

浙江工业大学

本科毕业设计文献综述

(2017 届)



论文题目 基于遗传算法的音乐作曲
研究与实现

作者姓名 陈文帆

指导教师 盛伟国

学科(专业) 计自 1302

所在学院 计算机科学与技术学院

提交日期 2017 年 3 月

模因作曲

Enrique Muñoz, Member, IEEE, Jose Manuel Cadenas, Senior Member, IEEE,

Yew Soon Ong, Senior Member, IEEE, and Giovanni Acampora, Senior Member, IEEE

摘要——计算机与人工智能在艺术创作中扮演着关键角色，通过设计人造代理，能够再现人类艺术家高品质的艺术品，比如绘画与雕塑。在这方面，音乐作曲代表了能够从计算智能中极大获益的艺术学科之一，近年来在该领域进行的大量研究就是最好的证明。然而，由于人类音乐家的天才与能力，自动作曲还远远不能完美地演艺。本文通过提出一个有效的、基于音乐方法的旋律生成智能方案，减少了人类与机器作曲之间的差距。该方法受到人类作曲技巧：非数字低音技术的启发。特别地，我们将作曲技术表示成一个优化问题，并利用包含各种元启发式的自适应多代理模因方法求解它。作曲代理以低音线作为输入，协作创作高品质的音乐片段。一系列基于巴赫四音合唱的实验研究表明，不同优化策略的混合表现比传统的混合进化算法更好。

关键词——自适应模因算法，自动作曲，多代理系统

一、简介

历史上，人工智能与计算智能方法学已经普遍应用于艺术家/科学家的创作与科研活动。其中一个引人注目的成就包括人造代理的设计，其目标是再现人类艺术家创作高品质艺术品的技艺，比如绘画和雕塑等等。这项技术，称为计算机艺术，它代表了计算机应用领域的一个真正的历史性突破，因为

它第一次试图复制纯粹的、人类所独有的能力：创造力。自从1965年，第一次计算机艺术展在纽约和斯图加特同时举行，人们已经在实现所谓的创造性算法方面取得了许多进步，以支持（替代）艺术家的创作。

如今，人工智能已经涵盖了大量艺术创作。其中，音乐是将智能方法应用于数据处理并能从中极大获益的代表性艺术学科，正如过去几十年间在该领域进行的大量研究所见证的（见第二节B）。事实上，对于解决不同音乐问题的研究工作可能可以追溯到计算机时代之前，比如：记谱程序、声音合成、实时演艺、数字乐器等等^[1]。然而，在这些音乐问题中，算法（自动）作曲，对于计算机科学家和音乐家而言绝对都是最迷人的。事实上，许多不同的计算机技术都已经应用于该极具挑战性的领域：随机数，正式乐理语法，自动划分，分形等等。特别地，近来^[2]，进化音乐被认为是一种新形式的计算机艺术方法，其重点在于通过进化算法（EA）来评估以及演化给定的音乐作品。因此，基于遗传算法（GA），遗传编程，粒子群优化（PSO）以及进化策略的音乐作曲技术，在计算机艺术领域得到越来越普遍的应用，并生成一些有趣的结果^[1]。

尽管如此，由于两个关键性问题，基于进化方法的自动作曲距离完善还很遥远：1）人为再现人类作曲家的天才能力的巨大挑战，2）对自动生成的旋律作美学评价的困难性。这些难点在生成技术上具有极大的相关性，比如非数字低音技术，人类作曲家为低音创造旋律，而不需要指定复调或和弦，

演奏者即兴演奏这些和弦（次中音，中音，高音）以更好地调整演奏出整段音乐的和谐。

本文的研究旨在提高进化算法在自动生成音乐旋律方面的能力，通过引入一个多代理系统，其包含了联合适应模因，解决了非数字低音合成技术在创作技巧与美学评估方面的缺陷。从科学的角度讲，基因指示蛋白质的合成，而模因是包含执行指令的基因的社会文化等价物^[4]。受大自然的启发，此处，模因被建模为指定搜索过程的指令。特别地，我们的系统拥有基于种群和基于局部的元启发式的模因集合，此处标记为模因作曲代理，使用基于模糊决策树（FDT）的机会学习机制，能够评估每个作曲代理在生成其独有的非数字低音的强度和成熟度。在执行学习和作曲过程中，每个模因作曲代理运用基于著名的和谐规则1的创新音乐适应函数，这使得每个代理能够以精准正式的方式评估其生成的音乐作品。接着，作曲代理间使用挖掘得到的知识来合作交换信息，以有效地探索三音旋律的线间空白并进行乐曲合成，这样得到的乐曲明显优于从单个作曲代理处获得的原始乐曲²。同时，模因间的协作与竞争（为固定的访问）促进了搜索过程中音乐片段的形成（不可能由单个作曲代理得到）。

如将在实验结果部分所示的，不同模因或元启发策略间的协作，允许我们的人造作曲代理对由传统的、最先进的（混合或非混合）进化算法获得的著名的巴赫四音生成曲，生成改进的解决方案并提高性能。

本文的余下部分安排如下。第二部分，是关于模因算法（MA）和进化音乐的简短报告。第三部分，展示了如何将音乐作曲的非数字低音技术建模成一个优化问题。之后，在第四步分，展示了自适应模式作曲。接下来，第五部分，展示了由提出的模因音乐作曲家（MMC）得到的结果，以及其相对于传统进化音乐技术的优越性。最后，第六部

分，得出结论。附录介绍了一些理解数字化低音问题的必要的音乐概念。

二、相关工作

本节简短地介绍了近年来与模因方法和进化音乐相关的研究活动。

4. MA 简介

MA 是旨在为复杂和困难的优化问题寻找解决方案的元启发式算法^[2]。它是传统进化算法的扩展，其搜索策略包含了局部搜索优化阶段。MA 的出现是为了解决进化算法的一些固有问题，进化算法通常由于未能利用局部信息，而导致要找到一个精确的解决方案，收敛速度极慢。这通常就限制了 EA 在许多计算时间是关键的考虑因素的大规模现实问题上的实用性。

从优化的角度来看^[3]，MA 已经被证明更高效（比如，要求使用较少的评估来找出最优解），同时其效果在多个问题领域较传统 EA 更好（比如，找出更优的解）。以结果论，MA 正被越来越广泛地接受，特别是在众所周知的组合优化问题中。一些组合优化问题已经通过 MA 解决，得到了最优解，而使用其他元启发式方法并不能得出可比较的结果。Krasnogor 和 Smith^[4] 通过收集 MA 在许多著名的组合优化问题的应用，对该领域作了完整的回顾。

然而，尽管 MA 得出的结论很有趣，设计有效且高效的 MA 的过程依旧存在缺陷。比如说，其控制参数微调的困难，这可能需要广泛的测试，具体地，找到一个适合指定问题的模因很困难^[5]。事实上，模因的选择会严重影响 MA 搜索的性能^[6]。这一点已经导致了研究界开发行为能够适应待解决问题特性的 MA，以获得第三代 MA 或自适应 MA。Ong 等人^[7] 展示了自适应

MA 的分类和比较。

B. 自动作曲和进化音乐

旋律的自动生成是计算机艺术领域最具挑战性的问题之一，它几乎与计算机本身一样古老：《伊利亚克组曲》^{[11]·[12]}是由计算机程序生成的音乐片段，其可以追溯到1957年，第一台计算机问世的几年之后。本节，我们将对自动作曲作一个简要回顾，并将重点关注基于进化策略的方法。该领域研究现状的完整回顾已经超出了本文的范畴。感兴趣的读者可以参考参考文献^[13]和^[14]，作更详细的调查。

过去的几年间，数个进化方法被提出，其中一些成功地被应用于生成悦耳的音乐片段。比如说，一个基于遗传算法的自动作曲工具 GenDash 已经被著名的美国作曲家 Waschka 二世^{[15]·[16]}用于他的现场表演。其他用于作曲的遗传方法被提出：Jacob 的系统^{[17]·[18]}介绍了一个基于“耳朵”模型的 GA，用于聆听与评估生成的旋律；相反地，Waschka 的系统，完全忽略了适应度评估的阶段。GaMusic^[19]是一个基于美学 GA 自动合成双八度音阶旋律的进化系统。Johanson 和 Poli^[20]展示了 GP-Music 系统，这是一个能根据导师的偏好与系统自身的自主方法演化短旋律序列的遗传编程方法。Spector 和 Alpern^{[21]·[22]}提出了 GenBebop，这是一个生成即兴短爵士的系

