FUNCTIONAL DECOMPOSITION OF A CLASS OF WILD POLYNOMIALS

ROBERT S. COULTER, GEORGE HAVAS AND MARIE HENDERSON

Dedicated to Professor Anne Penfold Street.

Abstract. No general algorithm is known for the functional decomposition of wild polynomials over a finite field. However partial solutions exist. In particular, a fast functional decomposition algorithm for linearised polynomials has been developed using factoring methods in skew-polynomial rings. This algorithm is extended to a related class of wild polynomials, which are sub-linearised polynomials.

1. Introduction

Let $\mathbb{F}_q$ represent the finite field with $q = p^e$ elements where p is prime. A polynomial $f \in \mathbb{F}_q[X]$ is called *wild* if p divides the degree of f and *tame* otherwise. The polynomial $f = g \circ h = g(h)$ is the functional *composition* of g and h. The polynomials g and h are called left and right *composition factors* of f respectively. Note that $degree(f) = degree(g)\, degree(h)$. A polynomial $f \in \mathbb{F}_q[X]$ is called *indecomposable* over $\mathbb{F}_q$ if for each decomposition $f = g(h)$ where $g, h \in \mathbb{F}_q[X]$ then either $degree(g) \leq 1$ or $degree(h) \leq 1$. A decomposition $f = f_1 \circ \cdots \circ f_k$ is called a *complete* decomposition of f if each f_i, $1 \leq i \leq k$ is indecomposable and $degree(f_i) \geq 1$. As $(aX + b), a^{-1}(X - b) \in \mathbb{F}_q[X]$ are inverses with respect to composition then

$$f(g(X)) = f(aX + b) \circ (a^{-1}(g(X) - b)).$$

Thus, any complete decomposition can be varied using linear compositions. In fact, even if we discount linear compositions, there is no truly unique complete decomposition for polynomials. However, certain restrictions do apply in the tame case.

In [7], Ritt shows that, over the complex numbers, any complete decomposition of f is unique in a certain sense. This is known as Ritt's first theorem. Fried and MacRae [3] extend Ritt's first theorem to include tame polynomials over finite fields. We state their result for finite fields only. Note that the full result in [3] is concealed within the text.

This work was partially supported by the Australian Research Council.

Theorem 1.1 (Fried & MacRae). *Let $f \in \mathbb{F}_q[X]$ where $(degree(f), p) = 1$. If*

$$f = g_1 \circ \cdots \circ g_r = h_1 \circ \cdots \circ h_s$$

are two complete decompositions of f over $\mathbb{F}_q$ then $r = s$ and there exists a permutation π of the symbols $1, \ldots, r$ such that $degree(g_i) = degree(h_{\pi(i)})$.

Therefore there is an "essentially" unique decomposition of $f \in \mathbb{F}_q[X]$ in the tame case. Ritt's first theorem can not be extended to the wild case. The following example supports this statement. It is a classic wild case example and was presented by Dorey and Whaples in [2].

Example 1.2. Let p be a positive prime integer and F any field of characteristic p. We have the following partial decompositions of $f \in F[X]$:

$$\begin{aligned} f(X) &= X^{p^3+p^2} - X^{p^3+1} - X^{p^2+p} + X^{p+1} \\ &= X^{p+1} \circ (X^p + X) \circ (X^p - X) \\ &= (X^{p^2} - X^{p^2-p+1} - X^p + X) \circ X^{p+1}. \end{aligned}$$

Evidently any polynomial of the shape $X^p + aX$ is indecomposable. If a polynomial of degree p^2 is decomposable then it is of the form

$$\begin{aligned} &(aX^p + bX^{p-1} + \cdots) \circ (cX^p + dX^{p-1} + \cdots) \\ &\qquad = ac^p X^{p^2} + (ad^p + bc^{p+1}) X^{p^2-p} + \text{lower degree terms} \end{aligned}$$

and so it can not contain an X^{p^2-p+1} term. Hence the polynomial $X^{p^2} - X^{p^2-p+1} - X^p + X$ is also indecomposable. The decompositions of X^{p+1} do not alter the fact that these two partial decompositions of f are distinct in ways not acceptable in Theorem 1.1. So in characteristic $p \neq 0$ it is not always true that two complete decompositions have the same number of components or that the sequence of degrees are the same except for order.

Joachim von zur Gathen looks at the functional decomposition of tame and wild type polynomials in the two separate articles: [8] and [9] respectively. It is stated in the conclusions of [9] that although a satisfactory solution for polynomial decomposition in the tame case has been found the wild case remains open. The tame case relies on the essential uniqueness of complete decomposition. In the wild case only partial solutions have been achieved, see [9] and [4] for results and further references. The difficulty of obtaining general results in the wild case arises from the absence of a uniqueness property.

