

1-PERFECT CODES IN SIERPIŃSKI GRAPHS

SANDI KLAVŽAR, UROŠ MILUTINOVIĆ AND CIRIL PETR

Sierpiński graphs $S(n, k)$ generalise the Tower of Hanoi graphs—the graph $S(n, 3)$ is isomorphic to the graph H_n of the Tower of Hanoi with n disks. A 1-perfect code (or an efficient dominating set) in a graph G is a vertex subset of G with the property that the closed neighbourhoods of its elements form a partition of $V(G)$. It is proved that the graphs $S(n, k)$ possess unique 1-perfect codes, thus extending a previously known result for H_n . An efficient decoding algorithm is also presented. The present approach, in particular the proposed (de)coding, is intrinsically different from the approach to H_n .

1. INTRODUCTION

The study of perfect codes in graphs was initiated by Biggs [4] in 1973. At first, perfect codes were studied mostly in distance regular graphs and related classes of graphs, see, for instance, [3, 8, 24]. In the late 1980s Kratochvíl and his co-workers followed with a series of papers studying perfect codes in general graphs, (see [13, 14, 16, 18]). Most of the results up to 1991 are collected in the monograph [15].

From the algorithmic point of view, several NP-complete results were obtained by Kratochvíl and Křivánek [17, 15, 16]. In particular, for any fixed t it is NP-complete to decide whether a graph has a t -perfect code ([15, Theorem 7.0.1]). The special case of 1-perfect codes was independently proved to be NP-complete by Bange, Barkauskas, and Slater [1] and by Fellows and Hoover [7]. Recently, yet another proof for 1-perfect codes appeared in [5, 6]. The 1-perfect code problem remains NP-complete on k -regular graphs (for any fixed $k \geq 3$) ([15, Theorem 7.2.2]), planar 3-regular graphs ([15, Theorem 7.2.4]) and planar graphs of maximum degree three ([7]), as well as on bipartite and chordal graphs ([23]). For additional information on complexity results we refer to [10, 15].

On a positive side, Fellows and Hoover [7] showed that the problem of existence of a 1-perfect code in a tree can be solved in linear time, while Kratochvíl, Manuel and Miller [19] proved that it is polynomial on interval graphs. Recently, Klostermeyer and Eschen [12] obtained efficient algorithms for finding 1-perfect codes in interval graphs

Received 21st December, 2001

The first two authors were supported by the Ministry of Education, Science and Sport of Slovenia under the grant 0101-504.

Copyright Clearance Centre, Inc. Serial-fee code: 0004-9727/02 \$A2.00+0.00.

and circular-arc graphs. Bange, Barkauskas, and Slater [1] gave a constructive characterisation of trees that contain 1-perfect codes, see also [2].

Recently, Cull, Li, and Nelson [5, 6, 20] considered perfect codes in the Tower of Hanoi graphs H_n . In [5] it is shown that the graphs H_n contain (essentially) unique 1-perfect codes. In addition, code vertices are described in a simple way that allows an easy decoding algorithm. These results are more or less repeated in [6]. Finally, in [20] it is proved that besides the 1-perfect codes and the trivial t -perfect codes (with either one code vertex or three code vertices) there are no other t -perfect codes in H_n .

A two parametric generalisation $S(n, k)$ ($n, k \geq 1$) of the Tower of Hanoi graphs was proposed in [11]. It was in particular shown that for any $n \geq 1$, the graph $S(n, 3)$ is isomorphic to the graph of the Tower of Hanoi with n disks. We call the graphs $S(n, k)$ *Sierpiński graphs* since their introduction was motivated by topological studies [21, 22] of Lipscomb's space, where it was shown that this space is a generalisation of the Sierpiński triangular curve (Sierpiński gasket).

In the next section we recall the concepts of the Sierpiński graphs and perfect codes in graphs, and give some basic facts about them. Then, in Section 3, we prove that any graph $S(n, k)$ contains (essentially) unique 1-perfect code, thus extending the corresponding result for $S(n, 3)$ from [5]. The proofs enable us to develop an efficient decoding algorithm that is presented in the last section.

2. PRELIMINARIES

For a graph $G = (V(G), E(G))$, the *distance* $d_G(u, v)$, or briefly $d(u, v)$, between vertices u and v , is defined as the number of edges on a shortest u, v -path. A set $C \subseteq V(G)$ is a t -code in G if $d(u, v) \geq 2t + 1$ for any two distinct vertices $u, v \in C$; t -codes are also known as $2t$ -packings. In addition, C is called a t -perfect code if for any $u \in V(G)$ there is exactly one $v \in C$ such that $d(u, v) \leq t$; 1-perfect codes are also called *efficient dominating sets*. Note that C is a 1-perfect code if and only if the closed neighbourhoods of its elements form a partition of $V(G)$.

A set D of vertices of a graph G is called *dominating* if every vertex $w \in V(G) - D$ is adjacent to some vertex $v \in D$. The *domination number* of a graph G , $\gamma(G)$, is the order of a smallest dominating set of G . The following result is well-known and has been independently established several times, see [9, Theorem 4.2].

PROPOSITION 2.1. *Let C be a 1-perfect code in a graph G . Then $|C| = \gamma(G)$.*

The above proposition implies that all 1-perfect codes in G have the same cardinality, a result that extends to all t -perfect codes, (see [15, Corollary 4.6]).

The graph $S(n, k)$ ($n, k \geq 1$) is defined on the vertex set $\{0, 1, \dots, k-1\}^n$, two different vertices $u = (i_1, i_2, \dots, i_n)$ and $v = (j_1, j_2, \dots, j_n)$ being adjacent if and only if $u \sim v$. The relation \sim is defined as follows: $u \sim v$ if there exists an $h \in \{1, 2, \dots, n\}$ such that

- (i) $\forall t, t < h \Rightarrow i_t = j_t,$
- (ii) $i_h \neq j_h,$
- (iii) $\forall t, t > h \Rightarrow i_t = j_h \text{ \& } j_t = i_h.$

In the rest of the paper we shall shortly write $i_1 i_2 \dots i_n$ for (i_1, i_2, \dots, i_n) . The graph $S(2, 4)$ is shown in Figure 1, while the relation \sim is schematically explained in Figure 2.

