Generalized Sudan's list decoding for order domain codes*

Olav Geil[1] and Ryutaroh Matsumoto[2]

[1] Department of Mathematical Sciences, Aalborg University, Denmark
olav@math.aau.dk,
[2] Department of Communications and Integrated Systems, Tokyo Institute of Technology, Japan
ryutaroh@it.ss.titech.ac.jp

Abstract. We generalize Sudan's list decoding algorithm without multiplicity to evaluation codes coming from arbitrary order domains. The number of correctable errors by the proposed method is larger than the original list decoding without multiplicity.

1 Introduction

Høholdt et al. [5] proposed the new framework for algebraic code construction, which they called *evaluation codes*. Evaluation codes are defined by either generator matrices or parity check matrices. Evaluation codes defined by parity check matrices include many classes of algebraic codes, including generalized Reed-Muller, Reed-Solomon, and one-point geometric Goppa codes $C_\Omega(D, G)$, and they provided lower bounds on the minimum Hamming distance and decoding algorithms in a unified manner, while relatively little work was done for evaluation codes defined by generator matrices in [5]. The framework of evaluation codes and order domains was later generalized by O'Sullivan [6], Geil and Pellikaan [3].

Andersen and Geil [1] studied the evaluation codes defined by generator matrices, which also include generalized Reed-Muller, Reed-Solomon, and one-point geometric Goppa codes $C_{\mathcal{L}}(D, G)$, and they also provided lower bounds on the minimum Hamming distance in a unified manner. Their work [1] can be regarded as a generator matrix counterpart of [5]. In this paper we study evaluation codes defined by generator matrices.

On the other hand, Sudan [9] and Guruswami-Sudan [4] proposed the list decoding algorithms for Reed-Solomon and one-point geometric Goppa codes, and the latter method dramatically increased the number of correctable errors of the conventional bounded distance decoding algorithm, such as the Berlekamp-Massey algorithm. Following those work, Shokrollahi and Wasserman [8] generalized the Sudan method [9] to one-point geometric Goppa codes, and Pellikaan

* This research is in part supported by the Danish National Science Research Council Grant FNV-21040368 and the MEXT 21st Century COE Program: Photonics Nanodevice Integration Engineering.

and Wu [7] generalized the Guruswami-Sudan method [4] to generalized Reed-Muller codes as the first algorithm among three new list decoding algorithms in [7]. Augot and Stepanov improved the estimation of error-correcting capability of the first algorithm in [7].

However, up to now, nobody has successfully generalized the list decoding algorithms [9, 4] to evaluation codes from arbitrary order domains. The difficulty lies in the fact that existing methods [9, 4, 7] deal with codes coming from polynomial rings or their factor rings and utilize their polynomial structure such as the degree of a polynomial and the pole order of an algebraic function.

We will distill essential ingredients from Sudan's original decoding method [9], which allow us to carry over it to evaluation codes from arbitrary order domains. After that, we examine the error-correcting capability of the proposed generalization when we apply it to generalized Reed-Muller and one-point geometric Goppa codes, and show that the proposed method can correct more errors than [8] and the first algorithm in [7]. We have to note that the proposed method usually cannot correct more errors than the Guruswami-Sudan method [4] with multiplicity.

The paper is organized as follows. In Section 2 we present the modified Sudan decoding algorithm without multiplicity. Our description does not require that the reader has any previous experience with order domains. Some knowledge about generalized Reed-Muller and one-point geometric Goppa codes should do. In Section 3 we study decoding of generalized Reed-Muller codes. We compare our findings to the results by the first algorithm of Pellikaan and Wu in [7] and by Augot and Stepanov in [2]. Then in Section 4 we apply our method to some codes coming from norm-trace curves.

2 Decoding of order domain codes

In this section we state the modified decoding algorithm for a large family of codes defined from order domains. We provide translations into the case of generalized Reed-Muller codes and one-point geometric Goppa codes. Our presentation relies on [1, 3, 6].

