

UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA ELETTRONICA E INFORMATICA

MODELLIZZAZIONE MUSICALE TRAMITE CATENE DI MARKOV

Relazione discussa in sede di esame finale
dal candidato *Federico Simonetta*

Docente tutore: Prof. *G. De Nicolao*

A.A. 2013/14

Indice

1	Esempi di pensiero algoritmico nella musica	1
1.1	L’Era precedente ai computer	1
1.1.1	Primi studi di acustica	1
1.1.2	Prime formalizzazioni matematiche del ritmo	2
1.1.3	Guido d’Arezzo, Kircher e il <i>Musikalisches Würfelspiel</i>	3
1.1.4	L’algoritmo nelle manifestazioni artistiche	5
1.2	L’Era dei computer	8
1.2.1	Le prime macchine	8
1.2.2	Determinismo e aleatorietà	10
1.2.3	Iannis Xenakis	11
1.2.4	György Ligeti	11
1.2.5	Musica algoritmica e popular music	12
2	Musica e catene di Markov: studi precedenti	13
2.1	Catene di Markov	13
2.2	Generatori tramite catene di Markov	14
2.3	Analisi musicale tramite catene di Markov	15
3	Software generatore di melodie	17
3.1	Algoritmo	17
3.1.1	Algoritmo a una voce	17
3.1.2	Algoritmo a più voci	20
3.2	Architettura	21
3.2.1	Strumenti utilizzati. Java e JFugue	21
3.2.2	Design patterns e diagrammi UML	22
3.3	Migliorie	24
3.3.1	Matrice principale e matrice secondaria	24
3.3.2	Istogrammi	25
3.4	Risultati e commenti	26
3.5	Limiti e problemi	34
3.6	Possibili linee guida per il futuro	34

Introduzione

Questo lavoro ha avuto come obiettivo principale l'esplorazione del mondo della composizione algoritmica, in particolar modo di quel settore che opera con le catene di Markov, e lo sviluppo di un software che produca musica tramite catene di Markov allenate su brani preesistenti. Il primo capitolo delinea brevemente le linee generali di una possibile storia del pensiero logico-razionale in musica. Non vuole essere una trattazione esaustiva, ma, riguardando argomenti già ampiamente conosciuti dagli studiosi, si pone come una proposta di percorso tematico.

Il secondo capitolo tratta, dopo una breve introduzione teorica, alcuni lavori scientifici cui si è venuti a conoscenza. Si è individuata nella letteratura qui trattata due grandi linee di lavoro: una che riguarda l'analisi musicale assistita dal calcolatore e una che tratta nuovi metodi per le pratiche musicali, siano esse improvvisate o meno. Molti dei lavori citati sono di recente data, a conferma che si tratta di un settore in via di evoluzione.

L'ultimo capitolo, infine, tratta del laboratorio annesso alla presente trattazione, durante il quale si è sviluppato un piccolo software in Java capace di elaborare brani tramite catene di Markov; alcuni risultati potrebbero essere utilizzati anche al fine dell'analisi musicale.

Capitolo 1

Esempi di pensiero algoritmico nella musica

L'arte musicale e il pensiero razionale sono connessi da complessi legami fin dalle più antiche fonti musicologiche a noi disponibili. In questo capitolo verranno presentati alcuni esempi del rapporto tra musica, matematica e algoritmi compositivi, con l'obiettivo di rendere un'immagine generale del contesto culturale del presente lavoro. In questo senso si farà perciò riferimento a manifestazioni artistiche che nella Storia della Musica hanno legato alla composizione musicale il fascino per il numero, per la matematica o per rigide regole logicamente predeterminate. Come è ovvio, l'accezione del termine "algoritmo" può essere molto varia: qualunque ragionamento condotto durante la fase compositiva può essere inteso come "algoritmo", razionale o meno che sia. In questa sede si considereranno algoritmi compositivi tutti quei procedimenti che hanno a fondamento un'importante quantità di ragionamento logico e razionale.

1.1 L'Era precedente ai computer

1.1.1 Primi studi di acustica

I fondamenti del legame tra musica e matematica sono sicuramente da ricercare nelle prime forme di musica a noi conosciute. Secondo alcuni studiosi già gli Egizi avrebbero potuto conoscere alcuni principi di acustica. [30] Sicuramente già i Cinesi, civiltà che sviluppò una cultura musicale almeno a partire dal 2000 a.C., possedevano conoscenze approfondite di acustica, tanto che il mito racconta della generazione dei dodici suoni della scala musicale (*liuh*) proprio in virtù dei rapporti fra le altezze. Leu Pu-wei scrive nel *Leu*

Shyh (239 a.C.) che secondo il mito i *liuh* furono trovati aggiungendo tra di loro pezzi di bambù di uguale spessore e lunghi “tre pollici e nove decimi”, la cui lunghezza definiva il suono fondamentale (*hoang-cong*). [30]

Anche i Babilonesi, che si stanziarono in Mesopotamia a partire dal 2000 a.C. fino al 1000 a.C. e poi da VII sec al 539 a.C., avevano un serio interesse per la musica e l’acustica. Secondo Plutarco (46 – 127 d.C.), i Babilonesi collegavano gli intervalli musicali alle stagioni dell’anno. Secondo altri autori greci, lo stesso Pitagora dovette la propria formazione matematica e musicale alle culture egizia e babilonese. [30]

Ruolo centrale nella teoria musicale occidentale fu svolto dalla scuola di Pitagora di Samo (VI-VII sec. a.C) fondata a Crotona nel 530 a.C. Secondo i pitagorici la matematica era non solo in grado di spiegare tutti i fenomeni del Mondo, ma era intrinsecamente associata agli elementi stessi del Mondo tramite degli stretti legami tra numeri e concetti, esplicitati nel *tetraktys*, un modello geometrico tramite il quale descrivevano l’armonia e l’ordine dell’Universo.

Alla scuola pitagorica si deve la scoperta dei rapporti tra le frequenze fondamentali delle note musicali occidentali. [3] Rifacendosi al *tetraktys*, costituito da quattro numeri base (1, 2, 3 e 4), essi trovarono che il rapporto 4:1 corrisponde a due ottave, 3:1 ad un’ottava più una quinta cioè una dodicesima, il rapporto 2:1 all’ottava, il 3:2 alla quinta e il 4:3 alla quarta. Di conseguenza, un intervallo di una seconda maggiore, deve necessariamente essere il rapporto tra 3:2 e 4:3, cioè $\frac{3}{2} \times \frac{3}{4} = \frac{9}{8}$. Partendo da queste osservazioni essi idearono anche la scala pitagorica, ancora oggi utilizzata. [3, 30]

Il rapporto tra musica e matematica per i pitagorici risulta quindi essere una diretta conseguenza della visione del Mondo e della “aritmo geometria” che permeava la loro vita in tutti i suoi aspetti.

1.1.2 Prime formalizzazioni matematiche del ritmo

Nella civiltà greca, benché la musica fosse in gran parte di origine improvvisativa, si ritrova uno stretto rapporto con la matematica anche nei ritmi utilizzati, derivanti dalle sillabe lunghe e brevi del testo.

Anche durante l’Alto Medioevo, nella scuola parigina di Notre Dame, dove si sperimentano varie possibilità contrappuntistiche, si giunge ad una concezione ritmica simile a quella greca, ma con fondamenti indipendenti dal testo. Le ritmiche utilizzate erano guidate dai 6 differenti modi in figura — da qui la dicitura *ritmo modale* —, ciascuno corrispondente ad una diversa ritmica. I valori di durata utilizzati erano esclusivamente la *longa* e

la *brevis*. [30]

Questi i modi utilizzati (B = brevis, L = longa): [30]



A questo si aggiunga il simbolismo medievale, per il quale il 3 aveva un significato intrinsecamente divino e quindi privilegiato; la stragrande maggioranza della musica, specie di quella religiosa, era scritta in tempi ternari, non a caso detti *perfetti* in contrapposizione a quelli binari detti *imperfetti*.

Si giunse così ad individuare dei rapporti matematici che descrivevano sia l'aspetto melodico sia l'aspetto ritmico, fenomeno unico nella Storia delle forme d'arte. Si giungerà poi ad un'importante svolta con Franccone da Colonia e il suo trattato "*Ars cantus mensurabilis*" (1250) che getterà le basi alla cosiddetta *notazione mensurale*, in cui vengono introdotte nuove durate, spariscono i modi ritmici e si instaurano i rapporti conosciuti oggi tra le durate delle note. [30]

1.1.3 Guido d'Arezzo, Kircher e il *Musikalisches Würfelspiel*

Un primo esempio di vera e propria musica algoritmica si ha con Guido d'Arezzo (991 – 1033), noto monaco di Ferrara, a cui si deve tra l'altro, oltre ai nomi delle nostre note musicali, il sistema della cosiddetta *ars solfandi* un complesso metodo di insegnamento del canto che fu in voga in tutta Europa per i secoli successivi, pur con varie modifiche, e a cui Guido dovette una grandissima notorietà che lo rivestì di numerose leggende popolari. [26, 30]

Guido d'Arezzo, attorno al 1026, scrisse il *Micrologus*, uno dei trattati musicali più letti del Medio Evo insieme al *De institutione musica* di Severino Boezio (475 - 525), nel quale espone un sistema per associare ai testi poetici le note musicali; in particolare, al cap. XVII, dopo aver espresso una serie di considerazioni stilistiche sul rapporto tra testo e musica, finisce con associare ad ogni vocale una nota. [26, 30] In questo modo, Guido d'Arezzo costruì il primo sistema automatico per produrre musica ad oggi conosciuto della Storia.

Un altro caso è quello di Padre Athanasius Kircher, il quale, oltre a scrivere importanti trattati teorici, sviluppò un metodo per comporre tramite schemi precostituiti, studiato per permettere di comporre senza conoscere la musica; chiamò il sistema "*Arca Musarithmica*" e lo presentò nell'otta-

Strophica IV. <i>Strophica IV.</i>	Note temporis <i>Strophica IV.</i>
Toni.V.VI.VII.VIII	o o o o o o o o
7771234555	o o o o o o o o
55568878	o o o o o o o o
33338323	o o o o o o o o
33366551	o o o o o o o o
112173332223	o o o o o o o o
56758888878	o o o o o o o o
34435556555	o o o o o o o o
14231184551	o o o o o o o o
72878687678	o o o o o o o o
555554555	o o o o o o o o
3232383223	o o o o o o o o
3715341551	o o o o o o o o
56535432878	o o o o o o o o
8878765555	o o o o o o o o
3428887323	o o o o o o o o
845634151	o o o o o o o o
3278767878	o o o o o o o o
8654545555	o o o o o o o o
5428383223	o o o o o o o o
1456341551	o o o o o o o o
71223171	o o o o o o o o
56655455	o o o o o o o o
2667882323	o o o o o o o o
55443651	o o o o o o o o

Figura 1.1: Colonna per comporre la quarta strofa, Athanasius Kircher - *Organum Mathematicum*. Licensed under Public domain via Wikimedia Commons

vo libro della *“Musurgia Universalis”* (1650), affermando di averlo ottenuto sfruttando in modo peculiare le proprietà del calcolo combinatorio. [5]

In effetti, l’*Arca Musarithmica* di Kircher costituisce un vero esempio di musica algoritmica, permettendo di comporre seguendo schemi e regole predeterminate.

