

Le Projet Harmoniser : un Outil Pratique à Quatre Couches pour la Composition Musicale Basé sur les Contraintes

Damien Sprockeels¹, Peter Van Roy¹

¹ Université catholique de Louvain, ICTEAM

{damien.sprockeels,peter.vanroy}@uclouvain.be

Résumé

Le projet Harmoniser réalise un outil basé sur la programmation par contraintes pour aider les compositeurs de musique. Nous définissons un modèle de composition pour la musique tonale en quatre couches. Les quatre couches sont (1) la structure globale, (2) les progressions d'accords et les modulations, (3) la disposition des accords à quatre voix, et (4) les ornements. Nous avons réalisé les couches 2 et 3 dans le solveur Gecode. Nous sommes en train de réaliser les autres couches avec une interface graphique comme plug-in pour une station de travail audio numérique.

Mots-clés

Musique tonale, Station de travail audio numérique, programmation par contraintes.

Abstract

The Harmoniser project is building a tool to aid music composers in creating musical pieces using constraint programming. We define a four-layer composition process for tonal music that is useful for a significant number of composers. The four layers are (1) global structure, (2) chord progressions and modulations, (3) voicing (four-voice chord layout), and (4) ornaments (adding local complexity). We have defined layers 2 and 3 and implemented them in the Gecode solver. We are building the remaining layers with a GUI as a plug-in for a Digital-Audio Workstation.

Keywords

Tonal music, Digital-Audio Workstation (DAW), constraint programming.

1 Introduction

La composition musicale est depuis longtemps considérée comme une application prometteuse de la programmation par contraintes (PC), car la musique tonale occidentale est hautement combinatoire et repose sur une base théorique solide ([9, 7]). La théorie musicale et la composition musicale sont deux domaines vastes, et nous ne prétendons pas les aborder de manière exhaustive. Nous avons plutôt choisi un sous-ensemble pertinent de la théorie musicale tonale et proposé un processus de composition qui, bien que non général, est utile à un nombre significatif de compositeurs.

Travaux antérieurs La programmation par contraintes est une technique populaire pour la génération ([17, 18, 4]). La génération musicale avec PC est également populaire. Elle est parfois ajoutée à certaines formes d'apprentissage ([14, 11]) pour offrir un meilleur contrôle à l'utilisateur et une meilleure structure globale. Cependant, l'apprentissage est limité par les données d'entraînement. Une approche alternative consiste à formaliser la théorie musicale et à s'appuyer uniquement sur PC ([13, 1, 3, 5, 6]). Cette alternative présente de multiples avantages. Premièrement, il n'y a aucune limitation liée aux données d'entraînement. Deuxièmement, elle garantit que les règles sont respectées dans chaque solution générée. Troisièmement, elle facilite la manipulation des solutions, offrant ainsi plus de contrôle aux compositeurs. Cependant, elle présente deux inconvénients. Premièrement, un ensemble conséquent de règles est nécessaire, et deuxièmement, la recherche de solutions nécessite souvent un temps de calcul important. Les travaux antérieurs utilisant PC pour créer des outils de composition se divisent en deux catégories. La première concerne les outils formalisant un aspect spécifique de la théorie musicale ([8, 23, 12]). La seconde concerne les outils capables de modéliser une théorie musicale ([2, 15, 20]), mais nécessitant de la programmation de la part du compositeur, ce qui limite le public d'utilisateurs. Nous proposons une approche différente qui modélise entièrement un style musical sans nécessiter de la programmation.

Le projet Harmoniser Le projet Harmoniser définit un processus de composition à quatre couches, chaque couche contenant un modèle de contraintes. Les contraintes sont tirées de traités de théorie musicale et formalisées mathématiquement (voir Sections 3 et 4). L'application de ces contraintes par le solveur décharge le compositeur d'un travail fastidieux, qui lui permet de se concentrer sur l'ajout d'idées musicales pour façonner les solutions et obtenir le résultat souhaité. Afin de réduire la complexité de ce calcul, nous suivons la décomposition du processus de composition introduite dans [21], conformément à [16], qui conclut qu'une structuration appropriée est nécessaire pour rendre le problème résolvable. Nous décomposons le processus de composition musicale en problèmes plus petits, pouvant être résolus indépendamment (voir Section 2).

