The guessing number of undirected graphs

Demetres Christofides

Institute for Theoretical Computer Science
Faculty of Mathematics and Physics
Malostranské Náměstí 25
188 00 Prague, Czech Republic

demetres@kam.mff.cuni.cz

Klas Markström

Department of Mathematics and Mathematical Statistics
Umeå University
90187 Umeå, Sweden

klas.markstrom@math.umu.se

Abstract

Riis [*Electron. J. Combin.*, 14(1):R44, 2007] introduced a guessing game for graphs which is equivalent to finding protocols for network coding. In this paper we prove upper and lower bounds for the winning probability of the guessing game on undirected graphs. We find optimal bounds for perfect graphs and minimally imperfect graphs, and present a conjecture relating the exact value for all graphs to the fractional chromatic number.

1   Introduction

An active area of research in communication theory during the last ten years has been the development of protocols for network coding [ACLY00]. In network coding there are several senders and receivers who wish to pass messages between each other, however the routers in the network can only send one message at a time. In order to avoid bottlenecks, network coding allows the routers to compute and distribute new messages, as long as the receivers can compute the senders' original message from the collection of new messages they received in its stead.

In [Rii07b], Riis connected network coding to the much older problem of finding an optimal Boolean circuit for a Boolean function. In that paper, Riis disproved a conjecture of Valiant in circuit complexity and showed that both network coding and this form of

circuit complexity were equivalent to a certain type of multiplayer guessing game on a graph.

The guessing game can be described as follows: Each vertex v of a, directed or undirected, graph G is assigned a player, also denoted by v, and a uniformly random integer from $\{1 \ldots s\}$. A player v can see the numbers assigned at the neighbours of v, or inneighbours in the directed case, but not the number at v. Without communication, each player must now make a guess as to the value of its own random number. The team of players wins if all of them guess correctly their own value, and loses if anyone is incorrect. The objective is to now find a strategy which makes the winning probability as large as possible. At first it might seem like there is no way of getting a higher winning probability than s^{-n} on an n vertex graph, but Riis observed that this is far from being the case. If G is the complete graph, the players can use the following simple strategy: Each player picks the unique number such that the sum of that and all other numbers is 0 modulo s. If every player follows this strategy they will all be correct when the sum of all numbers is 0 modulo s, which happens with probability s^{-1}, a value which does not even depend on n.

In [Rii07b], Riis considered this game on general directed graphs, where player v can see the number at vertex u if and only if there is an edge from u to v. Motivated by this general question, in this paper we study the problem for undirected graphs. As we shall see, the problem of finding optimal guessing game strategies can be translated into a question regarding the size of the largest independent set in an auxiliary graph, and using this graph we can find good bounds for the winning probabilities.

Here is an outline of the paper. In the next section we give some formal definitions and prove some basic bounds for the winning probability of the guessing games. In Section 3, we explain how the guessing numbers can be determined by computing the fixed points of some specific maps. We also use the methods developed there to show that there is a naturally defined limit of the guessing numbers which we call the asymptotic guessing number. In Section 4, we find lower bounds on the guessing numbers using fractional clique covers. In Section 5, we define the code graph of the guessing game and prove that the guessing number can be computed by determining the independence number of this graph. In Section 6, we find upper bounds on the guessing numbers using entropy inequalities and pose a conjecture regarding the asymptotic guessing number of each graph. Finally in Section 7 we consider some generalisations of the methods of the paper in similar contexts.

As pointed out by the anonymous referee several of the results in this paper has independently been developed by Riis, with various co-authors, for the directed version of the game and we have given references for these directed graph results where they appear.

2 Definitions and some basic bounds

We start out by making a more formal definition of the guessing game.

Definition 2.1. In the guessing game on a graph G, each vertex v is assigned an integer

$x_v \in \{1, \dots, s\}$ uniformly at random.

Given a vertex v of a graph G, a strategy for player v for the guessing game on G is a function $f_v : \{1, \dots, s\}^{N(v)} \to \{1, \dots, s\}$. The value of f_v is called v's guess. A strategy $\mathcal{F}$ for the guessing game on G is a sequence of functions $(f_v)_{v \in V(G)}$ such that each function f_v is a strategy for player v for the guessing game.

We say that the players win if all guesses are correct and we write $\mathrm{Cor}(\mathcal{F})$ to denote this event.

We now define the guessing number of the graph via the winning probability in the optimal strategy for that graph.

Definition 2.2. The guessing number $\mathrm{gn}(G, s)$ of a graph G with respect to the positive integer s is the largest β such that there exists a strategy $\mathcal{F}$ for the guessing game on G such that with probability $\frac{1}{s^{n-\beta}}$ every player v guesses its own value x_v, i.e. $\mathrm{Pr}\left(\mathrm{Cor}(\mathcal{F})\right) = \frac{1}{s^{n-\beta}}$.

In general, one could consider strategies where each player makes a random choice based on the available information. However there is always an optimal strategy which is deterministic, so we only consider deterministic strategies.

Lemma 2.3. *Every randomised strategy for the guessing game on a graph G has winning probability at most $\frac{1}{s^{n-\mathrm{gn}(G,s)}}$.*

Proof. A randomised strategy can be described by assigning a probability $\mathrm{Pr}(\mathcal{F})$ to each deterministic strategy $\mathcal{F}$. However the winning probability of such a strategy is

$$\sum_{\mathcal{F}} \mathrm{Pr}(\mathcal{F}) \mathrm{Pr}\left(\mathrm{Cor}(\mathcal{F})\right) \leqslant \max_{\mathcal{F}} \mathrm{Pr}\left(\mathrm{Cor}(\mathcal{F})\right) = \frac{1}{s^{n-\mathrm{gn}(G,s)}}. \qquad \square$$

Using this terminology we can now give a formal version of the strategy for the complete graph which we described in the introduction.

Example 2.4. Let G be the complete graph K_n on n vertices. We define a strategy for the guessing game on G by defining f_v to map the sequence $(x_u)_{u \in N(v)}$ to the unique integer $x'_v \in \{1, \dots, s\}$ such that $x'_v + \sum_{u \in N(v)} x_u$ is divisible by s. We will call this the *clique strategy*. Observe that all the players guess correctly if and only if $x'_v = x_v$ for every $v \in V(G)$ which holds if and only if $\sum_{v \in V(G)} x_v$ is divisible by s. The probability of this event is thus $\frac{1}{s}$. Since the probability that a single player guesses its own value correctly is also $\frac{1}{s}$, this cannot be improved and we find that $\mathrm{gn}(K_n, s) = n - 1$.

From the clique strategy we can define a natural strategy for general graphs by partitioning the vertex set into cliques.

Definition 2.5. A clique cover, or clique partition of a graph G is a partition of $V(G)$ into vertex disjoint cliques. The clique cover number $\mathrm{cp}(G)$ of G is the minimum cardinality of a clique cover of G.

