[5] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379–1396, Sept. 1995.
[6] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations. Concepts and Applications of Voronoi Diagrams*. Chichester, U.K.: Wiley, 1992.
[7] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell Syst. Tech. J.*, vol. 38, no. 3, pp. 611–656, May 1959.
[8] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, pp. 228–236, Mar. 1965.
[9] ____, "Group codes for the Gaussian channel," *Bell Syst. Tech. J.*, vol. 47, no. 4, pp. 575–602, Apr. 1968.
[10] G. D. Forney, Jr., "Geometrically uniform codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1241–1260, Sept. 1991.
[11] I. Ingemarsson, "Group codes for the Gaussian channel," in *Topics in Coding Theory. In Honor of Lars H. Zetterberg*, M. Thoma and A. Wyner, Eds. Berlin, Germany: Springer-Verlag, 1989, pp. 73–108.
[12] E. Agrell, "Voronoi regions for binary linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 310–316, Jan. 1996.
[13] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*. Berlin, Germany: Springer-Verlag, 1987.
[14] E. Viterbo and E. Biglieri, "Computing the Voronoi cell of a lattice: The diamond-cutting algorithm," *IEEE Trans. Inform. Theory*, vol. 42, pp. 161–171, Jan. 1996.
[15] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[16] H. J. Landau, "How does a porcupine separate its quills?," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 157–161, Mar. 1971.
[17] T.-Y. Hwang, "Decoding linear block codes for minimizing word error rate," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 733–737, Nov. 1979.
[18] P. Butovitsch, "Classification of signal sets. The classification capability of multi-layer perceptrons and soft-decision decoding of error-correcting codes," Ph.D. dissertation, Roy. Inst. Technol., Stockholm, Sweden, 1994.
[19] P. J. Green and R. Sibson, "Computing Dirichlet tessellations in the plane," *Comput. J.*, vol. 21, no. 2, pp. 168–173, May 1978.
[20] E. Agrell and P. Hedelin, "How to evaluate search methods for vector quantization," in *Proc. Nordic Signal Processing Symp.* (Ålesund, Norway, June 1994), pp. 258–263,.
[21] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
[22] G. D. Forney, Jr., "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1741–1752, Nov. 1994.
[23] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, nos. 3 and 4, pp. 379–423 and 623–656, July and Oct. 1948.
[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[25] J. N. Pierce, "Limit distribution of the minimum distance of random linear codes," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 595–599, Oct. 1967.
[26] W. W. Peterson, *Error-Correcting Codes*. Cambridge, MA: MIT Press and New York: Wiley, 1961.
[27] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, pp. 10–21, Jan. 1949.
[28] D. Slepian, "Some further theory of group codes," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1219–1252, Sept. 1960.
[29] J. H. van Lint, "Repeated-root cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 343–345, Mar. 1991.
[30] G. Castagnoli, "On repeated-root cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 337–342, Mar. 1991.
[31] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[32] R. J. McEliece, E. R. Rodemich, H. Rumsey, Jr., and L. R. Welch, "New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 157–166, Mar. 1977.
[33] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[34] A. Ashikhmin and A. Barg, "Minimal vectors in linear codes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2010–2017, Sept. 1998.

Bounded-Distance Decoding: Algorithms, Decision Regions, and Pseudo Nearest Neighbors

Ofer Amrani and Yair Be'ery, *Member, IEEE*

***Abstract*— For a code $\mathcal{C}$, bounded-distance decoding algorithms perform as optimal algorithms within the balls $B(c)$, centered at the codewords $c \in \mathcal{C}$, with radius equal to half the minimum Euclidean distance of the code. Thus distinct bounded-distance algorithms vary in performance due to their different behavior outside the balls $B(c)$. We investigate this issue by analyzing the decision regions of some known (e.g., GMD) and some new bounded-distance algorithms presented in this work. In particular, we show that there are three distinct types of nearest neighbors and classify them according to their influence on the decision region. Simulation results and computer-generated images of the decision regions are provided to illustrate the analytical results.**

***Index Terms*— Bounded-distance decoding, decision region, nearest neighbors, pseudo nearest neighbors, Voronoi region.**

I. Introduction

While the decision regions of optimal soft-decoding algorithms for block and lattice codes were studied in great detail (cf. [7], [1], respectively, [13], and the references therein), little is known about the decision regions associated with suboptimal algorithms mainly due to the following reasons. The shape of the decision regions of suboptimal algorithms is far less intuitive as demonstrated in this work. Also, the decision regions in the optimal case are determined only by the code, regardless of the specific optimal decoder used, whereas the decision regions of suboptimal decoders are algorithm- as well as code-dependent. The main purpose of this work is to present some analytical and pictorial observations about the decision regions of suboptimal bounded-distance soft-decoding algorithms for block and lattice codes used on additive white Gaussian noise (AWGN) channels.

Although our observations are valid in general, for illustration we consider some known and new bounded-distance soft-decoding algorithms designed for (n, k, d) q-ary linear block codes with rate $\frac{k}{n} \leq 1/2$ and $d = 4$. These include the $(8, 4, 4)$ binary extended Hamming code, the $(6, 3, 4)$ hexacode over F_4, the $(8, 4, 4)$ code over F_4, and the $(8, 4, 4)$ octacode over the ring Z_4. These codes are of some practical importance; for example, decoding the hexacode is the key step in efficient algorithms for decoding the Golay code and the Leech lattice [2], including the most efficient algorithms known [12]. The known algorithms that we consider are generalized minimum-distance (GMD) decoding [8], modified GMD decoding [9], [11], and minimum-distance (optimal) decoding. We also propose three new algorithms tailored for the codes described above, which involve fewer algebraic decoding operations than these known algorithms. The proposed algorithms can be generalized for other code

Manuscript received December 2, 1996; revised April 6, 1998. This work was supported in part by the Consortium for Digital Receiver Technologies, the Ministry of Industry and Commerce, Israel. The material in this correspondence was presented in part at the Mediterranean Workshop on Coding and Information Integrity, Mallorca, Spain, February 1996, and the International Symposium on Information Theory, Ulm, Germany, 1997.

The authors are with the Department of Electrical Engineering–Systems, Tel-Aviv University, Ramat-Aviv 69978, Tel-Aviv, Israel.

Publisher Item Identifier S 0018-9448(98)06739-X.

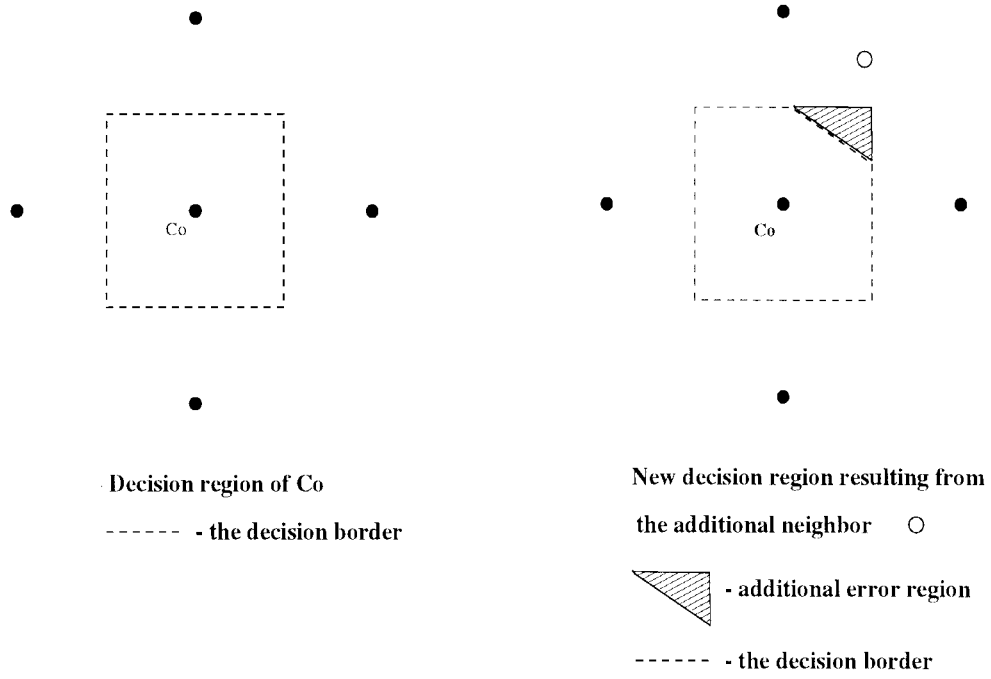

Fig. 1. Additional error region due to an additional noncodeword neighbor.

parameters [4], but these were not of sufficient interest to include here.

We assume that the q-ary alphabet is mapped into an appropriate q-ary signal set in Euclidean space; e.g., symbols from F_q are assumed to be mapped into a q-simplex in R^{q-1} with equal distance 2 between all points. The Euclidean-space image of a codeword $\boldsymbol{c}$ will be denoted by $s(\boldsymbol{c}) \in R^{n(q-1)}$. If the code $\mathcal{C}$ has minimum Hamming distance d, then the minimum squared distance between the images $s(\boldsymbol{c})$ and $s(\boldsymbol{c'})$ of two distinct codewords $\boldsymbol{c}, \boldsymbol{c'} \in \mathcal{C}$ is $4d$, and the squared error-correction radius is $\rho^2 = d$.

