

Exprimer les Idées Musicales avec la Programmation par Contraintes: un Assistant pour Compositeurs

Damien Sprockeels* Peter Van Roy

Université catholique de Louvain, Belgique
{damien.sprockeels, peter.vanroy}@uclouvain.be

Résumé

La composition musicale avec l'intelligence artificielle est un domaine en constante évolution, attirant un intérêt croissant. Cet article présente un projet utilisant une approche basée sur la programmation par contraintes pour générer des morceaux musicaux respectant les règles d'une technique musicale donnée ainsi que les préférences exprimées par les compositeurs. Cette approche exploite la puissance de la programmation par contraintes en tant que logique formelle pour modéliser rigoureusement les règles musicales et offrir un contrôle complet sur l'ensemble des règles appliquées. Cela permet aux compositeurs d'interagir de manière itérative avec le modèle, en ajoutant et en supprimant des contraintes, leur permettant ainsi de façonner les solutions selon leurs préférences.

Cet article définit un modèle de contraintes d'harmonie tonale diatonique, appelé Diatony. Nous démontrons que notre implémentation utilisant le solveur Gecode trouve des solutions optimales en un temps raisonnable et nous montrons comment elle peut être utilisée par un compositeur dans son processus de composition. Ce modèle servira de base pour générer des morceaux dans différents styles musicaux.

Abstract

The realm of musical composition with artificial intelligence is a relevant and constantly evolving field, attracting increasing interest. This article presents an ongoing project using a constraint-based programming approach to generate musical pieces that respect the rules of a given musical technique as well as the preferences expressed by composers. This approach leverages the power of constraint programming as a formal logic to rigorously model musical rules and offer complete control over the set of applied rules. This allows composers to interact

iteratively with the model, adding and removing constraints, thus enabling them to shape solutions according to their preferences.

This article defines a model of diatonic tonal harmony, called Diatony. We demonstrate that our implementation using the Gecode solver finds optimal solutions in a reasonable time and show how it can be used by a composer in their composition process. This model will serve as a basis for generating pieces in different musical styles.

1 Introduction

L'application de la programmation par contraintes au domaine de la composition musicale a été longuement étudiée [10, 11]. La relation étroite entre la musique et les mathématiques rend attrayante l'idée d'une formalisation mathématique de styles musicaux complets. Il existe de nombreux travaux modélisant des problèmes musicaux et règles musicales variés [2, 7], mais ces travaux se concentrent d'avantage sur la résolution de problèmes musicaux sous forme mathématique que sur l'utilisation possible pour les compositeurs.

Il existe cependant quelques tentatives de création d'outils utilisant de telles formalisations, mais il n'y a pas à l'heure actuelle d'outil modélisant complètement un style musical complexe. Ovans et Davidson [9] proposent un outil modélisant le contrepoint dans le style de Fux, mais seulement pour la première espèce à deux voix, et Truchet [13] propose une librairie permettant de résoudre certains problèmes musicaux.

Notre objectif est d'utiliser les connaissances acquises dans ces précédents travaux pour créer un outil modélisant plusieurs styles musicaux complets pouvant être utilisé par les compositeurs. Dans ce cadre, nous

*Papier doctorant : Damien Sprockeels est auteur principal.

présentons le modèle général d’harmonie tonale diatonique que nous avons développé. Ce modèle, bien que simple, peut déjà être utilisé par les compositeurs pour développer des phrases harmoniques. Il a cependant été développé pour être utilisé comme une base pour modéliser différents styles de musique tonale et modale. Ensuite, nous discuterons des prochaines étapes de notre travail.

2 Modèle d’harmonie tonale diatonique

Cette section définit un modèle de base de l’harmonie tonale que nous utilisons pour composer de la musique tonale. Notre modèle, appelé *Diatony*, définit une texture à quatre voix avec des triades diatoniques et des accords de septième de dominante, y compris leurs renversements¹. Le modèle contient des règles harmoniques et mélodiques ainsi que des préférences pour créer des progressions d’accords diatoniques. Il contient des règles pour l’occurrence des notes dans les accords, ainsi que des règles pour le mouvement des voix entre les accords. Le modèle est basé sur une sélection de règles issues de deux traités de théorie musicale. Ensemble, ils fournissent un modèle complet et cohérent des bases de l’harmonie tonale qui s’applique à la majorité de la musique tonale. Les travaux futurs étendront ce modèle pour prendre en charge les modulations ainsi que les accords non diatoniques. Nous prenons dans Duha [4] les chapitres sur les accords de 3 et 4 sons et les renversements (*Accords de 3 et 4 sons*). Dans Gauldin [5], nous prenons les chapitres sur la texture à quatre voix, l’harmonie diatonique, la septième de dominante et les renversements (chapitres 5 à 11, 16 et 17). Une liste complète des règles musicales et leur formalisation mathématique se trouve dans [12], mais est omise dans cet article pour des raisons de brièveté. Le modèle est implémenté à l’aide de Gecode [6].

