Eötvös Loránd University
Institute of Mathematics

Erika Renáta Bérczi-Kovács

Summary of the Ph.D. thesis entitled

Network Coding
Algorithms and Applications

Supervisor: András Frank
Professor, Doctor of Sciences

Advisor: Zoltán Király
Doctor of Philosophy

Doctoral School of Mathematics
Director: Miklós Laczkovich, Member of the Hungarian Academy of Sciences

Doctoral Program: Applied Mathematics
Director: György Michaletzky, Professor, Doctor of Sciences

Department of Operations Research, Eötvös Loránd University and
MTA-ELTE Egerváry Research Group on Combinatorial Optimization

Budapest, 2015
1 Introduction

Network coding is a new approach applied to information transmission in a network, enabling intermediate nodes to forward some function, typically a linear combination of the incoming messages. Compared to simple routing, network coding may have advantages, for example increasing throughput, or protection against eavesdropping or failures. From a theoretical viewpoint, network coding can be regarded as a tool for further generalizing min-max packing theorems in combinatorial optimization.

Definition 1. A network is an acyclic directed graph $D = (V, A)$ with a single source node $s \in V$ and a set of terminal/receiver nodes $T \subseteq V - s$. Nodes in $V \setminus (T + s)$ are called internal nodes. For a node $v \in V$, let $\lambda(s, v)$ denote the maximal number of arc-disjoint paths from $s$ to $v$. Assume $k$ messages are to be sent from $s$ to each terminal. Given a network, let $L \subseteq A \times A$ be the set of consecutive pairs of arcs: $L = \{(wu, uv) | w, u, v \in V, wu, uv \in A\}$. For the sake of shortness, members of $L$ are called pairs. A network code is a pair $(\alpha, c)$ referring to local and global coefficient functions, respectively. The local coefficient function of a network code is a function $\alpha : L \to \mathbb{F}_q$. The global coefficient function of a network code is a function $c : A \to \mathbb{F}_k^q$ such that $c(wv) = \sum_{wu \in A} \alpha(wu, uv)c(wu)$ for every arc $wv \in A, u \neq s$. For a network code $(\alpha, c)$, a node $v$ can decode (or receives) message $M_i$, if $e_i \in \langle \{c(wv) | uv \in A\} \rangle$. A network code is feasible for terminal set $T$, if for every terminal node $t \in T$ the dimension of $\langle c(vt) | vt \in A \rangle$ is $k$. Given a positive integer $k$ and a network with digraph $D = (V, A)$, source $s$, terminal set $T$, the network coding problem is to decide whether there exists a feasible network code in the network over some finite field.

Theorem 2 (Jaggi et al. [18]). Assume a network coding problem is given. There exists a feasible network code over some finite field if and only if $\lambda(s, t) \geq k$ for every terminal $t \in T$. If this connectivity criteria holds, then a feasible network code can be given in polynomial time over every finite field of size $q > |T|$.

This thesis considers some applications of network coding, with a focus on possible generalizations of the previous theorem.

2 Multi-layer video streaming

The appearance of new devices (smartphones, tablets, etc.) has highly increased user diversity in communication networks. As a consequence, when broadcasting a video stream, users may have very different quality demands depending on their resolution capabilities. Multi-resolution codes (MRC) are one successful way to handle this diversity, encoding data into a base layer and one or more refinement layers [13, 21]. Receivers can request cumulative layers, and the decoding of a higher layer always requires the correct reception of all lower layers including the base layer. The multi-layer multicast problem is to multicast as many valuable layers to as many receivers as possible. Network coding has been shown to be a successful tool in communication scenarios including multi-layered streaming, increasing
throughput compared to simple routing [19]. Kim et al. gave a network coding scheme called minCut based on restricting the set of layers that may be encoded at certain nodes [19].

**Definition 3.** In multi-resolution coding, for \( i > j \) we say that layer \( i \) is **higher** than layer \( j \), and layer \( j \) is **lower** than layer \( i \). The **height** of a network code on an arc \( uv \) is the highest layer with non-zero coefficient on that arc. For example, the first unit vector has height one and so on, \( e_i \) has height \( i \), and vector \( (1, 0, 1, 0) \) has height 3. The height of \( c \) is denoted by \( h_c : A \to \mathbb{N} \).

