

APPLICATION RESEARCH OF GENETIC ALGORITHMS ON THE MELODY COMPOSITION

**A Dissertation Submitted
to the Graduate School of Henan Normal University
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering**

By

Liu Yanmei

Supervisor: Prof. Cao Xizheng

April, 2014



摘要

随着计算机技术的不断发展,自动化技术已被广泛应用于旋律创作领域。算法作曲的出现,使得旋律创作变得容易、方便。作为算法作曲的方法之一,遗传算法能够灵活地产生各种不同风格的旋律。但是,应用遗传算法进行旋律创作时,适应度函数的构建仍是一大挑战。以往的适应度函数大多数很主观而且容易受设计者的偏见影响,评估过程耗费时间长且效率低下。本文着重研究将遗传算法应用于旋律创作时适应度函数的设计及其他相关问题。

首先,分析了将遗传算法应用于旋律创作时生成初始群体常用的几种人工智能方法,包括数学模型的方法、人工神经网络的方法、音乐文法的方法等。为了实时产生旋律,本文中采用基于数学模型的方法来生成初始群体。并且为了避免生成的群体过度随机,在生成初始群体时对相关旋律参数进行了限制,如调号、拍号等。其次,设计了一种新的旋律编码方式,即以旋律中的每个音符为单位,基于该音符的音级、时值和符号对旋律进行编码。该编码方法可以直观形象地表现旋律的信息,而且能够保证旋律信息的正确性。再次,构建了旋律特征提取器,定义了用于自动评价的多目标适应度函数。特征提取器用于提取旋律乐谱的特征,如旋律中各音符的自相似性,旋律的图形,音符的线性度,所使用的音域,等等。将这些特征进行线性组合作为多目标适应度函数,对所有生成的旋律个体进行适应度计算,根据计算得到的适应度函数值决定个体是否需要进化。然后,对进化过程中的交叉和变异操作作了研究,以使旋律的进化能够顺利进行。交叉操作时着重考虑了音符的时值修正问题,确保交叉前后每一小节中音符的时值之和保持不变。执行变异操作时,对于音级的变异,保证相邻音符的音程不超过8度。最后,搭建了基于多目标适应度函数的遗传算法旋律创作实验系统平台。实验结果表明,该系统在确保生成具有较高质量的旋律的前提下,缩短了评估所用的时间,大大提高了旋律创作的效率。

关键词: 算法作曲, 遗传算法, 旋律创作, 适应度函数

ABSTRACT

With the continuous development of computer technology, automation technology has already widely used in the field of melody composition. The occurrence of algorithm composition makes the melody composition easier and more convenient. As one of methods of algorithm composition, genetic algorithm can flexibly generate melodies with varying styles. However, when applied genetic algorithm to the melody composition, the creation of fitness function remains a challenge. Most of those previous fitness functions are subjective and can be affected by the designer's biases, the evaluation process is very time-consuming and with low efficiency. This paper emphasizes the research of the design of fitness function and some related questions when genetic algorithms are applied to compose melody.

Firstly, several artificial intelligence methods which are often used to create initial population when the genetic algorithm is applied to compose melody have analyzed, including mathematical model method, artificial neural network method, music grammar method and so on. In order to produce melody in real time, this paper employs the mathematical model method to create initial population. And with the purpose of avoiding the excessively randomization of the generated population, impose some restrictions on the related melody parameters when creating initial population, such as the key signature, time signature and so on. Secondly, a new method of encoding the melody has devised, that is to say, in the every note of the melody, encode the melody that based on the scale, value and symbol of the note. This method represents the melody's messages intuitively and vividly, and can make sure the correctness of the melody's messages. Thirdly, the feature extractors of melody have established, and the multi-objective fitness function which used to evaluate automatically has defined. The feature extractors are used to extract the melody score's features, such as the self-similarity of every note of the melody, the shape of the melody, the linearity of the note, the used pitch range and so on. These characteristics are linearly combined as the multi-objective fitness function, do fitness computation for all the generated melody individuals, determining whether the individual needs the evolution operator depending on the calculated fitness function values. Then, some research of the crossover and mutation operator of the evolution process have done, to make sure the evolution of melody can go on smoothly. When carrying out the crossover operator, think more about the

value correction problem of the note to guarantee the sum of the value of note in every measure stay the same no matter before or after the crossover operator. When executing the mutation operator, for the mutation of the scale, the interval between the neighboring notes have to ensure that not exceeding 8 degrees. Finally, the genetic algorithm melody composition experiment system platform which based on the multi-objective fitness function has built. The experiment result shows that on the premise of the generated melody with higher quality, this system cut down the time of the assessment, improve the efficiency of the melody composition heavily.

KEY WORDS: algorithmic composition, genetic algorithm, melody composition, fitness function

目 录

摘 要.....	I
ABSTRACT.....	III
目 录.....	V
第一章 绪论.....	1
1.1 引言.....	1
1.2 研究现状.....	2
1.3 本文的主要工作.....	6
第二章 相关理论知识.....	7
2.1 遗传算法.....	7
2.1.1 遗传算法的基本思想.....	7
2.1.2 遗传算法的基础理论.....	7
2.1.3 遗传算法的基本概念.....	8
2.1.4 遗传算法的基本操作.....	10
2.2 相关旋律学基础理论.....	13
2.2.1 音.....	13
2.2.2 音阶.....	14
2.2.3 乐音体系.....	15
2.2.4 音程.....	16
2.2.5 节奏与节拍.....	16
2.2.6 调式.....	18
第三章 应用于旋律创作的遗传算法设计与实验.....	21
3.1 使用遗传算法创作旋律的步骤.....	21
3.2 编码策略.....	23
3.2.1 调号的编码.....	23
3.2.2 音级的编码.....	24
3.2.3 音名的编码.....	24

3.2.4 音符时值的编码.....	24
3.2.5 旋律的编码.....	24
3.3 初始群体的生成.....	25
3.4 适应度函数的构建.....	26
3.5 进化操作.....	30
3.5.1 交叉操作.....	30
3.5.2 变异操作.....	32
第四章 总结与展望.....	35
4.1 总结.....	35
4.2 研究展望.....	36
参考文献.....	37
致 谢.....	41
攻读学位期间的科研成果.....	43
独 创 性 声 明.....	45
关于论文使用授权的说明.....	45

第一章 绪论

1.1 引言

近几年,随着我国经济与科技的迅速发展,人民的物质生活水平在不断提高的同时,对精神生活的需求也在不断增加。其中,音乐作为一种直接的娱乐方式,进入到人们的日常生活中,并且成长十分迅速。但是,由于音乐具有很强的专业性,对于很多非音乐专业的人来说,大多只是聆听、欣赏音乐,很少进行创作。随着计算机技术的不断发展,自动化技术已经广泛应用于旋律创作领域^[1]。截止到目前,关于计算机作曲,最常见的定义是:“借助于计算机,让其产生包含有音符、音色、节奏等音乐元素的旋律,并且把生成的旋律演奏出来。”^[2-4]计算机作曲是计算创造力的一个挑战区域,已经引出各种解决音乐自动生成问题的方法。然而,由于音乐理论具有在某些方面很难形式化的复杂性问题,这使得产生有说服力的音乐变得困难。计算机作曲与知识表征、计算机智力、逻辑推理、模式识别等一系列的前沿问题密切相关^[5]。其中,核心内容是算法作曲。算法作曲的出现,使得旋律创作变得更加容易、方便,而且也使旋律创作变得更加多元化和大众化^[6]。

算法作曲(Algorithmic Composition),或称自动作曲(Automated Composition),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的介入程度达到最小的研究^[7-8]。虽然存在很多成功的方法,但是遗传算法(Genetic Algorithm,简称GA)为产生各种类型的音乐输出提供了最大的灵活性,这是由适应度函数的抽象性和基因组表现的普遍性决定的。Freitas 和 Guimaraes 演示了遗传算法如何完成这点^[9]。他们使用多个适应度函数为旋律制作和声,产生了两种类型的输出,这取决于加权更高的那一个适应度函数。利用两种不同的旋律风格(即简单性或者不和谐音)之一来产生有趣的和声。

遗传算法能够成功地应用于音乐领域,但是执行一个有效的适应度函数仍是一个挑战^[10],这是因为量化一首旋律的优劣在很大程度上仍然是主观的——这是该领域的一个主要障碍。目前两种最通用的方法是有人参与的方法和算法的方法。

交互式遗传算法(Interactive Genetic Algorithm, 简称IGA)吸纳人类输入作为适应度

函数但是遭遇了吞吐量问题——适应度瓶颈^[11]。当评价者聆听音乐时每次只听一首旋律。这一过程耗费的时间使得通过一个人类评价者来评定更多数量的旋律是不切实际的。自动评估的适应度函数增加了吞吐量,而且大部分已经实现的适应度函数的应用范围比较狭窄,从传统上对评价质量有所限制。不论适应度函数是被设计用来寻找特定的四声部和声规则^[12]还是全音阶音符,这个函数均限制了输出范围。鉴于有人参与和自动的适应度函数都有重大的缺陷,因此需要一个新的方法——一个避免适应度瓶颈并且允许使用更加灵活的方式以避免程序设计员偏见的方法。

本文基于遗传算法构建了一个旋律创作系统:解决这些挑战并且允许遗传算法通过使用一组特征提取器来灵活地产生旋律。单个的提取器分析和声、节奏的分布、旋律线的大体形状、自相似性、重复以及输出的其他方面。然后一个多目标的适应度函数使用加权的特征提取器的组合产生适应度分数。遗传算法根据这个分数来推动进化模型。

1.2 研究现状

国外对计算机作曲系统这方面的研究非常活跃,这一过程需要计算机专业人员与音乐家的密切合作,但是,目前我国这项技术仍然处于起步阶段。

11世纪时,关于旋律创作的形式化技术就已经出现。Guido d'Arezzo 构建了一种模型,该模型能够为一本与宗教相关的书籍中的每一个元音拟定出不同的音高^[13]。到15世纪时,已经能够在节奏均匀的圣歌中比较系统地运用节奏模式。到了文艺复兴时期以及巴洛克(Baroque)时代,为了能够对旋律进行对位操作,作曲家们制定了严密的规则,例如:为了创作复调音乐,对于给出的主题动机(也即是一个小的音乐片段),可以对其进行倒影、延长主题、削减系统化的过程等操作。

进入20世纪后,研究者们再一次使用了形式化技术。20世纪初期,Arnold Schonberg 介绍并引进了音列技术,之后,Anton Webern 等人又作了进一步的系列化工作,包括对音乐进行特征化操作,提取并且控制旋律中音符的音高、时值及音色等参数。然后,根据这些参数,按照次序挑选出可能的值形成音列。最后根据当前的音列或者是进行倒置或逆行操作之后的音列对参数值进行操作。在50年代初期,Iannis Xenakis 已经使用基于随机过程的方法手工地产生旋律片段,而没有借助于计算机,虽然数字计算机在那时已经出现并且开始应用于创作旋律。Lejaren Hiller 的弦乐四重奏 Illiac 组曲^[14]是最早的而且是全部由计算机产生的旋律,发表于1956年。

研究算法作曲的过程中，发现了以下几个重要问题：

- (1) 知识的表征问题，也即是如何将音乐的知识表现出来的问题，即在音乐的表层结构和深层的逻辑之间如何建立相互对应的关系；
- (2) 创造性问题，即基于听众的直觉、所具有的美学标准来判断自动生成的音乐是否具有创造性；
- (3) 旋律创作的风格问题。目前存在着很多模拟著名作曲家的创作风格或者模拟特定时期的旋律风格的系统，但这些系统是否具有广泛的适用性还有待研究，而且很难精确地定义旋律的创作风格；
- (4) 生成旋律的质量评估问题。在算法作曲的系统中，质量评估机制引导着创作方向，并最终决定旋律成功与否，是非常重要的部分。可以通过构建形式规则库来评估生成的旋律的质量。

