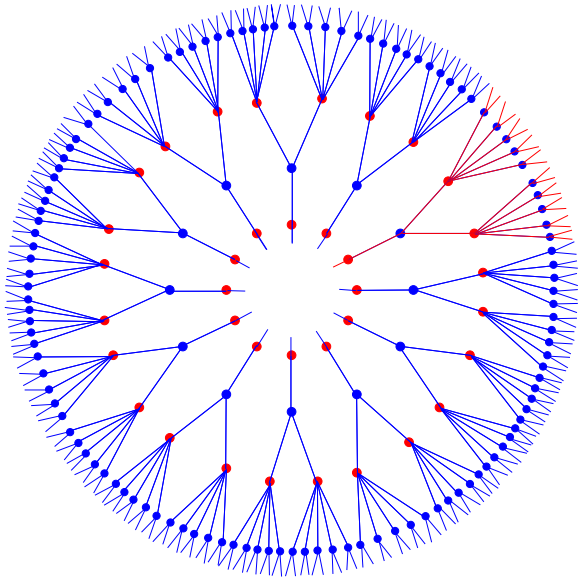
Codes auf bipartiten Graphen



M. Amin Shokrollahi

digitalfountain

Übersicht

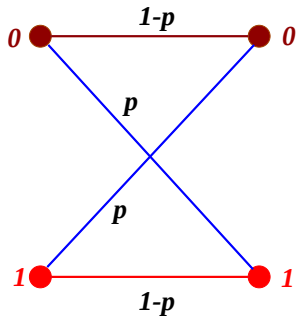
1. Codes auf bipartiten Graphen
2. Iterative Decodierung
3. Analyse
4. Codierung
5. Kapazitätsoptimalität
6. Anwendungen
7. Offene Fragen

Shannons Sätze (1948)

1. Existenz der Kapzität eines Kanals

2. Existenzbeweis kapazitätsoptimaler Codes

3. Unter Maximum likelihood Decodierung.



$$1 + p\log_2(p) + (1-p)\log_2(1-p)$$

Ziel

Wollen

1. Codes, die asymptotisch kapazitätsoptimal sind und

2. die Kapzität mittles effizienter Decodieralgorithmen erreichen und

3. deren Decodierkomplexität nur moderat mit der Nähe zur Kapzität zunimmt.

Low-Density Parity Check Codes

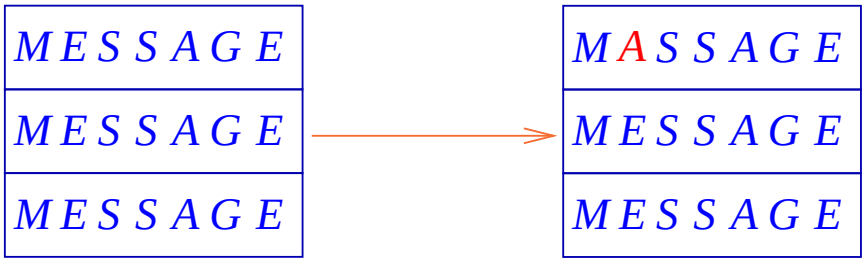
Gallager	1963
Zyablov	1971
Zyablov-Pinsker	1976
Tanner	1981
Turbo Codes	1993
Berroux-Glavieux-Thitimajshima	
Sipser-Spielman, Spielman	1995
MacKay-Neal, MacKay	1995
Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann	1997
Luby-Mitzenmacher-Shokrollahi-Spielman	1998
Richardson-Urbanke	1999
Richardson-Shokrollahi-Urbanke	1999

Codes

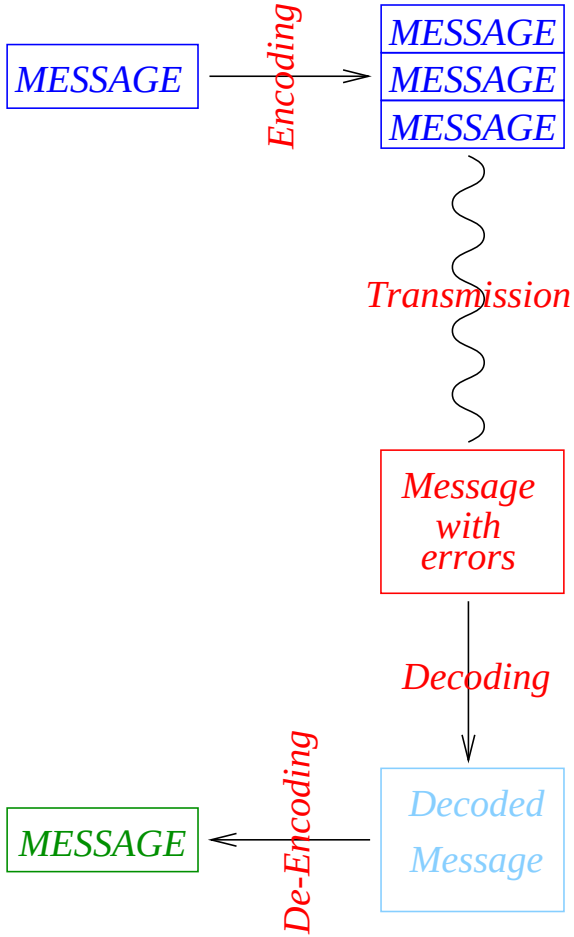
Informationsübertragung auf verrauschten Kommunikationskanälen.

Without coding

With coding



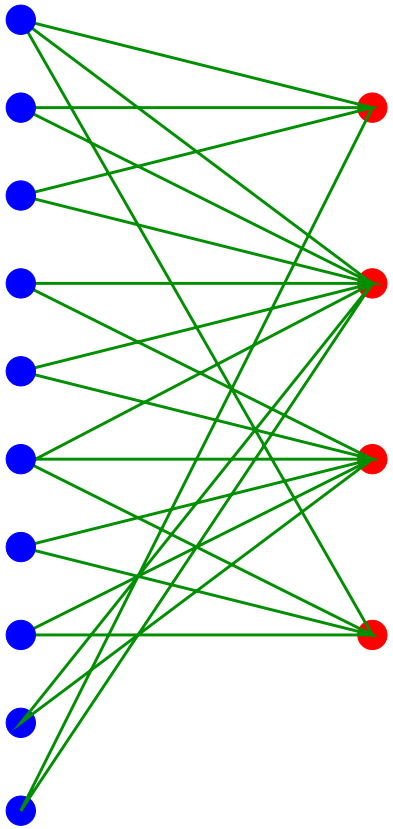
Codierung und Decodierung



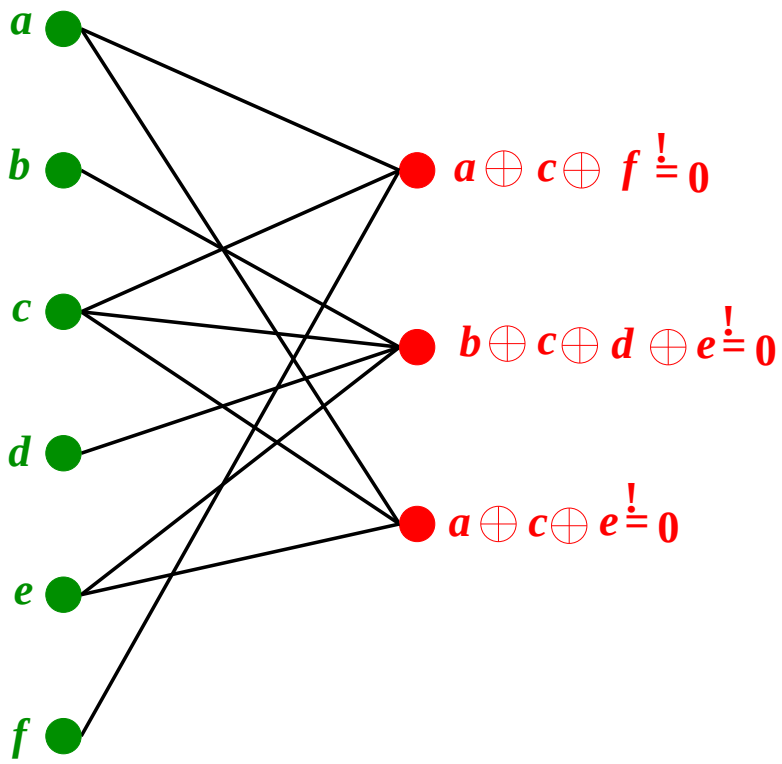
$\text{Codierung}^{-1} \neq \text{Decodierung!}$

LDPC-Codes

LDPC-Codes werden mittels dünner bipartiter Graphen konstruiert.