Definition 1. *Let R be an $\mathbb{F}_q$-algebra and let Γ be a subsemigroup of $\mathbb{N}_0^r$ for some r. Let $\prec_{\mathbb{N}_0^r}$ be a monomial ordering on $\mathbb{N}_0^r$. A surjective map $\rho : R \to \Gamma_{-\infty} := \Gamma \cup \{-\infty\}$ that satisfies the following six conditions is said to be a weight function*

(*W*.0) $\rho(f) = -\infty$ *if and only if* $f = 0$
(*W*.1) $\rho(af) = \rho(f)$ *for all nonzero* $a \in \mathbb{F}_q$
(*W*.2) $\rho(f+g) \preceq_{\mathbb{N}_0^r} \max\{\rho(f), \rho(g)\}$ *and equality holds when* $\rho(f) \prec_{\mathbb{N}_0^r} \rho(g)$
(*W*.3) *If* $\rho(f) \prec_{\mathbb{N}_0^r} \rho(g)$ *and* $h \neq 0$, *then* $\rho(fh) \prec_{\mathbb{N}_0^r} \rho(gh)$
(*W*.4) *If f and g are nonzero and* $\rho(f) = \rho(g)$, *then there exists a nonzero* $a \in \mathbb{F}_q$ *such that* $\rho(f - ag) \prec_{\mathbb{N}_0^r} \rho(g)$
(*W*.5) *If f and g are nonzero then* $\rho(fg) = \rho(f) + \rho(g)$.

An $\mathbb{F}_q$-algebra with a weight function is called an order domain over $\mathbb{F}_q$. The triple (R, ρ, Γ) is called an order structure and Γ is called the value semigroup of ρ.

We have the following two standard examples of weight functions.

Example 1. Consider the polynomial ring $R = \mathbb{F}_q[X_1, \ldots, X_m]$ and let $\prec_{\mathbb{N}_0^m}$ be the graded lexicographic ordering on $\mathbb{N}_0^m$ given by $(i_1, \ldots, i_m) \prec_{\mathbb{N}_0^m} (j_1, \ldots, j_m)$ if either $i_1 + \cdots + i_m < j_1 + \cdots + j_m$ holds or $i_1 + \cdots + i_m = j_1 + \cdots + j_m$ holds, but left most non-zero entry of $j_1 - i_1, \ldots, j_m - i_m)$ is positive. The map $\rho : R \to \mathbb{N}_0^m \cup \{-\infty\}$, $\rho(F) := \max_{\prec_{\mathbb{N}_0^m}} \{(i_1, \ldots, i_m) \mid X_1^{i_1} \cdots X_m^{i_m} \in \mathrm{Supp}(F)\}$ if $F \neq 0$ and $\rho(0) := -\infty$ is a weight function.

Example 2. Let Q be a rational place of a function field in one variable over $\mathbb{F}_q$. Then $R = \cup_{m=0}^{\infty} \mathcal{L}(mQ)$ is an order domain with a weight function given by $\rho(f) = -\nu_Q(f)$. Clearly, in this case the value semigroup Γ is simply the Weierstrass semigroup corresponding to Q and the monomial ordering is the unique monomial ordering on $\mathbb{N}_0$.

For the code construction we will need a few results.

Theorem 1. *Let (R, ρ, Γ) be an order structure. Then any set $\mathcal{B} = \{f_\gamma \mid \rho(f_\gamma) = \gamma\}_{\gamma \in \Gamma}$ constitutes a basis for R as a vector space over $\mathbb{F}_q$. In particular $\{f_\lambda \in \mathcal{B} \mid \lambda \preceq \gamma\}$ constitutes a basis for $R_\gamma := \{f \in R \mid \rho(f) \preceq \gamma\}$.*

A basis as in Theorem 1 is known in the literature as a well-behaving basis. In the remaining part of this section we will always assume that some fixed well-behaving basis has been chosen for the order domain under consideration.