The \( i \)th layer is **valuable** for a node only if all lower layers can also be decoded at that node, i.e., for every \( j \leq i \) message \( M_j \) is decodable. The **performance** of a network code at a node \( v \) is the index of the highest valuable layer for \( v \). The performance function of \( c \) is denoted by \( p_c : V \to \{0, 1, \ldots, k\} \), where \( p(v) = 0 \) denotes that layer 1 is not decodable at \( v \).

A **demand** is a sequence of mutually disjoint subsets of \( V - s \) denoted by \( \tau = (T_1, T_2, \ldots, T_k) \). The set of **receiver nodes** is the union of these request sets, denoted by \( T = T_1 \cup T_2 \cup \ldots \cup T_k \).

The nodes in \( T_i \) **request** the first \( i \) layers. Given a demand \( \tau \), we can define a **demand function** \( d_c : V \to \{0, 1, \ldots, k\} \) on the nodes a straightforward way setting \( d_r(v) = i \) if \( v \in T_i \), and \( d_r(v) = 0 \) if \( v \in V \setminus T \).

A network code is **feasible** for demand \( \tau \), if every receiver node \( t \in T_1 \) can decode \( M_j \) for all \( i \) and \( j \leq i \), that is, \( p_c(v) \geq d_r(v) \) for all \( v \in V \). If there exists a feasible network code for a demand, the demand is called **satisfiable**.

We showed NP-hardness of some very special cases of the multi-layered network coding problem.

**Theorem 4** (B-K, Király [5]). Given a directed acyclic graph \( D \) and a demand with three layers \( \tau = (T_1, \emptyset, T_3) \), it is NP-hard to decide whether there exists a satisfying network code for \( \tau \).

**Theorem 5** (B-K, Király [5]). Given a directed acyclic graph \( D \) and a demand \( \tau = (T_1, T_2) \) it is NP-hard to find a maximal cardinality subset \( T'_1 \) of \( T_1 \) so that for \( \tau' = (T'_1, T_2) \) there exists a satisfying network code.

**Corollary 6.** Given a network \( D \), a demand \( \tau = (T_1, T_2) \) and a number \( K \), it is NP-hard to decide whether there exists a network code satisfying at least \( K \) requests.

**Definition 7.** For a function \( f : A \to \{0, 1, \ldots, k\} \), a path \( P \) with arcs \( a_1, a_2, \ldots, a_r \) is called **monotone**, if \( f(a_1) \leq f(a_2) \leq \ldots \leq f(a_r) \). We define for such a monotone path \( \min(P) = f(a_1) \) and \( \max(P) = f(a_r) \). Let a node \( v \in V - s \), a function \( f : A \to \{0, 1, \ldots, k\} \) and a function \( g : V \to \{0, 1, \ldots, k\} \) be given. An **\( i \)-fan** of \( v \) consists of \( i \) pairwise arc-disjoint non-trivial (i.e., containing at least one arc) monotone paths \( P_1, \ldots, P_i \) ending at \( v \), where for all \( j \leq i \) we have \( j \leq \min(P_j) \leq \max(P_j) \leq i \), and \( P_j \) begins at a node \( v_j \) with \( g(v_j) \geq \min(P_j) \).

**Definition 8.** If functions \( f : A \to \{0, 1, \ldots, k\} \) and \( g : V \to \{0, 1, \ldots, k\} \) are given such a way that
i, for every node $v$ with $g(v) > 0$ there exists a $g(v)$-fan of $v$,

ii, for every arc $vw$, either $f(vw) \leq g(v)$, or there exists an incoming arc $uv$ with $f(uv) = f(vw)$,

then $g$ is called a fan-extension of $f$.

**Definition 9.** A function $f : A \to \{0, 1, \ldots, k\}$ is a **height function** if there exists a finite field $\mathbb{F}_q$ and a linear network code $c$ over $\mathbb{F}_q$ with $h_c = f$. Similarly we can define when a function $g : V \to \{0, 1, \ldots, k\}$ is a **performance function**, i.e., if there exists a linear network code $c$ over $\mathbb{F}_q$ with $p_c = g$. We say that functions $f : A \to \{0, 1, \ldots, k\}$ and $g : V \to \{0, 1, \ldots, k\}$ form a **height-performance-pair** if there exists a network code $c$ with $h_c = f$ and $p_c = g$. Given a function $f : A \to \{0, 1, \ldots, k\}$, a function $g : V \to \{0, 1, \ldots, k\}$ is called a realizable extension of $f$, if they form a height-performance-pair. A height function $f$ is feasible for a demand $\tau$ if it has a realizable extension $g$ such that $g \geq d_{\tau}$.