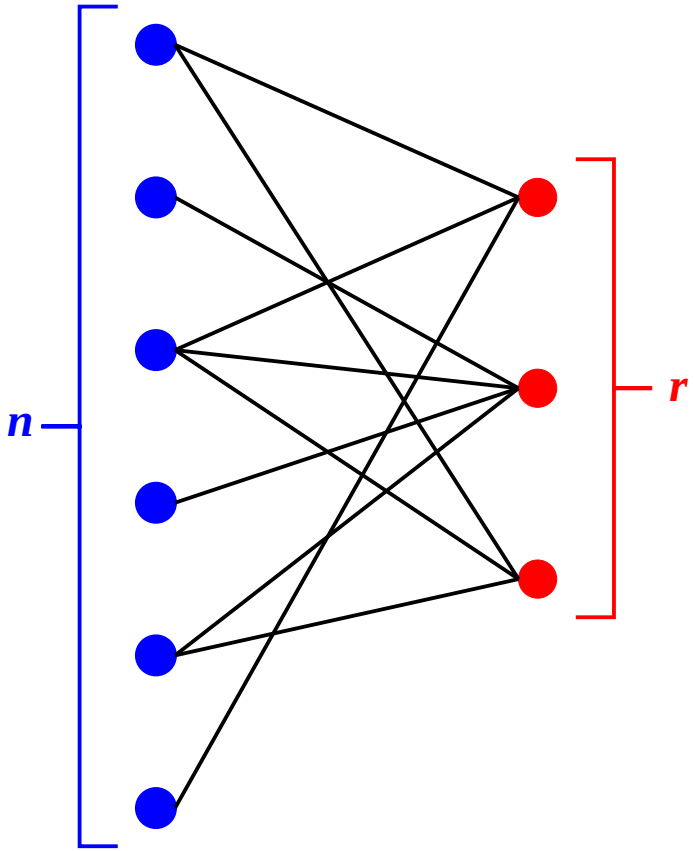
Konstruktion



Jeder binäre lineare Code hat eine graphische Darstellung.

Nicht jeder Code kann durch einen dünnen Graphen dargestellt werden.

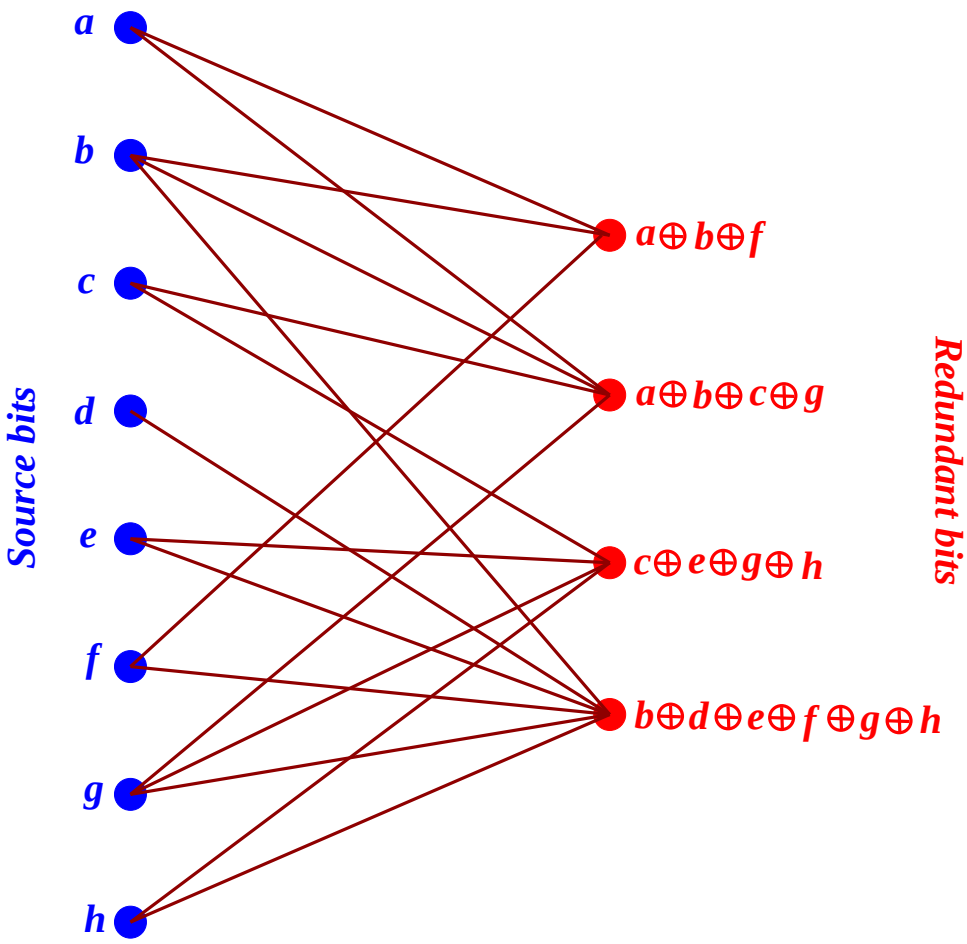
Parameter



$$Rate \geq \frac{n-r}{n}$$

$$Rate \geq 1 - \frac{\text{average left degree}}{\text{average right degree}}$$

Duale Konstruktion



Codierzeit ist proportional zu Anzahl der Kanten.

Konvention

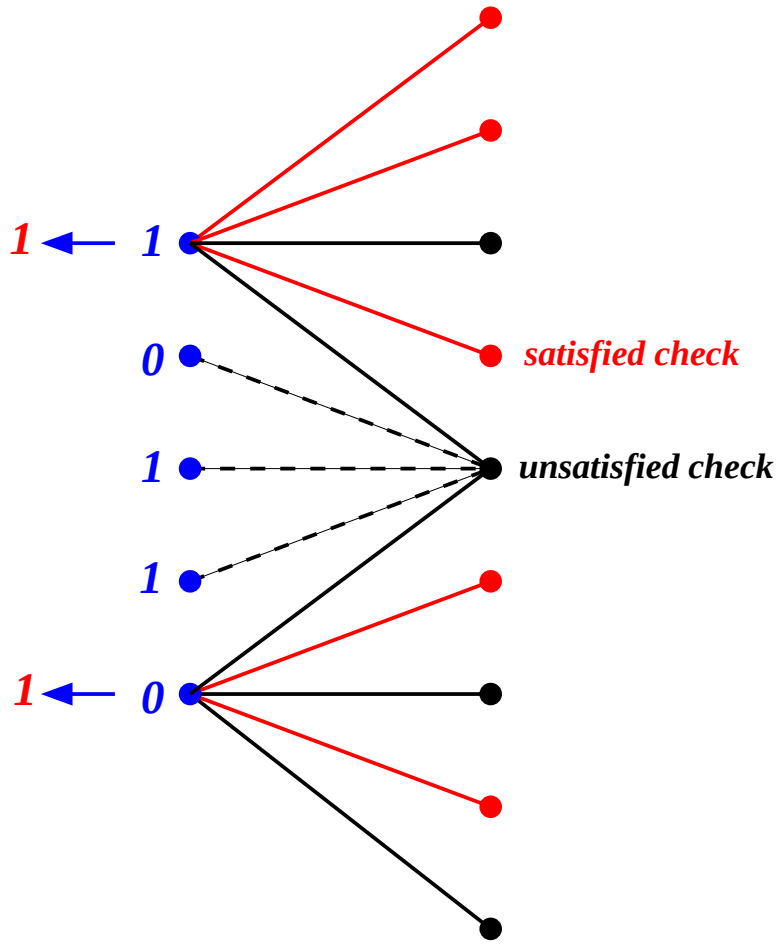
Linke Knoten im Graphen heissen Nachrichtenknoten.

Rechte Knoten im Graphen heissen Check-Knoten.

