2)$	4	1587	1612	1764	3884
$K_{12}(6,3)$	3	105	120	144	384
$K_{12}(7,3)$	4	729	732	878	3456
$K_{12}(7,4)$	3	64	72	104	264
$K_{12}(8,4)$	4	390	406	513	2304
$K_{12}(8,5)$	3	43	48	78	264

Table 11. $q = 13$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{13}(5,2)$	3	248	278	311	583
$K_{13}(6,3)$	3	132	145	193	495
$K_{13}(7,3)$	4	987	1006	1198	4294
$K_{13}(7,4)$	3	80	91	132	356
$K_{13}(8,4)$	4	526	537	684	3249
$K_{13}(8,5)$	3	53	61	98	356

Table 12. $q = 14$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{14}(5,2)$	3	307	342	381	686
$K_{14}(6,3)$	3	162	175	235	610
$K_{14}(7,3)$	4	1309	1340	1570	4802
$K_{14}(7,4)$	3	98	112	162	448
$K_{14}(8,4)$	4	694	718	917	4356
$K_{14}(8,5)$	3	65	70	119	448

Table 13. $q = 15$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{15}(5,2)$	3	374	414	465	855
$K_{15}(6,3)$	3	197	209	291	730
$K_{15}(7,4)$	3	119	134	192	519
$K_{15}(8,4)$	4	900	932	1181	5625
$K_{15}(8,5)$	3	78	88	141	519

Table 14. $q = 16$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{16}(5,2)$	3	451	491	576	1024
$K_{16}(6,3)$	3	237	248	344	896
$K_{16}(7,4)$	3	142	161	227	611
$K_{16}(8,4)$	4	1149	1180	1507	7569
$K_{16}(8,5)$	3	93	106	168	611

Table 15. $q = 17$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{17}(5,2)$	3	538	568	671	1241
$K_{17}(6,3)$	3	282	289	407	1241
$K_{17}(7,4)$	3	168	191	271	703
$K_{17}(8,5)$	3	110	121	198	703

Table 16. $q = 18$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{18}(5,2)$	3	635	646	807	1458
$K_{18}(6,3)$	3	332	342	471	1353
$K_{18}(7,4)$	3	198	223	316	774
$K_{18}(8,5)$	3	129	144	233	774

Table 17. $q = 19$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{19}(5,2)$	3	744	759	957	1779
$K_{19}(6,3)$	3	387	399	557	1568
$K_{19}(7,4)$	3	231	256	367	866
$K_{19}(8,5)$	3	150	167	271	866

Table 18. $q = 20$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{20}(5,2)$	3	864	896	1088	2000
$K_{20}(6,3)$	3	449	461	650	1600
$K_{20}(7,4)$	3	267	295	419	979
$K_{20}(8,5)$	3	173	191	312	979

Table 19. $q = 21$

$K_q(n,R)$	k	Sph.-Cov. Bound	Old Lower Bound [5]	New Lower Bound	Upper Bound [5]
$K_{21}(5,2)$	3	996	1043	1257	2381
$K_{21}(6,3)$	3	517	542	738	2058
$K_{21}(7,4)$	3	306	336	497	1029
$K_{21}(8,5)$	3	199	221	354	1029

References

[1] W. Chen, I. S. Honkala, *Lower bounds for q-ary covering codes,* IEEE Trans. Inform. Theory **36** (1990), 664-671.

[2] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering Codes*, North Holland, Amsterdam, 1997.

[3] W. Haas, J. Quistorff, J.-C. Schlage-Puchta, *Lower Bounds on Covering Codes via Partition Matrices*, J. Combin. Theory Ser. A **116** (2009), 478-484.

[4] W. Haas, J. Quistorff, J.-C. Schlage-Puchta, *New Lower Bounds for Covering Codes*, preprint.

[5] G. Kéri, *Tables for Covering Codes*, `http://www.sztaki.hu/~keri/codes/`, accessed 30 December 2008.

[6] W. Lang, J. Quistorff, E. Schneider, *Integer Programming for Covering Codes,* J. Comb. and Comb. Comp. **66** (2008), 279-288.

[7] E.R. Rodemich, *Coverings by rook domains,* J. Combin. Theory Ser. A **9** (1970), 117-128.