Reduction and Fixed Points of Boolean Networks and Linear Network Coding Solvability

Maximilien Gadouleau, *Member, IEEE*, Adrien Richard, and Eric Fanchon

Abstract

Linear network coding transmits data through networks by letting the intermediate nodes combine the messages they receive and forward the combinations towards their destinations. The solvability problem asks whether the demands of all the destinations can be simultaneously satisfied by using linear network coding. The guessing number approach converts this problem to determining the number of fixed points of coding functions $f : A^n \to A^n$ over a finite alphabet A (usually referred to as Boolean networks if $A = \{0, 1\}$) with a given interaction graph, that describes which local functions depend on which variables. In this paper, we generalise the so-called reduction of coding functions in order to eliminate variables. We then determine the maximum number of fixed points of a fully reduced coding function, whose interaction graph has a loop on every vertex. Since the reduction preserves the number of fixed points, we then apply these ideas and results to obtain four main results on the linear network coding solvability problem. First, we prove that non-decreasing coding functions cannot solve any more instances than routing already does. Second, we show that triangle-free undirected graphs are linearly solvable if and only if they are solvable by routing. This is the first classification result for the linear network coding solvability problem. Third, we exhibit a new class of non-linearly solvable graphs. Fourth, we determine large classes of strictly linearly solvable graphs.

M. Gadouleau is with School of Engineering and Computing Sciences, Durham University, UK. m.r.gadouleau@durham.ac.uk

A. Richard is with Laboratoire I3S, CNRS & Université de Nice-Sophia Antipolis, France. richard@unice.fr

E. Fanchon is with Université de Grenoble - CNRS, TIMC-IMAG UMR 5525, Grenoble, France. eric.fanchon@imag.fr

This work is partially supported by CNRS and The Royal Society through the International Exchanges Scheme grant *Boolean networks, network coding and memoryless computation.*

I. Introduction

A. Background: network coding solvability and coding functions

Network coding is a technique to transmit information through networks, which can significantly improve upon routing in theory [1], [2]. At each intermediate node v, the received messages $x_{u_1}, \dots, x_{u_k}$ are combined, and the combined message $f_v(x_{u_1}, \dots, x_{u_k})$ is then forwarded towards its destinations. The main problem is to determine which functions f_v can transmit the most information. In particular, the **network coding solvability problem** tries to determine whether a certain network situation, with a given set of sources, destinations, and messages, is solvable, i.e. whether all messages can be transmitted to their destinations. This problem being very difficult, different techniques have been used to tackle it, including matroids [3], Shannon and non-Shannon inequalities for the entropy function [4], [5], error-correcting codes [6], and closure operators [7], [8]. As shown in [5], [9], the solvability problem can be recast in terms of fixed points of (non-necessarily Boolean) networks.

Boolean networks have been used to represent a network of interacting entities as follows. A network of n automata has a state $x = (x_1, \dots, x_n) \in \{0,1\}^n$, represented by a Boolean variable x_i on each automaton i, which evolves according to a deterministic function $f = (f_1, \dots, f_n) : \{0,1\}^n \to \{0,1\}^n$, where $f_i : \{0,1\}^n \to \{0,1\}$ represents the update of the local state x_i. Boolean networks have been used to model gene networks [10], [11], [12], [13], neural networks [14], [15], [16], social interactions [17], [18] and more (see [19], [20]). Their natural generalisation where each variable x_i can take more than two values in some finite alphabet A has been investigated since this can be a more accurate representation of the phenomenon we are modelling [12], [21]. In order to avoid confusion, and despite the popularity of the term "Boolean network," we shall refer to any function $f : A^n \to A^n$ as a **coding function**.

The structure of a coding function $f : A^n \to A^n$ can be represented via its **interaction graph** $G(f)$, which indicates which update functions depend on which variables. More formally, $G(f)$ has $\{1, \dots, n\}$ as vertex set and there is an arc from j to i if $f_i(x)$ depends essentially on x_j. In different contexts, the interaction graph is known–or at least well approximated–, while the actual update functions are not. One main problem of research on (non-necessarily Boolean) coding functions is then to predict their dynamics according to their interaction graphs.

Among the many dynamical properties that can be studied, **fixed points** are crucial because they represent stable states; for instance, in the context of gene networks, they correspond to stable patterns of gene expression at the basis of particular biological processes. As such, they are arguably the property which has been the most thoroughly studied. The study of the number of fixed points and its maximisation in particular is the subject of a stream of work, e.g. in [22], [23], [24], [25], [26], [27], [28]. In particular, a lot of literature is devoted to determining when a Boolean coding function admits multiple fixed points (see [29] for a survey).

The network coding solvability problem can be recast in terms of fixed points of coding functions as follows [5], [9]. The so-called **guessing number** [5] of a digraph G is the logarithm of the maximum number of fixed points over all coding functions f whose interaction graph is a subgraph of G: $G(f) \subseteq G$. The guessing number is always upper bounded by the size of a minimum feedback vertex set of G; if equality holds, we say that G is **solvable** and the coding function f reaching this bound is called a solution. Then, a network coding instance N is solvable if and only if some digraph G_N (to be defined later) related to the instance N is solvable.

Linear network coding is the most popular kind of network coding, where the intermediate nodes can only perform linear combinations of the packets they receive [30]. The network coding instance N is then linearly solvable if and only if G_N admits a linear solution. Many interesting classes of linearly solvable digraphs have been given in the literature (see [31], [6]). However, as we shall explain in Section IV, all the linearly solvable undirected graphs G known so far are “easily” solved, because they are all vertex-full: the vertex set can be partitioned into $\alpha(G)$ cliques, where $\alpha(G)$ is the independence number of G [32].

Network coding solvability is closely related to **index coding** [33], [34]. In particular, the length of a minimal index code (for a given digraph) is the same as the information defect [5], [6]. Since a graph is solvable if and only if it is solvable in the sense of information defect [5], [7], there is an equivalence between index coding and network coding. This equivalence was independently given in [35] and extended to storage capacity in [36]; an asymptotic version of this equivalence is also given in [6].

The guessing number, and hence the maximum number of fixed points of Boolean networks, is also equal to the so-called **graph entropy** [31]. The latter is the maximum entropy of a family of discrete random variables (of entropy at most one each), where the dependencies amongst variables are described by a digraph. Hence, using properties of the entropy function, one can give upper bounds on the number of fixed points of Boolean networks [31], [32].

B. Our approach and contribution

Fixed points of coding functions and network coding are very closely linked; for instance [28] uses techniques from network coding and coding theory to derive bounds on the number of fixed points of specific coding functions. As such, in this paper we will derive **results of interest for both communities**.

More precisely, we expand a new technique to study the number of fixed points of coding functions and we apply it to the solvability problem. Recently, [37] introduced the reduction of coding functions in order to reduce the number of interacting automata while preserving some key dynamical properties. More precisely, for any loopless vertex v of $G(f)$ the v-reduction of f is obtained by evaluating f_v and then replacing its expression instead of x_v into all the other local functions f_i. The v-reduction notably preserves the number of fixed points [37]. A very similar reduction procedure was proposed in the context of systems of differential equations [38]; this procedure is also based on variable elimination and preserves the number of fixed points.

In this paper, we generalise the concept of reduction of a coding function by a vertex in two fashions. We consider successive reductions vertex per vertex, and we prove in Theorem 1 that this is equivalent to reducing all these vertices at once, provided that they induce an acyclic subgraph of the interaction graph. Since the reduction of a coding function has the same number of fixed points as the original coding function, we can then study the number of fixed points of fully reduced coding functions. We also introduce the concept of reduction of digraphs; again this can be done one vertex at a time or all at once, according to Theorem 2. The interaction graph of a reduced coding function is then a subgraph of the reduction of its interaction graph. In particular, reducing an entire maximal acyclic set of a digraph yields a digraph with a loop on each vertex. Similarly, we can always successively reduce a coding function to one whose interaction graph has a loop on each vertex. We then fully determine the maximum number of fixed points of coding functions for a given interaction graph with a loop on each vertex in Theorem 3.

We then apply this reduction approach to network coding solvability and derive four main results.

1) We consider solvability by non-decreasing coding functions, which naturally extend routing. We show in Theorem 4 that a digraph is solvable by a non-decreasing coding function if and only if it is solvable by routing.
2) We derive some important classification results for undirected graphs. We exhibit in Theorem 5 the first example of a non-vertex-full linearly solvable graph. We obtain in Theorem 6 a necessary condition for a graph G to be strictly linearly solvable, i.e. to have a linear solution f with $G(f) = G$. Using this condition, we then prove in Theorem 7 that a triangle-free undirected graph is linearly solvable if and only if it is vertex-full; we also prove that all strictly linearly solvable complements of triangle-free graphs are vertex-full in Theorem 8. For triangle-free graphs, our results indicate that the instance is linearly solvable if and only if it is solvable by routing; in other words, linear network coding does not help to solve these graphs.
3) Using Theorem 6, we exhibit in Theorem 9 a new class of digraphs which are not linearly solvable.

This is significant because few non-linearly solvable classes of digraphs are known so far, and proving non-linear solvability usually requires different techniques, such as graph entropy [31], [32] or digraph closure [8].

4) We show in Theorem 10 that a large class of digraphs are strictly linearly solvable. Strictly linearly solvable digraphs are not only interesting for some applications of coding functions (see Section IV-E), but they also represent network coding instances where no arc is detrimental to the transmission of information.

The rest of the paper is organised as follows. Section II studies the reduction of coding functions and the reduction of graphs and relates these two notions. Reductions of coding functions are then related to their fixed points in Section III. Finally, we apply the theory of coding function and graph reductions to the problem of linear network coding solvability in Section IV.

II. Reduction of coding functions

A. *Definitions*

We first review some concepts relating to coding functions. Let V be a finite set, possibly empty, of cardinality n. Let A be a finite set, referred to as the **alphabet**, of cardinality $q \geq 2$; depending on the context, we will consider $A = \mathrm{GF}(q)$ or $A = \mathbb{Z}_q$ or $A = [q] := \{0, \dots, q-1\}$. Let $f : A^V \to A^V$ be a **coding function** of dimension $\textsc{dim}(f) = n$. We shall usually simplify notation and identify A^V with A^n. We can then view $f : A^n \to A^n$ as $f = (f_1, \dots, f_n)$ where $f_v : A^n \to A$. For any $x \in A^n$ and any $I \subseteq V$, we also denote $x_{V \setminus I}$ as x_{-I}; we will usually identify a vertex v with its corresponding singleton $\{v\}$.

A digraph with vertex set V is a pair $G = (V, E)$ where $E \subseteq V^2$; we set $\textsc{dim}(G) = |V| = n$. If E is a symmetric set, we say that G is undirected (i.e. we identify undirected and bidirected graphs). We associate with f the digraph $G(f)$, referred to as the **interaction graph** of f, defined by: the vertex set is V; and for all $u, v \in V$, there exists an arc (u, v) if and only if f_v depends essentially on x_u, i.e. there exist $x, y \in A^n$ that only differ by $x_u \neq y_u$ such that $f_v(x) \neq f_v(y)$. We denote the set of all coding functions $f : A^n \to A^n$ for some A of size q with interaction graph G as $F(G, q)$.

We now review some basic concepts and introduce some notation for digraphs $G = (V, E)$ [39]. An **induced subgraph** of G is obtained by removing vertices of G, i.e. $G' = (V', E')$ is and induced subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' = \{(u, v) \in E : u, v \in V'\}$. A **spanning subgraph** of G is obtained by removing arcs, i.e. $G' = (V', E')$ is and induced subgraph of $G = (V, E)$ if $V' = V$ and $E' \subseteq E$. If $I \subseteq V$, we denote by $G[I]$ the subgraph of G induced by I, and we set $G \setminus I = G[V \setminus I]$. If

$G[I]$ has no cycle, then we say that I is an **acyclic set**. An acyclic set $I = \{i_1, \dots, i_m\}$ can be sorted in **topological order**, where $(i_k, i_l) \in G$ only if $k < l$. Thus if I is an acyclic set of $G(f)$, then f_{i_k} does not depend on the variables i_l with $l > k$ and we can write $f_{i_k}(x) = f_{i_k}(x_{-I}, x_{i_1}, \dots, x_{i_{k-1}})$. The complement of an acyclic set is a **feedback vertex set**, i.e. $J \subseteq V$ is a feedback vertex set of G if and only if $V \setminus J$ is an acyclic set of G. We denote the size of a minimum feedback vertex set of G as $k(G)$ and the size of a maximum acyclic set of G as $\alpha(G)$; we then have $\alpha(G) = n - k(G)$.