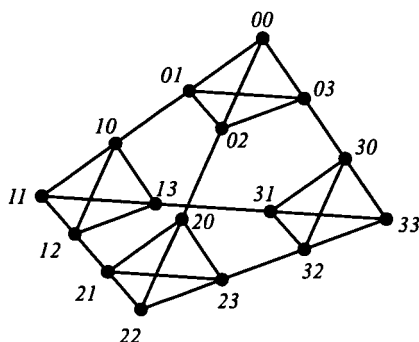


Figure 1: Sierpiński graph $S(2, 4)$

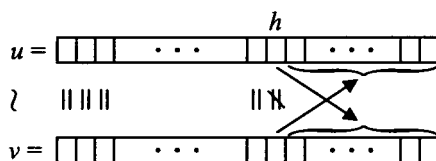


Figure 2: Visualisation of relation \sim

Clearly, $|S(n, k)| = k^n$. Let $i \in \{0, 1, \dots, k-1\}$, then the vertex $ii \dots i$ of $S(n, k)$ is called an *extreme vertex*. Clearly, $S(n, k)$ contains k extreme vertices, and they are of degree $k-1$, while all the other $k(k^{n-1} - 1)$ vertices are of degree k .

Let $r \in \{1, 2, \dots, n\}$ and let $i_1, i_2, \dots, i_r \in \{0, 1, \dots, k-1\}$. Then the subgraph of $S(n, k)$, induced by the vertices whose first r coordinates are $i_1 i_2 \dots i_r$, will be denoted

$$S(n, k; i_1, i_2, \dots, i_r).$$

Note that $S(n, k; i_1, i_2, \dots, i_r)$ is isomorphic to $S(n-r, k)$. In particular, the subgraph $S(n, k; i_1, i_2, \dots, i_{n-1})$ is isomorphic to $S(1, k)$ which is in turn isomorphic to the complete graph K_k , and $S(n, k; i_1, i_2, \dots, i_n)$ is the one vertex graph. Observe also that the vertices of $S(n, k)$ can be covered with the vertices of k^r disjoint subgraphs $S(n, k; i_1, i_2, \dots, i_r)$, obtained via all possible selections of i_1, i_2, \dots, i_r . We shall use the natural convention that the generic symbol $S(n, k; i_1, i_2, \dots, i_r)$ for $r = 0$ means the graph $S(n, k)$ itself.

A subgraph $S(n, k; i)$ contains one extreme vertex of $S(n, k)$, namely $ii \dots i$, and $k - 1$ vertices of the form $ijj \dots j$, $j \neq i$, that are, respectively, adjacent to vertices $jii \dots i$ of the subgraphs $S(n, k; j)$. Hence, in $S(n, k)$ there is exactly one edge between each pair of the k subgraphs $S(n, k; i)$, $i = 0, 1, \dots, k - 1$.

For any subgraph $S(n, k; i_1, i_2, \dots, i_r)$ of $S(n, k)$, $r = 1, \dots, n - 1$, a vertex v of the form $i_1 i_2 \dots i_r j j \dots j$, where j is taken $n - r$ times, will be called an *extreme vertex* of $S(n, k; i_1, i_2, \dots, i_r)$, as it is mapped to an extreme vertex of $S(n - r, k)$ by any isomorphism of $S(n, k; i_1, i_2, \dots, i_r)$ onto $S(n - r, k)$.

Let us introduce a useful shorthand notation that will simplify both formulation and visualisation of the coming statements. The full meaning and importance of it will become clear in Theorem 3.3, where we shall prove that the four cases described here are in fact the only possibilities.

Let C be a given 1-perfect code in $S(n, k)$ and $(i_1, \dots, i_r) \in \{0, \dots, k - 1\}^r$. Then $S(n, k; i_1, i_2, \dots, i_r)_*$ means:

“For $0 \leq \ell \leq k - 1$ all extreme vertices $i_1 i_2 \dots i_r \ell \dots \ell$ of $S(n, k; i_1, i_2, \dots, i_r)$ belong to C ”. In Theorem 3.3 we shall prove that this is possible only for even $m = n - r$.

$S(n, k; i_1, i_2, \dots, i_r)^j$ means:

“ $i_1 i_2 \dots i_r j \dots j$ is adjacent to a code vertex outside $S(n, k; i_1, i_2, \dots, i_r)$, and all other extreme vertices $i_1 i_2 \dots i_r \ell \dots \ell$, $0 \leq \ell \leq k - 1$, $\ell \neq j$, are adjacent to code vertices inside $S(n, k; i_1, i_2, \dots, i_r)$ ”. In Theorem 3.3 we shall prove that this is also possible only for even $m = n - r$.

$S(n, k; i_1, i_2, \dots, i_r)^*$ means:

“For $0 \leq \ell \leq k - 1$ all extreme vertices $i_1 i_2 \dots i_r \ell \dots \ell$ of $S(n, k; i_1, i_2, \dots, i_r)$ are adjacent to code vertices outside $S(n, k; i_1, i_2, \dots, i_r)$ ”. In Theorem 3.3 we shall prove that this is possible only for odd $m = n - r$.

$S(n, k; i_1, i_2, \dots, i_r)_j$ means:

“ $i_1 i_2 \dots i_r j \dots j$ belongs to C , and in $S(n, k; i_1, i_2, \dots, i_r)$ all other extreme vertices $i_1 i_2 \dots i_r \ell \dots \ell$, $0 \leq \ell \leq k - 1$, $\ell \neq j$, are adjacent to code vertices inside $S(n, k; i_1, i_2, \dots, i_r)$ ”. In Theorem 3.3 we shall prove that this is also possible only for odd $m = n - r$.

In Figure 3 the above four cases are schematically presented, while in Figure 4 several examples are depicted.

The usage of subscripts and superscripts was selected in such a way that subscripts are used in cases when $S(n, k; i_1, i_2, \dots, i_r)$ is “self-sufficient”, that is, when any vertex of $S(n, k; i_1, i_2, \dots, i_r)$ is either an element of C , or is adjacent to an element of C inside $S(n, k; i_1, i_2, \dots, i_r)$. A superscript is used when this is not the case.





$m = n - r$ is odd	 $S(n, k; i_1, i_2, \dots, i_r)_j$	 $S(n, k; i_1, i_2, \dots, i_r)^*$
$m = n - r$ is even	 $S(n, k; i_1, i_2, \dots, i_r)_*$	 $S(n, k; i_1, i_2, \dots, i_r)^j$

Figure 3: Possible relation of $S(n, k; i_1, i_2, \dots, i_r)$ to a given C

3. MAIN RESULTS

In this section we prove that for any $n \geq 1$ and $k \geq 1$ the graph $S(n, k)$ contains a unique 1-perfect code. More precisely, every 1-perfect code C contains at least one extreme vertex and, assuming that the extreme vertex $00 \dots 0$ belongs to C , we prove that C is unique. As $S(n, 1) = K_1$ for any n , there is nothing to be proved in this case. $S(n, 2)$ is isomorphic to the path on 2^n vertices and a simple argument gives the claim also in this case. As we already mentioned, the result for $k = 3$ was established in [5].

