

Zeitschrift: Bulletin de la Société Fribourgeoise des Sciences Naturelles = Bulletin der Naturforschenden Gesellschaft Freiburg
Herausgeber: Société Fribourgeoise des Sciences Naturelles
Band: 111 (2022)

Artikel: Composition algorithmique et théorie musicale
Autor: Vaucher, Michael
DOI: <https://doi.org/10.5169/seals-1005840>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 28.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Composition algorithmique et théorie musicale

MICHAEL VAUCHER
Travail de Master de Mathématiques
sous la supervision du
Prof. Dr. CHRISTIAN MAZZA

Avec l'avancée de l'informatique et de l'intelligence artificielle durant le XXe siècle, les musiciens se sont petit à petit intéressés à produire de la musique par ordinateur. En commençant à utiliser des algorithmes stochastiques se basant sur des modèles markoviens, les programmeurs se sont ensuite tournés vers des systèmes apprenants, comme le machine learning. Toutes ces différentes implémentations ont toutefois un seul but commun : créer une œuvre de meilleure qualité possible, ressemblant au mieux aux compositions pré-existantes de musique savante. Pour ce faire, la plupart des algorithmes — ou du moins les plus performants — sont soumis à une phase d'apprentissage, durant laquelle ceux-ci s'ajustent et s'imprègnent de divers paramètres, issus d'un corpus de pièces musicales défini. Une fois le programme informatique entraîné, ses probabilités internes déterminées, celui-ci peut alors être exécuté afin de composer une partition dans le style d'écriture de son corpus d'apprentissage.

Il va de soi que plus le corpus est copieux et plus le nombre de données prises en compte est grand, plus l'algorithme sera finalement apte à générer une œuvre de qualité. En effet, la musique savante est d'une complexité sans limite, c'est pourquoi un grand nombre de paramètres considérés permet de composer des œuvres d'une meilleure richesse structurelle et musicale. A titre d'exemple, l'algorithme Bee-thovANN ([COL21]) prend en compte dix caractéristiques différentes de son dataset d'apprentissage, telles que la tonalité, le degré harmonique et la position rythmique de la note dans la mesure, afin de créer de la musique d'une qualité optimale. Ce programme a d'ailleurs généré une "dixième symphonie" de Beethoven en 2021, complétant de manière algorithmique les esquisses posthumes du compositeur.

Le travail présenté dans cet article a pour but d'améliorer la composition algorithmique, non pas en augmentant le nombre de données prises en compte, mais en considérant davantage la théorie musicale. L'algorithme utilisé pour cette étude, intitulé BeethoPar, permet effectivement au programmeur de sélectionner parmi cinq arguments initiaux, modifiant entre autres la lecture des partitions du corpus d'apprentissage et les probabilités de la génération des notes. En variant la sélection des arguments, lesquels ont tous un lien direct avec la théorie musicale, diverses partitions sont ainsi créées par le programme. Par conséquent, dans cette recherche, il s'agit de relever et comparer les qualités des productions, dans le but ultime de définir une initialisation optimale de l'algorithme et déterminer quels arguments doivent être privilégiés ou évités.

L'algorithme BeethoPar

L'algorithme BeethoPar a été implémenté par mes soins en 2022, dans le cadre de ma thèse de Master à l'université de Fribourg, laquelle a été réalisée sous la supervision du Prof. Dr. Christian Mazza. Ce programme, comme dit dans l'introduction, permet la génération de partitions musicales dépendamment d'une sélection parmi cinq arguments initiaux. Ce choix entraîne la composition d'œuvres de qualité modulante, sur lesquelles l'analyse finale de l'étude est basée. En effet, il s'agit finalement de déterminer la richesse et l'impact qu'ont chacun de ces arguments sur la génération. Plus en détails, le processus de BeethoPar consiste à supprimer la voix du premier violon d'un quatuor à corde de Beethoven choisi, puis à la régénérer algorithmiquement. De cette manière, la partition produite par algorithme pourra typiquement être comparée à la pièce originale du compositeur allemand. Afin de permettre une qualité de composition optimale, l'algorithme BeethoPar est préalablement entraîné avec le corpus des 16 quatuors à corde de Beethoven, composés entre 1800 et 1826. Ce dataset, comprenant plus de 240'000 notes, a été annoté et numérisé en 2018 par NEUWIRTH et al. dans le cadre du laboratoire de musicologie numérique de l'EPFL (Lausanne, Suisse).

L'architecture de l'algorithme

L'algorithme BeethoPar se base sur une architecture de matrices stochastiques. Par conséquent, la séquence de N notes à générer $seq = (note[1], \dots, note[N])$ peut être vue comme un processus markovien, puisque chaque note successive est uniquement déterminée d'après des caractéristiques présentes à ce moment-là dans la partition, sans considérer des états antérieurs. En effet,

puisque'il s'agit uniquement de comparer les partitions résultantes entre elles et avec la voix originale, une simple génération par un processus stochastique est considérablement suffisante.

Dans l'algorithme BeethoPar, les valeurs possibles pour $note[i]$ sont les nombres entiers entre 0 (*do*) et 11 (*si*). En effet, il a été choisi de travailler uniquement avec la valeur des notes, à octave près. De cette manière, les matrices stochastiques sont restreintes à une dimension 12 et ainsi leurs 144 valeurs sont alors davantage précisément ajustées d'après le corpus d'apprentissage. Cette étude ne s'intéresse cependant pas au rythme des notes. Effectivement, par les matrices, seules les hauteurs des notes seront générées. Le rythme associé aux notes générées est gardé depuis la partition originale de Beethoven. Cela implique grossièrement que la génération de BeethoPar s'applique sur chaque note préexistante, en modifiant sa hauteur, d'après les probabilités des matrices stochastiques. Les partitions résultantes seront ainsi toutes un minimum semblables et leur comparaison sera davantage aisée. De plus, les valeurs rythmiques ne déterminent pas vraiment la qualité d'une partition, alors que la hauteur des notes et les lignes mélodiques sont primordiales pour cela.

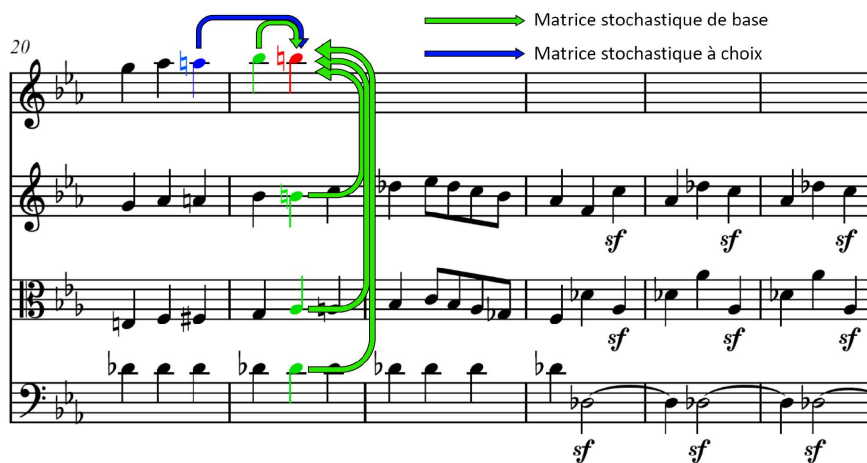


FIGURE 6.1 – *Schéma des implications matricielles de BeethoPar.* La note en rouge est générée avec trois matrices provenant des autres instruments et une matrice travaillant sur la voix du premier violon. La matrice en bleu, basée sur l'avant-dernière note de la portée supérieure, est à choix par le troisième argument de l'algorithme.

L'algorithme, pour définir chaque note consécutive utilise les probabilités de 4 à 5 matrices stochastiques. Dans un quatuor à cordes, trois autres instruments jouent en polyphonie avec le premier violon, lesquels peuvent également être utilisés en tant que paramètres pour la génération. Par conséquent, chaque note composée par BeethoPar dépend directement de la note précédente et des

notes jouées par le deuxième violon, l'alto et le violoncelle au même moment. La figure 6.1 montre les implications matricielles implémentées dans Beetho-Par. Lorsque l'algorithme entreprend de générer la note en rouge, il pioche dans les probabilités des quatre matrices, colorées en vert sur l'image. Une cinquième, indiquée en bleu, est prise en compte si le troisième argument est enclenché. Les probabilités de chaque matrice sont alors multipliées, afin de créer le vecteur stochastique de dimension 12, harmonisé pour la génération de la note en rouge, d'après l'entier des paramètres présents sur la partition à ce moment précis.