Simile all’*Arca Musarithmica* è il sistema ideato da Mozart, che sebbene non sia solitamente ricordato per l’aspetto speculativo della sua musica creò un noto sistema per comporre musica in automatico; il principio di base è identico a quello di Kircher, realizzato circa un secolo prima, e anche questo ha l’obiettivo di «comporre [...] senza avere la minima nozione di musica». [21, 28]

Il sistema viene pubblicato da Simrock con il nome di *Musikalisches*

Würfelspiel (“Dado musicale”) perchè le istruzioni prevedono l’utilizzo di due dadi laddove Kircher pensava ad una mente umana; sicuramente questo fu uno dei primi esempi di aleatorietà in musica. Il sistema è stato attribuito a Mozart, ma non vi è certezza sulla paternità dell’opera. [21]

1.1.4 L’algoritmo nelle manifestazioni artistiche

In questo paragrafo si presentano alcuni esempi di compositori e di fenomeni il cui processo compositivo è stato formalizzato dagli studiosi in algoritmi o in regole precise.

Ars Nova

Nel tardo Medioevo la produzione musicale più interessante si sposta in Francia, dove durante un lungo processo di secolarizzazione di tutta la cultura, la musica comincia ad uscire dal fatto religioso e dalla Chiesa. In questo periodo gli autori diventano uomini laici, spesso nobili, che girano di corte in corte cantando le loro poesie accompagnandosi con strumenti musicali.

All’interno di questo scenario, la musica di alta aspirazione comincia ad avere un carattere sempre più intellettuale. Johannes Boenn nel 1355 scriveva che i procedimenti compositivi dell’epoca erano “più per la vista che per l’udito”; e d’altronde i complessi risultati cui giungono i compositori applicando i principi della *talea* e del *color* confermano questa opinione. [30]

Figura di riferimento in quest’epoca è senz’altro Guillaume de Machaut (ca. 1300 – 1377) e anche nei suoi mottetti, forma nobile per eccellenza, si ritrovano complesse strutture architettoniche che utilizzano la ripetizione ritmica e/o melodica di brevi motivi di origine gregoriana. Lo schema ritmico, *talea*, e la figura melodica, *color*, si sovrappongono in maniera differente tra le varie voci creando effetti non facilmente udibili, specie nei mottetti a molte voci. [30] In, questi procedimenti ricordano molto le tecniche dodecafoniche di cui si parlerà più avanti. [30] Quello che segue è un breve esempio di utilizzo di *talea* e *color* (pratica detta *isoritmia*); le ‘T’ indicano l’inizio di una *talea*, mentre le ‘C’ indicano l’inizio di un *color*; si noti che con un differente tema — in questo caso la melodia detta dal soprano — l’incastro darebbe luogo a dissonanze non ammesse dal gusto del periodo.



La scuola Fiamminga

Durante il Rinascimento, l'Italia sarà anche nella musica come nelle altre arti al centro dell'attenzione della cultura Occidentale; ma i grandi musicisti di quest'epoca, dall'inizio del Quattrocento alla fine del Cinquecento, provengono da un'altra regione Europea che nello stesso periodo storico diede vita ad interessanti manifestazioni artistiche: la scuola fiamminga.

La scuola fiamminga — si adotta qui la terminologia diffusa di “scuola” indipendentemente dalla sua appropriatezza —, la quale deriva in maniera molto stretta dall'Ars Nova francese, e in generale la musica vocale del Cinquecento, assume concezioni per certi aspetti sempre più intellettualistiche; vengono utilizzati procedimenti compositivi che si basano su qualunque tipo di regola, di volta in volta decisa dal compositore. Principio base delle opere contrappuntistiche è l'imitazione, che giunge a divenire una vero e proprio algoritmo compositivo nei canoni (mensurali ed enigmatici). In particolare, i canoni enigmatici prevedono la stesura di una linea melodica e la deduzione delle altre voci tramite una regola comprensibile solo dopo aver risolto un indovinello; alcuni canoni enigmatici sono ancora ad oggi rimasti irrisolti. Altri principi molto usati sono il contrappunto doppio, l'inversione e il moto retrogrado, abbinato eventualmente al canone, e tutte le loro combinazioni. [9]

Gli esempi di possibili algoritmi compositivi ad opera di musicisti fiamminghi sono moltissimi e si rimanda pertanto ai testi specializzati per l'approfondimento; si ricorderà qui il mottetto “*Nuper rosarum flores*”, che secondo alcune ipotesi presenta le proporzioni della cattedrale di S. Maria del Fiore a Firenze, per la cui inaugurazione fu composto da Guillaume Dufay, primo musicista della scuola e il più importante del '400, e la “*Missa L'homme armé super voces musicales*”, basata su una melodia preesistente trasposta di brano in brano di un grado più in alto, scritta Josquin Desprez, il più celebre compositore della scuola, non ché uno dei più fini e alti com-

positori di tutti i tempi. [19, 27]

J.S. Bach

Un caso estremamente importante per tutta la musica a seguire, e anche di fondamentale interesse per la musica algoritmica moderna e contemporanea, è certamente il caso di J.S. Bach (1685-1750). Considerato solitamente, insieme a G.F. Handel (1685-1759), l'apice raggiunto dall'arte musicale in periodo Barocco, egli produsse opere di inestimabile valore che, anche se non note al pubblico europeo dell'epoca, saranno da modello per tutti i compositori a seguire, in particolar modo per i musicisti romantici e, ancora di più, per i musicisti seriali e, appunto, "algoritmici". [30]

Bach, caso quasi unico nel suo tempo, seppe unire lo stile compositivo dell'epoca, recependo elementi da vari stili nazionali, all'interesse per l'elemento logico e razionale, per il ragionamento e il fascino della regola, nonché per la sua rottura. Forma per eccellenza ricordata parlando di Bach è senz'altro la fuga, di cui Bach fisserà la struttura in maniera rigorosa e precisa; lo stile contrappuntistico e imitativo della fuga, obbliga il compositore a schemi compositivi simili a quelli fiamminghi, con in più obblighi formali definiti e più rigidi. [30]

Per evidenziare il carattere più speculativo della musica di Bach, che è sempre in qualche modo presente, si fa di solito riferimento a due delle opere che da molti studiosi sono considerate le vette più alte della produzione musicale di sempre: l'*Arte della Fuga* (ca 1740-1750) e l'*Offerta musicale* (1747). [30]

La prima è un'opera rimasta incompiuta che contiene 19 brani non strumentati — da qui l'interpretazione secondo la quale l'opera non aveva destinazione esecutiva — tutte sullo stesso soggetto, dimostrando quindi una grandissima abilità contrappuntistica nel saper rielaborare la stessa materia in modi sempre nuovi, pur sottostando a forti vincoli, dal canone perpetuo alle fughe a specchio, rovesciate ed altri. La seconda invece è un'opera dedicata dall'autore al re Federico II di Prussia su tema ideato da quest'ultimo. L'opera presenta una serie di brani di complessa forma e acuta ideazione: canoni inversi, perpetui, per moto contrario, enigmatici, complessi ricercari a 6 voci, volti a porre in rilievo l'ingegnosità dell'opera, che in quasi tutte le sue parti non è strumentata. [30]

Ultima opera di Bach che viene qui citata sono i 14 canoni sulle note del basso delle *Variazioni di Goldberg*. Esse, di recente scoperta, presentano 14 canoni enigmatici, proprio sul modello di quelli fiamminghi; in questo caso,

come nelle *Variazioni di Goldberg* (1741), la struttura matematica regola anche la disposizione dei singoli brani all'interno dell'opera. [1, 30]

Bach rappresenta sicuramente l'esempio più lampante di come l'artificio logico-razionale, inteso anche come algoritmo compositivo, fosse già stato alla base della produzione artistica ben prima della comparsa dei calcolatori elettronici, sebbene controllato dalla mente umana del compositore e all'occorrenza corretto in maniera anche, se necessario, irrazionale.

Dodecafonìa e musica seriale

Schönberg, nel cammino verso la dodecafonìa, recupererà gli stessi procedimenti di elaborazione tematica dell'epoca rinascimentale e che si dedicherà a esercizi intellettuali simili a quelli dell'*Arte della Fuga*. Ma, a differenza dei fiamminghi, nella dodecafonìa vi è un limite importante, ovvero quello della serie dodecafonica: una successione di dodici note in cui non compare mai due volte la stessa nota. Oltre a portare ad inediti effetti armonici, che sconvolgono il linguaggio armonico classico-tonale già portato agli estremi da Wagner, questo procedimento ha come conseguenza l'imposizione di un limite imponente alla libertà del compositore che si trova a dover scegliere tra 479 001 600 differenti serie come materiale tematico della propria composizione. [7, 28]

Saranno poi gli allievi di Schönberg, e in particolare Anton von Webern, ad elaborare la teoria seriale sui parametri musicali differenti dall'altezza, come ritmo, dinamiche e forma a cui si ispirerà la nuova generazione di musicisti seriali: Boulez, Stockhausen, Pousseur, Nono. [7, 28]

Ispirandosi ai procedimenti di trasposizione, moto retrogrado, inversione e loro combinazioni, i musicisti seriali realizzano quindi nuove esperienze musicali riportando in auge l'interesse per la regola e per l'attività speculativa e sottostando a rigide regole predeterminate, spesso in base matrici che costituirebbero l'algoritmo compositivo di fondo. [20, 28]

1.2 L'Era dei computer

1.2.1 Le prime macchine

La prima macchina in grado di produrre musica in modo più o meno autonomo fu costruita nel 1957 da Lejaren Hiller (1924 – 1994) e Leonard Isaacson presso l'università dell'Illinois. La macchina fu chiamata *ILLIAC* e tramite essa furono prodotte alcuni brani che entrarono in "*The ILLIAC Suite*" per quartetto d'archi. Il lavoro dei due musicisti fu descritto nel libro "Experimental music" (1959). Nello stesso periodo Hiller e Robert Baker

The image displays a musical score for a canon in G major, 2/4 time. It consists of five staves. The bottom staff (bass) is the given bass line from the Goldberg Variations. The other four staves (treble clefs) represent the other voices of the canon. The score shows the first four measures of the piece, with a double bar line and repeat sign at the end of the fourth measure. The notation includes various rhythmic values such as quarter notes, eighth notes, and sixteenth notes, along with accidentals and phrasing slurs.

