Reflections on "Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes"

Venkatesan Guruswami* Madhu Sudan†

January, 2002

A t-error-correcting code over a q-ary alphabet $\mathbb{F}_q$ is a set $C \subseteq \mathbb{F}_q^n$ such that for any received vector $\mathbf{r} \in \mathbb{F}_q^n$ there is at most one vector $\mathbf{c} \in C$ that lies within a Hamming distance of t from $\mathbf{r}$. The minimum distance of the code C is the minimum Hamming distance between any pair of distinct vectors $\mathbf{c}_1, \mathbf{c}_2 \in C$. In his seminal work introducing these concepts, Hamming pointed out that a code of minimum distance $2t+1$ is a t-error-correcting code. It also pointed out the obvious fact that such a code is not a t'-error-correcting code for any $t' > t$. We conclude that a code can correct half as many errors as its distance and no more.

The mathematical correctness of the above statements are indisputable, yet the interpretation is quite debatable. If a message encoded with a t-error-correcting code ends up getting corrupted in $t' > t$ places, the decoder may simply throw it hands up in the air and cite the above paragraph. Or, in an alternate notion of decoding, called *list decoding*, proposed in the late 1950s by Elias [10] and Wozencraft [43], the decoder could try to output a list of codewords within distance t' of the received vector. If t' is not much larger than t and the errors are caused by a probabilistic (non-malicious) channel, then most likely this list would have only one element — the transmitted codeword. Even if the errors are caused by a malicious jammer, the list cannot contain too many codewords provided t' is not too much larger than t. Thus, in either case, the receiver is in a better position to recover the transmitted codeword under the model of list decoding.

List decoding was initiated mainly as a mathematical tool that allowed for a better understanding of some of the classical parameters of interest in information and coding theory. Elias [10] used this notion to get a better handle on the error-exponent in the strong forms of Shannon's coding theorem. The notion also plays a dominant role in the Elias-Bassalygo [34, 4] upper bound on the rate of a code as a function of its relative distance.

Through the decades the notion has continued to be investigated in a combinatorial context; and more recently has seen a spurt of algorithmic results. The paper being reflected on [23] was motivated by a gap between the combinatorial understanding of Reed-Solomon codes, and the known algorithmic performance. Below we summarize the combinatorial state of knowledge, and describe the main result of [23], and also use the opportunity to survey some of the rich body of algorithmic results on list decoding that have emerged in the recent past. We also muse upon some useful asymptotic perspectives that eased the way for some of this progress, and reflect on some possibilities for future work.

*University of California at Berkeley, Computer Science Division, Berkeley, CA 94708. venkat@lcs.mit.edu
†MIT Laboratory for Computer Science, 200 Technology Square, Cambridge, MA 02139, USA. madhu@mit.edu.

1 Combinatorics of list decoding

We start by defining the notion of the list decoding radius of an (infinite family of) codes. This notion is adapted from a definition in [20], who term it the "polynomial list decoding radius".

Definition 1 *A family of codes $\mathcal{C}$ has a* list decoding radius $L : \mathbb{Z}^+ \to \mathbb{Z}^+$ *if there exists a polynomial $p(\cdot)$ such that for every code $C \in \mathcal{C}$ of block length n, and every received vector $\mathbf{r}$, it is that case that there are at most $p(n)$ codewords in C that have Hamming distance at most $L(n)$ from $\mathbf{r}$. We say that the code has a* relative list decoding radius $\ell(n)$ *if it has list decoding radius $L(n) = n \cdot \ell(n)$.*

The primary thrust of the combinatorial study is the relationship between $\ell(n)$ and the more classical parameters $\delta(n)$, the relative distance of a code, and $R(n)$, the rate of a code. (A family of codes has rate $R(n)$ (relative distance $\delta(n)$) if every member of $\mathcal{C}$ of block length n has information length at least $n \cdot R(n)$ (minimum distance at least $n\delta(n)$).)

