生成式艺术初探

使用短乐曲片段样本的马尔可夫链方法自动作曲

林汇平，王昕兆，王子健

{1800013104,1800013102,1800012915}@pku.edu.cn

摘要

自动作曲是使用某个形式化的过程，利用计算机进行乐曲创作。本文使用马尔可夫链进行自动作曲，主要方法为：选择某个已有的音乐片段作为生成样本，统计出其状态空间与概率转移矩阵，使用马尔可夫链生成随机音乐，利用评价函数选取出较好的音乐。我们选择了不同风格、不同特征的 8 条音乐片段，形式化地写出其旋律行进后，使用 Python 语言撰写了实验代码，并开展了相关实验，在不同的音乐片段样本上生成了相应的随机乐曲，分析并评价了对应的生成结果。

1 介绍

自动作曲 (算法作曲) 是试图使用某个形式化的过程, 以使人 (或作曲家) 在利用计算机进行音乐创作时介入程度达到最小的研究 [1]。目前在自动作曲中公认的有影响的技术主要有马尔可夫链、随机过程、基于规则的知识库系统、音乐文法、人工神经网络、遗传算法等。本文中我们使用马尔可夫链进行作曲。

1.1 马尔可夫模型

马尔可夫模型是一种统计模型，可以用于计算条件概率分布，为一系列的离散事件建模。马尔可夫链为状态空间中经过从一个状态到另一个状态的转换的随机过程。该过程要求具备“无记忆”的性质：下一状态的概率分布只能由当前状态决定，在时间序列中它前面的事件均与之无

关。这种特定类型的“无记忆性”称作马可夫性质。马尔科夫链作为实际过程的统计模型具有许多应用。

在马尔可夫链的每一步，系统根据概率分布，可以从一个状态变到另一个状态，也可以保持当前状态。状态的改变叫做转移，与不同的状态改变相关的概率叫做转移概率。随机漫步就是马尔可夫链的例子。随机漫步中每一步的状态是在图形中的点，每一步可以移动到任何一个相邻的点，在这里移动到每一个点的概率都是相同的（无论之前漫步路径是如何的）。

在此问题中，我们把状态空间记为 $\{X_t\}$，其中 X_t 表示时值为 t、音名为 X 的音符。$P(X_t)$ 是随机事件 X_t 的概率分布，满足 $\sum_{Y,s} p(X_t, Y_s) = 1$ 的归一化条件。

对于 n 阶马尔可夫链（其中 $n \geq 1$）马尔可夫模型可以基于「上文」做出判断和预测，未来状态只取决于当前状态或者限定范围的过去状态。

对于该建模过程，形式化定义如下：

所选择的旋律记为序列 $\{A_1, A_2, \cdots, A_n\} = \{A_i \mid i = 1, 2, \cdots, n\}$，$i$ 表示旋律中的音符的编号，即在第 i 个位置出现。。

状态空间集合：$\{X_t \mid \exists\, i, s.t. X_t = A_i\}$

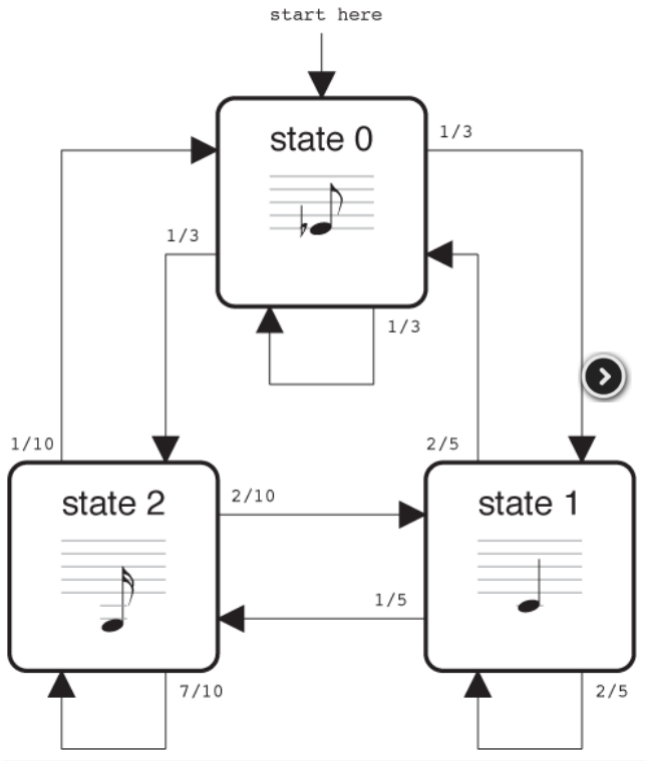
转移矩阵：$p(X_t, Y_s) = \dfrac{\sum_{i=1}^{n-1} [A_i = X_t \cap A_{i+1} = Y_s]}{\sum_{i=1}^{n-1} [A_i = X_t]}$

这里 $[\,\cdot\,]$ 是指示函数，$[x] = 1$ 当且仅当 x 为真。因此概率转移矩阵其实就是从一个状态转移到另一个状态的频率。

实现马尔可夫模型的学习算法有几个步骤：

- 构建一个 transition count table (state transition matrix)，计算每一种可能的上下文的频率分布。
- 对频率进行归一化。用每一种组合的数量除以所有的组合总数，即保证每一种情况的概率之和为 1。
- 随机选择一个起始值，根据概率表格选择下一个序列值。

举一个三状态的例子：



这首马尔可夫旋律以 state 0 开始，播放一个八分音符 ♭E。然后选择一个新的状态。选择 state 0，state 1 或 state 2 的概率相等，都是 $\frac{1}{3}$。假设选择了 state 2，则播放下加二间的十六分音符 G。从 state 2 开始，state 0 被选择的概率是 $\frac{1}{10}$，state 1 是 $\frac{1}{5}$，state 2 是 $\frac{7}{10}$。

马尔可夫模型已经被大量应用于生成式艺术和算法创作中。Lejaren Hiller 在 1957 年完成了算法生成的弦乐四重奏「依利亚克组曲」(Illiac Suite)，这也是历史上第一支完全由计算机生成的音乐作品。首先使用马尔可夫链模型来产生有限控制的随机音符，之后利用和声与复调的规则测试这些音符，最后选择符合规则的材料，修改、组合成传统音乐记谱的弦乐四重奏。

该作品分为四个乐章：

- 第一乐章：计算机生成的不同长度的固定主题旋律
- 第二乐章：使用变奏的规则生成的四声部音乐
- 第三乐章：通过计算机对节奏、动态和演奏法的不同处理生成的音乐
- 第四乐章：通过衍生算法和马尔可夫链的不同模型及概率生成的音乐（pitch, intervals and textures）

类似地，Iannis Xenakis 在他 1958 年的专辑 Analogique 中就使用了马尔可夫链来作曲。在他的著作 Formalized Music: Thought and Mathematics in Composition 里详细描述了使用马尔可夫模型的算法 [4]。

1.2 具体实现

我们使用了 python 语言作为此项目的主要框架，在 music21 库 [2] 的支持下实现了使用 Markov 链的机器作曲。我们的方法可以分为以下几个步骤：1、选取合适的原始音乐作为样本。2、生成单首或两首原始音乐对应的 Markov 矩阵。3、多次使用 Markov 链生成随机音乐。4、设计评价算法在大量的生成音乐中自动选取较好的音乐。

music21 库是由麻省理工学院 Cuthbert 实验室开发的计算机辅助音乐分析工具包，支持包括 MusicXML、MIDI、abc 等多种格式的音乐文件，具有强大的音乐分析功能，甚至可以从零开始构建音乐文件。在此项目中，我们主要利用 music21 库将（音高，时值）序列转化成音频文件和曲谱文件，转化部分代码如下：

```
from music21 import *

def save_stream(save_stream, path, name):
        save_stream.write('midi', fp=path+name+'.mid')
        save_stream.write('xml', fp=path+name+'.xml')

def save_music(music, path, name):
        mini_stream = stream.Stream(music)
        save_stream(mini_stream, path, name)

def generate_stream(note_list, interval_list, tune):
        gen_stream = stream.Stream()
        ts = meter.TimeSignature(tune)
        gen_stream.insert(0, ts)
        n = len(note_list)
        for i in range(n):
```

```
                f = note.Note(note_list[i])
                f.duration = duration.Duration(interval_list[i])
                gen_stream.append(f)

        return gen_stream

if ___name___ == '___main___':
        note_list = ['E4', 'G4', 'G4', 'G4']
        interval_list = ['eighth', 'eighth', 'quarter', 'eighth']
        tune = '2/4'
        gen_stream=generate_stream(note_list, interval_list, tune)
        gen_stream.show('midi')
        save_stream(gen_stream, 'test/', 'sample')
```

2 数据准备

我们选择了 8 首不同时代、风格、流派、地域、节拍与调号的短旋律，用以进行实验。该部分的数据在 python 下撰写格式为：

```
note_list = ['E4#', 'G4-', 'G4##', 'G4']
        interval_list = ['eighth', 'eighth', 'quarter', 'eighth']
        tune = '2/4'
```

其中数组 note_list 为音符的列举，一个音符由三部分构成：音名、音区、变音记号。数组 interval_list 为每个音所对应的节拍。变量 tune 为乐曲的节拍。

在数据选择中，我们摘取了一首乐曲中比较有代表性的旋律，为了避免状态空间过大，我们每次选取的音符个数不超过 20 个，保证生成的乐曲与原曲关系较接近。实际上，我们也可以理解为以这一小段旋律为“动机”，所进行的展开。

下面我们详细介绍选取的乐曲以及其特征。

1. 格里高利圣咏

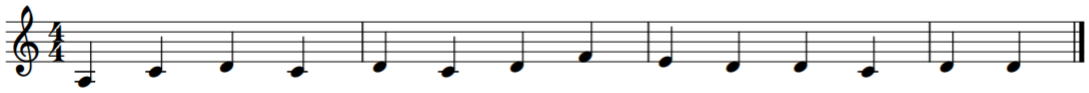
格里高利圣咏是以教皇格里高利一世命名的天主教礼仪式音乐，是西方教会单声圣歌的主要传统，是一种单声部、无伴奏的罗马天主教宗教音乐，主要是在第 9 世纪和第 10 世纪法兰克人到达西欧和中欧期间发展起来。