Note that the cliques in a clique cover of G induce a partition of the complement of G into independent sets, i.e. a proper vertex colouring of $\overline{G}$. Hence $\mathrm{cp}(G) = \chi\left(\overline{G}\right)$.

We can now give a lower bound for the guessing number in terms of the clique cover number, and an upper bound in terms of the independence number of G. As usual, we let $\alpha(G)$ denote the size of the largest independent set in G.

Lemma 2.6.

1. *For every graph G and every positive integer s, $\mathrm{gn}(G, s) \geqslant n - \mathrm{cp}(G)$*

2. *For every graph G and every positive integer s, $\mathrm{gn}(G, s) \leqslant n - \alpha(G)$*

Proof. 1. A strategy giving this bound can be constructed by taking a minimal clique cover and letting the players in each clique follow the clique strategy for that clique. This gives probability at least $\frac{1}{s^{\mathrm{cp}(G)}}$ that all players guess correctly.

2. Let I be a maximum independent set in G and let $\mathcal{F} = (f_v)_{v \in V(G)}$ be a strategy. We choose the random number $(x_u)_{u \in V(G)}$ in two stages. In the first stage, we generate all numbers x_u with $u \notin I$. Observe that since I is independent all functions f_u with $u \in I$ can now be determined. In the second stage, we generate all numbers x_u with $u \in I$ and observe that the probability that player v guesses correctly is exactly $1/s$ with the events being independent. It follows that $\Pr\left(\mathrm{Cor}(\mathcal{F})\right) \leqslant s^{-\alpha(G)}$ as required.

$\square$

We call the strategy used in part 1 of the proof of the above lemma, the *clique cover strategy*.

Even these two simple bounds are enough to determine the guessing numbers exactly for large classes of graphs. In fact, these two bounds determine the guessing number of G precisely when $\alpha(G) = \mathrm{cp}(G)$. One particularly natural class which satisfies this property is the class of *perfect graphs*, introduced by Berge [Ber63]. This is the class of graphs such that $\chi(H) = \omega(H)$ for all induced subgraphs of G. A classical theorem of Lovász [Lov72] tells us that a graph is perfect if and only if its complement is perfect. In our context this means that for a perfect graph G we have $\alpha(G) = \omega(\overline{G}) = \chi\left(\overline{G}\right) = \mathrm{cp}(G)$

Corollary 2.7. *If G is perfect then $\mathrm{gn}(G) = n - \alpha(G)$.*

This tells us among other things that if G is bipartite then $\mathrm{gn}(G, s) = n - \alpha(G)$. One striking consequence is that a disjoint union of n K_2's, i.e. a matching of size n has the same guessing number as the complete bipartite graph $K_{n,n}$, despite being a subgraph of it having a factor of n fewer edges. In other words, all this extra information that the players have in playing the guessing number for $K_{n,n}$ contributes nothing to the winning probability for the guessing game on this graph compared with the guessing game on the matching of size n.

3 Fixed points and the asymptotic guessing number

Another way of describing the guessing problem is in terms of fixed points of the mappings given by different strategies. Note that a strategy $\mathcal{F}$ can be viewed as a mapping from $A_s = \{1, \ldots, s\}^{V(G)}$ into itself. The strategy $\mathcal{F}$ guesses correctly on a given outcome of random numbers $x = (x_v)_{v \in V(G)}$ if $\mathcal{F}(x) = x$. Thus the problem of finding a good strategy for the guessing game can be interpreted as finding a mapping $\mathcal{F} : A_s \to A_s$ with as many fixed points as possible, where f_v only depends on $x_u \in N(v)$. We call a mapping with the given dependence structure a *strategy mapping* and denote the set of all strategy mappings on A_s by $\mathcal{S}(G, s)$. As similar discussion in terms of fixed points was given in [WCR09].

If we let $\mathrm{Fix}(\mathcal{F})$ denote the number of fixed points, then we can compute the guessing number as

$$\mathrm{gn}(G, s) = \max_{\mathcal{F} \in \mathcal{S}(G,s)} \log_s \mathrm{Fix}(\mathcal{F})$$

Let us define $\mathrm{Fix}(G, s) = \max_{\mathcal{F} \in \mathcal{S}(G,s)} \mathrm{Fix}(\mathcal{F})$.

While the guessing number is given by the maximum number of fixed points for a mapping of this type there are also mappings at the other extreme with no fixed points, for every non-trivial graph.

Example 3.1. Let $G = K_2$ and take $s = 2$. Let the first player guess the same value as he sees (i.e. the outcome of the random experiment of the second player), and let the second player guess the opposite of the value it sees. For this strategy, $\mathrm{Fix}(\mathcal{F}) = 0$. Thus for every graph with at least one edge there exists a strategy which never guesses all values correctly, and this can be extended to larger s as well. This is of course to be expected as the average number of fixed points over all strategy mappings is 1. So if we are able to find strategy mappings with more fixed points, then there must also exist strategy mappings with no fixed points at all.

So far, for all the examples of graphs we have seen, the guessing number was independent of s. Our earlier results show that in order to see a dependence on s we must consider non-perfect graphs. The strong perfect graph theorem [CRST06] tells us that a graph is perfect if and only if neither G nor its complement contains an induced odd cycle of length at least five. So we turn our attention to the five cycle C_5, which is the smallest non-perfect graph.

Example 3.2. Let $G = C_5$, be the cycle on 5 vertices. Since $\alpha(C_5) = 2$ and $\chi(\overline{C_5}) = 3$, Lemmas 2.6 and 2.6 give $2 \leqslant \mathrm{gn}(G, s) \leqslant 3$ for every s.

For $s = 2$, let $\mathcal{F}$ be the strategy where a vertex guesses 2 if both its neighbours have value 1, and guesses 1 otherwise. This strategy guesses correctly on the following x: $\{11212, 12112, 12121, 21121, 21211\}$. Thus we have $\mathrm{gn}(C_5, 2) \geqslant \log_2 5$, and a quick computer check shows that this is indeed optimal.

For $s = 3$, let $\mathcal{F}$ be the strategy where a vertex guesses 2 if both its neighbours have value 1, guesses 3 if at least one neighbour has values 3 and no neighbour has value 2, and guesses 1 otherwise. This strategy has $\mathrm{Fix}(\mathcal{F}) = 11$ and a computer check shows

that it is the best symmetric strategy. However by computer check, using the methods of the next section, we know that there exists a more complex, vertex dependent, strategy which is optimal and has $\text{Fix}(\mathcal{F}) = 12$. This strategy guesses correctly on the following x:

$\{11111, 11231, 12323, 12331, 21112, 22212, 23312, 23323, 23331, 31113, 31233, 31323\}$ In this strategy each vertex uses a strategy which is distinct from that of every other vertex. Thus we have $\text{gn}(C_5, 3) = \log_3 12 < \text{gn}(C_5, 2)$. As pointed out by the referee this settles a question by Yun posed in [Yun09], asking if the symmetric strategy is optimal.