For a "well-designed" code $\mathcal{C}$ of relative distance $\delta(n)$, one should expect the list decoding radius $\ell(n)$ to be at most $\delta(n)$. And from the fact that a code can correct half as many errors as its distance it follows that a family of codes $\mathcal{C}$ of relative distance $\delta(n)$ has relative list decoding radius $\ell(n) \geq \delta(n)/2$. The real question here is where in between $\delta/2$ and δ does the list decoding radius actually lie in general. The classical Johnson bound (or at least, its proof) shows that $\ell(n) \geq 1 - \sqrt{1 - \delta(n)}$ which turns out to be better than $\delta/2$ for all choices of δ. This bound motivates one of the principal algorithmic challenges associated with list decoding: For a code of relative distance $\delta(n)$, give a polynomial time algorithm to find a list of all codewords within a relative distance of $(1 - \sqrt{1 - \delta(n)})$ from a given received word $\mathbf{r}$. This is the question that motivated the work [23] and was answered positively therein. Before describing the algorithmic results, we wrap up the section with a summary of the combinatorial state of knowledge.

The inequality $\ell(n) \geq 1 - \sqrt{1 - \delta(n)}$ appears to be the best possible lower bound one can establish on the relative list decoding radius of a code as a function of its distance.[1] It is easy to prove the existence of non-linear codes which match this bound. The question of whether the bound is the best one can prove for linear codes remains open, though significant progress has been made towards resolving it in [25, 20, 18].

From the point of usage, it is more useful to compare the rate of a code with its list decoding radius. This question has been investigated over the years by [6, 7, 45, 11, 20]. It follows from the converse to Shannon's coding theorem that a q-ary code of relative list decoding radius $\ell(n)$ has rate at most $R(n) \approx 1 - H_q(\ell(n))$. The above mentioned works show that there exist codes approaching this bound. The associated algorithmic challenge, of constructing such codes explicitly and finding decoding algorithms for them remains wide open.

2 List decoding algorithms

Despite the obvious utility of list decoding algorithms, few results were obtained till the eighties. The first efficient list decoding algorithms, due to Dumer [9] and Sidelnikov [36] corrected a number of errors that were of the form $\ell(n) = (\frac{1}{2} + o(1))\delta(n)$ for some families of Reed-Solomon codes. This problem was introduced to the computer science literature by Goldreich and Levin [14] who gave a

[1] This applies to bounds that apply for all codes, regardless of their alphabet size. For small alphabets, eg. for binary codes, a better bound can be proven. Since our primary focus is Reed-Solomon codes, we do not elaborate on the improved bound on list decoding radius that takes into account the alphabet size.

highly efficient randomized list decoding algorithm for Hadamard codes, when the received vector was given implicitly. This work led to some extensions by Goldreich, Rubinfeld, and Sudan [16]. Yet no efficient list decoding algorithms were found for codes of decent rate (constant, or even slowly vanishing rate such as $R(n) = n^{-1+\varepsilon}$ for some $\varepsilon > 0$).

The first list decoding algorithm correcting $\alpha\delta(n)$ errors for $\alpha > \frac{1}{2}$ for codes of constant rate was due to Sudan [38], who gave such an algorithm for Reed-Solomon codes. The algorithm was subsequently extended to algebraic-geometric codes by Shokrollahi and Wasserman [35]. Yet these results did not decode up to the best known combinatorial bounds on list decoding radius; in fact, they did not correct more than $\delta(n)/2$ errors for any code of rate greater than 1/3. The obvious gap between the combinatorial bound ($\ell(n) \geq 1 - \sqrt{1-\delta(n)}$) and the algorithmic results motivated the work [23], where this gap was bridged for Reed-Solomon codes and algebraic-geometric codes. Specifically, the following theorem was proven for the class of Reed-Solomon codes.

Theorem 2 ([23]) *There exists an algorithm that, given a received vector* $\mathbf{r}$ *and a description of a* q*-ary Reed-Solomon code of dimension* $(k+1)$ *and block length* n*, finds a list of all codewords within a distance of* $n(1-\sqrt{k/n})$ *from the received vector. The running time of the algorithm is bounded by a polynomial in* n *and* q*.*

Below we give a brief overview of the algorithm and in particular, describe some of the history behind this algorithm.