For a fixed 1-perfect code C in $S(n, k)$, $k \geq 2$, and for any extreme vertex of $S(n, k; i_1, i_2, \dots, i_r)$, there are exactly three possibilities:

1. v is adjacent to a code vertex inside $S(n, k; i_1, i_2, \dots, i_r)$;
2. v is adjacent to a code vertex outside $S(n, k; i_1, i_2, \dots, i_r)$;
3. v is a code vertex.

Denote the number of extreme vertices of $S(n, k; i_1, i_2, \dots, i_r)$ of type 1, 2, and 3 by a , b , and c , respectively. Note that $a = k - b - c$. In order to simplify the notation set $m = n - r$; note that $1 \leq m \leq n$. Recall that $S(n, k; i_1, i_2, \dots, i_r)$ for $r = 0$ means the graph $S(n, k)$ itself.

Lemma 3.1 and Proposition 3.2 demonstrate how simple number-theoretic techniques can give us a lot of information about 1-perfect codes in $S(n, k)$.

LEMMA 3.1. *Let C be a fixed 1-perfect code in $S(n, k)$, $k \geq 2$. Let $i_1 i_2 \dots i_r$ be any element of $\{0, 1, \dots, k-1\}^r$, where $r \in \{0, 1, \dots, n-1\}$. If b , c , and m have the same meaning as above, then*

$$(k+1) \mid (k^m - ck - b).$$

PROOF: Denote $H = S(n, k; i_1, i_2, \dots, i_r)$ and note first that H has k^m vertices. An extreme vertex of H has $k-1$ neighbours in H . Hence in the closed neighbourhoods in

H of extreme vertices of type 3 there are ck vertices of H . If an extreme vertex u of H is adjacent to a code vertex v of C , $v \notin H$, then u is clearly adjacent to no code vertex of H . The remaining $k^m - ck - b$ vertices of H are of degree k and hence they must be covered by closed neighbourhoods of size $k + 1$. It follows that $k^m - ck - b$ must be divisible by $k + 1$. \square

PROPOSITION 3.2. *Under the assumptions of Lemma 3.1, if m is even, then either $c = k$ and $b = 0$, or $b = c + 1$, and if m is odd, then either $c = 0$ and $b = k$, or $c = b + 1$.*

PROOF: Since $-1 \equiv k \pmod{(k+1)}$, it follows $(-1)^m \equiv k^m \pmod{(k+1)}$, and then $(-1)^m - ck - b \equiv k^m - ck - b \pmod{(k+1)}$. Using Lemma 3.1 we infer $(-1)^m - ck - b \equiv 0 \pmod{(k+1)}$, thus $(-1)^m + c - b \equiv c(k+1) \pmod{(k+1)}$. We conclude that $(-1)^m + c - b \equiv 0 \pmod{(k+1)}$.

CASE 1. m is even. Since $1 + c - b \equiv 0 \pmod{(k+1)}$, there is an integer ℓ , such that

$$b - c = 1 + (k+1)\ell.$$

From $0 \leq b + c \leq k$ it follows $0 \leq 2c + 1 + (k+1)\ell \leq k$.

In case $\ell \geq 1$ we get a contradiction from the right hand side inequality: $2c + 1 \leq k(1 - \ell) - \ell < 0$. In case $\ell \leq -2$, the left hand side inequality gives a contradiction: $2c + 1 \geq (k+1)\ell \geq 2(2k+1)$, hence $c > k$. Therefore, there are only two possibilities: $\ell = 0$, or $\ell = -1$.

$\ell = 0$ gives $b = c + 1$, and $\ell = -1$ gives $b = c + k$. Since $b \leq k$ and $c \geq 0$, the second case is possible only when $c = 0$ and $b = k$.

CASE 2. m is odd. In this case we get $-1 + c - b \equiv 0 \pmod{(k+1)}$, and therefore there is an integer ℓ , such that

$$c - b = 1 + (k+1)\ell.$$

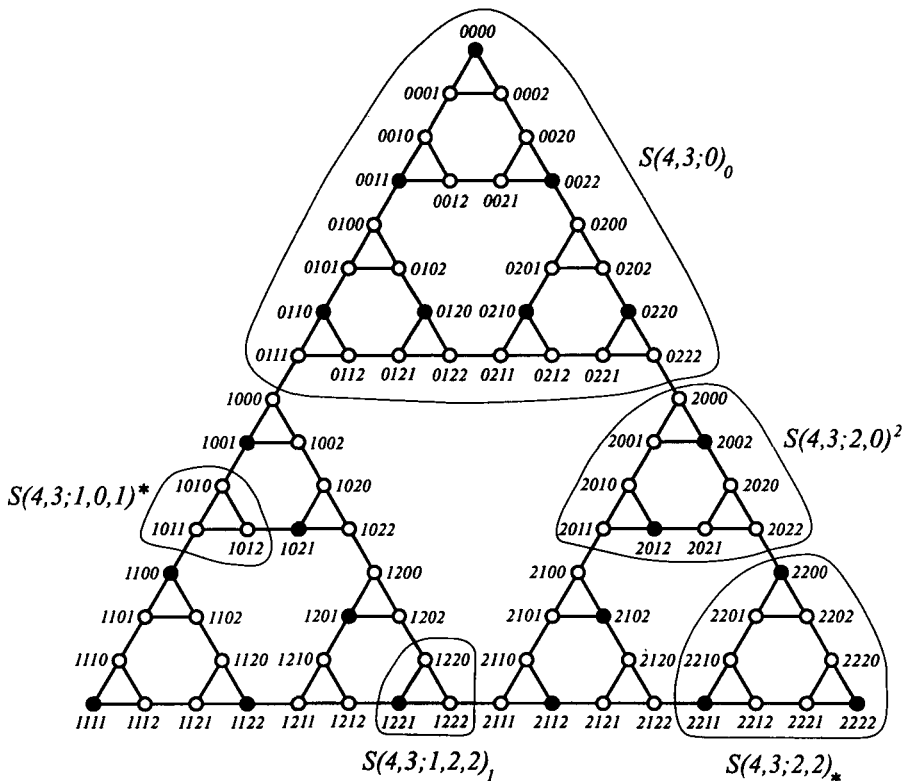
We now proceed as in the even case, except that we interchange b and c . \square

In fact, a stronger result than the one given in Proposition 3.2 can be proved:

THEOREM 3.3. *Under the assumptions of Lemma 3.1, if m is even, then either $c = k$ and $b = 0$, or $c = 0$ and $b = 1$, and if m is odd, then either $c = 0$ and $b = k$, or $c = 1$ and $b = 0$.*

PROOF: We are going to prove, by induction on $m = n - r \in \{1, \dots, n\}$, that for any subgraph $S(n, k; i_1, i_2, \dots, i_r)$ of $S(n, k)$ exactly one of the following holds true (the reader is advised to use Figure 4 to visualise these cases in the proof):

1. $m = n - r$ is even, $c = k$, $b = 0$, that is, $S(n, k; i_1, i_2, \dots, i_r)_*$;
2. $m = n - r$ is even, $c = 0$, $b = 1$, that is, $S(n, k; i_1, i_2, \dots, i_r)^j$;

Figure 4: Examples of subgraphs in $S(4,3)$

3. $m = n - r$ is odd, $c = 0$, $b = k$, that is, $S(n, k; i_1, i_2, \dots, i_r)^*$;
4. $m = n - r$ is odd, $c = 1$, $b = 0$, that is, $S(n, k; i_1, i_2, \dots, i_r)_j$.