For $s = 2^k$ we can create a strategy by writing each x_v in base 2 and following the clique strategy in each specific bit separately. This strategy will have $\text{Fix}(\mathcal{F}) = 5^k$, and so $\text{gn}(C_5, 2^k) \geqslant \text{gn}(C_5, 2)$.

Comparing the cases $s = 2$, $s = 3$ and $s = 4$ we see that $\text{gn}(C_5, s)$ is not monotone in s.

The guessing number $\text{gn}(G, s)$ is of course bounded by the number of vertices of G but as we have seen in the previous example it is not monotone. One may wonder whether the sequence $(\text{gn}(G, s))_{s \geqslant 1}$ tends to a limit or not. We will show that this is indeed the case but first we go on to prove a few simple lemmas concerning the properties of $\text{Fix}(G, s)$.

Lemma 3.3. $\text{Fix}(G, s + 1) \geqslant \text{Fix}(G, s) + 1$

Proof. Given any strategy mapping $\mathcal{F} = (f_v)_{v \in V(G)}$ for the guessing game on G with respect to s, we extend this to a strategy mapping $\mathcal{F}' = (f'_v)_{v \in V(G)}$ for the guessing game on G with respect to $s + 1$ as follows: If $x_u = s + 1$ for at least on $u \in N(v)$, then $f'_v(x) = s + 1$, otherwise $f'_v(x) = f_v(x')$, where x' is obtained from x by defining x'_u to be equal to x_u if $x_u \in \{1, \ldots, s\}$ and equal to 1 otherwise. Observe that $\mathcal{F}'$ is indeed a strategy mapping, every fixed point of $\mathcal{F}$ is also a fixed point of $\mathcal{F}'$ and moreover $\mathcal{F}'$ has $(s + 1, \ldots, s + 1)$ as a fixed point as well. $\qquad\square$

Lemma 3.4. *If H is a subgraph of G then $\text{Fix}(H, s) \leqslant \text{Fix}(G, s)$*

Proof. Given a strategy mapping $\mathcal{F}$ for the guessing game on H, we extend it to a strategy mapping $\mathcal{F}'$ for the guessing game on G by defining $f'_v(x)$ to be equal to $f_v(\{x_u : u \in V(H)\})$ if $v \in V(H)$ and to be identically 1 otherwise. It is immediate that $\mathcal{F}'$ is a strategy mapping having at least as many fixed points as $\mathcal{F}$. $\qquad\square$

The last inequality is very far from being strict, as shown by our earlier example with the balanced complete bipartite graph on $2n$ vertices versus the matching of size n.

We can also bound the number of fixed points for composite values of s

Lemma 3.5. $\text{Fix}(G, s_1 s_2) \geqslant \text{Fix}(G, s_1)\text{Fix}(G, s_2)$

Proof. This follows by simply writing each random number x_v as $(a - 1)s_1 + b$, with $b \in \{1, \ldots, s_1\}$ and $a \in \{1, \ldots, s_2\}$, and using the optimal mappings for s_1 and s_2 to guess a and b independently. $\qquad\square$

Theorem 3.6. *The limit* $\lim_{s\to\infty} \log_s \text{Fix}(G,s)$ *exists, and is at most* $|V(G)|$. *Moreover, it is equal to* $\sup_{s\in\mathbb{N}} \log_s \text{Fix}(G,s)$.

Proof. Suppose G has n vertices and let us write $a_s = \text{Fix}(G,s)$. By the definition of $\text{Fix}(G,s)$ it follows that $a_s \leqslant s^n$ for every s and so if the limit exists then it is at most n. By Lemma 3.3 we have that $a_{s+1} \geqslant a_s$ for every $s \in \mathbb{N}$ and by Lemma 3.5 we have that $a_{st} \geqslant a_s a_t$ for every $s,t \in \mathbb{N}$. Our aim is to show that $\lim_{s\to\infty} \log_s a_s$ exists and is equal to $\sup_{s\in\mathbb{N}} \log_s a_s$. Given $s,t \in \mathbb{N}$ we claim that

$$\log_s a_s \geqslant \frac{k}{k+1} \log_t a_t,$$

where $k = \lceil \log s / \log t \rceil$. The result immediately follows. Indeed, given $\varepsilon > 0$, let $\ell = \sup_{s\in\mathbb{N}} \log_s a_s$ and take t large enough such that $\log_t a_t > (1-\varepsilon)\ell$. With this t fixed, we can now pick s large enough such that $k/(k+1) > (1-\varepsilon)$. We deduce that $\log_s a_s > (1-\varepsilon)^2\ell$, thus $\liminf \log_s a_s \geqslant \ell$ and so $\lim \log_s a_s$ exists and is equal to ℓ. To prove the claim, observe that

$$\log_s a_s = \frac{\log_t a_s}{\log_t s} \geqslant \frac{\log_t a_{t^k}}{\log_t t^{k+1}} \geqslant \frac{\log_t a_t^k}{k+1} = \frac{k}{k+1} \log_t a_t. \qquad \square$$

Hence may introduce the following asymptotic version of the guessing number.

Definition 3.7. The asymptotic guessing number $\text{gn}(G)$ of G is

$$\text{gn}(G) = \lim_{s\to\infty} \log_s \text{Fix}(G,s)$$

Example 3.8. Let us return to C_5. We have already seen that $\text{gn}(C_5, 2^k) \geqslant \text{gn}(C_5, 2) = \log_2 5$. It follows that $\text{gn}(C_5) \geqslant \log_2 5$. In particular $\text{gn}(C_5) > 2$. In the next section we will improve further on this bound.

4   Lower bounds via fractional clique covers.

So far we have determined exactly the guessing numbers of perfect graphs and found a non-trivial lower bound for the asymptotic guessing number of C_5. In fact, a much better bound can be derived for both C_5 and other non-perfect graphs by making use of a fractional coverings. Before defining fractional coverings we introduce the t-fold blow up of graphs and the blow-up strategy.

Definition 4.1. Given a graph G we define the t-fold blow up of G, denoted by G^t, as follows: For each vertex v in G there are t vertices $v_1, \ldots, v_t$ in G^t with vertices v_i and u_j being neighbours in G^t if and only if v and u are neighbours in G.

Later we will also need the following special case of the strong graph product.

Definition 4.2. The strong product of K_t and G, denoted $K_t * G$ is the graph obtained from G^t by adding all edges of the form (v_i, v_j) for each vertex v in G.