The **in-neighbourhood** of a vertex i in G is denoted as $\mathrm{in}_G(i) := \{u \in V : (u,i) \in G\}$; its **in-degree** is $\mathrm{ind}_G(i) = |\mathrm{in}_G(i)|$; when there is no ambiguity, we shall remove the dependence in G. The **out-neighbourhood** and **out-degree** are defined similarly. Paths and cycles are always supposed to be directed. If $s = (s_1, \dots, s_k)$ is a sequence of distinct vertices of G, then $\{s\} = \{s_1, \dots, s_k\}$ denotes the **support** of s.

Definition 1 ([37]). For any $v \in V$ without a loop in $G(f)$, the v**-reduction of** f is the coding function $f^{-v} : A^{V \setminus v} \to A^{V \setminus v}$, where for all $i \neq v$ and $x \in A^V$

$$f_i^{-v}(x_{-v}) := f_i(x_{-v}, f_v(x_{-v})).$$

If $G(f)$ has a loop on v then $f^{-v} = f$ by convention.

The v-reduction of f is then obtained by first applying the local function f_v to update x_v to $f_v(x)$ and by then applying all the other local functions f_i, $i \neq v$.

Thus $\textsc{dim}(f^{-v}) = \textsc{dim}(f) - 1$ if and only if $G(f)$ has no loop on v. Let $s = (s_1, s_2, \dots, s_k)$ be a sequence of distinct vertices of V of length $|s| = k > 0$. We write

$$f^{-s} = f^{-s_1 s_2 \dots s_k} = (f^{-s_1})^{-s_2})^{\dots})^{-s_k}.$$

The sequence s is a **reduction sequence of *f*** if: $G(f)$ has no loop on s_1, and $G(f^{-s_1 \dots s_{r-1}})$ has no loop on s_r for each $1 < r \leq k$. So s is a reduction sequence if and only if

$$\textsc{dim}(f^{-s}) = \textsc{dim}(f) - |s|.$$

By convention the empty sequence ϵ is a reduction sequence, and $f^{-\epsilon} = f$.

Definition 2. Let $I = \{i_1, \dots, i_m\}$ be an acyclic set of $G(f)$ in topological order. We denote the

cumulative f-coding function on I as $F^I : A^{V\setminus I} \to A^I$ defined as

$$\begin{aligned} F^I_{i_1}(x_{-I}) &:= f_{i_1}(x_{-I}) \\ F^I_{i_2}(x_{-I}) &:= f_{i_2}(x_{-I}, F^I_{i_1}(x_{-I})) \\ &\vdots \\ F^I_{i_m}(x_{-I}) &:= f_{i_m}(x_{-I}, F^I_{I\setminus i_m}(x_{-I})). \end{aligned}$$

The I-**reduction of** $f : A^V \to A^V$ is defined as the coding function $f^{-I} : A^{V\setminus I} \to A^{V\setminus I}$ such that

$$f^{-I}_i(x_{-I}) := f_i(x_{-I}, F^I(x_{-I})).$$

Theorem 1. *If I is an acyclic set of $G(f)$, then any enumeration s of I is a reduction sequence of f such that $f^{-s} = f^{-I}$.*

Proof. We prove that if there is no arc from v to u and no loop on either vertex, then $f^{-uv} = f^{-vu}$. By direct application of the reduction rule, we have for all $i \notin \{u, v\}$,

$$\begin{aligned} f^{-uv}_i(x_{-uv}) &= f^{-u}_i(x_{-uv}, f^{-u}_v(x_{-uv})) \\ &= f_i(x_{-uv}, f^{-u}_v(x_{-uv}), f_u(x_{-uv}, f^{-u}_v(x_{-uv}))) \\ &= f_i(x_{-uv}, f_v(x_{-uv}, f_u(x_{-uv})), f_u(x_{-uv}, f^{-u}_v(x_{-uv}))) \end{aligned}$$

and since there is no arc from v to u we get

$$f^{-uv}_i(x_{-uv}) = f_i(x_{-uv}, f_v(x_{-uv}, f_u(x_{-uv})), f_u(x_{-uv})).$$

Again by direct application of the reduction rule, we have for all $i \notin \{u, v\}$,

$$\begin{aligned} f^{-vu}_i(x_{-uv}) &= f^{-v}_i(x_{-uv}, f^{-v}_u(x_{-uv})) \\ &= f_i(x_{-uv}, f^{-v}_u(x_{-uv}), f_v(x_{-uv}, f^{-v}_u(x_{-uv}))) \end{aligned}$$

and since there is no arc from v to u we have $f^{-v}_u(x_{-uv}) = f_u(x_{-uv})$ thus

$$f^{-vu}_i(x_{-uv}) = f_i(x_{-uv}, f_u(x_{-uv}), f_v(x_{-uv}, f_u(x_{-uv}))).$$

Thus $f^{-uv}_i = f^{-vu}_i$ and the claim is proved.

Let $I = \{i_1, \dots, i_m\}$ in topological order and let s and t be enumerations of I. Firstly, suppose that s and t only differ by a transposition of adjacent vertices, say $s = (s_1, \dots, s_m)$ and $t = (s_1, \dots, s_{k-2}, s_k, s_{k-1}, s_{k+1}, \dots, s_m)$. We then have

$$f^{-s_1 \dots s_k} = h^{-s_{k-1}s_k} = h^{-s_k s_{k-1}} = f^{-t_1 \dots t_k},$$

where $h = f^{-s_1 \cdots s_{k-2}}$, and hence $f^{-s} = f^{-t}$. Secondly, in the general case, it is well known that t can be obtained from s by transposing adjacent vertices: indeed the Coxeter generators of I generate the symmetric group on I. Thus $f^{-s} = f^{-t}$; in particular, if s is a topological order of I, then we obtain f^{-I} described above. □

Corollary 1. *If s and t are two reduction sequences of f with the same acyclic support then $f^{-s} = f^{-t}$.*

A coding function h is a **reduced form of f** if there exists a reduction sequence s such that $f^{-s} = h$. A **minimal reduced form of f** is a reduced form h such that every vertex of $G(h)$ has a loop. The set of reduced forms of f is denoted $\textsc{red}(f)$. We are particularly interested in finding, according to $G(f)$, reduced forms of dimension as small as possible. In the ideal case, we would like to obtain reduced forms of dimension

$$\textsc{mindim}(f) := \min_{h \in \textsc{red}(f)} \textsc{dim}(h).$$

We finish this section by an example illustrating the reduction of coding functions.

Example 1. Consider the following coding function $f : \{0,1\}^4 \to \{0,1\}^4$ given by

$$f_1(x) = x_3 \wedge (x_2 \vee x_4)$$
$$f_2(x) = x_1 \vee x_4$$
$$f_3(x) = x_2$$
$$f_4(x) = x_3.$$

Then f^{-4} is given by

$$f_1^{-4}(x_{-4}) = x_3 \wedge (x_2 \vee x_3) = x_3$$
$$f_2^{-4}(x_{-4}) = x_1 \vee x_3$$
$$f_3^{-4}(x_{-4}) = x_2,$$

$f^{-34} = f^{-43}$ is given by

$$f_1^{-34}(x_1, x_2) = x_2$$
$$f_2^{-34}(x_1, x_2) = x_1 \vee x_2,$$

and $f^{-134} = f^{-431}$ is given by

$$f_2^{-134}(x_2) = x_2 \vee x_2 = x_2.$$

B. Graph reduction

Definition 3. If G has no loop on v, we call **v-reduction of G**, and we denote by G^{-v}, the graph obtained from $G \setminus v$ by adding an arc (u, w) (not already present) whenever (u, v) and (v, w) are arcs of G. By convention, if G has a loop on v, then $G^{-v} = G$.

We shall use similar notation to that of the reduction of coding functions. A sequence $s = (s_1, \dots, s_k)$ of vertices of G is a **reduction sequence of G** if: G has no loop on s_1, and $G^{-s_1 \dots s_{r-1}}$ has no loop on s_r for each $1 < r \leq k$. So s is a reduction sequence if and only if G^{-s} has $|V| - k$ vertices.

Definition 4. For any acyclic set I of G, the **I-reduction of** G is the digraph $G^{-I} := (V \setminus I, E')$, where $(u, w) \in E'$ if and only if either $(u, w) \in E$ or there is a path in G from u to w through I (that is, a path from u to w whose internal vertices are all in I).

Theorem 2. *If I is an acyclic set of G, then any enumeration s of I is a reduction sequence of G such that $G^{-s} = G^{-I}$.*

Proof. The structure of the proof is similar to that of Theorem 1. We first prove that if $u, v \in V$ induce an acyclic subgraph, then $G^{-uv} = G^{-vu}$. Say that there is no arc from v to u and that there is no loop on either u or v. Let us simplify notation and denote the proposition $(x, y) \in G$ as xy and the proposition $(x, y) \in G^{-z}$ as xy^{-z} for any vertices x, y, and z. Then for any $a, b \notin \{u, v\}$,

$$\begin{aligned}
ab^{-u} &\iff ab \vee (au \wedge ub)\,, \\
(a,b) \in G^{-uv} &\iff ab^{-u} \vee \left(av^{-u} \wedge vb^{-u}\right) \\
&\iff ab \vee (au \wedge ub) \vee \{[av \vee (au \wedge uv)] \wedge vb\} \\
&\iff ab \vee (au \wedge ub) \vee (av \wedge vb) \vee (au \wedge uv \wedge vb)\,.
\end{aligned}$$

Similarly,

$$\begin{aligned}
ab^{-v} &\iff ab \vee (av \wedge vb)\,, \\
(a,b) \in G^{-vu} &\iff ab^{-v} \vee \left(au^{-v} \wedge ub^{-v}\right) \\
&\iff ab \vee (av \wedge vb) \vee \{au \wedge [ub \vee (uv \wedge vb)]\} \\
&\iff ab \vee (av \wedge vb) \vee (au \wedge ub) \vee (au \wedge uv \wedge vb)\,.
\end{aligned}$$

Now let $I = \{i_1, \dots, i_m\}$ in topological order and let s and t be enumerations of I. Firstly, suppose that s and t only differ by a transposition of adjacent vertices, say $s = (s_1, \dots, s_m)$ and $t = (s_1, \dots, s_{k-2},$

$s_k, s_{k-1}, s_{k+1}, \dots, s_m)$. We then have

$$G^{-s_1 \dots s_k} = H^{-s_{k-1}s_k} = H^{-s_k s_{k-1}} = G^{-t_1 \dots t_k},$$

where $H = G^{-s_1 \dots s_{k-2}}$, and hence $G^{-s} = G^{-t}$. Secondly, in the general case, it is well known that t can be obtained from s by transposing adjacent vertices: indeed the Coxeter generators of I generate the symmetric group on I; thus $G^{-s} = G^{-t}$.

In particular, we prove that if s is the topological order, then we obtain G^{-I} described above. The proof is by induction on $|I|$. For $|I| = 1$, the result is obvious. Suppose the result holds for all induced acyclic subgraphs of size $m-1$. Let $s = i_1, \dots, i_m$ be a topological order of I. By definition, we have $(u,v) \in G^{-s}$ if and only if $(u,v) \in G^{-i_1,\dots,i_{m-1}}$ or $(u,i_m),(i_m,v) \in G^{-i_1,\dots,i_{m-1}}$. Thus, by induction hypothesis, $(u,v) \in G^{-s}$ if and only if $(u,v) \in G^{-(I \setminus i_m)}$ or $(u,i_m),(i_m,v) \in G^{-(I \setminus i_m)}$. This is equivalent to

- either, $(u,v) \in G$;
- or G has a path from u to v through $I \setminus i_m$;
- or $(u,i_m),(i_m,v) \in G$;
- or G has a path from u to i_m through $I \setminus i_m$ and $(i_m,v) \in G$;
- or $(u,i_m) \in G$ and G has a path from i_m to v through $I \setminus i_m$ (impossible, since $\mathrm{out}(i_m) \cap I = \emptyset$);
- or G has a path from u to i_m through $I \setminus i_m$ and a path from i_m to v through $I \setminus i_m$ (also impossible).

This is clearly equivalent to either $(u,v) \in G$ or there is a path in G from u to v through I. Thus $(u,v) \in G^{-s}$ if and only if $(u,v) \in G^{-I}$. □

We make three remarks on the reduction of digraphs.

1) If s and t are two reduction sequences of G with the same acyclic support then $G^{-s} = G^{-t}$.
2) G^{-I} has a loop on each vertex if and only if I is a maximal acyclic set. Therefore, there is a bijection between the set of minimal reduced forms of G and the set of its minimal feedback vertex sets. Since it is well known that finding a minimum feedback vertex set is an NP-Complete problem, finding a minimum reduced form is also NP-Complete.
3) For any G and any acyclic set I, we have $G \setminus I \subseteq G^{-I}$. We prove below a converse to this result: depending on the initial digraph G, the reduced form G^{-I} of G may add any possible arc to $G \setminus I$.

