Graph Domination, Tabu Search and the Football Pool Problem

Rowan Davies and Gordon F. Royle

September 14, 1995

Abstract

We describe the use of a tabu search algorithm for generating near minimum dominating sets in graphs. We demonstrate the effectiveness of this algorithm by considering a previously studied class of graphs, the so-called "football pool" graphs, and improving many of the known upper bounds for this class.

1 Introduction

This paper describes a method for finding near-minimum dominating sets in graphs using a tabu search algorithm. Tabu search is the name given to a relatively new class of neighbourhood search algorithms that often appear to outperform other heuristic approximation methods on combinatorial optimisation problems (see Glover and Laguna [6]).

In Davies [4], an investigation into a wide variety of methods for finding minimum dominating sets was conducted. There it was concluded that the most promising method is to express the domination problem as a combinatorial optimization problem and use algorithms for integer linear programming, particularly those for the set covering problem (which includes the minimum dominating set problem as a special case). There are many articles describing such algorithms each of which gives experimental results to certain test problems, and the best of these algorithms have consistently solved test problems with hundreds of variables and constraints (see Davies [4] for details).

In this investigation a series of graphs $F_{t,b}$ was used as benchmarks to compare the algorithms. The graph $F_{t,b}$ is defined as follows: the vertex set is $\mathbf{Z}_3^t \times \mathbf{Z}_2^b$ and two vertices are adjacent if they have Hamming distance one. This series of graphs was chosen because they have already been quite extensively studied with regard to domination, particularly the case $F_{t,0}$, where the problem of finding a minimum dominating set is often called the football pool problem. The term *football pool problem* arises because in many countries there is a form of lottery where speculators must select the results of association football (soccer) matches as being either a win for the home team, a win for the away team or a draw. A dominating set for $F_{t,0}$ corresponds to a set of selections for t matches that guarantees at most one incorrect selection. If b matches are so one-sided that effectively only two outcomes are considered possible, then a dominating set for $F_{t,b}$ guarantees at most one incorrect selection on $t+b$ matches. Hämäläinen and Rankinen [7] collated the results of many (often commercial rather than academic) publications into a listing of the smallest known dominating sets for all the graphs $F_{t,b}$ with $t+b \leq 13$. Using this list therefore enables us to make more informed comparisons than using randomly generated test data. This table has recently been updated by Östergård and Hämäläinen [11].

The smallest of the football pool problems for which the size of a minimum dominating set is not known is the 729-vertex graph $F_{6,0}$. Finding a dominating set for $F_{6,0}$ can be expressed as an integer linear program with 729 variables and 729 constraints each with 13 variables. The best set covering algorithms described in the literature could reliably solve problems of a similar size and density. However it seems that the dominating set problem is a sufficiently special case of set covering that it completely defeats these algorithms, and none of them came remotely close to providing an optimal solution to just this one problem. One possible explanation is that the football pool problems are very highly symmetric, whereas the algorithms described were usually tested on randomly generated test data, with non-uniform cost functions.

As with any integer linear program the main problem is finding good lower bounds for the size of the dominating set. The relaxed linear program has a fractional solution whose cost can be significantly lower than that of the optimal integer solution. Davies [4] showed that any branch-and-bound type algorithm to find a minimum dominating set for $F_{t,0}$ must take at least $2^{\min(p,2t-1)}$ steps where p is the difference between the cost of the fractional solution and the integer solution.

Davies concluded that with current techniques it is extremely difficult to find exact solutions or even to find good lower bounds. Therefore in this paper we turn our attention to finding good upper bounds. These are much easier to find because any dominating set provides an upper bound, and hence we can concentrate on finding specific near-minimum dominating sets.

As previously mentioned, considerable effort has been expended in finding dominating sets for the graphs $F_{t,b}$ and in particular $F_{t,0}$ for $6 \leq t \leq 12$. Most of these efforts have involved either combinatorial constructions, heuristic search or a combination of the two (see Wille [14], van Laarhoven et. al. [10]).

