The Stochastic Synthesis of Iannis Xenakis

By Voinea Ionut-Florin
Mircica Melisa-Maria

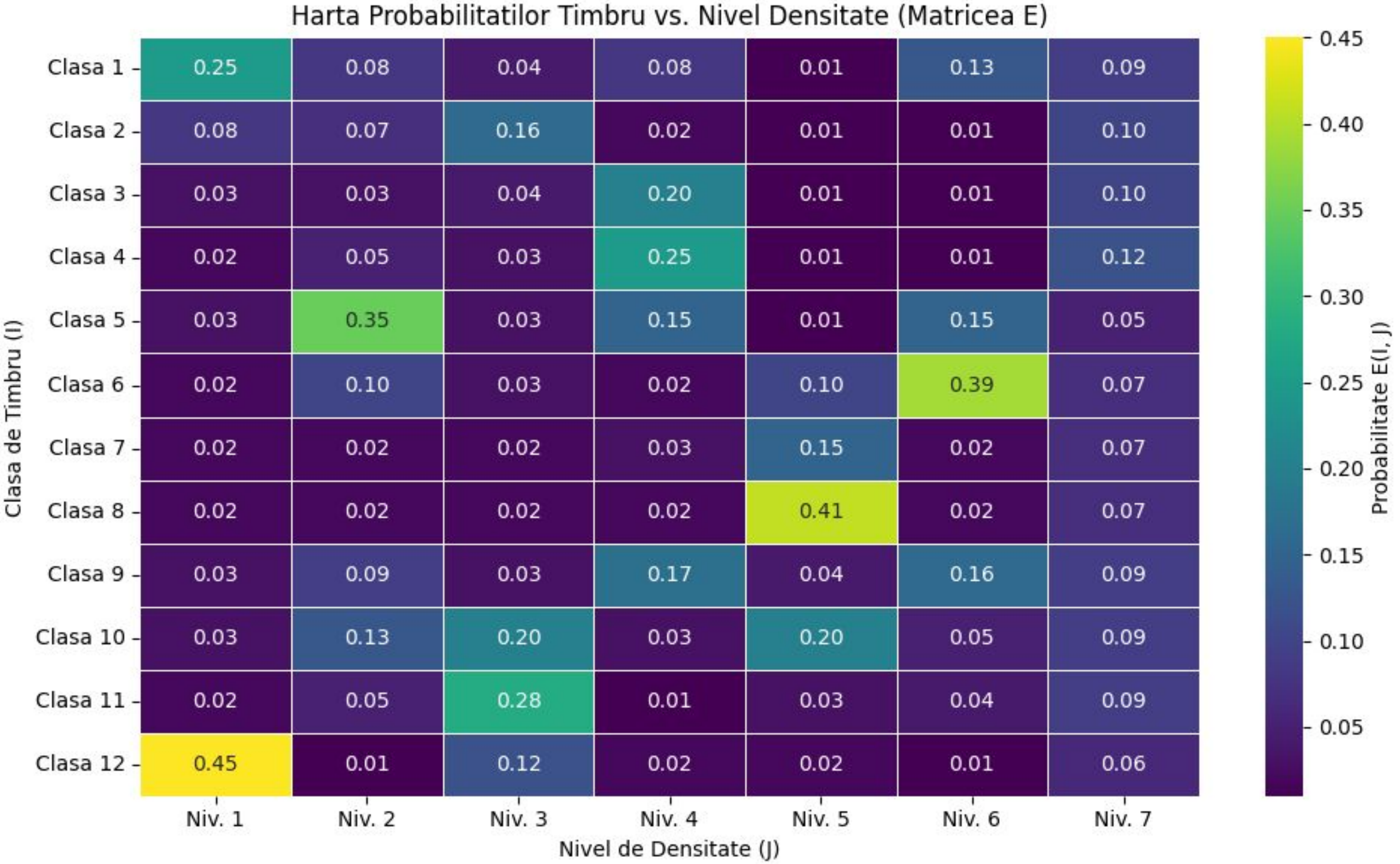
Introduction

- The primary challenge is the reproducibility and analysis of Iannis Xenakis's stochastic algorithms, written in FORTRAN IV for works like ST/10.
- Our objective is to rigorously reconstruct and validate the specific probabilistic logic and mathematical distributions detailed in Formalized Music.
- By translating Xenakis's stochastic synthesis methods into Python, this project aims not only to preserve historically significant algorithms but also to make them accessible for modern analysis, experimentation, and creative practice. This reconstruction bridges musicology, computer science, and mathematics, enabling a deeper understanding of algorithmic composition and the probabilistic thinking at the core of Xenakis's musical language.