Mathématiquement parlant, en se basant sur la figure 6.1, définissons $(V1_{i,j})$ la matrice traitant de la note du premier violon précédant la note à générer (flèche verte en haut du schéma). Puis, soient $(V2_{i,j})$, $(A_{i,j})$, $(C_{i,j})$ les matrices des autres instruments, respectivement du deuxième violon, de l'alto et du violoncelle, lesquelles déterminent verticalement les probabilités pour générer la note en rouge (les flèches vertes parallèles sur le schéma). Le premier violon joue un *si bémol* avant la note en rouge, qui est la dixième note de la gamme chromatique. C'est donc le vecteur $(V1_{10,j})_{0 \leq j \leq 11}$ qui sera opportun à utiliser. Celui-ci procure effectivement des probabilités de sélection pour une note successive à un *si bémol* justement. De la matrice du deuxième violon, lequel joue au moment de la note à générer un *si bécarré* (onzième note de la gamme), on retire la ligne $(V2_{11,j})_{0 \leq j \leq 11}$. Puis, de manière analogue, on extrait les vecteurs $(A_{8,j})_{0 \leq j \leq 11}$ et $(C_{1,j})_{0 \leq j \leq 11}$, puisque les deux derniers instruments jouent un *la bémol* et un *ré bémol*, respectivement huitième et première note de la gamme chromatique. Une fois que les quatre vecteurs stochastiques sont issus des matrices, en se conformant aux notes jouées par les différentes voix sur la partition, il suffit de multiplier leurs éléments afin de créer le vecteur optimal pour la note à générer. Ainsi, la probabilité conditionnelle que la note rouge prenne la valeur i ($i \in \{0, \dots, 11\}$), en connaissant les valeurs de la partition et sans enclencher le troisième argument, est donnée par :

$$\begin{aligned} \mathbb{P}(\text{note}V1[n+1] = i \mid \text{note}V1[n] = \text{si bémol}, \\ \text{note}V2_{n+1} = \text{si bécarré}, \\ \text{note}A_{n+1} = \text{la bémol}, \\ \text{note}C_{n+1} = \text{ré bémol}) \\ = V1_{10,i} \times V2_{11,i} \times A_{8,i} \times C_{1,i}, \end{aligned}$$

où $\text{note}V1[k]$ désigne le k -ième élément de la séquence des notes du premier violon. Plus précisément, $\text{note}V1[n+1]$ est la note en rouge et $\text{note}V1[n]$ sa précédente. Les valeurs de $\text{note}V2_{n+1}$, $\text{note}A_{n+1}$ et $\text{note}C_{n+1}$ sont les notes entendues au même moment que $\text{note}V1[n+1]$ au deuxième violon, à l'alto

et au violoncelle respectivement.

Les probabilités contenues dans les matrices sont déterminées d'après le corpus d'entraînement de l'algorithme. Plus précisément, dans le but de définir une valeur en particulier, BeethoPar commence par compter le nombre d'occurrences dans le corpus de la transition définie par le coefficient en question. Puis, après l'énumération de toutes les successions, il suffit de normaliser chaque ligne de la matrice, afin que celle-ci devienne stochastique. Par exemple, afin de connaître la probabilité de composer un *mi* après un *fa* dans la voix de premier violon, l'algorithme compte le nombre de fois que cet enchaînement apparaît dans le corpus, par rapport au total des successions partant de *fa*. La figure 6.2 illustre l'énumération des transitions d'une partition de quatre notes. Typiquement, la flèche verte du schéma relate la suite des notes *do* et *mi*, qui sont les éléments 0 et 4 de la gamme chromatique. Ainsi, l'élément $a_{0,4}$ (cercle vert) de la matrice augmente sa valeur de 1. Il faut noter toutefois que la matrice de la figure 6.2 n'est pas stochastique, puisque la somme des éléments de sa ligne initiale ne fait pas 1. Il est donc nécessaire de la normaliser, avant de pouvoir l'appliquer à la génération.

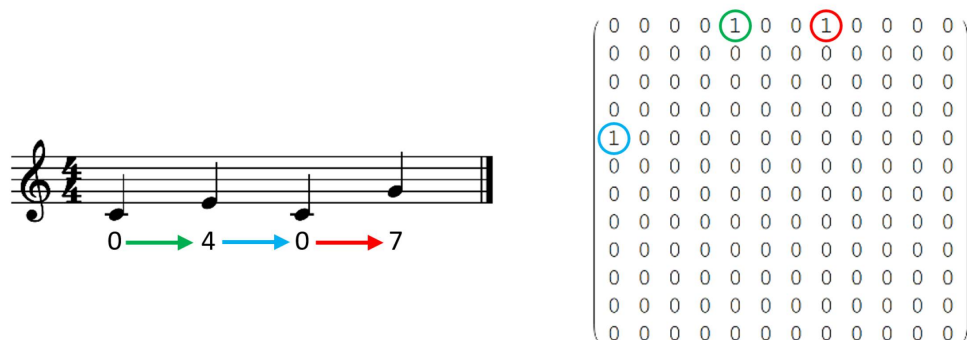


FIGURE 6.2 – *Fichier musical de 4 notes, avec la matrice d'énumération du premier violon associée. Les numéros reliés aux notes désignent leur rang dans la gamme chromatique et donc leur position dans la matrice. Chaque flèche, interprétant la transition d'une note à une autre, est illustrée dans la matrice par le cercle de même couleur.*

La création des matrices concernant les autres instruments est similaire, mais pas identique. Alors que la matrice du premier violon traite des notes successives horizontalement (comme sur la figure 6.2), les autres portent sur une implication verticale. En effet, ces tuples doivent pouvoir transmettre à l'algorithme les probabilités qu'une note particulière puisse être générée en parallèle de l'instrument en question. Pour compter le nombre d'occurrences, BeethoPar, pour chaque note jouée par le premier violon, relève les notes jouées par l'autre instrument au même moment. Par exemple, si dans une pièce

du corpus l'alto interprète un *mi* (quatrième note de la gamme chromatique) lorsque le premier violon joue un *sol* (septième note), alors l'élément $a_{4,7}$ de la matrice de l'alto augmente de 1. De cette manière, la probabilité de générer dans la phase suivante un *sol*, lorsque l'alto joue un *mi*, augmente. La figure 6.3 illustre par un schéma la création de cette matrice pour une musique polyphonique composée de 4 notes. Il faut également remarquer que la matrice de la figure n'est pas stochastique, puisqu'elle n'est pas normalisée.

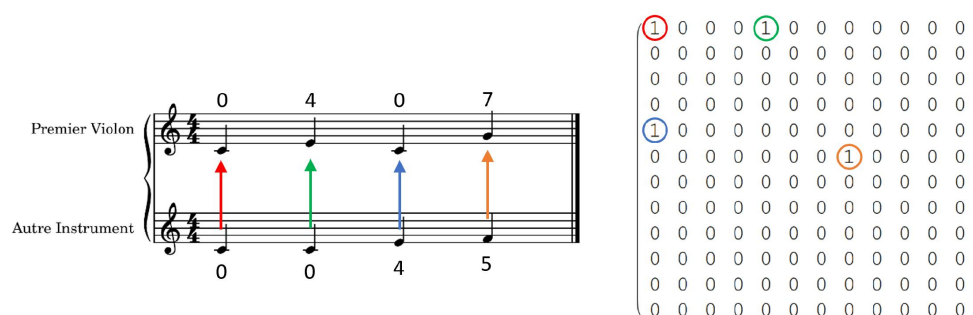


FIGURE 6.3 — *Fichier musical de 4 notes, avec la matrice d'énumération des autres instruments associée. Les numéros sous et sur les notes désignent leur rang dans la gamme chromatique et donc leur position dans la matrice. Chaque flèche, interprétant l'implication verticale des autres instruments sur la voix du premier violon, est illustrée dans la matrice par le cercle de même couleur.*

Les arguments à choix

Ce sous-chapitre détaille les cinq arguments, pouvant être sélectionnés dans l'algorithme BeethoPar. Ils sont expliqués par ordre d'apparition dans le code. En effet, les trois premiers agissent directement sur la création des matrices de probabilité, puis les deux derniers influencent la génération des notes.

1 — Transposition du corpus d'apprentissage

Beethoven a composé 16 quatuors à corde. Parmi eux, trois sont en tonalité¹ de *fa majeur*, deux sont en *mi bémol majeur*, deux sont en *si bémol majeur* et les restants sont en d'autres tonalités distinctes. Suivant si son mode est majeur ou mineur, chacune de ces tonalités est agencée selon une échelle diatonique², séparant les notes de la gamme par des intervalles précis. Toutefois, cette succession précise de notes permet surtout de hiérarchiser les sept notes d'une gamme en degrés, lesquels possèdent des fonctions différentes. Par exemple, le premier degré d'une gamme a tendance à soutenir la tonalité et la septième a le rôle de mélodiquement anticiper la tonique³.

