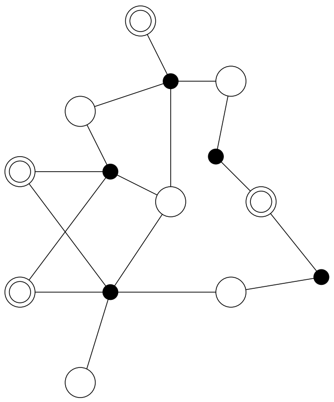
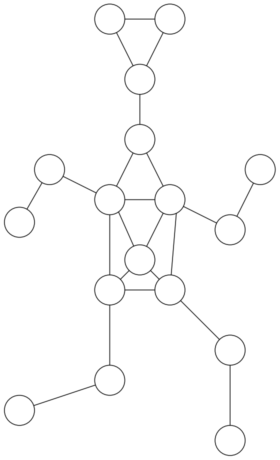
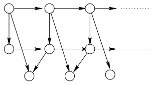
Graphical Models for Linear Systems, Codes, and Networks

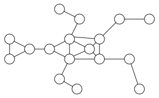
Ralf Koetter
University of Illinois at Urbana-Champaign

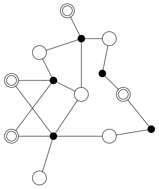




Graphical Models

Bayes Nets: 

Markov Random Fields: 

Factor Graphs: 

Algorithms in Graphical Models

Iterated Conditional Modes

Gibbs sampling

Expectation Maximization

Mean Field

Sum-Product

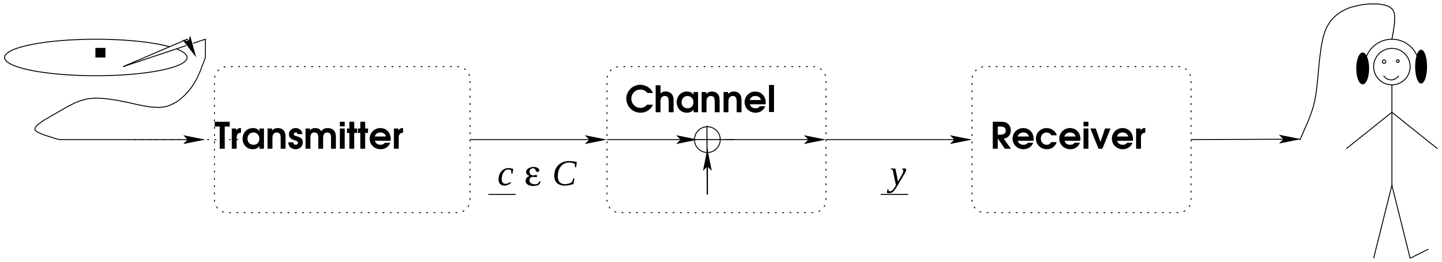
Some Questions

What do the algorithms do ?

How do we find the “best” model?

The problems are very much connected
Non-equivalent representations
Finding a good graph!
Finding a good (small) representation

Graphical Models for Coding and Linear Systems



Main problem:

$$\hat{\underline{c}} = \text{argmax}_{\underline{c} \in \mathcal{C}} \left\{ p(\underline{y} | \underline{c}) \right\}$$

⇒ Choose $\mathcal{C}$, a graphical model for $\mathcal{C}$ and your favorite inference algorithm.

Choices, Choices, ... and some structure

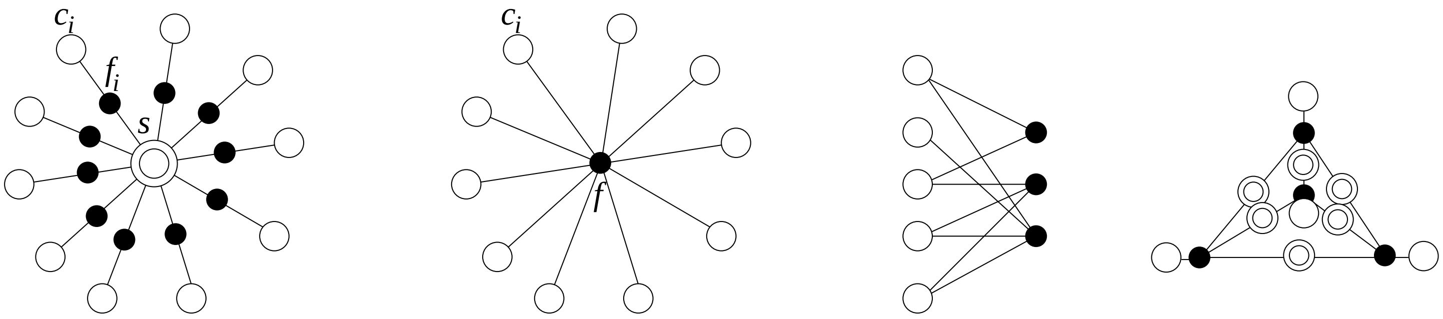
We can choose the time axis freely!

We can give the code a suitable structure!

$\Rightarrow$ We choose the code as a linear space over a finite field $\mathbb{F}$ (think $\mathbb{F}_2$).

Networks and network coding → linear behaviors.

Examples of graphs:



All graphs represent indicator functions $\mathcal{I}_{\mathcal{C}}(\underline{c}) = \begin{cases} 1 & \underline{c} \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}$

Interesting problems

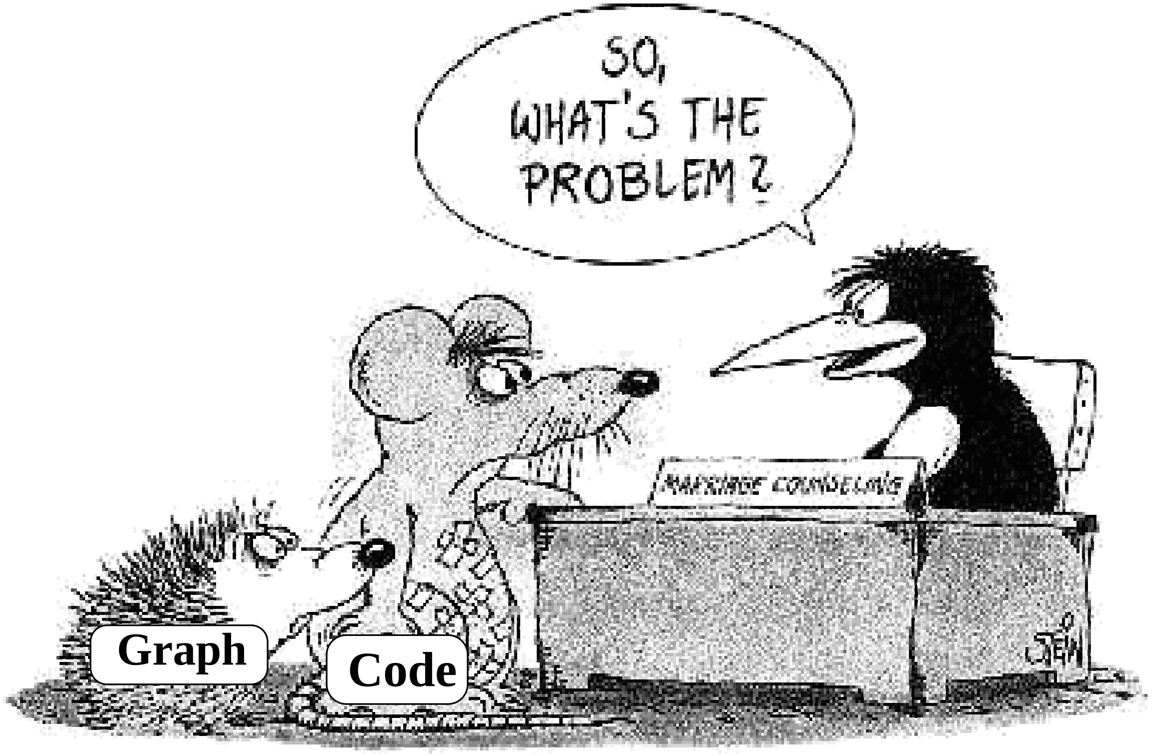
Find the “best” graph (expansion, girth, structure . . .)

