# Towards Optimality in Constellation Labeling

Richard D. Wesel,* Christos Komninakis, and Xueting Liu
University of California, Los Angeles, CA 90095-1594

## Abstract

*Robust trellis codes that perform well on channels with severe, correlated fading require low rates and do not employ uncoded (pass-through) bits. For this trellis code design problem, neither Gray-coding or set-partitioning is well motivated. We propose a new approach to constellation labeling and provide a low-rate 32-QAM trellis code design example to show how the new approach can provide improved metrics at low rates.*

## 1 Introduction

For low-rate trellis codes that do not utilize uncoded bits, there is no compelling structural motivation to use either Gray coding [1, see references therein] or set partitioning [2, 3, 4]. We present a new approach to constellation labeling that results in improved low-rate trellis codes.

Section 2 reviews the trellis code design problem. Section 3 examines ways to compare trellis codes and subsequently constellations. Section 4 uses edge-profile maximization to identify good constellation labelings for low-rate codes using 4-PSK, 8-PSK, and 16-QAM. Section 5 presents a 32-QAM labeling that has a better edge profile than can be achieved with any set-partitioning. Section 6 uses this labeling to design low rate trellis codes with improved metrics.

## 2 Trellis Code Design

Any two-dimensional trellis code is fully described by enumerating the set of valid sequences of constellation points and the mapping of input information sequences to that set. Both the set of sequences and the mapping to that set are important factors to code performance, but they can be addressed separately. Theoretically, any mapping of information sequences to a given set of output sequences can be achieved by preceding the encoder with the appropriate bijective mapping from the set of information sequences to itself. In practice, the choice between various “equivalent” feedback and feedforward encoder structures selects the particular mapping of information sequences to output sequences. In any case, this paper restricts attention to the search for the best possible set of output sequences.

A coded modulation scheme can be separated into a binary encoder and a signal mapper that maps the binary code symbols to constellation points. For a linear binary encoder, the set of valid binary symbol sequences and the set of possible binary symbol error sequences are exactly the same set. Thus, checking all possible binary symbol error sequences can be accomplished by examining only the set of valid code sequences rather than considering every possible pair of code sequences. For convolutional codes, this exhaustive search of code sequences can be accomplished efficiently for any additive metric using the Larsen algorithm [5].

However, the binary symbol errors do not directly map to Euclidean metrics. Consider the constellation to be a fully connected graph with an **edge** between pair of constellation points. The binary symbol error that corresponds to an edge is its **edge label**. For a constellation with $2^n$ points, each binary symbol error labels $2^{n-1}$ edges. The edges with a given binary symbol error may not all have the same Euclidean distance, as illustrated in Figure 1.

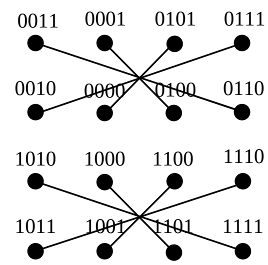



Figure 1: Symbol errors (edges) associated with the binary symbol error (edge label) 0101.

Typically, code searches simply assign to each binary symbol the Euclidean distance of the smallest edge labeled by that binary symbol [2, 3, 6]. This worst-case distance assignment allows the Larsen algorithm to be applied by forcing the code search to consider only the worst possible Euclidean distance sequence for each binary symbol error sequence. This approach can be pessimistic, but often it is not. The worst-case Euclidean distance sequences are guaranteed to exist for many common code rates and constellation sizes [2].

## 3 Codes and Constellations

A constellation labeling is as good as the best code that can be achieved with that labeling. Comparing labelings requires the ability to compare codes. We define three forms of code/constellation equivalence and one relative for comparison.

**Definition 1** *Two codes are strictly equivalent if they implement exactly the same mapping of input information sequences to output sequences.*

**Definition 2** *Two codes are range equivalent if the have the same set of possible output sequences*

Two different binary encoders combined with two different labelings may implement two strictly equivalent codes. Range equivalence is the notion of equivalence for convolutional codes considered in [7].

*This work was supported by grants from the UCLA Academic Senate and the Rockwell Corporation. email: *wesel@ee.ucla.edu* URL:http://www.ee.ucla.edu/faculty/Wesel.html

**Definition 3** *Two codes are distance equivalent if they have the same set of worst-case distance sequences.*

Distance equivalence is the weakest equivalence of the three, but it is the most important for practical code design. The metrics of Euclidean distance [2], product distance, effective code length [8, 9], periodic product distance, and periodic effective code length [10, 11] are all computed from the set of Euclidean distance sequences. Algorithms that use the worst-case distance assignment discussed at the end of Section 2 produce the same value of any of these metrics when applied to two distance-equivalent codes.