Definition 2. *Let R be an $\mathbb{F}_q$-algebra. A surjective map $\varphi : R \to \mathbb{F}_q^n$ is called a morphism of $\mathbb{F}_q$-algebras if φ is $\mathbb{F}_q$-linear and $\varphi(fg) = \varphi(f) * \varphi(g)$ for all $f, g \in R$, where $*$ denotes the componentwise multiplication of two vectors.*

The class of codes $E(\lambda)$ below includes as we shall recall generalized Reed-Muller codes as well as one-point geometric Goppa codes.

Definition 3. *Consider an order domain R over $\mathbb{F}_q$ and a corresponding morphism $\varphi : R \to \mathbb{F}_q^n$. For $\lambda \in \Gamma$ we define $E(\lambda) := \varphi(R_\lambda)$.*

Example 3. This is a continuation of Example 1. Consider $\mathbb{F}_q^{q^m} = \{P_1, \ldots, P_{q^m}\}$ and let $\varphi : \mathbb{F}_q[X_1, \ldots, X_m] \to \mathbb{F}_q^{q^m}$ be given by $\varphi(F) = (F(P_1) \ldots, F(P_{q^m}))$. If we choose $\lambda = (u, 0, \ldots, 0)$ then $E(\lambda)$ is simply the generalized Reed-Muller code $\mathrm{RM}_q(u, m)$ no matter how the well-behaving basis for the order domain $R = \mathbb{F}_q[X_1, \ldots, X_m]$ has been chosen. For simplicity we choose in this paper always the well-behaving basis $\mathcal{B}$ to be the set of monomials in $X_1, \ldots, X_m$.

Example 4. This is a continuation of Example 2. Let $\{P_1, \ldots, P_n\}$ be rational places different from Q and consider the morphism $\varphi : R \to \mathbb{F}_q^n$ given by $\varphi(f) = (f(P_1), \ldots, f(P_n))$. The code $E(\lambda)$ is the one-point geometric Goppa code $C_{\mathcal{L}}(D, \lambda Q)$ where $D = P_1 + \cdots + P_n$.

We next consider some terminology from [1].

Definition 4. *Let* $\alpha(1) := 0$ *and define for* $i = 2, 3, \ldots, n$ *recursively* $\alpha(i)$ *to be the smallest element in* Γ *that is greater than* $\alpha(1), \alpha(2), \ldots, \alpha(i-1)$ *and satisfies* $\varphi(R_\gamma) \subsetneq \varphi(R_{\alpha(i)})$ *for all* $\gamma < \alpha(i)$. *Write* $\Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$.

Definition 5. *For* $\eta \in \Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$ *define*

$$M(\eta) := (\eta + \Gamma) \cap \Delta(R, \rho, \varphi)$$

where $\eta + \Gamma$ *means* $\{\eta + \lambda \mid \lambda \in \Gamma\}$. *Let* $\sigma(\eta) := \#M(\eta)$.

The first part of the following theorem plays a fundamental role in our modification of the Sudan decoding algorithm without multiplicity.

Theorem 2. *If* $\boldsymbol{c} \in E(\lambda)$ *but* $\boldsymbol{c} \notin E(\eta)$ *for any* η *with* $\eta \prec_{\mathbb{N}_0^r} \lambda$ *then* $w_H(\boldsymbol{c}) \geq \sigma(\lambda)$ *holds. In particular we have* $d(E(\lambda)) \geq \min\{\sigma(\eta) \mid \eta \in \Delta(R, \rho, \varphi), \eta \preceq \lambda\}$.

Example 5. The above bound gives the true minimum distances of generalized Reed-Muller codes and of Hermitian codes. For the case of one-point geometric Goppa codes the bound is an improvement to the usual bound by Goppa which states that the minimum distance of a one-point geometric Goppa code $C_{\mathcal{L}}(D, \lambda Q)$ is at least $n - \lambda$. More precisely, we have $\sigma(\lambda) \geq n - \lambda$ for any $\lambda \in \Delta(R, \rho, \varphi)$. For high dimensions the inequality is in general sharp.