2.1 Decoding Reed-Solomon Codes

It might help to recall the definition of Reed-Solomon codes. Let $\mathbb{F}_q$ denote a field of size q and let $\mathbb{F}_d^k[x]$ denote the vector space of polynomial of degree at most k over $\mathbb{F}_q$. Recall that the Generalized Reed Solomon code of dimension $(k+1)$, is specified by distinct $x_1, \ldots, x_n \in \mathbb{F}_q$ and consists of the evaluations of all polynomials p of degree at most k at the points $x_1, \ldots, x_n$. More formally, letting $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ and letting $p(\mathbf{x})$ denote $\langle p(x_1), \ldots, p(x_n) \rangle$, we get that the associated code $\mathrm{RS}_{q,k,\mathbf{x}}$ is given by

$$\mathrm{RS}_{q,k,\mathbf{x}} = \{p(\mathbf{x}) | p \in \mathbb{F}_q^k[x]\}.$$

Viewed from this perspective (as opposed to the dual perspective, where the codewords of the Reed Solomon codes are coefficients of polynomials), the Reed Solomon decoding problem is really a "curve-fitting" problem: Given n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, find all polynomial $p \in \mathbb{F}_q^k[x]$ such that $\Delta(p(\mathbf{x}), \mathbf{y}) \leq e$, for some error parameter e. (Here and later, $\Delta(\cdot,\cdot)$ denotes the Hamming distance.) We now give a brief summary of the algorithmic ideas that led to the algorithm in [23]. This chain of ideas includes the Welch-Berlekamp algorithm [42, 5], an algorithm for a restricted decoding problem due to Ar et al. [1], and the list decoding algorithm of Sudan [38].

Traditional algorithms, starting with those of Peterson [32] attempt to "explain" $\mathbf{y}$ as a function of $\mathbf{x}$. This part becomes explicit in the work of Welch & Berlekamp [42, 5] (see, in particular, the expositions in [13] or [37, Appendix A]) where $\mathbf{y}$ is interpolated as a rational function of $\mathbf{x}$, and this leads to the efficient decoding. (Specifically a rational function $a(x)/b(x)$ can be computed such that for every $i \in \{1, \ldots, n\}$, we have $a(x_i) = y_i * b(x_i)$.)

Rational functions, however, are limited in their ability to extract the message from data with large amounts of error. In particular they fail to work when the data has exactly two explanations — i.e., there are two polynomials p_1 and p_2 such that for exactly half the points $y_i = p_1(x_i)$ and for the other half $y_i = p_2(x_i)$. In such a case it is still possible to find an algebraic explanation of the points $\{(x_i.y_i)\}_{i=1}^n$: we simply have that the polynomial $Q(x,y) = (y - p_1(x)) \cdot (y - p_2(x))$ is zero on every given (x_i, y_i). Furthermore the polynomial $Q(x,y)$ can be found by simple interpolation (which amounts to solving a linear system), and the candidate polynomials $p_1(x)$ and $p_2(x)$ are the roots of the polynomial $Q(x,y)$. (Notice that the factoring will find two polynomials p_1 and p_2 and, if $\omega^2 \neq 1$, the true candidate is p_1 iff it satisfies $p_2 = \omega p_1$.) This was the problem considered by Ar et al. [1] and the solution above is the one given by them.

The next step in this chain of ideas, due to Sudan [38], is the realization that the algorithm above already solves the Reed-Solomon list decoding problem for a non-trivial choice of parameters (rate vs. list decoding radius). In particular, a simple counting argument shows that there exists a non-zero polynomial $Q(x,y)$ of degree $\sqrt{n}$ each in x and y that is zero on *any* set of n points. Now, if a subset of more than $(k+1)\sqrt{n}$ these points satisfy $y_i = p(x_i)$, then $y - p(x)$ is a factor of $Q(x,y)$. Thus finding such a bivariate polynomials Q and factoring it, gives a small list of polynomials that includes all the candidates for output of the list decoding algorithm. By picking the degree of Q very carefully, one can improve its performance significantly to at least $n - \sqrt{2kn}$ errors (see [39] for a more complete analysis of the performance of this algorithm).

