



La Cité du Cinéma – 20 Rue Ampère BP 12 – 93213 La Plaine Saint-Denis
Téléphone : +33 (0)1 84 67 00 01 – <https://www.ens-louis-lumiere.fr>

Jean-André VITAL
jean.vital@gmail.com

Mémoire de Master

Spécialité Son, Promotion 2022
Soutenance de XXXXX 2022

VITAMIX

Création d'un algorithme de pré-mixage automatique
basé sur des méthodes d'optimisation globale et
de l'intelligence computationnelle

Directeur de mémoire : Laurent Millot

Gestion et organisation de la VAE : Julie Bornand

Résumé

Le travail d'un ingénieur du son vise à résoudre les problèmes techniques et créatifs lors du mixage d'un morceau de musique. Certains aspects importants d'un mixage réussi sont son équilibre, sa transparence, sa profondeur et sa largeur. Dans cette étude, nous avons cherché à démontrer comment les techniques modernes d'optimisation globale et d'intelligence artificielle pouvaient aider l'ingénieur du son dans les premières étapes du mixage.

Une courte enquête par questionnaire auprès d'ingénieurs du son professionnels de longue date nous a permis de connaître leurs principales habitudes et méthodes de travail qui contribuent à créer un mixage de qualité. Nous avons ensuite construit notre propre système de mixage (VitaMix), qui est contrôlé par un algorithme d'optimisation globale issue de l'intelligence computationnelle, et la nouvelle formulation du problème est innovante et diffère dans son approche des méthodes qui ont été examinées dans la littérature scientifique au cours de la dernière décennie.

Nous avons mené un certain nombre d'expériences et dérivé nos propres critères d'optimisation pour aboutir à un pré-mixage de haute qualité. Nous avons également introduit un mécanisme de contraintes lors de la formulation du problème d'optimisation. Cela permet de prendre en compte les désirs stylistiques et esthétiques de l'ingénieur du son et d'améliorer la qualité du mixage final.

Par la suite, à l'aide de tests subjectifs en aveugle, nous avons montré les tendances préférées par l'oreille humaine ainsi que la valeur et l'intérêt potentiels du mixage automatique. Nous suggérons que dans la plupart des cas, le mixage automatique pourrait servir de bonne aide à la décision pour l'ingénieur du son novice, mais pourrait aussi augmenter la productivité de l'ingénieur du son chevronné.

Dans cette étude, nous ne montrons qu'un prototype d'un système potentiel de mixage automatique, donc en conclusion nous discutons également des améliorations possibles et du développement ultérieur du système, ainsi que de la puissance de calcul nécessaire et des caractéristiques de son utilisation.

Mots-clefs : ingénierie du son, mixage automatique, spatialisation, démasquage, optimisation globale, évolution différentielle, intelligence computationnelle, intelligence artificielle, production musicale

Abstract

A sound engineer's job is to solve technical and creative problems when mixing a piece of music. Some important aspects of a successful mix are its balance, transparency, depth and width. In this study, we demonstrated how modern global optimization and artificial intelligence techniques can help the sound engineer in the early stages of mixing.

A short questionnaire survey of long-time professional sound engineers allowed us to learn about their main habits and working methods that contribute to creating a quality mix. We built our own mixing system (VitaMix), which is controlled by a global optimization algorithm derived from computational intelligence and the new problem formulation is innovative and differs in its approach from methods that have been reviewed in the scientific literature over the last decade.

We have conducted a number of experiments and derived our own optimization criteria to achieve a high quality pre-mix. We have also introduced a constraint mechanism in the formulation of the optimization problem. This allows to take into account the stylistic and aesthetic desires of the sound engineer and to improve the quality of the final mix.

Subsequently, using blind empirical tests, we have shown the trends preferred by the human ear and the potential value and benefit of automatic mixing. We suggest that in most cases, automatic mixing could serve as a good decision aid for the novice sound engineer, but could also increase the productivity of the experienced sound engineer.

In this study, we show only a prototype of a potential automatic mixing system, so in conclusion we also discuss possible improvements and further development of the system, as well as the required computing power and the characteristics of its use.

Keywords : sound engineering, automatic mixing, spacialization, unmasking, global optimization, differential evolution, computational intelligence, artificial intelligence, music production

Dédicace

Je consacre ce mémoire à deux des caractéristiques humaines les plus attrayantes :

la paresse et le désir de beauté.

C'est la réticence à faire un travail routinier, ainsi que le désir d'être un créateur de la nouveauté et du beau, qui m'ont poussé pendant de nombreuses années à créer des produits uniques destinés à libérer les hommes d'activités fastidieuses et à leur permettre de se concentrer exclusivement sur leur nature créative et à chercher à utiliser au mieux leur potentiel créatif naturel. L'outil de mixage automatique, au cœur de ce mémoire, s'inscrit dans cette démarche comme nous allons pouvoir le constater.

Remerciements

Je tiens tout d'abord à remercier l'administration de l'École Nationale Supérieure Louis-Lumière, ainsi que l'équipe d'enseignants qui ont accepté ma candidature atypique, pour leur confiance et leur foi en mes capacités, non seulement en tant que chercheur et musicien, mais surtout en tant qu'ingénieur du son qui pourrait contribuer à apporter un peu d'innovation à cette merveilleuse spécialité.

Merci en particulier à Alan Blum, qui a été mon premier interlocuteur, très agréable, et qui a passé beaucoup de temps à analyser mes idées et mes élans créatifs ainsi qu'à me donner des conseils avisés.

Mes remerciements les plus sincères à Julie Bornand, pour son aide précieuse tant sur le plan administratif que pour son avis et ses conseils objectifs sur la rédaction de mon dossier professionnel.

Un grand merci à Laurent Millot pour, tout d'abord, avoir cru en mes idées et mon projet, mon potentiel en tant qu'ingénieur du son, pour les longues discussions et la correspondance sur la partie scientifique, pour les nombreuses questions intéressantes et curieuses, dont les réponses ont constitué la base de ce travail, des conseils très utiles et pratiques sur l'organisation du travail en général, non seulement la partie scientifique, mais aussi méthodique.

Merci à Joshua Reiss, pour une introduction très rapide au domaine du mixage automatique, grâce aux liens très intéressants vers des articles sur les derniers développements dans le domaine. Merci également à ses nombreux étudiants et doctorants dont j'ai pu lire les mémoires et les thèses, et ceux d'entre eux avec qui j'ai pu discuter de leur vision du mixage automatique. Parmi eux, je tiens particulièrement à mentionner : Pedro Pestana, Brecht De Man, Sina Hafezi, Ajin Tom et David Ronan.

Je suis très reconnaissant à Sascha Dick pour ses conseils pratiques et son introduction aux modèles psychoacoustiques, et je suis également reconnaissant à Brian Moore d'avoir pris le temps de me proposer un accès à ses approches de la modélisation de l'audition humaine.

Je remercie tous les participants actifs à mes questionnaires liés à ce travail, ainsi que tous les participants aux forums sur l'ingénierie du son avec lesquels nous avons pu discuter plus en détail de certains aspects de la profession, et pour toutes les conversations personnelles. La plupart de leurs noms figurent dans Annexe A.

Merci aux sociétés commerciales qui ont répondu pour parler de mes idées et de mon projet, pour leur ouverture, leur profond intérêt et leur soutien. Pour des raisons de secret commercial, les noms de ces sociétés ne sont pas mentionnés ici.

Je remercie tout particulièrement la société SkyWorkflows pour avoir fourni les algorithmes d'optimisation, qui ont permis de concrétiser des idées hypothétiquement irréalistes et d'aboutir à un prototype complètement fonctionnel.

Table des matières

1 Introduction	13
1.1 Aperçu du sujet traité et du travail réalisé	13
1.2 Les études existantes	14
1.3 Les limites de l'étude proposée	15
1.4 Plan de ce document	16
2 Le mixage musical	17
2.1 Processus mis en œuvre lors d'une session de mixage typique	17
2.2 Que peut-on automatiser?	22
2.3 Les habitudes des ingénieurs du son	26
2.3.1 Comment est-ce que vous commencez votre session de mixage?	27
2.3.2 Préparation, édition et nettoyage des pistes (voix, bruits,...)	28
2.3.3 Balance	28
2.3.4 Spatialisation (pans)	30
2.3.5 Égalisation	31
2.3.6 Dynamique et compression, compression parallèle	33
2.3.7 Regroupement de pistes	34
2.3.8 Pistes monophoniques VS stéréophoniques, mixage en monophonique et/ou en stéréophonique	35
2.3.9 Masking et unmasking	36
2.3.10 Utilisation du Master bus (esthétiques, rééquilibrage global en fréquences, ...)	37
2.3.11 Aspects esthétiques et artistiques	40
2.3.12 Aspects techniques et technologies	41
2.3.13 Utilisation de modèles (pré-réglages, styles, workflow, ...)	42
2.3.14 L'importance de la balance et de la spatialisation	43
2.3.15 Quels objectifs esthétiques et techniques la spatialisation permet-elle de résoudre?	46
2.3.16 Quels sont les plug-ins ou les effets hardwares que vous utilisez le plus souvent?	48
2.3.17 Discussions libres	49
2.4 Au lieu d'une conclusion	50
3 Architecture du système de mixage automatique	51
3.1 Le langage de programmation	51
3.2 Le dictionnaire	52
3.3 L'abstraction de la scène virtuelle	53
3.3.1 Les trois plans de volumes	54
3.3.2 Les plans panoramiques	54
3.3.3 Les types d'égalisation et de compression	56
L'égalisation	56
La compression	57
3.4 Les groupes prédéfinis et les bus	57

3.5	La description du projet	57
3.6	La description du style	59
3.7	La scène "3D" interactive ou pilotée par l'utilisateur	61
3.8	L'aperçu de l'architecture du système	62
3.9	Les formats de données et les unités de mesure	64
3.10	Le chargeur du project – <i>Loader</i>	65
3.11	Le chargeur des bus – <i>Bus</i>	65
3.12	Le mixeur – <i>Mixer</i>	65
3.13	Les règles de mixage – <i>MixRules</i>	66
3.14	La classe interprète – <i>Mapper</i>	66
3.15	L'utilitaire de traitement audio – <i>Audiotools</i>	66
3.16	L'utilitaire des métriques audio - <i>Metrics</i>	68
3.17	Le paquet d'effets – <i>Pedalboard</i>	69
4	L'optimisation numérique	71
4.1	L'introduction à l'optimisation	71
4.2	Les types des fonctions objectifs et des contraintes	72
4.3	Les types des variables	73
4.4	Les fonctions objectifs spécifiques	73
4.5	L'optimisation multi critères	74
4.6	Scalarisation	75
4.7	Les méthodes d'optimisation	75
4.8	Les méthodes méta-heuristiques	76
4.9	La bibliothèque d'optimisation <i>VitaOptimum</i>	78
4.10	L'évolution différentielle	79
4.11	Un exemple d'utilisation du package <i>VitaOptimum</i>	82
5	Les critères d'optimisation	83
5.1	Le design de la fonction objectif	83
5.2	Les discussions sur les modèles psychoacoustiques, leur complexité et leur abstraction excessive	84
5.3	Les transformations de Fourier	85
5.4	L'énergie et l'équilibre des canaux droit et gauche	87
5.5	L'énergie conjointe et le masquage	87
5.6	Les expériences sur l'attention humaine et le caractère aléatoire des calculs	88
5.7	Les expériences avec des types de mixages pour le calcul du masquage	90
5.8	Les ressources mémoire nécessaire à un calcul performant	90
5.9	Les modèles de profondeur et de largeur, distance entre les instruments	91
5.10	Le problème d'optimisation multi critères	92
6	Les résultats et les tests perceptifs	93
6.1	La sélection des projets musicaux à mixer	93
6.2	La convergence et l'ensemble de solutions	93
6.3	Le questionnaire : les tests perceptifs	95
6.4	L'interprétation des résultats	96
6.4.1	Le graphe de convergence de la fonction objectif	97
6.4.2	Le graphe récapitulatif des paramètres du modèle psychoacoustique	99
6.4.3	Le graphe individuel des paramètres pour chaque instrument	101
6.5	L'observation et la discussion des résultats	102
6.6	Quelques citations	106
6.6.1	Classic : Rod Alexander – Canon	106
6.6.2	Folk : Leslie Mendelson – The Hardest Part 2	107

6.6.3	Ska : Big Stone Culture – Fragile Thoughts	107
6.6.4	Latino : Eddie Garrido – Una Semana Sin Ti	108
6.6.5	Country : Angela Thomas Wade – Milk Cow Blues	109
6.6.6	Choir : Don Camillo Choir – Marsh Marigolds Song	110
6.6.7	Jazz : Araujo – The Saga Of Harrison Crabfeathers	111
6.6.8	Ballad : MR 1001 – Jay Menon.	112
6.6.9	Quelques notes complémentaires	113
6.7	Quelles conclusions pouvons-nous tirer?	113
7	Comment améliorer le système de mixage automatique à l’avenir?	115
7.1	Quelles sont les limites en termes de vitesse de calcul et de mémoire?	115
7.2	Comment et par quelles méthodes peut-on améliorer la performance?	116
7.2.1	Plus de mémoire, moins de calculs	116
7.2.2	Le changement du langage et l’optimisation du code	116
7.2.3	Les calculs parallèles et distribués	116
7.2.4	Le principe de séparation pour la réduction de la dimensionnalité	117
7.3	Les options pour la mise en œuvre et l’utilisation du programme VitaMIX	118
7.3.1	L’accélération du calcul	118
7.3.2	Les serveurs et le calcul distribué	119
7.3.3	Les services en ligne	120
7.3.4	L’estimation de prix	120
7.4	Est-il nécessaire de complexifier le modèle de démasquage?	121
7.5	L’analyse préliminaire des pistes par des méthodes d’apprentissage profond	122
7.6	Les extensions de fonctionnalité	122
7.6.1	L’extension à <i>l’égalisation</i>	122
7.6.2	L’extension à <i>la compression</i>	122
7.7	L’extension des styles avec des plug-ins externes	123
7.7.1	Les <i>inserts</i> stylistiques	123
7.7.2	Les <i>sends</i> et la création de plans de réverbération	123
7.8	La redondance des solutions complique l’optimisation	124
7.9	En guise de conclusion	124
8	Conclusion	125
	Bibliographie	128
A	Liste des personnes participant aux questionnaires	139
A.1	Le mixage musical	139
A.2	Les tests perceptifs	139
B	L’intelligence artificielle	141
B.1	Le contexte philosophique	141
B.2	Les conditions technologiques	141
B.3	La naissance de la science : "Intelligence artificielle"	141
B.4	L’intelligence artificielle moderne	142
B.5	L’apprentissage automatique	142
B.6	L’apprentissage profond	143
B.7	L’intelligence computationnelle	143
B.8	Le changement du paradigme	144
B.9	L’optimisation numérique	144

C Les projets	145
C.1 Classic : Rod Alexander - Canon : 2 tracks	145
C.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks	146
C.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks	147
C.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks	149
C.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks	151
C.6 Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks	154
C.7 Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks	157
C.8 Ballad : MR 1001 - Jay Menon : 16 tracks	160
D Les styles	165
D.1 Classic : Rod Alexander - Canon : 2 tracks	165
D.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks	166
D.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks	167
D.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks	169
D.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks	171
D.6 Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks	172
D.7 Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks	174
D.8 Ballad : MR 1001 - Jay Menon : 16 tracks	175
E Les règles de mixage	179
E.1 Classic : Rod Alexander - Canon : 2 tracks	179
E.1.1 Les règles de mixage	179
E.1.2 Les graphes de convergence	181
La fonction objectif	181
Tous les instruments ensemble	182
Chaque instrument séparément	183
E.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks	184
E.2.1 Les règles de mixage	184
E.2.2 Les graphes de convergence	186
La fonction objectif	186
Tous les instruments ensemble	187
Chaque instrument séparément	188
E.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks	191
E.3.1 Les règles de mixage	191
E.3.2 Les graphes de convergence	194
La fonction objectif	194
Tous les instruments ensemble	195
Chaque instrument séparément	196
E.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks	202
E.4.1 Les règles de mixage	202
E.4.2 Les graphes de convergence	205
La fonction objectif	205
Tous les instruments ensemble	206
Chaque instrument séparément	207
E.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks	216
E.5.1 Les règles de mixage	216
E.5.2 Les graphes de convergence	219
La fonction objectif	219
Tous les instruments ensemble	220
Chaque instrument séparément	221

E.6	Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks	231
E.6.1	Les règles de mixage	231
E.6.2	Les graphes de convergence	235
La fonction objectif	235	
Tous les instruments ensemble	236	
Chaque instrument séparément	237	
E.7	Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks	248
E.7.1	Les règles de mixage	248
E.7.2	Les graphes de convergence	251
La fonction objectif	251	
Tous les instruments ensemble	252	
Chaque instrument séparément	253	
E.8	Ballad : MR 1001 - Jay Menon : 16 tracks	263
E.8.1	Les règles de mixage	263
E.8.2	Les graphes de convergence	267
La fonction objectif	267	
Tous les instruments ensemble	268	
Chaque instrument séparément	269	

Chapitre 1

Introduction

1.1 Aperçu du sujet traité et du travail réalisé

Le processus de transformation d'une composition musicale en une œuvre produite, prête pour la distribution, intègre un grand nombre de tâches diverses. Ces tâches sont généralement réalisées par une équipe de spécialistes travaillant en tandem, et comprenant des musiciens, un ou plusieurs producteurs, une maison de disques ainsi que des ingénieurs d'enregistrement, de mixage et de mastering.

Dans ce travail, nous allons nous concentrer sur l'étape du mixage. Le mixage musical multipiste est un processus itératif dans lequel différents paramètres de traitement tels que la balance des volumes sonores, l'égalisation et la compression sont ajustés pour atteindre un objectif esthétique et technique défini. Cet objectif est toujours conforme à des critères perceptifs et artistiques.

Un ingénieur de mixage musical peut créer un mixage « mental » de l'enregistrement simplement en écoutant les multipistes bruts s'il est suffisamment expérimenté. Ce mixage est basé sur la vision de l'ingénieur et peut être influencé grâce à l'apport du producteur et/ou de l'artiste. Pour certains mixeurs, la première étape consiste à chercher à créer un mixage dans lequel tous les instruments sonnent clairs, équilibrés et séparés dans l'espace sans masquage auditif. Cette tâche est considérée par certains mixeurs comme une tâche technique visant à constituer une base équilibrée à partir de laquelle le travail créatif pourra être mené plus facilement dans un second temps. Cela nous permet de tracer une ligne virtuelle et souvent invisible entre les tâches créatives exprimant la vision d'un ingénieur mixeur et les tâches plutôt techniques qui ont un caractère assez objectif du point de vue de la perception musicale et pouvant même être partiellement estimées via des mesures techniques ou des formules mathématiques.

L'arrivée récente de l'intelligence artificielle et computationnelle permet de développer un axe scientifique que l'on pourrait qualifier de « production musicale intelligente », et cette arrivée s'accompagne de nombreuses tentatives d'automatisation des aspects techniques du mixage musical.

On peut d'ailleurs se demander si, dans un avenir proche, on pourrait disposer d'une sorte d'assistant technique virtuel et intelligent qui nous aiderait à résoudre les problèmes techniques typiques « non artistiques » qui sont susceptibles de se présenter pendant une session de mixage. Par exemple, cet assistant virtuel pour proposer des pré-réglages de mixage afin de gagner du temps dans la préparation d'une session.

Dans ce travail, nous allons étudier si on peut effectivement envisager la conception d'un assistant virtuel de ce type. Nous allons donc proposer dans ce travail l'étude, la mise en œuvre et le test de stratégies de pré-mixage automatique. Ces stratégies viseront à optimiser la localisation des sources sonores dans l'espace virtuel tridimensionnel à l'aide d'une approche innovatrice basée sur des méthodes métaheuristiques d'optimisation globale [Feoktistov, 2006]. Compte tenu de l'ampleur du travail et des moyens informatiques limités dont nous avons pu disposer, nous limiterons l'étude à l'optimisation d'un nombre limité de critères : l'équilibre fréquentiel entre les canaux gauche et droit, le masquage des fréquences, la profondeur et la largeur du mixage.

En parallèle de ce travail d'optimisation, nous avons mené une enquête préliminaire auprès d'ingénieurs du son expérimentés afin d'essayer d'identifier les points importants pour eux pour mener à la réalisation d'un bon mixage, y compris le placement spatial des instruments. Nous avons également effectué des tests d'écoute à partir des pré-mixages proposés par l'outil développé pour mieux comprendre l'utilité pratique de la méthode proposée.

Ce travail se concentra sur les aspects scientifiques de l'optimisation et sur la mise en œuvre technique du système d'assistant virtuel proposé. Mais, on abordera aussi l'utilisation pratique potentielle d'un tel assistant virtuel ainsi que les exigences supplémentaires introduites s'agissant de la station de travail d'un ingénieur du son.

1.2 Les études existantes

Bien que constituant une composante essentielle de la production musicale, le mixage reste encore une tâche ésotérique avec pas à peu de "bonnes pratiques" établies [Tot, 2018; Case, 2012; De Man, 2017; Gibson, 2019; Izhaki, 2017; Owsinski, 1999]. Néanmoins, il existe quelques tentatives scientifiques pour étudier ce processus qui peut sembler quelque peu "mystique" :

- soit se concentrant sur la question de l'automatisation des égalisations [Perez-Gonzalez and Reiss, 2009; Hafezi and Reiss, 2015];
- soit se concentrant sur l'automatisation des compressions [Giannoulis et al., 2013; Ma et al., 2015];
- soit se concentrant sur l'automatisation du placement des sources dans l'espace stéréophonique, ce que nous désignerons dans la suite sous le nom de pans [Tom et al., 2019; Perez-Gonzalez and Reiss, 2010; Mansbridge et al., 2012; Pestana and Reiss, 2014];
- soit se concentrant sur l'automatisation par réseaux de neurones ou Neuro FX [Steinmetz, 2020; Steinmetz et al., 2021].

Les trois premières approches ont essayé d'utiliser autant que possible des critères objectivement mesurables qui déterminent la qualité finale du mixage, tandis que la dernière approche est basée sur des réseaux de neurones (Deep Learning) "formés" pour imiter partiellement le travail de mixage d'un ingénieur du son.

Mais, dans tous les cas, la qualité des mixages qui en résultent est indéniablement pire que le travail réalisé par un ingénieur de mixage expérimenté. Cela signifie qu'il existe de nombreux problèmes scientifiques et techniques dans ce domaine qui n'ont pas encore été résolus, et peut-être que pour certains d'entre eux, aucune solution ne sera trouvée dans les années à venir. En d'autres termes, de tels systèmes sont encore très loin d'être parfaits, et, dans un proche avenir, on ne peut au mieux qu'espérer disposer d'un « assistant intelligent » virtuel permettant aux ingénieurs mixeurs de faciliter leurs tâches routinières.

1.3 Les limites de l'étude proposée

Afin de circonscrire le problème, nous avons arrêté quelques abstractions, simplifications et hypothèses.

Nous avons divisé toutes les opérations réalisées par l'ingénieur mixeur en opérations de macro-mixage et opérations de micro-mixage. Le macro-mixage concerne l'image globale de la perception sonore, tandis que le micro-mixage concerne l'élaboration créative des détails de chaque instrument. Les opérations de micro-mixage visent à créer un espace créatif subjectif pour l'ingénieur, tandis que les opérations de macro-mixage visent à résoudre un certain nombre de défis techniques plus ou moins objectifs afin de garantir un mixage équilibré.

Parmi l'ensemble de défis techniques, la séparation spatiale des sources semble avoir un de plus grand impact s'agissant de l'amélioration globale du mixage [Tot, 2018; Tom et al., 2019]. Et, selon une étude statistique [Jillings and Stables, 2017] (cf. Tableau 1.1), un ingénieur du son passe jusqu'à 45% de son temps à travailler sur l'équilibre et la constitution de la scène sonore. Par conséquent, une automatisation même partielle de ces processus pourrait permettre de réaliser une partie du travail de routine et ainsi soulager considérablement le travail de l'ingénieur du son et lui permettre de se concentrer sur les aspects créatifs, donc subjectifs, du mixage.

Action	Nombre	%
Play/Pause/Stop	1 318	33.066
Set Volume	1 247	31.284
Set Pan position	500	12.543
Toggle Solo	457	11.465
Set Track Group	221	5.544
Toggle Mute	102	2.559
Add Bus	67	1.681
Add Send to track	66	1.655
Alter Track Name	8	0.201

Table 1.1 – Statistiques sur les opérations typiques d'un ingénieur du son lors de la création d'un mixage musicale (d'après [Jillings and Stables, 2017]).

Dans ce travail, nous allons traiter les aspects de spatialisation, donc chercher à placer les sources sonores dans le champ ou espace stéréophonique (la scène virtuelle [Gibson, 2019]) de manière à réduire voire éliminer les problèmes de masquage [Tom et al., 2019; Pestana and Reiss, 2014]. On cherchera donc à proposer un pré-mixage permettant de plonger l'auditeur dans un espace sonore clair, large et profond. D'après Wilson, [Wilson, 2017], ce processus de spatialisation des sources d'un enregistrement multipiste est un problème d'optimisation difficile, principalement en raison de la complexité de la perception auditive.

Nous proposons un nouveau modèle de spatialisation et un algorithme de pré-mixage associé qui vise à satisfaire les critères d'équilibre, de masquage multipiste, de profondeur et de largeur pour créer un mixage bien spatialisé avec la meilleure clarté possible.

Comme il n'est pas possible de résoudre tous les problèmes techniques liés au mixage automatique dans le cadre d'un seul mémoire et afin de réduire le périmètre de notre étude, nous avons considéré que les étapes suivantes déjà réalisées au préalable :

1. le travail de nettoyage et d'édition des pistes est fait;
2. les pistes sont groupées¹ [Mahanta et al., 2021; Fuhrmann, 2012];
3. le genre musical est connu ou détecté automatiquement [Hsu et al., 2021; Pimenta-Zanon et al., 2021; Agrawal and Nandy, 2020];
4. l'utilisateur a exprimé ces préférences de mixages (optionnel) cf. Annexe C et Annexe D.