格里高利圣咏共有 8 种调式。它的特征是无明显节拍，旋律平缓，以级进和三度进行为主，且音域控制在比较窄的范围内，且建立在自然音阶基础上，很少有音程的跳进。以级进和三度为主，偶尔会有四五度跳。

从下图可以看出，我们选择的旋律进行非常平缓，也并无大的起伏。

Music 1

Music21



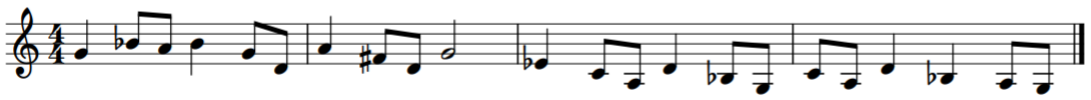
2. 加伏特舞曲

这首加伏特舞曲是巴赫的著名作品。巴赫是巴洛克时期德国作曲家。加伏特舞曲原是法国古代民间舞曲。17 世纪中叶由法国作曲家吕利 (1632-1687) 采用于歌剧、舞剧中，在法、德两国流行一时。其特点为中速，2/4 或 4/4 拍，旋律轻盈优雅。巴洛克音乐的节奏强烈、跳跃，比较强调曲子的起伏。乐句长度不一，气息较长，无论是声乐还是器乐，旋律都空前地华丽复杂，有相当多的装饰音和模进音型。和文艺复兴时期的旋律一样，曲调的进行常常带有形象化、象征性特征。

我们选择的这首曲子是旋律跳跃性很强、极具舞曲特点与巴洛克时期音乐特点的，选出的是右手声部中的代表性旋律。

Music 2

Music21

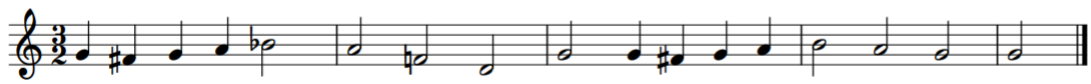


3. 恰空舞曲

这首恰空舞曲是维塔利的小提琴曲代表作。维塔利是意大利作曲家、小提琴家。恰空舞曲是最早出现 16 世纪的西班牙，是三拍子的，其特点为在固定低音的基础上保留和声框架并对和声织体进行变奏。维塔利的这首《恰空》是变奏曲式，共 17 个部分，我们所选取的是最开始的第一部分，这是该乐曲的主体部分，且是这首乐曲的主旋律，开始得庄重又极具动力性，旋律线条明朗，歌唱性强。

Music 3

Music21



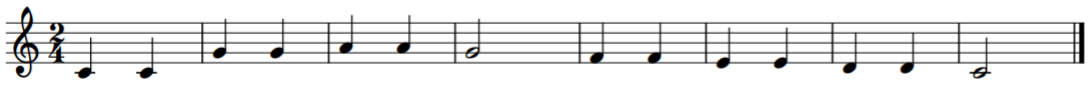
4. 小星星变奏曲

这首《小星星变奏曲》是 C 大调作品第 K. 265/300e，由莫扎特于 1778 年所创作的钢琴曲。原题直译的意思为在法国歌曲《妈妈请听我说》基础上创作的 12 段变奏曲。莫扎特是古典主义时期奥地利作曲家，维也纳古典乐派代表人物之一。这首乐曲音乐主题出自一首古老的欧洲民谣，这个主题的节奏与旋律单纯质朴，莫扎特为它配上十二段可爱又富有魅力的变奏，乐声一直自然而愉快的流淌着。其歌曲主题浪漫而梦幻，由六个四分音符加一个二分音符组成了轻灵的旋律。

我们所选择的这一段为主题部分，即是法国歌曲《妈妈请听我说》。

Music 4

Music21



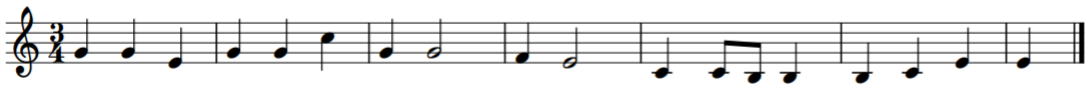
5. C 大调奏鸣曲

这首《C 大调奏鸣曲》是海顿的作品。海顿是奥地利古典主义时期作曲家，维也纳古典乐派奠基人。海顿的音乐形式多样，且充满趣味性。急速的音量高低对比、调性的转变，以及看似漫不经心的音乐句法，都构成了海顿创造幽默式浪漫的方式。

我们选取的这段旋律轻盈、跳跃，坚定且率真，是海顿作品特点的高度体现。

Music 5

Music21



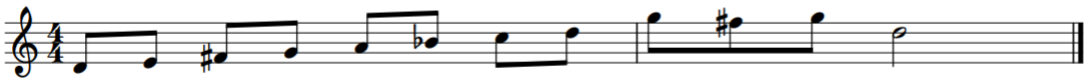
6. 六月 (船歌)

这首《船歌》是柴可夫斯基的作品。柴可夫斯基是俄罗斯浪漫乐派作曲家，作品带有浓郁的俄罗斯民歌的色彩。柴可夫斯基的创作几乎涉及了所有的音乐体裁和形式，其中交响乐创作处于重要位置。他继承了格林卡以来俄罗斯音乐的发展成就，同时又注意吸取西欧音乐文化发展的经验，把高度的专业技巧同俄罗斯民族音乐传统有机结合，创造出具有戏剧性冲突和浓郁民族风格的作品。

这首《船歌》为柴可夫斯基的套曲《四季》中的第六首，描绘的是六月夏日的夜晚，人们坐在小船上在映着月光的湖水中悠闲荡漾之情景。旋律优美，织体多样，和声丰富，色彩绚丽，每一首乐曲都是一幅五彩缤纷的风俗画。我们选取的这一段先是平稳的旋律推进，然后是大跨度的跳进，旋律感很强。

Music 6

Music21

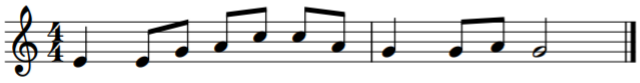


7. 茉莉花

这首《茉莉花》是中国传统音乐的典型代表，是一首脍炙人口的中国传统民歌。《茉莉花》产生于明末清初，后流传东北地区及全国各大城市的小调歌曲，距今约有 300 年的历史。这首曲子的旋律虽然简单，但是非常优美动听，让人回味无穷，应用中国五声调式，是中国传统民歌的典型代表。

Music 7

Music21

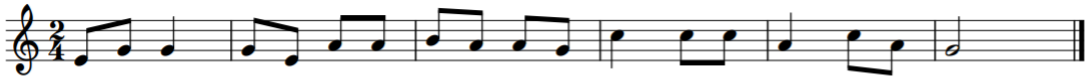


8. 童年

《童年》是由罗大佑作词作曲的一首流行乐。它的旋律活泼，节奏欢快，语调轻快，是一首描写童年纯真的歌曲。

Music 8

Music21



3 实验

考虑到一阶 Markov 链的下一个状态只和上一个状态有关，很难捕捉到更高层次的信息，因此我们的代码实现了 m 阶 Markov 链

$$\Pr(X_n = x_n \mid X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_1 = x_1)$$

$$= \Pr(X_n = x_n \mid X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-m} = x_{n-m}) \text{ for } n > m$$

这部分的代码参考了一些开源项目 [3]，我们把核心功能封装成了 python 中的三个 class: MarkovBuilder, MarkovMatrix, MusicGenerator。

3.1 MarkovBuilder

MarkovBuilder 是最底层的 Markov 链部分，我们使用 python 的 numpy 库来实现对高维 Markov 链的存储和操作，通过记录一个状态和编号的映射表和逆映射表来实现对状态的编码和解码 (下面的代码都只是示例)

```
class MarkovBuilder:
    def ___init___(self, value_list, order):
        # 映射表和逆映射表
        self.value_lookup = {}
        self.reverse_value_lookup = value_list
        self.order = order
        value_num = len(value_list)
        for i in range(0,value_num):
            self.value_lookup[value_list[i]] = i
        # 初始化转移矩阵
        self.matrix = np.zeros([value_num for i in range(order+1)],
   dtype = int)
```

实例方法包含 add, next_value, random_choose:

```
class MarkovBuilder:
    # 向Markov链中输入数据并更新Markov转移矩阵
    def add(self, from_value, to_value)
```

```
    # 根据from_value生成下一个状态
    def next_value(self, from_value)
    # 从choice_counts中按照和大小成正比的概率选择一个index
    def random_choose(self, choice_counts)
```

值得一提的是，因为我们的数据不够多，因此经常会出现当前状态到任何一个状态的转移概率都是零，为了应对这种情况，我们在 MarkovBuilder 中维护了一个之前接收过的所有状态出现次数的列表 previous_state，如果遇到没有状态可以跳转的的情况，那么会根据之前出现的所有状态的出现频率随机选择其中一个跳转，下面是 random_choose 部分的代码：

```
def randomly_choose(self, choice_counts):
    counted_sum = 0
    count_sum = sum(choice_counts)
    length = len(choice_counts)
    if count_sum == 0:
        # 如果转移概率都是0，那么从之前出现过的状态按照出现次数随机选一个状态
        index = self.randomly_choose(self.previous_state)
        return index
    selected_count = random.randrange(1, count_sum + 1)
    for index in range(0, length):
        counted_sum += choice_counts[index]
        if(counted_sum >= selected_count):
            return index
```

3.2 MarkovMatrix

MarkovMatrix 是在 MarkovBuilder 更高层次的封装，它会自动记录之前输入的状态，初始化状态空间，以及生成最后的概率转移矩阵。

我们通过 previous_state 记录之前输入的 m 个状态（m 是 Markov 链的阶数），在每次输入新的转态后都需要更新 previous_state。

```
class MarkovMatrix:
    def add(self, to_note, to_interval):
        to_state = (to_note, to_interval)
        ...
        self.state_markov_matrix.add(self.previous_state, to_state)
        self.previous_state = self.previous_state[1:] + [to_state]
```

因为之前的矩阵只是用正整数记录了不同转移发生的次数，因此，生成最后的概率转移矩阵的时候需要归一化概率分布，并且对于全零的转移向量，需要将其替换为上一部分提到过的概率向量（每个状态被选中的概率正比于它之前出现的次数）