Il faut cependant noter que les degrés sont tous différents suivant la tonalité choisie, puisque l'ordre dépend directement de la tonique. En *fa majeur* typiquement, le septième degré (*mi*) incitera, par sa fonction, l'apparition consécutive de la tonique (*fa*). Dans cette tonalité, la probabilité de rencontrer un *fa* après un *mi* est alors relativement élevée, puisqu'il s'agit d'un corollaire naturel du septième degré. Par contre, en tonalité de *do majeur*, le *mi*, qui est le troisième degré de la gamme, n'aura pas de tendance particulière à introduire un *fa* (voir figure 6.4). Par conséquent, les probabilités de générer une note particulière reposent considérablement sur les tonalités du corpus d'apprentissage, puisque les degrés — et donc leurs fonctions — en dépendent.

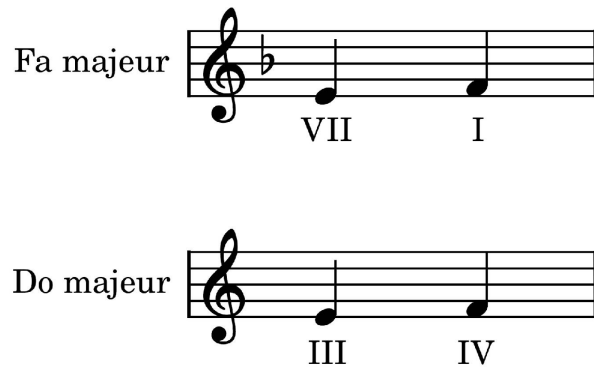


FIGURE 6.4 – *Succession des notes mi - fa en tonalités de do majeur et de fa majeur, avec leurs degrés dans les tonalités respectives.*

1. Tonalité : gamme principale d'une pièce musicale, donnée habituellement par sa note fondamentale (ou tonique) et son mode (majeur ou mineur). Le *fa majeur* est typiquement une tonalité de fondamentale *fa* et de mode majeur.

2. Echelle diatonique (aussi appelée gamme diatonique) : hiérarchie de 7 degrés utilisée communément dans la musique savante occidentale. Dans la musique tonale, on en emploie deux : une majeure et une mineure.

3. Tonique : premier degré d'une tonalité, donc de la gamme diatonique associée.

Le premier argument de BeethoPar permet de remédier à ces ambiguïtés. Alors qu'un algorithme de génération conventionnel s'ajuste d'après un corpus, sans tenir compte des différentes tonalités et des degrés, BeethoPar parvient, si on enclenche l'argument, à mettre l'entier des pièces du corpus d'entraînement sur une hiérarchie égale, en transposant toutes les partitions en *do*. De cette manière, les degrés possèdent tous la même note de base et chaque partition du corpus possède une gamme hiérarchique structurée identiquement.

2 — Restriction du corpus d'apprentissage par mode mineur ou majeur

La musique classique tonale connaît en général deux gammes diatoniques distinctes : une majeure et une mineure. La tonalité d'une pièce est alors donnée, la plupart du temps, par sa note fondamentale et son mode, majeur ou mineur, lequel détermine quelle échelle employer. Par exemple, le premier quatuor à cordes de Beethoven est en tonalité de *fa majeur*, ce qui indique qu'à partir d'un *fa*, les notes suivent majoritairement une gamme diatonique majeure.

Il faut savoir que les notes des gammes mineures et majeures ne sont pas identiques, leurs degrés ne se rapportent pas aux mêmes notes. En effet, les intervalles entre les deux échelles ne sont pas analogues, ce qui implique une grande différence dans l'usage de certaines notes dans une musique. Par exemple, en *do majeur*, il est commun de relever un *mi bécarré* dans la partition, troisième degré de la gamme, alors qu'en *do mineur*, cette note sera pratiquement inobservable puisqu'elle n'appartient pas à l'échelle diatonique mineure. Ainsi, ces distinctions entre les deux modes impliquent d'énormes changements dans les occurrences et donc dans les probabilités, puisque la composition d'une œuvre musicale dépend considérablement de la gamme choisie.

Par conséquent, à l'identique du premier argument, qui transpose l'entier du corpus d'apprentissage afin de n'avoir qu'une tonique commune, ce deuxième a pour but de ne travailler qu'avec un unique mode. De cette manière, l'algorithme ne travaille qu'avec une des deux échelles diatoniques et les degrés correspondent aux mêmes notes dans la hiérarchie. Alors que le premier argument restreint les gammes à ne partir que depuis le *do*, ce second limite le mode à être soit majeur, soit mineur.

Toutefois, il est important de relever qu'il n'est pas possible de modifier le mode d'une pièce. Le deuxième argument de l'algorithme n'a alors pas d'autre choix que de réduire le corpus d'apprentissage aux quatuors de mode voulu. D'après le mode de la pièce sélectionnée pour la génération du premier violon, l'entier du corpus sera initialement filtré. Typiquement, si le mouvement du quatuor à générer est en majeur, seules des œuvres du même mode sont

choisies pour entraîner les matrices stochastiques. De cette manière, la voix à créer est de mode identique au corpus d'entraînement restreint, ce qui permet de renforcer les degrés propres à la gamme diatonique du mode.

3 — Quatre ou cinq matrices stochastiques

Comme illustré par la figure 6.1, l'algorithme BeethoPar donne la possibilité, par ce troisième argument, d'ajouter une cinquième matrice stochastique au processus de génération. Au lieu de ne considérer que la dernière note générée de la voix du premier violon, BeethoPar prend en compte, avec l'argument enclenché, également la pénultième. Dans ce cas, l'algorithme crée une matrice stochastique supplémentaire caractérisant cet enchaînement (flèche bleue sur la figure 6.1). Pour reprendre l'exemple de la figure 6.1, la note pénultième de la voix du premier violon, avant la rouge, est un *la bécarré*, neuvième note de la gamme chromatique. Avec cette information supplémentaire et la matrice stochastique ($VB_{i,j}$) de l'argument, la probabilité conditionnelle que la note rouge prenne la valeur i lors de sa génération revient à :

$$\begin{aligned} \mathbb{P}(\text{noteV1}[n+1] = i \mid \text{noteV1}[n] = \text{si bémol}, \\ \text{noteV1}[n-1] = \text{la bécarré}, \\ \text{noteV2}_{n+1} = \text{si bécarré}, \\ \text{noteA}_{n+1} = \text{la bémol}, \\ \text{noteC}_{n+1} = \text{ré bémol}) \\ = V1_{10,i} \times VB_{9,i} \times V2_{11,i} \times A_{8,i} \times C_{1,i}. \end{aligned}$$

La matrice de cet argument est construite avec le corpus d'apprentissage, de manière analogue à celle agissant sur la dernière note générée (c.f. figure 6.2). Toutefois, pour cette nouvelle matrice, ce ne sont pas des enchaînements consécutifs de notes qui sont pris en compte, mais les successions à deux notes d'écart.

