

# 國立交通大學

工學院聲音與音樂創意科技碩士學位學程

## 碩士論文

節奏複雜度在自動作曲之應用與研究

A Study on Rhythm Complexity and Its  
Application to Automated Composition

研究生：吳明瑾

指導教授：黃志方 教授

成維華 教授

中華民國一百零一年七月

節奏複雜度在自動作曲之應用與研究

A Study on Rhythm Complexity and Its Application to  
Automated Composition

研究生：吳明瑾

Student : Ming-Jin Wu

指導教授：黃志方

Advisor : Chih- Fang Huang

成維華

Wei-Hua Chieng



國立交通大學  
工學院聲音與音樂創意科技碩士學位學程  
碩士論文  
1896  
A Thesis

Submitted to Master Program of Sound and Music Innovative Technologies  
College of Engineering  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of Master  
in

Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年七月

# 節奏複雜度在自動作曲之應用與研究

學生：吳明瑾

指導教授：黃志方

成維華

國立交通大學工學院聲音與音樂創意科技碩士學位學程

## 摘 要

本論文為節奏複雜度的分析研究，以求節奏複雜度在自動作曲上之應用，有別於一般自動作曲的探討，將重心放在節奏上。首先探討各種節奏複雜度定義，針對四種常見的節奏複雜度分析與比較，除了一般古典樂常見的節奏外，並延伸其定義在不同拍號以及不規則節奏上，以適應自由靈活的自動作曲，以實例分析四種複雜度的表現，找出最適合運用在自動作曲上的節奏複雜度定義。另外，在本文中提出了以節奏複雜度為基礎的節奏產生方法，以現有的節奏複雜度定義延伸，特別是針對不同拍號以及不規則節奏的產生。最後將節奏複雜度分析與節奏產生系統結合，實作出以節奏複雜度為基礎的自動作曲系統，將現有的樂曲在不更動音高的狀況下改編節奏，產生出單旋律的樂曲，並以實驗評估系統的效能，進一步討論節奏複雜度與音樂結構的關係，以及兩者之間的交互影響。希望透過本研究，讓自動作曲的節奏產生有更多的可能性，配合上其他演算法，使自動作曲更臻完備。

關鍵字：節奏複雜度、節奏產生、自動作曲、演算法作曲

# A Study on Rhythm Complexity and Its Application to Automated Composition

Student : Ming-Jin Wu

Advisor : Chih- Fang Huang

Wei-Hua Chieng

Master Program of Sound and Music Innovative Technologies

National Chiao Tung University

## Abstract

In this thesis, we provide a study on rhythm generator in order to apply it to automated composition. We focus the study on rhythm which is distinct from common study of automated composition. First, we discuss the various definition of rhythm complexity, and analyze and compare four common definitions of rhythm complexity in order to find the most suitable for automated composition. Besides the common rhythm in classical music, we take various time signatures and irrational rhythms as examples to compare four rhythm complexity definitions. Furthermore, we bring up an idea for rhythm generator based on rhythm complexity which can be applied for generating rhythm of various time signatures and irrational rhythm. Last but not least, we implement an automated composition system based on rhythm complexity which combines analysis of rhythm complexity and rhythm generator. Through the experiment, we verify the system and discuss the relationship between rhythm complexity and music structure. We hope this study can be applied to other algorithm in order to create a better system of automated composition.

Key words: rhythm complexity, rhythm generator, automated composition, algorithmic composition

## 誌謝

首先要感謝指導教授黃志方老師，感謝老師教導我們在電腦音樂領域上的專業知識，以及在研究上給予我們許多的指導與寶貴的建議，使我在這些年中獲益匪淺。再來感謝所有教導過我的老師，曾毓忠老師、董昭民老師、侯君昊老師等師長，從您們身上學到許多這個領域的相關知識，對於自己的研究也有相當大的幫助。

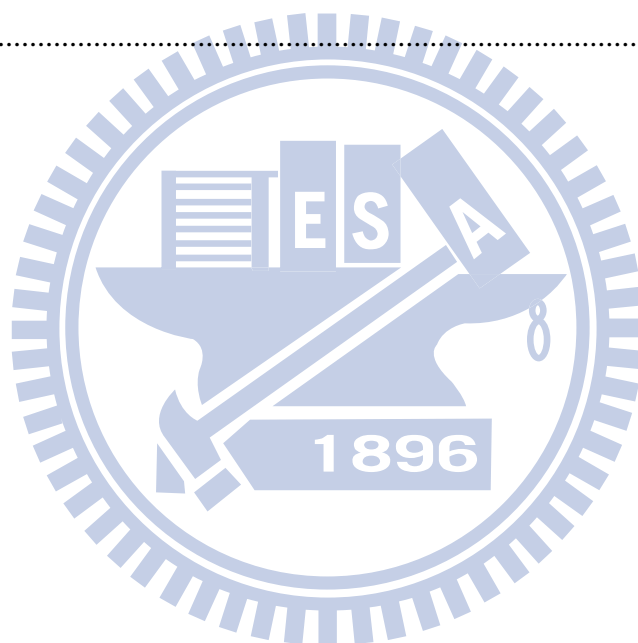
感謝實驗室的學長姐、同學以及學弟們。謝謝怡瑩學姊一直以來的關心與照顧，毅凡學長則是我們研究與學習上的好榜樣，謝謝學弟姚頭、小單、育賢、阿德、偉剛以及在口試前給予很多幫助的致暉，謝謝季芳、舒方、志翰、粘粘和囧豪，很開心在研究所的這個階段可以跟你們一起嬉鬧一起努力，遇見你們真的是一件很幸運的事。除此之外，也感謝在這段時間因為修課而認識的所有同學，不管是一起學習還是一起準備演出，這些日子從你們的身上學到很多。

最後感謝我的家人以及朋友們，沒有你們的包容與體諒，相信研究所的這些日子將會更加艱難。感謝朋友們這段時間的陪伴，讓我在研究之餘也能擁有豐富的生活。在準備研究所考試之前，就是家人告知我交大有這個學程，是家人的支持以及幫助讓我得以順利考取，也讓我順利完成研究所的學業。謹以此論文，獻給我最親愛的家人以及所有關心我的人。

# 目錄

摘要 .....	i
Abstract.....	ii
誌謝 .....	iii
目錄 .....	iv
圖目錄 .....	vi
表目錄 .....	viii
第一章、緒論 .....	1
1.1 研究動機 .....	1
1.2 背景介紹 .....	1
1.3 研究內容及貢獻 .....	2
1.4 論文架構 .....	3
第二章、相關研究 .....	4
2.1 背景知識 .....	4
2.1.1 節奏複雜度定義 .....	4
2.1.2 節奏表示法 .....	4
2.1.3 切分音 .....	6
2.1.4 MIDI .....	7
2.1.5 MusicXML .....	9
2.2 GTTM .....	10
2.2.1 Metrical Structure .....	11
2.2.2 Grouping Structure .....	12
2.3 節奏複雜度 .....	14
2.3.1 Metric Complexity .....	15
2.3.2 LHL Complexity .....	17
2.3.3 Keith's Complexity .....	20
2.3.4 Weighted Note-to-Beat Distance .....	22
第三章、研究方法 .....	25
3.1 節奏複雜度的分析與比較 .....	25

3.1.1 基本定義的分析與比較.....	25
3.1.2 延伸定義上的分析與比較.....	30
3.2 複雜度分析實作 .....	37
3.3 節奏產生方法 .....	39
3.4 系統架構 .....	42
第四章、實驗結果及討論 .....	44
4.1 實驗方法 .....	44
4.2 實驗結果 .....	46
4.3 延伸討論 .....	52
第五章、結論與未來展望 .....	59
參考文獻 .....	60



## 圖目錄

圖 1：以五線譜表示 the clave son .....	5
圖 2：以打擊樂器的樂譜表示 the clave son .....	5
圖 3：拿掉音符長度的表示法 .....	5
圖 4：box notation 以方框表示 the clave son .....	6
圖 5：以二元表示法表示 the clave son .....	6
圖 6：以 tied note 表示 the clave son .....	7
圖 7：切分音的實例 .....	7
圖 8：Hello, World in MusicXML .....	10
圖 9：Metrical hierarchy.....	12
圖 10：以 slur 在樂譜下方表示 grouping.....	13
圖 11：Grouping 群組關係 .....	13
圖 12：將小節切割成 16 個位置的 metrical hierarchy.....	15
圖 13：將小節切割成 12 個位置的 metrical hierarchy.....	16
圖 14：Metric Complexity 算法，以 the clave son 為例.....	17
圖 15：將小節切割成 16 個單位的 tree structure .....	18
圖 16：將小節切割成 12 個單位的 tree structure .....	19
圖 17：LHL 的切分音算法，以 the clave son 為例.....	20
圖 18：Keith 三種事件的定義 .....	20
圖 19：Keith 的切分音計算方法，以 the clasve son 為例.....	22
圖 20：WNBD 定義之與拍子的最短距離.....	23
圖 21：WNBD 計算方式，以 the clave son 為例 .....	24
圖 22：拍號 4/4，六個 onset，以十六分音符為基礎單位.....	26
圖 23：拍號 4/4，五個 onset，以八分音符為基礎單位.....	27
圖 24：拍號 3/4，五個 onset，以八分音符為基礎單位.....	28
圖 25：拍號 6/8，四個 onset，以八分音符為基礎單位.....	29
圖 26：6/8 與 3/4 重複交錯 .....	31
圖 27：Tchaikovsky's Symphony No. 6 in B minor, "Pathétique", op. 74 .....	31
圖 28：5/4 拍的節奏範例 .....	32
圖 29：將 5/4 拍的 metrical hierarchy 以 tree 表示 .....	32
圖 30：三連音的切割 .....	33
圖 31：以三連音取代原有十六分音符的權重 .....	34
圖 32：以新的權重定義分析不規則節奏 .....	34
圖 33：拍號 4/4，八個 onset.....	35
圖 34：複雜度分析流程圖 .....	38
圖 35：節奏產生流程圖 .....	40



圖 36：節奏產生方法範例 .....	41
圖 37：最簡單節奏複雜度不為 0 的例子 .....	42
圖 38：系統架構 .....	43
圖 39：小星星譜例 .....	45
圖 40：巴哈 G 大調小步舞曲譜例 .....	45
圖 41：編號 1 譜例（所有小節複雜度 1） .....	47
圖 42：編號 2 譜例（所有小節複雜度 6） .....	47
圖 43：編號 5 譜例（所有小節複雜度 0） .....	48
圖 44：編號 6 譜例（所有小節複雜度 0） .....	49
圖 45：編號 7 譜例（所有小節複雜度 0） .....	49
圖 46：編號 14 譜例（複雜度平均 1.25） .....	50
圖 47：編號 15 譜例（複雜度平均 1.25） .....	51
圖 48：編號 10 譜例（複雜度平均 0.625） .....	53
圖 49：編號 11 譜例（複雜度平均 1.0） .....	53
圖 50：編號 12 譜例（複雜度平均 1.0） .....	55
圖 51：編號 13 譜例（複雜度平均 5.0） .....	55
圖 52：編號 3 譜例（所有小節複雜度 6） .....	57
圖 53：編號 4 譜例（複雜度平均 1.83） .....	57



## 表目錄

表 1：Note-On 與 Note-Off 事件.....	8
表 2：Metrical Structure Rules.....	12
表 3：Grouping Structure Rules.....	14
表 4：以圖 22 為例之四種複雜度.....	26
表 5：以圖 23 為例之四種複雜度.....	27
表 6：以圖 24 為例之四種複雜度.....	28
表 7：以圖 25 為例之四種複雜度.....	29
表 8：以圖 28(b), (c) 為例之三種複雜度.....	32
表 9：以圖 28 為例之三種複雜度.....	35
表 10：以圖 33 為例之三種複雜度.....	36
表 11：WNBD 的最低複雜度，以 4/4 為例.....	37
表 12：系統中預設的 simple time signature.....	39
表 13：系統中預設的 compound time signature.....	39
表 14：受試者音樂背景.....	46
表 15：編號 1 及編號 2 複雜度較高之選擇.....	47
表 16：編號 1 與編號 2 複雜度比較.....	48
表 17：拍號辨識統計表.....	50
表 18：編號 14 及編號 15 複雜度較高之選擇.....	51
表 19：編號 14 與編號 15 複雜度比較.....	52
表 20：不規則節奏辨識統計表.....	52
表 21：編號 10 及編號 11 複雜度較高之選擇.....	54
表 22：編號 10 與編號 11 複雜度比較.....	54
表 23：編號 12 及編號 13 複雜度較高之選擇.....	56
表 24：編號 12 與編號 13 複雜度比較.....	56
表 25：編號 3 及編號 4 複雜度較高之選擇.....	57
表 26：編號 3 與編號 4 複雜度比較.....	58

# 第一章、緒論

## 1.1 研究動機

音樂包含節奏、曲調、和聲三大要素，然而要怎麼精確的定義「節奏」，卻不是一件容易的事，有人說節奏是拍子結構上的一種變化，有人說節奏包括了音樂中所有與時間相關的因素，而知名的音樂學家 Curt Sachs 曾經這樣形容「節奏」這個詞，「沒有廣泛被接受的意義」 [1]，雖然有各式各樣對於「節奏」這個詞的解釋，但是卻很難有明確而且容易讓所有人都理解的定義。

然而對於電腦科學，這確實是一個壞消息，一個精確的定義才有辦法將節奏以數值量化的方式表示在計算模型中，所以有人提出了節奏複雜度這樣的表示法。節奏複雜度以一個數值表示了節奏的不規則性，也就是說，所謂的複雜度高的節奏在人類的聽覺感知下，就是一連串不規則的聲響，而複雜度越低的節奏則越有規律性。

隨著資訊科技的發展，不僅改變了人類聆聽音樂的方式，人們捨去了黑膠唱片、錄音帶、CD 等音樂載體，使用數位音樂檔案在電腦上播放聆聽；另一方面，使用電腦創作音樂也漸漸的取代了原來的音樂創作形式，電腦可以記載樂譜進而編曲甚至演奏，電腦自動創作的需求也與日俱增。

經過研究，人類是擁有節奏創造的直覺本質 [2]，如果我們可以將節奏複雜度靈活地應用在電腦自動創作上，發揮人類在節奏創作的本質，勢必可以讓自動作曲變成更有趣的議題。

## 1.2 背景介紹

前面提過，音樂包含節奏、曲調、和聲三大要素，在電腦自動作曲自然也包含這三大要素。然而使用演算法產生旋律，雖已發展數十年之久，但多半是針對曲調以及和聲這兩部分為其主要發展，其中包括馬可夫鍊 [3] [4]，基因演算法發展出的 evolutionary 方法 [5]等，但卻較少看到以產生旋律的節奏為主的研究。

雖然節奏產生在電腦音樂裡面，並不算是一個新的主題，但在討論節奏產生時，多半是以純節奏性的音樂做為討論主題，像是非洲、南美洲等以打擊樂器為主要表現的民族音樂 [6]，卻較少人嘗試將其應用在旋律性的音樂上。

在 Povel 所提出的以樂理為基礎的旋律產生器 Melody Generator 系統中 [7]，雖然也提出了比較有架構性的方法，以時間為量度，配合拍子的階層性，將切割出來的單位組合而成為節奏，但是另一方面，為了配合調性，和聲以及音高走向都有可能使得節奏有所變動，因此並不是以節奏為出發點去做為自動作曲的基礎；加上此系統雖是可以藉由演算法產生出旋律，但主要還是必須靠使用者對於樂理的了解，去更動音樂的細節，以符合音樂創作本身的需求，不完全算是自動作曲的系統。

如果要在電腦實作出有規則的節奏產生系統，勢必得使用數學化的節奏分析去實踐，所以將節奏量化而成的節奏複雜度成為一個很好的選擇；將一長串的節奏化為一串數值的分析，又或者是相反地，將一串數值轉換成為一整段的節奏，只要透過了節奏複雜度，是有可能做到的。

以人類的聽覺感知來說，節奏是由聲響所交互形成的，所以如果要以節奏複雜度作為分析的基礎，將多聲部的音樂分別拆開來做分析，結果勢必不符合人類的感知，但要如何將多聲部的音樂綜合的來做分析，需要從更多不同的角度去切入人類的聽覺以及節奏的感知，需要了解在人類的聽覺之下，多聲部的音樂是如何被感知，以及在不同聲部之間是否會有交互影響作用，因此，我們的研究將分析著重在單聲部的樂曲上，以求達到更精確的結果。

### 1.3 研究內容及貢獻

本研究的初衷在於建立一套更完整的規則，透過對於現有節奏複雜度的分析與研究，改變現有的節奏複雜度，使其更廣泛的應用在自動作曲，以利於發展更完整的自動作曲的系統。

因此，研究的第一部分，為了將現有的節奏複雜度規則化，針對了目前現有的節奏複雜度，做了許多的分析與比較，除了要符合人類聽覺的感知，以音樂樂理為導向，更

要能精準且靈活地使用在自動作曲上。所以，除了將節奏複雜度應用在一般較常見、較規則的古典樂曲上，更要分別討論在不同的拍號，以及不規則節奏兩種狀況下，要如何調整現有的規則，讓節奏複雜度可以分析各種不同風格的樂曲，以方便進一步的應用在自由的自動作曲上。

而此研究的第二部分，則是將節奏複雜度的分析反推為節奏產生的方法，並實作節奏產生的方法在自動作曲上。以單旋律的音樂為分析的基礎，同樣產生出單旋律的樂曲，在不更動音高的狀況下，配合著原本音樂的結構，單獨改變了樂曲的節奏，也進一步針對音樂結構以及節奏的複雜度的關聯做了實驗以及探討。

#### 1.4 論文架構

本篇論文的第二章將介紹相關背景知識以及相關的文獻回顧；在第三章中，我們將說明研究方法，以及實作系統的架構；第四章將展示實驗的設計、實驗結果以及實驗評估；最後，在第五章總結我們的結果，以及討論未來研究可能的方向。



## 第二章、相關研究

### 2.1 背景知識

#### 2.1.1 節奏複雜度定義

早在 1905 年，心理學家 R. H. Stetson 提到節奏應該是「比較不規則、不純粹的」一連串的聲響 [8]，因此在早期的實驗與研究當中，節奏是代表著一種不規則性的概念，又或者也可以說是節奏複雜度的一種解釋。