又因为我们考虑的是对于 m 维高维数组的操作，所以需要用递归函数实现：

```
class MarkovMusic:
    # 生成最后的概率转移矩阵，需要调用递归函数enum
    def get_matrix(self):
            prob_matrix = self.state_markov_matrix.matrix.astype(
   float)
            state_num = len(self.state_list)
            self.previous_state_cnt = np.array(self.state_markov_matrix
    .previous_state, dtype=float)
            self.previous_state_cnt = self.previous_state_cnt/np.sum(self
    .previous_state_cnt)
            self.enum(state_num, 0, prob_matrix)
            return prob_matrix
    # 递归函数enum遍历m维数组
    def enum(self, state_num, depth, matrix):
        if depth == self.order:
            # 如果所有转移概率都是0
            if np.sum(matrix) == 0:
                for i in range(state_num):
```

```
                    matrix[i] = self.previous_state_cnt[i]
            else:
                s = np.sum(matrix)
                for i in range(state_num):
                    matrix[i] = matrix[i]/s
            return
        else:
            for i in range(state_num):
                self.enum(state_num, depth + 1, matrix[i])
```

3.3 MusicGenerator

MusicGenerator 是最后的音乐生成器，需要传入之前训练好的 MarkovMatrix 作为参数。MusicGenerator 只有一个实例方法 next_state 并且没有参数。

```
class MusicGenerator:
    def next_state(self)
```

3.4 示例

首先我们指定原始音乐片段的音高序列、时值序列和拍号，以及 Markov 链的阶数从而生成 MusicMatrix 的实例：

```
note_list = ['G4','B-4','A4','B-4', 'G4','D4','A4','F#4', 'D4','G4','E-4','C4'
    ,'A3', 'D4','B-3', 'G3', 'C4', 'A3', 'D4', 'B-3','A3','G3']
interval_list = ['quarter','eighth','eighth','quarter','eighth','eighth','quarter','
    eighth','eighth','half', 'quarter','eighth','eighth','quarter','eighth','eighth','
    eighth','eighth','quarter','quarter','eighth','eighth']
tune = '4/4'
order = 1
markov_instance = MusicMatrix(order)
```

输入数据，训练 Markov 链：

```
for i in range(length):
    #每次向markov_instance中丢一个note和interval，这些全部用来训练markov
    chain
    markov_instance.add(note_list[i], interval_list[i])
```

用训练好的 Markov 链生成 MusicGenerator 的实例：

```
generator = MusicGenerator(markov_instance, note_list[:order],
    interval_list[:order])
```

不断调用 generator.next_state 即可生成随机音乐，保存在 state_list 中：

```
for loop in range(len):
    state_list.append(generator.next_state())
```

4 评价算法

为了评价使用 Markov 链生成出的音乐的质量，以便在多轮随机生成过程中选出较好的结果，我们设计了生成音乐的评价函数。

我们先考虑原始音乐只有一段的情况。设待评价的生成音乐为 $G = \{g_1, g_2, \ldots, g_n\}$，原始音乐为 $O = \{o_1, o_2, \ldots, o_m\}$（假设 $n \geq m$）。其中 $g_i, o_j \in S = Scale \times Interval, Scale = \{A_3, \flat A_3, \flat\flat A_3, \sharp A_3, \times A_3, B_3, \ldots, \times G_3, \ldots, \times G_5\}, Interval = \{whole, half, quarter, eighth, 16th, 32th, 64th\}$，即所有可能出现的音符。但这样的表示仍然很不方便，考虑将这个二元组集合映射到一个正整数：建立映射函数 $S \xrightarrow{f} \mathbb{N}, f(s,i) = p_S(s) \times |Interval| + p_I(i)$，其中 $p_S(s)$ 表示 s 在集合 $Scale$ 转化成序列后在其中出现的位置，$p_I(i)$ 表示 i 在集合 $Interval$ 转化成序列后在其中出现的位置。

我们将生成音乐 G 与原始音乐 O 经过 f 映射后得到两个正整数序列 $P = f(G) = p_1 p_2 \ldots p_n$，$Q = f(O) = q_1 q_2 \ldots q_m$。由于随机音乐是在原始音乐的基础上生成的，所以我们很自然地想到用生成音乐与原始音乐的近似程度来衡量生成音乐的优劣。我们将一段音乐看成一个满足特定波动规律的随机变量，引入统计学中分析统计变量相关程度的 SROCC 指标（Spearman rank

order correlation coefficient，斯皮尔曼秩相关系数）：对于两个样本大小相等的统计变量 X, Y，其 SROCC 为

$$\rho(X,Y) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2} \in [-1, 1]$$

$\rho(X,Y)$ 越大说明两个变量的单调相关性越大，当 $\rho(X,Y) = 1$ 时，X 与 Y 完全单调相关。

由于序列 P Q 的长度并不相等，我们考虑在 P 中找到若干个连续段 $P[s_i : s_i + m - 1]$，用这些连续段与 Q 的 SROCC 之和作为 P, Q 的相关性指标 $relevance_k(P,Q)$，其中 k 为可调整的参数。

$$relevance_k(P,Q) = \max_{\substack{s_1 \neq s_2 \neq \ldots \neq s_k \\ 1 \leq s_i \leq n-m+1}} \sum_{i=1}^{k} \rho(P[s_i : s_i + m - 1], Q)$$

实验发现，通过这种评价标准选出的音乐可能出现旋律大量重复的问题，主要体现在出现很多相同的相邻对与间隔对，或是出现连续重复的子串。于是我们设计了对生成音乐的惩罚函数：

$$\begin{aligned} pun(P) = \sum_{i<j} &[(p_i, p_{i+1}) = (p_j, p_{j+1})] + [(p_i, p_{i+2}) = (p_j, p_{j+2})] \\ &+ [P[i : j-1] = P[j : 2j-i-1]] \times (j - i) \end{aligned}$$

使用 $score(G) = relevance_k(P,Q) \div pun(P)$ 作为生成音乐 G 的最终评价函数。

对于两个原始音乐 O_1, O_2 混合生成随机音乐 G 的情况，对 $relevance_k(P,Q)$ 函数做出以下修改：

$$\begin{aligned} relevance_k(P, Q_1, Q_2) = \max_{\substack{s_1 \neq s_2 \neq \ldots \neq s_k \\ 1 \leq s_i \leq n-m+1}} \sum_{i=1}^{k} \max\{&\rho(P[s_i : s_i + m - 1], Q1), \\ &\rho(P[s_i : s_i + m - 1], Q2)\} \end{aligned}$$

使用 $score(G) = relevance_k(P, Q_1, Q_2) \div pun(P)$ 作为生成音乐 G 的最终评价函数。

5 混合音乐

因为我们的数据较少而且类型各异，有些风格的音乐并不适合结合在一起，因此我们对所有音乐的相似度进行评估，选择其中最接近的几对来生成混合音乐。

```
#这段比较之前音乐的相似性
sim = np.zeros([8,8])
for i in range(8):
    for j in range(8):
        #排除掉自己和自己混合的情况
        if i == j:
            sim[i][j] = 0
        else:
            sim[i][j] = float(len(set(state_list[i])&set(state_list[j]))/
   min(len(set(state_list[i])),len(set(state_list[j]))))
print(sim)
sort_list = sim.flatten().argsort()
mix = [(sort_list[i]//8, sort_list[i]%8) for i in range(64-12, 64)]
```

6 实验分析

6.1 单原始音乐的实验分析

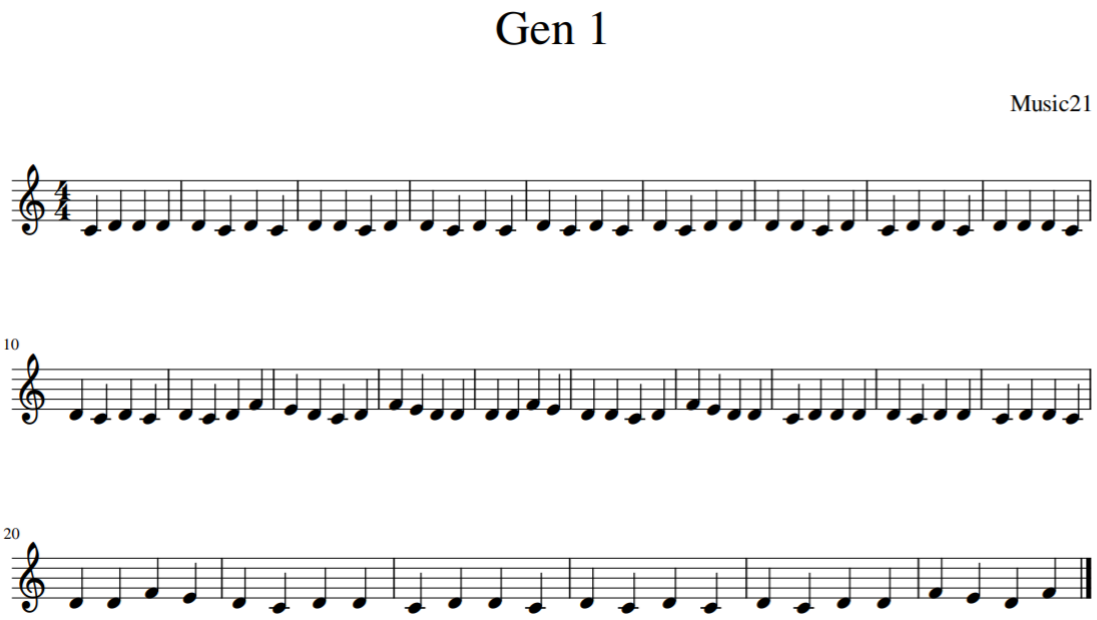
我们对于每一首原始音乐都进行了 Markov 音乐生成实验：首先构建输入音乐对应的一阶 Markov 矩阵，然后开始生成随机音乐，每个随机音乐由 100 个音符组成，共生成 1024 轮随机音乐。使用评价算法在这 1024 首生成音乐中选取分数最高的 3 首音乐，最后再从中人工筛选出听觉感受最好的音乐。