4 — Élément maximal ou probabilité pondérée

Pour chaque note à générer, un vecteur de probabilités est créé par l'algorithme BeethoPar, issu de toutes les matrices stochastiques d'après les valeurs présentes sur la partition. Ce 12-tuple comprend toutes les possibilités pour la note à générer. Typiquement, si on reprend l'exemple de la figure 6.1, le vecteur stochastique associé contient les probabilités de sélectionner les valeurs de 0 à 11 pour la génération de la note en rouge. Noté ici $V_{\text{note_rouge}}$, le tuple est donné par la multiplication suivante (sans enclencher le troisième argument) :

$$V_{\text{note_rouge}} = (V1_{10,i} \times V2_{11,i} \times A_{8,i} \times C_{1,i})_{0 \leq i \leq 11}$$

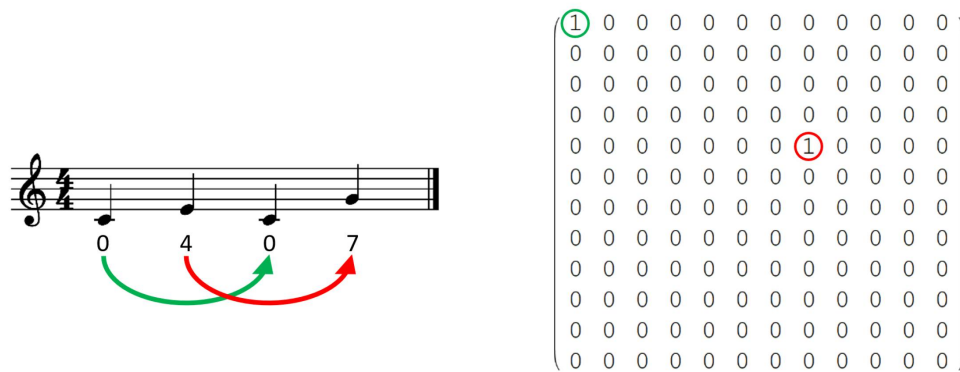


FIGURE 6.5 – *Fichier musical de 4 notes, avec la matrice d'énumération de l'argument associée. Sur les 4 notes de la portée, seules deux successions sont observées par l'argument. Elles sont illustrées par les flèches et représentées dans la matrice par les valeurs entourées d'un cercle de même couleur.*

Numériquement, pour ce cas précis, celui-ci équivaut au vecteur normalisé et arrondi suivant :

$$V_{note_rouge} = [0.032, 0.041, 0.036, 0.025, 0.112, 0.100, 0.003, 0.044, 0.407, 0.023, 0.018, 0.160]$$

C'est-à-dire que, d'après les valeurs des matrices, BeethoPar indique qu'il y a une probabilité de 0.1 de générer un *fa* pour la note en rouge, puisque c'est la cinquième note de la gamme chromatique et 0.1 est l'élément du vecteur V_{note_rouge} avec index 5. Toutefois, d'autres valeurs du tuple sont plus élevées, quelle valeur faut-il donc sélectionner ? Ce quatrième argument propose deux possibilités.

La première possibilité, qui est le procédé par défaut, est de simplement choisir l'élément du vecteur stochastique avec la plus grande valeur. De cette manière, la plus haute probabilité est sélectionnée pour la note à générer et ainsi, la partition résultante sera toujours de qualité harmonique maximale. Toutefois, choisir cette possibilité dans l'argument implique que les autres probabilités ne sont pas du tout prises en compte et, typiquement, lorsque les données de la partition sont plusieurs fois semblables, la note générée sera également successivement identique. Avec cette alternative, la voix composée par BeethoPar ne contient pas de fantaisie, mais décide toujours de la solution idéale. Un réseau de neurones conventionnel fonctionne de cette manière.

L'alternative proposée par le quatrième argument permet à l'algorithme de générer une partition avec davantage d'excentricité. Au lieu de sélectionner l'argument maximal, BeethoPar choisit un élément du vecteur stochastique, selon une loi de probabilité pondérée par les valeurs du tuple. Cela permet de générer des notes possédant une probabilité inférieure à l'extremum du vecteur et ainsi composer une musique avec quelques libertés supplémentaires.

Comme la sélection est pondérée, la probabilité de choisir un élément particulier est ainsi proportionnelle à sa valeur. L'hypothèse d'opter pour l'argument culminant est donc très probable, mais toutefois pas automatique.

5 — Poids du premier violon par rapport aux autres instruments

Comme déjà expliqué dans les sections ci-dessus, le vecteur stochastique associé à une note générée est issu de la multiplication de probabilités, lesquelles sont originaires des différentes matrices de Markov. Il faut toutefois relever que, dans ce calcul, les probabilités du premier violon sont de même poids que celles des autres instruments. En plus, il y a bien trois valeurs — $V_{211,i}$, $A_{8,i}$ et $C_{1,i}$ — qui font face au seul facteur $V_{110,i}$ provenant de la voix de ce violon. Par conséquent, ce dernier élément est quelque peu noyé par l'influence et le poids des trois instruments inférieurs, même si c'est bien la voix du premier violon qui est générée par l'algorithme BeethoPar.

Le dernier argument de l'algorithme essaie de remédier à ce problème en modifiant le poids des probabilités issues de la matrice du premier violon. Cet argument, qui prend comme valeur un nombre réel non négatif, permet ainsi d'ajuster l'incidence des différentes valeurs, afin de générer une note avec le compromis idéal entre les implications des notes précédentes et celles jouées en parallèle par les autres instruments. Spécifiquement, l'argument agit sur le vecteur stochastique en ajoutant une puissance aux probabilités voulues. Ainsi, en insérant le poids $j \in \mathbb{R}_{\geq 0}$ dans l'argument, le calcul du tuple pour la note en rouge de la figure 6.1 revient à :

$$V_{note_rouge} = ((V_{110,i})^j \times V_{211,i} \times A_{8,i} \times C_{1,i})_{0 \leq i \leq 11}.$$

Il faut donc remarquer que si $j = 0$, la note sera entièrement générée par les probabilités issues des autres instruments, ce qui peut être moins fascinant d'analyser. Avec une valeur entre 0 et 1 ($0 < j < 1$), le poids de la matrice du premier violon est diminuée par l'argument, ce qui accentue les probabilités provenant des trois voix inférieures. En revanche, en donnant une valeur supérieure à 1 ($j > 1$), la matrice (V_1) prend de l'envergure par rapport aux autres.

Résultats

Il s'agit maintenant de déterminer quelle sélection d'arguments est optimale, afin de générer une partition de qualité idéale. Bien que BeethoPar n'ait pas l'architecture élaborée d'un réseau de neurones, son but principal est de comparer, interpréter, analyser les compositions résultantes et décider s'il existe un ou plusieurs arguments permettant d'agréments la richesse des générations.

Dans le but de décrypter les résultats de l'algorithme BeethoPar, ce travail utilise trois techniques distinctes. La première consiste à comparer la partition générée avec la pièce originale, composée par Beethoven. On peut en effet partir du principe que l'écriture du compositeur contient les notes optimales. Par conséquent, cette première donnée de comparaison calcule le pourcentage de notes, issues de la génération, identiques à la partition de Beethoven, afin de connaître la proportion de notes choisies par BeethoPar étant incontestablement bonnes. Le second procédé consiste à davantage examiner les notes présentes sur la partition générée. En usant de techniques d'analyse, les mouvements musicaux et les harmonies seront relevées et évaluées. Finalement, une lecture de la partition générée sera écoutée grâce au logiciel MuseScore, afin d'interpréter les notes également grâce à l'oreille humaine. Les quatuors de Beethoven ont effectivement été composés dans le but d'être entendus par un public, il est donc aussi nécessaire de déterminer si les générations de BeethoPar sont autant agréables à écouter.

Ce chapitre procède à l'interprétation des arguments individuellement en commençant par le cinquième. En effet, les derniers arguments de l'algorithme ne possèdent pas — ou peu — d'incidence sur les premiers, puisque ceux-ci se trouvent en position postérieure dans le code de BeethoPar. Par conséquent, dans le but de trouver la sélection idéale, il est nécessaire de d'abord ajuster le poids du premier violon sans enclencher d'autre argument, puis de décider de la méthode pour choisir l'élément du vecteur stochastique, etc. De cette manière, il sera plus facile de discerner la capacité des premiers arguments, puisque les autres seront déjà ajustés de manière optimale lors de leur interprétation.

Afin de ne pas perdre de temps et de créer une analyse efficace, il a été choisi, pour ce chapitre, de se concentrer essentiellement sur le troisième mouvement du quatrième quatuor de Beethoven (fichier `op18_no4_mov3.mx1`). Ainsi, les arguments seront testés sur une pièce en particulier, afin de garder la comparaison possible et cohérente durant tout le chapitre. Le quatuor en question est idéal pour une telle analyse, puisqu'il contient une polyphonie riche et donc la génération de chaque note dépendra de beaucoup de paramètres issus de la partition originale. Toutes les partitions interprétées et analysées dans ce chapitre sont téléchargeables depuis le site <https://>

[//github.com/MichaelVaucher/BeethoPar](https://github.com/MichaelVaucher/BeethoPar).