之後 H. E. WEAVER 在 1939 年的研究中，他形容節奏的不規則性其實是音樂當中的切分音所造成的特性 [9]，切分音會在後面有更詳細的解釋；以切分音來定義節奏複雜度，可以說是最具有音樂導向的，以音樂以及樂理的層面來探討的研究也有提及 [10]，而在後面也會提及著名的 LHL 複雜度的定義 [11]，也是以切分音作為分析的最基本原則。

另外，以信息理論的亂度 Entropy 來定義節奏複雜度，也在 1960 年代的論文多被提及，在 1968 年 Paul Vitz [12] 以及 1969 年 Vitz & Todd [13] 的研究當中都可找到。而後，節奏複雜度在各個領域，以不同的角度去定義，在心理學的文獻當中，也有相當多節奏複雜度的測量與人類的感知的比較，從 1985 年 Povel & Essens [14]、1995 年 Peter Essens [15]、2000 年 Shmulevich & Povel [16]，一直到比較近期的 2007 年 Fitch & Rosenfeld 的研究當中 [17]，對於人類在節奏表現、節奏記憶等感知，與節奏複雜度的數值量化，做了許多的實驗以及討論，提出了很多正向相關的數據。

在我們的研究中，我們是希望將節奏複雜度應用在自動作曲上，因此在本篇論文中所討論到的節奏複雜度，皆是以音樂以及樂理的導向來解釋節奏複雜度，也就是以切分音去定義複雜度。

#### 2.1.2 節奏表示法

在此節中將展示四種節奏的表示方式，在此篇論文中，我們將以這四種方式來呈現；



前面兩種是以原本音樂的表示法，也就是以平常所見的樂譜上的形式來呈現，後面兩種則是在分析節奏是較常使用到的方式，去除了原本音符的長度，只留下音符出現的時間點。

第一種方式如圖 1 所示，也是音樂當中最熟悉、最常見的表現方式，這裡顯示的是 Afro-Cuban 節奏中著名的 the clave son。以五線譜的方式來表示節奏，只有一個小節拍號(time signature)為 4/4，因為加上了反覆記號，所以會重複一遍，最小的單位是十六分音符所以會將此小節切成十六等分。



圖 1：以五線譜表示 the clave son

第二種方式如圖 2 所示，通常是使用在打擊樂器上，同樣是樂譜的表示方式，比起圖 1 的五線譜，少了音高的表示，只是很單純的表示出節奏。但在打擊樂器中，樂器所發生的聲響只有一瞬間，實際上是無法表示出音符長度的，所以可以將原本音符的長度去掉，只留下發生聲音的那個時間點，其餘的位置由休止符補上，也就是我們在圖 3 所看到的樣子。而我們在討論節奏複雜度時，也是單純以聲音發生的點為代表，找出點跟點之間的關係來做計算。



圖 2：以打擊樂器的樂譜表示 the clave son

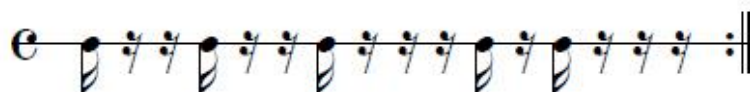


圖 3：拿掉音符長度的表示法

圖 4 中將小節切割成同等分後，以方框表示 the clave son 的節奏，這是由音樂學家 James Koetting 在研究非洲複節奏時所提出的表示法 box notation [18]，這樣的表示法取代了傳統樂譜，以圖像化的方式清楚地表現出節奏的模樣，比起樂譜更適合用在節奏複

雜度上，其中「x」標示著發出聲音的點，也就是音符開始的位置，而原本沒有聲音的位置，也就是休止符所在的點，則由「·」標示。

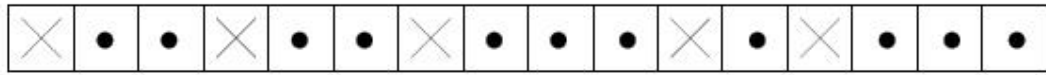


圖 4：box notation 以方框表示 the clave son

最後一個表示法如圖 5 所示，以電腦科學的方式來表示，在只有 0 跟 1 的電腦的世界，將先前的 box notation 轉換成為二元表示法，以 1 代替「x」標示著發出聲音的點，以 0 代替「·」標示休止符所在的點，這樣的表示法可以最直接的使用在電腦程式上，在此篇論文中，系統實作時就是以這樣的表示法在進行。

1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0

圖 5：以二元表示法表示 the clave son

### 2.1.3 切分音

關於切分音的觀念，在音樂學家之間有非常多樣的解釋 [19]，在 The Harvard Dictionary of Music 之中，對於切分音是這樣定義的為「與原有的拍子之間存在瞬間的矛盾」 [20]，也就是說切分音會使得聽者對於原本拍子的感覺造成混亂，當節奏與原本的拍子互相矛盾，也就是造成節奏在聽覺感知上複雜的原因。

而何謂與拍子之間存在有矛盾，Longuet-Higgins and Lee (LHL) 在 1984 針對節奏分析所提出的 context-free grammar [11] 中有提出一套完整的解釋；在 LHL 的理論當中，將所有休止符所在的位置，也就是沒有發聲的點，都以 tied note 代替，也就是一個依附在發聲位置的音符，雖然是音符，但是與前面的音符相連，所以並不會單獨發聲，只是延長被依附音符的音長，當然，在純節奏的觀念上，音長是可以被忽略的，所以 tied note 並不會改變節奏的樣貌。同樣以 the clave son 作為例子，圖 6 即是將休止符以 tied note 取代。



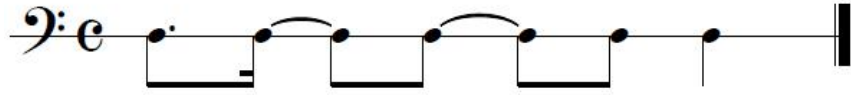


圖 6：以 tied note 表示 the clave son

而 LHL 所提出的切分音的定義，就是建立在 tied note 上面，當一個「較重」的音符依附在「較輕」的音符上時，切分音便產生了，也就是說，如果 tied note 比他前面發出聲響的音符還要重的時候，切分音就發生於此；而何謂音符的輕重，簡單地來說，每個音符根據它在小節裡面不同的位置都會有它各自的權重，這是建立在 metrical hierarchy 上，通常在拍子上的音符會重於在拍子以外的音符，所以說如果在拍子上的音符成為了 tied note，依附在非拍子上的音符，造成了聽覺上拍子的混亂，就是所謂的切分音了；而詳細關於 metrical hierarchy 的定義會在 2.2 GTTM 中說明，權重的定義則會分別在 2.3.1 Metric Complexity 以及 2.3.2 LHL Complexity 說明兩種不同的定義方式。

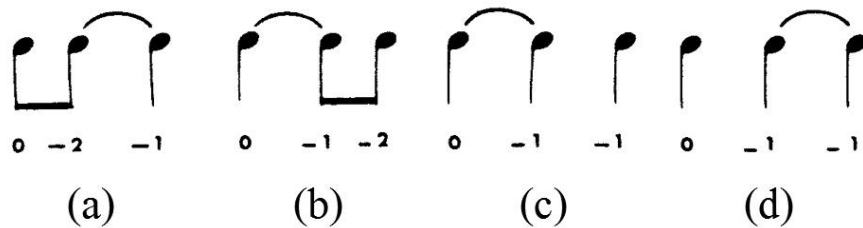


圖 7：切分音的實例

實際以譜為例來說明，在圖 7 中，(b)、(c)、(d)皆沒有切分音的存在，(b)和(c)的 tied note 權重小於被依附的音符，而(d)的 tied note 權重大小則跟被依附的音符是相同的，所以只有(a)是有切分音存在的；因為 tied note 權重為-1，比起前方權重為-2 的音符還要重，因此使得原本在聽覺上應該為拍子的音，在拍子之前就發出聲音了，進而造成了拍子上的矛盾，這就是所謂的切分音。

## 2.1.4 MIDI

現代數位音樂科技有兩種不同的音樂檔案類型，分別為 symbolic 和 acoustic 兩種。Acoustic 是將音樂以訊號方式記錄儲存，如 mp3, wav 檔等，其中取樣頻率會影響音樂的播放效果，雖然所包含的音樂資訊最為完整，但是音樂的內容相較之下卻不易取得；而

symbolic 則是將音樂以事件來記錄儲存，如果要產生音訊，則需要透過以硬體或軟體模擬樂器聲音來播放。

而在我們的系統中所使用的 MIDI<sup>1</sup>就是屬於 symbolic；MIDI (Musical Instrument Digital Interface) 格式是在 1983 年所提出，原是為了解決各種電子樂器、電腦還有其他演出設備之間的通訊問題所提出的通訊協定，而後 MIDI 成為一項關鍵的技術以用來紀錄音樂，MIDI 以事件儲存記錄音樂，使用 MIDI 的好處在於我們不需要透過繁雜的音訊處理技巧，就能夠擷取各種音樂參數。

以實例來說明，如果要從 acoustic 的音訊檔案中，擷取某個聲音的音高，我們需要計算頻率然後找出對應的音高，但過程中很有可能因為雜訊、泛音等原因而造成辨認失誤；但是在 MIDI 檔中，每一個聲音的產生所對應的都是訊息，裡面包含了音高、音長、音量等參數，所以我們可以直接從訊息中抓出數字來對應，如音高 60 等同於中央 C，換句話說，只要我們能夠正確讀到 MIDI 的訊息，就能直接找到我們所想要的參數。

以最常見的兩個 MIDI 事件為例：Note-On 以及 Note-Off，Note-On 事件代表音符開始發聲，Note-Off 則代表音符結束；兩個事件皆會記錄音高、力度、頻道等資訊，如下表所示。

表 1：Note-On 與 Note-Off 事件 [21]

	<i>Status Byte</i>	<i>Data Byte 1</i>	<i>Data Byte 2</i>
Description	Status/channel # (0-15)	Note # (0-127)	Attack Velocity (0-127)
Binary Data	(1001.0100)	(0100.0000)	(0101.1001)
Numeric Value	(Note on/CH#5)	(64)	(89)
Description	Status/channel # (0-15)	Note # (0-127)	Release Velocity (0-127)
Binary Data	(1000.0100)	(0100.0000)	(0001.1001)
Numeric Value	(Note off/CH#5)	(64)	(89)

<sup>1</sup> *MIDI Specification 1.0*, International MIDI Association, Los Angeles, 1983.

## 2.1.5 MusicXML

時至今日，MIDI 一直是 symbolic 音樂交換格式中最廣為使用的，但是 MIDI 最初的設計是為了克服在音樂表演上的通訊問題，很難直接表現樂譜上的較細微的音樂資訊，例如音樂中的小節、重複、連奏，還有音高因為以數字表示，因此也無法分辨 C 升半音和 D 降半音的區別，而 MusicXML<sup>2</sup>就是針對樂譜訊息作為交換格式而設計的，MusicXML 作為交換格式除了在樂譜軟體之外，還經常被使用在音樂分析、音樂資訊檢索，甚至音樂表演上。

MusicXML 是以成熟的 XML 技術標準為基礎，所以非常便利於網路交換，也因為結構清晰也方便閱讀，而且對於音樂訊息的記錄非常的完整，在 MusicXML 技術出現後，發展非常迅速。圖 8 是 MusicXML 的一個最簡單的例子，如同一般程式語言當中的 Hello, World [22]，裡面代表的是最簡單的音樂，一個樂器、一個小節、一個在中央 C 的全音符，根據各個節點的訊息亦可知：這段音樂的譜號為高音譜號也就是 G 譜號，拍號 4/4，C 大調，如同旁邊以五線譜記號所標示的。

在我們的系統中，分析的部分是以 MusicXML 為輸入的格式，以方便且準確的作為系統分析的依據，但在實驗中，仍是以 MIDI 作為輸出的格式，以方便受試者直接聆聽測試。

---

<sup>2</sup> MusicXML (Music Extensible Markup Language), <http://www.makemusic.com/musicxml>