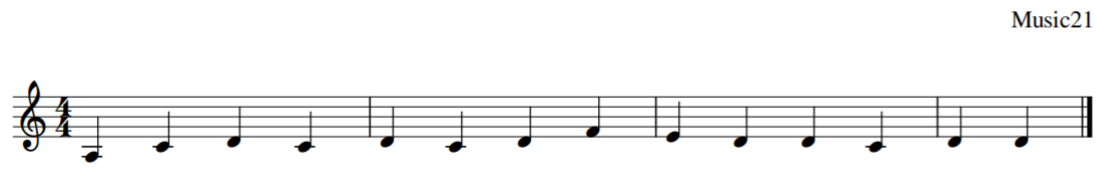
以下是我们的实验结果。我们分别罗列出状态空间、转移矩阵、生成结果与原始音乐，以及我们对该次生成的分析与评价。

6.1.1 Music 1 格里高利圣咏

('A3', 'quarter'),('C4', 'quarter'),('D4', 'quarter'),('E4', 'quarter'),('F4', 'quarter')

$$\begin{pmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.33 & 0.0 & 0.16 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix}$$





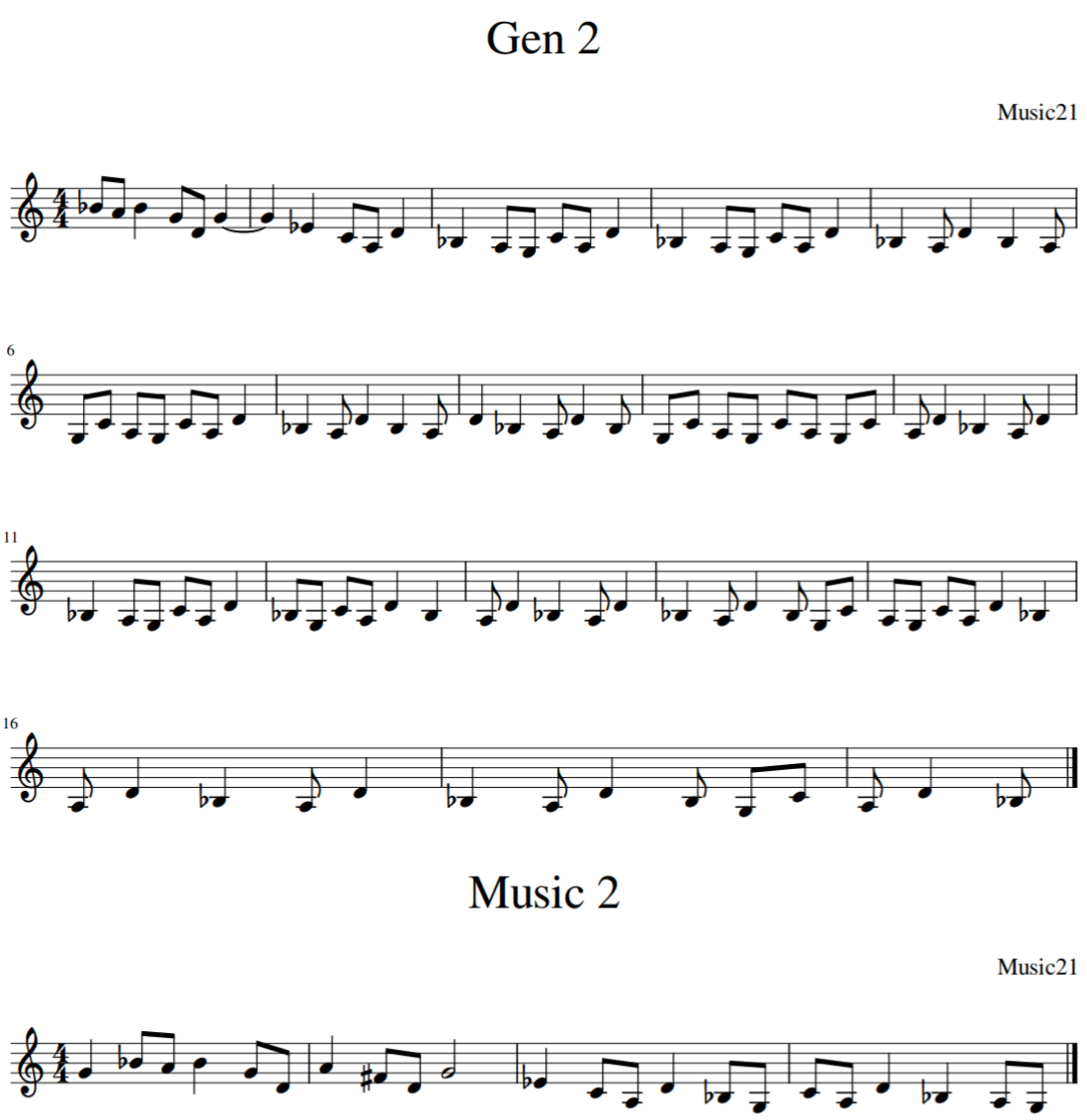
可以看到，生成的音乐有较多的相同片段的循环（如第一行中大量的 C、D 两个音），后续的乐曲行进更自然一些，但也是起伏跨度都很小，且重复偏多。这是因为格里高利圣咏本身的状态空间就很小，旋律也很简单。

6.1.2 Music 2 加伏特舞曲

('A3', 'eighth'),('B-3', 'quarter'),('B-3', 'eighth'),('G3', 'eighth'),('A4', 'quarter'),('A4', 'eighth'),('B-4', 'quarter'),('B-4', 'eighth'),('C4', 'eighth'),('D4', 'quarter'),('D4', 'eighth'),('E-4', 'quarter'),('F#4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth')

表 1: matrix 2

0.0	0.0	0.0	0.33	0.0	0.0	0.0	0.0	0.0	0.66	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0



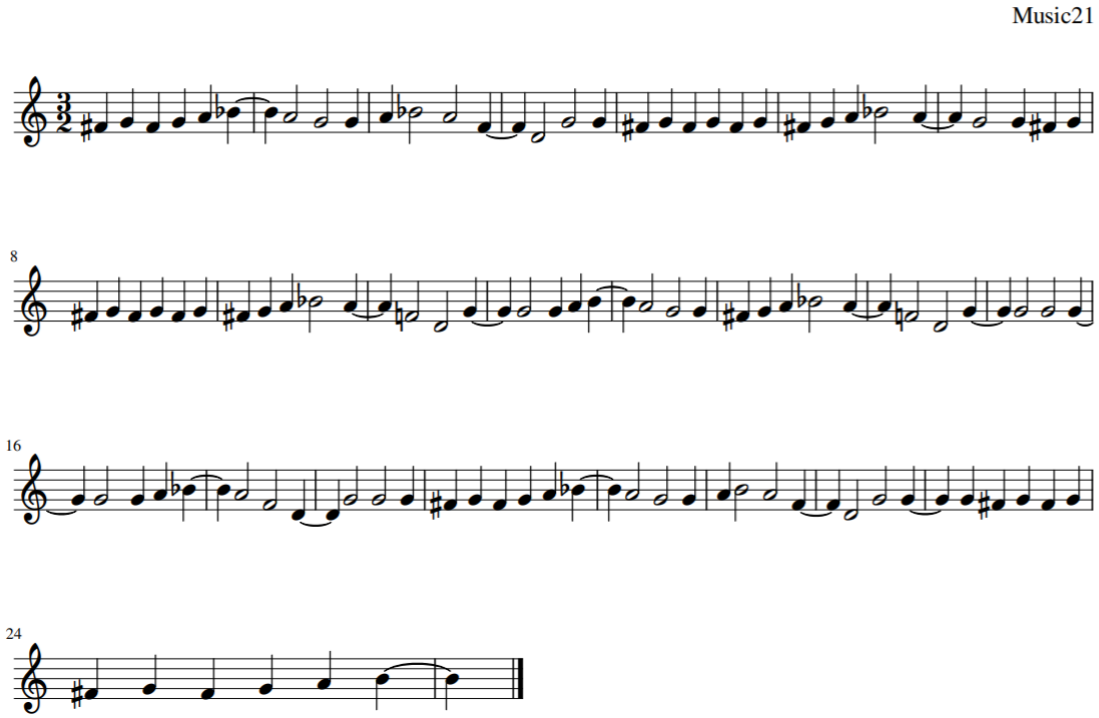
这里的生成的结果还是比较令人满意的，虽然仍有重复片段较多的情况，但整体上更像是在原版加伏特舞曲上的展开，乐曲中的一些旋律走向完美符合了加伏特舞曲，但展开得更丰富一些。

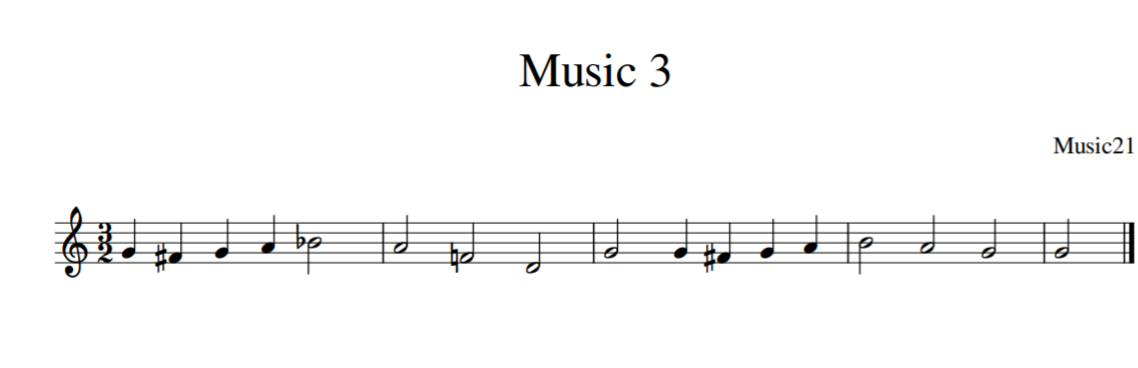
6.1.3 Music 3 恰空

('A4', 'half'),('A4', 'quarter'),('B4', 'half'),('B-4', 'half'),('D4', 'half'),('F4', 'half'),('F#4', 'quarter'),('G4', 'half'),('G4', 'quarter')

$$\begin{pmatrix}
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.5 & 0.0 \\
0.0 & 0.0 & 0.5 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.5 \\
0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0
\end{pmatrix}$$