遗传算法作为一种全局优化算法，基本思想是达尔文的进化论和孟德尔的遗传学说。自 20 世纪 40 年代起，生物模拟(也称仿生学)就已经成为计算科学的组成部分。1975 年，美国密歇根大学的霍勒德(J.H.Holland)教授首先提出了遗传算法的概念^[15]，并且发表了专著 *Adaptation in Natural and Artificial Systems*。遗传算法是一种随机搜索的最优化方法，具有隐并行性、智能性、多点搜索能力等特点。遗传算法使用适应度函数(Fitness Function)对候选者(染色体)进行进化，主要思路是将旋律看作自然界中生物的染色体，然后通过选择、交叉、变异等操作，形成新的染色体，之后根据适应度函数值判断哪些染色体能够幸存到下一代并进行下一阶段的进化。

遗传算法作曲是基于遗传算法并利用计算机来生成旋律^[16-18]。当进行旋律创作时，对于给定的旋律首先按照一定的方式进行编码，编码后的旋律称为染色体。之后使用遗传算法中的交叉、变异等算子来“进化”旋律，并且设计适应度函数进而用于评估进化结果的质量，重复此过程直至找到最终的满意解^[19]。使用遗传算法创作旋律时，适应度函数至关重要。可通过两种方式获得，一种是设计规则作为适应度函数，另一种是让人作为适应度函数，通常将这种方式称作交互式遗传算法，例如 Biles 的“GenJam”^[20-21]，这是一个关于爵士乐的交互即兴演奏系统。此外，还有许多成功的遗传算法作曲系统，如 Aladdin Ayesh 和 Andrew Hugill 构造的基于 EMFA 算法的系统^[22]；George Papadopoulos 和 Geraint Wiggins 创建的基于遗传算法的旋律进化系统^[23]，其中，该系统主要是针对爵士乐风格的旋律；Muneyuki Unehara 和 Takehisa Onisawa 的音乐作曲系统^[24]，该系统也

是基于交互式遗传算法,而且即使是非音乐专业的人员也能使用该系统来创作旋律;Unemi 构造的作曲工具 SBEAT^[25],这是使用模拟繁殖的方法产生的,等等。

基于遗传算法的方法首先随机生成音符的序列,然后再通过这些序列产生旋律,不需要知道待寻优问题其他的特殊信息。生成的音符序列具有较高的质量,排列顺序也比较合理,可以成倍增加音符串,产生出新的音乐知觉^[26]。但是当使用人作为适应度函数时,存在适应度瓶颈问题,影响作曲的效率与质量。

在使用遗传算法进行旋律创作时,首先需要生成初始群体,以便于进行进一步的优化操作。目前常用的是人工智能(Artificial Intelligence)的方法^[27],即构建和培养训练一个人工智能算法系统,根据这个系统来自动生成旋律。使用人工智能技术生成初始群体时主要采用以下方法:基于数学模型的方法、基于规则的知识库系统、基于音乐文法的方法、基于案例推理的方法、基于人工神经网络的方法,下面分别进行介绍。

(1) 基于数学模型的方法

初始群体的生成方法中最早采用的是基于数学模型的方法,基本思路是首先分析给定的旋律集并且构建数学模型,之后依据该数学模型生成新的旋律,其中最为常见的是概率模型。

Hiller 和 Isaacson 基于产生——测试的方法创作了弦乐四重奏^[28]。首先使用 Markov 链模型来产生伪随机音符,之后基于和声与复调的规则对这些音符进行测试,最后选择符合规则的音符来生成音乐旋律。该方法计算复杂度较低,因此,许多商业程序采用该方法进行旋律创作,例如 Cybernetic Composer 系统^[29]。这个系统能够生成具有不同风格的音乐片段,例如爵士乐、摇摆乐以及拉格泰姆(Ragtime)乐等。

基于数学模型的方法,算法简单,实时性也较好,可用于即兴演奏,比较适合用于商业方面。但是缺点是对旋律中蕴含的数学关系过分依赖,生成的旋律缺乏主题性,从中提炼更高层的或者抽象的音乐概念也比较困难。

(2) 基于规则的知识库系统

这种方法结合多个相关领域专家的知识建立起该领域的知识库,而且在系统中运用了符号。对于音乐知识,使用规则和约束对其进行描述。Ebcioğlu 提出了回溯说明语言(Backtracking Specification Language,简称BSL)并将其用于他们的系统 CHORAL 中^[30]。这个系统采用基于规则的方法,能够生成拥有巴赫的风格的四声部合唱曲。此外,Anders 和 Miranda 设计出一个遵循规则生成和声序列的有效方法^[31],Tanaka 等人把严密的二部

对位规则编码成能够产生有说服力的结果的随机模型^[32]。

基于规则的知识库系统的引导机制比较直观，能够对系统所做的选择行为进行解释，而且系统根据作曲者编写的程序实现那些“专家”的能力。但是，在音乐方面很难建立知识引导机制，而且系统中的“专家”不能灵活地表现知识，生成旋律的风格较形式化，更重要的是，音乐是主观的，难以用精确的规则进行表述。

(3) 基于音乐文法的方法

Steedman 构造了一种用来表述具有 12 小节的爵士乐蓝调和弦进程的生成文法^[33]，之后又采用范畴文法，改进了其之前所做的工作。他的系统能够对那些在传统上看作是右分叉的结构进行左分叉分析，因此提升了系统的解释能力。该系统模拟产生的直觉更加接近听者的直觉，而且人们也更倾向于认可其所生成的和弦演绎进程。David Cope 的音乐智能实验(Experiments in Musical Intelligence, 简称 EMI)比较注重理解音乐的风格并且将每个作曲家的创作风格进行复制^[34]。该系统存储并学习那些已经逝去的作曲家的旋律创作风格，进而生成具有相似风格的旋律，比如与巴赫的创意曲、器乐协奏曲和组曲相似的旋律，以及与莫扎特的奏鸣曲和肖邦的夜曲相似的旋律，等等。

基于音乐文法的方法在生成旋律时使用了不同的文法规则，易于得到大量的音乐序列。将该方法与统计方法结合，能够发现目前旋律中的各种事件(比如节奏、和声、音程等)的概率分布情况，根据这些特点产生的旋律也具有相似的风格。但是，可能会出现音乐结构不一致的情况，而且生成的音符序列的质量也不是很高。

(4) 基于案例推理的方法

完全基于规则的音乐文法的方法经常会失败，这是因为不是由规则来生成旋律，而是由旋律来生成规则。基于案例推理的方法首先将成功的和声配置旋律当作样本，然后再对新的旋律执行配置操作。当系统需要对给定的旋律执行和声配置操作时，首先查找与给定旋律相似且已执行过这种操作的案例，若查找不到，则寻找可用的和声规则；如果规则均不可用，则操作失败并返回到上一决策点。

基于案例推理的方法是通过经验进行学习的，而且如果系统中的样本数越多，那么重排规则的机会就越少，进而成功的可能性也就越大。

(5) 基于人工神经网络的方法

人工神经网络方法，即模拟人类的生物神经系统的人工智能方法，在模式识别、工业控制等领域均得到了广泛应用，而且能够成功地应用于旋律的识别与感知中。

MUASCT 为了学习和声模式而使用神经网络方法^[35], 目的在于获得具有和声性质的音乐直觉^[36]。Mozer 的 CONCERT^[37], 通过递归的神经网络技术构建而成, 训练时采用反向传播学习算法。当生成旋律时, CONCERT 按照音接音(Note-by-Note Composition)的方式进行创作。

基于人工神经网络的方法不具备先验知识, 完全通过人工神经网络来学习示例旋律的特征。这种方法可以捕获到旋律的细部结构, 并以此为基础产生新的旋律, 而且可以辨别出旋律的节奏与风格, 同时还具有反馈机制。但是一般情况下体现不出与旋律的调性、乐句等相关的音乐信息, 而且该方法对于预置了节奏和风格的作曲比较适用, 其余的则效果不是很好, 通用性较差。

当使用遗传算法进行旋律创作时, 为了生成初始群体, 可以使用上面介绍的这几种方法, 但均需要作一定的改进, 也可以使用这几种方法的综合, 以便生成更有效地群体。

1.3 本文的主要工作

本文基于遗传算法构建了使用多目标适应度函数来自动评估生成个体的旋律创作系统, 主要内容如下:

- (1) 对算法作曲的概念及其发展历程作了概述, 并且分析和比较了当使用遗传算法进行旋律创作时生成初始群体的几种常用的人工智能方法;
- (2) 全面分析和介绍了遗传算法的相关知识;
- (3) 对相关的旋律学基础理论作了介绍;
- (4) 详细介绍了将遗传算法应用于旋律创作时对遗传算子的设计, 构造了一种新的编码方式, 着重对适应度函数以及使用的特征提取器作了研究, 并通过实验对本文中构建的系统进行了验证。

第二章 相关理论知识

2.1 遗传算法

2.1.1 遗传算法的基本思想

遗传算法作为一种进化算法，其基本思想是达尔文的进化论和孟德尔的遗传学说。由达尔文的进化论可知，自然界中的各种生物在不断的生存斗争中，通过遗传和变异产生新的个体，新个体若能适应环境则保留，而不适应者则被淘汰，即“适者生存”法则。根据孟德尔的遗传学说，作为控制个体性状的指令码，遗传以基因的形式存在于染色体中。基因具有特定的位置并且决定着某一特殊性状。通过杂交和基因突变，产生的后代对环境的适应性更强，再通过自然选择(即优胜劣汰)，保留适应性强的个体，低的则将其淘汰。

给定一个实际的问题，遗传算法首先将问题编码为“染色体”，根据“适者生存”，选出用于复制的染色体，这一过程称为再生(Reproduction，或称选择，Selection)，之后再通过交叉(Crossover)、变异(Mutation)等操作产生新的染色体，这些新的染色体具有更强的适应能力。将此过程重复进行直到出现最能适应环境的个体，也即为最优解。图 2-1 为遗传算法的流程图。

2.1.2 遗传算法的基础理论

遗传算法的基础理论主要是分析待求解问题的收敛性，也即是计算群体收敛到全局最优解的概率。整体上而言，可以分为随机模型理论和进化动力学理论。

(1) 随机模型理论

该理论是基于随机过程的，而且，如果编码空间或者种群是有限的，可以采用马尔科夫链模型(Markov Chain Model)来表示搜索过程，注意，该模型是离散时间的。最后，基于已经存在的相关随机过程理论对该搜索过程进行分析。

(2) 进化动力学理论

与随机模型理论不同，进化动力学理论是基于模式理论的。模式理论可以分为模式定理与积木块假设。其中，模式定理是由 Holland 教授提出的，可以确保较好的模式的

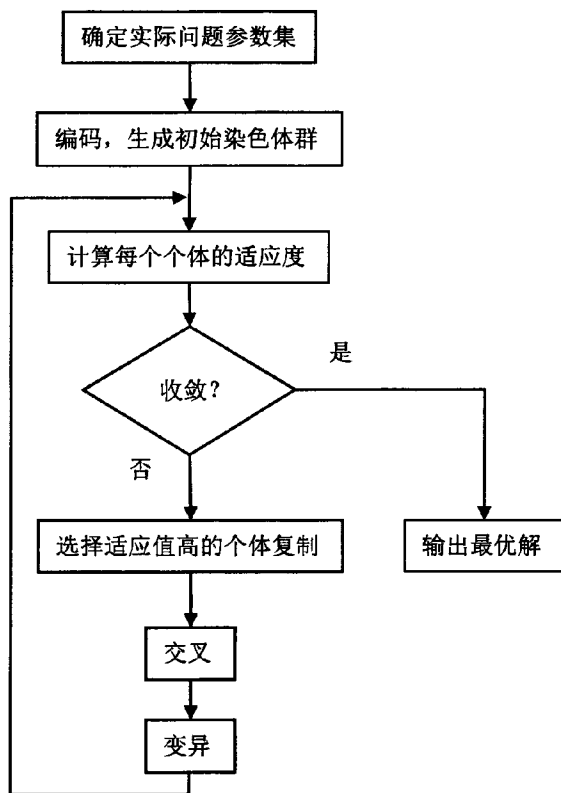


图 2-1 遗传算法的流程图

数量呈指数形式增加, 是进化动力学理论中的基本定理。而积木块假设则主要描述了遗传算法具有的重组功能。

当使用遗传算法求解问题时, 模式定理和积木块假设这两者可以确保发现全局最优解, 而且还可以对进化行为作一系列的分析。

2.1.3 遗传算法的基本概念

(1) 种群和个体

种群(Population), 是指当使用遗传算法求解问题时, 所给出问题的众多解的集合, 而且问题的求解过程从这个集合的子集开始。个体(Individual), 是指种群中的某一个元素, 常常表示为能够清楚地描述个体的基本遗传结构的形式。根据所使用的字符的个数来选择编码方式。若只使用 0、1, 是二进制编码; 若使用 0、1、…、9, 则是实数编码。

