Journal of Combinatorial Optimization (2022) 44:2963–2980
https://doi.org/10.1007/s10878-021-00825-y

On zero-error codes produced by greedy algorithms

Marcin Jurkiewicz[1]

Accepted: 5 October 2021 / Published online: 2 November 2021

Abstract
We present two greedy algorithms that determine zero-error codes and lower bounds on the zero-error capacity. These algorithms have many advantages, e.g., they do not store a whole product graph in a computer memory and they use the so-called distributions in all dimensions to get better approximations of the zero-error capacity. We also show an additional application of our algorithms.

Keywords Shannon capacity · Zero-error code · Greedy algorithm · Strong product · Independence number

1 Preliminaries

Let $G = (V, E)$ be a graph. The number of vertices and edges of G we often denote by n and m, respectively, thus $|V(G)| = n$ and $|E(G)| = m$. If $u, v \in V(G)$ and $\{u, v\} \in E(G)$, then we say that u is *adjacent* to v and we write $u \sim v$. The *open neighborhood* of a vertex $v \in V(G)$ is $N_G(v) = \{u \in V(G)\colon \{u, v\} \in E(G)\}$, and its *closed neighborhood* is the set $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of a vertex v, denoted by $d_G(v)$, is the cardinality of its open neighborhood. The *minimum* and *maximum degree* of G is the minimum and maximum degree among the vertices of G and is denoted by $\delta(G)$ and $\Delta(G)$, respectively. A graph G is *regular* if $\delta(G) = \Delta(G)$. By the *complement* of G, denoted by $\overline{G}$, we mean a graph which has the same vertices as G, and two vertices of $\overline{G}$ are adjacent if and only if they are not adjacent in G. If U is a subset of vertices of G, we write $G[U]$ and $G - U$ for $(U, E(G) \cap [U]^2)$ and $G[V(G)\backslash U]$, respectively. Furthermore, if $U = \{v\}$, then we write $G - v$ rather than $G - \{v\}$.

A preliminary version of this work was presented at COCOA'20 (Jurkiewicz 2020).

✉ Marcin Jurkiewicz
marjurki@pg.edu.pl

[1] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland

Given two graphs $G_1 = (V(G_1), E(G_1))$ and $G_2 = (V(G_2), E(G_2))$, the *strong product* $G_1 \boxtimes G_2$ is defined as follows. The vertices of $G_1 \boxtimes G_2$ are all pairs of the Cartesian product $V(G_1) \times V(G_2)$. There is an edge between (v_1, v_2) and (u_1, u_2) if and only if $\{v_1, u_1\} \in E(G_1)$ and $\{v_2, u_2\} \in E(G_2)$, or $v_1 = u_1$ and $\{v_2, u_2\} \in E(G_2)$, or $v_2 = u_2$ and $\{v_1, u_1\} \in E(G_1)$. The *union* $G_1 \cup G_2$ is defined as $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. In addition, if $\circ$ is a binary graph operation, then we write $G^{\circ r}$ to denote the rth *power* of G, i.e., $G \circ G \circ \cdots \circ G$, where G occurs r-times.

A *clique* (*independent vertex set*, resp.) in a graph $G = (V, E)$ is a subset $V' \subseteq V$ such that all (no, resp.) two vertices of V' are adjacent. The size of a largest clique (independent vertex set, resp.) in a graph G is called the *clique* (*independence*, resp.) *number* of G and is denoted by $\omega(G)$ ($\alpha(G)$, resp.). A *split graph* is one whose vertex set can be partitioned as the disjoint union of an independent set and a clique. A *legal coloring* of a graph G is an assignment of colors to the vertices of G ($C\colon V(G) \to \mathbb{N}$) such that any two adjacent vertices are colored differently.

2 Introduction

A *discrete channel* $W\colon \mathcal{X} \to \mathcal{Y}$ (or simply W) is defined as a stochastic matrix[1] whose rows are indexed by the elements of a finite *input set* $\mathcal{X}$ while the columns are indexed by a finite *output set* $\mathcal{Y}$. The (x, y)th entry is the probability $W(y|x)$ that y is received when x is transmitted. A sequence of channels $\{W^n : \mathcal{X}^n \to \mathcal{Y}^n\}_{n=1}^{\infty}$, where $W^n : \mathcal{X}^n \to \mathcal{Y}^n$ is the nth *direct power* of W, i.e.,

$$W^n(y_1 y_2 \dots y_n | x_1 x_2 \dots x_n) = \prod_{i=1}^{n} W(y_i | x_i)$$

and $\mathcal{X}^n$ is the nth Cartesian power of $\mathcal{X}$, is called a *discrete memoryless channel* (DMC) with stochastic matrix W and is denoted by $\{W : \mathcal{X} \to \mathcal{Y}\}$ or simply $\{W\}$. See Shannon (1956), Csiszár and Körner (2011), Körner and Orlitsky (1998), Cover and Thomas (2006) and McEliece (2004) for more details.

Let $W : \mathcal{X} \to \mathcal{Y}$ be a discrete channel. We define the *ω-characteristic graph* G of W as follows. Its vertex set is $V(G) = \mathcal{X}$ and its set of edges $E(G)$ consists of input pairs that cannot result in the same output, namely, pairs of orthogonal rows of the matrix W. We define *α-characteristic graph* $G(W)$ (we call it *characteristic graph* for short) of W as the complement of the ω-characteristic graph of W. Let $\{W : \mathcal{X} \to \mathcal{Y}\}$ be a DMC and so $W : \mathcal{X} \to \mathcal{Y}$ is the corresponding discrete channel. We define the *characteristic graph* $G(\{W\})$ of the discrete memoryless channel $\{W\}$ as $\{G(W^n)\}_{n=1}^{\infty}$. The *Shannon* (*zero-error*) *capacity* $C_0(W)$ of the DMC $\{W : \mathcal{X} \to \mathcal{Y}\}$ is defined as $C(G(W))$, where

$$C(G) = \sup_{n \in \mathbb{N}} \frac{\log \alpha(G^{\boxtimes n})}{n} = \lim_{n \to \infty} \frac{\log \alpha(G^{\boxtimes n})}{n}.$$

[1] We assume that W is non-empty.

See Csiszár and Körner (2011), Körner and Orlitsky (1998), Cover and Thomas (2006) and McEliece (2004) for more details. Let G be the characteristic graph of W and $\Theta(G) = \sup_{n\in\mathbb{N}} \sqrt[n]{\alpha(G^{\boxtimes n})}$. Then $\Theta(G)$ uniquely determines $C_0(W)$.

Let $W : \mathcal{X} \to \mathcal{Y}$ be a discrete channel with the characteristic graph G. A sequence of input letters is called an *input word*. Input words $x_1x_2\ldots x_n \in \mathcal{X}^n$ and $x'_1x'_2\ldots x'_n \in \mathcal{X}^n$ are *orthogonal* if the vectors $W^n(\cdot|x_1x_2\ldots x_n)$ and $W^n(\cdot|x'_1x'_2\ldots x'_n)$ are orthogonal. A *zero-error code* of block length n for a DMC is defined by a set of mutually orthogonal input words (Körner and Orlitsky 1998; Cover and Thomas 2006). Furthermore, an independent set I of the characteristic graph $G(W^n)$ corresponds to the zero-error code for W^n and $G(W^n)$ is the same as $G^{\boxtimes n}$ (Shannon 1956; Körner and Orlitsky 1998).

