Revista de Matemática: Teoría y Aplicaciones 2004 **11**(2) : 55–70
cimpa – ucr – ccss issn: 1409-2433

GRAPH DOMINANCE BY ROOK DOMAINS FOR $\mathbb{Z}_p^n$ AND $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$ GRAPHS

Eduardo Piza–Volio*

Received/Recibido: 16 Jan 2004

Abstract

Described within is the problem of finding near-minimum dominating subsets of a given graph by rook domains. Specifically, we study the graphs of the kind $\mathbb{Z}_p^n$ and $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$ and introduce a simulated annealing algorithm to compute upper bounds of the size of minimum dominating subsets.

We demonstrate the effectiveness of the algorithm by comparing the results with a previously studied class of graphs, including the so-called "football pool" graphs and others. We give some new upper bounds for graphs of the kind $\mathbb{Z}_p^n$, with $p \geq 4$. The codes of some dominating subsets are given in an appendix.

Keywords: Graph domination, simulated annealing, football pool problem, combinatorics.

Resumen

En este artículo se describe el problema de la dominación de los grafos del tipo $\mathbb{Z}_p^n$ y mezclas del tipo $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$ a través de subconjuntos dominantes de vértices de tamaño mínimo. Se introduce un algoritmo del tipo de recocido simulado para calcular cotas superiores de la cardinalidad de estos subconjuntos dominantes minimales.

Se demuestra la eficiencia del algoritmo al comparar los resultados obtenidos con los ya conocidos correspondientes a algunas clases de grafos, entre ellos los llamados grafos del "football pool problem". Se establecen cotas superiores en algunos de los grafos del tipo $\mathbb{Z}_p^n$, con $p \geq 4$. Los códigos de algunos subconjuntos dominantes se incluyen en un apéndice.

Palabras clave: Dominación de grafos, recocido simulado, problema de las apuestas en fútbol, combinatoria.

AMS Subject Classification: 68R05, 68R10, 05C69, 90C27.

*Centro de Investigación en Matemática Pura y Aplicada, CIMPA, Universidad de Costa Rica, 2060, San José, Costa Rica. E-Mail: epiza@cariari.ucr.ac.cr.

1 Introduction

The theme of graph domination by rook domains has been deeply studied during the last few years [3, 5, 7, 9, 11, 12, 13, 16], and a large variety of approximate solutions (occasionally exact) has been found to the given problems. The most useful methods used to find near-minimum dominating subsets seem to be the heuristic methods based on the combinatorial optimization techniques, like *tabu search*, *simulated annealing* and *genetic algorithms*. This paper precisely describes a *simulated annealing* algorithm [1, 10, 15] to resolve these kinds of problems.

Throughout this article p will represent a natural number ≥ 2 and n and m will represent natural numbers, with $n+m \geq 1$. We will be working with the graphs F_p^n and $F_{3,2}^{n,m}$, which are defined as follows:

- The sets of vertices of F_p^n and $F_{3,2}^{n,m}$ are $V = \mathbb{Z}_p^n$ and $V = \mathbb{Z}_3^n \times \mathbb{Z}_2^m$, respectively.
- In both graphs F_p^n and $F_{3,2}^{n,m}$ we stipulated that two given vertices are adjacent if they have *Hamming distance* equal to 1, that is, if they differ only in one of their corresponding coordinates.

The vertices are represented as vectors: of n coordinates for the graph F_p^n and $n+m$ coordinates for the graph $F_{3,2}^{n,m}$. Both graphs have been extensively studied in relation to the theme of domination, particularly the F_3^n graph, in which the problem of finding a minimum dominating subset is named as the *football pool problem*.

The terminology "*domination by rook domains*" refers to the kind of metric used for these graphs F_p^n and $F_{3,2}^{n,m}$ (Hamming distance) and comes from the chess context. In fact, the concept of domination by rook domains in the graph F_8^2 precisely coincides with the movements of a rook on a chess board, as illustrated in Figure 1.

On the other hand, the terminology *football pool problem* comes from a system of lottery existent in some countries (for example, "LOTTO" in France and Italy, "PROGOL" in Costa Rica), in which the gamblers have to bet on the results of n soccer games, each one having three possible results: victory of the home team, defeat of the home team, or equal score. In this context, a dominating subset of F_3^n will correspond to a set of lottery bills with the bets of the n games, in such a way that it is granted—under any eventuality of the games' results—that at least one of the bills contains at least $n-1$ correct bets; that is to say that there will be one bill that contains at most one incorrect bet. In this case, maybe the gambler is not going to become a millionaire by guessing all the n games, but nevertheless he will win for sure the second prize (and sometimes the first prize), which is also a sizable winning.

If the lottery under consideration additionally included m games of another sport in which any of them has 2 possible results (winning or losing of the home team, for example), then we are confronting an $F_{3,2}^{n,m}$ graph. A dominating subset of this graph will grant us—under any eventuality of the games' results—that there will be one bill with at most one incorrect bet (maybe with all the bets correct), allowing certainty in winning the second prize and occasionally the first one.

The problem of finding a minimum dominating subset of vertices for these graphs F_p^n and $F_{3,2}^{n,m}$ is a combinatorial optimization problem identified as difficult, not only due to

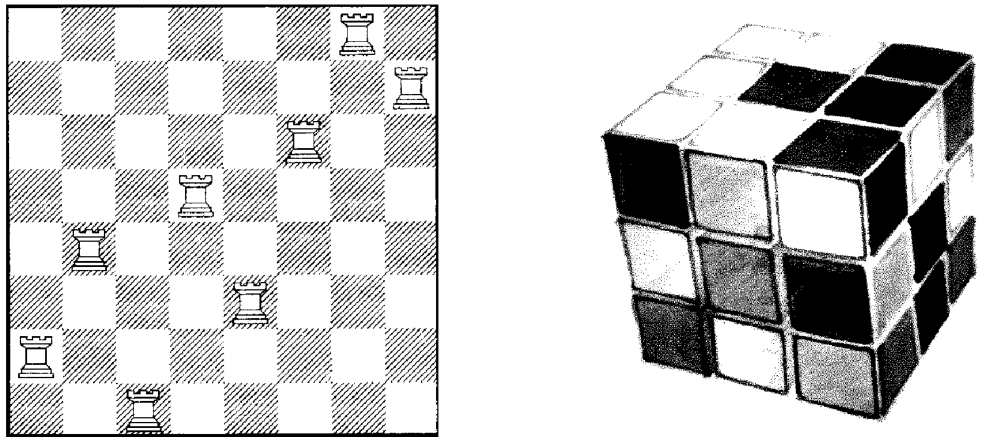

Figure 1: Two of the most popular objects of F_p^n: the movements of the rooks on the chess board in the graph $F_8^2 = (\mathbb{Z}_8^2, H)$, and the Rubik cube $F_3^3 = (\mathbb{Z}_3^3, H)$, where H is the Hamming distance. In the case of a chess board, the figure shows one of the exact solutions to the problem of domination by rook domains: 8 rooks are necessary and sufficient to dominate all the squares of an 8×8 chess board. In the case of a Rubik cube it is necessary and sufficient to have 5 specially chosen "vertices" (small cubes) to dominate the graph; for example these: 012, 021, 100, 211, 222.

the monstrous size of the configuration space that it involves, but also because of the intrinsic algorithmic complexity associated to it. A considerable effort has been dedicated by various authors to the research of dominating subsets for these graphs, especially for the graphs $F_{3,2}^{n,m}$ and F_3^n. Some of this effort involves combinatorial constructions, or heuristics searches, or a combination of both methods. Practically only a few exact solutions for some small values of n, m and p are known. In the majority of the studied cases only an upper bound for the size of the minimum dominating subset is known. A list of the known solutions (exact and approximate upper bounds) can be looked up in the articles of Hämäläinen & Rankinen [7] and Östergård & Hämäläinen [12].