A bounded-distance decoder is one for which the decoding region $D(\boldsymbol{c})$ associated with each codeword $\boldsymbol{c} \in \mathcal{C}$ contains the open ball $B(\boldsymbol{c})$ whose boundary is a sphere $S(\boldsymbol{c})$ of squared radius $\rho^2 = d$ centered on $s(\boldsymbol{c}) : B(\boldsymbol{c}) \subseteq D(\boldsymbol{c})$ for all $\boldsymbol{c} \in \mathcal{C}$. We will focus on the closest points to $s(\boldsymbol{c})$ on the boundary of $D(\boldsymbol{c})$, which must of course be at squared distance $\rho^2 = d$ from $s(\boldsymbol{c})$.

A minimum-distance decoder is one whose decision regions are, up to boundary ambiguities, the Voronoi regions $V(\boldsymbol{c})$; i.e., the set of points that are at least as close to $s(\boldsymbol{c})$ as to any other codeword image $s(\boldsymbol{c'})$. A Voronoi region $V(\boldsymbol{c})$ is bounded by portions of separating hyperplanes $H(\boldsymbol{c}, \boldsymbol{c'})$ halfway between $s(\boldsymbol{c})$ and $s(\boldsymbol{c'})$ for certain codewords $\boldsymbol{c'} \in \mathcal{C}$ known as *Voronoi relevant* [7]. The separating hyperplane $H(\boldsymbol{c}, \boldsymbol{c'})$ is tangent to the spheres $S(\boldsymbol{c})$ and $S(\boldsymbol{c'})$ and intersects with them at a single point, the midpoint between $s(\boldsymbol{c})$ and $s(\boldsymbol{c'})$. The number of such boundary points at squared distance $\rho^2 = d$ is the number N_d of codewords $\boldsymbol{c'} \in \mathcal{C}$ at Hamming distance d from $\boldsymbol{c}$. We shall call these "codeword" or "Type I" boundary points.

As many authors have observed, with suboptimal decoding algorithms there are typically additional closest boundary points of $D(\boldsymbol{c})$ lying halfway between $s(\boldsymbol{c})$ and certain points $s(\boldsymbol{x})$, where $\boldsymbol{x} \in (F_q)^n$ is not a codeword in $\mathcal{C}$. We shall call these "noncodeword" or "Type II" boundary points. The number of such boundary points is denoted by N_{BDD}. The "effective error coefficient" $N_{0,\text{eff}}$ is defined as the total number of Types I and II points, i.e., $N_{0,\text{eff}} = N_d + N_{BDD}$, and appears as the coefficient in the union bound estimate of error probability $\Pr(E)$. The geometrical scenario assumed is as depicted in Fig. 1, namely, an additional separating hyperplane per additional noncodeword neighbor.

One of our key observations is that the complement $D^c(\boldsymbol{c})$ of the decision region in the neighborhood of a Type II boundary point is typically a polygonal region that apart from the boundary point $[s(\boldsymbol{c}) + s(\boldsymbol{x})]/2$ lies completely outside the separating hyperplane $H(\boldsymbol{c}, \boldsymbol{x})$. This implies that the contribution to error probability from this neighborhood is less than for a Type I boundary point; in other words, the union bound estimate considerably overestimates $\Pr(E)$ since the number of Type II points is often much higher than Type I points ($N_{BDD} \gg N_d$).

We will also see that different suboptimal algorithms can have the same Type II boundary points, but then their decision regions in the neighborhood of these boundary points can be different; indeed, one can contain the other. This effect can result in significantly different error probabilities for two algorithms that have the same effective error coefficient.

Finally, we will also observe cases in which there are closest boundary points at squared distance $\rho^2 = d$ from $s(\boldsymbol{c})$, where these points are not midpoints between $s(\boldsymbol{c})$ and $s(\boldsymbol{x})$ for any $\boldsymbol{x} \in (F_q)^n$. We call these Type III boundary points, and since they are not associated with any ("conventional" Type I or II) point $\boldsymbol{x} \in (F_q)^n$ we associate them with what we call "pseudo neighbors." Type III boundary points usually appear in a continuous set or a "cluster," forming an m-dimensional subregion of the boundary of $D(\boldsymbol{c})$, where $0 < m < n(q-1)$. In such cases the complement $D^c(\boldsymbol{c})$ of the decision region in the neighborhood of a Type III boundary point is bounded by an m-dimensional subregion of the sphere $S(\boldsymbol{c})$, and the contribution to $\Pr(E)$ of such a Type III cluster is greater than for a Type I boundary point.

Our results suggest that the number of nearest neighbors in a bounded-distance algorithm (as used with the union bound) can be a very misleading measure for performance. Indeed, a bounded-distance algorithm that considerably increases the number of nearest neighbors, may perform very close to the optimum even for high

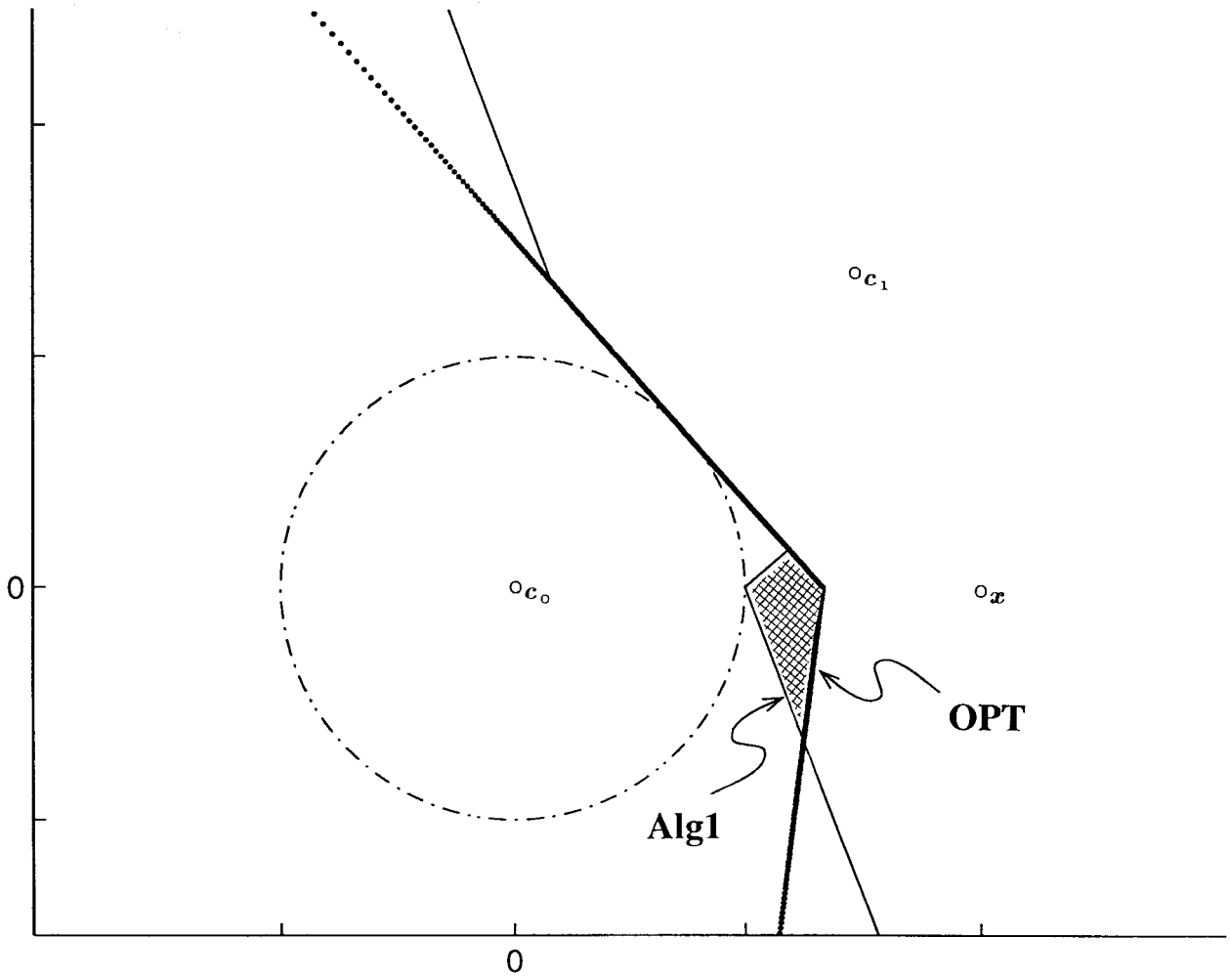

Fig. 2. Computer image of a cross section of the decision region of Alg1, the nearest neighbors are the codeword c_1 and noncodeword x.

noise levels, and can outperform an algorithm which has a smaller number of nearest neighbors.

In Section II, we analyze the decision regions of several bounded-distance decoding algorithms including some new algorithms presented in that section. Computer-generated images of decision regions of these algorithms are included to illustrate the analytical results. In Section III, simulation results for several codes and decoding algorithms are presented and their performance analyzed in terms of our new observations.

II. Bounded-Distance Decoding Algorithms and the Associated Decision Regions

Generalized minimum-distance (GMD) decoding is a well-known bounded-distance decoding algorithm, see [8], [9], [11], and the references therein. A GMD decoder first makes hard decisions on the symbols of the received vector $\boldsymbol{y} \in R^{n(q-1)}$, resulting in hard-decision word $\boldsymbol{z}$, and orders the received symbols in order of reliability. The reliability measure used here is the squared Euclidean distance. Then, it performs a series of algebraic decoding trials, where in each trial a different number of the least reliable symbols are erased. Each trial may result in a candidate codeword. The GMD decoder finally chooses the most reliable among the resulting candidate codewords. For a code with $d = 4$, a GMD decoder performs two decoding trials.

