

Generación y optimización de tablatura de guitarra a partir de lenguaje musical usando técnicas metaheurísticas

Autor: Juan Sánchez Jurado

Directores: José Francisco Chicano García, Enrique Alba Torres

Introducción

Las tablaturas son notaciones musicales más fáciles de entender y leer que la solfa debido a que muestran las posiciones que hay que adoptar en el instrumento para hacer sonar una nota o acorde. Mientras que la notación musical tradicional está orientada a indicar cuál o cuáles son las notas que deben sonar en el instrumento en cada instante, las tablaturas están orientadas a indicar cómo se deben tocar tales notas. Por este motivo, es una forma de representación musical adaptada al instrumento que goza de gran popularidad por su sencilla interpretación. En el caso particular de la guitarra, cordófono de cuerda pulsada, la tablatura indica al menos el(los) traste(s) que debe(n) pisarse en un momento determinado y la(s) cuerda(s) que debe(n) pulsarse.

The figure shows a musical score with two staves. The top staff is a standard musical notation in 6/8 time, featuring a treble clef and a key signature of one flat. It contains four measures of music, each with a single eighth note followed by a dotted eighth note. The bottom staff is a guitar tablature with two lines labeled 'T' (Treble) and 'B' (Bass). It shows the fret numbers for each note in the four measures, corresponding to the notes in the musical notation above. The fret numbers are: Measure 1: T=2, B=0; Measure 2: T=1, B=0; Measure 3: T=0, B=0; Measure 4: T=1, B=0.

Figura 1: Notación musical tradicional y tablatura para el mismo fragmento musical

Dada una partitura escrita en lenguaje musical tradicional existen múltiples formas de transcribir dicha partitura a tablatura para guitarra. El origen de esta diversidad se encuentra en las distintas opciones disponibles para tocar una misma nota usando la guitarra. Así por ejemplo, el re³ se puede tocar en el décimo traste usando la 6^a cuerda, en el quinto traste usando la 5^a cuerda o en la 4^a cuerda al aire. Si en lugar de hablar de notas individuales aplicamos este razonamiento a los acordes, el conjunto de posibilidades se multiplica aproximadamente por el número de notas del acorde. A cada una de estas posibles formas de tocar una nota o acorde se le llama *digitación* de dicha nota o acorde.

No todas las digitaciones son igualmente buenas. Por un lado, debemos tener en cuenta si dicha digitación es posible para un determinado guitarrista o si es más o menos cómoda para él. Distintos guitarristas con distinta experiencia y tamaño de mano preferirán digitaciones diferentes. Por otro lado, es necesario tener en cuenta la digitación de la nota o acorde anterior y del siguiente para encontrar una digitación que implique un esfuerzo mínimo para el guitarrista.

El problema de la digitación no es particular de la guitarra, sino que existe de una forma u otra en todos los instrumentos. Normalmente, durante los estudios musicales en conservatorios y escuelas de música, los intérpretes aprenden a resolver el problema por su cuenta de forma casi automática. Además, con el paso del tiempo, aprenden cuáles

son las mejores digitaciones para ellos. Por este motivo, la digitación no supone ningún problema para los músicos profesionales, sean guitarristas o no.

Sin embargo, existe una gran cantidad de guitarristas aficionados que son capaces de leer tablaturas pero no leen con soltura el lenguaje musical tradicional. Estos guitarristas no están acostumbrados, por tanto, a elaborar digitaciones y suelen acudir a versiones en tablatura de las piezas. Debido a que la digitación es un aspecto muy personal de la interpretación, en muchas ocasiones estas versiones tabuladas no son del agrado del intérprete. Existen herramientas automáticas que generan tablaturas a partir de notación musical tradicional, como es el caso de TableEdit [Tab]. No obstante, estos programas no tienen en cuenta las características particulares del intérprete que va a usar la tablatura.