Understand the behavior of algorithms (random graphs, threshold effects, graph covers . . .)

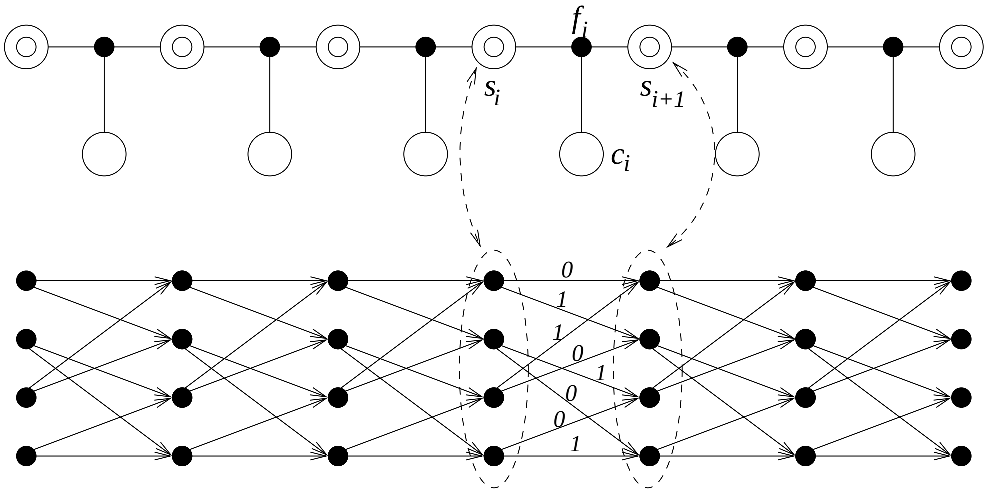
Graphical models for systems (Equalization, timing, estimation . . .)

The main problem in this talk: State space realizations

How to marry a given linear space to a given graph structure



The classical example: Trellises



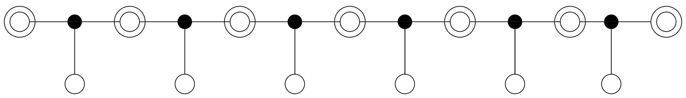
State variables: $s_i \in \mathcal{S}_i$

This is a factor graph for an indicator function $\mathcal{I}_\mathcal{C}$ such that: $\mathcal{I}_\mathcal{C}(\underline{c},\underline{s}) = \prod_{i=0}^{n-1} \mathcal{I}_{\mathcal{C}_i}(s_i, c_i, s_{i+1})$ and $\mathcal{I}_{\bar{\mathcal{C}}}(\underline{c},\underline{s}) = 1 \Rightarrow \mathcal{I}_\mathcal{C}(\underline{c}) = 1$

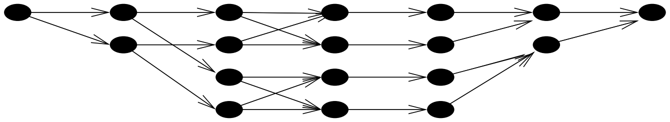
How should we choose the “local” checks $\mathcal{I}_{\mathcal{C}_i}(s_i, c_i, s_{i+1})$ such that $\mathcal{S}_i$ are small?

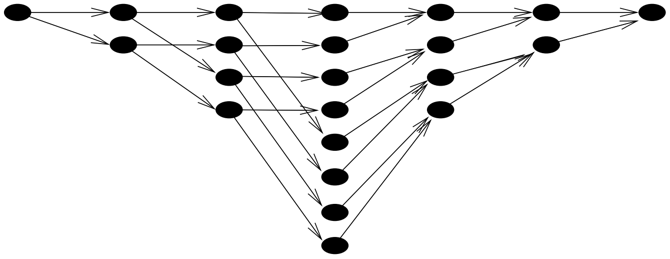
Choices, choices . . .

Example:

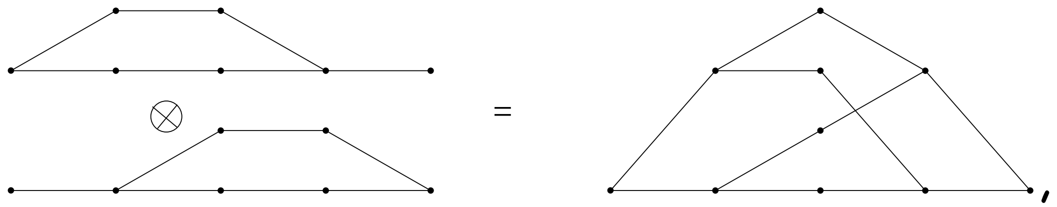


$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \qquad G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$





The Product Construction of Trellises



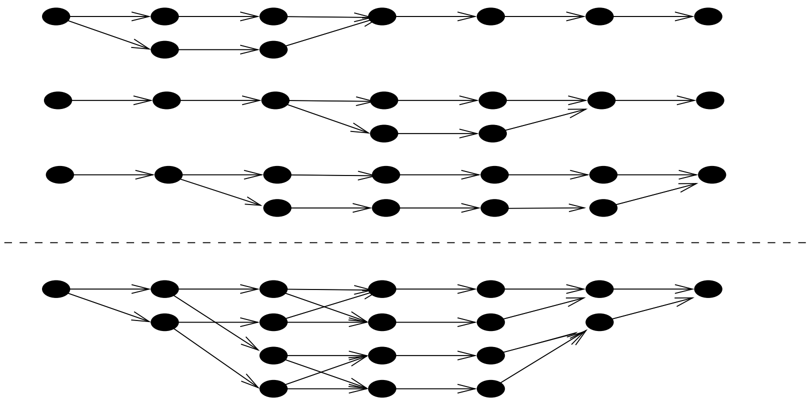
For trellis sections T_i and T_i':

$$T_i \subset \mathcal{S}_i \times \mathbb{F}_2 \times \mathcal{S}_{i+1} \qquad T_i' \subset \mathcal{S}_i' \times \mathbb{F}_2 \times \mathcal{S}_{i+1}'$$

$$T_i \otimes T_i' = \big\{((s_i, s_i'), c_i + c_i', (s_i, s_i')) \subset (\mathcal{S}_i \times \mathcal{S}_i) \times \mathbb{F}_2 \times (\mathcal{S}_{i+1}' \times \mathcal{S}_{i+1}')\big\}$$

Minimal Trellises and the product

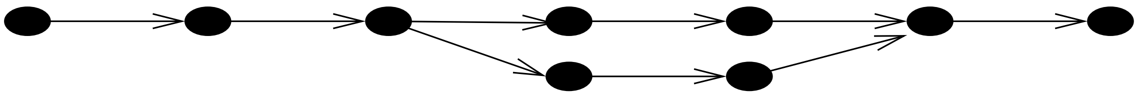
$$G = \begin{pmatrix} 1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1\,0 \\ 0\,1\,0\,1\,0\,1 \end{pmatrix}$$



Each elementary trellis corresponds to a one-dimensional space

A Factorization Theorem

A "prime trellis" is a trellis representing a one-dimesional space.

