LABRI - Bordeaux, university of science

# Covering codes in Sierpinski graphs

Glacet Christian
<christian.glacet@etu.u-bordeaux1.fr>

Tutor : Paul Dorbec

---

**Abstract**

This is a short introduction to the article "Covering codes in Sierpinski graphs" ( [1]). $(a, b)-$code, also named $[a, b]$-dominating, are a generalisation of covering codes and perfect weighted codes. That kind of codes are used for errors detection/corection and also for domination problems. In this paper, $(a, b)-$code are used to show some particularities of covreing codes overall. Sierpinski graph are used because they have a structure that allow to show properties on $(a, b)-$code easily. First I will explain some useful principles to understand the domain of [1]. Following will show main results of this paper and some keys to understand proofs. Finally, I will explain the idea of the proof of the theorem (5.1 in the original paper) that show how to make a particular $(a, b)-$code in a Sierpinski graph.

---

# 1 Definitions

## 1.1 Covering code - Error correcting code

The hamming distance is the number of digit that differ between two words. A 1-covering code (covering code of radius 1) is a code that allows to correct any single-bit error (hamming distance = 1), or detect all two(or less)-bit errors in a word (hamming distance $\leq 2$). Here is an example of such a code, the Hamming(7,4) introduced in [2] by Richard W. Hamming.

**Encoding :** The technique used is quite simple, we add 3 parity bit ($p_1$,$p_2$,$p_3$) to the original binary word ($d_1d_2d_3d_4$) of lenght 4 :

| original message | encoding | $p_1$ | $p_2$ | $d_1$ | $p_3$ | $d_2$ | $d_3$ | $d_4$ |
|---|---|---|---|---|---|---|---|---|
| 1100 | $\Longrightarrow$ | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

$$\begin{cases} p_1 = & d_1 + d_2 + d_4 = 0 \\ p_2 = & d_1 + d_3 + d_4 = 1 \\ p_3 = & d_2 + d_3 + d_4 = 1 \end{cases}$$

**Correcting single-bit errors :** Now assume that the data-bit $d_2$ has the wrong value ($d_2 = 0$), so we have :

| sent message | | received message |
|---|---|---|
| 0111100 | $\Longrightarrow$ | 011$\underline{\mathbf{0}}$000 |

$$\begin{cases} p'_1 = & 1 + 0 + 0 = 1 \\ p'_2 = & 1 + 0 + 0 = 1 \\ p'_3 = & 0 + 0 + 0 = 0 \end{cases}$$

We have two parity changes ($p_{1'} \neq p_1$ and $p_3 \neq p_3$), that means there is an error in $d_2$ or $d_4$. But $p_3$ parity hasn't changes, that means there is no error in all $d_1, d_3, d_4$ bit. So, we can now correct the error on $d_2$, changing from 0 to 1.

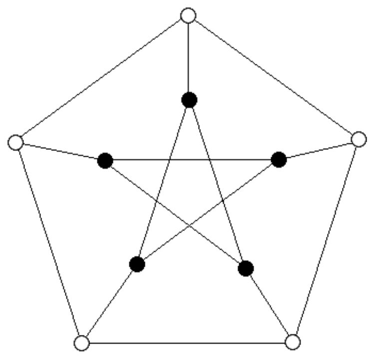
**Other covering codes :** You can see more examples and a classification of covering codes in [3].

## 1.2 $(a, b)-$code

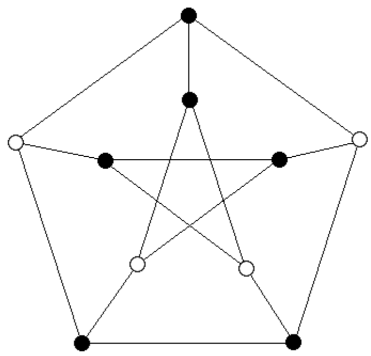
**Definition :** Let $G$ be a graph and $C$ a set of vertices, $G$ is an $(a, b)-$code if :

- all vertices belong to $C$ have $a$ neighbors in C,
- all vertices belong to $G \setminus C$ have $b$ neighbors in C

**Examples :** (black vertices belong to the code)



*a)* $(2, 1)-$*code in Petersen pentagon graph*



*b)* $(1, 3)-$*code in Petersen pentagon graph*

If all vertices belong to the code this is an $(3, 0)-$code. *Note that a* $(0, 1)-$*code is a perfect code, similar to the covering showed in 1.1.*

## 1.3 graphes de Sierpinski

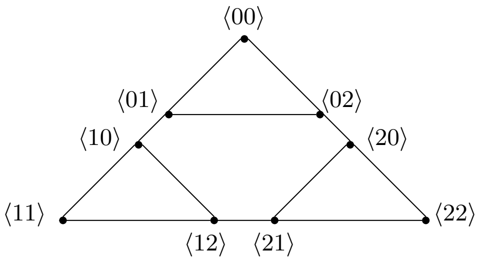
A Sierpinski graph ($S(n, k)$) is a fractal graph. The construction of such a graph is recursive :

i) Construct a $k-$clique ($K_k$) vertices set $S(1, k)$.

ii) Then build the next ($S(i, k)$) graph with $k$ subgraphs : $S(i - 1, k)$ and make a $k-$clique with those subgraphs (consider each subgraph as a vertice).

iii) Repeat ii) till $i >= n$.