Le modèle Diatony se compose de deux parties : des règles strictes, qui doivent être suivies en tout temps, et des préférences, qui peuvent ou non être suivies. Cette section donne une liste complète des règles musicales du modèle Diatony.

2.1 Règles strictes

Les règles musicales implémentées sont les suivantes :

- Les voix doivent être dans leur registre assigné et les accords doivent avoir la bonne fondamentale, qualité et état (**H1**, **H2** et **H3**).

1. La texture à quatre voix est une technique couramment utilisée pour représenter l’harmonie. Il y a quatre voix distinctes, à savoir la basse, le ténor, l’alto et le soprano, de la plus basse à la plus haute, qui jouent des accords. Les accords diatoniques sont des accords formés avec des notes de la tonalité donnée.

- Dans les accords à l’état fondamental, la basse doit être doublée (**H4**).
- D’un accord à l’état fondamental à un autre, les notes communes doivent être maintenues dans la même voix et les autres voix doivent se déplacer vers la note la plus proche (**P4**, **P5**).
- S’il n’y a pas de notes communes entre deux accords à l’état fondamental de degrés successifs, les voix supérieures doivent se déplacer en mouvement contraire à la basse (**M1**).
- l’intervalle harmonique entre deux voix ne peut pas être une quinte ou octave juste dans deux accords successifs, à moins que les notes soient les mêmes (**M2**).
- Lorsque le triton² de la tonalité est présent dans un accord, il doit être résolu dans l’accord suivant. La voix jouant le quatrième degré de la gamme doit descendre conjointement vers le troisième degré de la gamme, et la voix jouant la note sensible doit monter conjointement vers la tonique (**M3**).
- Dans le cas d’une cadence rompue, si le mode est mineur ou si la note sensible est au soprano, la note sensible doit monter conjointement vers la tonique et les autres voix doivent descendre. La tierce de l’accord doit être doublée pour l’accord de sixième degré au lieu de la fondamentale (**M4**, **H6**).
- Pour les accords en première inversion, chaque note doit être présente une fois et n’importe quelle note tonale³ peut être doublée. Si la basse est une note tonale, elle peut aussi être doublée (**H7**).
- Si la basse et le soprano bougent conjointement en mouvement contraire d’un accord à l’état fondamental à un accord en première inversion puis à un accord à l’état fondamental, la basse peut être doublée dans le second accord même si elle n’est pas tonale (**H9**).
- Pour l’accord diminué de septième degré en première inversion, la fondamentale doit être doublée (**H7**).
- Pour les accords en seconde inversion, la basse doit être doublée (**H8**).
- Si un accord en seconde inversion est l’appoggiature de l’accord de cinquième degré, la tonique doit être approchée par mouvement contraire ou oblique. Le troisième degré et la tonique doivent descendre conjointement (**M5**).
- Si un accord diminué est en seconde inversion, la

2. Le triton dans une tonalité est peut-être la notion la plus importante de l’harmonie tonale. C’est l’intervalle entre le quatrième degré et la note sensible, qui est un demi-ton en dessous de la tonique. Il est très dissonant et demande à être résolu.

3. Les notes tonales sont le premier, quatrième et cinquième degré de la tonalité.

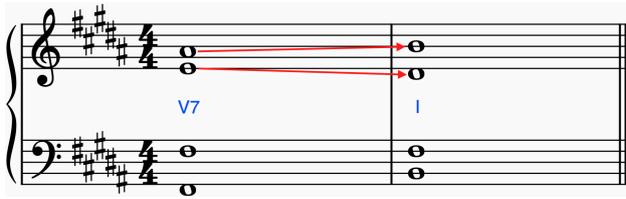


FIGURE 1 – Résolution de triton en si majeur.

tierce de l'accord doit être doublée (**H10**).