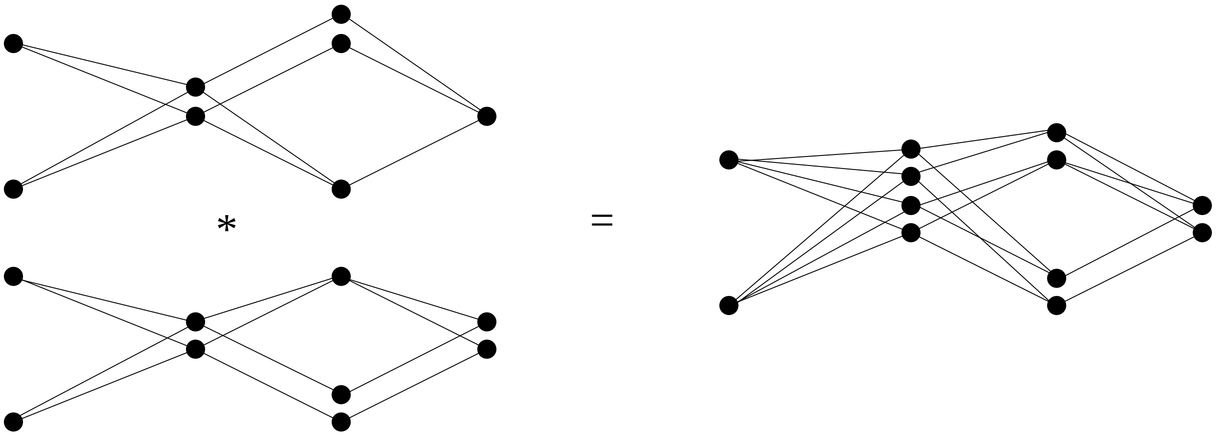


Any linear trellis is obtainable as product of "prime" trellises P_i.

Morover, the trellis product is is commutative and the set of linear trellises constitutes a Unique Factorization Domain: $T = \prod_i P_i^{e_i}$

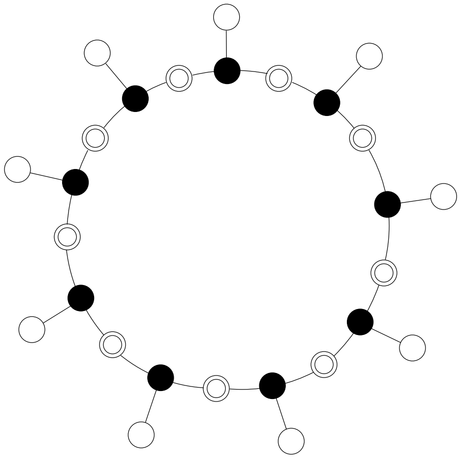
There exists a uniformly smallest "minimal" trellis corresponding to a decomposition of a space into "minimal" one-dimensional spaces!

The product construction for other graphs:

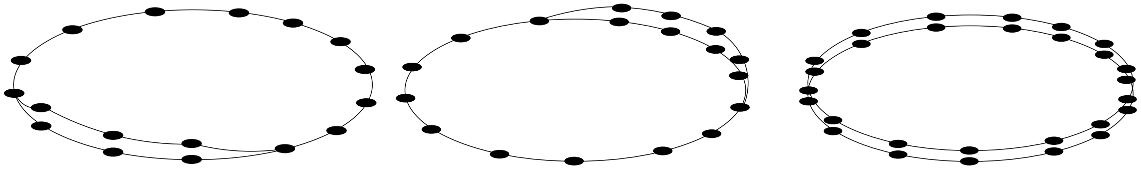


Other graphs?

Tail-biting trellises:



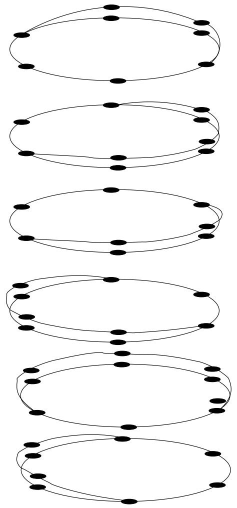
Prime trellises:



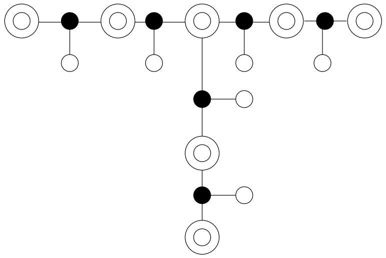
Tail-biting trellises

The factorization theorem can be used to indentify a set $\mathcal{P}$ on at most n prime trellises which contribute to the minimal tail-biting trellis.

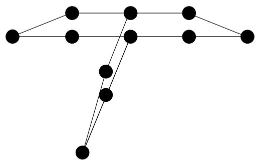
$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

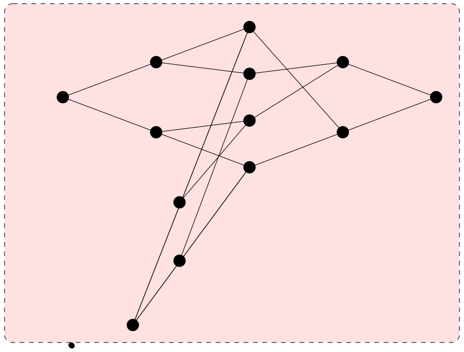


One more case



Primes:





2,3,5,7,11,.....

Any graph comes with a specific set of prime trellises! Characterizing these primes is a central problem for a structure theory of state space realizations.

For trees the set of primes is finite!

For graphs with cycles.... ???????