**Theorem 10 (B-K, Király [3]).** A function $f : A \to \{0, 1, \ldots, k\}$ and a fan-extension $g$ form a height-performance-pair, that is, $g$ is a realizable extension.

**Theorem 11 (B-K, Király [3]).** If a function $f : A \to \{0, 1, \ldots, k\}$ has a fan-extension, then it has a unique maximal fan-extension $g^*$ such that $g^*(v) \geq g(v)$ for every fan-extension $g$ of $f$ and every node $v$.

**Theorem 12 (B-K, Király [3]).** The maximal fan-extension of a function $f : A \to \{0, 1, \ldots, k\}$ can be determined algorithmically.

For two layers ($k = 2$), the feasible height functions can be characterized.

A demand is **proper** if $\lambda(s, t_i) \geq i$ for all $i$ and all $t_i \in T_i$. Being a proper demand is a natural necessary condition for a demand to have a feasible network code, however, not always sufficient.

**Theorem 13 (B-K, Király [5]).** A function $f : A \to \{1, 2\}$ is feasible for a proper demand $\tau = (T_1, T_2)$, if and only if for all arcs $uv \in A$, $u \neq s$

1. if $f(uv) = 2$, then $\exists wu \in A : f(wu) = 2$,

2. if $f(uv) = 1$, then either $\exists wu \in A : f(wu) = 1$, or $\lambda(s, u) \geq 2$, and moreover

3. for any receiver $t \in T_1$ with $\lambda(s, t) = 1$, there is a 1-valued arc entering $t$, and

4. for any $t \in T_2$ there is a 2-valued arc entering $t$.

**Corollary 14.** For two layers, function $f : A \to \{1, 2\}$ is feasible for a demand $\tau$ if and only if it has a fan-extension $g$ with $g(v) \geq d_{\tau}$ for all $v$.

We showed that given the condition that all receiver nodes have to be able to decode the first layer, there is a unique maximal set of nodes $X$ in the graph such that demand
\( \tau' = (T \setminus X, X) \) is satisfiable. We gave an algorithm for finding this maximal set, as well as constructing a feasible network code.

For the case of three layers we gave a new network coding algorithm. We proved that the algorithm sends the first layer to every receiver and within this constraint, the unique maximal set of receivers gets at least two layers, while some receivers may get three layers.

We compared our heuristic for three layers with minCut [19]. The comparison shows that on some average of the inputs our new heuristic outperforms the other with a peremptorily chosen objective. Implementations were carried out using LEMON C++ library [12].

We also showed how the fan-extensions can be generalized to determine the expected performance of a family of randomized layered network coding heuristics such as minCut [19].

**Theorem 15** (B-K, Király). For a fixed height bound function \( \ell \) and a finite field \( \mathbb{F}_q \), let \( h_{q,\ell} \) and \( p_{q,\ell} \) denote the random height and performance functions of the algorithm, respectively. There exists a height-performance pair \( (H_\ell, P_\ell) \) on \( D \) such that

- \( \lim_{q \to \infty} \operatorname{Prob}(h_{q,\ell}(uv) = H_\ell(uv)) = 1 \) for all \( uv \in A \),
- \( \lim_{q \to \infty} \operatorname{Prob}(p_{q,\ell}(v) = P_\ell(v)) = 1 \) for all \( v \in V \).

Moreover, \( H_\ell \) and \( P_\ell \) can be determined in polynomial time.

A useful consequence of these results is that the expected performance over large finite fields can be determined without simulations. Another application is a new proof for the performance guarantee of minCut.

### 3 Wireless multi-layer multicast

This section summarizes work carried out with Pedersen, Lucani and Fitzek during a visit at Aalborg University [7]. The main focus was to explore possible applications of multi-layer network coding, presented in the previous section, in a real-world setting. Results presented are more practical than theoretical, compared to other sections.