For $m = 1$, since $S(n, k; i_1, i_2, \dots, i_{n-1})$ is isomorphic to the complete graph K_k , there are only two possibilities: the subgraph $S(n, k; i_1, i_2, \dots, i_{n-1})$ contains no code vertex, or it contains exactly one code vertex. In the first case all vertices are adjacent to code vertices outside the subgraph, hence $S(n, k; i_1, i_2, \dots, i_{n-1})^*$. In the second case let $i_1 i_2 \dots i_{n-1} j \in C$. Since all other vertices of the subgraph are adjacent to $i_1 i_2 \dots i_{n-1} j$, the subgraph is of type $S(n, k; i_1, i_2, \dots, i_{n-1})_j$.

In the rest of the proof we shall use the following abbreviations. Let $H = S(n, k; i_1, i_2, \dots, i_r)$ and $H(\ell) = S(n, k; i_1, i_2, \dots, i_r, \ell)$.

Let m be an even number, and suppose the claim holds for $m - 1$. Consider the subgraph H with $n - r = m$. For each of its subgraphs $H(\ell)$, $0 \leq \ell \leq k - 1$, either $H(\ell)^*$ or $H(\ell)_j$ holds by the inductive assumption.

Suppose first that for any ℓ we have $H(\ell)_j$ (for each ℓ for some j). Then all k code

vertices that are the extreme vertices of $H(\ell)$, $0 \leq \ell \leq k-1$, must be the extreme vertices of H . Indeed, otherwise there would be an extreme vertex of some $H(\ell)$ adjacent to a code vertex outside $H(\ell)$, and such does not exist. Thus in this case we have H_* . That also means that for each ℓ the corresponding j equals ℓ , that is, when H_* then $H(j)_j$, $0 \leq j \leq k-1$.

Suppose next that $H(i)^*$ holds for at least one subgraph $H(i)$ of H . Any two different subgraphs $H(i')$ and $H(i'')$ are joined by an edge. Hence they cannot simultaneously satisfy $H(i')^*$ and $H(i'')^*$, for otherwise one of them would contain an extreme vertex belonging to C . Thus in this case there is a unique $H(i)$ with $H(i)^*$ and $H(\ell)_j$ for $0 \leq \ell \leq k-1$, $\ell \neq i$ (for each ℓ for some j). Then the extreme vertices $i_1 i_2 \dots i_r i i_1 \dots i_r$ of $H(i)$ are code vertices adjacent to the extreme vertices of $H(\ell)$. Indeed, otherwise there would be an extreme vertex of $H(i)$ adjacent to a non-code vertex that is an extreme vertex of one of the $H(\ell)$, $0 \leq \ell \leq k-1$, $\ell \neq i$, contrary to the requirement that it must be adjacent to a code vertex outside $H(i)$ (note that it cannot be adjacent to more than one vertex outside $H(i)$). Thus in this case we have H^i , since the remaining extreme vertex of $H(i)$ must be adjacent to a code vertex outside H . Hence it is an extreme vertex of H , and the extreme vertex of H belonging to $H(i)$ is $i_1 i_2 \dots i_r i i_1 \dots i_r$. By this we have also proved that H^i implies $H(i)^*$ and $H(j)_i$, $j \neq i$, since the extreme vertex $i_1 i_2 \dots i_r i j \dots j$ of $H(i)$ is adjacent to the code vertex $i_1 i_2 \dots i_r j i \dots i$, which is an extreme vertex of $H(j)$.

Let now $m > 1$ be an odd number, and suppose the claim holds for $m-1$. Consider the subgraph H with $n-r=m$. Each of its subgraphs $H(\ell)$, $0 \leq \ell \leq k-1$, satisfies by the inductive assumption either $H(\ell)_*$ or $H(\ell)^j$.

Assume first that for any ℓ we have $H(\ell)^j$ (for each ℓ for some j). Then all k extreme vertices of $H(\ell)$, $0 \leq \ell \leq k-1$, which are adjacent to code vertices outside $H(\ell)$ (one for each ℓ), must be in fact adjacent to code vertices outside H , since inside H they are adjacent only to other extreme vertices of subgraphs $H(\ell)$, and none of them is a code vertex. Therefore, these k vertices must be extreme vertices of H , since only they may be adjacent to vertices outside H . That means that we have H^* . From the argument we also infer that H^* implies $H(i)^i$, $0 \leq i \leq k-1$.

Suppose next that $H(i)_*$ holds for at least one subgraph $H(i)$ of H . Then such a subgraph is clearly unique, for otherwise C is not a 1-perfect code. Hence let $H(i)$ satisfies $H(i)_*$ and let $H(\ell)$, for $0 \leq \ell \leq k-1$, $\ell \neq i$, satisfy $H(\ell)^j$ (for each ℓ for some j). Then $k-1$ extreme vertices of $H(i)$ must be adjacent precisely to those extreme vertices of subgraphs $H(\ell)$, for $0 \leq \ell \leq k-1$, $\ell \neq i$, which are adjacent to code vertices outside $H(\ell)$. Indeed, other extreme vertices of $H(\ell)$, $0 \leq \ell \leq k-1$, $\ell \neq i$, are adjacent to code vertices inside corresponding $H(\ell)$. That means that in this case we have H_i . Since any extreme vertex $i_1 i_2 \dots i_r i j \dots j$ of $H(i)$, $j \neq i$, is adjacent to $i_1 i_2 \dots i_r j i \dots i$, which is an extreme vertex of $H(j)$, we have also proved that H_i implies $H(i)_*$ and $H(j)^i$, $j \neq i$.

This completes the inductive proof. \square

Note that the above proof demonstrates that, with respect to C , the structure of the subgraphs $S(n, k; i_1, i_2, \dots, i_r)$ is strictly determined by the structure of the subgraphs $S(n, k; i_1, i_2, \dots, i_r, \ell)$, $0 \leq \ell \leq k-1$. Figures 5 and 6 show schematically how the graphs $S(n, k; i_1, i_2, \dots, i_r)$ are built from the subgraphs $S(n, k; i_1, i_2, \dots, i_r, \ell)$, $0 \leq \ell \leq k-1$, in all possible cases.

