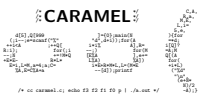
WAIFI — July 16–19 , 2012

Finding Optimal Formulae for Bilinear Maps

Razvan Barbulescu Jérémie Detrey Nicolas Estibals
Paul Zimmermann

CARAMEL project-team, LORIA
University of Lorraine, INRIA, CNRS



A bit of history

- Multiplication is an expensive arithmetic operation
- Well-studied problem
  - Karatsuba (1962)
  - Toom–Cook (1963), evaluation-interpolation schemes
  - CRT-based algorithms
  - Schönhage–Strassen algorithm (1971)
  - ...
- *Five-, six-, and seven-term Karatsuba-like formulae*, P. Montgomery (2005)
  - ad-hoc formulae
  - partly exhaustive search for five-, six- and seven-term multiplication
  - successfully used in practice in the M4RIE library (Albrecht, 2011)

Outline of the talk

- Problem description
- Resolution methods
- Results

Outline of the talk

- Problem description
- Resolution methods
- Results

Example: a 2-term polynomial times a 3-term one

- We want to compute

$$\begin{aligned} C(X) &= (a_1 \cdot X + a_0) \times (b_2 \cdot X^2 + b_1 \cdot X + b_0) \\ &= a_1 b_2 \cdot X^3 + (a_1 b_1 + a_0 b_2) \cdot X^2 + (a_0 b_1 + a_1 b_0) \cdot X + a_0 b_0 \end{aligned}$$

Example: a 2-term polynomial times a 3-term one

- We want to compute

$$\begin{aligned} C(X) &= (a_1 \cdot X + a_0) \times (b_2 \cdot X^2 + b_1 \cdot X + b_0) \\ &= a_1 b_2 \cdot X^3 + (a_1 b_1 + a_0 b_2) \cdot X^2 + (a_0 b_1 + a_1 b_0) \cdot X + a_0 b_0 \end{aligned}$$

- Only 5 products required instead of 6
  - use Karatsuba's trick

$$C(X) = a_1 b_2 \cdot X^3 + (a_1 b_1 + a_0 b_2) \cdot X^2 + ((a_0 + a_1)(b_0 + b_1) - a_1 b_1 - a_0 b_0) \cdot X + a_0 b_0$$

Example: a 2-term polynomial times a 3-term one

- We want to compute

$$\begin{aligned} C(X) &= (a_1 \cdot X + a_0) \times (b_2 \cdot X^2 + b_1 \cdot X + b_0) \\ &= a_1 b_2 \cdot X^3 + (a_1 b_1 + a_0 b_2) \cdot X^2 + (a_0 b_1 + a_1 b_0) \cdot X + a_0 b_0 \end{aligned}$$

- Only 5 products required instead of 6
  - use Karatsuba's trick

$$C(X) = a_1 b_2 \cdot X^3 + (a_1 b_1 + a_0 b_2) \cdot X^2 + ((a_0 + a_1)(b_0 + b_1) - a_1 b_1 - a_0 b_0) \cdot X + a_0 b_0$$

  - compute the products

$$\begin{aligned} g_0 &= a_0 \cdot b_0, \\ g_1 &= a_0 \cdot b_2, \\ g_2 &= a_1 \cdot b_1, \\ g_3 &= a_1 \cdot b_2, \text{ and} \\ g_4 &= (a_0 + a_1) \cdot (b_0 + b_1). \end{aligned}$$

  - reconstruct the result

$$C(X) = g_3 \cdot X^3 + (g_1 + g_2) \cdot X^2 + (g_4 - g_2 - g_0) \cdot X + g_0$$

General form of a multiplication formula

- We want to compute, over a given field K (or any K-algebra $\mathbf{K}$),
  $(a_{n-1} \cdot X^{n-1} + \cdots + a_0) \times (b_{m-1} \cdot X^{m-1} + \cdots + b_0) = c_{n+m-2} \cdot X^{n+m-2} + \cdots + c_0$

- All formulae for multiplication can be expressed as:
  - compute some linear combinations of the a_i's
    $$a'_j = \sum \alpha_{i,j} \cdot a_i$$
  - compute some linear combinations of the b_i's
    $$b'_j = \sum \beta_{i,j} \cdot b_i$$
  - perform some products
    $$g_j = a'_j \cdot b'_j$$