Structure Cet article donne un aperçu du projet Harmoniser en présentant quelques de ses idées les plus im-

portantes. Pour une présentation plus complète, nous renvoyons le lecteur aux publications [21, 22]. La Section 2 explique d’abord le processus de composition en quatre couches. Nous présentons ensuite les principales idées des deux couches actuellement implémentées : la Couche 2 (progressions d’accords et modulations, voir la Section 3) et la Couche 3 (les voix, voir la Section 4). Pour chaque couche, le modèle formel est défini mathématiquement par des contraintes, et son implémentation pratique est réalisée dans le solveur de contraintes Gecode [10]. La Section 5 propose une brève évaluation de l’outil actuel, notamment son efficacité de calcul et son utilisation par un compositeur. Enfin, la Section 6 explique la suite du projet.

2 Processus de Composition

Il existe probablement autant de façons de composer de la musique qu’il y a de compositeurs. Dans un outil pratique, il est néanmoins important de fournir une structure prédéfinie pour guider les compositeurs. Nous avons identifié un processus itératif à quatre couches, suffisamment général pour être pertinent pour un nombre significatif de compositeurs. Nous expliquons brièvement les quatre couches.

La structure globale (Couche 1) La première couche décompose la pièce musicale complète hiérarchiquement en thèmes, où chaque thème primitif est une juxtaposition de progressions incluant des modulations. Chaque progression contient une séquence d’accords dans une tonalité. Deux progressions successives sont reliées par une modulation.

Progressions et modulations (Couche 2) La deuxième couche réalise le développement harmonique au sein de chaque progression. Elle définit une séquence de degrés d’accords pour chaque progression, ainsi que des modulations pour la transition d’une progression à la suivante. Les degrés d’accords peuvent être diatoniques (appartenant à la tonalité) ou chromatiques (n’appartenant pas). La Section 3 présente les idées principales du modèle formel de cette couche. La définition complète est donnée dans [22].

Les voix (Couche 3) La troisième couche définit les voix, c’est-à-dire les notes pour chaque accord. La disposition standard d’un accord est à quatre voix. Deux aspects sont à prendre en compte : l’harmonie, c’est-à-dire l’interaction entre les notes d’un accord donné, et la mélodie, c’est-à-dire l’interaction temporelle entre les notes d’une même voix. Certaines règles des voix influencent les états des accords, ce qui signifie que la couche 2 contient également des règles des voix. La définition complète est donnée dans [21].

Les ornements (Couche 4) La quatrième couche est celle des ornements mélodiques. Étant donné l’harmonisation d’une progression d’accords, des ornements tels que des notes de passage ou des appoggiatures sont ajoutés pour enrichir la pièce. Cela ajoute une complexité essentielle.

3 Progression d’Accords et Modulations (Couche 2)

La couche 2 définit un modèle formel de d’accords et de modulations dans un contexte tonal. Nous donnons un

	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII	♭II	6Δ
I	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
II	1	1		1	1			1		1	1	1				
III						1							1			
IV	1	1		1	1		1	1	1			1		1	1	1
V	1		1	1	1	1				1		1	1			
VI		1		1	1			1	1		1	1			1	1
VII	1		1							1						
Vda	A			1					B						C	
V/II		1									1					
V/III			1									1				
V/IV				1									1			
V/V					1				1							
V/VI						1				1						
V/VII							1				1					
♭II				1				1								
6Δ	G			1				1	H						I	

FIGURE 1 – Matrice de transition T entre les degrés successifs d’un accord dans une tonalité, utilisée pour définir les progressions d’accords. Un emplacement vide dans la matrice correspond à la valeur zéro.

aperçu des principaux concepts du modèle (le modèle complet est présenté dans [22]). Nous définissons une *progression* comme une séquence de degrés d’accords dans une tonalité donnée et une *modulation* comme une transition entre deux progressions successives de tonalités différentes. Le modèle définit les accords possibles (degré, qualité et état) et les transitions entre eux, conformément à la théorie de la musique tonale occidentale, basée sur le concept de *tonalité*¹. Tous les concepts utilisés dans le modèle sont des concepts standards de la théorie musicale tonale, pour lesquels de nombreuses références existent².