In a wireless multi-layered multicast setting, receivers with different computational capabilities and demands make use of different types of encoded packets. We present a scheme that splits higher layers into sublayers and sends inter and intra-layer packets with different probabilities. The advantage of this flexibility is that it can increase the coding advantage of users with low-demand by extracting information from inter-layer packets. To the best of our knowledge, the effect of layer sizes on this coding advantage has not been investigated. In [7], we introduce a scheme for wireless multi-layer multicast which takes heterogeneity of users into account. It addresses the problem of finding the trade-off between sending intra-layer packets of the base layer, and inter-layer packets mixing the base layer and one refinement layer.

The overall goals of our work were the following:
• Reduce (and make more deterministic) the time needed for all receivers to get the base layer as well as reducing time needed to recover all desired layers.

• Exploit the inherent, heterogeneous computing capabilities of different devices to improve their overall performance.

• Since we assume the different data packets have the potential to be received at each destination, these destinations should have the ability to use them if needed.

• Provide an explicit trade-off in performance between different types of receivers.

The following descriptions are based on our implementation of the algorithms in the Kodo network coding library [20].

• Encoder: In order to implement the layered encoding, we used a simple scheme requiring only three minimal changes to an existing RLNC encoder. 1) Before encoding a symbol, randomly select a coding layer \( L_m \) according to the layer probabilities \( p_m \), where \( 0 \leq m \leq n \). 2) Generate only non-zero coding coefficients up until the size \( d_m \) of the chosen coding layer. 3) Include the layer index into the encoded symbol allowing the decoder to easily identify which layer was used for the encoding.

• Decoder: In order to implement the proposed scheme we needed to construct a decoder capable of decoding a specific layer \( L_i \) while utilizing \( j \) out of a total of \( n \) layers, where \( i \leq j \leq n \). Similarly to the encoder, this goal was achieved in three stages. 1) Extract the layer index of the incoming symbol. If the layer index is larger than \( j \) discard the symbol. 2) Otherwise pass the symbol to the elimination decoder. The purpose of the elimination decoder is to remove the \( L_j \) contribution in the incoming symbols so that it becomes useful for decoding layer \( L_i \). 3) If the elimination decoder successfully removed the \( L_j \) contribution from the incoming symbol it can be passed to the \( L_i \) decoder for actual decoding. With this structure we are able to deal with all choices of \( L_i \), \( L_j \) and \( L_n \).

• Recoder: Recoding at intermediate nodes without altering the coding structure requires the system to control which sub-layers can be combined for generating a coded packet of a given sub-layer. A simple approach lies in creating a random linear combinations of all coded packets of that sub-layer and sub-layers that have less data packets.

We gave exact values for the expected number of packages users need to receive in order to be able to decode the demanded layer(s). We gave formulas for all the three types of users presented. Calculations can be extended for the general case of \( n \) layers applying similar techniques. Our estimations coincided with experimental results [15].

4 Fixed local coefficients

In this section we present various linear network coding problems with partially predetermined coding coefficients. Two versions of the topic were investigated in the thesis, the first
was the so-called network code completion problem (NCCP) and the second was the problem of fixable pairs.

4.1 Network code completion

The first version of network code completion was introduced by Harvey, Karger and Murota [17]. They reduced both the unicast and multicast cases to matrix completion.

**Definition 16.** For a subset of pairs \( M \subseteq L \), a mapping \( \alpha_0 : M \to \mathbb{F}_q \) is **extendable**, if there exist local coefficients \( \alpha \) of a feasible network code such that \( \alpha = \alpha_0 \) on \( M \). Given a network coding problem with a subset of pairs \( M \subseteq L \) with a mapping \( \alpha_0 : M \to \mathbb{F}_q \), the **network code completion problem** is to decide whether \( \alpha_0 \) is extendable.

**Theorem 17** (Harvey, Karger, Murota [17]). If \( q > |T| \), a mapping is extendable over \( \mathbb{F}_q \) if and only if for every \( t \in T \) it is extendable for the one-element terminal set \( \{t\} \). Such an extension can be found in polynomial time.