统，即所谓的“trading fours”，即兴演奏四个小节以回应另一个表演者刚刚演奏的四个小节。Biles 探讨了一个类似的方法^{[23]·[26]}，并设计了 GenJam，一个基于 GA 的交互式系统，也是即兴演奏 trading four。在这之后，Thywissen^[27]设计了 GenNotator，一个混合算法的计算机辅助作曲工具。Gibson 和 Byrne^[28]介绍了一个生成全音阶，四部和声的短音乐片段的系统。这个系统叫做 NEUROGEN，其利用 GA 生成音乐，利用神经网络去评估生成结果的品质。Unemi^{[29]·[30]}创造了 SBEAT，一个交互式系统，在具有各种节奏、音高和音速的音乐模式中选择生成乐句（比如，节奏模式）。Ralley 提出了另一个基于交互式 GA 的旋律演化建议。

这里关注一下 De Prisco 和 Zaccagnino^[45]，因为他们试图用 GA 解决数字低音问题。然而，他们并没有用数学的方法将搜索形式化为优化问题。此外，我们提出的进化方法包括了各种元启发式自适应模因代理，比之简单的 GA，能够更有效且高效地发现高品质且创造性的解决方案。特别地，我们的方法采用在线搜索过程中先验执行的元启发式自动提取的知识，因此，能够很好地自适应待解实例。更深入地说，与单个模因代理相比，采用协作与竞争的模式，能取得更好的结果，并促进创意音乐片段的生成，而这在仅采用单个模因代理的情况是无法实现的。

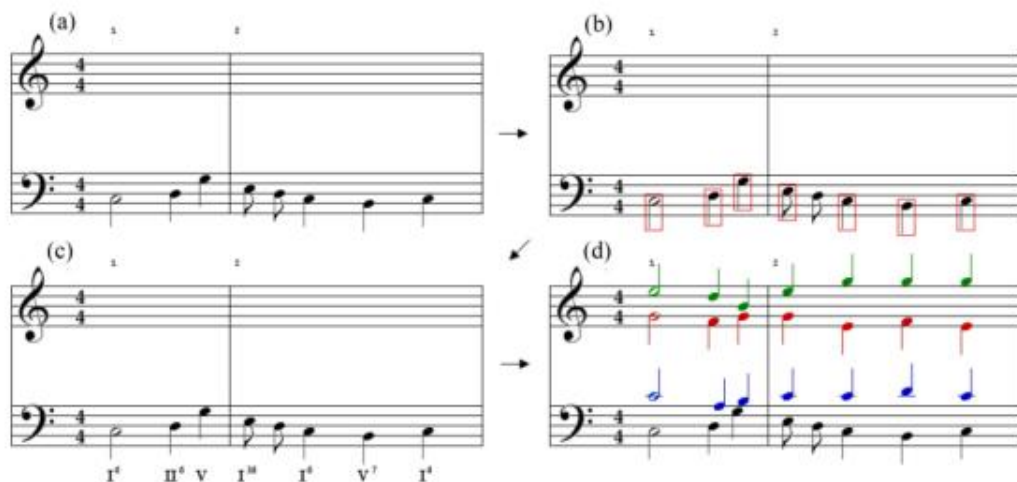


图 1

三、非数字低音作曲作为 NP 难搜索问题

本节，提供了非数字低音技术形式化表示为 NP 难优化问题的方法。如前所述，人类作曲家可以使用非数字低音技术创作低音旋律，但他/她又不必指定和声，比如，对应其他三个音：1) 次中音；2) 中音；3) 高音；然后，演奏乐曲的时候，表演者需要即兴演奏和弦³以保证乐段的和谐。因此，这项作曲技术视为一个优化问题，以一个给定的低音线作为输入，生成最合适的（美学上的）三个音，完成一个四音节。

然而，因为没有关于完成四音节所需的三个音节的先验知识，将非数字低音技术形式化地建模为优化问题，并通过进化进化求解它是相当困难度。事实上，缺少这样的信息，要创建可行解的初始种群是不可能的。因此，用数字标出低音部，对于获取给定低音线的合适初始三音节知识，从而最终将自动作曲定义为拥有其适应函数和约束条件的优化问题是必要的。

图 1 是我们提出的作曲过程的一个例子：(a) 表示输入（低音线）；(b) 标出了

恰当旋律（生成步骤）的符号；(d) 展示了最终的解，比方说，使用进化方法求得。

表 1

	I	II	III	IV	V	VI	VII
I	250	200	50	200	250	50	10
II	100	100	100	150	2000	150	10
III	100	100	100	200	100	250	10
IV	250	150	100	200	1500	100	10
V	2000	100	100	100	250	150	10
VI	100	200	150	150	200	200	10
VII	1000	50	150	50	50	100	200

A. 标注低音部步骤：从非数字到数字低音

标注低音的步骤决定了给定的输入低音线中哪些和弦能够产生最动听的旋律。从优化的角度来看，这个步骤缩小了搜索空间，根据传统和谐规则，将搜索空间从一个包含由低音线得出的所有可能和弦的空间，缩小为一个子集，该子集中的和弦能够产生更好的最终乐曲。准确地说，标注低音步骤是动态规划实现的。

然而，并非所有低音线上的音符都在标注低音步骤的考虑范围内。在标注低音步骤中，分析得到的音符子集，由输入的低音线中的音符决定，同时，生成的节奏的节拍长度，也由低音线决定。节奏通常由时间节拍来指定（图 1 所示的例子，为），其中，表示拍子，表示长度。特别地，我们使用基本时间单元作为一个拍子的长度。一个节拍

中所有的音符都显示在标注低音中。我们也允许音符的长度超过拍子的长度（事实上，大多都是如此）。但是，如果一个输入音符短于拍子长度，那么，我们仅在音符处于节拍上时才考虑它（否则，忽略它）。我们通过 HN 来指定需要做低音标注的音符。

因此，在低音标注步骤中，我们通过为和弦集中的两个连续音符的和弦赋予权重，来为低音线中每个 HN 选择最合适的和弦。这些权重代表著名的和谐规则，表示哪些和弦序列更和谐（见[36]）。那些更普通的和弦序列将被赋予更大的权值，其他的则赋更低的权值。注意，序列中前一个和弦的修改，会改变当前和弦的权值。

我们使用以下标准为每对连续的 HN 分配权值，其中 c_1 和 c_2 分别代表序列中第一个和第二个 HN 的和弦：

1) c_1 和 c_2 是相同的大调和弦。权重的赋值见表 1。2) c_1 和 c_2 是相同的小调和弦。权重的赋值见表 2。3) c_1 和 c_2 是从一个大调到另一个大调，或从一个小调到另一个小调。权重的赋值见表 3。4) c_1 和 c_2 是从一个大调到一个小调，或从一个小调到一个大调。权重的赋值见表 4。

表 2

	I	II*	III*	IV*	V*	VI*	VII*
I	250	200	50	200	250	50	10
II*	100	100	100	150	2000	150	10
III*	100	100	100	200	100	250	10
IV*	250	150	100	200	1500	100	10
V*	2000	100	100	100	250	150	10
VI*	100	200	150	150	200	200	10
VII*	1000	50	150	50	50	100	200

表 3

	C	G	D	A	E	B	G ^b	D ^b	A ^b	E ^b	B ^b	F
C	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500	0
G	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500
D	-500	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000
A	-1000	-5000	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500
E	-1500	-1000	-1000	0	2000	0	-1000	-1000	-1500	-2000	-2500	-2000
B	-2000	-1500	-1500	-5000	0	2000	0	-500	-1000	-1500	-2000	-2500
G ^b	-2500	-2000	-2000	-1000	-500	0	2000	0	-500	-1000	-1500	-2000
D ^b	-2000	-2500	-2500	-1500	-1000	-500	0	2000	0	-500	-1000	-1500
A ^b	-1500	-2000	-2000	-2000	-1500	-1000	-500	0	2000	0	-500	-1000
E ^b	-1000	-1500	-1500	-2500	-2000	-1500	-1000	-500	0	2000	0	-500
B ^b	-500	-1000	-1000	-2000	-2500	-2000	-1500	-1000	-500	0	2000	0
F	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500	0	2000