Tous ces points seront repris et décrits plus en détail dans le corps du texte.

1.4 Plan de ce document

Après l'introduction de ce chapitre (cf. Chapitre 1), dans le Chapitre 2, nous examinons une session de mixage typique, nous discutons de ce qui pourrait théoriquement ou pratiquement être automatisé dans cette session, et nous parlons en détail des habitudes et des méthodes de travail des ingénieurs du son expérimentés, sur la base de notre enquête par questionnaire. Dans le Chapitre 3, nous présenterons l'architecture du système de mixage automatique que nous avons développé dans le cadre de ce projet et nous plongerons dans les détails de sa mise en œuvre. Dans le Chapitre 4, nous présentons une très brève introduction aux méthodes d'optimisation et discutons plus en détail de l'utilisation de la bibliothèque d'algorithmes d'optimisation globale *VitaOptimum*, qui a constitué la base de ce travail. Le Chapitre 5 est entièrement consacré à la discussion des critères d'optimisation. Nous discuterons ici des critères d'optimisation que nous avons développés et proposés, ainsi que des méthodes de modélisation et de résolution du problème d'optimisation lui-même. Le Chapitre 6 décrit les projets de mixage que nous avons utilisés pour tester les performances de notre système et discute des aspects positifs et négatifs de ce système de mixage sur la base des tests perceptifs que nous avons effectués et des réponses que nous avons reçues d'experts indépendants, d'étudiants en ingénierie du son, de musiciens professionnels et d'audiophiles. Le Chapitre 7 décrit comment le système actuel peut être amélioré à l'avenir, y compris les exigences techniques et les méthodes de mise en œuvre pratique basées sur les avancées scientifiques et technologiques actuelles. Le Chapitre 8 résume le travail effectué.

Ce mémoire contient également un certain nombre d'annexes. Dans l'Annexe A, on trouve la liste des personnes qui ont participé au questionnaire et aux tests perceptifs. Dans l'Annexe B nous présentons brièvement la terminologie liée à l'intelligence artificielle, à l'apprentissage profond et aux techniques d'optimisation. En Annexe C nous présentons les fichiers décrivant les projets de mixage utilisés dans ce travail et en Annexe D nous montrons les fichiers décrivant des propositions stylistiques et techniques du mixage. L'Annexe E présente les résultats de l'optimisation des projets de mixage sous la forme d'un ensemble de règles de mixages et de graphiques de convergence correspondants.

1. cela pourrait être fait automatiquement, par exemple, avec des méthodes de classification de deep learning

Chapitre 2

Le mixage musical

2.1 Processus mis en œuvre lors d'une session de mixage typique

Il y a autant d'approches du mixage que d'ingénieurs du son. Chacun apporte quelque chose de différent, de nouveau, mémorable, reconnaissable et inoubliable. Dans cette section, nous allons essayer de résumer, en termes simples, les étapes de base du processus de mixage musical. Nous n'entrerons pas dans les détails et nous nous contenterons de présenter les éléments généraux.

Tout commence par l'immersion dans la session et la découverte du matériau musical constituant le projet. Certains mixeurs écoutent les enregistrements bruts, d'autres écoutent des mixages de référence, d'autres encore se contentent d'écouter de la musique dans un style donné, tandis que certains préfèrent écouter le silence (ou "fond d'air") du studio.

La préparation peut commencer par l'importation des pistes dans un projet au sein de la station audionumérique ou DAW¹, leur écoute, leur tri, le regroupement de pistes en sections, l'adoption de couleurs différentes bien précises pour les pistes suivant leur nature, ou encore le chargement de modèles (templates). La préparation peut ainsi consister à effectuer toutes les actions qui vont simplifier et faciliter le travail ultérieur.

Dans le cas d'une chanson, l'accent est mis sur le traitement du chant. Le montage vocal est une forme d'art qui consiste à couper et à sélectionner les bonnes pistes et prises, à les nettoyer en éliminant les bruits inutiles, les clics, à traiter les sifflantes gênantes ainsi que les consonnes explosives, à aligner le rythme, à corriger ou à créer un groove, à ajuster (tuning) la voix, voire même à construire des stems ("tiges"²) vocaux individuels.

Récemment, en raison de la spécificité de ce travail et selon le principe de la division du travail, certains ingénieurs du son préfèrent sous-traiter cette partie du travail à des personnes spécialisées dans le travail de traitement vocal.

Souvent, dès le début, pendant ou après l'écoute du matériau sonore, l'ingénieur du son a déjà une idée assez claire du son final, du moins des principaux éléments, des idées clés qu'il aimerait réaliser, y compris l'image spatiale du placement de tous les éléments. La constitution de cette

1. Digital Audio Workstation ou un logiciel de M.A.O (musique assistée par ordinateur)

2. Regroupement des pistes [Decherf, 2021]

représentation mentale artistique et esthétique constitue le point d'appui pour les étapes ultérieures du travail.

En fonction du paysage sonore qu'ils ont imaginé, certains ingénieurs du son préfèrent commencer par nettoyer le projet, c'est-à-dire supprimer tout ce qui leur apparaît superflu ou secondaire. Il peut s'agir de fréquences d'instruments inutiles, de pistes inutiles, et plus largement de tout ce qui les gêne.

Le regroupement des pistes représente une étape importante, car l'organisation des pistes ou leur structuration réduisent la complexité du projet et permettent d'économiser les ressources techniques. Certains ingénieurs du son préfèrent partir d'un schéma de groupement qui leur apparaît naturel et confortable, d'un traitement des groupes et d'une chaîne de signaux prédéfinis, tandis que d'autres préfèrent toujours faire table rase, partir d'une page blanche et laisser complètement libre cours à leur créativité.

Le plus souvent, le placement des sources sonores dans l'espace est décidé dès le départ. Ici aussi, chacun peut avoir une approche différente. Certains ingénieurs du son commencent par équilibrer les pistes en les écoutant en monophonie, tandis que d'autres travaillent en stéréophonie et vérifient périodiquement ce que cela donne une fois réduit en monophonique. Un mixeur va commencer par le panorama et ajuster ensuite les balances, quand un autre va régler les contrôleurs de volume (faders) et ceux de positionnement angulaire (panoramiques) en même temps. L'essentiel, au final, consiste à obtenir l'image souhaitée pour la disposition spatiale des instruments, où chaque instrument doit être bien perceptible dans le mixage, remplir sa fonction musicale et, en général, une image globale qui doit être bien équilibrée.

On constate qu'il existe un grand nombre d'approches mais on peut néanmoins proposer de distinguer certains modèles d'approches du mixage. Bien entendu, le style de musique a également un impact conséquent.

Dans la musique moderne, il est plus courant de commencer par équilibrer les pistes comportant les basses fréquences, comme les pistes de grosse caisse et de basse, puis de les compléter par la "colonne vertébrale" constituée par les instruments responsables du groove et la voix principale. Ensuite, on traite progressivement le reste des instruments, des plus importants au moins importants. De même, plus les fréquences fondamentales d'un instrument sont basses, plus ce instrument sera positionné proche du centre, et, à l'inverse, plus les fréquences sont hautes, plus l'instrument peut être éloigné du centre et basculé vers la périphérie. Des modèles similaires sont décrits de manière très détaillée dans le travail de thèse de Pedro Pestana [Pestana, 2013].

L'égalisation est l'un des outils les plus fréquemment utilisés. Elle est utilisée dès le début de la session du mixage, par exemple, pour nettoyer les bruits indésirables, supprimer les résonances ou couper le spectre inutile des fréquences basses et/ou de hautes. L'égalisation est également utilisée comme un outil de base pour résoudre des problèmes du masquage d'un instrument par un autre, ainsi que pour égaliser l'équilibre tonal. Seule ou en combinaison avec des effets temporaires (délai et réverbération), l'égalisation est utilisée pour construire la disposition et l'emplacement spatial, en profondeur, des sources audio dans le mixage.

Il y a ceux qui aiment "couper seulement"³ et ceux qui aiment "ajouter seulement"⁴ des gammes de fréquences. Ces décisions peuvent également dépendre du type d'égaliseur.

3. travailler négativement, c'est-à-dire atténuer de plusieurs décibels certaines fréquences

4. travailler positivement en amplifiant de plusieurs décibels des fréquences

Par ailleurs, une application fréquente de l'égaliseur permet d'ajouter ou de jouer avec la coloration du son ainsi que sur les déphasages relatifs des différentes composantes fréquentielles et correspondent à des effets utilisés à des fins artistiques et stylistiques.

Dans l'étude qui a été menée, nous avons uniquement utilisé l'égalisation pour couper les fréquences basses et hautes afin de résoudre des problèmes de masquage. Toutes les autres applications de l'égalisation pourraient également être partiellement ou totalement automatisées [Perez-Gonzalez and Reiss, 2009; Hafezi and Reiss, 2015; Spyridon, 2019], mais elles n'ont pas été testées dans cette étude.

La compression "sidechain" (aussi appelé "entrée latérale") est beaucoup utilisée en dernier. L'exemple typique d'utilisation est ce qu'on appelle en anglais le "ducking" et qui permet d'améliorer l'intelligibilité d'un animateur radiophonique ou d'un DJ lorsqu'il parle alors que de la musique continue à être diffusée.

La compression "sidechain" permet aussi de résoudre certains problèmes de mixage notamment le conflit entre une basse et une grosse caisse qui "se disputent" la même zone spectrale. On peut encore recourir à la compression "sidechain" pour améliorer l'insertion d'une voix dans un mixage.

Le sidechain peut aussi avoir son utilité pour des choses plus créatives comme, par exemple, l'animation d'une nappe de synthétiseur en fonction d'un kick.

L'outil suivant important est la compression. La compression correspond à la réduction de la dynamique du signal lorsque le niveau sonore des parties les plus fortes est diminué en fonction du paramètre de ratio, alors que le niveau global est augmenté.

Voici une liste de quelques-unes des tâches pour lesquelles un compresseur peut constituer une aide pour un ingénieur du son :

- épaissement du signal ;
- adoucissement des pics aigus ;
- égalisation du volume global et de la dynamique ;
- modification de l'enveloppe ;
- augmentation ou diminution de l'attaque de l'instrument ou de la voix ;
- augmentation ou diminution du sustain (durée du son d'une note au volume maximum) ;
- augmentation ou diminution du release (la durée du déclin de la note).

La compression est très utilisée de nos jours, car les auditeurs perçoivent un son plus fort comme étant de meilleure qualité, alors qu'en réalité il a une dynamique plus restreinte.

La compression est souvent nécessaire pour la voix afin de la rendre plus intelligible dans le mixage.

Il existe également un autre effet implicite de la compression : en fonction des valeurs d'attaque et du release, on peut créer de l'espace, même sans délai ni réverbération. Plus la valeur d'attaque est grande et plus le relâchement (release) est petit, plus l'auditeur se sent proche de l'instrument. Et, inversement, avec une petite attaque et un grand release, l'instrument passe au second plan.

Des cas particuliers de compression correspondent à la limitation, l'expansion, la distorsion et le gating.

Il existe différents types de compresseurs : compresseurs optiques, compresseurs à pente variable, compresseurs FET, compresseurs VCA. Tout comme les égaliseurs, les compresseurs sont souvent utilisés pour colorer esthétiquement le son. Par exemple, des compresseurs célèbres tels que Teletronix LA-2A, UREI LN 1176 (Universal Audio 1176), Fairchild 670, Manley Variable Mu sont les plus utilisés pour ce but.

L'étape suivante du mixage est l'application d'effets spatiaux qui comprennent le délai et ses dérivés (chorus, flanger, phaser) ainsi que la réverbération.

Dans la musique classique, les microphones sont utilisés pour capter et reproduire l'acoustique naturelle d'une salle de concert, par contre, la musique moderne utilise la réverbération pour créer un espace artificiel. Dans ce dernier cas, outre le traitement artistique, la réverbération est le plus souvent utilisée pour créer des plans sonores et une sensation de profondeur de l'image spatiale d'un mixage.

Les effets de délai participent également à la fois à la création de l'espace et à son remplissage artistique.

Nous ne nous attarderons pas sur tous les détails possibles, qui sont nombreux et dont les descriptions pourraient faire l'objet de plusieurs livres, car notre objectif consiste à présenter les grandes lignes du processus de mixage musical.

Il reste à dire quelques mots sur la saturation, qui peut être utilisée à la fois pour souligner des caractéristiques stylistiques telles que la chaleur des tubes, le gras du son, ou pour résoudre des problèmes techniques tels que la saturation harmonique pour obtenir une meilleure intelligibilité des basses fréquences lors de la diffusion par des équipements amateurs standards qui ne sont pas capables de transmettre correctement les fréquences basses.

L'automatisation est très largement utilisée dans des contextes très différents. Nous avons déjà mentionné le sidechain comme un cas particulier d'automatisation d'un compresseur par un signal externe, mais, de manière générale, nous pouvons parler de l'automatisation de presque n'importe quelle instrumentation (outil) dans le mixage, depuis le volume, le plus simple, pour souligner certains éléments et contrôler l'attention de l'auditeur, jusqu'à des automatisations plus complexes comme celle consistant à dessiner manuellement ou "algorithmiquement" le panoramique de diffusion [Pestana and Reiss, 2014] pour rendre un instrument particulier plus perceptible dans le mixage.

Le traitement des groupes mérite également d'être mentionné tel que, par exemple, la fusion de percussions, la compression parallèle de la batterie ou des voix, le regroupement des traitements pour réduire les coûts de calcul, y compris de canaux d'envoi (sends) qui peuvent être considérés comme un cas particulier de regroupement.

Chacun a une approche différente de l'utilisation du bus maître. Certaines personnes préfèrent ne rien avoir sur ce bus et pensent que c'est le signe d'un mixage réussi si aucun ajustement supplémentaire n'est nécessaire sur le "master" (bus maître). D'autres appliquent un limiteur sur ce

bus et font tout le mixage avec un limiteur activé⁵ afin de préserver l'image spatiale et dynamique du mixage. Le bus maître est souvent utilisé pour saturer l'ensemble du mixage et créer ainsi des harmoniques et corriger l'équilibre tonal si nécessaire.

Pour conclure cette section, il convient de noter que le processus de mixage est un **processus itératif** et qu'un ingénieur du son revient constamment aux étapes précédentes pour faire des ajustements, changer des choses ou les refaire, et ce processus itératif peut prendre beaucoup de temps, surtout si l'ingénieur du son est perfectionniste. Le plus souvent, la limite est le temps budgété pour une session.

C'est la nature itérative du processus de mixage que nous utiliserons dans notre approche pour construire un algorithme itératif de la spatialisation.

5. Souvent un limiteur "BrickWall" est utilisé, un limiteur avec capacité LookAhead intégrée qui permet d'anticiper le dépassement de niveau maximal de sortie et donc d'avoir une vraie limitation au niveau maximal fixé.

2.2 Que peut-on automatiser ?

Avec le retour des techniques d'intelligence artificielle et les capacités révolutionnaires des nouvelles technologies informatiques, de nombreuses tâches qui étaient auparavant impossibles à résoudre ont désormais des solutions complètes, robustes et assez faciles à mettre en œuvre.

L'apprentissage profond⁶ est l'une des tendances les plus révolutionnaires de la dernière décennie. Vous pouvez lire une brève introduction à l'intelligence artificielle y compris l'apprentissage profond dans Annexe B.

Nous mentionnons ces technologies ici pour une raison précise, car un grand nombre de petites automatisations qui pourraient être utiles peuvent être mises en œuvre à l'aide de ces approches. Nous en présentons quelques-unes dans la suite.

Une fois encore, nous tenons à souligner que cette section décrit des automatisations que nous ne réalisons pas dans le cadre de ce travail et de notre étude, mais qui pourraient être réalisées en conjonction avec les dernières avancées technologiques. Chaque fois que ce sera possible, nous indiquerons les publications dans lesquelles de telles automatisations ont été tentées.

- **Convertir la pseudo-stéréophonie en monophonie**⁷. Il est pratique d'analyser automatiquement les fichiers audio, de déterminer leur contenu stéréophonique, pseudo-stéréophonique ou monophonique et de les convertir au bon format. De cette façon on peut réduire considérablement l'utilisation du processeur, utiliser moins de voies, économiser de l'espace sur un disque dur ou permettre une meilleure organisation de la session. La mise en œuvre d'un tel utilitaire est très simple, il suffit de comparer les spectres d'amplitude des canaux gauche et droit, et pas les signaux des voies gauche et droite, étant donné que la piste stéréophonique n'est pas nécessairement centrée.
- **Classifier des instruments de musique et dénommer des pistes**. Il est très pratique d'avoir un système de dénomination des pistes organisé et unifié, car on s'y habitue et il devient plus facile de travailler avec. Les méthodes d'apprentissage profond nous permettent de construire des classificateurs d'instruments de musique assez précis sur la base de l'analyse du fichier audio. Pour leur mise en œuvre on peut s'appuyer sur les modèles décrits dans [Fuhrmann, 2012; Mahanta et al., 2021]. Ainsi, les résultats de la classification des fichiers audio, complétés au besoin par le nom du fichier audio lui-même, peuvent servir de base pour nommer une piste dans la session créée dans le DAW.
- **Identifier le style et le sous-style d'un morceau de musique**. Les méthodes d'apprentissage profond se caractérisent par leur robustesse et leur faible sensibilité aux interférences, propriété qui permet de déterminer à quel style [Agrawal and Nandy, 2020; Pimenta-Zanon et al., 2021] appartient un morceau sans disposer d'un mixage final et même éventuellement d'analyser le sous-style auquel il appartient [Hsu et al., 2021]. Pour améliorer la précision de l'identification, il est raisonnable d'ajouter aux données d'entrée les informations sur des instruments de musique qui seront déjà connus grâce à l'étape précédente décrite ci-dessus. La détection précise du style nous permettra ensuite d'ajuster correctement la fonction objectif, celle qui sera au centre du processus d'optimisation, ainsi que les contraintes prises en compte dans le modèle d'optimisation itérative que nous prévoyons d'utiliser.

6. Deep Learning en anglais

7. <https://www.soundizers.com/>

- **Identifier la fonction de l'instrument.** Par instrument, ici comme ailleurs, on entend non seulement l'instrument de musique, mais aussi la voix. Une classification assez simple mais efficace du rôle de l'instrument dans la composition peut être proposée : solo, rythme, texture, effet. Et plus tard, en fonction du rôle de l'instrument, des exigences spécifiques de regroupement et de traitement seront formées pour cette piste. Ce qui va améliorer la qualité du mixage automatique.
- **Grouper les instruments.** Les connaissances acquises lors des étapes précédentes sur les noms des instruments, le style de musique et le rôle de chaque instrument dans le mixage nous permettent de regrouper automatiquement les pistes en groupes et sous-groupes en fonction de ces informations ainsi que des pratiques et habitudes des ingénieurs du son propres à un style [Tot, 2018]. Les effets temporels (délais et réverbérations), utilisés pour construire des plans sonores et remplir l'espace sonore, peuvent également être ajoutés lors de la même étape.

Le regroupement, en plus d'avoir un effet positif sur les performances en structurant le projet et en libérant les ressources du processeur grâce au traitement par groupe, présente un avantage très important pour l'amélioration potentielle des approches d'optimisation proposées dans ce travail. En effet, l'utilisation du principe "diviser pour mieux régner" permettra de traiter des projets de très grande envergure, en résolvant séparément les problèmes d'optimisation spatiale pour chaque groupe. Nous reviendrons sur ce point, plus loin dans le mémoire, de manière plus détaillée.

- **Détecter des instruments similaires.** Très souvent, les instruments dont le spectre est similaire ont tendance à s'étaler davantage dans la scène stéréophonique [Pestana, 2013; Pestana and Reiss, 2014; Tom et al., 2019]. Il est donc très important de détecter la similarité spectrale des instruments afin de leur apporter des corrections de distance relative dans l'espace sonore pour réduire les problèmes de masquage.

Techniquement, une telle tâche est très facile à réaliser puisqu'il suffit d'appliquer une fonction de similarité à chaque couple d'instruments et de construire une matrice de similarité sur la base de ces métriques.

- **Identifier les doublons** Cette situation peut se présenter pour les voix, la guitare, les autres instruments ainsi que pour les situations où la même source sonore a été enregistrée simultanément sur plusieurs pistes.

Premièrement, l'identification des prises de son est utile pour les regrouper, de sorte que l'ingénieur du son puisse facilement sélectionner, supprimer ou couper les prises non désirées et les conserver pour un montage ultérieur.

Deuxièmement, très souvent, surtout pour l'enregistrement de la guitare dans la musique moderne, un nombre pair de prises de son de guitare est enregistré pour créer la largeur du mixage, ou le "mur du son", puis, ensuite ces prises de son sont réparties symétriquement aux extrémités de la scène stéréophonique.

Troisièmement, il faut identifier les cas de "layering"⁸, pour lesquels des parties identiques sont jouées par instruments avec des timbres différents mais complémentaires, par exemple en fréquence. Le résultat obtenu correspond alors à un nouveau timbre composite avec les propriétés désirées.

Quatrièmement, on peut considérer le cas où la source sonore unique est enregistrée plusieurs fois par différents chemins de signaux, cas particulier de layering. Par exemple, on peut considérer une guitare basse qui est enregistrée au travers d'une ligne directe et, simultanément, le son direct qui est capté par des microphones disposés devant (voire derrière) un amplificateur de guitare combo.

Techniquement, les problèmes 1, 2 et 4 peuvent être résolus en sélectionnant des seuils de similarité adaptés pour chacun de ces cas, en tenant compte du type d'instrument de musique et grâce à une analyse spectrale de ce signal. Le troisième problème consiste à identifier la similitude du motif musical, de la texture ou du rythme, et à faire des hypothèses s'agissant du layering d'instruments. Pour cette tâche, les méthodes d'apprentissage profond qui sont utilisées pour extraire des motifs musicaux, comme la détection de plagiat, sont plus appropriées [Schindler et al., 2020; Das and Kolya, 2020; He et al., 2021].

- **Démasquer par égalisation automatique.** En pratique, le problème du masquage des fréquences est le plus souvent résolu par l'utilisation d'un égaliseur. Afin d'automatiser ce processus la solution la plus simple et peut-être la plus efficace est décrite dans [Hafezi and Reiss, 2015].

Il existe également des approches plus sophistiquées utilisant des modèles psychoacoustiques⁹ qui tiennent compte de la physiologie du système auditif humain [Herre and Dick, 2019; Glasberg and Moore, 2005], mais ces modèles sont assez complexes à mettre en œuvre et très éloignés de l'expérience musicale d'un ingénieur du son, de sorte que leur utilisation ne semble se justifier que pour des fins scientifiques et expérimentales.

- **Ajuster la dynamique.** De nombreux facteurs influencent les décisions en matière de compression, comme le style, la rigidité ou la douceur du son, qui sont à leur tour dictés par les préférences artistiques ou les exigences du genre. Pour toutes ces raisons, cette opération ne doit pas être engagée au stade initial. Ceci étant, plusieurs tentatives ont été faites pour automatiser la compression [Giannoulis et al., 2012, 2013; Ma et al., 2015].

8. superposition des sons

9. <https://amtoolbox.org/>

Pour cette question, nous proposerons une autre approche de compression automatique qui correspondra au problème d'optimisation d'"emballage par boîtes tridimensionnelles"¹⁰ [Hu et al., 2017]. Dans cette approche, on supposera que le son d'une piste sera "emballé" par un compresseur dans une boîte virtuelle 3D, dont les dimensions seront estimées à l'aide de mesures spéciales [Vickers, 2001]. Et, la tâche d'optimisation consistera à minimiser le volume de l'intersection de toutes les boîtes.



10. 3d packing optimization problem

2.3 Les habitudes des ingénieurs du son

L'idée d'automatiser le processus de mixage est étroitement liée au constat que les ingénieurs du son, avec une grande expérience du mixage, procèdent par habitude pour certaines des opérations. Certaines opérations typiques et routinières passent dans le subconscient et commencent à être réalisées de manière plus au moins automatique.

Nous avons donc choisi d'utiliser un questionnaire pour tenter d'identifier ces habitudes, et principalement celles liées à la construction de l'espace sonore puisque notre travail portera exclusivement sur cet aspect. Cela signifie que ce questionnaire ne constitue pas l'objectif principal de ce travail mais juste un outil complémentaire permettant de clarifier les hypothèses déjà formulées. Ce questionnaire constitue aussi une façon d'entamer un dialogue avec des ingénieurs du son en exercice, afin d'obtenir leur avis sur la nécessité et l'utilité des développements considérés dans ce mémoire.

Je tiens à mentionner qu'il existe des thèses assez récentes et sérieuses [Pestana, 2013; De Man, 2017; Tot, 2018] qui se sont penchées plus en profondeur sur les méthodes de travail des ingénieurs du son et qui présentent des analyses plus approfondies et des statistiques plus étendues que ce qui a pu être déduit grâce à l'analyse du questionnaire.

Le questionnaire a été créé à l'origine en  français [Vital, 2021a], puis traduit en  russe [Vital, 2021b]. L'objectif premier consistait à étudier le travail des ingénieurs du son en France et en Russie. Par la suite, ce questionnaire pourrait être également traduit en anglais afin d'enquêter auprès du public des ingénieurs du son anglophones.

Le questionnaire porte sur les opérations de base du mixage musical et les personnes questionnées ont la possibilité d'exprimer librement leurs pensées puisque la plupart des questions sont ouvertes. Ce type de question a été spécifiquement choisi pour permettre aux ingénieurs du son une totale liberté d'expression, sans les limiter en leur posant des questions fermées. Cela a permis d'observer quelles sont les opérations qui sont réalisées avec attention, donc de manière consciente, et celles qui sont réalisées de manière très souvent automatiques et qui ont été "repoussées" dans le subconscient des ingénieurs du son.

L'invitation à remplir le questionnaire a été effectuée par le biais d'une publication sur un forum thématique, mais, une trentaine de personnes ont également été contactées personnellement pour leur demander leur avis.

Nous avons cherché à obtenir les réponses d'un maximum d'ingénieurs du son ayant une grande expérience, mais, des mixeurs débutants ont aussi répondu à nos questions.

