

UNIVERSITÀ DI BOLOGNA

Magistrale in Ingegneria Informatica

Attività Progettuale di Fondamenti di Intelligenza Artificiale M  
**Evolutionary Music**

Professors:

**Mello Paola**

**Roli Andrea**

Student:

**Lisa Innocenti Uccini**

Academic year 2023/2024

# Abstract

Questo progetto si propone di creare un brano musicale del genere ambient utilizzando un algoritmo di intelligenza artificiale basato su algoritmo genetico. L'obiettivo è quello di generare una sequenza di accordi che sia coerente con l'atmosfera e lo stile tipico della musica ambientale.

I risultati ottenuti dimostrano che l'algoritmo genetico è in grado di generare brani musicali ambient coerenti con l'obiettivo proposto.

Un'importante prospettiva futura di questo lavoro è l'espansione della metodologia per includere non solo sequenze di accordi, questo consentirebbe la creazione di brani più articolati e complessi, ampliando così le possibilità di applicazione dell'intelligenza artificiale nella composizione musicale.

# Contents

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Obiettivo</b>	<b>5</b>
<b>3</b>	<b>Genetic Algorithm</b>	<b>6</b>
3.1	Rappresentazione individuo . . . . .	7
3.2	Funzionamento algoritmo . . . . .	8
3.2.1	Crossover . . . . .	9
3.2.2	Mutuazione . . . . .	9
3.2.3	Selezione . . . . .	9
3.3	Funzione costo . . . . .	9
<b>4</b>	<b>Tecnologie utilizzate</b>	<b>11</b>
<b>5</b>	<b>Esecuzione e rappresentazione dei risultati ottenuti</b>	<b>12</b>
5.1	Soluzione iniziale . . . . .	12
5.2	Soluzione iniziale . . . . .	13
5.3	Evoluzione costo . . . . .	13
<b>6</b>	<b>Conclusioni</b>	<b>15</b>

# Chapter 1

## Introduzione

La creazione di musica è da sempre stata un'attività umana che richiede creatività, sensibilità e conoscenza musicale. La musica è una forma d'arte universale che ha il potere di trasmettere emozioni, creare atmosfere e suscitare reazioni emotive nei suoi ascoltatori.

Negli ultimi anni, l'intelligenza artificiale (IA) è arrivata ad un punto tale che è in grado di generare in maniera automatica brani musicali attraverso algoritmi innovativi.

Uno dei settori più promettenti in questo ambito è proprio l'evoluzione della musica, che utilizza algoritmi evolutivi come gli algoritmi genetici per creare composizioni musicali. Con l'avvento delle tecnologie digitali e dell'intelligenza artificiale, sono nate nuove prospettive nella creazione di musica attraverso processi di evoluzione e adattamento automatico.

Il presente report è strutturato come segue: nel Capitolo2 verrà fornita una panoramica dettagliata dell'obiettivo del progetto. Nel Capitolo3 sarà descritto l'algoritmo utilizzato per la composizione musicale, in particolar modo la rappresentazione di un individuo (possibile soluzione), il funzionamento dell'algoritmo e la funzione fitness. Il capitolo 4 introduce le tecnologie utilizzate per la realizzazione del progetto. Il capitolo 5 presenterà i risultati ottenuti. Infine, nel Capitolo 6 saranno esposte le conclusioni del progetto e le prospettive future.

## Chapter 2

# Obiettivo

L'obiettivo principale di questo progetto è quello di utilizzare un algoritmo genetico per generare un brano musicale genere ambientale. Questo brano sarà composto da una sequenza di accordi coerenti e gradevoli, i quali dovranno rispettare diverse regole musicali.

La scelta di utilizzare un algoritmo genetico è motivata dalla sua capacità di generare soluzioni originali ed efficaci attraverso processi di evoluzione naturale. L'idea è quella di adattare questo approccio all'ambito della composizione musicale, consentendo alla musica di evolversi in modo autonomo e creativo. Inoltre, l'utilizzo di un algoritmo genetico offre la possibilità di esplorare una vasta gamma di combinazioni musicali e di individuare soluzioni ottimali rispetto agli obiettivi prefissati.

