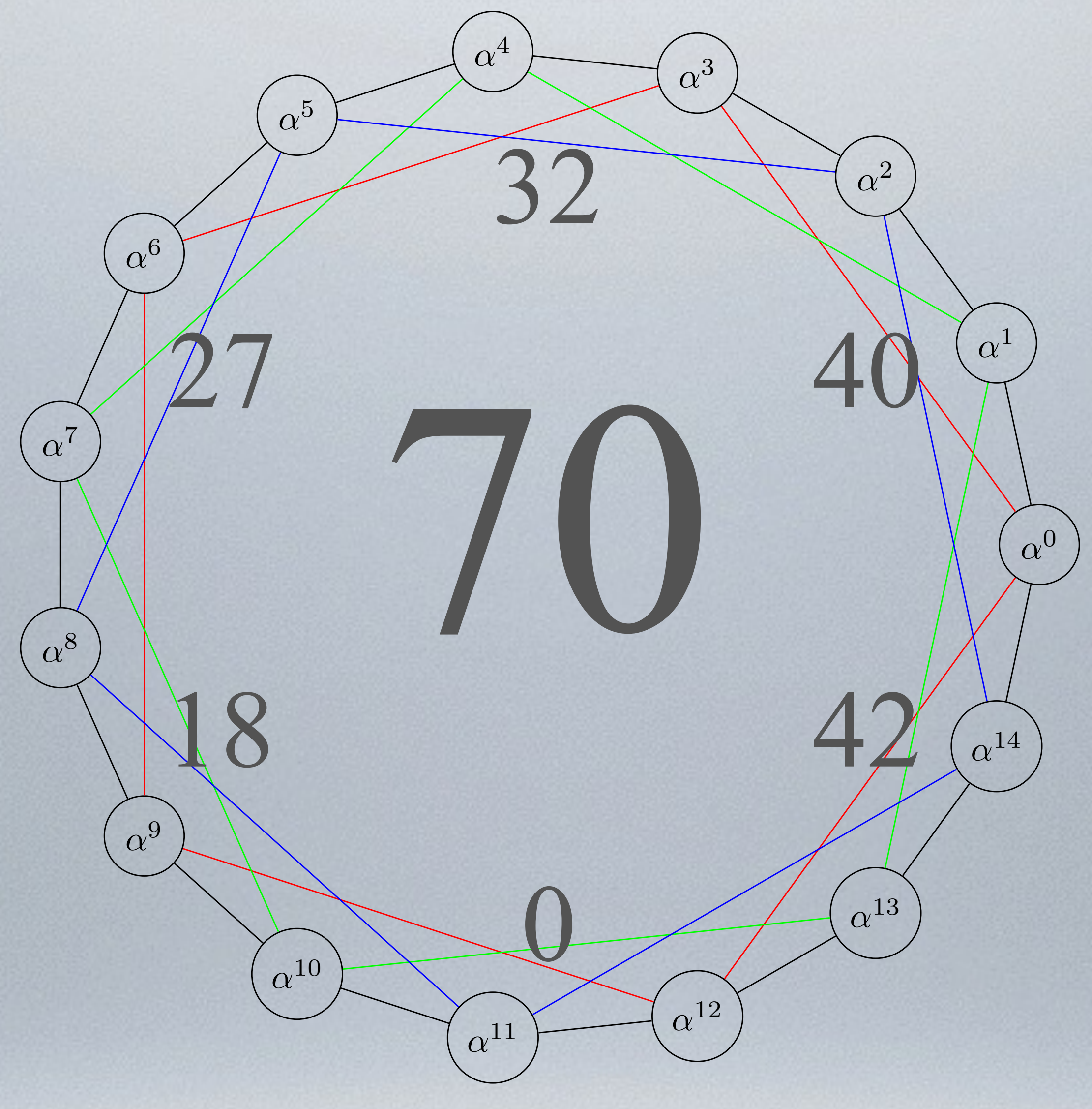


GRS Codes with Fast Encoding/Decoding Algorithms

Amin Shokrollahi
EPFL

Elements of Coding Theory

Message

Channel

Received

001001100100

001001100100

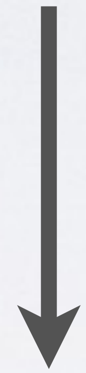
How do we correct the errors?

By cleverly appending redundant information

Elements of Coding Theory

By cleverly appending redundant information

0010011001001001100101



0011011000001001110100

Use the knowledge of how the redundancy was formed to correct the errors

What is Coding Theory About?

- How do we design the redundancy so that
  - we use as little redundancy as possible
  - we can correct as many errors as possible
- What are the fundamental limits?
- How do we “encode” efficiently?
- How de we “decode” efficiently?

More than 50 years of coding theory have been about answering these questions. Research has revealed many solutions using algebra, combinatorics, probability theory, algebraic geometry, graph theory, algorithm design, ...

Algebraic Theory

- How do we design the redundancy so that
  - we use as little redundancy as possible
  - we can correct as many errors as possible

- Attempt: use structure of a vector space.

- A <u>linear code</u> of <u>dimension</u> k and <u>length</u> n over an alphabet of size q is a k-dimensional subspace of $GF(q)^n$

- The "redundancy" is added by embedding $GF(q)^k$ into this vector space.

Algebraic Theory

- How do we design the redundancy so that
  - we use as little redundancy as possible
  - we can correct as many errors as possible

- Error-correction: minimum weight of a nonzero codeword ("weight" of a vector = number of nonzero coordinates -- note dependency on chosen basis).

- If the minimum weight is d then we can (theoretically) correct up to $(d-1)/2$ errors

Algebraic Theory

- How do we "encode" efficiently? Linear Algebra
- How de we "decode" efficiently? Question of the agens....

