

面向电子木琴自动演奏的MIDI信息智能转换算法研究

周涵艺^{1,2}, 刘卫玲^{2,3*}, 常晓明²

¹太原理工大学计算机科学与技术学院, 山西 晋中

²太原理工大学晓明研究室, 山西 太原

³太原理工大学人工智能学院, 山西 晋中

收稿日期: 2026年4月7日; 录用日期: 2026年5月1日; 发布日期: 2026年5月27日

摘要

针对电子木琴自动演奏系统中MIDI信息转换的关键问题, 文章提出了两种核心算法: 智能音高偏移优化算法和智能转换策略选择算法。智能音高偏移优化算法通过构建六维度加权评分体系, 自动计算最优音高偏移量, 将任意MIDI信息所包含的音域适配至木琴有效音域; 智能转换策略选择算法基于轨道数、总音符数和主轨占比三个特征, 通过规则评分自动选择全曲转换、主旋律轨转换或截取部分转换策略, 以适应不同音乐类型和系统资源限制。在Python环境下开发了MIDI转换程序, 实现了从MIDI信息到C语言头文件的完整转换流程。实验结果表明, 两种算法能有效解决音域不匹配和策略选择问题, 转换后的音高分布与原始旋律轮廓高度一致, 验证了算法的有效性。研究为传统打击乐器的智能化改造提供了技术支撑。

关键词

电子木琴, MIDI转换, 智能音高偏移优化算法, 智能转换策略选择算法, 自动演奏

Research on Intelligent Conversion Algorithm of MIDI Information for Automatic Performance of Electronic Xylophone

Hanyi Zhou^{1,2}, Weiling Liu^{2,3*}, Xiaoming Chang²

¹College of Computer Science and Technology, Taiyuan University of Technology, Jinzhong Shanxi

²Xiaoming Research Laboratory, Taiyuan University of Technology, Taiyuan Shanxi

*通讯作者。

文章引用: 周涵艺, 刘卫玲, 常晓明. 面向电子木琴自动演奏的 MIDI 信息智能转换算法研究[J]. 嵌入式技术与智能系统, 2025, 2(6): 363-374. DOI: 10.12677/etis.2025.26036

Abstract

Regarding the key issue of MIDI information conversion in the automatic electronic xylophone playing system, this paper proposes two core algorithms: the intelligent pitch offset optimization algorithm and the intelligent conversion strategy selection algorithm. The intelligent pitch offset optimization algorithm constructs a six-dimensional weighted scoring system to automatically calculate the optimal pitch offset amount, adapting the pitch range contained in any MIDI information to the effective pitch range of the xylophone; the intelligent conversion strategy selection algorithm, based on three features - the number of tracks, the total number of notes, and the proportion of the main track - automatically selects the full piece conversion, the main melody track conversion, or the partial extraction conversion strategy to adapt to different music types and system resource limitations. The MIDI conversion program was developed in the Python environment, achieving a complete conversion process from MIDI information to C language header files. Experimental results show that the two algorithms can effectively solve the problems of pitch range mismatch and strategy selection, and the pitch distribution after conversion is highly consistent with the original melody contour, verifying the effectiveness of the algorithms. This research provides technical support for the intelligent transformation of traditional percussion instruments.

Keywords

Electronic Xylophone, MIDI Conversion, Intelligent Pitch Offset Optimization Algorithm, Intelligent Conversion Strategy Selection Algorithm, Automatic Performance

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

传统键盘式电子琴系统需通过用户物理敲击键盘完成音乐演奏，对用户的乐理知识和演奏技巧有一定要求。随着数字音乐技术的发展，如何实现乐器的自动化演奏、降低演奏门槛成为研究热点之一。MIDI作为一种数字音乐接口标准，能够将乐谱信息转化为数字指令，为乐器自动化提供了技术基础。

在电子木琴自动演奏系统中，核心问题之一是如何将 MIDI 信息转换为硬件可执行的指令。MIDI 信息记录了音符的音高、起始时间和持续时间，是“计算机能理解的乐谱”，但无法直接被微控制器(MCU)识别。由于 MCU 通常使用 C 语言进行编程，因此需要将 MIDI 信息转换为 C 语言头文件格式。

目前，MIDI 信息转换研究主要集中于音高识别[1]和主旋律提取[2]，面向有限音域乐器的自适应转换算法较少。现有方法多采用固定升降调或基于调性检测的整体平移，无法解决超出木琴音域(仅 2 个八度)的音符映射问题；同时，针对嵌入式系统资源限制的转换策略选择也缺乏系统研究。相比 MP3 等音频格式，MIDI 信息具有结构化的音符数据，便于算法处理，更适合实现智能化转换。

本文针对上述问题，提出两种核心算法：智能音高偏移优化算法和智能转换策略选择算法，前者通过六维度加权评分模型自动计算最优音高偏移量，将 MIDI 信息中的音域适配至木琴有效音域；后者基