General form of a multiplication formula

- We want to compute, over a given field K (or any K-algebra $\mathbf{K}$),
  $(a_{n-1} \cdot X^{n-1} + \cdots + a_0) \times (b_{m-1} \cdot X^{m-1} + \cdots + b_0) = c_{n+m-2} \cdot X^{n+m-2} + \cdots + c_0$

- All formulae for multiplication can be expressed as:
  - compute some linear combinations of the a_i's
    $$a'_j = \sum \alpha_{i,j} \cdot a_i$$
  - compute some linear combinations of the b_i's
    $$b'_j = \sum \beta_{i,j} \cdot b_i$$
  - perform some products
    $$g_j = a'_j \cdot b'_j$$
  - reconstruct the result by linear combinations of the products
    $$c_k = \sum \gamma_{j,k} \cdot g_j$$

General form of a multiplication formula

- We want to compute, over a given field K (or any K-algebra $\mathbf{K}$),
  $(a_{n-1} \cdot X^{n-1} + \cdots + a_0) \times (b_{m-1} \cdot X^{m-1} + \cdots + b_0) = c_{n+m-2} \cdot X^{n+m-2} + \cdots + c_0$

- All formulae for multiplication can be expressed as:
  - compute some linear combinations of the a_i's
    $$a'_j = \sum \alpha_{i,j} \cdot a_i, \quad \text{with } \alpha_{i,j} \in K$$
  - compute some linear combinations of the b_i's
    $$b'_j = \sum \beta_{i,j} \cdot b_i, \quad \text{with } \beta_{i,j} \in K$$
  - perform some products
    $$g_j = a'_j \cdot b'_j, \quad \text{with } a'_j,\ b'_j \in \mathbf{K}$$
  - reconstruct the result by linear combinations of the products
    $$c_k = \sum \gamma_{j,k} \cdot g_j, \quad \text{with } \gamma_{j,k} \in K$$

General form of a multiplication formula

- We want to compute, over a given field K (or any K-algebra $\mathbf{K}$),
  $(a_{n-1} \cdot X^{n-1} + \cdots + a_0) \times (b_{m-1} \cdot X^{m-1} + \cdots + b_0) = c_{n+m-2} \cdot X^{n+m-2} + \cdots + c_0$

- All formulae for multiplication can be expressed as:
  - compute some linear combinations of the a_i's
    $$a'_j = \sum \alpha_{i,j} \cdot a_i, \quad \text{with } \alpha_{i,j} \in K$$
  - compute some linear combinations of the b_i's
    $$b'_j = \sum \beta_{i,j} \cdot b_i, \quad \text{with } \beta_{i,j} \in K$$
  - perform some products
    $$g_j = a'_j \cdot b'_j, \quad \text{with } a'_j,\ b'_j \in \mathbf{K}$$
  - reconstruct the result by linear combinations of the products
    $$c_k = \sum \gamma_{j,k} \cdot g_j, \quad \text{with } \gamma_{j,k} \in K$$

- This is also valid for any bilinear map
  $$\begin{array}{ccccccc} F: & K^n & \times & K^m & \longrightarrow & K^\ell \\ & ((a_0, \ldots, a_{n-1}) & , & (b_0, \ldots, b_{m-1})) & \longmapsto & (c_0, \ldots, c_{\ell-1}) \end{array}$$

Formal framework

$$F : K^n \times K^m \to K^\ell$$

Formal framework

$$F : K^n \times K^m \to K^\ell$$

- Represent the products and the coefficients of the result as elements of the nm-dimensional K-vector space

$$V = \operatorname{Span} \mathcal{V}, \qquad \text{with basis } \mathcal{V} = \{a_i b_j\}_{0 \le i < n,\ 0 \le j < m},$$

  where the $a_i b_j$'s are seen as formal elements

Formal framework

$$F : K^n \times K^m \to K^\ell$$

- Represent the products and the coefficients of the result as elements of the nm-dimensional K-vector space

$$V = \operatorname{Span} \mathcal{V}, \qquad \text{with basis } \mathcal{V} = \{a_i b_j\}_{0 \le i < n,\ 0 \le j < m},$$

  where the $a_i b_j$'s are seen as formal elements

- Target vectors: the coefficients of the result form a set

$$\mathcal{T} = \{c_i\}_{0 \le i < \ell} \subset V$$

Formal framework

$$F : K^n \times K^m \to K^\ell$$

- Represent the products and the coefficients of the result as elements of the nm-dimensional K-vector space