What is this Talk about?

Can we use Computer Algebra for Practical Decoding of algebraic codes?

Pro: asymptotically faster algorithms.

Con: Slow and costly in commodity hardware.

Example where methods from CA lead to better design.

Reed-Solomon Codes

$$\langle \alpha \rangle = \mathbb{F}_q^\times$$

$$g(x) = (x-1)(x-\alpha)\cdots(x-\alpha^{d-2})$$

$$\mathcal{C} = \{f \in \mathbb{F}_q[x]_{<q-1} \mid f \equiv 0 \bmod g\}$$

$$\dim(\mathcal{C}) = n - d + 1 =: k$$

$$\text{minimum distance} = d$$

Encoding

$$c_0 \quad c_1 \quad \cdots \quad c_{k-1}$$

$$c_0 x^{d-1} + c_1 x^d + \cdots + c_{k-1} x^{k-1}$$

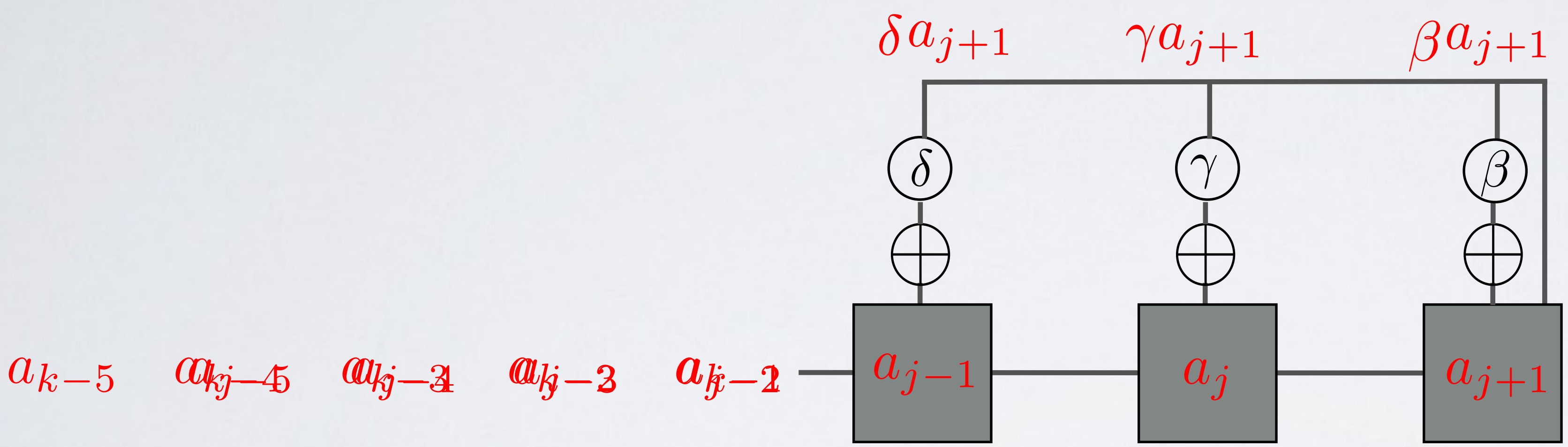
$$c(x) \bmod g(x) = r(x)$$

$$-r_0 \quad \cdots \quad -r_{d-2}$$

$g(x) = (x-1)(x-\alpha)\cdots(x-\alpha^{d-2})$

Example

$$g(x) = x^3 - \beta x^2 - \gamma x - \delta$$



LFSR(g)

LFSR

LFSR(g)

Output of the LFSR(g) is the remainder of the division of input by g.