Theorem 2 suggests the following improved code construction.

Definition 6. *Given any fixed basis* $\mathcal{B} = \{f_\gamma \mid \rho(f_\gamma) = \gamma\}_{\gamma \in \Gamma}$ *as in Theorem 1 we define* $\tilde{E}(\delta) := \mathit{Span}_{\mathbb{F}_q}\{\varphi(f_{\alpha(i)}) \mid \alpha(i) \in \Delta(R, \rho, \varphi)$ *and* $\sigma(\alpha(i)) \geq \delta\}$.

We have

Theorem 3. $d(\tilde{E}(\delta)) \geq \delta$.

The codes $\tilde{E}(\delta)$ are sometimes very much better than the corresponding codes $E(\lambda)$. This is for instance the case for the improved generalized Reed-Muller codes known as hyperbolic codes (or Massey-Costello-Justesen codes). Regarding one-point geometric Goppa codes the picture very much relies on which particular curve we consider, but the improvement may also in this case be significant. The idea of controlling the minimum distance of a code by choosing the functions f_λ to be used in the code construction in a clever way will be one of the main ingredients of our modified Sudan decoding algorithm without multiplicity.

We now describe the modified Sudan decoding algorithm without multiplicity for the codes $E(\lambda)$ and $\tilde{E}(\delta)$. To ease notation we state the algorithm for a larger class of codes, namely for any code C of the form

$$C = \mathrm{Span}_{\mathbb{F}_q}\{\varphi(f_{\lambda_1}), \ldots, \varphi(f_{\lambda_k})\} \text{ where } \{\lambda_1, \ldots, \lambda_k\} \subseteq \Delta(R, \rho, \varphi). \quad (1)$$

The first part of the decoding algorithm is to find a proper interpolation polynomial $Q(Z)$ with coefficients from the order domain R. To set up the decoding procedure for a given fixed code C we first need to describe sets from which we will allow the coefficients to be chosen. To this end consider the following definition.

Definition 7. *Given a code C as above let E be some fixed value (representing the number of errors we would like to correct). For $s \in \mathbb{N}_0$ define*

$$L(E,s) := \{\lambda \in \Delta(R,\rho,\varphi) \mid \textit{for all } \ i_1,\ldots,i_s \in \{1,\ldots,k\} \ \textit{ we have}$$
$$f_\lambda \prod_{v=1}^{s} f_{\lambda_{i_v}} \in \mathit{Span}\{f_{\alpha(1)},\ldots,f_{\alpha(n)}\} \textit{ and} \tag{2}$$
$$\sigma(\lambda_i) > E \textit{ for all } \ f_{\lambda_i} \in \mathit{Supp}_{\mathcal{B}}(f_\lambda \prod_{v=1}^{s} f_{\lambda_{i_v}})\} \tag{3}$$

Note, that there is no requirement that $i_1,\ldots,i_s$ are pairwise different. Note also that the set $L(E,s)$ relies on the actual choice of well-behaving basis $\{f_\lambda\}_{\lambda\in\Gamma}$. Further we observe that for large values of s we have $L(E,s) = \emptyset$. What we will need for the modified version of Sudan type decoding without multiplicity to work is a number E such that $\sum_{s=0}^{\infty} \#L(E,s) > n$. As indicated above the value E will be the number of errors we can correct and therefore we would of course like to find a large value of E such that the above condition is met. On the other hand the smallest value t such that

$$\sum_{s=0}^{t} \#L(E,s) > n \tag{4}$$

holds will to some extent reflect the complexity of the decoding algorithm. So in some situations it might be desirable to choose a smaller value of E than the largest possible one to decrease the complexity of the algorithm. Choosing parameters E and t and calculating the corresponding sets $L(E,0),\ldots,L(E,t)$ is something that is done when setting up the decoding system. Hence, the complexity of doing this is not of very high importance. However, as we will demonstrate in the case of generalized Reed-Muller codes, there are often tricks to ease the above procedure. We are now able to describe the modified Sudan decoding algorithm without multiplicity.

