Application of Shortest Path Algorithms in Telecommunication Channel Coding: A Comparative Study

Tamar Kamkhadze

MSc (Telecommunications), Invited Lecturer, Georgian American University, Tbilisi, Georgia.

Email: *tamar.kamkhadze@gau.edu.ge*

Abstract

Reliable communication over noisy channels requires efficient error-correcting mechanisms. Convolutional channel codes enable error correction by introducing redundancy, while decoding algorithms reconstruct the most probable transmitted sequence. The decoding process can be formulated as a shortest path problem in a weighted directed graph known as a trellis. The Viterbi algorithm performs maximum likelihood decoding by minimizing cumulative path metrics. This paper presents a mathematical formulation of convolutional decoding as a shortest path problem and compares the Viterbi algorithm with classical shortest path algorithms, including the Bellman–Ford algorithm and Dijkstra's algorithm. Structural similarities, complexity differences, and practical implications are analyzed.

Keywords: Channel Coding, Shortest Path Algorithm, Viterbi Algorithm, Bellman–Ford, Dijkstra, Trellis Graph

Introduction

Digital communication systems transmit data across noisy channels, where received signals are corrupted by additive noise and interference [3]. Let the transmitted sequence be:

$$\mathbf{u} = (u_1, u_2, \dots, u_k)$$

After encoding, the transmitted coded sequence becomes:

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

The received sequence is:

$$\mathbf{r} = \mathbf{x} + \mathbf{n} \qquad (1)$$

where **n** represents channel noise.

The decoding problem consists of finding the transmitted sequence $\hat{\mathbf{x}}$ that maximizes the conditional probability:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} P(\mathbf{r} \mid \mathbf{x}) \qquad (2)$$

This optimization problem can be reformulated as a shortest path problem in a weighted graph.

1. Convolutional Coding Model

A convolutional encoder with memory m can be modeled as a finite-state machine with:

$$S = 2^m$$

possible states [4].

Each input bit causes a state transition and generates output symbols according to generator polynomials.

The encoder can be represented as a state diagram, which when expanded over time forms a trellis graph.

2. Trellis as a Weighted Directed Graph

The trellis is a directed acyclic graph (DAG) defined as:

$$G = (V, E)$$

where:

V= set of states at each time step

E= transitions between states

Each edge $e \in E$ has an associated weight $w(e)$, defined by a distance metric between received and expected symbols.

For hard-decision decoding, the Hamming distance is used:

$$d_H(\mathbf{r}_t, \mathbf{x}_t) = \sum_{i=1}^{n} | r_i - x_i | \qquad (3)$$

For soft-decision decoding (AWGN channel), Euclidean distance is used:

$$d_E(\mathbf{r}_t, \mathbf{x}_t) = \sum_{i=1}^{n} (r_i - x_i)^2 \qquad (4)$$

The total path metric for path P is:

$$D(P) = \sum_{t=1}^{T} w_t \qquad (5)$$

Decoding becomes:

$$\hat{P} = \arg \min_{P} D(P) \qquad (6)$$

This is equivalent to the shortest path problem.

3. **Trellis as a Weighted Directed Graph**

The Viterbi algorithm applies dynamic programming [1].

Let:

$M_t(s)$be the minimum path metric to reach state s at time t.

Recursive update:

$$M_t(s) = \min_{s'} [M_{t-1}(s') + w(s', s)] \qquad (7)$$

Only the minimum (survivor) path is retained.

Complexity

If:

- S= number of states
- T= time steps

Then complexity is:

$$O(S \cdot T) \qquad (8)$$

Because the trellis is layered and acyclic, computation proceeds forward without revisiting states.

4. Classical Shortest Path Algorithms

4.1 Bellman–Ford Algorithm

The Bellman–Ford algorithm computes shortest paths from a source vertex $s$to all vertices in a graph $G = (V, E)$.

Initialization:

$$d(s) = 0, d(v) = \infty \text{ for } v \neq s$$

Relaxation step:

$$d(v) = \min\left(d(v), d(u) + w(u, v)\right) \qquad (9)$$

This process is repeated $\mid V \mid -1$ times.

Time complexity:

$$O(\mid V \mid\mid E \mid) \qquad (10)$$

Bellman–Ford can detect negative cycles.

4.2 Dijkstra's Algorithm

The Dijkstra's algorithm [2] operates under the constraint:

$$w(u,v) \geq 0$$

It selects the vertex with minimum temporary distance and relaxes adjacent edges. Time complexity (with priority queue):

$$O(|E| \log |V|) \qquad (11)$$

It cannot handle negative weights.

5. Comparative Mathematical Analysis

All three algorithms solve:

$$\min_{P \in \mathcal{P}(s,t)} \sum_{e \in P} w(e) \qquad (12)$$

where $\mathcal{P}(s,t)$ denotes all paths from s to t.

However, Table 1 shows:

Table 1-main distinctions between algorithms

Property	**Viterbi**	**Bellman–Ford**	**Dijkstra**
Graph type	Layered DAG	General graph	General graph
Weight sign	Non-negative	Any	Non-negative
Update rule	Eq. (7)	Eq. (9)	Greedy selection
Complexity	$O(ST)$	$O(VE)$	$O(E\log V)$

Key distinction:

- Viterbi exploits time-layered structure.
- Bellman–Ford iterates over entire edge set.
- Dijkstra applies global greedy optimization.

6. Engineering Implications

The trellis structure enables deterministic hardware implementation of the Viterbi algorithm with fixed memory and predictable computational complexity. This property makes it suitable for real-time communication systems, including mobile and satellite networks.

However, shortest path theory is not limited to channel decoding. In computer networking, routing protocols rely extensively on shortest path algorithms to determine optimal forwarding paths between nodes.

Distance-vector routing protocols such as Routing Information Protocol are based on the principles of the Bellman–Ford algorithm. Each router iteratively updates path costs based on information received from neighboring routers.

Similarly, link-state routing protocols such as Open Shortest Path First compute shortest paths using the Dijkstra's algorithm. Routers construct a global topology map and calculate optimal routes via greedy path selection.

This demonstrates that shortest path optimization forms a foundational principle across multiple layers of communication systems:

- TCP/IP Physical layer → Channel decoding (Viterbi)
- TCP/IP Network layer → Routing protocols (Bellman–Ford, Dijkstra)

Thus, shortest path theory serves as a unifying mathematical framework bridging digital communications and computer networking.

Conclusion

This paper presented a mathematical formulation of convolutional code decoding as a shortest path optimization problem. The Viterbi algorithm was analyzed as a specialized dynamic programming technique operating on layered trellis graphs. A comparative study with the Bellman–Ford and Dijkstra algorithms revealed both structural similarities and computational distinctions, despite their shared objective of path metric minimization.

Beyond channel decoding, shortest path algorithms play a central role in computer networking, forming the theoretical basis of routing protocols such as distance-vector and link-state approaches. This cross-layer applicability demonstrates that shortest path theory constitutes a unifying mathematical framework spanning the physical and network layers of communication systems.

The results confirm that graph-based optimization is fundamental to modern digital communication and networking architectures, reinforcing the importance of algorithmic theory in practical engineering design.

REFERENCES

1. A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
3. S. Haykin, *Communication Systems*, 5th ed. New York, NY, USA: Wiley, 2009.
4. J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2008.