- It performs bounded-Hamming-distance algebraic decoding on the hard-decision word $\boldsymbol{z}$ with the three least reliable symbols erased. This trial produces a candidate codeword $\boldsymbol{c}$ if and only if there exists such a codeword (necessarily unique) that agrees with $\boldsymbol{z}$ in all $n - 3$ unerased positions.
- Same as above, with only the least reliable symbol of $\boldsymbol{z}$ erased. This trial produces a candidate codeword $\boldsymbol{c}'$ if and only if there exists such a codeword (necessarily unique) that agrees with $\boldsymbol{z}$ in all but at most one of the $n - 1$ unerased positions.

If these two decoding trials yield two different codewords $\boldsymbol{c}$ and $\boldsymbol{c}'$, then the GMD decoder finally chooses the one with the minimum Euclidean distance to $\boldsymbol{y}$.

Let $D_{\mathrm{GMD}}(\boldsymbol{c})$ denote the decision regions of a GMD decoder for an (n, k, d) code. It has been shown in [11] for the binary case and in [9] for the general case that the set of closest boundary points of $D_{\mathrm{GMD}}(\boldsymbol{c})$ are precisely the set of midpoints between $s(\boldsymbol{c})$ and $s(\boldsymbol{x})$ for all $\boldsymbol{x} \in (F_q)^n$ at Hamming distance d from $\boldsymbol{c}$. These are the points that have components equal to $s(c_i)$ in $n - d$ positions and $\frac{s(c_i)+s(x_i)}{2}$ in the remaining d positions; in other words, the squared distance is 0 in $n - d$ positions and 1 in the remaining d positions. Thus the effective error coefficient is $N_{0,\,\mathrm{eff}} = \binom{n}{d}(q-1)^d$.

Modified GMD (MGMD) decoding involves performing one additional decoding trial in which the d least reliable symbols are erased and the decoder tries to complete the erased symbols by solving the

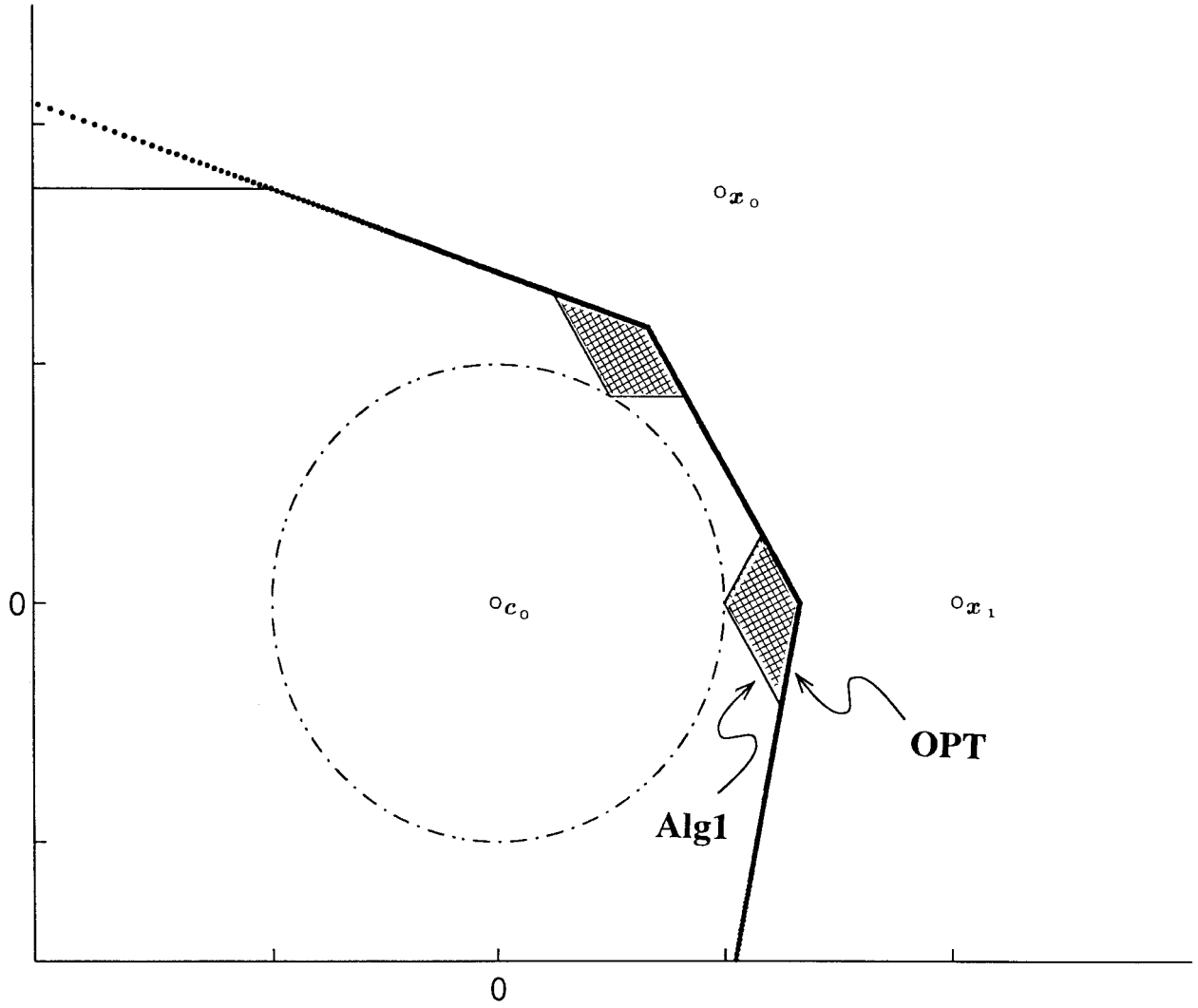

Fig. 3. Computer image of a cross section of the decision region of Alg1, the nearest neighbors are both noncodewords.

corresponding parity-check equations in d unknowns [9], [11]. This trial may result in several codeword solutions. In the nonbinary case, only solutions that agree with either z_i or z_i' (the most likely or the second most likely symbol), in the erased positions are considered [9]. With this modification, all Type II boundary points (which are associated with noncodeword nearest neighbors) are correctly decoded, and the effective error coefficient is thus reduced to the number N_d of Type I boundary points (associated with codeword nearest neighbors), which is the error coefficient of minimum-distance decoding.

We shall now propose several new bounded-distance decoding algorithms that do not involve any algebraic decoding operations, but rather repeated encodings that can be performed by matrix multiplication in F_q. For two of these algorithms, we shall show that if GMD decoding succeeds on a given received n-tuple, then the proposed algorithm must also succeed. For any of these algorithms, this will then imply the following:

- within the Voronoi regions $V(\boldsymbol{c})$, the decision regions $D(\boldsymbol{c})$ contain the GMD decision regions $D_{\mathrm{GMD}}(\boldsymbol{c})$;
- the algorithm is, therefore, bounded-distance;
- the closest boundary points of $D(\boldsymbol{c})$ are a subset of the set of closest boundary points of $D_{\mathrm{GMD}}(\boldsymbol{c})$ described above.

The first algorithm is based on the following observations. We assume that $\mathcal{C}$ is an $(n \geq 2k, k, d)$ linear block code over F_q, and furthermore that there exists a partition of $2k$ coordinates into two information sets of size k. (An information set is a k-tuple such that each k-tuple in $(F_q)^n$ corresponds to a unique codeword in $\mathcal{C}$; since $\mathcal{C}$ is linear, there always exists a generator matrix for $\mathcal{C}$ for which the corresponding k columns form a size k identity matrix.) The squared error-correction radius of the code is d. If $\boldsymbol{c}$ is transmitted and the received point $\boldsymbol{y}$ is in a ball $B(\boldsymbol{c})$ (i.e., if the square Euclidean distance $d^2(\boldsymbol{y}, s(\boldsymbol{c})) < d$) then at most $d-1$ hard decisions can be incorrect in $\boldsymbol{z}$, since a symbol error can occur only if $d^2(y_i, s(c_i)) \geq 1$. An algorithm which can correct all these events is a bounded-distance algorithm. For $d = 4$, if $\boldsymbol{y} \in B(\boldsymbol{c})$, there can be at most three hard-decision errors in $\boldsymbol{z}$, and then, one of the two information sets contains at most one error. For simplicity, we shall describe the new algorithms for $n = 2k$. The case $n \geq 2k$ is then straightforward.

Algorithm 1: Compute the hard-decision word $\boldsymbol{z}$. If $\boldsymbol{z} \in \mathcal{C}$, accept it. Otherwise, partition $\boldsymbol{z}$ into two halves corresponding to the two information sets, $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$. For each of the two information sets and for each of the $(q-1)k+1$ k-tuples z_i', satisfying $d_H(z_i', z_i) \leq 1$, $i = 1, 2$, encode z_i' into the corresponding codeword $c(z_i')$. Select