Attraverso questo progetto, non solo ci proponiamo di raggiungere l'obiettivo specifico della creazione di brani ambientali coerenti, ma intendiamo anche esplorare le potenzialità e le limitazioni di un approccio basato sull'intelligenza artificiale nella composizione musicale.

Ci aspettiamo che l'algoritmo genetico sia in grado di generare una sequenza di accordi coerente con l'atmosfera e lo stile della musica ambientale. Ci attendiamo che il sistema sia in grado di adattarsi e migliorarsi attraverso iterazioni successive, producendo composizioni musicali sempre più sofisticate e coinvolgenti.

## Chapter 3

# Genetic Algorithm

Gli algoritmi genetici sono metodi euristici di ricerca utilizzati nell'intelligenza artificiale e nell'informatica. Sono applicati per la ricerca di soluzioni ottimali a problemi complessi, come quello che abbiamo l'obiettivo di risolvere.

Sono eccellenti per la ricerca in seti di dati grandi e complessi. Sono considerati in grado di trovare soluzioni ragionevoli a problemi complessi poichè sono altamente in grado di risolvere problemi di ottimizzazione non vincolati e vincolati.

Gli algoritmi genetici sono ottimizzatori iterativi di tipo stocastico, basati sui concetti Darwiniani di selezione naturale e di evoluzione. Un algoritmo genetico utilizza tecniche ispirate alla biologia evolutiva come selezione, mutazione, ereditarietà e ricombinazione per risolvere un problema.

In un algoritmo genetico, un insieme di potenziali soluzioni viene fatto "evolvere" verso una soluzione globale ottimale. L'evoluzione (ottimizzazione) è il risultato di un insieme di operazioni di selezione e modifica dell'insieme che nel tempo tendono a scartare configurazioni meno adatte al problema, ovvero si allontanano dalla soluzione ottima.

Un algoritmo genetico differisce da un classico algoritmo di ottimizzazione su derivati principalmente in due modi:

- Un algoritmo genetico genera una popolazione di soluzioni ogni iterazione, mentre un algoritmo classico genera una singola soluzione a ogni iterazione;
- Un algoritmo genetico seleziona la popolazione successiva mediante un calcolo utilizzando generatori di numeri casuali, mentre un algoritmo classico seleziona il punto successivo mediante un calcolo deterministico.

Un'altra caratteristica di questa tipologia di algoritmi è che trasporta informazioni nel tempo quando vengono applicati degli operatori, nuove informazioni possono essere generate oppure perse. Inoltre gli algoritmi genetici lavorano con una codifica dei dati, non con i dati stessi.

Sono due gli elementi chiave di un algoritmo genetico:

- Si definisce un individuo una possibile soluzione al problema di ottimizzazione. Ciascun individuo è caratterizzato da un valore, corrispondente al risultato dell'applicazione della funzione fitness, tale elemento rappresenta il costo della soluzione in questione;

- Si definisce popolazione l'insieme delle possibili soluzioni, ovvero l'insieme di un numero finito di individui.

Gli individui vengono valutati con l'ausilio della funzione di valutazione fornita e descritta nel capitolo successivo. Agli individui viene quindi fornito un punteggio che evidenzia indirettamente l'idoneità alla situazione data.

### 3.1 Rappresentazione individuo

Ogni individuo è definito da una struttura ben precisa e studiata appositamente sulla base del problema che si vuole risolvere. La descrizione della struttura utilizzata per rappresentare l'individuo all'interno dell'algoritmo richiede una spiegazione, cio' permetterà anche di comprendere al meglio il funzionamento dell'algoritmo genetico utilizzato.

Ogni singola soluzione viene codificata rappresentando quest'ultima come una matrice di elementi, la cui dimensione corrisponde a:

- un numero di righe pari al numero di accordi che definiscono il brano;
- un numero di colonne pari al numero di note che compongono un singolo accordo.

Entrambi i valori in questione sono costanti e devono essere definiti prima dell'esecuzione.

Gli elementi della matrice sono numeri interi appartenenti ad un range precedentemente definito, il quale dipende dal numero di ottave musicali che si ha intenzione di prendere in considerazione per la definizione del brano.

