

音乐与数学-小组报告

蔡鹏霖 张润博 王翊霏 陈平 颜廷恺 于泮林
元培学院 元培学院 数学科学学院 数学科学学院 物理学院 地球与空间科学学院
选题：机器作曲 · 遗传算法

1 题目概述

机器作曲历来一直是人工智能领域中一个极具吸引力的课题。自从 20 世纪 50 年代开始，就有学者开始研究机器作曲，并提出了各种算法和技术。在过去的几十年里，随着计算机硬件和软件技术的快速发展，机器作曲技术也得到了长足的发展。

前人在机器作曲领域做了很多工作，他们提出了许多算法和技术，如基于规则、神经网络、遗传算法等。随着时间的推移，这些方法不断被改进和优化，使得机器作曲的效果越来越好。尽管如此，机器作曲仍然需要进一步的研究和发展，才能实现真正的智能作曲，创造出完美的音乐。

作为研究题目，我们在 Python 编程平台上建立了使用遗传算法进行机器作曲的模型。

小组分工 全组成员共同讨论了适应度函数的定义与算法流程，陈平、颜廷恺进行了只含音高的一维音乐代码尝试，王翊霏、于泮林进行了同时含有音高和时值的二维音乐代码尝试，最后由蔡鹏霖、张润博汇总补充、确定整体代码框架，并撰写报告。

2 遗传算法流程及实现

2.1 音符编码方式

本题中，我们考虑音符的两种基本要素——音级和时值，因此下面将分别考虑音级序列和时值序列两个维度，由此组合成一段旋律。

一方面，考虑乐音体系 $\{F_3, \sharp F_3, \dots, B_3, C_4, \sharp C_4, \dots, B_4, C_5, \sharp C_5, \dots, \sharp F_5, G_5\}$ ，把集合 $S = \{53, 54, \dots, 79\}$ 作为我们纳入考量的音级的全集，其中从 53 至 79 的连续整数与乐音体系形成一一对应；另一方面，我们考虑常见的时值集合 $T = \{4, 2, 1, \frac{1}{2}, \frac{2}{3}\}$ ，五种时值分别对应全音符、二分音符、四分音符、八分音符和三连音中的一个。

在上述定义下，音级序列可以用一个由集合 S 中元素组成的列表表示，时值序列可以用一个由集合 T 中元素组成的列表表示。两个集合的笛卡尔积构成了全部可能的乐音音符，因而最终生成的序列是由集合 $S \times T$ 中元素组成的列表。

	60	63	65	67	68	70	72	73	75
60	0	0	1	0	0	0	0	0	0
63	0	0	0	1	0	0	0	0	0
65	0	$\frac{1}{3}$	0	$\frac{2}{3}$	0	0	0	0	0
67	0	0	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	0	0
68	0	0	0	$\frac{1}{4}$	0	$\frac{3}{4}$	0	0	0
70	0	0	$\frac{1}{5}$	0	$\frac{2}{5}$	0	$\frac{1}{5}$	$\frac{1}{5}$	0
72	0	0	0	0	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$
73	0	0	0	0	0	0	1	0	0
75	0	0	0	0	0	0	0	1	0

表 1: 示例音级片段对应的转移概率矩阵

2.2 从指定音乐片段中计算（音级的）转移概率矩阵

我们采用把音级序列看作一条 Markov 链的基本假设，并设该 Markov 链是时齐的，由此可以计算转移概率矩阵 [3]。例如，对于给定的一个片段：

```
initial_list = [60, 65, 67, 68, 70, 68, 67, 65, 63, 67, 70, 73, 72, 68, 70, 72,
               75, 73, 72, 70, 68, 70, 65, 67]
```

可以对片段中的音级作如下统计（限于篇幅，这里仅展示不全为 0 的行和列），如表 1 所示。

转移概率矩阵 P 中第 i 行第 j 列的元素 P_{ij} 表示前一个音级为 i 时，后一个音级为 j 的概率 $\Pr(s_{t+1} = j | s_t = i)$ 。

本组成员认为片段中相邻两个音符的时值之间不具有强相关性，故时值序列不宜按 Markov 链进行处理。因此，我们采用依概率分布随机选取时值的方式，并加以适应度函数的约束，用以指导旋律进化的方向。这里，我们探究了依不同概率分布选取时值的方式对最终旋律的影响（以适应度作为指标，适应度函数的定义见 2.7 节），结果如图 1 所示。