Duality and a Normalization

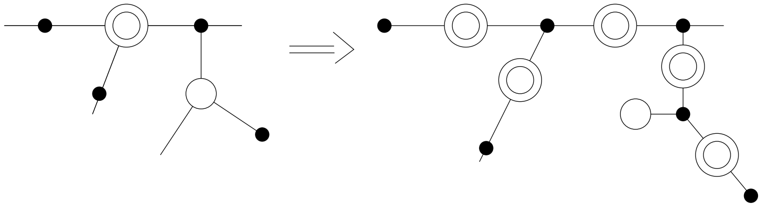
linear space $\mathcal{C}$ and an inner product $\langle \cdot, \cdot \rangle \quad \Rightarrow \mathcal{C}^{\perp}$

From now on:

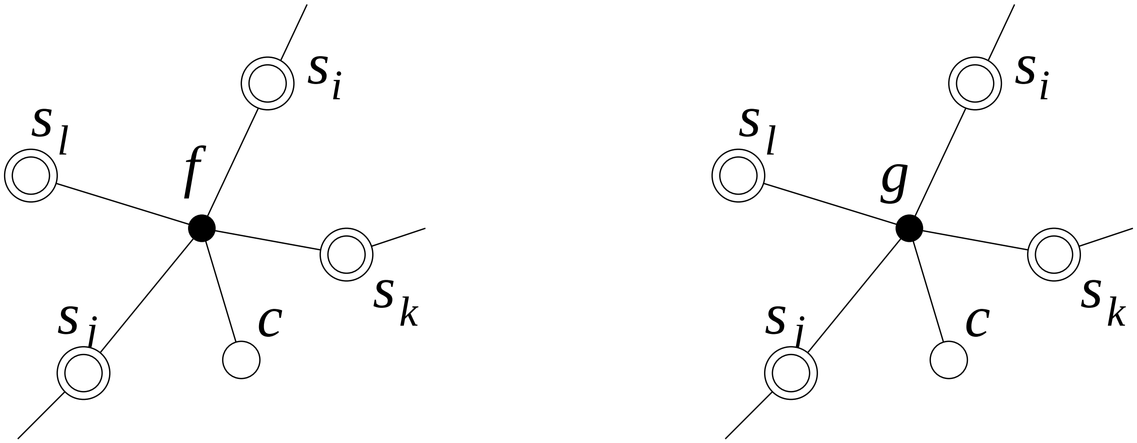
All state variables: degree two

All observed variables: degree one

Transformations:



A canonical dualization



Function f is an indicator function for a linear space V:

$$f(s_i, s_j, s_l, s_k, c) = I_V(s_i, s_j, s_l, s_k, c) = \begin{cases} 1 & (s_i, s_j, s_l, s_k, c) \in V \\ 0 & \text{otherwise} \end{cases}$$

$$g(s_i, s_j, s_l, s_k, c) = I_{V\perp}(s_i, s_j, s_l, s_k, c)$$

Duality and a normalization

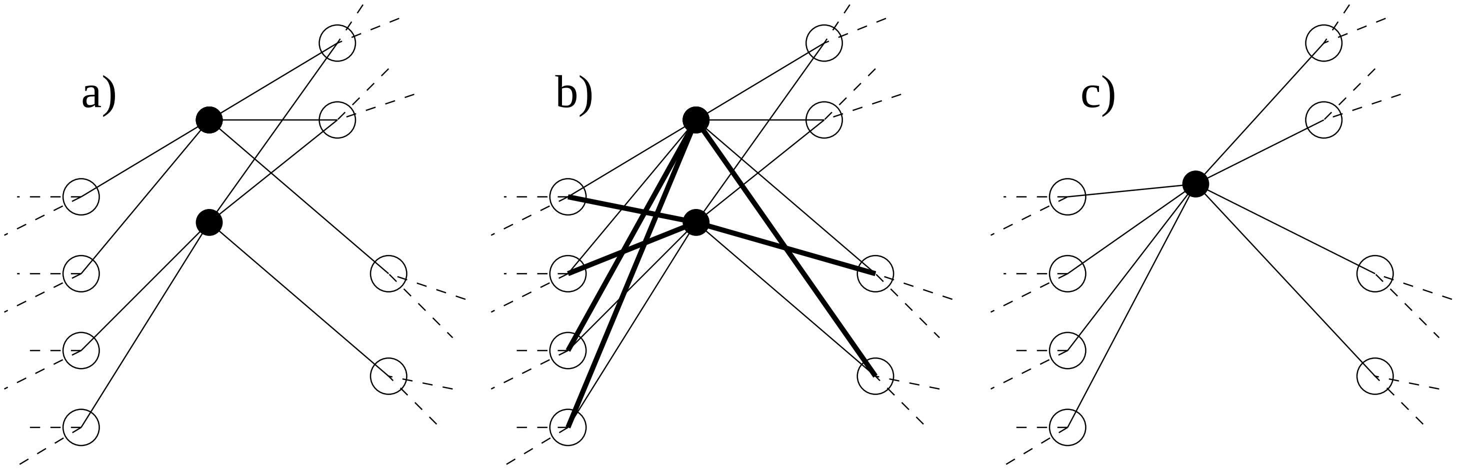
$$\begin{tikzcd}
\boxed{\begin{array}{l} \mathscr{C} = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}}\mathbf{H}(\mathcal{G},\mathfrak{C})^{\mathbf{T}} = \mathbf{0}\} \\ \quad = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}}\mathbf{G}(\mathcal{G},\mathfrak{C}^{\perp})^{\mathbf{T}} = \mathbf{0}\} \end{array}} \arrow[r, "(\cdot)_{|V_O}"] \arrow[d, leftrightarrow, "\perp"] & \boxed{\mathscr{C}_{|V_0} = \bar{\mathscr{C}}_{V_O}} \arrow[d, leftrightarrow, "\perp"] & \boxed{\begin{array}{l} \bar{\mathscr{C}} = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}} = \underline{\mathbf{i}}\mathbf{G}(\mathcal{G},\mathfrak{C})\} \\ \quad = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}} = \underline{\mathbf{i}}\mathbf{H}(\mathcal{G},\mathfrak{C}^{\perp})\} \end{array}} \arrow[l, "(\cdot)_{V_O}"'] \arrow[d, leftrightarrow, "\perp"] \\
\boxed{\begin{array}{l} \mathscr{C}^{\perp} = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}} = \underline{\mathbf{i}}'\mathbf{H}(\mathcal{G},\mathfrak{C})\} \\ \quad = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}} = \underline{\mathbf{i}}'\mathbf{G}(\mathcal{G},\mathfrak{C}^{\perp})\} \end{array}} \arrow[r, "(\cdot)_{V_O}"] & \boxed{(\mathscr{C}^{\perp})_{V_O} = (\bar{\mathscr{C}}^{\perp})_{|V_O}} & \boxed{\begin{array}{l} \bar{\mathscr{C}}^{\perp} = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}}\mathbf{G}(\mathcal{G},\mathfrak{C})^{\mathbf{T}} = \mathbf{0}\} \\ \quad = \{\underline{x} \in \mathbf{F_2^n} : \underline{\mathbf{x}}\mathbf{H}(\mathcal{G},\mathfrak{C}^{\perp})^{\mathbf{T}} = \mathbf{0}\} \end{array}} \arrow[l, "(\cdot)_{|V_O}"']
\end{tikzcd}$$

Duality

Theorem[Forney] If we dualize the individual checks in a state space realization for a linear space V we obtain a state space realization for the dual space $V^{\perp}$.

Minimal realizations

A crucial operation: Merging of states



Minimal state space realizations can be obained by merging non-minimal realizations.