**Definition 4** *Two labeled constellations $\mathcal{C}_1$ and $\mathcal{C}_2$ are strictly, range, or distance equivalent if for any code that uses labeled constellation $\mathcal{C}_1$ there is respectively a strictly, range, or distance equivalent code that uses labeled constellation $\mathcal{C}_2$.*

Any two labeled constellations whose edge labels are related by a change of basis are distance equivalent. If two such constellations label the same point with the zero label, then they are strictly equivalent.

**Definition 5** *Code $G_1$ is distance superior to code $G_2$ if the worst-case distance sequences of $G_1$ can be paired bijectively with those of $G_2$ such that each term in every distance sequence of $G_1$ is greater than or equal to the corresponding term in the corresponding sequence for $G_2$. There must be at least one strict inequality, otherwise the trellis codes are distance equivalent.*

**Definition 6** *Constellation $\mathcal{C}_1$ is distance superior to $\mathcal{C}_2$ if for every code $G_2$ designed using $\mathcal{C}_2$ there is a distance superior code $G_1$ defined using $\mathcal{C}_1$.*

**Definition 7** *A labeling is distance superlative if it is superior to all but distance equivalent labelings.*

Unfortunately, distance superlative labelings do not exist for every constellation size. In general, proving distance superiority of one labeling over another is difficult, and no technique either constructive or exhaustive is known for directly identifying a distance superlative labeling.

## 4 The Edge Profile

A necessary condition for a distance-superlative labeling is based on the edge profile of a labeled constellation, which is defined below.

**Definition 8** *The edge profile of a labeled constellation is the monotonically increasing list of minimum squared distances, one for each nonzero edge label.*

Examining Figure 1, the minimum squared distance associated with the edge label 0101 is 2, assuming that the nearest points are separated by unit distance. Table 1 presents the entire edge profile for this labeled constellation.

The edge profile imposes a partial ordering on labelings according to the following definitions.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 5 | 5 | 5 | 8 |

Table 1: Edge profile for constellation in Figure 1.

**Definition 9** *$\mathcal{C}_1$ is profile superior to $\mathcal{C}_2$ if the edge profile for $\mathcal{C}_1$ is element by element greater than or equal to the edge profile for $\mathcal{C}_2$, with at least one strict inequality.*

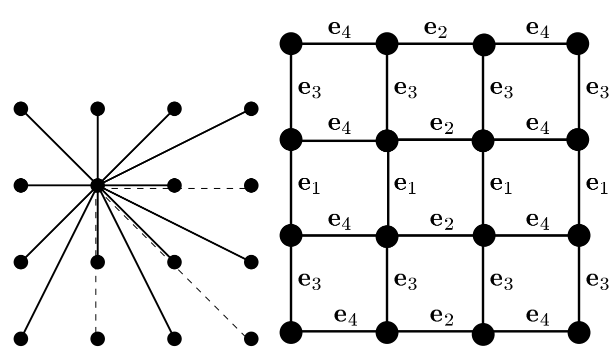
**Definition 10** *A profile-superlative constellation is profile superior to all labelings except those with the same profile.*

Either one of the profile superlative labelings is distance superlative or no distance-superlative labeling exists. We identify all profile superlative labelings for 4-PSK, 8-PSK, and 16-QAM constellations. For detailed proofs that the labelings presented below are profile superlative and that all profile superior labelings are distance equivalent for these constellations, see [10].

All 4-PSK labelings are profile superlative; they all have exactly the same edge profile (1 1 2). In fact, all 4-PSK labelings are distance equivalent and within a rotation or reflection of being strictly equivalent.

For 8-PSK and 16-QAM we upper bound the possible edge profiles and then show that the upper bound is achieved only by distance equivalent labelings. To construct the edge-profile bound, the set of edges emanating from a central point are considered. Regardless of the particular labeling, each edge in this set has a distinct label, and every label appears exactly once. Thus the squared lengths of these edges produce an upper bound on the edge profile.