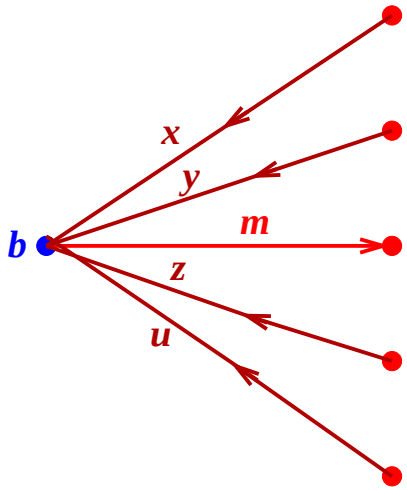
Algorithmische Fragen

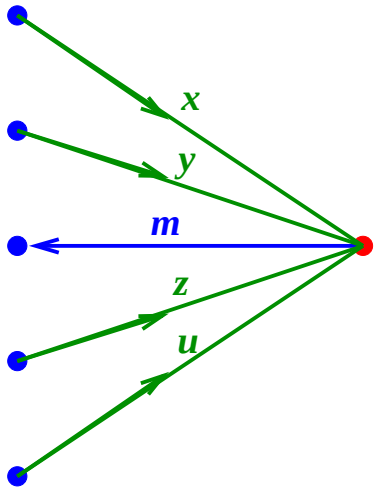
- Codierzeit
  - Ist linear für die duale Konstruktion
  - Ist quadratisch (nach Vorbereitung) für die Gallager Konstruktion. Mehr dazu später!
- Decodierung?
  - Hängt vom Kanal ab.
  - Hängt von der Anzahl der zu korrigierenden Fehler ab.

Decodieren auf dem BSC: Flipping



Decodieren auf dem BSC: Gallager Algorithm A (Message passing)



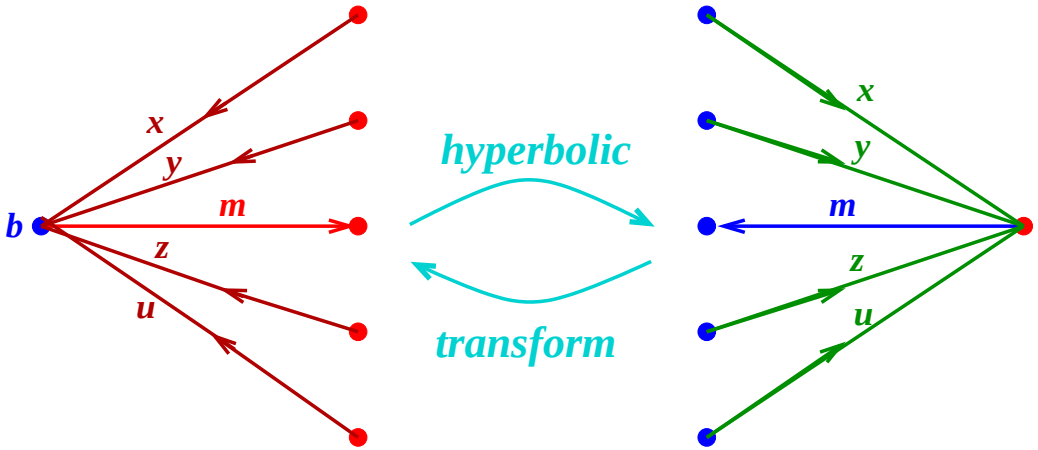


$$m = \begin{cases} x & \text{if } x=y=z=u \\ b & \text{else} \end{cases}$$

$$m = x \oplus y \oplus z \oplus u$$

MESSAGE **CHECK**

Decodieren auf dem BSC: Belief Propagation



$m = x+y+z+u+b$

$m = x * y * z * u$

$(a,b) * (c,d) := (a+c,\ b+d \bmod 2)$

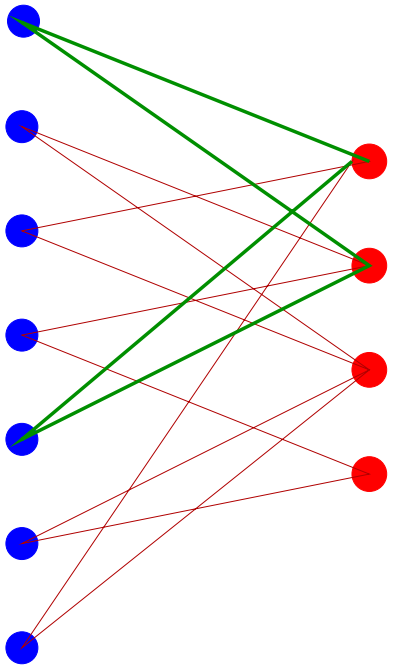
MESSAGE

CHECK

Die Nachrichten sind in Form von log-likelihood ratios.

Optimalität von Belief Propagation

Belief propagation ist bit-optimal, falls der Graph keine Zyklen hat.



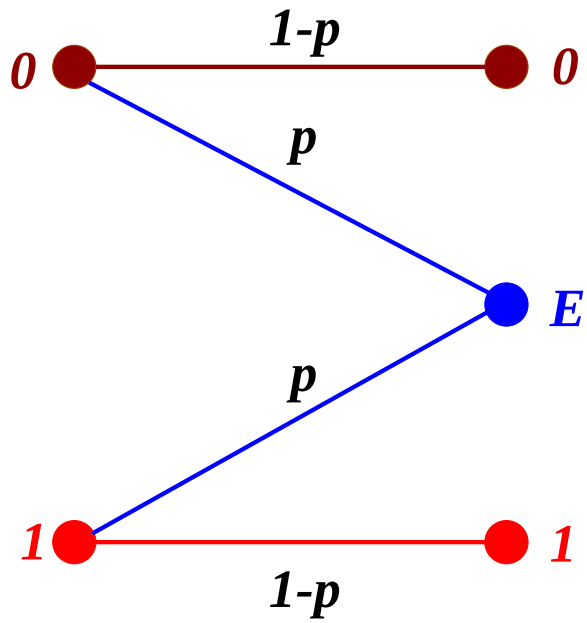
Maximiert die Wahrscheinlichkeit

$$P(c_m = b \mid y) = \sum_{c \in C} P(c \mid y).$$

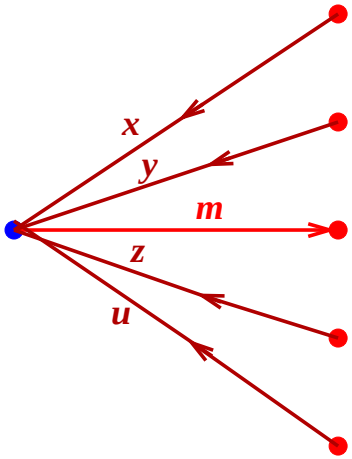
Qualität von (3,6)-Graphen

Shannon Schranke:	11%	
Flipping algorithm:	1%?	
Gallager A:	4%	
Gallager B:	4%	(6.27%)
Erasure decoder:	7%	
Belief propagation:	8.7%	(10.8%)

Der binäre Auslöschungskanal (BEC)

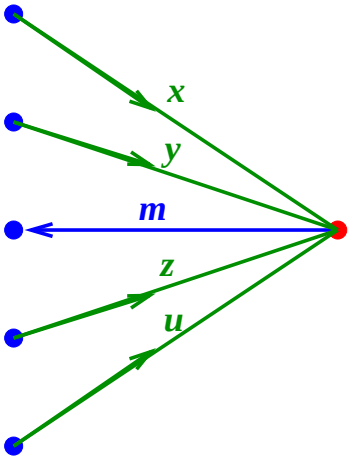


Decodieren auf dem BEC: Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann



$$m = \begin{cases} 1 & \text{if } x \vee y \vee z \vee u = 1 \\ 0 & \text{else} \end{cases}$$