$$V = \operatorname{Span} \mathcal{V}, \qquad \text{with basis } \mathcal{V} = \{a_i b_j\}_{0 \le i < n,\ 0 \le j < m},$$

  where the $a_i b_j$'s are seen as formal elements

- Target vectors: the coefficients of the result form a set

$$\mathcal{T} = \{c_i\}_{0 \le i < \ell} \subset V$$

- Generators: the set $\mathcal{G} \subset V$ of the potential products to use in a formula

$$\mathcal{G} = \{(\sum \alpha_i a_i) \cdot (\sum \beta_j b_j) \mid \forall i,\ \alpha_i \in K \text{ and } \forall j,\ \beta_j \in K\}$$

  If $g = \lambda g'$ for some scalar λ, we keep only one.

Problem definition

Definition

Given:

1. a finite dimensional vector space V,
2. a finite set $\mathcal{G}$ generating V and
3. a finite set of vectors $\mathcal{T}$,

find the minimal number k of generators which span $\mathcal{T}$.

- The problem can be formulated as the bilinear rank problem, which is known to be NP-hard.
- Searching formulae for polynomial and matrix multiplication are particular instances (NP-hard?).

Example (continued)

Consider the previous example: 2×3-term polynomial product in $\mathbb{F}_2[X]$

- V is a 6-dimensional vector space with basis

$$\mathcal{V} = \{a_0 b_0, a_0 b_1, a_0 b_2, a_1 b_0, a_1 b_1, a_1 b_2\}$$

- The set of targets is

$$\mathcal{T} = \{a_1 b_2, a_1 b_1 + a_0 b_2, a_0 b_1 + a_1 b_0, a_0 b_0\}$$

- The set of generators contains 21 products:

$$\begin{aligned}
\mathcal{G} = \{\ & a_0 \cdot b_0, && a_1 \cdot b_0, && (a_0 + a_1) \cdot b_0, \\
& a_0 \cdot b_1, && a_1 \cdot b_1, && (a_0 + a_1) \cdot b_1, \\
& a_0 \cdot (b_0 + b_1), && a_1 \cdot (b_0 + b_1), && (a_0 + a_1) \cdot (b_0 + b_1), \\
& a_0 \cdot b_2, && a_1 \cdot b_2, && (a_0 + a_1) \cdot b_2, \\
& a_0 \cdot (b_0 + b_2), && a_1 \cdot (b_0 + b_2), && (a_0 + a_1) \cdot (b_0 + b_2), \\
& a_0 \cdot (b_1 + b_2), && a_1 \cdot (b_1 + b_2), && (a_0 + a_1) \cdot (b_1 + b_2), \\
& a_0 \cdot (b_0 + b_1 + b_2), && a_1 \cdot (b_0 + b_1 + b_2), && (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}
\end{aligned}$$

Outline of the talk

- Problem description
- Resolution methods
- Results

Naive algorithm

- Goal: find the optimal formulae (*i.e.*, with a minimum number of products)
  - enumerate the subsets $\mathcal{W} \subset \mathcal{G}$ of exactly k products which yield a valid formula
  - starting with $k = \mathrm{rk}(\mathcal{T})$, increase k until a solution is found
- Look for $\mathcal{W}$ such that
  - $\mathcal{W}$ is a set of k generators:

$$\mathcal{W} \subset \mathcal{G} \quad \text{and} \quad \#\mathcal{W} = k$$

  - $\mathcal{W}$ linearly generates the coefficients of the results:

$$\mathcal{T} \subset \mathrm{Span}\,\mathcal{W}$$

Naive algorithm

- Goal: find the optimal formulae (*i.e.*, with a minimum number of products)
  - enumerate the subsets $\mathcal{W} \subset \mathcal{G}$ of exactly k products which yield a valid formula
  - starting with $k = \mathrm{rk}(\mathcal{T})$, increase k until a solution is found
- Look for $\mathcal{W}$ such that
  - $\mathcal{W}$ is a set of k generators:

$$\mathcal{W} \subset \mathcal{G} \quad \text{and} \quad \#\mathcal{W} = k$$

  - $\mathcal{W}$ linearly generates the coefficients of the results:

$$\mathcal{T} \subset \operatorname{Span} \mathcal{W}$$

- Naive approach:
  - enumerate the $\binom{\#\mathcal{G}}{k}$ subsets $\mathcal{W}$ of size k
  - for each subset, test whether it generates $\mathcal{T}$ or not