Théorie musicale Les règles implémentées dans le modèle sont tirées de [7] et [9], garantissant ainsi la cohérence avec le modèle des voix (Couche 3) [21]. Nous utilisons le chapitre de Duha sur les modulations ainsi que les chapitres 4 (triades et accords de septième), 6 (écriture des parties), 8-11 (harmonie diatonique), 13 (accords de dominante), 14 (accords prédominants), 16 (accord 6-4), 17 (troisième et sixième degré), 19 (accord de septième de sensible), 21 (accords de dominante secondaire), 29 (accord napolitain) et 30 (accord de sixte augmentée) de Gauldin. Le modèle a été élaboré en collaboration avec deux compositeurs.

Accord Un *accord* est un ensemble de trois notes ou plus, identifié de manière unique par un triplet (r, q, s) où r est la note fondamentale (l’une des douze notes de la musique occidentale), q est la qualité (qui définit les intervalles entre les notes de l’accord) et s est l’état (défini par la note la plus grave de l’accord). Dans une tonalité donnée, chaque note possède un degré d qui définit sa fonction au sein de la tonalité, ainsi que les qualités et états possibles pour construire des accords sur cette note comme fondamentale.

Matrice de transition d’accords Le cœur du modèle de la Couche 2 est la matrice de transition T , qui définit les transitions d’accords possibles entre deux degrés d’une tonalité (voir la Figure 1). Il s’agit d’une matrice constante qui encode les transitions permises par la théorie musicale tonale. La matrice T est complétée par trois autres matrices constantes M , P et L , qui définissent respectivement les re-

1. Une tonalité est définie comme une paire composée d’une clé (l’une des douze notes C, C#, D, jusqu’à B) et d’un mode (majeur ou mineur).

2. Quelques liens : [tonalité](#), [accord](#), [fonction](#), [dominante secondaire](#) et [accord de sixte augmentée](#), parmi beaucoup d’autres.

lations entre un degré d'accord et les qualités, états et notes fondamentales possibles de l'accord. Ces quatre matrices encodent de manière compacte une partie conséquente de la théorie musicale, ce qui, à notre connaissance, n'a été réalisé par aucun outil de composition précédent. À ces matrices, des contraintes sont ajoutées pour modéliser des aspects spécifiques de la théorie musicale tonale qui ne sont pas capturés par les matrices. De plus, des contraintes sont ajoutées pour modéliser les modulations.

La matrice de transition T se lit comme suit : l'accord *rang* peut être suivi de l'accord *colonne* si la valeur de la matrice est égale à 1. Par exemple, l'accord III peut être suivi du VI et du V/VI, mais pas de tout autre degré. La matrice T encode les règles générales de l'harmonie tonale, tirées de [7, 9]. Chaque bloc de la matrice contraint un aspect spécifique des progressions d'accords dans un contexte tonal :

- Le bloc **A (orange)** définit la succession d'accords possible entre les accords diatoniques³.
- Le bloc **B (bleu foncé)** définit les accords de dominante secondaire⁴ qui peuvent suivre les accords diatoniques.
- Le bloc **C (jaune)** définit quels accords chromatiques peuvent suivre les accords diatoniques.
- Les blocs **D (vert)** et **E (cyan)** définissent les accords diatoniques et les autres dominantes secondaires qui peuvent suivre les dominantes secondaires. Les dominantes secondaires doivent se résoudre vers un accord basé une quinte en dessous de leur fondamentale. Elles peuvent se résoudre soit vers l'accord diatonique correspondant (par exemple, II pour V/II), soit vers un autre accord de dominante basé sur cette même note (par exemple, V/V pour V/II).
- Le bloc **F (gris)** applique les règles des accords chromatiques suivant les accords de dominante secondaire. Ce n'est pas autorisé; cette partie de la matrice est donc vide.
- Les blocs **G (violet)** et **H (magenta)** appliquent les règles des accords chromatiques. La sixte napolitaine (bII) et la sixte augmentée (6 Δ) doivent aller à V, mais avant cela ils peuvent aller à l'appoggiature du cinquième degré (Vda).
- Le bloc **I (rouge)** impose des règles de succession d'accords chromatiques. Ce n'est pas autorisé; cette partie de la matrice est donc vide.

4 Les Voix (Couche 3)

La couche 3 définit un modèle formel pour l'enchaînement à quatre voix des accords, c'est à dire la traduction d'accords (exprimés en degrés) en ensembles de notes [21]. Le modèle est composé de deux parties : un modèle de base (pour les progressions d'accords simples) et une extension (pour les progressions multiples, afin de gérer le résultat de la couche 2). Le modèle de base définit les voix d'une progression simple avec des accords diatoniques dans une tex-

ture à quatre voix, une technique courante pour représenter l'harmonie. Le modèle de base comprend des triades diatoniques et des accords de septième de dominante, y compris leurs renversements. L'extension prend en charge la juxtaposition de progressions, les accords chromatiques dans les progressions, ainsi que les septièmes d'espèce et les modulations entre progressions.