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 0.5 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise>
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>

```



圖 8：Hello, World in MusicXML [22]

## 2.2 GTTM [23]

Lerdahl 和 Jackendoff 在 1983 年提出的 Generative Theory of Tonal Music (GTTM)，是基於人類的聽覺經驗，找出聆聽音樂時直覺上的意義，將調性音樂以規則、文法來具體描述，從音樂作品的表面，去尋找隱含於內的結構。

在 GTTM 裡，Lerdahl 和 Jackendoff 提出了四種模組，根據聽覺經驗者(experienced listeners)對於音樂作品的描述，定義了四種不同的結構，分別是 grouping structure, metrical structure, time-span tree 和 prolongational tree，在四種結構之中，都分別擁有兩種不同的準則：well-formedness rules 和 preference rules，其中 well-formedness rules 為必要條件，必須要完全符合，如果有兩個以上的結構都符合 well-formedness rules 時，則由 preference rules 來判斷何者為最合適的 [24]，四種結構的 preference rules 相互參照，才是完整的結構分析。

在本篇論文當中，只特別針對 metrical structure 以及 grouping structure，metrical structure 當中提出的 metrical hierarchy 會在 2.3.1 Metric Complexity 中使用到，而在我們的實驗部分，會運用 grouping structure，來做基礎的結構分析。

### 2.2.1 Metrical Structure

在討論 metrical structure 之前，我們必須先釐清重音的觀念，在 GTTM 當中提出，重音可以分為 phenomenal, structural, and metrical 三種，phenomenal 指的是任何音樂表面上特定加強的重音，例如說樂譜上所見的重音符號，structural 是因為旋律或和聲導致在樂句上所產生的重力點，特別是在終止式時會發生，而 metrical 才是指拍子之間的強弱，也是我們這裡所討論的，然而所謂 metrical 重音，其實是一種心理層面的結構，並不直接存在於音樂的表面，所以才建構出一個 metrical hierarchy，來描述節奏的形式。

在拍子的基本概念裡面，強拍與弱拍的交錯，是形成節奏最主要的原因，而因為拍子之間有強弱之別，所以勢必會存在一個 metrical hierarchy，也就是拍子之間階層的概念，而「強拍」與「階層」的關係在於，如果一個拍子被認為是強於某個特定的階層，那它會屬於一個較高的階層；以拍號 4/4 來說，第一拍跟第三拍，在一般的樂曲中，是被認為比第二拍和第四拍強的，所以第一拍跟第三拍會比第二拍跟第四拍高一個階層，而第一拍又比第三拍更強，所以第一拍又會比第三拍更高一個階層，如果以點做為表示方式則如圖 9(a)，以 poetic accent 來表示則如圖 9(b)，其中 - 代表強，~ 代表弱。

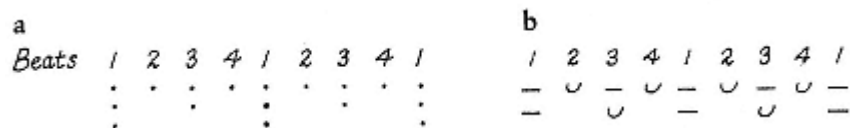


圖 9：Metrical hierarchy [23]

完整的 metrical structure 分析則需要針對 4 個 well-formedness rules 以及 10 個 preference rules，找出符合條件的結構，並與其他三個 preference rules 相互參照，才能達成完整的結構解析，rules 定義如表 2；在此篇論文中並無實作此部分的分析，因此不多作介紹。

表 2：Metrical Structure Rules

MWFR	Description	MPR	Description
MWFR 1	Every attack point must be associated with a beat at the smallest level of metrical structure.	MPR 1	Parallelism
		MPR 2	Strong beat early
		MPR 3	Pitch-event
MWFR 2	Every beat at a given level must also be a beat at all smaller levels.	MPR 4	Stress/Accent
		MPR 5	Length
MWFR 3	At each metrical level, strong beats are spaced either two or three beats apart.	MPR 6	Bass
		MPR 7	Cadence
		MPR 8	Suspension
MWFR 4	Each metrical level must consist of equally spaced beats.	MPR 9	Time-span interaction
		MPR 10	Binary regularity

## 2.2.2 Grouping Structure

所謂的 grouping 也就是分組的過程，其實是來自人類的本能，在面對一堆物品或者是一連串的項目之下，我們會自動地以某種方式幫對於這些物品或項目做分組或分類，而這樣的行為是否容易達成，則取決於這些輸入的資訊本身的組織，是否符合人類內在、無意識的分組原則；以音樂來說，對於聆聽者而言，輸入的資訊就是音高、發聲的時間點、音長、力度和音色等，當一個聆聽者在建構一首音樂作品的段落時，他必須很深刻



地意識到：這首音樂當中有哪些元素、哪些應該被歸類在一起而哪些不應該，而這些是在了解複雜的音樂架構當中最基本的部分。



圖 10：以 slur 在樂譜下方表示 grouping [23]

音樂 grouping 最基本的特點是在於聆聽時是有階層性的，動機是主題的一部份，主題是主題群的一部份，段落是整首作品的一部份，所以在 GTTM 裡提出了以連音符號 (slur) 為表示法，畫在樂譜下方如圖 10 所示，被另一個 slur 所圈住的符號就代表在聽覺上是一個群組屬於另一個更大的群組，以圖 11 的例子來說明， $p$  這個群組是屬於  $q$  這個更大的群組的一部份。



圖 11：Grouping 群組關係 [23]

在 grouping structure 中，有五項的 well-formedness rules，都是在敘述 group 的最基本的定義，例如 GWFR 4 指的是 group 之間不會有重疊的部分，如果有那勢必代表其中一個 group 是屬於另一個 group 的一部份，其他詳細定義如表 3 所示。

而 preference rules 則有七項，GPR1 是指應避免一個 group 裡面只有一個 event 存在，也就是通常一個 group 不會只有一個音符；GPR2 是指一連串的音符中，如果其中有兩個連續音符之間的時間差距較其他連續音符之間大的話，這兩個音符會被分為兩個 group；GPR3 是指如果有兩個連續音符，在音程、力度、演奏法、音長有了強烈的變化，則在這兩音符中間會是 group 的分界點；GPR4 是當 GPR2 與 GPR3 無法同時符合時，則會以更大的 group 取代；GPR5 對稱性是說在分段時，會以兩個同樣長度的段落做為切割的優先考量；GPR6 平行是說當有兩個以上的音樂平行而生的，在 groups 裡平行存在；GPR7 則是要參照 time-span 以及 prolongational 的結果，在我們的分析當中並沒有

使用，所以暫時不討論。

在 GPR 之中，GPR1、GPR2 以及 GPR3 是屬於比較小範圍的切割，通常在分段時是最先被考慮的，而 GPR4、GPR5 以及 GPR6 則是比較屬於大範圍的規則，通常是在前三個規則之後再對於大範圍的 group 進行分段切割。

表 3：Grouping Structure Rules

GWFR	Description	GPR	Description
GWFR1	Any contiguous sequence of pitch-events, drum beats, or the like can constitute a group, and only contiguous sequences can constitute a group.	GPR1	Avoid single event
		GPR2	Proximity: slur/rest and attack-point
GWFR 2	A piece constitutes a group.	GPR3	Change: register, dynamics, articulation, and length
GWFR 3	A group may contain smaller groups.		
GWFR 4	If a group G1 contains part of a group G2, it must contain all of G2.	GPR4	Intensification
		GPR5	Symmetry
GWFR 5	If a group G1 contains a smaller group G2, then G1 must be exhaustively partitioned into smaller groups.	GPR6	Parallelism
		GPR7	Time-span and prolongational stability

## 2.3 節奏複雜度

在眾多的節奏複雜度定義當中，我們選擇了四種作為主要的研究方向。其中由 Toussaint 在 2002 年所提出的 Metric Complexity [25]，以及 LHL [11] 的兩個複雜度的定義，皆是以 metrical weight 為基礎所發展的，也就是給予節奏權重，去尋找出複雜度的存在；另外由 Keith 在 1991 年以 metrical hierarchy 的階層為基礎 [26]，在沒有權重的狀況下，以類似樣式比對的方式去判斷切分音；最後是 Weighted Note-to-Beat Distance (WNBD)，由 F. Gómez 等四位在 2005 年提出 [27]，以音符跟拍子之間的最短距離作為切分音的判斷。

這四種節奏複雜度的定義，基本上都是以音樂上對切分音的角度為發想，在各式節奏複雜度的研究與討論中時常被提及，也被證實與人類的節奏感知較為類似 [28]，另外



在人類對於節奏的表現上也有相當的關連性 [29]。

### 2.3.1 Metric Complexity [25]

Metric Complexity 是由 Toussaint 所提出，藉由計算每一個發聲的點，也就是 onset 的權重的總和，以代表一個節奏。建立在 2.2.1 Metrical Structure 所提到的 metrical hierarchy，以階層的級數作為權重的數值。

以拍號 4/4 為例，當我們以十六分音符為基礎時，剛好可以將一個小節切割成十六個位置，依照一般古典音樂對於 4/4 拍的強弱拍定義，「強、弱、次強、弱」，如圖 12 所示，所以在這十六個位置中，最重要的是第一拍，也就是圖上橫軸 0 的位置，接下來是第三拍，也就是圖上橫軸 8 的位置，然後是第二拍跟第四拍，兩個擁有一樣的強度，依序往下推；現在我們把這樣的階層觀念延伸，建立一個通用的步驟。

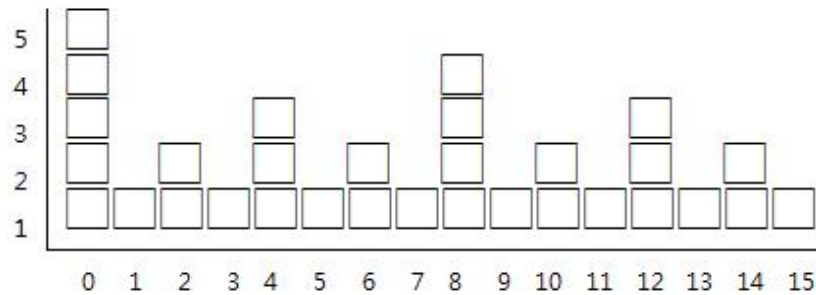


圖 12：將小節切割成 16 個位置的 metrical hierarchy

首先，假設我們將一個小節切割成  $n$  個位置，找出  $n$  所有的質因數，對  $n$  做質因數分解；如果以  $n=16$  為例，將 16 分解成  $2 \cdot 2 \cdot 2 \cdot 2$ ；假設  $w_i$  為權重，所有的權重一開始皆為 0， $i$  為位置，也就是說  $i = 0 \sim 15$ ， $m$  則對應到現在階層。

第一步，從  $m = 1$  開始，所有可以被  $m$  整除的位置權重加一， $w_i$  權重可表示為：

$$w_i = \begin{cases} w_i + 1, & \text{if } i \bmod m \text{ is } 0 \\ w_i, & \text{if } i \bmod m \text{ is not } 0 \end{cases} \quad (1)$$

第二步，把  $m$  乘上最後一個分解出來的質因數，也就是說此時  $m = 2$ ，一樣在所有可以被  $m$  整除的位置權重加一；第三步，把  $m$  乘上倒數第二個分解出來的質因數，也就是說此時  $m = 4$ ，一樣在所有可以被  $m$  整除的位置權重加一；第四步，把  $m$  乘上倒數第三

個分解出來的質因數，也就是說此時  $m = 8$ ，一樣在所有可以被  $m$  整除的位置權重加一；然後不斷重複上述的方法，直到所有質數都使用過，也就是  $m = 16$  時，即可完成 metrical hierarchy 的權重定義。

透過這樣的步驟，我們可以把原本建立在拍號 4/4 上面的 metrical hierarchy，也運用在其他拍號上，例如說在拍號 3/4 當中以十六音符為基本單位，則我們會把一個小節被切成十二等分，就會形成  $3 \cdot 2 \cdot 2$ ，在其他狀況下，十二等分也有可能會被分為  $2 \cdot 2 \cdot 3$  或者是  $2 \cdot 3 \cdot 2$ ，如圖 13 所示。

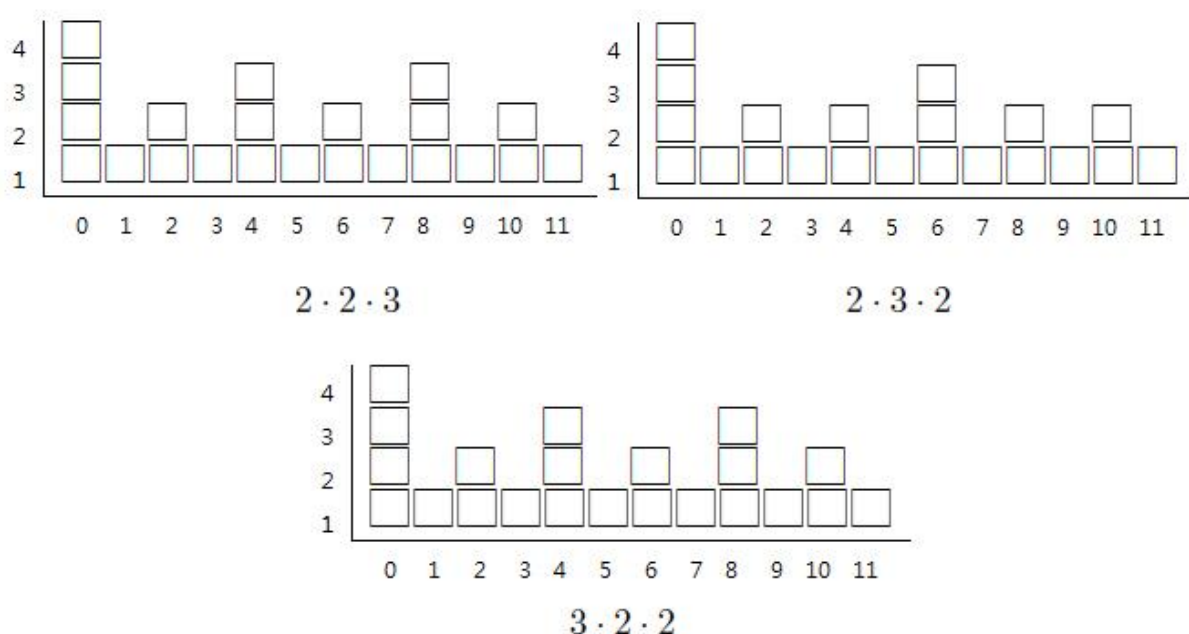


圖 13：將小節切割成 12 個位置的 metrical hierarchy

利用在 metrical hierarchy 上所建立的權重，Metric Complexity 的計算方式就是找出在節奏裡面每一個 onset 音符所對應的權重，算出它的總和；設  $k$  為 onset 的個數，可以如此表示：

$$M(R) = \sum_{i=0}^{k-1} w_i \quad (2)$$

以 the clave son 為例，如圖 14 所示，onset 分別在位置 0, 3, 6, 10, 12 上面，對應到的權重則為 5, 1, 2, 2, 3，加總起來就是 13。