(2) 染色体

对于个体, 当采用一定的编码方式对其进行编码后所得到的编码串就称为染色体(Chromosome)。染色体上的每一位称为基因, 其中由若干基因构成的一个有效信息段则

称为基因组。

(3) 适应度与适应度函数

适应度(Fitness)，是指根据自然界中的生物个体对其所处环境的适应程度，设计出一种用来表现待求解问题中的个体的优劣的方式。在待求解问题中的全部个体与其适应度之间构造的一种对应关系即为适应度函数(Fitness Function)。一般情况下，适应度函数是实值函数，用于衡量种群中的个体对于环境的适应性。遗传算法若要实现优胜劣汰操作，必须以经过计算得到的函数值为基础。选取适应度函数是遗传算法中非常重要的一步，其优劣将会对遗传算法的收敛速度产生重大影响，而且决定着该方法是否能够找到最优解。一般情况下，通过将目标函数 $g(x)$ 进行一系列的变换来得到适应度函数，主要有以下 3 种方式：

1) 通过变换待求解的目标函数来作为适应度函数，即

$$F(g(x)) = \begin{cases} g(x) & \text{if } g(x) \text{ 为最大化问题} \\ -g(x) & \text{if } g(x) \text{ 为最小化问题} \end{cases} \quad (2-1)$$

2) 若目标函数为最小化问题，则

$$F(g(x)) = \begin{cases} C_{\max} - g(x), & g(x) < C_{\max} \\ 0, & \text{其他} \end{cases} \quad (2-2)$$

若目标函数为最大化问题，则

$$F(g(x)) = \begin{cases} g(x) - C_{\min}, & g(x) > C_{\min} \\ 0, & \text{其他} \end{cases} \quad (2-3)$$

其中， C_{\max} 、 C_{\min} 分别是估计的 $g(x)$ 最大值、最小值，有多种选择方法，例如输入一个合适的值，或者是截止到目前的进化过程中的最大(小)值，值得注意的是， C_{\max} 、 C_{\min} 最好与群体无关。

3) 若目标函数为最小化问题，则

$$F(g(x)) = \frac{1}{1+c+g(x)}, \quad c \geq 0, c+g(x) \geq 0 \quad (2-4)$$

若目标函数为最大化问题，则

$$F(g(x)) = \frac{1}{1+c-g(x)}, \quad c \geq 0, c-g(x) \geq 0 \quad (2-5)$$

其中，以上两个式子中的 c 是通过目标函数界限的保守估计来获得的。

(4) 遗传算子

遗传算子(Genetic Operator),也称为遗传操作,主要有选择、交叉和变异三种算子。

选择算子(Selection Operator),也称为复制算子(Reproduction Operator),即根据种群中个体表现的优劣判断是否将其保留,通常选择适应度高的个体产生后代。常见的有:轮盘赌(Roulette Wheel)选择(也称比例选择)、排序选择、随机联赛选择、确定式采样选择、精英选择等。

交叉算子(Crossover Operator),是指按照某种方式,把两个待配对的染色体上的部分基因进行交换进而形成新个体的运算,目的是将父代的优良基因进行优化组合进而传至子代,并产生新的寻优空间。采用的编码方式不同,选用的交叉算子也有不同。若采用二进制编码,则有单点交叉、两点和多点交叉、一致交叉等;若采用实数编码,则有简单交叉和算术交叉。

变异算子(Mutation Operator),是指将种群中个体的染色体上的某些基因用其他的等位基因将其替换从而形成新个体的运算。与交叉算子类似,变异算子的选取也与编码方式有关。若采用二进制编码,则有基本位变异、均匀变异等;若采用实数编码,则有均匀变异和非均匀变异。

(5) 控制参数

遗传算法操作中需要设定一些参数来提高算法的性能,主要有:

种群规模 M : 群体的大小,也即是群体中个体的数量,其大小将会对算法的有效性有影响,如果太小,算法效果将会很差,或者寻不到最优解;如果太大,将会增加工作量,耗时增多,一般取 20-100。

进化代数 G : 即算法的最大进化代数,常作为判断算法是否终止的条件。

交叉概率 P_c : 即进行交叉运算的染色体数量占全部的染色体数量的比例,一般取 0.25-0.75。

变异概率 P_m : 即产生变异的基因位的数量占全部染色体的总的基因位数的比例,这个概率通常比较小,一般取 0.001-0.1。

这些参数没有明确统一的规定,一般根据经验进行设定。

2.1.4 遗传算法的基本操作

(1) 编码

由于遗传算法无法直接对问题空间的参数进行处理,因此需要一种转换操作,将问题空间表示为具有一定结构的染色体或者个体,这一过程就称为编码。常用的有二进制

编码和实数编码。

1) 二进制编码(Binary Encoding)

二进制编码,是指用集合 $\{0,1\}$ 来将问题空间中的参数表示为染色体位串。进行编码时,首先需要对二进制字符串的长度进行明确规定。这是因为该字符串的长度与所定义的变量的定义域以及待求解问题的计算精度都有很大关系。

2) 实数编码(Real Encoding)

实数编码,是指将各个体的染色体表示为一定范围内的实数(或者浮点数),该问题中变量的个数即为编码的长度。这种类型的编码方法将操作环境转移到实数空间上,具体的操作对象是实数。对于多维、要求高精度的连续函数优化问题,常常使用实数编码方法。

二进制编码和实数编码各有千秋,Michalewicz 指出^[38],二进制编码的操作对象是基因,而实数编码的操作对象是个体,所涉及的理论基础不同。而且,大量实验结果表明:对于同一类型的优化问题,这两种编码方法的性能类似。

(2) 适应性的度量

遗传算法经编码后将问题空间用染色体位串空间来表示,按照自然选择中的适者生存法则,需要评价各个体位串的适应性,此时,个体的生存环境则是适应度函数,然后根据个体的适应值决定其生存能力。通常,较好的染色体经计算后得到的适应度函数值会较高,获得的评价也会较高,进而生存能力也会较强。

遗传算法中适应度函数值决定着种群中个体的生存与否,为了更好地将适应度函数与个体的优劣联系起来,通常将适应值设定为非负,并且值越大越好。

(3) 选择算子

采用选择算子的目的在于获得用于交叉和变异的个体,并且规定所选个体产生的子代个体的数量。在计算适应度之后,根据得到的适应值选择待操作的个体。通常有以下几种选择方法:

1) 轮盘赌选择

轮盘赌选择是基于适应度函数值的比例选择。首先,计算种群中各个体的适应度函数值,之后计算该适应值与种群中所有个体的适应值之和的比例,该比例也即是该个体被选中的概率。若给定种群大小为 M 的群体 $P=\{1, 2, \dots, M\}$, 个体 i 的适应值为 $f(i)$,

则其选择概率为 $p_i = f(i) / \sum_{i=1}^M f(i)$ ($i=1, \dots, M$)。而且个体的适应值越大, 选中的机会也就越大。根据计算得到的选择概率 p_i , 把一个圆盘分成 M 份, 同时随机产生一个 $[0,1]$ 范围内的数 r , 若 $\sum_{j=1}^{i-1} p_j < r \leq \sum_{j=1}^i p_j$ ($i, j=1, \dots, M$), 则选择个体 i 。

由于 r 的产生是随机的, 因此这种方法的误差比较大。

2) 随机联赛选择

随机联赛选择, 其基本思想是每次从当前群体中随机选出数个个体并将其中适应值最大的个体保留至下一代。该方法的优点是对个体的适应度函数值的正负不作要求, 但是随机误差也会随之变得更大。联赛规模(N)是指每次参与比较适应度函数值大小的个体的数量, 通常情况下取 2。首先随机取出 N 个个体并比较它们的适应度, 将最大的保留至下一代, 重复 M 次, 即可得到子代的 M 个个体, 其中 M 是种群大小。

3) 精英选择

在遗传算法的众多选择策略中, 精英选择能够保证种群收敛到优化问题的最优解。分别计算父代和子代群体中个体的最佳适应值并将其进行比较, 若子代群体中的最佳个体适应值较小, 则将父代群体中大于子代最佳适应值的个体均复制到子代中并且随机地替换子代中具有较低适应值的相应数量的个体。

(4) 交叉算子

通过选择操作可以从种群中选出优秀的个体, 但不能产生新的染色体, 而交叉操作模拟自然界生物进化过程中的繁殖过程, 交换并将两个染色体重新组合进而产生新的优秀个体。基本步骤为: 首先从匹配池中随机选出两个待进行交叉操作的染色体, 在其上任意选出一一点或者多点作为交叉点, 再对这两个染色体的交叉点后的部分进行交换, 便可获得新的染色体。

常见的交叉算子有单点交叉、两点或多点交叉、算术交叉等。其中单点交叉的应用范围最广。

1) 单点交叉

单点交叉, 也称简单交叉, 是指在待进行交叉操作的染色体上只设有一个交叉点, 之后在该位置交换交叉点后的部分。

2) 多点交叉

与单点交叉相比,多点交叉即是设置了两个或者多个交叉点,然后再进行交叉操作。

3) 算术交叉

算术交叉,是指将待进行交叉操作的两个染色体进行线性组合从而形成新的个体。一般情况下是对用实数编码的染色体使用算术交叉。

(5) 变异算子

在自然界生物进化的过程中,通过变异来产生新的性状甚至是新的物种,而变异操作则是对此过程的模拟。

常用的变异算子有基本位变异、均匀变异和非均匀变异。

1) 基本位变异

基本位变异,首先,对染色体位串上的每一位基因根据变异概率 P_m 指定变异点;然后,在每个变异点处,对其基因值进行取反或者使用其他等位基因进行替换,进而产生新的个体。

2) 均匀变异

均匀变异,首先依次指定染色体位串上基因的变异点,然后对于每个变异点,根据变异概率 P_m ,将原基因值用从相对应的基因的取值范围中任意选出的一个数来将其进行替换。

3) 非均匀变异

非均匀变异的操作过程大体上与均匀变异相似,但也有不同,即非均匀变异着重搜寻与原染色体邻近的微小区域。

2.2 相关旋律学基础理论

2.2.1 音

旋律是由具有各种性质的音按照一定的顺序构成的^[39]。音,是自然界中的一种物理现象。虽然人类在自然界可以听到很多的音,但是并不是说所有的音都能用于产生音乐。音乐中能够使用到的音的频率(不包括泛音),一般仅限于 27-4100Hz 范围内。音乐中的音,是指人们通过长期的社会生活实践选出能够反映人们的生活情况或者表达人们的思想感情的声音,将其按照一定的方式组织起来形成固定的体系,主要功能为表达音乐中蕴含的思想和塑造音乐形象。