Decoding Chain

$$g(x) = (x-1)(x-\alpha)\cdots(x-\alpha^{d-2})$$

Received

$u_0 \quad u_1 \quad \cdots \quad u_{n-1}$

Syndromes

Error locator

BM unit

Chien Search

c_0

Error values

Finding the Error Locations

Error locator polynomial

$$h(x) = h_0 + h_1 x + \cdots + h_{t-1} x^{t-1}$$

Error locations

$$\{i \mid h(\alpha^{-i}) = 0\}$$

Finding the Error Locations

Find zeros of a polynomial

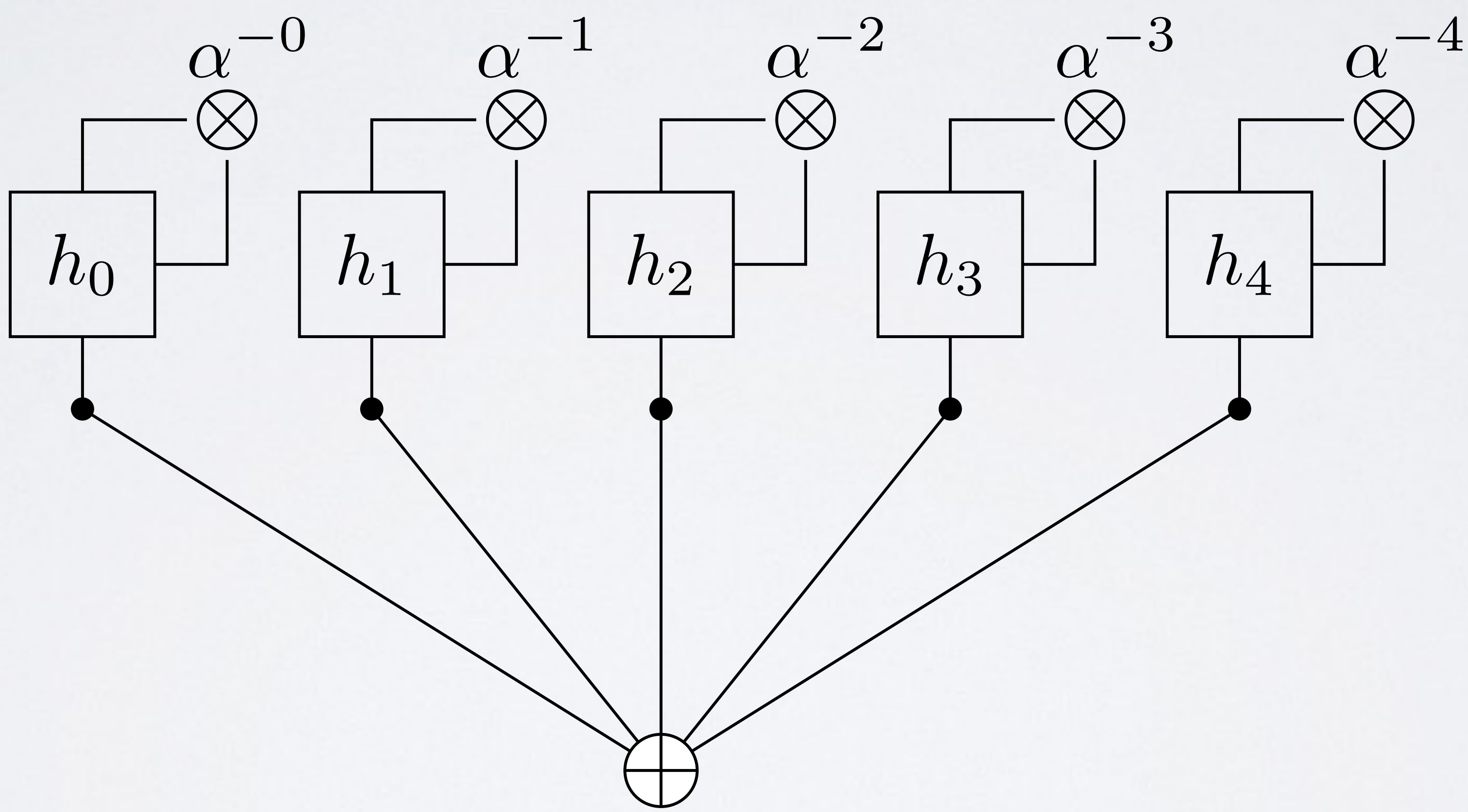
One could use methods from Computer Algebra

One should use methods from Computer Algebra

Well, no. Way too expensive in hardware.

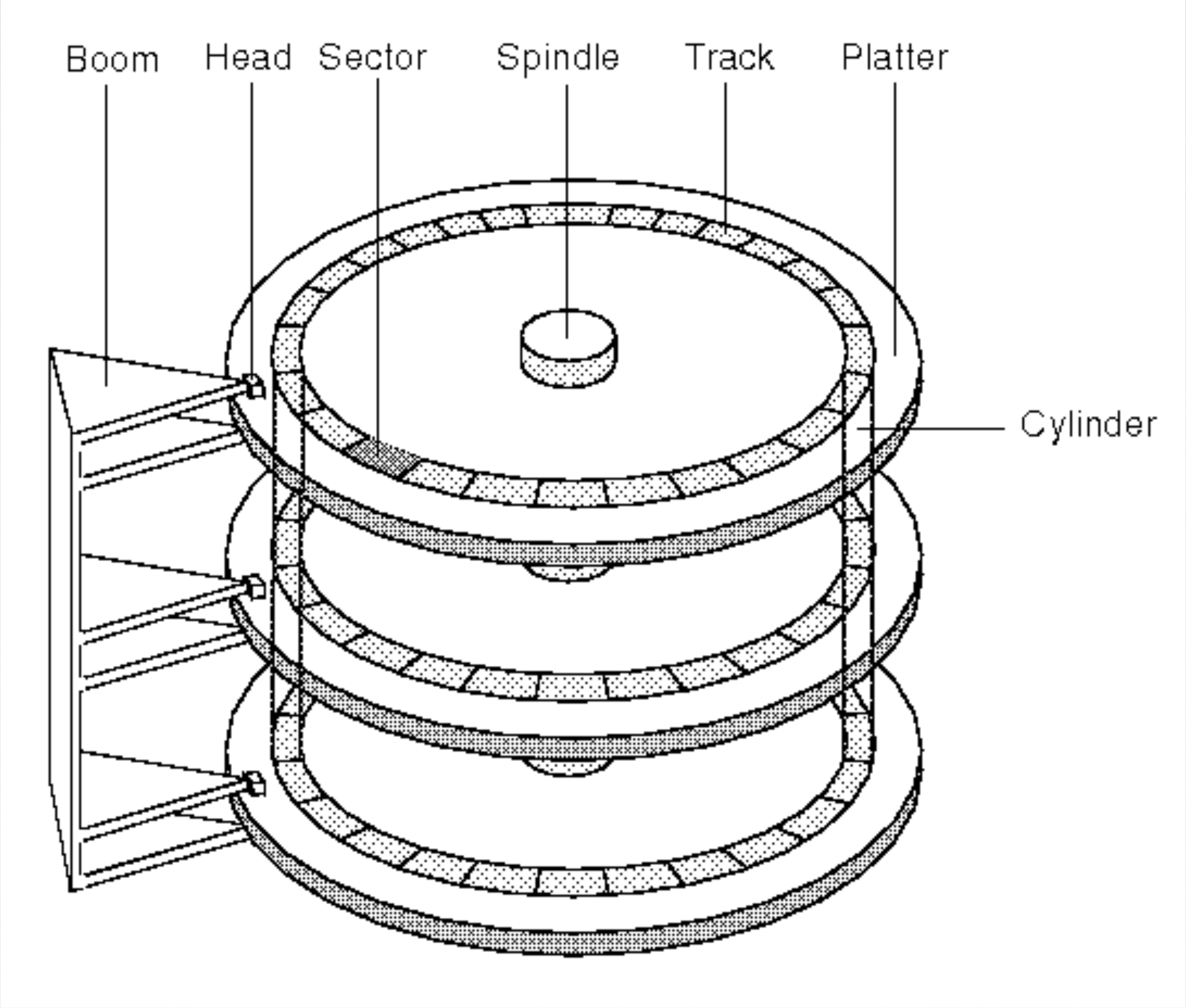
The Reality: Chien Search

Try all nonzero elements of the field!