L'expérience moyenne professionnelle des ingénieurs du son qui ont répondu est de l'ordre de 20 ans. Le questionnaire était totalement anonyme, mais, ceux qui le souhaitaient, pouvaient laisser leurs coordonnées ainsi que leur accord pour que leur nom soit cité dans ce mémoire. Les personnes qui ont accepté que leur nom soit mentionné figurent dans l'Annexe A.1.

Certains des ingénieurs du son m'ont contacté, soit pendant qu'ils répondaient aux questions, soit juste après, et, nous avons pu avoir une conversation libre. C'était très agréable de ressentir ce désir d'aider et leur volonté de partager leurs propres expériences.


Une vue d'ensemble du processus permet de dégager l'existence de trois groupes ("clusters") de réponses, tous de nature différente : 1) les débutants, 2) les indépendants et 3) les personnes travaillant sur un grand nombre de projets en studio.

Parmi les spécialistes, on trouve des personnes spécialisées dans la diffusion (40%) et des personnes spécialisées dans le mixage musical (95%). Pour les personnes spécialisées dans la diffusion, les ingénieurs du son travaillent principalement pour la télévision, la radio ou dans la diffusion sur Internet.

En termes de genres musicaux, ils travaillent principalement en musique pop, en rock, en hip-hop, en jazz comme en musique classique.

S'agissant des fonctions occupées, 95% des mixeurs ayant répondu sont impliqués dans le mixage, dont 85% font aussi la prise de son, 60% travaillent dans le mastering, et 23% sont en même temps des directeurs artistiques.

Dans ce questionnaire, nous avons accordé une attention particulière aux questions de spatialisations. 80% d'ingénieurs du son estiment qu'une bonne spatialisation (volume et panorama) est absolument essentielle pour obtenir un mixage réussi.

Dans la suite, nous passons en revue les principales questions posées dans le questionnaire en mettant en évidence, voire en citant, certaines des réponses qui nous semblent représentatives de l'enquête et essentielles par rapport au sujet traité. Les réponses détaillées sont disponibles sous forme électronique sur la page internet  *lien* [VITAL, 2022].

2.3.1 Comment est-ce que vous commencez votre session de mixage ?

russe: «с тишины»
français: "du silence"

russe: «прослушивание разной музыки»
français: "écoute de différentes musiques"

russe: «с прослушивания записанного материала»
français: "de l'écoute du matériel enregistré"

russe: «пытаюсь понять материал, его характер, настроение»
français: "j'essaie de comprendre le matériau, son caractère, son humeur"

"Préparation de session : organisation des pistes, vérification de la mise à plat, structure, importation template de mixage"

"Début de mixage : réglage de la tranche master (EQ, COMP, Limiteur) afin de fixer le cadre esthétique de travail global avant de rentrer dans le détail des pistes"

Un ingénieur du son très célèbre a déclaré lors d'une conversation verbale : "Je rentre en transe, et quand j'en sors, le morceau est prêt."

2.3.2 Préparation, édition et nettoyage des pistes (voix, bruits,...)

“Le nettoyage se fait en amont au montage.”

“Je réalise l’édit avant de mixer, en gardant les différentes étapes en playlist. Pour les bruits de bouche je corrige au fur et à mesure du mixage selon s’ils sont perceptibles ou non.”

russe: «Грубая очистка вырезание ненужных пауз перед началом сведения, тонкая очистка фэйд ины и ауты, шумоподавление, это всё в финале сведения , после эквализации и компрессии»

français: “Un nettoyage grossier supprimant les pauses indésirables avant le mixage, un nettoyage fin des entrées et sorties en fondu, une réduction du bruit, tout cela se trouve au stade de mixage final, après l’égalisation et la compression.”

russe: «По мере необходимости, как правило не чищу без надобности. Конечно же компилирование тейков происходит до сведения»

français: “Pour ce qui concerne le contenu nécessaire, en règle générale, je ne le nettoie pas inutilement. Bien sûr, la compilation des prises se fait avant le mixage.”

russe: «Редактирование, по возможности стараюсь делегировать»

français: “L’édition, si possible, j’essaie de la déléguer”

Ayant parlé à plusieurs ingénieurs du son, j’ai remarqué qu’ils ont tendance à déléguer ces opérations si possible, surtout s’ils doivent travailler sur un nombre important de projets.

2.3.3 Balance

“Je reproduis la balance de la mise à plat, qui sert de référence tout au long du mixage, et revérifie régulièrement que mon équilibre y reste conforme ou alors si modification de niveau celle-ci doit être justifiée artistiquement/esthétiquement.”

russe: «Первым делом»

français: “La première chose à faire”

russe: «Использую референс»

français: “J’utilise une référence”

russe: «Балансы потом»

français: “La balance après”

russe: «Слушаю в моно баланс»

français: “La balance j’écoute en mono”

- russe:* «Всё зависит от стиля и жанра, но в случае с поп, джаз и рок порядок такой: Кик, Снэр, Хайхет, Томы, Овэрхэды, Бас, Ритм гитары, Брас секция, Пэды, Соло гитары, Соло клавиши и другие сольные партии если есть, Вокал, Бэк вокал»
- français:* “Tout dépend du style et du genre, mais dans le cas de la pop, du jazz et du rock, l’ordre est le suivant : Kick, Caisse claire, Charleston, Toms, Overheads, Basse, Guitares rythmiques, Section cuivres, Pads, Guitares solos, Claviers solos et autres instruments solos s’il y en a, Voix, Chœurs.”
- russe:* «Строятся под панораму»
- français:* “La balance dépend de la scène sonore recherchée”
- russe:* «Сначала принципиальные решения по выраженной компрессии, а потом балансы»
- français:* “D’abord les solutions principales pour les compressions prononcées, puis les balances.”
- russe:* «Баланс не может выстраиваться до тех пор, пока каждый компонент микса не зафиксирован по динамике, и не выполнена частотная демаскировка. После выполнения динамической и спектральной обработки каждого компонента микса выполняется «предварительный баланс».»
- français:* “La balance ne peut pas être ajustée tant que chaque composant du mixage n’est pas fixé s’agissant de la dynamique et que le démasquage des fréquences n’est pas effectué. Une fois que la dynamique et le traitement spectral de chaque composant du mixage sont terminés, un "pré-équilibre" est effectué.”
- russe:* «Расставляю громкости, в зависимости от тесситуры. Низкие партии громче, высокие тише. Тихие места громче. Применяю динамическую обработку, после неё восстанавливаю спектр эквалайзером.»
- français:* “Réglage du volume, en fonction de la tessiture. Les parties basses rendues plus fortes, les parties aigües rendues plus faibles. Des parties plus silencieuses rendues plus fortes. Appliquer un traitement dynamique, puis restaurer le spectre avec un égaliseur.”
- russe:* «На этапе прослушивания сырого проекта расставляю приоритеты, стараюсь понять, что главное, а что второстепенное, соответственно и делаю баланс»
- français:* “En écoutant le projet brut, j’établis des priorités, j’essaie de comprendre ce qui est important et ce qui ne l’est pas, et j’en déduis une balance.”
- russe:* «Выстраиваю общий баланс так, чтобы всё само приемлемо играло, затем постепенно добавляю автоматизацию в процессе, голос, бас, солирующий инструмент как правило остаются на неизменном уровне»
- français:* “Je construis la balance générale pour que tout puisse jouer de manière acceptable, puis j’ajoute progressivement l’automatisation dans le processus ; la voix, la basse, l’instrument principal restent généralement au même niveau.”

Comme nous pouvons l’observer, les approches de la construction d’une balance sont complètement différentes. Pour certains, c’est une priorité absolue, pour d’autres elle est intégrée en dernier lieu. Pour certains, il existe une séquence claire d’actions, tandis que pour d’autres tout dépend du

moment. Dans presque tous les cas, la balance dépend fortement de la scène sonore recherchée, de l'utilisation des traitements spectraux pour le démasquage, de la compression, et bien sûr de l'intention artistique et de l'esthétique du son. Autrement dit, la balance est, d'une part, le fondement d'un mixage réussi et, d'autre part, une fonction multifactorielle dépendant d'autres processus de mixage.

2.3.4 Spatialisation (pans)

russe: «Пространственная обработка производится для выполнения демаскировки компонентов микса по плановости. Следует отличать пространственную демаскировку и панорамную. При работе с плановостью, принимается за данность то, что диффузное поле одинаково во всех точках пространства которое эмулируется. При работе с панорамной демаскировкой учитывается то, что любой источник локализуется в базе с точностью до 16 градусов если он статичен, и с точностью до 3-4 градусов, если он находится в «движении». При повышенной плотности компонентов базы, прописывается панорамная автоматизация начинающая с высокочастотных составляющих. Низкочастотные составляющие 50-150 гц не панорамируются и не посылаются для формирования на основе них диффузного поля. Источники ниже 50 гц, отсекаются для соблюдения технических аспектов трактов прослушивания.»

français: “Le traitement spatial est effectué pour démasquer les composants du mixage par plan. On distingue le démasquage spatial du démasquage panoramique. Lorsqu'on travaille avec le plan, on suppose que le champ diffus est le même en tous les points de l'espace qui est émulé. Lorsque l'on travaille avec le masquage panoramique, on tient compte du fait que toute source est localisée dans la base avec une précision de 16 degrés si elle est statique et avec une précision de 3-4 degrés si elle est en "mouvement". Si la densité des composants de base est augmentée, l'automation panoramique est prescrite en partant des composants à haute fréquence. Les composantes basse fréquence 50-150 Hz ne sont pas balayées ou envoyées pour former un champ diffus basé sur elles. Les sources inférieures à 50 Hz sont coupées pour respecter les aspects techniques des voies d'écoute.”

russe: «Как ни странно, я не уделяю ключевого внимания панораме, мне кажется, что микс должен хорошо звучать в моно, панорамирование - да, можно просто слегка развести инструменты влево-вправо, но для меня это не главное»

français: “Curieusement, je ne prête pas une grande attention au panorama, je pense que le mixage doit bien sonner en mono, le panning - oui, on peut juste répartir les instruments légèrement à gauche et à droite, mais ce n'est pas l'essentiel pour moi”

russe: «После основных балансов»

français: “Une fois que la balance principale est faite”

russe: «Соответственно жанру и наличию инструментов»

français: “Selon le genre et la disponibilité des instruments”

- russe:* «Начинаю с панорам и уровней согласно задуманному плану, при необходимости корректирую»
- français:* “Je commence par les panoramas et les niveaux selon le plan conçu, en corrigeant si nécessaire”
-
- russe:* «Пространство в конце»
- français:* “Spatialisation à la fin”
-
- russe:* «Часто с автоматизацией»
- français:* “Souvent avec l’automatisation”
-
- russe:* «После сведения в моно, ширина и глубина сцены зависит от жанра и состава ансамбля, но в любом случае чтобы напоминало живое расположение музыкантов»
- français:* “Après le mixage en mono, la largeur et la profondeur de la scène dépendent du genre et de la composition de l’ensemble, mais dans tous les cas, elle doit ressembler à un positionnement en direct des musiciens”
-
- russe:* «Как в симфоническом оркестре»
- français:* “Comme dans un orchestre symphonique”
-
- russe:* «Сtereo оставляю стерео, контролирую ширину и панораму в конце сессии микширования. Из моно делаю стерео, используя дубльтреки и стерео эффекты. Сам при разработке грува использую радикальное стерео и подвижные в панораме элементы. Часто переключаю моно - стерео контроль для совместимости и проверки текущего баланса.»
- français:* “Stéréo je laisse la stéréo, je contrôle la largeur et le panorama à la fin de la session de mixage. Je fais du stéréo à partir de mono en utilisant des pistes doubles et des effets stéréo. Lorsque je développe mon propre "groove", j'utilise des éléments radicalement stéréo et mobiles en panorama. Je passe souvent du contrôle mono au contrôle stéréo pour des raisons de compatibilité et pour vérifier la balance actuelle”

On peut définitivement dire que le panning est très étroitement lié au genre musical. De plus, presque tous les ingénieurs du son vérifient la compatibilité en monophonique. Pour certains, l’image stéréophonique est plus importante que pour d’autres. Et de préférence, cette image doit correspondre au positionnement effectif des musiciens sur une vraie scène. L’automatisation des panoramiques est une décision technique importante qui est souvent utilisée. Et dans certains cas, on opte pour un panoramique diffus, qui permet de localiser l’instrument avec plus de précision lorsqu’il y a une forte densité d’instruments dans le mixage. Il est important de se rappeler qu’il est inutile d’appliquer un panoramique pour les basses fréquences.

2.3.5 Égalisation

“Utilisation quasi systématique (en phase naturelle ou linéaire selon le degré de sensation de précision souhaité dans les extrêmes, parfois l’imprécision relative liée au fait de ne pas travailler en phase linéaire peut être intéressante)”

- russe:* «Сначала очевидные крупные движения – обрезка низа, резонансы, маскировка и коррекция тембра в плюс в процессе микса»
- français:* “Tout d’abord, les grands mouvements évidents – réglage des basses, résonances, masquage et correction du timbre en positive (amplifications) durant le processus du mixage”
-
- russe:* «Обязательно фильтры низких и высоких, чтобы заранее избежать частотных конфликтов и маскирования»
- français:* “Filtrer les basses et les hautes fréquences pour éviter les conflits de fréquences et les masquages au préalable.”
-
- russe:* «Устранение частотных конфликтов»
- français:* “Élimination des conflits de fréquences”
-
- russe:* «Добиваюсь в начале, чтоб каждый сигнал имел спектр с преобладающей по амплитуде основной гармоникой и не превышающими её тембральными и выше тоже, с помощью анализеров. В случае конфликтов, выбираю основную партию и делаю под неё провал в конкурирующей, часто сайдчейном к многополосному компрессору»
- français:* “Tout d’abord, je fais en sorte que chaque signal ait un spectre où la fréquence fondamentale prédomine par son amplitude et où les harmoniques du timbre ne la dépassent pas, et ce, à l’aide d’égaliseurs. En cas de conflit, je choisis une partie de base et je l’égalise dans une partie concurrente, souvent à l’aide d’un compresseur multibande”
-
- russe:* «Финальная эквалализация производится методом «инвертируемого колокола» когда спектр делится на низ, средний низ, средний верх, верх. В каждом диапазоне используя управляющую поверхность (не мышь) одновременно крутятся сочетания усиления/ослабления, частоты, добротности, при нахождении максимума (центра) не приемлемого спектрального участка при повышенном усилении усиление инвертируется в ослабление. Процедура повторяется для всех разделенных участков спектра (от 4х до 6-8)»
- français:* “L’égalisation finale est effectuée par la méthode de la "cloche inversée", où un spectre est divisé en une gamme basse, une gamme moyenne, une gamme moyenne et une gamme haute. Dans chaque bande, à l’aide d’une surface de contrôle (et non d’une souris), des combinaisons de gain/atténuation, fréquence, facteur Q sont tournées simultanément, et lorsqu’un maximum (le centre) d’une zone spectrale inacceptable est trouvé, on insère une atténuation lorsque le gain est augmenté. La procédure est répétée pour toutes les parties séparées du spectre (de 4 fois à 6 ou 8)”
-
- russe:* «Стараюсь вначале построить баланс без эквалазации, но если нужно – применяю. Голос и барабаны требуют эквалазации почти всегда, но радикальных движений избегаю, максимум 4-6 дБ»
- français:* “J’essaie de construire un équilibre sans égalisation au début, mais si nécessaire je l’applique. Les voix et la batterie ont presque toujours besoin d’une égalisation, mais j’évite les mouvements radicaux, maximum 4-6 dB”

L'égalisation est l'une des opérations de mixage les plus fréquemment utilisées. Elle corrige généralement le timbre et résout les conflits de fréquence. Une bonne balance spectrale et une spatialisation bien faite minimisent les problèmes d'égalisation. L'égalisation est souvent utilisée comme un élément artistique pour créer une certaine coloration. L'égalisation est également appliquée pour créer des plans. Chaque ingénieur du son développe ainsi son approche personnelle de l'égalisation.

2.3.6 Dynamique et compression, compression parallèle

“Suivi de dynamique à l'automation, peu de compression”

“Beaucoup d'utilisation de compression parallèle (mixage bus, bus kick, batterie, parfois les voix). Utilisation du side chain (autant en compression globale qu'en multibande). Compression : au cas par cas selon la nécessité. Travaillant souvent dans un contexte de musiques actuelles avec productions déjà compressées ce ne sont pas les outils que j'utilise le plus”

russe: «Максимально сохраняя естественность звучания инструментов, иногда клипер предпочтительнее, по ситуации»

français: “Maintenir le son naturel des instruments autant que possible, parfois le clipper est préférable, selon la situation”

russe: «По необходимости, часто параллельную вообще не использую»

français: “Selon les besoins, souvent je n'utilise pas du tout la compression parallèle”

russe: «Все должно хорошо звучать без финальной компрессии на мастере. Значит доводка на самих треках»

français: “Tout devrait sonner correctement sans compression finale sur le master. Donc, mise au point sur les pistes elles-mêmes”

russe: «Использую всегда, если не касается классической музыки»

français: “Je l'utilise toujours si ce n'est pas de la musique classique”

russe: «Не использую параллельную компрессию в DAW. Использую групповую на однородные партии. Синхронизирую её с текущим темпом, вычисляя чётную гармонику цепью деления темпа в миллисекундах на 2, 4, 8, 16... Получаю число для диапазона атаки, затухания калькулятором. Так же поступаю со многими настройками, но не со всеми по чётному принципу, с некоторыми – триолями...»

français: “Je n'utilise pas la compression parallèle dans le DAW. J'utilise la compression de groupe sur des pistes homogènes. Je la synchronise avec le tempo actuel en calculant un nombre pair d'harmoniques en divisant le tempo en millisecondes par 2, 4, 8, 16... J'obtiens un chiffre pour la portée de l'attaque et l'atténuation avec une calculatrice. Je fais la même chose avec de nombreux paramètres, mais pas tous en nombre pair, avec certains – triplés...”

- russe:* «Для выбора границы эффективной атаки, сигнал при соотношении 10-20:1 зажимается трешхолдом до значения, при котором исходная динамика теряет импульсы. В этом положении атака отстраивается путем поднятия от нуля, до предела, в котором изменения не перестают читаться, после чего трешхолд отпускается до момента информативной индикации о приемлемой глубине (влияющей на потерю динамического диапазона)»
- français:* “Pour sélectionner la limite d’une attaque efficace, le signal à un rapport 10-20:1 est bridé avec un threshold à la valeur à laquelle la dynamique originale perd son élan. Dans cette position, l’attaque est ajustée en augmentant de zéro jusqu’à la limite où les changements ne cessent pas d’être intelligibles, après quoi le threshold est relâché jusqu’à une indication informative d’une profondeur acceptable (affectant la perte de gamme dynamique)”
- russe:* «Часто применяю параллельную компрессию на некоторых барабанах и голосе, но стараюсь скорее подчеркнуть атаки, нежели управлять громкостью.»
- français:* “J’applique souvent une compression parallèle sur certaines batteries et voix, mais j’essaie d’accentuer les attaques plutôt que de contrôler le volume”

La pratique montre qu’il est préférable de ne pas exagérer avec la compression, mais de ne l’appliquer que là où il y a un réel besoin. Si possible, il faut préférer l’automation à la compression. La compression parallèle peut être utilisée ponctuellement pour accentuer les attaques. De plus, la compression peut participer à la création des plans (les sons plus éloignés sont plus compressés, et en contrôlant les transitoires). Il ne faut pas oublier les applications artistiques de la compression. La sélection des paramètres du compresseur est une opération correspondant presque à un art quand elle est effectuée le plus souvent à l’oreille, mais, certaines personnes préfèrent utiliser une calculatrice.

2.3.7 Regroupement de pistes

“Systématique, et sous-bus (auxiliaire) par fonction musicale afin de mixer les éléments d’arrangement (domaine "musical") et non les contributions individuelles (plus du domaine "sonore)”

- russe:* «Большие группы (перкуссия/гармония и т.п.) не люблю. Локальные группы - гитарный дабл, бэквокая пачка»
- français:* “Je n’aime pas les grands groupes (percussion/harmonie, etc.). Groupes locaux - double guitare, backing pack”
- russe:* «Ударные, Ритм гитары ритм клавиши, Бэк вокал, Брасс, Сольные партии»
- français:* “Batterie, guitare rythmique, claviers rythmiques, chœurs, cuivres, parties solos”
- russe:* «Ударные, басы, синты, эффекты, вокалы, гитары»
- français:* “Batterie, basse, synthétiseurs, effets, voix, guitares”
- russe:* «Группирую однородные партии в группу для разгрузки DSP и экономии времени настройки»
- français:* “Je regroupe les pistes homogènes en groupes pour soulager le DSP et gagner du temps de réglage”

russe: «Делаю, конечно, так удобнее работать»
français: “Je le fais, bien sûr, car c’est plus pratique de travailler de cette façon”

russe: «При подготовке сессии»
français: “Lors de la préparation d’une session”

Le regroupement de pistes est utilisé par tout le monde. Il est effectué pendant la phase de préparation de la session. Le regroupement permet de structurer le projet, de décharger le CPU ou DSP et de gagner du temps en minimisant le nombre de paramètres à régler. Souvent, les ingénieurs du son ont déjà un modèle de regroupement personnel établi des groupes, qui peuvent varier d’un genre musical à l’autre. Le regroupement apporte du confort au travail.

2.3.8 Pistes monophoniques VS stéréophoniques, mixage en monophonique et/ou en stéréophonique

“mixage en stéréophonique, évidemment”

“Conservation quasi systématique de la nature des pistes. Une piste stéréo (overhead, synthé, piano, couple en général, etc ...) sera mixée en stéréo et une piste mono en mono. Il m’arrive de découpler une piste stéréo afin de corriger un problème spécifique sur l’un des canaux. Du point de vue du strict mixage respect de la nature stéréo ou mono de la piste selon le même principe de conservation de fonction musicale de l’item précédent (groupage de piste)”

russe: «Регулярно заужаю стерео в минутах для лучшего позиционирования»
français: “Ajustement régulier de la stéréo, en minutes, pour un meilleur positionnement”

russe: «Сведение в моно, выстраиваю глубину сцены в моно, а затем перехожу в стерео и расставляю всех по ширине сцены в стерео конечно, на дворе заканчивается 2021 год), но балансы могу иногда проверить в моно (выключаю одну колонку)»
français: “mixage en mono, construction de la profondeur de scène en mono, puis passage en stéréo et disposition de tous selon la largeur de scène, en stéréo bien sûr, c’est la fin de 2021), mais je peux parfois vérifier la balance en mono (en éteignant un haut-parleur)”

russe: «Сведение в моно»
français: “mixage en mono”

russe: «Сначала в моно затем в стерео»
français: “D’abord en mono, puis en stéréo”

russe: «Стерео (кому вообще в 2022 году нужно моно?)»
français: “Stéréo (qui a besoin de mono en 2022 ?)”

- russe:* «Сtereo, моносовместимо. Моно помогает найти баланс в процессе. Stereo создаёт бинауральный психоделический эффект. Создаю stereo для всех монотреков и проверяю моносовместимость отдельных партий, исправляю фазовые искажения на начальной стадии микширования – фокусирую микрофонные треки к нормальному stereo, вводя задержки к опережающим трекам. Кореллометр всегда на глазах вместе со спектрометром»
- français:* “Stéréo, mono-compatible. La mono aide à trouver une balance dans le processus. La stéréo crée un effet psychédélique binaural. Je crée une stéréo pour toutes les pistes mono et je vérifie la compatibilité mono des différentes parties, en corrigeant la distorsion de phase au stade initial du mixage – je concentre les pistes de microphones en stéréo normale en introduisant des délais sur les pistes principales. Un coréloètre est toujours en vue, ainsi qu’un spectromètre”
- russe:* «Дело вкуса. Конечно предпочтительнее stereo или работа с прослушиванием сигнала через dolby декодер. В mono не свожу, но иногда проверяю микс на моносовместимость»
- français:* “C’est une question de goût. Bien sûr, la stéréo est préférable, ou le travail avec le décodeur dolby. Je ne mixe pas en mono, mais je vérifie parfois la compatibilité de mon mixage avec le mono”
- russe:* «Так, как записаны, при необходимости корректирую stereo, проверяю фазировку. В mono не свожу, просто стараюсь сохранять моно-совместимость. Stereo-клавиши часто разделяю и использую один наилучшим образом записанный трек, если необходимо – применяю psp pseudo-stereo. Звуки, чьё значение велико, обычно идут в центр, а всякие тарелки, звонелки, клавишные, пэды – по краям. Иногда впрочем какой-то звук лучше, выразительнее звучит из одной колонки, но поскольку все нынче пользуются наушниками, принимаю это во внимание»
- français:* “J’utilise comme c’était enregistré, je corrige la stéréo si nécessaire, je vérifie la phase. Je ne mixe pas en mono, j’essaie juste de le garder mono-compatible. Je divise (split) souvent les pistes stéréo et j’utilise l’une des meilleures pistes enregistrées, si nécessaire, j’utilise la pseudo-stéréo psp (plugin). Les sons, qui sont les plus importants, vont généralement au centre, et toutes sortes de cymbales, carillons, des claviers et des nappes sur les bords. Parfois, cependant, un son est meilleur, plus expressif, provenant d’une seule enceinte, mais comme tout le monde utilise des écouteurs de nos jours, j’en tiens compte.”

La compatibilité monophonique reste un facteur important pour beaucoup de monde, et la musique est souvent écoutée à partir de téléphones portables. La tendance à écouter la musique avec des écouteurs est également à prendre en considération. Il est toujours vrai qu’il est plus facile d’obtenir une bonne balance en monophonique qu’en stéréophonique. Mais en même temps, on peut noter une tendance à travailler uniquement en stéréophonie, et à l’abandon de la compatibilité monophonique.