于轨道数、总音符数和主轨占比，自动选择全曲转换、主旋律轨转换或截取部分转换策略。在 Python 环境下开发的转换程序，实现了从 MIDI 信息到 C 语言头文件的完整转换流程。本文中所提到的“MIDI 信息”均指标准 MIDI 文件所包含的结构化音乐数据。

本文后续章节安排如下：第二章介绍两种算法的理论模型；第三章阐述 MIDI 转换程序的实现；第四章对算法进行验证与评估；第五章总结全文。

2. MIDI 信息智能转换算法的数学模型

从 MIDI 信息到 C 头文件转换的总体思路，如图 1 所示。

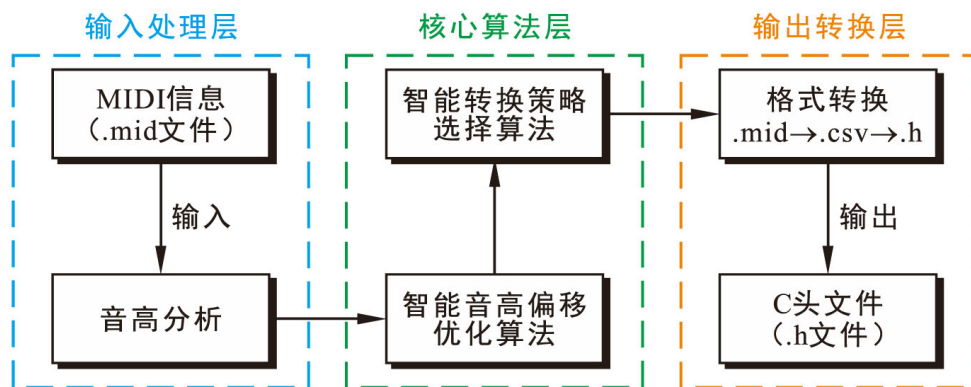


Figure 1. Overall conceptual diagram for MIDI information conversion
图 1. MIDI 信息转换总体思路图

图 1 中，输入的 MIDI 信息首先经过音高分析，统计各音符的音高及出现频率；然后由智能音高偏移优化算法将音域适配至木琴有效范围；再经智能转换策略选择算法根据音乐特征决定最优转换策略；最后通过格式转换生成控制器可执行的 C 语言头文件[3]。

2.1. 智能音高偏移优化算法的数学模型

智能音高偏移优化算法是一种多目标加权评分优化算法，目标是自动找到最优音高偏移量 d^* ，使 MIDI 信息中的音域最大限度适配电子木琴音域(MIDI 编号 43~67) [4]。

该算法的核心目标为最大化综合评分函数，找到最优音高偏移量 d^* ，计算公式如下：

$$d^* = \arg \max_{d \in [-48, 48]} S(d) \quad (1)$$

式中， d^* ：最优音高偏移量(半音数)，

d ：候选音高偏移量(半音数)，搜索空间为 $[-48, 48]$ 的整数，

$S(d)$ ：音高偏移量 d 的综合评分函数。

综合评分函数 $S(d)$ 采用六个维度的加权评分体系，每个维度从不同角度评估偏移量的优劣，计算公式为：

$$S(d) = (w_1 \cdot R(d) + w_2 \cdot E(d) + w_3 \cdot C(d) + w_4 \cdot G(d) + w_5 \cdot M(d) + w_6 \cdot P(d)) \times B(d) \quad (2)$$

式中， w_i ：各维度权重比例，

$R(d)$ ：音域内比例，权重 w_1 为 0.25，

$E(d)$ ：音高分布熵，权重 w_2 为 0.20，

$C(d)$: 中心音高贴近度, 权重 w_3 为 0.15,

$G(d)$: 音域利用率, 权重 w_4 为 0.15,

$M(d)$: 映射惩罚因子, 权重 w_5 为 0.10,

$P(d)$: 位置适应度, 权重 w_6 为 0.15,

$B(d)$: 八度移位奖励系数。

其中, $B(d)$ 为候选音高偏移量 d 的八度移位奖励系数, 当且仅当 d 为 12 的整数倍(即整八度移位)时, 予以奖励, 计算公式如下:

$$B(d) = \begin{cases} 1.05 & \text{if } d \bmod 12 = 0 \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

式(2)中, 六个维度函数通过对应公式归一化到 $[0, 1]$ 区间, 保证不同量纲的评分可直接加权求和。以音域内比例函数 $R(d)$ 为例, 说明公式形式, 其归一化计算公式如下:

$$R(d) = \frac{\sum_{p \in P} \text{freq}(p) \cdot \Pi(p+d \in V)}{\sum_{p \in P} \text{freq}(p)} \quad (4)$$