The research on zero-error codes was initiated by Shannon in 1956. He found capacities of a class of channels (graphs) that does not yield additional information benefits (Shannon 1956) and he provided a method which enables constructing codes for these channels. The research was continued by, among others, Lovász (1979) in his IEEE Information Theory Society award work, in which he determined the values of Shannon capacities for some channels with effective codes using the so-called Lovász function. The class of channels examined by Lovász is represented by the so-called vertex-transitive, self-complementary graphs. It is the only one class containing channels with effective codes, for which the an explicit formula of the Shannon capacity is known.

Recently, Polak and Schrijver (2019) and Mathew and Östergård (2017) made some progress in research on channels represented by strong powers of cycles. Moreover, Boche and Deppe (2020) proved that the zero-error capacity is uncomputable in the Banach–Mazur and Borel–Turing senses. Earlier, Alon and Lubetzky (2006) showed that the series of independence numbers of strong powers of a fixed graph can exhibit a complex and unpredictable structure. In this article, we propose polynomial algorithms that approximate the capacity for some channels.

In the next section, we describe the so-called fractional independence number defined by Rosenfeld (1967), which is strongly related to the considered problem.

3 Fractional independence number

Computing the independence number of a graph $G = (V, E)$ can be formulated by the following integer program.

$$\begin{aligned} \text{Maximize} \quad & \sum_{v\in V} x_v \\ \text{subject to} \quad & \underset{\{u,v\}\in E}{\forall} x_u + x_v \le 1 \text{ and } \underset{v\in V}{\forall} x_v \in S, \end{aligned} \tag{1}$$

where $S = \{0, 1\}$. Now let $S = [0, 1]$. Given a graph G, by $\alpha_2^*(G)$ we denote the optimum of the objective function in the integer program (1). However, for a graph G

and a set of not necessarily all its cliques[2] $\mathcal{C}$ by $\alpha^*_{\mathcal{C}}(G)$ we denote the optimum of the objective function in the following integer program.

$$\begin{aligned} \text{Maximize} \quad & \sum_{v \in V} x_v \\ \text{subject to} \quad & \underset{C \in \mathcal{C}}{\forall} \sum_{v \in C} x_v \leq 1 \text{ and } \underset{v \in V}{\forall} x_v \in [0, 1]. \end{aligned} \tag{2}$$

If $\mathcal{C}$ is the set of all maximal cliques of size at most r in G, then we denote $\alpha^*_{\mathcal{C}}(G)$ by $\alpha^*_r(G)$. If $\mathcal{C}$ contains the set of all cliques (or equivalently all maximal cliques) of G, then we denote $\alpha^*_{\mathcal{C}}(G)$ by $\alpha^*(G)$ and it is called *the fractional independence number* of G. It is worth to note that α^* is multiplicative with respect to the strong product (Scheinerman and Ullman 2011).

The following results present some properties of the linear program (2). In particular, the first observation establishes an order between the above-mentioned measures.

Observation 1 *Let G be a graph and $\mathcal{C}$, $\mathcal{C}'$ be sets of its cliques. If $\mathcal{C}' \subseteq \mathcal{C}$, then $\alpha^*_{\mathcal{C}}(G) \leq \alpha^*_{\mathcal{C}'}(G)$.*

Observation 2 *Let G be a graph. If $\omega(G) = r$, then $\alpha^*(G) = \alpha^*_r(G)$.*

Lemma 1 *For every graph G and a non-empty set of its cliques $\mathcal{C}$ we have*

$$\frac{|V(G)|}{\omega(G)} \leq \alpha^*_{\mathcal{C}}(G) \leq \frac{|\mathcal{C}|}{\varsigma(\mathcal{C})} + R_{\mathcal{C}}(G), \tag{3}$$

where $\varsigma(\mathcal{C}) = \min\{\sum_{C \in \mathcal{C}} |\{v\} \cap C| \colon v \in \bigcup_{C \in \mathcal{C}} C\}$ and $R_{\mathcal{C}}(G) = |V(G)| - |\bigcup_{C \in \mathcal{C}} C|$. Furthermore, the equalities hold in the inequality chain (3) if G is vertex-transitive[3] and $\mathcal{C}$ is the set of all largest cliques in G.

Proof It is well known (Gross et al. 2014) that for every graph G we have

$$\alpha^*(G) \geq \frac{|V(G)|}{\omega(G)}. \tag{4}$$

From (4) and Observation 1, the left inequality holds in (3).

Given a linear program (2) and its optimum $\alpha^*_{\mathcal{C}}(G)$. Since $\sum_{C \in \mathcal{C}} \sum_{v \in C} x_v \leq |\mathcal{C}|$, so

$$\sum_{v \in V(G)} x_v \leq \frac{|\mathcal{C}|}{\varsigma(\mathcal{C})} + R_{\mathcal{C}}(G).$$

Hence $\alpha^*_{\mathcal{C}}(G) \leq |\mathcal{C}|/\varsigma(\mathcal{C}) + R_{\mathcal{C}}(G)$.

[2] We emphasize that the number of maximal cliques in G is at most exponential with respect to $|V(G)|$ (Fomin and Kratsch 2010), while the number of edges in G is at most quadratic with respect to $|V(G)|$.

[3] A graph is *vertex transitive* if for any two vertices u and v of this graph, there is an automorphism such that the image of u is v.

If G is vertex-transitive and $\mathcal{C}$ is the set of all largest cliques in G, then $\mathcal{C}$ covers the whole vertex set, i.e., $V(G) = \bigcup_{C \in \mathcal{C}} C$. Hence $R_{\mathcal{C}}(G) = 0$. Furthermore, every vertex is contained in the same number of largest cliques. Hence $\varsigma(\mathcal{C})|V(G)| = \omega(G)|\mathcal{C}|$. □

It is interesting that the measure α^* has a particular interpretation in information theory (Shannon 1956; Körner and Orlitsky 1998).

4 Capacity approximation

It is well known (Shannon 1956) that[4]

$$\alpha(G) \leq \sqrt[i]{\alpha(G^{\boxtimes i})} \leq \Theta(G) \leq \alpha^*(G) \tag{5}$$

for each positive integer i. A graph G is of *type* I if $\Theta(G) = \alpha(G)$, otherwise is of *type* II. Furthermore, Hales (1973) showed that for arbitrary graphs G and H we have

$$\alpha(G \boxtimes H) \leq \min\{\alpha(G)\alpha^*(H), \alpha(H)\alpha^*(G)\}.$$

In contrast to the above results, in the next section we use the fractional independence number to calculate lower bounds on the Shannon capacity and the independence number of strong products.