Proposition 1. *For any digraph D with vertex set J and any spanning subgraph H of D, there exists a set I and a digraph G with vertex set $I \cup J$ such that $G \setminus I = G[J] = H$ and $G^{-I} = D$.*

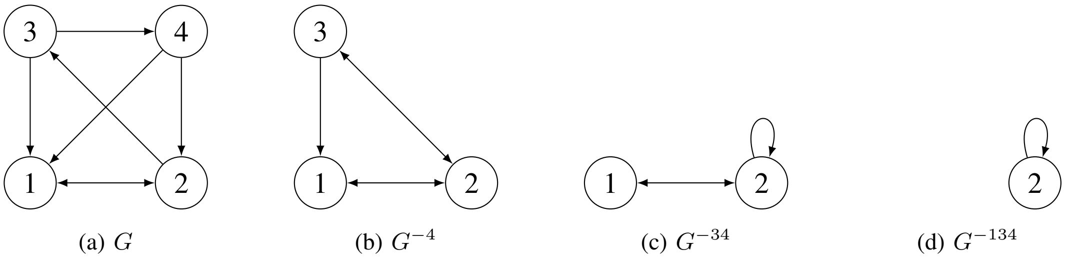


Fig. 1: Example of graph reduction.

Proof. Let I be the set of arcs in D but not in H and let G be the graph on $I \cup J$ such that $G[J] = H$ and for any arc $e = (u, v) \in I$, G contains the arcs (u, e) and (e, v). Then it is clear that $G^{-I} = D$. □

We finish this subsection by an example illustrating the reduction of digraphs in Figure 1.

C. Interaction graph of the reduced coding function

The reduction of digraphs yields an estimate on the interaction graph of the reduction of coding functions.

Proposition 2. *If I is an acyclic set of $G(f)$ then $G(f^{-I})$ is a subgraph of $G(f)^{-I}$.*

Proof. We only prove that if $G(f)$ has no loop on v, then $G(f^{-v})$ is a subgraph of $G(f)^{-v}$; the result is an easy consequence. We have

$$f_w^{-v}(x) = f_w(x_{-v}, f_v(x_{-v})),$$

hence (u, w) is an arc in $G(f^{-v})$ only if either (u, w) is already in $G(f)$ or $(u, v), (v, w) \in G(f)$, i.e. $(u, w) \in G(f)^{-v}$. □

Corollary 2. *Every reduction sequence of $G(f)$ is a reduction sequence of f.*

According to Proposition 2, we have $\text{MINDIM}(f) \leq k(G(f))$. We show that this bound on $\text{MINDIM}(f)$ is the best possible as a function of $G(f)$. The **min-net** of a digraph G over $[q] = \{0, \dots, q-1\}$ $(q \geq 2)$ is the coding function $f := \min(G, q)$ defined as

$$f_i(x) := \min\{x_j : j \in \text{in}_G(i)\}$$

with the convention that $\min(\emptyset) = q - 1$.

Proposition 3. *For any digraph G and any acyclic set I of G, $\min(G,q)^{-I} = \min(G^{-I}, q)$. Therefore, $\textsc{mindim}(\min(G,q)) = k(G)$.*

Proof. Again, we only prove the case where I is only one vertex v. If $f = \min(G,q)$, we have for all $i \neq v$

$$f_i^{-v}(x) = \min\{x_j : j \in \mathrm{in}_G(i) \setminus v, \min\{x_k : k \in \mathrm{in}_G(v)\}\} = \min\{x_k : k \in \mathrm{in}_{G^{-v}}(i)\}.$$

Thus $\min(G,q)^{-v} = \min(G^{-v}, q)$. □

Note that according to the preceding, finding a minimum reduced form of a min-net is equivalent to finding a minimum vertex set in a digraph. So finding a minimum form of a coding function is NP-Hard.

Although there exists a coding function (the min-net) whose reductions follow the reductions of its interaction graph, we prove in the following two propositions that in general we cannot say much about the interaction graph of the reduced coding function. First, we derive the analogue of Proposition 1 for the interaction graphs of coding functions.

Proposition 4. *Let D and H be any digraphs with vertex set J. Then for any $q \geq 2$ there exists a set I, a digraph G with vertex set $I \cup J$, and a coding function $f \in F(G,q)$ such that $G(f)[J] = D$ and $G(f)^{-I} = H$.*

Proof. Let $I = I_D \cup I_H$, where I_D is the set of all arcs in D but not in H and I_H is the set of all arcs in H but not in D. Then let G be the graph with vertex set $I \cup J$ such that $G[J] = D$ and for any $(u,v) \in I$, $(u,(u,v)), ((u,v),v) \in G$. For any $(u,v) \in I_D$ and any $x \in [q]^n$ (with $n = |I \cup J|$), we denote

$$y_{(u,v)} = x_u + q - 1 - x_{(u,v)}$$

and for any $j \in J$, y^j as the state with coordinates $y_{(u,j)}$ for all $u \in \mathrm{in}_D(j) \setminus \mathrm{in}_H(j)$.

Finally, let $f : [q]^n \to [q]^n$ be defined as

$$f_j(x_{\mathrm{in}_D(j) \setminus \mathrm{in}_H(j)}, x_{I_D}, x_{\mathrm{in}_D(j) \cap \mathrm{in}_H(j)}, x_{I_H}) = \min(y^j, x_{\mathrm{in}_D(j) \cap \mathrm{in}_H(j)}, x_{I_H \cap \mathrm{in}_G(j)})$$

for all $j \in J$ and

$$f_{(u,v)}(x) = x_u$$

for all $(u,v) \in I$. It is clear that $f \in F(G,q)$, hence $G(f)[J] = D$. Moreover, reducing I_D yields

$$f_j^{-I_D}(x_{\mathrm{in}_D(j)}, x_{I_H}) = \min(q-1, \dots, q-1, x_{\mathrm{in}_D(j) \cap \mathrm{in}_H(j)}, x_{I_H \cap \mathrm{in}_G(j)}) = \min(x_{\mathrm{in}_D(j) \cap \mathrm{in}_H(j)}, x_{I_H \cap \mathrm{in}_G(j)}),$$

and then reducing I_H yields

$$f_j^{-I}(x_J) = \min(x_{\mathrm{in}_D(j)\cap\mathrm{in}_H(j)}, x_{\mathrm{in}_H(j)\setminus\mathrm{in}_D(j)}) = \min(x_{\mathrm{in}_H(j)}),$$

thus $G(f^{-I}) = H$. □

Second, we prove that even reducing a single vertex may in fact remove any set of arcs from the original interaction graph.

Proposition 5. *Let G be a digraph with a vertex v such that $\mathrm{in}(v) = \mathrm{out}(v) = V \setminus v$, and let G have minimum in-degree at least 2. Then for any $q \geq 2$ and any spanning subgraph H of $G \setminus v$, there exists a coding function $f \in F(G, q)$ such that $G(f) = G$ and $G(f^{-v}) = H$.*

Proof. Say $v = n$ and for all i, let $y_i = \min\{x_i, 1\} \in \{0, 1\}$. We define the function for the vertex n:

$$f_n(x) = \bigvee_{i=1}^{n-1} y_i.$$

That way, we can focus on each vertex of H separately; without loss we only consider the vertex 1. Let $N = \mathrm{in}_G(1) \setminus n$, $P = \mathrm{in}_H(1)$ and $Q = N \setminus P$; then

$$f_1(x) = \left(\bigwedge_{p\in P} y_p\right) \wedge \left(y_n \vee \bigvee_{q\in Q} \neg y_q\right),$$

$$f_1^{-n}(x) = \left(\bigwedge_{p\in P} y_p\right) \wedge \left(\bigvee_{i=1}^{n-1} y_i \vee \bigvee_{q\in Q} \neg y_q\right) = \bigwedge_{p\in P} y_p,$$

with the convention that an empty conjunction is equal to 1 and an empty disjunction is equal to 0. □

We finish this section with an example illustrating the reduction of graphs and coding functions.

Example 2. Consider the coding function $f : \{0,1\}^4 \to \{0,1\}^4$ from Example 1. Then $G(f)$ is the graph G from Figure 1. Thus $G(f^{-4})$ is a strict subgraph of $G(f)^{-4}$ since f_1^{-4} does not depend on x_2, yet $G(f^{-34}) = G(f)^{-34}$ and $G(f^{-134}) = G(f)^{-134}$.

III. Fixed points of coding functions

A. Maximum number of fixed points

The q-**guessing number** [5] and q-**strict guessing number** of G are respectively defined as

$$g(G, q) := \log_q \max\{|\mathrm{Fix}(f)| : f \in F(G', q), G' \subseteq G\},$$

$$h(G, q) := \log_q \max\{|\mathrm{Fix}(f)| : f \in F(G, q)\}.$$

The guessing number of loopless digraphs was thoroughly investigated in [5], [40], [6], [32], [41], [42]; the strict guessing number is new. We first relate $h(G,q)$ to $g(G,q)$.

Lemma 1. *If $g(G,q) \geq 1$, then $h(G,q) \geq 1$.*

Proof. If $g(G,q) \geq 1$, then G has a cycle. Say that the vertices 0 up to $l-1$ form a chordless cycle and consider the coding function $f \in F(G,q)$ defined by

$$f_i(x) = \begin{cases} x_{i-1 \mod l} & \text{if } x_j \geq x_{i-1 \mod l} \text{ for all } j \in \text{in}(i) \\ x_{i-1 \mod l} + 1 & \text{otherwise} \end{cases}$$

if $0 \leq i \leq l-1$ and

$$f_j(x) = \min\{x_k : k \in \text{in}(j)\}$$

otherwise. Then it is clear that if G is of minimal in-degree at least one then for any $a \in [q]$, $(a, \dots, a)$ is fixed by f. Thus $h(G,q) \geq \log_q |\text{Fix}(f)| \geq 1$. Otherwise, let I_0 be the set of vertices of in-degree 0, and for $0 < k < n$, let I_k be the set of vertices i such that $\text{in}(i) \subseteq I_0 \cup \dots \cup I_{k-1}$. Then, for any $a \in [q]$, the point $x^a \in [q]^n$ such that $x^a_i = q-1$ if $i \in I_k$ for some k and $x^a_i = a$ otherwise is a fixed point of f, thus $h(G,q) \geq \log_q |\text{Fix}(f)| \geq 1$. □

Proposition 6. *For all $q \geq 2$ and any digraph G,*

$$g(G,q) \geq h(G,q) \geq g(G,q-1)\log_q(q-1) \geq g(G,q) - n\log_q\left(1 + \frac{1}{q-1}\right).$$

Proof. The first inequality is trivial. We now prove the second. Let $f : [q-1]^n \to [q-1]^n$ with interaction graph $G' \subseteq G$ and with $(q-1)^{g(G,q-1)}$ fixed points. Let $f' \in F(G,q)$ such that

$$f'_v(x) = \begin{cases} f_v(x) & \text{if } x_{\text{in}(v)} \in [q-1]^{\text{ind}(v)} \\ q & \text{otherwise.} \end{cases}$$

Then $\text{Fix}(f) \subseteq \text{Fix}(f')$ and hence $(q-1)^{g(G,q-1)} = |\text{Fix}(f)| \leq |\text{Fix}(f')| \leq q^{h(G,q)}$.

