# Optimal Sectionalization of a Trellis

IEEE TRANSACTIONS ON INFORMATION THEORY. VOL.42 NO.3 pp.689-703 MAY 1996

Alec Lafourcade and Alexander Vardy

National Chiao-Tung University ,
Communication Engineering
u8913522 Sheng-Shyan Wang

# Outline

- Background
- Motive
- Key point
- Achievement

# Background

While the complexity of trellis decoding for a given block code is essentially a function of the number of states and branches in its trellis, the decoding complexity may be often reduced by means of an appropriate sectionalization of the trellis.

# Motive

Notwithstanding the many examples of “good” sectionalizations for particular codes that may be found in some literature. However,no systematic method for finding the optimal sectionalization of a given trellis is presently known.

# Key point

- What is sectionalization?
- What is objective function?
- The Algorithm

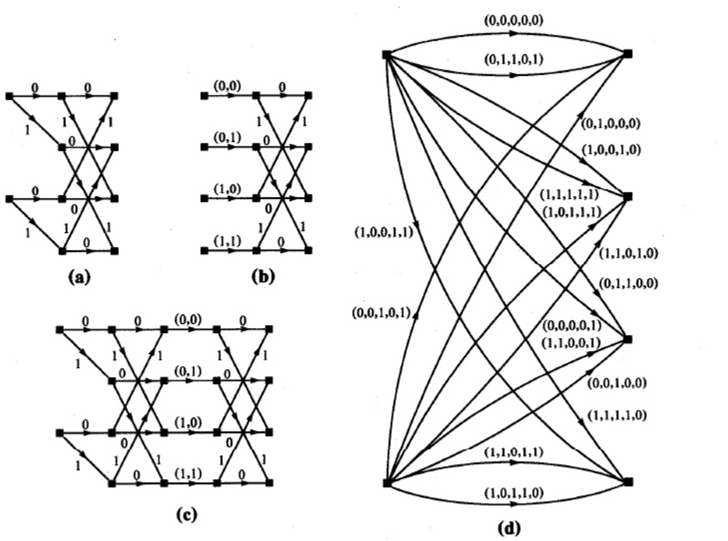


Fig. 1. Example of composition and amalgamation of trellises. (a) A trellis $T$. (b) A trellis $T'$. (c) The composition $T \circ T'$. (d) The amalgamation $T * T'$.

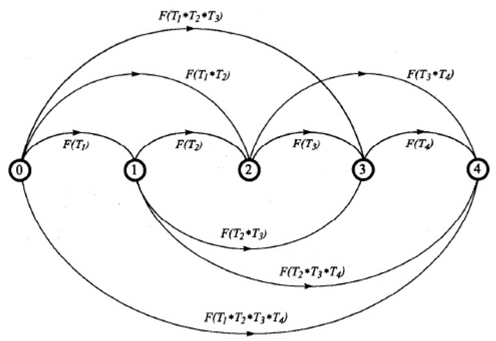


Fig. 2. The sectionalization digraph $\mathcal{G} = (V, \mathbb{N}, E)$ for a trellis of length 4.

# Objective function

- $F_1$(T)=the total number of branches in T
- $F_2$(T)=the minimum distance of the code generated by T
- $F_3$(T)=the number of operations required for Viterbi decoding of T

# The Algorithm

- Initialization:Set $T^{min}_{n}=T_{n}$ and $i=n-1$
- Minimization:Compute the value of
  $F(T_{i})+F(T^{min}_{i+1})$ (6)
  $F(T_{i}*T_{i+1}*....*T_{j})+F(T^{min}_{j+1})$
  for $j=i+1, i+2,....,n-1$ (7)
  $F(T_{i}*T_{i+1}*....*T_{n})$ (8)
  and find the minimum among the $n-i+1$ values in (6)-(8)

# The Algorithm

- Iteration:Set
  $T^{min}_{i}=T_{i} \circ T^{min}_{i+1}$
  or $T^{min}_{i}=(T_{i}*T_{i+1}*....*T_{jmin}) \circ T^{min}_{imin+1}$
  or $T^{min}_{i}=T_{i}*T_{i+1}*....*T_{n}$

  According as the minimum value in step 2 is achieved in (6),or by setting $j=j_{min}$ in (7),or in (8).If $i=1$ stop;then $T^{min}_{1}$ is desired trellis. Otherwise, set $i \leftarrow i-1$ and go back to minimization in step 2

TABLE V
DECODING COMPLEXITY FOR THE PRIMITIVE BINARY BCH CODES

| Code | Decoding complexity | | | |
|---|---|---|---|---|
| | Binary trellis | Best known | Reference | Optimal sectionalization |
| 1. BCH (8,4,4) | 53 | 23 | [17, 35] | 23 |
| 2. BCH (16,11,4) | 353 | 287 | [17, 35] | 167 |
| 3. BCH (16,7,6) | 537 | 295 | [35] | 231 |
| 4. BCH (16,5,8) | 193 | 110 | [17, 35] | 94 |
| 5. BCH (32,26,4) | 1,721 | 1,721 | [17, 35] | 1,255 |
| 6. BCH (32,21,6) | 26,873 | 22,980 | [16, 35] | 20,381 |
| 7. BCH (32,16,8) | 7,993 | 4,015 | [17, 35] | 2,383 |
| 8. BCH (32,11,12) | 16,121 | 4,375 | [35] | 3,994 |
| 9. BCH (32,6,16) | 729 | 384 | [17, 35] | 278 |
| 10. BCH (64,57,4) | 7,529 | 7,524 | [17, 35] | 6,507 |
| 11. BCH (64,51,6) | 418,553 | 418,600 | [16, 35] | 393,528 |
| 12. BCH (64,45,8) | 1,082,105 | 985,095 | [16, 35] | 891,819 |
| 13. BCH (64,39,10) | 38,436,857 | 34,359,000 | [35] | 30,982,731 |
| 14. BCH (64,36,12) | 18,972,665 | 13,572,000 | [35] | 12,829,263 |
| 15. BCH (64,30,14) | 41,844,729 | 18,180,000 | [35] | 16,598,063 |
| 16. BCH (64,24,16) | 1,679,353 | 471,480 | [16, 35] | 316,608 |
| 17. BCH (64,18,22) | 3,772,409 | 606,300 | [35] | 509,120 |
| 18. BCH (64,16,24) | 948,473 | 185,600 | [35] | 148,566 |
| 19. BCH (64,10,28) | 22,009 | 4,655 | [35] | 4,074 |
| 20. BCH (64,7,32) | 2,825 | 943 | [17, 35] | 806 |