音的主要性质包括音高、音长、音强以及音色。这些性质在表现音乐时显得非常重

要，尤其是音的音高和音长。

(1) 音高

音高，即是乐音的高低水平。乐音的高低是由物体在规定时间内振动次数决定的。物体振动的次数越多，音就越高；反之，则音越低。

(2) 音长

音长，即是乐音的长短情况，取决于乐音的持续时间。乐音的持续时间越长则音越长，反之，音越短。

(3) 音强

音强，即是乐音的强弱情况，取决于乐音的振动幅度的大小。幅度越大则音越强，反之，则音越弱。

(4) 音色

音色取决于发音体的性质。从大的范围来看，音乐分为两类，一类是由人声演唱的声乐，而另一类是由乐器演奏的器乐。其中，声乐可以分为男音、女音，也可以分为高音、中音、低音，还可以分为美声、民族、通俗等。而器乐则可以分为弦乐、管乐、弹拨乐以及打击乐等。

音还可以分为乐音和噪音，这主要是依据音的振动状态是否规则。其中，乐音主要用于音乐中，而噪音也是音乐中不可或缺的部分，比如架子鼓，它产生的音响就是符合某些规律的噪音。

2.2.2 音阶

音乐中最主要的音称为主音，其稳定程度最强。一般情况下，主音出现的次数最多，时值最长，所在位置也最重要。从主音到主音，根据音高的次序从低至高排列音符，如此形成的音列就称为音阶(Scale)。由于世界上各个地区具有不同的音阶，而且音乐水平也在不断进步，音乐理论和相关的系统得到完善。当学习音乐时，目前均采用的是西方的十二平均律。现在经常所说的音阶，主要是大音阶(大调)和小音阶(小调)。

根据调式中音的数目，将音阶分为“五声音阶”、“七声音阶”等。若音阶为由低到高，则称作上行；若为由高到低，则称作下行。五声音阶(Pentatonic scale)含有五个音，民族音乐中常采用这种调式，如DO、RE、MI、SOL、LA、(DO)。至于七声音阶(Heptachord)，相邻的二音之间的间距或者是全音，或者是半音，其中第三、四两音间为半音，第七、八两音间也为半音，其余为全音。

2.2.3 乐音体系

应用于音乐中并且音高固定的音的总和，称作乐音体系。

(1) 音列

音列是指将一首旋律或者一个乐段中的各音按照上行或者下行的次序进行的排列。为了便于区分具有相同音名但是不同音高的音，将音列中的音进行了分组。音列中处于中央位置的一组称为小字一组，比小字一组低四倍、低三倍、低二倍、低一倍的分别称为大字二组、大字一组、大字组和小字组，而高于小字一组一倍、二倍、三倍的则分别称为小字二组、小字三组和小字四组。图 2-2 所示为钢琴的音列分组。

(2) 音级

音级是指乐音体系中包含的每一个音。音级可以分为基本音级和变化音级。在乐音体系中，基本音级是指具有独立名称的七个音级，其中，音级的名称使用字母和唱名进行标记。相邻的两个具有相同名称的音称为八度。如图 2-2 中，小字组的 c 与小字一组的 c^1 相差了八度。将基本音级进行升高或者降低变化后得到的音，称作变化音级。若要将基本音级升高半音，标记为“升”或“#”；若要降低半音，则标记为“降”或“b”；若要升高全音，则标记为“重升”或“x”；若要降低全音，则标记为“重降”或“bb”；若要还原，则标记“ \natural ”。

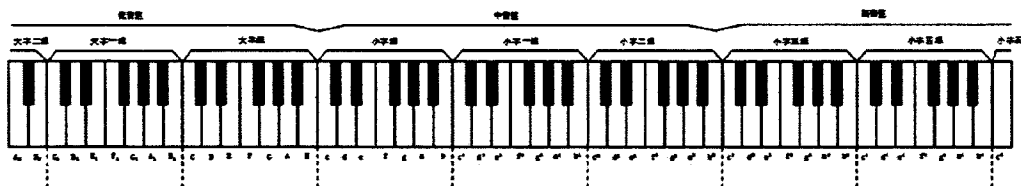


图 2-2 钢琴的音列分组

(3) 音域

音域是指人声或者乐器可以达到的音的范围，分为总音域和个别的人声或者乐器的音域。总音域，即音列的总范围，也即是从最低音至最高音(C_2-c^5)之间的范围。个别的人声或者乐器的音域，指的是人声或者乐器可以达到的那一部分音域，比如，钢琴的音域为 A_2-c^5 。

(4) 音区

音区是音域的一部分，可以划分为低音区、中音区和高音区。整个音域中，低音区为：大字组、大字一组和大字二组，中音区为：小字组、小字一组和小字二组，而高音

区为：小字三组、小字四组和小字五组。如图 2-2 所示。当划分人声的音区时，通常不符合上述规律，例如，女低音的低音区和男低音的高音区重叠。但是每一音区具有各自独特的音色，能在音乐的表现中体现出来。一般情况下，低音区低沉、浑厚，而高音区清脆、尖锐、嘹亮。

2.2.4 音程

(1) 音程的概念

音程，指的是两个音级关于音高的关系，也即是这两个音在音高上的距离，单位是度。度是指两个音符之间相差的自然音音名的数目，比如三度指的是从这一音开始的第三个自然音音名，例如，计算 DO 和 MI 之间的度数的方法是：从 DO 到 MI 共有三个自然音音名，所以 DO 和 MI 之间的度数是三度。度数无法表现出 DO 和 MI 之间的实际距离，这是因为实际距离是通过半音数来计算的。当计算音程时，如果只使用度数，就会出现以下情况：虽然音符之间的度数相同，但相差的半音数目却不一定不同，比如 DO 和 MI 相差 4 个半音，而 MI 和 SOL 相差 3 个半音，它们的度数相同，均是三度，但是实际距离却不相同。因此，在确定一组音的度数之后，为了能更准确地确定这组音的音程，还需要在得到的度数的前面加上增、减、大、小等形容词。

(2) 常见音程

常见的音程名称如图 2-3 所示。掌握了这些基本的自然音程，其余的就可以按照以下法则进行推导计算得出^[40]：如果音程是大、小、完全音程，则不用操作。否则，如果该音程大于大音程或者完全音程半音，则称之为增音程；如果该音程小于小音程或者完全音程半音，则称之为减音程。如果音程在八度之内，则称为单音程；否则，则称为复音程。

音程可以分为旋律音程和和声音程。旋律音程是由按照先后次序进行弹奏的两个音构成的，而和声音程则是由同时进行弹奏的两个音构成。

2.2.5 节奏与节拍

将音乐中的长音和短音按照一定方式有规律地组合起来就形成了节奏。节拍是衡量节奏的单位，是指具有相同时值的强拍和弱拍每隔一定时间循环出现。节奏和节拍在音乐中同时存在，节奏不考虑音高，仅关注的是音的长短，而节拍主要考虑音的强弱交替规律^[41]。

常见音程：

- 距离 0 个半音： 纯一度 (DO-DO、MI-MI)
- 距离 1 个半音： 小二度 (MI-FA)、增一度 (DO-#DO)
- 距离 2 个半音： 大二度 (DO-RE)、减三度 (#RE-FA)
- 距离 3 个半音： 小三度、增二度
- 距离 4 个半音： 大三度、减四度
- 距离 5 个半音： 纯四度
- 距离 6 个半音： 增四度、减五度
- 距离 7 个半音： 纯五度
- 距离 8 个半音： 小六度、增五度
- 距离 9 个半音： 大六度
- 距离 10 个半音： 小七度
- 距离 11 个半音： 大七度
- 距离 12 个半音： 纯八度

图 2-3 常见音程名称

“拍”，指的是使用某一具有固定时值的音符来表示节拍的单位^[42]，其中以四分音符或者八分音符为一拍比较常见。“拍子”是将“拍”按照某种强弱关系组织而成。一个小节中包含若干拍，当想获得一个乐段甚至一首完整旋律的长度时常使用小节作为计算单位。拍号是一种表示拍子的记号，如 4/4 表示的是以四分音符为一拍，其中每小节中有四拍。

拍子的形态丰富多样，在节奏设计中起着举足轻重的作用，包括单拍、复拍、混拍、变拍、隐拍、散拍等，其中最为常见的是单拍和复拍。如果每小节中只含有一个强拍，则称为单拍。而复拍是重复某一种单拍的拍子形式，可以重复一次、二次，甚至更多。复拍与单拍的相异之处在于次强拍的增加。复拍的每一小节中具有一个强拍以及若干个次强拍。例如，四四拍可以看成是两个四二拍去掉中间的小节线后结合在一起，这时去掉的小节线后的第一拍成了次强拍，构成“强、弱、次强、弱”的复拍。

节奏和节拍与音乐所表现出来的风格密切相关。一般情况下，节奏越紧凑，音乐就越激扬、欢快，否则，音乐就越安静、压抑。由于只有三拍能旋转起来，因此经常将三拍用于圆舞曲中。

2.2.6 调式

音乐中，调式是指多个具有不同程度音高的音(一般情况下不多于七个)以某一个具有稳定感的音为中心并且按照某种关系连接在一起所组成的体系。这是人们通过长期的实践构建出的乐音组织结构形式。

在调式体系中，音可以分为两类：稳定音和不稳定音。其中，稳定音是指在音乐中居于核心地位而且给人以稳定感的音，不稳定音则是给人以不稳定感的音。

调式可以分为大调式与小调式。大调式中有七个音，将其所含的稳定音组合起来能够形成大三和弦。小调式中也有七个音，其所含的稳定音组合起来则能够形成小三和弦。大调式中，主音与其上方第三个音称为能够表明大调式的色彩的大三度，而小调式中，主音与其上方第三个音称为能够表明小调式的色彩的小三度。无论是在大调式体系还是小调式体系中，都是第 I、III、V 级是稳定音。但是，这三级音的稳定程度却不尽相同，其中，第 I 级最稳定，而第 III 级和第 V 级则没有那么稳定。只有在与主音的三和弦共鸣的时候，这三个稳定音及其稳定性才能够表现出来。第 II、IV、VI、VII 级为不稳定音级，当满足一定条件时它们具有进行到稳定音的倾向。

(1) 大调

根据十二平均律系统，为了构造出新的大调，可以根据大调的音程排列次序从以下的任一个半音开始，这些半音为 DO、#DO、RE、#RE、MI、FA、#FA、SOL、#SOL、LA、#LA、SI。如图 2-4 所示的 C 大调。

I	II	III	IV	V	VI	VII	I
全音	全音	半音	全音	全音	全音	全音	半音

图 2-4 C 大调

大调中有七个音，常用罗马数字来表示这七个音的级数。第一个音为 I 级音，是该大调的主音，第七个音为 VII 级音，能够将音阶再引回到主音上，因此常称之为导音。由图 2-4 可知，大调中相邻两音之间相差的音程依次为“全-全-半-全-全-全-半”，这也就是大调的组成规则。

(2) 小调

小调比大调简单，附属于大调，并且与所依附的大调共用调号。小调的主音可以通过将大调的主音移小三度来得到。小调的组成规则是“全-半-全-全-半-全-全”。例如，

对于 C 大调，附属的小调为 a 小调。

小调可以分为自然小调式、和声小调式、旋律小调式和现代小调式四种。

1) 自然小调式

在简谱中，除了调号，没有其他的临时记号。其音阶的结构表示为：

6 7 1 2 3 4 5 6

2) 和声小调式

由于和声学中存在导音，而且导音与主音相差了半音，因此，和声小调式是通过将自然小调式的第七级音升高半音得到的。在简谱中，其音阶的结构表示为：

6 7 1 2 3 4 #5 6

3) 旋律小调式

由上述可知，和声小调式将第七级音升高了半音，此时第六级音与第七级音相差了三个半音，为了旋律的和谐性，在音阶上行时也将第六级音升高半音，这时，旋律给人的感觉就类似于大调。因此，当音阶下行时还原第六级音与第七级音，这时旋律就类似于自然小调式，将其称之为旋律小调式。在简谱中，其音阶的结构表示为：