Approach

- For our project, we will work with input data provided for the Atrées composition, which is a continuous stream of numbers. Since the data was originally formatted for FORTRAN, the numbers are not separated by spaces or commas. Therefore, we must use the character counting rules (Fortran FORMAT specifications like F3.3) in order to decode the data. This analysis allowed us to extract the HAMIN/HAMAX pitch limits and the minimum density V3 = 0.05 notes/sec directly from the input. Once the input is successfully decoded, we extract the probability matrix which is the primary source of stochastic distribution because it dictates the probability of selecting a given Timbre class (I) based on the current density level (J) within the movement.

- In the next step, we will implement the Stochastic Memory logic, which tracks both the Previous Density (UPR) and Previous Pitch (H). This memory allows the system to enforce smooth textural transitions and local melodic coherence, primarily achieved using the parabolic distribution function. Second, we will code the method for Continuous Orchestration by calculating Timbre Probabilities (Q) through linearly interpolating the Probability Matrix (E) against the continuous sequence density. Then, we will implement the process for Gaussian Parameter Generation which involves coding the efficient TETA table lookup for Inverse Transform Sampling to generate the precise Normal (Gaussian) distributions needed for parameters like note duration and glissando speed.


Harta Probabilitatilor Timbru vs. Nivel Densitate (Matricea E)
Clasa de Timbru (I)
Nivel de Densitate (J)
Probabilitate E(I, J)
Clasa 1 0.25 0.08 0.04 0.08 0.01 0.13 0.09
Clasa 2 0.08 0.07 0.16 0.02 0.01 0.01 0.10
Clasa 3 0.03 0.03 0.04 0.20 0.01 0.01 0.10
Clasa 4 0.02 0.05 0.03 0.25 0.01 0.01 0.12
Clasa 5 0.03 0.35 0.03 0.15 0.01 0.15 0.05
Clasa 6 0.02 0.10 0.03 0.02 0.10 0.39 0.07
Clasa 7 0.02 0.02 0.02 0.03 0.15 0.02 0.07
Clasa 8 0.02 0.02 0.02 0.02 0.41 0.02 0.07
Clasa 9 0.03 0.09 0.03 0.17 0.04 0.16 0.09
Clasa 10 0.03 0.13 0.20 0.03 0.20 0.05 0.09
Clasa 11 0.02 0.05 0.28 0.01 0.03 0.04 0.09
Clasa 12 0.45 0.01 0.12 0.02 0.02 0.01 0.06
Niv. 1 Niv. 2 Niv. 3 Niv. 4 Niv. 5 Niv. 6 Niv. 7
0.45 0.40 0.35 0.30 0.25 0.20 0.15 0.10 0.05

Other Articles Related to Our Research

- “Formalized Music THOUGHT AND MATHEMATICS IN COMPOSITION - Revised Edition” - Additional material compiled and edited by Sharon Kanach; Chapter 1. Free Stochastic Music
- “Iannis Xenakis’s Free Stochastic Music Program as an Aid to Analysis” - Ronald Squibbs
- “Iannis Xenakis’ Musique Stochastique System Design and Mathematical Background” - Seminar Computergestutzte Komposition, WS 2002/3 Universitat des Saarlandes Prof. Frobenius (Musikwissenschaft) and Prof. Wilhelm (Informatik) by Jeffrey Schone

Mathematical Models
(How the Music Achieves Coherence)

- The Exponential Distribution (Poisson's Law)
  - This model is used to determine the stochastic timing of note attacks and the duration of sequences, ensuring a rhythm that feels naturally irregular rather than strictly periodic.
- The Normal Distribution (Gaussian Model)
  - This model generates parameters like note duration and pitch displacement by ensuring that most values stay close to an average, providing controlled variation.
- The Parabolic Distribution (Short-Term Memory)
  - This function manages the system's musical memory. It ensures that the next note's pitch or density will almost certainly be very similar to the last one, making large, sudden jumps highly unlikely, which guarantees musical coherence.

Minimal Computational Requirements

- The ST program, being a product of 1960s mainframe computing, has minimal modern hardware requirements.
- GPU Usage: Zero. The program is purely arithmetic and logical, requiring no parallel processing capacity.
- CPU Power: Very Low. The computation involves only simple math functions and is instantaneous on any modern computer.
- RAM Memory: Minimal. Only needed to store small tables and matrices (like E and TETA).