Let us now prove the third inequality. Let $f : [q]^n \to [q]^n$ with interaction graph $G' \subseteq G$ and with $q^{g(G,q)}$ fixed points. Then for any vertex v and any permutation π of $[q]$, consider the coding function $f^{v,\pi}$ defined as

$$f^{v,\pi}_u(x) = \begin{cases} \pi(f_v(\pi^{-1}(x_v), x_{-v})) & \text{if } u = v \\ f_u(\pi^{-1}(x_v), x_{-v}) & \text{otherwise.} \end{cases}$$

Then $x \in \mathrm{Fix}(f)$ if and only if $(\pi(x_v), x_{-v}) \in \mathrm{Fix}(f^{v,\pi})$ and hence $|\mathrm{Fix}(f^{v,\pi})| = q^{g(G,q)}$. Denote

$$\begin{aligned} R(v,a) &:= |\{x \in \mathrm{Fix}(f) : x_v = a\}| = |\{x \in \mathrm{Fix}(f^{v,\pi}) : x_v = \pi(a)\}|, \\ r(v) &:= \min_{a \in [q]} R(v,a) \le q^{-1} q^{g(G,q)}. \end{aligned}$$

Consider a permutation σ of $[q]$ such that $r(v) = R(v, \sigma^{-1}(q-1))$; we then obtain

$$\begin{aligned} |\{x \in \mathrm{Fix}(f^{v,\sigma}) : x_v \in [q-1]\}| &= |\{x \in \mathrm{Fix}(f) : x_v \in \sigma^{-1}([q-1])\}| \\ &= \sum_{a \neq \sigma^{-1}(q-1)} R(v,a) \\ &= |\mathrm{Fix}(f)| - R(v, \sigma^{-1}(q-1)) \\ &\ge (1 - q^{-1}) q^{g(G,q)}. \end{aligned}$$

Thus, $f^{v,\sigma}$ has at least $(1-q^{-1})q^{g(G,q)}$ fixed points with $x_v \in [q-1]$. Applying this strategy recursively for all n vertices, we find that there exists a coding function with at least $(1-q^{-1})^n q^{g(G,q)}$ fixed points in $[q-1]^n$. By considering the restriction of this coding function to $[q-1]^n$, we obtain $(q-1)^{g(G,q-1)} \ge (1-q^{-1})^n q^{g(G,q)}$. □

Corollary 3. *We have* $\lim_{q \to \infty} h(G,q) = \lim_{q \to \infty} g(G,q) = H(G)$ *for all G, where* $H(G)$ *is the entropy of G [31].*

B. Fixed points and reduction

Proposition 7 (See [37])**.** *Let f be a coding function and h be a reduced form of f. With the convention that f has a unique fixed point if* $\mathrm{DIM}(f) = 0$*, f and h have the same number of fixed points.*

Proof. Again, we can assume that $h = f^{-v}$ for some vertex v without a loop in $G(f)$. We then have $f_i(x) = x_i$ for all i if and only if $f_v(x) = x_v$ and $f_i^{-v}(x) = f_i(x_{-v}, f_v(x)) = f_i(x) = x_i$ for all $i \neq v$. □

Example 3. Let f be the coding function in Example 1. The fixed points of f and its successive reductions are respectively given by

$$\begin{aligned} \mathrm{Fix}(f) &= \{(0,0,0,0),(1,1,1,1)\}, \\ \mathrm{Fix}(f^{-4}) &= \{(0,0,0),(1,1,1)\}, \\ \mathrm{Fix}(f^{-34}) &= \{(0,0),(1,1)\}, \\ \mathrm{Fix}(f^{-134}) &= \{0,1\}, \end{aligned}$$

and successive reductions preserve the number of fixed points. In particular, since $k(G(f)) = 1$ and f^{-134} is the identity of $A^{k(G(f))}$, Proposition 7 indicates that f is indeed a solution for $G(f)$.

Let $S = V \setminus I$ be a feedback vertex set of $G(f)$. Then according to Proposition 2, f has a reduced form f^{-I} with dimension $|S|$. So f^{-I} has obviously at most $q^{|S|}$ fixed points, and since f and f^{-I} have the same number of fixed points, f has at most $q^{|S|}$ fixed points. This provides an alternative proof of (a modified form of) a theorem of Aracena [25] (see Riis [31]): If S is a feedback vertex set of $G(f)$, then f has at most $q^{|S|}$ fixed points. In other words, $h(G, q) \leq k(G)$; by obvious extension, we obtain $g(G, q) \leq k(G)$ as well.

In particular, if $G(f)$ has no cycle, then f has a reduced form f^{-V} of dimension zero. Then f^{-V} has a unique fixed point and we deduce that f has a unique fixed point. This provides an alternative proof of a theorem of Robert [43]: If $G(f)$ is acyclic, then f has a unique fixed point.

As seen below, we cannot say anything interesting about the guessing number of reduced digraphs in general.

Proposition 8. *The guessing number of G and that of its reduction G^{-v} are related as follows.*

1) *Let G be a digraph and v a vertex of G, then $g(G, q) \leq g(G^{-v}, q)$ for all $q \geq 2$.*
2) *If G is acyclic, then $h(G, q) = g(G, q) = g(G^{-v}, q) = h(G^{-v}, q) = 0$.*
3) *For any $n \geq 3$ there exists G on n vertices such that $g(G, q) = h(G, q) = 1$, $h(G^{-v}, q) = \log_q(q^{n-1} - 1)$ and $g(G^{-v}, q) = n - 1$ for all q.*

Proof. The first two statements are clear. Now consider the complete bipartite graph $G = K_{n-1,1}$, also called the star on n vertices, where n is the centre of the star. Then this vertex forms a feedback vertex set and hence $g(G, q) = 1$, and by Lemma 1, $h(G, q) = 1$. Then G^{-n} is the complete graph on $n - 1$ vertices with a loop on each vertex, and we have $g(G^{-n}, q)$ and $h(G^{-n}, q)$ from Theorem 3 and Example 4 below. □

C. Fixed points of fully reduced coding functions

We are then interested in studying the number of fixed points of coding functions which are fully reduced, i.e. whose interaction graphs have a loop on each vertex. For any loopless digraph G, we denote the graph obtained from G by adding a loop on each vertex as $\mathring{G}$. Clearly, $g(\mathring{G}, q) = n$; moreover, $h(\mathring{G}, q) = n$ if and only if G is empty (this is the interaction graph of the identity function).

For any loopless G, an **in-dominating set** (IDS) is a set of vertices $X \subseteq V$ such that for all $v \in V$

with positive in-degree, either $v \in X$ or $\mathrm{in}(v) \cap X \neq \emptyset$. Denote the number of IDSs of G of size k as $I_k(G)$; clearly, $I_n(G) = 1$.

Theorem 3. *For any loopless graph G,*

$$h(\mathring{G}, q) = \log_q \sum_{k=0}^{n} (q-1)^k I_k(G).$$

Proof. For any property $\mathcal{P}$, we denote the function which returns 1 if $\mathcal{P}$ is satisfied and 0 otherwise as $\mathbb{1}\{\mathcal{P}\}$. Also, we write $\mathrm{in}(i)$ and $\mathrm{out}(i)$ for $\mathrm{in}_{\mathring{G}}(i)$ and $\mathrm{out}_{\mathring{G}}(i)$ so that $i \in \mathrm{in}(i) \cap \mathrm{out}(i)$. We define the coding function $g \in F(\mathring{G}, q)$ as

$$g_i(x) := \begin{cases} x_i & \text{if } \mathrm{ind}(i) = 1 \\ x_i + \mathbb{1}\{x_{\mathrm{in}(i)} = (0, \dots, 0)\} \mod q & \text{otherwise.} \end{cases}$$

For any x, $x = g(x)$ if and only if $\{v \in V : x_v \neq 0\}$ is an in-dominating set. This proves the lower bound.

Now let f with $G(f) = \mathring{G}$ and $q^{h(\mathring{G},q)}$ fixed points. Any local function of f is expressed as

$$f_i(x) = \begin{cases} a(x_i) & \text{if } \mathrm{ind}(i) = 1 \\ x_i + e_i(x_{\mathrm{in}(i)}) \mod q & \text{otherwise,} \end{cases}$$

where $e_i(x_{\mathrm{in}(i)}) = f_i(x) - x_i \mod q$. It is clear that the optimal choice for the function a is simply $a(x_i) = x_i$. Therefore, we only focus on the case where $\mathrm{ind}(i) \geq 2$ henceforth.

We now show that we can always assume that e_i takes a non-zero only once. Let $Y = \{y \in A^{\mathrm{ind}(i)} : e_i(y) \neq 0\}$ and let $y^i \in Y$. Now, let f' such that $f'_j = f_j$ for all $j \neq i$ and

$$f'_i(x) = x_i + \mathbb{1}\{x_{\mathrm{in}(i)} = y^i\} \mod q.$$

Suppose $f(x) = x$, then $f'_j(x) = f_j(x) = x_j$ for all $j \neq i$; moreover, $x_{\mathrm{in}(i)} \notin Y$ hence $x_{\mathrm{in}(i)} \neq y^i$ and $f'_i(x) = f_i(x) = x_i$. Therefore, $|\mathrm{Fix}(f')| \geq |\mathrm{Fix}(f)|$.

Hence we can consider f' instead. We now show that choosing $y^i = (0, \dots, 0)$ for any vertex i maximises the number of fixed points. Consider a vertex k and define a new function f'' as

$$f''_j = f'_j \quad \forall\, j \notin \mathrm{out}(k),$$
$$f''_i(x) = x_i + \mathbb{1}\{x_{\mathrm{in}(i)} = z^i\} \mod q \quad \forall\, i \in \mathrm{out}(k),$$

where $z^i_j = y^i_j$ if $j \neq k$ and $z^i_k = 0$. Let $x' \in \mathrm{Fix}(f') \setminus \mathrm{Fix}(f'')$. Then there exists $i \in \mathrm{out}(k)$ such that $z^i = x'_{\mathrm{in}(i)} \neq y^i$. Defining x'' by only changing the k-coordinate of x' to $x''_k := y^i_k$ we obtain

$z^i \neq x''_{\mathrm{in}(i)} = y^i$ and $z^j \neq x''_{\mathrm{in}(j)}$ for all $j \in \mathrm{out}(k)\backslash i$ (because $x''_k > 0 = z^j_k$). Thus $x'' \in \mathrm{Fix}(f'')\backslash\mathrm{Fix}(f')$. Hence, there is an injection from $\mathrm{Fix}(f') \setminus \mathrm{Fix}(f'')$ to $\mathrm{Fix}(f'') \setminus \mathrm{Fix}(f')$, thus f'' has at least as many fixed points as f'.

Thus, we can always choose $z^i_k = 0$ for all i and all k, which yields the coding function g. □

Corollary 4. *For any loopless G, we have*

$$h(\mathring{G}, q) \geq n \log_q(q-1) + \log_q\left(1 + \frac{n}{q-1}\right),$$

and hence $\lim_{q\to\infty} h(\mathring{G}, q) = n$.

Proof. For any $v \in V$, $V \setminus v$ is an IDS. Therefore, $I_{n-1}(G) = n$ and since V is also an IDS, $I_n(G) = 1$. Therefore, $h(\mathring{G}, q) \geq \log_q\big(n(q-1)^{n-1} + (q-1)^n\big)$. □

Example 4. In general, computing the sum $\sum_k (q-1)^k I_k(G)$ is #P-Complete. However, we can exhibit five special cases for which the formula is easy to derive; all graphs have vertex set $V = \{1, \dots, n\}$.

- For the clique K_n (with arcs (i,j) for all $i \neq j$),

$$h(\mathring{K}_n, q) = \log_q(q^n - 1).$$

- For the transitive tournament T_n (with arcs (i,j) for all $i < j$),

$$h(\mathring{T}_1, q) = 1 \quad \text{and} \quad h(\mathring{T}_n, q) = n - 2 + \log_q(q^2 - 1)\ \forall\, n \geq 2.$$

- For the inward directed star iS_n (with arcs (i,n) for all $1 \leq i \leq n-1$),

$$h(i\mathring{S}_n, q) = \log_q(q^n - 1).$$

- For the outward directed star oS_n (with arcs (n,i) for all $1 \leq i \leq n-1$),

$$h(o\mathring{S}_n, q) = \log_q(q^n - q^{n-1} + (q-1)^{n-1}).$$

- For the undirected star (with arcs (i,n) and (n,i) for all $1 \leq i \leq n-1$),

$$h(\mathring{S}_n, q) = \log_q(q^n - q^{n-1} + (q-1)^{n-1}).$$

Proof. For K_n, we have $I_0(K_n) = 0$ and $I_k(K_n) = \binom{n}{k}$ for all $1 \leq k \leq n$. Therefore, $\sum_k (q-1)^k I_k(K_n) = q^n - 1$. For T_n ($n \geq 2$), a set of vertices X is an in-dominating set if and only if it contains either the first or the second vertex. Therefore, $I_k(T_n) = \binom{n}{k} - \binom{n-2}{k}$ and $\sum_k (q-1)^k I_k(T_n) = q^n - q^{n-2}$. The proof for the stars is similar: we have $I_0(iS_n) = 0$ and $I_k(iS_n) = \binom{n}{k}$ for all $1 \leq k \leq n$; we also have $I_{n-1}(S_n) = I_{n-1}(oS_n) = n$ and $I_k(S_n) = I_k(oS_n) = \binom{n-1}{k-1}$ otherwise. □

IV. Application to linear network coding solvability

A. *Network coding solvability and guessing number*

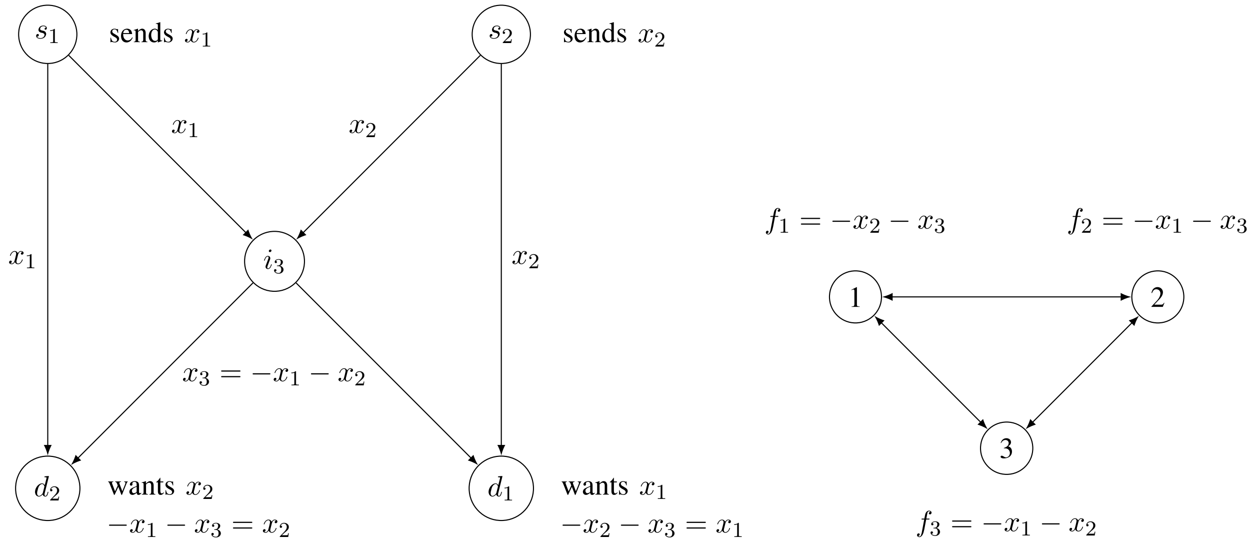
We now apply the theory of coding function reduction to linear network coding solvability. The network coding solvability problem asks whether a given network coding instance is solvable, i.e. whether all messages can be transmitted to their destinations simultaneously. In particular, if the local functions f_v are linear, then the instance is linearly solvable. For the study of solvability, any network coding instance can be converted into a **multiple unicast** without any loss of generality [44], [5]. A multiple unicast instance consists of an acyclic network N and a finite alphabet A of cardinality q, where

- each arc in the network carries an element of A;
- the instance is given in its so-called circuit representation, i.e. the same message flows on every arc coming out of the same vertex;
- the network has k sources $s_1, \dots, s_k$, k destinations $d_1, \dots, d_k$, and α intermediate nodes $i_{k+1}, \dots, i_{k+\alpha}$;
- each destination d_i ($1 \le i \le k$) requests an element from A from a corresponding source s_i.