Figura 1.2: Possibile soluzione del Canone 11 dai 14 canoni. Il basso è dato — basso delle *Variazioni di Goldberg* — mentre le altre voci formano due canoni contrari, la prima con la seconda e la terza con la quarta. Secondo Timothy Smith [29] il canone includerebbe inoltre una ricorrente simbologia bachiana, una *X* come simbolo di Cristo, rappresentata nelle ultime battute dall'incrocio tra il motivo di cinque semitoni discendenti in ottavi della seconda voce e di cinque sedicesimi ascendenti della prima, entrambi ripetuti per moto contrario dall'altra voce.

realizzarono *MUSICOMP* con cui scrissero *Computer Cantata*; *MUSICOMP* era progettato come una libreria di routine, in modo da permettere al musicista di adattare il sistema la proprio stile. [7, 20, 28] In entrambi icasi, e ancora per molto tempo, la musica venne eseguita da esseri umani. [20]

In entrambi i casi Hiller decise di produrre tramite la macchina uno spartito che venne poi rielaborato a mano laddove ritenuto necessario ed eseguito in un secondo momento da persone umane. [20]

È comunque nella seconda metà degli anni '50, da quando Hiller e Isaacson cominciarono a lavorare ad *ILLIAC*, che si gettarono le basi per la moderna musica algoritmica, che conquisterà i cuori di alcuni fra i più significativi compositori del Novecento, quali Stockhausen, Cage, Xenakis e Ligeti. Hiller, seguendo un'idea di Ada Augusta [20, 28], creatrice di una macchina ritenuta antenata dei moderni PC, fu sicuramente all'inizio di un cambiamento epocale della Storia della Musica, che, come spesso accade, fu mal visto dal mondo musicale ancora legato all'idea romantica di compositore, tanto che il Grove Dictionary of Music and Musicians inserì la voce di Hiller solo nel 1994, poco prima della sua morte. [7]

1.2.2 Determinismo e aleatorietà

Già nei secoli precedenti si possono individuare due approcci differenti alla musica algoritmica, i quali nei successivi anni diventeranno cruciali per l'evolversi della musica. Si tratta dell'elemento probabilistico e deterministico; infatti, mentre *ILLIAC* seguiva un procedimento aleatorio in base a Categorie di Markov [7] e Iannis Xenakis scriveva *Pithoprakta* (1956) e Karlheinz Stockhausen produceva *Klavierstück XI* (1956) [28], nel 1955 veniva eseguito per la prima volta *Le marteau sans maître* di Pierre Boulez e poco prima Stockhausen scriveva *Klavierstück III* [13]. Questi brani ben testimoniano che il binomio tra algoritmo deterministico e probabilistico non fu dovuto in alcun modo alla comparsa dei sistemi di generazione musicale, ma gettò le basi già prima di *ILLIAC*.

Sicuramente, il maggior compositore che sposò la musica aleatoria fu John Cage (1912 - 1992). Autore di numerosi brani di spiccato senso riflessivo e intellettuale, John Cage si dedicò anche alla computer music. Brani noti di Cage sono *Reunion*, in cui le note sono determinate in base alle mosse di una partita a scacchi e *Atlas Eclipticalis* in cui la musica viene scelta sovrapponendo il pentagramma ad una mappa delle stelle [20]. Cage collaborò con Hiller a *HPSCHD* (16 Maggio 1969), un brano per 7 clavicembali basato sull'elaborazione di musica dal *Musikalisches Würfelspiel* di Mozart e altri

compositori [7, 28]; al riguardo Cage affermò: “è la macchina che ci aiuterà a scoprire se comprendiamo il nostro stesso processo di pensiero”. [28]

Per quanto riguarda gli algoritmi deterministici nella composizione assistita da computer, essi si sono sviluppati maggiormente solo in una seconda fase; esempi sono brani di David Cope (*Experiments in Musical Intelligence*) e la macchina di William Shottstaedt per la composizione automatica delle specie di contrappunto basate sul *Gradus ad Parnassum* (1725) del Fux. [7, 20].

1.2.3 Iannis Xenakis

Xenakis (1922 - 2001) è stato uno dei compositori che più hanno fatto riferimento alla composizione algoritmica in tutto il Novecento, divenendo punto di riferimento per ogni autore che voglia rapportarsi al mondo della musica e matematica.

Musicista, ingegnere, architetto, Xenakis utilizzò a fondo l’approccio stocastico, che lo portò all’*SMP* (Stochastic Music Programme) e a brani totalmente innovativi e sconvolgenti, come il già citato *Pithoprakta* in cui i movimenti delle note sono associati alla teoria cinetica dei gas di Maxwell-Boltzmann, e *Eonta* (1963 - 1964) creato con *SMP*. [7] Altri brani da ricordare sono *Atrées* (1962) e *Morsima-Amorsima* (1962), composti lasciando scegliere al calcolatore le note in base a densità di probabilità specificate dall’autore. [20]

Oltre ad essere un fedele seguace dei metodi stocastici, anche Xenakis era legato all’“adattamento” del prodotto dell’algoritmo, come Hiller e molti altri. [7] È una posizione che lo accomuna alla musica algoritmica fin qui analizzata, dove, a parte Mozart, il compositore aveva sempre la possibilità di intervenire per migliorare il brano. Riguardo ad *Atrées* (1962), Nouritza Matossian afferma che Xenakis usò per il “75% materiale generato dal computer”, lasciando il resto a se stesso. [20] Xenakis era solito utilizzare il computer solo per elaborare dati da lui inseriti ed associati manualmente agli elementi musicali.

Per uno studio approfondito su Iannis Xenakis si rimanda per problemi di spazio ai testi specializzati, in quanto la grande maggioranza della sua produzione meriterebbe di essere trattata. [12]

1.2.4 György Ligeti

György Ligeti (1923–2006) è conosciuto al grande pubblico grazie alle colonne sonore dei film di Stanley Kubrick. *Lux Aeterna*, *Requiem*, *Lontano*,

Musica Ricercata furono utilizzati da Kubrick, anche con litigi sui diritti d'autore. [7, 11]

Ligeti, ungherese di famiglia ebrea, visse il nazismo e lo stalinismo, per poi lasciare l'Ungheria alla volta della Francia. Qui ebbe contatti con Gottfried Michael Koenig e Stockhausen, a fianco dei quali lavorò pur producendo anche lavori in proprio. Il suo interesse per la matematica e la geometria lo portò a comporre brani influenzati da frattali e teoria del caos, ad esempio, ma non giunse mai ad un approccio totalmente algoritmico. Fu lo stesso Ligeti ad affermare che esiste da qualche parte “un luogo comune nel nostro spirito” dove la musica e la matematica si incontrano, ma non incontrano mai il livello degli algoritmi o la composizione tramite calcolo. [7]

Parrebbe comunque che in alcune opere siano effettivamente rintracciabili dei veri e propri algoritmi, come in *Monumento* (1976) e nei tre libri di *Etudes* per pianoforte (1985, 1988-94, 1995). [11] Inoltre alcuni studiosi hanno ricostruito algoritmi di brani di Ligeti, ad esempio *Désordre*. [7]

Certamente, Ligeti non usava computer, ma è comunque difficilmente pensabile che non si servisse di algoritmi; più probabilmente la computazione degli algoritmi avveniva a mano, rendendo ancora più labile il confine tra musica algoritmica senza e con l'ausilio dei computer. [7]

1.2.5 Musica algoritmica e popular music

La musica algoritmica è stata trattata anche da compositori di formazione non accademica, come Brian Eno (1948–) — in *Music for Airports* (1978) e altri — su ispirazione dell'americano Steve Reich (1936–) e degli altri minimalisti americani. In questo caso, il momento della generazione e dell'esecuzione coincide poiché una volta impostato, il sistema è lasciato libero di generare la sua musica. [7] È evidente che l'intervento da parte dell'autore è pressoché nullo; il compositore diventa, come dice Eno, un “pianificatore” di musica. [7]

Altri sistemi sono pensati per ricreare specifici stili musicali, come gli esperimenti di David Cope o di George Lewis; Lewis, in particolare, programmò negli anni Ottanta “Voyager” in modo da permettergli di imparare da un musicista umano e poter improvvisare insieme a lui. Strumento analogo è OMAX, progettato all'IRCAM di Parigi. [7] *Emily Howell* di David Cope invece è stata progettata con l'obiettivo di rendere la macchina capace di imparare uno stile in base ai giudizi dell'ascoltatore. [4]

Capitolo 2

Musica e catene di Markov: studi precedenti

2.1 Catene di Markov

Per questa parte si è fatto riferimento al volume *Calcolo delle probabilità e statistica* di Paolo Baldi [2]. Sia data la definizione di *catena di Markov*:

Catena di Markov Si chiama catena di Markov un processo stocastico a tempo discreto tale che:

- Le variabili aleatorie X_n che costituiscono il processo assumono valori in un insieme discreto E ;
- Esistono numeri positivi p_{ij} , con $i, j \in E$, $n \in \mathbb{N}$ tali che:

$$\begin{aligned} P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1\} = \\ = P\{X_{n+1} = j | X_n = i\} = p_{ij}(n) \end{aligned}$$

In altre parole, detto E l'insieme degli stati del sistema, una catena di Markov è un processo in cui la conoscenza dello stato attuale X_n dà informazioni sullo stato futuro X_{n+1} , ma non sugli stati X_{n+i} con $i > 1$. Lo stato attuale del sistema dipende cioè solo dallo stato X_{n-1} e non da X_{n-2}, X_{n-3} etc.

In realtà vengono considerate anche le catene il cui stato dipende dagli i stati precedenti, dove i è detto *ordine* della catena.

Le catene qui trattate sono inoltre *omogenee nel tempo*, cioè il valore di $p_{ij}(n)$ è indipendente da n , in questo caso rappresenta il tempo.

Supponendo che l'insieme degli stati E sia finito, sia data ora la definizione di *matrice di transizione*:

Matrice di transizione Si chiama matrice di transizione associata a una catena di Markov una matrice di ordine E tale che:

- Il suo elemento E_{ij} è la probabilità p_{ij} di passare dallo stato E_i allo stato E_j ;
- Tutti i suoi elementi sono ≥ 0 ;
- La somma di tutti gli elementi di ogni riga vale 1.

Si danno ora alcune definizioni utili a classificare le matrici di transizione. Innanzitutto si dice che lo stato $i \in E$ *comunica* con $j \in E$ se $\exists n > 0 \mid p_{ij}(n) > 0$. Nel nostro caso, poiché tratteremo catene omogenee nel tempo, la definizione sarà indipendente da n .