Let's write σ_3^n for the size of a minimum dominating subset of the "football pool problem" with n games. The smallest of the problems for which the exact value of σ_3^n is still unknown is the 729-vertex graph F_3^6. In 1989 van Laarhoven, Aarts, van Lint and Wille [11] found a dominating subset of F_3^6 with 73 vertices, using a simulated annealing algorithm. Therefore, $\sigma_3^6 \leq 73$. Until now, that's the official record, and even if suspected that $\sigma_3^6 = 72$ (see Östergård [14]) nobody has demonstrated yet this supposition and maybe no one ever will. To better understand the magnitude of this problem, in the graph F_3^6 of 729 vertices there are $\binom{729}{72} \approx 0.57 \times 10^{101}$ different subsets of 72 vertices, a monster quantity that makes it impossible to try an exhaustive search for an exact solution. In Figure 2 we present a dominating subset for F_3^6 with 73 vertices (that is, equal to the record) found by the author.

By the use of our simulated annealing algorithm, we have found upper bounds for the size of the minimum dominating subset of F_p^n, for some values of $p \geq 4$. In addition, some

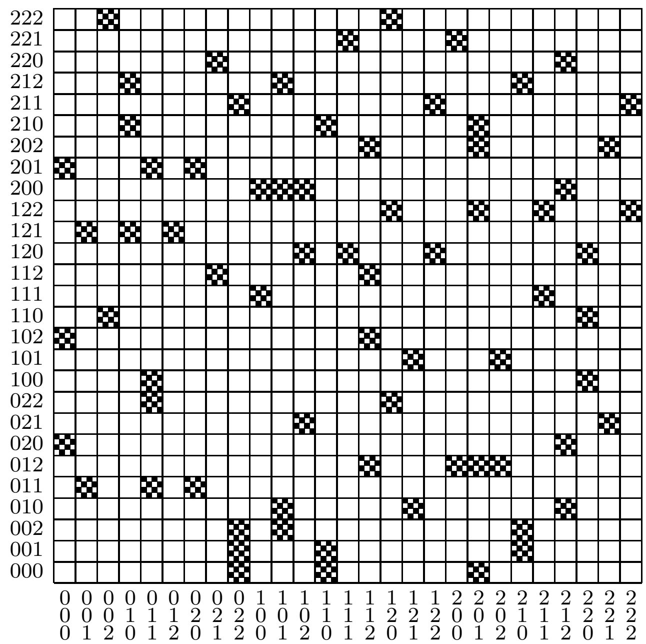

Figure 2: A dominating subset for $\mathbb{Z}_3^6$ graph by 73 vertices, found by the author. In this graph the position of the vertices (6-tuples of $\mathbb{Z}_3^6$) are represented by taking the first 3 coordinates as the abscisse and the last 3 coordinates as the ordinate.

of the upper bounds could be equaled in the analog problem for $F_{3,2}^{n,m}$, F_2^m and F_3^n graphs, proving the effectiveness of the method. The codes of some dominating subsets are given in an appendix.

2 Definitions, notations and basic results

In a graph $G = (V, E)$ we say that a vertex $v \in V$ *dominates* the vertices in the *closed neighborhood* $N[v] = \{v\} \cup \{u \in V : (u, v) \in E\}$. The vertices that are dominated by a subset of the set of vertices $V' \subseteq V$, are those contained in the set

$$\operatorname{dom} V' := \bigcup_{v \in V'} N[v].$$

A *dominating subset* of the graph G is $V' \subseteq V$ such that $\operatorname{dom} V' = V$. A *minimum dominating subset* of the graph G is the one that has the least number of vertices. In general, the problem of finding a minimum dominating subset of G and its cardinality has been well identified as an *NP-hard* problem [6].

Each vertex of $\mathbb{Z}_p^n$ can be viewed as a vector of n coordinates, each one taking values in $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$. Then, the Hamming distance implies that in the graph F_p^n two vertices will be adjacent if they differ in exactly one of their corresponding coordinates. Similarly, in the graph $F_{3,2}^{n,m}$ each element of $V = \mathbb{Z}_3^n \times \mathbb{Z}_2^m$ can be viewed as a vector of $n + m$ coordinates, where the first n coordinates are ternary (taking values in $\mathbb{Z}_3$) and the last m are binary (taking values in $\mathbb{Z}_2$). Consequently, in the graph $F_{3,2}^{n,m}$ two vertices will be adjacent if they differ in exactly one of their corresponding coordinates.

The graph F_p^n is *regular* (all the closed neighborhoods $N[v]$ has the same size) and have *valency* $n(p-1)$ (number of vertices adjacent to each given vertex). We shall denote the size of the minimum dominating subset of F_p^n by σ_p^n. As a vertex $v \in \mathbb{Z}_p^n$ dominates $n(p-1)+1$ vertices, then the following inequalities are verified:

$$\frac{p^n}{n(p-1)+1} \leq \sigma_p^n \leq p^{n-1}. \tag{1}$$

The inequality on the right in (1) is justified by looking at the problem in terms of n games with p different results, so p^{n-1} is a sufficient quantity of vertices to dominate a graph with $n-1$ games. The expression on the left in (1) gives us a lower bound for σ_p^n, usually referred to as the *sphere-packing* bound for F_p^n. A subset of vertices that exactly satisfies the sphere-packing bound is called a *perfect code*, and such a code dominates every vertex precisely once.

Similarly, the graph $F_{3,2}^{n,m}$ is also *regular* and has *valency* $2n+m$. We shall denote by $\kappa(n,m)$ the size of the minimum dominating subset of $\kappa(n,m)$.[1] As a vertex $v \in \mathbb{Z}_3^n \times \mathbb{Z}_2^m$ dominates $2n+m+1$ vertices, then we have the following inequalities:

$$\frac{3^n\, 2^m}{2n+m+1} \leq \kappa(n,m) \leq 3^{n-1}2^m. \tag{2}$$

The inequality on the right in (2) is justified by considering all the possible results of $n-1$ games in $\mathbb{Z}_3$ and m games in $\mathbb{Z}_2$, such that in total we guarantee a bill with $n+m-1$ right results. The expression of the left in (2) gives a lower bound for $\kappa(n,m)$, usually referred to as the *sphere-packing* bound for $F_{3,2}^{n,m}$. A subset of vertices of $F_{3,2}^{n,m}$ that exactly satisfies the sphere-packing bound is called a *perfect code* and such a code dominates every vertex exactly once.

The graphs F_2^m and F_3^n are usually called *hypercube* graph and *football pool* graph respectively. It is known that they contain perfect codes when n and m take certain values (see [4]). In particular, if $m = 2^r - 1$ then the hypercube graph F_2^m contains a perfect code of size 2^{2^r-r-1}. Thus, $\sigma_2^3 = 2$ and $\sigma_2^7 = 16$. Similarly, when $n = (3^r-1)/2$ the football pool graph F_3^n contains a perfect code of size $3^{(3^r-2r-1)/2}$. Therefore, we have $\sigma_3^4 = 9$ and $\sigma_3^{13} = 50049$.