A function $\beta\colon \mathcal{G} \to \mathbb{R}$ is *supermultiplicative* (resp. *submultiplicative*) on $\mathcal{G}$ with respect to the operation $\circ$, if for any two graphs $G_1, G_2 \in \mathcal{G}$ we have $\beta(G_1 \circ G_2) \geq \beta(G_1) \cdot \beta(G_2)$ (resp. $\beta(G_1 \circ G_2) \leq \beta(G_1) \cdot \beta(G_2)$). A supermultiplicative and submultiplicative function is called *multiplicative*. The independence number α is supermultiplicative on the set of all graphs with respect to the strong product, i.e., $\alpha(G \boxtimes H) \geq \alpha(G) \cdot \alpha(H)$ for any graphs G and H. Let B be a lower bound on the independence number α, i.e., $\alpha(G) \geq B(G)$. If $B(G^{\boxtimes i}) > (\alpha(G))^i$ $(i \geq 2)$, then G is of type II and is more interesting from an information theory point of view (Shannon 1956). It is possible if $B(G^{\boxtimes i}) > (B(G))^i$. Thus we require that B has the last two properties for at least one graph, i.e., B *recognizes* some graphs of type II.

The *residue* R of a graph G of degree sequence $S : d_1 \geq d_2 \geq d_3 \cdots \geq d_n$ is the number of zeros obtained by the iterative process consisting of deleting the first term d_1 of S, subtracting 1 from the d_1 following ones, and re-sorting the new sequence in non-increasing order (Favaron et al. 1993). It is well known (Favaron et al. 1991) that $\alpha(G) \geq R(G)$. Unfortunately, the following negative result holds.

Proposition 1 *Let G and H be regular*[5] *or split graphs. Then $R(G \boxtimes H) \leq R(G) \cdot R(H)$.*

Proof Let G and H be regular graphs. For a regular graph G, from Favaron et al. (1991), we have $R(G) = \lceil \sum_{i=1}^{n}(1/(1 + d_i)) \rceil = \lceil (n/(1 + d(G))) \rceil$, where $d(G)$

[4] There is a better upper bound on $\Theta(G)$, the so-called Lovász theta function (Lovász 1979).

[5] This part of the proposition was found by my colleague Tytus Pikies (2015, personal communication).

is the degree of each vertex of G. From Jurkiewicz (2017) we know that the ceiling function is submultiplicative on non-negative real numbers with respect to the multiplication. Hence $R(G \boxtimes H) = \lceil |V(G)||V(H)|/(1+(d(G)d(H)+d(G)+d(H)))\rceil \leq \lceil |V(G)|/(1+d(G))\rceil \lceil |V(H)|/(1+d(H))\rceil = R(G) \cdot R(H)$, since a strong product of regular graphs is regular.

Let G and H be split graphs. From Barrus (2012) and Hammack et al. (2011), we have $\alpha(G) = R(G)$ and $\alpha(G \boxtimes H) = \alpha(G) \cdot \alpha(H)$, respectively. Finally, we have $R(G \boxtimes H) \leq \alpha(G \boxtimes H) = \alpha(G) \cdot \alpha(H) = R(G) \cdot R(H)$. □

We conjecture that the residue is submultiplicative on the set of all graphs with respect to the strong product. This probably means that the residue does not recognize any graphs of type II. There are more such bounds, e.g., the average distance (Jurkiewicz 2017), the Caro–Wei bound and the Wilf bound (Jurkiewicz and Pikies 2015). On the other hand, it is hard to find bounds that recognize at least one graph of type II.

5 Greedy algorithm MIN

In this section, we analyze, in the context of DMCs codes, the so-called greedy algorithm Min (Algorithm 5.1) that determines an independent set and a lower bound on the independence number of a graph (Harant and Schiermeyer 2001). The algorithm Min has complexity $O(n^2)$. Similar greedy algorithms can be found in literature (Borowiecki and Rautenbach 2015).

Algorithm 5.1 Greedy Algorithm Min

```
1: function Min(G)
2:     I ← ∅
3:     while V(G) ≠ ∅ do
4:         assign to v* an element v ∈ V(G) with the smallest d(v)
5:         G ← G − N_G[v*]
6:         I ← I ∪ {v*}
7:     return I
```

A greedy algorithm always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution (Cormen et al. 2009). Vertices chosen (in such a way) by Min often strongly block an eventual choice of vertices in a further stage of the algorithm, making generated independent sets are small, especially for strong products of graphs of type II. In Table 1, we summarize results produced by Min for these graphs. On the other hand, Min works well for strong products of investigated graphs of type I. Although channels represented by graphs of type I do not yield additional information benefits, we also need a fast method that determines zero-error codes for these channels. There are at least two ways to do this, i.e., we can run Min on the characteristic graph G of a channel (since I^n is an independent set of $G^{\boxtimes n}$ if I is an independent set of G) or directly on a strong power of G. In Table 2, we summarize our results produced by

Table 1 For each graph $G \in \mathcal{G}^+_{n,2} = \{H : \alpha(H^{\boxtimes 2}) > \alpha^2(H) \wedge |V(H)| = n\}$ we determined $T = 10^{(11-n)}$ independent sets of the graph $G^{\boxtimes 2}$ using the algorithm MIN and from these T sets, we chose the largest one, which is denoted by I

Greedy algorithm MIN (results for $\mathcal{G}^+_{n,2}$)				
n	$\lvert\mathcal{G}^+_{n,2}\rvert$	$\lvert I\rvert \le \alpha^2(G)$	$\lvert I\rvert > \alpha^2(G)$	$\lvert I\rvert = \alpha(G^{\boxtimes 2})$
5	1	1	0	0
6	4	4	0	0
7	36	33	3	3
8	513	474	39	38
9	16,015	15,536	479	475
10	908,794	900,764	8030	7746

Table 2 For each graph $G \in \mathcal{G}^0_{n,2} = \{H : \alpha(H^{\boxtimes 2}) = \alpha^2(H) \wedge |V(H)| = n\}$ we determined an independent set I of the graph $G^{\boxtimes 2}$

Greedy algorithm MIN (results for $\mathcal{G}^0_{n,2}$)			
n	$\lvert\mathcal{G}^0_{n,2}\rvert$	$\lvert I\rvert < \alpha^2(G)$	$\lvert I\rvert = \alpha(G^{\boxtimes 2})$
5	33	0	33
6	152	0	152
7	1008	20	988
8	11,833	300	11,533
9	258,653	10,076	248,577
10	11,096,374	682,341	10,414,033

The independent set I is a larger set of $I' \times I'$ and I'', where $I' = \text{MIN}(G)$ and $I'' = \text{MIN}(G^{\boxtimes 2})$. In addition, we obtained $|I| = \alpha(G^{\boxtimes 2})$ for all graphs on $n = 1, 2, 3, 4$

MIN for some graphs of type I. It is important to note that, for all results, we randomly[6] chose vertices with the smallest degrees in MIN (in line 4).

6 Modification of greedy algorithm MIN

In this section, we present a new greedy algorithm that produces an independent set (a DMC code) and a lower bound on the independence number of a strong product. This value, from (5), also determines a lower bound on the Shannon capacity.

We try to improve MIN, since from our research it follows that it does not work well, i.e., it recognizes a small number of graphs of type II. Our goal is to get larger independent sets for strong products of graphs of type II by a modification of the mentioned algorithm. We begin by introducing definitions required in the rest of the paper.

[6] If the first/last found vertex with the smallest degree is chosen, then two last columns in Table 1 contain zeros.