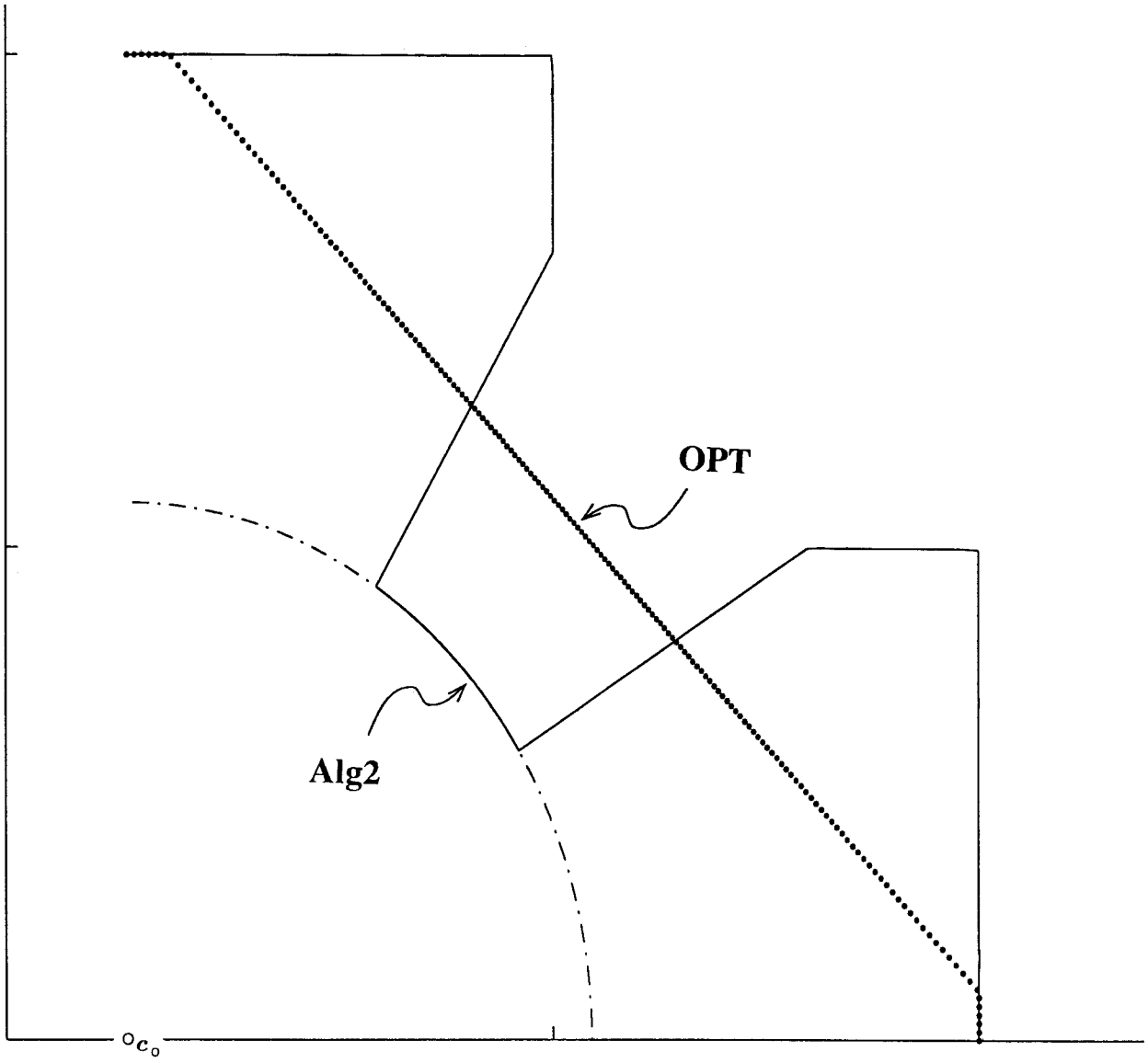

Fig. 4. Computer image of a cross section of the decision region of Alg2, corresponding to a pseudo nearest neighbor.

the best of the resulting candidate codewords using the Euclidean distance metric.

Algorithm 1 will always generate the codeword $\boldsymbol{c}$ as one of the candidates. Since $y \in B(\boldsymbol{c}) \in V(\boldsymbol{c})$, there cannot be any other codeword $\boldsymbol{c}'$ with smaller Euclidean metric, thus $\boldsymbol{c}$ will be selected as the output and hence Algorithm 1 is a bounded-distance algorithm. This property can also be proved from a different angle, by establishing the relation between Algorithm 1 and GMD as follows.

Lemma 1: All candidate codewords generated by a GMD decoder will also be generated by Algorithm 1.

Proof: If the three-erasure trial of GMD decoding produces a codeword $\boldsymbol{c}$, then relative to $\boldsymbol{c}$ there are at most three symbol errors in $\boldsymbol{z}$; thus either $\boldsymbol{z}_1$ or $\boldsymbol{z}_2$ has at most one error and $\boldsymbol{c}$ will be generated in one of the re-encodings of Algorithm 1. Similarly, if the one-erasure trial produces a codeword $\boldsymbol{c}'$, then relative to $\boldsymbol{c}'$ there are at most two symbol errors in $\boldsymbol{z}$ and $\boldsymbol{c}'$ will be generated by Algorithm 1. □

Corollary 1: Within the Voronoi regions $V(\boldsymbol{c})$, the decision regions $D_1(\boldsymbol{c})$ of Algorithm 1 contain the GMD decision regions $D_{\mathrm{GMD}}(\boldsymbol{c})$.

Proof: Within $V(\boldsymbol{c})$, if $\boldsymbol{c}$ is produced as a candidate codeword, it will be chosen over any other candidate codeword in the final selection process of either algorithm. □

The corollary implies that the decision regions $D_1(\boldsymbol{c})$ contain the balls $B(\boldsymbol{c})$, and thus Algorithm 1 is bounded-distance. Furthermore, the closest boundary points of $D_1(\boldsymbol{c})$ have components equal to $s(c_i)$ in $n-4$ positions and halfway between $s(c_i)$ and $s(x_i)$ for some $x_i \neq c_i$ in the remaining four positions. Now, Algorithm 1 can fail only if a) $\boldsymbol{x}$ is a codeword at distance 4 from $\boldsymbol{c}$ (Type I error), or b) there are at least two errors in both $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ (a Type II error, unless $\boldsymbol{x}$ is a codeword). Thus the Type II boundary points of $D_1(\boldsymbol{c})$ are those points in which both halves of $\boldsymbol{y}$ have components equal to $s(c_i)$ in $k-2$ positions and $\frac{s(c_i)+s(x_i)}{2}$ for some $x_i \neq c_i$ in the remaining two positions, where $\boldsymbol{x}$ is not a codeword. There are thus precisely $\binom{k}{2}^2(q-1)^4 - N_{2,2}$ Type II points, where $N_{2,2}$ denotes the number of weight-4 codewords with weight 2 in each information set.

When examining the shape of the error region in the vicinity of a Type II (noncodeword) boundary points, we find that Type II

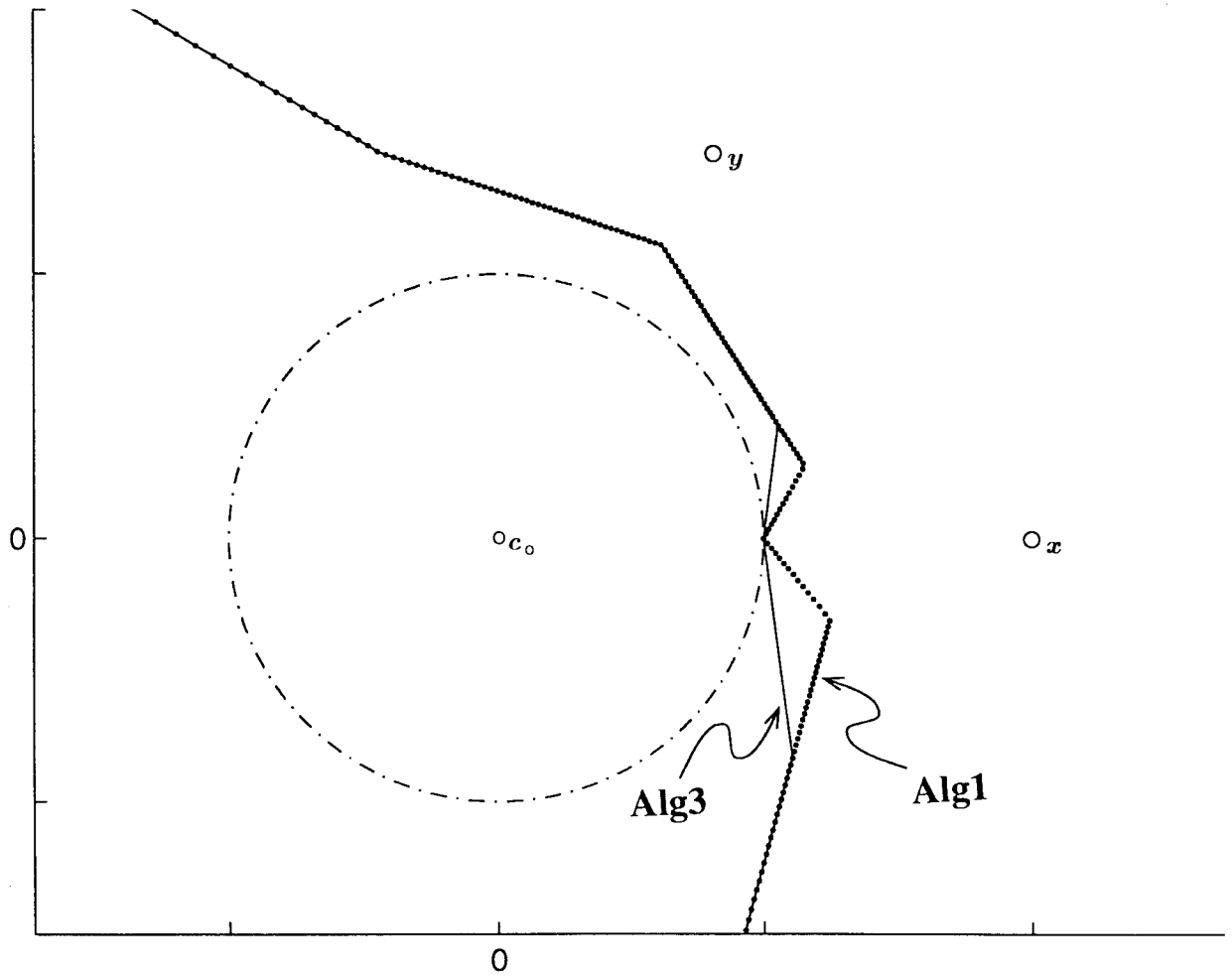

Fig. 5. Computer image of a cross section of the decision region of Alg1 versus Alg3.