式中, $\Pi(\cdot)$: 指示函数, 条件成立时值为 1, 否则为 0,

P : 原始 MIDI 信息中所有不同音高的集合,

p : 集合 P 中的某个音高值(MIDI 编号),

$\text{freq}(p)$: 原始音高 p 在 MIDI 信息中出现的次数(即频率),

d : 候选音高偏移量(半音数), 搜索空间为 $[-48, 48]$ 的整数,

V : 木琴有效音域区间, 范围为 $[43, 67]$, 即 24 个半音(木琴音域理论最大跨度)。

六个维度函数用于评估候选音高偏移量 d 的适应度, 每个函数的值域均归一化到 $[0, 1]$ 区间, 值越接近 1 表示该维度表现越优。六个维度函数意义, 如表 1 所示。

Table 1. Function significance of six dimensions

表 1. 六个维度函数意义

NO.	六个维度函数	函数意义
01	音域内比例函数 $R(d)$	值为 0 表示没有音符直接落在有效音域内; 值为 1 表示所有音符均直接有效
02	音高分布熵函数 $E(d)$	值为 0 表示所有音符集中在单一音高; 值越接近 1 表示 15 个有效音高越均匀分布
03	中心音高贴近度函数 $C(d)$	值为 0 表示所有音符均在音域两端(最差); 值为 1 表示所有音符均为中心音高 55
04	音域利用率函数 $G(d)$	值为 0 表示只使用一个音高(最集中); 值为 1 表示使用全部 24 个半音(最广泛)
05	映射惩罚因子函数 $M(d)$	值为 0 表示所有音符都需要映射(最差); 值为 1 表示没有音符需要映射(最佳)
06	位置适应度函数 $P(d)$	值为 0 表示整体平均音高严重偏离音域; 值为 1 表示整体平均音高在音域 $[43, 67]$ 内

2.2. 智能转换策略选择算法的数学模型

智能转换策略选择算法是一种基于规则的三元决策算法，目标是通过分析 MIDI 信息的三个关键特征(轨道数、主轨占比、总音符数)，按照预设的评分规则，为三种转换策略计算综合得分，最终选择得分最高的策略，以在转换质量与资源占用之间找到最佳平衡[5]。

MIDI 信息具有多样性：部分为单轨纯旋律，部分为多轨复合音乐(含伴奏、打击乐)；音符数量也从几十到数千不等。而嵌入式系统存在内存限制，通常要求输入音符数小于 1,000 个。因此，需要据 MIDI 信息的特征自动选择最优转换策略。定义 MIDI 信息 M 的三个关键特征如下：

- (1) 有效轨道数量 T ：包含有效音符(velocity > 0 的 note_on 消息)事件的轨道数量；
- (2) 总音符数 N ：所有有效轨道中有效音符事件的总数；
- (3) 主轨占比 R ：有效音符数量最多的轨道占总音符数的比例；

对上述三个输入特征，算法预设三种转换策略处理输入的 MIDI 信息，如表 2 所示。

Table 2. Three conversion strategies

表 2. 三种转换策略

NO.	策略类型	转换逻辑	适用场景
01	全曲转换 S_1	保留 MIDI 信息所有有效轨道的有效音符	单轨简单音乐
02	主旋律轨转换 S_2	仅保留音符数量最多的主旋律轨	多轨复杂音乐
03	截取部分转换 S_3	先筛选主旋律轨(无主轨则选全轨道)，再按预设最大音符数截取前 K 个有效音符	超长或复杂音乐

表 2 中，截取部分转换策略 S_3 ，需动态计算最大保留音符数 K ，计算公式如下：

$$K = \begin{cases} \min\left(500, \left\lfloor \frac{N}{2} \right\rfloor\right) & N > 1000 \\ \min(300, N) & 500 < N \leq 1000 \end{cases} \quad (5)$$

式中， N ：总音符数

算法采用规则评分体系，将 MIDI 信息的三个输入特征作为三个维度，量化评估三种策略对当前 MIDI 信息的适配程度，为每种策略加分，选择总分最高的策略。评分规则如下：

- (1) 有效轨道数量 T ：当为单轨($T=1$)时， S_1 加 3 分；当为多轨($T \geq 2$)时， S_2 加 3 分、 S_3 加 2 分；
- (2) 总音符数 N ：当 $N \leq 500$ 时， S_1 加 2 分；当 $500 < N \leq 1000$ 时， S_1 、 S_2 各加 1~2 分；当 $N > 1000$ 时， S_2 加 2 分、 S_3 加 3 分；
- (3) 主轨占比 R ：当 $R > 0.6$ 时， S_2 加 3 分；当 $0.3 < R \leq 0.6$ 时， S_2 加 2 分、 S_3 加 2 分；当 $R \leq 0.3$ 时， S_3 加 3 分。

该算法的决策逻辑为：首先遍历 MIDI 信息所有轨道，提取三个特征值；然后根据评分规则为三种策略累计加分；最后选择得分最高的策略作为输出。若各策略得分相同，按“ $S_2 \rightarrow S_3 \rightarrow S_1$ ”的优先级选择(优先保证旋律完整)。若最优策略为 S_3 (截取部分转换)，则根据总音符数，按式(5)动态计算最大保留音符数 K 。算法决策过程的决策树，如图 2 所示。