Gen 3



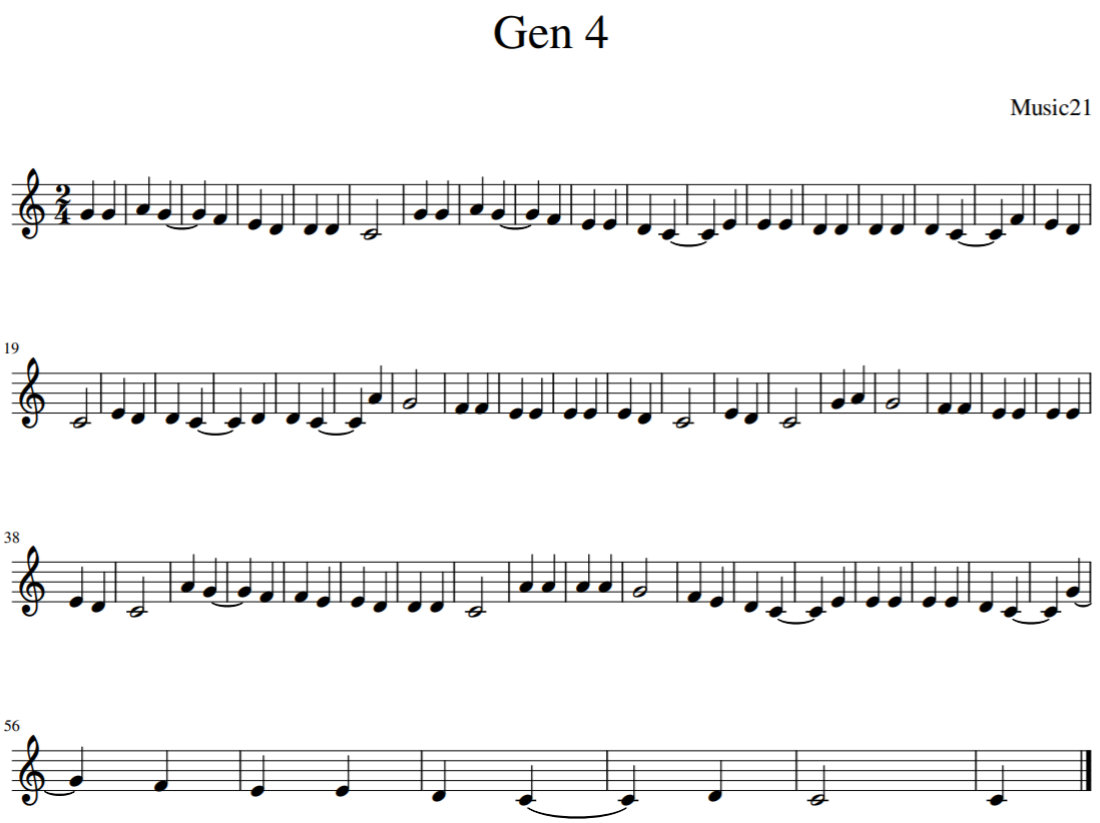


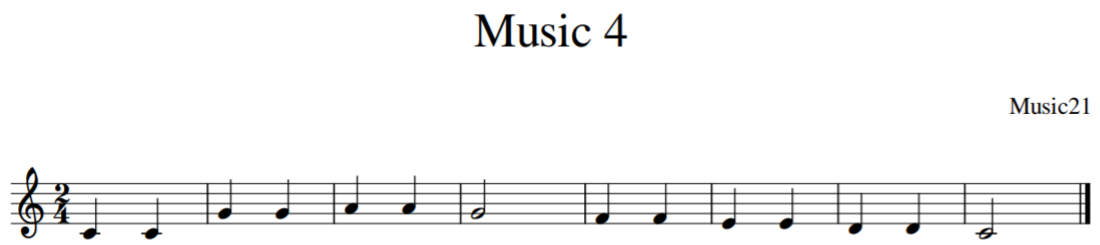
这里的生成的结果和上面的加伏特舞曲很接近，比较令人满意。重复片段较多，像是一种基于原曲的展开。

6.1.4 Music 4 小星星变奏曲

('A4', 'quarter'),('C4', 'half'),('C4', 'quarter'),('D4', 'quarter'),('E4', 'quarter'),('F4', 'quarter'),('G4', 'half'),('G4', 'quarter')

$$\begin{pmatrix} 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.142857 & 0.0714285 & 0.142857 & 0.142857 & 0.142857 & 0.142857 & 0.0714285 & 0.142857 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.5 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 \end{pmatrix}$$





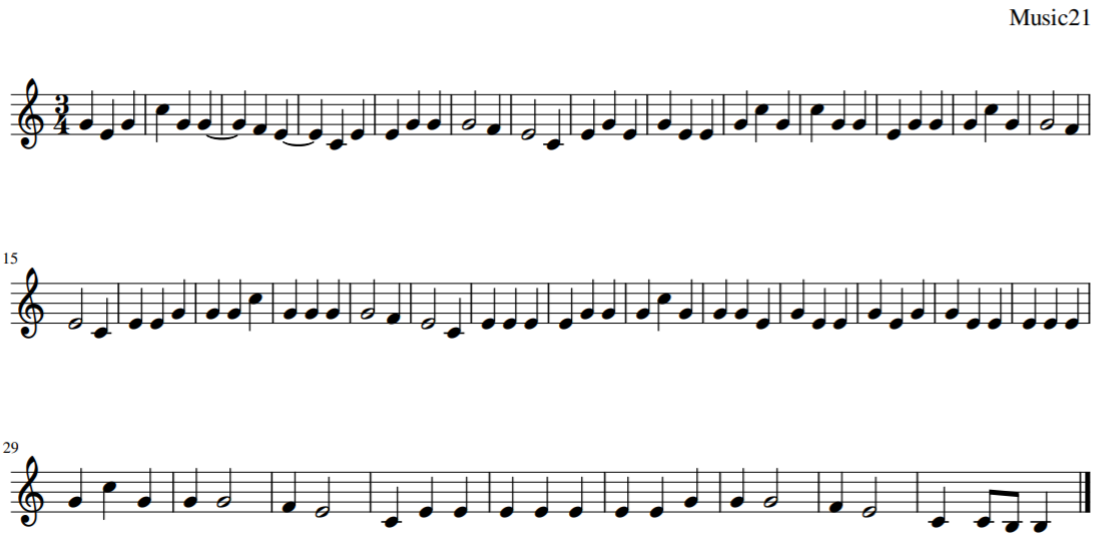
这里的生成结果与原曲有一点很相似：相同双音的大量重复使用。除此之外，该生成曲主要是连续式的旋律下行与跳跃式的旋律上行，这一点与原曲还是很相似的。

6.1.5 Music 5 C 大调奏鸣曲

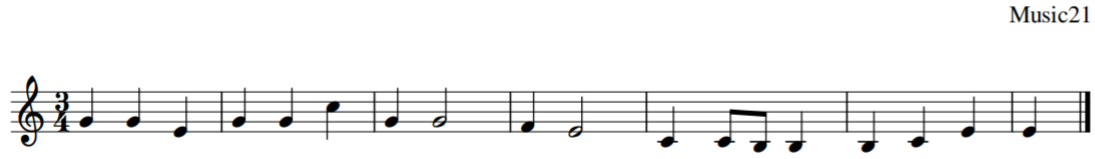
('B3', 'quarter'),('B3', 'eighth'),('C4', 'quarter'),('C4', 'eighth'),('E4', 'half'),('E4', 'quarter'),('F4', 'quarter'),('G4', 'half'),('G4', 'quarter'),('C5', 'quarter')

$$\begin{pmatrix}
0.5 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 & 0.5 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.2 & 0.4 & 0.2 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0
\end{pmatrix}$$

Gen 5



Music 5



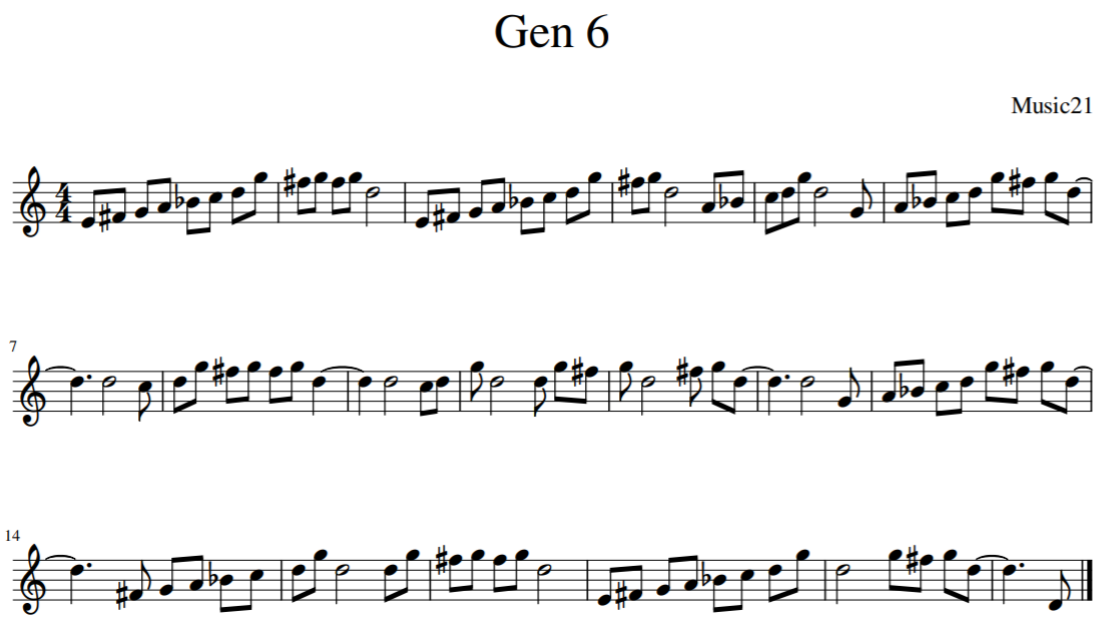
这里的生成结果与原曲相似性较低，更多的就是在 C 大三和弦上（即 C-E-G）的各种推进与跳跃，这是因为我们所选取的片段本身就是在 C 大三和弦的展开。但整体旋律和海顿的明亮、欢快风格十分接近。