Our intention in this paper is to show that a very straightforward implementation of tabu search is sufficient to generate good solutions to these problems that have already been extensively studied. We believe that this demonstrates that tabu search outperforms other heuristic algorithms previously used to generate upper bounds for dominating sets for these graphs. Our hope is that this initial progress will stimulate further research into possibly more complicated tabu-search based implementations.

In addition we generalise a construction of Blokhuis and Lam [2] and using it in conjunction with our tabu search algorithm we improve many of the bounds given by Hämäläinen and Rankinen [7]. A different, but also straightforward, implementation of tabu search by Östergård [12], performed independently of this work, has also improved some of these bounds, and further improvements continue to be made.

2 Definitions and notation

In a graph $G = (V, E)$ a vertex $v \in V$ is said to *dominate* the vertices in the closed neighbourhood $N[v] = \{v\} \cup \{u \in V \mid (u, v) \in E\}$.

The vertices dominated by a subset of the vertex set $V' \subseteq V$ are those contained in the set

$$\mathrm{dom} V' = \bigcup_{v \in V'} N[v]$$

A *dominating set* of G is a set $V' \subseteq V$ such that $\mathrm{dom} V' = V$, and a *minimum dominating set* of a graph is one with the least possible number of vertices. The problem of finding a minimum dominating set of an arbitrary graph is NP-hard [5].

Each element of the set $V = \mathbf{Z}_3^t \times \mathbf{Z}_2^b$ can be regarded as a "vector" of length $t + b$ with t ternary coordinates and b binary coordinates, hence in $F_{t,b}$ two vectors are adjacent if they differ in precisely one coordinate.

The graph $F_{t,b}$ is regular and has valency $2t + b$. We shall denote the size of the minimum dominating set of $F_{t,b}$ by $K(t, b)$. As a vertex dominates $2t + b + 1$ vertices we have the inequality

$$K(t, b)(2t + b + 1) \geq 3^t 2^b$$

which is usually referred to as the sphere-packing bound. A set of vertices that exactly satisfies the sphere-packing bound is called a *perfect code*, and such a code dominates every vertex precisely once.

The graphs $F_{0,b}$ and $F_{t,0}$ are usually called Hamming graphs and they contain perfect codes called Hamming codes (see for example [3]) whenever t and b take on certain values. In particular if $b = 2^r - 1$ then $F_{0,b}$ contains a perfect code of size 2^{2^r-r-1}. Thus we have $K(0,3) = 2$ and $K(0,7) = 16$. Similarly when $t = (3^r - 1)/2$ then $F_{t,0}$ contains a perfect code of size $3^{(3^r-2r-1)/2}$. Therefore we have $K(4,0) = 9$ and $K(13,0) = 59049$.

Given any 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, and where g is adjacent to gh for all $g \in G$, $h \in H$. If we regard $V = \mathbf{Z}_3^t \times \mathbf{Z}_2^b$ as an additively written group G and put $H = \{\pm e_1, \pm e_2, \ldots, \pm e_t, e_{t+1}, \ldots, e_{t+b}\}$ where e_i is the i'th standard basis vector, then we see that $F_{t,b}$ is actually just a Cayley graph of G with respect to H. This observation will play a role later. For more on Cayley graphs, see Biggs [1].

3 Tabu search and results

Tabu search is a general purpose approximation algorithm for combinatorial optimisation based on neighbourhood search. Consider a very general problem of the following form.

Minimise $c(x)$ subject to $x \in X$, with X a set of possible configurations.