2.3 初始种群的生成

我们对种群初始化为具有 10 个个体的列表，每个个体的时值是元素总和为 16（四小节， $\frac{4}{4}$ 拍）的随机序列；每个个体的音级都是以给定片段初始音作为初始音、依照转移概率矩阵随机产生的音级序列，且音级序列与时值序列中元素个数相等。

2.4 交叉遗传操作

遗传算法中，对亲本的交叉操作来源于生物体中同源染色体的交叉互换。具体实现上，我们随机选取两段（等长）亲本序列的位点 A、B，并交换两段序列在位点 A、B 之间的部分，从而产生两个新个体。

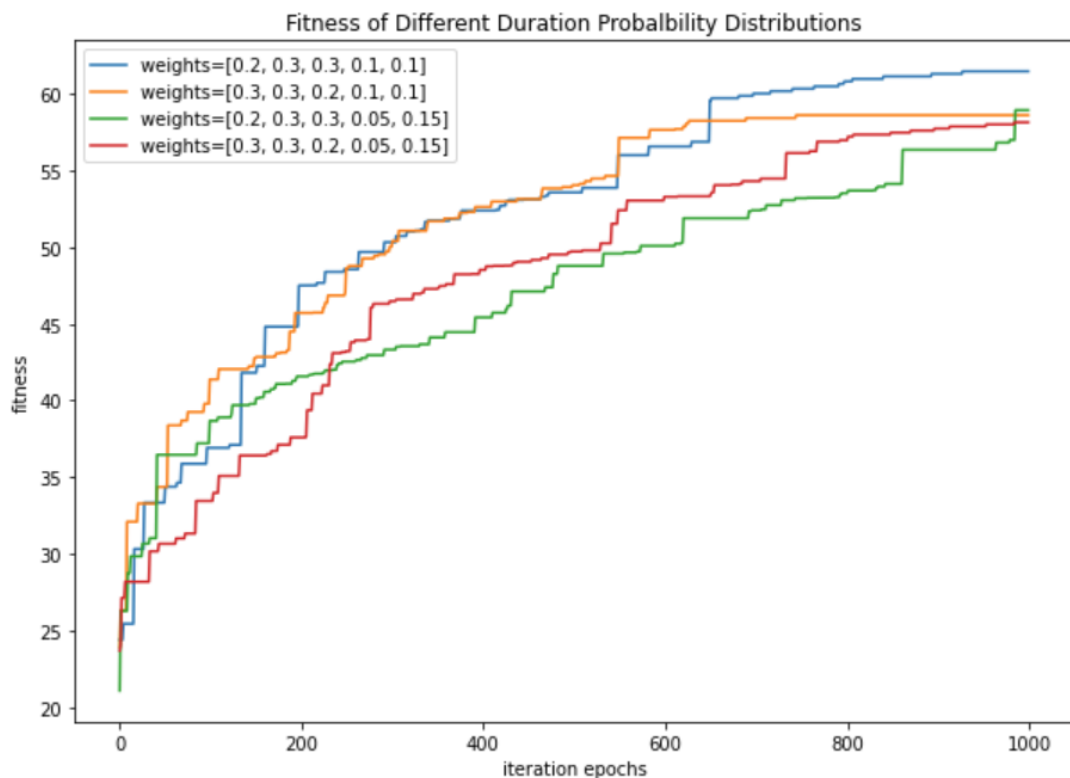


图 1: 不同概率分布选取时值的方式对适应度的影响

2.5 变异操作

设置一个概率阈值 $\text{threshold} \in (0, 1)$ 。遍历音级序列中的所有位点，当随机数超过概率阈值 threshold 时（意即以 threshold 的概率值进行变异），就将此位点进行变异。变异的结果仍然遵循转移概率矩阵，即根据前一位点的音级和前一位点对应的条件概率分布，随机选取当前位点的新音级。不同 threshold 值对适应度函数计算结果的影响参见图2。