- Dans les cadences parfaites, l'un des accords doit être incomplet⁴. Si l'accord de cinquième degré est incomplet, sa basse doit être doublée. Si l'accord de premier degré est incomplet, la basse doit être triplée (**H5**).
- Si un accord de septième de dominante est en seconde inversion, la résolution du triton peut être modifiée de sorte que le quatrième degré de la gamme monte conjointement au lieu de descendre (**M3**).

2.2 Préférences

Nous définissons les préférences suivantes, par ordre décroissant d'importance :

- Les accords doivent être complets (**P1**).
- Les accords diminués à l'état fondamental doivent être utilisés avec trois voix au lieu de quatre (**P2**).
- Les accords doivent avoir quatre valeurs de notes différentes⁵ (**P3**).
- Les intervalles mélodiques doivent être petits (**P4**).
- Les notes communes doivent être maintenues dans la même voix (**P5**).

2.3 Modèle

Comme détaillé dans les sections précédentes, les règles musicales sont distinguées en deux catégories : les règles strictes, qui doivent être respectées en permanence, et les préférences, qui sont des règles qui peuvent ne pas être respectées. Les règles strictes sont implémentées sous forme de contraintes, tandis que les préférences sont implémentées sous forme de coûts à minimiser dans un ordre lexicographique.

Un bon exemple de règle mélodique est que lorsque le triton de la tonalité est présent dans un accord, il doit se résoudre dans l'accord suivant. Cela signifie que la voix jouant le quatrième degré de l'échelle doit

4. Un accord est complet si toutes les notes qui composent l'accord sont présentes au moins une fois. Si une note n'est pas présente, cela doit être la quinte de l'accord.

5. À ne pas confondre avec le fait d'avoir quatre notes différentes. Cette préférence indique qu'une note peut être présente plus d'une fois, mais elle ne doit pas être dans la même octave.

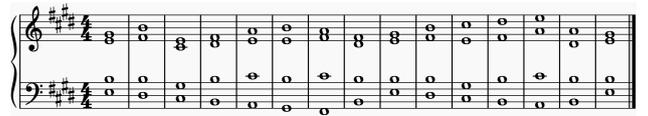


FIGURE 2 – Solution optimale trouvée par le solveur pour l'enchaînement d'accords I5-V6-VI5-V5-IV5-I6-II5-V5-I5-V6-VI5-V5-IV5-V7+-I5 en si majeur. Il peut être écouté ici : <https://on.soundcloud.com/4tpXL>

descendre conjointement vers le troisième, tandis que la voix jouant la sensible doit monter conjointement vers la tonique. Cette situation se produit généralement lorsqu'un accord de septième de dominante du cinquième degré est suivi de l'accord du premier degré. La figure 1 montre la résolution du triton dans une cadence parfaite en si majeur. Cette règle peut être exprimée mathématiquement comme suit :

$$\forall c \in [1, n], \forall v \in [1, 4]$$

$$N[c][v] \bmod 12 = T_q \implies M[c][v] \in \{-1, -2\} \quad (1)$$

$$N[c][v] \bmod 12 = T_s \implies M[c][v] = 1 \quad (2)$$

Où N est le tableau contenant les notes pour chaque accord, c représente un accord, v une voix de l'accord, T_q et T_s le quatrième degré et la sensible de la tonalité respectivement, et M représente les mouvements mélodiques de chaque voix.

Un exemple de préférence est le fait que l'on tend à minimiser les intervalles mélodiques au sein d'une même voix. Cette règle est implémentée en assignant un poids à chaque intervalle mélodique, et en faisant une combinaison linéaire.

$$cost_{mel} = \sum_i \sum_v w[|M[i][v]|] \quad (3)$$

La figure 2 montre un exemple de solution optimale générée par le solveur pour une séquence d'accords donnée. La formalisation mathématique a été implémentée avec le solveur Gecode et peut être trouvée ici : <https://github.com/sprockeelsd/Diatony>

2.4 Recherche et optimisation

Puisque le modèle utilise un système de préférences, il est logique de chercher la solution optimale aux problèmes. L'espace de recherche croît rapidement en fonction de la longueur du morceau puisque le modèle Diatony ne donne que la structure musicale de base du morceau et n'a pas de contraintes globales. L'intention est que le compositeur ajoute des idées musicales à cette structure pour restreindre l'espace de recherche et converger rapidement vers le résultat musical souhaité, par exemple une forme pour la mélodie de la

voix supérieure. La modélisation d'un style spécifique permettra aussi d'avoir des contraintes globales réduisant l'arbre de recherche. Cette approche donne une grande liberté au compositeur dans sa composition, mais rend la recherche moins efficace sans l'ajout de règles personnelles.