Neighbourhood search algorithms initially impose a neighbourhood structure on the set of configurations X. Starting at a (possibly randomly) chosen element $x_0 \in X$ the algorithm proceeds by repeatedly moving from a configuration to one of its neighbours, with the ultimate aim of finding a configuration of low cost. The choice of a good neighbourhood structure and a good schedule for organising the moves is highly problem dependent. Hill-climbing and simulated annealing are two popular neighbourhood search techniques that have had some success in combinatorial problems. Tabu search is a fairly recent technique that proceeds as follows:

Algorithm 3.1 *Start at any $x_0 \in X$. At step i choose x_i as a neighbour of x_{i-1} that minimises $c(x_i)$ subject to the constraint that the move from x_{i-1} to x_i does not 'undo' any* *of the t most recent moves. Finish after a number of iterations, returning the x_i which gave the least $c(x_i)$.*

Here t is a parameter called the *tabu tenure* and it is the use of this parameter that gives tabu search its power. We can view this parameter as the length of a list containing the t most recently performed moves. Reversing any of these t moves is then forbidden or *tabu*. Therefore the operation of the algorithm is constrained by being forced to avoid these particular moves. Without this restriction, the search is simply a greedy algorithm, always taking the move that is locally optimal. On reaching a local minimum the greedy algorithm usually takes a single step upwards, and then immediately returns to the same local minimum resulting in an infinite cycle of length two. The tabu list prevents an immediate return to that local minimum, and with luck the search is forced out of the region of attraction of that local minimum.

The form of tabu search described here is a "short term" form that makes use of recency-based memory. More general forms of tabu search include longer term considerations that incorporate additional memory structures, such as those based on frequency which underly intensification and diversification strategies (according to the sets over which frequency is defined). Although tabu search implementations normally rely on longer term elements to be fully effective, our study has successfully made use of just the short term aspects. Full details of these additional concepts may be found in [6].

The rationale behind tabu search in its short term form (to which we henceforth restrict our attention) is to spend as much time as possible investigating low cost configurations, while providing a mechanism to escape from local minima. By contrast, simulated annealing also enables an escape from local minima, but spends a good deal of the early portion of its search performing a near-random walk among configurations of relatively high cost.

To use this algorithm some neighbourhood structure on X needs to be defined, as does which pairs of moves 'undo' each other. How this is done depends on the problem being solved, and usually follows quite naturally from the structure of the set X.

Tabu search seems particularly well suited to graph domination because there is a very natural cost function and neighbourhood structure. To find a minimum dominating set for a graph $G = (V, E)$ we take X to be the set of all subsets of V. The *cost* of a subset V' is defined to be

$$c(V') = |V'| + |V \setminus \mathrm{dom} V'|$$

Note that X includes sets which are not dominating, but that any set $V' \in X$ determines a dominating set of size $c(V')$ because $V' \cup (V \setminus \mathrm{dom} V')$ is trivially a dominating set. The reason for allowing X to contain non-dominating sets is that this allows paths between small dominating sets which might otherwise need to go via a much larger dominating set. This technique is recommended by Glover and Laguna [6] for tabu search applications. It was also used in the application of simulated annealing to the football pool problem in van Laarhoven [9] as was the neighbourhood structure we used.

Two subsets $V_1, V_2 \subseteq V$ are defined as neighbours in X if one is obtained from the other by the addition of one vertex. Thus a move in the tabu search algorithm involves either the deletion or the addition of one vertex. Two moves are defined to 'undo' each other if one is addition and the other deletion of the same vertex.

Implementation of the algorithm is straightforward, once the above details have been decided. However it is important to be able to quickly decide which is the best possible move. For this purpose it is crucial to maintain a list storing for each vertex the cost associated with moving that vertex (ether into or out of V'), and only updating the few values which might change after each move is made. Another important feature is to make a random choice whenever there are several equally good optimal moves. This tends to diversify the search, and helps to prevent cycling. A variety of history lengths were chosen, and it was determined that the shortest possible length that avoided cycling gave the best results. We used lengths of between 5 and 8 for most of the runs. A very simple *aspiration criterion* was also used — if an otherwise tabu move gave an improvement on the best solution found to date, then it was performed anyway.