Another known property is that, for big values of n the quantity σ_p^n tends toward the sphere-packing bound (see [2, 4]), that is, for all $p \geq 2$ we have

$$\lim_{n\to\infty} \frac{\sigma_p^n}{n(p-1)+1} = 1. \tag{3}$$

In addition, from the formulation of the graph $F_{3,2}^{n,m}$ in terms of ternary and binary games, it is evident the inequality $2\kappa(n+1,m) \leq 3\kappa(n,m+1)$, that when rewriting it we

[1] This terminology is widely used. Notice that by definition we have $\kappa(n,0) = \sigma_3^n$, while $\kappa(0,m) = \sigma_2^m$.

obtain the next inequality, very useful to find rough upper bounds for $\kappa(n,m)$ for certain values of n and m:

$$\kappa(n,m) \leq \frac{3}{2}\,\kappa(n-1,m+1). \tag{4}$$

Given any additive group G together with a generating subset H that satisfies $H = H^{-1}$, we define the *Cayley* graph of G with respect to H to be the graph $X(G,H)$ whose vertices are the elements of G, where by definition g is adjacent to gh, for all $g \in G$ and $h \in H$. In particular, if we regard $\mathbb{Z}_p^n$ as the additive group G_1 and select $H_1 = \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$, where e_i is the i^{th} standard basis vector, then F_p^n is actually the Cayley graph of G_1 with respect to H_1. Similarly, if we take $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$ as the additive group G_2 and select $H_2 = \{\pm e_1, \ldots, \pm e_n, \pm e_{n+1}, \ldots, \pm e_{n+m}\}$, where e_i is the i^{th} standard basis vector, then $F_{3,2}^{n,m}$ is actually the Cayley graph of G_2 with respect to H_2. For more on Cayley graphs, see Biggs [2].

3 Description of the algorithm

Our goal is to find exact values or minimal upper bounds for σ_p^n and $\kappa(n,m)$, as well as the associated codes of the dominating subsets for the graphs F_p^n and $F_{3,2}^{n,m}$, respectively. To simplify, we will focus on the explanation of the applied methodology used in the case of the graph F_p^n, because for the others graphs the ideas are completely similar.

Let V' be an arbitrary subset of the set of vertices $V = \mathbb{Z}_p^n$ of the graph F_p^n. It could be that V' doesn't dominate the graph F_p^n, but the subset $V' \cup (V - \operatorname{dom} V')$ induced by V' always dominates F_p^n. So, we are looking for a subset $V^\star$ of the set of vertices V that could be a solution for the following combinatorial optimization problem:

$$\begin{array}{l}\text{Minimize } c(V') := |V'| + |V - \operatorname{dom} V'| \\ \text{subject to } V' \subseteq V.\end{array} \tag{5}$$

Therefore, $\sigma_p^n = c(V^\star)$, that is, the number of vertices of the solution $V^\star$ of (5). Our simulated annealing algorithm uses adequate codification for each vertex in V, as well as complete codification inside the computer memory of the closed neighborhoods $N[v]$ for each vertex. It is necessary to intensively use of codification-decodification algorithms from decimal base to base p and base $3^n \times 2^m$, details that are explained later.

The algorithm starts selecting at random a subset V' of vertices, in such a way that the inequality (1) is satisfied with $c(V') := |V'| + |V - \operatorname{dom} V'|$. Next, the following step sequence is repeated, using a parameter t_k for a system "temperature", which occasionally decreases in order to make $t_k \to 0$ slowly, when $k \to \infty$:

1. Any vertex $v \in V$ is chosen, which will give rise to a new subset of vertices V'' in the following way: $V'' = V' \cup \{v\}$, if $v \notin V'$, while $V'' = V' - \{v\}$, if $v \in V'$.

2. Next, $c(V'')$ is calculated, using the already-made calculation for $c(V')$, updating it according to the vertices dominated by the chosen vertex v. We keep the list of all the neighbors of each vertex inside the computer memory, so this calculation is made fast.

3. Next, we take the decision to accept or reject V'' according to probability equal to $\min\{1, e^{-\Delta c/t_k}\}$, where $\Delta c = c(V'') - c(V')$. This acceptance rule is called *Metropolis rule* [1].

Here Δc represents the change in the cost function produced by the inclusion or exclusion of selected vertex v. The Metropolis rule for accepting or rejecting v states that, if vertex v gives rise to a new subset V'' having less cost than V', then it is accepted with probability 1. Otherwise, the new generated subset V'' has greater cost than V' and then it is accepted only with probability $e^{-\Delta c/t_k}$, quantity which decreases to 0 when $t_k \to 0$. The specific details of the cooling schedule are as follows:

(a) **Decrease of temperature:** every certain number of steps the system is cooled down a little, decreasing the value of the temperature t_k using the geometric cooling scheme: $t_{k+1} = \lambda \cdot t_k$, where λ is a previously chosen constant between $[0.92, 0.98]$ ($\lambda \approx 0.95$ was a good selection in almost all our experiments). This makes the Metropolis rule become more strict each time, in the acceptance of vertices that make the cost increase.

(b) **Length of the temperature chains:** the temperature parameter is updated each nLimit steps, or when it has already accepted nOver new subsets V'' for which $c(V'') \geq c(V')$. We successfully use values of nOver $\in [10^5, 10^6]$ and nOver $\in [5000, 50000]$, depending on the size of the problem.

(c) **Initial temperature:** the initial temperature t_0 is selected at the beginning the Metropolis rule in order to let it be more tolerant, accepting nearly $\chi \times 100\%$ of the subsets V'' of which $c(V'') \geq c(V')$. Here χ is a previously chosen constant. With this criteria, $t_0 = (n+1)/2\ln(1/\chi)$. We've used generally $\chi = 0.7$ with good results.

(d) **Criteria to stop the algorithm:** a maximum of 150 cycles of temperature are completed, because in practice the quantity $t_{150} = (t_0)^{150}$ is almost null, although independently of the initial value t_0. Nevertheless, if for the last nCad temperature steps a new good dominating subset V'' doesn't appear, then the process is stopped. We're used nCad $= 3$ in our test with good results.

We use an additive algorithm for the generation of random numbers, proposed by Knuth [8]. In spite that in theory, the method of simulated annealing converges to a global minimum of the objective function $c(V)$ that is being minimized [1], in practice it is necessary to run the algorithm several times to achieve good upper bounds of σ_p^n and $\kappa(n, m)$. For example, $\sigma_3^6 \leq 73$ was obtained after 5 runs, and to find $\sigma_3^7 \leq 186$ about 200 runs were necessary. Besides, a good part of the success of these methods depends on good calibration of the system's parameters.

In Figure 3 the main results achieved are presented for the graph F_p^n, and in Figure 4 we present the corresponding results for the graph $F_{3,2}^{n,m}$. As shown in the tables of these figures, only the cases corresponding to small values of p, n and m are studied.