Figure 2(a) shows the edges used to construct an edge profile bound for the 16-QAM constellation. The bounding

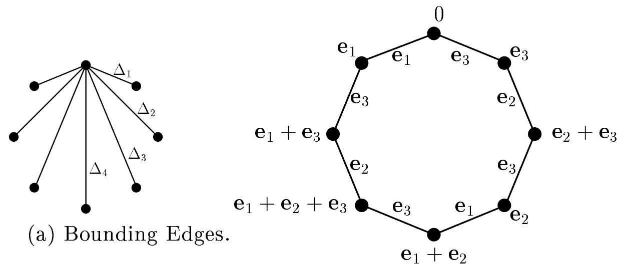



(a) Bounding Edges. (b) Optimal Structure.

Figure 2: 16-QAM bounding edges and edge labeling.

profile resulting from the edges in Figure 2(a) is the same as in Table 1. All labeled 16-QAM constellations that achieve this profile are distance equivalent; they all have the structure shown in Figure 2(b) where $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$ are any basis for the edge labels.

A similar procedure can be applied to 8-PSK. Figure 3(a) shows the edges emanating from a single point and identifies the four possible squared distances as $\Delta_1, \Delta_2, \Delta_3$, and $\Delta_4$. The first row of Table 2 shows the profile bound corresponding to Figure 3(a).




(a) Bounding Edges.

(b) Optimal Structure.

Figure 3: 8-PSK Bounding edges and edge labeling.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Figure 3(a) | $\Delta_1$ | $\Delta_1$ | $\Delta_2$ | $\Delta_2$ | $\Delta_3$ | $\Delta_3$ | $\Delta_4$ |
| Best Achievable | $\Delta_1$ | $\Delta_1$ | $\Delta_1$ | $\Delta_2$ | $\Delta_2$ | $\Delta_3$ | $\Delta_4$ |

Table 2: Edge profile bounds for 8-PSK

Every 8-PSK constellation must have at least three edge labels with minimum squared distance $\Delta_1$ since these minimum distance edge labels must form a basis. Further argument demonstrates that the fifth element in any 8-PSK edge profile cannot be larger than $\Delta_2$. These two observations produce the bottom edge profile bound shown in Table 2. All the constellations that achieve this improved bound are distance equivalent; they all share the structure of Figure 3(b) with $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ being any basis for the three-bit edge labels.

## 5 32-QAM

For 32-QAM, we have not yet proven a labeling strategy to be profile superlative. However, we can show that the best known achievable profile is superior to what can be obtained by either Gray coding or set partitioning. Gray coding is impossible for 32-QAM [10], and we show that set-partitioning forces a sub-optimal edge profile.

Enumerating the edges from a central point in the 32-QAM constellation produces the upper bound on the edge profile shown in the top row of Table 3. Since it is impossible to Gray code the 32-QAM constellation, there are at least six distinct edge labels for the unit-distance edges. Thus the edge profile must begin with six ones as shown in the second row of Table 3.

The edge profile bound given in the second row is not likely to be achievable, but more investigation is required to tighten it further. By seeking a labeling that has only the required six edge labels corresponding to unit distance, we found the labeling structure shown in Figure 4. Edge labels $\mathbf{e}_1$ through $\mathbf{e}_5$ are any basis for the five-bit edge labels and $\mathbf{e}_6 = \mathbf{e}_3 \oplus \mathbf{e}_4 \oplus \mathbf{e}_5$. The edge profile for a labeling with this structure is given in the third row of Table 3. The labeling structure in Figure 4 has the least possible number of unit-distance edge labels, and it's last four elements are as large as they can be. One labeling with the structure of Figure 4 is given (in octal) in Figure 5

The edge profile obtained by Figure 4 cannot be achieved with set-partitioning. The fourth row in Table 3 gives the best edge profile we found using set-partitioning. Rather than exhaustively search all possible set-partitionings, we show that the structure imposed by set-partitioning of 32-QAM restricts the maximum edge profile in at least two ways. With set partitioning, the last two elements in the profile can be at most 10 (rather than 13 which is achieved by Figure 4), and the first seven elements of the edge profile must be 1 (only six elements are 1 for Figure 4).

We prove these two restrictions for 32-QAM partitioned at least to the C-level (8 cosets). The three-bit edge labels implied by the coset labeling are called the coset edge labels. The complete 5-bit edge label is not completely determined by the coset edge label. We begin with a lemma partially characterizing the structure imposed by set-partitioning.