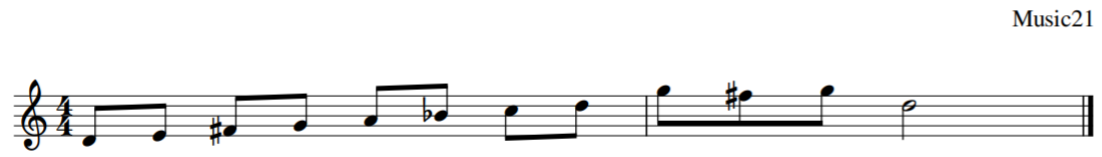
6.1.6 Music 6 六月船歌

('A4', 'eighth'),('B-4', 'eighth'),('D4', 'eighth'),('E4', 'eighth'),('F#4', 'eighth'),('G4', 'eighth'),('C5', 'eighth'),('D5', 'half'),('D5', 'eighth'),('F#5', 'eighth'),('G5', 'eighth')

表 2: matrix 6

0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.17
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0





此处的生成结果虽然和原曲十分接近，虽然大量用到了柴可夫斯基作曲中的行进动机，但整体上并不太符合音乐行进的规则，听起来比较奇怪，且几乎全部以 D 音进行收尾。这主要是因为我们选取的片段过短。这一句旋律本身听起来十分优美，但生成结果并不好。

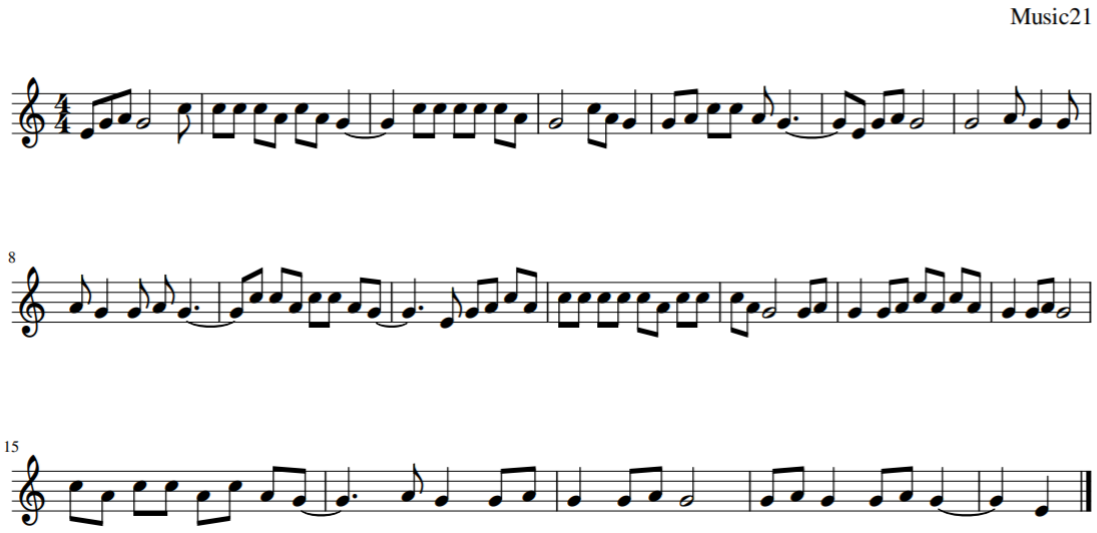
通过转移矩阵可以看到，该转移矩阵过于稀疏，且不同的块相似度过高，因此生成效果并不好。

6.1.7 Music 7 茉莉花

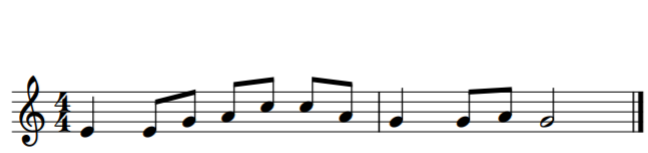
('A4', 'eighth'),('E4', 'quarter'),('E4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth'),('C5', 'eighth')

$$\begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.333 & 0.333 & 0.0 & 0.333 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.273 & 0.0909 & 0.0909 & 0.0909 & 0.0901 & 0.182 & 0.182 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 \end{pmatrix}$$

Gen 7



Music 7



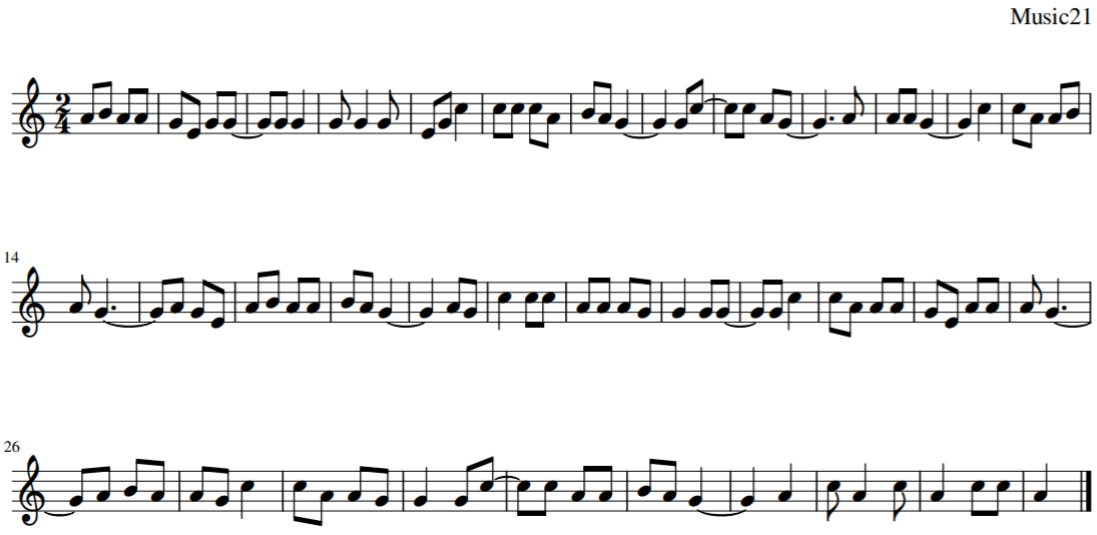
这一首的生成结果，我们主观评测上是 8 首中最好的。这首的生成结果是中国五声调式下的旋律优美的乐曲，与前面的相比更加自然，虽然也有《茉莉花》原曲的特点与明显的旋律走向，但听起来行进更加流畅。

6.1.8 Music 8 童年

('A4', 'quarter'),('A4', 'eighth'),('B4', 'eighth'),('E4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth'),('C5', 'quarter'),('C5', 'eighth')

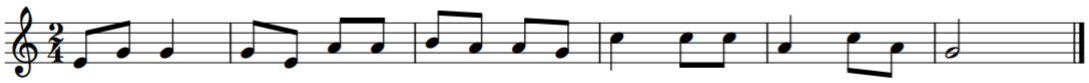
$$\begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.4 & 0.2 & 0.0 & 0.2 & 0.0 & 0.2 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.056 & 0.278 & 0.056 & 0.111 & 0.056 & 0.056 & 0.167 & 0.056 & 0.167 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.333 & 0.0 & 0.333 & 0.0 & 0.333 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.333 & 0.333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.333 \end{pmatrix}$$

Gen 8



Music 8

Music21



此处的生成结果旋律也和《童年》原曲有很大相似之处，且“AABA”四音动机经常出现，但有很多地方单音的重复过多，不过旋律的走向还是很符合原曲的活泼与欢快感，且符合一般乐曲的行进方式。

6.2 双原始音乐的实验分析

在双原始音乐部分，我们选择了本来较为接近的音乐片段组合生成了一些结果。

6.2.1 Music 1&4

('A3', 'quarter'),('B3', 'quarter'),('B3', 'eighth'),('A4', 'quarter'),('C4', 'half'),('C4', 'quarter'),('C4', 'eighth'),('D4', 'quarter'),('E4', 'half'),('E4', 'quarter'),('F4', 'quarter'),('G4', 'half'),('G4', 'quarter'),('C5', 'quarter')

Gen 1&4

Music21

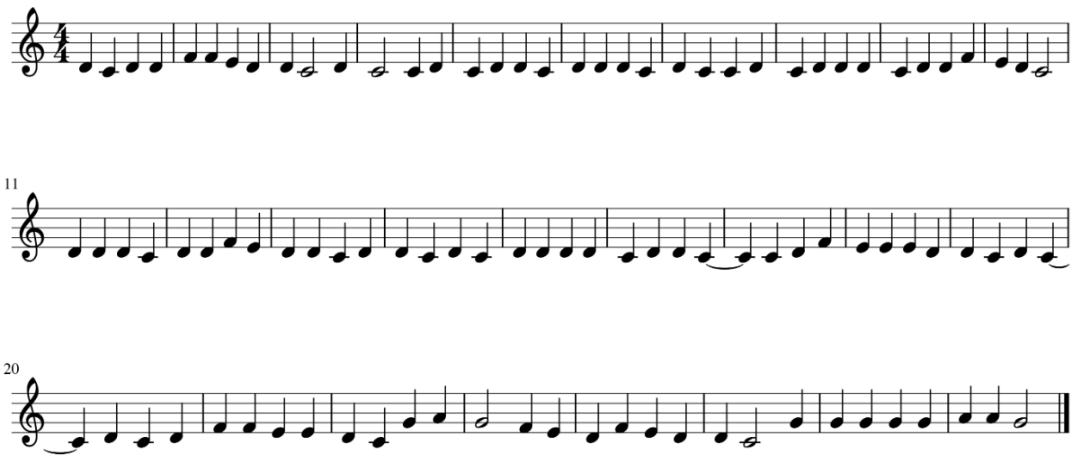
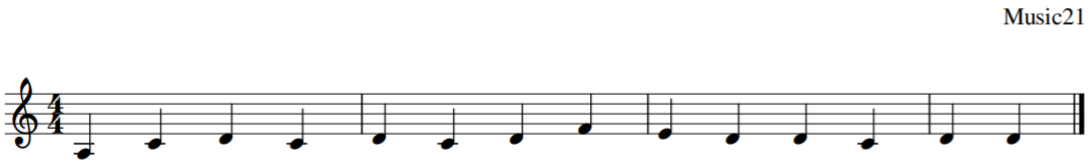