3.1 Representation in base p and base $3^n \times 2^m$

A substantial part of the success of our algorithm is based on the fact we kept a complete list of the neighbors of each vertex inside the computer main memory, coded in decimal

n	$\mathbb{Z}_2^n$	$\mathbb{Z}_3^n$	$\mathbb{Z}_4^n$	$\mathbb{Z}_5^n$	$\mathbb{Z}_6^n$	$\mathbb{Z}_7^n$	$\mathbb{Z}_8^n$	$\mathbb{Z}_9^n$	$\mathbb{Z}_{10}^n$
1	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$	$^{a}1$
2	$^{a}2$	$^{a}3$	$^{a}4$	$^{a}5$	$^{a}6$	$^{a}7$	$^{a}8$	$^{a}9$	$^{a}10$
3	$^{s,e}2$	$^{e}5$	$^{e}8$	$^{e}13$	$^{e}18$	$^{e}25$	$^{e}32$	$^{e}41$	$^{e}50$
4	$^{e}4$	$^{s,e}9$	$^{e}24$	$^{e}52$	$^{b}72$	$^{b}123$	$^{b}224$	$^{b}390$	
5	$^{e}7$	$^{e}27$	$^{e}64$	$^{b}200$	$^{b}540$				
6	$^{e}12$	$^{b,c}73$	$^{b}334$						
7	$^{s,e}16$	$^{c}186$							
8	$^{e}32$	$^{c}486$							
9	$^{c}62$	$^{c}1341$							
10	$^{c}120$	$^{c}3645$							
11	$^{c}192$	$^{c}9477$							
12	$^{c}380$	$^{c}27702$							
13	$^{c}736$	$^{s}59049$							
14		$^{c}177147$							

Figure 3: Best solutions for the problem of covering the graph $\mathbb{Z}_p^n$ by rook domains. The size of minimum dominating subset of $\mathbb{Z}_p^n$ already known is reported in the table. The superscripts to the left of each entry have the following meaning: "a" denotes a trivial and exact solution; "e" denotes an exact solution found by the algorithm in 100% of the runs; "b" denotes the best upper bound already known and also found by the author; "c" denotes the best known upper bound, found by other authors; "s" denotes an exact *sphere-packing* solution.

base. Thus, it is necessary to have efficient algorithms to run this process. Probably the reader is familiar with the representation of natural numbers in base p, but not with the representation of numbers in extended base $3^n \times 2^m$. Let's make a brief summary of this topic.

3.1.1 Representation in base p

Let s be a natural number. We shall write $[s]_p^n$ to denote the representation of s in base p, using n digits p-adic d_1, d_2, …, d_n. That is,

$$[s]_p^n := \underbrace{(d_n, \ldots, d_2, d_1)_p}_{\text{base } p} = \sum_{i=1}^{n} d_i\, p^i,$$

where each $d_i \in \{0, 1, \ldots, p-1\}$. The maximum natural number that can be represented with this scheme is $p^n - 1$. To find the p-adic digits d_i the efficient and classic algorithm is the well known *Euclidean division* algorithm. The obtained p-adic representation is compatible with the lexicographic order $\preceq$ of $\mathbb{Z}_p^n$, that is,

$$0 \le s \le s' < p^n \Longrightarrow [s]_p^n \preceq [s']_p^n.$$

3.1.2 Representation in base $3^n \times 2^m$

Let $s < 3^n$ and $s' < 2^m$ be two natural numbers. We shall write $([s]_3^n, [s']_2^m)$ to denote the representation of an integer a with $n+m$ digits, from which the first n digits are ternary (base 3) while the last m digits are binary (base 2). That is,

$$a = ([s]_3^n, [s']_2^m) := (\underbrace{d_n, \dots, d_2, d_1}_{\text{base 3}}, \underbrace{r_m, \dots, r_2, r_1}_{\text{base 2}}) := \sum_{i=1}^{m} r_i\, 2^{i-1} + \sum_{i=1}^{n} d_i\, 3^{i-1}\, 2^m,$$

where each $d_i \in \mathbb{Z}_3$ and $r_i \in \mathbb{Z}_2$. This definition is justified by the following result.

Proposition 1 (Base $3^n \times 2^m$) representation *Every natural number $a < 3^n 2^m$ can be represented as a vector of $n+m$ coordinates as*

$$a = ([q_a]_3^n, [r_a]_2^m), \tag{6}$$

where the first n coordinates correspond to the representation of a natural number $q_a < 3^n$ in base 3, while the last m coordinates correspond to the representation of a natural number $r_a < 2^m$ in base 2. The numbers q_a and r_a of this representation are unique.

Proof: Let's consider the quotient q_a and the remainder r_a of the Euclidean division of integer a by 2^m. Therefore, we have that q_a and r_a are the only integers that satisfy $a = 2^m\, q_a + r_a$, with $0 \le r_a < 2^m$. Then, clearly r_a admits a unique representation in base 2 using m digits, denoted by $[r_a]_2^m$. On the other hand,

$$0 \le 2^m\, q_a \le a < 3^n\, 2^m,$$

from which we deduce that $0 \le q_a < 3^n$. Therefore, q_a admits a unique representation in base 3 using n digits, denoted by $[q_a]_3^n$. ∎

The last representation in base $3^n \times 2^m$ is also compatible with the ordinary lexicographic order "$\preceq$" of $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$, in the next sense:

$$0 \le a \le a' < 3^n 2^m \Longrightarrow ([q_a]_3^n, [r_a]_2^m) \preceq ([q_{a'}]_3^n, [r_{a'}]_2^m).$$

4 Combinatorial construction

The smallest known dominating subset of F_3^8 has size 486 and was found by Laarhoven et al. [11] using simulated annealing algorithm in conjunction with a combinatorial construction, which reduces the problem to another one with less dimension. This combinatorial construction was originally formulated by Blokhuis & Lam [3] and in theory can be applied to any graph F_p^n or $F_{3,2}^{n,m}$, but only obtains good results in certain cases.

In this section we shall work with column vectors, instead of row vectors, because of notation convenience.

Definition 2 *Let $A = [a_1|a_2|\cdots|a_n]$ be a $q \times n$ matrix of rank q with entries from $\mathbb{Z}_p$. The set $S \subseteq \mathbb{Z}_p^q$ is said to* **cover $\mathbb{Z}_p^q$ using** *A if*

$$\mathbb{Z}_p^q = \{s + \alpha\, a_i : s \in S,\, \alpha \in \mathbb{Z}_p,\, 1 \leq i \leq n\}.$$

Note that according to this definition, when $q = n$ and A is the identity matrix, then S covers $\mathbb{Z}_p^q$ using A if and only if S is a dominating subset of F_p^n. In general, we have the next result.

Proposition 3 *If S covers $\mathbb{Z}_p^q$ using A, then $W = \{w \in \mathbb{Z}_p^n : Aw \in S\}$ is a dominating subset of F_p^n of size $|W| = |S|\, p^{n-q}$.*

Proof: Let $w \in \mathbb{Z}_p^n$. Then we shall have that $Aw \in \mathbb{Z}_p^q$. By virtue of S covering $\mathbb{Z}_p^q$ using A, there exist $s \in S$, $\alpha \in \mathbb{Z}_p$ and $1 \leq i \leq n$ such that $Aw = s + \alpha\, a_i$. Let e_i be the i^{th} vector of the canonical base of $\mathbb{Z}_p^n$. Then, $a_i = Ae_i$, from where

$$Aw = s + \alpha A e_i,$$

and therefore $A(w - \alpha\, e_i) \in S$. But by definition this means that $w - \alpha\, e_i \in W$, and so w is dominated by W in F_p^n. Finally, by hypothesis A has rank q and therefore $|W| = |S|\, p^{n-q}$, because for each $w \in S$ the inverse image $A^{-1}(\{w\})$ is a vector subspace of dimension $n - q$ and then has exactly p^{n-q} different elements. ■

For example, in the graph F_3^8 of the football pool problem, the set $S \subseteq \mathbb{Z}_3^4$ of 6 vertices

$$S = \{(2,2,2,2)^t, (2,1,2,1)^t, (2,0,1,1)^t, (0,2,1,1)^t, (2,0,1,2)^t, (1,1,2,2)^t\}$$

covers $\mathbb{Z}_3^4$ using the following matrix A of size 4×8:

$$A = \begin{pmatrix} 2 & 0 & 2 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 1 & 0 & 2 & 2 \\ 2 & 0 & 2 & 0 & 1 & 2 & 2 & 1 \\ 0 & 2 & 2 & 2 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Here $q = 4$. In this way Laarhoven et al. [11] found the upper bound $\sigma_3^8 \leq |S| \cdot 3^4 = 486$.