Ogni numero corrisponde ad una nota musicale ben precisa, a seconda se la scala di riferimento che si intende utilizzare è maggiore o minore.

Per chiarezza, di seguito è riportato un esempio di conversione da numeri a note musicali, prendendo in considerazione un numero di ottave pari a due e come scala di riferimento la scala di C maggiore:

$$0 \rightarrow C; 1 \rightarrow D; 2 \rightarrow E; 3 \rightarrow F; 4 \rightarrow G; 5 \rightarrow A; 6 \rightarrow B; 7 \rightarrow C;$$

$$8 \rightarrow D; 9 \rightarrow E; 10 \rightarrow F; 11 \rightarrow G; 12 \rightarrow A; 13 \rightarrow B; 14 \rightarrow C$$

Le diverse soluzioni si differenziano a seconda di quali elementi sono contenuti all'interno della matrice e dall'ordine di quest'ultimi.

Per esporre al meglio questa caratteristica di seguito è riportato un esempio di individuo esplicativo:

numero di accordi che definiscono il brano = 5;  
note che compongono un singolo accordo = 4.

0	4	9	12
3	8	12	14
2	6	11	14
5	7	10	13
0	4	9	12

La scelta della rappresentazione di ogni individuo appena descritta è stata principalmente dettata dall'obiettivo di rendere l'algoritmo il più possibile indipendente dai seguenti parametri:

- numero di accordi che definiscono un brano;
- numero di note che compongono un accordo;
- numero di ottave musicali da considerare;
- scala di riferimento.

In questo modo l'algoritmo sarà il più possibile generico e scalabile.

## 3.2 Funzionamento algoritmo

Descritta la modalità con la quale vengono rappresentate le possibili soluzioni, possiamo passare alla descrizione dell'algoritmo utilizzato per raggiungere l'obiettivo di ottimalità.

Il funzionamento dell'algoritmo genetico si può rappresentare come nel seguente grafico. Come si nota dall'immagine, l'algoritmo è composto da diverse operazioni le quali vengono eseguite più volte, fino a che la condizione di terminazione non viene soddisfatta.

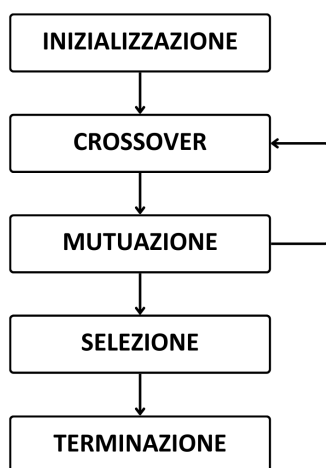


Figure 3.1: Funzionamento algoritmo

Adesso andiamo a descrivere più approfonditamente ognuna delle operazioni presenti nello schema.

La popolazione iniziale viene definita creando un numero finito di possibili soluzioni generate randomicamente.

Data una popolazione, l'evoluzione di quest'ultima avviene attraverso l'interazione del processo di ottimizzazione che permette, tramite una sequenza di operatori, di modificare gli individui della popolazione, andando così a generare una nuova generazione di individui.

L'algoritmo termina solo se una delle seguenti condizioni è soddisfatta:

- è stato raggiunto il numero di iterazioni eseguite massimo;



- il costo del miglior individuo della popolazione è pari a zero, è stata raggiunta la miglior soluzione possibile.

Gli operatori genetici implementati sono i seguenti: Crossover, Mutuazione e Selezione.

### 3.2.1 Crossover

Il Crossover è un operatore genetico utilizzato per generare nuovi individui a partire da quelli appartenenti alla precedente generazione. È analogo alla riproduzione e al crossover biologico, in quanto vengono selezionati due genitori e vengono prodotti più di un figlio utilizzando il materiale genetico dei genitori.

Per ogni individuo della popolazione viene generato un numero casuale, se questo è minore di una probabilità, viene preso casualmente un altro individuo della popolazione in questione, la coppia di individui viene data in input alla funzione crossover.