We can use the t-blow up of G to obtain better bounds for the guessing number of G with respect to values of s which are perfect t powers.

Definition 4.3. Let G be a graph, let t be a positive integer and let $s = s_1^t$ for some integer $s_1 > 1$. Given any strategy $\mathcal{F}$ for G^t with respect to s_1, we can obtain a strategy, which we call the blow-up strategy, for G with respect to s as follows: If x is the number assigned to a vertex v of G, we write $x = \sum_{i=1}^{t}(x_i - 1)s_1^{i-1}$, where $x_i \in \{1, \ldots, s_1\}$ for each i. Then we assign the numbers $x_1, \ldots, x_t$ to the vertices $v_1, \ldots, v_t$ of G^t. We then follow the strategy $\mathcal{F}$ on G^t. If $y_1, \ldots, y_b$ are the numbers guessed by $\mathcal{F}$ at vertices $v_1, \ldots, v_b$, then the number guessed by the blow-up strategy at vertex v will be $y = \sum_{i=1}^{t}(y_i - 1)s_1^{i-1}$.

We note that a similar blow-up strategy can be used in the case that s is a product of t (not necessarily equal) integers strictly greater than 1. We discuss how this can be done in subsection 7.1.

Theorem 4.4. *Let G be a graph and let t, s_1 be positive integers. Then $\mathrm{gn}(G, s_1^{t}) \geqslant \mathrm{gn}(G^t, s_1)$.*

Proof. It is immediate that we can partition G^t into t vertex disjoint copies of G. Let $\mathcal{F}$ be the strategy on G^t with respect to s_1 which follows the best possible strategy on each of these t copies of G. We now follow the blow-up strategy. The winning probability is at least $s_1^{-t(n - \mathrm{gn}(G, s_1))} = s^{-(n - \mathrm{gn}(G, s_1))}$ and the result follows. $\qquad\square$

We could have in fact proved this theorem using Lemma 3.5 instead. We will see a more powerful application of the blow-up strategy after we introduce fractional coverings of graphs.

Definition 4.5. A fractional clique cover of a graph G is a family of cliques $H_1, \ldots, H_t$ of G together with non-negative weights $w_1, \ldots, w_t$ such that $\sum_{\{i : v \in H_i\}} w_i \geqslant 1$ for all $v \in V(G)$. The minimum value of $\sum_{i=1}^{t} w_i$ over all clique covers $H_1, \ldots, H_t$ of G is known as the fractional clique cover number of G and is denoted by $\mathrm{cp}_f(G)$. It is also known as the fractional chromatic number of the complement $\overline{G}$ of G, denoted by $\chi_f(\overline{G})$.

It is well known that $\mathrm{cp}_f(G)$ is always a rational number and that $\alpha(G) \leqslant \mathrm{cp}_f(G) \leqslant \mathrm{cp}(G)$. Similarly, we have that $\omega(G) \leqslant \chi_f(G) \leqslant \chi(G)$. One of the basic results in fractional graph theory lets us relate the fractional chromatic number to the chromatic number of a suitable blow up of the original graph. For these and other results on fractional graph theory we refer the reader to [SU97].

Theorem 4.6. *For each graph G there exists a positive integer t, with such that $\chi_f(G) = \frac{\chi(K_t * G)}{t}$. Equivalently, considering the complement of $K_t * G$, we have $\mathrm{cp}_f(\overline{G}) = \frac{\mathrm{cp}(\overline{G^t})}{t}$*

Observe that for any positive integer t and any integer multiple s of t we have that $\chi(K_t * G)/t \leqslant \chi(K_s * G)/s \leqslant \chi_f(G)$. In particular, if t is an integer for which the consequence of Theorem 4.6 holds, then it also holds for any integer multiple of t. Note that considering the complement of G, there is an integer t (not necessarily the same as

the one given by Theorem 4.6) such that $cp(G^t) = tcp_f(G)$. So we can use a combination of the clique cover strategy and the blow-up strategy to obtain better strategies for the guessing game.

Definition 4.7. Let G be any graph and let t be any positive integer such that $\mathrm{cp}(G^t) = t\,\mathrm{cp}_f(G)$. Then the fractional clique strategy is defined by taking the blow-up strategy of G with respect to the clique cover strategy of G^t.

Example 4.8. Consider the fractional clique strategy for C_5. The fractional chromatic number of $\overline{C_5}$ is $\frac{5}{2}$ and a fractional clique strategy for C_5 can be obtained by using the clique cover strategy on the 2-fold blow up C_5^2 of C_5. This graph is depicted in Figure 1, with a clique covering given by the thick edges.
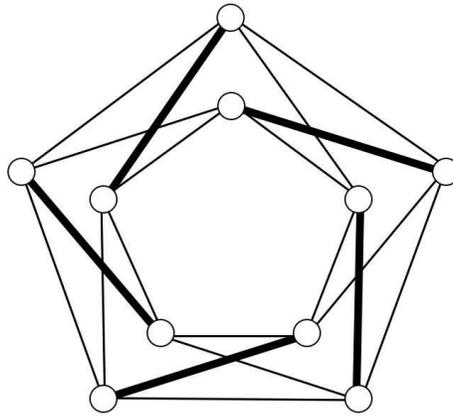
Figure 1: C_5^2 with a clique cover shown as thick edges.

Theorem 4.9. *Let t be any positive integer as in Definition 4.7 . Then $\mathrm{gn}(G, s) \geqslant n - \mathrm{cp}_f(G)$ whenever s is a perfect t-power. In particular, $\mathrm{gn}(G) \geqslant n - \mathrm{cp}_f(G)$.*

Proof. We apply the fractional clique strategy. We have $\mathrm{cp}(G^t) = t\,\mathrm{cp}_f(G)$ and so, if $|V(G)| = n$ and $s = s_1^t$, then the winning probability is

$$s_1^{-(tn - \mathrm{cp}(G^t))} = s_1^{-(tn - t\,\mathrm{cp}_f(G))} = s^{-(n - \mathrm{cp}_f(G))}.$$

The result follows. The result for the asymptotic guessing number also follows immediately since $\mathrm{gn}(G, s) \geqslant n - \mathrm{cp}_f(G)$ holds for infinitely many s. $\qquad\square$

For the five-cycle, and other symmetrical graphs, we can apply the following well known lemma (Proposition 3.1.1 in [SU97]).

Lemma 4.10. *If G is a vertex transitive graph on n vertices then $\chi_f(G) = \frac{n}{\alpha(G)}$ and $\mathrm{cp}_f(G) = \frac{n}{\omega(G)}$.*

Two simple classes of vertex transitive graphs are the odd cycles and their complements.