Our initial tests were on the graphs $F_{6,0}$ and $F_{7,0}$ for which dominating sets of size 73 and 186 (respectively) were the best known. The algorithm consistently found a dominating set of size 73 for $F_{6,0}$ within a few hundred thousand iterations, compared to the results of simulated annealing reported by van Laarhoven et. al. [10] where a large number of runs, each of one million iterations was required to find a dominating set of the same size (however, we must remember that the cost of a single iteration of simulated annealing is considerably less than that of a single iteration of tabu search). A dominating set of size 186 for $F_{7,0}$ was also found directly by this algorithm, with a run of several million iterations.

For the graphs $F_{m,n}$ the tabu search directly found a dominating set of size 128 for $F_{4,4}$ improving the bound from 132. The algorithm either matched or improved the given bounds for all of the smaller (under 1000 vertices) graphs in the tables of Hämäläinen and Rankinen [7], but the other improvements were also found by Östergård and Hämäläinen [11].

We conclude that tabu search in the short term form we have applied is an excellent algorithm for graph domination problems on moderately sized graphs. However as the graph size increases, the cost per iteration increases quite rapidly if all potential moves are evaluated at each iteration. Even with careful coding ensuring that only the moves whose values change are re-evaluated this problem still renders tabu search less and less attractive as the graph size increases. One possible approach to overcome this problem is to incorporate a *candidate list strategy* (see [6]) to restrict the moves examined to a subset of those possible. In his implementation of tabu search, Östergård [12] successfully restricts attention to moves that involve an uncovered vertex.

4 Combinatorial construction

The smallest known dominating set of $F_{8,0}$ has size 486 and was found using simulated annealing in conjunction with a combinatorial construction [van Laarhoven et al, 1988]. This construction was formulated by [Blokhuis and Lam, 1984] and can be applied to any $F_{n,0}$.

We generalise their proof to $F_{m,n}$, and then use this in conjunction with tabu search to obtain a powerful algorithm.

Our generalised construction is as follows:

Definition 4.1 *Let $A = [a_1|a_2|\dots|a_m]$ be a $q \times m$ matrix of rank q with entries from $\mathbf{Z}_3$ (considered as a field). Similarly, let $B = [b_1|b_2|\dots|b_n]$ be an $r \times n$ matrix of rank r with entries from $\mathbf{Z}_2$ (considered as a field). Then $S \subseteq \mathbf{Z}_3^q \times \mathbf{Z}_2^r$ is said to* **cover** $\mathbf{Z}_3^q \times \mathbf{Z}_2^r$ **using** *A and B if*

$$\begin{aligned} \mathbf{Z}_3^q \times \mathbf{Z}_2^r &= \left\{ \begin{pmatrix} s_1 + pa_i \\ s_2 \end{pmatrix} : \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in S, p \in \mathbf{Z}_3, 1 \le i \le m \right\} \\ &\cup \left\{ \begin{pmatrix} s_1 \\ s_2 + pb_i \end{pmatrix} : \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in S, p \in \mathbf{Z}_2, 1 \le i \le n \right\} \end{aligned}$$

Theorem 4.2 *If S covers $\mathbf{Z}_3^q \times \mathbf{Z}_2^r$ using A and B then*

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

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

Proof. For any $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbf{Z}_3^m \times \mathbf{Z}_2^n$ we have $\begin{pmatrix} Ax_1 \\ Bx_2 \end{pmatrix} \in \mathbf{Z}_3^q \times \mathbf{Z}_2^r$

so we can find $\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \in S$ to give either

$$\begin{pmatrix} Ax_1 \\ Bx_2 \end{pmatrix} = \begin{pmatrix} s_1 + pa_i \\ s_2 \end{pmatrix}, p \in \mathbf{Z}_3, 1 \leq i \leq m$$

or

$$\begin{pmatrix} Ax_1 \\ Bx_2 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 + pb_i \end{pmatrix}, p \in \mathbf{Z}_2, 1 \leq i \leq n$$

In the first case, with e_i the i^{th} basis vector in $\mathbf{Z}_3^m$, we have

$$\begin{pmatrix} A(x_1 - pe_i) \\ Bx_2 \end{pmatrix} \in S$$

so

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - p \begin{pmatrix} e_i \\ 0 \end{pmatrix} \in W$$

thus $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is dominated by W in $F_{m,n}$.

