On Multivariate Interpolation Decoding of 
Folded Reed-Solomon codes

Peter Trifonov
petert@dcn.ftk.spbstu.ru
Saint-Petersburg State Polytechnic University

Abstract. The problem of list decoding of folded Reed-Solomon codes is consid-
ered. A generalization of the Wu list decoding algorithm to the case of multivariate 
interpolation decoding is proposed. The reformulated algorithm achieves higher 
error correction capability than the one of Guruswami-Rudra.

1 Introduction

Reed-Solomon codes are widely used in modern communication and storage 
systems. Recently, there was a great interest to list decoding of these codes. 
After the introduction of the Guruswami-Sudan algorithm [3], which was able 
to correct up to \( t_{GS} = \left\lfloor n - \sqrt{n(k-1)} \right\rfloor \) errors in \((n,k)\) RS code, numerous 
 attempts were made to overcome this bound. In particular, it was shown by 
Guruswami and Rudra that one can recover up to a fraction \( 1 - R - \epsilon \) of errors 
in rate \( R \) folded RS code [2].

In this paper an alternative formulation of the multivariate decoding algo-
rithm for folded RS codes is presented, which is based on Wu list decoding 
method [6] for conventional RS codes. The novel formulation enables one to 
achieve higher error correction capability in the case of high-rate codes com-
pared to the case of Guruswami-Rudra algorithm.

The paper is organized as follows. The Guruswami-Rudra list decoding 
algorithm for folded RS codes is reviewed in Section 2. The generalization of 
the Wu algorithm to the case of folded RS codes is presented in Section 3. The 
numeric results are given in Section 4. Finally, some conclusions are drawn.

2 List decoding of folded Reed-Solomon codes

Let \( \gamma \in \mathbb{F} \) be an element of order at least \( n \). The \( m \)-folded RS code is a set 
of vectors \( (c_0, \ldots, c_{N-1}) \), \( N = n/m \), where the symbols \( c_j \in \mathbb{F}^m \) are \( m \)-tuples 
\( c_j = (f(\gamma^{jm}), \ldots, f(\gamma^{jm+m-1})), \deg f(x) < k \). That is, \( m \)-folded RS code of 
length \( n/m \) is obtained from a conventional \((n,k)\) RS code by considering \( m \) 
adjacent symbols in a codeword as a single one.

---

1This research is partially supported by the grant of the President of Russian Federation 
for young scientists MK-1195.2009.9.
List decoding of a folded RS code can be implemented using $M$-variate interpolation decoding. Let $(y_0, \ldots, y_{n-1})$ be an unfolded noisy vector to be decoded. The algorithm presented in [1] is based on construction of a polynomial $Q(X, Y_1, \ldots, Y_{M-1})$, such that the points

$$(\gamma^{jm+l}, y^{jm+l}, \ldots, y^{jm+l+M-1}), j = 0 \ldots n/m - 1, l = 0 \ldots m - M + 1$$

are its roots of multiplicity $r^2$. If $(1, k-1, \ldots, k-1)$-weighted degree of this polynomial is less than $D = \tau(m-M+2)r$, then all polynomials $f(x)$ (i.e. codewords of the folded RS code), such that $\deg f(x) < k$, and $f(\gamma^{jm+l}) = y^{jm+l}, l = 0 \ldots m - 1$, for at least $\tau$ distinct values of $j$ satisfy $Q(x, f(x), f(\gamma x), \ldots, f(\gamma^{M-2}x)) = 0$.