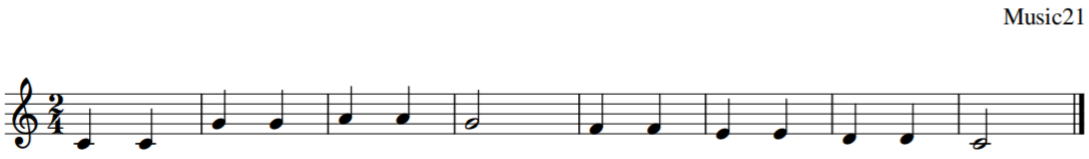
表 3: matrix 1&4

0.5	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.1	0.2	0.4	0.0	0.2	0.0	0.0	0.1	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.111	0.333	0.0	0.333	0.0	0.0	0.111	0.0	0.111	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2857	0.0	0.4286	0.0	0.0	0.2857	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.2	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0833	0.0	0.0	0.0	0.0	0.0	0.1667	0.0	0.1667	0.4167	0.1667
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

Music 1



Music 4



这里的生成结果，在旋律走向上更符合格里高利圣咏，即起伏较少，非常平稳；在音符的组

合上更像星星变奏曲，常常有双音重复。

6.2.2 Music 1&5

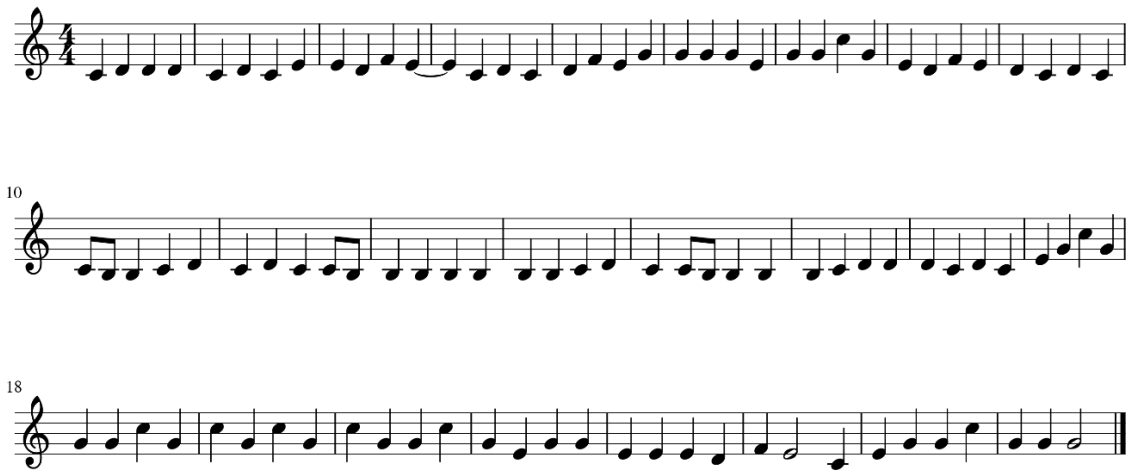
('A3', 'quarter'),('B3', 'quarter'),('B3', 'eighth'),('C4', 'quarter'),('C4', 'eighth'),('D4', 'quarter'),('E4', 'half'),('E4', 'quarter'),('F4', 'quarter'),('G4', 'half'),('G4', 'quarter'),('C5', 'quarter')

表 4: matrix 1&5

0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.1667	0.667	0.0	0.1667	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.5	0.0	0.333	0.0	0.0	0.1667	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.333	0.0	0.333	0.0	0.0	0.333	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	0.4	0.2
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

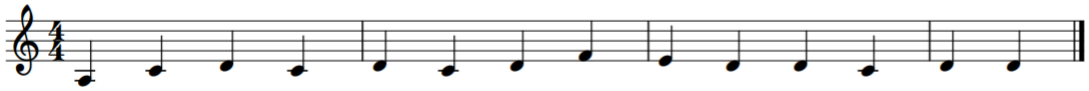
Gen 1&5

Music21



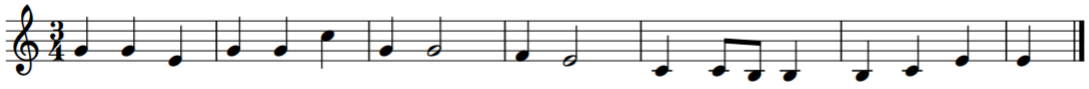
Music 1

Music21



Music 5

Music21



这里的生成结果，前半部分更像旋律舒缓、起伏不大的格里高利圣咏，但后面有较多的跳跃，更符合海顿的作曲方式。

6.2.3 Music 2&7

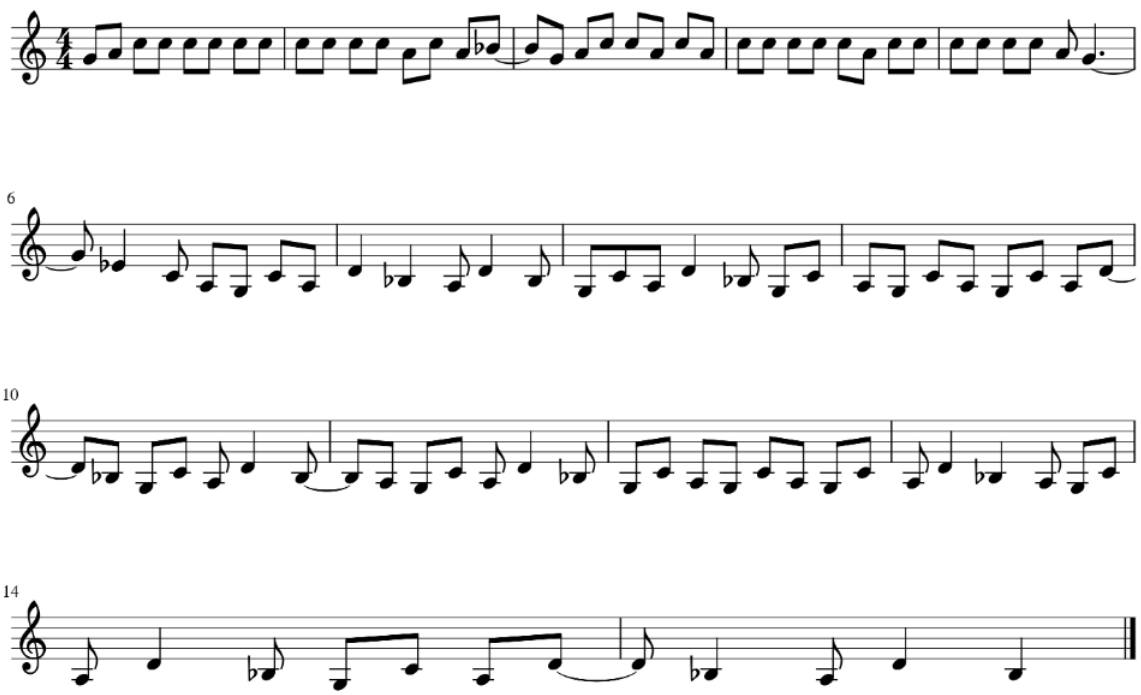
('A3', 'eighth'),('B-3', 'quarter'),('B-3', 'eighth'),('G3', 'eighth'),('A4', 'quarter'),('A4', 'eighth'),('B-4', 'quarter'),('B-4', 'eighth'),('C4', 'eighth'),('D4', 'quarter'),('D4', 'eighth'),('E4', 'quarter'),('E4', 'eighth'),('E-4', 'quarter'),('F#4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth'),('C5', 'eighth')

表 5: matrix 2&7

0.0	0.0	0.0	0.333	0.0	0.0	0.0	0.0	0.0	0.667	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.25	0.0	0.25
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0
0.0	0.0	0.0	0.0	0.0	0.667	0.0	0.0	0.0	0.0	0.333	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5

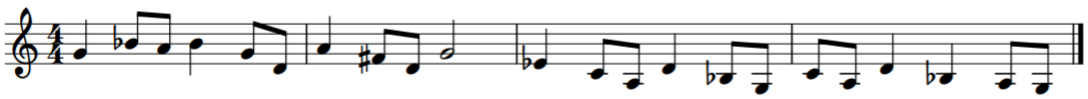
Gen 2&7

Music21



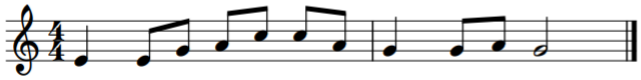
Music 2

Music21



Music 7

Music21



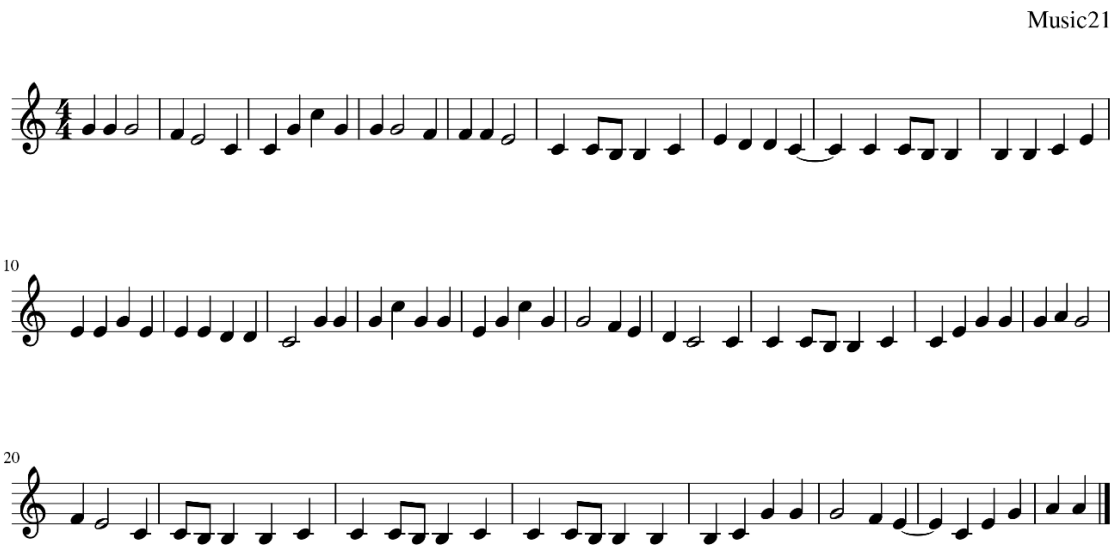
这部分的生成结果比较“分裂”，一部分很接近茉莉花，另一部分很接近加伏特舞曲，在一些单音有重合的地方会进行转换。茉莉花和加伏特舞曲的特点风格区别较大，因此叠加后效果并不理想。