Algorithm 1.
Input: A code C as in (1), parameters E,t such that (4) is met and corresponding sets $L(E,0),\ldots,L(E,t)$. A received word $\boldsymbol{r}$
Output: A list of at most t codewords that contains all codewords within distance at most E from $\boldsymbol{r}$
Step 1. Find $Q_0,\ldots,Q_t \in R$ not all zero such that $Q_s \in \mathrm{Span}_{\mathbb{F}_q}\{f_\lambda \mid \lambda \in L(E,s)\}$ for $s = 0,\ldots,t$ and such that $\sum_{s=0}^{t}(\varphi(Q_s)) * \boldsymbol{r}^s = \mathbf{0}$ holds. (Here $\boldsymbol{r}^s$ means the component wise product of $\boldsymbol{r}$ with itself s times and $\boldsymbol{r}^0 = \mathbf{1}$.)
Step 2. Factorize $\sum_{s=0}^{t} Q_s Z^s \in R[Z]$ and detect all possible $f \in R$ such that $Z - f$ appears as a factor, which can be done by the method of Wu [10].
Step 3. Return $\{\varphi(f) \mid f \ \text{ is a solution from step 2}\}$.

Theorem 4. *Algorithm 1 gives the claimed output.*

Proof: Condition (4) ensures that the set of linear equations in step 1 has more indeterminates than equations. Therefore $Q_0, \ldots, Q_t$ as described in step 1 indeed do exist.

Consider any code word $\boldsymbol{c}$. That is, let $\boldsymbol{c} = \varphi(f)$ where f is of the form $f = \sum_{v=1}^{k} \beta_v f_{\lambda_v}$. From the conditions (2) and (3) we get that

$$\sum_{i=0}^{s} Q_i f^i \in \mathrm{Span}\{f_{\alpha(1)}, \ldots, f_{\alpha(n)}\} \tag{5}$$

holds and that

$$\text{all} \quad f_{\alpha(v)} \in \mathrm{Supp}_{\mathcal{B}}(\sum_{i=0}^{s} Q_i f^i) \quad \text{satisfies} \quad \sigma(\alpha(v)) > E. \tag{6}$$

Assume now that $\boldsymbol{c} = \varphi(f)$ is a code word within Hamming distance at most E from $\boldsymbol{r}$. But then $\sum_{s=0}^{t} (\varphi(Q_s)) * (\varphi(f))^s$ differs from $\sum_{s=0}^{t} (\varphi(Q_s)) * \boldsymbol{r}^s = \boldsymbol{0}$ in at most E positions implying

$$w_H(\varphi(\sum_{s=0}^{t} Q_s f^s)) \leq E \tag{7}$$

Combining (5), (6) and (7) with the first part of Theorem 2 lead to the conclusion that $\varphi(\sum_{s=0}^{t} Q_s f^s) = 0$ must hold, and Eq. (2) implies $\sum_{s=0}^{t} Q_s f^s = 0$. That is, f is a zero of $Q(Z)$. But order domains are integral domains and therefore $\mathrm{Quot}(R)$ is a field. It follows that $Z - f$ divides $Q(Z) \in \mathrm{Quot}(R)[Z]$. As the leading coefficient of $Z - f$ is 1 we conclude that $Q(Z) = (Z - f)K(Z)$ for some $K(Z)$ with coefficients in R. Hence, indeed $Z - f$ appears in the factorization in step 2 of the algorithm. Finally, as $Q(Z)$ has degree at most t the list in step 3 is of length at most t. □

Remark 1. We have used the Hamming weight to ensure $Q(Z) = 0$ in the above argument. The conventional method [9, 8] used the degree of a polynomial and the pole order of an algebraic function to ensure $Q(Z) = 0$. The use of Hamming weight allows us to list-decode codes from any order domains.