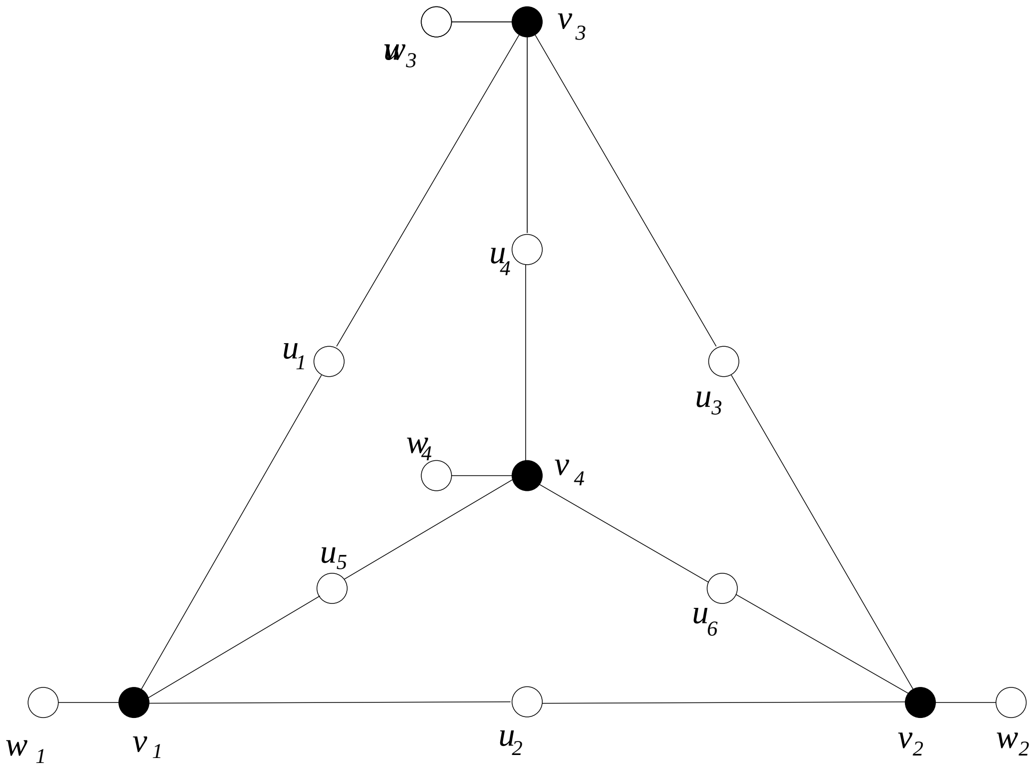
— easy for trellises!

— in general a global operation!

Merging in graphs with cycles

Example:

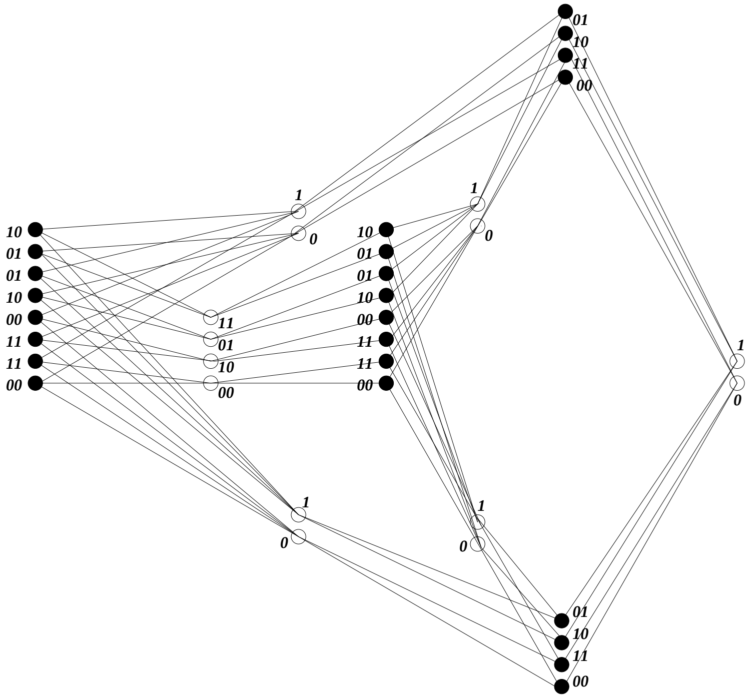
$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$



Merging in graphs with cycles

A global problem.......

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$



Merging in graphs with cycles

Theorem If a given state space realization $\mathscr{G}$ contains a pair of mergeable vertices , then the canonical dual state space $\mathscr{H}$ realization is either disconnected or noton-to-one.

An Algorithm

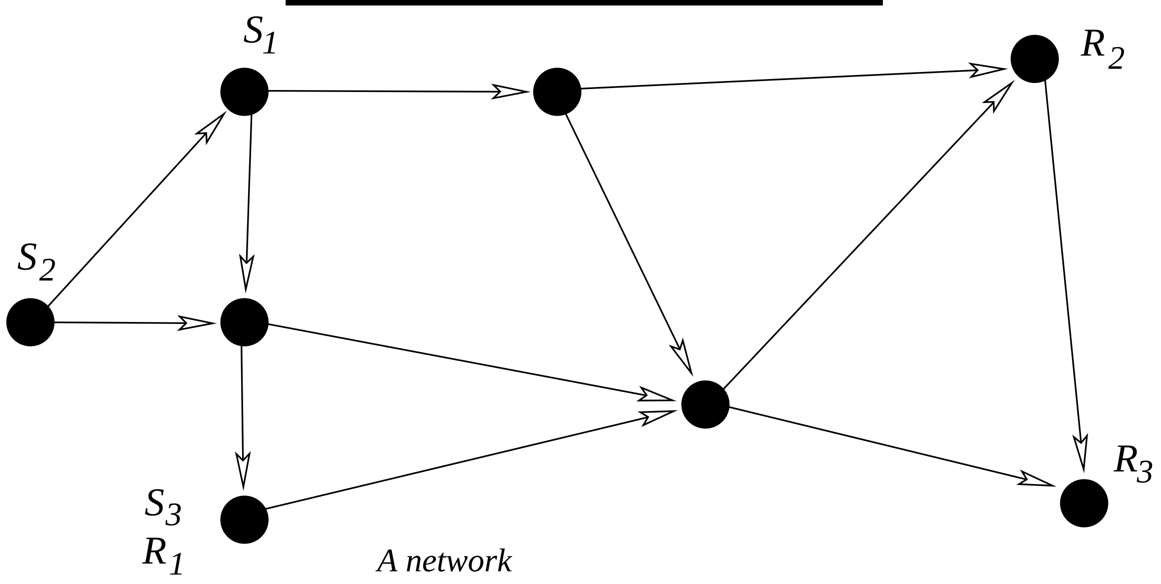
1. Given $\mathscr{G}$ construct $\mathscr{H}$
2. If $\mathscr{G}$ or $\mathscr{H}$ is not one-to-one or is disconnected $\Rightarrow$ merge vertices
3. Reconstruct the canonical dual to the changed $\mathscr{G}$ $\Rightarrow$ GOTO 2.

The running time of this algorithm is polynomial in the number of checks.