Cinquième argument — Le poids idéal du premier violon

Il faut se rappeler que le cinquième argument peut prendre n'importe quelle valeur réelle non négative. Il n'est donc pas possible ici de simplement faire une comparaison entre deux partitions, afin de découvrir si BeethoPar produit de meilleurs résultats en enclenchant ou non l'argument. Il est nécessaire de tester plusieurs valeurs et trouver l'intervalle optimal pour la variable en question. À cette fin, 40 valeurs dans un intervalle de 0 et 2 (donc à incrément de 0.05) ont été insérées dans le cinquième argument de BeethoPar, afin de découvrir le pourcentage de notes correspondantes à la partition originale de Beethoven. Le graphe affiché par la figure 6.6 montre les valeurs résultantes, lesquelles sont calculées en insérant la partition du troisième mouvement du quatrième quatuor. Cela relève typiquement qu'avec un poids de 0.3 ou 1.45, la génération de BeethoPar possède une proportion supérieure à 22% de notes identiques par rapport à la composition de Beethoven. Ce qui est également intéressant de relever est que les valeurs contenues dans l'intervalle de 0.5 à 1.2 sont clairement à éviter (c.f. figure 6.6). Toutefois, le graphe peut être différent suivant la pièce choisie à générer avec BeethoPar. La figure 6.7 affiche alors la moyenne des valeurs issues de quatre quatuors différents. Il faut remarquer que ce graphe a le même comportement que le précédent ; il est donc récurrent d'avoir de bons résultats avec des valeurs autour de 0.3 (pic de gauche) et dans les environs de 1.5 (pic de droite). Ce sont les deux valeurs qui seront testées dans le but d'interpréter la qualité d'écoute des générations de BeethoPar pour le cinquième argument.

Grâce aux graphes des figures 6.6 et 6.7, deux valeurs pour l'argument ont ainsi été choisies : 0.3 et 1.5. Ce sont deux nombres tout à fait antipodiques, puisqu'un d'entre eux augmente le poids du premier violon (1.5), alors que l'autre l'abaisse fortement (0.3). Il s'agit maintenant de savoir laquelle des deux permet une meilleure écoute. Deux partitions, une avec une valeur de 1.5 et une autre avec un poids de 0.3, ont été générées par BeethoPar et sont présentées dans la figure 6.8. Les mouvements conjoints avec la composition originale de Beethoven (portée du milieu) ont été relevés avec les flèches montantes et descendantes. Ces montées et descentes sont effectivement importantes, puisqu'elles définissent le relief d'une pièce, des cohésions entre les notes, ce sont des enchaînements qui créent une écoute davantage confortable.

Plus en détails, il est facile de remarquer que dans la pièce avec un poids de 1.5 (portée supérieure sur la figure 6.8), il n'y a que très peu de mouvements et donc l'écoute de cette partition peut s'avérer pénible. Notez typiquement que les cinq premières notes de la portée sont des *sol*, puis les cinq suivantes sont des *do*, les mouvements sont bien horizontaux (flèches oranges). Cela montre alors beaucoup de répétitions, peu de délicatesse, une partition non

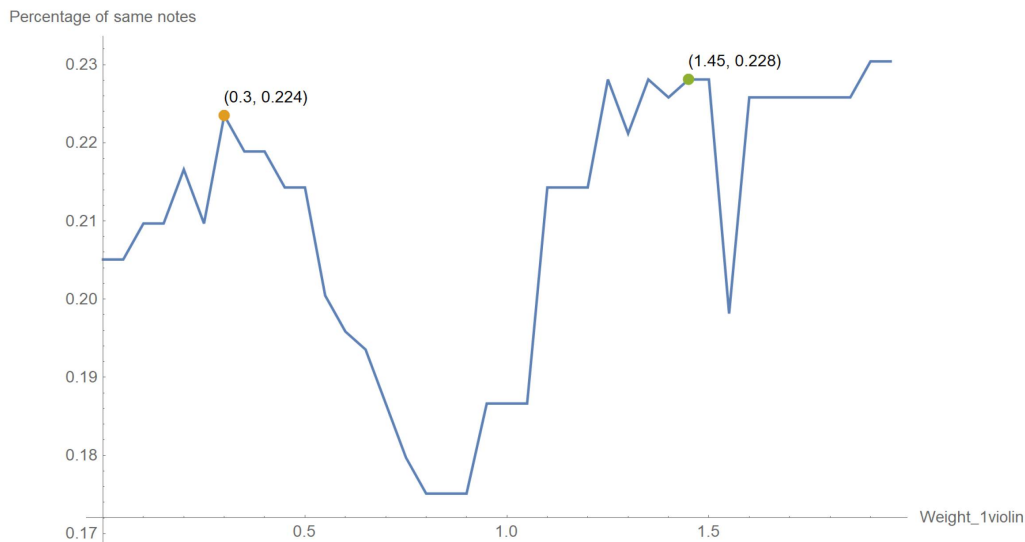


FIGURE 6.6 – *Graphe des proportions de notes identiques en fonction du poids donné au premier violon par le cinquième argument. Ce graphe est construit d'après le troisième mouvement du quatrième quatuor de Beethoven (fichier `op18_no4_mov3.mxl`).*

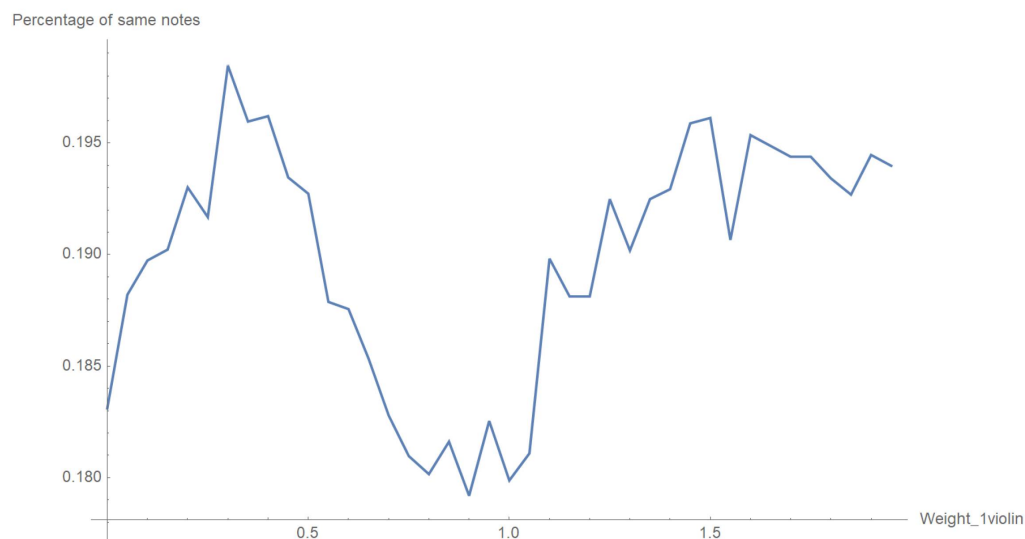


FIGURE 6.7 – *Graphe des proportions de notes identiques, issues de quatre quatuors, en fonction du poids donné au premier violon par le cinquième argument. Ce graphe est construit en prenant la moyenne de quatre quatuors différents : le no1 (mouvement 1), le no3 (mouvement 3), le no4 (mouvement 3) et le no8 (mouvement 1).*

sophistiquée. L'explication à cela est sans doute issue d'un poids trop élevé. En effet, les matrices du premier violon, qui ont été entraînées d'après le corpus, privilégient des successions de notes identiques, puisque les coefficients

de la diagonale sont dominants par rapport aux autres. Alors que les autres instruments du quatuor jouent des mouvements conjoints de note, ils n'auront toutefois pas assez de poids pour surmonter les probabilités issues du premier violon. Bien que la partition possède un bon pourcentage de notes identiques à la pièce originale (c.f. figure 6.6), ce ne sont pas des notes aptes à produire une écoute agréable.

Au contraire, la génération avec un poids de 0.3 est plutôt satisfaisante. En effet, celle-ci possède des mouvements similaires à la partition de Beethoven (flèches vertes sur la figure 6.6), ses notes sont variées, recherchées, son écoute est plutôt plaisante. C'est alors une génération de nettement meilleure qualité que celle avec un argument de 1.5. Bien que le pourcentage de notes correspondantes à la pièce de Beethoven est plutôt faible (22.4% seulement), BeethoPar a ici généré une musique de qualité (presque) similaire. Ce taux plutôt bas est typiquement dû au fait que l'algorithme a choisi de composer un mouvement montant (première flèche verte de la troisième portée de la figure 6.8) à intervalle de tierce par rapport à la partition originale, ce qui est très consonant et harmonieux, mais ce ne sont néanmoins pas des notes similaires à la portée centrale.