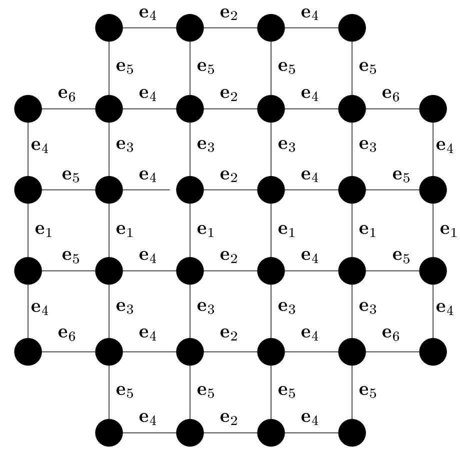



Figure 4: Labeling structure that achieves best known 32-QAM profile.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 23 | 21 | 25 | 27 | | |
| 20 | 03 | 01 | 05 | 07 | 24 | |
| 22 | 02 | 00 | 04 | 06 | 26 | (1) |
| 32 | 12 | 10 | 14 | 16 | 36 | |
| 30 | 13 | 11 | 15 | 17 | 34 | |
| | 33 | 31 | 35 | 37 | | |

Figure 5: A Labeling with the structure of Figure 4.

**Lemma 1** *Set-partitioning the 32-QAM constellation to eight cosets causes unit-distance horizontal (or vertical) coset edge labels alternate between exactly two different three-bit labels, which differ only in the third (the most significant) bit. Together, the vertical and horizontal unit-distance coset edge labels consist of four distinct labels.*

**Proof:** This lemma follows from an examination of the partitioning tree for 32-QAM such as the one shown in [3, page 13]. In this tree the A-level coset is the entire constellation. B-level cosets have sixteen points whose labels all share the same least significant bit. C-level cosets have eight points whose labels all share the same two least significant bits. D-level cosets have four points whose labels all share the same three least significant bits.

| Edges in Figure 4(a) | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 9 | 9 | 10 | 10 | 10 | 10 | 13 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tighter bound | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 9 | 9 | 10 | 10 | 10 | 10 | 13 | 13 |
| Best known | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 10 | 10 | 13 | 13 |
| A set-partitioning | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 9 | 10 |

Table 3: 32-QAM edge profiles.

Consider the nearest neighbors of a point in the C-level coset C0. These nearest neighbors will all be in B1, the B-level coset that does not contain C0. Either the horizontal nearest neighbors are in C1 and the vertical neighbors are in C3, or vice versa. Because all neighbors of a particular orientation (horizontal or vertical) are in the same C-level coset, they have their two least significant bits in common. Thus the unit-distance edges of a given orientation connecting a point in C0 to these neighbors have coset edge labels that differ only in the third bit. The horizontal and vertical unit-distance edges emanating from C2 have the same labels as the respective edges emanating from points in C0 by symmetry. Thus all the unit distance edges have labels that obey the lemma. □

We now use this lemma to prove two earlier statements about the edge-profile limitations of a set-partitioned 32-QAM constellation.

**Lemma 2** *Set-partitioned 32-QAM has no edge label with minimum normalized squared distance greater than 10.*

**Proof:** For a 32-QAM constellation normalized so that the closest points are separated by unit distance, consider the central point shown inside a square in Figure 6. Every non-zero five-bit pattern must label exactly one of the 31 edges emanating from this point. The largest squared distance for an edge emanating from this point is 13 (occurring twice), and the next largest squared distance is 10.

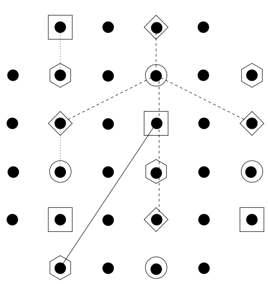


Figure 6: Proof of Lemma 2.

To show that set partitioning forces all edge labels to have minimum distance of at most 10, we need only show that the five-bit labels for the two edges with squared distance 13 emanating from the central square point must also be labels for edges with squared distances of 10 or less.

The alternating behavior proved in Lemma 1 implies that with set partitioning, the three least significant bits (LSBs) of the five-bit edge labels for the two dotted edges in Figure 6 must be identical. Every point in a given coset has the same three least significant bits. Thus all labels for edges connecting a diamond point to a circle point have the same three LSBs, and all labels for edges connecting a hexagon point to a square point have these same three LSBs.