This network coding instance is **solvable** over A if all the demands of the destinations can be satisfied at the same time.

The solvability of a multiple unicast instance can be decided by determining the guessing number of a related digraph. By merging each source with its corresponding destination node into one vertex, we form the digraph G_N on $n := k + \alpha$ vertices. In general, we have $g(G_N, q) \le k$ for all q and the original network coding instance is solvable over A if and only if $k(G_N) = k$ (this condition is purely graph-theoretic and does not involve coding functions; as such, we assume it is always satisfied) and $g(G_N, q) = k$, in which case we say that G_N is solvable over A [5] (an analogous result holds for linear solvability). Therefore, while network coding considers how the information flows from sources to destinations, the guessing number captures the intuitive notion of how much information circulates through the digraph.

We illustrate the conversion of a network coding instance to a guessing number problem for the famous butterfly network in Figure 2. It is well-known that the butterfly network is solvable over all alphabets, and conversely the clique K_3 has guessing number 2 over any alphabet. The solutions are shown in Figure 2 and indeed the operations done in the butterfly network correspond to the fixed point equations on the clique.



(a) Butterfly Network coding instance

(b) Corresponding graph K_3

Fig. 2: The butterfly network.

B. Solvability by non-decreasing coding functions

We first apply the reduction approach to network coding solvability by non-decreasing coding functions. Here, we consider $A = [q] = \{0, \dots, q-1\}$ with the usual linear order. We then say that a local function f_v is non-decreasing if it is non-decreasing in every variable x_u; the coding function is **non-decreasing** if all its local functions are non-decreasing. For instance, the min-net introduced in Section II-C is a non-decreasing coding function. Non-decreasing coding functions have been widely studied (see [25], [29], [28]); they are usually represented by an interaction graph with positive signs on all arcs (see [29] and the references therein for a survey of the work on signed interaction graphs).

More closely related to network coding, **routing** can be viewed as a non-decreasing coding function. Indeed, routing corresponds to local functions of the form $f_v(x_u) = x_u$ for some $u \in \mathrm{in}(v)$. Routing then achieves a guessing number of $c(G)$, where c is the maximum number of **disjoint cycles** in G; thus a graph is solvable by routing if and only if $c(G) = k(G)$. We first extend this result slightly by considering **unary** coding functions, where $f_v(x_u)$ can be any function of one variable.

Lemma 2. *For any digraph G, the following are equivalent:*

1) $c(G) = k(G)$.
2) *G is solvable by routing over any alphabet.*

3) *G is solvable by routing over some alphabet.*

4) *G is solvable by a unary coding function over any alphabet.*

5) *G is solvable by a unary coding function over some alphabet.*

Proof. Clearly, 1) $\Rightarrow$ 2) $\Rightarrow$ 3) $\Rightarrow$ 5) and 2) $\Rightarrow$ 4) $\Rightarrow$ 5).

5) $\Rightarrow$ 1). If f is a unary coding function with $G(f) \subseteq G$ and $q^{k(G)}$ fixed points, then $G(f)$ is a disjoint union of at most $c(G)$ cycles. Hence f has at most $q^{k(G(f))} \leq q^{c(G)}$ fixed points and $c(G) = k(G)$. □

It is shown in [28] that general non-decreasing coding functions can significantly outperform routing in terms of guessing number: for instance, on the clique K_n, routing achieves a guessing number of $c(K_n) = \lfloor n/2 \rfloor$, while non-decreasing functions achieve $n - 3 - \epsilon$ when the alphabet is large enough [28, Proposition 6]. However, Theorem 4 below proves that non-decreasing functions do not outperform routing in terms of solvability. We say that G is solvable by routing if it satisfies any of the equivalent properties of Lemma 2.

Theorem 4. *For any digraph G, the following are equivalent:*

1) $c(G) = k(G)$.

2) *G is solvable by routing.*

3) *G is solvable by a non-decreasing coding function over any alphabet.*

4) *G is solvable by a non-decreasing coding function over some alphabet.*

Proof. By Lemma 2, 1) $\Leftrightarrow$ 2) $\Rightarrow$ 3) $\Rightarrow$ 4).

4) $\Rightarrow$ 1). Let $f : [q]^n \to [q]^n$ be a non-decreasing coding function with $G(f) \subseteq G$ and $q^{k(G)}$ fixed points. Let I be a maximal acyclic set of G such that $|V \setminus I| = k(G)$. Since f^{-I} and f have the same number of fixed points, f^{-I} is the identity on $[q]^{V \setminus I}$.

For every vertex $v \notin I$, let $e_v \in [q]^{V \setminus I}$ be the v-th unit vector defined by $(e_v)_v = 1$ and $(e_v)_u = 0$ for all $u \neq v$; we set $\overline{e_v} = 1 - e_v$. Consider

$$C_v := \{v\} \cup \{i \in I : F_i^I(e_v) > F_i^I(\overline{e_v})\}.$$

Claim. *For all distinct $u, v \notin I$, $C_u \cap C_v = \emptyset$.*

Proof. For any $i \in I$, F_i^I is a non-decreasing function of x_{-I}. Since $\overline{e_u} \geq e_v$, we obtain for any $i \in C_u \cap C_v$:

$$F_i^I(e_u) > F_i^I(\overline{e_u}) \geq F_i^I(e_v) > F_i^I(\overline{e_v});$$

yet $\overline{e_v} \geq e_u$ implies $F_i^I(\overline{e_v}) \geq F_i^I(e_u)$, which is the desired contradiction. □

Claim. *For all $v \notin I$, C_v contains a cycle.*

Proof. We only need to show that for all $i \in C_v$, $C_v \cap \mathrm{in}(i) \neq \emptyset$. Firstly, let $i \in C_v \setminus \{v\}$, and suppose that $C_v \cap \mathrm{in}(i) = \emptyset$, then $F^I_{\mathrm{in}(i)\cap I}(\overline{e_v}) \geq F^I_{\mathrm{in}(i)\cap I}(e_v)$ and $(\overline{e_v})_{-v} \geq (e_v)_{-v}$. Hence

$$F^I_i(\overline{e_v}) = f_i\left((\overline{e_v})_{-v}, F^I_{\mathrm{in}(i)\cap I}(\overline{e_v})\right) \geq f_i\left((e_v)_{-v}, F^I_{\mathrm{in}(i)\cap I}(e_v)\right) = F^I_i(e_v),$$

which contradicts the fact that $i \in C_v$. Secondly, let $i = v$, and again suppose that $C_v \cap \mathrm{in}(v) = \emptyset$; thus $F^I_{\mathrm{in}(v)\cap I}(\overline{e_v}) \geq F^I_{\mathrm{in}(v)\cap I}(e_v)$. Recall that f^{-I} is the identity, hence

$$0 = f^{-I}_v(\overline{e_v}) = f^{-I}_v\left((\overline{e_v})_{-v}, F^I_{\mathrm{in}(v)\cap I}(\overline{e_v})\right) \geq f^{-I}_v\left((e_v)_{-v}, F^I_{\mathrm{in}(v)\cap I}(e_v)\right) = f^{-I}_v(e_v) = 1.$$

□

By the claims above, we have $n - |I| = k(G)$ disjoint cycles in the graph G. □

C. Linearly solvable undirected graphs

We are now interested in linear coding functions. A **linear coding function** is any coding function $f : R^V \to R^V$, where R is a commutative ring of order q and such that $f_i(x) = \sum_{u \in \mathrm{in}(i)} a_{i,u} x_u$ for some $a_{i,u}$ invertible in R. For any G we denote the set of linear coding functions with interaction graph G over a commutative ring of order q as $L(G, q)$. The set of fixed points of a linear coding function forms a submodule of R^V, hence we denote the q-**linear guessing number** [5], [6], [45] and q-**strict linear guessing number** of G respectively as

$$g_L(G, q) = \max\{\dim \mathrm{Fix}(f) : f \in L(G', q), G' \subseteq G\},$$

$$h_L(G, q) = \max\{\dim \mathrm{Fix}(f) : f \in L(G, q)\}.$$

We say that a digraph G is **linearly solvable** if $g_L(G, q) = g(G, q) = k(G)$ for some q. We say it is **strictly linearly solvable** if $h_L(G, q) = h(G, q) = k(G)$. It is easy to prove that G is linearly solvable if and only if G has a strictly linearly solvable spanning subgraph H with $k(G) = k(H)$.

The minimum number of parts in any partition of the vertex set of G into cliques is denoted as $\mathrm{cp}(G)$; if G is undirected, then $\mathrm{cp}(G) = \chi(\bar{G})$, the chromatic number of its complement. We say that a digraph G is **vertex-full** if all its vertices can be covered by $\alpha(G)$ cliques. In other words, G is vertex-full if and only if $\mathrm{cp}(G) = \alpha(G)$.

We can easily obtain a classical lower bound on the guessing number. Firstly, the clique K_n is always linearly solvable over all alphabets by the following coding function f (see Figure 2):

$$f_i(x_{-i}) = -\sum_{j \neq i} x_j \mod q.$$

Indeed, all states summing to zero mod q are fixed by f and hence $g_L(K_n, q) = g(K_n, q) = n - 1$. (For $n = 1$, we simply set $f(x) = 0$.) Therefore, if we partition the vertex set of G into $\mathrm{cp}(G)$ cliques and apply the corresponding coding function on each clique, we obtain a linear coding function with $q^{n-\mathrm{cp}(G)}$ fixed points, thus yielding [32]

$$g_L(G, q) \geq n - \mathrm{cp}(G).$$

This lower bound implies that vertex-full graphs are linearly solvable over all alphabets. On the other hand, many classes of linearly solvable digraphs are not vertex-full, e.g. the directed cycle (see [6] for more striking examples). Until now, however, the only known linearly solvable undirected graphs are vertex-full. Based on the results in [42], we can construct the first example of a linearly solvable undirected graph which is not vertex-full. Firstly, for two digraphs G_1 and G_2 on disjoint vertex sets of sizes n_1 and n_2 respectively, their **bidirectional union** is $G := G_1 \bar{\cup} G_2$ where G_1 and G_2 are linked by all possible edges between them. The linear guessing number then satisfies for all q [6]

$$g_L(G, q) = \min\{n_1 + g_L(G_2, q), n_2 + g_L(G_1, q)\}.$$

Theorem 5. *There exists an undirected graph which is linearly solvable, and yet it is not vertex-full.*

Proof. Let $G_1 := E_6$ denote the empty graph on $n_1 := 6$ vertices and with linear guessing number 0 for any q. Let $G_2 := \mathfrak{C}$ denote the Clebsch graph: $\mathfrak{C}$ has $n_2 := 16$ vertices, independence number $\alpha(\mathfrak{C}) = 5$, and $g_L(\mathfrak{C}, 3) \geq 10$ [42]. Since $\mathfrak{C}$ is triangle-free but not vertex-full, it is not linearly solvable as we shall see below. Nonetheless, the graph $G := E_6 \bar{\cup} \mathfrak{C}$ is linearly solvable but not vertex-full, since $n = 22$, $\alpha(G) = 6$ and $g_L(G, 3) = 16$. □

Definition 5. Let I be a non-empty acyclic set I of a digraph G.