In practice it is very difficult to find $q < n$ and a set $S \subseteq \mathbb{Z}_p^q$ together with a matrix A of size $q \times n$, in such a way that S covers $\mathbb{Z}_p^q$ using A. This problem can also be formulated as a combinatorial optimization problem, in the following way: for any given positive integer k and a selection of $r \in \{1, \ldots, n-1\}$, find a subset $S \subseteq \mathbb{Z}_p^n$ with k r-tuples and a matrix $r \times n$ such that the size of the set

$$\mathbb{Z}_p^r - \{s + \alpha\, a_i : s \in S,\, \alpha \in \mathbb{Z}_p,\, 1 \leq i \leq n\} \tag{7}$$

be minimal.

Again we use a simulated annealing algorithm to solve this optimization problem: a "move" now is either the replacement of one of the r-tuples from S (selected at random) by other r-tuple not belonging to S, or the replacement of a column of A (selected at random) by another column not belonging to A. If the algorithm finds a subset S and a matrix A for which S covers $\mathbb{Z}_p^r$ using A, then the value of k is decreased by 1 and the algorithm is executed again. The process stops when k is such that the algorithm is not able to find a subset S and a matrix A for which the set in (7) be empty.

If we define $G = \mathbb{Z}_p^n$ and $H = \{\pm a_1, \ldots, \pm a_n\}$ then we see that S is simply a dominating subset in the Cayley graph $X = X(G, H)$. This graph X has the same number of vertices than F_p^n, but is denser. By standard results on automorphisms of Cayley graphs, we may assume without loss of generality that $a_i = e_i$ for $1 \le i \le n$, so A consists of full rank leading identity submatrices together with some additional columns. Then X is actually equal to F_p^n with some extra edges determined by these additional columns.

The benefits of this combinatorial construction reside in the fact that they can help us to find dominating subsets in smaller denser graphs, where the simulated annealing algorithm works better. Every dominating subset found in X induces another dominating subset of F_p^n. However, this procedure could not find all the dominating subset of F_p^n, because not all of them have this particular shape. So, even with this technique we could fail to obtain the minimum dominating subset of F_p^n or good upper bounds for their cardinality σ_p^n.

The algorithm just described simplifies itself a little bit if we already have the matrix A. On the matter, Davies & Royle [5] have reported the following result, although without an adequate theoretical justification: in order to find the matrix A the orbits of the projective group $PGL(q, p)$ are studied, extracting a set from it of n projective vectors $a_1, \ldots, a_n$ of q components. These vectors form the columns of matrix A. For these calculations they use a computer package oriented to group theory, named CAYLEY.

For the graph $F_{3,2}^{n,m}$ we have a completely analog combinatorial construction, that is described as follows.

Definition 4 *Let $A = [\,a_1|a_2|\ldots|a_n\,]$ be a matrix of size $q \times n$ of rank q with entries from $\mathbb{Z}_3$. Similarly, let $B = [\,b_1|b_2|\ldots|b_m\,]$ be a matrix of size $r \times m$ of rank r with entries from $\mathbb{Z}_2$. Then, $S \subseteq \mathbb{Z}_3^q \times \mathbb{Z}_2^r$ is said to* **cover** $\mathbb{Z}_3^q \times \mathbb{Z}_2^r$ **using** *A and B if*

$$\begin{aligned}\mathbb{Z}_3^q \times \mathbb{Z}_2^r \quad = \quad & \left\{ \begin{pmatrix} s_1 + \alpha\, a_i \\ s_2 \end{pmatrix} : \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in S,\ \alpha \in \mathbb{Z}_3,\ 1 \le i \le n \right\} \\ & \cup \left\{ \begin{pmatrix} s_1 \\ s_2 + \alpha\, b_j \end{pmatrix} : \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in S,\ \alpha \in \mathbb{Z}_2,\ 1 \le j \le m \right\}\end{aligned}$$

Proposition 5 *If S covers $\mathbb{Z}_3^q \times \mathbb{Z}_2^r$ using A and B, then*

$$W := \left\{ \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbb{Z}_3^n \times \mathbb{Z}_2^m : \begin{pmatrix} Aw_1 \\ Bw_2 \end{pmatrix} \in S \right\}$$

is a dominating subset of $F_{3,2}^{n,m}$ of size $|W| = |S|\, 3^{n-q}\, 2^{m-r}$.

Proof: For any $\binom{x_1}{x_2} \in \mathbb{Z}_3^n \times \mathbb{Z}_2^m$ we have $\binom{Ax_1}{Bx_2} \in \mathbb{Z}_3^q \times \mathbb{Z}_2^r$. So, we can find $\binom{s_1}{s_2} \in S$ to give either

$$\begin{pmatrix} Ax_1 \\ Bx_2 \end{pmatrix} = \begin{pmatrix} s_1 + \alpha\, a_i \\ s_2 \end{pmatrix}, \qquad \alpha \in \mathbb{Z}_3,\ 1 \le i \le n,$$

or

$$\begin{pmatrix} Ax_1 \\ Bx_2 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 + \alpha\, b_j \end{pmatrix}, \quad \alpha \in \mathbb{Z}_2,\ 1 \le j \le m.$$

In the first case, taking e_i the i^{th} basis vector in $\mathbb{Z}_3^n$, we have

$$\begin{pmatrix} A(x_1 - \alpha\, e_i) \\ Bx_2 \end{pmatrix} \in S,$$

and so $\binom{x_1}{x_2} - \alpha \binom{e_i}{0} \in W$, that means that $\binom{x_1}{x_2}$ is dominated by W in $F_{3,2}^{n,m}$. In the second case, taking e_j the j^{th} basis vector in $\mathbb{Z}_2^m$, we have

$$\begin{pmatrix} Ax_1 \\ B(x_2 - \alpha\, e_j) \end{pmatrix} \in S,$$

and therefore $\binom{x_1}{x_2} - \alpha \binom{0}{e_j} \in W$, so $\binom{x_1}{x_2}$ is dominated by W in $F_{3,2}^{n,m}$. Thus, putting the two cases together, we see that W is a dominating subset of $F_{3,2}^{n,m}$. Elementary linear algebra yields that its size is $|W| = |S|\, 3^{n-q}\, 2^{m-r}$. ■

5 Some conclusions

In Figure 3 we present the list of upper bounds of σ_p^n corresponding to the graph F_p^n, for small values of p and n. Some of them are new and were obtained with our simulated annealing algorithm. Other listed upper bounds for σ_p^n are already known.

In Figure 4 the known results concerning upper bounds of $\kappa(n, m)$ are listed. Some of these upper bounds were also found by us in an independent way, through our simulated annealing algorithm.