In Figure 5 the bottom row shows the only two possibilities when $m = n - r$ is even. The left hand side of the figure shows that when $S(n, k; i_1, i_2, \dots, i_r)_*$, then $S(n, k; i_1, i_2, \dots, i_r, i)_i$, $0 \leq i \leq k-1$, and the right hand side shows that when $S(n, k; i_1, i_2, \dots, i_r)^i$, then $S(n, k; i_1, \dots, i_r, i)^*$ and $S(n, k; i_1, \dots, i_r, j)_i$, for all $j \neq i$.

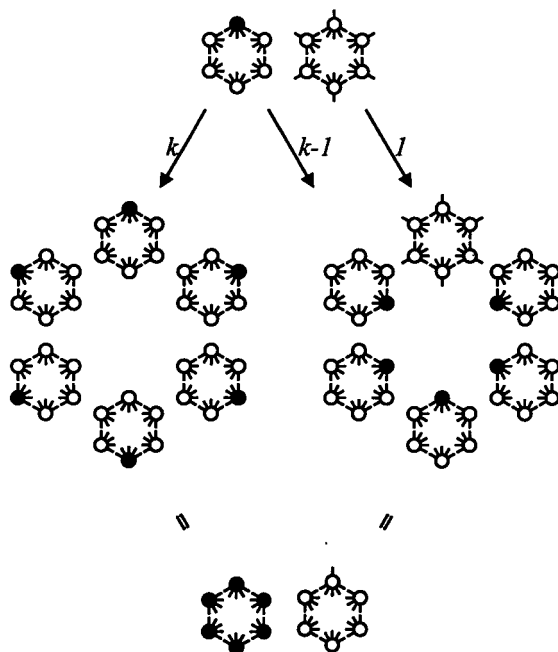


Figure 5: The structure of $S(n, k; i_1, i_2, \dots, i_r)$ when $m = n - r$ is even

In Figure 6 the bottom row shows the only two possibilities when $m = n - r$ is odd. The left hand side of the figure shows that if $S(n, k; i_1, \dots, i_r)_i$, then $S(n, k; i_1, \dots, i_r, i)_*$ and $S(n, k; i_1, \dots, i_r, j)_i$, for all $j \neq i$, while the right hand side shows that $S(n, k; i_1, \dots, i_r)^i$ implies $S(n, k; i_1, \dots, i_r, i)^i$, $0 \leq i \leq k-1$.

In the following theorem we include what we have just illustrated and proved while proving Theorem 3.3:

THEOREM 3.4. *Let C be a fixed 1-perfect code in $S(n, k)$, $k \geq 2$ and let $r \in \{0, 1, \dots, n-1\}$. Then:*

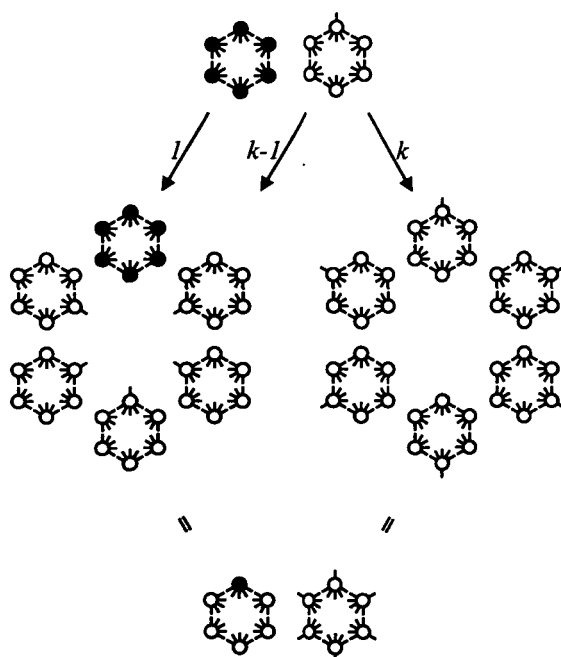


Figure 6: The structure of $S(n, k; i_1, i_2, \dots, i_r)$ when $m = n - r$ is odd

1. $S(n, k; i_1, \dots, i_r)_*$ implies $S(n, k; i_1, \dots, i_r, j)_j$, $0 \leq j \leq k - 1$;
2. $S(n, k; i_1, \dots, i_r)^i$ implies $S(n, k; i_1, \dots, i_r, i)^*$ and $S(n, k; i_1, \dots, i_r, j)_i$ for $0 \leq j \leq k - 1$, $j \neq i$;
3. $S(n, k; i_1, \dots, i_r)^*$ implies $S(n, k; i_1, \dots, i_r, i)^i$, $0 \leq i \leq k - 1$;
4. $S(n, k; i_1, \dots, i_r)_i$ implies $S(n, k; i_1, \dots, i_r)_*$ and $S(n, k; i_1, \dots, i_r, j)^i$ for $0 \leq j \leq k - 1$, $j \neq i$.

Cases 1 and 2 are possible when $m = n - r$ is even, and cases 3 and 4 are possible when m is odd.

Understanding the structure of the subgraphs $S(n, k; i_1, \dots, i_r)$ with respect to a given 1-perfect code will enable us to prove our main result on the existence and uniqueness of 1-perfect codes in graphs $S(n, k)$, as well as to describe a decoding algorithm and prove its correctness. Before that, we state a simple consequence of Theorem 3.3.

COROLLARY 3.5. *Let C be a fixed 1-perfect code in $S(n, k)$, $k \geq 2$. If n is odd, C contains exactly one extreme vertex $jj \dots j$, and if n is even, C contains all extreme vertices of $S(n, k)$.*

PROOF: Clearly, for the whole graph $S(n, k)$ we have $b = 0$. Hence Theorem 3.3 implies $c = k$ if m is even, and $c = 1$ if m is odd. \square

THEOREM 3.6. *For any $n \geq 1$ and any $k \geq 1$ the graph $S(n, k)$ has a unique 1-perfect code, if n is even, and there are exactly k 1-perfect codes, if n is odd. More precisely, let n be odd and let $jj \dots j$ be the extreme vertex of $S(n, k)$ belonging to a 1-perfect code C . Then C is unique.*

PROOF: We first show the uniqueness claims by repeated applications of Theorem 3.4.