2.3.9 Masking et unmasking

“selon l’écoute, pas de procédé systématique”

- russe:* «Эквализация, сайдчейн»
français: “Égalisation, sidechain”

russe: «Эквалайзер, далее автоматизацией»
français: “Égaliseur, puis automation”

russe: «Конфликт. Сначала решается нужен ли конфликт вообще? Выключается ненужная партия. Если нужна, перемещается по тесситуре или эквализуется динамическим прибором – сайдчейн к многополосному компрессору от ведущей партии»

français: “Conflit. Il faut d’abord décider si le conflit est nécessaire ou non. Je désactive la piste non désirée. Si c’est le cas, elle est déplacée en tessiture, ou égalisée par un instrument dynamique – sidechain vers un compresseur multi-bandes à partir de la piste maître”

russe: «Частотной, пространственной и панорамной демаскировкой»
français: “Par démasquage fréquentiel, spatial et panoramique”

russe: «Выясняю, что главнее, оставляю естественное звучание этого, остальное, что не так важно, "кастрирую" эквалайзером»

français: “En déterminant ce qui est le plus important, je laisse le son naturel et je "neutralise" le reste, qui n’est pas si important, avec l’égaliseur.”

russe: «Если звук мешает песне, его можно не использовать вовсе»
français: “Si le son perturbe la chanson, on ne devrait pas l’utiliser du tout”

russe: «Плагин trackspacer или вручную»
français: “Plugin Trackspacer ou manuellement”

Comme on peut le constater, pour les problèmes de masquage, l’outil principal est l’égaliseur. Il n’y a pas beaucoup d’options pour résoudre le conflit. La solution la plus courante consiste à couper tout ce qui est "superflu", jusqu’à et y compris la piste qui "interfère". On peut également recourir à des méthodes plus spécifiques de démasquage, telles que la révision de la spatialisation ou le panoramique diffus. Un ajustement plus subtil est effectué par des automatisations et le side chain. Il existe également un plug-in dédié qui peut résoudre le problème du masquage en temps réel avec un égaliseur à 32 bandes (Trackspacer).

2.3.10 Utilisation du Master bus (esthétiques, rééquilibrage global en fréquences, ...)

“Première étape du mixage (EQ, Comp, Limiteur)”

russe: «Никак. На место шины НИЧЕГО. Это дело мастеринг инженера»
français: “Rien. Il n’y a RIEN sur le bus. C’est le travail d’un ingénieur de mastering”

russe: «Незначительное вмешательство на мастере, в основном многополосный компрессор в режиме эквалайзера и легкое лимитирование. Никогда не жму мастер в "кирпич"»

français: “Petites manipulations sur le master, principalement un compresseur multi-bande en mode EQ et une légère limitation. Je ne pousse jamais le master en "brique””

- russe:* «Нет, только чуть максимайзера и все»
français: “Non, juste un peu de maxizer et c’est tout”
- russe:* «В мастер ставится FFT компрессор с разрешением 4000-8000 гармоник и шаблоном естественного спада амплитуды от частоты, типа розовый шум, с небольшим коэффициентом 2,7 x 1. И пик лимитор в режиме слегка. В конце обрезной фильтр на тесситуру фонограммы, чтоб обрезать артефакты переработки»
français: “Sur le master, on met un compresseur FFT avec la résolution 4000-8000 harmoniques et un modèle de décroissance d’amplitude naturelle de la fréquence, comme un bruit rose, avec un petit facteur de 2,7 x 1. Et un limiteur de crête en mode léger. A la fin, un filtre trim sur la tessiture du phono pour couper les artefacts de traitement”
- russe:* «Если на мастер-шине требуется перебаланс частот, это свидетельствует о необходимости вернуться к работе над начальной и эффективной эквалаизацией»
français: “Si vous devez rééquilibrer les fréquences sur le bus maître, cela montre qu’il vous faut retravailler l’égalisation initiale de façon plus efficace”
- russe:* «На мастер шине у меня только анализаторы и лимитер. НО у меня всегда есть mixage BUS, и на него идут все треки микса, а он уже идет на мастер шину. Тут у меня стоит эмулятор консоли и склеивающий компрессор. Понятно, что это типа шило на мыло, но мне так удобнее, больше пространства для манипуляций и не надо создавать дорожку мастера в проекте, если понадобится делать автоматизацию громкости мастера»
français: “Sur le bus maître, je n’ai que des analyseurs et un limiteur. MAIS j’ai toujours le mixage BUS, et toutes les pistes du mixage y vont, et il va ensuite au master bus. Ici, j’ai un émulateur de console et un compresseur de colle. Bien sûr, c’est une sorte de casse-tête, mais c’est plus pratique pour moi, car j’ai plus d’espace pour la manipulation et je n’ai pas besoin de créer une piste master dans mon projet, si j’ai besoin d’automation du volume principal”
- russe:* «Нет...все при миксе»
français: “Non... tout est dans le mixage”
- russe:* «Использую при сведении фильмов, музыку свожу в аналоге с последующим мастерингом»
français: “Je l’utilise pour le mixage de films, le mixage de musique je le fais en analogique puis le mastering”
- russe:* «Лимитер для демо, больше никак. Бывает, автоматизирую мастер-фейдер, для большей выразительности при смене частей песни»
français: “Limiteur pour la démo, rien d’autre. Parfois, automation du fader principal, pour une meilleure expressivité lors des changements de parties du morceau”
- russe:* «Использую с лимитером и компрессией»
français: “Je l’utilise avec un limiteur et une compression”

Il convient de noter la grande variété d'approches de l'utilisation du bus maître. Souvent, déjà à la phase de préparation d'une session, l'égaliseur, le compresseur, le limiteur et les saturateurs sont placés sur le bus maître afin de travailler sur le mixage dans des conditions proches de "mastering". Cela permet de minimiser le nombre de surprises lors du mastering. Mais il y a aussi des défenseurs de l'idée que le bus maître doit être vide, et que le désir d'effectuer des ajustements sur le bus maître est le signe d'erreurs commises durant le processus de mixage. Le bus maître est souvent utilisé pour l'automation du volume principal d'un mixage pour plus d'expression.

2.3.11 Aspects esthétiques et artistiques

“Donnés par la mise à plat fournie avec le multipiste, libertés prises en concertation avec l’artiste et la réalisation”

russe: «Референсы заказчика, вкусовые временные пристрастия»

français: “Références des clients, goûts de préférences et du temps”

russe: «Много слушать хорошо сделанной и желательно фирменной музыки в разных жанрах, анализировать как меняются ширина и глубина сцены, компрессия и баланс в зависимости от жанра и стиля, приучаться слушать не только музыку а САУНД»

français: “Écouter beaucoup de musiques bien faites et de préférence des musiques de référence dans différents genres, analyser comment la largeur et la profondeur de la scène, la compression et l’équilibre changent en fonction du genre et du style, s’habituer à écouter non seulement de la musique mais aussi du SON.”

russe: «Должно быть сочно»

français: “Il faut que ce soit juteux”

russe: «Пространство и звучание инструментов в нём»

français: “L’espace et le son des instruments qui s’y trouvent”

russe: «Песня диктует все аспекты»

français: “La chanson dicte tous les aspects”

russe: «С того начинается и этим заканчивается. Вообще процесс регулируется объективно – по приборам, а контролируется субъективно на слух пошагово: стало лучше или хуже»

français: “C’est là que ça commence et c’est là que ça finit. En général, le processus est régulé de manière objective – par l’instrumentation, et contrôlé de manière subjective par l’audition : s’améliore ou empire”

russe: «Всегда делать так, чтобы самому нравилось, тогда и клиенту будет норм ;) Не без исключений, конечно, вкусы же у всех разные»

français: “Faites toujours les choses à votre goût, le client s’en portera bien aussi ;) Pas sans exceptions, bien sûr, car chacun a des goûts différents”

russe: «Иду за музыкой, стараюсь находить отдельные выразительные якоря. Мне нравится легкое звучание, так, чтобы микс было легко слушать, то есть не перекомпрессированный, с натурально звучащими высокими частотами»

français: “Je vais vers la musique, j’essaie de trouver des points d’ancrage expressifs individuels. J’aime que le son soit facile à écouter, qu’il ne soit pas surcompressé et que les aigus soient naturels”

Comme nous le voyons, il est important de rester dans les limites des exigences et des souhaits du client, de rester dans l’air du temps, de rendre le son cohérent avec le style et le genre de musique, de trouver le point fort et de le mettre en valeur, et bien sûr d’aimer son travail et sa création.

2.3.12 Aspects techniques et technologies

“Pro Tools 2021 / Sommateur externe Neve 8816 / Chaine Master Analogique Wes Audio à recall numérique (émulations EQ Hyperion , Comp Dione, Comp Rhéa) / Convertisseur Master Burl B2 Bomber”

russe: «Знаю как эмулировать глубокое пространство с помощью двух ревербераторов»

français: “Je sais comment émuler un espace profond avec deux réverbérateurs”

russe: «Начинаю сведение с уже поставленным на микс шине склеивающим компрессором, и свожу через него всегда. Это позволяет не удивляться от развала микса в конце сведения, когда все готово, ставишь на микс шину компрессор... и все летит к черту. А так изначально при правильной настройке слышу "финальный"звук микса»

français: “Je commence à mixer avec un compresseur de colle déjà sur le bus de mixage, et je mixe toujours à travers lui. De cette façon, on n’est pas surpris par des problèmes du mixage à la fin du mixage, lorsque vous avez terminé et que vous mettez un compresseur sur le bus de mixage... tout part en vrille. Ainsi, au départ, avec la bonne configuration, je peux entendre le son "final" du mixage”

russe: «Чем проще и информативнее плагин, тем легче с ним работать на управляющей поверхности»

français: “Plus un plugin est simple et informatif, plus il est facile de travailler avec lui sur une surface de contrôle”

russe: «Люблю реальные (железные) устройства, ламповую обработку»

français: “J’aime les vrais appareils (hardware analogique), le traitement par des tubes”

russe: «Делать быстро, пока не замылился слух»

français: “Je le fais rapidement, avant que l’oreille ne devienne floue”

russe: «Нет таких, пожалуй что лучше работает на песню, то и выигрывает»

français: “Non, en fait ce qui marche le mieux sur une chanson gagne”

russe: «Из нестандартных - люблю префейдерные посылы на ревера»

français: “De l’inhabituel – j’aime bien les sends pré-fader de réverbération”

russe: «Люблю внешние синтезаторы, микшеры, и обработку, стараюсь сделать звук до того, как он попадает в компьютер»

français: “J’aime les synthés externes, les mixages et le traitement analogique, j’essaie de faire le son avant qu’il n’entre dans l’ordinateur”

russe: «Минимум компрессии, не люблю ее. Насыщение гармониками микса»

français: “La compression minimale, je ne l’aime pas. Enrichir le mixage avec des harmoniques”

russe: «FFT, спектральные динамические процессоры»

français: “FFT, processeurs dynamiques de spectre”

- russe:* «AI плагины, smartcomp + smarteq, Eventide elevate, плагины спектральной обработки, iZotope RX»
français: “Plugins IA (intelligence artificielle), smartcomp + smarteq, Eventide elevate, plugins de traitement spectral, iZotope RX”

2.3.13 Utilisation de modèles (pré-réglages, styles, workflow, ...)

“Non”

“Systématique, template préparé avec auxiliaires de spatialisation (reverb, delay), routing vers le sommateur externe, tranche master”

- russe:* «Да есть, но свои наработки, к определенным вокалистам»
français: “Oui, je le fais, mais c’est mes propre trucs, pour certains chanteurs”

- russe:* «Не люблю, только лэйаут проекта с заготовленным порядком и группировкой трэков»
français: “Je n’aime pas, que la simple mise en page d’un projet avec un ordre et un regroupement de pistes prédéfinis”

- russe:* «Никогда. Каждая работа неповторима»
français: “Jamais. Chaque œuvre est unique”

- russe:* «Использую. Разрабатываю для повторяющихся задач шаблоны и готовлю шаблон для завтрашней задачи»
français: “J’en utilise. Je développe des modèles pour les tâches répétitives et je prépare un modèle pour la tâche du lendemain”

- russe:* «Темплейты всегда удобны и ускоряют процесс работы. Не надо отвлекаться на технические мелочи, все под рукой и на горячих клавишах»
français: “Les modèles sont toujours pratiques et accélèrent le processus de travail. Pas besoin de se laisser distraire par des détails techniques, tout est à portée de main et peut être activé par des touches de raccourci”

- russe:* «Использование темплейтов ограничивает»
français: “L’utilisation de modèles restreint”

- russe:* «Да, в основном для предустановок мастеринга. Набор плагинов на мастере»
français: “Oui, surtout pour les présélections de mastering. Un ensemble de plugins sur le mastering”

- russe:* «С чистого листа»
français: “A partir de rien”

- russe:* «Никогда, пробовал – не получилось. Каждая песня, каждый звук уникальны»
français: “Jamais, j’ai essayé – ça n’a pas marché pour moi. Chaque chanson, chaque son sont uniques”

russe: «При работе ФОН ускоряют подготовку к лайву»
français: “Pour le travail de FОН, les préparatifs pour le live sont accélérés”

L'attitude à l'égard des modèles est très ambivalente. Certaines personnes aiment et utilisent des modèles dans leur travail quotidien parce qu'ils considèrent que les modèles accélèrent et simplifient le processus de travail. D'autres rejettent les modèles, affirmant que les modèles les limitent. On peut noter que les modèles sont nettement préférés par ceux dont le travail représente un flux important de projets et dont le temps de session est limité. Les ingénieurs du son qui peuvent consacrer plus de temps à un projet et s'engager dans une exploration créative préfèrent commencer un projet à partir de zéro.

2.3.14 L'importance de la balance et de la spatialisation

Veillez discuter librement de l'importance de la balance, de la spatialisation (panpot + volume), du masquage (temporel et fréquentiel), de la clarté du mixage et son transparence, et des moyens techniques pour y parvenir.

“Je vois plutôt le mixage comme une activité plus artisanale qu'artistique, dont le cadre esthétique est à envisager relativement à celui fourni par la mise à plat. La problématique professionnelle principale d'un mixage est qu'il soit validé. Or le contexte de production actuel amène les intervenants antérieurs au mixage à développer de façon très prononcée les différents enjeux esthétiques du morceau. Le mixeur est donc actuellement plus amené à prolonger une ligne esthétique déjà prononcée qu'à imposer sa propre vision. Les différentes notions que tu développes sont toutes très importantes dans la bonne conduite d'un mixage mais leur réalisation dépend plus de choix antérieurs (mise à plat) que de ma propre opinion”

russe: «Абсолютно всё, что есть в треках должно быть слышно на любой акустике. Или пропадает сам смысл такого количества треков. Это должно делаться эквалайзером и громкостью. Возможно динамическим. Панорама не решит проблемы нечитаемости инструментов. Ясность добивается исключительно вырезанием не нужного, а не прибавлением чего-то. Прозрачный, четкий трек не всегда хорош, хотя бы просто из-за музыки»

français: “Absolument tout ce qui se trouve dans les pistes doit être audible sur n'importe quel haut-parleur. Ou bien on perd l'intérêt même d'avoir tant de pistes. Cela doit être fait avec l'équaliseur et le volume. Possiblement dynamique. Le panorama ne résoudra pas le problème des instruments non intelligibles. La clarté ne s'obtient qu'en supprimant ce dont vous n'avez pas besoin, et non en ajoutant quelque chose. Une piste transparente et claire n'est pas toujours bonne, ne serait-ce qu'à cause de la musique”

russe: «Без правильного баланса микс не зазвучит. Никогда. И мастерингом этого не исправить. Предвосхищая следующий вопрос-оценку – важно создать некое пространство убедительно»

français: “Sans la bonne balance, le mixage ne sonnera pas juste. Jamais. Et le mastering ne le réparera pas. En anticipant la prochaine question – il est important de créer une sorte d'espace de manière convaincante”

russe: «Баланс это когда все партии слышно и в данный момент времени акцентирована та, что наиболее значима по драматургии. Пространство формируется внутренними процессами источника и в основном реверберацией. При записи акустических инструментов фиксируется и пространство, обстановка. Пространство записи не должно превалировать над источником. При искусственном ревербировании, пространство не должно превышать исходный сигнал, за исключением трюков. Пишу акустический источник способом М/С, стараясь разместить источник симметрично, не по центру помещения. Моно близко, стерео дальше, как близкий зритель. При сведении стараюсь воссоздать фигуру в пространстве и поместить её в естественном отношении к ансамблю. Электроакустические инструменты могут создавать самостоятельные пространства большего масштаба, использую это качество для радикальных стереоэффектов, слежу чтоб такое играло и в моно. Для соединения в общее пространство использую натуралистичные ревербераторы на грани заметности, посылая на те большинство партий. На ритмичные партии использую дилей с фидбэком для заполнения пустот, но не везде, а там где явно пусто или для сглаживания драматических частей. Использую генераторы гармоник для пустоватых на тембры партий. Почти всегда создаю стерео из моно партий. Само по себе моно не биологично, звучит неестественно. Маскировка предупреждается аранжировочным методом. Конкурирующие партии могут быть разведены в разные пространства, например панорамированием и ревербированием. Использую для общего пространства пару стереоревербераторов разного размера и демпфирования, по типам далёкий и близкий, что позволяет приближать и удалять кажущийся источник. Пространственная аранжировка сильнодействующий метод. В собственных проектах я стараюсь плотно заполнить всё пространство, слить партии в цельный звук, в котором существуют драматические фигуры, обычно текст и мелодический агент в фоне гармоничных им шумов. Музыка в балансе это одна подвижная синусоида внизу, чётные - нечётные к ней гармоники, иногда пересекаемые ритмичными шумами и плавающее по поверхности сообщение, обычно текст или солист. Все эти компоненты должны создавать самодостаточный цельный образ. Цепь образов лучше соединять последовательно»

français: “La balance est atteinte lorsque toutes les parties sont entendues et qu’à un moment donné, celle qui est la plus significative en termes de dramaturgie est accentuée. L’espace est façonné par des processus de source interne et principalement par la réverbération. Lors de l’enregistrement d’instruments acoustiques, l’espace, l’ambiance, est également fixe. L’espace d’enregistrement ne doit pas prendre le pas sur la source. Avec la réverbération artificielle, l’espace ne doit pas dépasser le signal source, sauf pour les trucages. J’écris la source acoustique à la manière M/S (mid-side), en essayant de placer la source de manière symétrique, pas au centre de la pièce. La prise mono est de près, la stéréo est de plus loin, comme un spectateur de près. Lors du mixage, j’essaie de recréer la figure dans l’espace et de la placer en relation naturelle avec l’ensemble. Les instruments électroacoustiques peuvent créer des espaces indépendants à plus grande échelle, j’utilise cette qualité pour des effets stéréo radicaux, en veillant à ce qu’ils puissent également être joués en mono. Pour la liaison dans l’espace global, j’utilise des réverbérations naturalistes à la limite de la perceptibilité, en leur envoyant la plupart des parties. Sur les parties rythmiques, j’utilise des délais avec des feedbacks pour combler les vides, mais pas partout, juste là où ils sont manifestement vides ou pour construire les parties dramatiques. J’utilise également des générateurs d’harmoniques pour les parties qui sont un peu creuses en termes de timbre. Je crée presque toujours de la stéréo à partir de parties mono. La mono en soi n’est pas biologique, elle ne sonne pas naturelle. Le masquage est évité par la méthode d’arrangement. Les parties concurrentes peuvent être divisées en différents espaces, par exemple par des panoramiques et des réverbérations. En utilisant une paire de réverbérations stéréo de taille et d’amortissement différents pour l’ensemble de l’espace, on utilise les types de réverbération distant et proche, ce qui permet d’approcher et d’éloigner la source apparente. La disposition spatiale est une technique puissante. Dans mes propres projets, j’essaie de remplir étroitement tout l’espace, de fusionner les parties en un son cohérent, dans lequel il y a des figures dramatiques, généralement du texte et un agent mélodique avec des bruits harmoniques en arrière-plan. La musique avec une bonne balance est une onde sinusoïdale en mouvement, avec des harmoniques paires et impaires, parfois traversée par des bruits rythmiques et un message flottant, généralement un texte ou un soliste. Tous ces éléments doivent créer une image complète et autonome. Une chaîne d’images est mieux connectée dans l’ordre”

russe: «Пространство создается эквализацией, громкостью, панорамой и пространственными обработками. Но можно обойтись и просто эквализацией и пространственными обработками (ревер, дилей)»

français: “L’espace est créé par l’égalisation, l’intensité sonore, les panoramiques et les traitements spatiaux. Mais vous pouvez aussi vous en sortir avec l’égalisation et les traitements spatiaux (réverbération, délai)”

russe: «Любую из этих проблем нужно решать прежде всего с художественной точки зрения. Художественные средства диктуют необходимость применения тех или иных технических средств, а не наоборот»

français: “Tous ces problèmes doivent être résolus avant tout d’un point de vue artistique. Les moyens artistiques dictent la nécessité des moyens techniques, et non l’inverse”

- russe:* «Для ясности очень осторожно с делями и реверами и убрать частотные конфликты»
- français:* “Pour la clarté, faites très attention aux délais et aux réverbérations et supprimez les conflits de fréquence”
- russe:* «Важно, чтобы складывалось впечатление прекрасного исполнения, чтобы ритм казался подвижным и ровным, чтобы все строили и играли вместе, именно этого и надо искать и добиваться любыми способами и средствами»
- français:* “Il est important d’avoir l’impression d’une belle performance, de faire en sorte que le rythme semble fluide et régulier, tout le monde est s’accordé et joue ensemble, c’est ce qui doit être recherché et atteint par tous les moyens et par tous les chemins”
- russe:* «Прозрачный микс – следствие прозрачной аранжировки, плюс устранение маскировки, важно прочистить хвосты реверов»
- français:* “Un mixage transparent est la conséquence d’un arrangement transparent, plus l’élimination du masquage, il est important de nettoyer les queues de réverbération”
- russe:* «Очень важен баланс, важнее всего остального»
- français:* “La balance est très importante, plus importante que toute autre chose”

La balance et l’espace sont des aspects très importants du mixage. Il est également important de tenir compte des souhaits d’un client. La portabilité du mixage est l’un des principaux objectifs du mixage. Une mauvaise balance ne peut pas être corrigée au stade du mastering. Un bon arrangement est la clé d’une bonne spatialisation. Le traitement spatial, ainsi que l’égalisation, sont des outils essentiels pour créer l’espace. L’aspect artistique doit dominer tout le reste.

2.3.15 Quels objectifs esthétiques et techniques la spatialisation permet-elle de résoudre ?

“Techniquement, du démasquage binaural ; esthétiquement, des découvertes sonores, des mouvements inouïs, etc.”

“Démasquage, équilibre monophonique du mixage (important pour l’écoute laptop et téléphone)”

- russe:* «В фильме это ощущение помещения или леса или иного места, это важно»
- français:* “Dans un film, c’est la sensation d’une pièce, d’une forêt ou d’un autre endroit, c’est important”
- russe:* «Стиль и жанр, комфортное прослушивание в течение длительного времени без утомления слуха»
- français:* “Style et genre, l’écoute confortable pendant de longues périodes sans fatiguer vos oreilles”
- russe:* «Погружение в предложенные обстоятельства»
- français:* “Immersion dans les circonstances offertes”

- russe:* «Приводит слушателя в мир сообщения, как карта. Выполняет задачу самопрограммирования слушателя. Чётко выражает идею сообщения. В целом, техно-артефакт звукозаписи позволяет сообщению проникнуть в социальный слой»
français: “Conduit l’auditeur dans le monde du message, comme une carte. Effectue la tâche d’auto-programmation de l’auditeur. Exprime clairement l’idée du message. Dans l’ensemble, le techno-artefact de l’enregistrement sonore permet au message de pénétrer dans la couche sociale”
- russe:* «Кайф от прослушивания!)) Как пример Fool’s Garden – Man of Devotion»
français: “Le buzz de l’écoute !)) A titre d’exemple Fool’s Garden – Man of Devotion”
- russe:* «Практически все»
français: “Pratiquement tous”
- russe:* «Эмоциональность»
français: “Émotion”
- russe:* «Локализация объекта в пространстве»
français: “Localisation de l’objet dans l’espace”
- russe:* «Донести замысел песни так, чтобы хотелось слушать. Чтобы было ощущение путешествия, приключения, чтобы было приятно на слух»
français: “Transmettre le message de la chanson pour qu’il soit souhaitable de l’écouter. Un sens du voyage, de l’aventure, agréable à l’oreille”
- russe:* «Эффект присутствия и правильное настроение»
français: “L’effet de présence et l’humeur souhaité”
- russe:* «В целом создать весь микс на 80%»
français: “Créer généralement tout le mixage à 80%”
- russe:* «То, что называют атмосферой и настроением»
français: “Ce qu’ils appellent l’atmosphère et l’humeur”

2.3.16 Quels sont les plugins ou les effets hardware que vous utilisez le plus souvent ?

“GRM tools”

“Software : Pro Tools 2021/ Fabfilter / Universal Audio / Limiteur Weiss MM-1 / Soundtoys / Plugin Alliance”

“Logic Pro, Fabfilter, Steven Slate, Voxengo, Klangheim, UVI, SSL, Valhalla”

“Avid s6; Waves R Comp, Q6; Altiverb”

“Logic Pro, Lexicon, iZotope Ozone. Hardware: TL Audio, dbx 160A, Drawmer”

“Fabfilter, iZotope Ozone”

“Waves L2, CLA-1176, C1, Kramer Pie, maxBass, H-delay, DMGAudio equality, Altiverb”

“FabFilter, Uad, Waves”

“Slate digital, Uad, NEVE Portico, Waves, Eventide”

“Melda Production”

“SSL, chandler limited, Reaper plugs, Bitwig, Uad, Softube”

2.3.17 Discussions libres

“La mise à plat !!!!”

russe: «Надо быть смелым художником в миксинге, но при этом нужно понимать - физику не обманешь. Точнее надо знать, как ее обманывать. Это и есть профессионализм звукорежиссера»

français: “Il faut être un artiste courageux dans le mixage, mais il faut comprendre qu’on ne peut pas tromper la physique. Ou plutôt, il faut savoir comment la tromper. C’est le professionnalisme d’un ingénieur du son”

russe: «Продюсер – тот кто вырабатывает товар, в частности звукозапись. Главное в его деятельности социально-психологический аспект, координация коллектива. Другие задачи могут быть решены практически автоматически. Автомикшер SoundDevises неплохо микширует. Много в изготовлении записи может быть автоматизировано. Потому сосредоточиться хорошо на образе жизни, позволяющему создавать артефакты – сообщения другим, например в виде песен»

français: “Un producteur est une personne qui réalise un produit, un enregistrement sonore en particulier. L’essentiel dans son activité est l’aspect socio-psychologique, la coordination du collectif. D’autres tâches peuvent être résolues de manière quasi automatique. Le mixeur automatique SoundDevises fait un très bon travail de mixage. Une grande partie de la production d’un enregistrement peut être automatisée. Il est donc bon de se concentrer sur un mode de vie qui permet de créer des artefacts - des messages aux autres, par exemple, sous forme de chansons”

russe: «Посадили двух звукорежиссеров на пятнадцать лет за убийство певицы... Отсидели они в одной камере срок, вышли в один день, поговорили еще минут сорок про аппарат и разошлись.»

français: “Deux ingénieurs du son ont été emprisonnés pendant 15 ans pour avoir tué une chanteuse... Ils ont purgé leur peine dans la même cellule, sont sortis le même jour, ont parlé encore quarante minutes de leur équipement et se sont séparés”

russe: «От микрофона и оборудования не зависит готовый результат. Самые лучшие средства те, с которыми работаете Вы, причем до тех пор пока Вы с ними работаете»

français: “Le résultat final ne dépend pas du microphone et de l’équipement. Les meilleurs outils sont ceux avec lesquels vous travaillez, pour autant que vous travailliez avec eux”

russe: «Звукорежиссер – это не только хороший и тренированный слух, но и то что находится между его ушами»

français: “Un ingénieur du son n’est pas seulement une bonne oreille bien entraînée, mais aussi ce qui se trouve entre ses oreilles”

russe: «Пожалуйста, поделитесь выводами после дипломной!»

français: “Veuillez partager les conclusions de votre mémoire, s’il vous plaît !”