Si dice che $C \subset E$ è una *classe chiusa* se tutti gli stati di C non comunicano con il complementare di C . Se tutti gli stati di C comunicano tra loro, essa è detta *irriducibile*. Se uno stato costituisce da solo una classe irriducibile, esso è detto *assorbente* — uno stato che comunica solo con se stesso è ad esempio lo stato di equilibrio di un sistema. Una catena di Markov è detta *irriducibile* se E è l'unica classe irriducibile.

2.2 Generatori tramite catene di Markov

Fra i lavori già svolti sulla modellizzazione musicale tramite catene di Markov, si citano alcuni articoli utili al lavoro proposto.

Uno è di Chuan e Chew [6], i quali hanno creato un sistema per l'armonizzazione automatica di melodie. Il loro sistema, chiamato *ASSA*, si basa sulla teorizzazione geometrica del *Tonnetz* e su alcune misure metriche tipiche della *teoria dei segnali* come *entropia* e *perplexity*, oltre che, ovviamente, sulle catene di Markov. L'idea è quella di predire la probabilità di un accordo o delle sue varianti dato un frammento di melodia, supponendo che lo stesso frammento possa avere differenti armonizzazioni — si pensi a melodie popolari o agli standard jazz, ad esempio. Pur non essendo esplicitamente rivolto all'analisi musicale, l'idea di utilizzare queste metriche può essere interessante, soprattutto in relazione al lavoro di Yi-Wen Liu [18] (vd par. 2.3).

Altri lavori si sono invece concentrati su come sia possibile imporre alcuni vincoli sulle catene generate in modo da realizzare una costruzione formale.

Il primo è quello di Pachet, Roy e Barbieri [25] in cui l'idea di base è quella di modificare la matrice di transizione inibendo la possibilità di cadere sugli stati che i vincoli imposti proibiscono, implicitamente o esplicitamente. Questo approccio ha ottimi effetti sull'elaborazione in tempo reale, perché non necessita di conoscere l'intera catena per imporre i vincoli. Nel lavoro ci si preoccupa del fatto che imponendo degli zeri in certi elementi della matrice di transizione, si otterrebbe un processo non stocastico e sarebbe quindi necessaria una normalizzazione dei valori ottenuti; nel software da me sviluppato, vengono memorizzate solo il numero di occorrenze degli stati, lasciando il calcolo della probabilità al momento dell'elaborazione, non ci sarebbero quindi problemi di normalizzazione.

Il secondo lavoro è invece di Ronnander [17] e si basa sull'individuazione di catene per sezioni differenti; questo permette di caratterizzare bene lo stile di differenti sezioni — idealmente i divertimenti di una fuga sarebbero resi ben marcati rispetto all'incipit o al finale, oppure i primi temi di una sonata rispetto ai secondi e rispetto ai ponti e allo sviluppo. Lo stato della ricerca è comunque ancora da sviluppare in questo senso: il lavoro di Ronnander si limita all'individuazione di una frase iniziale, una finale e una sezione centrale.

2.3 Analisi musicale tramite catene di Markov

Per quanto riguarda l'analisi musicale tramite catene di Markov, è stimolante una relazione di Yi-Wen Liu [18] riguardo ad un esperimento in cui tramite le catene di Markov si è cercato di modellizzare gli stili degli autori per provare ad identificare la paternità di altri brani misurando la distanza di Kullback-Leibler. Principale limite di questo lavoro è l'aver utilizzato solo catene del primo ordine; come si vedrà, nel mio esperimento si è trovato che per ordini inferiori a 2 la catena non è solitamente in grado di ripetere le caratteristiche stilistiche di un brano.

Di conseguenza potrebbe essere proficuo ripetere le misure con Kullback-Leibler usando però catene di ordine superiore, eventualmente sfruttando la ricerca degli ordini ottimi di cui al par. 3.3.2.

Un altro lavoro che si cita sull'analisi musicale tramite catene di Markov è quello di David M. Franz [8] che, dati gli accordi di uno standard jazz, ha modellizzato la melodia improvvisata dai grandi jazzisti. Da qui propone una serie di possibili misure di parametri interessanti, come *fluidità*, *flessibilità*, *elaborazione* e *originalità*. L'approccio di questo lavoro è certamente

molto filosofico e pertanto si presta a critiche di pensiero; in ogni caso il significato delle misure citate è interessante perché di indubbio valore musicale; resta aperta la discussione sul metodo utilizzato per effettuare tali misure.

Capitolo 3

Software generatore di melodie

Il software prodotto è un generatore di musica tramite catene di Markov. Verranno qui riassunti le principali caratteristiche, i problemi e le conclusioni a cui si è giunti, nonché le migliorie da implementare.

3.1 Algoritmo

L'algoritmo utilizzato è di estrema semplicità. Durante l'implementazione del programma si è variato l'algoritmo per implementare la possibilità di utilizzare brani a più voci per allenare il sistema, in modo da ottenere brani polifonici in uscita. Verranno quindi descritti i due algoritmi, sebbene il secondo presenti ancora alcuni problemi.

3.1.1 Algoritmo a una voce

Il problema principale di applicare le catene di Markov alla musica è la difficoltà di individuare adeguatamente gli stati del sistema. Ad esempio, sarebbe possibile scegliere di individuare manualmente, eventualmente anche con un sistema automatizzato, gli elementi musicali principali e quelli secondari; gli stati potrebbero allora scaturire dalla combinazione delle due classi di elementi. Se si individuassero tutte le note reali e le si distinguessero dalle note di abbellimento, passaggio etc., si renderebbe una notevole semplificazione dei brani, che potrebbe forse permettere una migliore efficienza del sistema.

Nonostante queste osservazioni, si è scelto per semplicità del programma, di seguire il metodo più ovvio: si considera come stato del sistema una

sequenza di n note, e si parlerà quindi di “stato a n note”. Questo tipo di modellizzazione darà sicuramente problemi per valori di n bassi, mentre sarà senz’altro migliore per n più alti. Analogamente converrà utilizzare catene di ordine più alto, poichè gli ordini bassi riusciranno con difficoltà a ripetere le caratteristiche stilistiche dei brani di input.

In particolare, si sono presi in considerazione due possibili algoritmi per individuare stati di questo tipo. Sia definito il multiinsieme delle note V come, il multiinsieme che contiene tutte le note del brano, ciascuna con cardinalità determinata dal numero di volte per cui viene ripetuta all’interno del brano. Nel primo si sono considerati tutti gli stati A e B di n note senza intersezioni tra di loro, cioè tali che, detta $a \in V$:

$$a \in A \wedge a \in B \Rightarrow A = B$$

In altre parole, secondo questo metodo, gli stati A individuati sono partizioni di V . Questo metodo verrà in seguito chiamato *standard method*. Nel secondo metodo, qui denominato *note-by-note method*, si sono invece considerate tutte le possibili combinazioni di n note che si presentano nel brano. In questo modo si evita l’effetto dato dal metodo precedente per il quale se il numero di note dell’intero brano non è multiplo di n si ottiene uno stato ‘monco’ alla fine.

Data una sequenza di note $[a, b, c, d, e, f, g, h, i, l, m, n, o]$ gli stati individuati saranno i seguenti:

1. Standard Method

- a, b, c
- d, e, f
- g, h, i
- l, m, n
- o

2. Note-by-note method

- a, b, c
- b, c, d
- c, d, e
- d, e, f
- e, f, g

- f, g, h
- g, h, i
- h, i, l
- i, l, m
- l, m, n
- m, n, o

Data una sequenza di q note differenti tra loro, con lo *standard method* si ottengono m stati dove m è il più piccolo intero $\geq \frac{q}{n}$, con il *note-by-note method* si ottengono esattamente q stati.

A questo punto è facile intuire come si forma la matrice di transizione. Per la seguente sequenza di note: $[a, b, c, d, a, b, c, e, f, d, a, b, g, h, i]$ lo *standard method* genera la seguente.

	a, b, c	d, a, b	c, e, f	g, h, i
a, b, c	0	1	0	0
d, a, b	0	0	0.5	0.5
c, e, f	0	1	0	0
g, h, i	0	0	0	0

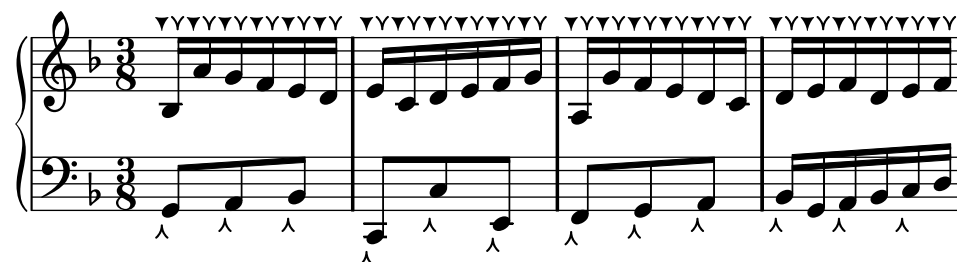
Si noti che lo stato finale non presenta nessuno stato verso cui andare, esso è quindi uno stato anomalo. Questo può portare ad un loop; si consideri questa sequenza di stati: $A, B, C, A, A, B, D, E, D$. In questo caso allo stato D segue solo E , e allo stato E segue solo lo stato D ; di conseguenza, supponendo una catena di ordine 1 con stati ad una nota, giunti sullo stato D si otterrà necessariamente una catena come la seguente: $[...D, E, D, E, D, E, etc]$, quindi si avrebbe una *classe chiusa* che comprometterebbe il funzionamento (vd. par. 2.1). Per evitare questo sconveniente si è scelto di dare la possibilità alla catena di procedere dallo stato D al primo stato, in questo caso A , in modo da interrompere il loop.

Come ultima osservazione, si osservi che una catena di ordine k si differenzia da una catena di ordine 1 solo per la definizione dello stato; di fatto, l'implementazione per la produzione della catena una volta allenato il sistema è identica e indipendente da n . Inoltre, utilizzando il *note-by-note method*, stati ad n note producono catene di ordine 1 identiche a quelle prodotte da catene di ordine n con stati ad una nota; Lunghezza dello stato e ordine della catena sono pertanto concetti interscambiabili per questo metodo.

3.1.2 Algoritmo a più voci

Per quanto riguarda l'implementazione di più voci, il problema consiste ancora nel determinare quale sia il modo migliore per identificare gli stati. Per evitare problemi di dissonanze, il metodo ritenuto più semplice è quello di considerare le varie voci in ogni istante. Questo sistema porta ad una concezione della musica simile al *voice-space leading* di Tymoczko [31].