Let n be even and let C be a 1-perfect code in $S(n, k)$. By Corollary 3.5 all extreme vertices of $S(n, k)$ belong to C , therefore $S(n, k)_*$ holds. Recall that $S(n, k)$ is induced by the (disjoint) subgraphs $S(n, k; i)$. Case 1 of Theorem 3.4 implies that $S(n, k; i)_*$ holds for all i . $S(n, k)$ is also induced by the subgraphs $S(n, k; i, j)$, so using Case 4 of the same theorem we get $S(n, k; i, i)_*$ and $S(n, k; i, j)^i$, for $j \neq i$. Moreover, for each of the subgraphs it is uniquely determined which property it satisfies. Applying Cases 1 and 2 and considering $S(n, k)$ as the graph induced by the subgraphs $S(n, k; i, j, \ell)$, we find that for each of them it is uniquely determined whether it satisfies $S(n, k; i, j, \ell)_p$ or $S(n, k; i, j, \ell)^*$. Alternating applications of Cases 3 and 4, and Cases 1 and 2 and considering $S(n, k)$ as the graph induced by the subgraphs $S(n, k; i_1 \dots i_{n-1})$, we conclude that for each of them it is uniquely determined whether it satisfies $S(n, k; i_1, i_2, \dots, i_{n-1})^*$ or $S(n, k; i_1, i_2, \dots, i_{n-1})_q$ (for each choice of indices i_1, i_2, \dots, i_{n-1} the property holding true is also uniquely determined, as well as the index q in the second case). If $S(n, k; i_1, i_2, \dots, i_{n-1})^*$, then $S(n, k; i_1, i_2, \dots, i_{n-1})$ contains no code vertex, and $i_1 i_2 \dots i_{n-1} q$ is the only code vertex of $S(n, k; i_1, i_2, \dots, i_{n-1})$ if $S(n, k; i_1, i_2, \dots, i_{n-1})_q$ (recall that these graphs are isomorphic to the complete graphs). Hence, vertices belonging to C are uniquely determined.

Proof for the odd case is analogous, except that we start from $S(n, k)_j$ and apply Case 4 Theorem 3.4 in the first step.

We next show that the only possible candidates for 1-perfect codes described above indeed exist. We proceed by induction on n . Simultaneously we prove by the same induction that for any even n , there is a subset $D_n \subseteq V(S(n, k))$, such that any vertex $v \neq 00 \dots 0$ of $S(n, k)$ either belongs to D_n or is adjacent to exactly one vertex from D_n , and that $00 \dots 0$ is neither in D_n nor adjacent to any vertex of D_n . Also, by the same induction we prove that for any odd n , there is a subset $D_n \subseteq V(S(n, k))$, such that any vertex $v \neq ii \dots i$, $0 \leq i \leq k-1$, either belongs to D_n or is adjacent to exactly one vertex from D_n , and that any vertex $ii \dots i$, $0 \leq i \leq k-1$, is neither in D_n nor adjacent to any vertex of D_n .

Obviously any one-element subset of $V(S(1, k))$ is a 1-perfect code in $S(1, k)$. It is also clear that the set of all extreme vertices in $V(S(2, k))$ is a 1-perfect code in $S(2, k)$, see Figure 1. Also, $D_1 = \emptyset$ and $D_2 = \{10, 20, \dots, (k-1)0\}$ satisfy all the requirements of the additional claims for $S(1, k)$ and $S(2, k)$, respectively.

Let $n > 2$ be an even number. Then $n-1 > 1$ is an odd number, and by inductive assumption there is exactly one 1-perfect code $C \subseteq V(S(n-1, k))$ containing $00 \dots 0$.

For each i , $0 \leq i \leq k-1$, let $f_i : V(S(n-1, k)) \rightarrow V(S(n, k))$ be an isomorphism from $S(n-1, k)$ onto the subgraph $S(n, k; i)$ of $S(n, k)$ such that $f_i(00 \dots 0) = ii \dots i$. Then $f_0(C) \cup \dots \cup f_{k-1}(C)$ is a 1-perfect code in $S(n, k)$. Namely, any vertex from $S(n, k; i)$ either belongs to $f_i(C)$, or is adjacent to exactly one vertex from $f_i(C)$, and is not adjacent to any of the vertices from $f_j(C)$ for $j \neq i$. This follows from the observation that the only edge connecting the subgraph $f_i(C)$ with the subgraph $f_j(C)$ is the edge between $ijj \dots j$ and $jii \dots i$, and that an extreme vertex of $S(n-1, k)$, different from $00 \dots 0$, is mapped by f_j to $jii \dots i$, and it does not belong to C .

Let $D_{n-1} \subseteq V(S(n-1, k))$ be a set such that any vertex of $S(n-1, k)$ but $ii \dots i$, $0 \leq i \leq k-1$, either belongs to D_{n-1} , or is adjacent to exactly one vertex from D_{n-1} . For $i = 0, \dots, k-1$ let $g_i : V(S(n-1, k)) \rightarrow V(S(n, k))$ be an isomorphism from $S(n-1, k)$ onto the subgraph $S(n, k; i)$ of $S(n, k)$ such that $g_i(00 \dots 0) = i0 \dots 0$. Then any vertex of $S(n, k)$ but $00 \dots 0$ either belongs to $g_0(D_{n-1}) \cup g_1(C) \cup \dots \cup g_{k-1}(C)$, or is adjacent to exactly one of its vertices. This follows by isomorphism properties for all but extreme vertices of $S(n, k; 0)$. The claim for any vertex of the form $0i \dots i$, $1 \leq i \leq k-1$ follows from the fact that $0i \dots i$ is adjacent to $i0 \dots 0$, and $i0 \dots 0 \in g_i(C)$. Since $00 \dots 0$ is not adjacent to any vertex not in $g_0(D_{n-1})$, it also satisfies the required properties.

Let $n > 2$ be an odd number. Then $n-1 > 1$ is an even number, and by inductive assumption there is exactly one 1-perfect code $C \subseteq V(S(n-1, k))$. Also, there is a subset $D_{n-1} \subseteq V(S(n-1, k))$, such that any vertex of $S(n-1, k)$ but $00 \dots 0$ either belongs to D_{n-1} , or is adjacent to exactly one vertex from D_{n-1} . For $i = 0, \dots, k-1$ let $g_i : V(S(n-1, k)) \rightarrow V(S(n, k))$ be defined as above. Then any vertex of $S(n, k)$ either belongs to $g_0(C) \cup g_1(D_{n-1}) \cup \dots \cup g_{k-1}(D_{n-1})$, or is adjacent to exactly one of its vertices. This again follows from the fact that $0i \dots i$ is adjacent to $i0 \dots 0$.