The interpolation constraints for points given by (1) lead to $n m^{-\rho (M-2)} (r + M - 1)$ linear equations. Obviously, the total degree of $Q(X, Y_1, \ldots, Y_{M-1})$ in $Y_i$ cannot exceed $\rho = \lceil (D - 1)/(k - 1) \rceil$. Hence, the number of terms in the polynomial is given by

$$\hat{N} = \sum_{j=0}^{\rho} \binom{j + M - 2}{M - 2} (D - (k - 1)j) (\rho + M - 1)$$

If this value exceeds the number of interpolation constraints, then the coefficients of this polynomial can be recovered as a solution of the corresponding system of linear equations. That is, the parameters of the above described algorithm must satisfy

$$\left( D - (k - 1)\rho \frac{M - 1}{M} \right) \left( \rho + M - 1 \right) > n \frac{m - M + 2}{m} \left( r + M - 1 \right)$$

It is possible to show that for sufficiently large $r$ the Guruswami-Rudra algorithm can be used to correct a fraction of $\theta = \frac{n/m - \tau}{n/m}$ errors, provided that the code rate$^3$ satisfies

$$R < \frac{m - M + 2}{m} \frac{1}{M} \sqrt{(1 - \theta)^M}.$$ 

### 3 A multivariate generalization of the Wu algorithm

It was suggested in [6] to perform list decoding of RS codes by means of algebraic continuation of the classical Berlekamp-Massey algorithm. This method was reformulated in [4] using the language of Gröbner bases. We will adopt the latter approach, since it leads to simpler derivations.

$^2$A polynomial $Q(X, Y_1, \ldots, Y_{M-1})$ has a root of multiplicity $r$ at point $(x_0, y_0, \ldots, y_{M-1})$ if all its partial Hasse derivatives of total order $j_0 + \ldots + j_{M-1} < r$ are equal to zero.

$^3$Here code rate is defined as $R = \frac{k-1}{n}$ in order to keep the notation consistent with [2].
The problem of decoding of RS codes (including folded ones) reduces to finding all pairs of polynomials \([\sigma(x), f(x)]\), such that

\[
\sigma(\gamma^i) f(\gamma^i) = y_i f(\gamma^i), \quad i = 0..n - 1,
\]

where \(\deg f(x) < k\), and \(\sigma(\gamma^i) = 0\) iff \(y_i\) is corrupted. This is equivalent to finding a bivariate polynomial \(Q(x, y) = \sigma(x)y - p(x)\), such that \(Q(\gamma^i, y_i) = 0\), and \(\text{LT}(x, y) = x^q\), where \(\text{LT}(x, y)\) denotes the leading term of the polynomial with respect to \((1, k - 1)\)-weighted degree lexicographic ordering with \(y < x\). \(T\) is the total number of errors in the vector \((y_0, \ldots, y_{n-1})\), and \(\deg p(x) \leq T + k - 1\). Decoding of folded codes requires one also to demand that the set of roots of \(\sigma(x)\) should be covered by at most \(t\) sets \(\{\gamma^{im+j}, \ldots, \gamma^{im+m-1}\}\), \(i = 0..n/m - 1\), i.e. there should be at most \(t\) groups of consecutive errors of length \(m\). This implies \(T = tm\).

All such bivariate polynomials can be found in the module \(\mathcal{M} = \{Q(x, y) = q_0(x) + q_1(x)y|Q(\gamma^i, y_i) = 0, i = 0..n - 1\}\), which can be considered as a two-dimensional vector space over \(\mathbb{F}[x]\). Let \(q_{00}(x) + yq_{01}(x)\) and \(q_{10}(x) + yq_{11}(x)\) be a Gröbner basis of this module with respect to \((1, k - 1)\)-weighted degree lexicographic ordering. Then any polynomial \(Q(x, y)\) corresponding to a solution of the decoding problem satisfies \(Q(x, y) = a(x)(q_{00}(x) + yq_{01}(x)) + b(x)(q_{10}(x) + yq_{11}(x))\), where \(\deg a(x) \leq w_1 = T + k - 1 - \deg q_{00}(x)\), and \(\deg b(x) \leq w_2 = T - \deg q_{11}(x)\). In particular,

\[
\sigma(x) = a(x)q_{01}(x) + b(x)q_{11}(x).
\]

In the case of folded RS codes this implies that one should find \(a(x), b(x) : a(\gamma^{im+j})q_{01}(\gamma^{im+j}) + b(\gamma^{im+j})q_{11}(\gamma^{im+j}) = 0, j = 0..m - 1\) for at most \(t\) distinct values of \(i\).