Théorie musicale De [7] nous reprenons les chapitres sur les accords de 3 et 4 sons et les renversements. De [9] nous reprenons 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).

Concepts de base La texture à quatre voix définit quatre voix distinctes, appelées basse, ténor, alto et soprano, classées de la plus grave à la plus aiguë en référence à l'amplitude des voix humaines correspondantes. Le modèle de base définit 15 règles (contraintes) et 5 préférences (contraintes optionnelles avec coûts) pour les quatre voix. Les contraintes harmoniques s'appliquent aux notes d'un même accord, tandis que les contraintes mélodiques s'appliquent aux transitions entre accords consécutifs ainsi que les voix dominantes.

5 Évaluation

Nous présentons une évaluation de l'état actuel de l'outil en deux parties. Nous abordons d'abord le temps de calcul du solveur, puis nous expliquons comment un compositeur peut utiliser l'outil. Le code source est sur github⁵.

5.1 Efficacité et nombre de solutions

Le modèle étant conçu pour offrir une liberté maximale aux compositeurs, les seules contraintes appliquées par défaut sont celles nécessaires pour garantir que les séquences d'accords générées respectent les règles de l'harmonie tonale. Dans ce cas, le nombre de solutions pour un problème donné est énorme. De plus, ces solutions sont trouvées extrêmement rapidement, même pour des problèmes de taille significative; l'efficacité n'est donc pas encore notre priorité. Par exemple, l'enchaînement d'accord pour le morceau musical présenté dans la Section 5.2 a été généré en 3 ms sur un MacBook Pro M1. Un morceau plus long, comportant 60 accords et cinq modulations, a été généré en 20 ms. Un aspect essentiel de l'utilisation de l'outil réside dans l'ajout par le compositeur d'idées musicales, sous forme de contraintes, pour guider le solveur. Nous prévoyons que cela se fasse via une interface graphique, afin que le compositeur n'ait pas besoin d'écrire de code. L'ajout d'idées musicales entraînera une augmentation du temps de calcul. Le problème passera d'un problème de satisfaction (trouver une solution valide) à un problème d'optimisation (trouver la meilleure solution), les critères d'optimisation étant fournis par le compositeur. Par exemple, la Couche 1 pourrait ajouter des contraintes globales pour appliquer la structure AABA avec les thèmes A et B, avec des contraintes de similarité entre les A, qui sont des variations d'un même thème, et une contrainte de différence pour le thème B contrasté.

3. L'appoggiature double du cinquième degré (Vda) est traitée séparément car sa fonction musicale est complètement différente.

4. Une dominante secondaire est la dominante d'un degré diatonique.

5. <https://github.com/sprockelsd/Harmoniser>

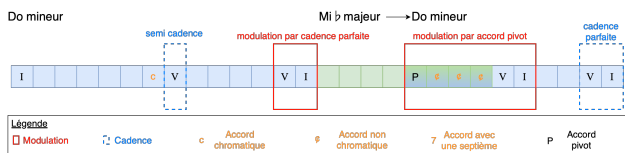


FIGURE 2 – Représentation de l'entrée au solveur.

5.2 L'utilisation par un compositeur

Nous nous mettons dans la peau d'un compositeur pour montrer comment utiliser l'outil. Nous expliquons l'exemple en termes de contraintes, mais dans un outil pratique, ces contraintes seraient fournies via une interface graphique. Pour notre exemple, nous souhaitons écrire une progression d'accords commençant en do mineur, modulant vers sa tonalité relative mi bémol majeur, puis revenant à la tonalité de do mineur, soit 28 accords. Nous souhaitons une cadence parfaite sur l'accord 12 pour terminer le premier enchaînement en do mineur et une modulation d'accord pivot de l'accord 18 à l'accord 23 pour y revenir.