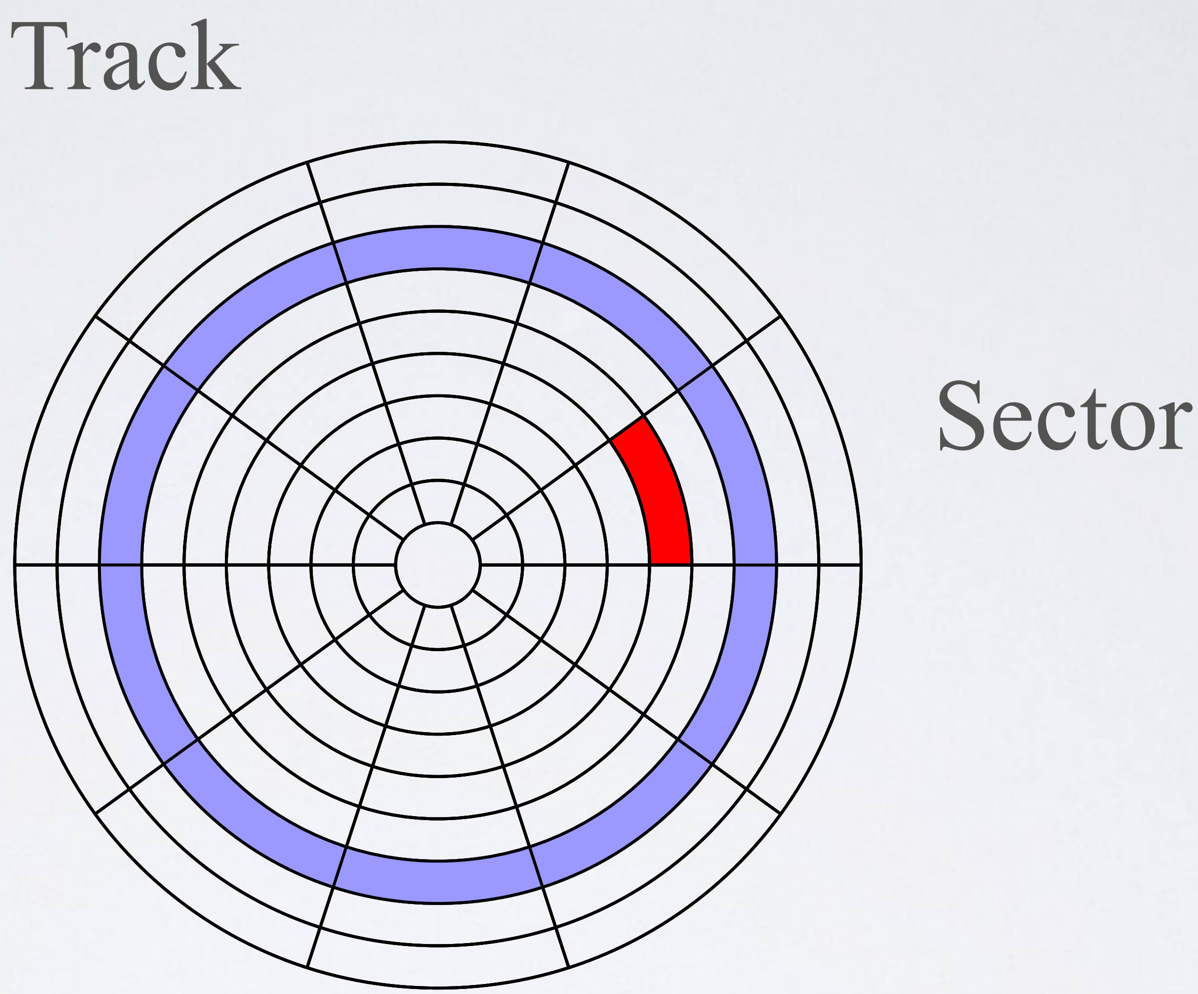


Hard Disks

Hard Disks



Hard Disks



Chien Search

Old sector size: 512 bytes

New sector size: 4 kilobytes

RS-code is defined over $\mathbb{F}_{4096}$