Following [4], we introduce a partially homogenized polynomial (PHP)

\[
Q(x, y_1, z_1, \ldots, y_{M-1}, z_{M-1}) = \sum_{j_1=0}^\rho \cdots \sum_{j_{M-1}=0}^\rho q_{j_1,\ldots,j_{M-1}}(x)y_1^{j_1}z_1^{\rho-j_1}\cdots y_{M-1}^{j_{M-1}}z_{M-1}^{\rho-j_{M-1}},
\]

which will be used to identify all suitable polynomial pairs \([a(x), b(x)]\).

**Lemma 1.** Let \(Q(x, y_1, z_1, \ldots, y_{M-1}, z_{M-1})\) be a PHP such that for all \(\alpha_j\) the points \((x_0, \alpha_1u_1, \alpha_1v_1, \ldots, \alpha_{M-1}u_{M-1}, \alpha_{M-1}v_{M-1})\) are its roots of multiplicity \(r\). Then for all \(a_i(x), b_i(x)\):

\[
v_ia_i(x_0) - u_ib_i(x_0) = 0, \quad i = 1..M - 1,
\]

the polynomial \(Q(x, a_1(x), b_1(x), \ldots, a_{M-1}(x), b_{M-1}(x))\) is divisible by \((x-x_0)^r\).

**Proof.** The polynomial has roots \((x_0, \alpha_1u_1, \alpha_1v_1, \ldots, \alpha_{M-1}u_{M-1}, \alpha_{M-1}v_{M-1})\) of multiplicity \(r\) iff all its Hasse derivatives of total order less than \(r\) are equal.
to zero at these points. In the case of PHP this can be written as

$$Q(x,y_1,z_1,\ldots,y_{M-1},z_{M-1}) = \sum_{j_0+j_1+\ldots+j_{M-1} \geq r} q_{j_0,\ldots,j_{M-1}}(x-x_0)^{j_0} \prod_{i=1}^{M-1} (y_iv_i-z_iu_i)^{j_i}$$

Clearly, \((x - x_0)(v_ia_i(x) - u_ib_i(x))\). Hence, \((x - x_0)^r\) divides \(Q(x,a_1(x),b_1(x),\ldots,a_{M-1}(x),b_{M-1}(x))\). \hfill \Box

For the sake of simplicity, the arbitrary constants \(\alpha_i\) will be omitted in the subsequent derivations. The following is a reformulation of [2, Lemma 4.1].

**Lemma 2.** Let \(Q(x,y_1,z_1,\ldots,y_{M-1},z_{M-1})\) be a PHP having roots \((\gamma_{im+j}, q_{11}(\gamma_{im+j}), q_{12}(\gamma_{im+j}), \ldots, q_{11}(\gamma_{im+j}+M-2)), -q_{10}(\gamma_{im+j}+M-2)), j = 0..m - M + 1, i = 0..n/m - 1, \) of multiplicity \(r\). If \((1, w_1, w_2, \ldots, u_1, u_2)\)-weighted degree of this polynomial is less than \(D = r(m - M + 2)t\), then \(Q(x,a(x),b(x),a(\gamma(x)),b(\gamma(x)),\ldots,a(\gamma^{M-2}x),b(\gamma^{M-2}x)) = 0\) for all \(a(x), b(x)\), such that \(\deg a(x) \leq w_1, \deg b(x) \leq w_2,\) and

\[
a(\gamma_{im+j})q_{10}(\gamma_{im+j}) + b(\gamma_{im+j})q_{10}(\gamma_{im+j}) = 0, j = 0..m - 1,
\]

for at least \(t\) distinct values of \(i\).