The following example illustrates the nature of our modification.

Example 6. Consider a one-point geometric Goppa code $E(\eta)$ where $\eta < n$. Let, g be the genus of the function field or equivalently let $g = \#\mathbb{N}_0 \backslash \Gamma$. The set

$$L'(E, s) = \{\lambda \in \Gamma \mid \lambda + s\eta < n - E\}$$

is easily calculated and we have $L'(E, s) \subseteq L(E, s)$. Replacing $L(E, s)$ with $L'(E, s)$ in Algorithm 1 gives the traditional algorithm [8] without multiplicity for the one-point geometric Goppa code $E(\eta)$. Hence, for one-point geometric Goppa codes the modified algorithm can correct at least as many errors as the original one and in cases where the sets $L(E, s)$ are larger than the sets $L'(E, s)$ we will be able to correct more errors by the modified algorithm.

3 Generalized Reed-Muller codes

In this section we consider the implementation of Algorithm 1 to the case of generalized Reed-Muller codes of low dimensions. Recall, from Example 1 that we have a weight function $\rho : \mathbb{F}_q[X_1, \ldots, X_m] \to \mathbb{N}_0^m$ given by $\rho(F) = (i_1, \ldots, i_m)$ if $X^{i_1} \cdots X^{i_m}$ is the leading monomial of F with respect to the monomial ordering from Example 1. Recall from Example 3 that we always choose the well-behaving basis $\mathcal{B}$ of $\mathbb{F}_q[X_1, \ldots, X_m]$ to be simply the set of monomials in $X_1, \ldots, X_m$. Observe that for the weight function under consideration the σ function is easily calculated as follows

$$\sigma\left((i_1, \ldots, i_m)\right) = \prod_{v=1}^{m} (q - i_v).$$

We get the following Lemma that significantly eases the job with finding $L(E, s)$.

Lemma 1. *Let $u < q$ and consider the generalized Reed-Muller code $RM_q(u, m)$. The description of $L(E, s)$ simplifies to*

$$\begin{aligned} L(E,s) = \{(l_1, \ldots, l_m) \in \mathbb{N}_0^m \mid & \\ & l_1 + su, \ldots, l_m + su < q, && (8)\\ & (q - l_1 - su)(q - l_2) \cdots (q - l_m) > E, \\ & \vdots && (9)\\ & (q - l_1) \cdots (q - l_{m-1})(q - l_m - su) > E\} \end{aligned}$$

Proof: To see that (9) corresponds to (3) we observe that the σ function from this section is concave. The fact that (8) corresponds to (2) follows from similar arguments. □

To decide how many errors our algorithm can correct we should according to (4) look for the largest possible E such that a t exists with $\sum_{s=0}^{t} \#L(E, s) > n = q^m$. Of course such an E can always be found by an extensive trial and error. For the case of $m = 2$ that is, codes of the form $\mathrm{RM}_q(u, 2)$ we now give an approximative trial and error method that requires only few calculations. It turns out that this approximative method is actually rather precise.

For a fixed s the conditions to be satisfied are

$$l_1 + su < q, \qquad l_2 + su < q \tag{10}$$

$$(q - l_1 - su)(q - l_2) > E, \qquad (q - l_1)(q - l_2 - su) > E \tag{11}$$

We make the (natural) assumption

$$0 \leq l_1, l_2 < q. \tag{12}$$

Equations (11) and (12) imply (10) which we therefore can forget about. When $E < q$, it is easy to lower-bound the number of solutions to (11) and (12). Under the assumption $E \geq q$ we now want to count the number of possible solutions

to (11) and (12). The number of such solutions is bounded below by the area in the first quadrant of the points that are under both the curve

$$l_2 = q - \frac{E}{q - l_1 - su} \tag{13}$$

as well as are under the curve

$$l_2 = q - su - \frac{E}{q - l_1} \tag{14}$$

By symmetry these two curves intersect in two points of the form (γ, γ). We have to use the point closer to the origin, which we calculate to be

$$\gamma = \frac{2q - su - \sqrt{s^2u^2 + 4E}}{2}.$$

Therefore (again by symmetry) the area is

$$2\left(\int_0^{\gamma} (q - su - \frac{E}{q - l_1})dl_1 - \frac{1}{2}\gamma^2\right)$$
$$= 2(\gamma(q - su) - E(\ln(q) - \ln(q - \gamma)) - \frac{1}{2}\gamma^2)$$