表 4

	C	G	D	A	E	B	G ^b	D ^b	A ^b	E ^b	B ^b	F
C	-500	-200	0	500	0	-200	-500	-700	-1000	-1200	-1000	-700
G	-700	-500	-200	0	500	0	-200	-500	-700	-1000	-1200	-1000
D	-1000	-700	-500	-200	0	500	0	-200	-500	-700	-1000	-1200
A	-1200	-1000	-700	-500	-200	0	500	0	-200	-500	-700	-1000
E	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200	-500	-700
B	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200	-500
G ^b	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200
D ^b	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0
A ^b	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500
E ^b	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0
B ^b	0	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200
F	-200	0	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500

我们使用动态规划的方法（见算法 1）来计算最合适的和弦集，其元素是可能与第 i 个 HN 相关的和弦 C_i 。该算法输出得到的解，是与原非数字低音线关联的数字低音版。下一节，我们将用数字低音线来定义 NP 难优化问题，其得到的高效解决方案对应于高品质的音乐作曲活动。

B. 将数字低音技术形式化为优化问题

假设集合 V 中，有 n 个 HN，其中每个 HN 都可以被演奏为一个和弦 c_i ，而 c_i 是由低音标注阶段获得的 C_i 的可获得集。我们定义 $c = \{c_1, \dots, c_n\}$ 作为协奏曲，而 $\xi(c)$ 作为通过演奏 c 获得的全局美学收益。数字低音问题需要找到最大化 $\xi(c)$ 的最优协奏曲 c^* 。

Algorithm 1: Pseudo-Code Used to Select the Most Suitable Chords

```

Input: HN: array of  $n$  harmonizable notes.
Output: most suitable chords.
begin
  for  $i = 1; i < n; i++$  do
     $A[i]$  = calculate set of chords that can be
      obtained using HN[i];
  end
  for  $i = n; i > 0; i--$  do
    for Chord  $c \in A[i]$  do
      for Chord  $d \in A[i+1]$  do
        weight = calculate weight using
          Tables I, II, III and IV;
        weight += maxWeight[d];
        if weight > maxWeight[c] then
          maxWeight[c] = weight;
          mostSuitableFollowingChord[c] = d;
        end
      end
    end
  end
  mostSuitableChord[1] = chord with maximum weight
  for HN[1];
  for  $i = 2; i < n; i++$  do
    mostSuitableChord[i] =
      mostSuitableFollowingChord [
        mostSuitableChord [i-1]];
  end
end

```

在这个问题中，我们将 $\xi(c)$ 近似成每个 HN 获得的局部奖励总和。另外，我们可以假设这些奖励依赖于 c_i （为 HN i 选择的和

弦)和从 V 的子集中选择的和弦, 定义为 $T(i)$, 表示 HN_i 的邻域。用于确定每个局部奖励的和弦集合被定义为 c_i , 其从中取值。所以, 我们定义由和弦获得的局部奖励为, 定义全局奖励为

本文中, 我们仅考虑定义在最多两个和弦上的奖励函数。因此, 我们可以将问题表示成无向图 $G=(V, E)$, 其中每个点表示一个 HN , 而每条边表示相关 HN 间的相互影响。因此, 有及。当仅考虑两个和弦间的关系时, 图的边就变成了仅连接两个连续的 HN 。这张图被称为协调图 (HG)。根据以下全局奖励的记法, $\xi(c)$ 可以重写成

其中 是使用和弦 c_i 演奏时, 由 HN_i 得到的奖励, 是由和弦对得到的奖励。

要定义函数和, 我们需要从普通练习时段引入音乐作曲的规则。特别地, 我们考虑一个美学规则集, 其细节可参见表 5。这些规则表明了对违反规则的惩罚, 而不是奖励。因此, 它们是消极的。举例来说, 如果一个音的第四跳做了增强, 将收到值为 100 的惩罚。对一些违规(我们称之为严重错误), 将会受到更严重的惩罚。比方说, 如果一个音跳超过八度或者两个音曝光了平行五度, 可考虑予以 3000 的惩罚。

Single voice error	Cost	Type
sixth	-1000	normal
aug. fourth	-1000	normal
aug. fifth	-1000	normal
seventh	-3000	critical
> octave	-3000	critical
Single voice cost	Cost	Type
jump	-interval	no error
Two voices error	Cost	Type
hidden unison	-100	normal
hidden fifth	-100	normal
hidden octave	-100	normal
unison	-3000	critical
parallel fifth	-3000	critical
parallel octave	-3000	critical
Error within chord	Cost	Type
octave leap	-100	normal

这个问题等价于在一个图模型中寻找随机变量的最大后验配置问题, 一个从蛋白质折叠到图像处理的应用的重要问题。总而言之, 这是一个 NP 难问题 [44]。因此, 确定性方法不可能找到最优解, 或者需要花费太多时间才能找出最优解, 特别是当实例的大小很大时, 比如数字低音问题。解决该问题的一个好的策略是, 搜索具有高品质的、好的近似解, 即使它们不是最优解。在这方面, 也许可以使用近似策略, 比如 MA, 它可以在一个合理的时间内找出好的近似解。本文中, 我们应用该策略以应对数字低音问题表示问题, 使用并调整应用优化代理集的方法, 通过合作, 得到能够自适应问题的不同情况的模因策略。

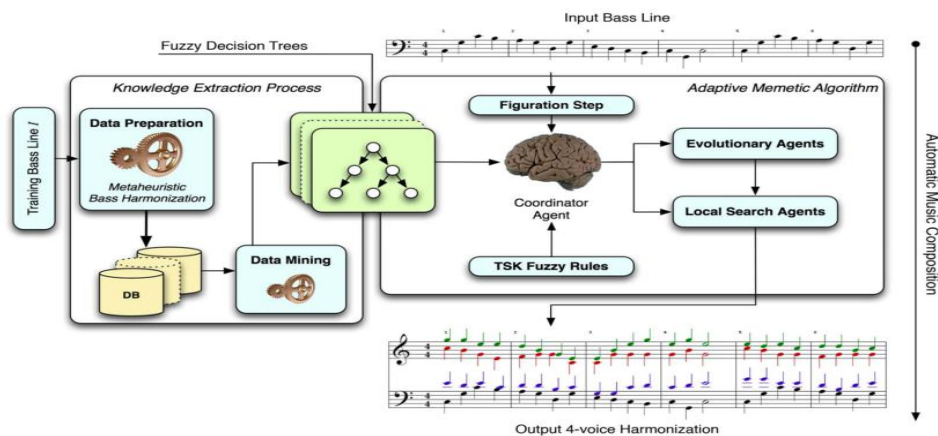


图 2

四、通过作曲代理间的合作与竞

争进行音乐作曲

本节,我们将介绍一个自动作曲的框架(见图 2)。通过解决第三部分介绍的问题,采用通用练习生成规则,其能够得到高质量的音乐片段。

从一个高层次的观点来看,该框架被设计成一个多代理系统,具有两个关键步骤:第一步,一个基于种群的作曲代理以(数字)低音线作为输入,并行地以自适应的方式进行协作,以高效地探索四音音乐片段的空间,找出一个合适的旋律子空间。第二步,一个局部作曲代理并行地以自适应的方式探索由基于人群的代理计算得到的区域,并找出一个高品质的四音音乐片段来完成输入的低音线。特别地,我们使用了六个不同的模因作曲代理,三个进化作曲代理,分别是 GA, PSO, 和蚁群优化 (ACO), 和三个局部作曲代理,分别是禁忌搜索 (TS), 模拟退火 (SA), 和可变邻域搜索 (VNS)。

这个多代理系统的写作和自适应功能由所谓的协调代理提供,能够学习并掌握每一代进化和局部作曲代理在以给定低音线为输入,演化四音旋律的优势与特色。更深入地将,这个协调代理的知识通过一个训练阶段获得,该阶段从包含知识的 FDT 集中习得。协调代理使用包含在 FDT 的知识和一个 Takagi-Sugeno-Kang (TSK) 模糊规则来评估每个代理生成的旋律,并“智能”地在这些旋律间跨种群移动,以提高每个作曲代理的能力。接下来的子节将正式介绍所提出的框架的不同子模块。

A. 学习协调代理知识

为了获得定义了协调代理知识的 FDT,有必要应用一个知识提取过程,来提取并总结有关作曲代理性能的有用信息。特别地,协调器使用两个 FDT 的集合,分别称为参数树和权重树。前者包含了对于每个可能的低音线,由于作曲代理实现的元启发式的最

便捷的参数值描述。而后者包含的知识,是权值的形式,根据每个元启发式的预期性能排序。这些权值可被控制旋律交换的一组规则所使用,将在后面进行说明。知识提取过程被分成两个不同的阶段:数据准备和数据挖掘,将在后面进行解释。