MESSAGE

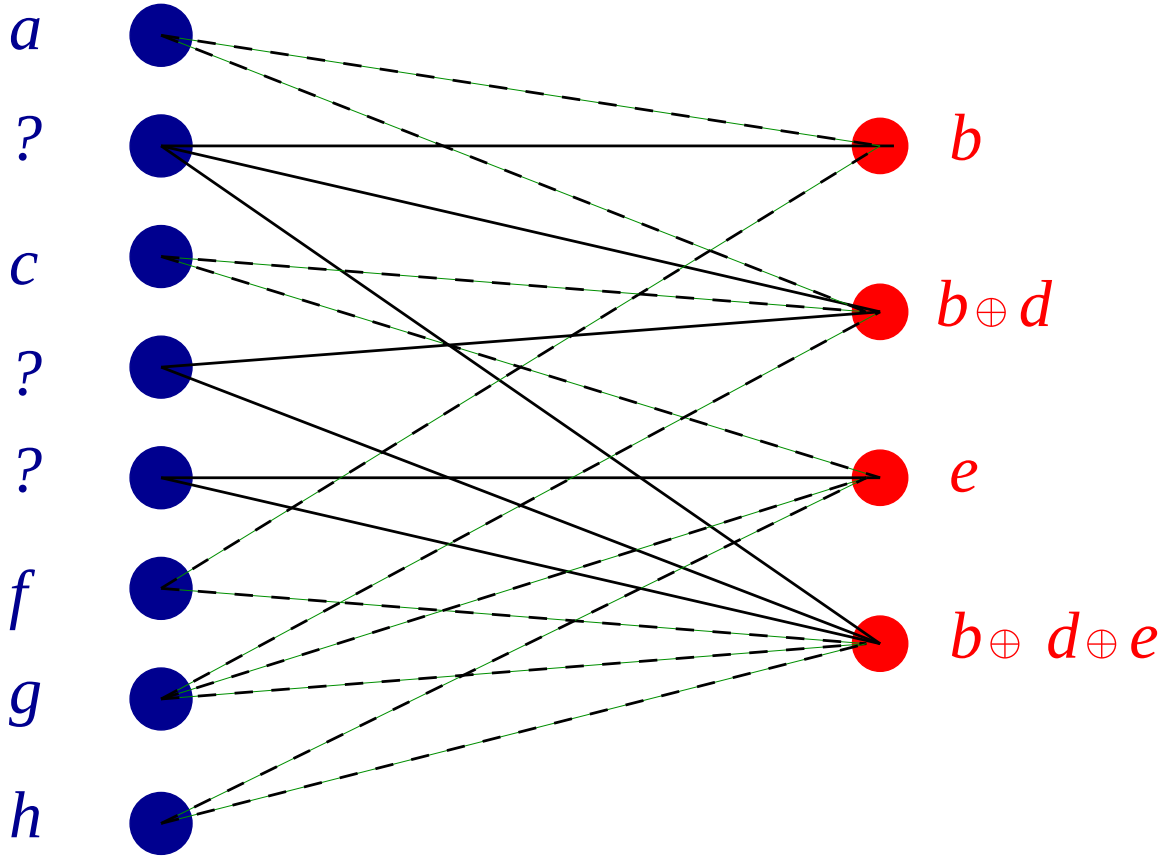


$$m = \begin{cases} 1 & \text{if } x = y = z = u = 1 \\ 0 & \text{else} \end{cases}$$

CHECK

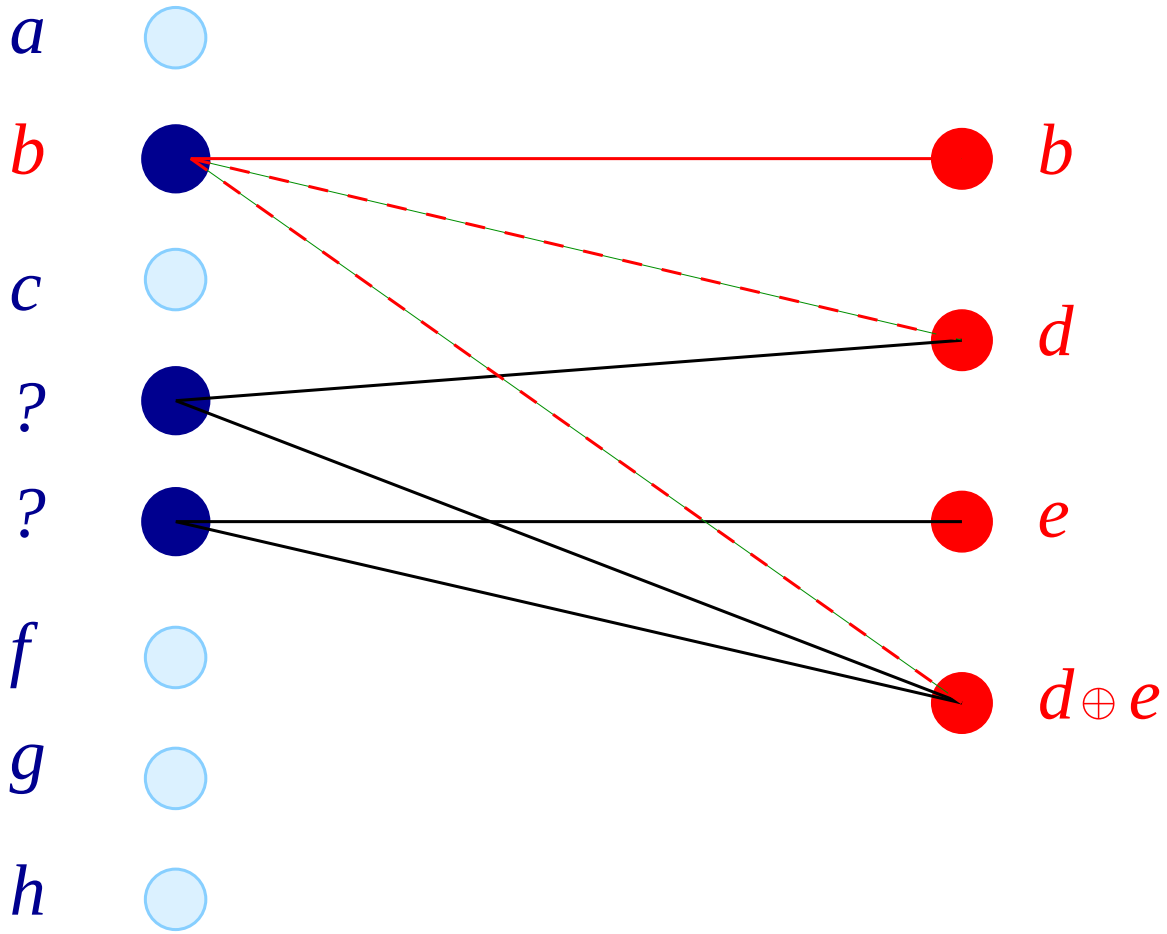
Decodieren auf dem BEC

Phase 1: Direkte Brechnung

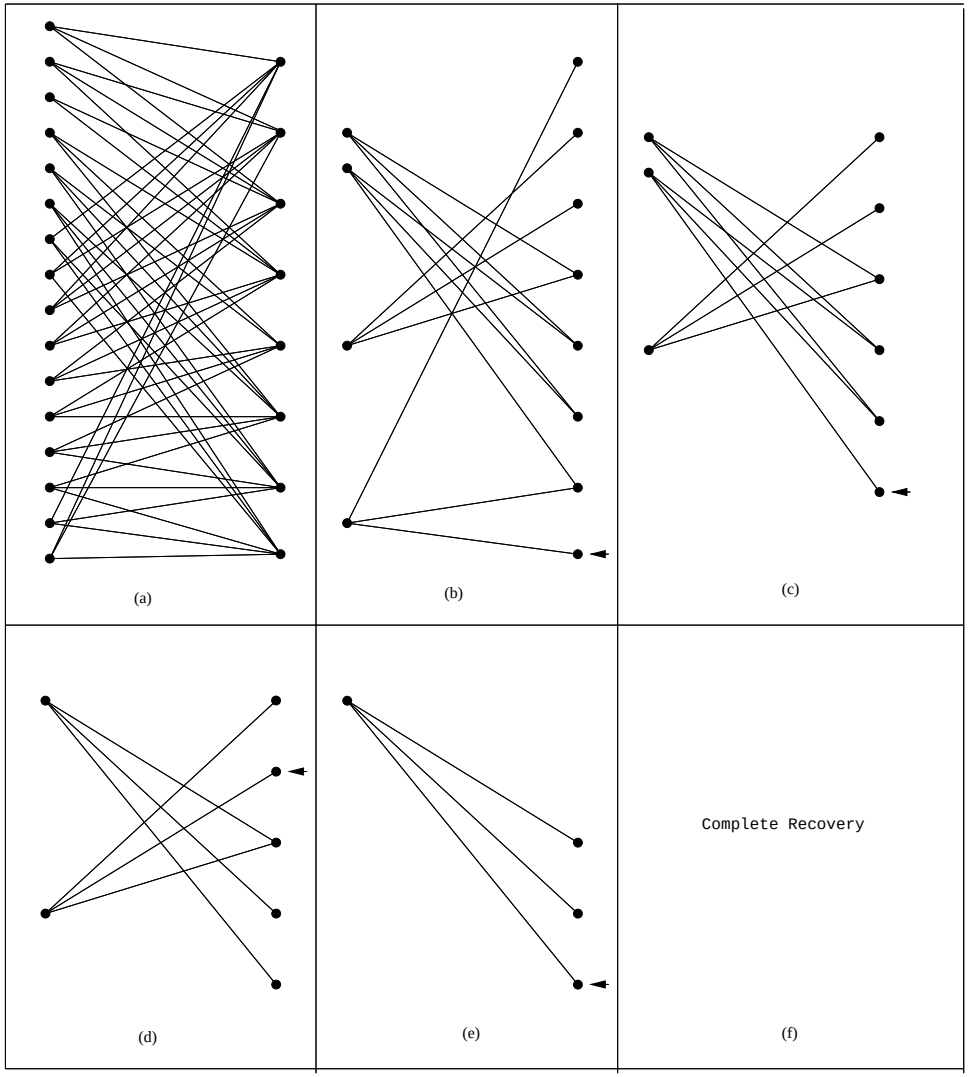


Decodieren auf dem BEC

Phase 2: Substitution



Beispiel



Das (inverse) Problem

Haben: schnellen Decodieralgorithmus.