A rougher but simpler estimate is found by approximating the above area with the area of the polygon with corners $(0,0), (0, q - \frac{E}{q} - su), (\gamma, \gamma), (q - \frac{E}{q} - su, 0)$. Here the second point is found by substituting $l_1 = 0$ in (14) and the fourth point is found by substituting $l_2 = 0$ in (13). The estimate can serve as a lower bound due to the fact that both functions in (13) and (14) are concave. The area of the polygon is found to be $\gamma(q - (E/q) - su)$. Whether we use the first estimate or the second estimate we would next like to know the largest value of t such that $L(E,t) \neq \emptyset$. But this is easily calculated from the requirement $\gamma \geq 0$ implying $t = \lfloor (q - (E/q))/u \rfloor$. Combining the above results with Theorem 4 we get.

Proposition 1. *Consider the code $RM_q(u, 2)$ with $u < q$. For $E \geq q$ Algorithm 1 can correct at least E errors if the following holds*

$$\sum_{s=0}^{\lfloor (q - E/q)/u \rfloor} (2(\gamma(q - su) - E(\ln(q) - \ln(q - \gamma)) - \frac{1}{2}\gamma^2)) > q^2.$$

Corollary 1. *Consider the code $RM_q(u, 2)$ with $u < q$. For $E \geq q$ Algorithm 1 can correct at least E errors if the following holds*

$$\sum_{s=0}^{\lfloor (q - E/q)/u \rfloor} (\gamma(q - \frac{E}{q} - su)) > q^2.$$

Augot and Stepanov in [2] gave an improved estimate of the sum of multiplicities in terms of the total degree of a multivariate polynomial as follows

Theorem 5. *The sum of multiplicities in $\mathbb{F}_q^m$ of an m-variate polynomial of total degree d is upper bounded by dq^{m-1}. The number of zeros with multiplicity r of such a polynomial is upper bounded by dq^{m-1}/r.*

The above bound is better than the combination of Lemmas 2.4 and 2.5 in [7]. As noted by Augot and Stepanov Theorem 5 allows us to use more monomials in the first list decoding algorithm in [7], and the resulting decoding algorithm has the larger error-correcting capability.

The error correcting capability of the modified list decoding algorithm with Theorem 5 is compared with ours and the original Pellikaan-Wu. The multiplicity used in Augot and Stepanov's estimate is 10. E_{PW}, E_{PWA}, E_{ours} are the error correcting capability of the original Pellikaan-Wu, Augot-Stepanov, and our method, respectively. Finally, E_{PWA1} respectively E_{PWA2} are the error correcting capability of the Augot-Stepanov modified the Pellikaan-Wu algorithm when multiplicity is 1 respectively 2. $q = 16$, $m = 2$, $n = 256$.

u	2	3	4	5	6	7	8	9	10	11	12
E_{PW}	63	46	34	26	19	14	10	7	5	3	2
E_{ours}	76	55	44	34	27	21	15	13	11	9	6
E_{PWA}	118	99	83	70	59	49	41	33	25	19	11
E_{PWA1}	47	31	15	-1	-17	-33	-33	-49	-49	-65	-65
E_{PWA2}	87	63	47	31	23	7	-1	-9	-17	-25	-25

Remark 2. The authors of the present paper have done a lot of computer experiments regarding the error correcting capability of the proposed decoding method for generalized Reed-Muller codes. In all of these experiments we were able to correct as many errors as Remark 2.1 in [7] guarantees Pellikaan-Wu algorithm (with multiplicity) to be able to.