2.4 Au lieu d'une conclusion

Un espace créé avec succès, selon la majorité, représente 80% d'un mixage réussi, comme le confirment les statistiques des réponses des participants au questionnaire.

La résolution du problème du masquage améliore considérablement la qualité du mixage. Une distinction est faite entre le masquage de fréquences et le masquage temporel. Les principaux outils techniques qui permettent de résoudre ce problème sont l'égaliseur, le side chain, la dilution spatiale des sources et le panoramique diffus.

D'autres aspects importants d'un bon mixage sont sa profondeur et sa largeur. La profondeur est obtenue en créant des plans, c'est-à-dire la bonne combinaison de filtrage, de compression et de traitements de temporels (délais et réverbération). La largeur est créée par une spatialisation correctement construite, par les mêmes moyens que ceux décrits ci-dessus.

Il est important de respecter l'équilibre spectral de chaque instrument, ainsi que l'équilibre tonal de l'ensemble du mixage. Pour cela, certaines personnes préfèrent se fier à des références, tandis que d'autres utilisent des plug-ins comme Tonal Balance Control (iZotope), quand d'autres le font à l'oreille.

Tout ce qui n'a pas été pris en compte au stade de l'arrangement doit être résolu par l'ingénieur du son. Mais, il arrive qu'une solution ne soit pas efficace ou satisfaisante du point de vue du mixage auquel cas on doit modifier l'arrangement.

Je tiens à souligner une fois de plus que dans ce mémoire, nous nous limitons à considérer uniquement la question de la création d'un espace sonore (spatialisation des sources sonores) avec un ensemble minimal d'outils : ajustement du volume sonore, panoramiques, filtrage des graves et des aigus.

Chapitre 3

Architecture du système de mixage automatique

3.1 Le langage de programmation

Dans ce chapitre, nous allons examiner en détail le fonctionnement du système créé. Nous présenterons également l'architecture globale et les principaux modules. Nous allons discuter de la composition et du format des données d'entrée. Nous allons parler de l'abstraction du modèle de scène virtuelle. Nous justifierons le choix du langage de programmation pour le prototypage.

Dans notre cas, nous parlons d'un système multifonctionnel plutôt complexe, et nous avons donc choisi l'approche orientée objet parmi toute la gamme des langages de programmation. L'approche orientée objet nous permettra de décomposer un système complexe en modules et en classes (objets), et donc de passer d'un système très complexe à un ensemble d'éléments simples inter-connectés.

Les concepts de base de la programmation orientée objet sont l'encapsulation, l'héritage et le polymorphisme. L'*encapsulation* nous permet de cacher les détails techniques, souvent complexes, des implémentations de fonctions, et de nous concentrer sur les caractéristiques architecturales du logiciel. L'*héritage* implique la réutilisation du code, ce qui résout le problème du support, de l'interchangeabilité et, dans notre cas, de la capacité à améliorer les modules et les classes du système développé. Le *polymorphisme* nous permet de construire des fonctions abstraites capables d'accepter n'importe quel type de données (ou plusieurs types de données), et d'adapter les algorithmes et le comportement en fonction des types de données à d'entrée.

Les langages principaux et les plus utilisés sont Java, C#, C++ et Python. Le langage C++ permet d'obtenir des applications avec la vitesse la plus élevée car les compilateurs peuvent optimiser le code pour les architectures les plus sophistiquées des processeurs modernes. Les langages C# et Java, utilisés pour le développement de systèmes complexes, se distinguent par leur brièveté, leur polyvalence et la productivité de développement. Dans les deux cas, la compilation se fait en *byte-code* intermédiaire, et ce byte code est ensuite interprété par la machine virtuelle correspondante sur diverses plateformes (Windows, Linux,...).

Python est un langage de programmation qui a gagné une énorme popularité ces dernières années. Cette popularité est due à l'apprentissage facile de ce langage grâce à sa syntaxe simple et intuitive et à l'existence de nombreuses bibliothèques gratuites et open source. Ce langage est

idéal pour le prototypage, car il permet de se concentrer sur le sujet de la mise en œuvre plutôt que sur les particularités techniques de la programmation. Python utilise également une machine virtuelle ce qui est pratique car cela permet d'avoir une capacité multi-plateforme mais, malheureusement, il n'est pas possible d'optimiser les performances du code Python. Toutefois, de nombreux *paquets*¹ de Python utilisent des langages de bas niveau tels que le C/C++ et Python n'est qu'une enveloppe (wrap) autour de ce code. Cette approche permet d'obtenir un compromis entre la vitesse de développement et la performance d'exécution du code, ce qui est idéal pour le prototypage. Pour ces raisons, nous avons choisi Python comme langage pour la programmation de notre prototype.

3.2 Le dictionnaire

Il existe de nombreuses approches pour la création de logiciels. Par exemple, on peut commencer par l'analyse et la modélisation des flux d'activités ou de processus selon les normes TOGAF², ce qui est typique pour le développement des grands logiciels complexes. Le plus souvent, et dans de nombreuses approches (Agile, Iconix, XP), on commence par introduire la terminologie et ainsi créer un *vocabulaire*. Ensuite, les éléments de ce vocabulaire sont souvent "convertis" en classes et le logiciel prend ainsi sa forme.

Les principaux éléments qui interviennent dans le système de mixage automatique sont les suivants.

- Le *mixeur* proprement dit, qui est l'élément responsable du mixage des pistes du projet et de la transformation de ces pistes en mixage final en fonction des réglages des groupes (les bus) et du master. Il est possible de parler d'un mixage complet de toutes les pistes ou d'un mixage partiel de certaines pistes pré-sélectionnées.
- La *piste* (track) audio qui se présente sous forme d'un fichier audio au format wav.
- Le *groupe* de pistes qui est une abstraction qui regroupe plusieurs pistes audio sur un seul bus afin d'appliquer un traitement commun à ces pistes groupées.
- Le *bus* qui est une abstraction vers laquelle un ou plusieurs signaux audio (pistes) peuvent être acheminés pour un traitement spécifique. Souvent, un groupe de pistes est acheminé vers le même bus. Il convient de distinguer le bus *maître* (master bus) en tant que traitement final de toutes les pistes du mix.
- Les *règles de mixage* (mix rules). Nous introduisons ce concept pour désigner un ensemble de règles de traitement des pistes audio. Les règles de mixage peuvent également être appliquées séparément à un groupe de pistes et à un bus maître. Les règles de mixage comprennent des instructions pour appliquer des effets aux insertions (inserts) et aussi aux envois (sends). De manière générale, les règles peuvent inclure la sélection et le réglage d'égaliseurs, de compresseurs, de saturateurs et d'autres éléments de traitement du signal. Dans ce mémoire, nous nous limiterons à un ensemble limité de règles. Il s'agira notamment de l'écrêtage des fréquences supérieures (high-cut) et inférieures (low-cut), du volume et du panoramique.
- L'*optimiseur* qui est un algorithme d'optimisation numérique qui optimise les règles de mixage en fonction de critères d'optimisation définis.
- Les *critères* d'optimisation qui sont un ensemble de critères définissant la qualité d'un mixage, exprimés par des formules mathématiques discrètes. Ces formules, qui tiennent compte des règles de mixage, transforment les pistes audio en quelques nombres réels proportionnels à la qualité du mixage. Ces critères peuvent être considérés comme une abstraction de la perception psycho-acoustique de l'ingénieur du son, mais une abstraction très simplifiée.

1. packages

2. <https://www.opengroup.org/togaf>

- Le *mapper* qui est un élément supplémentaire nécessaire pour traduire les règles de mixage du langage de l'ingénieur du son en un langage mathématique compréhensible pour le calcul des critères d'optimisation.
- Le *projet de mixage* (mix project) qui est un ensemble de tous les fichiers audio concernés par le mixage, ainsi qu'une description des exigences et des souhaits du client, de l'ingénieur du son ou bien du producteur, exprimés en langage naturel et de métier. Les spécificités du mixage sont étroitement liées au style de musique.
- Le *style de mixage* (mix style) qui est un ensemble de règles de mixage spécifiques à un style particulier, les types de traitement sur les bus de groupes et le bus maître, ainsi que sur les bus d'envoi (sends).
- Le *chargeur* (loader) qui est un module auxiliaire qui charge le projet dans la mémoire de l'ordinateur à partir du disque dur, vérifie l'intégrité des données du projet, effectue les conversions nécessaires pour unifier les données et présente aussi les données en mémoire dans un format adapté aux travaux ultérieurs (mixage et optimisation).
- Le *chargeur de bus* ou de styles (bus) qui est un module auxiliaire qui, en fonction d'un style de musique donné, génère le traitement requis sur les bus (groupes) prédéfinies et les charge en mémoire pour une utilisation ultérieure dans le mixeur.
- Le *mixdown* qui est le mixage final résultant du travail itératif de l'optimiseur sur un projet chargé. Le but ultime est d'obtenir un mixage de qualité.

3.3 L'abstraction de la scène virtuelle

Le producteur de musique, l'ingénieur du son ou le client savent souvent à l'avance comment doit sonner le résultat final de son morceau de musique. Nous pouvons imaginer une sorte de *scène virtuelle* en 3 dimensions dans laquelle nous plaçons nos sources sonores. Cette scène est caractérisée par 3 dimensions : la position du son en profondeur, la position du son horizontalement et la position du son verticalement.

Étant donné le choix limité d'instruments dans ce travail de mémoire, nous faisons l'hypothèse très caricaturale que la profondeur est caractérisée par l'intensité sonore de la source (volume), sa position horizontale par son panoramique (pan), et sa position verticale par ses caractéristiques spectrales, contrôlées par un écrêtage des hautes (hcut) et basses (lcut) fréquences.

Nous pourrions exécuter l'algorithme d'optimisation en "flottaison libre" et l'optimisation ne serait contrôlée que par les critères d'optimalité, mais comme le montrent de nombreux travaux antérieurs [Tom et al., 2019; Pestana, 2013; Pestana and Reiss, 2014] sur les systèmes de mixage automatique, cette approche d'optimisation ne permet pas d'obtenir la qualité souhaitée du mixage final.

Par conséquent, dans notre approche, nous décidons d'*aider* l'algorithme d'optimisation en ajoutant des contraintes sur l'espace des solutions possibles, les mixages finaux. Ces contraintes sont établies sur la base des souhaits du client ou des recommandations de l'ingénieur du son.

Nous supposons que ces recommandations ont déjà été définies sous forme descriptive lors des étapes précédentes d'analyse du projet : ce travail peut être réalisé soit par l'utilisateur du système, soit automatiquement en utilisant des méthodes d'intelligence artificielle, comme décrit en 2.2.

3.3.1 Les trois plans de volumes

On peut potentiellement créer autant de plans que l'on souhaite en fonction de l'intention artistique, mais beaucoup d'ingénieurs du son se limitent initialement à trois plans pour des raisons pratiques : proche (front), moyen (mid) et lointain (far).

Nous avons décidé de rester modestes (ou prudents) et d'introduire seulement 3 plans pour le volume. Ces plans définissent les limitations imposées pour les valeurs de volume³ en fonction du plan dans lequel nous voulons placer la source sonore.

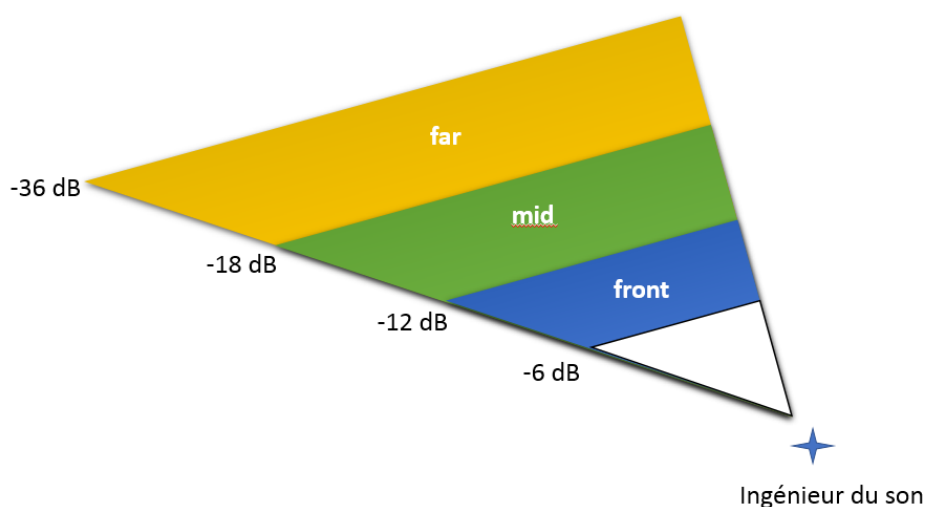


Figure 3.1 – 3 plans pour le volume.

Voici les valeurs des intervalles fixant les limites de *Volume* en fonction du plan sélectionné :

- ◇ **front** – proche de l'auditeur [-6 dB, -12 dB];
- ◇ **mid** – au milieu de la scène [-12 dB, -18 dB];
- ◇ **far** – loin de l'auditeur [-18 dB, -36 dB];
- ◇ **any** – toute distance [-6dB, -36dB].

3.3.2 Les plans panoramiques

Contrairement au découpage choisi pour les niveaux de volume, nous avons décidé de diviser le panorama en un plus grand nombre de valeurs fixes ou de secteurs prédéfinis. Cela nous permettra de prédéterminer avec plus de précision la position souhaitée pour l'instrument dans le mixage.

Nous avons fixé trois valeurs : centre (C), gauche (L) et droite (R). Pour de nombreux instruments, comme les voix ou les basses, il est important de placer l'instrument au centre strict du mixage. Ceci est également vrai pour les autres instruments, grâce auxquels nous assurons la largeur de la scène en les positionnant aux valeurs extrêmes fixes de gauche et de droite.

3. contraintes de bornes

Nous avons mis en évidence le cas où l'instrument peut être placé n'importe où, sans restriction (any).

Nous avons également mis en évidence le cas où il est souhaitable que l'instrument soit placé au centre, mais pas nécessairement dans une position centrale fixe (mid).

Nous avons divisé l'espace horizontal entier en 2 secteurs, gauche (left) et droite (right) et avons permis aux instruments d'occuper n'importe quelle position dans ces secteurs, permettant ainsi une plus grande liberté de choix. Cela convient aux cas où l'on veut absolument entendre l'instrument d'un côté particulier mais sans pour autant délimiter strictement sa position.

Pour permettre un positionnement plus précis, nous avons divisé chacun des secteurs, gauche et droite, en 3 sous-secteurs : extrême (eleft, eright), médian (mleft, mright), centré (cleft, cright). Cela a permis de créer 6 sous-secteurs supplémentaires pour un positionnement plus précis de la source musicale.

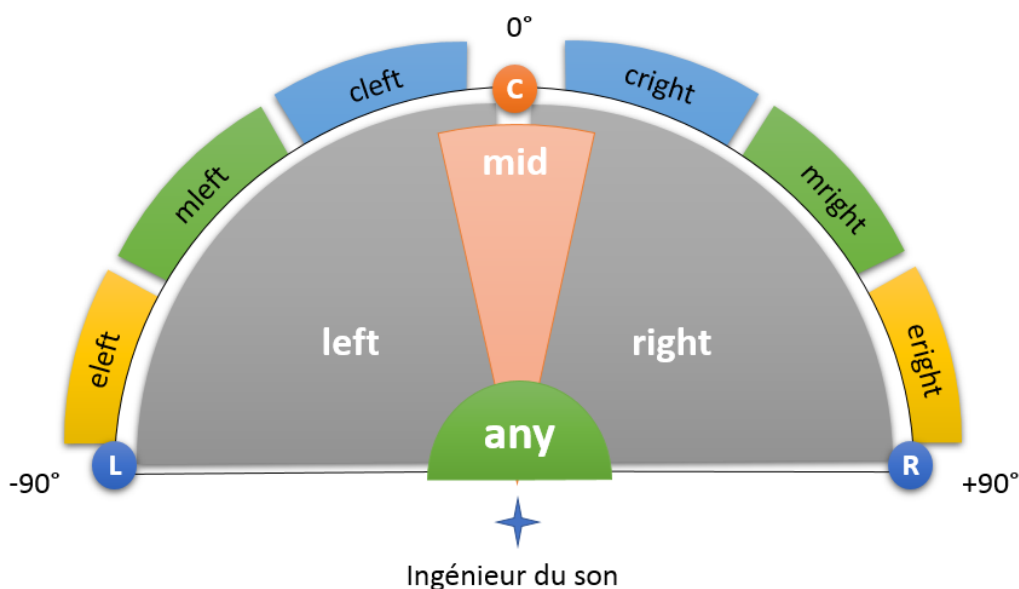


Figure 3.2 – Les plans panoramiques : secteurs, sous-secteurs et valeurs fixes.

Nous voudrions rappeler une fois de plus que cette division est conditionnelle, et qu'elle peut être révisée en fonction des objectifs et des souhaits. Nous avons choisi cette division dans un souci de simplification.

Vous trouverez ci-dessous les valeurs d'intervalle correspondant à chacun des secteurs, sous-secteurs ou valeurs fixes du panorama :

- ◇ **C** – source strictement centrée $[0^\circ]$;
- ◇ **L** – strictement gauche $[-90^\circ]$;
- ◇ **R** – strictement droite $[+90^\circ]$;
- ◇ **any** – toute position possible $[-90^\circ, +90^\circ]$;

- ◇ **mid** – panoramique central préféré $[-10^\circ, +10^\circ]$;
- ◇ **left** – panoramique vers la gauche $[-90^\circ, 0^\circ]$;
- ◇ **right** – panoramique vers la droite $[0^\circ, +90^\circ]$;
- ◇ **eleft** – extrême gauche $[-90^\circ, -60^\circ]$;
- ◇ **mleft** – milieu gauche $[-60^\circ, -30^\circ]$;
- ◇ **cleft** – centré à gauche $[-30^\circ, 0^\circ]$;
- ◇ **eright** – extrême droite $[+60^\circ, +90^\circ]$;
- ◇ **mrigh** – milieu droit $[+30^\circ, +60^\circ]$;
- ◇ **crigh** – centré à droite $[0^\circ, +30^\circ]$.

3.3.3 Les types d'égalisation et de compression

Nous ne l'utilisons pas dans ce travail de mémoire, mais nous en avons tenu compte pour une extension future et nous avons donc ajouté les paramètres nécessaires pour qu'il soit facile d'intégrer ultérieurement la compression et l'égalisation.

L'égalisation

Le système de mixage mis en œuvre actuellement n'utilise que des filtres d'écrtage passe-haut et passe-bas rudimentaires, qui sont caractérisés par les paramètres suivants :

- ◇ **locut** – la fréquence maximale de coupure des basses fréquences en Hz;
- ◇ **hicut** – la fréquence minimale de coupure des fréquences aigues en Hz.

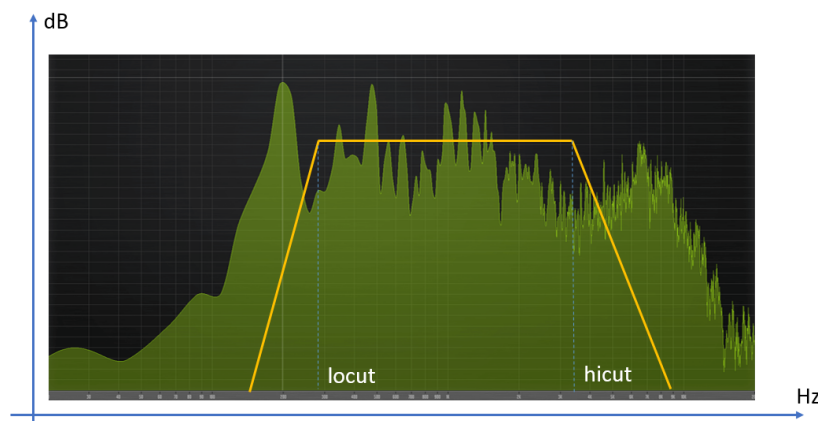


Figure 3.3 – Les paramètres passe-haut (locut) et passe-bas (hicut).

Les deux options d'égalisation suivantes pourraient être automatisées à l'avenir :

1. définir les fréquences de coupure haute et basse automatiquement via l'analyse du spectre de la piste en tenant compte du style de morceau ;
2. définir automatiquement les fréquences conflictuelles pour un couple de pistes.

Pour résoudre les problèmes de conflits de fréquences et pour corriger l'équilibre spectral, à l'avenir, les paramètres de contrôle suivants sont prévus pour l'égaliseur :

- ◇ **none** – ne pas appliquer l'égalisation ;
- ◇ **light** – une correction légère [0 dB, -3 dB] ;
- ◇ **normal** – toute correction, aucune restriction ;
- ◇ **agressive** – une correction agressive [-6 dB, -18 dB].

La compression

Pour les développements futurs, nous avons également envisagé les types de compression qui pourraient être appliquées lors de l'optimisation de spatialisation :

- ◇ **none** – pas de compression ;
- ◇ **any** – aucune restriction sur la compression ;
- ◇ **attack** – attaque longue, relâchement court, ratio moyen, seuil -6 dB ;
- ◇ **hard** – attaque courte, relâchement moyen, gros ratio, seuil -12 dB.

Les paramètres de compression peuvent être ajustés plus précisément et individuellement pour chaque piste, grâce à des systèmes experts basés sur l'expérience de l'ingénieur du son [De Man, 2017; Pestana, 2013], ainsi qu'à des techniques d'apprentissage profond [Mahanta et al., 2021].

3.4 Les groupes prédéfinis et les bus

Nous avons utilisé un schéma très simple de regroupement des pistes qui peut être appliqué à presque n'importe quelle chanson ou morceau de musique contemporains. Nous avons alloué un bus séparé pour la batterie et les percussions, un bus séparé pour les voix, y compris les chœurs, et un bus séparé pour tout le matériel musical. Le bus de la batterie est envoyé au bus de la musique tandis que les deux autres bus, voix et musique, sont envoyés vers le bus maître.

Voici les paramètres utilisés :

- ◇ **drums** – tout instrument de batterie et de percussion ;
- ◇ **vocal** – toute voix principale et toute voix de soutien ;
- ◇ **music** – tout instrument de musique autre que la batterie.

Dans ce mémoire, pour la simplicité du prototype, nous avons utilisé un ensemble minimal fixe de groupes et de bus correspondants. À l'avenir, il sera possible de procéder à un regroupement dynamique des instruments [Tot, 2018; Hsu et al., 2021] et donc d'adapter le nombre et le nom des groupes à chaque projet individuel et à chaque style de musique.

Dans ce projet, nous avons utilisé trois plans de réverbération comme effets d'envoi : Room, Plat et Hall. À l'avenir, le nombre de plans pourra être augmenté et étendu avec des effets de retard également.

3.5 La description du projet

Un projet musical pour le mixage est décrit dans un seul fichier. Nous avons décidé d'utiliser le format *json*, car nous pensons qu'il est le mieux adapté à cette tâche et qu'il possède une syntaxe intuitive.

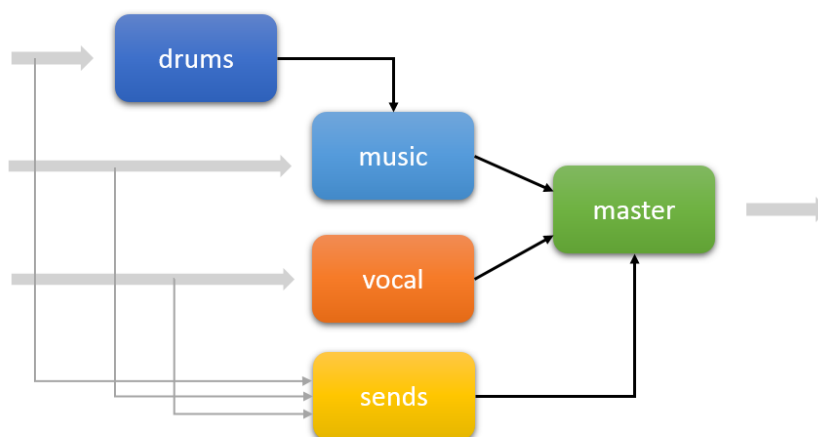


Figure 3.4 – Les regroupements de pistes et les bus.

Dans le travail actuel, nous avons dû décrire et définir tous les paramètres de ce fichier manuellement. À l'avenir, il est envisagé que ce fichier soit généré automatiquement grâce à une automatisation intelligente.