A *semigroup* is a set S with an associative binary operation on S. A *semiring* is defined as an algebra $(S, +, \cdot)$ such that $(S, +)$ and $(S, \cdot)$ are semigroups and for any $a, b, c \in S$ we have $a \cdot (b + c) = a \cdot b + a \cdot c$, $(b + c) \cdot a = b \cdot a + c \cdot a$ (Adhikari and Adhikari 2014). Note that $(\mathcal{G}, \cup, \boxtimes)$ is a semiring, where $\mathcal{G}$ is the set of all finite graphs. In addition, $\cup$ and $\boxtimes$ are commutative operations with neutral elements $(\emptyset, \emptyset)$ and K_1, respectively.

Lemma 2 *Let p, r be positive integers and $G_1, G_2, \ldots, G_r$ be graphs. Then*

$$\left(\bigcup_{i\in[r]} G_i\right)^{\boxtimes p} = \bigcup_{p_1+p_2+\ldots+p_r=p} \left[\binom{p}{p_1, p_2, \ldots, p_r} \boxtimes_{i\in[r]} G_i^{\boxtimes p_i}\right]$$

and

$$\alpha\left(\left(\bigcup_{i\in[r]} G_i\right)^{\boxtimes p}\right) = \sum_{p_1+p_2+\ldots+p_r=p} \left[\binom{p}{p_1, p_2, \ldots, p_r} \alpha\left(\boxtimes_{i\in[r]} G_i^{\boxtimes p_i}\right)\right],$$

where summations extend over all ordered sequences $(p_1, p_2, \ldots, p_r)$ of nonnegative integers that sum to p.

Proof The first part of the theorem can be proved in analogous way to the one in (Loehr 2011, Theorem 2.12) for rings (we only need the above mentioned properties of the semiring $(\mathcal{G}, \cup, \boxtimes)$).

The second part of the theorem follows from the fact that the independence number is multiplicative with respect to the disjoint union $\cup$ for all graphs. $\square$

The considered modification of the greedy algorithm takes as input arbitrary graphs $G_1, G_2, \ldots, G_r$ and produces as output an independent set of $G^{\boxtimes} = G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_r$. From Lemma 2, we can find connected components of $G^{\boxtimes}$. Hence, our greedy algorithm can be applied to each connected component of $G^{\boxtimes}$ separately, or to the entire graph $G^{\boxtimes}$ at once. We prefer the first method.

The next step of our modification is a reduction of factors of a strong product. For each $i \in \{1, 2, \ldots, r\}$ and any $u, v \in V(G_i)$, if $N_{G_i}[u] \subseteq N_{G_i}[v]$, then $\alpha(G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_i \boxtimes \ldots \boxtimes G_r) = \alpha(G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes (G_i - v) \boxtimes \ldots \boxtimes G_r)$ (Jurkiewicz 2020). Let G be a factor of a strong product $G^{\boxtimes}$, for example $G = G_i$. Let $>$ be a strict total order on $V(G)$. We reduce the factor G by Algorithm 6.1 (Reduction GR), which has complexity $O(\Delta^2 m)$. This algorithm is correct (in the considered context) since we remove vertices from the strong product, and hence we can only decrease or leave unchanged its independence number.

For some graphs, which we take as input, e.g., for a path on $n \geq 6$ vertices, we need to recursively repeat (at most n times) the algorithm GR to get a smaller graph. Sometimes, the algorithm GR produces vertices with degree zero. Such vertices should be removed from a graph, but taken into account in the outcome.

Let G be a graph and k be a positive integer. Let A be a k-tuple of subsets of $V(G)$. By $B_G(A)$ we denote a sequence containing upper bounds on $\alpha(G[A_i])$ for

Algorithm 6.1 Reduction GR

function GR(G)
  $R \leftarrow \emptyset$
  for all $\{u, v\} \in E(G)$ **do**
    if $u > v$ **then**
      if $N_G[u] = N_G[v]$ **then**
        $R \leftarrow R \cup \{u\}$
      else if $N_G[u] \subseteq N_G[v]$ **then**
        $R \leftarrow R \cup \{v\}$
      else if $N_G[v] \subseteq N_G[u]$ **then**
        $R \leftarrow R \cup \{u\}$
  return $G - R$

$i \in \{1, 2, \ldots, k\}$. Let $G = 2K_3$. Then, for example, $V(G) = \{1, 2, \ldots, 6\}$, $E(G) = \{\{1, 2\}, \{2, 3\}, \{3, 1\}, \{4, 5\}, \{5, 6\}, \{6, 4\}\}$ and

$$B_G((\{1, 2, 3\}, \{4, 5, 6\})) = (1, 1).$$

Let A' be a k'-tuple of subsets of $V(G)$. A *distribution* $D_G(A')$ is a k'-tuple of non-negative integers, and is our prediction about an arrangement of independent vertices of G in sets from A'. Let $G^{\boxtimes} = G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_r$. Let $i \in \{1, 2, \ldots, r\}$, $S \subseteq V(G_i)$ and $V_{G_i}(S) = V(G_1) \times V(G_2) \times \cdots \times V(G_{i-1}) \times S \times V(G_{i+1}) \times \cdots \times V(G_r)$. From Hammack et al. (2011), for each clique Q of G_i we have

$$\alpha\left(\mathop{\boxtimes}_{j \in [r]\setminus\{i\}} G_j\right) = \alpha(G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_{i-1} \boxtimes G_i[Q] \boxtimes G_{i+1} \boxtimes \ldots \boxtimes G_r).$$

Thus, if $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_k\}$ ($k \in \mathbb{N}_+$) is a set of cliques of G_i and

$$\alpha_i \geq \alpha\left(\mathop{\boxtimes}_{j \in [r]\setminus\{i\}} G_j\right), \tag{6}$$

then we can choose $B_{G^{\boxtimes}}((V_{G_i}(Q_1), V_{G_i}(Q_2), \ldots, V_{G_i}(Q_k))) = (\alpha_i, \alpha_i, \ldots, \alpha_i)$, where α_i occurs k times. Let

$$\alpha_i = \left\lfloor \prod_{j \in [r]\setminus\{i\}} \alpha^*_{E(G_j)}(G_j) \right\rfloor.$$

The function α^* is multiplicative with respect to the strong product for all graphs (Scheinerman and Ullman 2011). Thus, from Observation 1 and (5) we get

$$\alpha_i \geq \left\lfloor \prod_{j \in [r]\setminus\{i\}} \alpha^*(G_j) \right\rfloor = \left\lfloor \alpha^*\left(\mathop{\boxtimes}_{j \in [r]\setminus\{i\}} G_j\right) \right\rfloor \geq \alpha\left(\mathop{\boxtimes}_{j \in [r]\setminus\{i\}} G_j\right) \tag{7}$$

and finally α_i holds the condition (6). Furthermore, from (7) and (3), for graphs without vertices with degree zero, also the following substitution

$$\alpha_i = \left\lfloor \prod_{j \in [r] \setminus \{i\}} \left(|E(G_j)| \varsigma^{-1}(E(G_j)) + R_{E(G_j)}(G) \right) \right\rfloor = \left\lfloor \prod_{j \in [r] \setminus \{i\}} \frac{|E(G_j)|}{\delta(G_j)} \right\rfloor$$

holds the condition (6).