One of the first applications of network code completion was calculating the unicast capacity of a deterministic wireless relay network. Avestimehr et al. [9] introduced the model of such networks and gave a min-max characterization for its capacity [10]. Several polynomial time algorithms were given for the unicast version of the deterministic wireless relay network capacity problem. Amaudruz and Fragouli [8] used augmenting paths, and Goemans, Iwata and Zenklusen [16] solved the problem with matroid union or intersection. All of the approaches rely on the layered property of the model. Later, the connection to network coding was discovered, which gave characterization for non-layered networks as well.

The multicast version of the NCCP is equivalent to determining the simultaneous max rank completion of the transfer matrices, so if the field size is greater than the number of terminals, the multicast problem is feasible if and only if all its unicast cases are. We gave a new, algorithmic proof for the above statement, which was also applied for the design of the randomized algorithm for the multicast case in Theorem 18.

**Theorem 18** (B-K, Király[6]). If \( q > |T| \), and there exists a feasible extension over \( \mathbb{F}_q \), then the given randomized algorithm finds one with probability one in polynomial expected running time.

4.2 Fixable pairs problem

The second topic considers network coding problems where values on a subset of the local coding coefficients are fixed to unknown, but non-zero values. The question is when can one guarantee that the given values can be extended to a feasible network code.

**Definition 19.** We say that \( M \subseteq L \) is **fixable** if any nonzero-valued mapping \( \alpha_0 : M \to \mathbb{F}_q - \{0\} \) is extendable. The **fixable pairs problem** is to decide whether a given set \( M \) is fixable or not. For a pair \( \ell = (wu, uv) \in L \), \( wu \) and \( uv \) are the **first** and **second** arcs of the pair, respectively, and \( u \) is the **central node** of the pair. Two pairs \( \ell_1 \) and \( \ell_2 \) are
consecutive if the second arc of \( \ell_1 \) is the first arc of \( \ell_2 \). A path contains a pair, if it contains both of its arcs. For a subset of pairs \( M \subseteq L \), a node is \( M \)-influenced if it is the central node of a pair in \( M \). A set of paths is \( M \)-independent if they are pairwise arc-disjoint and any \( M \)-influenced node is contained by at most one of them.

We gave a sufficient condition for a subset \( M \) of pairs to be fixable and present some applications in heterogenous networks.

**Theorem 20** (B-K, Király [4]). Let \( D = (V, A) \) be an acyclic directed graph, \( T \subseteq V - s \) a terminal set having a feasible network code for \( k \) messages over \( \mathbb{F}_q \) with \( q > |T| \), and let \( M \subseteq L \) be a subset of pairs. If for every terminal \( t \in T \) there exist \( k \) \( M \)-independent paths from \( s \) to \( t \) such that none of the paths contains two consecutive pairs in \( M \), then \( M \) is fixable.

As an application we gave a characterization for the network capacity including the case when some of the nodes are broadcasting. Let a network coding problem be given and suppose that a subset \( W \subseteq V \setminus (T \cup \{ s \}) \) of intermediate nodes can only broadcast messages, that is, such a node sends the same message on each of its outgoing arc. The \( W \)-broadcasting network coding problem is to decide the existence of a network code where every node in \( W \) broadcasts. We say that a set of \( st \)-paths is \( W \)-disjoint if the paths are pairwise arc-disjoint and each node in \( W \) is contained in at most one of the paths. Note that the existence of \( k \) \( W \)-disjoint \( st \)-paths can be checked in polynomial time.

**Theorem 21** (B-K, Király [4]). Given a \( W \)-broadcasting network coding problem with \( q > |T| \), there exists a feasible network code if and only if for every \( t \in T \) there are \( k \) \( W \)-disjoint \( st \)-paths.

The fixable pairs problem can similarly model restrictions on incoming messages, that is, when each node in a subset \( W \) can only receive a fixed nowhere zero linear combination of their incoming messages. These linear combinations can also be handled by fixed local coefficients.

**Theorem 22** (B-K, Király [4]). Let a network coding problem be given with \( q > |T| \) and a subset \( W \subseteq V \) such that every node in \( W \) only receives a fixed nowhere zero linear combination of its incoming messages. There exists a feasible network code if and only if for every \( t \in T \) there are \( k \) \( W \)-disjoint \( st \)-paths.