The interesting aspect of the above algorithm is that it takes some very elementary algebraic concepts, such as unique factorization, Bezout's theorem, and interpolation, and makes algorithmic use of these concepts in developing a decoding algorithm for an algebraic code. This may also be a good point to mention some of the significant advances made in the complexity of factoring multivariate polynomials that were made in the 1980's. These algorithms, discovered independently by Grigoriev [17], Kaltofen [26], and Lenstra [28], form the technical foundations of the decoding algorithm above. Modulo these algorithms, the decoding algorithm and its proof rely only on elementary algebraic concepts. Exploiting slightly more sophisticated concepts from commutative algebra, leads to even stronger decoding results that we describe next.

The algorithm of Guruswami and Sudan [23] is best motivated by the following weighted curve fitting question: Suppose in addition to vectors $\mathbf{x}$ and $\mathbf{y}$, one is also given a vector of positive integers $\mathbf{w}$ where w_i determines the "weight" or confidence associated with a given point (x_i, y_i). Specifically we would like to find all polynomials p such that $\sum_{i|p(x_i)=y_i} w_i \geq W$ (for as small a W as possible). How can one interpret the weights in the algebraic setting? A natural way at this stage is to find a "fit" for all the data points that corresponds to the weights: Specifically, find a polynomial $Q(x,y)$ that "passes" through the point (x_i, y_i) at least w_i times. The notion of a curve passing through a point multiple times is a well-studied one. Such points are called singularities. Over fields of characteristic zero, these are algebraically characterized by the fact that the partial derivatives of the curve (all such, up to the $(r-1)$th derivatives, if the point must be visited by the curve r times), vanish at the point. The relevant component of this observation is that insisting that a curve pass through a point r times is placing $\binom{r+1}{2}$ linear constraints on the coefficients. This fact remains true over finite fields, though the partial derivatives don't yield these linear constraints any more. Formalizing this algorithm carefully and optimizing the degree of Q appropriately, gives the following lemma:

Lemma 3 ([23]) *Given vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, and $\mathbf{w} \in (\mathbb{Z}^+)^n$, a list of all polynomials $p \in \mathbb{F}_q^k[x]$ satisfying $\sum_{i|p(x_i)=y_i} w_i > \left\lfloor \sqrt{k \sum_{i=1}^{n} w_i(w_i+1)} \right\rfloor$ can be found in time polynomial in $n, \sum_i w_i$, provided all pairs (x_i, y_i) are distinct.*

The surprising element in the above lemma is that the performance is not invariant to scaling of the w_i's — and the requirement on the amount of agreement decreases as one scales the weights up. This holds even if all the weights are equal, in which case the problem being solved is just the Reed-Solomon list decoding problem in a disguised form. In particular, by setting the weights appropriately large gives the algorithm claimed in Theorem 2. Thus we have a better unweighted decoding algorithm, that uses the weighted version as an intermediate step! Of course, it is also possible to state what the algorithm achieves for a general set of weights. For this part, we will just assume that the weight vector is an arbitrary vector of non-negative reals, and get the following:

Theorem 4 ([23, 24]) *Given vectors* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$*, a weight vector* $\mathbf{w} \in \mathbb{R}_{\geq 0}^n$*, and a real number* $\varepsilon > 0$*, a list of all polynomials* $p \in \mathbb{F}_q^k[x]$ *satisfying* $\sum_{i|p(x_i)=y_i} w_i > \sqrt{k(\varepsilon + \sum_{i=1}^n w_i^2)}$ *can be found in time polynomial in* n *and* $\frac{1}{\varepsilon}$*, provided the pairs* (x_i, y_i) *are all distinct.*

This result summarizes the state of knowledge for list decoding of Reed Solomon codes, subject to the restriction that the decoding algorithm runs in polynomial time. However this criterion, that the decoding algorithm runs in polynomial time, is a very loose one. The practical nature of the problem deserves a closer look at the components involved and efficient strategies to implement these components. This problem has been considered in the literature, with significant success. In particular, it is now known how to implement the interpolation step in $O(n^2)$ time, when the output list size is a constant [31, 33]. Similar running times are also known for the root finding problem (which suffices for the second step in the algorithms above) [3, 12, 29, 31, 33, 44]. Together these algorithms lead to the possibility that a good implementation of list decoding may actually even be able to compete with the classical Berlekamp-Massey decoding algorithm in terms of efficiency. A practical implementation of such an algorithm in C++, due to Rasmus Refslund Nielsen, is available from from his homepage (`http://www.student.dtu.dk/~p938546/index.html`).