Per comprendere meglio come procede il software si osservi il seguente passo dell'*Invenzione n. 4* di Bach:



Le frecce verso l'alto sono disposte ogni ottavo, quelle verso il basso ogni trentaduesimo, quelle piene verso il basso ogni sedicesimo. A seconda delle frecce che si stanno considerando, si possono individuare degli stati differenti; osservando il passo ad ogni ottavo si ottengono i seguenti accordi:



L'idea è che a ciascuno di questi accordi deve corrispondere uno stato in una catena ad 1 nota; in una catena ad n note, uno stato sarà costituito dalla successione di n di questi accordi. Si capisce allora che la precisione con cui si osserva lo spartito è estremamente importante: in questo caso, ad esempio, estraendo gli stati ad ogni ottavo, si perdono molte informazioni. Si possono allora adottare due approcci:

- utilizzare come precisione la durata della nota più breve di tutto il brano.
- utilizzare la precisione più piccola che il sistema mette a disposizione.

Benché il sistema migliore sia chiaramente il primo, il software al momento non prevede un algoritmo per individuare automaticamente la precisione migliore, anche se sarebbe comunque di facile implementazione. Ad ogni modo, è sempre necessario prevedere la possibilità di incontrare note

di durata superiore alla precisione; si è scelto di aggiungere alla nota l'informazione per cui essa è una nuova nota o una nota già iniziata; è importante distinguere tra una nota già iniziata a causa di una legatura di valore e una nota già iniziata per l'algoritmo di ricerca degli stati: sono due informazioni differenti.

Un sistema del genere potrebbe funzionare sia con lo *standard method* sia con il *note-by-note method*, ma si è scelto di implementare, al momento, solo il primo algoritmo.

Altro elemento degno di nota è la differenza tra la modellizzazione di un brano a voce sola con l'algoritmo ad una voce e con quello a più voci. Si vedrà che per ottenere un livello intermedio tra libertà del software ed aderenza ai brani di input, è spesso più conveniente l'algoritmo a più voci, perché, aumentando la precisione, aumenta anche la gradazione con cui si passa da una bassa aderenza alla copia totale dei brani di input al variare dell'ordine della catena (vd par. 3.6).

Per quanto riguarda il tempo, il sistema distingue note in tempo binario da note in tempo ternario; JFugue infatti adotta una conversione da durata in secondi normalizzata con la durata della misura a lettera dell'alfabeto — *s* per sedicesimo, *h* per metà e così via. JFugue non è in grado di operare questa conversione sulle note di tempi ternari e in generale su tutte le note di durata differente da 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ o $\frac{1}{32}$, etc. Poiché l'individuazione degli stati si basa sulla loro rappresentazione in forma di stringa JFugue, il sistema impedisce di mischiare note prese da un brano ternario con note prese in un brano binario. Questa caratteristica è stata considerata un buon comportamento.

Ultima precisazione opportuna è che nei casi di voci temporanee o di accordi non facilmente distinguibili in voci autonome — fenomeno tipico della musica strumentale — il software prevede di considerare gli accordi come voce unica, una sorta di 'nota speciale'. Vi sono però delle difficoltà dovute alla non totale compatibilità di questi passaggi durante la trasformazione da musicxml o midi in stringa JFugue [15].

3.2 Architettura

3.2.1 Strumenti utilizzati. Java e JFugue

Il software costruito è stato implementato in Java, con l'obiettivo di avere a disposizione un prodotto facilmente portabile, eventualmente anche su server online — tramite l'uso di server Tomcat — o su sistemi Android e smartphone in generale.

Si è scelto di usare la versione 7 di OpenJDK [23] per garantire il massimo rispetto degli standard; purtroppo però, l'implementazione a più voci è stata realizzata con OracleJDK 8 [24] per problemi tecnici — la nuova versione della libreria JFugue attualmente in beta non è compatibile con la versione 7 dello standard Java.

La libreria utilizzata è stata JFugue, che ha come principale caratteristica la rappresentazione della musica sotto forma di stringhe. [16] In particolare questo tipo di rappresentazione è stata adottata nella versione ad una voce, in cui si è utilizzata l'attuale versione standard. Nella versione a più voci si è invece fatto maggiormente uso del paradigma ad oggetti, introducendo una classe 'MarkovState' che rappresenta uno stato come matrice di note $m \times n$, dove m è il numero di voci e n è l'ordine dello stato.

In questa seconda versione si è anche utilizzata una classe 'MyNote' che estende la classe 'Note' di JFugue aggiungendo l'informazione per cui essa è una nuova nota o meno (vd par. 3.1.2 pg. 20).

È stato inoltre reimplementato un nuovo 'ParseListener' che elimina le informazioni non interessanti o che comunque il software non è in grado di elaborare.

3.2.2 Design patterns e diagrammi UML

Lo sviluppo è proceduto con l'idea di realizzare una sorta di libreria per sistemi basati su catene di Markov, eventualmente integrabile con JFugue.

Per questo motivo si è scelto di usare il design pattern 'Mediator' [10] che permette di mantenere indipendenti i vari oggetti sviluppati. Ovviamente il 'Mediator' può possedere più implementazioni, ed è perciò protetto da un'interfaccia. Questa architettura permette di scegliere tra *standard method* e *note-by-note method* a run time, semplicemente modificando il creatore di matrici di transizioni e l'oggetto costruttore di catene — 'IMatrixCreator' e 'IChainBuilder'.

Il Diagramma UML risultante è in figura 3.1; il ruolo di 'DefaultHistogram', di 'SpecialStandardBuilder' e di 'SpecialStandardMatrixCreator' verrà spiegato nel paragrafo 3.3.

Si osserva che le matrici di transizioni implementano un 'Fly Weight' [10] in quanto ogni stato deve essere presente una sola volta all'interno delle chiavi della matrice. L'implementazione della matrice di transizione, infatti, consiste sostanzialmente in un campo di tipo 'HashMap' che ha per chiavi gli stati — quindi tutti differenti tra di loro — e per argomenti un 'ArrayList' di stati ciascuno abbinato al numero di volte che ricorre nel brano.

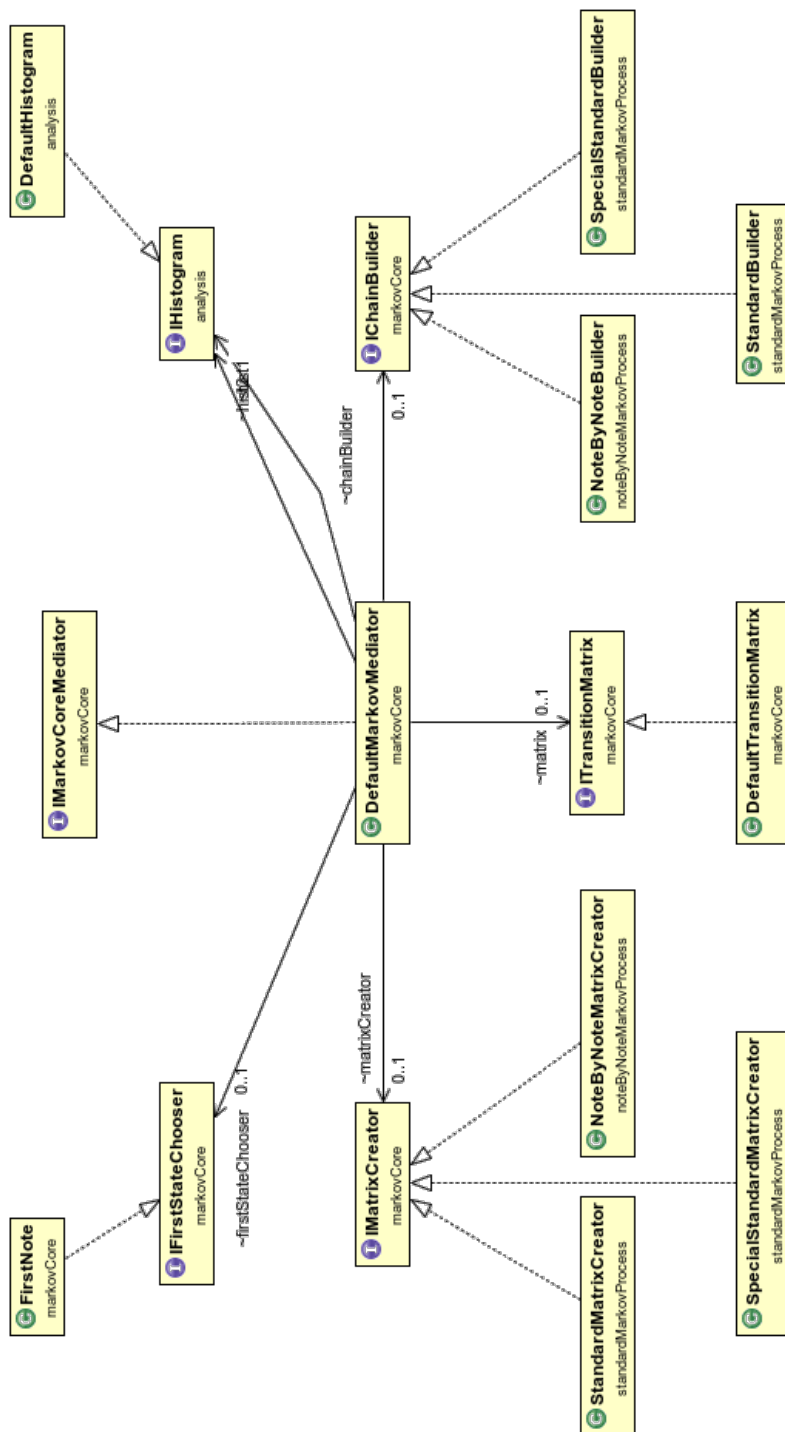


Figura 3.1: Diagramma UML

3.3 Migliorie

3.3.1 Matrice principale e matrice secondaria

L’algoritmo descritto al paragrafo 3.1 permette di creare matrici di transizione da cui poi facilmente costruire le catene. Purtroppo però, quando si utilizzano stati ad n note con valori di n alti e si allena il sistema su un numero di brani > 1 si ottiene l’effetto di avere due matrici di transizione incluse nello stesso ‘TransitionMatrix’ ma non comunicanti. In altre parole, la matrice di transizione risultante non è *irriducibile* e i brani non hanno stati *comunicanti* se si ricerca in un brano B una lunga sequenza di note estratta da un brano A. (vd par. 2.1). Di conseguenza, il valore di n per il quale il sistema riproduce solo stati di un determinato brano può rivestire un significato rilevante nell’analisi dei brani. Di questo si parlerà al paragrafo 3.3.2.