**Proof.** By lemma 1, \(g(x) = Q(x,a(x),b(x),\ldots,a(\gamma^{M-2}x),b(\gamma^{M-2}x))\) is divisible by \((x - \gamma_{im+j})^r, j = 0..m - M + 1\) for all \(i : a(\gamma_{im+j})q_{10}(\gamma_{im+j}) + b(\gamma_{im+j})q_{10}(\gamma_{im+j}) = 0, j' = 0..m - 1\). However, \(\deg g(x)\) is less than \(r(m - M + 2)t\). Hence, \(g(x) = 0\). \hfill \Box

This lemma implies that one can correct up to \(t\) errors in a folded RS code by constructing a \((2M - 1)\)-variate PHP with sufficiently small \((1, w_1, w_2, \ldots, w_1, w_2)\)-weighted degree having \(n\) \(\frac{m - M + 2}{m}\) roots of multiplicity \(r\). This can be done as long as the number of terms in it exceeds the number of equations. The number of terms of total degree less than \(D\) in (4) is given by \(N = \sum_{j_0=0}^r \cdots \sum_{j_{M-1}=0}^r \left(D - \sum_{i=1}^{M-1} (w_1j_i + w_2(\rho - j_i))\right) = (D - w_2\rho(M - 1))\rho + 1)^{M-1} - (w_1 - w_2)(M - 1)\rho + 1)^{M-1} = (D - w\rho(M - 1))(\rho + 1)^{M-1},\)

where \(w = w_1 + w_2 = 2tm + k - 1 - \deg q_{11}(x) - \deg q_{00}(x) = 2tm + k - 1 - n,\) and the latter equality follows from the properties of the Gröbner basis of \(\mathcal{M}\) [4, 5]. Hence, the parameters of the algorithm must satisfy

\[
(r(m - M + 2)t - \frac{2tm + k - 1 - n}{2}\rho(M - 1)) (\rho + 1)^{M-1} > \frac{n(m - M + 2)}{mM!} \prod_{j=0}^{M-1} (r+j).
\]

Relaxing \(\rho\) to be a continuous variable and optimizing over it, one obtains that the left-hand side of this maximized by setting \(\rho = \frac{D-w\rho/2}{Mr/2}\). The optimal value
Table 1: Comparison of the decoding algorithms: $n = 255, M = 3$

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$t$</th>
<th>$m$</th>
<th>$R_{\text{opt}}$</th>
<th>$k$</th>
<th>$r$</th>
<th>$\rho$</th>
<th>$R_{\text{opt}}$</th>
<th>$k$</th>
<th>$r$</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>25</td>
<td>1</td>
<td>0.81</td>
<td>208</td>
<td>46</td>
<td>13</td>
<td>0.81</td>
<td>208</td>
<td>5</td>
<td>51</td>
</tr>
<tr>
<td>0.3</td>
<td>8</td>
<td>3</td>
<td>0.57</td>
<td>147</td>
<td>19</td>
<td>20</td>
<td>0.82</td>
<td>212</td>
<td>7</td>
<td>18</td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td>5</td>
<td>0.68</td>
<td>175</td>
<td>18</td>
<td>19</td>
<td>0.82</td>
<td>211</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>0.8</td>
<td>76</td>
<td>1</td>
<td>0.49</td>
<td>125</td>
<td>23</td>
<td>33</td>
<td>0.49</td>
<td>125</td>
<td>9</td>
<td>32</td>
</tr>
<tr>
<td>0.5</td>
<td>25</td>
<td>3</td>
<td>0.39</td>
<td>100</td>
<td>12</td>
<td>14</td>
<td>0.50</td>
<td>131</td>
<td>33</td>
<td>43</td>
</tr>
<tr>
<td>0.5</td>
<td>15</td>
<td>5</td>
<td>0.47</td>
<td>120</td>
<td>15</td>
<td>18</td>
<td>0.52</td>
<td>136</td>
<td>33</td>
<td>43</td>
</tr>
<tr>
<td>0.8</td>
<td>127</td>
<td>1</td>
<td>0.25</td>
<td>64</td>
<td>26</td>
<td>52</td>
<td>0.25</td>
<td>64</td>
<td>25</td>
<td>51</td>
</tr>
<tr>
<td>0.8</td>
<td>42</td>
<td>3</td>
<td>0.24</td>
<td>61</td>
<td>22</td>
<td>31</td>
<td>0.22</td>
<td>59</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>0.8</td>
<td>25</td>
<td>5</td>
<td>0.28</td>
<td>73</td>
<td>14</td>
<td>20</td>
<td>0.27</td>
<td>72</td>
<td>34</td>
<td>55</td>
</tr>
<tr>
<td>0.8</td>
<td>204</td>
<td>1</td>
<td>0.04</td>
<td>11</td>
<td>41</td>
<td>209</td>
<td>0.04</td>
<td>11</td>
<td>165</td>
<td>206</td>
</tr>
<tr>
<td>0.8</td>
<td>68</td>
<td>3</td>
<td>0.059</td>
<td>16</td>
<td>61</td>
<td>138</td>
<td>-0.15</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.8</td>
<td>40</td>
<td>5</td>
<td>0.071</td>
<td>19</td>
<td>6</td>
<td>14</td>
<td>-0.06</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