Naive algorithm

- Goal: find the optimal formulae (*i.e.*, with a minimum number of products)
  - enumerate the subsets $\mathcal{W} \subset \mathcal{G}$ of exactly k products which yield a valid formula
  - starting with $k = \mathrm{rk}(\mathcal{T})$, increase k until a solution is found
- Look for $\mathcal{W}$ such that
  - $\mathcal{W}$ is a set of k generators:

$$\mathcal{W} \subset \mathcal{G} \quad \text{and} \quad \#\mathcal{W} = k$$

  - $\mathcal{W}$ linearly generates the coefficients of the results:

$$\mathcal{T} \subset \operatorname{Span} \mathcal{W}$$

- Naive approach:
  - enumerate the $\binom{\#\mathcal{G}}{k}$ subsets $\mathcal{W}$ of size k
  - for each subset, test whether it generates $\mathcal{T}$ or not
- $\mathcal{G}$ has to be finite:
  - look for formulae over finite fields: $K = \mathbb{F}_q$ (typically, $q = 2, 3, 4$)
  - restrict to a finite subset of the generators $\mathcal{G}$ (but search not exhaustive)

Improvement 1: subspaces instead of subsets

- Drawback of the naive approach: Distinct subsets may span the same subspace

Improvement 1: subspaces instead of subsets

- Drawback of the naive approach: Distinct subsets may span the same subspace
- Look instead for subspaces W of V such that
  - W can be generated by products only: $\operatorname{Span}(W \cap \mathcal{G}) = W$
  - only k products are required: $\dim W = k$
  - W contains the target space spanned by the target vectors: $T = \operatorname{Span} \mathcal{T} \subset W$

Improvement 1: subspaces instead of subsets

- Drawback of the naive approach: Distinct subsets may span the same subspace
- Look instead for subspaces W of V such that
  - W can be generated by products only: $\operatorname{Span}(W \cap \mathcal{G}) = W$
  - only k products are required: $\dim W = k$
  - W contains the target space spanned by the target vectors: $T = \operatorname{Span} \mathcal{T} \subset W$
- Solution space $W \Rightarrow$ several bases $\mathcal{W} \Rightarrow$ one formula per basis.

Improvement 1: subspaces instead of subsets

- Drawback of the naive approach: Distinct subsets may span the same subspace
- Look instead for subspaces W of V such that
  - W can be generated by products only: $\operatorname{Span}(W \cap \mathcal{G}) = W$
  - only k products are required: $\dim W = k$
  - W contains the target space spanned by the target vectors: $T = \operatorname{Span} \mathcal{T} \subset W$
- Solution space $W \Rightarrow$ several bases $\mathcal{W} \Rightarrow$ one formula per basis.
- No change of the worst case complexity.

Improvement 1: subspaces instead of subsets

- Look instead for subspaces W of V such that
  - W can be generated by products only: $\operatorname{Span}(W \cap \mathcal{G}) = W$
  - only k products are required: $\dim W = k$
  - W contains the target space spanned by the target vectors: $T = \operatorname{Span} \mathcal{T} \subset W$