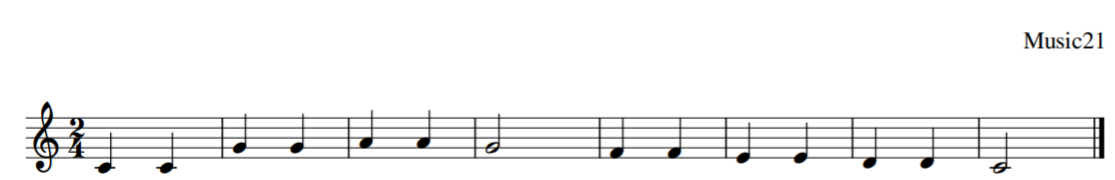
6.2.4 Music 4&5

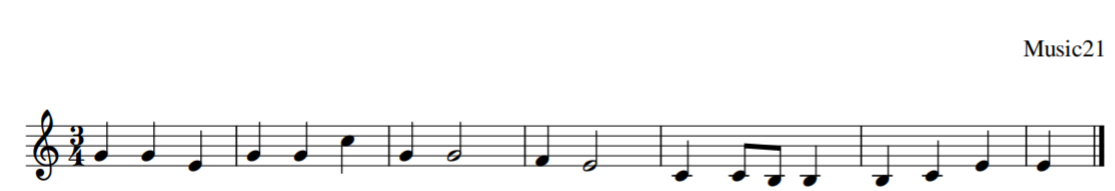
('B3', 'quarter'),('B3', 'eighth'),('A4', 'quarter'),('C4', 'half'),('C4', 'quarter'),('C4', 'eighth'),('D4', 'quarter'),('E4', 'half'),('E4', 'quarter'),('F4', 'quarter'),('G4', 'half'),('G4', 'quarter'),('C5', 'quarter')

表 6: matrix 4&5

0.5	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0
0.0645	0.0323	0.0645	0.0323	0.129	0.0323	0.0645	0.0323	0.1613	0.0968	0.0645	0.1925	0.0323
0.0	0.0	0.0	0.0	0.25	0.25	0.0	0.0	0.25	0.0	0.0	0.25	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.0	0.5	0.0	0.0	0.25	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.333	0.333	0.333	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.1429	0.0	0.0	0.0	0.0	0.0	0.1429	0.0	0.1429	0.4286	0.1429
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0







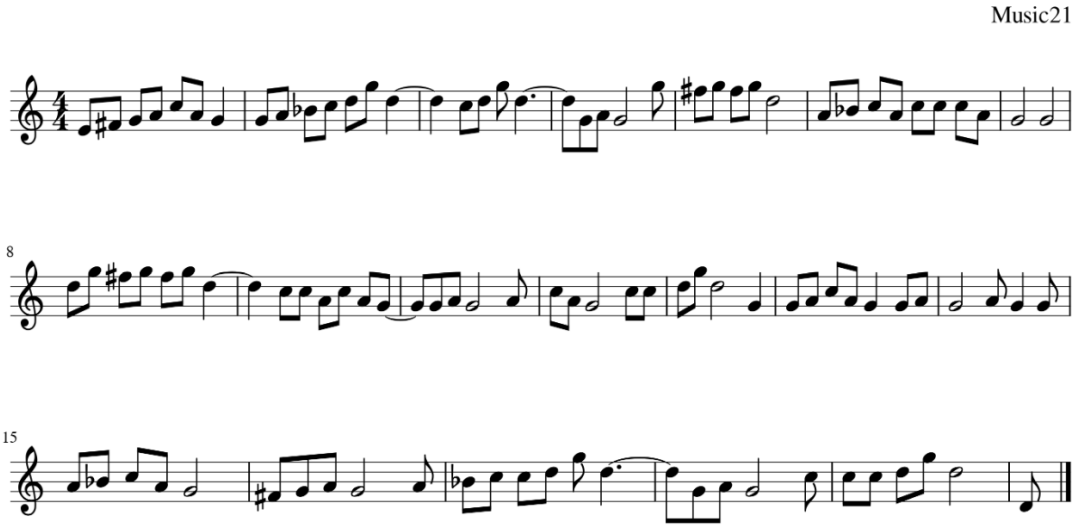
6.2.5 Music 6&7

('A4', 'eighth'),('B-4', 'eighth'),('D4', 'eighth'),('E4', 'quarter'),('E4', 'eighth'),('F#4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth'),('C5', 'eighth'),('D5', 'half'),('D5', 'eighth'),('F#5', 'eighth'),('G5', 'eighth')

表 7: matrix 6&7

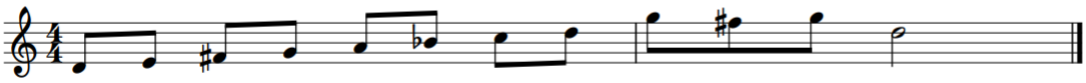
0.0	0.25	0.0	0.0	0.0	0.25	0.25	0.0	0.25	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.1818	0.0455	0.0455	0.091	0.0455	0.0455	0.0455	0.1364	0.1364	0.0455	0.0455	0.0455	0.091
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.333	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.333	0.0	0.333	0.0	0.0
0.1818	0.0455	0.0455	0.091	0.0455	0.0455	0.0455	0.1364	0.1364	0.0455	0.0455	0.0455	0.091
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0

Gen 6&7



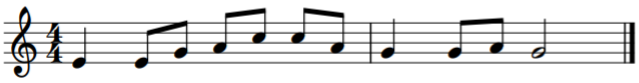
Music 6

Music21



Music 7

Music21



这里的生成结果非常自然，感觉上很符合小星星变奏曲和海顿奏鸣曲的结合体。这是因为两首曲子风格比较接近，都是很典型的维也纳派古典乐曲，都是用 C 大调下最基础的 C 大三和弦进行。

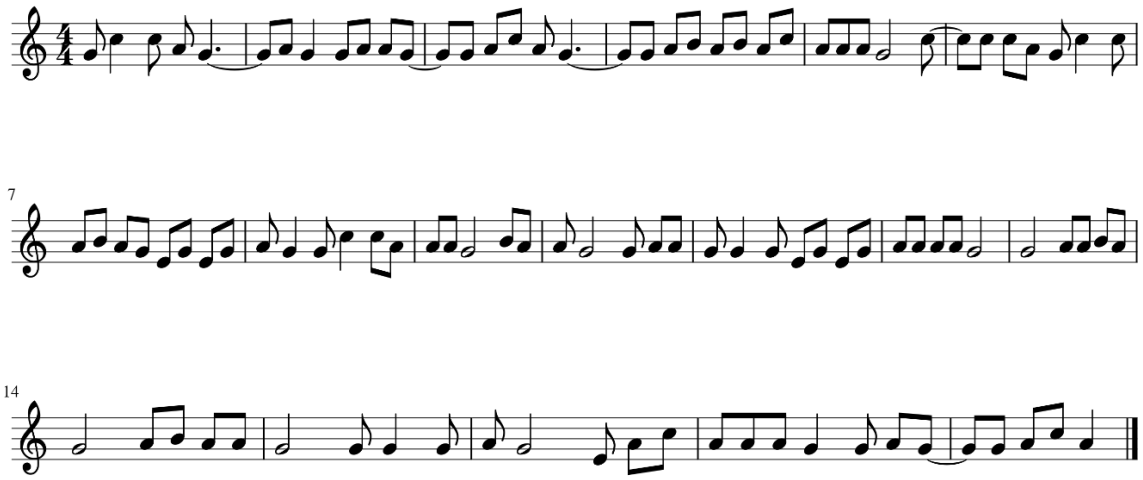
6.2.6 Music 7&8

('A4', 'quarter'),('A4', 'eighth'),('B4', 'eighth'),('E4', 'quarter'),('E4', 'eighth'),('G4', 'half'),('G4', 'quarter'),('G4', 'eighth'),('C5', 'quarter'),('C5', 'eighth')

$$\begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.25 & 0.125 & 0.0 & 0.25 & 0.125 & 0.125 & 0.0 & 0.125 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.667 & 0.0 & 0.0 \\ 0.0357 & 0.2857 & 0.0357 & 0.1071 & 0.0714 & 0.0714 & 0.1786 & 0.0357 & 0.1786 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.4 & 0.0 & 0.2 & 0.0 & 0.2 & 0.0 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.2 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 \end{pmatrix}$$

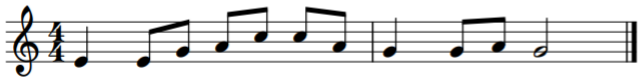
Gen 7&8

Music21



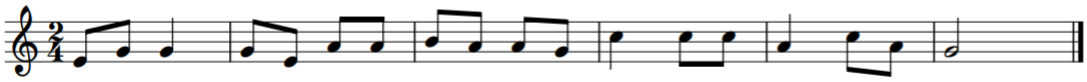
Music 7

Music21



Music 8

Music21



这里的生成结果也很好，茉莉花是中国传统民歌，童年也是在中国五声调式下的流行乐（第一句中只多了一个 B 作为点缀，但也是该曲的重要动机 AABA），因此两者的叠加效果很好，生成的乐曲中除了茉莉花的优美，也有童年中短促、欢快的音符组合，因此虽然旋律比较悠扬，但多了更多的短音点缀，更加活泼悦耳。

7 结论

实验效果表明，在较短的乐曲片段下使用马尔可夫链生成的随机乐曲效果表现参差不齐，相比较而言，旋律简单且直接使用常用和弦的乐曲片段效果更好，而转移矩阵过于稀疏的效果并不好。风格更接近的两种乐曲叠加后生成的音乐效果也更好，但有时调性接近、风格不同的音乐叠加后可能会产生较有趣的结果。

参考文献

[1] F. P. Brooks, A. L. Hopkins, P. G. Neumann, and W. V. Wright. An experiment in musical composition. *IRE Transactions on Electronic Computers*, EC-6(3):175–182, 1957.

[2] Michael Cuthbert. music21 documentation. `http://web.mit.edu/music21/doc/`.

[3] jcbozonier. Markovmusic. `https://github.com/jcbozonier/MarkovMusic`.

[4] Iannis Xenakis. Formalized music: Thought and mathematics in composition . hillsdale, ny. *Pendragon Press. Haworth, C.(2015). Sound Synthesis Procedures as Texts: An Ontological Politics in Electroacoustic and Computer Music. Computer Music Journal*, 39(1):41–58, 1992.