Per ogni paio di individui viene eseguita la seguente operazione di modifica: scambio una sezione di colonne nella coppia che stiamo considerando. Al termine di ogni singola esecuzione dell'operatore in questione otterremo due nuovi individui (figli).

### 3.2.2 Mutuazione

Questa parte del genetic algorithm consiste in una modifica mirata ad ottenere nuove soluzioni, l'obiettivo, come nel caso del cross over, è far "evolvere" gli individui in ingresso in modo tale da definire una nuova generazione di possibili soluzioni. Questo operatore apporta delle modifiche casuali, guidate da una percentuale definita precedentemente, al fine di diminuire il fitness degli elementi della popolazione.

Per ogni individuo in ingresso vengono eseguite le seguenti operazioni sulla matrice rappresentante la soluzione:

- scambio due righe prese casualmente;
- scambio due elementi presi casualmente;
- sostituzione di un elemento casuale della matrice con un nuovo elemento generato dal range dipendente dal numero di ottave.

Ognuna di queste operazioni viene eseguita con una determinata probabilità precedentemente stabilita.

### 3.2.3 Selezione

Funzione il cui obiettivo è selezionare i candidati più adatti a definire la nuova generazione. Ad ogni passaggio generazionale, dopo aver applicato gli altri operatori, un pool di soluzioni viene scelto in base ai valori di fitness di ciascun individuo utilizzando un meccanismo di selezione, in modo tale che gli individui con minor costo vengano selezionati.

## 3.3 Funzione costo

Gli algoritmi genetici sono procedure complesse, hanno l'obiettivo di trovare la soluzione migliore per un problema altrettanto complesso. Ciò viene fatto esplorando lo spazio delle soluzioni utilizzando gli operatori descritti dettagliatamente del capitolo precedente.

Per capire se una soluzione è migliore o meno viene definita una caratteristica associata ad ogni possibile soluzione appartenente alla popolazione: *il costo*.

Quest'ultima viene calcolata dando un individuo contenuto nella popolazione in ingresso alla funzione fitness. La procedura in questione è in grado di associare un valore ad ogni soluzione, il quale misura il livello di adeguatezza degli individui stessi.

Il calcolo del costo dipende da diversi fattori, i quali non sono standard per ogni applicazione, ma sono studiati appositamente per la risoluzione del problema che ci siamo posti in questo elaborato.

La funzione costo definisce un valore associato ad una determinata soluzione, in base alla costruzione. Quest'ultimo ha l'obiettivo di esplicitare quanto è buona una soluzione rispetto ad un'altra. Ciò ci permette di memorizzare la migliore soluzione trovata all'interno delle popolazioni prodotte dalle varie iterazioni dell'algoritmo genetico. La migliore soluzione, ovviamente, è quella definita dal costo minore.

I fattori presi in considerazione per il calcolo del costo sono:

- numero di note uguali in uno stesso accordo;
- numero di accordi dissonanti;
- numero di ripetizioni consecutive di un accordo;
- numero di note uguali nel brano;
- numero di accordi uguali nel brano;

Dalla funzione notiamo la presenza di 5 componenti che determinano il risultato. Ognuno di questi ha un'influenza diversa sul costo associato ad una determinata soluzione, il peso di ogni fattore viene definito dalla presenza di moltiplicatori e divisori.

I fattori utilizzati per determinare il costo sono frutto di un'attenta analisi della teoria musicale, delle aspettative di risultato e dei risultati effettivamente ottenuti. Sono stati scelti in modo tale da garantire che il brano risultante sia caratterizzato da una sequenza di accordi il più coerente, gradevole e musicalmente corretta possibile.

Affinché la sequenza di accordi risulti gradevole e rispetti le regole della musica, è fondamentale che ciascun accordo sia definito in modo tale che tutte le note che lo compongono siano diverse e che non vi siano dissonanze. Il mancato rispetto di tali condizioni comporta un aumento del costo della soluzione.

Tuttavia, non è sufficiente valutare il costo considerando singolarmente ciascun accordo; è necessario esaminare l'intera sequenza di accordi e le relazioni tra di essi. In particolare, un numero elevato di ripetizioni di un accordo o di una nota nel brano determina un aumento del costo della soluzione.