Ciò che preme sottolineare qui, è che se i brani musicali di input sono molto diversi, basterà un valore di n , a volte anche solo 3 o 4, per ottenere questo effetto. Ancora maggiore, come si può immaginare, è il problema se oltre all’ordine aumenta anche il numero di voci.

Per risolvere questa difficoltà si è introdotta una matrice secondaria con stati brevi e uno stato ‘special’ in occorrenza del quale si passa all’estrazione della catena tramite la matrice secondaria. Dopo un’estrazione dalla matrice secondaria si cerca di ritornare alla matrice principale — creata con n alto — ricostruendo uno stato di n note tramite le $n - p$ ultime note dello stato che ha causato lo stato ‘special’ e le p note ottenute dalla matrice secondaria. Se lo stato così calcolato non è presente nella matrice principale si continua con quella secondaria.

L’occorrenza di uno stato ‘special’ o di uno stato non presente in nessuna matrice genera una ‘Special State Exception’ che causa lo switch della matrice nel mediator, il quale dovrà ora contenere due matrici, una principale e una secondaria, realizzando così una doppia dipendenza da ‘TransitionMatrix’.

L’implementazione poteva in realtà essere ottenuta anche con un’unica matrice di transizione, usando quindi il ‘DefaultMarkovMediator’ semplicemente inserendo tutti gli stati nella matrice, indipendentemente dalla lunghezza dello stato, creando volontariamente un’ulteriore classe irriducibile; si è preferita comunque questa soluzione per rendere il sistema più modulare.

Attualmente, il sistema a doppia matrice è implementato solo ad una voce; grazie a questo metodo si è assistito ad un netto miglioramento del prodotto risultante (vd par. 3.3.2). Attualmente è possibile impostare la

probabilità di cadere in uno stato ‘special’ fino a 0.5; per valori compresi tra 0.5 e 1, si possono ottenere solo i valori di probabilità pari a

$$\frac{x}{x+1} \forall x \in \mathbb{N}$$

.

3.3.2 Istogrammi

Si è detto che uno dei problemi è stato quello di individuare la lunghezza ottima degli stati per la catena, ovvero quella che garantisce una sufficiente aderenza all’input in modo da possedere un certo “senso musicale”, ma che al contempo non si riduca ad essere un semplice rimescolamento di parti di un unico brano.

Ovviamente questo problema non è di facile soluzione, ma si è cercato comunque di facilitare l’individuazione manuale della lunghezza ottima.

In particolare, si è deciso di calcolare l’istogramma delle note presenti nei brani; in presenza di più voci, ciascuna voce è disposta in sequenza alle altre.

Si è valutato poi il quadrato (Ξ) della norma della differenza tra gli istogrammi calcolati sulla catena prodotta dal sistema e quelli calcolati sui brani originali. Come atteso, allenando il sistema su un gruppo di brani $Q = \{A, B, C, D\}$, Ξ presenterà valori nulli per tutti le lunghezze σ maggiori di σ_0 , dove σ_0 è la più piccola lunghezza degli stati per la quale $\Xi = 0$.

Più precisamente, definiamo il quadrato della norma della differenza di due istogrammi Ξ come

$$\Xi_{(Q, Q_i, \sigma)} = \|H_{(M_Q^\sigma)} - H_{(Q_i)}\|^2$$

con M_Q^σ catena di Markov generata dal sistema allenato su tutto Q con ordine σ , Q_i è un singolo elemento di Q e $H_{(\cdot)}$ funzione istogramma.

Allora, data $\bar{\sigma} = 0$

$$\bar{\sigma} = \sigma_0 \Leftrightarrow \Xi_{(Q, Q_i, \sigma)} > 0 \forall \sigma < \bar{\sigma}$$

Usando stati ad n note, il valore di σ_0 potrebbe essere utile all’identificazione dell’autore o dello stile del brano (vd. par. 2.3) poiché quasi sempre — ma non necessariamente — indica il più piccolo valore di n per il quale la catena prodotta è identica al brano Q_i . Si potrebbe pensare ad una sorta di “memoria” compositiva, specie se si considera il *note-by-note method*. Purtroppo non si ha avuto la possibilità di testare il sistema su un grande numero di brani, quindi non può essere affermata con certezza l’utilità di σ_0 all’analisi musicale.

Tuttavia, σ_0 è certamente un valore utile per determinare la lunghezza ottima $\bar{\sigma}$ degli stati poiché essa sarà certamente inferiore a σ_0 . Inoltre, si nota una diminuzione più o meno graduale di Ξ all'aumentare di σ . Dando per certo che gli stati a 1 e 2 note non sono ottimi, fatto valutato empiricamente, si può pensare di individuare la lunghezza ottima ottimo come

$$2 < \bar{\sigma} < \sigma_0$$

Una prima formula potrebbe essere l'ordine medio tra 2 e σ_0 :

$$\bar{\sigma} = \frac{\sigma_0 + 2}{2} = \frac{\sigma_0}{2} + 1$$

Si noti che utilizzando il sistema a doppia matrice, gli istogrammi restano molto più stabili e non è più individuabile il valore σ_0 . Questo significa che è possibile produrre catene con stati lunghi ottenendo la stessa aderenza ai brani di input degli stati brevi; per ottenere buoni risultati è comunque sconsigliabile utilizzare stati ad n note con valori grandi di n per evitare delle lunghe sequenze provenienti da un unico brano o dalla matrice secondaria — con n grande nella matrice principale, è improbabile riuscire ricreare uno stato partendo dalle catene della matrice secondaria costituite da stati brevi. Allo stesso modo con stati a poche note nella matrice principale, si ottengono comunque sequenze senza un “senso” musicale. Un buon metodo potrebbe essere studiare gli istogrammi con una sola matrice, determinare da qui la lunghezza ottima, ed utilizzarla nel sistema a matrice doppia.

3.4 Risultati e commenti

Di seguito si riportano i risultati degli istogrammi valutati sulla Partita per Flauto BWV 1013 di J.S. Bach individuando stati ad n note. Si noti come i valori, una volta andati a zero, non vi restano in modo costante, ma possono esistere valori di n in cui alcune formule musicali ripetute danno luogo a stati comuni, presenti in parti differenti del brano, e che di conseguenza generano dei valori degli istogrammi diversi da zero — è il caso ad esempio della *Corrente* con $n = 7$ modellizzata con lo *standard method*.

In questo caso, la lunghezza ottima dovrebbe stare tra 3 e 5.

Usando invece il *note-by-note method*, si ottengono valori leggermente diversi e tendenzialmente più stabili. σ_0 per questi brani tende ad essere più grande e quindi potrebbe in linea teorica dare luogo a migliori risultati.

Usando il sistema a doppia matrice, i valori di Ξ risultano essere molto più stabili e non andare mai a zero. Questo perché l'intervento di una catena a basso ordine impedisce al sistema di ripetere esattamente il brano e permette di utilizzare informazioni provenienti da più brani.

Tabella 3.1: Standard Method. Valori del quadrato della norma della differenza tra gli istogrammi di un brano e di una catena generata dal sistema allenato su tutti e quattro i brani al variare di n , cioè della lunghezza degli stati. Il numero di note della catena è uguale al numero di note del brano con cui viene confrontata, così come lo stato iniziale.

n	Allemanda	Corrente	Sarabanda	Bourrée
1	1446.61	7328.3	1985.05	5224.28
2	20688.64	13009.55	5264.44	7081.78
3	44951.01	36342.19	19052.56	14698.28
4	99433.51	32293.02	21961.17	1908.3
5	63374.21	421.74	8413.78	30.8
6	0.0	0.0	1226.74	0.0
7	0.0	30243.41	0.0	0.0
8	1561.28	0.0	0.0	375.2
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	126.72	0.0	0.0	212.8
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0

Tabella 3.2: Note-by-note Method. Valori del quadrato della norma della differenza tra gli istogrammi di un brano e di una catena generata dal sistema allenato su tutti e quattro i brani al variare di n , cioè della lunghezza degli stati. Il numero di note della catena è uguale al numero di note del brano con cui viene confrontata, così come lo stato iniziale.

n	Allemanda	Corrente	Sarabanda	Bourrée
1	625.0	4142.0	2340.0	5700.0
2	800.0	1304.0	1966.0	692.0
3	4809.0	20803.0	1672.0	559.0
4	510.0	3614.0	1905.0	1298.0
5	6555.0	12186.0	1814.0	1005.0
6	2816.0	1413.0	2416.0	2154.0
7	569.0	19293.0	159.0	237.0
8	352.0	9705.0	1533.0	310.0
9	2659.0	2410.0	1755.0	241.0
10	52.0	0.0	0.0	5021.0
11	195.0	0.0	0.0	5021.0
12	38.0	0.0	0.0	354.0
13	10515.0	0.0	0.0	5021.0
14	124.0	0.0	0.0	5021.0
15	241.0	0.0	0.0	5021.0
16	32.0	0.0	0.0	5021.0
17	32.0	0.0	0.0	323.0
18	106.0	0.0	0.0	0.0
19	0.0	0.0	0.0	5021.0
20	0.0	0.0	0.0	5021.0
21	0.0	0.0	0.0	287.0
22	0.0	0.0	0.0	288.0
23	0.0	0.0	0.0	5021.0
24	0.0	0.0	0.0	302.0
25	0.0	0.0	0.0	5021.0
26	0.0	0.0	0.0	314.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0
32	0.0	0.0	0.0	0.0

Tabella 3.3: Metodo a due matrici. In questo caso i valori sono ottenuti utilizzando una matrice primaria ad n note indicato e una matrice secondaria con stati a 2 note. La probabilità di cambiare matrice è 0.5

n	Allemanda	Corrente	Sarabanda	Bourrée
2	19411.8	7687.2	4962.2	5676.0
3	72764.2	12253.8	4613.4	16784.8
4	137114.2	18798.4	18187.0	6692.4
5	126491.2	19553.6	10487.8	34217.0
6	248450.4	32396.6	7764.2	35205.2
7	184003.2	21672.6	11288.6	12075.4
8	192781.8	38867.4	17205.8	39980.4
9	138857.0	20704.8	9021.0	61208.8
10	184492.2	49466.4	19072.0	64074.0
11	154232.2	17455.8	16820.6	16491.8
12	178457.6	18308.6	2747.8	19681.6
13	136784.8	21627.0	16041.2	29952.6
14	157549.6	10095.2	10838.6	62841.6
15	169983.0	27160.0	9463.0	3128.4
16	146317.0	24795.8	16216.0	40660.0
17	175140.0	11332.2	5681.6	40980.8
18	145645.6	11827.8	9064.2	22701.6
19	141284.4	36924.4	11652.8	48640.6
20	132261.6	24453.8	13764.6	52100.8
21	149576.8	18497.0	13449.8	15967.4
22	117536.2	5589.4	16363.2	25018.8
23	137704.8	14215.0	14379.8	19074.4
24	139673.6	13820.8	13182.4	35111.2
25	135478.6	28470.8	10804.8	17318.2
26	149578.8	30908.2	13193.2	17561.2
27	148377.0	8353.4	15446.4	25045.2
28	150539.4	13171.4	17697.4	15852.4
29	130352.8	14710.2	9388.2	7856.0
30	153236.6	30598.8	1+2291.8	27698.0