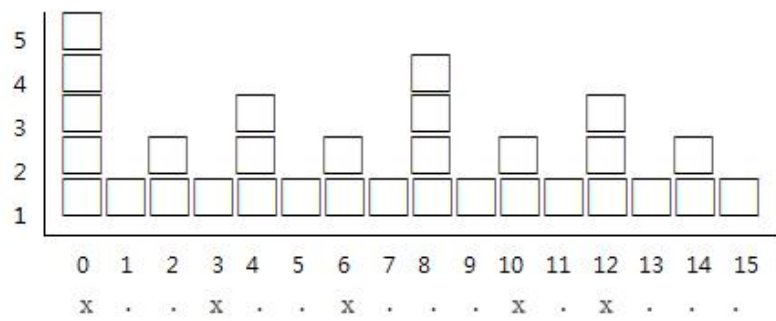


圖 14：Metric Complexity 算法，以 the clave son 為例

然而在 metrical hierarchy 的定義之下，權重越重，代表的是節奏的切分狀況越少，越接近原有的拍點，所以這個數值越大，反而代表的是越簡單的節奏。於是，為了讓數值的大小與複雜程度正向對應，Toussaint 在所有 onset 個數一樣的節奏裡面，找出最大值作為基準，與節奏所計算出的權重總和相減，也就是讓最簡單的節奏，複雜度為 0，同樣設  $k$  為 onset 個數， $maxw_0 \sim maxw_{k-1}$  為所有權重值裡面最大的  $k$  個，則 Metric 的複雜度可以如此表示：

$$Metric(R) = \sum_{i=0}^{k-1} maxw_i - M(R) \quad (3)$$

同樣以 the clave son 為例，權重總和為 13，但是在 onset 數量為五的所有節奏當中，總和的最大值為 17 ( $= 5 + 4 + 3 + 3 + 2$ )，那複雜度的值就是 4 ( $= 17 - 13$ )。

### 2.3.2 LHL Complexity [11]

LHL Complexity 是來自 Longuet-Higgins 以及 Lee 兩位在 1984 年所提出的一個針對音樂節奏的 context-free grammar，如同把節奏當成句子分析文法結構一般，與 metrical hierarchy 相同，給予每個節奏單位不同的等級；LHL 所提出的加權節奏層級，基本與先前 2.3.1 Metric Complexity 的想法是一樣的，差別在於 LHL 是以 tree 的架構來建立權重。

在近幾年，也有研究以 LHL 的基本定義為依據，將 tree 的架構以及複雜度的算法實作出來，分別有 Smith 以及 Honing 在 2006 年所提出，針對切分音的計算模型之評估與研究 [30]，和 Fitch 以及 Rosenfeld 在 2007 年所提出，切分音節奏的概念與產生 [17]，

兩個研究都有把 tree 的架構，實際以圖例展示出來；而此篇論文中 LHL 的 tree structure 圖表，是參考 Smith 以及 Honing 架構為基礎所畫製的。

以 4/4 拍號為例，當我們以十六音符作為基本單位時，每小節可切割成 16 個單位，如圖 15 所示；然而，與 Metric Complexity 不同的是，LHL 是將最重要的單位權重設為 0，依序往下減，以負數的形式來呈現，但其實階層的相對關係是一樣的。

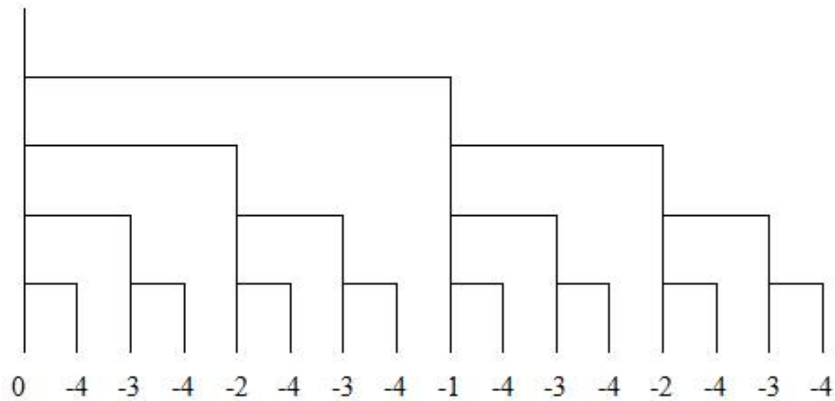


圖 15：將小節切割成 16 個單位的 tree structure

如果說要延伸應用至其他拍號，找出一個通用步驟的話，首先同樣必須以質因數分解的方式來做，以  $n=16$  為例，將 16 分解成  $2 \cdot 2 \cdot 2 \cdot 2$ ；tree 的 root 是一個完整的小節，從第一個分解出的質因數開始，將它一分為二，將最重要的單位權重設為 0，也就是小節中的第一拍，一分為二的中間位置，也就是小節中的第三拍，將上一個階層的數值減一，設為 -1。

第二步，找出第二個分解出的質因數，在這個例子當中，同樣也是 2，再將現在的分支一分為二，也就是小節中的第二拍以及第四拍，而權重依舊是上一個階層的數值減一，設為 -2；依序這樣下去，直到所有單位都有被分配到權重的值。

經由這樣的方式，我們就可以延伸到其他拍號，例如在拍號 6/8 以十六分音符為基本單位，一個小節會被切為 12 個單位，首先 6/8 拍子的強弱是「強、弱、弱、次強、弱、弱」，也就是說以 6/8 來說，會先將小節一分為兩，再來各分為三等份，把 12 質因數分解會是  $2 \cdot 3 \cdot 2$ ，在其他狀況下，十二等分也有可能會被分為  $2 \cdot 2 \cdot 3$  或者是  $2 \cdot 3 \cdot 2$ ，如圖 16 所示。

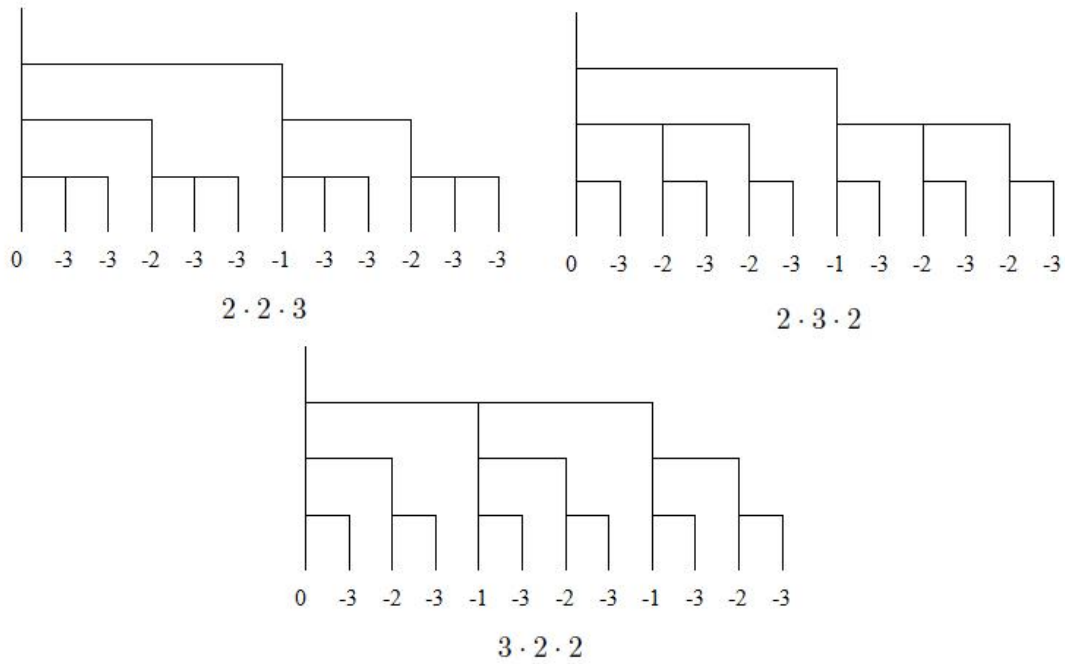


圖 16：將小節切割成 12 個單位的 tree structure

在 tree 建立好之後，如同在 2.1.3 所提到的，LHL 對於切分音的定義，當一個「較重」的音符依附在「較輕」的音符上時，切分音便產生了，因此，我們只需要針對每一個 onset，往後去尋找，在下一個 onset 出現前，是否有比它的大值，如果有就代表了切分音的出現，而每一個切分音的「強度」，即為  $S_i$  的權重與前面 onset 音符的權重的差，所有切分音的強度加總也就是複雜度的值；假設  $S_i$  為切分音出現的點， $k$  為切分音出現的個數，我們可以方程式來表示 LHL 的複雜度，如下所示：

$$S_i = w_s - w_{onset} \quad (4)$$

$$LHL(R) = \sum_{i=0}^{k-1} S_i \quad (5)$$

同樣以 the clave son 為例，如圖 17 所示， $S_0$  跟  $S_1$  皆是切分音，於是我們可以算出， $S_0$  為  $2 (= -2 - (-4))$ ， $S_1$  為  $2 (= -1 - (-3))$ ，在 the clave son 這個例子之下，LHL 的複雜度值為 4。

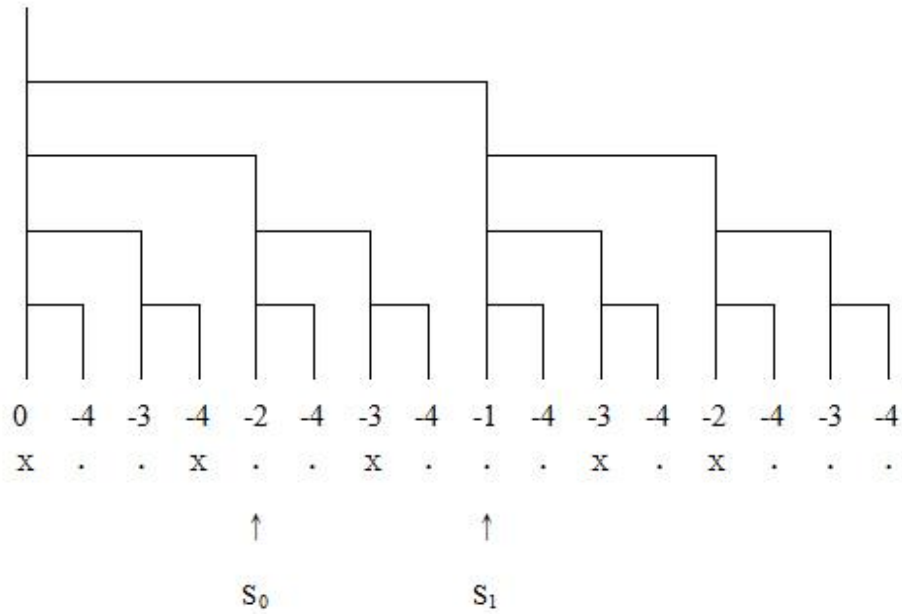


圖 17：LHL 的切分音算法，以 the clave son 為例

### 2.3.3 Keith's Complexity [26]

Keith 在 1991 年提出一個判別切分音的方法，建立在三種節奏事件上：hesitation、anticipation 和 syncopation，hesitation 是指開始在拍子上、結束不在拍子上的事件，anticipation 是指開始不在拍子上、結束在拍子上的事件，而 syncopation，也就是一般泛指的切分音形式，是指開始以及結束都不在拍子上的事件。Keith 就是以這三個類型的事件來做比對，然後計算出複雜度。

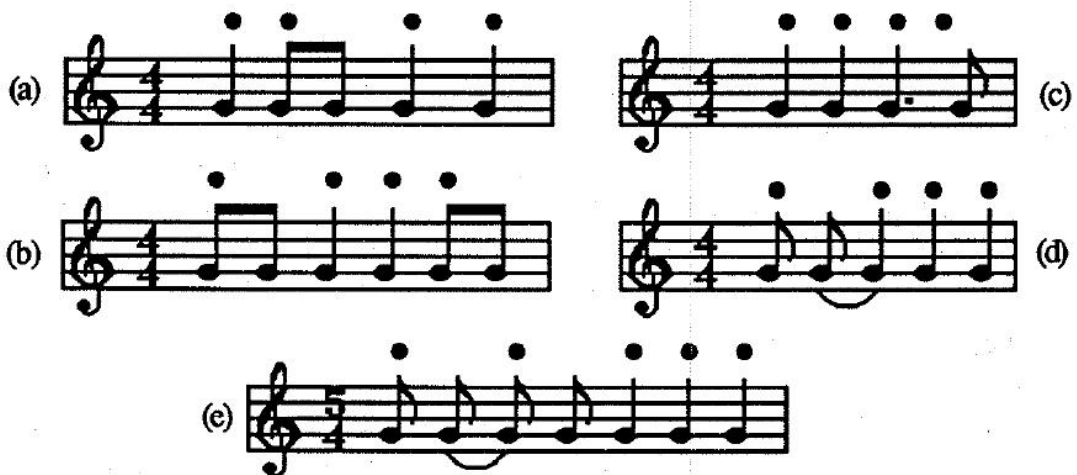


圖 18：Keith 三種事件的定義



如圖 18 所示，(a)和(b)兩個例子裡面都沒有以上三種事件的發生，所有的音符都是開始在拍子上，而且也結束在拍子上，而(c)的第三個 onset 音符開始在拍子上，但結束不在拍子上，代表的就是 hesitation，(d)的第二個 onset 音符結束在拍子上，但開始不在拍子上，所以是 anticipation，(e)的第二個 onset 音符開始與結束都不在拍子上，也就是 syncopation；其中，hesitation 的強度是小於 anticipation，anticipation 的強度小於 syncopation，而 syncopation 其實也就是 hesitation 與 anticipation 兩個事件同時發生時的事件。

以下我們以方程式詳細來說明 Keith 的複雜度定義，首先，針對每一個事件，假設  $i$  為 onset 的位置， $j$  為下一個 onset 出現的位置， $i$  和  $j$  分別代表的是一個事件開始跟結束的地方， $D$  是這個 onset 事件的長度，也就是  $D=j-i$ 。假設  $\delta$  為比  $D$  小且最靠近  $D$  的二的冪次方，而  $\delta$  就是判斷開始與結束是否在拍子上的依據，如果  $i$  或  $j$  可以被  $\delta$  整除，則代表是在拍子上，反之，如果無法被  $\delta$  整除，則代表不在拍子上；而 Keith 按照三個事件所造成的切分音的強度，分別將 hesitation 的值設為 1，anticipation 的值設為 2，syncopation 的值設為 3，如以下方程式所示：

$$\delta = D \text{ rounded down to the nearest power of 2} \quad (6)$$

$$s(x_i) = \begin{cases} 0, & \text{if } i \bmod \delta \text{ is } 0 \text{ and } j \bmod \delta \text{ is } 0 \\ 1, & \text{if } i \bmod \delta \text{ is } 0 \text{ and } j \bmod \delta \text{ is not } 0 \\ 2, & \text{if } i \bmod \delta \text{ is not } 0 \text{ and } j \bmod \delta \text{ is } 0 \\ 3, & \text{if } i \bmod \delta \text{ is not } 0 \text{ and } j \bmod \delta \text{ is not } 0 \end{cases} \quad (7)$$

以此方法，算出每一個 onset 事件的值，然後把所有事件的值加總，就是整個節奏的複雜度，設  $k$  為 onset 的個數，則我們可以將 Keith 的複雜度表示成：

$$Keith(R) = \sum_{i=0}^{k-1} s(x_i) \quad (8)$$

以 the clave son 為例，在拍號 4/4 以十六音符為基本單位，將一個小節切割成 16 份，onset 分別在 0, 3, 6, 10, 12 上面，如圖 19 所示。



圖 19：Keith 的切分音計算方法，以 the clave son 為例

第一個 onset 開始在 0 的位置，結束在 3 的位置，表示  $D$  為 3， $\delta$  為 2，開始的位置 0 可以被 2 整除，代表是開始在拍子上，結束的位置 3 無法被 2 整除，代表結束不在拍子上，也就是說這個 onset 是一個 hesitation 的事件，值為 1。第二個 onset 開始在 3 的位置，結束在 6 的位置，表示  $D$  為 3， $\delta$  為 2，開始的位置 3 無法被 2 整除，代表開始不在拍子上，結束的位置 6 可以被 2 整除，代表結束是在拍子上，也就是說這個 onset 是一個 anticipation 的事件，值為 2。第三個 onset 開始在 6 的位置，結束在 10 的位置，表示  $D$  為 4， $\delta$  為 4，開始的位置 6 無法被 4 整除，代表開始不在拍子上，結束的位置 10 也無法被 4 整除，代表結束同樣不在拍子上，也就是說這個 onset 是一個 syncopation 的事件，值為 3。

遵循同樣的步驟，我們可以算出第四個以及第五個 onset 值皆為 0；如此，依序找出所有 onset 計算出它的值，把所有值加總就是整個節奏複雜度，也就是代表 the clave son 這個節奏在 Keith 的複雜度算法下，複雜度的值為 6。

在 Keith 的複雜度算法之下，二的冪次方成為了一個關鍵，它限制住了這個演算法的範圍，只能針對一個小節切成二的冪次方的同樣等分時來計算，但是如果將一個小節切成非二的冪次方的等分時，這樣的算法很顯然地並不適用，所以 Keith 的複雜度很難應用在拍號 2/4、拍號 4/4 以外的情況。

### 2.3.4 Weighted Note-to-Beat Distance [27]

Weighted Note-to-Beat Distance (以下簡稱 WNBD) 是以「距離」作為切分音的計算方式，由 F. Gómez 等四位在 2005 年所提，為了打破之前幾種複雜度所擁有的限制而制定的新定義；而所謂的「距離」指的是 onset 與拍子之間的最短距離，然後再依照 onset 的類型，分別乘上不同的加權值，最後將所有 onset 事件的值加總除以事件的個數，就是 WNBD 所定義的複雜度數值。



如果以方程式來詳細描述 WNBD 的定義的話，首先，我們必須找出 onset 與拍子之間的最短距離，假設  $e_i$  和  $e_{i+1}$  為兩個連續的拍子，且任何兩個  $e_i$  和  $e_{i+1}$  之間的距離均設為 1，而  $x$  代表的是一個在  $e_i$  這個拍子上或者在這個拍子之後開始的音符，而 onset 與拍子之間的最短距離如下：

$$T(x) = \min\{d(x, e_i), d(x, e_{i+1})\} \quad (9)$$

以圖 20 為例，四個 onset 的  $T(x)$  值分別為， $\frac{1}{2}$ 、 $\frac{1}{4}$ 、 $\frac{1}{3}$ 、 $\frac{1}{3}$ 。

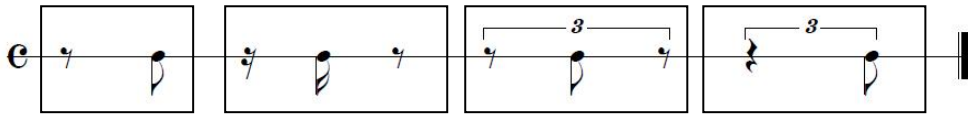


圖 20：WNBD 定義之與拍子的最短距離

在算出最短距離之後，依照 onset 結束的時間點分成三種狀況，第一種是「當  $x$  結束在  $e_{i+1}$  這個拍子之前，或是結束在這個拍子上」；第二種是「當  $x$  結束在  $e_{i+1}$  這個拍子之後，但是在下一個拍子  $e_{i+2}$  上，或是在  $e_{i+2}$  前」；最後是「當  $x$  結束在在  $e_{i+2}$  這個拍子之後」，三種不同的狀況分別給予不同的加乘，如以下的方程式所示：

$$D(x) = \begin{cases} \frac{1}{T(x)}, & \text{if } x \text{ ends before or on } e_{i+1} \\ \frac{2}{T(x)}, & \text{if } x \text{ ends after } e_{i+1} \text{ but before or on } e_{i+2} \\ \frac{1}{T(x)}, & \text{if } x \text{ ends after } e_{i+2} \\ 0, & \text{if } T(x) = 0 \end{cases} \quad (10)$$

$$WNBD(R) = \frac{1}{k} \sum_{i=0}^{k-1} D(x_i) \quad (11)$$

最後把所有 onset 的  $D(x)$  權重值加起來，除以 onset 的個數  $k$ ，就是 WNBD 的複雜度值。