Improvement 1: subspaces instead of subsets

- Look instead for subspaces W of V such that
  - W can be generated by products only: $\operatorname{Span}(W \cap \mathcal{G}) = W$
  - only k products are required: $\dim W = k$
  - W contains the target space spanned by the target vectors: $T = \operatorname{Span} \mathcal{T} \subset W$
- Algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and T ⊂ W then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace({0})
```

Improvement 2: expanding the target space

Exploration:

Improvement 2: expanding the target space

Exploration:

- Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T

Improvement 2: expanding the target space

Exploration:

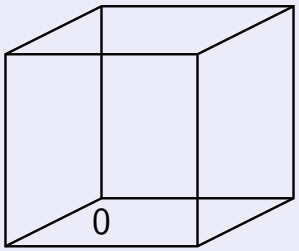
- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

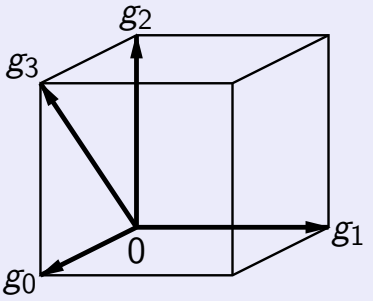


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

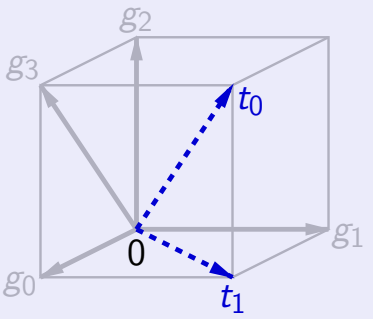


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

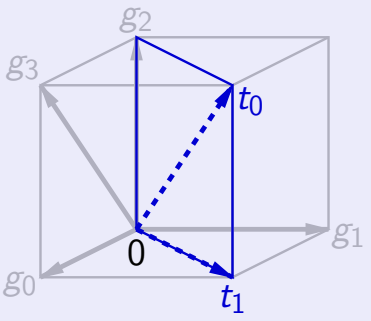


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

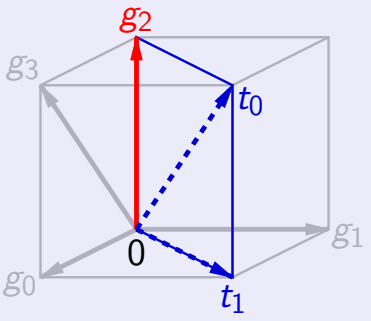


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

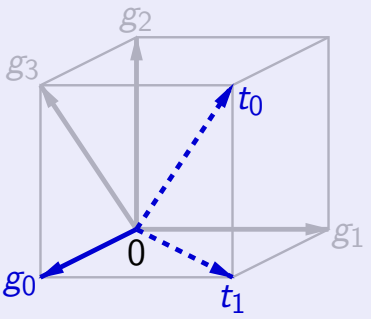


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

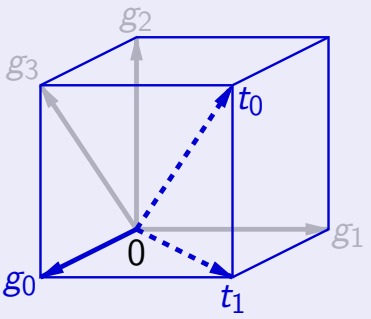


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

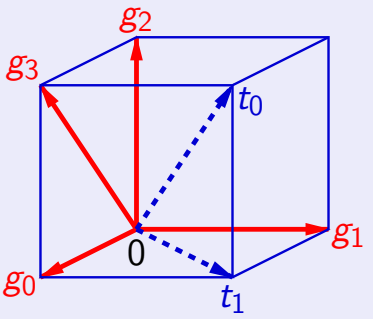


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example

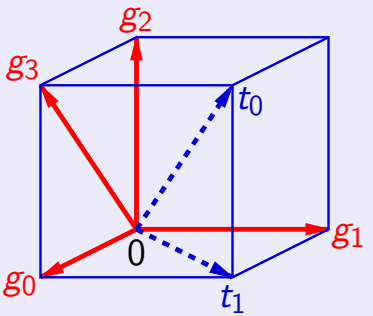


Improvement 2: expanding the target space

Exploration:

- ~~Choose a vector space W spanned by elements of $\mathcal{G}$ and test if it contains T~~
- Choose W containing T and detect all the vectors of $\mathcal{G}$ in W. Test if they are enough to form a basis of W

Example



Moreover, the Incomplete Basis Theorem allows us to explore only the spaces

$$W = T \oplus \operatorname{Span} \mathcal{W}' \text{ with } \mathcal{W}' \subset \mathcal{G}$$

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and T ⊂ W then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace({0})
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and T ⊂ W then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(~~{0}~~)
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and T ⊂ W then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and T ⊂ W then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and rk(W ∩ G) = k then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and rk(W ∩ G) = k then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and rk(W ∩ G) = k then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

- Complexity now depends on

$$\binom{\#\mathcal{G}}{k - \operatorname{rk} \mathcal{T}}$$

Improvement 2: expanding the target space

- Modified algorithm:

```
1: procedure expand_subspace(W)
2:     if dim W = k and rk(W ∩ G) = k then
3:         W is a solution
4:     else if dim W < k then
5:         for each g ∈ G \ W do
6:             expand_subspace(W ⊕ Span(g))
7: end procedure
8:
9: expand_subspace(T)
```

- Complexity now depends on

$$\binom{\#\mathcal{G}}{k - \operatorname{rk} \mathcal{T}}$$

- For instance for 5×5-term polynomial multiplication we have $\binom{961}{13-9} = 3.53 \cdot 10^{11}$ instead of $\binom{961}{13} = 8.82 \cdot 10^{28}$.

Example (continued)

2×3-term polynomial product in $\mathbb{F}_2[X]$

Example (continued)

2×3-term polynomial product in $\mathbb{F}_2[X]$

- Targets: $\mathcal{T} = \{a_1 b_2, a_1 b_1 + a_0 b_2, a_0 b_1 + a_1 b_0, a_0 b_0\}$
  - at least $\mathrm{rk}(\mathcal{T}) = 4$ products required

Example (continued)

2×3-term polynomial product in $\mathbb{F}_2[X]$

- Targets: $\mathcal{T} = \{a_1 b_2, a_1 b_1 + a_0 b_2, a_0 b_1 + a_1 b_0, a_0 b_0\}$
  - at least $\mathrm{rk}(\mathcal{T}) = 4$ products required
- Attempt with $k = 4$:
  - $W = T$
  - $T \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ \ a_1 \cdot b_2,\ \ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$

Example (continued)

2×3-term polynomial product in $\mathbb{F}_2[X]$

- Targets: $\mathcal{T} = \{a_1 b_2, a_1 b_1 + a_0 b_2, a_0 b_1 + a_1 b_0, a_0 b_0\}$
  - at least $\mathrm{rk}(\mathcal{T}) = 4$ products required
- Attempt with $k = 4$:
  - $W = T$
  - $T \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ \ a_1 \cdot b_2,\ \ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$
  - $\mathrm{rk}(T \cap \mathcal{G}) = 3 < k$
  - no solutions with $k = 4$ products only

Example (continued)

Attempt with $k = 5$:

- Try with $W = T \oplus \mathrm{Span}\{a_0 b_1\}$
- $W \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ a_1 \cdot b_0,\ (a_0 + a_1) \cdot b_0,$
  $a_0 \cdot b_1,\ a_1 \cdot b_2,\ (a_0 + a_1) \cdot (b_1 + b_2),$
  $a_0 \cdot (b_0 + b_1),\ a_1 \cdot (b_0 + b_2),\ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$

Example (continued)

Attempt with $k = 5$:

- Try with $W = T \oplus \mathrm{Span}\{a_0 b_1\}$
- $W \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ \ a_1 \cdot b_0,\ \ (a_0 + a_1) \cdot b_0,$
  $a_0 \cdot b_1,\ \ a_1 \cdot b_2,\ \ (a_0 + a_1) \cdot (b_1 + b_2),$
  $a_0 \cdot (b_0 + b_1),\ \ a_1 \cdot (b_0 + b_2),\ \ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$
- $\mathrm{rk}\,(W \cap \mathcal{G}) = 5 = k$, W is a solution!

Example (continued)

Attempt with $k = 5$:

- Try with $W = T \oplus \mathrm{Span}\{a_0 b_1\}$
- $W \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ \ a_1 \cdot b_0,\ \ (a_0 + a_1) \cdot b_0,$
  $a_0 \cdot b_1,\ \ a_1 \cdot b_2,\ \ (a_0 + a_1) \cdot (b_1 + b_2),$
  $a_0 \cdot (b_0 + b_1),\ \ a_1 \cdot (b_0 + b_2),\ \ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$
- $\mathrm{rk}\,(W \cap \mathcal{G}) = 5 = k$, W is a solution!
- $\{a_0 b_0, a_1 b_0, a_0 b_1, a_1 b_2, (a_0 + a_1)(b_1 + b_2)\}$ is a basis of W, and gives a formula