注意,这个配置过程只在框架被用于作曲之前进行一次。之后, FDT 将独立于被和谐的低音线来配置框架。

1)数据准备:获得有效的 FDT 的第一步是创建一个数据库集,这些数据库包含了作曲代理实现的元启发式的性能的具有高度代表性的数据。这些数据库必须使用不同的数字低音问题的实例来准确描述元启发式的性能。要获得这些数据库,我们采用一种获得四音旋律的过程,其使用具有不同参数配置的不同元启发式。

特别地,我们选择了 30 个数字低音问题的代表性实例,并使用四个元启发式来求解。这些实例从巴赫合奏曲⁴中获得,代表了不可达的和谐例子。每个元启发式采用多个参数值的配置,根据参数个数、可能的值、行为的先验知识不同而有所不同。256, 81, 216, 72, 36 和 130 个不同的参数组合分别应用于 GA, PSO, ACO, SA, TS 和 VNS。另外,每条低音线被每个参数组合和谐十次。对于每个元启发式,我们记录一个原始数据库,这个数据库包含了获得的旋律的重要信息。特别地,每行包含以下信息:

- 1、 生成个低音线描述,如下:
 - a) 音乐节奏,或每分钟的节拍数,表明每个音乐片段的速度或步调。
 - b) 由时间拍号指明的节拍数。比如说 3/4 中的 3。
 - c) 由时间拍号指明的节拍长度。比如说 3/4 中的 4。
 - d) 作曲的关键拍号。比如 C 大调或 A 小调。
 - e) HN 的数量。

- f) 生成音乐片段的方法数。 3、 旋律的最终适应度值。
- 2、 用于和谐的参数值。

表 5

Instance	Par. comb.	it.	Attributes						Parameters				Final fitness
			Tempo	# beats	beat length	key sign.	# NHs	# measures	Sel. strategy	Mut. prob.	Cross prob.	# indiv.	
Bach's chorale BWV 26	1	1	240	4	4	A min	39	10	Tournament	0.01	0.5	20	92342
		10	240	4	4	A min	39	10	Tournament	0.01	0.5	20	91358
	256	1	240	4	4	A min	39	10	Roulette	0.5	0.9	40	87964
		10	240	4	4	A min	39	10	Roulette	0.5	0.9	40	90957
		1	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124831
		10	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124330
Bach's chorale BWV 70	1	1	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124831
		10	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124330
	256	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	132587
		10	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	133641
	1	1	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	76670
		10	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	78549
Bach's chorale BWV 153.9	1	1	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	69325
		10	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	71537
	256	1	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	76670
		10	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	78549
		1	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	69325
		10	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	71537

表 6 展示了一个 GA 相关的原始数据库例子。

然而,原始数据库还不能被 FDT 学习算法所使用。因为,它们存储了过多信息,并不能产生预期的输出(权值与参数)。要减小原始数据库的大小,将其处理成学习算法可用的,我们采用一个精炼过程,以产生以下精炼的数据库。

- 1) 进化权值数据库: 该数据库用于训练 FDT, 来产生进化相的权值。
- 2) 局部权值数据库: 该数据库用于训练 FDT, 来产生局部相的权值。
- 3) 参数数据库集: 每个不同元启发式的参数对应有一个数据库。比如说,如表 6 所示的, GA 需要 4 个参数数据库,每个数据库对应一个参数,其余的元启发式也需要类似的数据库。这些数据库被用于学习树,来为每个实例找到最合适的参数。

按以下步骤创建精炼数据库。

- 1) 首先,我们需要计算每个元启发式获得的每个参数组合与每个低音

线获得的平均适应度值。如表 6 所示的,对于巴赫合唱曲 BWV 26⁵,我们使用参数组合 1 (使用竞争选择,突变率为 0.01,交叉概率为 0.5,个体数为 20)来计算 GA 得到的平均适应度值。然后,我们使用参数组合 2, 3 等来计算相同的平均适应度。我们对巴赫合唱曲 70 和其他的,重复此过程。对 PSO, SA 和 TS 进行类似的处理。

- 2) 之后,对于每个低音线和元启发式,我们隔离最佳的参数组合,因为该组合已经获得了最高的适应度值。调用该组合,其中 a 表示实例, i 表示元启发式。与 相关的适应度值为 。
- 3) 为每个元启发式和其每个参数,创建一个参数数据库,每个低音线用一行一个记录表示。每一行,包含了该低音线的描述,其参数值,以 表示。

- 4) 为每个相（进化和局部搜索），生成一个数据库，每个实例一行。每一行，包含了实例的描述和权值，权值的计算公式如下：

其中， M 表示在该相中涉及的作曲代理集。

表 6

Instance	Par. comb.	it.	Attributes						Parameters				Final fitness
			Tempo	# beats	beat length	key sign.	# NHs	# measures	Sel. strategy	Mut. prob.	Cross prob.	# indiv.	
Bach's chorale BWV 70	...	1	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124831
	1	10	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124330

	...	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	132587
	256	10	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	133641

↓ ↓ Fitness Average Computation ↓ ↓													
Bach's chorale BWV 70	1	avg.	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124662.3

	256	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	134903.2
↓ ↓ New entries in the refined databases ↓ ↓													
Weights database								Sel. Strategy database					
Tempo	# beats	beat length	key sign.	# NHs	ω_{GA}	ω_{PSO}		Tempo	# beats	beat length	key sign.	# NHs	parameter value
60	3	4	G Maj	62	0.62	0.38		60	3	4	G Maj	62	Roulette

精炼过程如表 7 所示，其是一个两部精炼的例子。

2) 数据挖掘：一旦通过精炼数据库获得了性能信息，我们就能够在数据挖掘阶段提取知识模型。这些模型包括，两个 FDT（进化和局部阶段），与 FDT 相等数量的不同元启发式的参数。这些 FDT 中，每一个都是通过对前一步创建的数据库应用 FDT 学习算法直接获得的。选择 FDT，是因为它们之前在类似的问题中取得过成功 [37]-[39]，并且它们的可解释性也有利于对所获得的模型的理解。

B. 协作与协调控制

在习得了定义协调代理知识的 FDT 之后，就能够控制作曲代理的协作与协调了。然而，协作与协调控制并不重要，因为旋律的不受限制交换可能会导致不完全满足品质标准的旋律出现（如经常遇到的传统优化问题 [41]）。该问题可通过监测旋律的交换，

防止不期望的交换发生来避免。要做到这一点，协调代理使用一组模糊规则，应用 FDT 来控制交换。

特别地，FDT 分析带标准的低音线，并评估得到一个权值，该权值表示作曲代理 i 实现的元启发式的期望性能，且表示作曲代理 i 比 j 能得到更好的结果。

另外，每一步中（进化和局部），协调代理使用一组 TSK 模糊规则来决定何时有必要在两个或多个作曲代理间交换旋律。模糊规则，用权值表示，反映的结构如下：

```

if ( $\omega_1 \cdot d_1$ ) is enough
then poor_musical_quality1 = 1
...
if ( $\omega_{h-1} \cdot d_{h-1}$ ) is enough
then poor_musical_qualityh-1 = 1
if ( $\omega_{h+1} \cdot d_{h+1}$ ) is enough
then poor_musical_qualityh+1 = 1
...
if ( $\omega_n \cdot d_n$ ) is enough
then poor_musical_qualityn = 1

```

其中,

n 是参与给定步骤的作曲代理数;

是由规则评估的作曲代理;

$d_i = (\xi(c_i) - \xi(c_h)) / \max(\xi(c_i), \xi(c_h))$ 其中, ξ 是第三节中定义的适应函数;

是 c_i 的权值;

enough 具有由四连音定义的梯形隶属函数的模糊集 (a, b, c, d) , 且其论域的范围是 $[0, 1]$;

poor_musical_quality_i $i = 1, \dots, n$ 是 TSK 变量, 取值范围是 $[0, 1]$ 。更大的值表示 c_h 可能从与 c_i 的旋律交换中获益的可能性更大。

框架中包含的每个作曲代理都与一组模糊规则相关联, 又关联到同一步骤的其余作者代理。比如说, 一个进化阶段包含了 GA 和 PSO 的系统, 我们需要一个实现 GA 的作曲代理与实现 PSO 的代理相关的规则, 和另一个从 PSO 关联到 GA 的规则。这两个规则表示从其他作曲代理接收旋律的便捷性。