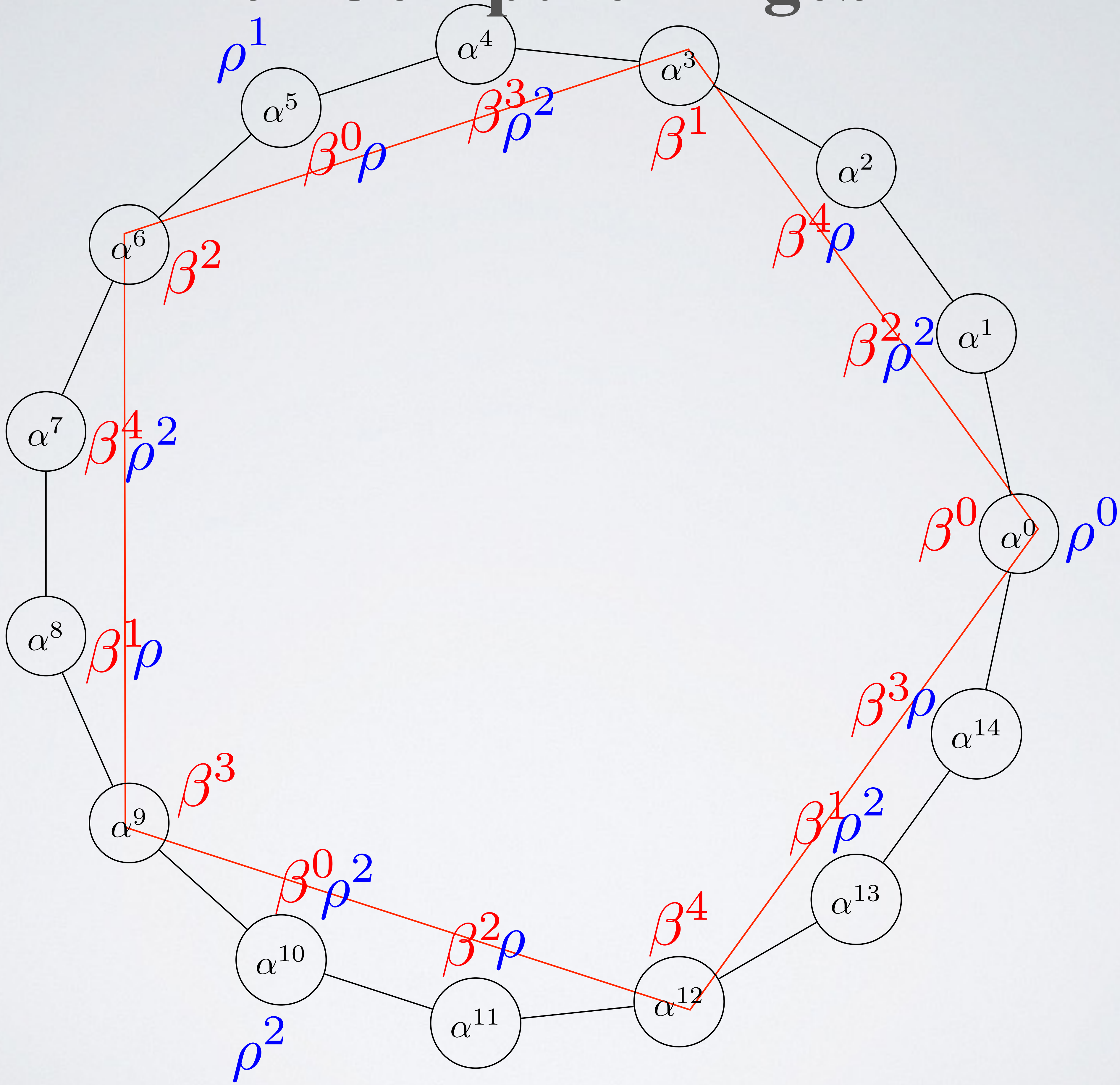
Chien search becomes bottleneck

Chien Search

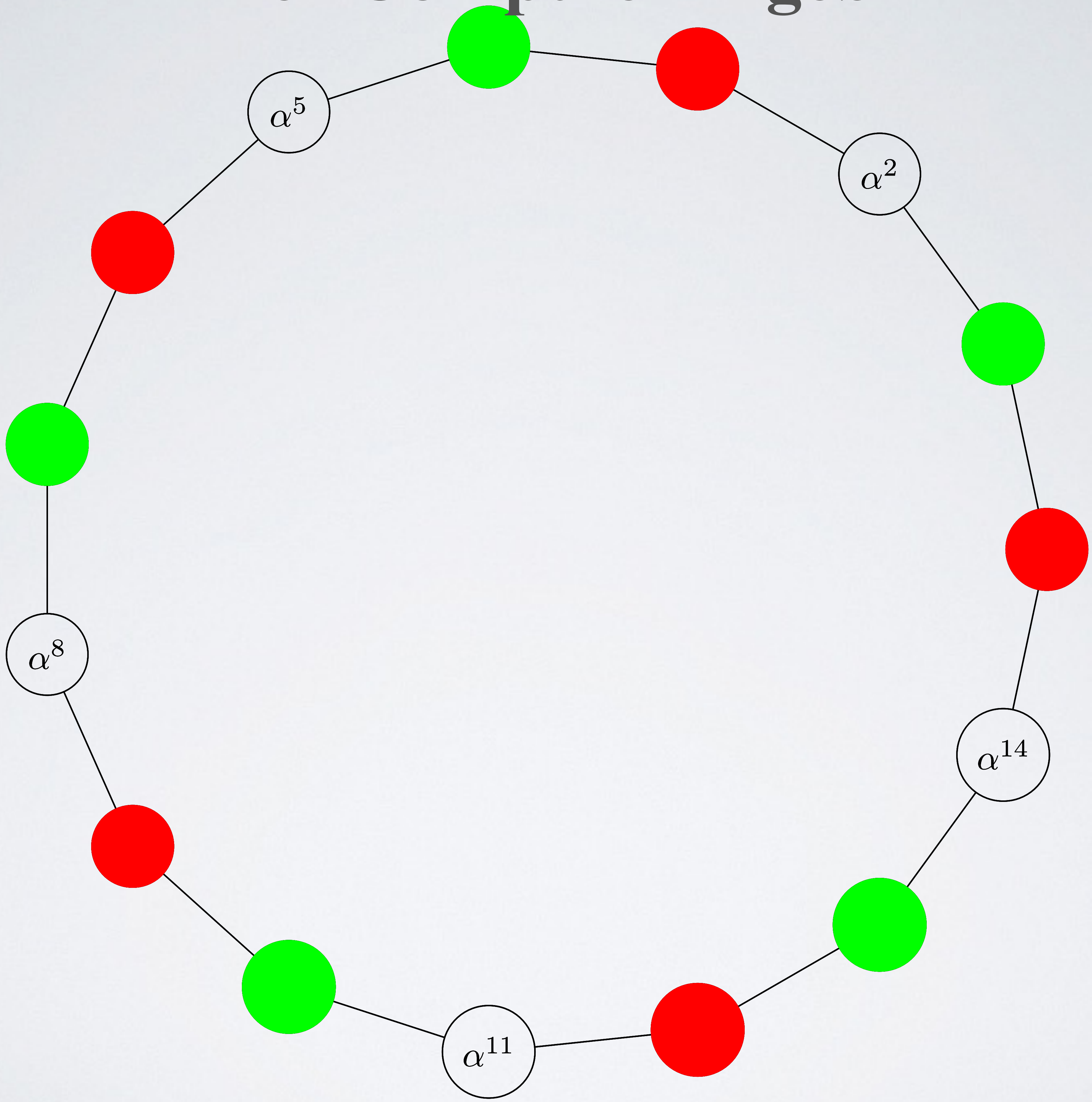
Martin Hassner, Hitachi Global Storage Solutions:

Can we reduce the running time of the Chien Search?

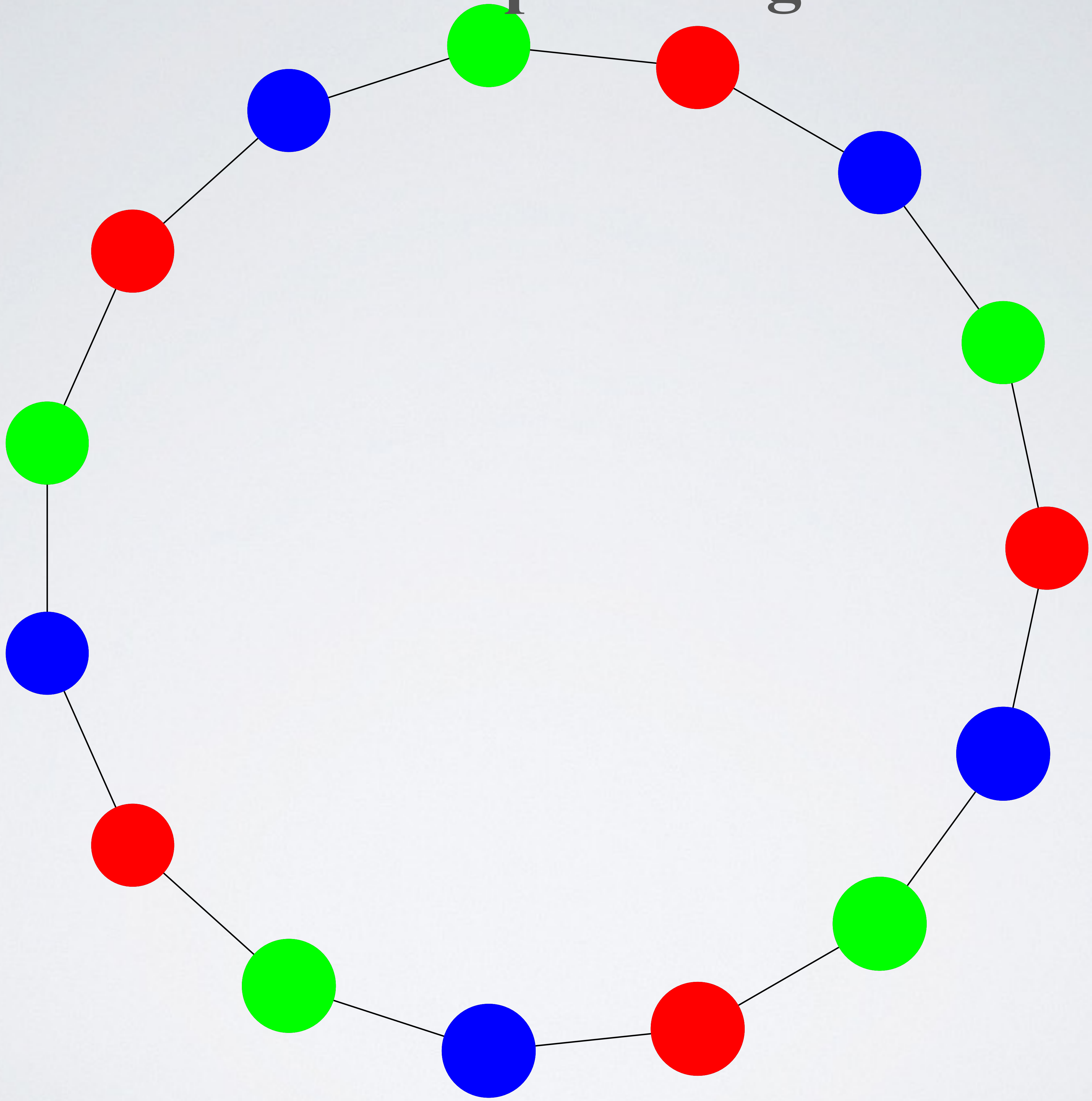
Enter Computer Algebra



Enter Computer Algebra



Enter Computer Algebra



Chien Search

$$h_0 + h_1 x + h_2 x^2 + h_3 x^3 + h_4 x^4 + h_5 x^5$$

$$h_0 + h_1 \beta^i \rho^j + h_2 \beta^{2i} \rho^{2j} + h_3 \beta^{3i} + h_4 \beta^{4i} \rho^j + h_5 \beta^{5i} \rho^{2j}$$