Applications

- Coding Theory
- Control Theory
- Linear Systems
- Network Coding

Problem Description A



A network

Vertices: V

Edges: $E \subseteq V \times V$, $e = (v, u) \in E$

Edge capacity: $C(e)$

Network: $\mathcal{G} = (V, E)$

Source nodes: $\{v_1, v_2, \ldots, v_N\} \subseteq V$

Sink nodes: $\{u_1, u_2, \ldots, u_K\} \subseteq V$

Input random processes at v:
$\mathscr{X}(v) = \{X(v,1), X(v,2), \ldots, X(v,\mu(v)\}$

Output random processes at u:
$\mathscr{Z}(u) = \{Z(u,1), Z(u,2), \ldots, Z(u,\nu(u))\}$

Random processes on edges: $Y(e)$

A connection:
$c = (v, u, \mathscr{X}(v,u))$, $\mathscr{X}(v,u) \subseteq \mathscr{X}(v)$

A connection is established if $\mathscr{Z}(u) \supset \mathscr{X}(v,u)$

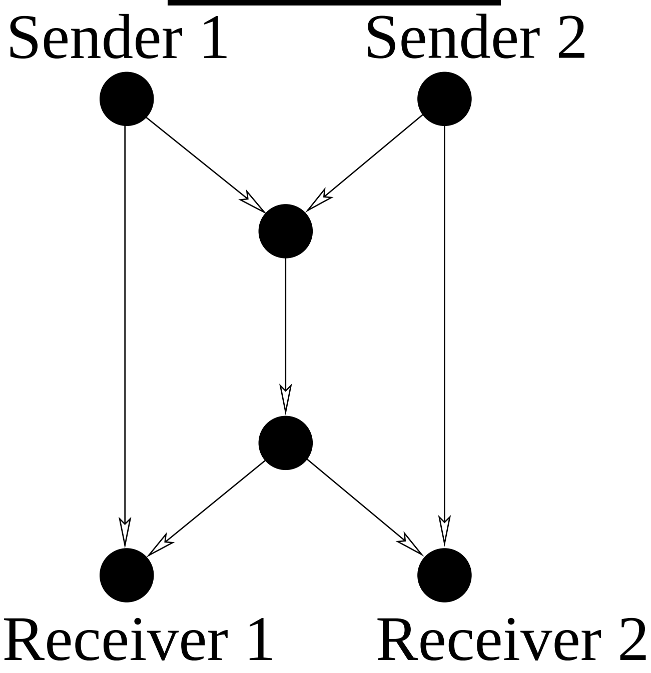
Set of connections: $\mathscr{C}$

The pair $(\mathcal{G}, \mathscr{C})$ defines a network coding problem .

Is the problem $(\mathcal{G}, \mathscr{C})$ solvable?

How do we find a solution?

An Example

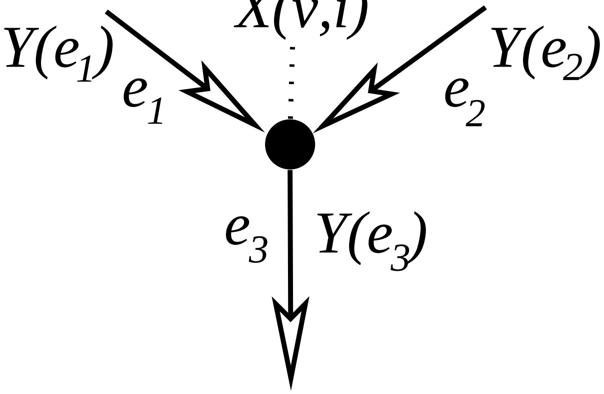


[1] Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network Information Flow”, IEEE-IT, vol. 46, pp. 1204-1216, 2000

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai “Linear Network Coding”, preprint, 2000

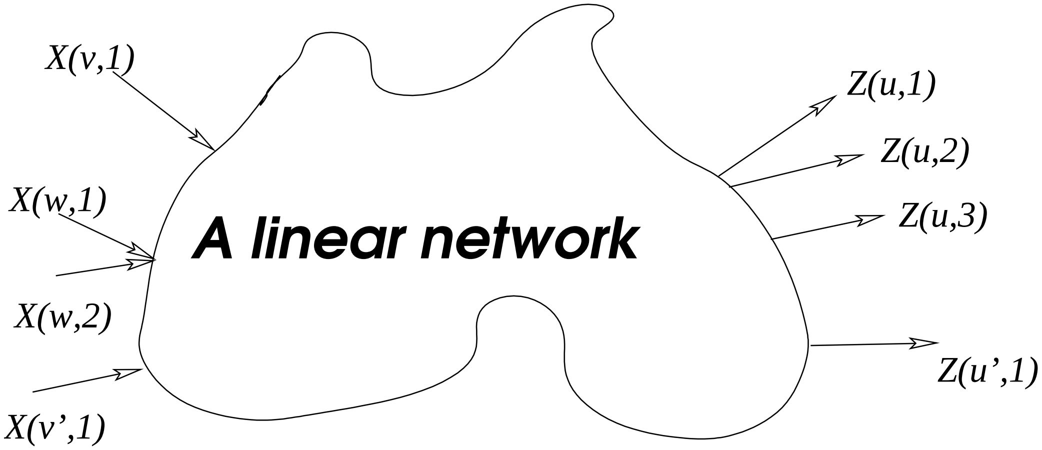
Linear Network Codes

$C(e) = 1$ (links have the same capacity)
$H(X(v,i)) = 1$ (sources have the same rate)
The $X(v,i)$ are mutually independent.
Vector symbols of length m elements in $\mathbb{F}_{2^m}$.



$Y(e_3) = \sum_i \alpha_i X(v,i) + \sum_{j=1,2} \beta_j Y(e_j)$

A linear system



Input vector: $\underline{x} = (X(v,1), X(v,2), \ldots, X(v', \mu(v')))$

Output vector: $\underline{z} = (Z(u,1), Z(u,2), \ldots, Z(u', \nu(u')))$

Transfer matrix: M, $\underline{z} = \underline{x}M$ $\underline{\xi} = (\xi_1, \xi_2, \ldots,) = (\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots$

$$\boxed{M_{i,j} \in \mathbb{F}_2[\underline{\xi}]}$$

An alg. Min-Cut Max-Flow condition

Let a linear network be given. The following three statements are equivalent:

1. A point-to-point connection $c = (v, v', \mathscr{X}(v, v'))$ is possible.
2. The Min-Cut Max-Flow bound is satisfied for a rate $R(c) = |\mathscr{X}(v, v')|$.
3. The determinant of the $R(c) \times R(c)$ transfer matrix M is nonzero over the ring $\mathbb{F}_2[\underline{\xi}]$

3. $\Rightarrow$ We have to study the solution sets of polynomial equations.

Algebraic characterization: A network problem P is solvable if and only if an associated algebraic variety $(V(P))$ is not empty.