Let $i \in \{1, 2, \ldots, r\}$. Algorithm 6.3 (Distribution DISTR), which takes as input a graph $G = G_i$ and an upper bound $\alpha_b = \alpha_i$, determines a distribution for a graph $G^{\boxtimes}$. The algorithm DISTR, whose running time is $O(n^2)$, uses Algorithm 6.2 (Greedy Coloring GC), which has complexity $O(n + m)$ (Kubale 2004). The algorithm GC takes as input a graph G and an arbitrary permutation P of the vertex set of G. GC in DISTR legally colors the complement of G and hence produces a partition $\mathcal{Q}$ of the vertex set of G into cliques (the so-called *clique cover* of G). Subsequently, DISTR distributes $\alpha_b = \alpha_i$ potential elements of an independent set roughly evenly (about $\alpha_b / |Q|$ elements or less depending on the sum from line 19) among all vertices of Q (as well as among all subgraphs of $G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_{i-1} \boxtimes G_i[Q] \boxtimes G_{i+1} \boxtimes \ldots \boxtimes G_r$) for all $Q \in \mathcal{Q}$.

Algorithm 6.2 Greedy Coloring GC

```
function GC(G, P)
comment: In all algorithms, loops contained the keyword in are performed in a given order.
    for each v in P do
        assign to v the smallest possible legal color C(v) in G
    return C
```

As we mentioned before, vertices chosen by MIN strongly block an eventual choice of vertices in a further stage of the algorithm. Our greedy algorithm, i.e., Algorithm 6.4 (Greedy Algorithm MIN-SP), significantly diminishes the mentioned effect by the use of generated distributions. The vertex set of $G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_r$ can be interpreted as the r-dimensional cuboid of the size $|V(G_1)| \cdot |V(G_2)| \cdot \ldots \cdot |V(G_r)|$. MIN-SP uses distributions in all r dimensions. Earlier Baumert et al. (1971), Jurkiewicz et al. (2015) and Codenotti et al. (2003), only one distribution was used at one time in algorithms for the maximum independent set problem in subclasses of the strong product of graphs to reduce a search space. The important point to note here is that in cases that are more interesting from an information theory point of view, i.e., if $G_1 = G_2 = \cdots = G_r$, some parts of MIN-SP are much simpler, e.g., we can determine one distribution and then we use it in all dimensions.

MIN-SP defines four sets N, V, F and I, where N is the closed neighborhood of a chosen vertex $\overline{v}^*$ (line 10), V is a set of vertices that are available for the next iterations, F is a set of forbidden vertices that are not available for the next iterations and I is an actual solution (an actual independent set). In lines 13–14 and lines 18–21, MIN-SP updates distributions and degrees of all vertices from V, respectively. In line

Algorithm 6.3 Distribution DISTR

```
function DISTR(G, α_b)
comment: V(G) = {v_1, ..., v_n}
  for each v ∈ V(G) do
     distr_v ← 0
  assign to P vertices of G arranged in non-increasing order according to their degrees
  C ← GC(Ḡ, P)
  create the clique cover CC of G from the coloring C
  sort cliques from CC in non-increasing order according to their sizes
  sort vertices in cliques from CC in non-decreasing order according to their degrees
  for each Q in CC do
     q ← ⌊α_b / |Q|⌋
     r ← α_b mod |Q|
     i ← 0
     for each v in Q do
        K ← q
        if i < r then
           K ← K + 1
        m ← α_b
        for each Q' in CC do
           M ← max { k ∈ {0, ..., K} : Σ_{v' ∈ N(v) ∩ Q'} distr_{v'} + k ≤ α_b }
           if m > M then
              m ← M
        distr_v ← m
        i ← i + 1
  return (distr_{v_1}, ..., distr_{v_n})
```

17, elements of N and F are removed from V, but only degrees of vertices from N are updated.

An advantage of MIN-SP is that we do not need to store edges of $G^{\boxtimes} = G_1 \boxtimes G_2 \boxtimes \ldots \boxtimes G_r$ in a computer memory. This is important since $|E(G^{\boxtimes})|$ almost always fast increases with r. In the memory, we only keep factors of $G^{\boxtimes}$, and the adjacency relation $\sim$ is directly checked from the conditions specified in the definition of the strong product (line 20).

Sometimes MIN-SP produces I such that $V(G^{\boxtimes}) - N_{G^{\boxtimes}}[I] \neq \emptyset$, where $N_G[I] = \bigcup_{v \in I} N_G[v]$ for $I \in V(G)$ and a graph G. Thus, finally, it is possible to get a larger independent set of $G^{\boxtimes}$, i.e., $I' = I \cup \text{MIN}(G^{\boxtimes} - N_{G^{\boxtimes}}[I])$. We prefer such a method in our computations. It turns out that we also do not need to store edges of $G^{\boxtimes}$ if we want to execute $\text{MIN}(G^{\boxtimes} - N_{G^{\boxtimes}}[I])$. It can be done by a modification of MIN similar to that we performed, when we constructed MIN-SP.

In Table 3, we summarize results produced by MIN-SP. The algorithm has a running time of $O(|V|^3)$.

We can approximate the Shannon capacity using (5) and the algorithm MIN-SP. We show it by the following example.

Algorithm 6.4 Greedy Algorithm MIN- SP

function MIN- SP$((G_1, G_2, \ldots, G_r))$
  comment: $\overline{v} = (v_1, v_2, \ldots, v_r)$, $\overline{v}^* = (v_1^*, v_2^*, \ldots, v_r^*)$
  $I \leftarrow \emptyset$
  for $i \leftarrow 1$ **to** r **do**
    $distr^{(i)} \leftarrow$ DISTR(G_i, α_i)
  $V \leftarrow V(G_1) \times V(G_2) \times \ldots \times V(G_r)$
  for each $\overline{v} \in V$ **do**
    $d(\overline{v}) \leftarrow |N_{G_1}[v_1]| \cdot |N_{G_2}[v_2]| \cdot \ldots \cdot |N_{G_r}[v_r]| - 1$
  while $V \neq \emptyset$ **do**
    assign to $\overline{v}^*$ an element $\overline{v} \in V$ with the smallest $d(\overline{v})$
    $N \leftarrow N_{G_1}[v_1^*] \times N_{G_2}[v_2^*] \times \ldots \times N_{G_r}[v_r^*]$
    $N \leftarrow N \cap V$
    $F \leftarrow \emptyset$
    for $i \leftarrow 1$ **to** r **do**
      $distr^{(i)}_{v_i^*} \leftarrow distr^{(i)}_{v_i^*} - 1$
      if $distr^{(i)}_{v_i^*} = 0$ **then**
        append to F elements $\overline{v} \in V$ with $v_i = v_i^*$
    $V \leftarrow V \setminus (N \cup F)$
    for each $\overline{v} \in V$ **do**
      for each $\overline{v}' \in N$ **do**
        if $\overline{v} \sim \overline{v}'$ **then**
          $d(\overline{v}) \leftarrow d(\overline{v}) - 1$
    $I \leftarrow I \cup \{\overline{v}^*\}$
  return I