Multiple Evaluation

DFT(3)

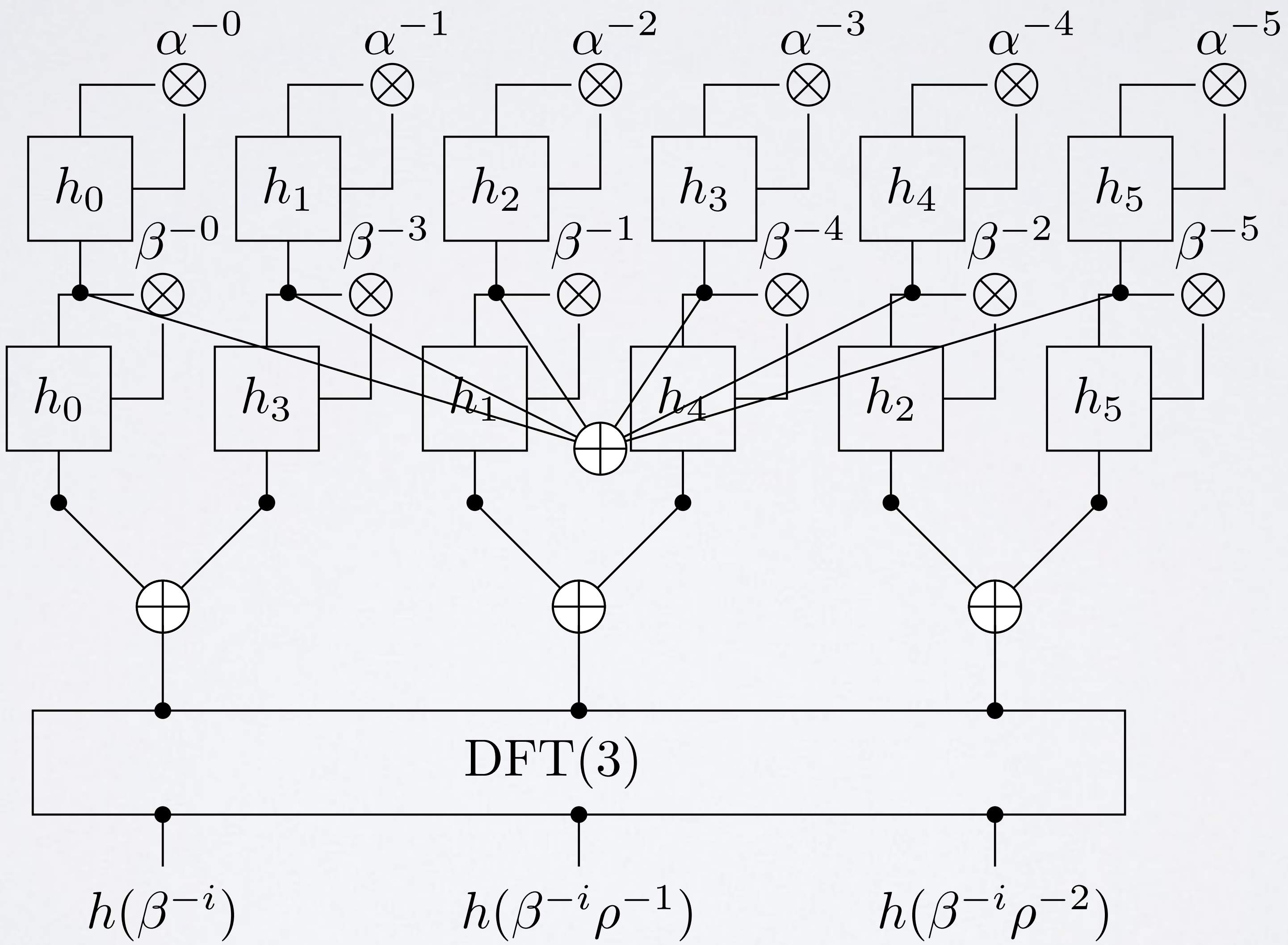
$$h_0 + h_3\beta^{3i} \longrightarrow \quad \longrightarrow h(\beta^i)$$

$$h_2\beta^{2i} + h_5\beta^{5i} \longrightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & \rho & \rho^2 \\ 1 & \rho^2 & \rho \end{bmatrix} \longrightarrow h(\beta^i\rho)$$

$$h_1\beta^{i} + h_3\beta^{4i} \longrightarrow \quad \longrightarrow h(\beta^i\rho^2)$$

3 values in every cycle

Circuits



α^{-0}
α^{-1}
α^{-2}
α^{-3}
α^{-4}
α^{-5}
h_0
h_1
h_2
h_3
h_4
h_5
β^{-0}
β^{-3}
β^{-1}
β^{-4}
β^{-2}
β^{-5}
h_0
h_3
h_1
h_4
h_2
h_5
DFT(3)
$h(\beta^{-i})$
$h(\beta^{-i}\rho^{-1})$
$h(\beta^{-i}\rho^{-2})$

Circuits

Essentially same area as before

Circuits

Three times the speed

Decoding Chain

Syndromes ?