Example 4.11. For C_5, Lemma 4.10 improves our previous bounds to $\mathrm{gn}(C_5, s) \geqslant \frac{5}{2}$ for every s which is a perfect square. This is exactly half way between the two simple bounds given by $\alpha(C_5)$ and $\chi(\overline{C_5})$. For the odd cycles in general, we find that $\mathrm{gn}(C_{2k+1}, s) \geqslant \frac{2k+1}{2}$ for every s which is a perfect square, and for their complements we find $\mathrm{gn}(\overline{C_{2k+1}}, s) \geqslant (2k+1) - \frac{2k+1}{k} = 2k - 1 - \frac{1}{k}$ for every s which is a perfect k-power. Hence, by the strong perfect graph theorem [CRST06], we have found improved bounds for the class of minimally imperfect graphs.

For the odd cycles, the improvement given by considering $\mathrm{cp}_f(G)$ instead of $\mathrm{cp}(G)$ is bounded. However there are families of graphs where the improvement can be arbitrarily large.

Example 4.12. Given positive integers n, r with $n > 2r$, the Kneser graph $G = K_{n:r}$ has the family of all r-subsets of $\{1, \ldots, n\}$ as its vertices with two vertices being neighbours if and only if the corresponding sets are disjoint. These graphs are clearly vertex transitive on $\binom{n}{r}$ vertices. It is immediate that $\omega(G) = \lfloor n/r \rfloor$. The Erdős-Ko-Rado theorem implies that $\alpha(G) = \binom{n-1}{r-1}$ and so by Lemma 4.10 we get that $\chi_f(G) = n/r$. So the guessing number of $\overline{G}$ is $\mathrm{gn}(\overline{G}, s) = \binom{n}{r} - \frac{n}{r}$, for every s which is a perfect rth power. For the Kneser graphs taking the r-fold blow up of $K_{n:r}$ will give a working clique cover strategy, see Chapter 3 of [SU97] for full details of the blow ups of Kneser graphs..

On the other hand, by Lovász' solution to Kneser's conjecture, we have $\chi(G) = n - 2r + 2$ and so the simple clique cover strategy only gives $\mathrm{gn}(\overline{G}, s) \geqslant \binom{n}{r} - (n - 2r + 2)$. So if $n = tr$, then we get an improvement of $(t-2)(r-1)$.

The family of Kneser graphs has some additional importance in connection with fractional cover due to the fact that the fractional chromatic number of a graph can be completely described in terms of which Kneser graphs G has homomorphisms into, see Chapter 3 of [SU97].

Other graphs with large gaps between the bounds can be found by e.g. using the Mycielski construction or the strong graph product, under which fractional chromatic numbers are multiplicative. See [SU97] for more details on these and other constructions for graphs with prescribed fractional chromatic numbers.

5 The code graph

In the previous section we have seen that the guessing number can be described in terms of the number of fixed points of the strategy maps, and since this is a finite set of mappings of a finite set it is in principle possible to find the optimal strategy for a given G and s by an exhaustive search. We also saw how to provide good lower bounds for the guessing number in terms of $\chi_f(\overline{G})$, but we did not find matching upper bounds in general. Our next aim is to introduce the code graph for the guessing game with a given G and s

and, using this graph, both provide a more efficient computational procedure for finding optimal strategies and a conjecture as to what the asymptotic guessing number $\text{gn}(G)$ should be.

Independently of us a "guessing graph" is introduced by Gadoleau and Riis in [GR10]. For undirected graphs our code graph and their guessing graph are equivalent, however they also develop the guessing graph for general directed graphs. In that setting they also develop analogue theorems of our Theorem 5.2 and the first part of Theorem 5.4, and prove that the guessing graph is vertex transitive. We refer the reader to [GR10] for a further discussion of the relationship between these and other conflict graphs used in the literature.

Let $\Omega(n, s)$ denote the set of all strings of length n over the alphabet $\{1, \ldots, s\}$. Given a graph G on n vertices we will usually identify $\Omega(n, s)$ with $A_s = \{1, \ldots, s\}^{V(G)}$. Our ultimate aim is to determine which subsets of $\Omega(n, s)$ are fixed point sets of some strategy map $\mathcal{F}$ on G.

Definition 5.1. The code graph $X(G, s)$ has vertex set $\Omega(n, s)$ with two vertices x and y of $X(G, s)$ being adjacent if and only if there is a vertex v of G such that $x_v \neq y_v$ but $x_u = y_u$ for all $u \in N(v)$

As the next theorem shows, the code graph completely describes the set of optimal strategies for the guessing game.

Theorem 5.2. *Let I be a set of vertices of $X(G, s)$. Then I is the set of fixed points of an optimal strategy mapping $\mathcal{F}$ if and only if it is a maximal independent set.*

Proof. Let $\mathcal{F}$ be a strategy mapping (not necessarily optimal) having I as its set of fixed points. Let $x, y \in I$ and suppose that they are neighbours in $X(G, s)$. Then there exist at least one vertex v of G such that $x_v \neq y_v$ but $x_u = y_u$ for all $u \in N(v)$. But x, y are fixed points of $\mathcal{F}$ and so x_v and y_v can be determined by the values of x_u and y_u with $u \in N(v)$. Thus $x_v = y_v$, a contradiction.

Conversely, given a maximum independent set I, we define a mapping $\mathcal{F} : A_s \to A_s$ as follows: For each vertex v of G and each vertex x of $X(G, s)$, if there is an $x' \in I$ such that $x'_u = x_u$ for each $u \in N(V)$ then we define $f_v(x) = x'_v$. Otherwise, we define $f_v(x) = 1$. Note that since I is an independent set then each f_v is well-defined and the mapping $\mathcal{F} = \{f_v\}_{v \in V(G)}$ is a strategy mapping. We claim that $\mathcal{F}$ has I as its the set of fixed points. Indeed, by construction every element of I is a fixed point of $\mathcal{F}$. Moreover, since the set of fixed points of $\mathcal{F}$ is an independent set containing the maximum independent set I, then it must be equal to I. It now follows that $\mathcal{F}$ is optimal as no other strategy mapping can have more fixed points. $\square$

It follows that we can compute the guessing number of G by computing the independence number of $X(G, s)$

Corollary 5.3. *For every positive integer s and every graph G we have that*

$$\text{gn}(G, s) = \log_s(\alpha(X(G, s))).$$

Our simple bounds for the guessing number can easily be translated into properties of the code graph.

Theorem 5.4.

1. *If $\alpha(G) = t$, then $\alpha(X(G, s)) \leqslant s^{n-t}$.*

2. *If $\chi(\overline{G}) = k$, then $\alpha(X(G, s)) \geqslant s^k$.*

Proof.

1. Let I be an independent set of size $\alpha(G) = t$ in G. Given $x, y \in \Omega(n, s)$ we say that they are equivalent if and only if $x_u = y_u$ for every $u \notin I$. Observe that this is an equivalence relation with each equivalence class being a clique in $X(G, s)$ and having size exactly s^t. Every independent set of $X(G, s)$ can contain at most one element from every such clique and therefore it can have size at most s^{n-t}.