6 7 1 2 3 #4 #5 6

4) 现代小调式

随着现代和声技术不断地进步，作曲家们对新音阶的需求日益强烈，现代小调式应运而生。不同于旋律小调式，除了调号以外，现代小调式在音阶下行时不用还原第六级音与第七级音。

此页不缺内容

第三章 应用于旋律创作的遗传算法设计与实验

由于遗传算法具有智能性、编码方式灵活、隐并行性等特点，而且易于与其他智能算法和技术结合，因此可以使用遗传算法来创作旋律^[43]。

遗传算法作曲，即是基于遗传算法的音乐自动生成过程，执行期间要求人类尽可能地干预。Phon-Amnuaisuk, Tuson 和 Wiggins 给出了当结合遗传算法和音乐时很多需要考虑的问题的综述^[44]，展示出几个成功地使用遗传算法并用和声来演唱音乐片段的例子，并且说明了在遗传算法中对大量的音乐知识和实践进行编码的必要性。Biles 把这些称为音乐上有意义的变异^[45]，而且创作时没有这些方面的知识将很难产生有说服力的音乐。Freitas 和 Guimaraes 证明了创作旋律时使用音乐上有意义的变异的能力，而且对旋律实施交叉和变异操作还有如下方法：例如，在小节间交换音符，随机化和弦以及复制其他小节等。他们的旋律和声系统创造出接近人类质量的和声。这一现象表明当被授予特定的音乐知识时使用遗传算法创作旋律将会很成功。

3.1 使用遗传算法创作旋律的步骤

基于遗传算法的方法创作旋律是一个不断进行的迭代过程，基本步骤如下：

- (1) 由用户设定相关的旋律参数，计算机根据这些参数生成旋律；
- (2) 将这些旋律进行编码，生成初始的旋律种群；
- (3) 构建适应度函数；
- (4) 根据所构建的适应度函数，为每首旋律计算适应度函数值；
- (5) 判断是否大于设定的适应度函数阈值，若大于该值，则转向步骤(8)，否则，执行步骤(6)；
- (6) 选择种群中适应度函数值较高的个体，按照预先设定的遗传参数，进行交叉、变异等操作，产生新的旋律；
- (7) 返回并执行步骤(4)；
- (8) 终止，输出产生的新旋律。

本文中构建了基于遗传算法的旋律创作系统，该系统的流程图如图 3-1 所示。

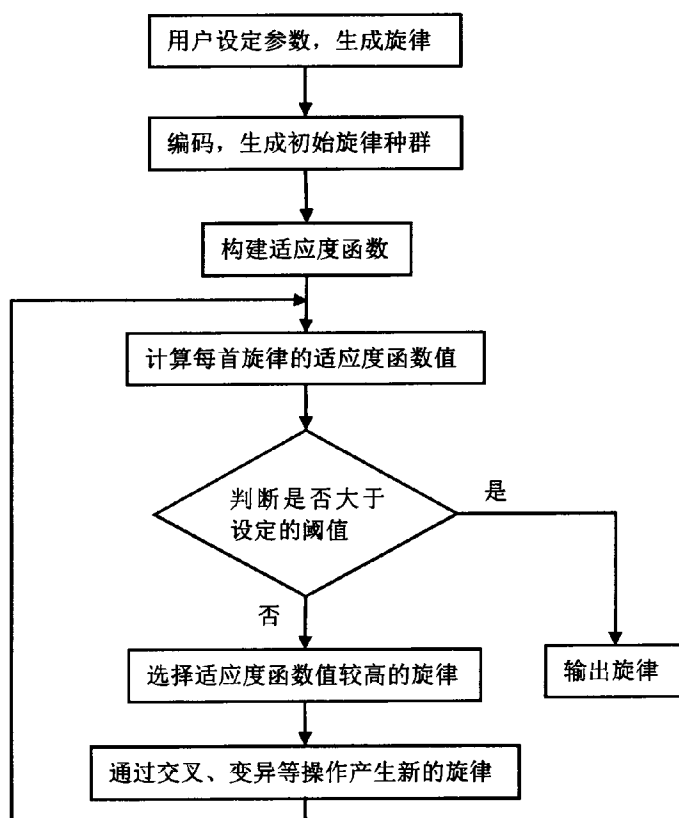


图 3-1 基于遗传算法的旋律创作系统流程图

本文中对传统的遗传算法作了稍微的改进，但是仍然保持了算法的完整性，具体算法如图 3-2 所示。

算法 1 遗传算法

- (1) 初始化群体;
- (2) 当未结束时:
 - (a) 为所有的个体计算适应度;
 - (b) 按照适应度对个体进行排序;
 - (c) 构造个体的概率性配对池;
 - (d) 使用交叉算子从配对池中产生新的后代;
 - (e) 在新的后代上使用变异算子;
 - (f) 挑选后代和现有群体的子集作为新的群体;
- (3) 输出适应度 $>T$ 的 M 个个体.

图 3-2 遗传算法

本文是在 Microsoft Visual C++ 6.0 环境下进行编码实验的。系统界面如图 3-3 所示。

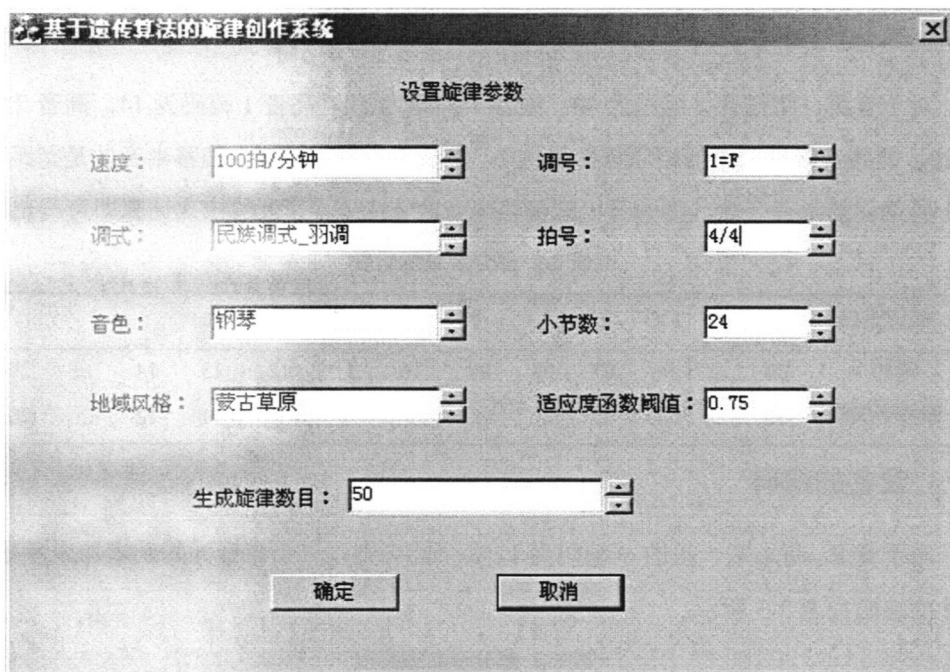


图 3-3 基于遗传算法的旋律创作系统

3.2 编码策略

编码是当使用遗传算法求解问题时用于描述问题的可行解的方式，也就是通过转换，将问题的解空间变为遗传算法能够在其上直接进行操作的搜索空间。在使用遗传算法时首先需要解决的问题即是编码，这是因为编码不仅决定着染色体的排列形式，而且还会影响解码，同时对交叉、变异操作也会有影响，也即是所选用的编码方式在很大程度上将会决定群体进化的方向并且影响进化的效率。

本文中在使用遗传算法进行旋律创作时，采用的是实数编码。具体操作过程后面小节中将会详细介绍。

3.2.1 调号的编码

对于调号，将调号 1=C 编码为 01，剩余的调号以相差的半音数为基础依次增加 1，具体的编码如表 3-1 所示。

表 3-1 调号编码表

调号	1=C	1= [#] C	1=D	1= ^b E	1=E	1=F	1= [#] F	1=G	1= ^b A	1=A	1= ^b B	1=B
编码 M	01	02	03	04	05	06	07	08	09	10	11	12

3.2.2 音级的编码

对于音级，将低音 1 编码为 00，中音 1 编码为 07，高音 1 编码为 14，高音 7 编码为 20，即将音级按照高低顺序依次增加 1，如表 3-2 所示，其中偏移半音数是指距离中音 1 所偏移的半音个数，将中音 1 的偏移半音数置为 0，其余的依次计算即可得出。

表 3-2 部分音级编码表

简谱音级	!	...	?	1	2	3	4	5	6	7	i	...	j
编码 S	00	...	06	07	08	09	10	11	12	13	14	...	20
偏移半音数	-12	...	-1	0	2	4	5	7	9	11	12	...	23

3.2.3 音名的编码

对于音名，将小字一组的 c¹ 编码为 1，其余的音则以一个半音为基础增加或者减少，具体的编码如表 3-3 所示。

表 3-3 部分音名编码表

音名	c	'c	d	'd	e	f	'f	g	'g	a	'a	b
编码 P	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
音名	c ¹	'c ¹	d ¹	'd ¹	e ¹	f ¹	'f ¹	g ¹	'g ¹	a ¹	'a ¹	b ¹
编码 P	1	2	3	4	5	6	7	8	9	10	11	12
音名	c ²	'c ²	d ²	'd ²	e ²	f ²	'f ²	g ²	'g ²	a ²	'a ²	b ²
编码 P	13	14	15	16	17	18	19	20	21	22	23	24

3.2.4 音符时值的编码

对于音符时值，当时值增加时，编码值也会随之增大。用整数来表示音符时值，以三十二分音符为标准，将其定义为 1，其余音符的整数值按照与三十二分音符的比值来计算。对于休止符，类似于相对应的普通音符，如果与普通音符的时值相等，那么编码的整数值也会相同，具体的编码如表 3-4 所示。

表 3-4 音符时值编码表

简谱	$\underset{\cdot}{5}$	$\underset{\cdot}{5}$	$\underset{\cdot}{5}$	$\underset{\cdot}{5}$	$\underset{\cdot}{5}$	5	5 [•]	5-	5--	5---
编码 D	1	2	3	4	6	8	12	16	24	32

3.2.5 旋律的编码

旋律中的每一个乐段都是由众多音符组成的，将每一个音符进行编码，即可得到乐

段的编码,进而可以得到旋律的编码。本文中采用音级、音符时值和符号来编码音符,如图 3-4 所示。



图 3-4 旋律编码表示

其中,符号为“;”或者“&”。如果该音符不是小节中最后一个音符,那么符号位为“;”,表示音符之间的间隔,而“&”表示一个小节结束。例如,对于下面这一小节:

$$\underline{1 \quad 5} \quad 2 \quad \underline{2 \quad 1} \quad \underline{7 \quad 1}$$

根据表 3-2、表 3-4、图 3-4 可知,经过编码,该小节为:074; 044; 088; 084; 074; 064; 074&。

3.3 初始群体的生成

初始群体可以通过两种方式来获取,一种是使用目前已经存在的乐段,另一种是使用由计算机自动生成的乐段。由于第一种方式中目前存在的乐段大都是已经相对成熟的音乐,如果再对其继续进行演化,意义不是很大。因此本文中采用第二种方式并使用基于数学模型的方法来获取初始群体。

根据音乐理论可知,在音乐旋律中,乐段是用于表现完整乐思或者相对完整乐思的最小结构^[46],因此在本文中将乐段看作染色体,而将乐段中的音符看作是染色体上的基因,这样经过多次迭代所得到的最优解也即是具有最优表现力的乐段。为了能够使遗传操作顺利进行,在自动生成初始群体即多个乐段时需要按照一定的规则进行。下面介绍一下计算机自动生成乐段的过程。