The paper [23] also presents a generalization of the weighted decoding algorithm to the case of algebraic-geometric codes. Using it as an intermediate step with a suitable choice of weights, one gets an algorithm that decodes algebraic-geometric codes beyond half the minimum distance for every value of rate. In fact, as noted in [27], a careful choice of weights enables decoding up to the combinatorial bound on list decoding radius.

2.2 Other algorithmic results

A rich body of algorithmic results concerning list decoding have appeared following the publication of [23]. We have already mentioned the works that addressed the question of more efficient implementations of the list decoding algorithms for Reed-Solomon and algebraic-geometric codes from [23]. Goldreich, Ron, and Sudan [15] considered the question of list decoding a number-theoretic code called the *Chinese Remainder code* (henceforth, CRT code). Here, the messages are identified with integers m in the range $0 \leq m < K$ and a message m is encoded as: $m \mapsto \langle m \pmod{p_1}, \ldots, m \pmod{p_n} \rangle$ where $p_1 < p_2 < \cdots < p_n$ are n relatively prime integers. When $K = p_1 \cdot p_2 \cdots p_k$, the Chinese Remainder Theorem implies that the code thus defined has distance $(n - k + 1)$. The combinatorial bounds then indicate that such a code can be list decoded with small lists up to about $n - \sqrt{kn}$ errors. Goldreich et al. [15] initiated the study of list decoding CRT codes and this was continued in Boneh [8]. However, these algorithms corrected only about $n - \Omega\left(\sqrt{kn \frac{\log p_n}{\log p_1}}\right)$ errors and therefore their performance was poor when the p_i's had widely different magnitudes.

Subsequently, in [22], it was realized that algebraic and number-theoretic codes can be unified under the umbrella of *ideal-based* codes. Loosely speaking, the messages of an ideal-based code are all elements of small "size" in a "nice" commutative ring, and a message is encoded by the sequence of its residues modulo a set of pairwise coprime *ideals* of the ring. Moreover, [22] also showed that the idea behind the list decoding scheme from [23] can be generalized to work for ideal-based codes as well. In addition to giving a "unified" approach to list decoding Reed-Solomon, algebraic-geometric and CRT codes, this also resulted in an improved algorithm for CRT codes that could list decode from up to $n - \sqrt{k(n+\varepsilon)}$ errors (for arbitrary $\varepsilon > 0$) and thus essentially up to the combinatorial list decoding radius.

The result of Theorem 4 has seen elegant applications in list decoding algorithms for concatenated codes with outer code being Reed-Solomon or algebraic-geometric and with certain choices of inner code. Nielsen [30] considers the case of inner codes with small distance. Elegant analytic results for the case when the inner code is Hadamard are obtained in [24]. In [20], the authors use "tailor-made" inner codes that work very well in conjunction with the weighted Reed-Solomon decoding algorithm. In the latter two works, the inner codes are first decoded to provide, for each position i of the outer Reed-Solomon code, a "weight" $w_{i,\alpha}$ for each field element α. The weight $w_{i,\alpha}$ is a measure of the confidence that the i'th symbol of the Reed-Solomon codeword is α. These weights are then used to list decode the outer Reed-Solomon code as per Theorem 4. Analyzing such a decoding procedure with a careful choice of weights gives algorithms to list decode certain concatenated codes up to or reasonably close to their list decoding radius. We refer the reader to [24, 20], or [19, Chapter 8] for further details.

Besides algebraic-geometric codes, Reed-Solomon codes can be generalized in another way, by allowing polynomials on more than one variable to encode the message. This gives the class of *Reed-Muller codes*. The technique used in [23] unfortunately does not seem to generalize in any simple way to decode Reed-Muller codes up to their list decoding radius, or for that matter even beyond half the distance for all rates, and this remains an interesting open question. However, in [2, 40], using clever reductions to the univariate case, an algorithm to list decode Reed-Muller codes well beyond half the distance is presented for codes of low rate.