- I is **strongly compatible** if for all $u, v \notin I$, $(u, v) \in G$ if and only if there is a path from u to v through I.
- I is **weakly compatible** if for all $u, v \notin I$, the following holds: if (u, v) is an arc, then there is a path from u to v through I; otherwise, there is either no path from u to v through I or there are at least two paths from u to v through I.

Theorem 6. *If G is strictly linearly solvable over some alphabet, then all maximum acyclic sets of G are weakly compatible.*

The proof of the theorem is based on the following lemma: if we consider the interaction graph $G(f)$

of a linear coding function f, we can only *erase* an arc from $G(f)$ if we use a path through I.

Lemma 3. *Let f be a linear coding function and I be an acyclic set of $G(f)$, and any vertices u, v outside of I such that $(u,v) \in G(f)$ but $(u,v) \notin G(f^{-I})$. Then there exists a path in $G(f)$ from u to v through I.*

Proof. Suppose $(u,v) \in G(f)$ but $(u,v) \notin G(f^{-I})$ and that there is no path in $G(f)$ from u to v through I. Denote $f_v(x) = \sum_{j \in \mathrm{in}(v)} a_j x_j$. Denote $N = \mathrm{in}(v) \cap I$, then there is no path in $G(f)$ from u to N through I; as such there is no arc from u to N in $G(f^{-(I \setminus N)})$. Thus, we have

$$f_v^{-I}(x) = a_u x_u + \sum_{j \neq u} b_j x_j;$$

the only occurrence of the variable x_u is due to the original $f_v(x)$. □

Proof of Theorem 6. Suppose that I is a maximum acyclic set of G which is not weakly compatible. There are two ways weak compatibility can be violated.

1) Let $u, v \notin I$ such that $(u,v) \in G$ and yet there is no path from u to v through I. Then by Lemma 3 for any linear coding function $f \in L(G,q)$, (u,v) is an arc in $G(f^{-I})$, thus by Theorem 3 f has fewer than $q^{k(G)}$ fixed points.
2) Let $u, v \notin I$ such that $(u,v) \notin G$ and yet there is a unique path from u to v through I. If $f_a(x) = \sum_b c_{a,b} x_b$ for all a and $b \in \mathrm{in}(a)$ and if the path is $u_0 = u, u_1, \dots, u_k = v$, it is easy to check that the x_u term in f_v^{-I} is $\prod_{i=1}^k c_{u_i, u_{i-1}} \neq 0$. Again f^{-I} is not the identity and f has fewer than $q^{k(G)}$ fixed points.

□

Not all undirected graphs G where all the maximum independent sets are weakly compatible are vertex-full. For instance, the bidirectional union $G = E_3 \bar{\cup} \bar{\mathfrak{G}}$ of an independent set of size three E_3 with the complement of the Grötzsch graph $\bar{\mathfrak{G}}$ is a counter-example. The Grötzsch graph is illustrated in Figure 3; it is triangle-free and has chromatic number 4. Therefore, its complement is not vertex-full: $\alpha(\bar{\mathfrak{G}}) = 2$ while $\mathrm{cp}(\bar{\mathfrak{G}}) = 4$. In G, E_3 then forms a maximum independent set, which is clearly weakly compatible; however $\alpha(G) = 3$ while $\mathrm{cp}(G) = 4$, thus G is not vertex-full.

Nonetheless, we can classify linearly solvable triangle-free undirected graphs. A **matching** in a digraph is a union of disjoint undirected edges in the digraph. We denote the number of edges in a maximum matching in the digraph G as $\mu(G)$. If G is undirected, then it is easily seen that $c(G) = \mu(G)$; hence G is solvable by routing if and only if $\mu(G) = k(G)$. Moreover, if G is triangle-free, then these properties

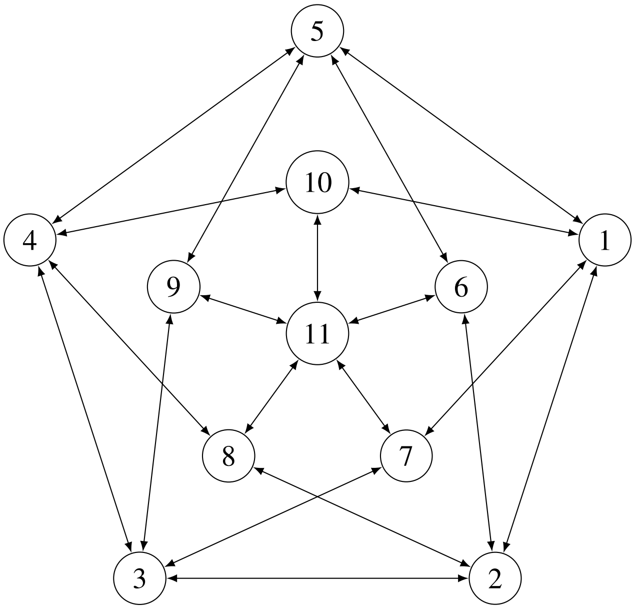


Fig. 3: The Grötzsch graph $\mathfrak{G}$.

are in turn equivalent to G being vertex-full. Theorem 7 then proves that if an undirected triangle-free graph is solvable by linear network coding, then it is solvable by routing.

Theorem 7. *Let G be an undirected triangle-free graph. The following are equivalent:*

1) $\mu(G) = k(G)$.
2) *G is vertex-full.*
3) *G is solvable by routing.*
4) *G is linearly solvable over all alphabets.*
5) *G is linearly solvable over some alphabet.*

Proof. 1) $\Leftrightarrow$ 2). If G is undirected, then $\mu(G) = k(G)$ if and only if G has a minimum feedback vertex set J and a maximum matching M such that every vertex in J belongs to exactly one edge of M and every edge of M has exactly one endpoint in J [46, Theorem 5.15]. If G is triangle-free, this is equivalent to covering all the vertices of G with $\alpha(G)$ cliques, namely the $k(G)$ edges of M and the remaining $\alpha(G) - k(G)$ vertices not covered by M.

1) $\Leftrightarrow$ 3) by Lemma 2 and the fact that $\mu(G) = c(G)$ for undirected graphs.

Clearly, 3) $\Rightarrow$ 4) $\Rightarrow$ 5).

5) $\Rightarrow$ 1). Suppose G is triangle-free and linearly solvable over some alphabet. Then there exists

a subgraph H of G such that H is strictly linearly solvable and $k(H) = k(G)$. H is not necessarily undirected, hence we denote the undirected graph obtained from H by adding any arc (u, v) if $(v, u) \in H$ as $\bar{H}$; thus $\bar{H}$ is a spanning subgraph of G with $k(\bar{H}) = k(G)$. Let I be a maximum independent set of G, then I is also a maximum acyclic set of H; by Theorem 6, it is weakly compatible in H. Then if (u, v) is an arc in H outside of I, there exists $i \in I$ such that u, i, and v form a triangle in $\bar{H}$, which is impossible. Thus $\bar{H}$ is bipartite and by the König-Egerváry theorem [47], $\mu(\bar{H}) = k(\bar{H}) = k(G)$. Since $\mu(G) \geq \mu(\bar{H})$, we obtain that $\mu(G) = k(G)$. □

We now prove that strictly linearly solvable complements of triangle-free graphs are vertex-full as well.

Theorem 8. *Let G be an undirected graph with $\alpha(G) = 2$. If G is strictly linearly solvable over some alphabet, then G is vertex-full (or equivalently, G is the complement of a bipartite graph).*

Proof. If G is strictly linear solvable over some alphabet, then by Theorem 6, every non-edge is weakly compatible, and we prove that this implies that G is vertex-full, i.e. that the vertex set of G can be partitioned into two cliques. Let ab be a non-edge in G, let C_a be a maximal clique containing a, and let C_b be a maximal clique containing b. If C_a and C_b cover all vertices, we are done. Otherwise, there exists c which does not belong to either clique.

Claim 1. *There exist $d \in C_a$, $e \in C_b$, disjoint from a and b, such that a, b, c, d, e induce a graph with exactly 7 edges and the following 3 non-edges: ab, cd and ce.*

Proof. Since a, b, c cannot form an independent set, without loss ac is an edge. By maximality of C_a, there exists $d \in C_a$ such that ad is an edge and cd is a non-edge. Then ab is weakly compatible, cd is a non-edge, and they have a common neighbour (namely, a) in ab: cd must have another common neighbour, namely b, which means that bc and bd are edges. In turn, there exists $e \in C_b$ such that be is an edge and ce is not an edge; as above, ae is also an edge. Finally, since c, d, e cannot form an independent set, de is an edge. □

Claim 2. *If c and f do not belong to C_a or C_b, then cf is an edge.*

Proof. The vertices corresponding to c are a to e as in Claim 1; let f not in C_a or C_b either and suppose that cf is not an edge. Since c, d, f cannot form an independent set, fd is an edge. Thus there exists $g \in C_a$ with $q \neq d$ such that fg is not an edge, and similarly there exists $h \in C_b$ with $h \neq e$ such that fh is not an edge. Now, cd is weakly compatible, fg is not an edge and they only have d as common neighbour in cd, which is a contradiction. □

Therefore, the vertices outside of C_a or C_b form a clique, which we shall refer to as C_c.

Claim 3. *Let $f \in C_c$ and $g \in C_a$ such that fg is not an edge. Then for any $i \in C_b$, gi is an edge.*

Proof. Suppose that gi is not an edge. Since f, g, i cannot form an independent set, fi is an edge. Then using the notation above, gi is weakly compatible, fh is not an edge and they only have i as a common neighbour in gi, which is a contradiction. □

By the above, the following two sets of vertices induce disjoint cliques and cover all vertices:

1) C_c and all the vertices in C_a connected to all the vertices in C_c;
2) C_b and all the remaining vertices of C_a.

□

D. Non linearly solvable digraphs

Theorem 6 yields and easy way to construct digraphs that are not strictly linearly solvable. Indeed, let $I = \{i_1, \ldots, i_m\}$ in topological order be a maximum acyclic set of G and let (u, v) be an arc outside of I such that the out-neighbourhood of u in I is after (in topological order) than the in-neighbourhood of v. Then there is no path from u to v through I and I is not weakly compatible.

More interestingly, based on Theorem 6, we can construct digraphs which are not linearly solvable. The strategy to construct such a digraph G uses two main ideas. Firstly, we force any possible linear solution to use an arc (u, v) in a minimum feedback vertex set J. This can be done by ensuring that the graph obtained by removing (u, v) has a smaller feedback vertex set than G. Secondly, we make sure that there is no path from u to v through the corresponding maximum acyclic induced subgraph $I = V \setminus J$. Thus, I is not weakly compatible and by Theorem 6, the graph is not linearly solvable.

Let $G_k = (I \cup J, E)$ be any digraph such that

- $I = \{i_1, \ldots, i_{k-1}\}$ and $J = \{j_1, \ldots, j_k\}$ are disjoint;
- I is acyclic;
- $J \setminus \{j_k\}$ is acyclic;
- J contains a path from j_1 to j_k;
- j_k only has one out-neighbour in J, namely j_1;
- I and J are connected using undirected edges as follows: $i_1 j_1$, $i_a j_b$ for all $1 \leq a \leq k-1$ and $2 \leq b \leq k-1$, and $i_c j_k$ for all $2 \leq c \leq k-1$.

A graph G_3 is illustrated in Figure 4, where we have chosen the graph which included all possible arcs.

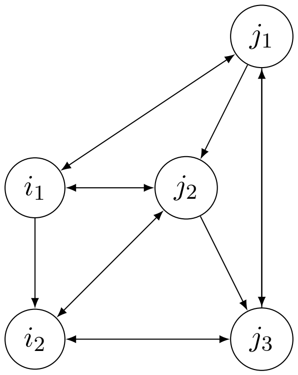


Fig. 4: Example of a non linearly solvable digraph: G_3.