關於  $D(x)$  權重值的不同，當「 $x$  結束在  $e_{i+1}$  這個拍子之後，但是在下一個拍子  $e_{i+2}$  上，或是在  $e_{i+2}$  前」發生時，因為  $x$  這個 onset 有發出聲音，但是在它之後的  $e_{i+1}$  這個拍子是沒有發出聲音的，這樣的狀況就是造成切分音的關鍵，所以才在這樣的狀況給予較高的權重。

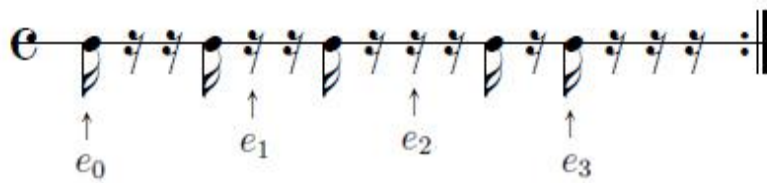


圖 21：WNBD 計算方式，以 the clave son 為例

再次以 the clave son 為例，我們以樂譜的方式表示，如圖 21 所示。第一個 onset 所在的位置，也就是第一個拍子  $e_0$  上，所以最短距離為 0， $D(x)$ 也等於 0；第二個 onset 所在的位置，以十六分音符為單位，與  $e_0$  間隔是 3，與  $e_1$  間隔是 1，因為兩個拍子之間的距離必須是 1，因此第二個 onset 與拍子的最短距離 0.25，因為結束在下一個拍子  $e_1$  之後、在  $e_2$  之前，所以  $D(x)$ 等於 8；第三個 onset 所在的位置，以十六分音符為單位，與  $e_1$  間隔是 2，與  $e_2$  間隔是 2，同樣因為兩個拍子之間的距離必須是 1，因此第三個 onset 與拍子的最短距離 0.5，因為結束在下一個拍子  $e_2$  之後、在  $e_3$  之前，所以  $D(x)$ 等於 4；第四個 onset 的位置與拍子的最短距離為 0.5，因為結束在下一個拍子  $e_3$  之前，所以  $D(x)$ 等於 2；第五個 onset 的位置也就是第四個拍子上，所以最短距離為 0， $D(x)$ 也等於 0。算出所有  $D(x)$  權重值之後，加總起來為 14，再除以 onset 的個數 5，就可以得到 WNBD 所定義的複雜度的數值 0.28。

WNBD 在這些複雜度演算法當中，是最能靈活運用在不同狀況的，因為不需要將一個小節切成同樣的等分，所以可以適應在不規則節奏，或者是不同的拍號上面。

## 第三章、研究方法

以下，我們將針對前面提及之四種節奏複雜度的定義，以實際的例子來做分析與比較，進一步找出適合自動作曲的複雜度定義，並訂立出實際的規則，運用在我們系統當中，以分析不同的節奏，然後以複雜度為基礎，說明產生節奏的方法。

### 3.1 節奏複雜度的分析與比較

#### 3.1.1 基本定義的分析與比較

在這一節當中，首先，我們以最常見也是最基礎的拍號 4/4 為例，分別以十六分音符和八分音符作為基礎單位，針對 onset 個數一樣的幾個節奏，將四個複雜度的定義做比較；因為 onset 個數會影響複雜度的計算，所以必須要以同樣的 onset 個數作為比較的基礎。

接著，我們以古典樂曲當中相當常見的拍號 3/4 為例，舞曲形式的華爾滋(waltz)、小步舞曲(minuet)和馬厝卡舞曲(mazurka)都是拍號 3/4，也有許多民族舞曲或者是民謠是以拍號 3/4 組成；在我們的範例中，拍號 3/4 以八分音符作為基礎單位，針對 onset 個數一樣的幾個節奏，將四個複雜度的定義做比較。

最後，我們以拍號 6/8 為例，在進行曲當中有部分是以 6/8 為拍號，還有船歌也經常是以 6/8 寫成，許多傳統的民族舞曲亦是；而 6/8 與 3/4 乍看之下兩個拍號似乎一樣，但其實在音樂上卻是截然不同的。

6/8 是所謂的複拍子(compound meter)，複拍子每拍可以分做三等分，所以在表示上一拍為一個複點音，而 6/8 是兩拍的複拍子(compound duple meter)，所以通常在表達 6/8 的強弱時，會是「強、弱、弱、次強、弱、弱」，但是在速度快的樂曲中，聽起來就是兩拍子的音樂，與三拍子(triple)「強、弱、弱」的 3/4 是截然不同的；在我們的範例中，拍號 6/8 以八分音符作為基礎單位，針對 onset 個數一樣的幾個節奏，將四個複雜度的定義做比較。

◆ 拍號 4/4

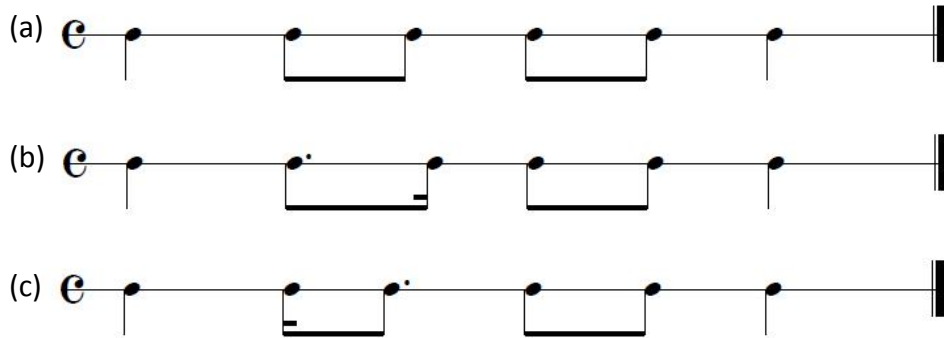


圖 22：拍號 4/4，六個 onset，以十六分音符為基礎單位

以圖 22 為例，(a) (b) (c)三種不同的節奏，在樂理上來說，(a)在三者之中是最簡單的節奏，接著是(b)，(b)出現了附點的節奏，也就是 Keith 所定義的 hesitation，但是按照先前切分音的定義，在(b)當中是沒有切分音的，最複雜的則是(c)，以複雜度的基本定義來說，(c)在第三個 onset 有切分音的出現，是 Keith 所定義的 anticipation；四種複雜度的值以及每一種複雜度的數值範圍(同一 onset 個數的最小值到最大值)如下表。

表 4：以圖 22 為例之四種複雜度

Complexity	Metric	LHL	Keith's	WNBD
(a)	0	0	0	$\frac{2}{3}$
(b)	1	0	1	1.5
(c)	1	1	2	1.5
數值範圍	0 ~ 13	0 ~ 13	0 ~ 18	$\frac{2}{3} \sim \frac{20}{3}$

在四個複雜度當中，只有 Keith's 很明確的分隔出三個節奏的不同，Metric 與 WNBD 對於(b)和(c)無法有所區隔，而 LHL 則是對於將(b)附點的狀況視為無切分音，所以無法與(a)區隔，但是因為在(c)當中，在切分音出現之後，後面的 onset 都在拍子上，所以以聽覺上來說，第三個 onset 對於拍子的混淆並沒有造成那麼大的影響。

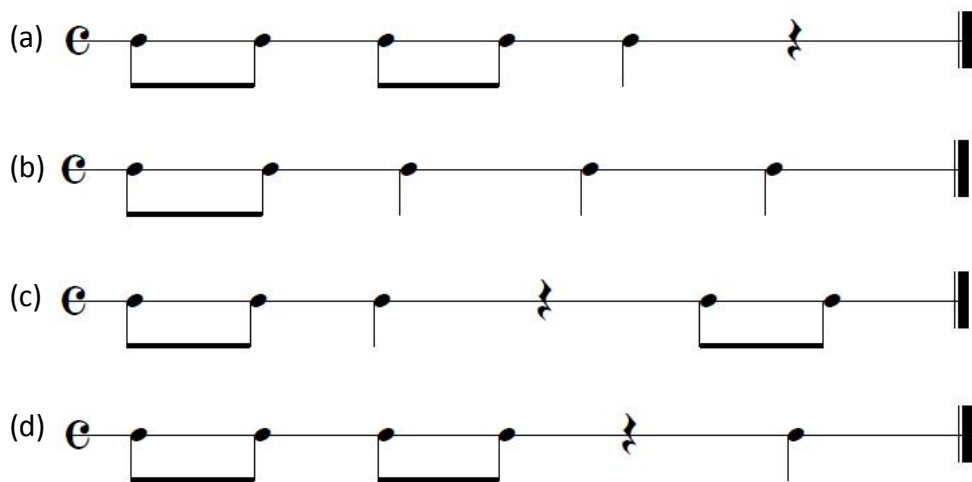


圖 23：拍號 4/4，五個 onset，以八分音符為基礎單位

接著以圖 23 為例，(c)和(d)的複雜度明顯比(a)和(b)要來的高，在 metrical hierarchy 當中第三拍是一個小節中，僅次於第一拍的重拍，因為(c)和(d)在第三拍當中沒有發出聲音，因此在聽覺上對拍子的混淆會有不小的作用，而(a)和(b)都是最基本的節奏，兩個的複雜度應該都接近 0；表 5 分別顯示了四種複雜度的值。

表 5：以圖 23 為例之四種複雜度

Complexity	Metric	LHL	Keith's	WNBD
(a)	1	0	0	0.8
(b)	0	0	0	0.4
(c)	2	1	3	0.8
(d)	2	2	2	1.2
數值範圍	0 ~ 12	0 ~ 12	0 ~ 15	0.4 ~ 3.2

在四個複雜度中，只有 WNBD 的(c)和(a)的值相同，無法明顯區別出兩者的差別，其他三個複雜度定義，則有效的區分了(c)(d)與另外兩個的差別，因為 WNBD 是建立在 onset 與拍子的最小距離上，所以如同(c)第三個 onset 本身在拍子上時，便無法分辨出切分音的存在。

有趣的是，LHL 跟 Keith's 雖然在(a)和(b)都很明確地顯示出了其節奏的低複雜度，

但是對於複雜度接近的(c)和(d)，卻有完全不同的判斷；以 Keith 所定義的複雜度角度來看，(c)是 syncopation 而(d)是 anticipation，以 LHL 的角度來看，使(c)和(d)形成切分音的 tied note 都是權重值為-1 的第三拍，但是因為依附在不同強度的音上，(c)被依附的音符權重為-2，而(d)被依附的音符權重為-3；雖然兩種複雜度同樣都判斷出了切分音的存在，可是對於切分音的強度有不同樣的解釋。

◆ 拍號 3/4

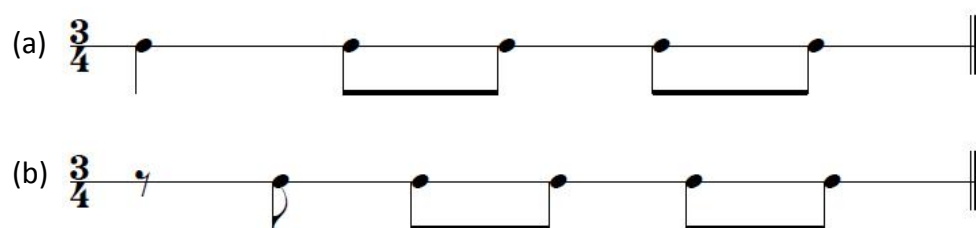


圖 24：拍號 3/4，五個 onset，以八分音符為基礎單位

以圖 24 為例，(a)為小步舞曲當中常見的節奏，是在「拍號 3/4，五個 onset，以八分音符為基礎單位」當中最簡單的其中一種節奏，而理論上來說，(b)應該是在同樣的狀況下複雜度最大的；四種複雜度的分析如下表。

表 6：以圖 24 為例之四種複雜度

Complexity	Metric	LHL	Keith's	WNBD
(a)	0	0	X (0)	0.8
(b)	2	2 (0)	X (2)	1.6 (1.2)
數值範圍	0 ~ 2	0 ~ 2	X	0.8 ~ 1.6

在分析四個複雜度之前，必須先針對 Keith 的複雜度做說明，因為 Keith's 的定義是以二的冪次方作為計算的關鍵，再來比對事件是屬於何種類型，所以當在 3/4 拍以八分音符為基礎單位時，一個小節會被切割為六等分，如果打破二的冪次方的算法，再來比對，可能會失去原本三種事件類型的意義。

在四個演算法當中，Metric 是最無爭議的，而在計算(b)的複雜度時，LHL 與 WNBD 必須參考下一個小節的第一個 onset，才能計算出正確的值，如果下一個小節重複(b)的



節奏，則兩者都可以反映出(a)與(b)複雜度的差別，但若是(b)的下一個小節節奏與(a)相同，在第一拍有 onset 的話，則(b)的複雜度明顯會降低。

◆ 拍號 6/8

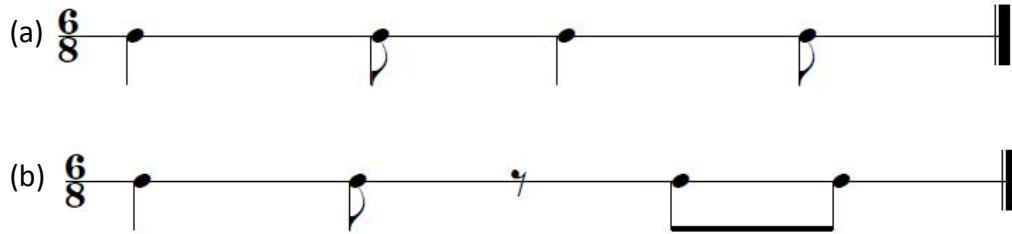


圖 25：拍號 6/8，四個 onset，以八分音符為基礎單位

以圖 25 為例，(b)在重要的拍子上少了發聲的點，理論上，(b)的複雜度應該要比(a)來得大，我們試者以四種複雜度分別計算，如下表；其中 WNBD 根據不同的定義，數值範圍不同，詳細定義差別會在後面說明。

表 7：以圖 25 為例之四種複雜度

Complexity	Metric	LHL	Keith's	WNBD
(a)	0	0	X (0)	1.5 (0)
(b)	1	1	X (0)	3 (0)
數值範圍	0 ~ 3	0 ~ 3	X	1.5 ~ 4.5 (0 ~ 0)

在 Metric 與 LHL 的計算方法之下，比較沒有爭議，而且也可以正確反映出複雜度的區別；我們試者將 Keith's 的二的冪次方打破，找出可以被計算的方法；我們依照 6/8 拍的結構，在計算  $\delta$  這個決定 onset 是否在拍子上的數值時，以 1, 3, 6 這樣的數列，取代二的冪次方，找出比  $D$  小但最靠近的值。

至於為何選擇 1, 3, 6 這樣的數列，是因為依照拍號 6/8 本身的結構，我們可以將 6/8 拆成  $3/8 + 3/8$ ，然後更進一步拆成  $(1/8 + 1/8 + 1/8) + (1/8 + 1/8 + 1/8)$  這樣的形式；然而計算出來的結果，還是無法反映出兩者複雜度的區別，因為第二個 onset 與第三個 onset 之間的距離，是在一個拍子的距離之內，可是沒有更小的解析度單位可以去做分析，因而造成了如此的結果；這樣的結果也證明了 Keith's 的算法上，確實是有所限制，無法

靈活運用在其他拍號上。

而依照 WNBD 的定義，onset 到 beat 的距離也就是拍子的最短距離，因為在 WNBD 的原文當中，並沒有針對 compound meter 多做說明，因此我們按照 compound meter 的原始定義，在 6/8 之中應該是只有兩個拍子，如果是兩拍，則可以反映出正確的複雜度結果，然而也有人以八分音符作為拍子的單位<sup>3</sup>，將 6/8 作為六拍的結構，但是這樣的結果顯然並不符合 compound meter 的複雜度。

在我們舉出的幾個例子當中，WNBD 相較之下，是比較無法區別出複雜度的高低，而且對於複拍子的定義不夠明確，而 Keith's 則是因為二的冪次方而多有限制。

### 3.1.2 延伸定義上的分析與比較

在針對比較基礎常見的幾種拍號分析比較了之後，我們將節奏複雜度的定義延伸到不規則拍號(irregular meter or complex time signature)以及不規則節奏(irregular rhythm, also irrational rhythm or tuplet)兩個面向，這些都是在古典音樂上比較少見的狀況，但在較無限制的現代音樂當中則經常出現，而我們為了讓節奏複雜度的定義可以靈活的在自動作曲當中被使用，所以在原本定義當中不被討論的這些狀況，我們將在此節中詳細討論與分析，以原本定義為基礎，延伸出適用於這兩個面向的方法；依照先前分析的結果，Keith's 因為二的冪次方的限制，所以無法運用至其他拍號，因此在此節當中，Keith's 將被排除在外，不做討論。

#### ◆ 在不同拍號的應用

Metric 與 LHL 兩個有相同的特性，只要能定義出每個單位的權重，就有辦法算出複雜度的值，所以如果要延伸到不同的拍號，只要能找出拍子的強弱關係定義清楚，在拍子之下的階層也都能定義出權重值，因此 Metric 和 LHL 基本上都是有辦法算出複雜度的。而 WNBD 本來就是為了靈活運用在各種狀況中所設計的，所以運用在各式各樣的拍號都是沒有問題的。

拍號大致上可以分成 simple, compound, complex, mixed 四種；simple 指的就是最基

<sup>3</sup> <http://www.cs.mcgill.ca/~cs644/Godfried/2005/Fall/pcastr/wnbd.html>





對第二小節與第四小節的節奏作分析比較。

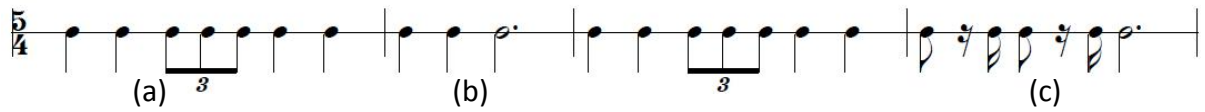


圖 28：5/4 拍的節奏範例

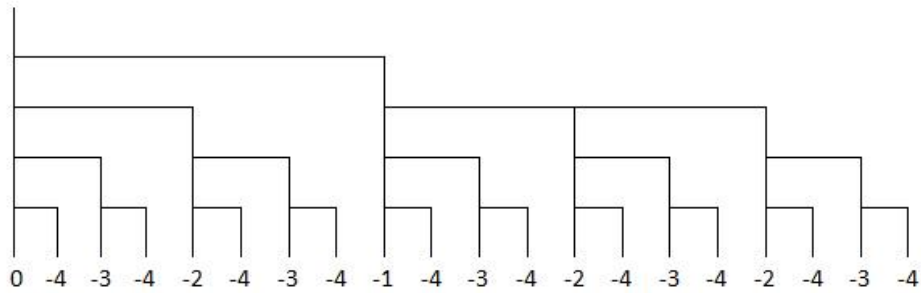


圖 29：將 5/4 拍的 metrical hierarchy 以 tree 表示

柴可夫斯基第六號交響曲的第二樂章所使用的 5/4，我們可以將它拆解為(2/4 + 3/4)，如果以權重表示之則為[0, -2, -1, -2, -2]，以 metrical hierarchy 來說，第一拍為最強的拍子，接下來是第三拍，最後才是其餘的拍子，然後再按照先前的權重給予的方式，將四分音符以下的每個點都給予權重，所有點的權重則會像是這樣：[0, -4, -3, -4, -2, -4, -3, -4, -1, -4, -3, -4, -2, -4, -3, -4, -2, -4, -3, -4]，如圖 29 所示，我們將以這樣的權重為基礎來分析 Metrical 與 LHL；因為兩者的 onset 數量不同，所以數值範圍不同，個別的结果以及數值範圍如表 8。

表 8：以圖 28(b), (c) 為例之三複雜度

Complexity	Metric	LHL	WNBD
(b)	0	0	0
數值範圍	0 ~ 9	0 ~ 9	0 ~ 8
(c)	4	0	1.6
數值範圍	0 ~ 13	0 ~ 13	0 ~ 8

在三種複雜度當中，LHL 無法表現出附點的複雜度，如之前所提，但是三種複雜度仍然都可以適應在拍號 5/4 的複雜度計算。

#### ◆ 在不規則節奏的應用

如同在圖 28 的例子當中看到的，在以十六分音符為基礎單位之下，三連音無法被基礎單位所解析成整數等分，所以三連音就成為了不規則節奏，三連音是不規則節奏最常見的一種，不規則節奏在節奏複雜度的計算造成了一個很大的問題，除了不需要權重的 WNBD 可以適用，其餘要把小節以同樣單位切成多等分的狀況下，不規則節奏成為了一個例外無法被處理，因此我們將以例外處理的方式來看待不規則節奏。