## Chapter 4

# Tecnologie utilizzate

Per l'implementazione del progetto sono state impiegate diverse tecnologie che hanno contribuito alla realizzazione dell'obiettivo finale. In particolare, sono stati utilizzati il linguaggio di programmazione Python e il software MuseScore.

Python è stato il linguaggio di programmazione principale utilizzato per lo sviluppo dell'algoritmo genetico responsabile della composizione del brano musicale. La sua flessibilità, facilità d'uso e ricchezza di librerie lo rendono una scelta ideale per implementare algoritmi di intelligenza artificiale e per la manipolazione di dati musicali.

L'algoritmo genetico scritto in Python produce come output una sequenza musicale convertita nel linguaggio di markup ABS (Abstract Score) utilizzato da MuseScore per la rappresentazione e la visualizzazione delle partiture musicali.

MuseScore è stato utilizzato come strumento per la visualizzazione e l'ascolto del brano musicale generato dall'algoritmo genetico. Grazie alla sua interfaccia intuitiva e alle potenti funzionalità di editing musicale, MuseScore ha consentito di trasformare la sequenza musicale codificata in formato ABS in una partitura leggibile e comprensibile.

Utilizzando MuseScore, è stato possibile esplorare il brano generato in dettaglio, visualizzando le singole note, le dinamiche, gli accordi e altri elementi musicali. Inoltre, il software in questione ha fornito la possibilità di ascoltare il brano attraverso la riproduzione audio integrata, consentendo una valutazione accurata delle caratteristiche musicali del risultato ottenuto.

## Chapter 5

# Esecuzione e rappresentazione dei risultati ottenuti

Per valutare l'efficacia dell'algoritmo per la generazione di brani musicali ambient sono state effettuate diverse prove.

Dopo aver eseguito l'algoritmo genetico con i parametri definiti, i risultati ottenuti sono stati rappresentati in modo completo e significativo.

Inizialmente, si ottiene un risultato corrispondente a una sequenza musicale codificata nel linguaggio ABS (Abstract Score), risultato dell'esecuzione dell'algoritmo.

Successivamente, utilizzando il software MuseScore, si trasforma la sequenza musicale in linguaggio ABS in uno spartito musicale visivamente accessibile e ascoltabile.

L'ascolto del brano fornisce, inoltre, l'opportunità di valutare la composizione anche da un punto di vista emano, emotivo ed estetico.

Il risultato riportato di seguito è stato generato settando i parametri nel seguente modo:

- numero accordi nel brano: 10;
- numero note per accordo: 3;
- numero di individui in una popolazione: 100;
- numero iterazioni: 100.

Vengono riportati i dati e le rappresentazioni sia della soluzione iniziale e che della soluzione finale, in modo tale da poter fare un confronto ed esaminare al meglio i risultati ottenuti.

### 5.1 Soluzione iniziale

Costo soluzione: 103.5

Linguaggio ABC:

M:4/4

L:1/1

| [FGA] | [Aeg] | [Eeg] | [Gdd] | [GBd] | [Dca] | [EEe] | [cdb] | [Gfg] | [Dda]

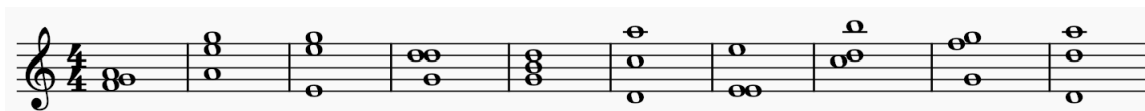


Figure 5.1: Spartito soluzione iniziale

## 5.2 Soluzione iniziale

Costo soluzione: 0

Linguaggio ABC:

M:4/4

L:1/1

| [GBb] | [Egb] | [ceg] | [DFA] | [CcG] | [GBg] | [GBd] | [Fda] | [GBg] | [EcG]

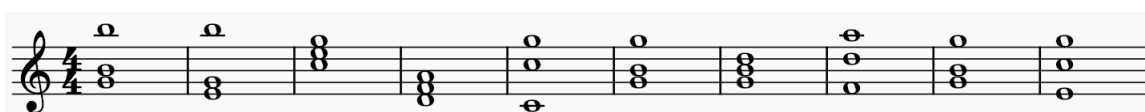


Figure 5.2: Spartito soluzione iniziale

## 5.3 Evoluzione costo

Nel grafico seguente è mostrato l'andamento della fitness della popolazione nel corso delle generazioni.