A polynomial of the shape

$$L(X) = \sum_{i=0}^{n} a_i X^{p^i} \tag{1}$$

is called a *linearised polynomial*. More precisely, if there exists a positive integer s such that $a_i = 0$ unless s divides i then L is called a p^s-polynomial. For background material on linearised polynomials see [6, Chapter 3]. Let $L \in \mathbb{F}_q[X]$ be a p^s-polynomial and d be a divisor of $p^s - 1$. Then $L(X) = XT(X^d)$ where $T \in \mathbb{F}_q[X]$. The polynomial $S(X) = XT^d(X)$ is called a *sub-linearised polynomial*, or, again more precisely, the (p^s, d)-polynomial associated with L. We see that L and S are associated if and only if $L(X^d) = S^d(X)$. These two classes of polynomials are certainly of wild type as p^s divides their degree. In [4], Giesbrecht introduces a decomposition algorithm for linearised polynomials. Here, we extend this algorithm to cover (p, d)-polynomials. Note that, although every p^s-polynomial is a p-polynomial, a (p^s, d)-polynomial is only a (p, d)-polynomial if d divides $p-1$ (from the definition).

2. The decomposition algorithms

We extend the algorithms for linearised polynomial decomposition from [4] to the class of sub-linearised (p, d)-polynomials. In [4], two decomposition problems are considered:

The complete decomposition problem: given a non-constant $f \in \mathbb{F}_q[X]$, find indecomposable $f_1, \ldots, f_k \in \mathbb{F}_q[X]$ such that $f = f_1 \circ \cdots \circ f_k$.

The bi-decomposition problem: given a non-constant $f \in \mathbb{F}_q[X]$ and $n \in \mathbb{N}$ where $n <$ degree(f), determine if there exist $f_1, f_2 \in \mathbb{F}_q[X]$ such that $f = f_1(f_2)$ and degree$(f_2) = n$, and, if so, find f_1, f_2.

In [4], Giesbrecht obtains the following theoretical running times for linearised polynomials in regards to the above decomposition problems. The decomposition results for this wild class are actually applications of the central results of [4]: factoring in skew-polynomial rings. Note that $M(e) = e^2$ or $M(e) = e\log e\log\log e$ depending on the multiplication algorithm used and $MM(n) = n^3$ or $MM(n) = n^{2.376}$ depending on the matrix multiplication algorithm used (see the opening remarks of [4]).

Theorem 2.1 (Giesbrecht)**.** *Let $q = p^e$, $L \in \mathbb{F}_q[X]$ be a linearised polynomial of degree p^n, and $m = p^t < p^n$. We can produce a complete decomposition of L in $\mathbb{F}_q[X]$ and determine if there exist $L_1, L_2 \in \mathbb{F}_q[X]$ such that degree$(L_2) = m$ and $L = L_1(L_2)$, and, if so, find such L_1, L_2, with a deterministic algorithm requiring $(nep)^{O(1)}$ operations in $\mathbb{F}_p$ or a probabilistic algorithm requiring $O(n^4 eM(e) + n^3 e^2 M(e)\log e + nMM(ne)\log(ne)\log p)$ operations in $\mathbb{F}_p$.*

It is well known that the class of linearised polynomials is closed with respect to composition. In addition, the decomposition factors of a linearised polynomial are either linearised polynomials or transformations of

a linearised polynomial such as $L(X) + a \in \mathbb{F}_q[X]$. This was shown in [2]. See [1] for a succinct proof. This is a key point in Giesbrecht's application. It can be shown that the sub-linearised polynomials satisfy a similar property. For a proof of the next result see [5].

Theorem 2.2. *Let S be a (p^s, d)-polynomial which decomposes over $\mathbb{F}_q$ with $S = f_1(f_2)$ for some $f_1, f_2 \in \mathbb{F}_q[X]$. Then $S = f_1'(f_2')$ where $f_1'(X) = f_1\big(X + f_2(0)\big)$, $f_2'(X) = f_2(X) - f_2(0)$ and f_1' and f_2' are (p^r, d)-polynomials where r divides s.*

Therefore, if the (p^s, d)-polynomial S decomposes it can be written as the composition of (p^r, d)-polynomials where r divides s. Note that the behaviour is not as "free" as the linearised case as the value d is set and must divide $p^r - 1$. The next theorem, also taken from [5], connects the composition behaviour of the linearised and sub-linearised polynomial classes.