There's still a lot of work to do, particularly on the graph F_p^n with $p \ge 4$, for which all the upper bounds studied by us have been found directly by the simulated annealing algorithm, without the use of combinatorial constructions described in the last section. Actually, we are working on a more efficient program of the simulated annealing algorithm that includes these combinatorial constructions, with the hope to find upper bounds of σ_n^p for other values of p and n greater than the ones already studied, and maybe improve some of the actually known upper bounds.

Appendix: codes of some dominating subsets

Some of the codes of dominating subsets found by the author are presented here. With the purpose of maintaining consistency with the terminology employed by other authors, we use the compressed notation of Östergård & Hämäläinen [12]. Let's consider all the vectors of $\mathbb{Z}_p^n$ and $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$, listed according to their lexicographic orders. Then, to specify a code of dominating subsets we can simply enumerate the quantity of consecutive positions that they have to skip in the listing.

For example, a code like "11, 0, 5, 2, ..." means that at the beginning we skip the first 11 vectors before selecting the first vector, then we select the next vector, then we

$\mathbb{Z}_2^m \rightarrow$

$\mathbb{Z}_3^n \downarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0		$^{e}1$	$^{e}2$	$^{e}2$	$^{e}4$	$^{e}7$	$^{e}12$	$^{e}16$	$^{e}32$	$^{e}62$	120	192	380	736
1	$^{e}1$	$^{e}2$	$^{e,d}3$	$^{e,d}6$	$^{e}8$	$^{e}16$	$^{e,d}24$	$^{e,d}48$	$^{e}84$	160	284	548	1024	
2	$^{e,d}3$	$^{e,d}4$	$^{e}6$	$^{e,d}12$	$^{e}20$	$^{e,d}36$	$^{e}64$	124	232	408	768	1504		
3	$^{e}5$	$^{e,d}9$	$^{e}16$	$^{e}24$	$^{e}48$	92	171	312	576	1080	2016			
4	$^{e}9$	$^{e}18$	$^{e,d}36$	$^{e,d}72$	128	238	432	$^{d}864$	1296	2592				
5	$^{e,d}27$	$^{e,d}54$	96	168	324	639	1188	$^{d}1944$	$^{d}3888$					
6	73	132	$^{d}252$	468	864	1620	$^{d}2916$	$^{d}5832$						
7	186	333	648	$^{d}1296$	2304	$^{d}4374$	8586							
8	486	$^{d}972$	1728	$^{d}3456$	6480	$^{d}12879$								
9	1341	$^{d}2592$	4860	9639	17496									
10	3645	7047	13122	25192										
11	9477	18894	$^{d}37788$											
12	27702	52488												
13	$^{e}59049$													

Figure 4: Better known solutions to the problem of domination of the graph $\mathbb{Z}_3^n \times \mathbb{Z}_2^m$ by rook domains. In the table, the sizes of the known minimal subset that dominates the graph are listed. The superscript on the left of each entry has the following meaning: "e" indicates an exact solution found by the algorithm in 100% of the runs; "d" indicates an upper bound of $\kappa(n,m)$ derived from the inequality $\kappa(n,m) \leq \frac{3}{2}\kappa(n-1,m+1)$.

skip the next 5 vectors and select the next one, then we skip the next 2 vectors and select the next one, etc.

There is an Internet site that contains the rest of the codes of dominating subsets for the graph $F_{3,2}^{n,m}$, that can be consulted using the "`ftp`" facility inside the directory `pub/graphs/pools` of the Web site `ftp.cs.uwa.edu.au`.

6.1 $\sigma_{10}^3 = 50$. Exact solution found by the algorithm in 100% of the runs.
1, 24, 11, 3, 44, 22, 43, 10, 7, 25, 13, 41, 13, 0, 23, 13, 18, 4, 13, 34, 20, 18, 15, 10, 37, 27, 44, 3, 11, 14, 28, 33, 6, 13, 20, 11, 20, 7, 4, 40, 32, 34, 13, 14, 13, 12, 15, 8, 15, 40.

6.2 $\sigma_9^3 = 41$. Exact solution found by the algorithm in 100% of the runs.
7, 3, 25, 42, 21, 9, 19, 7, 3, 19, 7, 33, 34, 19, 14, 16, 7, 7, 37, 3, 21, 7, 11, 19, 7, 21, 34, 31, 7, 15, 5, 12, 14, 15, 25, 30, 28, 7, 14, 14, 7.

6.3 $\sigma_9^4 \leq 390$. Better upper bound known, found by the algorithm.
19, 8, 9, 6, 32, 5, 31, 0, 30, 9, 20, 23, 32, 30, 2, 18, 13, 24, 34, 21, 7, 7, 4, 12, 18, 46, 12, 31, 7, 2, 21, 25, 20, 3, 15, 2, 20, 13, 27, 9, 11, 23, 4, 23, 8, 21, 33, 3, 6, 12, 5, 30, 23, 14, 12, 24, 12, 3, 48, 1, 0, 3, 28, 19, 9, 0, 33, 19, 1, 1, 25, 23, 33, 6, 23, 0, 24, 27, 10, 28, 0, 4, 5, 14, 12, 36, 21, 27, 5, 11, 12, 22, 6, 5, 23, 22, 1, 2, 6, 3, 9, 29, 12, 29, 22, 42, 27, 6, 25, 4, 19, 24, 16, 1, 21, 12, 38, 16, 5, 37, 14, 5, 3, 12, 44, 9, 29, 3, 21, 27, 5, 15, 38, 42, 4, 9, 10, 21, 4, 16, 14, 26, 11, 11, 24, 21, 10, 27, 31, 32, 12, 16, 4, 10, 30, 6, 4, 19, 11, 14, 9, 5, 24, 9, 18, 8, 16, 15, 15, 24, 4, 9, 33, 11, 20, 7, 6, 22, 13, 4, 6, 42, 0, 4, 4, 6, 17, 20, 15, 0, 0, 3, 15, 34, 30, 15, 9, 33, 3, 29, 11, 13, 2, 11, 29, 6, 16, 7, 16, 24, 7, 4, 10, 27, 55, 0, 22, 7, 5, 12, 38, 15, 18, 5, 6, 54, 15, 16, 15, 9, 52, 24, 3, 30, 5, 8, 18, 18, 18, 12, 24, 2, 12, 5, 18, 36, 9, 1, 2, 25, 16, 9, 6, 3, 5, 33, 15, 17, 15, 21, 10, 7, 19, 7, 5, 48, 18, 13, 14, 13, 2, 41, 37, 13, 11, 36, 11, 7, 7, 18, 28, 5, 4, 20, 11, 22, 0, 15, 25, 16, 9, 6, 0, 10, 3, 20, 34, 28, 28, 20, 36, 6, 25, 9, 19, 5, 19, 11, 3, 42, 6, 21, 14, 28, 22, 15, 18, 1, 18, 10, 3, 9, 37, 10, 9,

31, 10, 9, 9, 43, 4, 14, 9, 8, 13, 23, 5, 33, 12, 20, 36, 2, 20, 9, 28, 3, 33, 10, 9, 1, 29, 1, 26, 31, 0, 4, 23, 28, 7, 28, 13, 6, 5, 31, 14, 4, 9, 26, 10, 13, 15, 32, 0, 19, 40, 22, 9, 41, 7, 4, 13, 24, 8, 12, 5, 21, 14, 6, 15, 40.