En utilisant un petit projet de deux pistes audio comme exemple (cf. Figure 3.5), considérons la structure du fichier *json* décrivant le projet :

- ◇ **project_name** – décrit le nom du projet, le même nom est utilisé pour nommer le mixage final ;
- ◇ **project_path** – spécifie le chemin où sont situés tous les fichiers audio du projet ;
- ◇ **sample_rate** – spécifie la fréquence d'échantillonnage, en Hertz ;
- ◇ **start_time** – spécifie la valeur du temps de départ, en secondes, à partir duquel la piste est analysée puis mixée ;
- ◇ **duration** – le temps, en secondes, définit la longueur d'un fragment audio, car pour le prototype nous n'avons utilisé que de petits fragments de musique ;
- ◇ **tracks** – décrit un ensemble de pistes et leurs paramètres individuels.

Ensuite, pour chaque piste :

- ◇ **track_name** – le nom du fichier audio sur le disque ;
- ◇ **track_vol** – le plan de volume (front, mid, far, any) ;
- ◇ **track_pan** – le plan de pan (C, R, L, left, right, mid, any, ...) ;
- ◇ **track_locut** – la fréquence maximale de filtrage des fréquences basses ;
- ◇ **track_hicut** – la fréquence minimale de filtrage des fréquences aigues ;
- ◇ **track_eq** – le mode d'égalisation (none, light, normal, agressive) ;
- ◇ **track_comp** – le mode de compression (none, any, attack, hard) ;
- ◇ **track_group** – indique à quel groupe appartient la piste (drums, vocal, music).

Ce fichier est chargé dans la mémoire de l'ordinateur par la classe *Loader* (*loader.py*), l'intégrité et la compatibilité du projet sont vérifiées et, en fonction des informations contenues dans le fichier, les algorithmes d'optimisation vont ensuite rechercher le mixage souhaité.

```
1 {
2     "project_name": "Rod Alexander Canon - 2 guitars",
3     "project_path": "D:\\RodAlexander_Canon_Full\\",
4     "sample_rate": 44100,
5     "start_time": 0,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_MainPair.wav",
10            "track_vol": "mid",
11            "track_pan": "any",
12            "track_locut": "80",
13            "track_hicut": "20000",
14            "track_eq": "normal",
15            "track_comp": "none",
16            "track_group": "music"
17        },
18        {
19            "track_name": "02_DI.wav",
20            "track_vol": "mid",
21            "track_pan": "any",
22            "track_locut": "80",
23            "track_hicut": "10000",
24            "track_eq": "agressive",
25            "track_comp": "attack",
26            "track_group": "music"
27        }
28    ]
29 }
```

Figure 3.5 – Un exemple de description de projet.

Les fichiers de description de tous les projets utilisés dans ce mémoire se trouvent en Annexe C.

3.6 La description du style

Chaque projet musical est accompagné d'un style décrit dans le fichier de style *json*. Le style affecte les pré-réglages du processus de mixage, en particulier certains des traitements utilisés sur les bus principaux (drums, vocal, music, master), ainsi que les effets d'envoi (sends) et autres effets éventuels qui donnent du style au projet.

Dans le cadre de notre travail, nous avons essayé de minimiser la "stylisation", en laissant les dessins aussi naturels que possible, afin que l'accent soit mis sur la disposition spatiale des instruments.

À l'avenir, il sera possible de détecter automatiquement le genre, le style et éventuellement le sous-style d'un morceau de musique [Hsu et al., 2021] et, en fonction de cela ainsi que de l'utilisation de systèmes experts à base de connaissances, de générer, à la volée, un fichier de style pour le projet.

Voici un exemple de fichier de style :

```
1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {
7                 "path" : "C:/Program Files/Common Files/VST3/Plugin
8                     Alliance/Brainworx/bx_townhouse Buss Compressor.
9                     vst3",
10                "program": null,
11                "thresh_db": "-6.0",
12                "mix": "20"
13            }
14        },
15        "vocal": [
16            {
17                "type": "vst3",
18                "name": "vocal_bus",
19                "parameters": {
20                    "path" : "C:/Program Files/Common Files/VST3/Plugin
21                        Alliance/Brainworx/bx_townhouse Buss Compressor.
22                        vst3",
23                    "program": null,
24                    "thresh_db": "-3.0",
25                    "mix": "20"
26                }
27            },
28            "music": [
29            ],
30            "master": [
31            ],
32            "sends": [
33                {
34                    "type": "vst3",
35                    "name": "room",
36                    "parameters": {
37                        "path" : "C:\\Program Files\\Common Files\\VST3\\TC
```

```
42         Electronic\\VSS3.vst3",
43         "program": "Clear Room"
44     },
45     {
46         "type": "vst3",
47         "name": "plate",
48         "parameters": {
49             "path" : "C:\\Program Files\\Common Files\\VST3\\TC
50                 Electronic\\VSS3.vst3",
51             "program": "Bright Plate"
52         }
53     },
54     {
55         "type": "vst3",
56         "name": "hall",
57         "parameters": {
58             "path" : "C:\\Program Files\\Common Files\\VST3\\TC
59                 Electronic\\VSS3.vst3",
60             "program": "Large Hall Clear"
61         }
62     ]
63 }
```

Tous les fichiers de style utilisés dans ce travail se trouvent en Annexe D.

Lorsque le programme démarre, le fichier de style est chargé à l'aide de la classe *Bus* (*buses.py*), les effets d'insertion (*inserts*) et d'envois (*sends*) sont ensuite chargés, leurs paramètres sont initialisés et la configuration des bus spécifiés est créée.

3.7 La scène "3D" interactive ou pilotée par l'utilisateur

Outre la génération entièrement automatique des fichiers de projet et de style, l'utilisateur peut également les modifier manuellement, comme cela a été fait pour les besoins de ce prototype.

Toutefois, à l'avenir, il sera possible d'envisager une interaction avec l'utilisateur, qu'il s'agisse d'un ingénieur du son, d'un producteur ou d'un musicien amateur. On pourrait imaginer que, comme dans un jeu vidéo, l'utilisateur puisse modifier la disposition spatiale des instruments dans une scène tridimensionnelle, et qu'il puisse également combiner différents styles ou emprunter des parties de styles, ce qui lui donnerait un espace combinatoire incroyablement vaste pour exprimer sa créativité. Et on pourrait imaginer que selon les actions itératives de l'utilisateur, l'algorithme d'optimisation ajuste les instruments pour obtenir un paysage sonore plus net et plus large spatialement.

3.8 L'aperçu de l'architecture du système

Dans la suite, par souci de brièveté, de simplicité et de clarté de la rédaction, nous nommerons VitaMIX le système de mixage automatique que nous sommes en train de mettre au point.

Figure 3.8 montre l'architecture simplifiée du prototype VitaMIX.

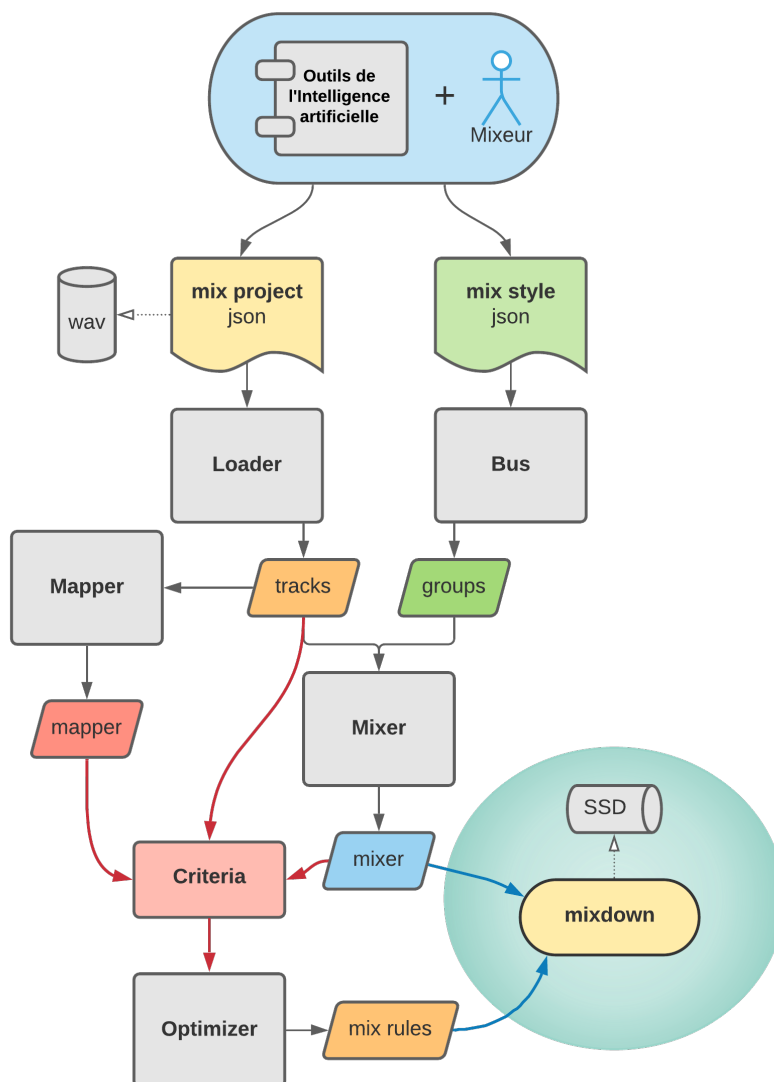


Figure 3.6 – L'architecture de VitaMIX.

Dans cette section nous décrivons brièvement le fonctionnement de l'ensemble du système.

Tout commence par la préparation de deux fichiers json : un fichier de projet (*mix project*) et un fichier de style (*mix style*). Comme nous l'avons déjà mentionné précédemment, les fichiers peuvent

être créés manuellement par l'utilisateur, soit sur la base d'une analyse algorithmique utilisant des techniques d'apprentissage profond, soit de manière interactive.

Le fichier de mixage du projet, *mix project*, pointe vers les pistes sonores originales au format wav, et donne également des informations caractéristiques sur le traitement préféré pour ces pistes.

Le fichier de style de mixage, *mix style*, décrit comment les groupes et les bus principaux de la table de mixage virtuelle doivent être formés, quels effets d'insertion et d'envoi doivent être utilisés et avec quels réglages.

Il existe deux classes :

1. *Loader* pour travailler avec le fichier de projet (*mix project*);
2. *Bus* pour travailler avec le fichier de style (*mix style*).

Ces deux classes créent respectivement un objet qui comprend des pistes audio et des informations d'accompagnement, *tracks*, et les bus (groups) avec leurs paramètres prédéfinis, *groups*.

Sur la base de ces deux objets (*tracks* et *groups*), un objet *mixer* est construit à l'aide de la classe *Mixer*, qui inclut à la fois les pistes et les bus avec leurs paramètres inhérents.

De plus, un objet auxiliaire, *mapper*, est créé sur la base des informations des pistes audio, en utilisant la classe *Mapper*, qui est responsable de la traduction du langage naturel, propre aux ingénieurs du son, en formulations mathématiques vectorielles, propres à l'algorithme d'optimisation.

Tous les objets mentionnés ci-dessus, à savoir les pistes (*tracks*), les groupes ou bus (*groups*), le mixeur (*mixer*) et le mappeur (*mapper*), font partie de l'objet *criteria* créé par la classe *Criteria*.

L'objet *criteria* est utilisé par l'objet *optimizer* créé par la classe *Optimizer*. L'optimiseur, dans une boucle itérative et selon les critères d'optimisation, génère des règles de mixage optimales (*mix rules*) qui, avec l'aide de l'objet mixeur, créent le mixage final (*mixdown*) qui est sauvegardé ensuite sur un disque externe.

Dans ce qui suit, nous allons examiner en détail chacun des éléments du système, et considérer les classes et utilitaires auxiliaires pour aider à mettre en œuvre les mécanismes de calcul et fonctionnels de base.

3.9 Les formats de données et les unités de mesure

Nous avons décidé de transformer tous les **fichiers audio** du projet en fichiers audio au format wav 32 bit à *virgule flottante*. Par rapport aux fichiers à virgule fixe (16 ou 24 bits), les fichiers 32 bits flottants stockent les nombres dans un format à virgule flottante. Ce format est fondamentalement différent du format à virgule fixe, car les nombres dans ces fichiers wav sont stockés en "notation scientifique", avec des points décimaux et des exposants (par exemple $1.2345 \cdot 10^6$ au lieu de 1234500). Cette différence est significative car il est possible de représenter des nombres beaucoup plus grands et plus petits qu'avec une représentation en virgule fixe. Le formatage et l'encodage de mots binaires en 32 bits ne sont pas intuitifs : ils ont été optimisés pour que les ordinateurs puissent exécuter des fonctions mathématiques courantes plutôt que pour être lisibles par l'homme. Ainsi, le premier bit indique si la valeur est positive ou négative, les 8 bits suivants définissent l'exposant, et les 23 derniers bits correspondent à la mantisse.

Pour l'enregistrement d'une gamme ultra-haute-dynamique, les nombres 32 bits à virgule flottante constituent un format d'enregistrement idéal. Le principal avantage de ces fichiers est leur capacité à enregistrer des signaux dépassant 0 dBFS, et donc de gérer les problèmes de saturation. En fait, la marge de manœuvre est telle que, du point de vue de la fidélité, le réglage des gains pendant l'enregistrement n'a aucune importance. Les niveaux audio des fichiers wav en 32 bits flottants peuvent être ajustés à la hausse ou à la baisse après l'enregistrement avec la plupart des principaux logiciels *DAW*, sans bruit ni distorsion supplémentaire.

Nous avons choisi les *décibels* comme unités de **volume**. La nature de la représentation par les sens humains et ceux des animaux des changements dans le déroulement de nombreux processus physiques et biologiques est proportionnelle au logarithme de l'intensité du stimulus (la loi de Weber-Fechner). Cette caractéristique rend l'utilisation des échelles logarithmiques, des quantités logarithmiques et de leurs unités tout à fait naturelle. De plus, les ingénieurs du son ont l'habitude de penser en décibels. Par contre, pour les calculs mathématiques, nous convertissons les décibels en une valeur réelle comprise entre 0 et 1.

Pour déterminer le placement du son dans le plan horizontal, qui correspond au **panorama**, nous avons utilisé des mesures en *degrés* allant de -90° , pour un positionnement complètement à gauche, à $+90^\circ$, pour un positionnement complètement à droite car nous avons trouvé cette approche plus intuitive.

3.10 Le chargeur du projet - *Loader*

Le chargeur du projet, *Loader*, reçoit les données du projet au format *json* lues dans le fichier du projet. Il convertit ces données en variables dans l'objet *loader*. Il s'agit notamment du nom du projet, de la fréquence d'échantillonnage (*sampling rate*), des pistes audio et des données relatives à chaque piste audio, le début du projet et sa durée en secondes.

Seules les parties de fichier audio spécifiées (par début et durée) sont chargées en mémoire, afin d'optimiser son utilisation. On vérifie si la source audio est monophonique ou stéréophonique, avec une conversion en source stéréophonique si nécessaire, car tout le travail d'optimisation de mixage est effectué en stéréophonie. On vérifie également que les fichiers sont de la même longueur, et avec le même taux d'échantillonnage.

Lorsque toutes les vérifications sont réalisées, un fondu en entrée/sortie linéaire (*fade in/out*) est ajouté aux fichiers audio et ces fichiers sont normalisés avec une marge de sécurité (*headroom*) spécifiée à -12 dB.

3.11 Le chargeur des bus - *Bus*

Le chargeur des bus, *Bus*, reçoit en entrée les données du style de mixage au format *json* lues dans le fichier *mix style* et la valeur de la fréquence d'échantillonnage. Ensuite, il crée des bus principaux (*master, music, vocal, drums*), et des bus d'effets d'envoi (*sends*), selon les paramètres spécifiés dans le fichier *json*.

Tous les *inserts* sont chargés en premier, et les *sends* (effets d'envoi) à la fin. Il existe des options pour charger des plug-ins *vst*⁴ ainsi que pour charger des plug-ins internes du projet de *pedalboard* (voir la description plus loin en Section 3.17).

3.12 Le mixeur - *Mixer*

Le mixeur est l'un des principaux éléments du système. La classe *Mixer* reçoit à l'entrée des pistes (*tracks*), des groupes pré-créés (*groups*), et l'option pour activer des effets d'envoi (*sends*). Sur la base de ces informations, cette classe initialise les bus de chacun des groupes, et si nécessaire, les bus d'envoi. Dans ce travail, nous avons limité les bus d'envoi à 3 bus fixes (*Room, Plate, Hall*) avec des paramètres fixes. En effet, ils ne constituent pas l'élément principal de la tâche à résoudre.

Le mixeur, *mixer*, contient également un certain nombre de fonctions utilitaires, telles que l'activation ou la désactivation d'effets d'envoi, l'effacement des informations des bus et la sélection du groupe. Il existe également des fonctions utilitaires permettant d'appliquer des règles de mixage à une seule piste, ainsi que de mixer un groupe de pistes et le bus principal (*master*).

D'autres fonctions de service permettent d'appliquer des règles de mixage, d'envoyer la piste sur le bus de groupe et de l'envoyer sur le bus d'effet (*sends*). Il est possible également de sauvegarder le mixage final sur disque.

4. Le Virtual Studio Technology (VST) est un format ouvert de plug-in audio créé par Steinberg.

Les principales fonctions de travail correspondent à mixer toutes les pistes (`mix_all`), à mixer toutes les pistes sauf une et séparément la piste isolée elle-même (`mix_all_one`), et à mixer une paire de pistes (`mix_ij_tracks`).

Le mixeur est utilisé activement pour calculer les critères d'optimisation ainsi que pour créer le mixage final.

3.13 Les règles de mixage – *MixRules*

C'est une classe auxiliaire combinant des instructions sur la façon de mixer une piste audio, exprimées en unités naturelles pour l'ingénieur du son. Actuellement, elle contient des informations sur le volume, le panorama, les filtres passe-haut et passe-bas. A l'avenir, elle pourra être étendue à l'application de l'égaliseur et du compresseur.

Cette classe contient également des fonctions utilitaires permettant de générer une règle de mixage aléatoire et d'imprimer les règles de mixage dans un format facilement lisible.

3.14 La classe interprète – *Mapper*

La classe *Mapper* est une autre classe auxiliaire qui prend en entrée des informations sur les pistes audio, y compris les souhaits de mixage exprimés par l'ingénieur du son.

La tâche principale de cette classe est de servir de "traducteur" du langage des ingénieurs du son dans le langage des vecteurs mathématiques. En d'autres termes, l'objet *Mapper* contient deux fonctions : l'une d'elles traduit les règles de mixage en vecteur des valeurs numériques pour l'algorithme d'optimisation (`mixrules2vec`) et l'autre traduit les arguments du vecteur numérique en règles de mixage compréhensibles par l'homme (`vec2mixrules`).

3.15 L'utilitaire de traitement audio – *Audiotools*

Nous avons décidé de regrouper toutes les fonctions liées d'une manière ou d'une autre au traitement audio dans une classe contenant un ensemble de fonctions statiques indépendantes, et l'avons nommée *Audiotools*.

On peut trouver ici différents types de normalisation des pistes : selon le volume de crête spécifié en décibels ou bien selon les mesures de volume évaluées selon les normes *EBU*.

On peut aussi régler le volume de la piste et appliquer un réglage de panoramique à la piste.

On trouve plusieurs méthodes pour contrôler le panorama dans un canal stéréophonique. Nous avons choisi une méthode basée sur la transformation trigonométrique des niveaux de volume des canaux droit et gauche.

```
@staticmethod
def pan_values(pan_degree: float):
    angle = pan_degree * Metrics.PI_360
    l_gain = Metrics.SQRT2 * (math.cos(angle) - math.sin(angle))
    r_gain = Metrics.SQRT2 * (math.cos(angle) + math.sin(angle))
    return l_gain, r_gain
```

$$\begin{aligned}\alpha &= \pi \cdot \frac{pan^\circ}{360^\circ} \\ l_{gain} &= \sqrt{2} \cdot [\cos(\alpha) - \sin(\alpha)] \\ r_{gain} &= \sqrt{2} \cdot [\cos(\alpha) + \sin(\alpha)]\end{aligned}\tag{3.1}$$

On y trouve aussi des fonctions qui appliquent des filtres d'écrêtage à la piste, et une implémentation des filtres eux-mêmes. Des filtres numériques Butterworth d'ordre 5 ont été pris comme base et trois filtres (passe-haut, passe-bas et passe-courbe) ont été mis en œuvre.

```
@staticmethod
def butter_bandpass(lo_cut, hi_cut, sample_rate, order=5):
    nyq = 0.5 * sample_rate
    low = lo_cut / nyq
    high = hi_cut / nyq
    sos = butter(order, [low, high], analog=False, btype='band', output='sos')
    return sos

@staticmethod
def butter_lowpass(hi_cut, sample_rate, order=5):
    nyq = 0.5 * sample_rate
    Wn = float(hi_cut / nyq)
    sos = butter(order, Wn=Wn, analog=False, btype='low', output='sos')
    return sos

@staticmethod
def butter_highpass(lo_cut, sample_rate, order=5):
    nyq = 0.5 * sample_rate
    Wn = float(lo_cut / nyq)
    sos = butter(order, Wn=Wn, analog=False, btype='high', output='sos')
    return sos

@staticmethod
def butter_low_high_pass_filter(track, sample_rate, lo_cut, hi_cut, order=5):
    # hi and lo exists => bandpass
    if hi_cut < 20000.0 and lo_cut > 20.0:
        sos = AudioTools.butter_bandpass(lo_cut, hi_cut, sample_rate, order)
        processed_track = sosfiltfilt(sos, track, axis=0)
    # only lo
    elif lo_cut > 20.0:
        sos = AudioTools.butter_highpass(lo_cut, sample_rate, order)
        processed_track = sosfiltfilt(sos, track, axis=0)
    # only hi
    elif hi_cut < 20000.0:
        sos = AudioTools.butter_lowpass(hi_cut, sample_rate, order)
        processed_track = sosfiltfilt(sos, track, axis=0)
    # not lo and not hi
    else:
        processed_track = track.copy()
    return processed_track
```

Il existe des fonctions qui combinent à la fois les opérations de volume, de panorama et des filtres.

Il existe également des fonctions graphiques : la visualisation des pistes audio et la visualisation de l'avancement de l'optimisation.

Il existe des fonctions qui convertissent un fichier stéréophonique en sa représentation *mid-side*, une fonction qui sépare le fichier stéréophonique en deux canaux monophoniques indépendants. Il existe aussi une fonction de conversion simple d'une piste monophonique en une piste pseudo stéréophonique.

Un certain nombre de fonctions disponibles permettent de convertir le fichier audio en une représentation fréquentielle à l'aide d'une transformée de Fourier numérique.

Audiotools inclut également les fonctions qui calculent les critères partiels d'optimisation. Il s'agit notamment des critères : d'*équilibre* stéréophonique, d'*énergie* jointe, de *profondeur* et de *largeur* du mixage. Nous reviendrons plus tard sur ces fonctions dans le Chapitre 5.

Enfin, on trouve aussi une fonction de fondu (fade in/out) à des fins d'écoute esthétique.

3.16 L'utilitaire des métriques audio - *Metrics*

Metrics est une autre classe constituée d'un ensemble de fonctions statiques et indépendantes. Cette classe contient : des constantes fréquemment utilisées, des opérations de comparaison, des opérations de conversion, des métriques audio typiques et diverses métriques de similarité.

Nous donnons maintenant un aperçu des opérations de conversion.

1) Conversion de l'Amplitude en décibels et inversement

$$\begin{aligned} dB &= 20 \cdot \log_{10}(amp) \\ amp &= 10^{dB/20} \end{aligned} \tag{3.2}$$

où *dB* correspond au niveau exprimé en décibels et *amp* à l'amplitude linéaire.

```
@staticmethod
def amp2db(amp):
    if np.abs(amp) > 1.0:
        sys.exit("ERROR: abs(amp) must be lower or equal to 1.0")
    return 20.0 * np.log10(amp)

@staticmethod
def db2amp(dB):
    if dB > 0:
        sys.exit("ERROR: dB must be lower or equal to 0.0")
    return np.power(10, dB / 20.0)
```

2) Conversion des fréquences Hertz en Mel⁵ et inversement

$$\begin{aligned}mel &= 2595 \cdot \log_{10} (1 + Hz/700) \\ Hz &= 700 \cdot \left[10^{(mel/2595)} - 1 \right]\end{aligned}\tag{3.3}$$

où *mel* correspond à la valeur de la fréquence calculée en mels et *Hz* à sa valeur exprimée en Hertz.

```
@staticmethod
def hz2mel(Hz):
    return 2595 * math.log10(1 + Hz / 700)

@staticmethod
def mel2hz(mel):
    return 700 * (pow(10, mel / 2595) - 1)
```

3.17 Le paquet d'effets – *Pedalboard*

Pedalboard est une bibliothèque Python⁶ permettant d'appliquer des effets à un signal audio. Elle prend en charge un certain nombre d'effets audio courants et permet également l'utilisation des formats de plugins VST3® et Audio Unit pour les effets tiers. Elle a été conçue par l'Audio Intelligence Lab de *Spotify* pour permettre l'utilisation d'effets audio de qualité studio à partir de Python et de TensorFlow.

Pedalboard constitue une aide pour le processus de création de contenu audio puisque cette librairie permet, en effet, d'appliquer des effets au signal audio sans utiliser une station de travail audionumérique (DAW).

Pour notre mémoire, nous avons activement utilisé ce paquet pour construire une chaîne d'insertions et d'envois vers les bus de groupes. Malheureusement, le projet de *Pedalboard* se révèle, à l'utilisation, très rudimentaire et mal supporté : de nombreuses fonctions sont instables ou ne fonctionnent pas du tout. Par contre, l'idée d'une telle bibliothèque est très pertinente.

Nous avons donc dû programmer les filtres nous-mêmes. Avec *Pedalboard* il est très pratique d'utiliser des effets vst externes, malheureusement il n'y a pas de possibilité d'utiliser des effets au format vst2, et tous les plugins ne sont pas toujours bien compatibles ou bien contrôlés par *Pedalboard* à cause de la présence de nombreux bugs dans cette bibliothèque.