Finally, for $i = 1, \dots, k-1$ let $f_i : V(S(n-1, k)) \rightarrow V(S(n, k))$ be defined as above. Then $f_0(D_{n-1}) \cup \dots \cup f_{k-1}(D_{n-1})$ satisfies the requirement, that each vertex of $S(n, k)$, except the extreme vertices, either belongs to this set, or is adjacent to exactly one vertex from it. This follows from the isomorphism embedding properties, and from the observation that no extreme vertex belongs to D_{n-1} , which is easily proved by the same induction, if this claim is added to the list of claims proved by the induction. \square

The obtained results allow us to calculate $\gamma(S(n, k))$ as follows. Let $r = 0, 1, \dots, n-1$, $m = n-r$. If m is odd, let

$$\begin{aligned} c(m) &= \left| C \cap V(S(n, k; i_1, i_2, \dots, i_r)) \right|, & \text{if } S(n, k; i_1, i_2, \dots, i_r)_j, \text{ and} \\ b(m) &= \left| C \cap V(S(n, k; i_1, i_2, \dots, i_r)) \right|, & \text{if } S(n, k; i_1, i_2, \dots, i_r)^*. \end{aligned}$$

If m is even, let

$$\begin{aligned} c(m) &= \left| C \cap V(S(n, k; i_1, i_2, \dots, i_r)) \right|, & \text{if } S(n, k; i_1, i_2, \dots, i_r)_*, \text{ and} \\ b(m) &= \left| C \cap V(S(n, k; i_1, i_2, \dots, i_r)) \right|, & \text{if } S(n, k; i_1, i_2, \dots, i_r)^j. \end{aligned}$$

Introduction of this notation simplifies the formulation of the following corollary, and is in the same time justified by it, since the inductive proof of the corollary shows that the cardinalities of the above intersections do not depend on the choice of i_1, i_2, \dots, i_r , and j .

COROLLARY 3.7. For any $\ell \geq 1$,

$$\begin{aligned} c(2\ell - 1) &= \frac{k^{2\ell-1} + 1}{k + 1}, & b(2\ell - 1) &= k \frac{k^{2\ell-2} - 1}{k + 1}, \\ c(2\ell) &= k \frac{k^{2\ell-1} + 1}{k + 1}, & b(2\ell) &= \frac{k^{2\ell} - 1}{k + 1}. \end{aligned}$$

PROOF: Obviously $c(1) = 1$, $b(1) = 0$. From Theorem 3.4 we obtain the recursive formulas

$$\begin{aligned} c(2\ell) &= kc(2\ell - 1), \\ b(2\ell) &= b(2\ell - 1) + (k - 1)c(2\ell - 1), \\ c(2\ell + 1) &= c(2\ell) + (k - 1)b(2\ell), \\ b(2\ell + 1) &= kb(2\ell), \end{aligned}$$

for $\ell \geq 1$. From these we obtain the following inductive proof:

$$\begin{aligned} c(2\ell) &= kc(2\ell - 1) = k \frac{k^{2\ell-1} + 1}{k + 1}; \\ b(2\ell) &= b(2\ell - 1) + (k - 1)c(2\ell - 1) = k \frac{k^{2\ell-2} - 1}{k + 1} + (k - 1) \frac{k^{2\ell-1} + 1}{k + 1} = \frac{k^{2\ell} - 1}{k + 1}; \\ c(2\ell + 1) &= c(2\ell) + (k - 1)b(2\ell) = k \frac{k^{2\ell-1} + 1}{k + 1} + (k - 1) \frac{k^{2\ell} - 1}{k + 1} = \frac{k^{2\ell+1} + 1}{k + 1}; \\ b(2\ell + 1) &= kb(2\ell) = k \frac{k^{2\ell} - 1}{k + 1}. \end{aligned}$$

□

Applying Proposition 2.1 and Corollary 3.7 we get:

THEOREM 3.8. For any $n \geq 1$ and any $k \geq 1$,

$$\gamma(S(n, k)) = \begin{cases} k \frac{k^{n-1} + 1}{k + 1}; & n \text{ even,} \\ \frac{k^n + 1}{k + 1}; & n \text{ odd.} \end{cases}$$

4. THE DECODING ALGORITHM

Let C be a given 1-perfect code in $S(n, k)$ and let $v = v_1 v_2 \dots v_n$ be a given vertex of $S(n, k)$. The algorithm presented below decides whether v is a code vertex, and if it is not, computes the adjacent code vertex v' . The algorithm is presented in a pseudo code to make it as clear as possible.

```

1. subroutine  $S(n, k; v_1, v_2, \dots, v_r)_j$  begin
2.   if  $(r = n - 1)$ 
3.     then if  $(v_n = j)$ 
4.       then  $v \in C$ 
5.       else  $v' = (v_1, v_2, \dots, v_{n-1}, j)$ 
6.     else if  $(v_{r+1} = j)$ 
7.       then call  $S(n, k; v_1, v_2, \dots, v_{r+1})_*$ 
8.       else call  $S(n, k; v_1, v_2, \dots, v_{r+1})^j$ 
9.   end
10. subroutine  $S(n, k; v_1, v_2, \dots, v_r)_*$  begin
11.   if  $(r = n - 1)$ 
12.     then  $v = v_1 \dots v_s i j \dots j, j \neq i, v' = v_1 \dots v_s j i \dots i$ 
13.     else call  $S(n, k; v_1, v_2, \dots, v_{r+1})^{v_{r+1}}$ 
14.   end
15. subroutine  $S(n, k; v_1, v_2, \dots, v_r)_*$  begin
16.   call  $S(n, k; v_1, v_2, \dots, v_{r+1})_{v_{r+1}}$ 
17. end
18. subroutine  $S(n, k; v_1, v_2, \dots, v_r)^j$  begin
19.   if  $(v_{r+1} = j)$ 
20.     then call  $S(n, k; v_1, v_2, \dots, v_{r+1})_*$ 
21.     else call  $S(n, k; v_1, v_2, \dots, v_{r+1})_j$ 
22.   end
23. main_program begin
24.   if  $(n \text{ odd})$ 
25.     then call  $S(n, k)_j$ 
26.     else call  $S(n, k)_*$ 
27. end

```

The algorithm first decides whether n is odd or even. In the first case we must know which extreme vertex $jj \dots j$ belongs to C and at line 25 the algorithm calls the subroutine $S(n, k; v_1, v_2, \dots, v_r)_j$ with $r = 0$. If n is even, the algorithm calls the subroutine $S(n, k; v_1, v_2, \dots, v_r)_*$ with $r = 0$ at line 26. The flow of the algorithm is graphically presented in Figure 7.

