

# Brute Force Approach for Guitar Chord Optimization

Andres Aguilar, Daniel Vega, Adrian Quiros

**Abstract**—This research explores a brute force approach to minimize left-hand movement when playing guitar chords across a musical piece. The algorithm exhaustively searches all possible chord fingering combinations to find the optimal sequence that minimizes physical effort during transitions. While guaranteeing optimal results, testing revealed the approach becomes computationally infeasible beyond 20 chords, with execution times growing exponentially.

**Index Terms**—Optimization, Brute Force, Guitar Chords.

## I. PROBLEM FORMULATION

The problem of minimizing left-hand movement in guitar chord sequences can be formulated mathematically as follows:

Let  $\sigma$  be a vector where  $\sigma_i$  represents a specific chord variant to be played on the guitar (e.g. C0, D2, etc.). Since each chord can typically be played in three different positions on the fretboard, we can define:

- Solution space  $S = \mathbb{N}^n$  where  $n$  is the number of chords
- Practical solution space  $S' = {}^nR$  where  $R \in \{0, 1, 2\}$
- Size of solution space  $|S'| = 3^n$

The brute force algorithm explores every possible combination in this solution space to find the sequence that minimizes the total displacement between consecutive chords. The time complexity is  $O(3^n)$ , making it impractical for sequences longer than 20 chords. For example:

- 10 chords: 59,049 combinations
- 15 chords: 14,348,907 combinations
- 20 chords: 3,486,784,401 combinations

## II. RESULTS

Brute-force algorithm: Execution Time Chart using 5, 10, 15, 20 and all chords

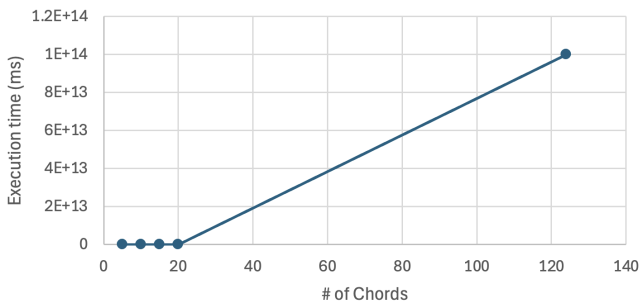


Fig. 1. Execution time of brute force algorithm showing exponential growth. Note that for inputs larger than 20 chords, the algorithm becomes computationally infeasible, failing to complete within reasonable time constraints.

Testing revealed that while the brute force approach guarantees optimal solutions, its execution time grows exponentially:

- 5 chords: 4.8ms
- 15 chords: 944.74ms
- 20 chords: 66.4 seconds

Extrapolating to 147 chords, the estimated runtime would be approximately  $3.6 \times 10^{70}$  years, demonstrating the algorithm's impracticality for real-world applications with longer chord sequences.

## III. CONCLUSION

While the brute force approach successfully guarantees optimal solutions for short chord sequences, its exponential time complexity makes it impractical for real-world applications. The astronomical computation times for sequences longer than 20 chords clearly demonstrate the need to explore alternative optimization approaches. Future work should focus on developing heuristic algorithms or dynamic programming solutions that can achieve near-optimal results with polynomial time complexity, making them suitable for practical musical applications.