对于时值序列同样根据概率阈值 threshold 进行变异操作，需要根据实际情况进行截断、补全等处理，具体实现可以参考源代码。

2.6 移调、倒影、逆序

2.6.1 移调

在给定范围内随机产生一个偏移值 N ，对给定音乐片段（包括音级与时值）中的音级部分整体升高或降低 N 个音高。

2.6.2 倒影

对于给定音乐片段，以第一个音为基准（或另选取对称音），进行竖直方向的对称变换。

2.6.3 逆序

将给定音乐片段的音级与时值序列进行逆序操作。

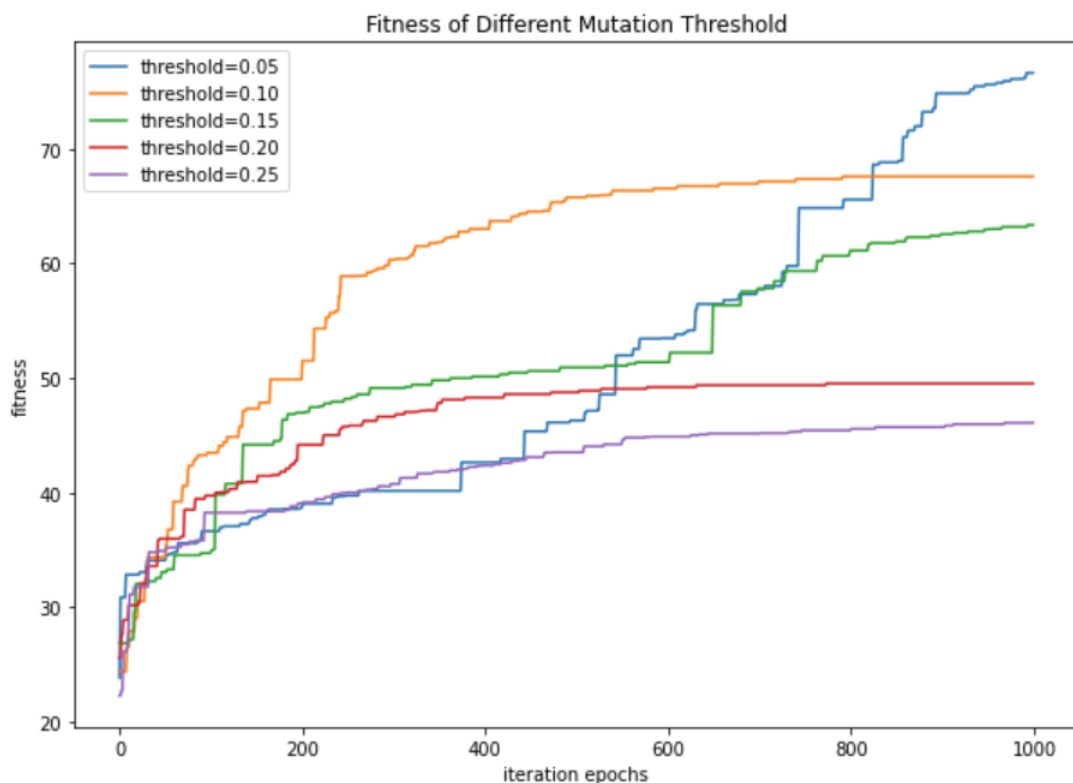


图 2: 不同 threshold 阈值选取对适应度的影响

2.7 适应度函数的定义及选取

适应度函数的定义分为音级与节奏型两个考察部分。

对于音级，我们考虑序列中相邻两个音符构成的音程，根据音程的协和程度，对不同的音程关系赋予不同的分值。评分方式如表2所示。

相邻两个音符构成的音程	评分
大三度、纯四度、纯五度	2
纯一度、纯八度、小三度、大六度	1
增四度、小六度	0
小二度、大二度、小七度、大七度	-1
超过纯八度	-2