Theorem 2.3. *(i) Let L_1 and L_2 be p^s-polynomials over $\mathbb{F}_q$ and S_1 and S_2 be the associated (p^s, d)-polynomials, respectively. Then $S_1(S_2)$ is the (p^s, d)-polynomial associated with $L_1(L_2)$.*
(ii) Let $L \in \mathbb{F}_q[X]$ be a p^s-polynomial and S be the associated (p^s, d)-polynomial. Let r be some integer where r divides s and d divides $p^r - 1$. Then $L = L_1(L_2)$ for p^r-polynomials L_1 and L_2 if and only if $S = S_1(S_2)$ for (p^r, d)-polynomials S_1 and S_2 and $L_i^d(x) = S_i(x^d)$, $i = 1, 2$.

Part (ii) of this theorem suits our purpose as it links the decomposition of the two classes. In part (ii) we see that the decomposition factors of a p^s-polynomial are p^r-polynomials where r divides s. It may be the case that $r < s$. This is possible as coefficients of p^r terms may be annihilated in the composition. This is an important point as a p^s-polynomial may decompose into p^r-polynomials but the associated (p^s, d)-polynomial can not correspondingly decompose unless d divides $p^r - 1$ (as we have already indicated). Therefore, to decompose a (p^s, d)-polynomial S, the first step is to find the least positive integer r such that d divides $p^r - 1$ and reinterpret S as a (p^r, d)-polynomial.

There is an even more subtle and important point concealed here. Suppose that $L = L_1(L_2) = L_3(L_4)$ where L, L_1, L_2 are p^r-polynomials but L_3, L_4 are p^t-polynomials for some t dividing r. Giesbrecht's decomposition algorithms may return the composition factors L_3, L_4 which do not correspond to a decomposition of the associated (p^r, d)-polynomial S. Thus, we can not decide, on the evidence of Giesbrecht's algorithm, that the polynomial S is indecomposable. This problem will not arise when $s = 1$, the class of (p, d)-polynomials. For this reason, we restrict ourselves to (p, d)-polynomials. We can make another simple restriction using the next theorem.

Lemma 2.4. *Let $L \in \mathbb{F}_q[X]$ be as in (1) with degree > 1. Let $a_i = 0$ for $0 \leq i \leq (m-1)$ and $a_m \neq 0$. Then $L(X) = L_1(X) \circ X^{p^m}$ and L_1 is a p-polynomial. If S is the associated (p,d)-polynomial to L then $S(X) = S_1(X) \circ X^{p^m}$ where $L_1(X^d) = S_1^d(X)$.*

Proof. To see the first part put $L_1(X) = \sum_{i=0}^{n-m} a_{i+m} X^{p^i}$. Now

$$\begin{aligned} S_1(X) \circ X^{p^m} &= X^{p^m} \left(\sum_{i=0}^{n-m} a_{i+m} (X^{p^m})^{(p^i-1)/d} \right)^d \\ &= X \left(\sum_{i=0}^{n-m} a_{i+m} X^{(p^m-1+p^{m+i}-p^m)/d} \right)^d \\ &= X \left(\sum_{i=0}^{n} a_i X^{(p^i-1)/d} \right)^d \end{aligned}$$

which is $S(X)$. □

Thus we may assume that the (p,d)-polynomial S to be decomposed has a non-zero coefficient of X. Otherwise we can perform the simple decomposition in Lemma 2.4 and consider S_1 instead.

To solve either of the problems listed, we need to convert a sub-linearised polynomial S to a linearised polynomial L, call the relevant algorithm by Giesbrecht, and then convert the output back into sub-linearised polynomials.

Algorithm: Complete-decomposition
Input: A (p,d)-polynomial $S \in \mathbb{F}_q[X]$ and the integer d.
Output: Indecomposable (p,d)-polynomials $S_1, \ldots, S_k \in \mathbb{F}_q[X]$ where $S = S_1 \circ \cdots \circ S_k$.

1. Convert S to a linearised polynomial L.
2. Find indecomposable $L_1, \ldots, L_k \in \mathbb{F}_q[X]$ satisfying $L = L_1 \circ \cdots \circ L_k$ using Giesbrecht's algorithm;
3. Convert each L_j, $1 \leq j \leq k$ to respective (p,d)-polynomials S_j and return $S_1, \ldots, S_k$.

Algorithm: Bi-decomposition
Input: A (p,d)-polynomial $S \in \mathbb{F}_q[X]$ of degree p^m, the integer d, and a positive integer $n = p^k$.
Output: $S_1, S_2 \in \mathbb{F}_q[X]$ with degree$(S_2) = n$ and $S = S_1(S_2)$, or a message that no such bi-decomposition exists.