In the other case, with e_i the i^{th} basis vector in $\mathbf{Z}_2^n$, we have

$$\begin{pmatrix} Ax_1 \\ B(x_2 - pe_i) \end{pmatrix} \in S$$

so

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - p \begin{pmatrix} 0 \\ e_i \end{pmatrix} \in W$$

thus $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is dominated by W in $F_{m,n}$.

Thus putting the two cases together, we see that W is a dominating set of $F_{m,n}$. Elementary linear algebra yields that its size is $|S|3^{m-q}2^{n-r}$. ■

If we define $G = \mathbf{Z}_3^q \times \mathbf{Z}_2^r$ and $H = \{\pm a_1, \pm a_2, \ldots, \pm a_m, b_1, b_2, \ldots, b_n\}$ then we see that S is simply a dominating set in the Cayley graph $X = X(G, H)$. The graph X has the same number of vertices as $F_{q,r}$ but is denser. By standard results on automorphisms of Cayley graphs we may assume without loss of generality that $a_i = e_i$ for $1 \leq i \leq q$ and that $b_i = e_i$ for $1 \leq i \leq r$, so A and B both consist of full rank leading identity submatrices together with some additional columns. Then X is actually equal to $F_{q,r}$ with some extra edges determined by these additional columns.

The power of this construction arises from the fact that it enables us to search for dominating sets in smaller and denser graphs, where tabu search works well. Any dominating set found in X can immediately be "blown up" to a dominating set for $F_{m,n}$. Of course this procedure cannot find all dominating sets of $F_{m,n}$ because they may not all be of this form, so in particular we may miss the minimum dominating set.

To use this construction to find a dominating set of $F_{m,n}$ we need to choose $q \leq m$ and $r \leq n$ and then the matrices A and B. Then the graph X is constructed, and dominating sets in X are found using the tabu search algorithm. In general there are several different choices for the matrices A and B and hence several possibilities for the graph X. There is no *a priori* reason to favour one of the choices over the others, and it seems difficult to justify any particular selection strategy that involves examining only a subset of the possibilities (although such a strategy would be extremely useful).

In order to construct all the possibilities for the matrix A with as few repetitions as possible, we recall the following result on automorphisms of Cayley graphs. If there is an automorphism σ of the group G such that $H_1^\sigma = H_2$ then the Cayley graphs $X(G, H_1)$ and $X(G, H_2)$ are isomorphic. Now the automorphism group of $V = (GF(3)^q, +)$ contains the general linear group $GL(q, 3)$. Therefore we must find orbits of this group on inverse-closed subsets of vectors H. As H is inverse-closed, we can identify the two vectors $\pm a_i$ with the 1-dimensional subspace $\langle a_i \rangle$. The group $GL(q, 3)$ acting on the 1-dimensional subspaces of V is the projective group $PGL(q, 3)$. Using the group theory package CAYLEY, we constructed this projective group and found its orbits on the m-sets of projective points (where we can assume that such a set contains the points represented by the standard basis vectors). For each such m-set, the matrix A is obtained by taking the m corresponding vectors as the columns of A. A similar process using the projective group $PGL(r, 2)$ yields all the possibilities for B.

5 Improved upper bounds

In this section we give the results of our computations. The following tables contains the sizes of the smallest dominating sets for all the graphs $F_{m,n}$ where $m + n \leq 13$.

For reasons of space we separate out the results for the hypercubes $F_{0,b}$ into Table 1 and the results for the "pure" football pool graphs into Table 2.

Our results improved on two values given in Hämäläinen and Rankinen [7] ($F_{0,9}$ and $F_{9,0}$) but the result for $F_{0,9}$ was independently matched by Östergård and Hämäläinen [11] and the result for $F_{9,0}$ was improved further by Östergård [12]. The improved results for $F_{11,0}$ and $F_{12,0}$ are from Östergård [13].