表 2: 对于相邻两个音符构成的音程的评分

通过遍历音级序列，计算相邻两个音符间的评分，得到音级评分列表 $C = \{c_1, c_2, \dots, c_{n-1}\}$ ，其中 n 是序列中音符的数量。相应地，我们可以计算出音级评分列表的均值

$$\bar{c} = \frac{1}{n-1} \sum_{i=1}^{n-1} c_i$$

与方差

$$\text{Var}(C) = \frac{1}{n-1} \sum_{i=1}^{n-1} (c_i - \bar{c})^2$$

对于节奏型的评分，由以下三部分构成：

- 节奏型的密集度评分 m_1 。我们不希望在有限的总时长（四小节， $\frac{4}{4}$ 拍）内出现过多或过少的音符数量，故对于节奏型密集度评分的定义为

$$m_1 = \frac{L - ||D| - \frac{1}{2}L|}{L}$$

其中 L 为总时长（本题中其值为 16）， $D = \{d_1, d_2, \dots, d_n\}$ 为时值序列。

- 根据三连音的情况给出连音评分 m_2 。这里采用简化的形式：如果出现三连音，则 $m_2 = 1$ ；否则 $m_2 = 0$ 。
- 根据节拍的合理性计算 m_3 。 $m_3 \in \{0, 1, 2, 3, 4\}$ 表示 4 个小节中未出现跨小节音符的节数 [1]。对每个小节，如果该小节的末位音符在本小节结束时终止，则称该小节“未出现跨小节音符”。

节奏型的总评分 $M = m_1 + m_2 + m_3$ 。

综合音级评分（列表）与节奏型评分，我们最终定义如下适应度函数 [2]：

$$\text{fitness} = \alpha \bar{c} + \beta e^{-\text{Var}(C)} + \gamma M$$

其中 α, β, γ 为权重参数，本项目中选取 $\alpha = 5, \beta = 1, \gamma = 3$ 。

2.8 迭代过程

采用如下所示的算法。

Algorithm 1: 遗传算法

Data: 输入给定旋律序列 I

- 1 根据序列 I ，计算转移概率矩阵 P
- 2 依据 P ，初始化初始种群 $Q = \{x_1, x_2, \dots, x_N\}$ ，其中 N 是初始个体总数
- 3 **foreach** $i = 1, 2, \dots, ngen$ ($ngen$ 为迭代次数) **do**
- 4 计算每个个体的适应度 $f(x_1), f(x_2), \dots, f(x_N)$
- 5 根据轮盘赌算法 [3]，选出用于遗传的两个亲本 x_i, x_j
- 6 亲本 x_i, x_j 经交叉、变异得子代 y_1, y_2
- 7 亲本 x_i, x_j 经倒影变换得子代 y_3, y_4
- 8 亲本 x_i, x_j 经逆序变换得子代 y_5, y_6
- 9 亲本 x_i, x_j 经移调变换得子代 y_7, y_8
- 10 $Q \leftarrow Q \cup \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$
- 11 **end**
- 12 **while** 当前种群中个体数量超过规定数量 **do**
- 13 去除种群中适应度最低的个体
- 14 **end**

Result: 输出现存种群中适应度最高的个体

2.9 实验结果

经过上述迭代过程，最终得到如下的生成旋律（样例）：

音级序列: [63 58 63 67 63 59 54 50 55 60 60 60 56 51 47 52 56 49 53 49 54 61 65
72 67 60 64 57 62 57]

时值序列: [4.0 2.0 1.0 1.0 4.0 4.0
 2.0 2.0 1.0 0.66666667 0.66666667 0.66666667
 0.5 1.0 2.0 1.0 2.0 0.5
 2.0 4.0 4.0 4.0 4.0 4.0
 2.0 1.0 1.0 2.0 2.0 2.5]

上述旋律对应的曲谱如图3所示。



图 3: 输出旋律样例

3 总结

通过这次研究题目,我们对基于遗传算法的机器作曲流程有了更进一步的认识,加强了编程能力和音乐鉴赏能力。

纵观整个技术,虽然机器作曲技术已经取得了许多进步,但这个领域仍然存在很多挑战和困惑。其中一个最大的难点是如何让机器创造出优美、富有感染力的音乐。此外,机器作曲还需要考虑到音乐理论、文化背景等方面的知识,以及如何表达情感和创意。诸多问题,有待后面继续去探究。

参考文献

- [1] Majid Farzaneh and Rahil Mahdian Toroghi. "GGA-MG: Generative Genetic Algorithm for Music Generation". In: *arXiv preprint arXiv:2004.04687* (2020).
- [2] Dragan Matić. "A genetic algorithm for composing music". In: *Yugoslav Journal of Operations Research* 20.1 (2010), pp. 157–177.
- [3] 王杰. 音乐与数学. Peking University Press, 2019.