En plus de ces instructions nécessaires, nous ajoutons les contraintes suivantes pour exprimer nos intentions musicales (voir la Figure 2). (1) Le premier accord doit être I. (2) Aucun accord ne peut être III ou VII, sauf pendant la modulation de l'accord pivot. (3) Seuls les accords de dominante peuvent avoir une septième. (4) Il y a une demi-cadence sur l'accord sept. (5) Il y a un accord chromatique juste avant (position six). (6) L'accord II ne doit être utilisé qu'en première inversion. (7) Il y a une cadence à la fin.

Si nous exécutons le solveur avec ces souhaits, il produit la sortie suivante. Pour la première progression en do mineur, les accords suggérés par le solveur sont *I-II-Vda-V-VI-IV-bII-V-I-V-I-IV-V-I*. Ceci est intéressant, mais selon nos goûts personnels, nous apportons quelques modifications : *I-II-Vda-V-I-IV-6Δ-V-I-V-I-Vda-V-I*. C'est également une solution acceptée par le solveur. Pour la progression de mi b majeur vers do mineur, le résultat est *I-VI-II-V-VI|I-V|VI|V-VI|I-VII|II-V-VI-I-II-V-I*, où *VI|I* est l'accord pivot démarquant la modulation (sixième degré en mi b majeur et premier degré en do mineur) et les accords suivants sont dans les deux tonalités jusqu'à la cadence en do mineur. Pour cette partie, nous n'apportons qu'une petite modification : nous modifions la cadence qui termine la modulation pour qu'elle soit interrompue. Ceci est dû au goût personnel.

Les accords finaux de cette partie sont donc *I-VI-II-V-VI|I-V|VI|V-VI|I-VII|II-V-VI-I-II-V-I*. Les états d'accords ont été omis dans la liste, mais ils sont tels qu'illustrés dans la Figure 3. Ensuite la Couche 3 génère une texture à quatre voix représentant notre morceau. La Figure 3 montre une texture à quatre voix possible. Elle peut être écoutée ici⁶. Un rythme harmonique a été donné aux accords, ainsi que quelques notes ornementales, par le compositeur. Nous prévoyons que le rythme et les notes ornementales soient fournis par l'outil dans une prochaine version.



FIGURE 3 – Pièce musicale basée sur la progression d'accords générée par la Couche 2. Les voix ont été ajoutées par la Couche 3 et le rythme par le compositeur.

6 Travaux Futurs

L'outil Harmoniser complet intégrera les modèles des quatre couches dans un seul système. Pour compléter les contraintes harmoniques, nous ajouterons des contraintes rythmiques aux quatre modèles. Nous travaillons sur une application intégrant un système de gestion des solutions afin que le compositeur puisse itérer sur différentes parties de la pièce musicale, conserver les bons résultats, ajouter ou supprimer des idées musicales, combiner des solutions partielles et écouter des solutions. Nous travaillons sur une interface utilisateur graphique permettant au compositeur d'utiliser l'outil sans aucune programmation. L'application sera fournie sous forme de plug-in pour une station de travail audionumérique (DAW) afin de pouvoir être intégrée à l'atelier du musicien. Nous faisons actuellement un plug-in expérimental utilisant la plateforme JUCE [19].

L'application Harmoniser sera évaluée par des compositeurs qui donneront leur avis sur son ergonomie. Cette évaluation s'inscrit dans le cadre de notre collaboration continue avec Karim Haddad de l'IRCAM et Adrien Tsilogianis du Conservatoire royal de Bruxelles. Nous espérons que l'application finale sera utilisable par un large public de compositeurs amateurs et servira de base à de futures extensions visant à l'adapter et à l'étendre.

7 Conclusion

Cet article présente l'état d'avancement du projet Harmoniser, qui vise à développer un outil de composition basé sur la programmation par contraintes. Ce projet définit un processus de composition en quatre couches. Nous avons développé la Couche 2 (progressions d'accords et modulations) et la Couche 3 (les voix). Les travaux en cours portent sur la création des Couches 1 et 4, l'ajout de contraintes rythmiques et la création d'un plug-in avec interface graphique pour une station de travail audionumérique.

Remerciements

Les auteurs souhaitent remercier le Dr. Karim Haddad de l'IRCAM et le Pr. Adrien Tsilogianis du Conservatoire royal de Bruxelles pour leurs conseils.