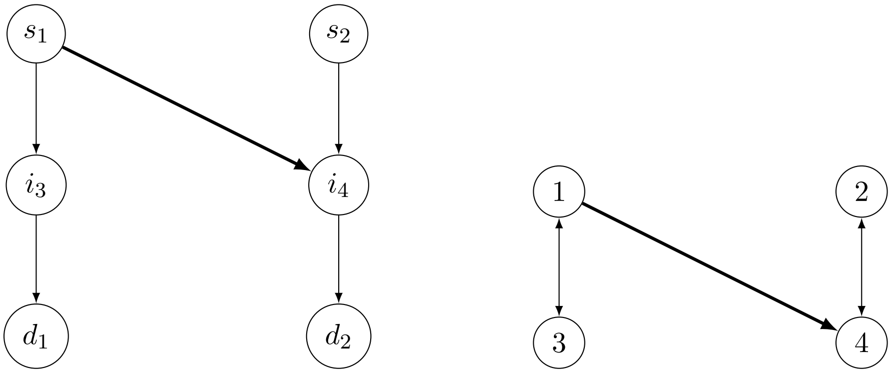
Theorem 9. *For any $k \geq 2$, $k(G_k) = k$ and $g_L(G_k, q) = k - 1$ for all q.*

Proof. We first verify that I is a maximum acyclic set, i.e. that no set of k vertices is acyclic. Let S be a set of k vertices. If $S = J$, S is not acyclic. Now suppose S contains a vertex $i \in I$. Firstly, suppose $i = i_1$, then if S contains j_b for $1 \leq b \leq k-1$, S contains the cycle $i_1 j_b$, otherwise the only case left is $S = I \cup \{j_k\}$, which again has a cycle $i_{k-1} j_k$. Secondly, suppose $i = i_a$ for $2 \leq a \leq k-1$, then if S contains j_b for $2 \leq b \leq k$, S contains the cycle $i_a j_b$, otherwise the only case left is $S = I \cup \{j_1\}$, which has a cycle $i_1 j_1$.

Now suppose that $g_L(G_k, q) = k$, that is, G is linearly solvable for some q. Then it has a strictly linearly solvable subgraph H such that $k(H) = k$. Then we force $(j_k, j_1) \in H$ because if $(j_k, j_1) \notin H$, then I becomes a feedback vertex set of H of size $k - 1$. Now, by construction, H has no path from j_k to j_1 through I; thus I is a maximum acyclic set of H which is not weakly compatible, thus by Theorem 6 H is not strictly linearly solvable, a contradiction. Thus $g_L(G_k, q) \leq k - 1$. Conversely, G contains a matching of size $k - 1$, namely $\{i_a j_a : 1 \leq a \leq k-1\}$, thus $g_L(G_k, q) = k - 1$ for all q. □

E. Strictly linearly solvable graphs

The reduction of coding functions also allows to construct strictly linearly solvable digraphs. Theorem 6 and its applications to Theorems 7 and 9 already illustrated how to use strictly linearly solvable digraphs as a means to study linearly solvable graphs. Nonetheless, we would like to motivate the study of strictly (linearly) solvable digraphs. Firstly, in the context of (Boolean) coding functions used as models of gene networks, an arc (u, v) in the interaction graph illustrates the fact that the gene u directly influences the gene v: such an influence may not be ignored. Secondly, studying strictly solvable digraphs indicates



(a) Network coding instance

(b) Corresponding graph

Fig. 5: A non-strictly solvable network coding instance.

which arcs must be ignored in order to correctly transmit information by network coding. Indeed, suppose G is solvable but not strictly solvable, then there exists an arc in G which must not be used in any solution of G. Therefore, that arc is not only useless, but it is actually detrimental to network coding. An example is given in Figure 5; the graph is clearly solvable, yet the thick arc makes it non-strictly solvable. Thirdly, by focusing on strictly linearly solvable digraphs, we show in Theorem 10 that a large class of digraphs are linearly solvable.

Theorem 10. *If G has a strongly compatible maximum acyclic set and no loop, then G is strictly linearly solvable.*

Proof. The reader is reminded of the $\mathbb{1}\{\mathcal{P}\}$ notation used in the proof of Theorem 3. Let I be a strongly compatible maximum acyclic set, say $I = \{i_1, \dots, i_m\}$ in topological order. For all vertices $u, v \in V \setminus I$, the number of paths from u to v through I is denoted $N_I(u,v)$. Let q be a prime number greater than $\max_{u,v \in V \setminus I} N_I(u,v)$, and $f \in L(G,q)$ as follows:

$$f_i(x) = \sum_{u \in \mathrm{in}(v)} x_u, \quad \forall i \in I$$

$$f_v(x) = \sum_{i \in \mathrm{in}(v) \cap I} \frac{1}{N_I(v,v)} x_i - \sum_{j \in \mathrm{in}(v) \setminus I} \frac{N_I(j,v)}{N_I(v,v)} x_j \quad \forall v \notin I.$$

We remark that $N_I(v,v) \neq 0$ since $V \setminus I$ is a minimal feedback vertex set and that $N_I(j,v) \neq 0$ since I is strongly compatible. The inverse of $N_I(v,v)$ then exists since q is a prime; since q is larger than any $N_I(j,v)$, we have $N_I(j,v) \neq 0 \mod q$ either. Therefore, $G(f) = G$.

We shall prove that f^{-I} is the identity. For that purpose, we prove the following by induction on

$0 \leq b \leq m$. Let $I_0 = \emptyset$ and $I_b = \{i_1, \dots, i_b\}$, then

$$f_i^{-I_b}(x) = \sum_{u \notin I_b} (N_{I_b}(u,i) + \mathbb{1}\{(u,i) \in G\}) x_u, \quad \forall i \in I \setminus I_b$$

$$f_v^{-I_b}(x) = \sum_{u \notin I_b} \frac{N_{I_b}(u,v)}{N_I(v,v)} x_u + \sum_{i \in \text{in}(v) \cap (I \setminus I_b)} \frac{1}{N_I(v,v)} x_i - \sum_{j \in \text{in}(v) \setminus I} \frac{N_I(j,v)}{N_I(v,v)} x_j \quad \forall v \notin I.$$

This clearly holds for $b = 0$. We have $f^{-I_{b+1}} = f^{-I_b - i_{b+1}}$. Let $i \in I \setminus I_{b+1}$; by induction hypothesis, the x_u term in $f_i^{-I_b}$ is

$$f_i^{-I_b}(x_u) = N_{I_b}(u,i) + \mathbb{1}\{(u,i) \in G\}; \tag{1}$$

the $x_{i_{b+1}}$ term in $f_i^{-I_b}$ is

$$f_i^{-I_b}(x_{i_{b+1}}) = N_{I_b}(i_{b+1}, i) + \mathbb{1}\{(i_{b+1}, i) \in G\} = \mathbb{1}\{(i_{b+1}, i) \in G\};$$

and the x_u term in $f_{i_{b+1}}^{-I_b}$ is

$$f_{i_{b+1}}^{-I_b}(x_u) = N_{I_b}(u, i_{b+1}) + \mathbb{1}\{(u, i_{b+1}) \in G\}.$$

By applying the reduction, we obtain that the x_u term in $f_i^{-I_{b+1}}$ is

$$\begin{aligned} f_i^{-I_{b+1}}(x_u) &= \mathbb{1}\{(u,i) \in G\} + N_{I_b}(u,i) + \mathbb{1}\{(i_{b+1}, i) \in G\} \left(N_{I_b}(u, i_{b+1}) + \mathbb{1}\{(u, i_{b+1}) \in G\} \right) \\ &= \mathbb{1}\{(u,i) \in G\} + N_{I_{b+1}}(u,i). \end{aligned}$$

Now let $v \notin I$: we have two cases to consider for $f_v^{-I_{b+1}}$. First, let $i \in I \setminus I_b$; by induction hypothesis, the x_i term in $f_v^{-I_b}$ is

$$f_v^{-I_b}(x_i) = \frac{1}{N_I(v,v)} \left[N_{I_b}(i,v) + \mathbb{1}\{(i,v) \in G\} \right] = \frac{\mathbb{1}\{(i,v) \in G\}}{N_I(v,v)}, \tag{2}$$

since there is no path from i to v through I_b; and the x_i term in $f_{i_{b+1}}^{-I_b}$ is

$$f_{i_{b+1}}^{-I_b}(x_i) = N_{I_b}(i, i_{b+1}) + \mathbb{1}\{(i, i_{b+1}) \in G\} = 0,$$

for similar reasons. By applying the reduction, we obtain that the x_i term in $f_v^{-I_{b+1}}$ is

$$f_v^{-I_{b+1}}(x_i) = \frac{\mathbb{1}\{(i,v) \in G\}}{N_I(v,v)}.$$

Secondly, let $u \notin I$; by induction hypothesis, the x_u term in $f_v^{-I_b}$ is

$$f_v^{-I_b}(x_u) = \frac{1}{N_I(v,v)} \left[N_{I_b}(u,v) - N_I(u,v) \mathbb{1}\{(u,v) \in G\} \right];$$

the $x_{i_{b+1}}$ term in $f_v^{-I_b}$ is

$$f_v^{-I_b}(x_{i_{b+1}}) = \frac{\mathbb{1}\{(i_{b+1}, v) \in G\}}{N_I(v,v)}$$

(similarly to (2)) and the x_u term in $f_{i_{b+1}}^{-I_b}$ is

$$f_{i_{b+1}}^{-I_b}(x_u) = N_{I_b}(u, i_{b+1}) + \mathbb{1}\{(u, i_{b+1}) \in G\}$$

(similar to (1)). By applying the reduction, we obtain that the x_u term in $f_v^{-I_{b+1}}$ is

$$\begin{aligned} f_v^{-I_{b+1}}(x_u) &= \frac{1}{N_I(v,v)} \left[N_{I_b}(u,v) + \mathbb{1}\{(i_{b+1},v) \in G\} \left(N_{I_b}(u,i_{b+1}) + \mathbb{1}\{(u,i_{b+1}) \in G\}\right) - N_I(u,v)\mathbb{1}\{(u,v) \in G\}\right] \\ &= \frac{N_{I_{b+1}}(u,v) - N_I(u,v)\mathbb{1}\{(u,v) \in G\}}{N_I(v,v)}. \end{aligned}$$

Having proved the claim, we can use it for $b = m$: this yields

$$f_v^{-I}(x) = \sum_{u \notin I} \frac{N_I(u,v) - \mathbb{1}\{(u,v) \in G\} N_I(u,v)}{N_I(v,v)} x_u.$$

The x_v term in $f_v^{-I}(x)$ is then 1 (since G has no loop on v); if $u \in \mathrm{in}(v)$, the term is $(N_I(u,v) - N_I(u,v))/N_I(v,v) = 0$; if $u \notin \mathrm{in}(v)$, we have $N_I(u,v) = 0$ since I is strongly compatible and hence the term in x_u also vanishes. Thus, $f_v^{-I}(x) = x_v$. □

Corollary 5. *For any loopless digraph D, there exists a strictly linearly solvable graph G such that D is an induced subgraph of G.*

Proof. We shall use a construction similar to that in the proof of Proposition 1. Let J be the vertex set of D, then let G be the graph with $G[J] = D$ and such that, for any arc (u,v) of D, G contains $|J|+1$ vertices $(u,v,1), \dots, (u,v,|J|+1)$ and the arcs $(u,(u,v,i))$ and $((u,v,i),v)$ for all $1 \le i \le |J|+1$. Then the vertices outside of J form a strongly compatible maximum acyclic set and G is strictly linearly solvable. □

Corollary 5 indicates that non-solvability is not a local property. One cannot isolate an induced subgraph of a graph and decide that this graph is not solvable.

Note that the converse of Theorem 10 is not true: the complete bipartite graph $K_{2,2}$, illustrated in Figure 6 is strictly linearly solvable but does not have any strongly compatible maximum independent set. The strict solution for $K_{2,2}$ is given, for any odd field, by

$$\begin{aligned} f_1(x_3,x_4) &= \frac{x_3 + x_4}{2}, \\ f_2(x_3,x_4) &= \frac{x_3 - x_4}{2}, \\ f_3(x_1,x_2) &= x_1 + x_2, \\ f_4(x_1,x_2) &= x_1 - x_2. \end{aligned}$$

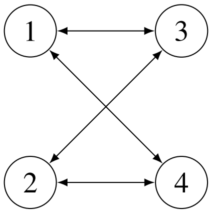


Fig. 6: $K_{2,2}$: a strictly linear solvable graph which is not edge-full.

Notably, the graph G on Figure 5 is a subgraph of $K_{2,2}$. Therefore, the thick arc, which is detrimental in G, becomes useful in $K_{2,2}$.

We generalise this observation to all balanced complete bipartite graphs.

Proposition 9. *The complete bipartite graph $K_{k,k}$ with $k \geq 1$ is strictly linearly solvable over all sufficiently large finite fields.*

Proof. The case $k = 1$ being clear, we assume $k \geq 2$ henceforth. We first prove that for all prime power $q \geq 3k^2$, there exists a $k \times k$ nonsingular matrix $M \in \mathrm{GL}(k, q)$ such that M and M^{-1} have no zero entry. Denote the set of $k \times k$ matrices over $\mathrm{GF}(q)$ with no zero entry as $Z(k, q)$; we then have

$$|Z(k,q)| = (q-1)^{k^2} \geq q^{k^2}\left(1 - \frac{k^2}{q}\right) \geq \frac{2}{3}q^{k^2},$$

while the number of nonsingular matrices is famously lower bounded by

$$|\mathrm{GL}(k,q)| \geq q^{k^2} \prod_{j=1}^{\infty}(1 - q^{-j}) = q^{k^2} \sum_{l \in \mathbb{Z}} (-1)^l q^{-l(3l-1)/2} \geq q^{k^2}(1 - q^{-1} - q^{-2}) \geq \frac{9}{10}q^{k^2},$$

using Euler's pentagonal number theorem. We obtain

$$|\mathrm{GL}(k,q) \cap Z(k,q)| \geq q^{k^2}\left(\frac{9}{10} + \frac{2}{3} - 1\right) > \frac{1}{2}q^{k^2} > \frac{1}{2}|\mathrm{GL}(k,q)|.$$

Hence $|\mathrm{GL}(k,q) \cap Z(k,q)| > |\mathrm{GL}(k,q) \setminus Z(k,q)|$. Thus, inversion cannot be an injection from $\mathrm{GL}(k,q) \cap Z(k,q)$ to $\mathrm{GL}(k,q) \setminus Z(k,q)$ and such a matrix M exists.