A consequence of Theorem 2 is that, for arbitrary $\varepsilon > 0$, efficient list decoding up to a fraction $(1-\varepsilon)$ of errors can be performed using codes of rate ε^2. The only drawback of Reed-Solomon codes is their large alphabet size (which is at least their block length). While this is alleviated by algebraic-geometric codes and the generalization of Theorem 2 to them, the construction and decoding complexity become rather high. Using Reed-Solomon codes together with suitable highly expanding graphs, Guruswami and Indyk [21] present a simple construction of a code over a *fixed* alphabet size that achieves rate $\Omega(\varepsilon^2)$ and can be efficiently list decoded from a fraction $(1-\varepsilon)$ of errors. They also present a construction with rate $\Omega(\varepsilon)$ (and thus is "better" than Reed-Solomon codes), though the decoding complexity becomes sub-exponential (2^{n^γ} for arbitrary $\gamma > 0$) in the block length [21]. This latter result raises the hope that even better codes and algorithms can be obtained by devising non-algebraic approaches to list decoding.

3 Future directions

It is well-known that the *capacity* of the binary symmetric channel with cross-over probability p equals $(1 - H(p))$. In other words, over the channel which flips each bit independently with probability p, one can achieve arbitrarily reliable communication at any rate less than $1-H(p)$. Now consider the noise model where the channel *adversarially* corrupts up to a fraction p of positions. In such a case, "traditional" unique decoding is limited by the half the distance barrier, and thus one has to use codes of relative distance $2p$. In turn, this means one cannot achieve the capacity $1-H(p)$. List decoding exhibits that this limitation is not entirely inherent to the adversarial error model, and can be overcome if one is allowed to output a small list of codewords as answers. In fact, a result due to [20] shows that one can get within ε of the capacity, even under the adversarial model, provided one is permitted list decoding with lists of size $1/\varepsilon$. This raises the intriguing possibility of a "worst-case" theory of information hinging upon list decoding as the basic notion of error-recovery.

The above-mentioned codes from [20] that achieve "capacity" under a worst-case setting are, however, highly non-explicit. An explicit construction of such codes together with efficient list decoding algorithms poses a enormous challenge for future work on list decoding, and constitutes in the authors' mind the single biggest open question in the area. There has been steady progress in this pursuit for the low-rate regime using clever concatenation schemes combined with the weighted Reed-Solomon list decoding algorithm (see, for example, [20]). Nevertheless, we are still very far from any construction of "capacity-approaching" codes that nearly achieve the optimal rate vs. list decoding radius trade-off. Algebraic codes possibly augmented with more sophisticated concatenation-like ideas still hold some promise. But, in light of the recent coding-theoretic developments using combinatorial objects such as "extractors" and "expanders" [41, 21], it is quite possible that non-algebraic approaches will be important in this pursuit.

References

[1] Sigal Ar, Richard Lipton, Ronitt Rubinfeld, and Madhu Sudan. Reconstructing algebraic functions from mixed data. *SIAM Journal on Computing*, 28(2):488–511, 1999.

[2] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 485–495, El Paso, Texas, 4-6 May 1997.

[3] Daniel Augot and Lancelot Pecquet. A Hensel lifting to replace factorization in list decoding of algebraic-geometric and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 46:2605–2613, November 2000.

[4] L.A. Bassalygo. New upper boundes for error-correcting codes. *Problems of Information Transmission*, 1(1):32–35, 1965.

[5] Elwyn Berlekamp. Bounded distance +1 soft-decision Reed-Solomon decoding. *IEEE Transactions on Information Theory*, 42(3):704–720, 1996.

[6] Volodia M. Blinovsky. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22(1):7–19, 1986.

[7] Volodia M. Blinovsky. *Asymptotic Combinatorial Coding Theory.* Kluwer Academic Publishers, Boston, 1997.

[8] Dan Boneh. Finding smooth integers in short intervals using CRT decoding. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 265–272, 2000.

[9] Ilya I. Dumer. Two algorithms for the decoding of linear codes. *Problems of Information Transmission*, 25(1):24–32, 1989.

[10] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[11] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37:5–12, 1991.

[12] Shuhong Gao and M. Amin Shokrollahi. Computing roots of polynomials over function fields of curves. *Coding Theory and Cryptography: From Enigma and Geheimschreiber to Quantum Theory (D. Joyner, Ed.), Springer*, pages 214–228, 2000.