Si può osservare una tendenza generale verso il miglioramento della fitness, indicando una progressiva convergenza verso soluzioni di qualità ottimale. In particolar modo si può notare anche la velocità di convergenza, in quanto dopo 25 iterazioni il costo è già pari a 0.

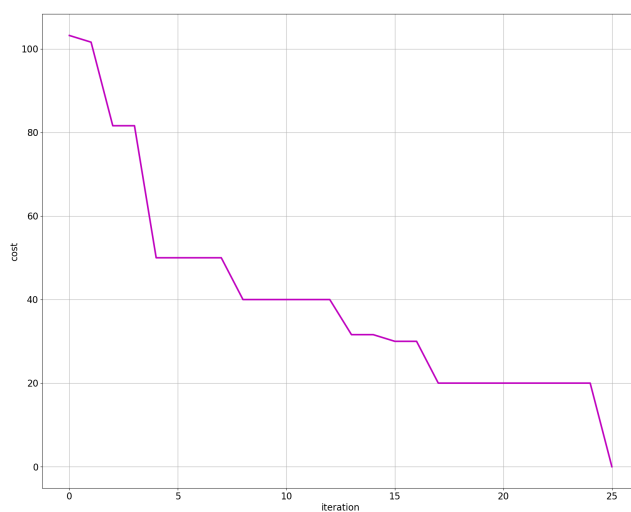


Figure 5.3: Evoluzione costo iterazioni

I risultati ottenuti mettono in luce l'efficacia degli algoritmi genetici nella creazione di brani musicali ambient composti da una sequenza di accordi coerenti e suggestivi, confermando il notevole potenziale dell'intelligenza artificiale nel campo della composizione musicale.

Inizialmente, la soluzione iniziale proposta presenta un costo elevato e un'impressione sonora poco gradevole, caratterizzata da ripetizioni e dissonanze. Tuttavia, eseguendo l'algoritmo ad ogni iterazione il costo diminuisce fino a convergere verso lo zero.

Il risultato finale, come evidenziato dalle immagini riportate, è una sequenza di accordi logicamente strutturata e armonicamente coerente, rispettando le regole stabilite per la valutazione del costo, come ad esempio l'assenza di dissonanze o di ripetizioni eccessive di accordi.

## Chapter 6

# Conclusioni

In conclusione, i risultati ottenuti da questo progetto rispecchiano le aspettative e gli obiettivi prefissati. L'algoritmo genetico implementato si è dimostrato efficace nella generazione di brani musicali nel genere ambientale, producendo composizioni coerenti e suggestive.

Le sequenze di accordi prodotte mostrano una struttura armonica ben definita e una progressione melodica coerente con le regole musicali.

L'ascolto dei brani attraverso il software MuseScore ha confermato la qualità delle composizioni generate dall'algoritmo.

Tuttavia, va notato che il compito assegnato all'algoritmo, ovvero la creazione di una sequenza di accordi per comporre un brano, non presentava particolari difficoltà in termini di complessità delle regole musicali da rispettare.

Nonostante ciò, il successo del progetto apre la porta a possibili espansioni e sviluppi futuri. Una direzione interessante potrebbe essere l'implementazione di algoritmi più complessi e sofisticati per la composizione musicale, che tengano conto non solo della sequenza di accordi, ma anche di altri elementi musicali come melodie, ritmi e texture sonore. Inoltre, si potrebbe esplorare l'utilizzo di approcci ibridi che combinino l'intelligenza artificiale con l'esperienza umana nella composizione musicale, per ottenere risultati ancora più creativi e originali.

Questo progetto rappresenta un punto di partenza per esplorare il vasto territorio dell'evolutionary music e dell'intelligenza artificiale nella composizione musicale.