Pour l'algorithme de recherche, nous avons choisi d'utiliser une approche basée sur les redémarrages avec un solveur de type "brancher et borner" lors de l'exploration de l'arbre de recherche. L'approche brancher et borner permet de trouver des solutions optimales en ajoutant une contrainte chaque fois qu'une solution est trouvée, et l'approche basée sur les redémarrages permet de réduire progressivement l'espace de recherche. L'efficacité de notre approche est suffisante pour rendre le modèle utilisable pour les compositeurs. Comme discuté dans la section 3, elle peut cependant être améliorée.

3 Prochaines étapes

Afin d'avoir un outil utilisable en pratique par les compositeurs, il est capital d'avoir une interface à la fois intuitive à utiliser et qui s'insère facilement dans leur routine de travail. Ceci est la prochaine étape de notre travail. Nous avons décidé de développer un plugin pour séquenceur comme Ableton [1] et Logic Pro [3]. Il sera crucial d'interagir avec des compositeurs pendant le développement de cette interface.

D'autre part, il conviendra d'étendre le modèle musical de base, à la fois avec les autres accords de 4 sons, les accords empruntés et les modulations. Les règles musicales inhérentes à ces concepts sont bien définies.

Pour avoir un outil véritablement utile aux compositeurs, il faut qu'il ait connaissance des règles de base inhérentes à un style musical donné. Il est donc nécessaire d'implémenter ces règles musicales sous forme de contraintes afin de fournir aux compositeurs un environnement de travail utile. Cette tâche est plus complexe parce que ce qui définit un style musical n'est pas toujours évident à déterminer. Des travaux comme cette thèse de Nobile [8] pourront servir de base pour extraire ces règles.

L'aspect rythmique de la génération de musique devra aussi être abordé. Notre modèle est un modèle harmonique et non rythmique. Définir le rythme avant de s'occuper de l'harmonie a comme inconvénient de limiter les solutions potentielles, mais générer le rythme simultanément entraîne un nombre de variables non constant.

Enfin, la recherche de solutions pour Diatony pourrait être améliorée, notamment en améliorant la qualité de la première solution trouvée, ce qui pourrait diminuer grandement le temps de recherche. Une solution

pourrait être de brancher sur les intervalles mélodiques au lieu des notes de l'accord, ou d'améliorer l'heuristique de valeur.

Références

- [1] ABLETON AG : Ableton Live, 1999.
- [2] Torsten ANDERS : *Composing music by composing rules : Design and usage of a generic music constraint system*. Thèse de doctorat, Queen's University Belfast, 2008.
- [3] APPLE INC. : Logic Pro, 1993.
- [4] Isabelle DUHA : *L'harmonie en liberté : de la mémoire à l'improvisation*. Gérard Billaudot, Armiane Imp., 2016.
- [5] Robert GAULDIN : *Harmonic Practice in Tonal Music. Second Edition*. W. W. Norton and Company, Inc, 2004.
- [6] GECODE TEAM : *Gecode : Generic Constraint Development Environment*, 2019.
- [7] Mikael LAURSON et Mika KUUSKANKARE : A constraint based approach to musical textures and instrumental writing. *In Seventh International Conference on Principles and Practice of Constraint Programming, Musical Constraints Workshop*. Citeseer, 2001.
- [8] Drew F NOBILE : *A structural approach to the analysis of rock music*. City University of New York, 2014.
- [9] Russell OVANS et Rod DAVISON : An iterative constraint-based expert assistant for music composition. *In Proceedings of the Biennial Conference-Canadian Society for Computational Studies of Intelligence*, pages 76–76. Citeseer, 1992.
- [10] François PACHET et Pierre ROY : Musical harmonization with constraints : A survey. *Constraints*, 6:7–19, 2001.
- [11] Örjan SANDRED : Constraint-Solving Systems in Music Creation. *Handbook of Artificial Intelligence for Music : Foundations, Advanced Approaches, and Developments for Creativity*, pages 327–344, 2021.
- [12] Damien SPROCKEELS et Peter VAN ROY : Expressing musical ideas with constraint programming using a model of tonal harmony. Accepted in the 33rd International Joint Conference on Artificial Intelligence 2024. <http://hdl.handle.net/2078.1/287476>.
- [13] Charlotte TRUCHET et Philippe CODOGNET : Musical constraint satisfaction problems solved with adaptive search. *Soft Computing*, 8(9):633–640, 2004.