FIGURE 6.8 – *Comparaison des générations avec poids de 0.3 et 1.5. Deux portées produites par BeethoPar, générées d'après le quatuor no4 (mouvement 3), encadrent la partition originale au milieu, composée par Beethoven.*

L'étude des résultats concernant le dernier argument a permis de montrer qu'ajouter du poids aux matrices stochastiques, construites sur la voix à générer, n'est pas un avantage pour la composition de musique par algorithme. En d'autres termes, les instruments en polyphonie doivent posséder davantage d'incidence sur la portée. C'est effectivement eux qui soutiennent et définissent les accords, il est donc important que les notes du premier violon suivent ce contexte harmonique. Avec un poids pour cet instrument défini à 0.3, l'algorithme BeethoPar est calibré efficacement, afin de générer la meilleure partition possible. Cette valeur est donc gardée comme valeur pour les exécutions suivantes, dans l'analyse des autres arguments.

Quatrième argument — Pondération ou sélection automatique

Le quatrième argument de l'algorithme BeethoPar permet de choisir le processus de sélection de chaque note consécutive, d'après les probabilités du vecteur stochastique. Seules deux alternatives sont acceptables par l'algorithme : sélection par pondération ou élément maximal. L'analyse de cet argument est donc plus aisée que pour le précédent, puisque celui-ci n'implique que deux résultats différents.



FIGURE 6.9 – *Comparaison des générations avec et sans quatrième argument.* Deux portées produites par BeethoPar, générées d'après le quatuor no 4 (mouvement 3), encadrent la partition originale au milieu, composée par Beethoven.

En choisissant la pondération, la partition sera naturellement plus fantaisiste, correspondant ainsi moins à la pièce originale de Beethoven. D'ailleurs, en optant pour cet argument, la partition générée n'est qu'à environ 15% similaire à la voix initiale, contre 22% en sélectionnant l'élément maximal du vecteur (cf. argument précédent). Cette valeur peut même descendre jusqu'à 13% si l'algorithme sélectionne par pondération des notes particulièrement singulières. La qualité des notes générées baisse alors nettement en enclenchant ce quatrième argument. Toutefois, ces valeurs étaient prévisibles par le fait de choisir les notes suivant une probabilité pondérée et créer une partition davantage fantaisiste. Il s'agit alors d'analyser sa qualité d'écriture et son écoute.

La figure 6.9 affiche et compare les partitions générées avec une pondération (haut) et en sélectionnant l'élément maximal (bas). Il est facile de relever les multiples mouvements montants et descendants de la composition supérieure (flèches oranges). Cela crée une écoute davantage instable, imprévisible, qui n'est pas si dérangeante, mais qui ne reflète pas du tout le type d'écriture classique de Beethoven. Par conséquent, la voix du premier violon générée ne correspond pas du tout au style de jeu des autres instruments du quatuor, ce qui crée un gouffre stylistique à l'intérieur-même de la partition résultante. Ce n'est néanmoins pas l'élément musical le plus difficile à écouter. En effet, il est

important également de relever les notes tout à fait aberrantes, générées par BeethoPar (cercles rouges sur la figure 6.9). Ces notes sont soit absolument dissonantes, donc en désaccord avec l'harmonie, ou créent une écoute trop boiteuse. Typiquement, finir la phrase musicale sur un *sol* (dernier cercle) est tout à fait déplaisant.

C'est donc sans quatrième argument que la qualité de la génération est préférable. En effet, les notes générées par BeethoPar avec une sélection de pondération ne procurent pas de plaisir à l'écoute, à cause de leurs dissonances, et sont davantage éloignées de la partition originale de Beethoven. Par conséquent, la suite de l'analyse des résultats n'utilisera pas cet argument.

Troisième argument — Une probabilité supplémentaire ou non

Le troisième argument de BeethoPar permet d'ajouter une cinquième probabilité dans la génération des notes. Cette donnée provient d'une matrice stochastique considérant l'avant-dernière note générée. C'est donc à nouveau deux résultats différents que produit BeethoPar, suivant le choix pris pour cet argument.

Dans la partition générée avec 4 matrices stochastiques et un poids du premier violon de 0.3, 22.4% des notes sont identiques à la pièce originale de Beethoven (c.f. figure 6.6). Avec le même poids, mais en rajoutant une probabilité (i.e. en enclenchant le troisième argument), ce pourcentage descend à 20.7%. Par conséquent, en appuyant l'analyse uniquement sur l'œuvre initiale, cet argument réduit la qualité de la génération. Cependant, il est évidemment nécessaire d'interpréter l'écoute et les éléments musicaux figurant sur les partitions créées par BeethoPar. Il faut tout de même relever que ces deux partitions générées, avec et sans cet argument, sont similaires à presque 80%. Il s'agit donc d'analyser et commenter spécifiquement les quelques petites différences qui y figurent.

La partition représentée sur la figure 6.10 permet de comparer les générations avec (portée "True") et sans (portée "False") le troisième argument. On remarque directement la grande ressemblance entre les deux pièces, toutefois, tous les mouvements musicaux ne sont pas identiques. En effet, bien que les deux portées montent au début, la génération en enclenchant l'argument s'élève plus rapidement, puis se stabilise. La fin de la section montrée par la figure 6.10 souligne le relief plutôt lisse de la portée "False". Il faut relever que des mouvements trop impromptus ou turbulents ne permettent pas une écoute agréable et calme, alors qu'un relief particulièrement plat dans une partition reflète un manque d'élaboration, de raffinement. Par conséquent, les deux générations possèdent autant d'avantages que de désavantages quant à leurs mouvements internes.

La figure 6.10 soulève surtout les mouvements musicaux sur la partition. Il



FIGURE 6.10 – *Comparaison des générations avec et sans troisième argument (mm. 1-8)*. La portée supérieure représente la partition créée par BeethoPar avec l'argument et l'inférieure sans. Au milieu se trouvent les notes originales, composées par Beethoven.

faut se rappeler que le troisième argument de BeethoPar permet de prendre en compte l'avant-dernière note générée dans les probabilités. Cela est censé créer davantage de structure mélodique dans la voix engendrée par l'algorithme. C'est effectivement ce qui se passe typiquement à la mesure 4 de la partition (encadré vert à la figure 6.11). Au lieu de créer une suite de trois *si bémol* consécutifs, comme sur la portée générée sans cet argument, le court motif musical dans ce cas présente un mouvement descendant vers un *la bécarré*, septième degré de l'harmonie précédente. Ceci permet de former une montée de trois notes, introduisant ainsi de manière mélodieuse le *do* de la mesure 5. La génération du *la bécarré* est évidemment due à la cinquième probabilité issue du troisième argument, puisque cette note n'apparaît pas dans la portée "False". Par conséquent, la structure de quelques sections musicales est renforcée grâce à la sélection du troisième argument, ce qui est très avantageux.

Grâce au troisième argument, comme montré dans la figure 6.11, la structure interne de la génération est meilleure, avec davantage de petits motifs musicaux dans la voix du premier violon. La figure 6.10 affiche peut-être beaucoup de fluctuations sur la portée générée, mais ce sont tous des mouvements appréciables, qui ne dérangent en rien l'écoute de l'œuvre. Ce relief est même particulièrement agréable, puisque les notes aiguës tombent sur des *sforzandi*, qui sont des accents musicaux soudains. Cela procure ainsi un mouvement ondulatoire très séduisant. Il a donc été choisi de garder le troisième argument pour la sélection optimale.

Deuxième argument — Mélange de modes ou restriction du corpus

Le deuxième argument permet de restreindre le corpus d'apprentissage de BeethoPar, afin de ne garder qu'un mode entre majeur et mineur. De cette



FIGURE 6.11 – *Génération de BeethoPar, avec le troisième argument enclenché*. L’encadré vert souligne un passage dont la structure est renforcée grâce à cet argument. Les flèches représentent les implications des probabilités, issues de la cinquième matrice stochastique.

manière, dans le but de composer une pièce dans un mode particulier, il est possible d’entraîner les matrices stochastiques avec des pièces du même mode. Cet argument touche alors les probabilités de l’algorithme, comme le montre la figure 6.12. Par exemple, la probabilité de poursuivre un septième degré par la tonique est clairement supérieure en tonalité mineure que majeure (c.f. les coefficients avec coordonnées (11,0)).

Bien que les probabilités ont l’air plus ou moins identiques entre les différentes matrices de la figure 6.12, la partition générée en enclenchant cet argument n’est qu’à 70% similaire à celle composée sans. Ces petites modifications dans les coefficients ont donc une implication relativement ample sur la génération des notes. Également, le pourcentage de notes identiques à la partition de Beethoven est, avec l’argument, de 21.4%. La restriction du corpus permet alors de faiblement augmenter la correspondance avec la pièce originale, puisque cette valeur était de 20.7% avant. Il faut savoir que le troisième mouvement du quatrième quatuor de Beethoven est en tonalité mineure. Par conséquent, avec l’argument enclenché, c’est un corpus entièrement en mineur qui entraîne les matrices stochastiques. Comme d’habitude dans cette analyse, il s’agit maintenant de décortiquer les notes et l’écoute en détail.