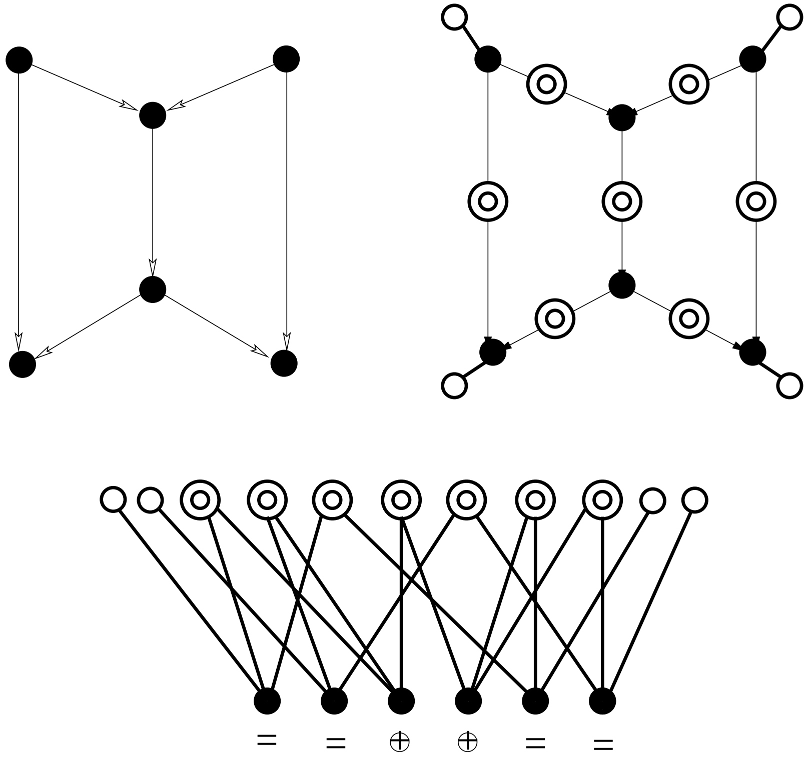
Receiver based recovery: A network problem P can be set up robustly! The collection of local codes does not have to change in order to protect against link failures - provided there exists a solution to the network problem for these link failures.

Network management: Tight bounds can be given for the number of different overall behaviors of the network in order to pretect against various link failure scenarios. (ISIT 02, Ho, Médard, K.)

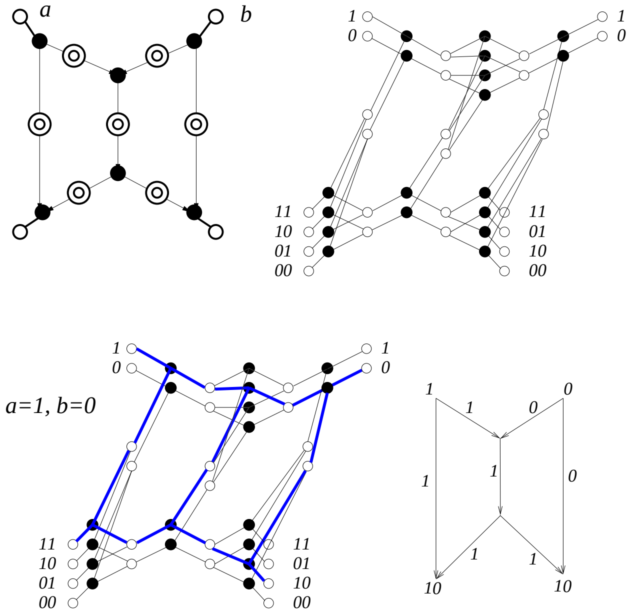
Connections to linear systems, codes......and state space realizations

Graphical models

The network as a linear system:



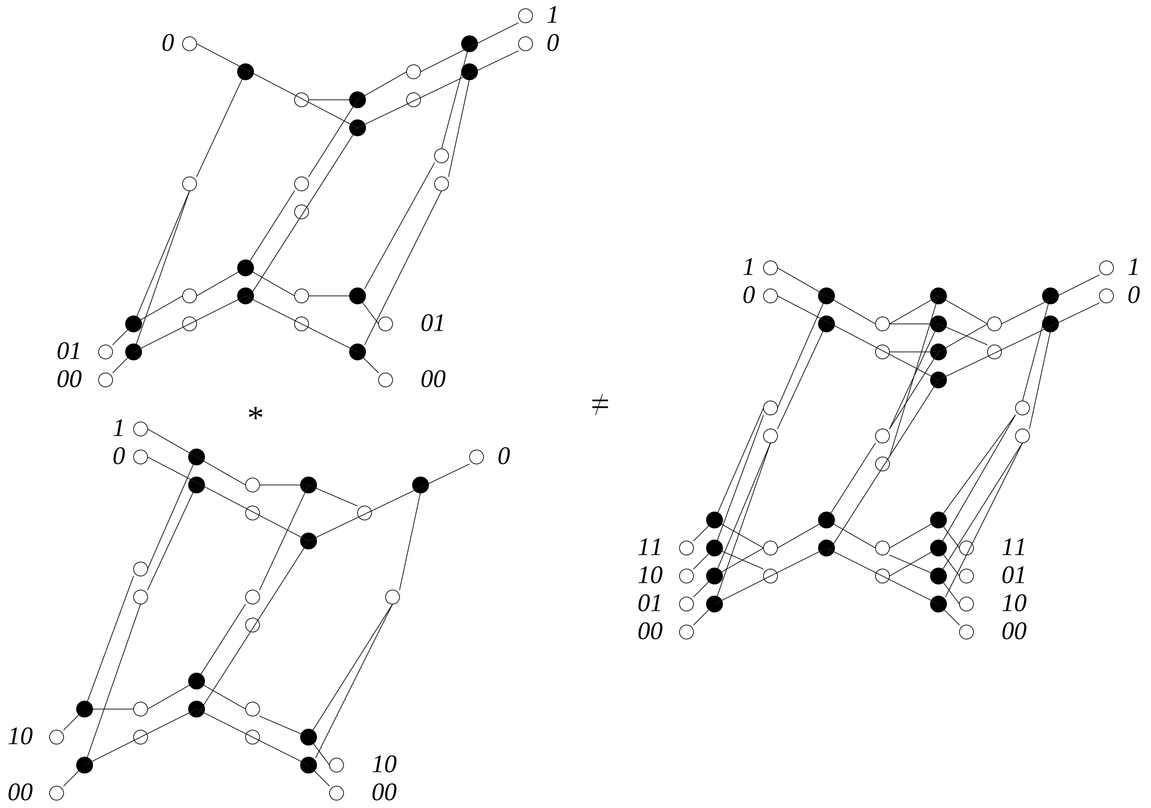
A state space description



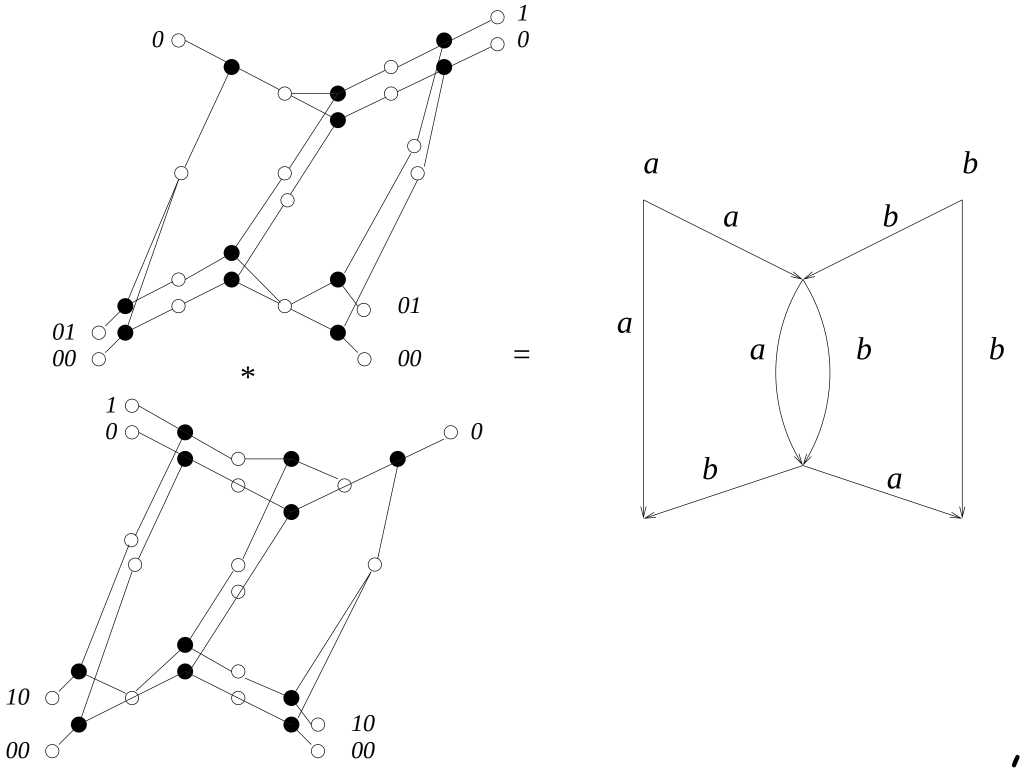
Embedding a code with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & 10 & 10 \\ 0 & 1 & 01 & 01 \end{pmatrix}$$

The product construction

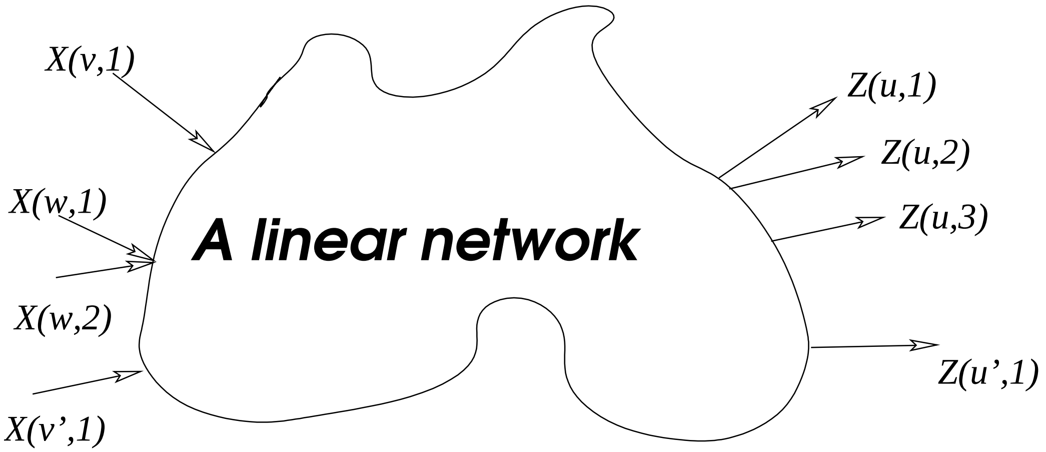


Packet routing and the product construction



The product construction is equivalent to the "routing" solution for the network problem - network coding corresponds to a new "prime".

Finding good networks

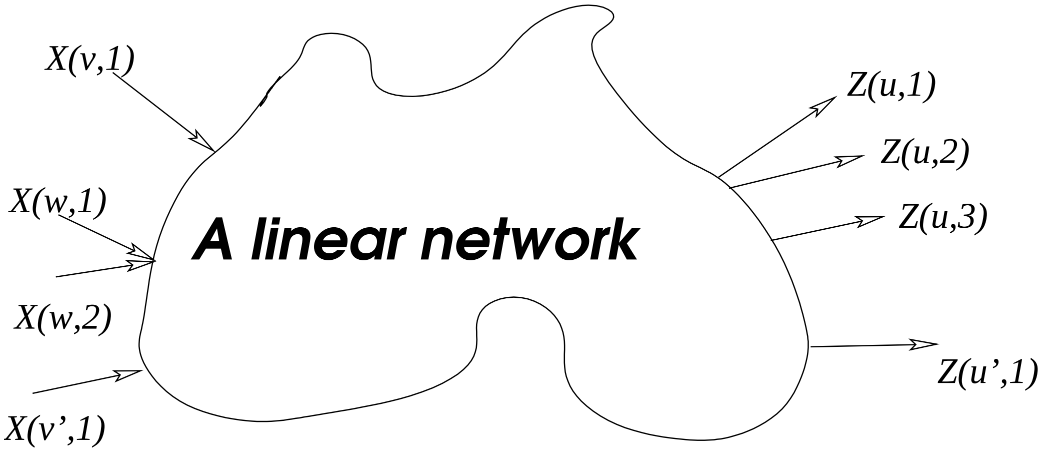


In general hard to find the "best" network.

Finding a non-mergable network a good intermediate solution!

This can be done in polynomial time.

Duality and Reversability

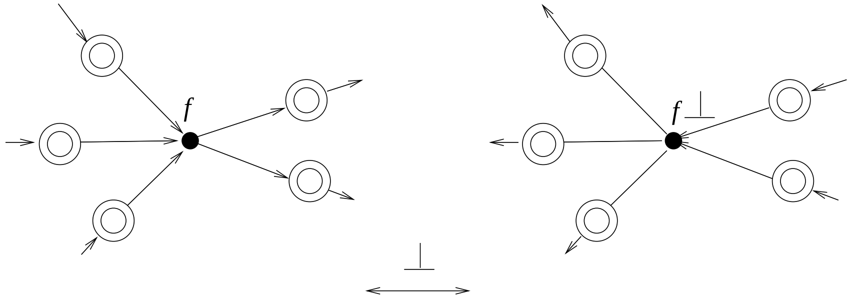


For routing: We can reverse the role of the sinks and the sources......

Network coding: ??????

Duality and Reversability

Theorem If a network can accomodate disjoint connections between a set of sources and sinks it is always possible to reverse the operation of the network to interchange sinks and sources. The network code achieving the reverse operation corresponds to the canonical dual of the state space description of the network code.



The embedded linear space is self dual and generated by:

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Networks for codes - codes for networks

Finding an efficient transmission strategy $\Leftrightarrow$ Finding a graphical model with small state spaces

Routing data streams $\Leftrightarrow$ Product construction of state space realizations.

Network coding is closely related to the theory of linear systems on graphs.

Conclusions

- A structure theory of graphical models for linear spaces is evolving
- Applications in linear systems, coding, control....
- Network coding closely related to graphical models
- This is a big open field with many ramifications ....

...........and a lot of fun!