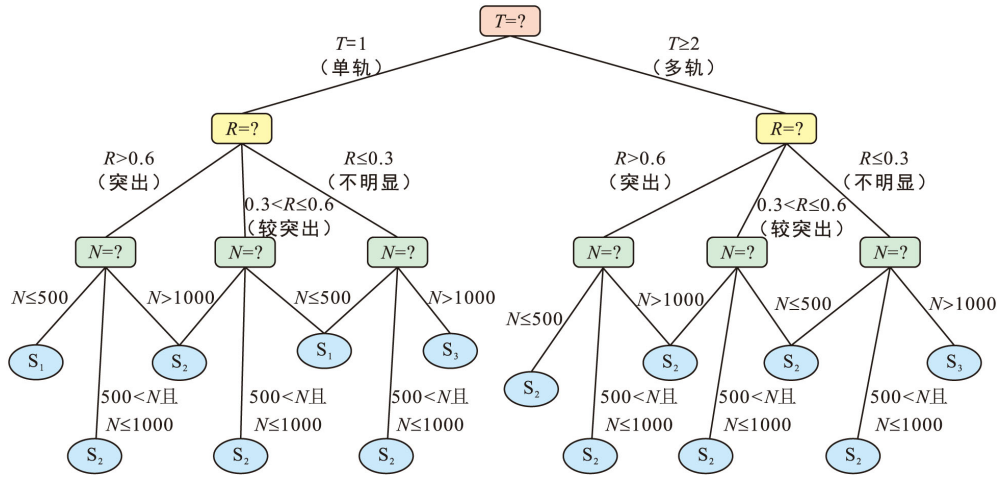


Figure 2. Decision tree for intelligent conversion strategy selection algorithm
图 2. 智能转换策略选择算法决策树

3. MIDI 信息智能转换程序设计与实现

根据第二章的理论模型，本文在 Python 环境下开发了 MIDI 信息转换程序。主函数是整个程序的入口，负责控制整体流程：遍历待转换文件[6]、调用各子函数、处理错误和验证，其执行逻辑如图 3 所示。

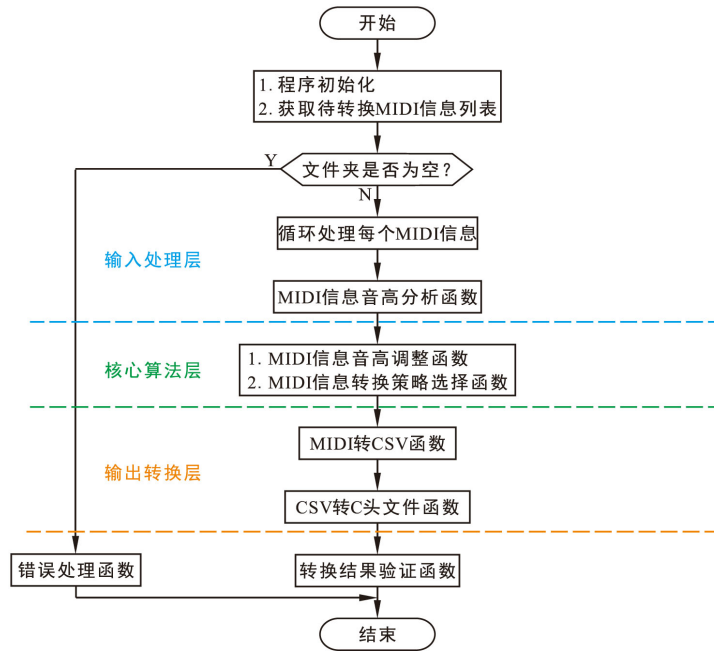


Figure 3. Main function flowchart
图 3. 主函数流程图

图 3 中，程序首先初始化并获取待转换的 MIDI 信息列表；若文件夹为空则触发错误处理；否则循环处理每个文件，依次调用音高分析、音高调整、策略选择、MIDI 转 CSV、CSV 转 C 头文件子函数。所有文件处理完毕后，调用验证函数检查结果，若过程中出现异常则转入错误处理函数。其中，MIDI 信息音高调整函数和 MIDI 信息转换策略选择函数分别为第二章提出的两个核心算法的实现。

3.1. MIDI 信息音高调整函数

MIDI 信息音高调整函数实现了第二章 2.1 节所述的智能音高偏移优化算法。该函数根据输入的音高统计字典,自动计算最佳音高偏移量,将原始 MIDI 信息中的音域适配到木琴有效音域(MIDI 编号 43~67)。算法的具体实现步骤如下:

输入: 音高统计字典 pitch_stats (键为 MIDI 编号, 值为该音高出现次数), 木琴有效音域 $V = [43, 67]$;