With only two unspecified bits, there are only four labels with the three LSBs of interest. Each of these four labels must label exactly one of the four dashed edges shown in Figure 6 emanating from the circle point. The largest of these edges has squared distance 10. One of these edge labels must be the same as the label of the solid-line edge with squared distance 13 connecting the central square point to a hexagon point. Thus, the edge label of this edge has minimum squared distance 10 or less.

The same argument applies to the other edge with squared distance 13 emanating from the central square point. Thus set partitioning forces the minimum distance associated with every edge label to be less than or equal to 10. □

**Lemma 3** *Set-partitioned 32-QAM has at least seven distinct edge labels for minimum-distance edges.*

**Proof:** From the Lemma 1 there are two horizontal and two vertical three-bit unit-distance coset edge labels $\mathbf{h}_1$, $\mathbf{h}_2$, $\mathbf{v}_1$, and $\mathbf{v}_2$ with $\mathbf{h}_1 \oplus \mathbf{h}_2 = \mathbf{v}_1 \oplus \mathbf{v}_2 = 100$. Assign coset edge labels for the unit-distance edges between the points with five-bit labels $b_1$, $b_2$, and $b_3$ as shown in Figure 7. Coset edge labels only fix the three least significant bits of the edge label.

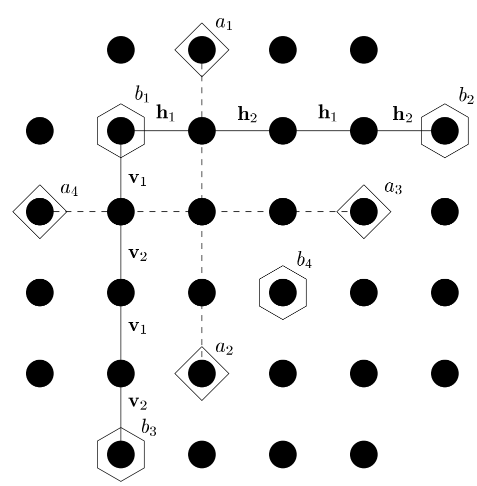



Figure 7: Proof of Lemma 3.

Since $b_1 \oplus b_2 \neq 000$, either the two $\mathbf{h}_1$ coset edges or the two $\mathbf{h}_2$ coset edges have different 2MSBs. This implies the existence of a fifth distinct label for a unit-distance edge. The same reasoning applies to $b_1$ and $b_3$, demonstrating a sixth distinct label.

If both the two $\mathbf{h}_1$ coset edges and the two $\mathbf{h}_2$ coset edges have different 2MSBs then the the argument is complete.

The same reasoning applies to the vertical coset edges $\mathbf{v}_1$ and $\mathbf{v}_2$. For the rest of the argument, assume that all $\mathbf{h}_1$ coset edges have the same 2 MSBs and that all $\mathbf{v}_1$ coset edges have the same 2 MSBs.

Since $b_2$ and $b_3$ are distinct point labels,

$$b_1 \oplus b_3 \neq b_1 \oplus b_2 \,. \tag{2}$$

Thus there must be unit distance edge labels such that two $\mathbf{h}_2$ edge labels and two $\mathbf{v}_2$ edge labels have distinct nonzero exclusive-or's. However, the coset structure also forces

$$a_1 \oplus a_2 = a_3 \oplus a_4 \neq 0 \,. \tag{3}$$

Thus there must be unit distance edge labels such that two $\mathbf{h}_2$ edge labels and two $\mathbf{v}_2$ edge labels have the same nonzero exclusive-or. Equations (2) and (3) force at least five distinct edge labels having $\mathbf{h}_2$ or $\mathbf{v}_2$ as least significant bits. Combined with one $\mathbf{h}_1$ edge label and one $\mathbf{v}_1$ edge label, we have at least seven distinct labels for the unit-distance edges. □

## 6 Code Design Results

Low rate trellis codes that are robust to severe fading were the original motivation for identifying constellation labelings with good edge profiles. As shown in [11], robust trellis codes for a specific periodic interleaver can be obtained by maximizing the periodic effective code length (PECL) and the code periodic product distances (CPPDs) with orders ranging from the PECL to the interleaver period. The choice of constellation labeling does not affect the periodic effective code length, but it does affect the periodic product distances.