points contribute less to error probability than do Type I points. For simplicity, first consider the binary case. Then, in the neighborhood of a Type II boundary point $\boldsymbol{y}_b = \frac{s(\boldsymbol{c})+s(\boldsymbol{x})}{2}$, in both halves $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ (corresponding to information sets) of $\boldsymbol{y}_b$, and in both of the two positions in which $\boldsymbol{x}$ disagrees with $\boldsymbol{c}$ the distances from $s(c_i)$ both have to be greater than one (so that y_i is closer to $s(x_i)$) in order for Algorithm 1 to fail. The boundary surface is thus defined as the set of points for which

$$\begin{aligned} d(y_i,\, s(c_i)) &\geq 1 \\ d(y_{i'},\, s(c_{i'})) &\geq 1 \\ d(y_j,\, s(c_j)) &\geq 1 \\ d(y_{j'},\, s(c_{j'})) &\geq 1 \end{aligned}$$

for the two positions (i, i') and (j, j') in the first and second halves of the coordinates, respectively, where at least one of these positions is fixed to 1. Clearly, (i, i', j, j') correspond to the four positions in which $\boldsymbol{x}$ disagrees with $\boldsymbol{c}$. This boundary surface is evidently piecewise-linear, and lies completely outside the separating hyperplane $H(\boldsymbol{c}, \boldsymbol{x})$, except for the boundary point $\boldsymbol{y}_b$ itself. By contrast, in the vicinity of a Type I point corresponding to a codeword $\boldsymbol{c}'$, the boundary surface is a portion of the separating hyperplane $H(\boldsymbol{c}, \boldsymbol{c}')$. This phenomenon can be further illustrated for the more general q-ary case by the following example.

Example 1: Let ϵ_1 and ϵ_2 be small positive scalars. Assume that the codeword $\boldsymbol{c}$ is transmitted and the point $\boldsymbol{y} \in R^{n(q-1)}$, corresponding to the event below, is received.

- First half $\boldsymbol{z}_1$: one symbol error z_i with distance $d(s(z_i),\, y_i) = 1 - \epsilon_1$ such that $d(s(c_i),\, y_i) = 1 + \epsilon_1$, i.e., y_i is on the line connecting $s(c_i)$ with $s(z_i)$; one correct symbol $z_{i'} = c_{i'}$, with distance $d(s(c_{i'}),\, y_{i'})) = 1 - \epsilon_2$, such that there exists a symbol $\hat{z}_{i'} \in F_q$ with distance $d(s(\hat{z}_{i'},\, y_{i'}) = 1 + \epsilon_2$.
- Second half $\boldsymbol{z}_2$: exactly two symbol errors, z_j and $z_{j'}$ (on the decision border) satisfying

$$\begin{aligned} d(s(z_j),\, y_j) = d(s(c_j),\, y_j) &= d(s(z_{j'}),\, y_{j'}) \\ &= d(s(c_{j'}),\, y_{j'}) = 1. \end{aligned}$$

Consider the word $\boldsymbol{x}$ which contains z_i, $z_{i'}$, z_j, and $z_{j'}$, and is equal to $\boldsymbol{c}$ otherwise. When $\epsilon_1 > \epsilon_2$, the point $\boldsymbol{y}$ is closer to $s(\boldsymbol{x})$ than to $s(\boldsymbol{c})$. Thus if $\boldsymbol{x}$ is a codeword, both $\boldsymbol{x}$ and $\boldsymbol{c}$ will be generated by Algorithm 1, and $\boldsymbol{x}$ will be selected as the output (Type I error). If, however, $\boldsymbol{x}$ is not a codeword, then $\boldsymbol{c}$ will be selected provided that $\boldsymbol{y} \in V(\boldsymbol{c})$. In conclusion, the decision region in the vicinity of Type II points is "better" than Type I points. Note that $\boldsymbol{y}$ actually represents a volume, rather than a singular point, in which this phenomenon occurs as it can quite liberally move in all directions. □

Figs. 2 and 3 illustrate this phenomenon by computer-generated images of cross sections of decision regions of Algorithm 1 for the

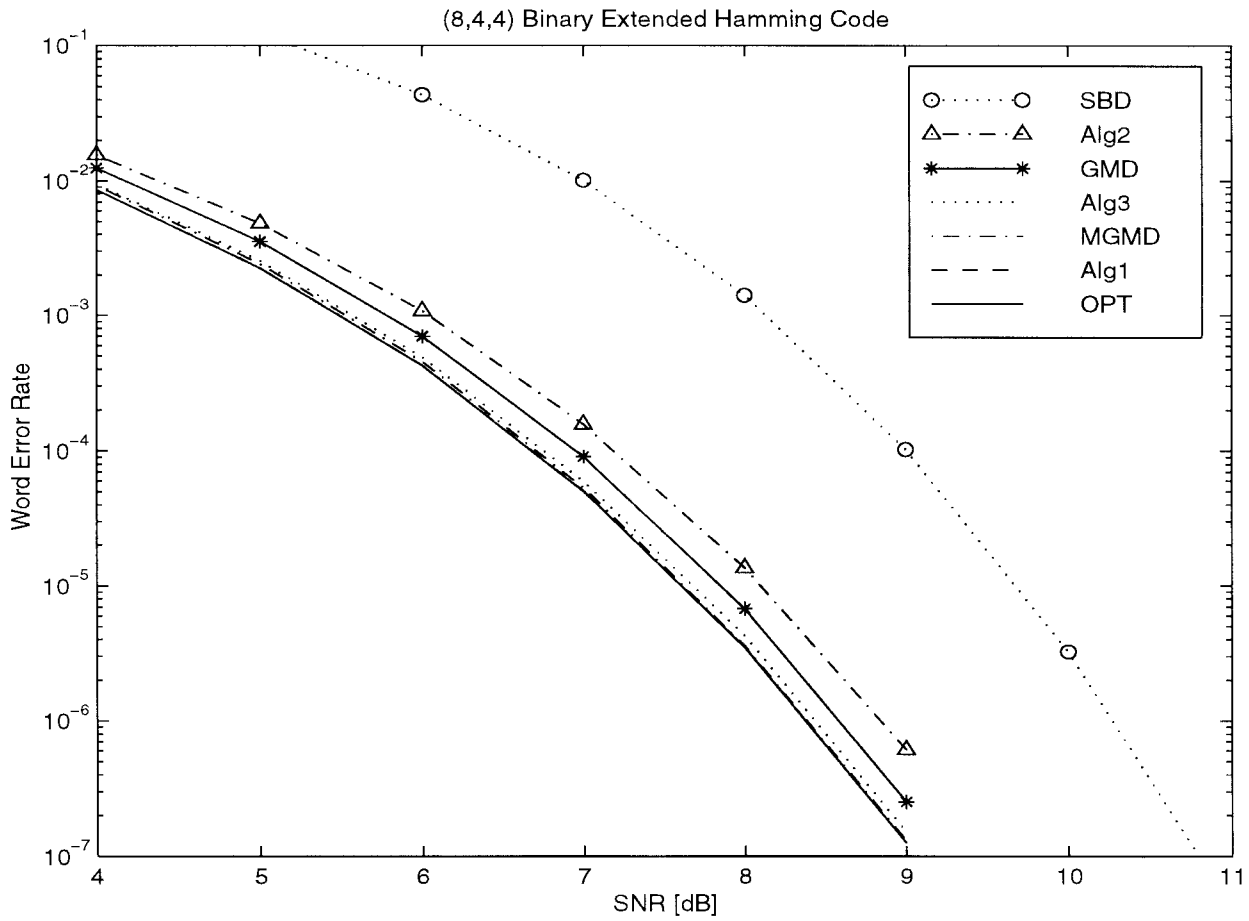

Fig. 6. Computer simulation results for the $(8, 4, 4)$ binary extended Hamming code.

$(8, 4, 4)$ binary extended Hamming code. Fig. 2 lies in a plane that contains the images of two codewords $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ and a noncodeword $\boldsymbol{x}$. The dotted and solid lines represent the boundary of the decision region of c_0 under optimal decoding (OPT) and Algorithm 1, respectively. The dash-dotted line represents the boundary of the ball $B(\boldsymbol{c}_0)$ with radius 2, half the minimum Euclidean distance of the code. The point midway between $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$, $\frac{s(\boldsymbol{c}_0)+s(\boldsymbol{c}_1)}{2}$, is a Type I boundary point, and the decision border is the equidistance line between the codewords (as in optimal decoding). The point $\frac{s(\boldsymbol{c}_0)+s(\boldsymbol{x})}{2}$ is a typical Type II boundary point. The decision region in the vicinity of this point is a polygonal region which clearly "favors" $\boldsymbol{c}_0$. The main contribution to error probability due to this point relative to optimal decoding is the cross-hatched area. From this neighborhood (of a Type II point) the contribution to error probability is less than for a Type I point. Fig. 3 lies in a plane that contains the images of two noncodewords $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ and a codeword $\boldsymbol{c}_0$. Again here, the cross-hatched area represents the difference between Algorithm 1 and optimal decoding. Note that the additional error regions due to the two noncodewords is much smaller than if these two points were codewords.

Now we consider a second algorithm that requires only half the number of encodings of Algorithm 1, which is based on the following lemma.

Lemma 2: If $d^2(\boldsymbol{y}, s(\boldsymbol{c})) < 4$, then the information set that has the better overall metric, has at most one hard-decision error.