首先需要设定一些旋律参数,如调号(1=C, 1=E 等)、拍号(如 2/4, 3/4, 4/4 等)、调式(如大调式、小调式等)、小节个数等。设定好这些参数后,接下来即是对这些参数进行编码,编码方式则按照 3.2 节中的介绍进行编码。

例如,设定调号:1=F;拍号:4/4;小节数:24;地域风格:蒙古草原;等等。由表 3-1 可知,1=F 的编码为 M=06。根据这些参数,采用基于数学模型的方法来自动生成初始群体。这里,由系统自动生成 50 首初始旋律,从中随机选取 8 首用于分析。这 8 首旋律的乐谱图像如图 3-5 所示。

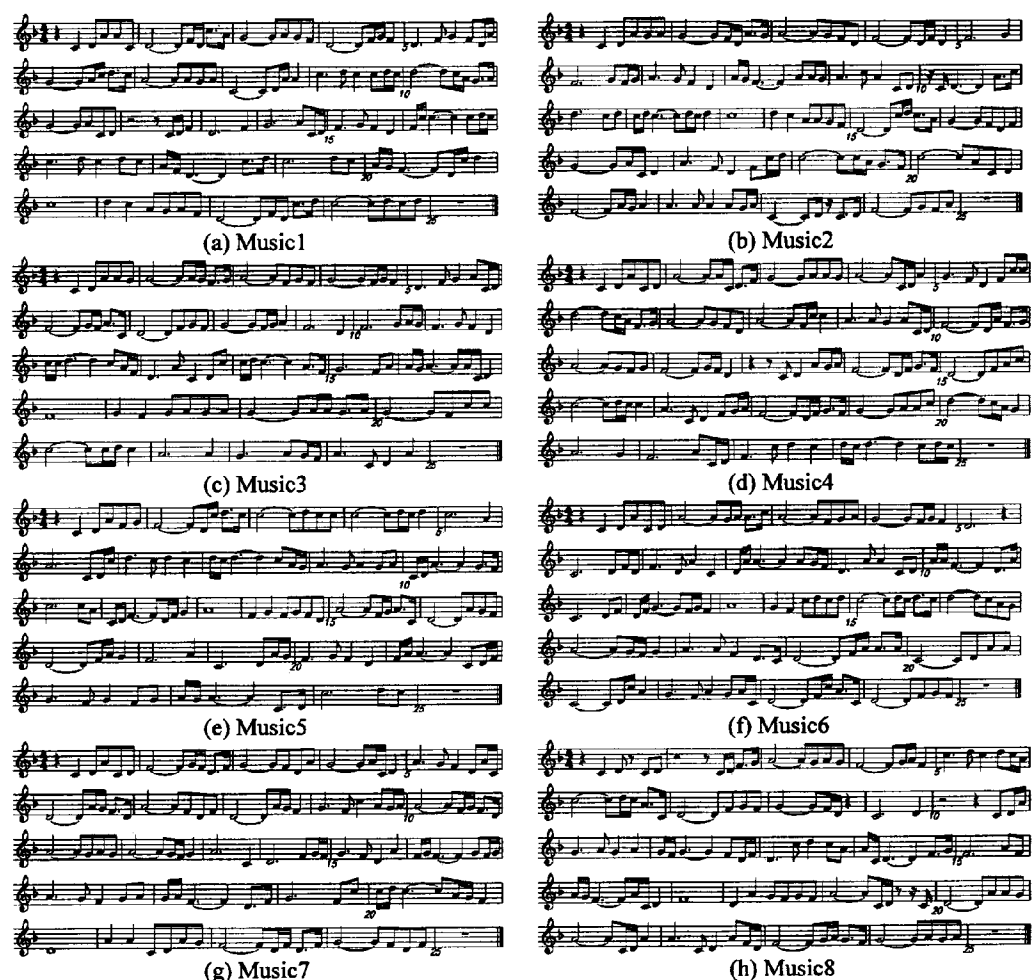


图 3-5 随机选取的 8 首旋律的乐谱图像

3.4 适应度函数的构建

在使用遗传算法创作旋律的过程中,适应度函数的构建至关重要,是所有遗传算法的核心部分。适应度函数通过为群体中的每个个体分派适应度分数并以此来推动进化过程。在大多数的遗传算法中这个分数用于决定哪一个个体将会幸存到下一次迭代并且产生后代。收敛到一个较优的解决方案关键取决于这个适应度函数的基因组质量的表现,但是由于音乐是一个不明确的概念,因此很难进行精确地限定。

然而,当评价的范围有限时,适应度函数能有效地推动一个群体收敛于高质量的解决方案中。Freitas 和 Guimaraes 在 2011 年使用双适应度函数的方法,记下从他们的系统中输出的和声的分数。其中,一个函数基于旋律个体的简单性和是否遵守通常的和声

规则而记下输出的分数,另一个则是基于旋律个体的不和谐音水平。经过多代和声之后,幸存的个体通过至少这两个适应度函数之一展示出得分较高的个体的特性,说明了两种不同的适应度函数如何能够产生两种类型的输出。

一般情况下,适应度函数的选取主要有以下两种方式:

(1) 使用人作为适应度函数

遗传算法作曲中,若使用人作为适应度函数,则称为交互式遗传算法。这种方法根据人对个体的评估来决定进化过程,容易受评价者的喜好、情感、心理特征以及直觉的影响,有很大的主观性,而且评价的过程耗费的时间比较长,评价者容易产生疲劳,也即是适应度瓶颈问题。这一问题使得交互式遗传算法在应用时具有较大的局限性。

(2) 设计进化规则作为适应度函数

这种方式需要在系统中预先设计规则来编码音乐知识,然后据此设计适应度函数自动评价产生的旋律,再根据适应值判断是否需要进行交叉和变异操作。这一过程中希望人的干预达到最小。

本文中采用第二种方式来设计适应度函数,以避免定性评价及适应度瓶颈问题。选择能够表现旋律的合适特征可以在大多数情况下使得很多用于处理旋律的计算方法变得容易。例如:Yip 等人证明了在编目旋律时从旋律中提取特征非常有意义^[47]; McKay 基于音乐特征提取(使用后来公布在一个开源程序库中的软件)创建了一个成功的音乐类型分类系统^[48]; Brown 创立了使用音乐特征的组合产生在美感度方面令人愉悦的旋律的衍生法^[49]。

在创建适应度函数时使用到了一组特征提取器。其中,特征提取器为适应度函数提供输入,并计算群体中各个体的适应度分数。每一个特征提取器分析种群中的个体 I 和计算函数 $e: \Pi \rightarrow [0,1]$, 其中 Π 是所有个体的集合。

该函数的输出反映了特殊的特征在个体中表现的良好程度。举一个简单的例子,一个特征可以从调号为 G 大调的乐段中计算出表示该特征的音符的百分比,并返回这个百分比作为输出。

这里介绍一下在描述特征提取器时本文中将会使用到的记号法。一个个体 $I \in \Pi$ 是一个音符序列, $I = i_1 i_2 i_3 \dots i_n$ 。单个的音符 i 可以取四八度音域内的任何音高值,每一个值表示为区间 $[0,48]$ 中的一个数,因此 $i_j \in [0,48], 1 \leq j \leq n$ 。

本文中实现特征提取器需要遵循如下规则:

1) 自相似性: 测量在 I 中出现的重复音程序列的频繁程度并且将其作为自相似性的一种测量——如果相同的音程重复出现, 与那些如果有很多音程但出现不频繁的旋律相比, 这首旋律是更自相似的。

$$Ss = \max \left\{ 1, \frac{2w}{|I|} \right\} \quad (3-1)$$

其中, $w = \frac{1}{|L|} \sum_{r \in L} count_r(I)$, 并且 L 是出现在 I 中所有音程序列长度为 2 的集合, 而且 $count_r(I)$ 是出现在 I 中时间音程序列 r 的数量。

2) 旋律图形: 由特殊的旋律图形参数化的五种函数的集合。旋律图形参数可以取以下五种类型值之一: 平行, 上行, 下行, 上弧, 下弧。

$$St(I) = (1 - \frac{g_{\max} - g}{2g_{\max}}) \left(1 - \frac{\varepsilon^2}{\varepsilon^2 + 10000} \right) \quad (3-2)$$

其中, $g = i_n - i_0$ 是 I 的群体斜率, ε 是图形类型和 I 之间的均方误差(使用线性回归进行计算), $g_{\max} = \frac{Nr}{|I|}$ 且 $Nr = 49$ 。

3) 线性度: 测量 I 中各音符的平滑度。在 I 中的每一个音符上粗略估计其二阶导数, 使用音符的绝对值来计算其近似值。 θ 是用来调整是否迅速线性接近 1 的一个平滑术语。

$$L(I) = \frac{\theta P(I)^2}{\theta P(I)^2 + 1} \quad (3-3)$$

其中, $P(I)$ 是一个二阶偏导数的近似值, 类似于一个拉普拉斯核:

$$P(I) = \sum_{j=2}^{n-1} |\alpha i_{j-1} + \varphi i_j + \alpha i_{j+1}| \quad (3-4)$$

4) 调号普及率: 每一个可能的调号集合中有 12 个函数。从 I 中计算表现那个调号的音符所占的比例。

$$Kp_t(I) = \frac{|K_t|}{|I|} \quad (3-5)$$

其中, $K_t = \{i \in I | i \in Key_t\}$, $1 \leq t \leq 12$, 并且 Key_1 是 C 大调, Key_2 是 G 大调, $\dots\dots Key_{12}$ 是 F 大调。

5) 音域: 记下 I 中使用的音符所占的音域的分。得分为 0 意味着没有使用音域而得分为 1 则意味着使用了整个音域。 $T(I)$ 在 I 覆盖的四八度音域内计算音符的百分比。

这里使用一个非线性缩放因子 δ ，它使得前两个八度的使用与第三个和第四个八度相比加权值更大。

$$Pr(I) = \frac{\delta T(I)^2}{\delta T(I)^2 + 1} \quad (3-6)$$

6) 音程类普及率：类似于 Kp_0 特征。上升或者下降的音程返回相同的值并且超过八度的音程减小为它们在八度之内的等价量。这样便产生了 12 个分离函数。其中， $0 \leq j \leq 11$ 。

$$Ip_j(I) = \frac{\sum_{k=1}^{n-1} \gamma(j, |i_{k+1} - i_k| \bmod 12)}{n-1} \quad (3-7)$$

由于进化的压力能够最大化适应度，因此需要一种方法把任何特定的特征分数(而不是仅仅对任何给定的特征推动群体达到 1)作为目标。本文使用一个包裹函数，把目标值 t 与特征提取器 $e_f(I)$ 的值进行比较并且返回一个在 $[0,1]$ 区间内的值，值越高表明匹配越好：

$$Fs(t, e_f(I)) = \frac{-1}{(y(t) - t)^2} (e_f(I) - t)^2 + 1 \quad (3-8)$$

$$\text{其中,} \quad y(t) = \begin{cases} 1, & \text{if } t < 0.5 \\ 0, & \text{if } t \geq 0.5 \end{cases} \quad (3-9)$$

多目标适应度函数是通过将来自特征提取器的加权输出进行线性组合得到的，而且偏向于旋律结果，其中权值充当了“偏爱物”。因此， $f(I) = \sum_{e \in E} \alpha_e Fs(t_e, e_f(I))$ 是一个表现偏爱旋律风格的多目标适应度函数，由目标值集合 $\{t_e\}$ 和权值集合 $\{\alpha_e\}$ 对其进行参数化。这里， E 是特征提取器的集合， t_e 是目标特征值而 α_e 用于加权提取器。要注意的是这两种参数的集合之间有一些相互影响但是作用于不同的函数。目标值决定着不同音乐特征的质量，而权值则决定着它们的重要性。例如，考虑下面的函数：

$$f(I) = \frac{1}{4} Fs(0.4, Pr(I)) + \frac{3}{4} Fs(0.9, Kp_2(I)) \quad (3-10)$$