然而, 并非每个激活的规则都会触发旋律交换。只有那些 poor_musical_quality 高于给定阈值的规则才会交换旋律。当这种情况发生时, 旋律的交换将以下述形式进行:

- 1) 如果交换包含了使用进化技术的作曲代理, 那么旋律种群的自己将会被替换为(触发规则的元启发式)的种群中最优的个体。此处权值与相关。

- 2) 如果是局部搜索策略的作曲代理之间的交换, 那么当前旋律将被替换为“类似: 于由获得的最优旋律的旋律。此处, “类似”代表变异算子的一个应用。

五、实验分析

本节将研究提出的技术在解决数字低音问题上的有效性及其效果。我们已经进行了不同的测试以评估包含了新的元启发式, 是怎样影响技术的性能的。特别地, 实验以不采用自适应 MA 开始, 其仅包含了一个基于种群的算法和局部搜索算法。之后, 向其加入更多的元启发式, 基于种群的和基于局部搜索的都有, 直到获得一个包含了 6 个不同元启发式的协作策略为止。最后。将提出的策略与先前应用于求解数字低音问题的方法作比较

A. 实验配置

实验就以下 13 个策略进行了比较。

- 1、一组包含了一个基于种群的策略和一个局部搜索策略的非自适应 MA, 如下:
 - a) GA 和 SA (GA+SA)。
 - b) GA 和 TS (GA+TS)。
 - c) PSO 和 SA (PSO+SA)。
 - d) PSO 和 TS (PSO+TS)。
- 2、一组通过向先前的组合中添加新的元启发式的自适应 MA。这些策略被称为 MMC, 为了区分不同的组合, 我们用上标表示包含了基于种群的元启发式的策略, 用下标表示应用了局部搜索元启发式的策略, 如下:
 - a) GA, PSO 和 SA (MMC G&P S)。
 - b) GA, PSO 和 TS (MMC G&P T)。
 - c) GA, SA 和 TS (MMC G S&T)。
 - d) PSO, SA 和 TS (MMC P S&T)。

e) GA, PSO, SA 和 SA(MMC G&P S&T)。

f) GA, PSO, ACO, SA 和 TS (MMC G&P&A S&T)。

g) GA, PSO, SA, TS 和 VNS(MMC G&P S&T&V)。

h) GA, PSO, ACO, SA, TS 和 VNS (MMC G&P&A S&T&V)。

3、进化音乐作曲 (EMC)，是由 De Prisco 和 Zaccagnino^[45] 的进化算法。据我们所知，这是唯一的一篇论文，尝试去解决类似于本文定义的数字低音问题的问題。

每个策略都有 50000 个解评估，以解决实验中包含的每个实例。不使用基于时间的停止条件，我们使用解评估的数量来获得一个与其他技术做更公平的比较。使用这个方法，我们能够不依赖编程语言，或运行测试的计算机体系结构，或者程序员优化代码的能力，对结果进行比较

然而，时间被视为考量性能的一个重要因素，并且，应当注意到，协调器的开销是

可忽略的，协作策略的执行时间总是小于目标函数的相同数量的评估中较慢的单个元启发式的执行时间。

关于执行模式，如果时间很重要，协作策略可以采用并行的方式，或者在只有单处理器的计算机上模拟并行。以下是本文采取的策略，过程很简单。我们构造了一组解算器数组，并循环地运行它们。该实现方案采用了一个异步通信模式，过程如下：解算器在目标函数的一定数量的评估期间执行。评估的数量是 [5000, 5500] 间的一个随机数字，并且一个周期时间之后，更新信息。重复以上步骤，直至目标函数评价的终止条件。

实验使用的在一组实例是从巴赫合唱曲中选择的，还包括了：BWV 4.8, BWV 8.6, BWV 12.7, BWV 16.6, BWV 17.7, BWV 19.7, BWV 25.6, BWV 28.6, BWV 31.9, BWV 33.6, BWV 39.7, BWV 40.6, BWV 43.11, BWV 47.5, BWV 55.5, BWV 62.6, BWV 67.7, BWV 77.6, BWV 81.7, BWV 88.7。

表 7

Bach's chorale	GA+SA	GA+TS	PSO+SA	PSO+TS	$MMC_S^{G\&P}$	$MMC_T^{G\&P}$	$MMC_{S\&T}^G$
n. 408	0.77% _(0,01)	0.62% _(0,00)	1.80% _(0,02)	2.99% _(0,04)	0.31% _(0,00)	0.62% _(0,01)	0.45% _(0,00)
n. 806	1.22% _(0,01)	0.96% _(0,01)	1.69% _(0,02)	2.01% _(0,02)	1.22% _(0,01)	1.52% _(0,02)	0.86% _(0,01)
n. 1207	0.23% _(0,00)	2.18% _(0,03)	0.42% _(0,00)	3.55% _(0,04)	0.49% _(0,00)	2.08% _(0,03)	0.09% _(0,00)
n. 1606	0.32% _(0,00)	0.24% _(0,00)	0.32% _(0,00)	0.41% _(0,00)	0.18% _(0,00)	0.74% _(0,01)	0.15% _(0,00)
n. 1707	0.67% _(0,00)	1.12% _(0,01)	1.46% _(0,02)	3.92% _(0,05)	0.36% _(0,00)	1.13% _(0,01)	0.59% _(0,00)
n. 1907	0.25% _(0,00)	0.57% _(0,00)	0.25% _(0,00)	0.98% _(0,01)	0.13% _(0,00)	0.30% _(0,00)	0.09% _(0,00)
n. 2506	0.58% _(0,00)	1.20% _(0,02)	1.62% _(0,02)	3.90% _(0,10)	0.46% _(0,00)	1.86% _(0,03)	0.15% _(0,00)
n. 2806	0.46% _(0,00)	1.98% _(0,02)	0.80% _(0,01)	3.30% _(0,04)	0.16% _(0,00)	2.19% _(0,02)	0.38% _(0,00)
n. 3109	0.40% _(0,00)	2.13% _(0,04)	1.57% _(0,02)	4.32% _(0,15)	0.62% _(0,00)	2.28% _(0,03)	0.42% _(0,00)
n. 3306	0.77% _(0,00)	0.37% _(0,00)	0.42% _(0,00)	0.64% _(0,00)	0.34% _(0,00)	1.17% _(0,01)	0.40% _(0,00)
n. 3907	0.49% _(0,00)	0.66% _(0,01)	0.64% _(0,01)	1.28% _(0,02)	0.09% _(0,00)	0.60% _(0,00)	0.17% _(0,00)
n. 4006	0.47% _(0,00)	0.44% _(0,00)	0.38% _(0,00)	0.75% _(0,01)	0.37% _(0,00)	0.58% _(0,01)	0.45% _(0,00)
n. 4311	0.33% _(0,00)	0.31% _(0,00)	0.86% _(0,01)	0.39% _(0,00)	0.20% _(0,00)	0.37% _(0,00)	0.44% _(0,00)
n. 4705	0.46% _(0,01)	0.12% _(0,00)	0.11% _(0,00)	0.12% _(0,00)	0.12% _(0,00)	0.25% _(0,00)	0.11% _(0,00)
n. 5505	0.24% _(0,00)	0.43% _(0,00)	0.39% _(0,00)	0.86% _(0,01)	0.19% _(0,00)	0.31% _(0,00)	0.23% _(0,00)
n. 6206	0.21% _(0,00)	0.06% _(0,00)	0.37% _(0,00)	0.06% _(0,00)	0.05% _(0,00)	0.38% _(0,00)	0.24% _(0,00)
n. 6707	0.67% _(0,00)	0.42% _(0,00)	0.19% _(0,00)	0.26% _(0,00)	0.25% _(0,00)	0.59% _(0,01)	0.16% _(0,00)
n. 7706	0.71% _(0,01)	1.26% _(0,02)	0.75% _(0,01)	2.04% _(0,03)	0.22% _(0,00)	1.09% _(0,01)	0.40% _(0,00)
n. 8107	0.68% _(0,01)	0.24% _(0,00)	0.80% _(0,01)	1.56% _(0,02)	0.21% _(0,00)	0.76% _(0,00)	0.60% _(0,00)
n. 8807	0.29% _(0,00)	1.85% _(0,02)	0.23% _(0,00)	2.58% _(0,03)	0.23% _(0,00)	1.35% _(0,01)	0.40% _(0,00)
Average	0.51% _(0,00)	0.86% _(0,01)	0.75% _(0,01)	1.79% _(0,03)	0.31% _(0,00)	1.01% _(0,01)	0.34% _(0,00)