El presente proyecto de fin de carrera tiene como objetivo desarrollar una herramienta que permita generar las tablaturas más adecuadas para un intérprete de guitarra a partir de una partitura en lenguaje musical tradicional. Definida una función de adecuación para evaluar las tablaturas, el problema a resolver es un problema de optimización en el que el objetivo es encontrar una solución de entre un conjunto de soluciones posibles (el conjunto de todas las tablaturas posibles para una partitura dada) que maximice dicha función de adecuación. Para resolver este problema de optimización, y puesto que el espacio de búsqueda es demasiado extenso como para aplicar algoritmos exhaustivos, proponemos el uso de técnicas *metaheurísticas*.

Las metaheurísticas [BR03] son técnicas de optimización surgidas en los años setenta cuya idea básica es combinar diferentes métodos heurísticos a un alto nivel para conseguir una exploración del espacio de búsqueda de forma eficiente y efectiva. El término metaheurística fue introducido por primera vez por Glover [Glover86]. Antes de que el término fuese aceptado completamente por la comunidad científica, estas técnicas eran denominadas *heurísticas modernas* [Reeves93]. Esta clase de algoritmos incluye colonias de hormigas, algoritmos evolutivos, búsqueda local iterada, enfriamiento simulado, y búsqueda tabú. De las diferentes descripciones de metaheurísticas que se encuentran en la literatura se pueden destacar ciertas propiedades fundamentales que caracterizan a este tipo de métodos. A modo de resumen, podemos decir que una metaheurística es una plantilla general no determinista que debe ser rellenada con datos específicos del problema (representación de las soluciones, operadores para manipularlas, etc.) y que permiten abordar problemas con espacios de búsqueda de gran tamaño.

Objetivos

El objetivo del presente proyecto de fin de carrera es desarrollar una herramienta en Java que permita generar las tablaturas óptimas para un intérprete a partir de una partitura en lenguaje musical tradicional. Para que la herramienta sea de gran utilidad es necesario que pueda tomar la entrada y escribir la salida en un formato de fichero estándar para notación musical. Dicho formato es MusicXML [Mus] que surgió con el objetivo de unificar los formatos de publicación de partituras. La gran mayoría del software de edición de partituras (Sibelius, Finale, etc.) puede leer y escribir ficheros en formato MusicXML

Fases del proyecto

El proyecto se realizará siguiendo un proceso de desarrollo en espiral. En cada iteración se irán incorporando nuevas funciones a la herramienta. A continuación se muestran las distintas iteraciones para el desarrollo de la herramienta final. Estas iteraciones se corresponden con las distintas fases del proyecto.

1. Diseño de la representación interna para las partituras y funciones asociadas a la determinación de las digitaciones posibles de cada nota o acorde.
2. Diseño de la función de adecuación e incorporación de la búsqueda usando metaheurísticas para fragmentos musicales.
3. Desarrollo del módulo para la entrada y salida en formato MusicXML.
4. Análisis y diseño de un módulo para la identificación de fragmentos similares en la partitura y mejora de la búsqueda.
5. Pruebas con grandes partituras y comparación con tablaturas generadas por otras aplicaciones disponibles.
6. Análisis de la función de adecuación y posible ampliación para aspectos más complejos del intérprete.
7. Documentación de la herramienta y Web asociada.

Medios materiales

- Equipo portátil o PC
- Entorno de desarrollo Java
- Adobe Dreamweaver para el desarrollo de las páginas web, hoja de estilos y documentos XML.
- Editor de textos y compilador de LaTeX para la memoria y documentos intermedios.
- Libros y artículos aportados por los directores del proyecto. Revistas de la hemeroteca y de acceso electrónico.

Referencias

- [BR03] C. Blum and A. Roli. *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*. ACM Computing Surveys, 35(3):268–308, 2003.
- [Glover86] F. Glover. *Future paths for integer programming and links to artificial intelligence*. Computers & Operations Research, 13:533–549, 1986.
- [Mus] <http://www.recordare.com>
- [Reeves93] C.R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publishing, Oxford, UK, 1993.
- [Tab] <http://www.tabledit.com>