Example (continued)

Attempt with $k = 5$:

- Try with $W = T \oplus \mathrm{Span}\{a_0 b_1\}$
- $W \cap \mathcal{G} = \{\ a_0 \cdot b_0,\ \ a_1 \cdot b_0,\ \ (a_0 + a_1) \cdot b_0,$
  $a_0 \cdot b_1,\ \ a_1 \cdot b_2,\ \ (a_0 + a_1) \cdot (b_1 + b_2),$
  $a_0 \cdot (b_0 + b_1),\ \ a_1 \cdot (b_0 + b_2),\ \ (a_0 + a_1) \cdot (b_0 + b_1 + b_2)\ \}$
- $\mathrm{rk}\,(W \cap \mathcal{G}) = 5 = k$, W is a solution!
- $\{a_0 b_0, a_1 b_0, a_0 b_1, a_1 b_2, (a_0 + a_1)(b_1 + b_2)\}$ is a basis of W, and gives a formula
- 3 such solution spaces $\Rightarrow$ 162 bases $\Rightarrow$ 162 formulae

A generic algorithm

- This algorithm works for every bilinear maps:
  - short products, middle products, cross products
  - multiplication in complexes, quaternions, field extensions, matrices
  - multiplication of sparse polynomials and matrices
  - etc.

A generic algorithm

- This algorithm works for every bilinear maps:
  - short products, middle products, cross products
  - multiplication in complexes, quaternions, field extensions, matrices
  - multiplication of sparse polynomials and matrices
  - etc.
- The quadratic forms form a vector space too.
  - simply requires extending the definition of $\mathcal{G}$:

$$\mathcal{G} = \{(\sum \alpha_i a_i) \cdot (\sum \alpha'_i a_i) \mid (\alpha_0, \ldots, \alpha_{n-1}) \preccurlyeq_{\text{lex}} (\alpha'_0, \ldots, \alpha'_{n-1})\} \setminus \{0\}$$

  - example: squaring of a 2-term polynomial in $\mathbb{F}_3[X]$

$$\begin{aligned}\mathcal{G} = \{\ & a_0 \cdot a_0, \\ & a_0 \cdot a_1, && a_1 \cdot a_1, \\ & a_0 \cdot (a_0 + a_1), && a_1 \cdot (a_0 + a_1), && (a_0 + a_1) \cdot (a_0 + a_1), \\ & a_0 \cdot (a_0 - a_1), && a_1 \cdot (a_0 - a_1), && (a_0 + a_1) \cdot (a_0 - a_1), && (a_0 - a_1) \cdot (a_0 - a_1)\ \}\end{aligned}$$

Outline of the talk

- Problem description
- Resolution methods
- Results

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

Example of symmetric product: $(a_0 + a_2 + a_5) \cdot (b_0 + b_2 + b_5)$

n × m-term polynomial multiplication

Ring	n × m	#$\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$F_2[X]$	2 × 2	9	3	1	1	1	0.00
	3 × 3	49	6	9	3	9	0.00
	4 × 4	225	9	$6.60 \cdot 10^3$	4	4	0.03
	5 × 5	961	13	$9.65 \cdot 10^9$	27	27	$2.28 \cdot 10^5$
	6 × 6	3 969	14	$4.37 \cdot 10^9$	—	—	$6.03 \cdot 10^5$
	6 × 6	(Sym.) 63	17	$8.08 \cdot 10^6$	6	54	17.7
	7 × 7	(Sym.) 127	22	$3.38 \cdot 10^{12}$	2 618	19 550	$1.59 \cdot 10^7$
$F_3[X]$	2 × 2	16	3	1	1	4	0.00
	3 × 3	169	6	24	22	1 493	0.00
	4 × 4	1 600	9	$4.11 \cdot 10^5$	726	50 640	14.9
	5 × 5	14 641	11	$4.89 \cdot 10^7$	—	—	$4.02 \cdot 10^4$
		(Sym.) 121	12	$3.93 \cdot 10^4$	31	6 460	0.14
	6 × 6	(Sym.) 364	15	$2.37 \cdot 10^8$	4	1 024	$3.79 \cdot 10^3$
	7 × 7	(Sym.) 1 093	17	$2.69 \cdot 10^{10}$	—	—	$1.50 \cdot 10^6$

Multiplication in small finite-field extensions

Finite field	$\#\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$\mathbf{F}_{2^2}$	9	3	3	3	3	0.00
$\mathbf{F}_{2^3}$	49	6	$7.03 \cdot 10^3$	105	147	0.01