输出: 最优音高偏移量 d^* (半音数, 取值范围-48~48)。

- (1) 初始化 $S^* \leftarrow -\infty$, $d^* \leftarrow 0$;
- (2) **for** $d = -48$ to 48 **do**:
 - a) 初始化 $\text{in_range} \leftarrow 0$, $\text{mapped} \leftarrow 0$;
 - b) **for each** $(p, \text{freq}) \in \text{pitch_stats}$ **do**:
 - $p' \leftarrow p + d$;
 - if** $p' \in V$ **then** $\text{in_range} \leftarrow \text{in_range} + \text{freq}$;
 - else** $\text{mapped} \leftarrow \text{mapped} + \text{freq}$;
 - c) 计算音域内比例 R ;
 - d) 计算音高分布熵 E ;
 - e) 计算中心音高贴近度 C ;
 - f) 计算音域利用率 G ;
 - g) 计算映射惩罚因子 M ;
 - h) 计算位置适应度 P ;
 - i) 计算综合评分 $S \leftarrow 0.25R + 0.20E + 0.15C + 0.15G + 0.10M + 0.15P$;
 - j) **if** $d \bmod 12 = 0$ **then** $S \leftarrow S \times 1.05$;
 - k) **if** $S > S^*$ **then** $S^* \leftarrow S$, $d^* \leftarrow d$;
- (3) 返回 d^* 。

3.2. MIDI 信息转换策略选择函数

MIDI 信息策略选择函数实现了第二章 2.2 节所述的智能转换策略选择算法。该函数根据 MIDI 信息的轨道结构特征,自动选择最优转换策略,以适应不同的音乐类型和系统资源限制。算法的具体实现步骤如下:

输入: MIDI 信息路径;

输出: 策略代码 $\text{strategy} \in \{1, 2, 3\}$, 最大保留音符数 K (仅当 $\text{strategy} = 3$ 时有效)。

- (1) 读取 MIDI 信息, 遍历所有轨道, 统计: 有效轨道数 T 、总音符数 N 及主轨占比 R ;
- (2) 初始化三种策略得分 $\text{score}_1 = \text{score}_2 = \text{score}_3 = 0$;
- (3) 根据有效轨道数加分:
 - if** $T = 1$ **then** $\text{score}_1 + = 3$;
 - else** $\text{score}_2 + = 3$, $\text{score}_3 + = 2$;
- (4) 根据总音符数加分:
 - if** $N \leq 500$ **then** $\text{score}_1 + = 2$;
 - else if** $500 < N \leq 1000$ **then** $\text{score}_1 + = 1$, $\text{score}_2 + = 2$;
 - else** $\text{score}_2 + = 2$, $\text{score}_3 + = 3$;
- (5) 根据主轨占比加分:

- if $R > 0.6$ then $score_{2+} = 3$;
- else if $0.3 < R \leq 0.6$ then $score_{2+} = 2$, $score_{3+} = 2$;
- else $score_{3+} = 3$;
- (6) 选择得分最高的策略: $strategy = \text{argmax}(score_1, score_2, score_3)$; 若平分, 按 $S_2 \rightarrow S_3 \rightarrow S_1$ 优先级选择;
- (7) if $strategy = 3$ then:
 - if $N > 1000$ then $K = \min(500, \lfloor N/2 \rfloor)$;
 - else if $500 < N \leq 1000$ then $K = \min(300, N)$;
- (8) 返回 $strategy, K$.

4. 核心算法验证与分析

4.1. 智能音高偏移优化算法的验证

以《Little Star》和《Happy Birthday》为例, 在 Python 环境下, 对两个 MIDI 文件进行音高偏移量穷举搜索(范围-48 至 48 半音), 各维度评分随偏移量变化的曲线, 如图 4 所示。

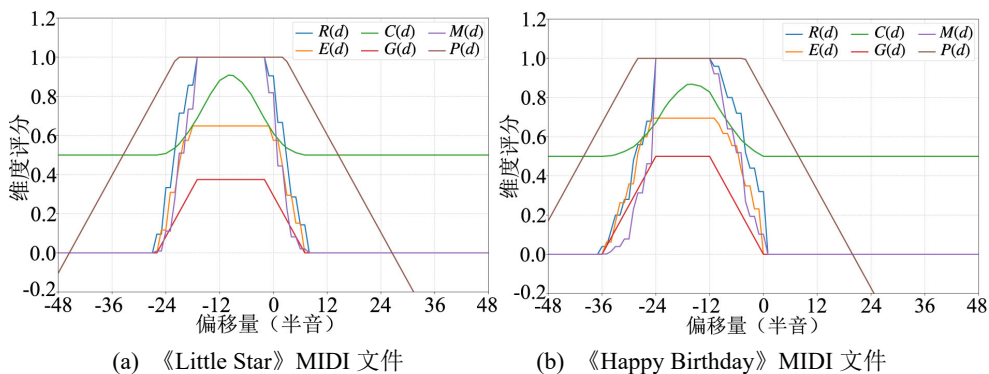


Figure 4. Dimensional scoring curves of two MIDI files
图 4. 两个 MIDI 文件维度评分曲线图

图 4 中, 据第二章表 1 分析可知, 各评分函数值越接近 1 表示该维度表现越优。因此, 《Little Star》MIDI 文件在原始音高(0 半音)或轻微降调(-12 半音)时, 整体音乐表现最佳; 《Happy Birthday》MIDI 文件在轻微降调(-12 半音)或-24 半音时, 整体音乐表现最佳。