[13] Peter Gemmell and Madhu Sudan. Highly resilient correctors for multivariate polynomials. *Information Processing Letters*, 43(4):169–174, September 1992.

[14] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15-17 May 1989.

[15] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46(5):1330–1338, July 2000. Extended version appears as ECCC Technical Report TR98-062 (Revision 4), `http://www.eccc.uni-trier.de/eccc`.

[16] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM Journal on Discrete Mathematics*, 13(4):535–570, November 2000.

[17] Dima Grigoriev. Factorization of polynomials over a finite field and the solution of systems of algebraic equations. *Translated from Zapiski Nauchnykh Seminarov Lenningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR*, 137:20–79, 1984.

[18] Venkatesan Guruswami. Limits to list decodability of linear codes. *Manuscript*, 2001.

[19] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. PhD thesis, Massachusetts Institute of Technology, August 2001.

[20] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, pages 603–612, October 2000.

[21] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, October 2001.

[22] Venkatesan Guruswami, Amit Sahai, and Madhu Sudan. Soft-decision decoding of Chinese Remainder codes. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 159–168, Redondo Beach, California, 12-14 November 2000.

[23] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[24] Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 181–190, 2000.

[25] Jørn Justesen and Tom Høholdt. Bounds on list decoding of MDS codes. *IEEE Transactions on Information Theory*, 47(4):1604–1609, May 2001.

[26] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM Journal on Computing*, 14(2):469–489, 1985.

[27] Ralf Koetter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, October 2000.

[28] Hendrik W. Lenstra. Codes from algebraic number fields. In L.G.L.T. Meertens M. Hazewinkel, J.K. Lenstra, editor, *Mathematics and computer science II, Fundamental contributions in the Netherlands since 1945*, pages 95–104. North-Holland, Amsterdam, 1986.

[29] R. Matsumoto. On the second step in the Guruswami-Sudan list decoding algorithm for AG-codes. *Technical Report of the Institute of Electronics, Information and Communication Engineers (IEICE)*, pages 65–70, 1999.

[30] Rasmus R. Nielsen. Decoding concatenated codes using Sudan's algorithm. *Manuscript submitted for publication*, May 2000.

[31] Rasmus R. Nielsen and Tom Høholdt. Decoding Hermitian codes with Sudan's algorithm. *Proceedings of AAECC-13, LNCS 1719*, pages 260–270, 1999.

[32] W. Wesley Peterson. Encoding and error-correction procedures for Bose-Chaudhuri codes. *IEEE Transactions on Information Theory*, 6:459–470, 1960.

[33] Ronny Roth and Gitit Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, January 2000.

[34] Claude E. Shannon, Robert G. Gallager, and Elwyn R. Berlekamp. Lower bounds to error probability for coding on discrete memoryless channels. *Information and Control*, 10:65–103 (Part I), 522–552 (Part II), 1967.

[35] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

[36] V. M. Sidelnikov. Decoding Reed-Solomon codes beyond $(d-1)/2$ errors and zeros of multivariate polynomials. *Problems of Information Transmission*, 30(1):44–59, 1994.

[37] Madhu Sudan. *Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems.* PhD thesis, University of California at Berkeley, October 1992. Also appears as *Lecture Notes in Computer Science*, vol. 1001, Springer, 1996.

[38] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[39] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction diameter. *Proceedings of the 35th Annual Allerton Conference on Communication, Control and Computing*, 1997.

[40] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 537–546, 1999.

[41] Amnon Ta-Shma and David Zuckerman. Extractor Codes. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 193–199, July 2001.

[42] Lloyd R. Welch and Elwyn R. Berlekamp. Error correction of algebraic block codes. *US Patent Number 4,633,470*, December 1986.

[43] John M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[44] Xin-Wen Wu and Paul H. Siegel. Efficient list decoding of algebraic geometric codes beyond the error correction bound. *Proceedings of the International Symposium on Information Theory*, June 2000.

[45] Victor V. Zyablov and Mark S. Pinsker. List cascade decoding. *Problems of Information Transmission*, 17(4):29–34, 1981 (in Russian); pp. 236-240 (in English), 1982.