Table 3 For each graph $G \in \mathcal{G}^+_{n,2} = \{H : \alpha(H^{\boxtimes 2}) > \alpha^2(H) \wedge |V(H)| = n\}$ we determined an independent set I' of the graph $G^{\boxtimes 2}$ using the algorithm MIN-SP

Greedy algorithm MIN- SP (results for $\mathcal{G}^+_{n,2}$)				
n	$\lvert\mathcal{G}^+_{n,2}\rvert$	$\lvert I'\rvert \le \alpha^2(G)$	$\lvert I'\rvert > \alpha^2(G)$	$\lvert I'\rvert = \alpha(G^{\boxtimes 2})$
5	1	0	1	1
6	4	0	4	4
7	36	4	32	32
8	513	127	386	386
9	16,015	6306	9709	9652
10	908,794	505,089	403,705	403,469

It is worth to note that the gap between $\alpha^2(G)$ and $\alpha(G^{\boxtimes 2})$ is small for the n that are small (Gyárfás et al. 2012)

Example 1 We consider strong products of some fullerenes,[7] since they are regular, symmetrical (Fowler and Manolopoulos 2007) and hence are not so easy for solvers and programs that calculate the independence number. Furthermore, fullerenes are often of type II. The algorithm MIN- SP produced the following upper bounds: $\alpha(F_{20} \boxtimes F_{20}) \geq 56$, $\alpha(F_{24} \boxtimes F_{24}) \geq 85$ and $\alpha(F_{28} \boxtimes F_{28}) \geq 123$, where symbols F_{20}, F_{24} and F_{28}

[7] A *fullerene* graph is the graph formed from the vertices and edges of a convex polyhedron, whose faces are all pentagons or hexagons and all vertices have degree equal to three.

mean 20-fullerene (dodecahedral graph), 24-fullerene and 28-fullerene ($\alpha(F_{20}) = 8$, $\alpha(F_{24}) = 9$ and $\alpha(F_{28}) = 11$(Fowler et al. 2007)), respectively. Therefore, from (5), the Shannon capacity $\Theta(F_{24}) \geq \sqrt[2]{85} = 9.21954.. > \alpha(F_{24}) = 9$ and $\Theta(F_{28}) \geq \sqrt[2]{123} = 11.09053.. > \alpha(F_{28}) = 11$, but $\Theta(F_{20}) \geq \alpha(F_{20}) = 8$ (we conjecture that $\alpha(F_{20} \boxtimes F_{20}) = 64$).

7 Another modification with additional conditions and relaxed distributions

In this section, we propose another modification of greedy algorithm MIN, which is similar to MIN-SP, but in addition, it uses distances between vertices of the strong product. We write down the new polynomial algorithm MIN-SP2 (Algorithm 7.1) in a more compact form to make additional elements more visible. The modification, besides the advantages of MIN-SP, has better accuracy (Table 4), but sometimes MIN-SP gives a larger independent set than MIN-SP2.

Algorithm 7.1 Greedy Algorithm MIN- SP2

1: **function** MIN- SP2($(G_1, G_2, \ldots, G_r)$)
  comment: $\overline{v} = (v_1, v_2, \ldots, v_r)$, $\overline{v}^* = (v_1^*, v_2^*, \ldots, v_r^*)$
2:   $I \leftarrow \emptyset$
3:   **for** $i \leftarrow 1$ **to** r **do**
4:     $distr^{(i)} \leftarrow$ RDISTR(G_i, α_i)
5:   $V \leftarrow V(G_1) \times V(G_2) \times \ldots \times V(G_r)$
6:   **for each** $\overline{v} \in V$ **do**
7:     $d(\overline{v}) \leftarrow |N_{G_1}[v_1]| \cdot |N_{G_2}[v_2]| \cdot \ldots \cdot |N_{G_r}[v_r]| - 1$
8:   **while** $V \neq \emptyset$ **do**
9:     assign to V^* the set of elements $\overline{v} \in V$ with the min. $d(\overline{v})$
10:     assign to V^* the set of elements $\overline{v} \in V^*$ with the max. $\sum_{i=1}^{r} distr_{v_i}^{(i)}$
11:     **for each** $\overline{v} \in V^*$ **do**
12:       $D(\overline{v}) = \infty$
13:       **for each** $\overline{v}' \in I$ **do**
14:         **if** $\overline{v}$ and $\overline{v}'$ differ on exactly one coordinate i **then**
15:           **if** $D(\overline{v}) > d_{G_i}(v_i, v_i')$ **then** $D(\overline{v}) = d_{G_i}(v_i, v_i')$
16:     assign to $\overline{v}^*$ an element $\overline{v} \in V^*$ with the largest $D(\overline{v})$
17:     $I \leftarrow I \cup \{\overline{v}^*\}$
18:     **for** $i \leftarrow 1$ **to** r **do**
19:       $distr_{v_i^*}^{(i)} \leftarrow distr_{v_i^*}^{(i)} - 1$
20:     $N \leftarrow N_{G_1}[v_1^*] \times N_{G_2}[v_2^*] \times \ldots \times N_{G_r}[v_r^*]$
21:     assign to V the set of elements $\overline{v} \in V \backslash N$ with $\Pi_{i=1}^{r} distr_{v_i}^{(i)} > 0$.
22:     update the degree $d(\overline{v})$ of each vertex $\overline{v} \in V$
23:   **return** I

In contrast to MIN-SP, MIN-SP2 determines a set V^* of all vertices of V with the smallest degree (line 9). The set V is defined exactly the same as in MIN-SP. In each iteration, MIN-SP2 takes vertices from V^* with the largest $\sum_{i=1}^{r} distr_{v_i}^{(i)}$ (i.e., it chooses vertices from „the least crowded region", line 10). We also realized that

Table 4 For each graph $G \in \mathcal{G}^{+}_{n,2} = \{H : \alpha(H^{\boxtimes 2}) > \alpha^2(H) \wedge |V(H)| = n\}$ we determined an independent set I' of the graph $G^{\boxtimes 2}$ using the algorithm MIN- SP2, which has better accuracy than MIN-SP. By accuracy of these algorithms for examined graphs, we mean the sum of all numbers in the last column of an appropriate table

Greedy algorithm MIN-SP2 (results for $\mathcal{G}^{+}_{n,2}$)

n	$\lvert\mathcal{G}^{+}_{n,2}\rvert$	$\lvert I'\rvert \le \alpha^2(G)$	$\lvert I'\rvert > \alpha^2(G)$	$\lvert I'\rvert = \alpha(G^{\boxtimes 2})$
5	1	0	1	1
6	4	0	4	4
7	36	2	34	34
8	513	112	401	401
9	16,015	5945	10,070	10,047
10	908,794	465,706	443,088	441,226

independent vertices in a copy of a factor of a strong product are often placed far apart. Hence, in lines 11–16, MIN-SP2 calculates $D(\overline{v})$ for each $\overline{v} \in V^*$ and selects one with the largest $D(\overline{v})$, where $D(\overline{v})$ measures the minimum distance from $\overline{v}$ to the set of vertices of I that differ on exactly one coordinate with $\overline{v}$. In contrast to MIN, MIN-SP2 gives the best results if we choose the first vertex $\overline{v}$ with the smallest $d(\overline{v})$. It is worth to note here that we always randomly relabel all input graphs at the beginning of our algorithm.