综合评分随偏移量变化的曲线, 如图 5 所示。

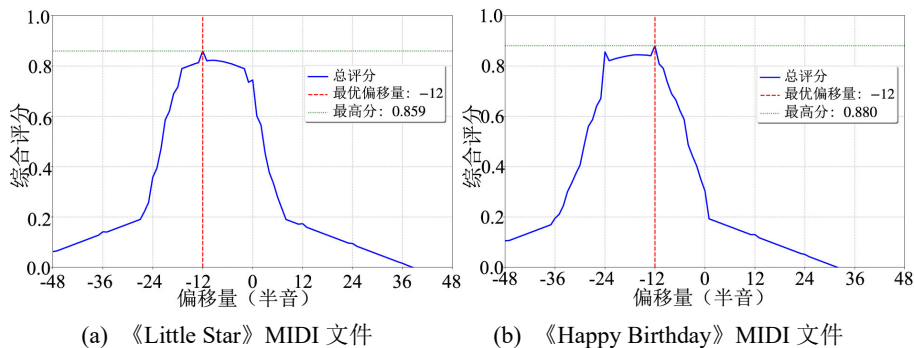


Figure 5. Comprehensive scoring curve of two MIDI files
图 5. 两个 MIDI 文件综合评分曲线图

图 5 表明, 两个 MIDI 文件均在偏移量为-12 半音时取得最高综合评分, 证明算法能有效将音域适配至木琴有效范围。

4.2. 智能转换策略选择算法的验证

以《Little Star》和《Happy Birthday》为例, 在 Python 环境下, 算法首先提取两个 MIDI 文件的三个关键特征, 如表 3 所示。

Table 3. Three key characteristics of two MIDI files

表 3. 两个 MIDI 文件的三个关键特征

NO.	歌曲名称	有效轨道数量 T	总音符数 N	主轨占比 R
01	《Little Star》	1	42	1.00
02	《Happy Birthday》	1	356	1.00

根据第二章的评分规则计算三种策略的综合得分, 结果如表 4 和表 5 所示。

Table 4. Scoring calculation table for intelligent conversion strategies of “Little Star”

表 4. 《Little Star》智能转换策略评分计算表

NO.	评分维度	特征区间	策略加分			选择结果
			S_1	S_2	S_3	
01	有效轨道数量	$T = 1$ (单轨)	+3	+1	+0	策略 1 (S_1)
02	总音符数	$N = 42$ (少)	+2	+1	+0	
03	主轨占比	$R = 1.00$ (突出)	+1	+3	+1	
总计			6	5	1	策略 1 (S_1)

Table 5. Calculation table for the score of the intelligent conversion strategy of “Happy Birthday”

表 5. 《Happy Birthday》智能转换策略评分计算表

NO.	评分维度	特征区间	策略加分			选择结果
			S_1	S_2	S_3	
01	有效轨道数量	$T = 1$ (单轨)	+3	+1	+0	策略 1 (S_1)
02	总音符数	$N = 356$ (少)	+2	+1	+0	
03	主轨占比	$R = 1.00$ (突出)	+1	+3	+1	
总计			6	5	1	策略 1 (S_1)

两个案例中策略 1 (全曲转换) 得分最高, 与单轨简单音乐的特征相符, 验证了算法决策的正确性。

4.3. 算法综合评估

为直观展示算法效果, 以《Little Star》和《Happy Birthday》为例, 对两个 MIDI 文件最优音高偏移量得分进行可视化分析, 如图 6 所示。