2. Let $H_1, \ldots, H_k$ be a clique cover of G of size k and let I be the set of all elements of $x \in X(G, s)$ such that $\sum_{v \in H_i} x_v \equiv 0 \bmod s$ for each $1 \leqslant i \leqslant k$. Then it is easily seen that I is an independent set of $X(G, s)$ of size s^k. $\qquad\square$

The bound from Theorem 4.9 can also be translated into this setting by considering the code graph of G^t, for a suitable value of t.

The code graph has s^n vertices. Given any pair of strings of length n we can decide in time n^2 if they are adjacent in the code graph or not. Having constructed the code graph we can now use standard algorithms for finding maximum independent sets to find both the guessing number and optimal guessing strategies. Using e.g. the maximum independent set algorithm from [TT77] we get the following upper bound on the complexity of determining the optimal strategy.

Corollary 5.5. *There exists an algorithm which constructs the optimal guessing strategy in time at most $2^{\frac{1}{3}s^n}$*

The upper bound here is of course prohibitively large, however it can be reduced by using the large automorphism group of $X(G, s)$, e.g. using methods similar to those of [RAMS04].

Lemma 5.6. $X(G, s)$ *is vertex transitive and* $\operatorname{Aut}(G) \rtimes \mathbb{Z}_s^n \subseteq \operatorname{Aut}(X(G, s))$, *with the semidirect group product $\rtimes$ as described in the proof.*

Proof. Let ϕ be an automorphism of G and let $y \in \mathbb{Z}_s^n$. Then the automorphism (ϕ, y) of $\operatorname{Aut}(X(G, s))$ is defined by

$$(\phi, y)(x)_v = (x + y_{\phi(v)})_{\phi(v)}$$

where the elements of $\Omega(n, s)$ are viewed as vectors in $\mathbb{Z}_s^n$ with addition defined componentwise. It is straightforward to verify that this is indeed an automorphism of $X(G, s)$ and that $(\phi, y) \circ (\phi', y') = (\phi \circ \phi', y + y')$ as required. $\qquad\square$

6 Upper bounds via entropy inequalities

Our next aim is to show that our bound on the guessing number for odd cycles and their complements given in Example 4.12 is sharp. As a tool we will use entropy inequalities of discrete random variable. Recall that given a discrete random variable X with outcomes labelled $1, \ldots, N$, its binary entropy is defined as

$$H(X) = \sum_{i=1}^{N} \Pr(X = i) \log_2 \Pr(X = i),$$

It will be more convenient for what follows to work with the s-entropy instead

$$H(X) = \sum_{i=1}^{N} \Pr(X = i) \log_s \Pr(X = i).$$

From now on, we will drop the subscripts and follow the convention that all our logarithms are with base s. The conditional entropy of X given another random variable Y is the entropy

$$H(X|Y) = \sum_{i,j} \Pr(X = i, Y = j) \log \Pr(X = i|Y = j).$$

We will use the following properties of the entropy function.

Theorem 6.1. *Let $X, Y, X_1, \ldots, X_n$ be discrete random variables*

1. *$H(X) \geqslant 0$ with equality if and only if X is determenistic.*

2. *If X takes values in a set S, then $H(X) \leqslant \log |S|$ with equality if and only if X is uniformly distributed on S.*

3. *$H(X, Y) = H(X|Y) + H(Y)$.*

4. *$H(X|Y) \geqslant 0$ with equality if and only if X is determined by Y.*

5. *$H(X|Y) \leqslant H(X)$ with equality if and only if X and Y are independent.*

6. *For $A \subseteq \{1, \ldots, n\}$, let X_A denotes the random vector of X_i's with $i \in A$. With this notation, H is a* submodular *function, i.e.*

$$H(X_{A \cup B}) + H(X_{A \cap B}) \leqslant H(X_A) + H(X_B).$$

For these and other properties of the entropy function we refer the reader to [CT06, AS08]. Recently, entropy inequalities have been used used to derive bounds for other problems in combinatorics, see [Rad03] for a survey.

Theorem 6.2. *For each $k, s \in \mathbb{N}$ we have that $\mathrm{gn}(C_{2k+1}, s) \leqslant \frac{2k+1}{2}$.*

Proof. Assume that the vertices of C_{2k+1} are numbered as $1, \ldots, 2k+1$ with $N(i) = \{i-1, i+1\}$ where addition and subtraction are done modulo $2k+1$. Let I be a maximum independent set of $X(C_{2k+1}, s)$ and let $X = (X_1, X_2, \ldots, X_{2k+1})$ be the characteristic vector of an element picked uniformly at random from I. By property (2), we have that $H(X) = \log|I|$. We will now proceed to bound $H(X)$ from above. When estimating entropies we will need to consider many quantities of the form $H(X_B)$ and in order to simplify our notation we will write them as $H(B)$ instead.

Since X is a fixed point of a strategy mapping, we know that X_i is determined by X_{i-1} and X_{i+1}. In particular, X is determined just by the random variables $X_2, X_3, X_5, X_7, \ldots, X_{2k+1}$ and so by properties (3) and (4) we have that

$$H(X) = H(X|2, 3, 5, \ldots, 2k+1) + H(2, 3, 5, \ldots, 2k+1)$$
$$= H(2, 3, 5, \ldots, 2k+1)$$

Applying now properties (3) and (5) we get that

$$H(X) \leqslant H(2,3) + H(5, 7, \ldots, 2k+1),$$

and by property (2) it follows that

$$H(X) \leqslant H(2,3) + \log(s^{k-1}) = H(2,3) + k - 1.$$

Likewise, we have

$$H(X) \leqslant H(1,2,3,4) + H(6, 8, \ldots, 2k) \leqslant H(1,2,3,4) + k - 2.$$

Adding these inequalities and using property (6) we get that

$$2H(X) \leqslant H(2,3) + H(1,2,3,4) + 2k - 3$$
$$\leqslant H(1,2,3) + H(2,3,4) + 2k - 3.$$

But since X_2 is determined by X_1 and X_3 we have that $H(1,2,3) = H(1,3) \leqslant \log(s^2) = 2$ and similarly $H(2,3,4) \leqslant 2$. Putting all these together we get

$$\log|I| = H(X) \leqslant \frac{2k+1}{2}$$

and so by Lemmas 5.2 and 5.3 we get that $\mathrm{gn}(C_{2k+1}, s) \leqslant \frac{2k+1}{2}$ $\qquad \square$

For C_5, the case $k = 2$ in the theorem, this inequality was proved in [Rii07a] as well.

Using a larger family of index sets we can similarly find a bound for the guessing number of the complement of an odd cycle.