We observed an interesting phenomenon: When distributions used in MIN-SP2 (lines 3–4) are not so precise (are relaxed), i.e., if numbers in distributions are too large (too optimistic), then this algorithm often produces optimal results. For example, let C_n be a cycle on n vertices, the following distribution (2, 2, 2, 2, 2, 2, 2) is too optimistic for $C_7^{\boxtimes 2}$ since $\alpha(C_7^{\boxtimes 2}) = 10 < 7 \times 2 = 14$, but MIN-SP2 for this strong product and distribution always gives a maximum independent set.

Theorem 1 *Let* $distr^{(1)} = distr^{(2)} = (2, 2, 2, 2, 2, 2, 2)$. *Then*

$$|\text{MIN-SP2}((C_7, C_7))| = \alpha(C_7 \boxtimes C_7).$$

Proof Without loss of generality (wlog), we relabel vertices of C_7. Let $V(C_7) = \{0, 1, 2, 3, 4, 5, 6\}$ and

$$E(C_7) = \{\{0, 1\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 0\}\}.$$

At the beginning, the degree of each vertex is 8. From symmetry, we assume (wlog) that the first chosen vertex by MIN-SP2 is $(1, 1)$. Now, $(1, 3)$, $(3, 1)$, $(1, 6)$, $(6, 1)$ have the minimum degree equal to 5. Again, from symmetry, we assume (wlog) that $(1, 3)$ is the chosen vertex. Since $distr_1^{(1)} = 0$ and $distr_1^{(2)} = distr_3^{(2)} = 1$ it follows that $V^* = \{(3, 2), (6, 2)\}$ after line 10. From symmetry, we assume (wlog) that $\overline{v}^* = (3, 2)$. Now, $(3, 0)$, $(3, 4)$ have the minimum degree equal to 4. Again, from symmetry, we assume (wlog) that $(3, 4)$ is the chosen vertex. In the next iteration $distr_1^{(1)} = distr_3^{(1)} = 0$, hence $\overline{v}^* = (2, 6)$, which has the minimum degree equal to 4. Then (0, 5), (0, 6), (4, 0), (4, 6) are vertices from V with the minimum degree equal to 4, but (0, 6), (4, 6) have the less sum ($3 < 4$) from line 10. Let $\overline{v}^* = (0, 5)$ (resp. $\overline{v}^* = (4, 0)$). Then (5, 4), (5, 5) (resp. (6, 1), (6, 0)) are vertices from V with the minimum degree equal to 3.

We consider a new chosen vertex $\overline{v}^* = (5, 4)$ (resp. (6, 1)). Other two cases will be considered later. Then (4, 6), (5, 2), (6, 2) (resp. (0, 6), (5, 3), (6, 3)) have the minimum degree equal to 3, but (6, 2) and (4, 6) (resp. (5, 3) and (0, 6)) have the largest sum ($3 > 2$) from line 10 and $D((6, 2)) = 3$ (resp. $D((5, 3)) = 3$) is larger than $D((4, 6)) = D((5, 2)) = 2$ (resp. $D(((0, 6)) = D((6, 3)) = 2$). Thus $\overline{v}^* = (6, 2)$ (resp. (5, 3)). In the next iteration, Min-SP2 choose (6, 0) (resp. (5, 5)) since only this vertex has the minimum degree equal to 2 and then V contains the last two adjacent vertices: (4, 0), (4, 6) (resp. (0, 5), (0, 6)) with equal degrees. Hence, in these cases $|\text{Min-SP2}((C_7, C_7))| = 10 = \alpha(C_7 \boxtimes C_7)$.

We now consider a new chosen vertex $\overline{v}^* = (5, 5)$ (resp. (6, 0)). Then, in the next two iterations, Min-SP2 chooses (4, 0) and then (6, 0) (resp. (0, 5) and then (5, 5)) since these vertices have the minimum degree. In the last iteration, V contains four vertices: (5, 2), (5, 3), (6, 2), (6, 3) that form the clique on four vertices. Thus, also in these cases $|\text{Min-SP2}((C_7, C_7))| = 10 = \alpha(C_7 \boxtimes C_7)$. □

In our computational experiments, we got many optimal results produced by Min-SP2 with relaxed distributions and we did not find any counterexamples for $|\text{Min-SP2}((C_{2k+1}, C_{2k+1}))| = \alpha(C_{2k+1} \boxtimes C_{2k+1})$ with $distr^{(1)} = distr^{(2)} = (\lceil k/2\rceil, \lceil k/2\rceil, \dots, \lceil k/2\rceil)$, where $\lceil k/2\rceil$ occurs $(2k+1)$–times and $k \geq 2$.

According to the above considerations, we propose a relaxed version of the function Distr, which we call Relaxed Distribution RDistr (Algorithm 7.2).

Algorithm 7.2 Relaxed Distribution RDistr

```
function RDistr(G, α_b)
comment: V(G) = {v_1, ..., v_n}
    for each v ∈ V(G) do
        distr_v ← ∞
    for each u ∈ V(G) do
        Q ← {u} ∪ Min(Ḡ − N_Ḡ[u])
        for each v ∈ Q do
            if distr_v > ⌈α_b/|Q|⌉ then
                distr_v ← ⌈α_b/|Q|⌉
    return (distr_{v_1}, ..., distr_{v_n})
```

8 Community detection problems

Chalupa and Pospíchal (2014) investigated the growth of large independent sets in the Barabási–Albert model of scale-free complex networks. They formulated recurrent relations describing the cardinality of typical large independent sets and showed that this cardinality seems to scale linearly with network size. Independent sets in social networks represent groups of people, who do not know anybody else within the group.

Hence, an independent set of a network plays a crucial role in community detection problems, since vertices of this set are naturally unlikely to belong to the same community (Chalupa and Pospíchal 2014; Whang et al. 2013). These facts imply that the number of communities in scale-free networks seems to be bounded from below by a linear function of network size (Chalupa and Pospíchal 2014).

Leskovec et al. (2010) introduced the Kronecker graph network model that naturally obeys common real network properties. In particular, the model assumes that graphs have loops and corresponds to the strong product (Hammack et al. 2011). Let $i \geq 1$ and $G' = G^{\boxtimes i}$. As mentioned earlier, the function α is supermultiplicative and α^* is multiplicative with respect to the strong product for all graphs. Thus $(\alpha^*(G))^i = \alpha^*(G') \geq \alpha(G') \geq (\alpha(G))^i$ and hence $|V(G')|^{c'} \geq \alpha(G') \geq |V(G')|^c$, where[8] $c' = \log(\alpha^*(G))/\log(V(G))$ and $c = \log(\alpha(G))/\log(V(G))$. We have just showed that the cardinality of maximum independent sets, in the mentioned model, scale sublinearly with network[9] size. Furthermore, if G is of type I, then $\alpha(G') = |V(G')|^c$. These considerations show that the number of communities in scale-free networks seems to be bounded from below by a sublinear (rather than a linear) function of network size. It is worth pointing out that we can approximate (resp. predict) the number of communities, in the mentioned model (resp. real complex network), using Algorithms 6.4 or 7.1 (Greedy Algorithm Min-SP, Min-SP2).