Bach's chorale	$MMC_{S\&T}^P$	$MMC_{S\&T}^{G\&P}$	$MMC_{S\&T}^{G\&P\&A}$	$MMC_{S\&T\&V}^{G\&P}$	$MMC_{S\&T\&V}^{G\&P\&A}$	EMC
n. 408	2,05% _(0,03)	0,36% _(0,00)	0,02% _(0,00)	0,51% _(0,00)	0,01% _(0,00)	20,56% _(1,66)
n. 806	2,41% _(0,00)	0,66% _(0,00)	0,03% _(0,00)	0,69% _(0,00)	0,02% _(0,00)	5,12% _(0,20)
n. 1207	0,13% _(0,00)	0,38% _(0,00)	0,03% _(0,00)	0,17% _(0,00)	0,02% _(0,00)	0,24% _(0,00)
n. 1606	0,12% _(0,00)	0,08% _(0,00)	0,03% _(0,00)	0,27% _(0,00)	0,02% _(0,00)	1,75% _(0,03)
n. 1707	1,30% _(0,01)	0,36% _(0,00)	0,03% _(0,00)	0,88% _(0,00)	0,02% _(0,00)	49,01% _(4,92)
n. 1907	0,18% _(0,00)	0,25% _(0,00)	0,02% _(0,00)	0,20% _(0,00)	0,01% _(0,00)	16,55% _(1,09)
n. 2506	0,79% _(0,01)	0,41% _(0,00)	0,06% _(0,00)	0,24% _(0,00)	0,04% _(0,00)	10,36% _(0,78)
n. 2806	0,33% _(0,00)	0,25% _(0,00)	0,02% _(0,00)	0,38% _(0,00)	0,01% _(0,00)	2,13% _(0,04)
n. 3109	1,54% _(0,02)	0,33% _(0,00)	0,02% _(0,00)	0,87% _(0,01)	0,02% _(0,00)	15,57% _(0,84)
n. 3306	0,30% _(0,00)	0,47% _(0,00)	0,03% _(0,00)	0,43% _(0,00)	0,02% _(0,00)	29,74% _(2,31)
n. 3907	0,43% _(0,00)	0,23% _(0,00)	0,02% _(0,00)	0,08% _(0,00)	0,01% _(0,00)	0,33% _(0,00)
n. 4006	1,10% _(0,02)	0,12% _(0,00)	0,02% _(0,00)	0,12% _(0,00)	0,02% _(0,00)	0,23% _(0,00)
n. 4311	0,26% _(0,00)	0,22% _(0,00)	0,02% _(0,00)	0,34% _(0,00)	0,01% _(0,00)	33,24% _(2,68)
n. 4705	0,23% _(0,00)	0,10% _(0,00)	0,03% _(0,00)	0,10% _(0,00)	0,02% _(0,00)	0,17% _(0,00)
n. 5505	0,34% _(0,00)	0,31% _(0,00)	0,02% _(0,00)	0,24% _(0,00)	0,02% _(0,00)	0,87% _(0,01)
n. 6206	0,06% _(0,00)	0,40% _(0,00)	0,01% _(0,00)	0,05% _(0,00)	0,01% _(0,00)	0,14% _(0,00)
n. 6707	0,09% _(0,00)	0,17% _(0,00)	0,02% _(0,00)	0,11% _(0,00)	0,01% _(0,00)	0,11% _(0,00)
n. 7706	1,25% _(0,01)	0,40% _(0,00)	0,03% _(0,00)	0,49% _(0,00)	0,02% _(0,00)	5,68% _(0,44)
n. 8107	0,19% _(0,00)	0,20% _(0,00)	0,03% _(0,00)	0,49% _(0,00)	0,02% _(0,00)	6,07% _(0,43)
n. 8807	0,17% _(0,00)	0,22% _(0,00)	0,02% _(0,00)	0,48% _(0,00)	0,02% _(0,00)	0,97% _(0,02)
Average	0,66% _(0,01)	0,30% _(0,00)	0,03% _(0,00)	0,36% _(0,00)	0,02% _(0,00)	9,94% _(0,77)

每个单独的实例都被各策略求解了十次。结果用平均值和标准差表示（下标）。

B. 结果

1) 非自适应与自适应 MA 对比: 表 8 显示了由不同策略获得的结果。这些结果以针对每个实例获得的最高适应度值的平均误差百分比来表示。为了演示由所提出的技术实现的适应性机制的有效性,我们将首先通过比较自适应与非自适应 MA 来分析结果。为了进行更严格的比较,我们使用由 García et al.^[43] 提出的方法,该方法采用非参数统计测试来研究通过不同方法获得的结果间的差异。

一开始,通过使用 Friedman 检验,我们评估每个元启发式获得的结果间是否存在任何显著差异。特别地, Friedman 检验表明差异是统计学上的显著差异,为 0.05 水平。然后,通过使用 Wilcoxon 的无符号秩检验将每个策略的性能与其他策略的性能进行成对比较。对于零假设,假定等效结果分布存在。另一方面,所考虑的替代假设的真实位置偏移不等于 0。当超过两项技术进行比较,需要调整获得的 p 值。我们使用 Benjamini-Hochberg 方法对它们进行调整。表 9 对获得的结果做了总结。当两个策略的性能差异明显(p 值小于 0.05)时,用“+”表征,若结果在统计学上无意义,则使用“-”。

表 8

	GA+SA	GA+TS	PSO+SA	PSO+TS	$MMC_S^{G\&P}$	$MMC_T^{G\&P}$	$MMC_{S\&T}^{G\&P}$	$MMC_{S\&T}^P$	$MMC_{S\&T}^{G\&P}$	$MMC_{S\&T\&V}^{G\&P\&A}$	$MMC_{S\&T\&V}^{G\&P}$	EMC
GA+SA	=	-	-	+	+	+	+	-	+	+	+	+
GA+TS	-	=	-	+	+	-	+	-	+	+	+	+
PSO+SA	-	-	=	+	+	-	+	-	+	+	+	+
PSO+TS	+	+	+	=	+	+	+	+	+	+	+	+
$MMC_S^{G\&P}$	+	+	+	+	=	+	-	+	-	+	-	+
$MMC_T^{G\&P}$	+	-	-	+	+	=	+	-	+	+	+	+
$MMC_{S\&T}^{G\&P}$	+	+	+	+	-	+	=	-	-	+	-	+
$MMC_{S\&T}^P$	-	-	-	+	+	-	-	=	-	+	-	+
$MMC_{S\&T}^{G\&P\&A}$	+	+	+	+	-	+	-	-	=	+	-	+
$MMC_{S\&T}^{G\&P}$	+	+	+	+	+	+	+	+	+	=	+	+
$MMC_{S\&T\&V}^{G\&P\&A}$	+	+	+	+	-	+	-	-	+	+	=	+
$MMC_{S\&T\&V}^{G\&P}$	+	+	+	+	+	+	+	+	+	+	+	=
EMC	+	+	+	+	+	+	+	+	+	+	+	=



图 3

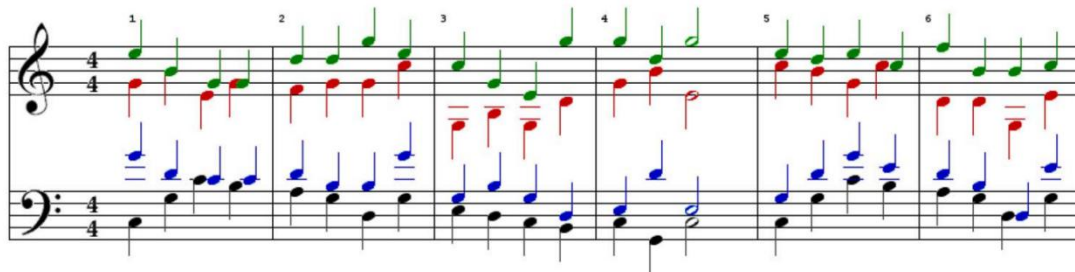


图 4

从获得的结果中可以得到第一个结论是在提出的系统中,引入的适应与协作的效用性。特别地,所有自适应模因策略,除了两个(和),结果都显著优于非自适应模因策略的结果。另外,两个不能提升非自适应策略的自适应策略的结果在统计学上等效,除了 GA+SA 在一个例子中优于。