BM

Chien Search

Error values

Syndromes

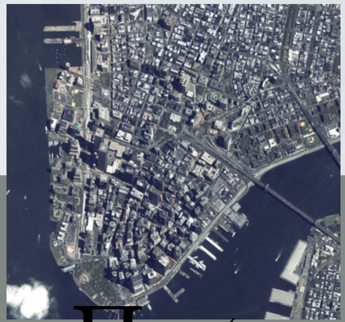
$\{1, \alpha, \alpha^2, \cdots, \alpha^{d-2}\}$ Not closed under ρ

$$u_0 + u_1 x + \cdots + u_{n-1} x^{n-1}$$

$$u(1), u(\alpha), \ldots, u(\alpha^{d-2})$$

Enter Coding Theory

Change the code. $$g(x) = \prod_{i=0}^{d-2} (x - \alpha^i)$$



DFT(3)

LFSR(g_0)

LFSR(g_1)

LFSR(g_2)

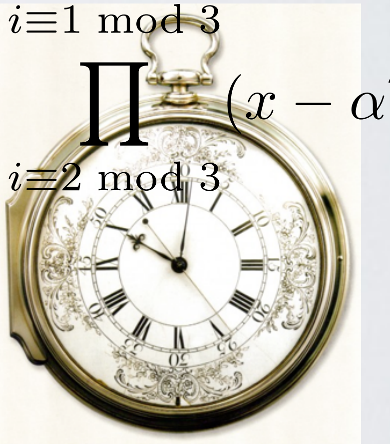
IDFT(3)

R(g)

$$g_0(x) = \prod_{i \equiv 0 \bmod 3} (x - \alpha^i)$$

$$g_1(x) = \prod_{i \equiv 1 \bmod 3} (x - \alpha^i)$$

$$g_2(x) = \prod_{i \equiv 2 \bmod 3} (x - \alpha^i)$$



Theorem: Resulting code is generalized RS.

New Syndrome Calculation

1

$\rho \cdot \{\mathbf{1}, \boldsymbol{\alpha}, \alpha^2, \cdots, \alpha^{(d-2)/3}\}$ Not closed under ρ

ρ^2

$$u_0 + u_1 x + \cdots + u_{n-1} x^{n-1}$$

$$u(1), \ldots, u(\alpha^{(d-2)/3})$$

$$u(1), u(\alpha), \ldots, u(\alpha^{d-2})$$

$$u(\rho), \ldots, u(\rho\alpha^{(d-2)/3})$$

$$u(\rho^2), \ldots, u(\rho^2\alpha^{(d-2)/3})$$

Decoding Chain

Syndromes ?

BM ?

Chien Search

Error values

BM Unit

Still not really bottleneck.

Speed-up using DFT's probably possible.

Final Remarks

Method works for every divisor of q-1 (not just 3).

For a divisor s of q-1, we get speed-up by factor s at the expense of DFT units.

Surprise: method also works for the standardized [255,239,17] code, even though the set of roots is not closed!

Method can be used in legacy systems for speed-up.

Theorem

Code as described above is generalized RS.

Proof

Show that check matrix is of the form

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & a_{n-1}^2 & a_n^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \cdots & \alpha_{n-1}^{r-1} & \alpha_n^{r-1} \end{pmatrix}$$

$\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ distinct

$\Delta_1 \cdot \Delta_2 \cdots \Delta_n \neq 0$

Proof

Check matrix is of the form

$$\left(\begin{array}{c|c|c} V_0 & 0 & 0 \\ \hline 0 & V_1 & 0 \\ \hline 0 & 0 & V_2 \end{array}\right) \cdot \left(\begin{pmatrix} 1 & 1 & 1 \\ 1 & \rho & \rho^2 \\ 1 & \rho^2 & \rho \end{pmatrix} \otimes I_m \right)$$

V_0, V_1, V_2 Vandermonde matrices

Proof

Check matrix is of the form

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & a_{n-1}^2 & a_n^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \cdots & \alpha_{n-1}^{r-1} & \alpha_n^{r-1} \end{pmatrix}$$

$$\{\alpha_1, \ldots, \alpha_n\} = \bigcup_{j=0}^{2} \{\rho^j, \rho^j \alpha, \ldots, \rho^j \alpha^{m-1}\}$$

Publication

US008296632B1

(12) **United States Patent** (10) **Patent No.: US 8,296,632 B1**
Shokrollahi (45) **Date of Patent: Oct. 23, 2012**

(54) **ENCODING AND DECODING OF GENERALIZED REED-SOLOMON CODES USING PARALLEL PROCESSING TECHNIQUES**

(76) Inventor: **Mohammad Amin Shokrollahi**, Preverenges (CH)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 644 days.

(21) Appl. No.: **12/479,605**

(22) Filed: **Jun. 5, 2009**

Related U.S. Application Data

(60) Provisional application No. 61/059,456, filed on Jun. 6, 2008.

(51) **Int. Cl.**

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,845,713	A *	7/1989	Zook	....	714/784
4,873,688	A *	10/1989	Maki et al.	....	714/784

* cited by examiner

Primary Examiner — Fritz Alphonse
(74) *Attorney, Agent, or Firm* — Kilpatrick Townsend & Stockton LLP

(57) **ABSTRACT**

A system, computer program, and/or method for encoding data that can correct r/2 errors. The original symbols are transformed using a Fourier transform of length p. Generator polynomials are used to encode the p blocks separately, and an inverse Fourier transform is applied to obtain the redundant symbol. In a decoding system, Fourier transforms are applied to every set of p consecutive symbols of the received vector, to obtain p blocks of symbols which in total have the same size as the received vector. Next, a syndrome calculator is applied to each of these blocks to produce p syndromes. The syndromes are forwarded to a Berlekamp-Massey unit and an

Future Work?

Try to come up with method for the case where q-1 is 2 times a prime.

Speedup of the BM unit?

FPGA implementation?