6.4 $\sigma_8^3 = 32$. Exact solution found by the algorithm in 100% of the runs.
3, 5, 30, 22, 18, 11, 5, 16, 30, 8, 4, 19, 31, 3, 10, 14, 11, 14, 27, 13, 13, 3, 29, 14, 28, 6, 9, 11, 14, 6, 38, 11.

6.5 $\sigma_8^4 \leq 226$. Better upper bound known, found by the algorithm.
5, 34, 23, 18, 0, 24, 18, 16, 26, 8, 24, 8, 4, 5, 19, 2, 21, 6, 28, 23, 10, 32, 33, 7, 9, 14, 1, 29, 9, 20, 16, 6, 34, 20, 2, 5, 30, 24, 9, 28, 0, 19, 17, 18, 15, 33, 9, 16, 36, 2, 28, 9, 19, 18, 20, 18, 14, 9, 46, 8, 19, 8, 26, 6, 9, 1, 40, 7, 26, 0, 8, 14, 37, 3, 19, 20, 1, 21, 38, 10, 24, 2, 17, 14, 25, 4, 36, 10, 8, 35, 18, 42, 9, 4, 10, 6, 27, 45, 3, 6, 21, 23, 3, 4, 29, 23, 8, 27, 4, 17, 17, 7, 20, 31, 8, 25, 15, 5, 1, 28, 0, 23, 6, 34, 20, 8, 33, 18, 3, 34, 9, 36, 2, 28, 26, 12, 21, 7, 2, 40, 23, 35, 2, 44, 10, 24, 30, 4, 3, 22, 32, 6, 20, 6, 0, 17, 11, 10, 23, 36, 16, 6, 2, 37, 29, 12, 31, 23, 2, 31, 46, 3, 26, 28, 10, 24, 30, 8, 19, 8, 8, 2, 25, 10, 12, 34, 3, 18, 4, 42, 5, 25, 8, 26, 21, 1, 12, 33, 8, 44, 23, 34, 16, 6, 33, 10, 33, 6, 2, 10, 14, 23, 2, 8, 31, 23, 3, 34, 35, 6, 26, 31, 4, 24.

6.6 $\sigma_7^3 = 25$. Exact solution found by the algorithm in 100% of the runs.
1, 9, 29, 2, 10, 2, 26, 9, 21, 1, 9, 31, 9, 8, 19, 3, 28, 10, 2, 10, 25, 3, 22, 8, 1.

6.7 $\sigma_7^4 \leq 123$. Better upper bound known, found by the algorithm.
1, 39, 13, 29, 31, 24, 0, 4, 39, 18, 9, 10, 32, 0, 29, 8, 17, 2, 26, 29, 13, 39, 23, 0, 29, 7, 29, 19, 17, 2, 39, 24, 0, 14, 9, 10, 30, 31, 16, 31, 10, 13, 11, 32, 23, 8, 0, 36, 21, 13, 1, 24, 31, 0, 20, 15, 3, 27, 14, 8, 19, 30, 25, 10, 7, 37, 22, 5, 38, 24, 32, 12, 10, 7, 27, 15, 3, 18, 38, 18, 14, 8, 30, 32, 4, 30, 33, 22, 23, 23, 24, 23, 19, 13, 1, 33, 31, 7, 31, 0, 18, 13, 11, 16, 0, 36, 17, 14, 8, 27, 10, 7, 21, 30, 23, 15, 3, 33, 32, 0, 38, 29, 22.

6.8 $\sigma_6^3 = 18$. Exact solution found by the algorithm in 100% of the runs.
6, 9, 6, 15, 22, 4, 14, 6, 0, 28, 0, 9, 14, 25, 4, 5, 22, 7.

6.9 $\sigma_6^4 \leq 72$. Better upper bound known, found by the algorithm.
0, 10, 40, 8, 30, 12, 15, 14, 28, 10, 8, 0, 34, 0, 42, 14, 28, 10, 19, 8, 30, 12, 2, 10, 38, 12, 31, 6, 6, 8, 43, 16, 4, 2, 43, 10, 28, 14, 3, 6, 38, 16, 9, 4, 44, 6, 33, 8, 26, 8, 7, 4, 44, 6, 13, 6, 38, 16, 25, 14, 16, 10, 27, 16, 4, 2, 51, 6, 6, 8, 39, 12.

6.10 $\sigma_6^5 \leq 540$. Better upper bound known, found by the algorithm.
7, 18, 1, 9, 9, 27, 6, 11, 27, 8, 26, 7, 18, 24, 7, 15, 21, 0, 27, 15, 19, 23, 2, 9, 13, 13, 19, 2, 9, 37, 7, 5, 8, 31, 8, 35, 10, 2, 4, 15, 24, 14, 19, 6, 32, 14, 1, 26, 15, 2, 9, 9, 23, 13, 42, 1, 14, 6, 12, 14, 9, 18, 9, 16, 13, 31, 24, 3, 8, 21, 8, 34, 27, 7, 25, 11, 14, 4, 22, 21, 20, 4, 1, 31, 3, 10, 7, 15, 27, 18, 14, 7, 4, 19, 25, 4, 28, 32, 7, 7, 26, 6, 13, 16, 9, 21, 1, 12, 0, 2, 40, 5, 6, 2, 12, 14, 11, 26, 31, 10, 6, 2, 23, 16, 2, 38, 25, 9, 14, 18, 7, 5, 4, 13, 42, 20, 3, 7, 10, 3, 4, 7, 7, 17, 9, 3, 29, 13, 1, 15, 20, 9, 1, 10, 28, 15, 11, 19, 15, 6, 41, 9, 18, 16, 7, 24, 1, 6, 12, 9, 16, 16, 23, 8, 6, 6, 27, 12, 17, 6, 14, 22, 6, 4, 5, 18, 20, 6, 26, 20, 10, 0, 7, 29, 18, 10, 6, 13, 31, 8, 5, 17, 14, 18, 19, 3, 0, 22, 22, 16, 5, 4, 41, 10, 9, 13, 8, 1, 15, 21, 9, 14, 3, 48, 16, 7, 23, 8, 39, 0, 10, 11, 45, 21, 5, 12, 9, 23, 3, 10, 8, 4, 5, 16, 3, 22, 0, 21, 33, 6, 15, 4, 19, 8, 14, 14, 3, 9, 11, 21, 21, 12, 21, 27, 4, 8, 27, 5, 13, 25, 2, 7, 3, 21, 10, 4, 26, 12, 13, 0, 10, 11, 9, 3, 19, 12, 15, 9, 33, 2, 19, 20, 21, 7, 0, 24, 10, 9, 13, 19, 14, 7, 20, 22, 0, 13, 13, 10, 19, 14, 26, 18, 12, 5, 13, 16, 7, 2, 13, 17, 12, 3, 3, 33, 3, 12, 18, 24, 7, 15, 12, 14, 4, 11, 6, 3, 11, 34, 9, 5, 19, 11, 6, 9, 1, 2, 27, 7, 40, 8, 31, 8, 14, 12, 9, 1, 1, 17, 12, 15, 14, 18, 19, 6, 8, 9, 20, 8, 6, 19, 12, 9, 13, 27, 1, 21, 0, 9, 5, 15, 6, 24, 15, 16, 0, 25, 4, 31, 4, 15, 20, 14, 17, 10, 14, 7, 17, 5, 7, 15, 4, 5, 19, 1, 15, 5, 8, 10, 8, 9, 10, 3, 19, 6, 1, 16, 39, 2, 1, 14, 10, 29, 3, 27, 9, 10, 23, 6, 19, 4, 1, 10, 6, 10, 24, 1, 10, 29, 6, 13, 7, 4, 10, 29, 6, 7, 12, 33, 8, 27, 13, 10, 32, 9, 7, 26, 6, 14, 9, 17, 15, 21, 10, 9, 27, 26, 8, 18, 38, 8, 21, 14, 10, 19, 14, 2, 1, 2, 51, 9, 26, 1, 1, 18, 5, 4, 16, 15, 22, 24, 18, 18, 7, 29, 11, 9, 12, 9, 18, 4, 8, 28, 3, 10, 14, 14, 10, 21, 10, 10, 10, 3, 29, 18, 13, 13, 13, 11, 22, 33, 3, 18, 4, 2, 17, 15, 7, 4, 11, 15, 21, 6, 5, 13, 9, 28, 17, 10, 1, 12.