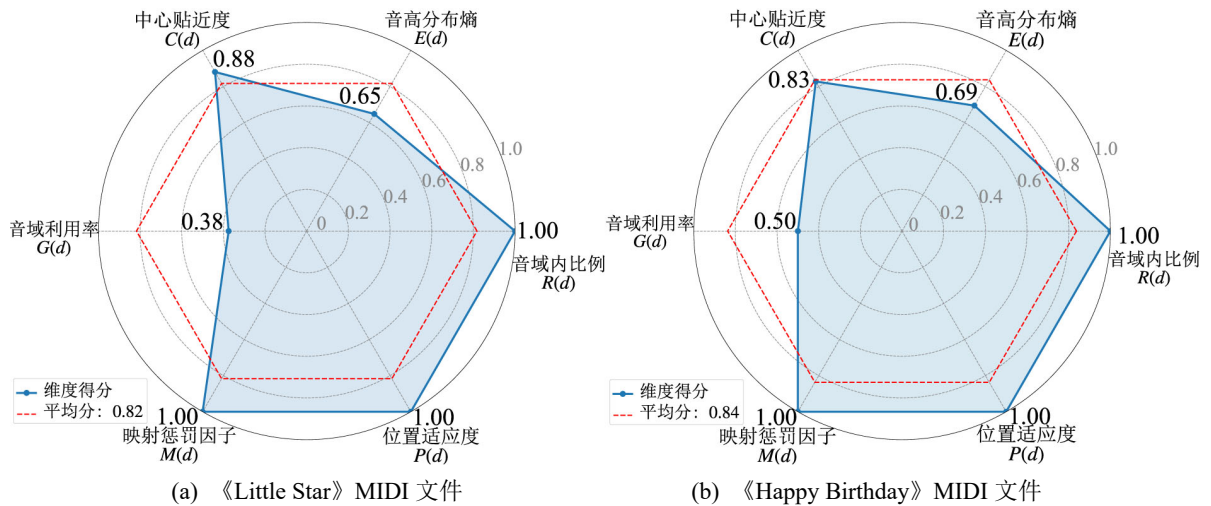


Figure 6. Radar chart of the optimal pitch offsets for two MIDI files
 图 6. 两个 MIDI 文件最优音高偏移量雷达图

图 6 中，雷达图展示了两个 MIDI 文件在最优偏移量下的六个维度得分，多边形面积越大表示综合表现越好，说明智能音高偏移优化算法能识别音乐的类型与特征，据音乐特性进行智能优化。

在 Python 环境下，对多个 MIDI 文件运用智能转换策略选择算法选择最优转换策略，得到的最优转换策略选择统计柱状图，如图 7 所示。

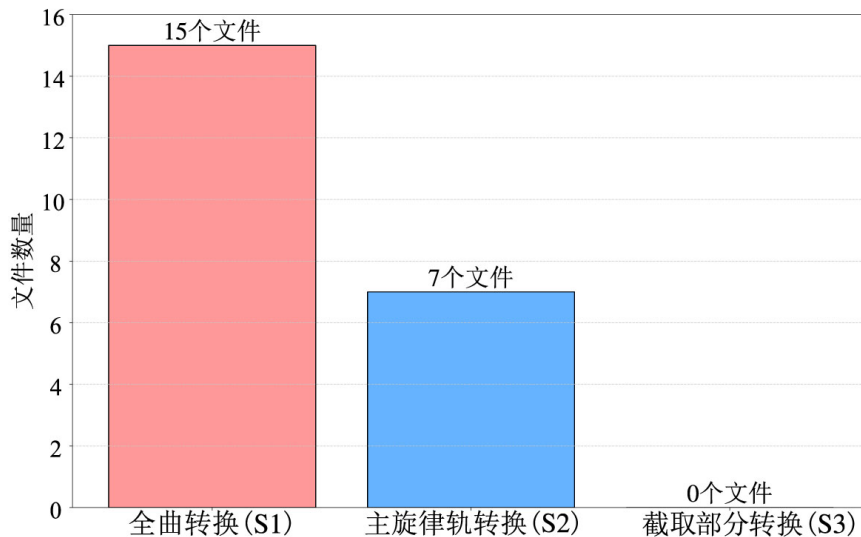


Figure 7. Statistical bar chart of optimal conversion strategy selection
 图 7. 最优转换策略选择统计柱状图

由图 7 可知，智能转换策略选择算法在大多数情况下倾向于选择全曲转换策略(尽量不选截取部分转换策略)，以保证歌曲旋律的完整性。同时，少部分情况选择主旋律轨转换策略，说明算法可据音乐内容动态选择策略。