6. <https://soundcloud.com/anonymous123-414149293/ijcai-example>

Références

- [1] Torsten Anders. *Composing Music by Composing Rules : Design and Usage of a Generic Music Constraint System*. PhD thesis, Queen's University Belfast, 2008.
- [2] Torsten Anders, Christina Anagnostopoulou, and Michael Alcorn. Strasheela : Design and Usage of a Music Composition Environment Based on the Oz Programming Model. In *Multiparadigm Programming in Mozart/Oz : Second International Conference (MOZ 2004)*, pages 277–291. Springer, 2005.
- [3] Torsten Anders and Eduardo R Miranda. A Computational Model that Generalises Schoenberg's Guidelines for Favourable Chord Progressions. In *proceedings of the Sound and Music Computing Conference*, pages 48–52, 2009.
- [4] Alexandre Bonlarron and Jean-Charles Régin. Intertwining CP and NLP : The Generation of Unreasonably Constrained Sentences. In *Thirty-Third International Joint Conference on Artificial Intelligence {IJCAI-24}*, pages 7600–7608. International Joint Conferences on Artificial Intelligence Organization, 2024.
- [5] Grégoire Carpentier, Gérard Assayag, and Emmanuel Saint-James. Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach. *Journal of Heuristics*, 16 :681–714, 2010.
- [6] Stephen Davismoon and John Eccles. Combining Musical Constraints with Markov Transition Probabilities to Improve the Generation of Creative Musical Structures. In *European Conference on the Applications of Evolutionary Computation*, pages 361–370. Springer, 2010.
- [7] Isabelle Duha. *L'Harmonie en Liberté : de la Mémoire à l'Improvisation*. Gérard Billaudot, Armiane Imp., 2016.
- [8] Kemal Ebcioglu. An Expert System for Harmonizing Chorales in the Style of JS Bach. *The Journal of Logic Programming*, 8(1-2) :145–185, 1990.
- [9] Robert Gauldin. *Harmonic Practice in Tonal Music. Second Edition*. W. W. Norton and Company, Inc, 2004.
- [10] Gecode Team. *Gecode : Generic Constraint Development Environment*, 2019.
- [11] Luca Giuliani, Francesco Ballerini, Allegra De Filippo, and Andrea Borghesi. MusiComb : a Sample-based Approach to Music Generation Through Constraints. In *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 194–198, 2023.
- [12] Dorien Herremans and Kenneth Sörensen. Composing Fifth Species Counterpoint Music with a Variable Neighborhood Search Algorithm. *Expert systems with applications*, 40(16) :6427–6437, 2013.
- [13] Cheng Zhi Anna Huang and Elaine Chew. Palestrina Pal : a Grammar Checker for Music Compositions in the Style of Palestrina. In *Proceedings of the 5th Conference on Understanding and Creating Music*, 2005.
- [14] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Imposing Higher-level Structure in Polyphonic Music Generation using Convolutional Restricted Boltzmann Machines and Constraints. *Journal of Creative Music Systems*, 2 :[1]–31, 2018.
- [15] Mikael Laurson and Mika Kuuskankare. Extensible Constraint Syntax through Score Accessors. In *Journées d'Informatique Musicale*, 2005.
- [16] François Pachet and Pierre Roy. Musical Harmonization with Constraints : A Survey. *Constraints*, 6 :7–19, 2001.
- [17] François Pachet and Pierre Roy. Markov Constraints : Steerable Generation of Markov Sequences. *Constraints*, 16(2) :148–172, 2011.
- [18] Alexandre Papadopoulos, Pierre Roy, Jean-Charles Régin, and François Pachet. Generating all Possible Palindromes from n-gram Corpora. In *IJCAI 2015*, 2015.
- [19] Raw Material Software Limited. *JUCE Audio Application Development Framework*, 2025.
- [20] Örjan Sandred. PWMC, a Constraint-solving System for Generating Music Scores. *Computer Music Journal*, 34(2) :8–24, 2010.
- [21] Damien Sprockeels and Peter Van Roy. Expressing Musical Ideas with Constraint Programming Using a Model of Tonal Harmony. In *International Joint Conference on Artificial Intelligence*, 2024.
- [22] Damien Sprockeels and Peter Van Roy. Towards a Practical Tool for Music Composition : Using Constraint Programming to Model Chord Progressions and Modulations. In *International Joint Conference on Artificial Intelligence*, 2025.
- [23] Charlotte Truchet, Gérard Assayag, and Philippe Codognet. OMClouds, Petits Nuages de Contrainte dans OpenMusic. In *Journées d'Informatique Musicale*, 2003.