Wollen: Codes entwerfen die viele Fehler mittels dieser Algorithmen korrigieren können.

Betrachten im folgenden den BEC.

Experimente

Wähle reguläre Graphen.

Ein (d, k)-regulärer Graph hat mindestens Rate $1 - d/k$. Kann höchstens einen d/k-Anteil von Auslöschungen korrigieren.

Wähle zufälligen (d, k)-Graphen.

p_0 := Maximaler Anteil von korrigierbaren Auslöschungen.

d	k	d/k	p_0
3	6	0.5	0.429
4	8	0.5	0.383
5	10	0.5	0.341
3	9	0.33	0.282
4	12	0.33	0.2572

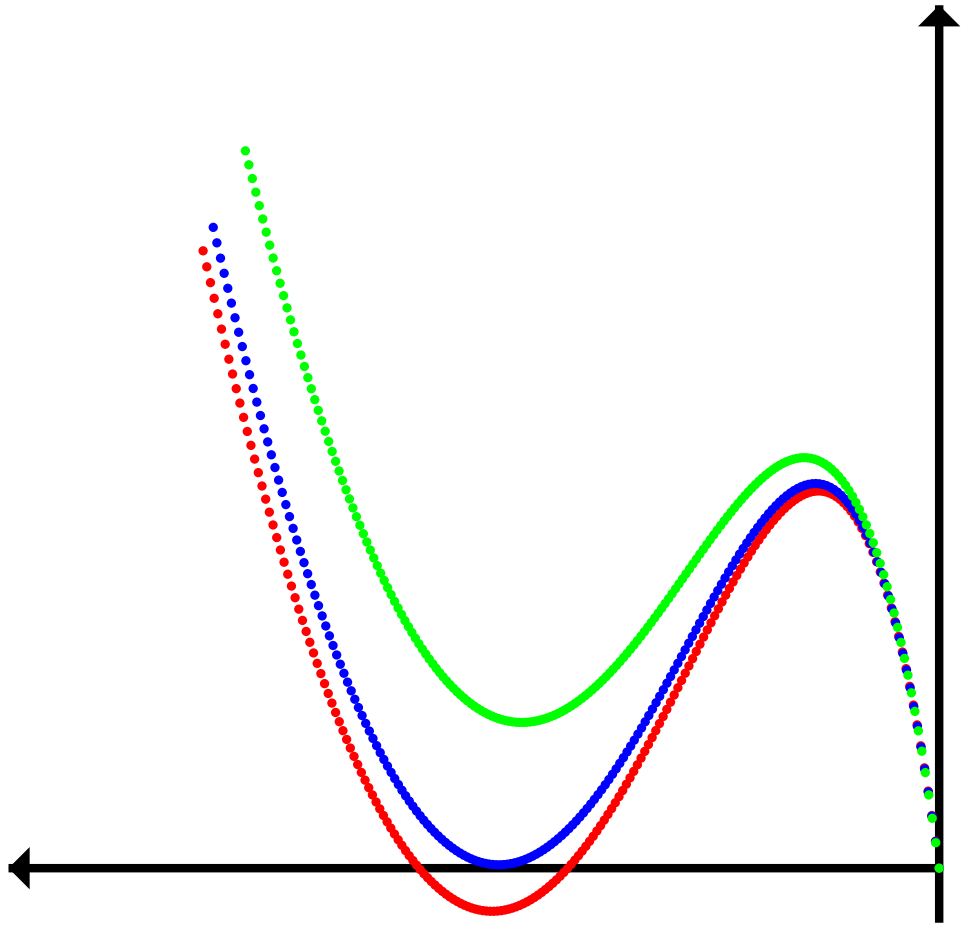
Wo kommen die Zahlen her?

Das Theorem

Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann, 1997:

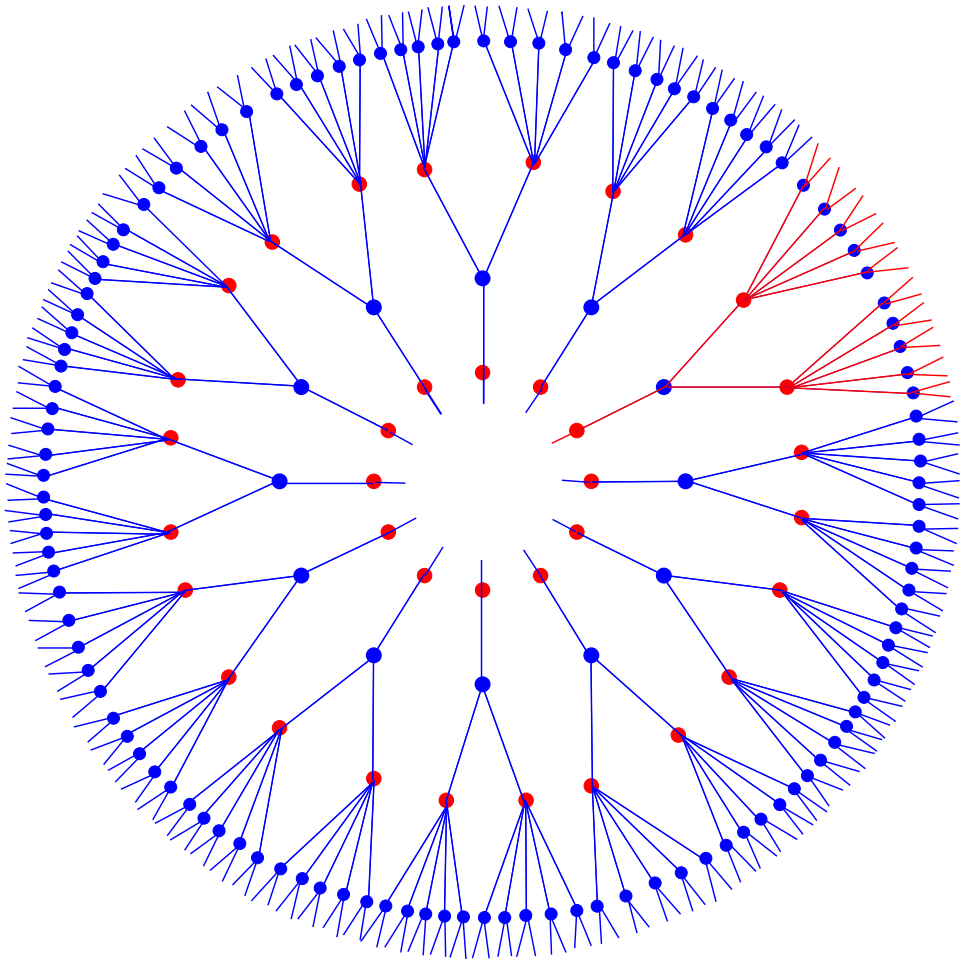
Ein zufällig gewählter (d, k)-Graph kann einen p_0-Anteil von Auslöschungen mit hoher Wahrscheinlichkeit korrigieren dann und nur dann, wenn

$$p_0 \cdot (1 - (1 - x)^{k-1})^{d-1} < x \qquad \text{for } x \in (0, p_0).$$



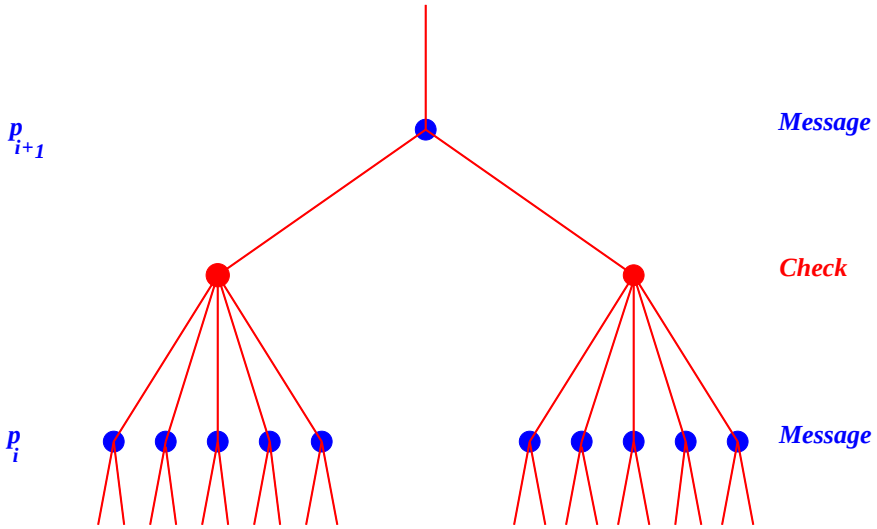
Analyse: $(3, 6)$-Graphen

Expandiere Nachbarschaft von Nachrichtenknoten.