4 One-point geometric Goppa codes

As already mentioned our proposed decoding algorithm applies among other things to one-point geometric Goppa codes. In this section we will be concerned with codes defined from the norm-trace curve. These are defined by the polynomial $X^{(q^r-1)/(q-1)} - Y^{q^{r-1}} - Y^{q^{r-2}} - \cdots Y \in \mathbb{F}_{q^r}[X, Y]$. We consider codes

$$C_{\mathcal{L}}(P_1 + \cdots + P_{q^{2r-1}}, sP_\infty) \tag{15}$$

where $P_1, \ldots, P_{q^{2r-1}}, P_\infty$ are the rational places of the corresponding function field and P_∞ is the unique place among these with $\nu_{P_\infty}(x) < 0$. We do not go into detail with how to implement the proposed algorithm but present only some examples.

Example 7. In this example we consider the norm-trace curve corresponding to $q = 2$ and $r = 6$. These are of length $n = 2^{11}$. In the table below s is the value used in (15) whereas E_{our} is the error correcting capability of the proposed method and E_{GS1} is the error correcting capability of Sudan's algorithm [9]

without multiplicity. By 900-929 we indicate that maximal performance is a number between 900 and 929. With multiplicity, Guruswami-Sudan's algorithm [4] outperform the proposed method.

s	64	96	192	288	480
E_{our}	1008	900-929	660-669	527	346
E_{GS1}	962	804	479	237	14

Example 8. In this example we consider the norm-trace curve corresponding to $q = 3$ and $r = 3$. These are of length $n = 3^5$. In the table below s is the value used in (15) whereas E_{our} is the error correcting capability of the proposed method and E_{GS1} is the error correcting capability of Sudan's algorithm [9] without multiplicity. With multiplicity, Guruswami-Sudan's algorithm [4] outperform the proposed method.

s	63	70	80	88
E_{our}	55	51	43	38
E_{GS1}	53	47	39	33

References

1. H. E. Andersen and O. Geil, "Evaluation codes from order domain theory," *Finite Fields and Their Appl.*, doi:10.1016/j.ffa.2006.12.004, 2007 (in press).
2. D. Augot and M. Stephanov, "Decoding Reed-Muller codes with the Guruswami-Sudan's algorithm," slides of talk given by D. Augot at workshop D1 of *Special Semester on Gröbner Bases and Related Methods 2006*, RICAM, Linz, 2006, "http://www.ricam.oeaw.ac.at/specsem/srs/groeb/download/Augot.pdf"
3. O. Geil and R. Pellikaan, "On the structure of order domains," *Finite Fields and Their Appl.*, 8:369–396, 2002.
4. V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inform. Theory*, 45(4):1757–1767, Sept. 1999.
5. T. Høholdt, J. van Lint and R. Pellikaan, "Algebraic geometry codes," in Handbook of Coding Theory, Eds. V. S. Pless and W. C. Huffman, Elseview, 1998, pp. 871-961.
6. M. E. O'Sullivan, "New codes for the Berlekamp-Massey-Sakata algorithm," *Finite Fields and Their Appl.*, 7:293–317, 2001.
7. R. Pellikaan and X.-W. Wu, "List decoding of q-ary Reed-Muller codes," Extended version of paper that appeared in IEEE Trans. Inform. Theory, vol. 50, 2004, pp. 679-682.
8. M. A. Shokrollahi and H. Wasserman, "List decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, 45(2):432–437, Mar. 1999.
9. M. Sudan, "Decoding of Reed Solomon codes beyond the error correction bound," *J. Complexity*, 13:180–193, 1997.
10. X.-W. Wu, "An algorithm for finding the roots of the polynomials over order domains," in Proc. of ISIT-2002, Lausanne, 2002, p. 202.