An important application is when some intermediate nodes only receive the XOR of their incoming messages (we assume that messages are from a finite field of size \( 2^d \) represented by \( d \) bits). The XOR of the incoming messages can be regarded as the sum over the finite field, and for each \( w \in W \), every local coefficient on a pair of the form \((u_0w_1, w_1w_0)\) is fixed to 1.

## 5 Resilient network codes

In this section we present some applications of network coding for efficient failure protection. The notion of failure protection can be defined in several ways. Our goal was to find a
network code that resists a certain number of arc failures, that is, deleting any subset of the arcs of bounded size, the network code remains sufficient on the remaining subgraph without altering the local coefficients.

**Definition 23.** Let \( D = (V, A) \) denote an acyclic digraph with source node \( s \), terminals \( T \subseteq V - s \) and assume that a network code \( c \) over \( \mathbb{F}_q \) is given with local coefficient function \( \alpha \). For a subset of arcs \( H \subseteq A \), we define the network code \( c_H \) resulting from the deletion of all arcs in \( H \), or equivalently, setting all local coefficients to zero on pairs intersecting \( H \). If \( c \) was a sufficient network code for sending \( k \) messages to all terminals, we say that \( c \) protects for failure set \( H \), if \( c_H \) is also sufficient. For an integer \( d \in \mathbb{N} \) a network code is \( d \)-failure-protecting, if \( c \) protects for any failure set of size at most \( d \). Similarly, given a set \( \mathcal{H} \subseteq 2^A \) of possible failure sets, a network code \( c \) is \( \mathcal{H} \)-protecting if it protects for every subset \( H \in \mathcal{H} \).

**Theorem 24** (Harvey, Karger, Murota [17]). There exists an \( \mathcal{H} \)-protecting network code \( c \) if and only if for every \( H \in \mathcal{H} \) there are \( k \) arc-disjoint paths from \( s \) to every terminal after the deletion of arcs in \( H \). Moreover, a protecting network code can be chosen over any field of size \( q > |T||\mathcal{H}| \) in time \( O(|T||\mathcal{H}|(m^3 \log m|L|)) \), where \( L \) denotes the set of consecutive pair of arcs.

**Corollary 25.** There exists a \( d \)-failure protecting code if and only if \( \lambda(s, t) \geq k + d \) for every terminal \( t \). Such a code can be found over any finite field of size at least \( |T|(\binom{m}{k} + \cdots + \binom{m}{0}) \) in time \( O(|T|((\binom{m}{k} + \cdots + \binom{m}{0})(m^3 \log m|L|)) \).

We gave a new upper bound for the sufficient field size for \( d \)-failure protection, which is independent of the size of the network.

**Theorem 26** (B-K [2]). If a \( d \)-failure protecting network code exists, then such a network code can be found over any field of size \( q > |T|(\binom{(k+d)^3}{k} + \cdots + \binom{(k+d)^3}{0}) \).

This theorem giving sufficient lower bounds for the field size can be regarded as analogues of Theorem 2, where any field of size at least \( |T| \) was sufficient. The important property of these bounds is the independence from the size of the digraph.

This lower bound on the field size may be much smaller for large digraphs. The idea of the proof is based on a completely different topic called network encoding complexity, similarly to the techniques used in [14].

In a slightly different version of failure protecting network code construction, instead of a set of terminals a unicast connection is considered, but arcs may have different capacities. Similarly to other network coding problems, some special cases of this problem only require actual coding at the source and terminal nodes.

**Theorem 27** (Babarczi, Tapolcai, Rónyai, Médard [11]). If \( k = 2 \), that is, two messages are sent from \( s \) to \( t \) in a digraph with arc capacities one or two, such that there remains an \( st \)-flow of value two after the failure of any arc, then encoding is only needed at the source node over \( \mathbb{F}_2 \).
Their key idea in a nutshell is to show that the network can be decomposed into three subnetworks such that the failure of any arc leaves at least two of them \(st\)-connected. In [1], with Babarzai, Tapolcai, Pašić, Rónyai, and Médard we examined whether the above decomposition property can be generalized for less special scenarios. Two straightforward generalizations may be to increase the number of data flows or the number of possible link failures. We showed that such flow decomposition property may not exit for these scenarios.

The thesis is based on the following papers


Bibliography