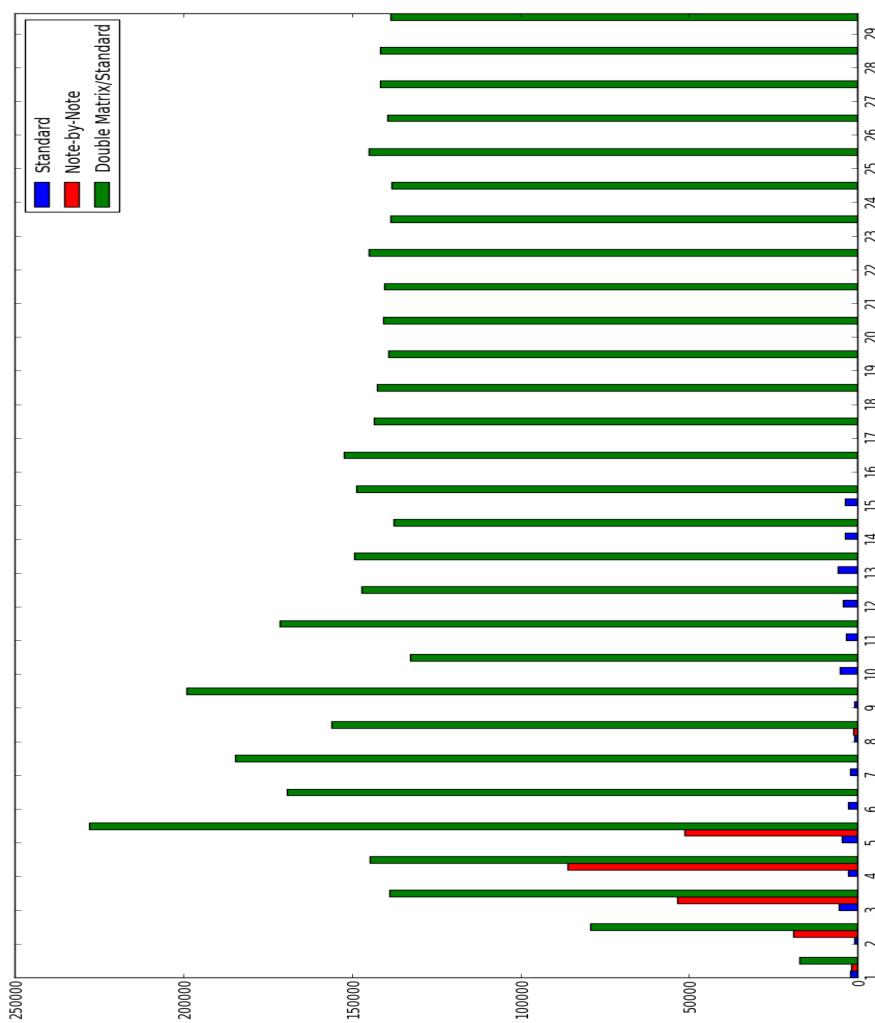


Figura 3.2: BWV1013, Allemanda. I valori di \mathcal{E} risultanti dai tre metodi presi in considerazione al variare della lunghezza degli stati.

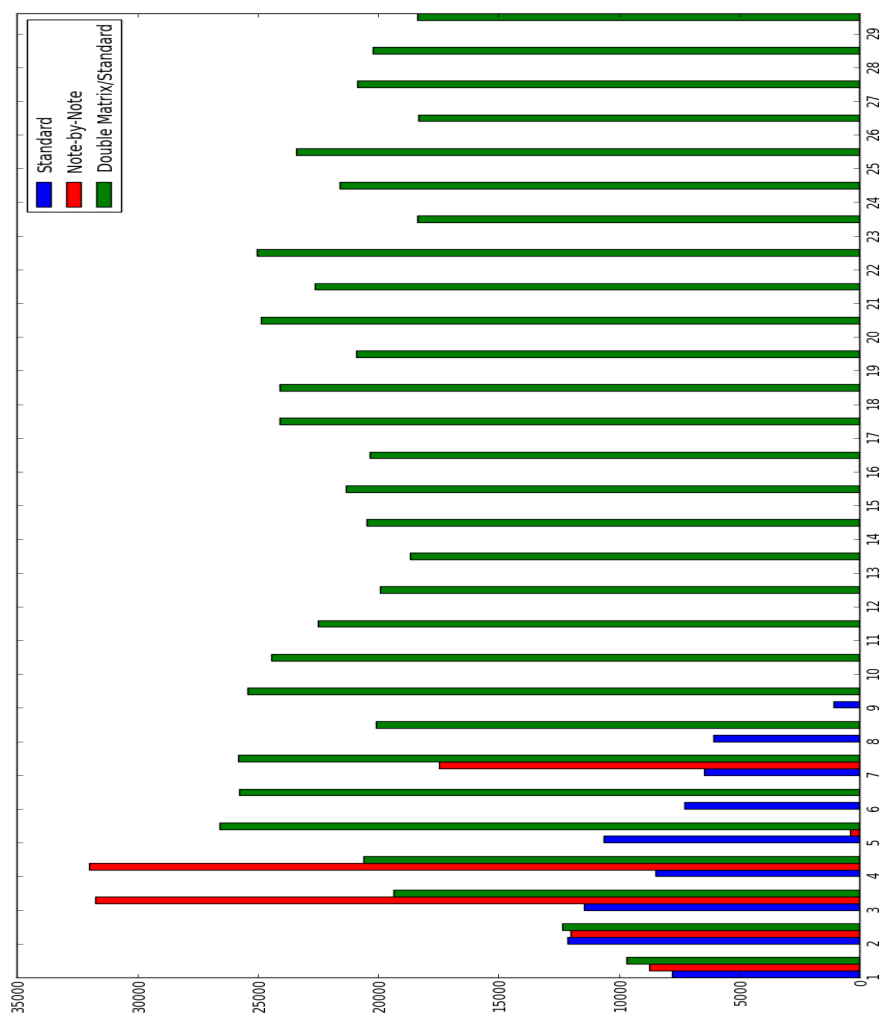


Figura 3.3: BWV1013, Corrente. I valori di \mathcal{E} risultanti dai tre metodi presi in considerazione al variare della lunghezza degli stati.

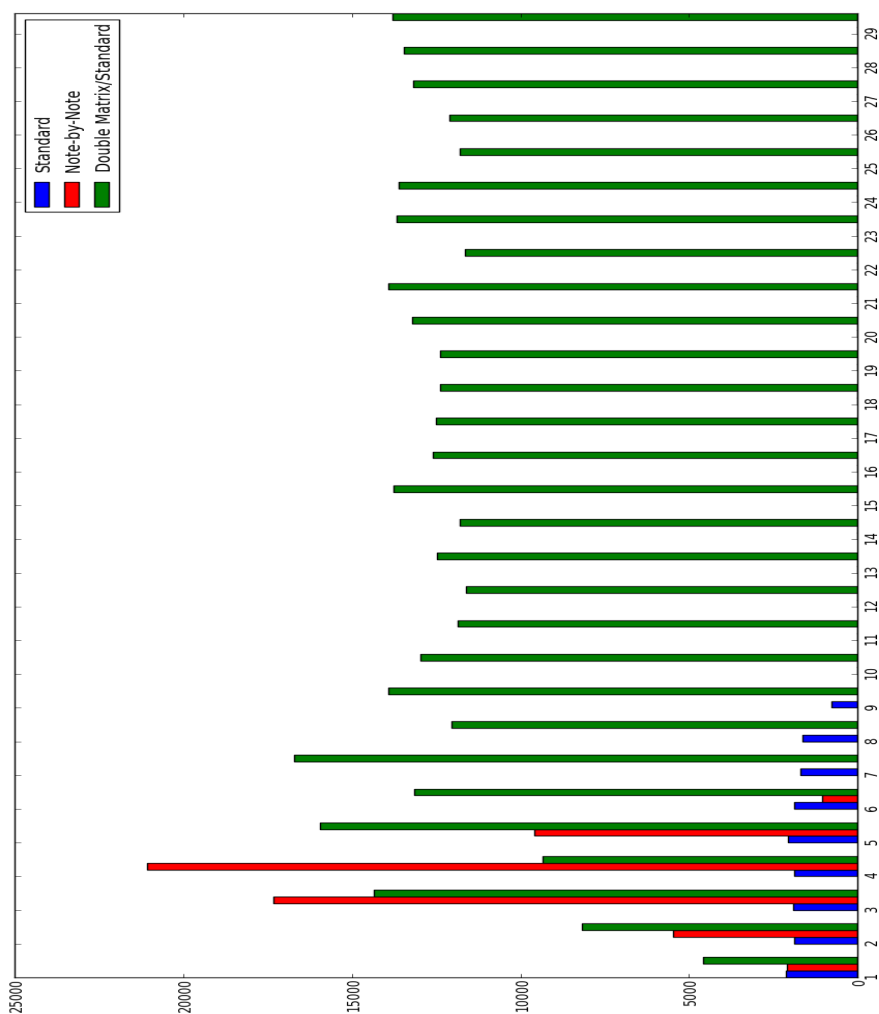


Figura 3.4: BWV1013, Sarabanda. I valori di E risultanti dai tre metodi presi in considerazione al variare della lunghezza degli stati.