Proof: Let $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$ corresponding to the two information sets $(\boldsymbol{z}_1, \boldsymbol{z}_2)$. The overall metric of the set $\boldsymbol{z}_i$, $i = 1, 2$, is $d^2(\boldsymbol{y}_i, s(\boldsymbol{z}_i))$. Now if $d^2(\boldsymbol{y}, s(\boldsymbol{c})) < 4$, then $d^2(\boldsymbol{y}_i, s(\boldsymbol{c}_i)) < 2$ for at least one of $i = 1, 2$. Since an error can occur only if $d^2(y_j, s(c_j)) \geq 1$, there can be at most one symbol error in this information set. □

Algorithm 2: Find the information set with the better overall metric, and continue according to Algorithm 1 with that set only.

This algorithm saves half the number of encodings of Algorithm 1 and by Lemma 2 is bounded-distance. It has, however, Type III boundary points, as may be seen in the following. Suppose that $d^2(\boldsymbol{y}_i, s(\boldsymbol{c}_i)) = 2$ for both halves, where one half, say $\boldsymbol{y}_1$, has components equal to $s(c_i)$ in $k-2$ positions and halfway between $s(c_i)$ and $s(x_i)$ for some $x_i \neq c_i$ in the remaining two positions, and the other half $\boldsymbol{y}_2$ is an arbitrary k-tuple satisfying $d^2(\boldsymbol{y}_2, s(\boldsymbol{c}_2)) = 2$. This defines a multidimensional continuous subregion of the sphere $S(\boldsymbol{c})$ of squared radius 4 about $\boldsymbol{c}$ which is evidently on the boundary of the decision region $D_2(\boldsymbol{c})$ of Algorithm 2. Note that all Type III points on the sphere $S(\boldsymbol{c})$ are interconnected. Namely, there is a trail on the surface of $S(\boldsymbol{c})$ connecting any two Type III points. Moreover, all points of Type I and II with Hamming weight-2 in each information set are also interconnected.

Fig. 4 illustrates this phenomenon by a computer-generated image of a cross section of a decision region of Algorithm 2 for the $(8, 4, 4)$ binary code, in a plane containing the image of a codeword $\boldsymbol{c}_0$. The dotted and solid lines represent the boundary of the decision region of c_0 under optimal decoding and Algorithm 2, respectively. The dash-dotted line represents the boundary of the ball $B(\boldsymbol{c}_0)$ with radius 2.

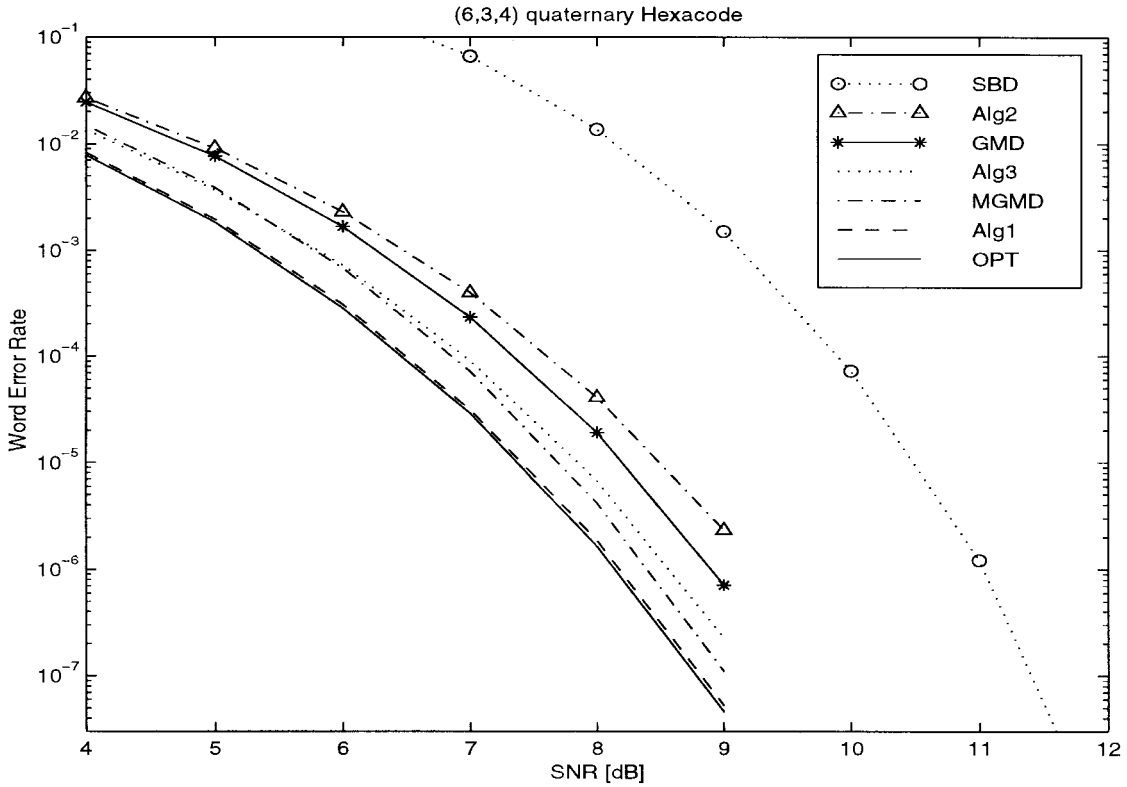

Fig. 7. Computer simulation results for the $(6, 3, 4)$ hexacode over F_4.

Here, the boundary of the decision region of Algorithm 2 lies inside any hyperplane tangent to it, and the contribution to error probability from this continuous cluster of Type III boundary points is worse than if the boundary were a portion of such a hyperplane.

A more interesting and practical modification to Algorithm 1 is suggested by the following lemma.

Lemma 3: If $d^2(\boldsymbol{y}, s(\boldsymbol{c})) < 4$, then in one of the two information sets $(\boldsymbol{z}_1, \boldsymbol{z}_2)$, if there is a single hard-decision error, it must be in the least reliable coordinate.

Proof: If $d^2(\boldsymbol{y}, s(\boldsymbol{c})) < 4$, then $d^2(\boldsymbol{y}_i, s(\boldsymbol{c}_i)) < 2$ for at least one of $i = 1, 2$. Since an error can occur only if $d^2(y_j, s(c_j)) \geq 1$, there can be at most one symbol error in this information set. This error must be in the least reliable position, because if $d(y_i, s(c_i)) = 1 + \epsilon$ and a correct symbol y_i' is no more reliable than y_i, then for the correct symbol $d(y_i', s(c_i')) \geq 1 - \epsilon$ and

$$d^2(y_i, s(c_i)) + d^2(y_i', s(c_i')) \geq (1+\epsilon)^2 + (1-\epsilon)^2 \geq 2$$

contrary to the supposition $d^2(\boldsymbol{y}_i, s(\boldsymbol{c}_i)) < 2$. □

Algorithm 3: Compute the hard-decision word $\boldsymbol{z}$. If $\boldsymbol{z} \in \mathcal{C}$, accept it. Otherwise, partition $\boldsymbol{z}$ into two halves corresponding to the two information sets, $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$. For each of the two sets find the least reliable symbol of $\boldsymbol{z}_i$, $i = 1, 2$, and for each of the q possible words $\boldsymbol{z}_i'$ obtained by substituting all elements of F_q in this symbol, encode $\boldsymbol{z}_i'$ into the corresponding codewords $c(\boldsymbol{z}_i')$. Select the best of the resulting $2q$ candidate codewords using the Euclidean distance metric.

Note that the algorithm obtained by combining Lemma 2 and 3 is not a bounded-distance algorithm.

Lemma 4: All candidate codewords generated by a GMD decoder will also be generated by Algorithm 3.

Proof: If the three-erasure trial of GMD decoding produces a codeword $\boldsymbol{c}$ then relative to $\boldsymbol{c}$ there are at most three symbol errors in $\boldsymbol{z}$; thus either $\boldsymbol{z}_1$ or $\boldsymbol{z}_2$ has at most one error, which moreover must be in the least reliable position of this half, since the three erased symbols are the least reliable symbols of $\boldsymbol{z}$. Thus $\boldsymbol{c}$ will be generated in one of the re-encodings of Algorithm 3. Similarly, if the one-erasure trial produces a codeword $\boldsymbol{c}'$, then relative to $\boldsymbol{c}'$ there are at most two errors in $\boldsymbol{z}$ with one (the erased) being in the least reliable symbol; if both errors fall in the same half, then the other half is correct, whereas if one error falls in each half, then at least one will be in the least reliable position of its half. Therefore, $\boldsymbol{c}'$ will also be generated by Algorithm 3. □

Within $V(\boldsymbol{c})$, if $\boldsymbol{c}$ is produced as a candidate codeword, it will be chosen over any other candidate codeword in the final selection process of either algorithm. Thus we have the following corollary.

Corollary 2: Within the Voronoi regions $V(\boldsymbol{c})$, the decision regions $D_3(\boldsymbol{c})$ of Algorithm 3 contain the GMD decision regions $D_{\mathrm{GMD}}(\boldsymbol{c})$.