结果同时也显示了提出的技术的可扩展性。在大多数例子中,(16 分之 10),当新的元启发式加入之前的组合时,获得的结果显著地优于之前的。在剩余的 6 个例子中,结果在统计学上并没有显著的不同,但都没有更坏了。并且,可获得更好的结果的策略包含了所有 6 个单独的元启发式。特别地,该策略的结果显著地优于其他所有测试的模因策略,不管是自适应或非自适应的

2) 音乐作曲技术对比:现在,我们将 MA 的性能与 EMC 进行比较,结果如表 8 所示。再次声明,为了进行更公平的比较,我们采用与前一节相同的方法进行分析。表 9

的最后一列表示统计检验的结果。可以看出,所有 MA 从统计的角度来看,都优于 EMC。这些结果显示了 MA 在解决复杂问题上的能力,比如数字低音技术。

C. 解的分析

分析与所获得的适合度值相关的结果能够让我们找出用于解决问题的最佳策略。然而,这不会带来对所找到的解决方案的音乐特点的洞察。为了弥补这个缺陷,我们将在这个子小节中研究巴赫合唱曲 BWV 25.6 的解。由于解空间的限制,我们将本文限制于第一个方法。

图 3 表示合唱曲的五线谱(作为输入)。如果我们使用简单的随机搜索来解决该实例,我们可能可以获得如图 4 所示的解。该解决方案是在实例解析的初始步骤期间获得的,其结果乍一看,很混乱。事实上,它有许多第三节中定义的和声错误,因此结果听起来会不和谐不令人愉悦。

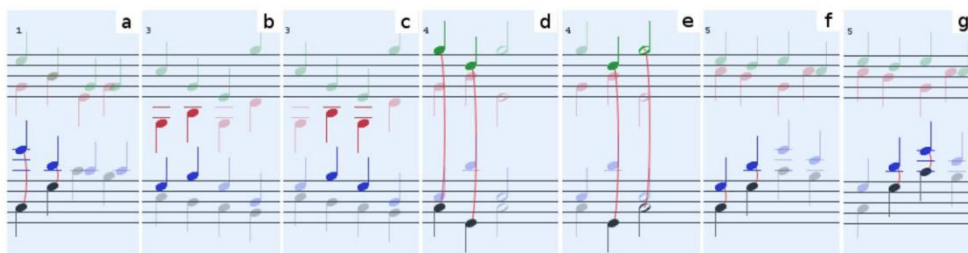


图 5

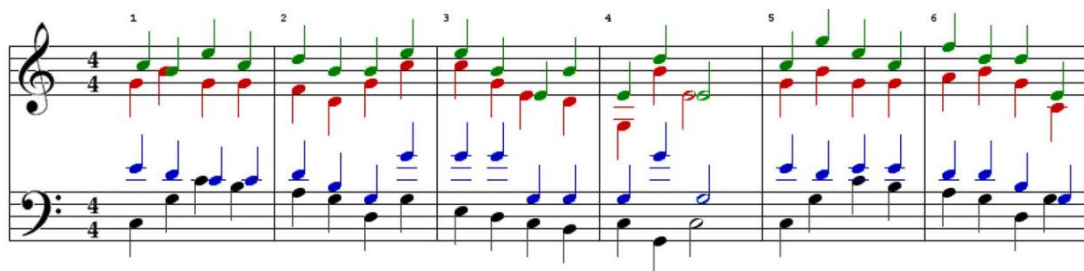


图 6

图 5 显示了可以在随机搜索解中隔离避免的关键错误。在随机和谐片段的前六个测量中总共出现了七个关键错误。同时，还有 12 个小问题（第三节中标为普通问题的）未显示。错误的音符用更深的颜色标出。这些错误如下：

- 1) 连续五度：该错误表示两个声音的变化，一个完美的五度之后跟着一个不同的完美五度。如图中的 *a, d, e, f* 和 *g* 部分。
- 2) 同度：该错误可在图中 *b* 和 *c* 部分看到，当两个音朝同一个方向一致变化时出现。

选择不同的和弦，或交换每个音演奏的音符，可能可以优化解，并消除错误。然而，这么做可能会引入新的错误。图 6 显示了巴赫合唱曲 25.6 的 EMC 最优解（就适应函数而言）。其看起来比随机生成的解组织得更好，听起来也更好。随机解中出现的所有关键错误都被消除了，但又产生了新的错误，如图 7 所示。EMC 解中有两个关键错误，八个正常问题。这两个错误包括：

- 1) 平行八度：该错误见图 7 的 *a* 和 *b* 部分，当两个音在相同方向上从较小间隔跳跃到完美八度音程的间隔。

如果使用恰当的方法，是可以消除所有的关键性错误的，和大多数非关键性错误。

图 8 显示了 的最优解（就适应函数而言）。该解没有之前定义的任何错误。因此，它听起来比本节中所有其余解都动听。

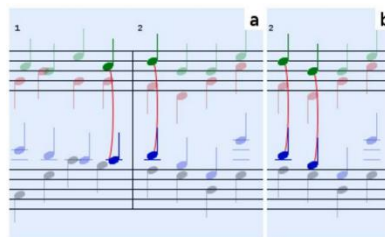


图 7

D. 音乐品质

正如介绍中所指出的，音乐作曲的品质是一个主观的判断。然而，所提供的解没有任何关键性错误。

我们进行了鉴赏测试，由具有音乐背景的人聆听我们所提出的模因作曲生成的音乐。鉴赏测试的反馈表明，这些音乐作品是健全的，并遵守一般作曲规则。人类作曲家可能在创造性上表现得更好，但大多数人都惊叹于计算机居然能创造这样的曲子。

我们提出的 MMC 取得的结果基本满足预期。事实上，它们代表具有显著质量和和谐的解决方案。这样的结果是可行的，因为我们提出的系统的协作功能能够消除所有关键性错误和许多和谐规则允许的错误的。

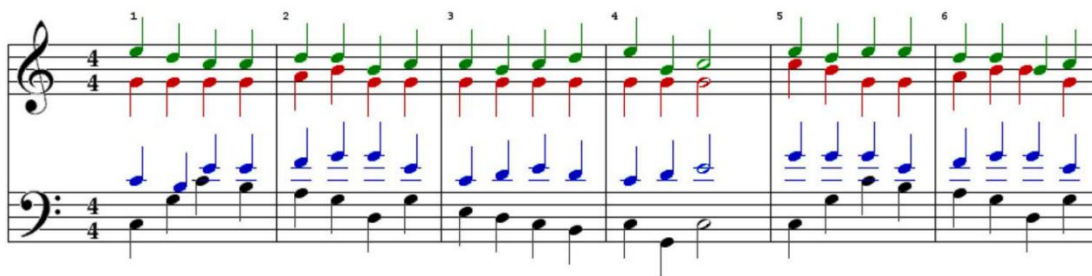


图 8

六、结论

自动音乐作曲是计算机艺术的最重要领域之一,最近的学科研究旨在通过人工智能技术复制人类创造性的艺术设计。不同方法被应用于解决该挑战,其中,因为进化算法评估和演化给定音乐作品片段的能力,遗传音乐在这方面扮演着越来越重要的角色。然而,基于遗传技术的自动音乐作曲离完善还很遥远,因为以下两个关键因素:1)人为的再现人类作曲家的天才与能力的巨大挑战;2)对于自动生成的旋律的美学评估做一个正式的准则很困难。

本文展现了进化音乐领域突破性的技术,由于对音乐的多元进化过程的模仿,组合不同模因作曲代理,能有效地解决上述问题。事实上,所提出的问题正式将非数字五线谱作曲技术建模为基于一个创新的音乐适应函数的优化问题,该应用函数使用著名的和谐规则来使每个作曲代理以准确和正式的方式评估其作曲的技巧。该适应函数是用于实现提出的自动音乐作曲系统的自适应模因方法的关键,使得该方法能够比当前的进化方法具有更好的性能。事实上,在实验部分我们见到过了,自适应模因方法从统计的角度看,克服了所有其他在音乐作曲领域使用的混合或非混合的进化算法。而且,我们提出的自适应 MA 已经被证明在观测与监控方面比传统的进化音乐方法更好。进化音乐由一组具有音乐背景的人进行过检