Consider the example of a rate-1/5 32-QAM trellis code designed to maximize PECL and the CPPDs for a period 6 interleaver using 4 memory elements. The largest possible PECL is 5. We seek a code with PECL 5 for which the CPPDs of order 5 and 6 are as large as possible. Table 4 shows the largest values of the sum of the logs of $\text{CPPD}_5$ and $\text{CPPD}_6$ obtained with each of the two labeling strategies. As expected, the superior edge profile yields better CPPD values.

| | $\log_2 \text{CPPD}_5$ | $\log_2 \text{CPPD}_6$ | Sum |
|---|---|---|---|
| Set partitioning | 14.8138 | 11.3219 | 26.1357 |
| Figure 5 labeling | 16.0447 | 13.1674 | 29.2122 |
| Gain | 1.2317 | 1.8455 | 3.0765 |

Table 4: Best log sum of order 5 and order 6 code periodic product distance (cppd) with four memory elements for rate-1/5 32-QAM trellis codes using set partitioning and the labeling of Figure 5.

Table 5 shows the largest free Euclidean distance obtained by trellis codes with four memory elements at rates 1/5 and 4/5 produced with each of the two labeling strategies.

The superior profile provides 0.87 dB more free Euclidean distance than set partitioning for the rate 1/5 trellis code design. However, the superior profile provides 0.79 dB *less* Euclidean distance than set partitioning for the rate 4/5 trellis code design. We conjecture that to maximize free distance for a rate $k/n$ trellis code, the labeling should maximize the largest minimum distance in a $2^k$ element algebraic coset of edge labels. When $k = 1$ this corresponds to maximizing the edge profile. When $k = n - 1$ this corresponds to set partitioning.

| Rate | 1/5 | 4/5 |
|---|---|---|
| Set partitioning | 36 | 6 |
| Figure 5 labeling | 44 | 4 |
| Gain | 0.87 dB | -1.76 dB |

Table 5: Best free Euclidean distances with four memory elements for rate-1/5 and rate-4/5 32-QAM trellis codes using set partitioning and the labeling of Figure 5. The values 4 and -1.76 dB are corrections to the mini-conference proceedings.

## 7 Conclusions

For low-rate trellis codes, maximizing the edge profile of a constellation can provide improved metrics compared to set partitioning and Gray coding. However, profile superior constellations can cause reduced Euclidean distance relative to set partitioning for high rate codes. We conjecture that there is a general rate-dependent labeling policy that includes edge-profile maximization and set partitioning as extreme points.

## References


[1] K. Zeger and A. Gersho. Psueudo-Gray Coding. *IEEE Trans. on Comm.*, 38(12):2147–2158, Dec. 1990.

[2] G. Ungerboeck. Channel Coding with Multilevel/Phase Signals. *IEEE Trans. on Inform. Theory*, 28(1):55–67, Jan. 1982.

[3] G. Ungerboeck. Trellis-Coded Modulation with Redundant Signal Sets: Parts I and II. *IEEE Comm. Mag.*, 25(2):5–20, Feb. 1987.

[4] G. D. Forney, Jr. Coset Codes – Part I: Introduction and Geometrical Classification. *IEEE Trans. on Inform. Theory*, 34(5):1123–1141, Sept. 1988.

[5] K. Larsen. Comments on 'An Efficient Algorithm for Computing Free Distance'. *IEEE Transactions on Info. Theory*, 19(4):577–579, July 1973.

[6] J. Du, Y. Kamio, H. Sasaoka, and B. Vucetic. New 32–QAM Trellis Codes for Fading Channels. *Electronics Letters*, 29(20):1745–1746, Sept. 1993.

[7] G. D. Forney, Jr. Convolutional Codes I: Algebraic Structure. *IEEE Trans. on Inform. Theory*, 16(6):720–738, Nov. 1970.

[8] S. G. Wilson and Y. S. Leung. Trellis Coded Phase Modulation on Rayleigh Channels. In *Proceedings of ICC–87*, pages 739–743, June 1987.

[9] D. Divsalar and M. K. Simon. The Design of Trellis Coded MPSK for Fading Channels: Performance Criteria. *IEEE Trans. on Comm.*, 36(9):1004–1012, September 1988.

[10] R. D. Wesel. *Trellis Code Design for Correlated Fading and Achievable Rates for Tomlinson Harashima Precoding*. PhD thesis, Stanford University, Aug. 1996.

[11] R. D. Wesel and J. M. Cioffi. Joint Trellis code and Interleaver Design. In *Globecom '97*, Nov. 1997.