这个函数记下大部分得分很高的音乐，它们大量使用 F 大调并采用一个适中的音域。调号特征的重要性是音域特征的三倍并且不考虑其他的特征。本文中使用公式(3-10)来计算每首旋律的适应度函数值。当然，特征的非线性组合和负加权也是旋律风格的潜在表现形式。然而，这里考虑的仅限于线性的、正加权的情形。

当使用遗传算法进行旋律创作时，终止条件的选择与所采用的适应度函数相关。当使用人作为适应度函数时，根据评价者对旋律的感受作为评价旋律优劣的标准；当设计进化规则作为适应度函数时，终止条件可以是事先设定的最大进化代数，也可以是进化后该旋律与经典旋律的相似程度，等等^[50]。本文中采用预先设定的适应度函数值 T 作为阈值，以此判断旋律是否需要进化。大量实验结果表明， T 取 0.75。

针对图 3-5 中的 8 首旋律，分别对其按照公式(3-10)进行适应度函数值计算，所得结果见表 3-5。

表 3-5 8 首旋律的适应度函数值	
旋律	适应度函数值
Music1	0.43
Music2	0.48
Music3	0.35
Music4	0.26
Music5	0.31
Music6	0.29
Music7	0.42
Music8	0.39

根据表 3-5 可知，这 8 首旋律经计算得到的适应度函数值均小于阈值，因此需要进行进化操作。

3.5 进化操作

在给每首旋律计算适应度函数值之后，若不满足输出的条件，则需要对旋律进行交叉和变异操作，以达到进化的目的。

3.5.1 交叉操作

常用的交叉算子为单点交叉和两点交叉。根据前面所述的旋律编码方式，本文中交叉时主要是音级或者时值的交叉，这里交叉概率取为 0.6。

本文中交叉分为音级交叉、时值交叉、音级和时值同时交叉三种。如图 3-6 所示。

对于每一首旋律，给定了拍号，则每一小节内所有音符的时值的和通常是固定的。比如 4/4 拍，每一小节中有 4 拍。通常，将四分音符设定为 1 拍，则十六分音符为 0.25 拍，八分音符为 0.5 拍，二分音符为 2 拍，全音符为 4 拍，其余的可以依次通过计算得

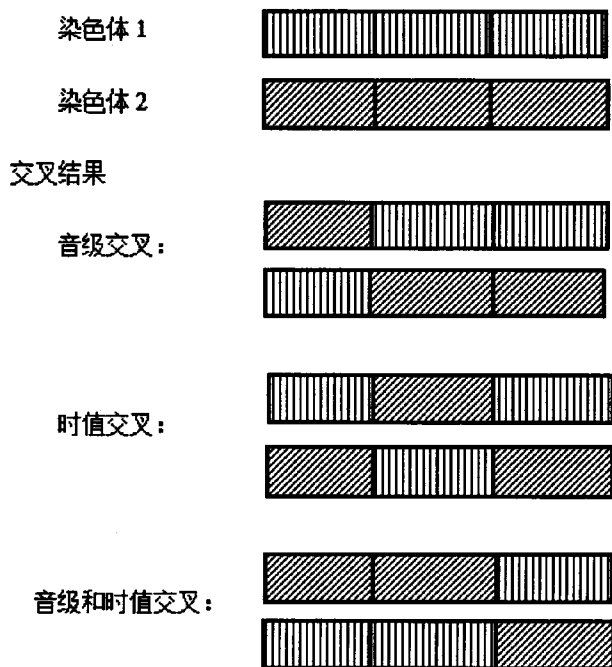


图 3-6 交叉操作

到。当进行交叉操作时，由于交叉是随机发生的，交叉后音符的时值可能会发生变化，打破原来音符的时值之和所具有的平衡。这种情况是不允许发生的，因此必须对交叉后的音符时值作进一步的修正操作，具体步骤如下：

(1) 以拍为计算单位，计算出乐段中每一小节内各个音符的时值并对其求和；

(2) 给定旋律的拍号，将每一小节中应具有的标准拍数之和设为 S ，进行交叉操作后得到的实际的拍数之和为 S' 。将 S 与 S' 进行比较：

- ① 若 $S' = S$ ，则不进行修正，直接进行下一步的操作；
- ② 若 $S' > S$ ，则需要对音符的时值进行压缩。假设需要进行压缩的音符的时值为 a ，如果 $a > S' - S$ ，则需要将 a 进行压缩，直至 $a = S' - S$ ；如果 $a < S' - S$ ，则将该音符删除并寻找下一个需要进行压缩的音符，重复此过程直至 $S' = S$ 。
- ③ 若 $S' < S$ ，则需要增加音符的时值。将主音(大调式的主音为 1，小调式的主音为 6)作为待增加的音符，时值取为 $S - S'$ ，并将其置于各小节的尾部。

当寻找需要进行压缩或者删除的音符时，首先考虑的是不稳定音符，这是因为在一首旋律中不稳定音符没有稳定音符那么重要，持续时间也不会太长。大调式中稳定音为 1、3、5，不稳定音为 2、4、6、7，其中 1 为主音。小调式中稳定音为 6、1、3，不稳

定音为 7、2、4、5，其中 6 为主音。当进行压缩或者删除操作时，对于大调式，音符选择的顺序为 7、6、4、2，对于小调式，顺序为 5、4、2、7。如果某一小节中不含不稳定音符，则可以考虑除主音外的稳定音符，选择顺序大调式为 5、3，小调式为 3、1，注意，主音不能删除。

3.5.2 变异操作

本文中按照变异概率进行变异操作，这里变异概率取 0.008。

根据上面所述的编码方式，变异是对音级和时值的同时变异。由于在变异的范围内对时值的任何变化都符合规则，因此对于时值的变异可以不作过多的约束。但对于音级，必须保证相邻音符的音程不超过 8 度。

假设需要进行变异操作的音符的音级为 S_r ，变异后的音级为 S_r' ，与 S_r 相邻的前后音级分别为 S_l 、 S_r ，比 S_r 低 8 度、高 8 度的音符的音级分别记为 S_{r-8} 、 S_{r+8} ，比 S_l 低 8 度、高 8 度的音符的音级分别记为 S_{l-8} 、 S_{l+8} 。

(1) 若 $S_r = S_l$ ，则 $S_r' \in [S_{r-8}, S_{r+8}]$ ；

(2) 若 $S_r > S_l$ ，则 $S_r' \in [S_{r-8}, S_{l+8}]$ ；

(3) 若 $S_r < S_l$ ，则 $S_r' \in [S_{l-8}, S_{r+8}]$ 。

根据这三个式子，即可得出 S_r' 的取值范围。

变异操作如图 3-7 所示。

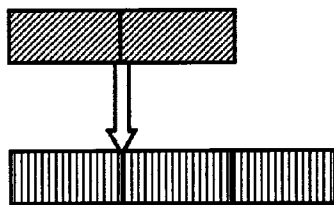


图 3-7 变异操作

将图 3-5 中的 8 首旋律按照上述进化操作进行进化，每进化一次需要再进行适应度函数值计算并判断是否满足输出条件，重复此过程。当进化 47 代后，所得乐谱图像如图 3-8 所示。

(a) Music1

(b) Music2

(c) Music3

(d) Music4

(e) Music5

(f) Music6

(g) Music7

(h) Music8

图 3-8 进化 47 代后的 8 首旋律乐谱图像

此时, 对这 8 首旋律再进行适应度函数值计算, 所得结果见表 3-6。

表 3-6 进化后 8 首旋律的适应度函数值

旋律	适应度函数值
Music1	0.83
Music2	0.88
Music3	0.82
Music4	0.77
Music5	0.85
Music6	0.86
Music7	0.84
Music8	0.81

由表 3-6 可知,这 8 首旋律的适应度函数值均大于阈值,达到输出要求,此时,可将旋律进行输出。为了判断输出的旋律是否令人满意,本文中采用三组人员对生成的旋律进行评估,每组 10 人。三组人员分别是作曲家、普通听众和特殊听众,其中,特殊听众是指旋律所直接面向的人员,如《军队节奏》是面向军人的旋律,而《教师礼赞》则是面向教师的旋律。分别让这三组人员对进化前后的这 8 首旋律进行评估,满分为 10 分,并计算每首旋律的平均分,结果见表 3-7 和表 3-8。

表 3-7 进化前 8 首旋律的评估分数

旋律	作曲家	普通听众	特殊听众	平均分数
Music1	4	3	3	3.33
Music2	4	5	4	4.33
Music3	3	4	5	4.00
Music4	2	3	3	2.67
Music5	3	5	4	4.00
Music6	3	4	4	3.67
Music7	3	5	3	3.67
Music8	4	3	4	3.67

表 3-8 进化后 8 首旋律的评估分数

旋律	作曲家	普通听众	特殊听众	平均分数
Music1	7	5	7	6.33
Music2	8	6	7	7.00
Music3	6	7	5	6.00
Music4	5	6	5	5.33
Music5	7	5	6	6.00
Music6	7	6	6	6.33
Music7	7	7	5	6.33
Music8	7	8	6	7.00

根据表 3-7 和表 3-8 中计算得到的平均分数可知,进化后这 8 首旋律大体令人满意,也即使用本文构建的适应度函数可使得旋律进化结果比较明显。若在遗传算法中采用人作为适应度函数,需要人对生成的旋律进行评估,根据评估得到的分数来判断是否需要进化。每个人评估一首旋律所需时间大约是 3 分钟,若要评估这 8 首旋律,则需要 24 分钟。而通过本文设计的适应度函数来评价时,5000 首旋律只需要 10s,评价这 8 首旋律则只需 0.016s,大大减少了评估所用的时间,提高了创作的效率。

第四章 总结与展望

4.1 总结

本文中采用基于遗传算法的方法创作旋律，所做的主要工作总结如下：

(1) 介绍了将遗传算法应用于旋律创作时生成初始群体的几种人工智能方法，如数学模型的方法、人工神经网络的方法等。这些方法在创作旋律的过程中有一定的局限性：数学模型的方法对旋律中蕴含的数学关系过分依赖，生成的旋律也缺乏主题性；人工神经网络的方法得到的旋律往往取决于在训练时所使用的数据，等等。本文中为了实时产生旋律采用基于数学模型的方法来生成初始群体。由于采用该方法生成的群体具有过度随机的可能性，因此，在生成初始群体时对旋律的相关参数进行了限制，如调号、拍号、调式、音色等。

(2) 对初始群体中所有个体进行编码时，采用了实数编码方法。对旋律的调号、音符的音级、音名、音符时值等的编码方式作了规定。编码时，以个体中的每个音符为单位，根据该音符的音级、音符时值和符号进行编码。其中，设计的符号位是用于辨别该音符是否处于某一小节的尾部。该编码方法可以直观形象并且正确地表现旋律的信息。

(3) 着重研究了用于自动评估生成旋律的适应度函数。本文中构建了用于提取生成旋律的乐谱特征(如旋律中各音符的自相似性，旋律的图形，音符的线性度，所使用的音域等)的提取器，通过一个包裹函数将目标值与计算得到的特征值进行比较并返回 $[0,1]$ 区间内的值，再对该包裹函数赋予权值并进行线性组合作为多目标适应度函数。之后使用该多目标适应度函数对每个个体进行适应度计算，并判断个体是否需要进化。

(4) 对进化操作中的交叉和变异作了研究。交叉和变异操作分别是在预先设定的概率下进行的。交叉操作包括音级交叉，音符时值交叉以及这两者的同时交叉，讨论了时值修正问题，确保交叉操作前后每一小节中音符的时值之和保持不变。变异操作采用了对音级和音符时值的同时变异，其中，对音符时值的变异不加限制，但对音级进行变异时相邻音符的音程不能超过8度。