La figure 6.13 illustre une comparaison des mouvements musicaux présents sur les générations avec et sans deuxième argument. Le relief est plutôt similaire dans les deux partitions, sans gros avantage ou désavantage particulier sur une d’entre elles. Les montées et descentes sont effectivement analogues en nombre et également toutes d’intervalles assez mélodieux, très appréciables

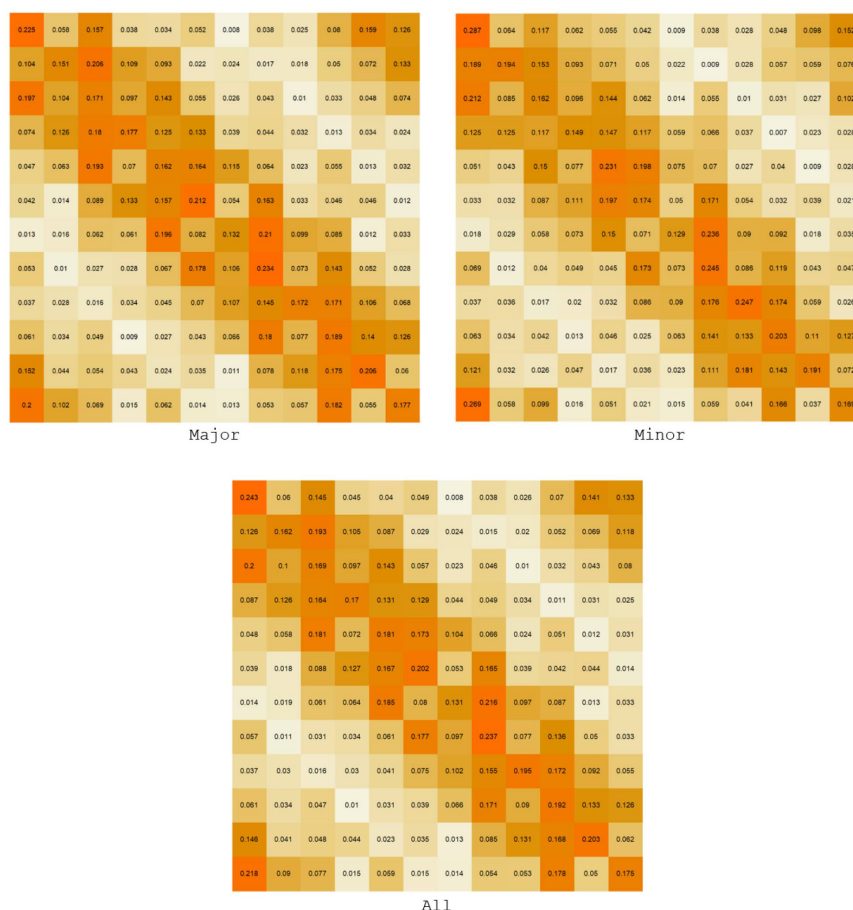


FIGURE 6.12 – *Comparaison des matrices stochastiques, suivant la restriction du corpus. Les trois matrices sont issues de la voix du premier violon et sont entraînées par un corpus majeur (en haut à gauche), mineur (en haut à droite) ou sans restriction (en bas).*



FIGURE 6.13 – *Comparaison des générations avec et sans deuxième argument. La portée supérieure représente la partition créée par BeethoPar avec l'argument et l'inférieure sans. Au milieu se trouvent les notes originales, composées par Beethoven.*

à l'écoute. Si l'analyse ne considère que cette propriété, les deux partitions engendrées par BeethoPar sont ainsi de très bonne qualité. Toutefois, c'est davantage les degrés d'une gamme mineure ou majeure qui sont mis en évidence par l'argument. Il peut être donc intéressant d'analyser les harmonies des générations de BeethoPar.

Un passage de la génération qui a particulièrement subi les modifications du deuxième paramètre, quant à ses harmonies, est illustré par la figure 6.14. En effet, au lieu de présenter la suite d'accords *do mineur* - *si bémol majeur* - *mi bémol majeur* de la partition originale, BeethoPar avec le deuxième argument enclenché compose une suite avec du *sol mineur* au centre. Il faut savoir que l'harmonie de *sol mineur* n'est pas très bienvenue à ce moment, sans sensible¹, après un accord de *do mineur*. Également, elle n'a pas du tout le caractère d'introduire le *mi bémol majeur* qui suit. La suite d'accords créée par BeethoPar est donc plutôt désagréable, puisque le parcours tonal est peu convaincant, produisant ainsi un discours musical non structuré, faible. Le fait de toujours préférer des harmonies mineures avec le deuxième argument n'est donc pas un avantage pour la génération de musique. Au contraire, la partition produite sans argument a sélectionné pour cette note un *si bémol*, l'auditeur ressent alors, comme dans la composition de Beethoven, une harmonie de *si bémol majeur*, laquelle est parfaitement à sa place dans le parcours tonal de la pièce.

FIGURE 6.14 – *Comparaison harmonique entre une génération avec deuxième argument et la partition originale. La section de gauche, produite par BeethoPar, présente la suite harmonique do mineur - sol mineur - mi bémol majeur, alors que la composition de Beethoven contient un accord de si bémol majeur.*

La suite harmonique d'une œuvre est très importante, afin que l'audi-

1. Sensible : note de la gamme chromatique, toujours à un intervalle de septième majeure par rapport à la tonique.

teur puisse être captivé par son écoute. Les accords servent effectivement à créer une progression cohérente à l'intérieur d'une pièce, permettant ainsi de transmettre un discours musical. Ces accords, afin de pouvoir exprimer n'importe quelle émotion, doivent pouvoir autant être majeurs que mineurs. Par conséquent, les matrices stochastiques d'un algorithme de génération musicale doivent être capables d'engendrer des accords des deux modes à probabilité égale, sans préférence pour l'un ou l'autre. Seules les notes présentes dans la polyphonie et avant, dans la voix du premier violon, permettront de pencher pour une harmonie plutôt majeure ou mineure, d'après les probabilités issues de leurs matrices respectives. En restreignant le corpus d'apprentissage avec le deuxième argument, la génération de BeethoPar prend parti pour un mode particulier et, comme le montre les notes de la figure 6.14, elle aura tendance à engendrer des harmonies de même mode que son apprentissage, quand bien même celles-ci seraient en contradiction avec les notes de la partition. Sans cet argument, l'algorithme BeethoPar a parfaitement su composer une partition cohérente, ne contenant que des accords de mode adéquat. Alors, le deuxième argument ne sera pas sélectionné pour la génération optimale de musique.

Premier argument — Mélange de tonalités ou transposition du corpus

Le premier argument de l'algorithme BeethoPar permet de transposer l'entier du corpus d'apprentissage, afin de ne considérer qu'une tonique : le *do*. Les degrés des gammes diatoniques de *do majeur* et *do mineur*, sont alors renforcés par ce procédé. En effet, comme le montrent les matrices stochastiques de la figure 6.15, les notes comme le *do dièse* ou le *fa dièse*, ne faisant partie ni de l'échelle de *do mineur*, ni de *do majeur*, sont représentées par des probabilités considérablement réduites (c.f. colonnes avec coordonnées 1 et 6 de la matrice de droite). Au contraire, les coefficients concernant les degrés communs aux deux gammes, comme le *ré*, le *fa* et le *sol*, sont augmentés notablement (c.f. colonnes 2, 5 et 7). Par conséquent, les probabilités de générer des notes n'appartenant pas aux échelles de *do mineur* ou *majeur* sont alors minimisées, au profit des degrés propres aux deux gammes diatoniques. Ce comportement est similaire dans toutes les autres matrices stochastiques créées avec le premier argument enclenché.