Table 3 contains the results for the remaining "mixed" graphs. Each entry consists either of 1 or 2 numbers — the top number is the value of the dominating set given in the tables of Hämäläinen and Rankinen [7], and the lower values are the currently best known bounds. Each improved value is preceded by a superscript in the range $\{1, 2, 3\}$, which has the following meaning:

(1) Bound due to this paper.
(2) Bound due to Östergård and Hämäläinen [11] duplicated independently.
(3) Bound due to Östergård and Hämäläinen [11].

We note that completely new bounds have therefore been found for $K(4,4)$, $K(4,5)$, $K(6,5)$, $K(7,6)$, $K(8,5)$ and $K(9,3)$.

6 Conclusions and Future Work

The results given in the previous section indicate clearly the rapid progress in improving the upper bounds for this class of graphs. It is of particular interest to note the large number of cases, mainly for the smaller values of $K(t, b)$, for which the upper bounds found by our tabu search algorithm match those given by Östergård and Hämäläinen [11] which were mainly found by combinatorial construction. This provides good circumstantial evidence that the tabu search algorithm is working well for those smaller values, though the table clearly shows the performance falling away for the larger values, where it produces inferior solutions.

Our implementation of tabu search is very straightforward, and as indicated throughout the text, there are several areas where further development and fine-tuning of the algorithm are indicated. In particular, some candidate list strategy should be implemented, with the results of Östergård [12] giving promising results in this direction. Experimentation with longer-term strategies, intensification and diversification should also yield improvements in performance. We believe that this problem provides an excellent test-bed for research into tabu search techniques, with the extensive tables of upper bounds providing readily available performance benchmarks.

For future work on the football pool problem itself, Östergård informs us (private commu-

b	1	2	3	4	5	6	7	8	9	10	11	12	13
$K(0,b)$	1	2	2	4	7	12	16	32	62	120	192	380	736

Table 1: Dominating sets for hypercubes

t	1	2	3	4	5	6	7	8	9	10	11	12	13
$K(t,0)$	1	3	5	9	27	73	186	486	1341	3645	9477	27702	59049

Table 2: Dominating sets for football pool graphs

b t	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	6	8	16	24	48	84	162 ^{2}160	288 ^{3}284	548	1024
2	4	6	12	20	36	64	126 ^{3}124	234 ^{2}232	428 ^{3}408	768	1536 ^{2}1504	
3	9	16	24	48	92	176 ^{3}171	332 ^{2}312	576	1152 ^{3}1080	2304 ^{3}2016		
4	18	36	72	132 ^{1}128	240 ^{1}238	432	864	1296	2592			
5	54	96	168	336 ^{2}324	648 ^{2}639	1296 ^{2}1188	1944	3888				
6	132	252	486 ^{2}468	864	1728 ^{1}1620	2916	5832					
7	372 ^{2}333	729 ^{2}648	1296	2592 ^{3}2304	4374	8748 ^{1}8586						
8	972	1944 ^{3}1728	3888 ^{3}3456	6480	12960 ^{1}12879							
9	2916 ^{2}2592	5832 ^{3}4860	9720 ^{1}9639	19440 ^{2}17496								
10	7290 ^{3}7047	14580 ^{3}13122	29160 ^{3}25192									
11	21870 ^{3}18894	39366 ^{3}37788										
12	59049 ^{3}52488											

Table 3: Upper bounds on sizes of dominating sets for the graphs $F_{t,b}$

nication) that the following bounds have been claimed in an Italian magazine, but the codes cannot be traced: $K(0,13) \leq 730$, $K(7,6) \leq 8352$, $K(8,4) \leq 6408$. Verifying these values would be a good initial aim for further work.