**Example :** here is a $S(2,3)$




All $\langle ii...i \rangle$ vertices are called extreme vertices, others are called inner.
In $S(n,k)$ graphs, two vertices $u = \langle i_1 i_2 ... i_n \rangle$ and $v = \langle j_1 j_2 ... j_n \rangle$ are adjacents iif an index $h$ exists in $\{1, 2, \dots, n\}$ such that :

- $i_t = j_t$ for $t = 1, \dots, h-1$
- $i_h \neq j_h$
- $i_t = j_h$ and $i_h = j_t$ for $t = h+1, \dots, n$

**Examples :** $u = \langle 135755 \rangle$ and $v = \langle 135577 \rangle$ are adjacents ($h = 4$)

$$\begin{array}{c|c|c} 135 & 7 & 55 \\ 135 & 5 & 77 \end{array}$$

ditto for $u = \langle 12 \rangle$and $v = \langle 21 \rangle$ ($h = 1$)

$$\begin{array}{c|c} 1 & 2 \\ 2 & 1 \end{array}$$

# 2 $(a,b)-$code in graphes de Sierpinski

## 2.1 Interdependence of a and b in graphes de Sierpinski

The authors of [1] have proved that the only possible (a,b) values for an $(a,b)-$code in a Sierpinski graph are : **(a,a), (a,a+1) and (a,a+2)**. Extreme vertices or $k-$clique properties are essentialy used to prove that (lemma 3.2 to lemma 3.5 in [1]). For example in lemma 3.2 : Let $C$ be an $(a,b)-$code in $S(n,k)$ and $K_k$ its $k-$clique, they have proved that

$$| C \cap K_k | \leq a + 1$$

using $k-$clique properties ; If the number of vertices $| C \cap K_k | > a+1$ then any vertice from this set has more than $a$ neighbors, which is impossible because any vertice from the code should have exactly a neighbors.
And they have proved that

$$b - 1 \leq | C \cap K_k |$$

using extreme vertices properties (existence of extreme vertices) ; If there is extreme vertices, then the $S(n,k)$ graph is not regular so the code cannot be the whole graph. Let $u \notin C$, so $u$ has $b$ neighbors in $C$, but one of them can be out of the $k-$clique, so $| C \cap K(u) | \geq b - 1$ (and $b \leq k$). Finaly, we have :

- $| C \cap K_k | \geq b - 1$ if $C \cap$ $K_k$
- or $| C \cap K_k | = k (\geq b)$ if $C \subset K_k$

## 2.2 Construction of $(a, a+1)-$code in $S(n,k)$

This construction is based on recusion on $n$, the following explanation of theorem 5.1 (from [1]) is driven alongside an $(1,2)-$code in $S(n,3)$ construction. At each step, let $i < n$, build $k = 3$ complete graphs by matching $k$ sub-graphs

$(S(i-1,k))$ together, these $k$ graphs are called $(S_1, S_2, S_3, \ldots, S_k)$. For all $u \in$S(n,k) the following properties should be respected throughout the construction :

$$\left.\begin{array}{lcccc} |N(u)\cap C)| & = & a & \Rightarrow & \bullet_* \\ |N(u)\cap C)| & = & a-1 & \Rightarrow & \bullet_+ \end{array}\right\} if\ u \in C$$

or

$$\left.\begin{array}{lcccc} |N(u)\cap C)| & = & a+1=b & \Rightarrow & \circ_* \\ |N(u)\cap C)| & = & a & \Rightarrow & \circ_+ \end{array}\right\} if\ u \notin C$$

*The notation + is short for "need one more neighbor that belong to the code".* Inner vertices have necessarily the correct number of neighbors from the code (recussion hypothesis), so the notation is used only for extremal vertices. And the only possible extremal vertices matches are : $\circ_+ — \bullet_+$, $\bullet_+ — \bullet_+$, $\circ_* — \circ_*$ otherwise, $(a,b)$ restrictions will be broken (that is the key of this proof). Let define the following near codes, wich are the only possible ones :

$$\left.\begin{array}{cccccc} SO^n & has & a+1 & \bullet_* & and & k-a-1 & \circ_* \\ WO^n & has & a & \bullet_+ & and & k-a & \circ_+ \end{array}\right\} n\ is\ odd$$

$$\left.\begin{array}{cccccc} SE^n & has & a & \bullet_+ & and & k-a & \bullet_* \\ WE^n & has & a+1 & \circ_* & and & k-a-1 & \circ_* \end{array}\right\} n\ is\ even$$