Analyse: $(3, 6)$-Graphen

p_i Wahrscheinlichkeit, daß Nachrichtknoten nach der i-ten Iteration nicht korrigiert ist.



$$p_{i+1} = p_0(1 - (1 - p_i)^5)^2 < p_i.$$

Analyse: $(3, 6)$-Graphen

Rigoroses Argument:

- Nachbarschaft ist Baum: Mit hoher Wahrscheinlichkeit. Standardargument.

- Obiges Argument funktioniert für erwarteten Anteil von Auslöschungen in der i-ten Iteration.

  Eigentlicher Wert ist konzentriert um den Erwartungswert p_ℓ: Edge exposure martingale, Azumas Ungleichung.

Der allgemeine Fall

λ_i und ρ_i Anteil der Kanten vom Grad i auf der linken und der rechten Seite des Graphen.

$\lambda(x) := \sum_i \lambda_i x^{i-1}$, $\rho(x) := \sum_i \rho_i x^{i-1}$.

Bedingung für erfolgreiches Decodieren bei Auslöschungswahrscheinlichkeit p_0:

$$p_0 \lambda\left(1 - \rho(1-x)\right) < x$$

für alle $x \in (0, p_0)$.

Belief Propagation

Richardson-Urbanke, 1999:

f_ℓ: Wahrscheinlichkeitsdichtefunktion der Nachrichten von Check-Knoten zu Nachrichtenknoten in Runde ℓ des Algorithmus.

P_0: Dichte des Originalfehlers (in log-likelihood Darstellung, d.h., log $\frac{p(x=1)}{p(x=-1)}$).

Betrachten (d, k) reguläre Graphen.

$$\Gamma\left(f_{\ell+1}\right) = \left(\Gamma\left(P_0 \otimes {f_\ell}^{\otimes(k-1)}\right)\right)^{\otimes(d-1)},$$

wobei

$$\Gamma(f)(y) := f(\ln \coth y/2)/\sinh(y),$$

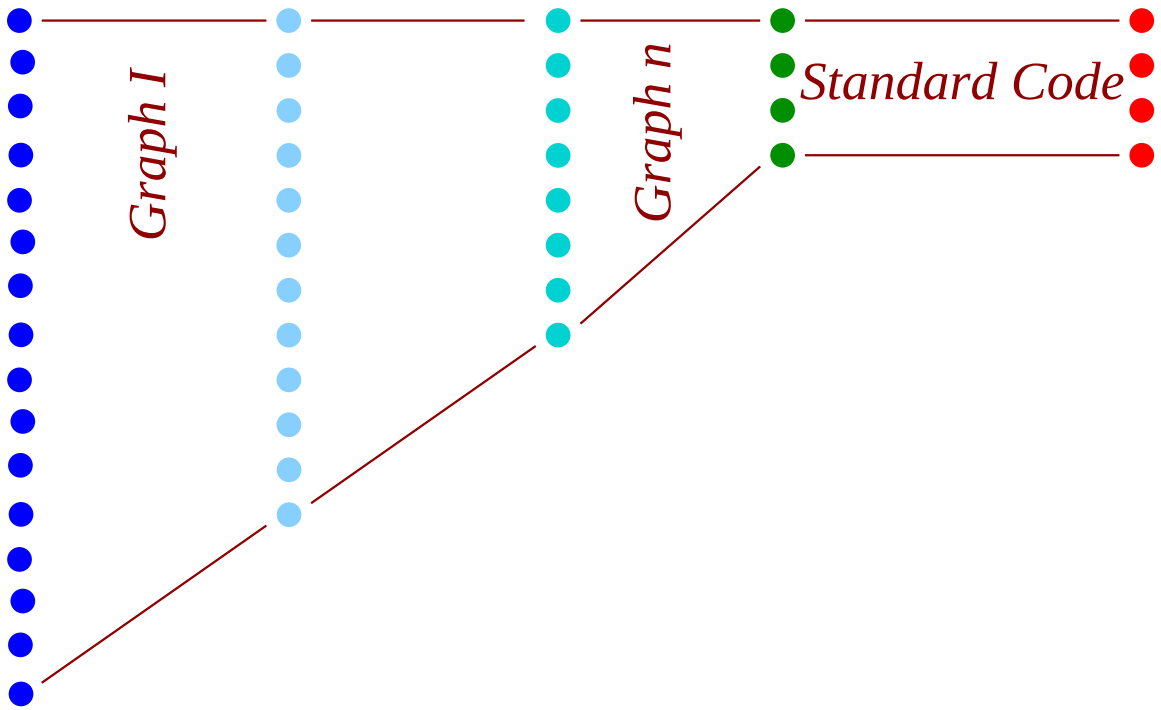
und $\otimes$ für Faltung steht.

Wollen, daß f_ℓ gegen Delta Funktion in ∞ konvergiert.

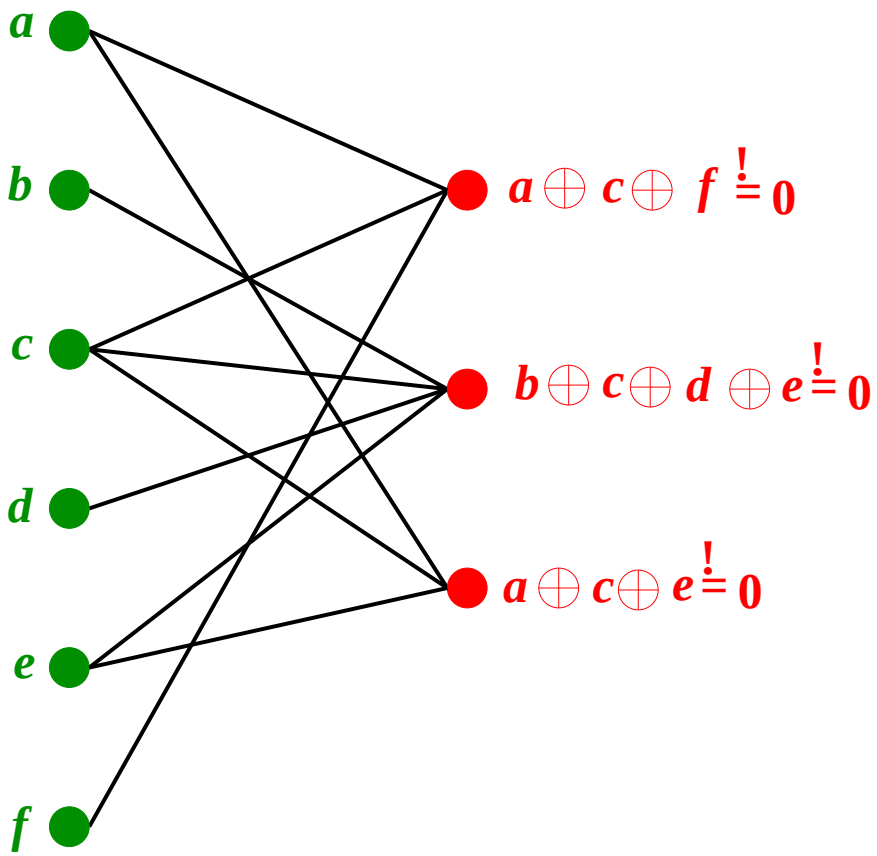
Hochdimensionales Optimierungsproblem.