7 The actual codes

In this section we give explicitly the six codes for which our bounds are better than any previously known. In order to maintain consistency we use the compressed notation of Östergård and Hämäläinen [11]. Consider all the vectors in $\mathbf{Z}_3^m \times \mathbf{Z}_2^n$ to be listed in lexicographic order. To specify the codewords we merely give the "skip-vectors" which are the number of words between consecutive codewords. Codes that arise from the matrix construction are specified by giving the appropriate codewords in $\mathbf{Z}_3^q \times \mathbf{Z}_2^r$ in compressed notation together with the additional column vectors forming the matrices A and B.

Explicit listings of these codes and any further improvements can be obtained via anonymous ftp from `ftp.cs.uwa.edu.au` in the directory `pub/graphs/pools`.

7.1 $K(4,4) \leq 128$

This code was found by direct tabu search.

11, 0, 5, 2, 33, 17, 10, 5, 5, 29, 4, 9, 8, 24, 0, 8, 3, 19, 1, 8, 1, 16, 5, 1, 0, 1, 27, 10, 22, 10, 3, 6, 3, 3, 2, 3, 19, 30, 3, 6, 3, 8, 39, 0, 4, 0, 0, 0, 4, 0, 1, 1, 4, 3, 9, 8, 22, 8, 1, 3, 2, 1, 24, 9, 6, 5, 35, 10, 8, 2, 10, 22, 19, 0, 7, 5, 8, 9, 25, 8, 26, 0, 12, 20, 3, 2, 12, 2, 21, 0, 33, 1, 18, 9, 11, 13, 12, 5, 13, 8, 21, 2, 3, 1, 3, 6, 3, 24, 25, 3, 2, 3, 3, 29, 6, 9, 3, 4, 27, 1, 0, 1, 11, 10, 1, 8, 1, 32

7.2 $K(4,5) \leq 238$

This was found from $F_{4,4}$ using the additional vector $b_5 = (1,1,0,0)^t$.

30, 1, 10, 1, 3, 5, 2, 9, 7, 23, 14, 5, 12, 7, 6, 7, 2, 10, 10, 7, 13, 8, 8, 16, 2, 8, 21, 6, 10, 2, 1, 12, 57, 2, 1, 4, 19, 2, 9, 12, 10, 6, 2, 10, 53, 5, 4, 5, 16, 0, 9, 10, 17, 13, 8, 5, 17, 12, 4, 20, 1, 1, 6, 25, 8, 24, 7, 4, 8, 4, 0, 4, 5, 10, 31, 7, 4, 5, 18, 18, 1, 6, 13, 13, 13, 8, 9, 7, 7, 8, 17, 10, 4, 9, 24, 5, 9, 5, 22, 9, 25, 1, 0, 4, 0, 1, 22, 6, 4, 8, 16, 17, 2, 6, 14, 11, 4, 28, 8

7.3 $K(6,5) \leq 1620$

This was found from $F_{4,4}$ using the additional vectors $a_5 = (0,1,2,0)^t$, $a_6 = (1,0,0,2)^t$ and $b_5 = (1,1,0,1)^t$.

37, 3, 4, 4, 9, 6, 1, 29, 19, 9, 17, 19, 9, 52, 1, 2, 11, 6, 1, 13, 17, 8, 8, 9, 24, 19, 5, 52, 5, 2, 3, 8, 0, 52, 8, 9, 5, 12, 21, 15, 9, 21, 15, 2, 10, 2, 52, 0, 14, 4, 29, 1, 8, 4, 14, 25, 15, 2, 6, 2, 51, 6, 30, 1, 31, 6, 1, 25, 2, 2, 15, 25, 2, 2, 32, 2, 30, 5, 27, 6, 15, 1, 20, 5, 8, 21, 2, 6, 28, 6

7.4 $K(7,6) \leq 8586$

This was found from $F_{3,6}$ using the additonal vectors $a_4 = (1,0,1)^t$, $a_5 = (1,0,2)^t$, $a_6 = (1,1,2)^t$ and $a_7 = (1,2,2)^t$.