验,通过将由模因作曲代理生成的音乐片段与其他技术生成的音乐片段进行比价。

未来,我们将扩展我们的方法,以处理不同的音乐作曲技术和美学进化算法,成为进化音乐领域最完善与高质量的方法。

致谢

E. Muñoz 对自适应模因算法的整体设计做出了贡献。G. Acampora 是本研究的主要监督者。其他作者在撰写与修改论文方面做出了重大贡献。

参考文献

- [1] E. R. Miranda, "Composing music with computers," Music Technology Series. Oxford, U.K.: Focal Press, 2001.
- [2] L. Fogel, "Evolving art," in Proc. 1st Annu. Conf. Evol. Program., San Diego, CA, USA, 1992, pp. 183–189.
- [3] M. Dostál, "Evolutionary music composition," in Handbook of Optimization (Intelligent Systems Reference Library), vol. 38, I. Zelinka, V. Sněšil, and A. Abraham, Eds. Berlin, Germany: Springer, 2013, pp. 935–964.
- [4] X. S. Chen and Y. S. Ong, "A conceptual modeling of meme complexes in stochastic search," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 42, no. 3, pp. 612–625, Sep. 2012.
- [5] D. W. Johnson and R. T. Johnson, Cooperation and Competition: Theory and Research. Edina, MN, USA: Interaction Book Company, 1989.
- [6] P. Moscato, "On evolution, search, optimization, GAs and martial arts: Toward memetic algorithms," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.
- [7] N. Krasnogor and J. E. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy and design issues," IEEE Trans. Evol. Comput., vol. 9, no. 5, pp. 474–488, Oct. 2005.
- [8] A. Torn and A. Zilinskas, Global Optimization (Lecture Notes in Computer Science), vol. 350. Berlin, Germany: Springer, 1989.
- [9] Y. S. Ong and A. J. Keane, "Meta-Lamarckian in memetic algorithm," IEEE Trans. Evol. Comput., vol. 8, no. 2, pp. 99–110, Apr. 2004.
- [10] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 36, no. 1, pp. 141–152, Feb. 2006.
- [11] L. Hiller, "Computer music," Sci. Amer., vol. 201, no. 6, pp. 109–120, 1959.
- [12] L. Hiller and L. M. Isaacson, Experimental Music: Composition With an Electronic Computer. New York, NY, USA: McGraw-Hill, 1959.
- [13] J. A. Biles, "Evolutionary computation for musical tasks," in Evolutionary Computer Music, E. R. Miranda and J. A. Biles, Eds. London, U.K.: Springer, 2007.
- [14] A. R. Brown, "Opportunities for evolutionary music composition," in Proc. Aust. Comput. Music Conf., Melbourne, VIC, Australia, 2002, pp. 27–34.
- [15] R. Waschka, II, "Avoiding the fitness 'bottleneck': Using genetic algorithms to compose orchestral music," in Proc. Int. Computer Music Conf., San Francisco, CA, USA, 1999, pp. 201–203.
- [16] R. Waschka, II, "Composing with genetic algorithms: GenDash," Evolutionary Computer Music, E. R. Miranda and J. A. Biles, Eds. London, U.K.: Springer, 2007, pp. 117–136.
- [17] B. L. Jacob, "Composing with genetic algorithms," in Proc. Int. Comput. Music Conf., Banff, AB, Canada, 1995, pp. 452–455.

- [18] B. L. Jacob, "Algorithmic composition as a model of creativity," *Organ. Sound*, vol. 1, no. 3, pp. 157–165, 1996.
- [19] J. H. Moore. (1994). *GAMusic: Genetic Algorithm to Evolve Musical Melodies*. [Online]. Available: <http://www.cs.cmu.edu/afs/cs/project/airepository/ai/areas/genetic/ga/systems/gamusic/0.html>
- [20] B. Johanson and R. Poli, "GP-music: An interactive genetic programming system for music generation with automated fitness raters," in *Proc. 3rd Int. Conf. Genet. Program.*, Helsinki, Finland, 1998, pp. 181–186.
- [21] L. Spector and A. Alpern, "Criticism, culture, and the automatic generation of artworks," in *Proc. 12th Nat. Conf. Artif. Intell.*, vol. 1, Menlo Park, CA, USA, 1994, pp. 3–8.
- [22] L. Spector and A. Alpern, "Induction and recapitulation of deep musical structure," in *Proc. IFCAI 1995 Workshop Artif. Intell. Music*, Montreal, QC, Canada, pp. 41–48.
- [23] J. A. Biles, *GenJam: A Genetic Algorithm for Generating Jazz Solos*. Ann Arbor, MI, USA: Univ. Michigan Library, 1994.
- [24] J. A. Biles, "Interactive GenJam: Integrating real-time performance with a genetic algorithm," in *Proc. 1998 Int. Comput. Music Conf.*, San Francisco, CA, USA.
- [25] J. A. Biles, "Life with GenJam: Interacting with a musical IGA," in *Proc. 1999 IEEE Int. Conf. Syst. Man Cybern.*, Tokyo, Japan, pp. 652–656.
- [26] J. A. Biles, "GenJam: Evolution of a jazz improviser," in *Creative Evolutionary Systems*, P. Bentley and D. Corne, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 165–187, 2002.
- [27] K. Thywissen, "GeNotator: An environment for investigating the application of genetic algorithms in computer assisted composition," in *Proc. 1996 Int. Comput. Music Conf.*, San Francisco, CA, USA, pp. 274–277.
- [28] P. M. Gibson and J. A. Byrne, "NEUROGEN: Musical composition using genetic algorithms and cooperating neural networks," in *Proc. 2nd 1994 Int. Conf. Artif. Neural Netw.*, Bournemouth, U.K., 1991, pp. 309–313.
- [29] T. Unemi, "A design of genetic encoding for breeding short musical pieces," in *Proc. Workshop Artif. Life Models Music. Appl. II Search Music. Creativity*, Sydney, NSW, Australia, 2002, pp. 25–29.
- [30] T. Unemi, "SBEAT3: A tool for multi-part music composition by simulated breeding," in *Proc. 8th Int. Conf. Artif. Life*, Cambridge, MA, USA, 2003, pp. 410–413.
- [31] D. Ralley, "Genetic algorithms as a tool for melodic development," in *Proc. Int. Comput. Music Conf.*, Banff, AB, Canada, 1995, pp. 501–502.
- [32] D. Cope, *Experiments in Musical Intelligence (Computer Music and Digital Audio Series 12)*. Madison, WI, USA: A-R Editions, 1996.
- [33] D. Cope, *The Algorithmic Composer (Computer Music and Digital Audio Series 16)*. Madison, WI, USA: A-R Editions, 2000.
- [34] D. Cope, *Virtual Music: Computer Synthesis of Musical Style*. Cambridge,

- MA, USA: MIT Press, 2004.
- [35] G. Loy, *Musimathics: The Mathematical Foundations of Music*, vol. 1. Cambridge, MA, USA: MIT Press, 2006.
- [36] W. Piston and M. DeVoto, *Harmony*. New York, NY, USA: Norton, 1987.
- [37] G. Acampora, J. M. Cadenas, V. Loia, and E. M. Ballester, "Achieving memetic adaptability by means of agent-based machine learning," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 557–569, Nov. 2011.
- [38] G. Acampora, J. M. Cadenas, V. Loia, and E. M. Ballester, "A multi-agent memetic system for human-based knowledge selection," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 5, pp. 946–960, Sep. 2011.
- [39] J. M. Cadenas, M. C. Garrido, and E. M. Ballester, "Using machine learning in a cooperative hybrid parallel strategy of metaheuristics," *Inf. Sci.*, vol. 179, no. 19, pp. 3255–3267, 2009.
- [40] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, "A fuzzy random forest," *Int. J. Approx. Reason.*, vol. 51, no. 7, pp. 729–747, 2010.
- [41] T. G. Crainic, M. Gendreau, P. Hansen, and N. Mladenovic, "Cooperative parallel variable neighborhood search for the p-median," *J. Heuristics*, vol. 10, no. 3, pp. 293–314, 2004.
- [42] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds. Boston, MA, USA: Kluwer Academic, 2003.
- [43] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study statistical of techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput.*, vol. 13, no. 10, pp. 959–977, 2009.
- [44] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007, pp. 553–560.
- [45] R. De Prisco and R. Zaccagnino, "An evolutionary music composer algorithm for bass harmonization," in *Applications of Evolutionary Computing (Lecture Notes in Computer Science)*, vol. 5484. Berlin, Germany: Springer, 2009, pp. 567–572.
- [46] S. Kostka and D. Payne, *Tonal Harmony With an Introduction to Twentieth Century Music*. New York, NY, USA: McGraw-Hill, 2008.