Grâce à cet argument, la partition produite par BeethoPar se rapproche encore davantage de la pièce originale de Beethoven. En effet, la génération contient plus de 29.7% de notes similaires à la composition initiale, ce qui est 9% de plus qu'auparavant. La figure 6.16 affiche justement les premières mesures du quatuor no4 de Beethoven (mouvement 3), généré avec et sans premier argument. On remarque alors facilement une assez bonne amélioration quant aux mouvements musicaux de la génération avec argument enclenché (haut), puisque celle-ci suit assidûment les montées et descentes de la pièce

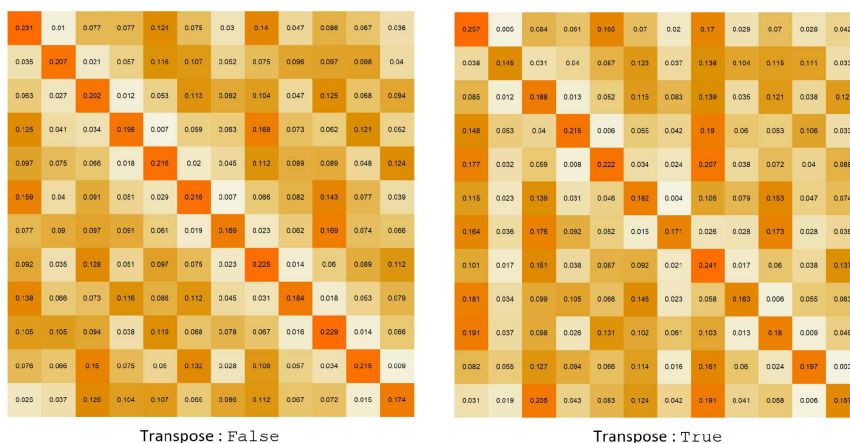


FIGURE 6.15 – *Comparaison des matrices stochastiques, suivant la transposition du corpus. Les deux matrices sont issues de la voix du violoncelle et sont entraînées par un corpus transposé en do (droite) ou non (gauche).*

composée par Beethoven, laquelle est la référence pour cette analyse. D'autres points positifs de la pièce avec premier argument sont représentés sur la partition de la figure 6.16, comme notamment le *mi bémol* de la troisième mesure (cercle vert), qui complète l'accord de septième diminuée, joué par les autres instruments. Cette dernière harmonie est typiquement propre à certains degrés de la gamme, ne pouvant donc être générée par BeethoPar qu'avec l'argument enclenché.



FIGURE 6.16 – *Comparaison des générations avec et sans premier argument. La portée supérieure représente la partition créée par BeethoPar avec l'argument et l'inférieure sans. Au milieu se trouvent les notes originales, composées par Beethoven.*

Un autre élément musical intéressant, généré avec le premier argument, se trouve aux mesures 5 à 8 de la partition et est affiché par la figure 6.17. La suite harmonique de ce passage est excellente et soutenue par la voix du premier violon produite par BeethoPar. Spécifiquement, celle-ci présente la

progression type d'une cadence complète, consistant en un enchaînement des degrés ii - i(6/4) - V - i (c.f. figure 6.17). Cette cadence permet de ponctuer le discours musical de manière très conclusive et particulièrement appréciable à l'oreille. La portée du premier violon, générée par algorithme, suit et souligne ces harmonies en jouant deux mouvements conjoints impeccables. Le premier est une succession montante (flèche verte initiale sur la figure 6.17), sur des accords de dominante¹ et tonique², introduisant sensiblement le quatrième degré *fa mineur*. Le deuxième relief est descendant (seconde flèche verte), afin de revenir délicieusement à un accord de premier degré. Les notes générées par BeethoPar sont alors en parfaite association avec les harmonies de la pièce, ce qui améliore considérablement l'écoute de la composition.

Violin I

Violin II

Viola

Violoncello

do : iv (V) → iv(6) ii(6/5) i(6/4) V(5) i

FIGURE 6.17 – *Génération de BeethoPar, avec le premier argument (mm. 5 - 8). La couleur verte présente sur la figure souligne l'impeccable progression harmonique, en association avec les notes du premier violon.*

Beaucoup de fragments musicaux sont bien mieux grâce à cet argument. Celui-ci permet effectivement à la génération de toujours se référer à une échelle diatonique graduée, puisque les matrices sont entraînées sur un corpus transposé entièrement à la même tonique. Ainsi, comme les accords et les degrés dépendent directement de cette note fondamentale, l'algorithme possédera une meilleure compréhension des progressions tonales et pourra ensuite générer une partition davantage en rapport avec l'harmonie de la pièce insérée. Le premier argument est donc sélectionné pour la génération optimale.

1. Accord de dominante : harmonie dont la fondamentale est le cinquième degré de la gamme diatonique.

2. Accord de tonique : harmonie dont la fondamentale est le premier degré de la gamme diatonique.

Sélection optimale et conclusion

Grâce à l'analyse ci-dessus, il a été possible de désigner un choix d'arguments et de les ajuster afin d'améliorer au mieux la génération de l'algorithme BeethoPar. Ces implémentations possèdent effectivement une incidence relativement élevée sur les matrices stochastiques, les différentes probabilités, le procédé de sélection et ainsi sur le contenu des partitions. Certains arguments permettent à BeethoPar de créer des pièces avec davantage de structure interne ou de cohérence harmonique, alors que d'autres lèsent la génération en composant des notes répétitives, dissonantes ou discordantes. Par des analyses des partitions, en s'intéressant aux mouvements musicaux, aux suites d'accords, à la polyphonie, à l'écoute de l'exécution, une sélection d'arguments optimaux a été définie.

Premièrement, il a été choisi de fixer le poids du premier violon à 0.3. De cette manière, la correspondance entre la génération et la pièce originale de Beethoven est idéale et les mouvements sur la partitions sont plutôt satisfaisants. Ce sont donc les probabilités issues des autres instruments qui possèdent davantage d'implication dans la génération optimisée de BeethoPar. Ensuite, pour la sélection de chaque note successive, l'analyse a opté de prendre l'index du vecteur stochastique avec argument maximal. En effet, le procédé de choisir les notes d'après une loi de probabilités pondérée par les éléments du tuple produit des œuvres musicales dissonantes, remplies de fantaisies malvenues. Pour créer ces vecteurs, il a été préféré d'utiliser une cinquième matrice markovienne, concernant chaque avant-dernière note par rapport à celle à générer. Grâce à ces probabilités, des motifs musicaux sont apparus sur la partition, rendant ainsi l'écoute davantage cohérente et plaisante. Au contraire, le fait de restreindre le corpus par mode n'améliore pas l'exécution de la génération. En effet, BeethoPar, entraîné avec une telle base de données, préfère composer des notes produisant des harmonies de même mode, ce qui n'est pas forcément opportun. Pour finir, une transposition du corpus entier, afin qu'il contienne une même tonique commune, permet de baser l'harmonie sur une seule note fondamentale. Par ce procédé, qui a été sélectionné par l'analyse, l'algorithme de BeethoPar génère des notes pour le premier violon davantage en rapport avec la polyphonie et les accords des autres instruments.

Pour résumer le choix final des arguments, le tableau 6.6 illustre les valeurs choisies et insérées dans le code d'exécution de BeethoPar, dans le but de générer une partition optimale. Grâce à cette sélection, la partition produite par l'algorithme est la plus similaire à la pièce originale de Beethoven et elle est beaucoup plus agréable à l'écoute que la création sans argument. Typiquement, le pourcentage de notes identiques entre la génération de BeethoPar, en insérant le troisième mouvement du quatrième quatuor, et la composition originale passe effectivement de 18.6% sans argument à 29.7% avec l'implé-

Numérotation	1	2	3	4	5
Argument	Transposition du corpus	Restriction du corpus par mode	Cinquième matrice stochastique	Argument maximal ou probabilité pondérée	Poids du premier violon
Valeur choisie	True	False	True	Élément maximal	0.3

TABELLE 6.6 – *Sélection optimale des arguments.*

mentation optimale. C’est une amélioration non négligeable, d’autant plus que les mouvements musicaux et les harmonies sont soutenus dans le résultat de l’algorithme, grâce aux différents arguments choisis.

Cette étude a donc pu déterminer la richesse qu’impliquent des arguments portant sur la théorie musicale dans la génération de musique par algorithme. Alors que réseaux de neurones actuels préfèrent ajouter des caractéristiques supplémentaires à leur apprentissage dans le but d’améliorer leurs résultats, l’implémentation d’hyperparamètres précis pourrait ainsi être plus avantageux. Par exemple, afin d’augmenter la cohérence harmonique d’une génération musicale, au lieu de considérer les accords du corpus d’entraînement, il suffirait de transposer l’entier du dataset à une tonalité commune, comme le fait le premier argument de BeethoPar. De plus, il s’agit d’un code plus simple et rapide, capacités toujours bienvenues dans un programme informatique. L’implémentation de ce type d’arguments ne doit alors pas être négligée par les réseaux de neurones actuels, ceux-ci devraient davantage profiter de leurs qualités.

Bibliographie

[COL21] Florian Colombo, Ecole Polytechnique Fédérale de Lausanne, Suisse, *Learning music composition with recurrent neural networks*, 2021