Algorithm 3 is a bounded-distance algorithm although considerably less complex than Algorithm 1. Also, it is easy to see that it has the same closest boundary points, and therefore the same effective error coefficient as for Algorithm 1. However, we make the interesting observation that in the neighborhood of these closest boundary points, the decision region $D_3(\boldsymbol{c})$ is strictly contained in $D_1(\boldsymbol{c})$, so that the contribution to error probability of each Type II boundary point of Algorithm 3 is greater. This follows from the fact that the candidate codewords generated by Algorithm 3 are a subset of those generated by Algorithm 1. For example, consider the point $\boldsymbol{y}$, in the vicinity of a Type II boundary point, as given in Example 1. While Algorithm 1 correctly decodes $\boldsymbol{y}$ to $\boldsymbol{c}$, Algorithm 3 will always

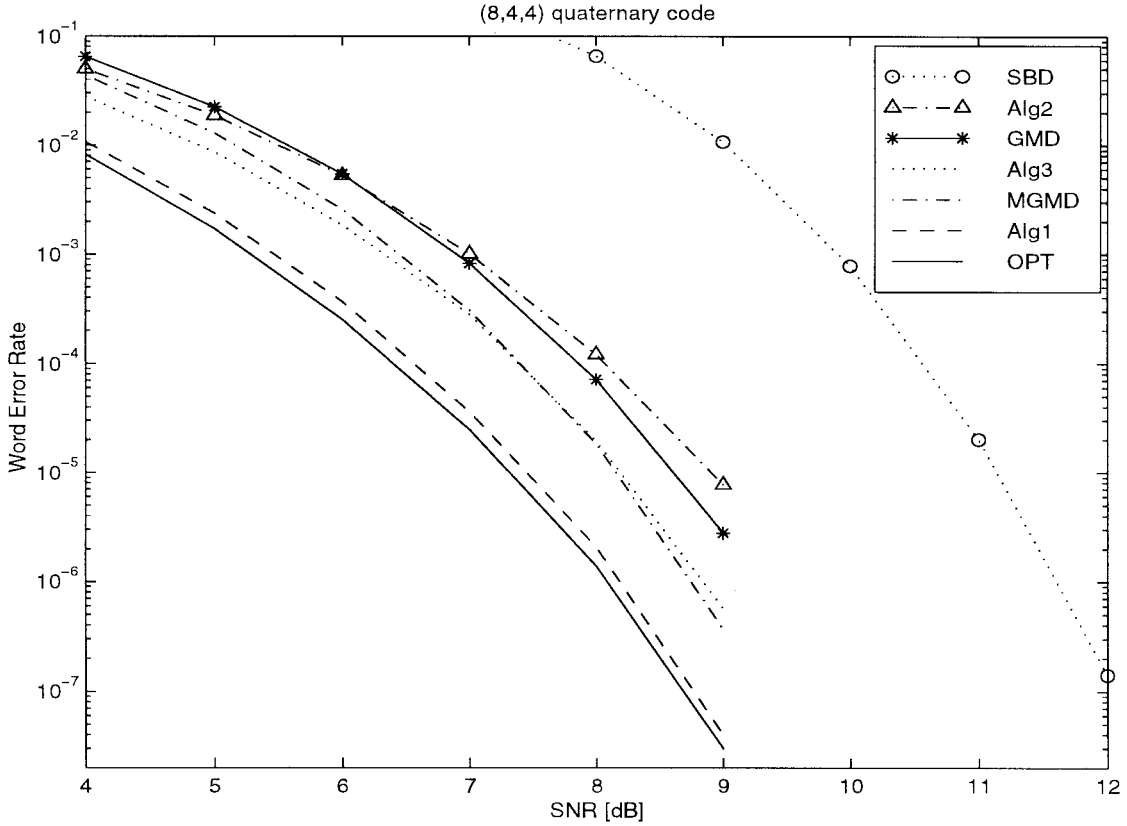

Fig. 8. Computer simulation results for an $(8, 4, 4)$ code over F_4.

fail, since there are two errors in $\boldsymbol{z}_2$ and the least reliable symbol in $\boldsymbol{z}_1$ is the correct symbol $z_i{}'$ rather than the incorrect symbol z_i.

This phenomenon is illustrated in Fig. 5 by a computer-generated image of a cross section of decision regions of Algorithms 1 and 3 for the $(8, 4, 4)$ binary code. The plane shown contains a codeword $\boldsymbol{c}_0$, a noncodeword $\boldsymbol{x}$ which corresponds to a Type II boundary point, and a distance-4 noncodeword $\boldsymbol{y}$ which does not correspond to a Type II boundary point. Clearly, the point $\boldsymbol{x}$ is a noncodeword nearest neighbor of $\boldsymbol{c}_0$ in both algorithms. It has, however, a different effect on their decision regions in the neighborhood of the corresponding Type II boundary point $\frac{s(\boldsymbol{c})_0+s(\boldsymbol{x})}{2}$. Indeed, $D_1(\boldsymbol{c}_0)$ contains $D_3(\boldsymbol{c}_0)$, and thus Algorithm 1 has significantly better error probability, in spite of the fact that both algorithms have the same nearest neighbors (error coefficient).

It can be easily verified by means of the scenario of Example 1 (and also for $d > 4$), that the region $D^c_{\mathrm{GMD}}(\boldsymbol{c})$ of GMD algorithm, in the neighborhood of a (Type II) boundary point which corresponds to noncodeword nearest neighbor, is also typically a polygonal region and thus the boundary of this region is not a separating hyperplane. This is also the situation with other bounded-distance algorithms, such as Chase algorithm 1 and 2 [6]. Elsewhere [5], we show that all candidate codewords generated by a GMD decoder are also generated by Chase Algorithms 1 and 2. Hence, using the same arguments as in the proof of Corollary 1, we conclude that within the Voronoi region $V(\boldsymbol{c})$, the decision regions $D_{\mathrm{CA1}}(\boldsymbol{c})$ and $D_{\mathrm{CA2}}(\boldsymbol{c})$ of Chase Algorithm 1 and 2, respectively, satisfy the following relation: $D_{\mathrm{GMD}}(\boldsymbol{c}) \subset D_{\mathrm{CA2}}(\boldsymbol{c}) \subset D_{\mathrm{CA1}}(\boldsymbol{c})$. This geometrical observation explains why the three algorithms, although having the same nearest neighbors, perform quite differently.

A. The Nearest Neighbors of the Decoding Algorithm of [12]

Type III boundary points also exist in other, previously published, decoding algorithms. As an example, consider the decoder for the hexacode H_6, a $(6, 3, 4)$ MDS code over F_4, given in [12]. It will be described only to the extent required for proving the existence of Type III boundary points.

Let $\boldsymbol{y}$ satisfy $d(\boldsymbol{y}, s(\boldsymbol{c})) = 2$ as in the following: say z_1 in error, with $d(y_1, s(z_1)) = d(y_1, s(c_1)) = 1$; another error, say z_2, with $d(y_2, s(z_2)) = \frac{1}{2}$ and $d(y_2, s(c_2)) = \frac{3}{2}$; three correct symbols, say z_3, z_4, z_5, with $d(y_3, s(z_3)) = d(y_4, s(z_4)) = d(y_5, s(z_5)) = \frac{1}{2}$; z_6 correct with $d(y_6, s(z_6)) = 0$. Upon receiving $\boldsymbol{y}$, the algorithm of [12] first makes hard decisions on the received symbols, resulting in the hard-decision word $\boldsymbol{z}$. Since there are two errors in $\boldsymbol{z}$ relative to $\boldsymbol{c}$ there is no codeword $\boldsymbol{c}'$ at Hamming distance 1 from $\boldsymbol{z}$, and thus it proceeds to Step 3a) where it finds the two most reliable symbols, which moreover are assumed to be correct. The most reliable symbol is z_6, while the second most reliable symbol is chosen among the (equally reliable) symbols z_2, z_3, z_4, z_5, out of which z_2 is incorrect. Therefore, $\boldsymbol{y}$ is clearly a boundary point and since it is not a midpoint between $s(\boldsymbol{c})$ and $s(\boldsymbol{x})$ for any $\boldsymbol{x} \in (F_4)^6$, $\boldsymbol{y}$ is a Type III boundary point.

Each vector $\boldsymbol{y}$ represents a continuous subregion of the sphere $S(\boldsymbol{c})$ of radius 2 about $\boldsymbol{c}$, which is on the boundary of the decision region of $\boldsymbol{c}$ for this algorithm. There are altogether $6 \cdot 5(q-1) \cdot 4(q-1) = 1080$ such different continuous subregions in the algorithm of [12]. Unlike with Algorithm 2, these regions are not interconnected between them, nor are they connected to Type I or II boundary points. It is noteworthy that the GMD hexadecoder, although having the same 1215 Type I and II boundary points, does not have such Type III boundary points.