Néanmoins, nous avons été heureux de voir qu'un tel produit existait et ce produit a rendu beaucoup plus facile l'utilisation d'effets au format vst externes dans notre mixeur.

5. L'échelle des mels est une échelle de hauteurs jugées par les auditeurs comme étant à égale distance les unes des autres. Le point de référence entre cette échelle et la mesure normale des fréquences est défini en assimilant un son de 1000 Hz, 40 dB au-dessus du seuil de l'auditeur, à une hauteur de 1000 mels.

6. <https://github.com/spotify/pedalboard>

Chapitre 4

L'optimisation numérique

4.1 L'introduction à l'optimisation

L'application que nous avons créée et que nous présentons dans ce mémoire est basée sur une méthode d'optimisation qui cherche, itérativement, la spatialisation optimale en fonction de certains critères d'optimisation et sous contraintes. C'est la raison pour laquelle, dans ce chapitre, nous allons présenter très brièvement les méthodes d'optimisation.

Un problème d'optimisation au sens large implique qu'il existe un critère d'optimisation qui dépend d'un ou de plusieurs paramètres et que nous voulons sélectionner les valeurs de ces paramètres de manière à ce que la valeur du critère d'optimisation soit maximale ou minimale, selon l'objectif du problème.

Formalisons cela mathématiquement.

Supposons qu'il existe un vecteur de paramètres \mathbf{x} défini dans l'espace $\mathbb{X} \subset \mathbb{R}^n$, soit $\mathbf{x} \in \mathbb{X}$, et qu'il existe une fonction $f(\mathbf{x}) : \mathbb{X} \rightarrow \mathbb{R}$, appelée la *fonction objectif*, fonction qui traduit de manière unique le vecteur de paramètres \mathbf{x} de l'espace de définition \mathbb{X} dans l'espace des nombres réels $f(\mathbf{x}) \in \mathbb{R}$. Il est nécessaire alors de trouver un vecteur \mathbf{x} optimal, noté \mathbf{x}^* , qui fournira la valeur minimale de la fonction $f(\mathbf{x})$:

$$\mathbf{x}^* \in \mathbb{X} : f(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbb{X}} f(\mathbf{x}). \quad (4.1)$$

En d'autres termes, on cherche \mathbf{x}^* tel que $f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in \mathbb{X}$.

Habituellement, on considère plutôt le problème de minimisation, puisque tout problème de maximisation peut facilement être traduit en un problème de minimisation par la multiplication par -1 :

$$\max f(\cdot) \Leftrightarrow \min -f(\cdot). \quad (4.2)$$

Nous avons présenté ci-dessus un problème d'*optimisation globale*, mais si la fonction n'est pas convexe, elle peut comporter plusieurs extrema (minima ou maxima) locaux, et, on parle alors d'un problème d'*optimisation locale*, donc d'un problème de recherche du minimum le plus proche dans un voisinage donné.

Pour un voisinage de largeur $\delta > 0$, ce problème de recherche d'un extremum local s'écrit :

$$\begin{aligned} \exists \mathbf{x}^* : \quad & \forall \mathbf{x} \in \mathcal{A} \quad \text{tel que} \quad \|\mathbf{x} - \mathbf{x}^*\| \leq \delta, \\ & f(\mathbf{x}^*) \leq f(\mathbf{x}). \end{aligned} \tag{4.3}$$

Si aucune contrainte n'est imposée à l'ensemble des solutions potentielles, le problème est dit d'optimisation inconditionnelle (ou *sans contraintes*), sinon on parle d'optimisation conditionnelle (ou *sous contraintes*).

4.2 Les types des fonctions objectifs et des contraintes

Suivant la fonction objectif, on distingue plusieurs types d'optimisations :

- *l'optimisation linéaire* pour laquelle la fonction objectif et les contraintes sont linéaires :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{c} \cdot \mathbf{x}, \\ & A \cdot \mathbf{x} \leq \mathbf{b}; \end{aligned} \tag{4.4}$$

- *l'optimisation quadratique* pour laquelle la fonction objectif est quadratique et les contraintes sont linéaires :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{x}^T Q \mathbf{x} + \mathbf{c} \cdot \mathbf{x}, \\ & A \mathbf{x} \leq \mathbf{b}; \end{aligned} \tag{4.5}$$

- *l'optimisation non linéaire* pour laquelle la fonction objectif est non linéaire et les contraintes, quand elles sont présentes, peuvent être de toute nature :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}), \\ & g(\mathbf{x}) \leq 0, \\ & h(\mathbf{x}) = 0. \end{aligned} \tag{4.6}$$

Il convient de noter que les fonctions contraintes peuvent inclure à la fois des fonctions d'inégalité $g(\mathbf{x}) \leq 0$ et des fonctions d'égalité $h(\mathbf{x}) = 0$, ce qui réduit considérablement l'espace des solutions faisables et complique l'optimisation.

4.3 Les types des variables

Ci-dessus, nous avons considéré, pour simplifier, que seules les variables continues $\mathbf{x} \in \mathbb{R}^n$ étaient des arguments de la fonction objectif $f(\mathbf{x})$, et par conséquent, un tel problème d'optimisation est appelé *optimisation continue* ou simplement *optimisation* parce que le mot continu est souvent omis.

Cependant, la plupart des problèmes du monde réel ne sont que très rarement continus. Les arguments variables de l'optimisation peuvent être des nombres entiers $\mathbf{x} \in \{1, 2, 3, \dots\}^n \subset \mathbb{N}^n$. Par exemple, lors du comptage d'articles, on parle alors d'*optimisation en nombres entiers*.

Les variables peuvent être des quantités discrètes, par exemple des valeurs issues d'un ensemble standardisé des valeurs $\{1.25, 2.36, 3.47, \dots\}$, auquel cas on parle d'*optimisation discrète*.

L'*optimisation binaire* est également un cas particulier de l'optimisation en nombres entiers, où les arguments prennent les valeurs 0 ou 1 : $\mathbf{x} \in \{0|1\}^n \subset \mathbb{B}^n$. Cette approche est utile pour modéliser les événements de type "oui/non".

Une autre sous-section très importante de l'optimisation est l'*optimisation combinatoire*, où l'on recherche une *permutation* optimale π^* de certains objets : $f(\pi^*) \leq f(\pi)$ pour $\forall \pi \in \mathbb{C}^n$.

Dans le monde réel, on rencontre très rarement l'un de ces types purs d'optimisation. En général, on cherche à décrire notre problème en termes aussi naturels que possible et proches du langage humain afin d'obtenir des modèles et des résultats plus précis, et on a donc de plus en plus recours à l'*optimisation en variables mixtes*. Bien évidemment, ces tâches sont plus difficiles à résoudre.

Les scientifiques doivent souvent faire des compromis entre un modèle qui se rapproche de la réalité (approximation) et la simplicité de sa résolution ainsi que la précision de la solution optimale.

4.4 Les fonctions objectifs spécifiques

Jusqu'à présent, nous avons considéré une fonction objectif f qui convertit un vecteur d'arguments \mathbf{x} en un espace de nombres réels \mathbb{R} . Mais dans les problèmes réels, on rencontre des fonctions plus complexes. Par exemple, un bruit aléatoire ou systématique peut être présent auquel cas la fonction correspond plutôt à $f(\mathbf{x}) + \epsilon$. La fonction peut aussi varier dans le temps auquel cas on a $f(\mathbf{x}, t)$, la fonction peut être de nature stochastique auquel cas $f(\mathbf{x}) \sim \mathcal{N}(\mu, \sigma)$, ou bien encore le temps de calcul de la fonction peut être trop long pour permettre d'effectuer suffisamment d'itérations afin d'obtenir la précision requise pour la solution cherchée. Tout cela rend le processus d'optimisation excessivement compliqué et nécessite des méthodes spéciales capables de gérer des tâches aussi spécifiques.

Par exemple, pour les fonctions qui nécessitent de grandes ressources informatiques, des méthodes d'optimisation de substitution¹ ont été développées : au lieu de la fonction originale, on optimise une approximation de cette fonction dont les valeurs sont rapidement calculables.

Pour traiter une fonction bruitée (donc comportant du bruit) , il faut appliquer des filtres de réduction du bruit afin d'améliorer l'estimation de la valeur réelle de la fonction.

1. surrogate-based optimization

Une fonction de nature stochastique nécessite le développement d'estimateurs de sa vraie valeur à partir de plusieurs mesures.

4.5 L'optimisation multi critères

Jusqu'à présent, nous n'avons considéré qu'une seule fonction $f(\mathbf{x})$ qui doit être optimisée. Cependant, il existe des situations où plus d'une fonction doit être optimisée $\{f_i(\mathbf{x})\}$, $i = 1, \dots, m$, et, souvent, ces fonctions ont des critères d'optimisation contradictoires parce qu'une augmentation d'un critère entraîne une diminution de l'autre. Néanmoins, il existe de nombreux problèmes où il est nécessaire de trouver la meilleure solution possible en fonction de plusieurs critères, même s'ils sont mutuellement exclusifs. Dans ce cas, on parle d'*optimisation multi critères* :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{X}} \quad & \mathbf{f}(\mathbf{x}), \\ & g(\mathbf{x}) \leq 0, \\ & h(\mathbf{x}) = 0, \\ & \mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^n, \\ & \mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m. \end{aligned} \tag{4.7}$$

Le plus souvent, on considère 2 ou 3 critères à optimiser, mais il y a des cas où il est nécessaire d'optimiser simultanément de 5 à 10 critères voire plus, on utilise alors d'autres types des méthodes d'optimisation et on parle dans ce cas, pour les distinguer, de *many-objective optimization*².

Comme les critères peuvent souvent être contradictoires, le concept de solutions *non dominantes* est introduit pour comparer deux solutions différentes. Pour \mathbf{x} et $\hat{\mathbf{x}}$ deux solutions et $\mathbf{f}(\mathbf{x})$ et $\mathbf{f}(\hat{\mathbf{x}})$ les valeurs des fonctions objectif associées, on a (pour m critères) :

- $\mathbf{f}(\hat{\mathbf{x}})$ **domine** $\mathbf{f}(\mathbf{x})$ **strictement** (au sens fort) noté $\mathbf{f}(\hat{\mathbf{x}}) \prec \mathbf{f}(\mathbf{x})$ si, et seulement si, $f_i(\hat{\mathbf{x}}) < f_i(\mathbf{x}) \forall i = 1, \dots, m$;
- $\mathbf{f}(\hat{\mathbf{x}})$ **domine faiblement** $\mathbf{f}(\mathbf{x})$ noté $\mathbf{f}(\hat{\mathbf{x}}) \leq \mathbf{f}(\mathbf{x})$ si, et seulement si, $f_i(\hat{\mathbf{x}}) \leq f_i(\mathbf{x}) \forall i = 1, \dots, m$;
- $\mathbf{f}(\hat{\mathbf{x}})$ **domine** $\mathbf{f}(\mathbf{x})$ noté $\mathbf{f}(\hat{\mathbf{x}}) \preceq \mathbf{f}(\mathbf{x})$ si, et seulement si, $\mathbf{f}(\hat{\mathbf{x}}) \leq \mathbf{f}(\mathbf{x})$ et $\mathbf{f}(\hat{\mathbf{x}}) \neq \mathbf{f}(\mathbf{x})$.

On parle de dominance dans l'espace des objectifs mais cette notion est aussi étendue à la notion d'efficacité dans l'espace de décision :

- si $\nexists \mathbf{x} \in \mathbb{X}$ tel que $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\hat{\mathbf{x}})$ on dit que $\hat{\mathbf{x}}$ est **faiblement efficace** et le point $\mathbf{f}(\hat{\mathbf{x}})$ est **faiblement non-dominé**;
- si $\nexists \mathbf{x} \in \mathbb{X}$ tel que $\mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\hat{\mathbf{x}})$ on dit que $\hat{\mathbf{x}}$ est **efficace** et le point $\mathbf{f}(\hat{\mathbf{x}})$ est **non-dominé**.

Une solution est dite faiblement efficace si le point associé est faiblement non-dominé, c'est-à-dire s'il n'existe aucune autre solution qui la domine au sens fort. La solution est dite efficace si le point associé est non-dominé (aucune autre solution ne la domine).

L'ensemble des solutions efficaces forme, dans l'espace objectif, une *frontière de Pareto*. La frontière de Pareto correspond à la solution finale du problème d'optimisation multicritères. Il appartient alors au décideur de sélectionner dans la frontière de Pareto, résultant du calcul d'optimisation, une ou plusieurs solutions adaptées à la situation.

2. Il n'y a pas de terme français pour cela. Mais on peut proposer un. Par exemple, l'optimisation avec beaucoup d'objectifs.

4.6 Scalarisation

En général, il faut beaucoup plus de calculs (itérations) pour résoudre un problème d'optimisation multi critères. Parfois, lorsque le calcul de la fonction objectif prend beaucoup de temps, cette approche devient peu pratique. Dans ce cas, l'optimisation multi objectifs est réduite à une optimisation mono objectif, à la résolution d'un problème d'optimisation qui est beaucoup plus facile du point de vue algorithmique et des calculs.

Dans la plupart des cas, un décideur peut supposer *a priori* le poids relatif ω_i de chacune des fonctions à optimiser $f_i(\mathbf{x})$, puis, par la *scalarisation linéaire* pour un ou plusieurs ensembles k de poids prédéfinis $\{\omega_i\}^k \in \mathbb{R}^m$ où $i = 1, \dots, m$, le problème d'optimisation est réduit à la forme suivante :

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \sum_{i=1}^m \omega_i \cdot f_i(\mathbf{x}), \\ & \sum_{i=1}^m \omega_i = 1 \end{aligned} \tag{4.8}$$

Dans notre travail, nous utiliserons cette formulation, car la complexité de calcul du modèle psychoacoustique proposé est trop élevée pour être résolue par d'autres méthodes. De plus, une pondération optimale ω_i des différents critères pourra être efficacement choisie à l'avance.

4.7 Les méthodes d'optimisation

Il existe de nombreuses classifications des méthodes d'optimisation, aussi nous en proposons ici une version simplifiée mais intuitive.

Tout d'abord, et historiquement, il existe des méthodes de programmation mathématique. Il s'agit notamment de la *programmation linéaire*, qui permet de résoudre des problèmes d'optimisation linéaire. On peut distinguer ici deux domaines principaux de méthodes : (1) les méthodes du simplexe et (2) les méthodes des points intérieurs.

Les méthodes de *programmation quadratique* sont utilisées pour résoudre les problèmes d'optimisation quadratique ou linéaire. Il s'agit notamment des méthodes : (1) d'activation, (2) du gradient conjugué, (3) du lagrangien augmenté, (4) du gradient projeté et aussi (5) de points intérieurs.

Les méthodes de *programmation non linéaire* résolvent des problèmes d'optimisation non linéaire et si la fonction objectif est non-convexe, donc une fonction avec de nombreux minima locaux, on obtient le plus souvent un optimum local. Les méthodes utilisées sont : (1) de Newton ou quasi-Newton et (2) du gradient conjugué. Pour résoudre un problème non-convexe et obtenir l'optimum global des approches typiques sont (1) multi-start³ et (2) branch-and-bound⁴.

3. Le même problème d'optimisation avec différents points de départ est résolu, puis la meilleure solution est choisie.

4. Un algorithme par séparation et évaluation

Toutes les méthodes citées font partie des méthodes traditionnelles d'optimisation et doivent être utilisées en premier lieu si la fonction objectif (ou le modèle) le permet. Ces méthodes ont une convergence, rapide et prouvée, vers une solution optimale.

Parfois, la solution obtenue par ces méthodes n'est pas satisfaisante. Par exemple, dans le cas de la programmation non linéaire, on souhaite une solution globale et numériquement stable, ou s'il est impossible de formuler le modèle en termes de la programmation mathématique, ou encore si les fonctions-contraintes sont très complexes. Pour tous ces cas, nous sommes alors obligés de rechercher des approches alternatives.

L'une des premières recommandations serait d'utiliser autant de connaissances *a priori* du modèle que possible et d'intégrer ces connaissances dans l'algorithme d'optimisation. C'est ainsi que l'on arrive à des *méthodes heuristiques*, donc des méthodes créées spécifiquement pour un problème particulier.

Une autre situation où nous avons recours à une solution alternative est celle où nous ne résolvons pas un problème en variables continues, comme dans la version classique, mais, par exemple, un problème en variables en nombres entiers, en variables discrètes, binaires ou en variables mixtes, ou bien, par exemple, un problème d'optimisation combinatoire où il s'agit des permutations.

Il existe de nombreux cas d'optimisation où il est impossible d'obtenir des informations *a priori* sur un problème, ou bien où ces informations ne peuvent pas être intégrées dans l'algorithme pour une raison technique quelconque. Dans ce cas, on aboutit à une classe d'heuristiques universelles qui ont la propriété de s'adapter à tout problème. Du fait de cette polyvalence, ces méthodes sont appelées méthodes *méta-heuristiques*.

4.8 Les méthodes méta-heuristiques

Historiquement, les méthodes métaheuristiques ont été élaborées pour la recherche d'extrema locaux, et, on appelle cette méthode la *méthode d'escalade* [Johnson and Jacobson, 2002a,b], qui ressemble un peu à un algorithme de gradient simplifié. À partir de trois points aléatoires dans l'espace de recherche et des valeurs associées pour la fonction objectif, un quatrième point dans la direction de l'extremum est construit géométriquement (comme pour la méthode simplex) [Gao and Han, 2012]. L'algorithme progresse alors dans cette direction. Ensuite, l'opération est répétée itérativement jusqu'à ce qu'un certain critère de convergence, tel que la proximité des points les uns par rapport aux autres, soit atteint. Mais, trop souvent, cet algorithme converge vers l'extremum local le plus proche.

Pour surmonter ce problème de localité, une méthode de *recuit simulé* [Kirkpatrick et al., 1983] a été proposée, dans laquelle des étapes non optimales sont autorisées et où, à chaque itération, la probabilité d'accepter de faire un pas non optimal diminue selon une règle de recuit empruntée à la thermodynamique. Cette approche permet de surmonter assez bien le problème de la localité, mais elle ne garantit pas le caractère global des solutions trouvées.

Parallèlement à cette méthode, de nombreuses méthodes similaires de recherche locale améliorée ont vu le jour. Parmi les plus célèbres, on peut citer la *recherche tabou* [Glover and Laguna, 1997], où la mémoire humaine a été simulée pour se souvenir des solutions qui ont déjà été visitées, solutions constituant la liste tabou.

La véritable percée a été le passage d'une solution unique à une population de solutions. Par exemple, on peut imaginer que l'espace de recherche soit exploré simultanément par une douzaine de méthodes de recuit simulé différentes. Ce faisant, la probabilité de trouver une solution globale augmente considérablement. Les algorithmes les plus connus fonctionnant selon un tel principe sont les *algorithmes génétiques* [Holland, 1992], qui ont été inventés par association avec le croisement de gènes. À l'origine, les algorithmes génétiques fonctionnaient dans des espaces "codés", et la qualité de la solution était grandement influencée par le bon choix du codeur. La représentation binaire était souvent utilisée. Les méthodes proposées ultérieurement ont permis de supprimer le codage, grâce à l'introduction d'opérations "génétiques"⁵ plus avancées.

Ensuite, l'ère des algorithmes basés sur la population de solutions a débuté. Les scientifiques ont commencé à s'inspirer de processus complètement différents rencontrés, par exemple, dans la nature, et les nouveaux algorithmes ont été créés par imitation de ces processus. On peut citer les exemples suivant de processus imités : les statistiques pour les *stratégies d'évolution* [Beyer and Schwefel, 2002] et l'algorithme à *estimation de distribution* [Hauschild and Pelikan, 2011]; l'immunologie pour les algorithmes appelés *systèmes immunitaires artificiels* [Hart et al., 2011]; la biologie pour les *algorithmes évolutionnaires* [Fogel, 1995]; la sociologie et le comportement collectif des animaux pour les algorithmes d'*optimisation par essaims de particules* [Kennedy and Eberhart, 1995] et *optimisation par colonies de fourmis* [Dorigo and Blum, 2005]; la géométrie et les systèmes d'auto-apprentissage pour les algorithmes d'*évolution différentielle* [Feoktistov, 2006].

Le principal avantage de ces algorithmes réside dans leur capacité à rechercher collectivement une solution globale. Cela est rendu possible par de nombreux facteurs, parmi lesquels : des valeurs de départ multiples, les principes de sélection de la population d'élite, les rôles de leader local et global, la capacité à s'adapter à l'espace de recherche et la capacité à auto-apprendre en ajustant automatiquement les paramètres.

Il convient également de noter que les algorithmes basés sur la population sont les algorithmes les mieux adaptés à la résolution de problèmes d'optimisation multi critères [Tan et al., 2002]. L'existence d'une population de solutions se transforme très naturellement en une frontière de Pareto grâce aux opérations introduites de sélection des solutions non dominantes.

Tout n'est pas parfait et cela vaut également pour les algorithmes basés sur la population, dont l'un des principaux inconvénients réside dans la lenteur de leur convergence. En pratique, dans la plupart des cas, il faut beaucoup plus d'itérations que dans les approches traditionnelles pour parvenir à une bonne solution. Cela est dû, d'une part, à l'existence simultanée de nombreuses solutions qui doivent évoluer à chaque itération, et, d'autre part, à la nature stochastique des méthodes elles-mêmes.

Un autre inconvénient moins important est l'absence de preuve de la convergence théorique. Là encore, en raison de la nature stochastique des algorithmes, souvent, en partant de populations initiales différentes, la solution finale obtenue peut parfois être aussi différente. Pour pallier ce problème, on fait généralement plusieurs démarrages de l'algorithme⁶ et on sélectionne ensuite la meilleure solution.

Les algorithmes hybrides, encore connus sous le nom d'*algorithmes mémétiques* [Molina et al., 2018], sont parfois utilisés pour accélérer la convergence, en rajoutant souvent une recherche locale

5. mutation et croisement

6. restarts

autour de chaque solution de la population. Pour les problèmes de grande échelle, les approches mémétiques sont très efficaces.

4.9 La bibliothèque d'optimisation VitaOptimum

Pour optimiser le modèle présenté dans ce mémoire, nous avons décidé d'utiliser une bibliothèque d'optimisation commerciale *VitaOptimum*, aimablement accordée par la société SkyWorkflows⁷ à des fins de recherche scientifique.

Cette bibliothèque est écrite en Fortran de bas niveau et elle est optimisée pour les processeurs modernes⁸. Elle comprend plusieurs dizaines d'algorithmes pour presque tous les cas d'optimisation globale.

Ces algorithmes sont basés sur les derniers développements clés en matière d'optimisation globale, y compris des éléments d'intelligence artificielle, de statistiques avancées, d'intelligence computationnelle, d'apprentissage automatique et d'ajustement automatique des paramètres de contrôle. La bibliothèque est très simple à utiliser et conviviale. Il existe également un "wrapper" Python pour un prototypage rapide⁹, que nous avons utilisé pour construire notre système de mixage automatique (cf. Chapitre 3).

Pour mieux comprendre les utilisations permises par cette bibliothèque, nous considérons un de ses algorithmes les plus représentatifs dans sa version simplifiée, à savoir l'évolution différentielle.

7. <http://skyworkflows.com/vo/>

8. multithreading, vectorisation SIMD

9. <https://pypi.org/project/vitaoptimum/>

4.10 L'évolution différentielle

Au cours des 20 dernières années, l'algorithme d'évolution différentielle a été l'algorithme de choix pour l'optimisation globale méta-heuristique. Comme montré dans mon livre [Feoktistov, 2006], grâce à la formule d'optimisation universelle, d'autres algorithmes d'optimisation connus peuvent être considérés à travers le prisme de l'évolution différentielle comme des cas particuliers de celle-ci. Cet algorithme présente la meilleure convergence pour presque tous les types de problèmes, ce qui a été prouvé par des années de compétitions durant les conférences IEEE CEC¹⁰.

Nous proposons une brève description du fonctionnement de l'algorithme.

Un problème d'optimisation est représenté par un ensemble de variables. Ces variables forment un vecteur à D -dimensions dans l'espace continu $X = (x_1, \dots, x_D) \in \mathbb{R}^D$. Il existe un critère d'optimisation $f(X) : \mathbb{R}^D \rightarrow \mathbb{R}$, généralement appelé fonction de *fitness* ou *fonction de coût*. Le but de l'optimisation consiste alors à trouver les valeurs des variables qui minimisent le critère, c'est-à-dire à trouver :

$$X^* : f(X^*) = \min_X f(X). \quad (4.9)$$

Souvent, les variables satisfont des contraintes de bornes¹¹ :

$$L \leq X \leq H : L, H \in \mathbb{R}^D. \quad (4.10)$$

Comme tous les algorithmes évolutionnaires, l'évolution différentielle traite une *population* de solutions. La population \mathbb{P} d'une génération g est constituée de NP vecteurs, appelés *individus* de la population. Chacun de ces individus représente une solution optimale potentielle.

$$\mathbb{P}^g = X_i^g, \quad i = 1, \dots, NP. \quad (4.11)$$

De son côté, l'individu contient D variables, appelées *gènes*.

$$X_i^g = x_{i,j}^g, \quad j = 1, \dots, D \quad (4.12)$$

La population est *initialisée* en générant de manière aléatoire des individus dans les limites des contraintes de bornes :

$$\mathbb{P}^0 = x_{i,j}^0 = rand_{i,j} \cdot (h_j - l_j) + l_j, \quad (4.13)$$

où la fonction *rand* génère des valeurs aléatoires uniformément réparties dans l'intervalle $[0, 1]$.

Ensuite, à chaque nouvelle *génération*, les individus d'une population sont mis à jour au moyen d'un schéma de *reproduction*. Ainsi, pour chaque individu *ind*, un ensemble d'autres individus π est extrait aléatoirement d'une population. Pour produire un nouvel individu les opérations de *Différenciation* et de *Recombinaison* sont appliquées l'une après l'autre. Puis, la *Sélection* est utilisée pour choisir le meilleur. Examinons maintenant brièvement ces opérations.