我們以圖 28(a) 的狀況來做說明，原本拍號 5/4 以十六分音符為基礎單位的權重會如同圖 29 一般，第三拍的三連音則成了例外，於是我們找出十二分音符(三連音的音符單位)跟十六分音符的最大公因數，以更小的單位將第三拍做切割，讓三連音的所有點都可以有權重，如圖 30 所示一般，箭頭所指的就是原本三連音的三個 onset 的點。

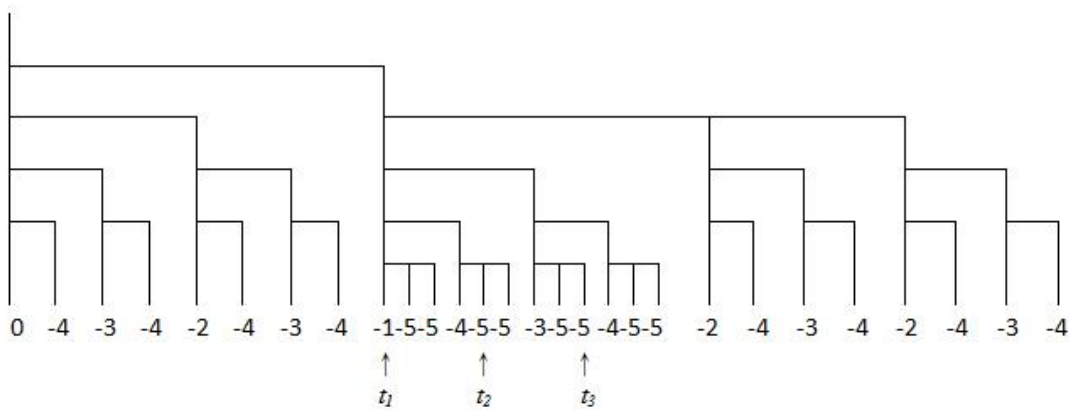


圖 30：三連音的切割

接著，我們把樂譜上沒有 onset 出現的點都拿掉，只留下三連音所在的位置，如圖 31 所示，以如此的步驟，針對所有不規則節奏我們都可以處理；我們將步驟簡化，當碰到一種不規則節奏時，就將其擺在原本 metrical hierarchy 最低階的下一階層，也就是找出最小的權重值  $w_{min}$ ，將不規則節奏當中，除了在拍子上本來就有權重的點以外的所有點權重設為  $w_{min} - 1$ ，如果出現兩種不規則節奏，譬如說同時出現三連音與五連音，則把三連音的權重值設為  $w_{min} - 1$ ，把五連音的權重值設為  $w_{min} - 2$ ，單一個拍子切成越多份的不規則節奏權重值越小。

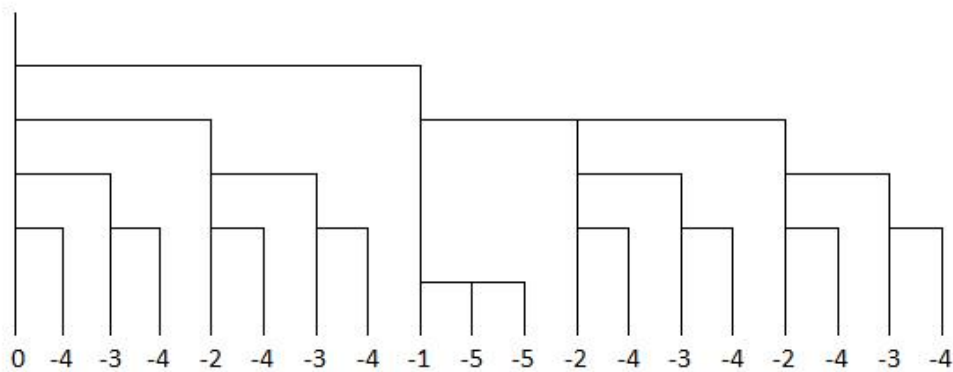


圖 31：以三連音取代原有十六分音符的權重

我們設定單一個拍子切成越多份的不規則節奏權重值越小，是因為以人對於同等份的切割處理，當數值越大時，越無法準確的去做切割，放在音樂上也是如此，演奏者在碰到不規則節奏時，在切割的等份較少時，如三連音、五連音，還有辦法精準的將節奏做時間上的切割，但當等份越來越多時，則是無法如此精確的切割，而是盡量快速的將所有的音符都在一個拍子裡面完成，所以等份越多，複雜度相對的應該會提高。

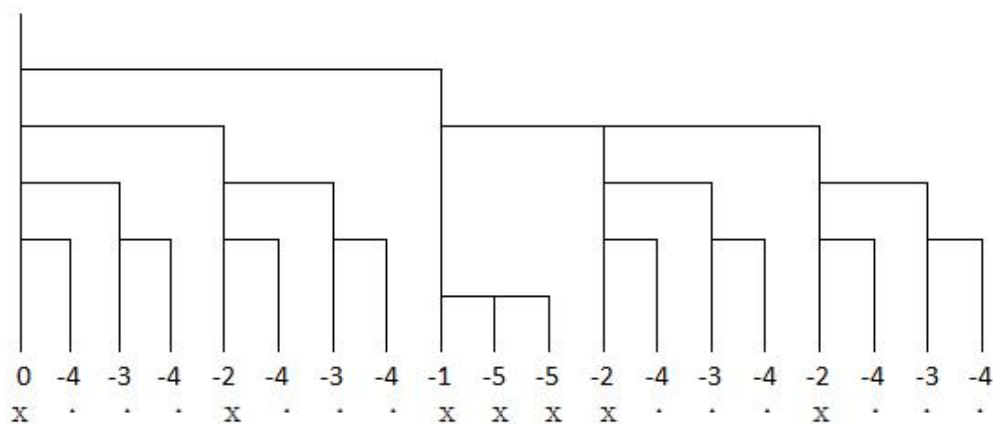


圖 32：以新的權重定義分析不規則節奏

以圖 31 的權重我們來看圖 28(a)的節奏，首先以 Metric 為例，如圖 32 所示，依照 Metric 複雜度的算法，所有權重加總之後的值為-17（或原 Metric 定義的 25，在這裡我們將 Metric 的權重值以 LHL 的負數值表示，因為在 Metric 的算法中是算出加總與最大值的差，所以以負數值運算並無影響），在這個節奏有 7 個 onset，然而要找出任何 7 個 onset 權重加總的最大值，則必須要從圖 29 中，尚未調整過的權重值尋找（因為調整過

的權重值會將原本可能選取到的較大的數值取代為不規則節奏中較小的數值)，而最大值為-13（或 29），所以 Metric 的值為 4。

表 9：以圖 28 為例之三複雜度

Complexity	Metric	LHL	WNBD
(a)	4	0 (3)	$\frac{6}{7}$
數值範圍	0 ~ 17	0 ~ 17	$\frac{4}{7} \sim \frac{43}{7}$
(b)	0	0	0
數值範圍	0 ~ 9	0 ~ 9	0 ~ 8
(c)	4	0	1.6
數值範圍	0 ~ 13	0 ~ 13	0 ~ 8

接著以 LHL 為例，如果就圖 32 的權重來看，是沒有切分音的，複雜度為 0，但在兩個權重-5 的 onset 後面，都是有可能會有比其更大的權重值，也就是必須以圖 30 來做計算，在  $t_2$  和  $t_3$  之後分別有-3 與-4，所以我們也可以說 LHL 的值為 3。

最後不需要權重的 WNBD，我們可以算出複雜度為  $\frac{6}{7}$ ，但有不規則節奏的節奏(a)與沒有不規則節奏的節奏(c)相比，複雜度卻反而比較低；三種複雜度結果如表 9 所示。

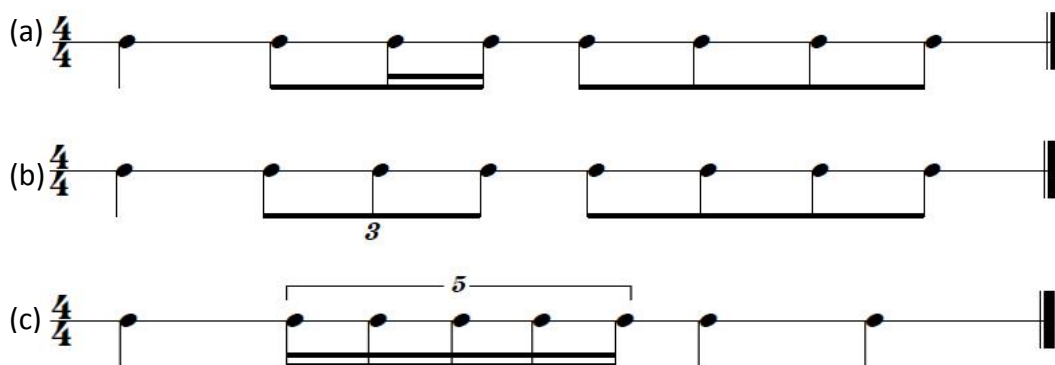


圖 33：拍號 4/4，八個 onset

我們再以圖 33 為例，針對不規則節奏做更多的比較，這兩種節奏當中，(a)為最基本的節奏；而(b)則有三連音，其中 WNBD 對於兩個複雜度沒有辦法區別，而 LHL 如果以原本切分音的定義也是無法區別，但若以權重值來算則為 1 或 3（八分音符或 16 分音

符為單位的權重不同)，Metric 以八分音符為基礎單位時複雜度為 2，以十六分音符基礎單位則為 4；(c)當中有五連音，WNBD 複雜度有小幅度的增加，而 LHL 如果以原本切分音的定義依舊無法區別，但若以權重值來算則為 4，Metric 以八分音符為基礎單位時複雜度為 4，以十六分音符基礎單位則為 8；LHL 與 Metric 雖然在不同單位之下會有不同的值，但都還是有辦法區別不規則節奏的複雜度；另外數值範圍則因為(a)沒有不規則節奏，而(b)和(c)各自擁有不同的不規則節奏導致三者的數值範圍有所不同，其中(b)和(c)都分別列出以八分音符以及十六分音符(如括號中所示)為基礎單位的不同數值範圍，所有的複雜度值如下表。

表 10：以圖 33 為例之三種複雜度

Complexity	Metric	LHL	WNBD
(a)	0	0	1.25
數值範圍	0 ~ 15	0 ~ 15	1 ~ 6
(b)	2 (4)	0 (1or3)	1.25
數值範圍	0 ~ 9 (0 ~ 17)	0 ~ 9 (0 ~ 17)	1 ~ 6
(c)	4 (8)	0 (4)	1.875
數值範圍	0 ~ 11 (0 ~ 19)	0 ~ 11 (0 ~ 19)	1 ~ 7.5

對於四種複雜度的定義，我們以實例做了比較，其中 Keith's 因為二的冪次方而多有限制，因此不適合用於自動作曲上；WNBD 雖然靈活，可是其定義卻無法分別許多節奏的複雜度高低，加上 WNBD 在不同的 onset 個數時，最低的複雜度不盡相同，如表 11，使得其複雜度的數值在不同的 onset 個數時會有不同的標準，如果要將不同 onset 個數的最低複雜度標準化，則會使得數值過於複雜；LHL 雖然有少部分節奏無法區別，但在各種狀況下都可以運用，唯一比較有爭議的在於如果存在跨越小節的切分音，對於前後小節的複雜度都會有影響，然而 LHL 本身的定義，是建立在每一個小節的第一拍都有 onset 的假設之下，尤其是當拍號如果交替產生時，出現了跨越小節的切分音，就必須要另外對於這個部分重新定義與分析；而 Metric 雖然亦有少部分節奏無法區別，但在各種狀況下也都可以運用，而且可以對於每個小節獨立做運算，就算是小節之間有不



同的拍號，也還是可以處理，於是我們決定以 Metric 作為本研究的複雜度分析實作以及節奏產生的基礎。

表 11：WNBD 的最低複雜度，以 4/4 為例

Onset 個數	1	2	3	4	5	6	7	8	9	10	11	12
最低複雜度	0	0	0	0	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{6}{7}$	1	$\frac{4}{3}$	$\frac{8}{5}$	$\frac{20}{11}$	2

### 3.2 複雜度分析實作

在針對四種複雜度做了許多不同的分析與比較了之後，我們將原來 Metric 對於複雜度的定義，加上前面所提及應用在不同拍號以及不規則節奏的延伸定義，實作在系統當中，以 MusicXML 作為輸入的格式，依照拍號把每個小節切割出來，再按照指定的基礎單位，建立所有點的權重，因為 Metric 必須以同樣 onset 個數的權重總和最大值作為基準相減，所以分別必須算出每個小節的 onset 個數，再按照先前建立好的權重找出最大總和，如果沒有不規則節奏，則按照此權重算出每個小節的複雜度，如果在節奏當中發現不規則節奏，則調整權重，依照調整後的權重算出複雜度，複雜度分析部分的流程圖如圖 34。

拍號上方的數字代表的是一個小節有幾拍，稱為 beat count，在此系統當中可以接受所有整數的 beat count，拍號下方的數字代表的是音符時值，也就是以何種單位當作一拍，稱為 beat unit，在此系統當中只能接受二的冪次方為音符時值，雖然也是有以非二的冪次方作為 beat unit 的音樂作品，但其實這樣的拍號與音樂節奏，是可以調整而成以二的冪次方作為 beat unit 的樣子，所以我們沒有實作非二的冪次方作為 beat unit 的拍號在此系統當中。

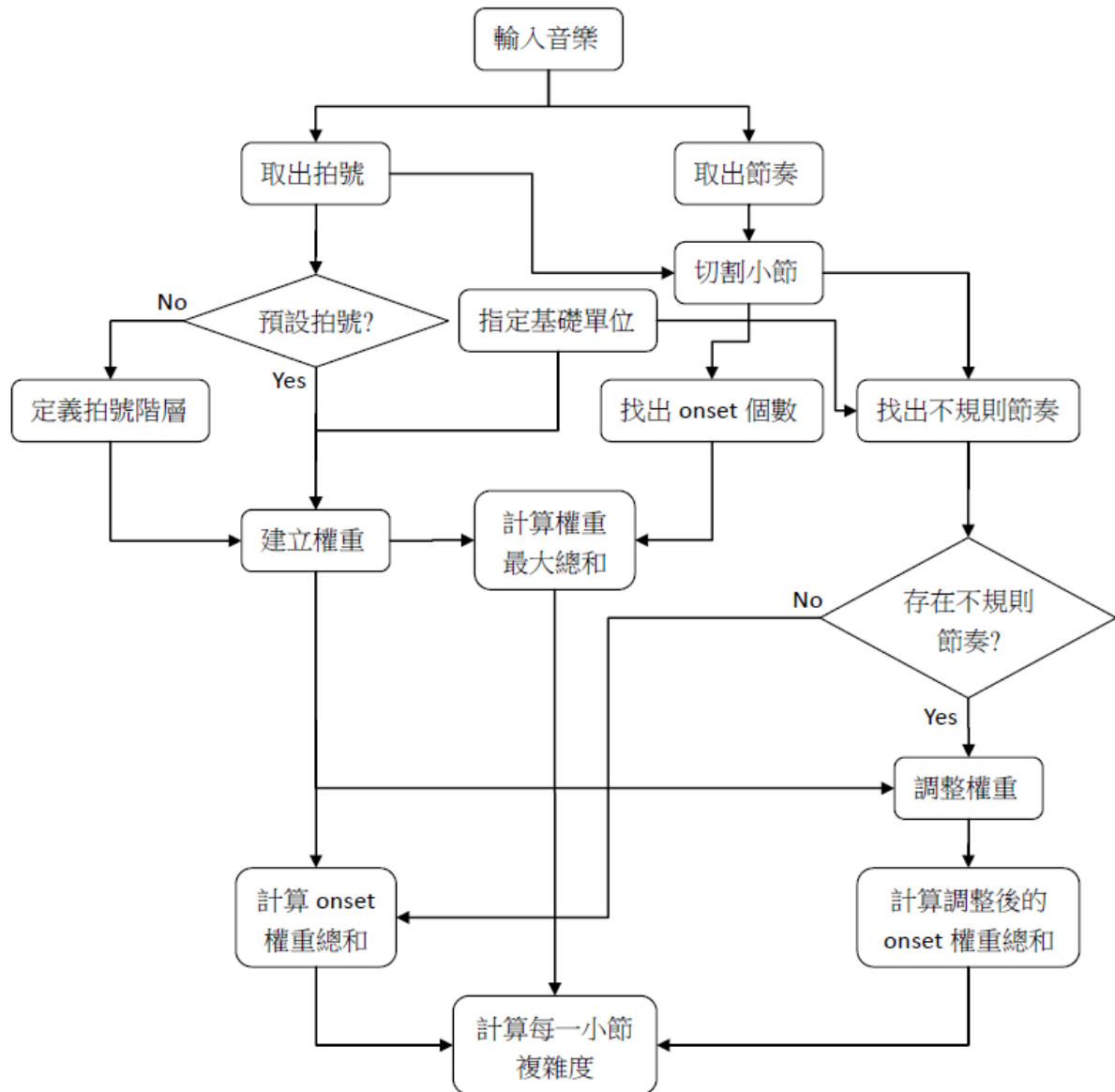


圖 34：複雜度分析流程圖

其中預設的拍號有六種 beat count，我們將 simple 以及 compound 兩種拍號的 metrical hierarchy 定義在程式裡面，如表 12 和表 13 所示，如果使用者輸入這六種以外的拍號，則會要求使用者自行定義 metrical hierarchy，如果使用者希望將預設的拍號重新定義，也可以依照輸入的 metrical hierarchy 去建立權重。

表 12：系統中預設的 simple time signature

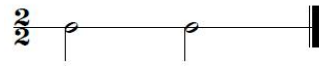





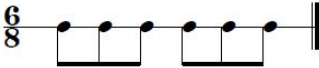
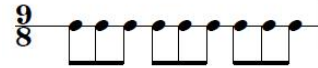

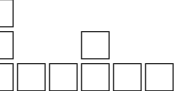
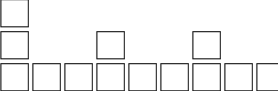

2 (duple)	3 (triple)	4 (quadruple)
		
		
[0, -1]	[0, -1, -1]	[0, -2, -1, -2]

表 13：系統中預設的 compound time signature

6 (duple)	9 (triple)	12 (quadruple)
		
		
[0, -2, -2, -1, -2, -2]	[0, -2, -2, -1, -2, -2, -1, -2, -2]	[0, -3, -3, -2, -3, -3, -1, -3, -3, -2, -3, -3]

在此系統當中，所有的權重都以 0 為最大值作為最高階層的數值，以下的階層則以負數表示，以方便在出現不規則節奏時，找到最小階層的權重值往下減一，如果依照原來 Metric 的正數階層，在碰到不規則節奏需要設定下一階層的權重值的時候，則會需要將所有權重值往上加一，因為 Metric 是算出與最簡單節奏也就是權重總和最大值的差，所以只要階層之間的相對位置沒有被改變，正數負數並不影響結果。

### 3.3 節奏產生方法

此系統除了將樂曲分析節奏複雜度以外，也實作了以節奏複雜度為基礎的節奏產生方法，將一個複雜度的數值反過來產生出一小節的節奏，以 Metric 複雜度定義為基礎，

讓使用者指定拍號、基礎單位以建立權重，指定 onset 個數、輸入複雜度，建立出最簡單節奏也就是權重總和最高、複雜度最低的節奏，然後以移動 onset 使其複雜度增加，直到符合指定的複雜度，另外也可選擇加入不規則節奏，提高複雜度，流程圖如圖 35。

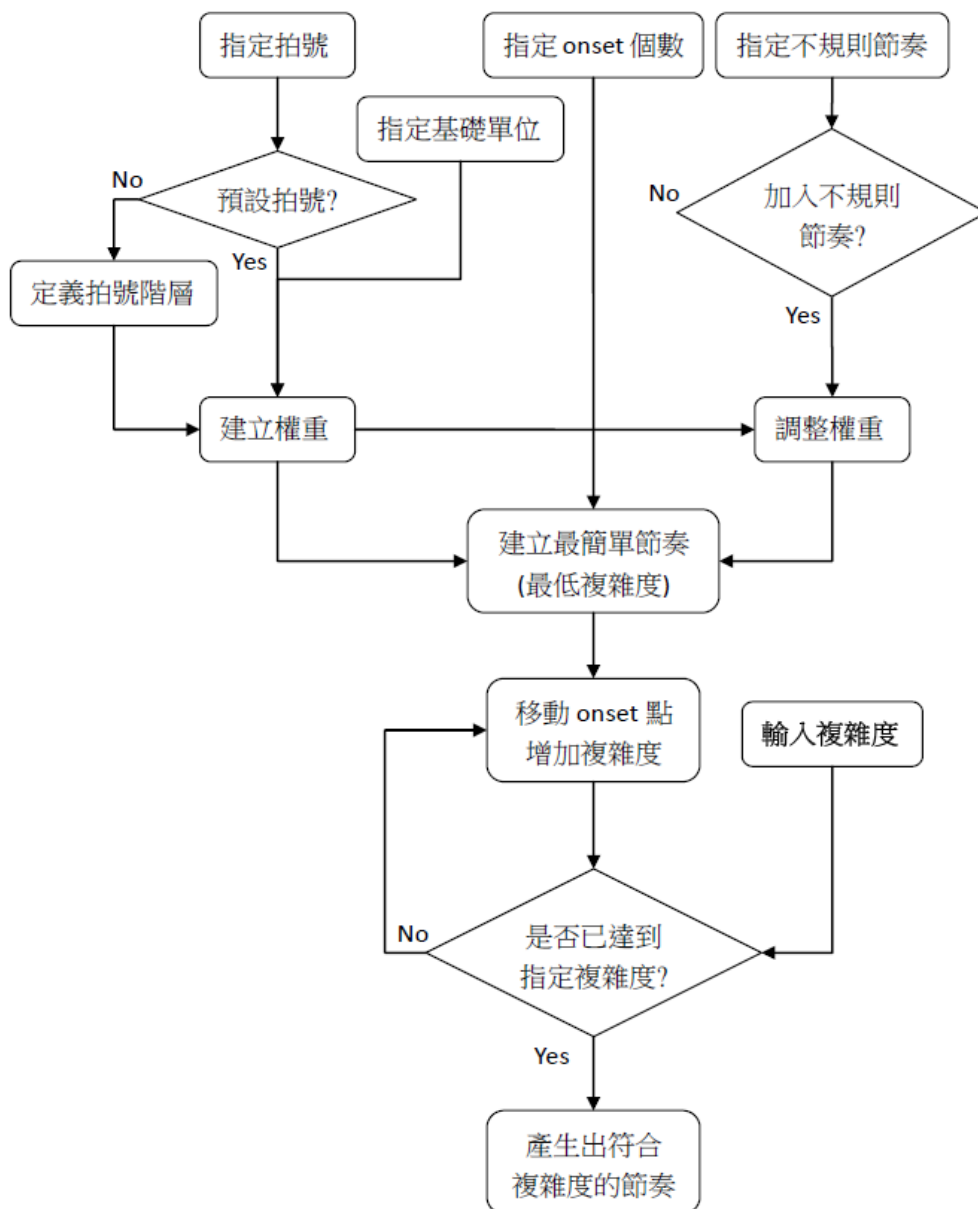


圖 35：節奏產生流程圖

節奏產生的詳細步驟如圖 36 所示，假設我們要產生出一個拍號 4/4、基礎單位為十六分音符、小節裡面一個有六個 onset 的節奏，第一步，首先我們找出權重值當中六個最大的數值，第一個一定是 0 所在的位置也就是第一拍，第二個是 -1 所在的位置也就是第三拍，然後第三個與第四個則是第二拍與第四拍，權重為 -2 的兩個位置，接著剩下兩

個，我們從下一層-3 當中隨機選出兩個點，於是將 onset 都放上這些點的位置，成為了複雜度為 0 的最簡單節奏。

建立了最簡單的節奏之後，再依照輸入的複雜度，去移動 onset 的點，符合我們所指定的複雜度值，以複雜度 3 為例子，首先，我們隨機指定一個 onset 的點，假設我們找到了權重-3 的這個位置，如圖左邊箭頭的起點，然後移動到-4 這個位置，如圖左邊箭頭的終點，我們的複雜度就變成了 1，然後我們再隨機指定一個 onset 的點，假設我們找到了權重-2 的這個位置，如圖右邊箭頭的起點，然後移動到同樣是-4 的另一個位置，如圖右邊箭頭的終點，複雜度就變成了 3，如此一來，就可產生出符合我們所指定的複雜度的節奏。

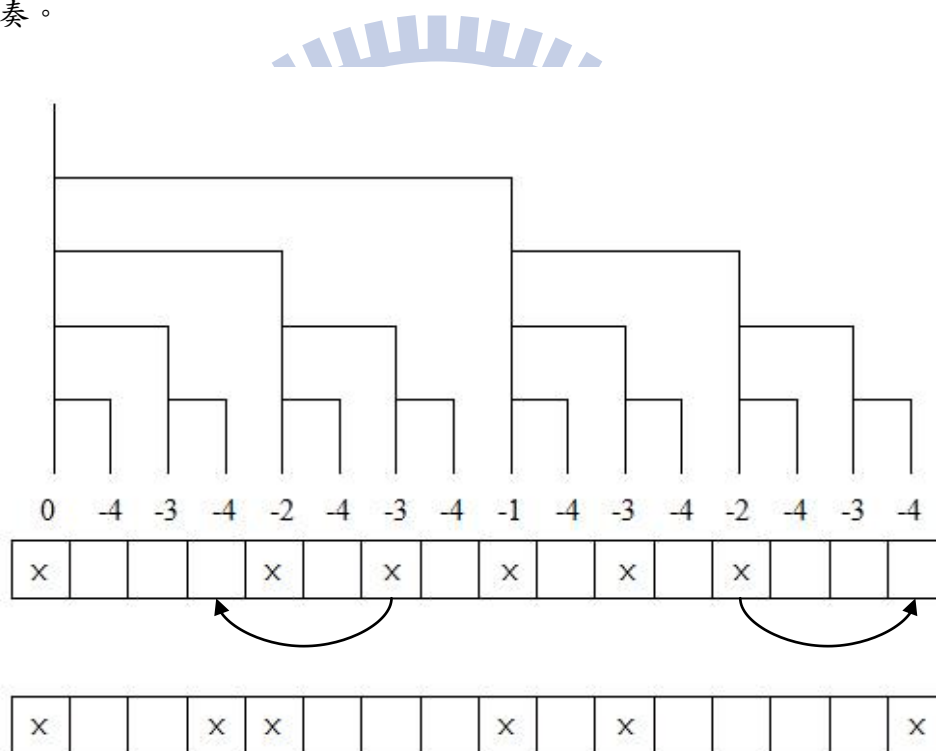


圖 36：節奏產生方法範例

只有在產生不規則節奏的時候，有一點需要特別注意的，如果設定了某些拍子會有不規則節奏產生時，我們會從調整過的權重去尋找最簡單的節奏，但此時所建立出來的最簡單節奏的複雜度值，有可能不是 0；以圖 37 為例子來說明，這是在拍號 4/4 之下，以八分音符為基礎單位，第三拍跟第四拍都有三連音的狀況下所建立的權重，如果在這樣的權重下，要找出八個 onset 的最簡單節奏的話，會選擇了兩個複雜度為-4 的值，但

是這樣的狀況複雜度為 2，因為複雜度的計算必須跟原本沒有調整過的權重值做比較，所以在八個 onset 複雜度為 0 的時候，應該是選出另外兩個在第三拍與第四拍當中以八分音符切割的兩個-3 的值，因此我們以圖 37 的節奏作為最簡單節奏時，複雜度已經是 2 了，在移動 onset 之前必須要特別注意到。

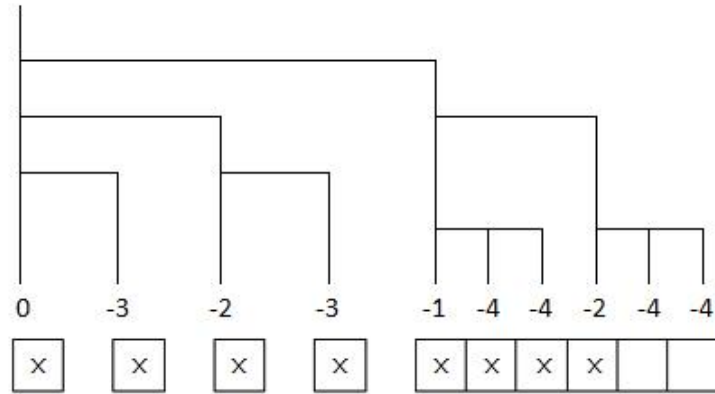


圖 37：最簡單節奏複雜度不為 0 的例子

### 3.4 系統架構

我們將複雜度分析與節奏產生兩個部分結合再一起，讓輸入的音樂可以改變拍號以及複雜度，輸出一首全新節奏的樂曲，其中每個小節的 onset 個數不會改變；對於音高的部分則不去做更動，在輸入的音樂檔中取出每一個小節的音高，在產生完節奏之後，在將每一個小節的音高依序與節奏合併，產生出一首全新節奏的樂曲；我們希望可以透過此系統去研究節奏複雜度更多的可能，因為在一般聽覺感知上，音高的變化較節奏的變化明顯，所以我們選擇不改變音高，讓聆聽者可以專注在節奏上的變化；整個系統架構的流程圖如圖 38。



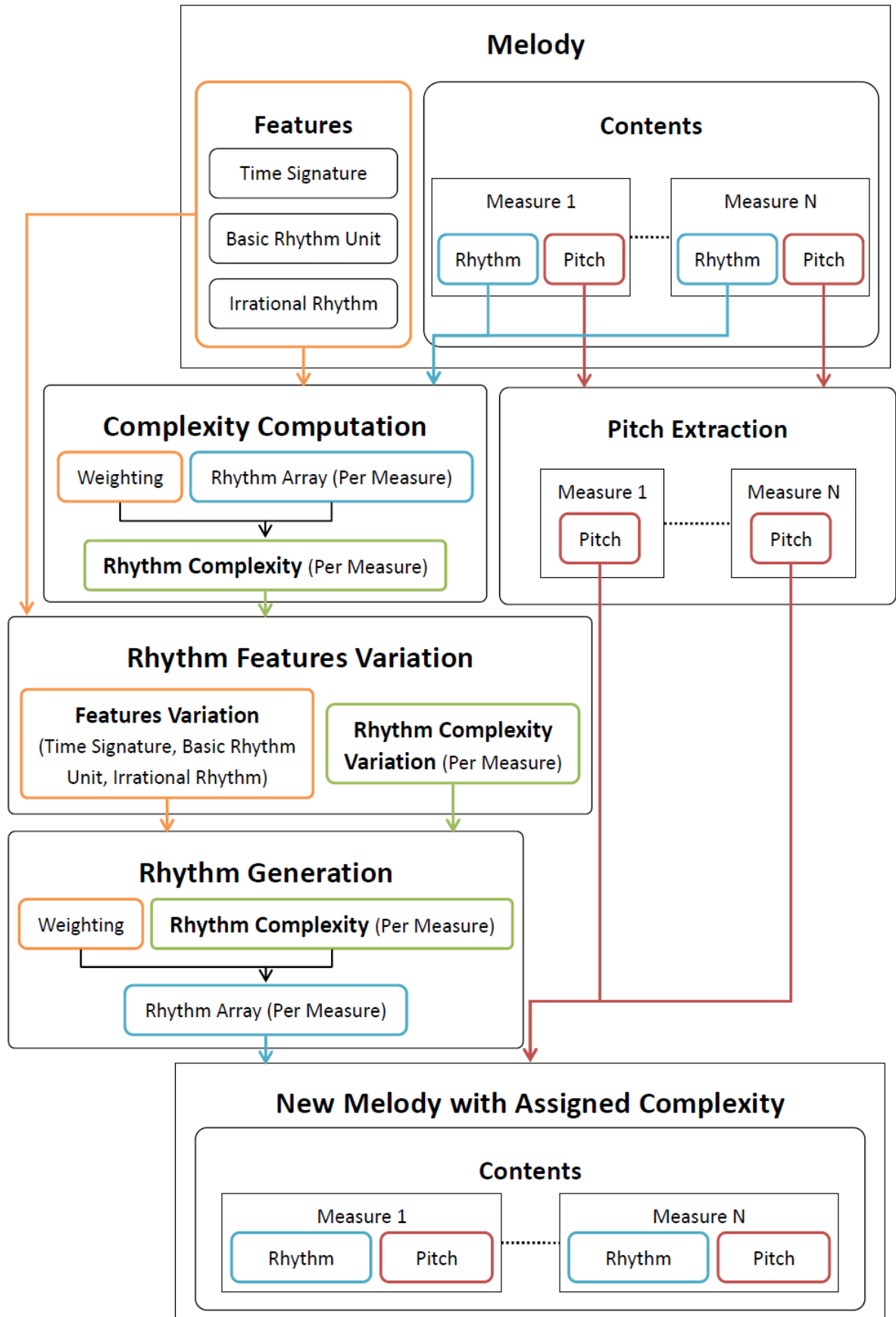


圖 38：系統架構

## 第四章、實驗結果及討論

### 4.1 實驗方法

在實驗方面，我們從三個面向評估此系統的效果。首先是驗證系統所產生之節奏的複雜度變化是否明顯，讓受試者去聆聽兩個由系統自動產生的不同複雜度的樂曲，可否明確的顯現出複雜度的高低；第二是拍號的辨識率，由系統自動產生的三種拍號，讓受試者辨別個別的拍子屬於哪一種；最後是不規則節奏的辨識率，由系統自動產生的兩個複雜度相同的樂曲，以最常見的三連音加入其中一個樂曲當中，讓受試者辨別是否有三連音的出現，且驗證兩者是否複雜度相當。所有的樂曲都由小星星(譜例如圖 39)以及巴哈 G 大調小步舞曲(譜例如圖 40)所改編，在指定個別的變數之後，由系統自動產生輸出 midi 檔，在不改變音高以及結構的狀況下，讓受試者分別聆聽與判斷。另外，我們針對結構與複雜度的變化做了實驗，分別實驗了三種不同的狀況下，結構是否會影響人所感知的樂曲複雜度的高低。這部分的樂曲同樣也由小星星以及小步舞曲改編，由系統自動產生輸出 midi 檔，在不改變音高的狀況下，讓受試者分別聆聽與判斷。