(5) 本文中构建的基于遗传算法的旋律创作系统是在VC++平台下实现的。实验结果表明，通过多代进化操作，该系统在生成具有较高质量的旋律的前提下，缩短了评估

所用的时间，提高了旋律创作的效率，而且旋律的进化效果也比较明显。

4.2 研究展望

本文中构建的基于遗传算法的旋律创作系统具有严密的全局结构，生成旋律的长度也可以进行事先指定，但是对旋律的和声以及复调等复杂的音乐元素却没有多作研究，之后的工作重点将转移为把些元素添加到构建的系统中去。适应度函数的构建还需进一步完善。当使用特征提取器提取生成旋律的特征时，所选用的特征需更能表现出旋律的特点并有助于进化操作。遗传算子的设计也需改进，以便使得生成的旋律在美学方面更加容易为人所接受。

参考文献

- [1] 孙浩. 计算机音乐编程的基本类型研究[D]. 北京: 中央音乐学院, 2012.
- [2] 谢力荣. 计算机作曲中存在的问题及对策探究[J]. 课程与教学, 2013, (3): 91-94.
- [3] 王文. 数字技术与音乐创作——计算机音乐技术特征与应用[J]. 教育教学论坛, 2013 (34): 251-252.
- [4] 曹西征, 毛文涛, 乔锐, 等. 基于音高旋律元的柔和乐曲的自动作曲算法[J]. 自动化学报, 2012, 38(10): 1627-1638.
- [5] 于立功, 卜佳俊, 陈纯. 计算机音乐研究初探[J]. 计算机工程与应用, 2005, (4): 66-68.
- [6] 张英俐, 刘弘, 李少辉. 遗传算法作曲中的应用[J]. 计算机应用研究, 2005, 11: 143-145.
- [7] 冯寅, 周昌乐. 算法作曲的研究进展[J]. 软件学报, 2006, 2: 209-215.
- [8] 张英俐, 刘弘, 宋宝亚. 一种交互式遗传算法生成带主题乐曲的方法[J]. 数学的实践与认识, 2012, 24(16): 143-150.
- [9] Freitas, A., Guimaraes, F. Melody harmonization in evolutionary music using multiobjective genetic algorithms. In Proceedings of the Sound and Music Computing Conference. 2011.
- [10] Matić D. A genetic algorithm for composing music. The Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043, 2013, 20(1).
- [11] Biles, J. Genjam: A genetic algorithm for generating jazz solos. In Proceedings of the International Computer Music Conference, 1994, 131-137.
- [12] 黄虎威. 四部和声写作规则之我见[J]. 音乐探索, 2011 (4): 17-20.
- [13] Grout DJ. History of Western Music. 5th ed., New York: W. W. Norton & Company, 1996.
- [14] Järveläinen H. Algorithmic musical composition. 2000. <http://www.tml.tkk.fi/Studies/Tik-111.080/2000/papers/hanna/alco.pdf>.
- [15] 高济, 朱森良, 何钦铭. 人工智能基础[M]. 北京: 高等教育出版社, 2002: 249.
- [16] Nagelberg A. MICA: A Hybrid Method for Corpus-Based Algorithmic Composition of Music Based on Genetic Algorithms, Zipf's Law, and Markov Models. The University of Manitoba, 2014.
- [17] Yu P, Li H, Zhou H, et al. Palmprint Recognition Based on Subclass Discriminant Analysis. Computer Engineering and Networking. Springer International Publishing, 2014: 465-472.
- [18] An J, Wang Z, Liu Z, et al. Collaborative Logistics Service Composition Based on Co-Evolutionary Genetic Algorithm. Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 2. Springer Berlin Heidelberg, 2014: 11-17.
- [19] 张英俐, 刘弘, 马金刚. 遗传算法作曲系统研究[J]. 信息技术与信息化, 2005, 5: 106-108.
- [20] Gonsalves T, Kawai A. USER PREFERRED COLOR COMBINATION DESIGN USING INTERACTIVE GENETIC ALGORITHM[J]. Computer Science, 2014.

- [21] Biles J A. GenJam in transition: From genetic jammer to generative jammer. 2002. <http://www.generativeart.com/papersGA2002/8.pdf>.
- [22] Aladdin Ayesha, Andrew Hugill. Genetic Approaches for Evolving Form in Musical Composition.(453-228).Artificial Intelligence and Applications. 2005: 318-321.
- [23] George Papadopoulos, Geraint Wiggins. A Genetic Algorithm for the Generation of Jazz Melodies.The Music Informatics Research Group. 1998.
- [24] Muneyuki Unehara, Takehisa Onisawa. Interactive Music Composition System. IEEE, 2002.
- [25] Unemi T. A tool for multi-part music composition by simulated breeding. In: Gedau A, ed. Artificial Life VIII. Cambridge: MIT Press,2002: 410-413.
- [26] 韩艳玲. 可能性构造空间理论与计算机作曲思维模型研究[D]. 武汉: 华中科技大学, 2010, (8).
- [27] Kazi N, Bhatia S. Various Artificial Intelligence Techniques For Automated Melody Generation[J]. International Journal of Engineering, 2013, 2(7).
- [28] 刘嘉璐. 论弦乐四重奏的演奏特点和注意技巧以及写作的技术问题[J]. 经济与社会发展, 2012, 10(5): 117-119.
- [29] Howard D M, Daffern H, Brereton J. Four-part choral synthesis system for investigating intonation in a cappella choral singing. Logopedics Phoniatrics Vocology, 2013, 38(3): 135-142.
- [30] Anders, T., and Miranda, E. R. A computational model that generalizes Schoenberg's guidelines for favorable chord progressions. In Proceedings of the Sound and Music Computing Conference. 2009.
- [31] Tanaka ,T.; Nishimoto,T.; Ono, N.; and Sagayama, S. Automatic music composition based on counterpoint and imitation using stochastic models. In Proceedings of the Sound and Music Computing Conference. 2010.
- [32] 林华. 音乐符号的内涵文法与外延文法[J]. 上海音乐学院学报, 2011 (2): 53-60.
- [33] Cope D. Virtual Music: Computer Synthesis of Musical Style.Cambridge: MIT Press, 2001.
- [34] J. J. Bharucha and P. M. Todd. Modeling the Perception of Tonal Structure with Neural Nets. In P. M. Todd and G. Loy, editors, Music and Connectionism,128-137. MIT Press, 1991.
- [35] 殷波. 论音乐表演的直觉及素质[J]. 华章, 2013 (10).
- [36] Mozer MC. Neural network composition by prediction: Exploring the benefits of psychophysical constraints and multi-scale processing. Cognitive Science, 1994,6:247-280.
- [37] 刘军伟, 余小清, 万旺根, 等. 基于改进型 BP 神经网络的音频多分类[J]. 上海大学学报(自然科学版), 2012, 18: 127-131.
- [38] 张超群, 郑建国, 钱洁. 遗传算法编码方案比较倡[J]. 计算机应用研究, 2011, 28(3).
- [39] 杨瑞庆. 实用旋律学初探[M]. 上海: 上海音乐出版社, 2006: 3-10.
- [40] 张英俐. 基于遗传算法的作曲系统研究[D]. 济南: 山东师范大学, 2006.
- [41] 王璐. 浅析节拍与节奏在音乐表现中的作用[J]. 新一代: 理论版, 2012 (12): 178-178.
- [42] 卿武明. 一种基于遗传算法的作曲方法研究[D]. 昆明: 昆明理工大学, 2008.
- [43] Deb, K., and Kalyanmoy, D. Multi-Objective Optimization Using Evolutionary Algorithms.Wiley,

2001, 1st edition.

- [44] Fernández J D, Vico F. AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*, 2013, 48: 513-582.
- [45] Dostál M. Musically meaningful fitness and mutation for autonomous evolution of rhythm accompaniment. *Soft Computing*, 2012, 16(12): 2009-2026.
- [46] 曹西征, 张爱丽, 徐久成. 基于遗传算法的智能作曲技术研究[J]. *计算机工程与应用*, 2008, 44(32): 206-209.
- [47] Yip, C. L., and Kao, B. A study of musical features for melody databases. 1999. In Bench-Capon, T. J. M.; Soda, G.; and Tjoa, A. M., eds., *Database and Expert Systems Applications*, 10th International Conference, DEXA '99, Florence, Italy, August 30 - September 3, 1999, Proceedings, volume 1677 of *Lecture Notes in Computer Science*, 724-733. Springer.
- [48] McKay, C., and Fuginaga, I. jSymbolic: A feature extractor for MIDI files. In *Proceedings of the International Computer Music Conference*, 2006, 305-305.
- [49] Brown, A. R. An aesthetic comparison of rule-based and genetic algorithms for generating melodies. *Organized Sound*, 2004, 9(2): 193-199.
- [50] Murray S J, Ventura D. *Algorithmically Flexible Style Composition Through Multi-Objective Fitness Functions*. Brigham Young University. Department of Computer Science, 2012.

此页不缺内容

致 谢

光阴似箭，日月如梭。两年的研究生生活转瞬即逝，这一段既愉快而又充满挑战的学习经历也将告一段落。回首既往，奋斗和辛劳成为丝丝的记忆。在我的毕业论文即将完成之际，向所有给予我关心与帮助的师长、朋友和同学表示最诚挚的感谢与最美好的祝愿。

首先，衷心而诚挚地感谢我的导师曹西征副教授。在两年的时间里，导师给了我很多帮助和指导，不仅帮我确立正确的研究方向，还教会我用科学的方法进行学习与研究。从论文的选题、研究、实验，再到后来的审查直至完成，都是在曹老师的指导下完成的，这其中凝聚了曹老师的大量心血，在此谨向曹老师表达我崇高的敬意和最衷心的感谢。另外，曹老师深厚的专业知识，严谨的治学态度，精益求精的工作作风，让我深刻认识到了什么是一个真正的学者，同时他朴实无华、平易近人的人格魅力也使我深受感染。作为曹老师的学生，我心中充满了无比的自豪。

同时，衷心地感谢我读研究生期间的各位老师，正是由于你们的谆谆教诲，我才能不断地学习新的知识，不断地取得进步。这一切都是计算机学院的所有老师无私奉献和帮助的结果。

特别感谢我的父母，是他们无私的支持和关爱促使我能够完成学业，他们是我不断取得进步的永恒能力。

最后，衷心感谢各位专家、老师不辞辛劳地评阅本论文，恳请各位专家、老师批评指正。

刘艳梅

2014年4月

此页不缺内容

攻读学位期间的科研成果

撰写论文:

- [1] Xizheng CAO, Lin SUN, Jingwen NIU, Ruiqi WU, Yanmei LIU, Huijuan CAI. Automatic composition of happy melodies based on relations[J]. Multimedia Tools and Applications. (SCI 源刊, 已录用).
- [2] 蔡会娟, 曹西征, 刘艳梅, 张宁. 基于一种 BP 改进算法的大学生学习成绩评价模型构建[J]. 郑州师范教育. (已录用)

参与项目:

- [1] 面向学前儿童的算法作曲系统开发研究, 河南省重点科技攻关项目, No. 102102210177, 已鉴定: 国内领先, 第 6 参与者.
- [2] 面向强迫症音乐治疗的计算机作曲关键技术研究, 河南省重点科技攻关项目, No.122102210054.
- [3] 面向音乐治疗的自动作曲算法研究, 河南省高校青年骨干教师项目, No.2011GGJS-061.
- [4] 基于情感的算法作曲的工程实现研究, 河南省教育厅科技攻关项目, No. 2010A520023.
- [5] 数字音频模式下的计算机作曲技术研究, 河南省教育厅基础研究项目, No.2011A520020.

参与编写著作:

- [1] 《Oracle 数据库技术与应用》, 科学出版社, 2013.8.

此页不缺内容

独 创 性 声 明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得河南师范大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名： 刘艳梅 日期： 2014.5.29

关于论文使用授权的说明

本人完全了解河南师范大学有关保留、使用学位论文的规定，即：有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河南师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。（保密的学位论文在解密后适用本授权书）

作者签名： 刘艳梅 导师签名： 曹红 日期： 2014.5.29