is given by

$$N^* = \frac{(D+(M-1)w/2)^M}{M^M (w/2)^{M-1}} = \frac{(r(m-M+2)t+(M-1)(2m+k-1-n)/2)^M}{M^M ((2m+k-1-n)/2)^{M-1}}.$$  

Then (5) can be rewritten as

$$\frac{m-M+2\theta + M-1(\theta + (R - 1)/2))^M}{M^M (\theta + (R - 1)/2)^{M-1}} \geq \frac{m-M+2}{mM!} \prod_{j=0}^{M-1} \left( 1 + \frac{j}{r} \right).$$

For sufficiently large $r$ this inequality can be solved if

$$\frac{m-M+2}{mM!} \frac{(m-M+2)^{M-1} \theta^M}{M^M (\theta + (R - 1)/2)^{M-1}},$$

i.e.

$$R < 1 - 2\theta + 2\frac{m-M+2}{m} \left( \frac{M}{M^M \theta^M} \right)^{\frac{1}{M-1}}.$$  (6)

This expression implies that the multivariate Wu algorithm always outperforms the classical decoding techniques, which require $R < 1 - 2\theta$.

4 Numeric results

This section presents comparison of the Guruswami-Rudra and Wu-based decoding algorithms. For each $\theta$ the maximal achievable rate was computed from (3) and (6). The exact values of parameters $r, \rho$, and $k$, such that (2) and (5) are satisfied, are reported for the case of $m$-folded RS code of length 255/m.

It can be seen that the multivariate Wu algorithm outperforms the Guruswami-Rudra one in the case of small $\theta$, i.e. high-rate codes. In the case of $m = 1$ (i.e. list decoding of conventional RS codes) the error correction
capability of both algorithms is identical. For \( m = 3 \) and small \( \theta \) the performance of the Guruswami-Rudra algorithm turns out to be worse than for \( m = 1 \), as it was observed in [2]. However, the achievable rate improves with \( m \) and \( \theta \). The multivariate Wu algorithm described in this paper outperforms the Guruswami-Rudra algorithm for small values of \( \theta \), but increasing \( \theta \) causes its performance to degrade. For \( \theta \geq 0.8 \) it turns out to be impossible to find suitable parameters \( r \) and \( \rho \). It can be also seen that the actual rate of the codes which can be decoded using the reformulated Wu algorithm exceeds slightly the value given by (6).

5 Conclusions

In this paper a multivariate generalization of the Wu list decoding algorithm to the case of folded RS codes was proposed. It was shown that for relatively small values of \( \theta \) the proposed algorithm can be used to list decode rate \( R \) \( m \)-folded RS codes from a fraction \( \theta \) of errors for larger values of \( R \) compared to the case of the Guruswami-Rudra algorithm. However, the proposed algorithm cannot be used for \( \theta \) close to 1.

References