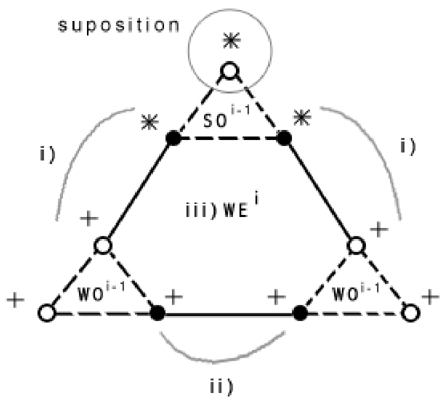
Furthermore, at last step of the construction $(i = n-1)$, the number of + should be at most egal to $k-1$, because only $k-1$ vertices could have a new neighbor that belong to the code (Sierpinski structure imposes that).

- For $i = 1$, two possible graph (modulo isomorphisms)


  - **W**eak-**O**dd graphs have $a$ $\bullet_+$ vertices, $k-a$ others are $\circ_+$. All neighbors for the next step $(i+1)$ should obviously be $\bullet_+$ (comming from another WO);
  - **S**trong-**O**dd graphs have $a+1$ $\bullet_*$ vertices, $k-a-1$ others are $\circ_*$. They obviously can't have $\bullet$ neighbors;

- For $i$ even, supose that there is a $\circ_*$ in one of the previous sub-graphs $(S(i-1,3) : S_1, S_2, S_3)$ that will remain extremal. Without lost of generality, we now assume that it is $S_1$ :
  i) Thus $S_1$ is isomorphic to $SO^{i-1}$ (cause $WO$ graphs do not contain $\circ_*$), so there is $a+1$ (here 2) $\bullet_*$ that must be matched with a $\circ_+$.
  ii) Therefore $S_2$ and $S_3$ are isomorphic to $WO^{i-1}$. These two graphs ($S_2$ and $S_3$) can be matched in one way, each $\bullet_+$ vertice should be matched to another $\bullet_+$ vertice.
  iii) It remains finally $k = 3$ unmatched vertices (extremal vertices), $k-a-1$ $\circ_*$ and $a+1$ $\circ_+$, in other words, $S(i,3)$ is isomorphic to $WE^i$.

  Here is a graphical representation of the process :



If the original vertex is not a $\circ_*$ the proof is exactly the same, with $\circ_+$ the construction ends to an $WE^i$, with $\bullet_+$ or $\bullet_*$ that ends to an $SE^i$.

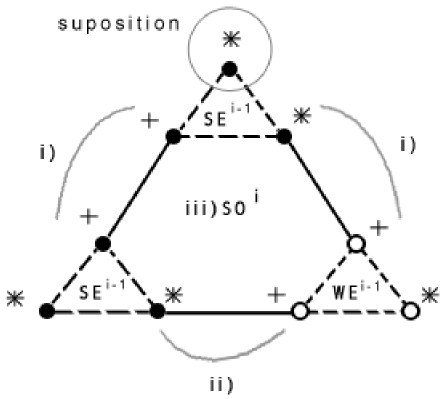
- For $i$ odd, supose that there is a $\bullet_*$ in one of the previous sub-graphs ($S(i-1,3)$ : $S_1, S_2, S_3$) that will remain extremal. Without lost of generality, we now assume that it is $S_1$ :

  i) Thus $S_1$ is isomorphic to $SE^{i-1}$ (cause $WE$ graphs do not contain $\bullet_*$), so there is $k-a-1$ (here 1) $\bullet_*$ that must be matched with a $\circ_+$. And $a$ (here 1) $\bullet_+$ that must be matched with a $\bullet_+$.

  ii) Therefore $S_2$ is isomorphic to $WE^{i-1}$ and $S_3$ is isomorphic to $SE^{i-1}$. These graphs ($S_2$ and $S_3$) can be matched in one way, each $\circ_+$ vertice should be matched to another $\bullet_*$ vertice.

  iii) It remains finally $k=3$ unmatched vertices (extremal vertices), $a+1$ $\bullet_*$ and $k-a-1$ $\circ_*$, $S(i,3)$ is isomorphic to $SO^i$. The actual graph is an $(a,a+1)-$code.

Here is a graphical representation of the process :




If the original vertex is not a $\bullet_*$ the proof is exactly the same, with $\circ_*$ the construction ends to an $SO^i$ (wich is an $(a,a+1)-$code too), with $\bullet_+$ or $\circ_+$ that ends to an $WO^i$.

# 3 Conclusion

It could be interseting to do a more precise construction (for 2.2) using label (seen in 1.3 example). Or maybe show that constructed code in this section are unique or not.

# References


[1] L. Beaudou, S. Gravier, S. Klavzar, M. Kovse, and M. Mollard, "Covering codes in sierpinski graphs."

[2] R. W. Hamming, "Error detecting and error correcting codes," 1950.

[3] P. R. J. Ostergard and W. D. Weakley, "Classification of binary covering codes," 1999.