Codierung

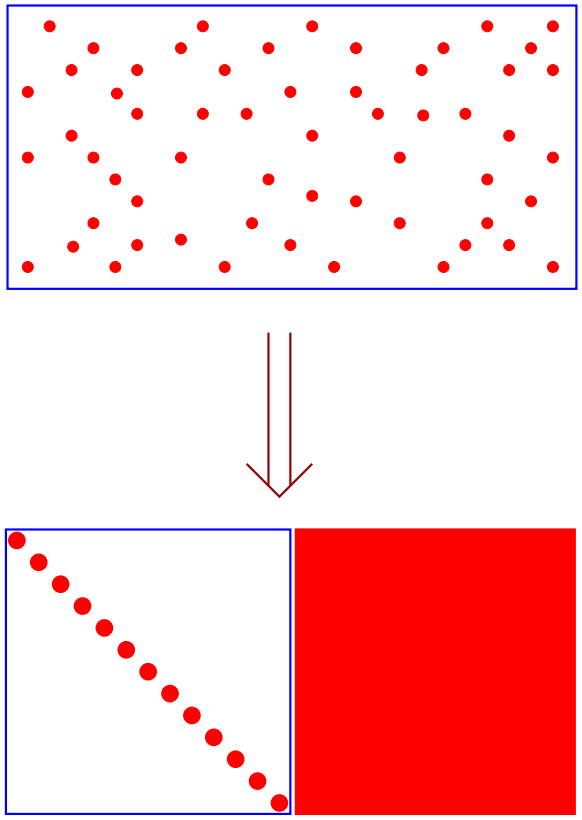
Spielman, 1995: Trivial für duale Konstruktion, aber Kasakde von Graphen nötig.



Codierung: Gallager Konstruktion



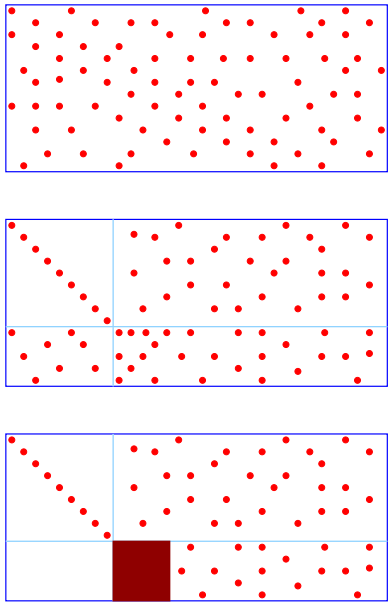
Codierung: Gallager Konstruktion



In quadratischer Zeit.

Codierung: Gallager Konstruktion

Verwende Auslöschungskorrektur!



Müssen kleinere Matrix invertieren.

Codierung

Richaradson-Urbanke, 1999: Unter leichten Bedingungen ist Codierung in Linearzeit möglich.

Genauer brauchen wir:

- Genug Nachrichtenknoten vom Grad 2: $\lambda_2 \rho'(1) > 1$.
- $\rho(1 - \lambda(1 - x)) < x$ für $x \in (0, 1)$.

$(3, 6)$-Graph kann in Zeit $(0.07n)^2$ codiert werden.

Kapzitätsoptimalität

Wollen Codes entwerfen, die asymptotisch einen $1-R$-Anteil Auslöschung korrigieren können.

Wollen λ und ρ so entwerfen, daß

$$p_0 \lambda(1-\rho(1-x)) < x$$

für alle $x \in (0, p_0)$, und p_0 beliebig nahe an

$$1-R = \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}.$$

Tornado Codes

Extrem irreguläre Graphen, die für jede Rate R Folgen von Codes liefern, die beliebig nahe an der Kapazität des Auslöschungskanals sind.

Gradstruktur?

Wähle Entwurfsparameter D.

$$\lambda(x) := \frac{1}{H(D)}\left(x + \frac{x^2}{2} + \cdots + \frac{x^D}{D}\right)$$

$$\rho(x) := \exp\left(\mu(x-1)\right),$$

wobei $H(D)$ die harmonische Summe ist $1 + 1/2 + \cdots + 1/D$ and $\mu = H(D)/\left(1 - 1/(D+1)\right)$.

Rechtsreguläre Codes

Shokrollahi, 1999:

Graphen, die auf der rechten Seite regulär sind.

Gradstruktur auf der linken Seite: Taylor-Entwicklung von

$$(1-x)^{1/m}.$$

Wahrscheinlich beste Folge.

Oswald und Shokrollahi, 2000:

Weitere Folgen und funktionentheoretischer Ansatz.

Andere Kanäle?

f Dichtefunktion.

$\lambda(f) := \sum_i \lambda_i f^{\otimes(i-1)}$.

$\rho(f) := \sum_i \rho_i f^{\otimes(i-1)}$.

$$\Gamma\left(f_{\ell+1}\right) = \rho\left(\Gamma\left(P_0 \otimes \lambda(f_\ell)\right)\right).$$

Wollen P_0, so daß $f_\ell \to \Delta_\infty$.

Bedingungen

Richardson-Shokrollahi-Urbanke, 1999:

- Symmetrie: $f(x) = f(-x)\mathrm{e}^x$ (falls Kanal symmetrisch).

- Fixpunktsatz: Falls $\int_{-\infty}^{0} f_i \mathrm{d}x = \int_{-\infty}^{0} f_j \mathrm{d}x$ für $i < j$, dann ist $f_i = f_j$ Fixpunkt der Iteration.

- Stabilität: Sei $r := -\lim_{n\to\infty} \frac{1}{n} \log \int_{-\infty}^{0} (\mathrm{d}x P_0^{\otimes n})$, und $\lambda_2 \rho'(1) > e^r$. Dann existiert ϵ, so daß für alle ℓ gilt: $\int_{-\infty}^{0} (\mathrm{d}x f_\ell) > \epsilon$.

  (Falls $\lambda_2 \rho'(1) < e^r$, dann ist Fixpunkt Δ_∞ stabil.)

Stabilität

- Auslöschungskanal mit Auslöschungswahrscheinlichkeit p_0:

$$\lambda_2 \rho'(1) \leq \frac{1}{p_0}.$$

- BSC mit Wahrscheinlichkeit p:

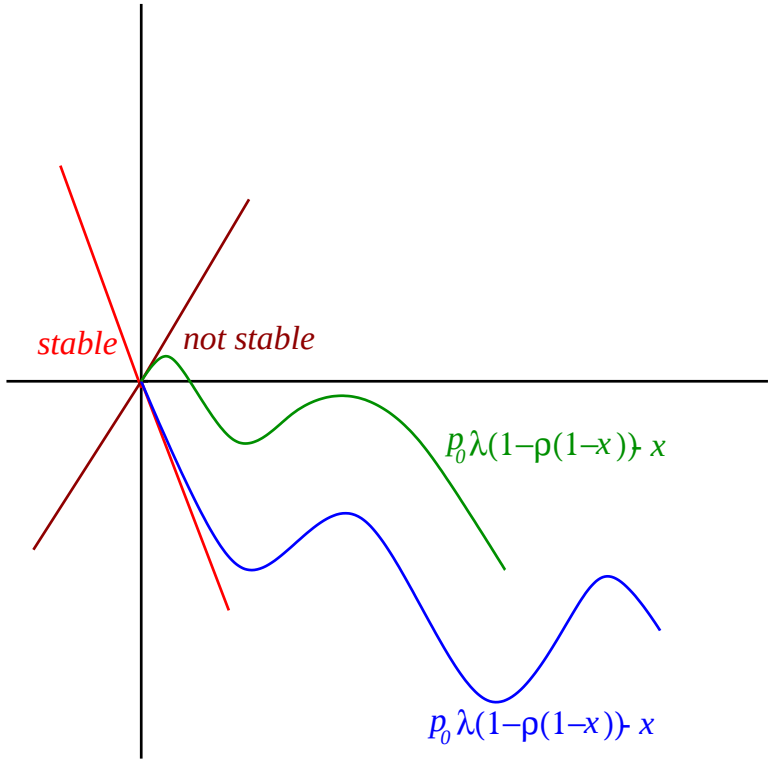
$$\lambda_2 \rho'(1) \leq \frac{1}{2\sqrt{p(1-p)}}.$$

- AWGN Kanal mit Varianz σ^2:

$$\lambda_2 \rho'(1) \leq \mathrm{e}^{-\frac{1}{2\sigma^2}}.$$

Stabilität für den Auslöschungskanal

Shokrollahi, 1999:



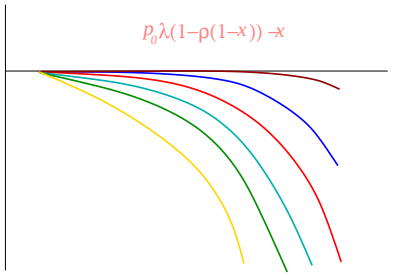
Flachheit: Höhere Stabilität

Shokrollahi, 2000:

$(\lambda_m(x), \rho_m(x))$ kapazitätsoptimal. Dann:

$$(1 - R)\lambda_m(1 - \rho_m(1 - x)) - x$$

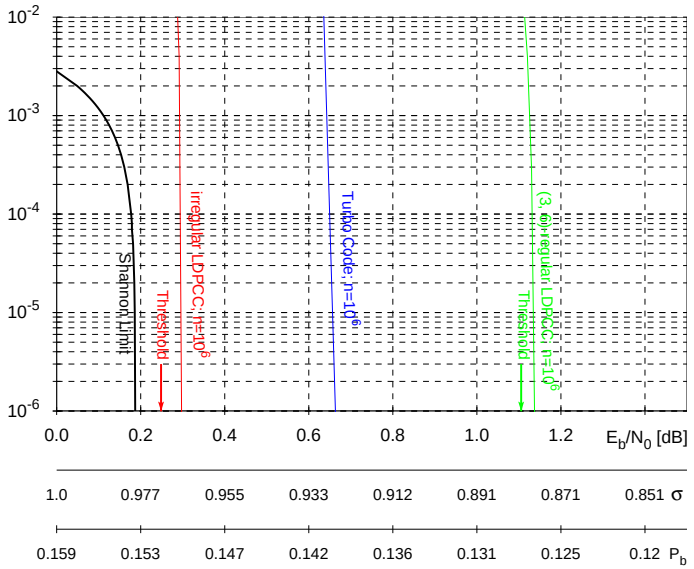
und alle Ableitungen konvergieren uniform gegen die Nullfunktion auf dem Interval $[0, 1 - R]$.



Kapazitätsoptimalität

Keine kapazitätsoptimalen Gradverteilungen für andere Kanäle bekannt.

Vermutung: Sie existieren!



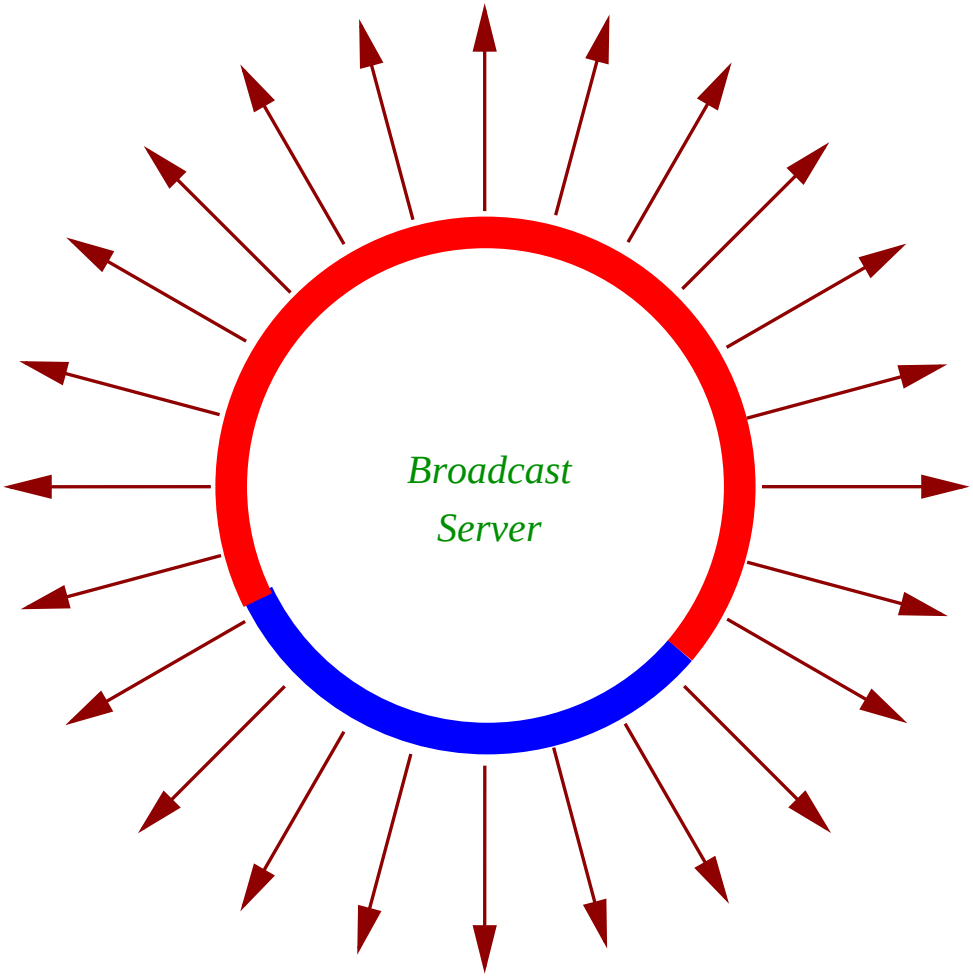
Anwendungen zu Computer-Netzwerken

Verteilung von großen Datenmengen an große Anzahl von Klienten.

Wollen:

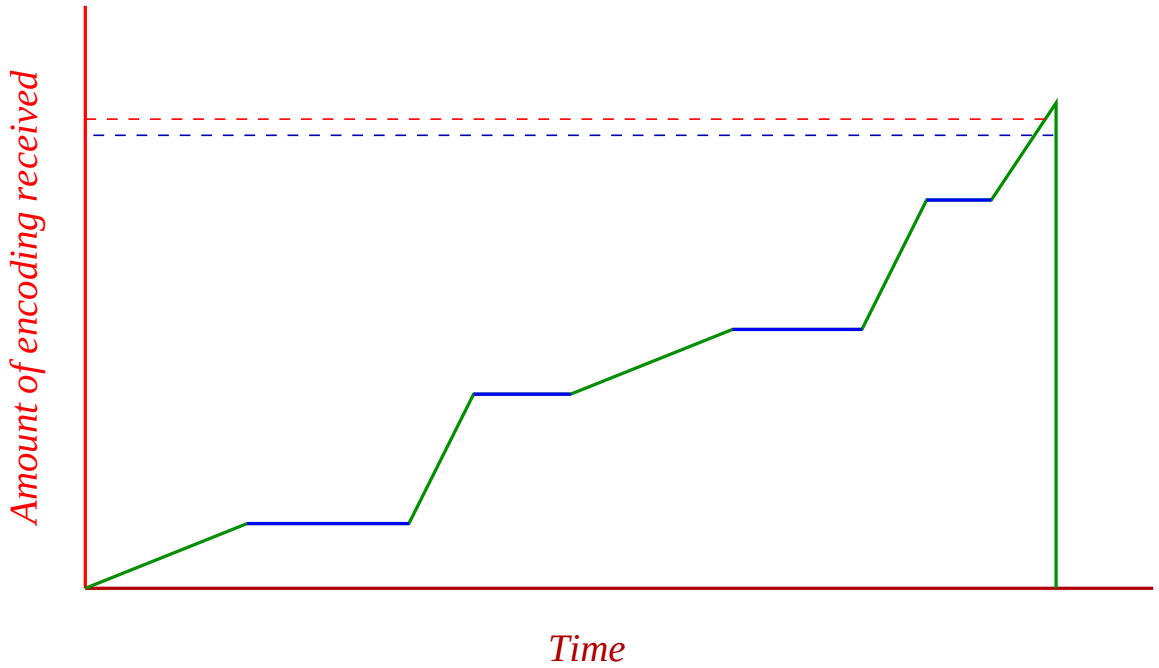
- Zuverlässigkeit,
- Wenig Netwerk-Overhead,
- Unterstützung von vielen Empfängern mit heterogenen Charakteristiken,
- daß Benutzer die Daten zu beliebigen Zeiten herunterladen und das Download unterbrechen können.

Eine Lösung



Eine Lösung

Klient meldet sich bei der Multicast-Gruppe an, bis er genug Daten erhalten hat, und decodiert Originaldaten.



digitalfountain

Offene Fragen

Asymptotische Theorie

1. Klassifikation kapazitätsoptimaler Gradfolgen für den Auslöschungskanal.
2. Kapazitätsoptimale Folgen für andere Kanäle.
3. Exponentiell kleine Fehlerwahrscheinlichkeit des Decodierers (anstatt polynomiell klein).

Explizite Konstruktionen

1. Konstruktionen mittles endlicher Geometrien.
2. Konstruktion mittels Reed-Solomon-Codes.
3. Algebraische Konstruktionen.

Kurze Codes

Graphen mit Zyklen.

Algorithmische Fragen

1. Entwurf und Analyse von neuen Decodieralgorithmen.
2. Entwurf von neuen Codierern.

Anwendungen

Paket-orientierte Mobilfunknetzwerke.