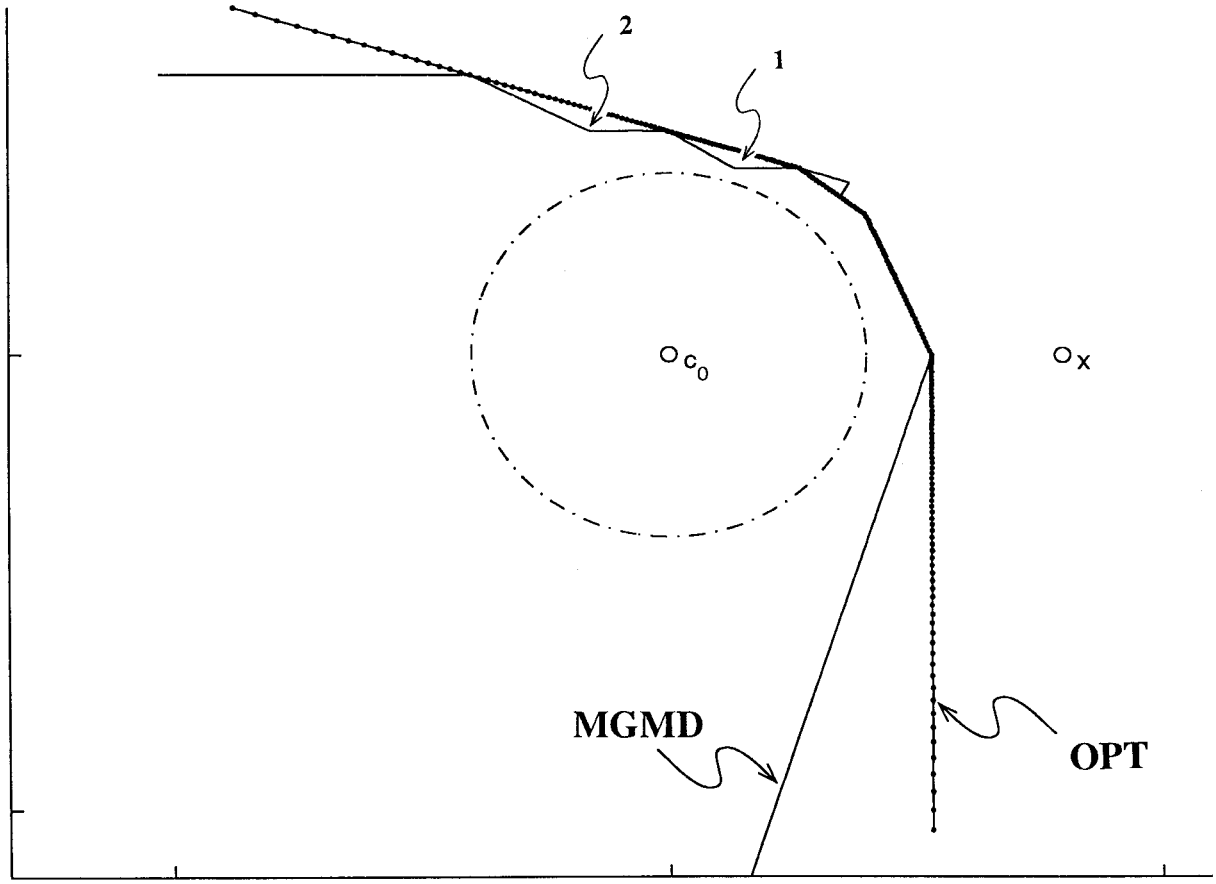

Fig. 9. Computer image of a cross section of the decision region of MGMD.

TABLE I
$(8, 4, 4)$ BINARY CODE, $N_o = 14$

Algorithm	$\frac{N_{o,eff}}{N_o}$	N_{BDD}	Loss [dB]
OPT	1	0	0
Alg1	3.14	30	$\sim$0
Alg2	3.14	(+) 30	0.45
Alg3	3.14	30	0.05
GMD	5	56	0.25
MGMD	1	0	$\sim$0

III. SIMULATION RESULTS AND CONCLUSIONS

Comprehensive computer simulations have been performed for several codes. These include the $(8, 4, 4)$ binary extended Hamming code whose results are presented in Fig. 6; the hexacode, in Fig. 7; and an $(8, 4, 4)$ code over F_4, in Fig. 8. The results demonstrate the **average** effect of the different phenomenon described in this work on the error probability. For each code, the probability of word error is given for a wide range of signal-to-noise ratios, and for each of the following decoding algorithms: *Strictly bounded-distance* algorithm, namely, an algorithm that decodes correctly **only** within the balls $S(\boldsymbol{c})$ of radius 2 about $\boldsymbol{c}$, denoted by SBD; Algorithms 1, 2, and 3 of Section II, denoted by Alg1, Alg2, and Alg3, respectively; a GMD-type algorithm [9], denoted by GMD, in which all generated codewords are accepted as candidates; modified GMD algorithm of [9], denoted by MGMD; and finally, optimal soft-decision decoding, denoted by OPT.

In Tables I–III, for each of the different algorithms, we list the number of nearest neighbors, N_{BDD}, corresponding to Type II boundary points; the factor of increase in the number of nearest neighbors $\frac{N_{o,\text{eff}}}{N_o}$, where $N_{o,\text{eff}} = N_o + N_{BDD}$; and the corresponding coding-gain loss as obtained form the simulations (at WER $\sim 10^{-6}$). The $(+)$, when preceding the number of Type II points, indicates that there are Type III boundary points in addition to N_{BDD} in the corresponding algorithm.

Alg1, Alg2, and Alg3, have the same nearest neighbors of Types I and II, however, their performance are quite different. Alg2 is the worst (Table III reports gain loss of about 1.55 dB) due to the existence of Type III boundary points which contribute to error probability more than do Type I points. Alg1 outperforms Alg3 since, within the Voronoi regions $V(\boldsymbol{c})$ its decision regions $D_1(\boldsymbol{c})$ strictly contain the decision regions $D_3(\boldsymbol{c})$ of Alg3, i.e., the error region in the neighborhood of the same Type II boundary point can be different for different algorithms. In fact, the corresponding difference in performance can be quite significant as can be seen in Tables II and III. It also means that, within the Voronoi regions, there is a lot to gain by decoding correctly outside the bounded-distance balls $B(\boldsymbol{c})$. This last remark is also evident from the curves of the strictly bounded-distance algorithm.

The number of nearest neighbors in Alg1 is much higher than in the case of optimal decoding, nevertheless Alg1 is practically optimal. The reason for this interesting phenomenon is now obvious.

TABLE II
$(6, 3, 4)$ HEXACODE OVER F_4, $N_o = 45$

Algorithm	$\frac{N_{o,eff}}{N_o}$	N_{BDD}	Loss [dB]
OPT	1	0	0
Alg1	16.5	702	$\sim$0
Alg2	16.5	$_{(+)}$702	1.05
Alg3	16.5	702	0.45
GMD	27	1170	0.65
MGMD	1	0	0.25
Alg. of [12]	27	$_{(+1080)}$1170	

TABLE III
$(8, 4, 4)$ CODE OVER F_4, $N_o = 42$

Algorithm	$\frac{N_{o,eff}}{N_o}$	N_{BDD}	Loss [dB]
OPT	1	0	0
Alg1	70	2898	0.1
Alg2	70	$_{(+)}$2898	1.55
Alg3	70	2898	$\sim$0.8
GMD	135	5628	1.25
MGMD	1	0	$\sim$0.8

Indeed, Alg1 increases the number of nearest neighbors (from Type I points only) by N_{BDD} (Type II points), yet each of these additional neighbors contributes very little to the error probability since the error region in the neighborhood of the Type II boundary points is very small compared to Type I points. Also, disregarding decoding complexity issues, although Alg1 has considerably more nearest neighbors, its error probability is always better than MGMD which has the same nearest neighbors as in optimal decoding. Therefore, for bounded-distance algorithms, a smaller number of nearest neighbors does not guarantee better performance, the relevant criterion is the shape of the decision regions. Indeed, MGMD has no Type II or III boundary points at distance 2 (first shell), it has, however, shells of (Type III, noncontinuous) boundary points very close to the first shell, and thus their contribution to error probability is significant. Fig. 9 illustrates this phenomenon by computer-generated image of a cross section of the decision region of MGMD for the $(8, 4, 4)$ binary code, in a plane containing the image of a codeword $\boldsymbol{c}_0$ and a noncodeword x. The dotted line represents the boundary of the decision region of c_0 under optimal decoding and the solid line represents the boundary of the decision region of MGMD decoding. The points labeled 1 and 2 on the solid line are examples of Type III boundary points which are close to the first shell. This notion is supported by the recent results of [10].

ACKNOWLEDGMENT

The authors thank I. Yosef and E. Fishler for the simulations and computer-generated images, respectively. They also thank G. D. Forney and A. Vardy, as well as M. P. C. Fossorier and S. Lin, for preprint of their papers [9], [10]. The authors are indebted to one of the referees for detailed suggestions and comments on an earlier version of the manuscript. O. Amrani wishes to dedicate this work to the memory of Etka Igelski as a token of appreciation for her life-long support and encouragement.

REFERENCES

[1] E. Agrell, "Voronoi regions for binary linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 310–316, 1996.
[2] O. Amrani and Y. Be'ery, "Efficient bounded-distance decoding of the hexacode and associated decoders for the Leech lattice and the Golay code," in *Proc. IEEE Int. Symp. Inform. Theory* (Trondheim, Norway, June 1994).
[3] ——, "Methods of efficient bounded-distance decoding for a family of block codes and associated error correction methods for the hexacode, the Golay code and the Leech lattice," US Patent 5 805 613.
[4] ——, "Bounded-distance decoding algorithms and decision regions," Tech. Rep. EE-S-96-61, Tel-Aviv Univ., Tel-Aviv, Israel, Nov. 1996.
[5] ——, "On the relations between the decision regions of GMD and Chase algorithms," Tech. Rep. EE-S-97-31, Tel-Aviv Univ., Tel-Aviv, Israel, Sept. 1997.
[6] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, 1972.
[7] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. New York: Springer-Verlag, 1992.
[8] G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, 1966.
[9] G. D. Forney, Jr., and A. Vardy, "Generalized minimum distance decoding of Euclidean-space codes and lattices," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1992–2026, 1996.
[10] M. P. C. Fossorier and S. Lin, "A unified method for evaluating the error-correcting radius of reliability-based soft decision algorithms for linear block codes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 691–700, Mar. 1998.
[11] B.-Z. Shen, K. K. Tzeng, and C. Wang, "A bounded-distance decoding algorithm for binary linear block codes achieving the minimum effective error coefficient," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1987–1991, 1996.
[12] A. Vardy, "Even more efficient bounded-distance decoding of the hexacode, the Golay code, and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1495–1499, 1995.
[13] E. Viterbo and E. Biglieri, "Computing the Voronoi cell of a lattice: The diamond-cutting algorithm," *IEEE Trans. Inform. Theory*, vol. 42, pp. 161–171, 1996.