$\mathbf{F}_{2^4}$	225	9	$2.57 \cdot 10^9$	2 025	2 025	$1.13 \cdot 10^4$
$\mathbf{F}_{2^5}$	961	9	$3.10 \cdot 10^{10}$	—	—	$8.11 \cdot 10^5$
	(Sym.) 31	13	$3.49 \cdot 10^6$	2 015	2 015	6.24
$\mathbf{F}_{2^6}$	(Sym.) 63	15	$2.21 \cdot 10^{10}$	21	21	$6.63 \cdot 10^4$
$\mathbf{F}_{2^7}$	(Sym.) 127	15	$1.34 \cdot 10^{12}$	—	—	$6.17 \cdot 10^6$
$\mathbf{F}_{3^2}$	16	3	3	4	16	0.00
$\mathbf{F}_{3^3}$	169	6	$2.42 \cdot 10^5$	11 843	105 963	1.08
$\mathbf{F}_{3^4}$	1 600	8	$2.27 \cdot 10^{11}$	—	—	$1.08 \cdot 10^7$
	(Sym.) 40	9	$1.10 \cdot 10^5$	234	615 240	0.45
$\mathbf{F}_{3^5}$	(Sym.) 121	11	$2.66 \cdot 10^9$	121	121	$1.45 \cdot 10^4$
$\mathbf{F}_{3^6}$	(Sym.) 364	12	$3.01 \cdot 10^{12}$	—	—	$4.50 \cdot 10^7$

Multiplication in small finite-field extensions

Finite field	$\#\mathcal{G}$	k	# of tests	# of solutions	# of formulae	Calculation time [s]
$\mathbf{F}_{2^2}$	9	3	3	3	3	0.00
$\mathbf{F}_{2^3}$	49	6	$7.03 \cdot 10^3$	105	147	0.01
$\mathbf{F}_{2^4}$	225	9	$2.57 \cdot 10^9$	2 025	2 025	$1.13 \cdot 10^4$
$\mathbf{F}_{2^5}$	961	9	$3.10 \cdot 10^{10}$	—	—	$8.11 \cdot 10^5$
	(Sym.) 31	13	$3.49 \cdot 10^6$	2 015	2 015	6.24
$\mathbf{F}_{2^6}$	(Sym.) 63	15	$2.21 \cdot 10^{10}$	21	21	$6.63 \cdot 10^4$
$\mathbf{F}_{2^7}$	(Sym.) 127	15	$1.34 \cdot 10^{12}$	—	—	$6.17 \cdot 10^6$
$\mathbf{F}_{3^2}$	16	3	3	4	16	0.00
$\mathbf{F}_{3^3}$	169	6	$2.42 \cdot 10^5$	11 843	105 963	1.08
$\mathbf{F}_{3^4}$	1 600	8	$2.27 \cdot 10^{11}$	—	—	$1.08 \cdot 10^7$
	(Sym.) 40	9	$1.10 \cdot 10^5$	234	615 240	0.45
$\mathbf{F}_{3^5}$	(Sym.) 121	11	$2.66 \cdot 10^9$	121	121	$1.45 \cdot 10^4$
$\mathbf{F}_{3^6}$	(Sym.) 364	12	$3.01 \cdot 10^{12}$	—	—	$4.50 \cdot 10^7$

Conclusion

- General algorithm (polynomials, matrices, squaring, etc.)
- Method that proves lower bounds on the number of subproducts (no heuristics used)
- Gives all formulae
  - Provides new formulae that cannot be found with previous methods
  - We can cherry-pick the one with minimum number of additions and scalar multiplications

Conclusion

- General algorithm (polynomials, matrices, squaring, etc.)
- Method that proves lower bounds on the number of subproducts (no heuristics used)
- Gives all formulae
  - Provides new formulae that cannot be found with previous methods
  - We can cherry-pick the one with minimum number of additions and scalar multiplications
- Work in progress and perspectives
  - Lifting formulae for higher-characteristic or characteristic-0 fields
  - Find formulae for your favorite bilinear application!

3×3 polynomial multiplication in odd characteristic

Compute the following products:

$$\begin{aligned}
g_0 &= a_0 \cdot b_0 \\
g_1 &= a_2 \cdot b_2 \\
g_2 &= (a_0 + a_1) \cdot (b_0 + b_2) \\
g_3 &= (a_0 + a_2) \cdot (b_1 + b_2) \\
g_4 &= (a_1 + a_2) \cdot (b_0 + b_1) \\
g_5 &= (a_0 + a_1 + a_2) \cdot (b_0 + b_1 + b_2).
\end{aligned}$$

Reconstruct the coefficients of the result.
$(a_0 + a_1X + a_2X^2) \cdot (b_0 + b_1X + b_2X^2) =$
$g_1X^4 + (g_0 + g_3 + g_4 + g_5)X^3 + (g_2 + g_3 + g_5)X^2 + (g_1 + g_2 + g_4 + g_5)X + g_0$