在 Python 环境下，运行 MIDI 信息转换程序，在多个算法支持下得到的转换前后四个 MIDI 文件的音高分布折线图，如图 8 所示。

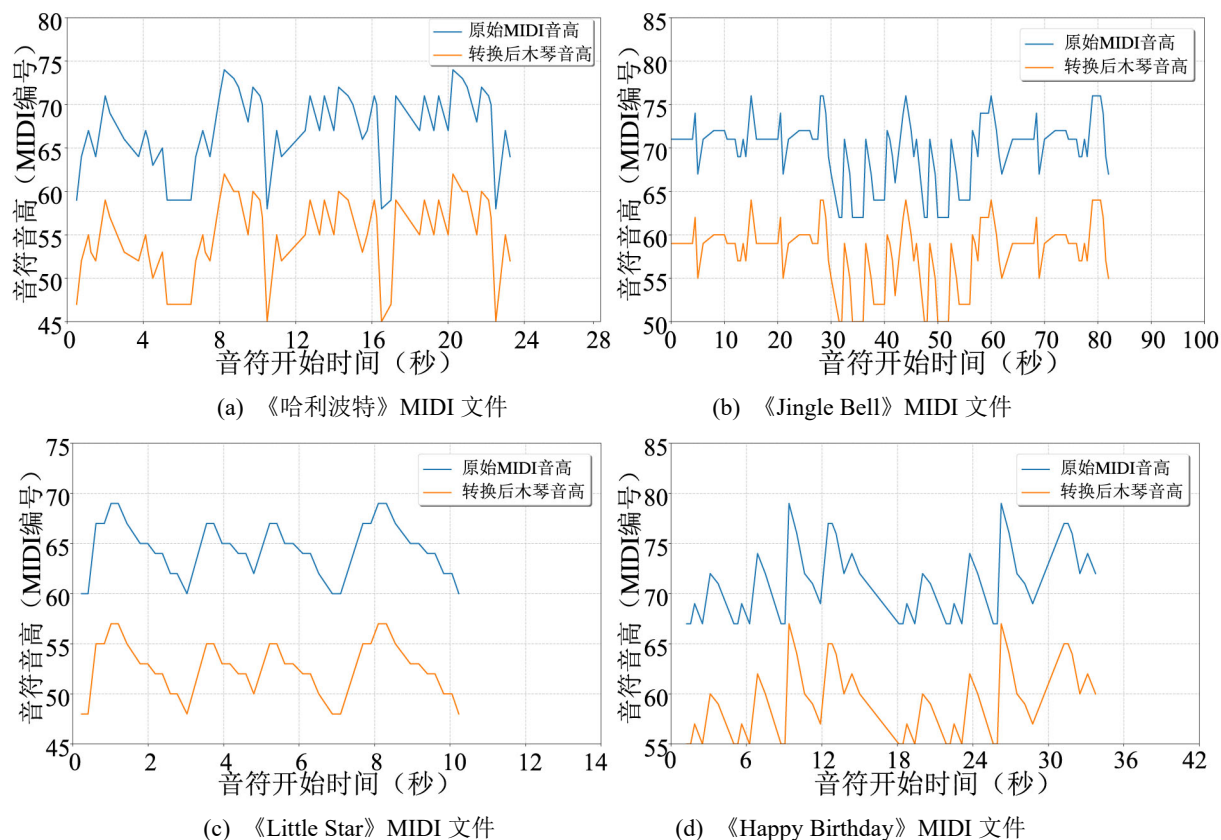


Figure 8. Line graph showing the pitch distribution of four MIDI files

图 8. 四个 MIDI 文件音高分布折线图

由图 8 可知，四个 MIDI 文件转换后的音高全部落在十五音木琴音域内(MIDI 编号 43~67)，且转换前后折线形状大致相似。表明程序能够正确适配不同音高范围和时长的 MIDI 文件，完整保留原始旋律轮廓与时间节奏。进一步分析两种算法的特性：智能音高偏移优化算法通过多维度加权评分全面评估音高分布，能找到较优偏移量、保持音乐质量，但计算开销较大且存在局部最优风险，适用于对质量要求高的关键曲目；智能转换策略选择算法简单高效、决策快速，适合实时应用，但特征维度较单一，可能误选策略，适用于大量 MIDI 信息的初步筛选。

5. 结论

本文针对电子木琴自动演奏系统中的 MIDI 信息智能转换问题，提出了两种核心算法，并进行了系统性的实现与验证。智能音高偏移优化算法通过六维度加权评分体系，自动计算最优音高偏移量，有效解决了有限音域乐器与宽音域 MIDI 信息之间的音域适配问题。智能转换策略选择算法基于轨道数、总音符数和主轨占比三个特征，通过规则评分自动选择最优转换策略，兼顾了旋律完整性与系统资源限制。在 Python 环境下开发的转换程序，实现了从 MIDI 信息到 C 语言头文件的完整转换流程。实验结果表明，两种算法均有效，转换后的音高分布与原始旋律轮廓高度一致，验证了算法的正确性和适用性。

参考文献

- [1] 郭凌华, 刘明磊, 丁亨文, 等. 基于 LabVIEW 数字图像生成音乐旋律算法的研究[J]. 西安理工大学学报, 2020, 36(1): 65-71.

- [2] 谢林玲. 多音轨 MIDI 音乐主旋律提取方法探析[J]. 科技创新导报, 2021, 18(19): 72-74.
- [3] 刘嘉欣. 嵌入式 MIDI 文件格式解析设计与实现[J]. 微计算机信息, 2006(32): 66-68+42.
- [4] 师冬冬. 基于深度学习的 MIDI 信号对电子音乐旋律自动转换研究[J]. 自动化与仪器仪表, 2024(11): 20-23+28.
- [5] 杨世瑞, 程科, 李想, 等. 基于多音轨聚类的 MIDI 音乐主旋律提取方法[J]. 计算机与数字工程, 2024, 52(10): 2897-2901.
- [6] 严桂林, 陈学煌, 赵云娥. 基于 S3C44B0 的 MIDI 音乐发生器的设计与实现[J]. 电子测量技术, 2008(8): 158-161.