實驗中總共有 13 首由系統產生出來的 midi 檔，其中有 7 首是由小星星改編，另外 6 首是由小步舞曲所改編，所有的速度皆為 120bpm，除了讓受試者判別拍號的 3 首之外，其他 10 首在音樂進行中是沒有給任何節拍的提示，僅在樂曲出現前有兩小節的提示；實驗分作兩階段，第一階段的所有樂曲都是由小星星改編，第二階段的所有樂曲都是由小步舞曲改編，因為兩首樂曲的原複雜度都為 0，以其為標準讓受試者判斷複雜程度，利用 rating scale，讓受試者選擇在五個等級當中，從複雜度最高 5 到複雜度最低 1 當中選出相應的選項。

我們以 GTTM 的 grouping structure 為音樂結構的分析基礎，將兩首樂曲做結構分析，因為節奏產生是以小節為單位，所以在小節以下較小的 group 以及跨越小節之間的 group 則不使用，僅依照小節之上的 group 來去實作，也因為我們主要是以小節以上的 group 為主，所以在 preference rules 當中如果存在 rules 之間的衝突，我們將以 GPR5 和 GPR6

為主要依據，而小星星的 group 如圖 39，小步舞曲的 group 如圖 40，因為大範圍的 group 在圖當中不好標示，所以在圖中會跨越行之間的 group 被我們省略了。

1  $\text{♩} = 120$

5

9

圖 39：小星星譜例

1  $\text{♩} = 120$

5

9

13

圖 40：巴哈 G 大調小步舞曲譜例

實驗的受試者共三十位，我們調查了受試者的音樂背景，分別針對受過音樂訓練的時間長短（包括學習樂器、樂理等）、受過節奏訓練的時間長短（學習打擊樂器、節奏創作等）以及一天當中平均接觸音樂多久（如聆聽音樂、音樂創作、接觸樂器等）三項調查。在三十位受試者中接受過音樂訓練五年以上的比例佔了一半，只有兩人沒有接受

過任何音樂相關訓練，而平均接受過節奏訓練的時間則較短，接受過節奏訓練五年以上的僅有 13%，有三分之一沒有接受過任何的節奏訓練，而一天當中接觸音樂的時間一小時以上的則占了三分之二，詳細比例如表 14 所示。

表 14：受試者音樂背景

接受音樂訓練時間長短			接受節奏訓練時間長短			一天當中接觸音樂時間		
項目	個數	百分比	項目	個數	百分比	項目	個數	百分比
10 年以上	8	27%	10 年以上	3	10%	一小時以下	10	33%
5~10 年	7	23%	5~10 年	1	3%			
3~5 年	4	13%	3~5 年	3	10%	一至兩小時	6	20%
1~3 年	6	20%	1~3 年	6	20%			
1 年以內	3	10%	1 年以內	7	23%	兩小時以上	14	47%
從未受過音樂訓練	2	7%	從未受過節奏訓練	10	34%			

所有的實驗，我們都是讓受試者判斷整首樂曲的複雜度，然後與原來每一小節的複雜度平均去做比對，因為此系統主要是希望將節奏複雜度運用在自動作曲上，所以我們必須以樂曲整體的感覺來做比較，而不是只針對單一小節的節奏複雜度做實驗。

## 4.2 實驗結果

以下的實驗結果，我們將按照上面所提之三個系統評估的順序分別說明，並沒有按照實驗的順序來進行說明，而所有譜例編號則按照原實驗 midi 檔的編號。

### ◆ 系統產生之節奏的複雜度變化

在這個階段的實驗當中，我們以小星星改編的編號 1 以及編號 2 兩首樂曲讓受試者聆聽，選出較為複雜、較為不規則的節奏，並以一到五的等級各自評斷其複雜的程度；兩首歌曲皆為拍號 4/4，以八分音符作為基礎單位，其中編號 1 所有小節的複雜度均為 1，編號 2 所有小節的複雜度均為 1，而結構的部分是依照上面小星星的結構，在平行的段落當中以重複的節奏出現，譜例如圖 41 以及圖 42。



圖 41：編號 1 譜例（所有小節複雜度 1）



圖 42：編號 2 譜例（所有小節複雜度 6）

在 30 人當中只有 1 人選擇兩者差不多，其餘 29 人都選擇編號 2 為複雜度較高者，百分比如表 15，因為兩者本身的複雜度差異相當大，所以幾乎所有人都可以辨識出複雜度高低的差別。

表 15：編號 1 及編號 2 複雜度較高之選擇

項目	個數	百分比
編號 1	0	0%
編號 2	29	97%
差不多	1	3%

編號 1 的樂曲有超過 80% 的人給予 1 到 2 的複雜程度，平均為 1.967，依照拍號 4/4 基本單位為八分音符的複雜度最低值 0 以及最高值 6.5(onset 數不同有一樣的最高值)，

使數值以原複雜度定義範圍標準化之後為 2.327，而編號 2 的樂曲有 60% 的人給予 4 到 5 的複雜程度，平均為 3.533，標準化之後為 4.483，兩者相差較原本複雜度值來得小，有可能是因為受試者多以中間數值作為選擇，使得評分分數較為集中，詳細資訊如表 16；在此階段驗證了系統可以產生不同複雜度的節奏，並且使用者也認同複雜度的變化。

表 16：編號 1 與編號 2 複雜度比較

編號 1	個數	百分比	編號 2	個數	百分比
1	8	27%	1	0	0%
2	17	57%	2	5	17%
3	3	10%	3	7	23%
4	2	6%	4	15	50%
5	0	0%	5	3	10%
平均		1.967	平均		3.533
標準差		0.809	標準差		0.900
標準化		1.568	標準化		4.116

◆ 不同拍號之節奏產生

在這個階段的實驗，我們以小星星改編的編號 3、編號 4 以及編號 5 三首樂曲讓受試者聆聽，以三拍子、四拍子、五拍子、六拍子、七拍子與無法辨識六個選項，分別讓受試者判別拍號，其中編號 5 為 5/4，編號 6 為 3/4，編號 7 為 6/8，所有複雜度皆為 0，因為只有在複雜度為 0 時，不會打破原來的節奏階層，可以按照原本的節奏階層顯示出暗藏在拍號之內的強弱之別，三首譜例分別如圖 43、圖 44 和圖 45。



圖 43：編號 5 譜例（所有小節複雜度 0）





圖 44：編號 6 譜例（所有小節複雜度 0）



圖 45：編號 7 譜例（所有小節複雜度 0）

三首樂曲分別都有 40% 以上的辨識率，因為樂曲只有單旋律，加上沒有以力度另外強調重音，所以如果要使辨識率再增加，可能必須針對強拍的增加重音，又或者是加上伴奏，提示拍子的結構性；其中三拍子的樂曲也有 37% 的人選擇了六拍子，代表在三拍子與六拍子的拍號結構上，必須要有更關鍵性的提示，以讓聆聽者區別；選擇無法分辨的受試者，多為僅接受 1 年以下節奏訓練或從未受過節奏訓練的受試者，可能對於不同拍號的節奏較不熟悉，但有一人在編號 5 以及編號 6 都選擇了正確的拍號，可是編號 7 選擇了無法分辨，而此受試者接受過十年以上節奏訓練，猜測受試者對於六拍子節奏的預期為 [x.xx.x]，跟我們所產生的 [x..xxx] 不相同所致；詳細比較如表 17。

表 17：拍號辨識統計表

編號 5	個數	百分比	編號 6	個數	百分比	編號 7	個數	百分比
三拍子	2	7%	三拍子	12	40%	三拍子	7	23%
四拍子	5	17%	四拍子	0	0%	四拍子	0	0%
五拍子	13	43%	五拍子	0	0%	五拍子	3	10%
六拍子	1	3%	六拍子	11	37%	六拍子	14	47%
七拍子	0	0%	七拍子	3	10%	七拍子	0	0%
無法分辨	9	30%	無法分辨	4	13%	無法分辨	6	20%

◆ 不規則節奏產生

在這個階段的實驗當中，我們以小步舞曲改編的編號 14 以及編號 15 兩首樂曲讓受試者聆聽，選出較為複雜、較為不規則的節奏，並以一到五的等級各自評斷其複雜的程度，然後針對兩首樂曲分別詢問受試者是否覺得樂曲當中有三連音；兩首歌曲皆為拍號 3/4，以八分音符作為基礎單位，只有編號 15 出現了三連音，但編號 14 與編號 15 的平均複雜度都是 1.25，而結構的部分是依照上面小步舞曲的結構，在平行的段落當中以重複的節奏出現，譜例如圖 46 以及圖 47。



圖 46：編號 14 譜例（複雜度平均 1.25）



圖 47：編號 15 譜例（複雜度平均 1.25）

在 30 人當中有 8 人選擇兩者差不多，選擇編號 14 與編號 15 分別為 12 人和 10 人，百分比如表 18，因為每個人對於複雜度的感知不盡相同，但兩者原本複雜度就相當，所以結果如預期三個選項的百分比並沒有差距太大。

表 18：編號 14 及編號 15 複雜度較高之選擇

項目	個數	百分比
編號 14	12	40%
編號 15	10	33%
差不多	8	27%

編號 14 的樂曲有 70% 的人給予 2 到 3 的複雜程度，平均為 2.633，依照拍號 3/4 基本單位為八分音符的複雜度最低值 0 以及最高值 2.5(onset 數不同有不一樣的最高值)，使數值以原複雜度定義範圍標準化之後為 1.066，而編號 15 的樂曲有將近 75% 的人給予 3 到 4 的複雜程度，平均為 3.1，標準化之後為 1.313，兩者都與原複雜度 1.25 相距不大，而沒有三連音的編號 14 依照受試者的評判複雜度略為低一點，詳細資訊如表 19；這樣的數據表示出本系統的不規則節奏產生是可行的。

表 19：編號 14 與編號 15 複雜度比較

編號 14	個數	百分比	編號 15	個數	百分比
1	4	13%	1	0	0%
2	10	33%	2	5	17%
3	11	37%	3	7	23%
4	3	10%	4	15	50%
5	2	7%	5	3	10%
平均		2.633	平均		3.1
標準差		1.066	標準差		1.029
標準化		1.021	標準化		1.313

在三連音的判別方面，為了不暗示只有單一首樂曲當中有三連音，所以讓受試者分別針對兩首樂曲都回答，有 87% 的受試者可以分辨編號 15 當中有三連音的存在，代表系統所產生的三連音節奏是有辨識度的，有少數受試者無法分辨編號 14 無三連音的存在，其節奏訓練的時間也多為一年以下或是不曾接受過節奏訓練者，詳細數據如表 20。

表 20：不規則節奏辨識統計表

編號 14	個數	百分比	編號 15	個數	百分比
是	2	6%	是	26	87%
否	23	77%	否	1	3%
無法分辨	5	17%	無法分辨	3	10%

### 4.3 延伸討論

以下的實驗結果，我們將以三種節奏複雜度與結構的交互影響分別說明，並沒有按照實驗的順序來進行說明，而所有譜例編號則按照原實驗 midi 檔的編號。

#### ◆ 低複雜度與結構的表現

在這個階段的實驗當中，我們以小步舞曲改編的編號 10 以及編號 11 兩首樂曲讓受試者聆聽，選出較為複雜、較為不規則的節奏，並以一到五的等級各自評斷其複雜的程度；兩首歌曲皆為拍號 3/4，以十六分音符作為基礎單位，編號 10 與編號 11 的平均複雜度分別為 0.625 與 1，只有編號 10 是依照上面小步舞曲的結構，在平行的段落當中以

重複的節奏出現，編號 11 的節奏則不依照結構重複出現，全部由系統隨機產生，譜例如圖 48 以及圖 49。

1  $\text{♩} = 120$

5

9

13

圖 48：編號 10 譜例（複雜度平均 0.625）

$\text{♩} = 120$

5

9

13

圖 49：編號 11 譜例（複雜度平均 1.0）

在 30 人當中有 11 人選擇編號 11，占了所有受試者的 77%，原本編號 11 複雜度就較高，加上節奏不重複出現，所以如預期有較多人選擇，百分比如表 21。

表 21：編號 10 及編號 11 複雜度較高之選擇

項目	個數	百分比
編號 10	3	10%
編號 11	23	77%
差不多	4	13%

編號 10 的樂曲有超過 90% 的人給予 1 到 3 的複雜程度，平均為 2.167，依照拍號 3/4 基本單位為十六分音符的複雜度最低值 0 以及最高值 8.0(onset 數不同有不一樣的最大值)，使數值以原複雜度定義範圍標準化之後為 2.334，與原本的 0.625 有 1.7 的差別，而編號 11 的樂曲有超過 90% 的人給予 3 到 5 的複雜程度，平均為 3.367，標準化之後為 4.734，與原本的 1 有 3.7 的差別，為編號 11 的兩倍，表示打破了結構，節奏沒有重複出現，使複雜度的感覺被放大了，詳細資訊如表 22。

表 22：編號 10 與編號 11 複雜度比較

編號 10	個數	百分比	編號 11	個數	百分比
1	8	27%	1	0	0%
2	13	43%	2	2	6%
3	7	23%	3	17	58%
4	0	0%	4	9	30%
5	2	7%	5	2	6%
平均		2.167	平均		3.367
標準差		1.053	標準差		0.718
標準化		2.334	標準化		4.734

#### ◆ 無結構的複雜度感知

在這個階段的實驗當中，我們以小步舞曲改編的編號 12 以及編號 13 兩首樂曲讓受試者聆聽，選出較為複雜、較為不規則的節奏，並以一到五的等級各自評斷其複雜的程度；兩首歌曲皆為拍號 3/4，以十六分音符作為基礎單位，編號 12 與編號 13 的平均複雜度分別為 1.0 與 5.0，編號 12 與編號 13 的節奏都沒有依照結構重複出現，全部由系統隨機產生，譜例如圖 50 以及圖 51。



1  $\text{♩} = 120$

5

9

13

圖 50：編號 12 譜例（複雜度平均 1.0）

1  $\text{♩} = 120$

5

9

13

圖 51：編號 13 譜例（複雜度平均 5.0）

在 30 人當中有 24 人選擇編號 13，占了所有受試者的 80%，原本編號 13 複雜度就高於編號 12 許多，所以如預期有較多人選擇，百分比如表 23。

表 23：編號 12 及編號 13 複雜度較高之選擇

項目	個數	百分比
編號 12	2	7%
編號 13	24	80%
差不多	4	13%

編號 12 的樂曲有超過 90% 的人給予 2 到 4 的複雜程度，平均為 2.967，標準化之後為 3.934，將近原本的 1.0 的四倍，而編號 13 的樂曲有超過 90% 的人給予 3 到 5 的複雜程度，平均為 4.167，標準化之後為 6.334，與原本的 5.0 則只有 1.334 的差別，代表打破了結構，原本低複雜度的樂曲複雜度被大幅度地提高了，而原本複雜度就高的樂曲，增加的幅度則沒那麼多，也顯示了結構對於一首音樂的重要性，詳細資訊如表 24。

表 24：編號 12 與編號 13 複雜度比較

編號 12	個數	百分比	編號 13	個數	百分比