6.11 $\sigma_5^3 = 13$. Exact solution found by the algorithm in 100% of the runs.
0, 7, 13, 13, 7, 7, 2, 17, 15, 1, 11, 3, 12.

6.12 $\sigma_5^4 = 52$. Exact solution found by the algorithm in 100% of the runs.
15, 4, 8, 3, 19, 5, 26, 0, 23, 0, 14, 3, 0, 27, 24, 15, 16, 5, 19, 3, 14, 0, 27, 3, 19, 5, 2, 5, 22, 0, 32, 0, 13, 12, 5, 6, 12, 3, 8, 19, 5, 19, 5, 26, 0, 22, 1, 16, 16, 3, 19, 5.

6.13 $\sigma_5^5 \leq 200$. Better upper bound known, found by the algorithm.

18, 27, 3, 3, 41, 10, 4, 17, 3, 0, 3, 12, 45, 3, 38, 30, 17, 3, 11, 4, 28, 3, 8, 22, 31, 4, 28, 15, 4, 31, 3, 1, 32, 5, 32, 6, 13, 12, 4, 14, 43, 11, 4, 32, 18, 3, 10, 3, 0, 3, 28, 7, 41, 28, 18, 15, 4, 32, 3, 7, 4, 15, 44, 1, 3, 37, 7, 3, 26, 4, 14, 15, 20, 5, 7, 13, 13, 14, 7, 10, 16, 4, 16, 43, 5, 36, 3, 22, 27, 3, 1, 43, 3, 8, 4, 11, 3, 27, 4, 30, 13, 4, 33, 26, 17, 13, 3, 0, 3, 26, 10, 39, 10, 30, 0, 3, 13, 1, 40, 5, 27, 3, 0, 3, 26, 17, 0, 3, 33, 22, 21, 13, 4, 30, 6, 3, 6, 21, 4, 12, 43, 3, 1, 40, 22, 3, 27, 5, 7, 35, 8, 7, 18, 4, 33, 18, 13, 16, 2, 3, 40, 3, 17, 15, 4, 28, 8, 26, 27, 4, 28, 3, 5, 5, 4, 35, 18, 31, 3, 45, 6, 3, 0, 3, 17, 4, 12, 3, 41, 7, 0, 37, 26, 3, 6, 4, 15, 27, 5, 31.

6.14 $\sigma_4^3 = 8$. Exact solution found by the algorithm in 100% of the runs.
0, 4, 20, 4, 11, 2, 2, 2.

6.15 $\sigma_4^4 = 24$. Exact solution found by the algorithm in 100% of the runs.
12, 4, 1, 18, 3, 17, 5, 17, 3, 20, 1, 2, 18, 5, 18, 1, 22, 1, 13, 1, 20, 1, 20, 5.

6.16 $\sigma_4^5 = 64$. Exact solution found by the algorithm in 100% of the runs.
7, 21, 4, 21, 21, 5, 22, 5, 14, 25, 10, 25, 9, 9, 24, 9, 19, 5, 30, 5, 13, 21, 12, 21, 16, 9, 16, 9, 17, 25, 2, 25, 5, 25, 8, 25, 9, 9, 26, 9, 14, 21, 6, 21, 21, 5, 20, 5, 27, 9, 18, 9, 17, 25, 0, 25, 16, 5, 28, 5, 13, 21, 14, 21.

6.17 $\sigma_3^6 \leq 73$. Better upper bound known. This is another different solution to the one already presented in Figure 2.
8, 1, 23, 6, 3, 2, 17, 10, 18, 6, 18, 0, 10, 0, 0, 0, 27, 16, 7, 2, 9, 28, 0, 6, 0, 5, 9, 2, 6, 4, 13, 16, 12, 14, 3, 6, 2, 8, 22, 13, 12, 7, 1, 10, 15, 12, 9, 3, 17, 24, 13, 10, 7, 2, 3, 2, 6, 10, 9, 12, 8, 8, 22, 5, 4, 13, 6, 12, 1, 8, 6, 3, 19.

References

[1] Aarts, E.; Korst, J. (1990) *Simulated Annealing and Boltzmann Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Chichester.

[2] Biggs, N. (1974) *Algebraic Graph Theory*. Cambridge University Press.

[3] Blokhuis, A.; Lam, C.W.H. (1984) "More coverings by rook domains", *Journal of Combinatorial Theory*, Series A **36**: 240–244.

[4] Cameron, P.J.; van Lint, J.H. (1991) *Designs, Graphs, Codes and their Links*. London Mathematical Society Student Texts, **22**, Cambridge University Press.

[5] Davies, R.; Royle, G.F. (1997) "Graph domination, tabu search and the football pool problem", *Discrete Applied Mathematics* **74**(3): 217–228.

[6] Garey, M.; Johnson, D. (1979) *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco.

[7] Hämäläinen, H.O.; Rankinen, S. (1991) "Upper bounds for football pool problem and mixed covering codes", *Journal of Combinatorial Theory*, Series A, **56**: 84–95.

[8] Knuth, D.E. (1981) *Seminumerical Algorithms*. Second edition, volume 2 of the book *The Art of Computer Programming*. Addison-Wesley, Reading, Mass.

[9] Koschnick, K.U. (1993) "A new upper bound for the football pool problem for nine matches", *Journal of Combinatorial Theory*, Series A, **62**: 162–167.

[10] van Laarhoven, P.J.M. (1988) "Theoretical and computational aspects of simulated annealing", Centrum voor Wiskunde in Informatica, Tract 57.

[11] van Laarhoven, P.J.M.; Aarts, E.J.L.; van Lint, J.H.; Wille, L.T. (1989) "New upper bounds for the football pool problem for 6, 7 and 8 matches", *Journal of Combinatorial Theory*, Series A, **52**: 304–312.

[12] Östergård, P.R.J.; Hämäläinen, H.O. (s.f.) "A new table of binary/ternary mixed covering codes", Preprint.

[13] Östergård, P.R.J. (1994) "New upper bound for the football pool problem for 11 and 12 matches", *Journal of Combinatorial Theory*, Series A, **67**: 161–168.

[14] Östergård, P.R.J. (1995) "A combinatorial proof for the football pool problem for six matches", *Journal of Combinatorial Theory*, Series A: 160–163.

[15] Piza, E.; Trejos, J.; Murillo, A. (1998) "Global stochastic optimization techniques applied to partitioning", in A. Rizzi et al. (Eds.) *Advances in Data Science and Classification*, Springer Verlag, Berlin: 185–191.

[16] Wille, L.T. (1987) "The football pool problem for 6 matches: A new upper bound obtained by simulated annealing", *Journal of Combinatorial Theory*, Series A, **45**: 171–177.