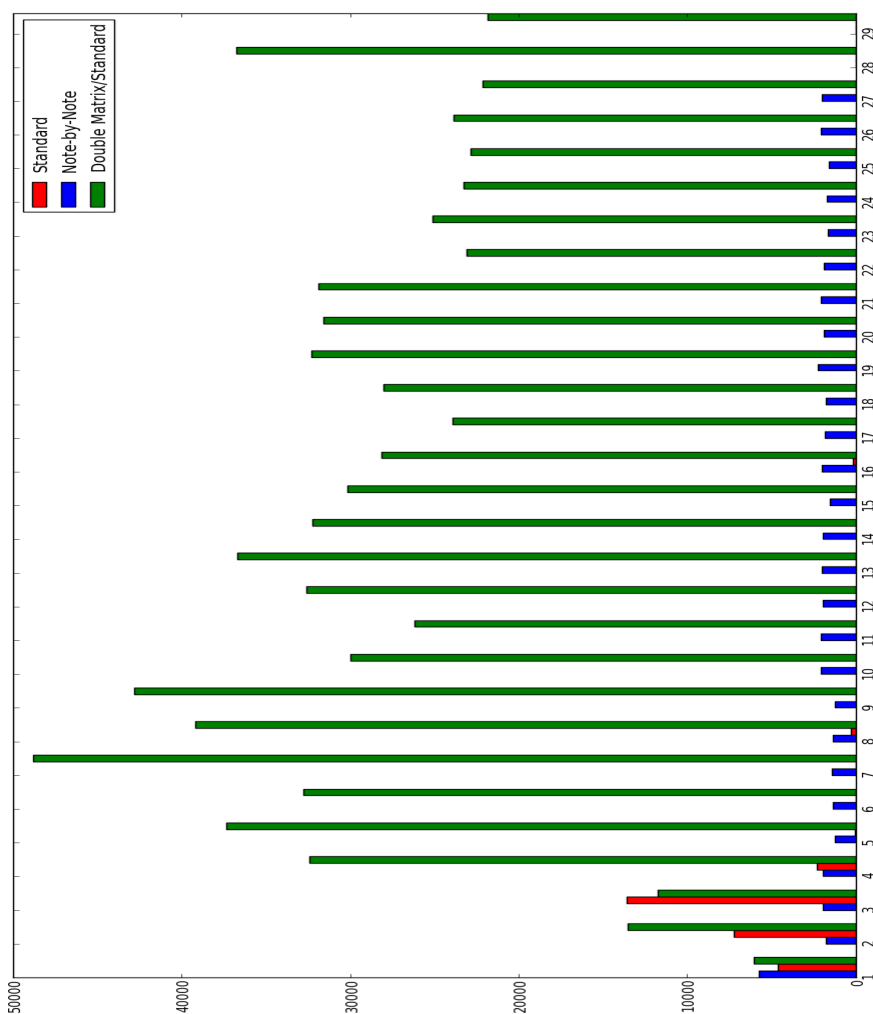


Figura 3.5: BWV1013, Bourrée. I valori di E risultanti dai tre metodi presi in considerazione al variare della lunghezza degli stati.

3.5 Limiti e problemi

I limiti del sistema sono principalmente:

- non completa implementazione, specie per il sistema a più voci;
- necessità di grandi basi di dati per testare il sistema;
- valori di σ_0 molto bassi per brani a più voci; questo limita molto e spesso rende impossibile una buona modellizzazione senza utilizzare il sistema a doppia matrice.

3.6 Possibili linee guida per il futuro

Per migliorare i problemi nei brani a più voci dovuti alla troppo forte caratterizzazione di piccoli frammenti – cioè di uno stato — si potrebbe pensare di sfruttare il *voice space leading* di Tymoczko [31]. È possibile che dalla formalizzazione geometrica possano nascere nuove idee o elementi su cui applicare le catene di Markov. Ad esempio, si potrebbero considerare come stati gli intervalli tra le note delle varie voci; occorrerebbero due catene, una per il ritmo e una per le altezze.

Si potrebbe cercare di sfruttare il software realizzato per l'analisi musicale, specie per quanto riguarda il riconoscimento della paternità di un brano. Al riguardo, si vedano i paragrafi 2.2 e 2.3.

Inoltre si potrebbe pensare di usare il sistema a più voci per modellizzare brani a una voce sola. Il sistema a più voci utilizzato su una voce sola dovrebbe consentire di rendere ancora più graduale la diminuzione di Ξ all'aumentare di σ , perché consente di utilizzare durate intermedie a quelle delle note; ad esempio, in un passo con molti sedicesimi analizzato con precisione di sessantaquattresimi, $\sigma = 4$ corrisponde a $\sigma = 1$ del sistema a voce sola. $\sigma = 6$ nel sistema a più voci corrisponderebbe a $\sigma = 1.5$ nel sistema a voce sola, che è evidentemente un paradosso, dovendo essere $\sigma \in \mathbb{N}$. Questo dovrebbe consentire in linea teorica di ottenere valori di Ξ altrimenti non ottenibili.

Un altro aspetto da evolvere è la costruzione formale, di cui al paragrafo 2.2. Si potrebbe cercare di implementare questi metodi, specie quello di Pachet, Roy e Barbieri [25], in modo da poter generare forme particolari e di poter utilizzare forme diverse per allenare il sistema. Sicuramente, essendo dei vincoli da imporre alla catena, questi metodi necessitano di ordini sufficientemente bassi, perché improbabile trovare la corrispondenza tra lunghe

sequenze di note e i vincoli imposti — che presumibilmente sarebbero lo stato finale, progressioni, temi etc.

È infine necessario testare il sistema su un grande numero di brani, ad esempio su un database estratto da IMSLP [14] come quello di ‘Peach Note’ [22]. Durante questa operazione è importante confrontare tutti gli algoritmi studiati: *standard method*, *note-by-note method*, doppia matrice, algoritmo per voce sola e per più voci e tutte le combinazioni fra queste.

Bibliografia

- [1] J. S. Bach. *14 canons BWV 1087/1078*. 1740 - 1750. URL: <http://gallica.bnf.fr/ark:/12148/btv1b550059626/f39.item> (cit. a p. 8).
- [2] P. Baldi. *Calcolo delle probabilità e statistica*. McGraw-Hill Companies, 1998 (cit. a p. 13).
- [3] A. Barbera. “Pythagoras”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/22603> (cit. a p. 2).
- [4] J. Cheng. *Virtual composer makes beautiful music*. arstechnica.com. 2009. URL: <http://arstechnica.com/science/2009/09/virtual-composer-makes-beautiful-musicand-stirs-controversy/> (cit. a p. 12).
- [5] C. M. Chierotti. “Comporre senza conoscere la musica”. In: *Nuova Rivista Musicale Italiana* Luglio/Settembre, 3 (1994). URL: <http://www.chierotti.net/kircher/articolo3> (cit. a p. 4).
- [6] C.-H. Chuan e E. Chew. “Generating and Evaluating Musical Harmonizations That Emulate Style”. In: *Computer Music Journal* 35.4 (2011), pp. 64–82. URL: http://www.mitpressjournals.org/doi/abs/10.1162/COMJ_a_00091 (cit. a p. 14).
- [7] M. Edwards. “Algorithmic Composition: Computational Thinking in Music”. In: *Communications of the ACM* 54.7 (2011). URL: <http://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext> (cit. alle pp. 8, 10–12).
- [8] D. M. Franz. “Markov Chains as Tools for Jazz Improvisation Analysis”. Master of Science thesis. Virginia Polytechnic Institute e State University, 1998. URL: <http://vtechworks.lib.vt.edu/handle/10919/36831> (cit. a p. 15).
- [9] C. Gallico. *L'età dell'Umanesimo e del Rinascimento*. v. 4. EDT, 1991 (cit. a p. 6).
- [10] E. Gamma. *Design Patterns: Elementi Per Il Riuso Di Software a Oggetti*. Pearson Education Italia, 2002 (cit. a p. 22).

BIBLIOGRAFIA

- [11] P. Griffiths. “Ligeti, György”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/16642> (cit. a p. 12).
- [12] P. Hoffmann. “Xenakis, Iannis”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/30654> (cit. a p. 11).
- [13] M. Iddon. *New Music at Darmstadt: Nono, Stockhausen, Cage, and Boulez*. Cambridge University Press, 2013 (cit. a p. 10).
- [14] *IMSLP — Petrucci Music Library*. URL: <http://imslp.org> (cit. a p. 35).
- [15] D. Koelle. *JFugue*. URL: <http://www.jfugue.org/> (cit. a p. 21).
- [16] D. Koelle. *The Complete Guide to JFugue*. 2008. URL: <http://www.jfugue.org/book.html> (cit. a p. 22).
- [17] Y.-W. Liu. *Markov chain algorithmic composition with musical form*. University of Minnesota, 2013. URL: <http://facultypages.morris.umn.edu/math/Ma4901/Sp2014/Prop/Reid-Ronnander-Proposal.pdf> (cit. a p. 15).
- [18] Y.-W. Liu. *Modeling Music as Markov Chains. Composer Identification*. CCRMA Stanford University, 2002. URL: http://esf.ccarh.org/254/254_LiteraturePack1/ComposerID_Liu.pdf (cit. alle pp. 14, 15).
- [19] P. Macey. “Josquin (Lebloitte dit) des Prez”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/1449> (cit. a p. 7).
- [20] J. A. Maurer. *A Brief History of Algorithmic Composition*. CCRMA, Stanford University. 1999. URL: <https://ccrma.stanford.edu/~blackrse/algorithm.html> (cit. alle pp. 8, 10, 11).
- [21] W. A. attr Mozart. *Musikalisches Würfelspiel, K.516f*. URL: http://imslp.org/wiki/File:PMLP47543-mozart_-_dice_waltz.pdf (cit. alle pp. 4, 5).
- [22] *Music Ngram Viewer*. URL: <http://www.peachnote.com/> (cit. a p. 35).
- [23] *OpenJDK*. URL: <http://openjdk.java.net/> (cit. a p. 22).
- [24] *OracleJDK*. URL: <http://www.oracle.com/technetwork/java/javase> (cit. a p. 22).
- [25] F. Pachet, P. Roy e G. Barbieri. “Finite-Length Markov Processes with Constraints”. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI*. 2011, pp. 635–642. URL: <http://cs1.sony.fr/bibdetails.php?reference=pachet:11b> (cit. alle pp. 15, 34).

BIBLIOGRAFIA

- [26] C. V. Palisca e D. Pesce. “Guido of Arezzo”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/11968> (cit. a p. 3).
- [27] A. E. Planchart. “Du Fay [Dufay; Du Fayt], Guillaume”. In: *Grove Music Online*. Oxford University Press. URL: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/08268> (cit. a p. 7).
- [28] M. Simoni. “The History and Philosophy of Algorithmic Composition”. In: *Algorithmic Composition: A Gentle Introduction to Music Composition Using Common LISP and Common Music*. Michigan Publishing, University of Michigan Library, 2003. URL: <http://dx.doi.org/10.3998/spobooks.bbv9810.0001.001> (cit. alle pp. 4, 8, 10, 11).
- [29] T. A. Smith. “That Crown of Thorns”. In: *BACH Journal* 28 (1997) (cit. a p. 9).
- [30] E. Surian. *Manuale di Storia della Musica*. Rugginenti Editore, 2006 (cit. alle pp. 1–3, 5, 7, 8).
- [31] D. Tymoczko. “The Geometry of Musical Chords”. In: *Science* 313.5783 (2006), pp. 72–74. DOI: 10.1126/science.1126287. eprint: <http://www.sciencemag.org/content/313/5783/72.full.pdf>. URL: <http://www.sciencemag.org/content/313/5783/72.abstract> (cit. alle pp. 20, 34).