1. Convert S to a linearised polynomial L.
2. Using Giesbrecht's algorithm, determine if there exist p-polynomials $L_1, L_2 \in \mathbb{F}_q[X]$ satisfying $L = L_1(L_2)$ and degree$(L_2) = n$. If L_1, L_2

are found then convert L_1 and L_2 to S_1 and S_2 and return them. Otherwise return "S has no such bi-decomposition".

So, to extend Giesbrecht's algorithms we make conversions from (p,d)-polynomials S to p-polynomials L and back again. These operations are presented in the following two algorithms.

Algorithm: A (convert S to L)
Input: A (p,d)-polynomial $S \in \mathbb{F}_q[X]$
Output: An associated p-polynomial $L \in \mathbb{F}_q[X]$

1. Calculate $f(X) = S(X^d)$;
2. Differentiate $f(X)$ repeatedly $d-1$ times to obtain $f^{[d-1]}(X)$;
3. Change $f^{[d-1]}(X)$ to a monic to obtain L.

Algorithm: B (convert L to S)
Input: A p-polynomial $L \in \mathbb{F}_q[X]$
Output: The associated (p,d)-polynomial $S \in \mathbb{F}_q[X]$

1. Calculate $T(Y) = L(X)/X$ where $Y = X^d$;
2. Return $S(X) = XT^d(X)$.

Recall that if the p-polynomial L and (p,d)-polynomial S are associated, then $L(X) = XT(X^d)$ and $S(X) = XT^d(X)$ for some $T \in \mathbb{F}_q[X]$. It is evident from this that Algorithm B produces the desired polynomial. In Algorithm A, $f(X) = S(X^d) = X^d T^d(X^d) = L^d(X)$. As $a_0 \neq 0$ then the $(d-1)$th derivative of f is $d! a_0^{d-1} L(X)$.

For the complexity analysis, note that d is $O(p)$. In Algorithm A, steps 1, 2 and 3 will require $O(pn)$, $O(p^2 n)$ and $O(ne)$ operations in $\mathbb{F}_p$, respectively. This gives an overall cost of $O((p^2+e)n)$ operations. For Algorithm 2, step 1 requires $O(n)$ operations. The number of operations required for step 2 is $O(\log(p)M(npe))$. All of the costs for the two conversion algorithms are contained, and thus absorbed, in the complexity of the algorithm of Giesbrecht. Thus the extension of Giesbrecht's algorithm to (p,d)-polynomials is asymptotically free. Note that, if the input for the decomposition algorithms was changed to be $T(X)$ and d, rather than $S(X)$ and d, Algorithm A would be redundant as conversion would be a simple task.

3. Conclusion

In general, it has proved difficult to obtain decomposition results in the wild case. We have presented a straightforward extension of a decomposition algorithm of the linearised polynomials to a related class of wild polynomials: the (p,d)-polynomials. This increases the range of applicability of Giesbrecht's algorithm by the number of divisors of $p-1$. Effectively, this has been done without cost. It may also be possible to extend the results

from [4], by less direct methods, to the larger class of (p^s, d)-polynomials. Improvements in the average running time may also be gained on the algorithms of [4] and those given here by applying other results from [5]. We note that the methods employed here to decompose (p, d)-polynomials using Giesbrecht's algorithm could be used to decompose (p, d)-polynomials using any algorithm which decomposes linearised polynomials.

References

1. S.D. Cohen, *The irreducibility of compositions of linear polynomials over a finite field*, Compositio Math. **47** (1982), 149–152.
2. F. Dorey and G. Whaples, *Prime and composite polynomials*, J. Algebra **28** (1974), 88–101.
3. M.D. Fried and R.E. MacRae, *On the invariance of chains of fields*, Illinois J. Math. **13** (1969), 165–171.
4. M. Giesbrecht, *Factoring in skew-polynomial rings over finite fields*, J. Symbolic Computation (to appear).
5. M. Henderson and R. Matthews, *Composition behaviour of linearised and sublinearised polynomials over a finite field*, preprint.
6. R. Lidl and H. Niederreiter, *Finite Fields*, Encyclopedia Math. Appl., vol. 20, Addison-Wesley, Reading, 1983, (now distributed by Cambridge University Press).
7. J.F. Ritt, *Prime and composite polynomials*, Trans. Amer. Math. Soc. **23** (1922), 51–66.
8. J. von zur Gathen, *Functional decomposition of polynomials: the tame case*, J. Symbolic Computation **9** (1990), 281–299.
9. ———, *Functional decomposition of polynomials: the wild case*, J. Symbolic Computation **10** (1990), 437–452.

Centre for Discrete Mathematics and Computing, Department of Computer Science and Electrical Engineering, The University of Queensland, Queensland, 4072, Australia