Theorem 6.3. *For each $k, s \in \mathbb{N}$ we have that $\mathrm{gn}(\overline{C_{2k+1}}, s) \leqslant 2k - 1 - \frac{1}{k}$.*

Proof. Assume that the vertices of $\overline{C_{2k+1}}$ are numbered as $1, \ldots, 2k+1$ with $N(i) = [2k+1] \setminus \{i-1, i+1\}$ where addition and subtraction are done modulo $2k+1$. Let I be a maximum independent set of $X(\overline{C_{2k+1}}, s)$ and let $X = (X_1, X_2, \ldots, X_{2k+1})$ be the characteristic vector of an element picked uniformly at random from I. By property (2) we have that $H(X) = \log |I|$. We will now proceed to bound $H(X)$ from above.

For each $i \in [2k+1]$, let $J_i = [2k+1] \setminus \{i-1, i+1\}$ and $K_i = [2k+1] \setminus \{i\}$. Observe that X_i is uniquely determined from the values of all X_j with $j \neq i$ and so $H(X) = H(K_i)$. Observe also that for any $S \subseteq J_i$ we have that

$$H(X) = H(K_{i-1}) \leqslant H(J_i) + H(S \cup \{i+1\}) - H(S) \leqslant 2k - 2 + H(S \cup \{i+1\}) - H(S).$$

Indeed, the first inequality follows from property (6) by taking $A = J_i$ and $B = S \cup \{i+1\}$, and the second inequality follows from property (2). Similarly, we have

$$H(X) \leqslant 2k - 2 + H(S \cup \{i-1\}) - H(S),$$

since X_i is determined by X_{i-1} and $x_i + 1$.

Let $T = \{3, 5, \ldots, 2k+1\}$. Applying the second inequality with $S = T$ and $i = 2$ we get

$$H(X) \leqslant 2k - 2 + H(T \cup \{1\}) - H(T)$$

Applying the first inequality repeatedly with $S = T \cup \{1\}, T \cup \{1,2\}, T \cup \{1,2,4\}, \ldots, T \cup \{1,2,4,\ldots,2k-6\}$ and $i = 1, 3, 5, \ldots, 2k-5$ respectively we get

$$H(X) \leqslant 2k - 2 + H(T \cup \{1,2\}) - H(T \cup \{1\})$$
$$H(X) \leqslant 2k - 2 + H(T \cup \{1,2,4\}) - H(T \cup \{1,2\})$$
$$\cdots$$
$$H(X) \leqslant 2k - 2 + H(T \cup \{1,2,4,\ldots,2k-4\}) - H(T \cup \{1,2,4,\ldots,2k-6\})$$

and summing all these together with the previous inequality up we get

$$(k-1)H(X) \leqslant 2(k-1)^2 + H(J_{2k-1}) - H(T) \leqslant 2k(k-1) - H(T).$$

Since also

$$H(X) = H(3, 4, \ldots, 2k+1) \leqslant H(T) + H(4, 6, \ldots, 2k) \leqslant H(T) + (k-1),$$

we get

$$kH(X) \leqslant (2k+1)(k-1)$$

and the result follows. $\square$

Here we have found that if G is a minimally imperfect graph then its guessing number is at least $\text{gn}(G, s) \leqslant n - \chi_f(\overline{G})$ for every value of s. We also know from the previous section that equality holds for infinitely many values of s. In particular it holds for every s which is a perfect b-power for some b which can be determined from G. We also know from Section 2 that $\text{gn}(G, s) \geqslant n - \chi_f(\overline{G})$ for every perfect graph G, where in fact we even have equality for every value of s. We conjecture that this inequality is true in general.

Conjecture 6.4. *For every graph G and every positive integer s we have $\mathrm{gn}(G, s) \leqslant n - \chi_f(\overline{G})$. In particular $\mathrm{gn}(G, s) = n - \chi_f(\overline{G})$ for every value of s for which the fractional clique strategy can be used and so $\mathrm{gn}(G) = n - \chi_f(\overline{G})$.*

It would also be of interest to find the guessing number for values s where the fractional clique strategy cannot be used. Even for C_5 we know only the value for $s = 3$, and we can obtain some bounds for other non-perfect squares via Lemmas 3.3 and 3.5.

For values of s other than perfect squares we do not even have a full understanding of the guessing numbers for odd cycles.

Example 6.5. For $s = 2$ we have computed the independence numbers of the code graphs for small odd cycles using a standard linear programming solver. We found that $\alpha(X(C_7, 2)) = 8$, $\alpha(X(C_9, 2)) = 16$ and $\alpha(X(C_{11}, 2)) = 32$, i.e. exactly the values attained by the clique strategy on these graphs. For C_7, we also found that $\alpha(x(C_7, 3)) = 29 > 3^3$, so here the clique strategy is not optimal.

We close this section with two problems.

Problem 6.6. *Is the clique strategy optimal for $\mathrm{gn}(C_{2k+1}, 2)$ for all $k \geqslant 3$?*

Problem 6.7. *Is the clique strategy optimal for $\mathrm{gn}(C_{2k+1}, 3)$ for all $k \geqslant 4$?*

7 Generalisations

7.1 Nonuniform random numbers

In the results so far we have assumed that all random numbers x_v were picked uniformly at random from the single set $\{1, \ldots, s\}$. However there are two extensions which could naturally appear. The first is to assume that the random numbers are independent as before but each x_v comes from some non-uniform distribution P_v, which may depend on v. The second is to allow the different x_v to come from sets S_v which may vary in size with v. The code graph approach from the previous section can easily be adapted to both of these modifications.

In order to deal with distinct P_v we can give each string in $\Omega(n, s)$ a weight $\prod_v P_v(x_v)$ and use a strategy given by a maximum weight independent set. With this weighting the maximum weight will correspond to the maximum winning probability.

Likewise the code graph can easily be adapted to differing S_v by replacing the uniform $\Omega(n, s)$ with the product set $\prod_v S_v$, using the same neighbour relation as before.

7.2 Directed graphs

In [Rii07b] the emphasis was on directed graphs, as this is the case used in network coding and boolean circuits. While our discussion has mainly been in terms of undirected graphs there are several methods and bounds which can be generalized to directed graphs as well. The directed case has been studied in great detail in [GR10], where the code graph for

directed graphs is introduced and studied. We will here only give a brief discussion of the connections to our work on fractional covergins.

The code graph $X(G, s)$ can be defined for a directed graphs by replacing the neighbourhood of v used in in Definition 5.1 with the set of in-neighbours of v, i.e. the set of vertices from which there is a directed edge to v. As before the guessing number is given in terms of the maximum independent sets of $X(D, s)$.