Now let $q \geq 3k^2$ and M such that $M, M^{-1} \in Z(k,q)$. Let the vertex set of $K_{k,k}$ be $L \cup R$ and consider the following linear coding function on $K_{k,k}$:

$$f_R(x_L) = x_L M, \quad f_L(x_R) = x_R M^{-1}.$$

Then clearly every vector of the form $(x_L, x_R = x_L M)$ is fixed by f. □

We make a note on undirected graphs with a strongly compatible maximum independent set; according to Theorem 10, they are strictly linearly solvable. We say G is **edge-full** if all its arcs can be covered by exactly $\alpha(G) - i(G)$ cliques, where $i(G)$ is the number of isolated vertices of G. Clearly, if G is edge-full, then it is undirected; a characterisation of edge-full graphs is given in [48] and we shall give another one in Proposition 10 below.

An **intersection model** for an undirected graph G is an ordered pair (S, X), where S is a set and $X = (X_1, \dots, X_n)$ is a collection of n subsets of S such that for all vertices u, v of G, uv is an edge if and only if $X_u \cap X_v \neq \emptyset$. The size of the intersection model is simply the size of S; the minimum size of an intersection model for G is denoted as $\epsilon(G)$. Then $\epsilon(G) \geq \alpha(G) - i(G)$. Indeed, any non-isolated vertex in a maximum independent set needs at least a singleton in the model; all these are disjoint, hence any intersection model must have at least that many elements.

Proposition 10. *Let G be an undirected graph. Then the following are equivalent.*

1) $\epsilon(G) = \alpha(G) - i(G)$.
2) *G is edge-full.*
3) *All maximum independent sets of G are strongly compatible.*
4) *A maximum independent set of G is strongly compatible.*
5) *An independent set of G is strongly compatible.*

Proof. Let U be the set of isolated vertices of G.

2) $\Rightarrow$ 3). Suppose all edges of G are covered by $\alpha(G)$ cliques $c_1, \dots, c_\alpha$, then any maximum independent set I contains one vertex $i_1, \dots, i_\alpha$ per clique; clearly, each i_l belongs to exactly one clique c_l. Suppose u and v are vertices outside of I. If uv is an edge, then it belongs to some clique c_β and hence $ui_\beta, i_\beta v$ are edges in G. Conversely, if uv is not an edge, then u and v cannot belong to a common clique and hence there is no vertex $i \in I$ such that ui, iv are edges. Thus, I is strongly compatible.

Clearly, 3) $\Rightarrow$ 4) $\Rightarrow$ 5).

5) $\Rightarrow$ 2). Let I be a strongly compatible independent set (non necessarily maximum), then $I \setminus U = \{i_1, \dots, i_m\}$ is a strongly compatible independent set too with $m \leq \alpha(G) - i(G)$. The closed neighbourhood of each i_l is then a clique. We claim that these m cliques cover all edges in G. Indeed, there are no edges in I; any edge with one vertex in I is clearly covered by these cliques; finally, for any edge uv outside of I, then there is $i \in I$ such that uv is in the clique corresponding to i. Conversely, it clearly takes at least $\alpha(G)$ cliques to cover all the vertices of G, and hence at least $\alpha(G) - i(G)$ cliques to cover all edges of G. Thus $m = \alpha(G) - i(G)$ and G is edge-full.

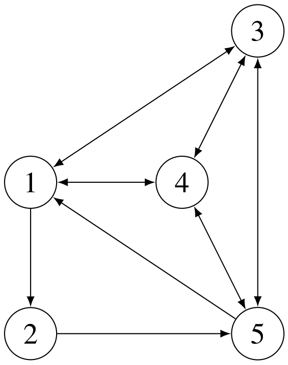


Fig. 7: Example of a non edge-full digraph which is strictly linearly solvable.

1) $\Rightarrow$ 3). If G has a model $(S = \{s_1, \dots, s_\epsilon\}, X)$ of size $\epsilon = \alpha(G) - i(G)$, then if I is a maximum independent set, we must have an enumeration $\{i_1, \dots, i_\epsilon\}$ of $I \setminus U$ such that $X_{i_1} = s_1, \dots, X_{i_\alpha} = s_\epsilon$. Thus, for any $u, v \notin I$, uv is an edge if and only if $s_\beta \in X_u \cap X_v$ for some β, which is equivalent to ui_β and $i_\beta v$ being edges, and I is strongly compatible.

4) $\Rightarrow$ 1). If I is a strongly compatible maximum independent set, then let $S = I \setminus U$ and $X_v = (v \cup \mathrm{in}(v)) \cap S$. □

We give an example of a digraph which is not edge-full and yet is strictly linearly solvable in Figure 7. The set $\{1, 2\}$ is a strongly compatible maximum acyclic set, hence by Theorem 10 the graph is strictly linearly solvable. Since the graph is not undirected, it is not edge-full; moreover, we remark that $\{1, 5\}$ is a maximum acyclic set which is not strongly compatible.

V. Acknowledgment

The authors would like to thank George Mertzios for interesting discussions leading to Proposition 10 and the anonymous referees whose comments improved the paper, notably leading to Lemma 2.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*, ser. Foundation and Trends in Communications and Information Theory. Hanover, MA: now Publishers, 2006, vol. 2, no. 4-5.

[3] R. Dougherty, C. Freiling, and K. Zeger, "Unachievability of network coding capacity," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2365–2372, June 2006.

[4] ——, "Networks, matroids, and non-Shannon information inequalities," *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 1949–1969, June 2007.

[5] S. Riis, "Information flows, graphs and their guessing numbers," *The Electronic Journal of Combinatorics*, vol. 14, pp. 1–17, 2007.

[6] M. Gadouleau and S. Riis, "Graph-theoretical constructions for graph entropy and network coding based communications," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6703–6717, October 2011.

[7] M. Gadouleau, "Closure solvability for network coding and secret sharing," *IEEE Transactions on Information Theory*, no. 12, pp. 7858–7869, December 2013.

[8] ——, "Entropy of closure operators and network coding solvability," *Entropy*, vol. 16, no. 9, pp. 5122–5143, September 2014.

[9] S. Riis. (2007, November) Graph entropy, network coding and guessing games. [Online]. Available: http://arxiv.org/abs/0711.4175

[10] S. A. Kauffman, "Metabolic stability and epigenesis in randomly connected nets," *Journal of Theoretical Biology*, vol. 22, pp. 437–467, 1969.

[11] R. Thomas, "Boolean formalization of genetic control circuits," *Journal of Theoretical Biology*, vol. 42, pp. 563–585, 1973.

[12] R. Thomas and M. Kaufman, "Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior," *Chaos*, vol. 11, no. 1, pp. 170–179, 2001.

[13] H. De Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, pp. 67–103, 2002.

[14] W. S. Mac Culloch and W. S. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Bio. Phys.*, vol. 5, pp. 113–115, 1943.

[15] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sc. U.S.A.*, vol. 79, pp. 2554–2558, 1982.

[16] E. Goles, "Dynamics of positive automata networks," *Theoretical Computer Science*, vol. 41, pp. 19–32, 1985.

[17] S. Poljak and M. Sura, "On periodical behaviour in societies with symmetric influences," *Combinatorica*, vol. 3, pp. 119–121, 1983.

[18] E. Goles and M. Tchuente, "Iterative behaviour of generalized majority functions," *Mathematical Social Sciences*, vol. 4, pp. 197–204, 1983.

[19] R. Thomas and R. D'Ari, *Biological Feedback*. CRC Press, 1990.

[20] E. Goles and S. Martínez, *Neural and automata networks: Dynamical behavior and applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1990.

[21] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature*, vol. 9, pp. 770–780, October 2008.

[22] F. Robert, *Discrete iterations: a metric study*, ser. Series in Computational Mathematics. Springer, 1986, vol. 6.

[23] J. Aracena, J. Demongeot, and E. Goles, "Fixed points and maximal independent sets in AND-OR networks," *Discrete Applied Mathematics*, vol. 138, pp. 277–288, 2004.

[24] E. Remy, P. Ruet, and D. Thieffry, "Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework," *Advances in Applied Mathematics*, vol. 41, pp. 335–350, 2008.

[25] J. Aracena, "Maximum number of fixed points in regulatory Boolean networks," *Bulletin of Mathematical Biology*, vol. 70, pp. 1398–1409, 2008.

[26] A. Richard, “Positive circuits and maximal number of fixed points in discrete dynamical systems,” *Discrete Applied Mathematics*, vol. 157, pp. 3281–3288, 2009.

[27] J. Aracena, A. Richard, and L. Salinas, “Maximum number of fixed points in AND-OR-NOT networks,” *Journal of Computer and System Sciences*, vol. 80, pp. 1175–1190, 2014.

[28] M. Gadouleau, A. Richard, and S. Riis. (2014) Fixed points of Boolean networks, guessing graphs, and coding theory. [Online]. Available: http://arxiv.org/abs/1409.6144

[29] A. Richard, “Fixed point theorems for boolean networks expressed in terms of forbidden subnetworks,” *Theoretical Computer Science*, vol. 583, pp. 1–26, 2015.

[30] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, February 2003.

[31] S. Riis, “Utilising public information in network coding,” in *General Theory of Information Transfer and Combinatorics*, ser. Lecture Notes in Computer Science, vol. 4123/2006. Springer, 2006, pp. 866–897.

[32] D. Christofides and K. Markström, “The guessing number of undirected graphs,” *Electronic Journal of Combinatorics*, vol. 18, no. 1, pp. 1–19, 2011.

[33] S. E. Rouayheb, A. Sprintson, and C. Georghiades, “On the index coding problem and its relation to network coding and matroid theory,” *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3187–3195, July 2010.

[34] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, “Index coding with side information,” *IEEE Transactions on Information Theory*, vol. 57, pp. 1479–1494, 2011.

[35] M. Effros, S. El Rouayheb, and M. Langberg, “An equivalence between network coding and index coding,” in *Proc. IEEE International Symposium on Information Theory*, 2013, pp. 967–971.

[36] A. Mazumdar, “On a duality between recoverable distributed storage and index coding,” in *Proc. IEEE International Symposium on Information Theory*, 2014, pp. 1977–1981.

[37] A. Naldi, E. Remy, D. Thieffry, and C. Chaouiya, “Dynamically consistent reduction of logical regulatory graphs,” *Theoretical Computer Science*, vol. 412, pp. 2207–2218, 2011.

[38] T. Kobayashi, L. Chen, and K. Aihara, “Modeling genetic switches with positive feedback loops,” *Journal of Theoretical Biology*, vol. 221, pp. 379–399, 2003.

[39] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, ser. Springer Monographs in Mathematics. Springer, 2009.

[40] T. Wu, P. J. Cameron, and S. Riis, “On the guessing number of shift graphs,” *Journal of Discrete Algorithms*, vol. 7, pp. 220–226, 2009.

[41] R. Baber, D. Christofides, A. N. Dang, S. Riis, and E. R. Vaughan, “Multiple unicasts, graph guessing games, and non-Shannon inequalities,” in *Proc. NetCod 2013*, 2013.

[42] P. J. Cameron, A. N. Dang, and S. Riis. (2014, October) Guessing games on triangle-free graphs. [Online]. Available: http://arxiv.org/abs/1410.2405

[43] F. Robert, *Les Systèmes Dynamiques Discrets*. Springer, 1995.

[44] R. Dougherty and K. Zeger, “Nonreversibility and equivalent constructions of multiple unicast networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 1287–1291, November 2006.

[45] G. J. Chang, K. Feng, L.-H. Huang, and M. Lu, “The linear guessing number of undirected graphs,” *Linear Algebra and Applications*, vol. 449, pp. 119–131, 2014.

[46] Y. Crama and P. L. Hammer, Eds., *Boolean Functions: Theory, Algorithms, and Applications*. Cambridge University Press, 2011.

[47] J. Bondy and U. Murty, *Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2008, vol. 244.

[48] R. C. Brigham and R. D. Dutton, "On clique covers and independence numbers of graphs," *Discrete Mathematics*, vol. 44, pp. 139–144, 1983.