132, 6, 4, 14, 9, 0, 4, 2, 4, 5, 2, 11, 10, 0, 18, 9, 0, 24, 4, 6, 5, 0, 25, 72, 12, 27, 10, 9, 7, 0, 11, 3, 2, 3, 7, 9, 0, 61, 63, 13, 5, 4, 0, 4, 2, 5, 9, 10, 9, 74, 10, 27, 12, 78, 0, 9, 18, 0, 10, 11, 9, 0, 12, 12, 11, 14, 32, 46, 4, 6, 5, 0, 25, 4, 5, 13, 3, 0, 8, 1, 6, 11, 5, 97, 98, 4, 20, 0, 5, 6, 4, 6, 13, 12, 27, 10, 76, 10, 27, 12, 5, 28, 0, 8, 0, 10

As a consequence of the straightforward result that

$$K(t,b) \leq 3/2K(t-1,b+1)$$

we also get that

$$K(8,5) \leq 12879$$

7.5 $K(9,3) \leq 9639$

This was found from $F_{5,3}$ using the additonal vectors $a_6 = (0,0,0,1,1)^t$, $a_7 = (0,0,0,1,2)^t$, $a_8 = (0,1,1,0,0)^t$ and $a_9 = (0,1,2,0,0)^t$.

9, 21, 20, 13, 15, 17, 20, 21, 5, 10, 21, 4, 34, 12, 17, 2, 21, 21, 13, 28, 9, 22, 21, 4, 17, 10, 21, 12, 9, 10, 18, 17, 10, 13, 12, 27, 9, 24, 17, 17, 9, 26, 17, 15, 13, 26, 9, 9, 20, 9, 10, 47, 13, 12, 9, 25, 17, 18, 21, 1, 0, 1, 0, 0, 0, 0, 50, 30, 18, 5, 20, 26, 20, 1, 18, 19, 21, 22, 13, 18, 9, 26, 17, 15, 13, 26, 9, 63, 0, 0, 0, 0, 0, 0, 0, 9, 18, 13, 12, 35, 17, 18, 21, 3, 20, 9, 10, 34, 9, 18, 5, 33, 13, 20, 1, 38, 21, 22, 13

References

[1] Biggs, N. *Algebraic Graph Theory*, Cambridge University Press, 1974.

[2] Blokhuis, A and Lam, C.W.H. More coverings by rook domains. *Journal of Combinatorial Theory, Series A* **36**, 240–244, 1984.

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

[4] Davies, Rowan. Tabu search, graph domination and the football pool problem. Honours Thesis, Department of Computer Science, University of Western Australia, 1992.

[5] Garey, M and Johnson, D. *Computers and Intractability: a guide to the theory of NP-completeness.* W.H. Freeman, San Francisco, 1979.

[6] Glover, F. and Laguna, M. Tabu Search. In *Modern heuristic techniques for combinatorial problems* (editor: Colin Reeves), Halsted Press, New York, 1993.

[7] Hämäläinen, H and Rankinen, S. Upper bounds for football pool problems and mixed covering codes. *Journal of Combinatorial Theory, Series A* **56**, 84–95, 1991.

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

[9] van Laarhoven, P.J.M. Theoretical and computational aspects of simulated annealing Centrum voor Wiskunde en Informatica, Tract 57, 1988.

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

[11] Östergård, P.R.J and Hämäläinen, H.O. A new table of binary/ternary mixed covering codes. *Preprint.*

[12] Östergård, P.R.J. Constructing covering codes by tabu search. *Preprint*

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

[14] The football pool problem for 6 matches: A new upper bound obtained by simulated annealing. *Journal of Combinatorial Theory, Series A* **45**, 171–177, 1987.

Rowan Davies
Department of Computer Science
Carnegie-Mellon University
e-mail:Rowan_Davies@gs151.sp.cs.cmu.edu

Gordon F. Royle
Department of Computer Science
University of Western Australia
e-mail: gordon@cs.uwa.edu.au