Recall that a directed graph is *acyclic* if it is not possible to walk along a cycle by following the directions of its edges. Given a directed graph D let $\vec{\alpha}(D)$ denote the size of the largest induced subgraph of D which is acyclic. This parameter can be used in place of the independence number to bound the guessing number of D. The following theorem was proven in [WCR09]

Theorem 7.1 ([WCR09]). *Let D be a digraph on n vertices. Then $gn(D, s) \leqslant n - \vec{\alpha}(D)$*

As in the clique strategy we may construct strategies for larger directed graphs by partitioning their vertex set into disjoint copies of smaller graphs, following the optimal strategies independently on each part of the partition. However unlike for ordinary graphs, where the cliques played a pivotal role, there does not seem to be a canonical family of directed graphs to partition the graph into. The most natural analogue of the complete graphs are tournaments, which are directed complete graphs, but since there are tournaments which are acyclic, some additional conditions seem to be required. Given any family of directed graphs with known values of their guessing numbers we may also use fractional covers using these graphs in the same way as we did in Theorem 4.9.

7.3 Infinite graphs

It is also natural to ask what happens if we play the guessing game on infinite graphs. We begin by consider the countably infinite complete graph with $s = 2$, say. As a comparison, recall that the winning probability on any finite complete graph with $s = 2$ is $1/2$.

Consider the set $\{0, 1\}^{\mathbb{N}}$ of countably infinite $0, 1$-sequences. We would like to find a 'large' subset A of this set such that any two sequences in A differ in at least two places. Having found such an A, we can then define a natural strategy $\mathcal{F}$ which has A as its set of fixed points. It is not difficult to partition $\{0, 1\}^{\mathbb{N}}$ into two sets A and B such that both of them have the property that any two of their sequences differ in at least two places: One just defines an equivalence relation $\sim$ on $\{0, 1\}^{\mathbb{N}}$ by saying that two sequences x, y are equivalent if and only if they differ in a finite number of places. For every equivalence class C of $\sim$ we then pick a sequence $x \in C$ and define A_x to be the set of all sequences in C which differ from x in an odd number of places and B_x to be the set of all sequences in C which differ from x in an even number of places. It is easy to check that $A = \cup_x A_x$ and $B = \cup_x B_x$ have the required properties.

This is great as it seems to show that the winning probability for this graph is also $1/2$. However there is one catch. The sets A, B above were constructed using the axiom of choice and in fact it is not too difficult to see that they are not measurable. For our purposes however, we want to calculate the winning probability and so we must require

that each individual strategy $f_v : \{0,1\}^{\mathbb{N}} \to \{0,1\}$ is measurable. So the question now translates to finding a measurable subset A of $\{0,1\}^{\mathbb{N}}$ such that any two sequences of A differ in at least two places and A has as large a measure as possible. Then for each $k \in \mathbb{N}$, the strategy $f_k : \{0,1\}^{\mathbb{N}} \to \{0,1\}$ defined by $f_k(x) = 0$ if the sequence x' obtained from x by making its k-th digit 0 belongs to A and $f_k(x) = 1$ otherwise, is a measurable strategy. Moreover the set of fixed points A' of the strategy mapping $\mathcal{F} = (f_k)_{k \in \mathbb{N}}$ is a measurable subset of $\{0,1\}^{\mathbb{N}}$ containing A and so it has the same measure as A.

We proceed to show that any such A must have measure 0. Observe that this is immediate if A is an open subset of $\{0,1\}^{\mathbb{N}}$ (under the product topology). Indeed the sets $U_{x,n} = \{y : y_i = x_i \ \forall \ i \leqslant n\}$ form a basis for the topology and clearly no such set can be a subset of A as $x, x' \in U_{x,n}$ where x' is obtained from x by changing its $(n+1)$-th digit. If there is such a set A of positive measure, then we can find a basic open set $U_{x,n}$ such that $m(A \cap U_{x,n}) \geqslant 51 m(U_{x,n})/100$. Now let A_0 be the set of all sequences in $A \cap U_{x,n}$ whose $(n+1)$-th digit is 0 and let B_0 be the set of all sequences obtained from A_0 by changing the $(n+1)$-th digit to 1. We define A_1 and B_1 analogously. Now observe that A_0, A_1, B_0, B_1 are disjoint subsets of $U_{x,n}$ with $m(A_0) + m(A_1) \geqslant 51 m(U_{x,n})/100$ and $m(A_0) = m(B_0), m(A_1) = m(B_1)$. It follows that $m(A_0 \cup A_1 \cup B_0 \cup B_1) > m(U_{x,n})$, a contradiction.

So even in the complete countably infinite graph it turns out that the winning probability for the guessing game is 0.

Acknowledgments

References

[ACLY00] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46(4):1204–1216, 2000.

[AS08] Noga Alon and Joel H. Spencer. *The probabilistic method.* John Wiley & Sons Inc., Hoboken, NJ, third edition, 2008.

[Ber63] C. Berge. Perfect graphs. In *Six Papers on Graph Theory.* Indian Statistical Institute, 1963.

[CRST06] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Ann. of Math. (2)*, 164(1):51–229, 2006.

[CT06]   Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2006.

[GR10]   M. Gadouleau and S. Riis. Graph-theoretical Constructions for Graph Entropy and Network Coding Based Communications. arXiv:1010.2619, October 2010.

[Lov72]   L. Lovász. A characterization of perfect graphs. *J. Combinatorial Theory Ser. B*, 13:95–98, 1972.

[Rad03]   J. Radhakrishnan. *Computational Mathematics, Modelling and Algorithms*, chapter Entropy and counting. Narosa Publishing House, 2003.

[RAMS04]   Arathi Ramani, Fadi A. Aloul, Igor L. Markov, and Karem A. Sakallah. Breaking instance-independent symmetries in exact graph coloring. In *Proceedings of the conference on Design, automation and test in Europe - Volume 1*, DATE '04, pages 10324–, Washington, DC, USA, 2004. IEEE Computer Society.

[Rii07a]   S. Riis. Graph Entropy, Network Coding and Guessing games. arXiv:0711.4175, November 2007.

[Rii07b]   Søren Riis. Information flows, graphs and their guessing numbers. *Electron. J. Combin.*, 14(1):R44, 2007.

[SU97]   Edward R. Scheinerman and Daniel H. Ullman. *Fractional graph theory*. John Wiley & Sons Inc., New York, 1997.

[TT77]   Robert Endre Tarjan and Anthony E. Trojanowski. Finding a maximum independent set. *SIAM J. Comput.*, 6(3):537–546, 1977.

[WCR09]   Taoyang Wu, Peter Cameron, and Søren Riis. On the guessing number of shift graphs. *J. Discrete Algorithms*, 7(2):220–226, 2009.

[Yun09]   Sun Yun. *Network Coding and Graph Entropy*. PhD thesis, Queen Mary University of London, 2009.