1	1	3%	1	0	0%
2	7	24%	2	1	3%
3	15	50%	3	5	17%
4	6	20%	4	12	40%
5	1	3%	5	12	40%
平均		2.967	平均		4.167
標準差		0.850	標準差		0.834
標準化		3.934	標準化		6.334

#### ◆ 節奏與結構的重要性

在這個階段的實驗當中，我們以小星星改編的編號 3 以及編號 4 兩首樂曲讓受試者聆聽，選出較為複雜、較為不規則的節奏，並以一到五的等級各自評斷其複雜的程度；兩首歌曲皆為拍號 4/4，以八分音符作為基礎單位，其中編號 3 所有小節的複雜度均為 6，而編號 2 平均複雜度為 1.83，只有編號 3 是依照上面小步舞曲的結構，在平行的段落當中以重複的節奏出現，編號 4 的節奏則不依照結構重複出現，全部由系統隨機產生，譜例如圖 52 以及圖 53。



圖 52：編號 3 譜例（所有小節複雜度 6）



圖 53：編號 4 譜例（複雜度平均 1.83）

在 30 人當中只有 3 人選擇兩者差不多，選擇原複雜度較高的編號 3 有 17 人佔了 57%，選擇原複雜度較低的編號 4 有 13 人佔了 33%，百分比如表 25，雖然兩者本身的複雜度差異相當大，但是打破了結構，節奏沒有重複出現，也使得在聽覺感知上節奏的複雜度受到影響。

表 25：編號 3 及編號 4 複雜度較高之選擇

項目	個數	百分比
編號 3	17	57%
編號 4	10	33%
差不多	3	10%

編號 3 的樂曲複雜度的選擇非常分散，平均為 1.967，依照拍號 4/4 基本單位為八分

音符的複雜度最低值 0 以及最高值 6.5(onset 數不同有一樣的最高值)，使數值以原複雜度定義範圍標準化之後為 3.738，原本的複雜度為 6，因為小星星結構簡單，節奏不斷重複出現，反而降低了原來節奏的複雜度，而編號 4 的樂曲幾乎所有人都給予 2 到 4 的複雜程度，平均為 3.033，標準化之後為 3.304，原本的複雜度平均為 1.83，但是因為打破了結構，讓節奏隨機出現，也使得複雜度被放大了，詳細資訊如表 26。

表 26：編號 3 與編號 4 複雜度比較

編號 3	個數	百分比	編號 4	個數	百分比
1	0	0%	1	0	0%
2	6	20%	2	11	37%
3	12	40%	3	8	27%
4	9	30%	4	10	33%
5	3	10%	5	1	3%
平均		3.3	平均		3.033
標準差		0.915	標準差		0.927
標準化		3.738	標準化		3.304

在這裡我們分別以三種狀況來對於節奏複雜度以及音樂結構的影響做了實驗與分析，結果也再再顯示了人類在聆聽音樂時，因為習慣了重複出現的節奏，所以對於無法預知的節奏出現時，原本低複雜度的節奏也會被認定為高複雜度的節奏，因此如果要將節奏複雜度實際運用在自動作曲上時，最好是可以配合結構分析，讓節奏與樂曲的結構結合，以創造出符合複雜度的樂曲。

## 第五章、結論與未來展望

本論文中，我們針對節奏複雜度做了各式的分析與比較，找出最適合自動作曲的複雜度定義，並將其延伸運用在不同拍號以及不規則節奏上，也提出了節奏產生的方法，並將節奏複雜度分析以及節奏產生方法實作，在實驗當中也驗證了系統可產生出不同複雜度之節奏，以及不同拍號的樂曲和含有不規則節奏的樂曲，也進一步討論了節奏複雜度與音樂結構之關係。

其中在不同拍號的樂曲產生中，結果並不是非常理想，或許可以透過力度上的改變，或者是加上伴奏等方法，使拍號的辨識程度再提高；另外在音樂結構的部分，我們只針對 GTTM 當中的 grouping structure 作分析，但如果可以將整個 GTTM 的結構套用在樂曲分析上，並把每個小節的階層關係建立出來，勢必可以使節奏複雜度與音樂結構更完整的結合，讓自動作曲能有更多的可能性。

在未來的研究中，可以考慮以多聲部的方式進行節奏的分析，針對人類在多聲部音樂下對於節奏複雜度的感知，研究多聲部的聲音當中，節奏是如何互相影響，在人類的感知當中，又是以什麼樣的形式去辨識多聲部的節奏，將原本僅限於單聲部的研究擴展至更多音樂，也可以進一步針對不同風格的音樂分析其節奏複雜度，討論節奏複雜度是否為決定風格的特徵。

我們希望透過本篇論文，提供一個以節奏為主的自動作曲概念，希望在未來可以結合其他演算法，控制曲調以及和聲上的變化，再與節奏複雜度結合，產生出多聲部的音樂，讓自動作曲更臻完善。

## 参考文献

- [1] Curt Sachs, *Rhythm and Tempo: A Study in Music History*. New York: Norton, 1953.
- [2] John Bispham, "Rhythm in Music: What is it? Who has It? And Why?," *Music Perception*, vol. 24, pp. 125-134, Dec. 2006.
- [3] Charles Ames, "The Markov Process as a Compositional Model: A Survey and Tutorial," *Leonard*, vol. 22, no. 2, pp. 175-187, 1989.
- [4] Emilios Cambouropoulos, "Markov Chains as an Aid to Computer Assisted Composition," *Musical Praxis*, vol. 1, no. 1, pp. 41-52, 1994.
- [5] George Papadopoulos, Somnuk Phon-amnuaisuk, Andrew Tuson Geraint Wiggins George, "Evolutionary Methods for Musical Composition," *International Journal of Computing Anticipatory Systems*, 1999.
- [6] Godfried T. Toussaint, "The Euclidean Algorithm Generates Traditional Musical Rhythms," *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pp. 47-56, July 2005.
- [7] Dirk-Jan Povel, "Melody Generator: A Device for Algorithmic Music Construction," *Journal of Software Engineering and Applications*, vol. 3, pp. 683-695, July 2010.
- [8] Raymond Herbert Stetson, "A motor theory of rhythm and discrete succession," *Psychological*, vol. 12, no. 4, pp. 250-270, July 1905.
- [9] H. E. WEAVER, "Syncopation: A Study of Musical Rhythms," *The Journal of General Psychology*, vol. 20, pp. 409-429, 1939.
- [10] Jeff Pressing, "Cognitive complexity and the structure of musical patterns," *Noetica*, vol. 3, no. 8, pp. 1-8, 1998.
- [11] H. Christopher Longuet-Higgins and Christopher S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception*, vol. 1, no. 4, pp. 424-441, 1984.
- [12] Paul C. Vitz, "Information, run structure and binary pattern complexity.," *Perception and Psychophysics*, vol. 3, no. 4, pp. 275-280, 1968.
- [13] Paul C. Vitz and Thomas C. Todd, "A coded element model of the perceptual processing of," *Psychological Review*, vol. 76, no. 5, pp. 433-449, 1969.
- [14] D. Povel and P. Essens, "Perception of temporal patterns," *Music Perception*, vol. 2, pp. 411-440, 1985.
- [15] Peter Essens, "Structuring temporal sequences: Comparison of," *Perception and Psychophysics*, vol. 57, no. 4, pp. 519-532, 1995.
- [16] Ilya Shmulevich and Dirk-Jan Povel, "Measures of temporal pattern complexity," *Journal of New Music Research*, vol. 29, no. 1, pp. 61-69, 2000.

- [17] W. Tecumseh Fitch and Andrew J. Rosenfeld, "Perception and production of syncopated rhythms," *Music Perception*, vol. 25, no. 1, pp. 43-58, 2007.
- [18] James Koetting, "Analysis and notation of west African drum ensemble music," *Selected Reports in Ethnomusicology*, vol. 1, no. 3, pp. 115-146, 1970.
- [19] David Huron and Ann Ommen, "An empirical study of syncopation in American popular," *Music Theory Spectrum*, vol. 28, no. 2, pp. 211-231, 2006.
- [20] Don Michael Randel, "Syncopation," *The Harvard Dictionary of Music, Fourth*. Cambridge: Belknap Press of Harvard University Press, 2003, pp. 861-862.
- [21] David Miles Huber, *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*, 3rd ed.: Focal Press, 2007.
- [22] Michael Good, "MusicXML: An Internet-Friendly Format for Sheet," *Proceedings of the XML Conference & Exposition*, Dec. 2001.
- [23] Fred Lerdahl and Ray S. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge: MIT Press, 1983.
- [24] Masatoshi Hamanaka and Satoshi Tojo, "Interactive GTTM analyzer," *Proceedings of the 10th International Conference on Music Information Retrieval Conference (ISMIR)*, pp. 291-296, Oct. 2009.
- [25] Godfried T. Toussaint, "A mathematical analysis of African, Brazilian, and Cuban clave rhythms," *BRIDGES: Mathematical Connections in Art, Music and Science*, pp. 157-168, July 2002.
- [26] Michael Keith, *From Polychords to Pólya: Adventures in Musical Combinatorics*:. Vinculum Press, 1991.
- [27] F. Gómez, A. Melvin, D. Rappaport, and G. T. Toussaint, "Mathematical measures of syncopation," *BRIDGES: Mathematical Connections in Art, Music and Science*, pp. 73-84, July 2005.
- [28] F. Gómez, E. Thul, and G. T. Toussaint, "An experimental comparison of formal measures of rhythmic syncopation," *Proceedings of the International Computer Music*, pp. 101-104, Aug. 2007.
- [29] E. Thul and G. T. Toussaint, "On the Relation Between Rhythm Complexity Measures and Human Rhythmic Performance," *C3S2E '08: Proceedings of the 2008 C3S2E conference*, pp. 199-204, 2008.
- [30] Leigh M. Smith and Henkjan Honing, "Evaluating and extending computational models of rhythmic," *Proceedings of the International Computer Music Conference*, pp. 688-691, 2006.