References

Adhikari MR, Adhikari A (2014) Basic modern algebra with applications. Springer, New Delhi. https://doi.org/10.1007/978-81-322-1599-8

Alon N, Lubetzky E (2006) The Shannon capacity of a graph and the independence numbers of its powers. IEEE Trans Inf Theory 52(5):2172–2176. https://doi.org/10.1109/TIT.2006.872856

Barrus MD (2012) Havel–Hakimi residues of unigraphs. Inf Process Lett 112(1–2):44–48. https://doi.org/10.1016/j.ipl.2011.10.011

Baumert LD, McEliece RJ, Rodemich E, Rumsey HC Jr, Stanley R, Taylor H (1971) A combinatorial packing problem. In: Computers in algebra and number theory (Proc SIAM-AMS Sympos Appl Math, New York, 1970). American Mathematical Society, Providence, RI, pp 97–108. SIAM–AMS Proc., Vol. IV

Boche H, Deppe C (2020) Computability of the zero-error capacity of noisy channels. arXiv: 2010.06873

Borowiecki P, Rautenbach D (2015) New potential functions for greedy independence and coloring. Discrete Appl Math 182:61–72. https://doi.org/10.1016/j.dam.2013.12.011

[8] We can use the so-called Lovász theta function instead of α^*, since it is also multiplicative with respect to the strong product for all graphs (Lovász 1979).

[9] We assume that a network (graph) G is non-empty, i.e., $|E(G)| \neq 0$.

Chalupa D, Pospíchal J (2014) On the growth of large independent sets in scale-free networks. In: Nostradamus 2014: prediction, modeling and analysis of complex systems. Springer, pp 251–260
Codenotti B, Gerace I, Resta G (2003) Some remarks on the Shannon capacity of odd cycles. Ars Combin 66:243–257
Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge
Cover TM, Thomas JA (2006) Elements of information theory, vol 2. Wiley, Hoboken
Csiszár I, Körner J (2011) Information theory, coding theorems for discrete memoryless systems, 2nd edn. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511921889
Favaron O, Mahéo M, Saclé JF (1991) On the residue of a graph. J Graph Theory 15(1):39–64. https://doi.org/10.1002/jgt.3190150107
Favaron O, Mahéo M, Saclé JF (1993) Some eigenvalue properties in graphs (conjectures of Graffiti. II). Discrete Math 111(1-3):197–220, graph theory and combinatorics (Marseille-Luminy, 1990). https://doi.org/10.1016/0012-365X(93)90156-N
Fomin FV, Kratsch D (2010) Exact exponential algorithms. Texts in theoretical computer science. An EATCS Series, Springer, Heidelberg. https://doi.org/10.1007/978-3-642-16533-7
Fowler PW, Manolopoulos D (2007) An atlas of fullerenes. Courier Corporation, North Chelmsford
Fowler P, Daugherty S, Myrvold W (2007) Independence number and fullerene stability. Chem Phys Lett 448(1):75–82. https://doi.org/10.1016/j.cplett.2007.09.054
Gross JL, Yellen J, Zhang P (eds) (2014) Handbook of graph theory, 2nd edn. Discrete mathematics and its applications (Boca Raton). CRC Press, Boca Raton
Gyárfás A, Sebő A, Trotignon N (2012) The chromatic gap and its extremes. J Comb Theory Ser B 102(5):1155–1178. https://doi.org/10.1016/j.jctb.2012.06.001
Hales RS (1973) Numerical invariants and the strong product of graphs. J Comb Theory Ser B 15:146–155
Hammack R, Imrich W, Klavžar S (2011) Handbook of product graphs, 2nd edn. Discrete mathematics and its applications (Boca Raton). CRC Press, Boca Raton, FL, with a foreword by Peter Winkler
Harant J, Schiermeyer I (2001) On the independence number of a graph in terms of order and size. Discrete Math 232(1–3):131–138. https://doi.org/10.1016/S0012-365X(00)00298-3
Jurkiewicz M (2017) Average distance is submultiplicative and subadditive with respect to the strong product of graphs. Appl Math Comput 315:278–285. https://doi.org/10.1016/j.amc.2017.06.025
Jurkiewicz M (2020) An approximation of the zero error capacity by a greedy algorithm. In: Wu W, Zhang Z (eds) Combinatorial optimization and applications. Springer, Cham, pp 91–104
Jurkiewicz M, Pikies T (2015) Selected topics in modern mathematics, Publishing House AKAPIT, Kraków, inst. Politechnika Krakowska, chap Some classical lower bounds on the independence number and their behavior on the strong product of graphs
Jurkiewicz M, Kubale M, Ocetkiewicz K (2015) On the independence number of some strong products of cycle-powers. Found Comput Decis Sci 40(2):133–141. https://doi.org/10.1515/fcds-2015-0009
Körner J, Orlitsky A (1998) Zero-error information theory. IEEE Trans Inf Theory 44(6):2207–2229. https://doi.org/10.1109/18.720537
Kubale M (2004) Graph colorings, contemporary mathematics, vol 352. American Mathematical Society, Providence. https://doi.org/10.1090/conm/352
Leskovec J, Chakrabarti D, Kleinberg J, Faloutsos C, Ghahramani Z (2010) Kronecker graphs: an approach to modeling networks. J Mach Learn Res 11:985–1042
Loehr NA (2011) Bijective combinatorics. Discrete mathematics and its applications (Boca Raton). CRC Press, Boca Raton
Lovász L (1979) On the Shannon capacity of a graph. IEEE Trans Inf Theory 25(1):1–7. https://doi.org/10.1109/TIT.1979.1055985
Mathew KA, Östergård PRJ (2017) New lower bounds for the Shannon capacity of odd cycles. Des Codes Cryptogr 84(1–2):13–22. https://doi.org/10.1007/s10623-016-0194-7
McEliece RJ (2004) The theory of information and coding, encyclopedia of mathematics and its applications, with a foreword by Mark Kac, vol 86. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511819896
Polak SC, Schrijver A (2019) New lower bound on the Shannon capacity of C_7 from circular graphs. Inf Process Lett 143:37–40. https://doi.org/10.1016/j.ipl.2018.11.006
Rosenfeld M (1967) On a problem of C. E. Shannon in graph theory. Proc Am Math Soc 18:315–319. https://doi.org/10.2307/2035288

Scheinerman ER, Ullman DH (2011) Fractional graph theory. Dover Publications, Inc., Mineola, NY, a rational approach to the theory of graphs, With a foreword by Claude Berge, Reprint of the 1997 original

Shannon CE (1956) The zero error capacity of a noisy channel. Trans Inf Theory IT 2(September):8–19

Whang JJ, Gleich DF, Dhillon IS (2013) Overlapping community detection using seed set expansion. In: Proceedings of the 22nd ACM international conference on information and knowledge management, association for computing machinery, New York, NY, USA, CIKM '13, p 2099–2108. https://doi.org/10.1145/2505515.2505535

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.