The algorithm ends when $r = n - 1$. This happens either in the subroutine $S(n, k; v_1, v_2, \dots, v_r)_j$ or in $S(n, k; v_1, v_2, \dots, v_r)_*$. In the first case, if $v_n = j$, vertex v is determined to be a code vertex (at line 4), else its neighbour $v_1 v_2 \dots v_{n-1} j$ is a code vertex (at line 5). In the second case v is not a code vertex, neither is a code vertex any of its neighbours in the complete graph $S(n, k; v_1, v_2, \dots, v_{n-1})$. Since we ended in the subroutine $S(n, k; v_1, v_2, \dots, v_{n-1})_*$, vertex v is adjacent to a unique code vertex v'

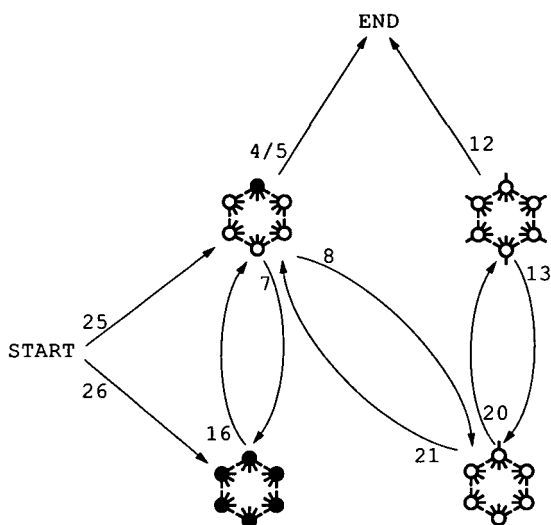


Figure 7: State transition diagram

outside of the complete graph (at line 12). The vertex v' is determined in the following way. If $v = v_1 \dots v_s i j \dots j$, $j \neq i$, then $v' = v_1 \dots v_s j i \dots i$. To compute the value of s , we have two possibilities. Either we keep track of the last index r for which two consecutive components v_{r-1} and v_r differed during the flow of the algorithm, or we separately compute such value at the very end of the algorithm, namely at line 12.

This completes our description of the algorithm. The fact that it works as claimed follows immediately from (the proofs of) Theorems 3.6 and 3.4.

We conclude the paper by noting that the decoding algorithm is optimal, that is, it is linear in the length n of an input word.

REFERENCES

- [1] D.W. Bange, A.E. Barkauskas and P.J. Slater, 'Efficient dominating sets in graphs', in *Applications of Discrete Mathematics, Clemson, 1986* (SIAM, Philadelphia, 1988), pp. 189–199.
- [2] D.W. Bange, A.E. Barkauskas and L.H. Clark, 'The number of orientations of a tree admitting an efficient dominating set', *Ars Combin.* **45** (1997), 201–207.
- [3] E. Bannai, 'Codes in bipartite distance-regular graphs', *J. London Math. Soc.* **16** (1977), 197–202.
- [4] N. Biggs, 'Perfect codes in graphs', *J. Combin. Theory Ser. B* **15** (1973), 289–296.
- [5] P. Cull and I. Nelson, 'Error-correcting codes on the Towers of Hanoi graphs', *Discrete Math.* **208/209** (1999), 157–175.
- [6] P. Cull and I. Nelson, 'Perfect codes, NP-completeness, and Towers of Hanoi graphs', *Bull. Inst. Combin. Appl.* **26** (1999), 13–38.

- [7] M.R. Fellows and M.N. Hoover, 'Perfect domination', *Australas. J. Combin.* **3** (1991), 141–150.
- [8] P. Hammond, 'On the nonexistence of perfect and nearly perfect codes', *Discrete Math.* **39** (1982), 105–109.
- [9] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, *Fundamentals of domination in graphs* (Marcel Dekker, New York, 1998).
- [10] S.T. Hedetniemi, A.A. McRae and D.A. Parks, 'Complexity results', in *Domination in Graphs: Advanced Topics*, (T.W. Haynes, S.T. Hedetniemi and P.J. Slater, Editors) (Marcel Dekker, New York, 1998), pp. 233–269.
- [11] S. Klavžar and U. Milutinović, 'Graphs $S(n, k)$ and a variant of the Tower of Hanoi problem', *Czechoslovak Math. J.* **47** (1997), 95–104.
- [12] W.F. Klostermeyer and E.M. Eschen, 'Perfect codes and independent dominating sets', *Congr. Numer.* **142** (2000), 7–28.
- [13] J. Kratochvíl, 'Perfect codes over graphs', *J. Combin. Theory Ser. B* **40** (1986), 224–228.
- [14] J. Kratochvíl, 'Perfect codes and two-graphs', *Comment. Math. Univ. Carolin.* **30** (1989), 755–760.
- [15] J. Kratochvíl, *Perfect codes in general graphs*, Rozprawy Československé Akad. Věd Řada Mat. Přírod Věd **7** (Akademia, Praha, 1991).
- [16] J. Kratochvíl, 'Regular codes in regular graphs are difficult', *Discrete Math.* **133** (1994), 191–205.
- [17] J. Kratochvíl and M. Křivánek, 'On the computational complexity of codes in graphs', in *Mathematical Foundations of Computer Science, Carlsbad, 1988*, Lecture Notes in Comput. Sci. **324** (Springer-Verlag, Berlin, 1988), pp. 396–404.
- [18] J. Kratochvíl, J. Malý and J. Matoušek, 'On the existence of perfect codes in a random graph', in *Random Graphs*, (M. Karoński, Jerzy Jaworski and Andrzej Ruciński, Editors) (J. Wiley and Sons, Chichester, 1990), pp. 141–149.
- [19] J. Kratochvíl, P. Manuel and M. Miller, 'Generalized domination in chordal graphs', *Nordic J. Comput.* **2** (1995), 41–50.
- [20] C.-K. Li and I. Nelson, 'Perfect codes on the Towers of Hanoi graphs', *Bull. Austral. Math. Soc.* **57** (1998), 367–376.
- [21] S.L. Lipscomb and J.C. Perry, 'Lipscomb's $L(A)$ space fractalized in Hilbert's $l^2(A)$ space', *Proc. Amer. Math. Soc.* **115** (1992), 1157–1165.
- [22] U. Milutinović, 'Completeness of the Lipscomb space', *Glas. Mat. Ser. III* **27(47)** (1992), 343–364.
- [23] C.B. Smart and P.J. Slater, 'Complexity results for closed neighborhood order parameters', *Congr. Numer.* **112** (1995), 83–96.
- [24] D.H. Smith, 'Perfect codes in the graphs O_k and $L(O_k)$ ', *Glasgow Math. J.* **21** (1980), 169–172.

Department of Mathematics, PeF
 University of Maribor
 Koroška 160
 2000 Maribor
 Slovenia
 e-mail: sandi.klavzar@uni-lj.si
 uros.milutinovic@uni-mb.si

Iskotel Telecommunications Systems Ltd.
 Tržaška 37a
 2000 Maribor
 Slovenia
 e-mail: petr@iskotel.si