10. Congress on Evolutionary Computation

11. boundary constraints

Nous montrons ici un modèle typique de la *Différenciation*, d'autres modèles peuvent être trouvés dans le livre [Feoktistov, 2006]. Pour cela, trois individus différents $\pi = \{\xi_1, \xi_2, \xi_3\}$ sont extraits au hasard d'une population. Ainsi, le résultat, un individu *test*¹², est donné par :

$$\tau = \xi_3 + F \cdot (\xi_2 - \xi_1), \quad (4.14)$$

où $F > 0$ est la *constante de différenciation*.

Après, l'individu d'essai τ est recombinaison avec l'individu initial *ind*. La *Recombinaison* représente un cas typique d'échange de gènes. L'individu d'essai hérite des gènes avec une certaine probabilité. Ainsi,

$$\omega_j = \begin{cases} \tau_j & \text{if } rand_j \geq C_r \\ ind_j & \text{otherwise} \end{cases} \quad (4.15)$$

où $j = 1, \dots, D$ et $C_r \in [0, 1)$ est la *constante de recombinaison*.

La *Sélection* est réalisée en comparant les valeurs de la fonction de coût de l'individu initial et l'individu test. Si l'individu test minimise mieux la fonction de coût, il remplace l'individu initial.

$$ind = \begin{cases} \omega & \text{if } f(\omega) \leq f(ind) \\ ind & \text{otherwise} \end{cases} \quad (4.16)$$

Notons qu'il n'y a que trois *paramètres de contrôle* dans cet algorithme. Ce sont NP – la taille de la population, F – la constante de différenciation et C_r – la constante de recombinaison. Quant aux conditions terminales, on peut soit fixer le nombre de générations g_{max} , soit une précision souhaitable, soit un critère de stagnation de l'optimisation.

Le schéma de l'algorithme d'évolution différentielle est présenté de la manière suivante :

12. A trial individual en anglais.

Algorithm 1 L'évolution différentielle

Require $params$:

F, Cr, NP

$f(\cdot) : \mathcal{R}^D \rightarrow \mathcal{R}$

$L, H \in \mathcal{R}^D$

g_{max}

- ▷ les paramètres de contrôle
 - ▷ la fonction objectif
 - ▷ les contraintes de borne
 - ▷ les conditions terminales

procedure DifferentialEvolution($params$)

Initialize $\mathbb{P}^0 \leftarrow \{ind_1, \dots, ind_{NP}\} \in \{L, H\}$

Evaluate $f(\mathbb{P}^0) \leftarrow \{f(ind_1), \dots, f(ind_{NP})\}$

$g \leftarrow 1$

while not stopping conditions : $g \leq g_{max}$ **do**

for all $ind \in \mathbb{P}^g$ **do**

 Sample $\pi = \{\xi_1, \xi_2, \xi_3\}$ from \mathbb{P}^g

 Reproduce

$\tau \leftarrow Differentiation(\pi, F)$

$\omega \leftarrow Crossover(\tau, ind, Cr)$

 Evaluate $f_\omega \leftarrow f(\omega)$

 Select (ω vs ind) $\rightarrow ind$

- ▷ opération souvent coûteuse

end for

$g \leftarrow g + 1$

- ▷ passer à la prochaine génération

end while

end procedure

4.11 Un exemple d'utilisation du package *VitaOptimum*

Voici un exemple de script Python utilisant le package *VitaOptimum* :

```
import os

from vitaoptimum.base import Strategy
from vitaoptimum.vo.cus import Cus
from vitaoptimum.voplus.cus import Cus as CusPlus

class Optimizer:
    def __init__(self, criteria):
        os.environ["OMP_NUM_THREADS"] = '1' # set mono thread
        self.solver = Cus(fobj=criteria.fobj,
                        dim=criteria.mapper.low.size,
                        low=criteria.mapper.low,
                        high=criteria.mapper.high,
                        nfe=1000,
                        np=10*criteria.n,
                        f=0.7,
                        cr=0.1,
                        strategy=Strategy.rand2best)

class OptimizerPlus:
    def __init__(self, criteria):
        os.environ["OMP_NUM_THREADS"] = '1' # set mono thread
        self.solver = CusPlus(fobj=criteria.fobj,
                            dim=criteria.mapper.low.size,
                            low=criteria.mapper.low,
                            high=criteria.mapper.high,
                            np=10*criteria.n, # proportional to num of tracks
                            stagnation=10*criteria.n)

from optimizer import Optimizer, OptimizerPlus

if __name__ == "__main__":
    # .....
    # optimize
    optimizer = OptimizerPlus(criteria) # advanced optimization
    # optimizer = Optimizer(criteria) # standard optimization
    results = optimizer.solver.run(verbose=True)
    # .....
```

Chapitre 5

Les critères d'optimisation

Dans ce chapitre, nous examinons toutes les questions relatives aux critères d'optimisation. Plus précisément, nous proposons l'utilisation de plusieurs fonctions objectif interdépendantes qui modélisent, à notre gré, l'évaluation de la qualité du mixage, bien que sous une forme très simplifiée. L'idée de base consiste à partir d'un modèle simple pour aboutir à un modèle complexe, et à essayer d'utiliser des modèles très primitifs pour évaluer le potentiel de la technologie du mixage automatique.

5.1 Le design de la fonction objectif

Nous partons des deux questions suivantes :

- qu'est-ce qu'est un mixage réussi ?
- quel type de tâches de mixage voulons-nous réaliser de manière automatique ?

Tout d'abord, un bon mixage est un mixage équilibré. Dans les termes les plus simples, on peut penser à un *équilibre* entre les canaux droit et gauche. À plus long terme, nous pouvons également considérer l'équilibre spectral de chaque instrument, et l'équilibre musical au sens étroit d'expression du style et de l'intention artistique, ainsi, qu'au sens plus large, à l'équilibre de la forme d'un morceau de musique. À ce stade du travail, nous nous concentrerons uniquement sur l'équilibre entre les canaux droit et gauche.

Le deuxième aspect important est l'intelligibilité du mixage, sa transparence. Souvent ce résultat est obtenu en résolvant les problèmes de *masquage*. On considère alors les problèmes de masquage de fréquences et de masquage temporel. Dans ce mémoire, nous nous concentrerons uniquement sur le masquage de fréquences, et, nous proposerons un concept très simple d'estimation du masquage basé sur le calcul de l'énergie conjointe des deux pistes audio.

Le troisième point important de la tâche de mixage consiste à donner un "grand espace" à la scène virtuelle. Ceci est réalisé simultanément en créant la *largeur* par le panoramique et la *profondeur* par la différence de volume perçu entre les instruments (ou les sources sonores) les plus éloignés et les plus proches.

Donc, pour résumer, dans ce chapitre nous allons modéliser l'équilibre, le masquage, la profondeur et la largeur de la scène, et, à sur la base de ces critères, en utilisant des techniques d'optimisation, nous allons proposer des règles de mixage optimal.

5.2 Les discussions sur les modèles psychoacoustiques, leur complexité et leur abstraction excessive

Afin de réaliser notre tâche, nous devons savoir ce que l'ingénieur du son entend non seulement pour savoir ce qu'il entend, mais, pour être, en plus, capables de le modéliser de la manière la plus réaliste possible.

Le son généré par les ondes de pression de l'air peut être mesuré avec précision grâce à des équipements modernes. Mais comprendre comment ces ondes sont reçues et interprétées dans notre cerveau n'est pas une tâche facile. Le son est un signal analogique continu, qui (en supposant que les molécules d'air soient infiniment petites) pourrait théoriquement transporter une quantité infinie d'informations (peut-être un nombre infini de fréquences, contenant des informations d'amplitude et de phase).

La compréhension des processus perceptifs nous permettra de nous concentrer sur les spécificités de l'audition et de la perception des sons. Il est également important de noter que la question de "qu'est-ce que l'on entend ?" n'est pas seulement une question sur les capacités physiologiques de l'oreille, mais aussi, à bien des égards, une question de psychologie, de clarté de la perception.

Alors comment entendons-nous ? L'audition humaine ressemble beaucoup à l'audition spectrale, en ce sens que l'oreille reconnaît la composition spectrale des ondes sonores sans analyser la phase de l'onde. En réalité, les informations de phase sont reconnues et sont essentielles à la perception directionnelle du son, mais, cette fonction est assurée par les parties du cerveau responsables du traitement du son. La différence de phase entre les ondes sonores arrivant sur l'oreille droite et l'oreille gauche nous permet de déterminer la direction de la source sonore, l'information sur la différence de phase étant de la plus haute importance, contrairement à la variation de volume perçue par les différentes oreilles. L'effet de filtrage de la transmission du cerveau joue également un rôle très important à cet égard.

Qu'est-ce que l'effet de masquage ? Dans certains cas, un son peut être masqué par un autre son. Par exemple, il peut être totalement impossible de parler à un arrêt de bus si un bus bruyant s'arrête. Cet effet s'appelle le masquage. On dit qu'un son faible est masqué s'il devient indiscernable en présence d'un son plus fort.

Ces dernières années, les scientifiques ont développé de nombreux modèles psychoacoustiques pour la perception des sons. Un aperçu des principaux modèles peut être trouvé dans l'article [Herre and Dick, 2019]. L'un des modèles les plus récemment proposés [Glasberg and Moore, 2005] serait peut-être le mieux adapté à la mise en œuvre d'un système de mixage automatique, mais ce modèle, comme les autres, présente un certain nombre de limitations.

Tout d'abord, il s'agit de modèles théoriques trop abstraits, qui sont très éloignés des conditions typiques dans lesquelles travaille un ingénieur du son. Ces modèles reposent souvent sur l'utilisation soit de signaux sinusoïdaux purs sur un fond de bruit rose parfait, soit de sons courts simples, comme le chant des oiseaux, sur un fond de bruit. Par contre, un morceau de musique a un développement logique et artistique qui s'inscrit dans le temps et une durée correspondante, et, il comporte de nombreux sons interdépendants. Souvent le masquage n'est pas dû à la présence de bruit, mais, plutôt à un son qui masque un autre son et le bruit est utilisé à des fins purement artistiques.

De plus, un ingénieur du son travaille toujours à créer sur un espace sonore à partir de signaux au moins stéréophoniques. Or, presque tous les modèles de perception psychoacoustique utilisent des signaux monophoniques, même s'il existe des adaptations plutôt discutables ou improbables à la perception stéréophonique.

Nous avons donc l'audace d'affirmer que les modèles acoustiques actuels, malgré leur sophistication computationnelle, sont très éloignés des conditions réelles de travail de l'ingénieur du son et de sa capacité d'attention auditive. En gardant à l'esprit ces faits, nous avons décidé de proposer, dans ce mémoire, un modèle très simple, mais que nous espérons suffisamment efficace, reflétant l'effet de masquage d'un son par un autre.

Il est également important de noter que, puisque nous créons un *pré-mixage* qui implique une intervention ultérieure de l'ingénieur du son, il n'est donc pas nécessaire de rechercher une très grande précision pour les calculs réalisés pour les paramètres étudiés, lors de l'utilisation du modèle psychoacoustique, puisqu'il est inutile de chercher la "meilleure" solution à ce problème au sens des critères d'optimisation mathématique. Il suffit donc de trouver une tendance similaire à la situation réelle, par exemple le masquage de fréquences, et d'essayer de faire quelques pas d'optimisation en direction de la solution du problème selon le modèle adopté.

Une autre confirmation importante de l'inadéquation des modèles psychoacoustiques actuels pour résoudre le problème du mixage par optimisation tient au fait que l'optimisation est un algorithme itératif qui nécessite de calculer un grand nombre de fois le critère d'optimisation. Or, si le critère d'optimisation est très difficile à satisfaire en termes de temps de calcul, alors le problème devient insoluble et l'approche n'est pas viable, puisque que, dans le monde réel, un ingénieur du son ne dispose pas de beaucoup de ressources informatiques et n'est pas non plus prêt à attendre plus de quelques minutes pour obtenir un résultat (un pré-mixage). Aussi, un temps trop long pour calculer un pré-mixage est quasi rédhibitoire car cela va réduire fortement l'efficacité de l'ingénieur du son.

5.3 Les transformations de Fourier

La transformée de Fourier à court terme¹ (STFT) est une transformée de Fourier utilisée pour déterminer l'ensemble des fréquences et des phases d'un jeu limité de sinusoides pour des portions locales d'un signal lorsqu'il évolue dans le temps, portions nommées fenêtres temporelles. En pratique, la procédure de calcul des STFT consiste à calculer une transformée de Fourier pour chacune des fenêtres temporelles. On trace ensuite généralement les spectres d'amplitude en fonction du temps, ce que l'on appelle un spectrogramme ou un graphique en cascade².

Les STFT ainsi que les transformées de Fourier standard (calculées pour le signal complet) sont fréquemment utilisées pour analyser la musique. Le spectrogramme peut, par exemple, être tracé avec les fréquences selon l'axe horizontal, les fréquences les plus basses à gauche et les plus hautes à droite. La hauteur de chaque barre représente l'amplitude des fréquences calculées. La profondeur représente le temps et chaque nouvelle barre correspond à un spectre distinct. Les ingénieurs du son utilisent ce type de tracé pour obtenir des informations sur un échantillon audio, par exemple, afin de localiser les fréquences de bruits spécifiques ou pour trouver les fréquences qui peuvent s'avérer plus ou moins résonnantes. Ces informations peuvent aussi être utilisées pour l'égalisation ou le réglage d'autres effets audio.

1. Short-time Fourier transform en anglais

2. waterfall en anglais

Pour ce qui concerne notre travail, nous n'avons pas besoin, à ce stade, de travailler avec des changements dynamiques du spectre au cours du temps. Nous sommes plus intéressés par des estimations statistiques en fonction du temps. Pour ce faire, nous calculons l'amplitude moyenne, selon l'axe temporel, pour chaque bande de fréquences (`freq_bins`), puis nous normalisons ces amplitudes afin qu'elles restent comprises entre 0 et 1.

Mathématiquement, cela s'écrit comme suit :

$$\text{STFT} \{x(t)\}(\tau, \omega) \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)\omega(t - \tau)e^{-i\omega t} dt \quad (5.1)$$

où $\omega(t)$ correspond à la fenêtre de pondération, généralement une fenêtre de Hann, et $x(t)$ au signal à transformer. $X(\tau, \omega)$ est la transformée de Fourier du signal $x(t)\omega(t - \tau)$, donc une fonction complexe représentant la phase et l'amplitude du signal en fonction du temps et de la fréquence. La fonction intervenant dans la STFT correspond à la multiplication du signal à transformer $x(t)$ par la fonction de fenêtre $\omega(t - \tau)$, de durée limitée et décalée de τ . Chaque STFT correspond donc à la transformée de Fourier d'un signal de durée limitée ce qui justifie son appellation de transformée de Fourier fenêtrée ou de transformée de Fourier dépendante du temps ou bien encore de transformée de Fourier à court terme.

L'implémentation en code Python est la suivante :

```
def time2freq(mono, settings):
    W = settings['W']
    stft = librosa.stft(mono,
                       n_fft=settings['N'],
                       win_length=settings['M'],
                       hop_length=settings['H'],
                       window=W)
    mag, phase = librosa.magphase(stft)
    mag /= np.sum(W) # normalizing stft
    freq_bins = np.average(mag, axis=1)
    return freq_bins
```

Ensuite, on normalise les valeurs de fréquences entre 0 et 1 :

```
def time2freq_norm(mono, settings):
    freq = AudioTools.time2freq(mono, settings)
    freq /= np.max(freq) # normalize in [0, 1]
    return freq
```

avec les paramètres suivants :

```
b20 = round(20 * 2**15 / tracks.sample_rate)
b20k = round(20000 * 2 ** 15 / tracks.sample_rate)
settings = {'sr': tracks.sample_rate, # sample rate
           'N': 2**15, # FFT size
           'M': 2**15, # window size
           'H': 2**15 // 4, # hop size
           'W': np.hanning(2**15), # hanning window
           'b20': b20, # bin corresponding to 20Hz
           'b20k': b20k # bin corresponding to 20kHz
}
```


Il est à noter que tous les calculs ultérieurs dans la gamme de fréquences seront effectués strictement entre $20Hz$ et $20kHz$, nous avons donc calculé les valeurs des *bins* (bandes de fréquences) correspondant comme b^{20} et b^{20k} .

5.4 L'énergie et l'équilibre des canaux droit et gauche

Pour commencer, introduisons le concept d'énergie spécifique à une bande de fréquences f donnée :

$$\mathbb{E}_x(f) = \hat{x}^2(f), \quad \forall f \in [b^{20}, b^{20k}] . \quad (5.2)$$

où $\hat{x}(f)$ est l'amplitude moyenne normalisée de la bande de fréquences f .

L'équilibre entre les canaux gauche L et droit R peut alors être calculé selon la formule suivante :

$$\mathbb{B} = \sum_f |\mathbb{E}_L(f) - \mathbb{E}_R(f)| \quad (5.3)$$

L'implémentation en code Python est la suivante :

```
@staticmethod
def balance_stereo(stereo, settings):
    l, r = AudioTools.stereo_split(stereo)
    l_freq = AudioTools.time2freq_norm(l, settings)
    r_freq = AudioTools.time2freq_norm(r, settings)
    b20 = settings['b20']
    b20k = settings['b20k']
    balance = np.sum(np.abs(l_freq[b20:b20k]**2 - r_freq[b20:b20k]**2))
    return balance
```

5.5 L'énergie conjointe et le masquage

Soit $\hat{x}(f)$ et $\hat{y}(f)$ sont les amplitudes moyennes normalisées de deux signaux $x(t)$ et $y(t)$ pour une bande de fréquences f .

Alors, l'énergie conjointe à la bande de fréquences f pour les deux pistes audio x et y pourra être calculée à l'aide de la formule suivante :

$$\mathbb{E}_{xy}(f) = \hat{x}(f) \cdot \hat{y}(f). \quad (5.4)$$

Notez que les valeurs de \hat{x} et \hat{y} sont comprises entre 0 et 1 et que la multiplication de \hat{x} par \hat{y} a les propriétés suivantes :

- Si l'amplitude du signal $\hat{x}(f)$ est beaucoup plus petite que celle de $\hat{y}(f)$, ou si l'amplitude de signal $\hat{y}(f)$ est beaucoup plus petite que celle de $\hat{x}(f)$, dans les deux cas cela signifie que les signaux ne se masquent pas mutuellement sur la bande de fréquences f , auquel cas leur énergie conjointe tend vers zéro : $\mathbb{E}_{xy}(f) \rightarrow 0$.
- La suppression d'un des signaux dans une bande de fréquences donnée annule l'énergie conjointe à cette bande de fréquences : $(\hat{x}(f) = 0 \vee \hat{y}(f) = 0) \Rightarrow \mathbb{E}_{xy}(f) = 0$.
- L'énergie conjointe des deux signaux normalisés ne peut être supérieure à 1 pour une bande de fréquences sélectionnée : $\max \mathbb{E}_{xy}(f) = 1$.

- La diminution de la valeur de la fonction d'énergie conjointe correspond au processus de séparation en fréquence des deux signaux, c'est-à-dire au *démasquage*.

Ainsi, l'énergie conjointe $\mathbb{E}_{xy}(f)$ peut être considérée comme un modèle simplifié de masquage de deux signaux.

Le masquage pour deux signaux monophoniques $x(t)$ et $y(t)$ est alors défini par :

$$\mathbb{M}_{xy}^{\text{mono}} = \sum_f \hat{x}(f) \cdot \hat{y}(f), \quad \forall f \in [b^{20}, b^{20k}]. \quad (5.5)$$

Pour calculer le masquage d'un signal stéréophonique (L, R) , on doit d'abord décomposer le signal stéréophonique en ses parties mid, $M = L + R$, et side, $S = L - R$. Ensuite on calcule le masquage pour ces composantes M et S séparément.

Cette transformation est nécessaire pour supprimer toute corrélation éventuelle entre les canaux gauche L et droit R . Indirectement, cela correspond à la vérification de la compatibilité monophonique (*mono check*), ce qui résout le problème du masquage en mode monophonique, comme la plupart des ingénieurs du son préfèrent le faire.

Ainsi, le masquage de deux signaux stéréophoniques $x(t)$ et $y(t)$ se calcule de la façon suivante :

$$\mathbb{M}_{xy}^{\text{stereo}} = \mathbb{M}_{xy}^M + \mathbb{M}_{xy}^S. \quad (5.6)$$

L'implémentation en code Python s'écrit :

```
@staticmethod
def joint_energy(stereo_a, stereo_b, settings, bins: None):
    mid_a, side_a = AudioTools.stereo_to_midside_norm(stereo_a)
    mid_b, side_b = AudioTools.stereo_to_midside_norm(stereo_b)
    freq_ma, freq_sa, freq_mb, freq_sb = AudioTools.time2freq_joint_norm(mid_a, side_a, mid_b,
                                                                           side_b, settings)
    energy = np.sum(freq_ma[bins] * freq_mb[bins] + freq_sa[bins] * freq_sb[bins])
    return energy
```

5.6 Les expériences sur l'attention humaine et le caractère aléatoire des calculs

Dans notre travail de mémoire, nous avons mené une petite expérience pour simuler l'attention humaine. Nous avons émis l'hypothèse que lorsqu'un ingénieur du son travaille avec du matériel musical, son attention ne couvre pas l'ensemble du spectre de fréquences du son en une seule fois, mais passe plutôt par certaines plages de fréquences établies (habituelles). Nous avons également supposé que l'on ne regarde pas (n'écoute pas) séquentiellement et en détail toutes les fréquences de cette bande où se trouve actuellement son attention, mais que l'on se concentre sélectivement sur certaines d'entre elles, comme celles qui sont moins homogènes par exemple.

Sur la base de cette hypothèse, nous avons développé une méthode stochastique d'estimation du masquage. Nous avons supposé que cette approche permettrait d'imiter un travail plus plausible de l'ingénieur du son et, que, par effet secondaire, cette approche permettrait de réduire la quantité de calculs nécessaires.

```
@staticmethod
def classical():
    return [(20, 59, 5),
            (60, 249, 9),
            (250, 499, 9),
            (500, 1999, 7),
            (2000, 3999, 5),
            (4000, 5999, 5),
            (6000, 20000, 5)]
```

```
@staticmethod
def pascal_spitz():
    return [(20, 29, 3),
            (30, 59, 3),
            (60, 79, 3),
            (80, 99, 3),
            (100, 119, 3),
            (120, 249, 7),
            (250, 499, 9),
            (500, 999, 7),
            (1000, 1999, 5),
            (2000, 3999, 5),
            (4000, 7999, 5),
            (8000, 15999, 5),
            (16000, 20000, 3)]
```

Figure 5.1 – Deux découpages en bandes de fréquences : 1) classique ; 2) recommandé par Pascal Spitz.

Plus précisément, nous avons programmé à cet effet deux types de découpages de l'ensemble de la gamme de fréquences en bandes (Figure 5.1). La premier découpage est assez *classique* tandis que le second est celui recommandé par Pascal Spitz dans le cadre des cours qu'il dispense dans le cadre du Master Son de l'ENS Louis-Lumière. Ces découpages fréquentiels sont définis dans le code du programme, dans le fichier `ranges.py`, à l'aide de la classe *RangeAttention*.

Chaque bande de fréquences (Figure 5.6) est caractérisée par trois nombres : le premier et le deuxième nombres correspondent aux fréquences de début et de fin de cette bande, le troisième décrit le degré d'attention que l'on souhaite accorder à l'analyse des fréquences dans cette bande fréquentielle.

Ainsi, la classe *RangeAttention* met en œuvre les méthodes stochastiques (`put_attention` et `put_attention_bins`) nécessaires à l'application du mécanisme d'attention décrit ci-dessus.

```
@staticmethod
def put_attention(ranges: list):
    alist = list()
    for low, high, count in ranges:
        a = np.random.randint(low, high, count)
        alist.append(a.sort())
    attention_on_frequencies = np.concatenate(alist)
    return attention_on_frequencies
```

5.7 Les expériences avec des types de mixages pour le calcul du masquage

L'objet mixeur (Section 3.12) permet trois types de mixage : mixage de toutes les pistes (*mix_all*) ; mixage d'une piste "contre" l'ensemble du mixage à l'exception de cette piste (*mix_all_one*) ; mixage d'une paire de pistes (*mix_ij_tracks*).

Nous avons utilisé les deux derniers types de mixage (*mix_all_one* et *mix_ij_tracks*) pour évaluer l'effet du masquage de fréquence : (1) soit pour évaluer comment une piste est masquée par le reste du mixage (*mix_all_one*), (2) soit pour évaluer comment une paire de pistes se masquent mutuellement (*mix_all_one* et *mix_ij_tracks*).

Si l'ensemble du mixage contient t pistes, alors t mixages différents doivent être réalisés pour calculer le premier critère \mathbb{M}^1 , et, pour calculer le deuxième critère \mathbb{M}^2 , il faut calculer $t(t-1)/2$ mixages.

$$\mathbb{M}^1 = \sum_{i=1}^t \mathbb{M}_{\text{all_one}} = \sum_{i=1}^t \sum_f \hat{x}_{\text{all}}(f) \cdot \hat{y}_{\text{one}}(f) \quad (5.7)$$

$$\mathbb{M}^2 = \sum_{i,j: i>j}^{t(t-1)/2} \mathbb{M}_{ij} = \sum_{i,j: i>j}^{t(t-1)/2} \sum_f \hat{x}_i(f) \cdot \hat{y}_j(f) \quad (5.8)$$

Le calcul du premier type de critère \mathbb{M}^1 est plus laborieux que le second \mathbb{M}^2 pour lequel seules deux pistes doivent être calculées.

De plus, après une courte série d'expériences, nous avons remarqué une très forte corrélation entre les valeurs des critères du premier et du deuxième type : $\mathbb{M}^1 \sim \mathbb{M}^2$. Sur la base de cette observation, nous avons décidé d'abandonner le critère le plus difficile à calculer, \mathbb{M}^1 , sans compromettre le résultat final.

5.8 Les ressources mémoire nécessaire à un calcul performant

Les opérations de calcul des fonctions objectifs des critères d'optimisation nécessitent des quantités relativement importantes de mémoire (RAM). Étant donné que nous développons un *prototype* et que nous sommes limités par l'utilisation d'un ordinateur portable, nous avons sacrifié la vitesse de calcul au profit de la réduction des besoins en mémoire.

Plus précisément, nous n'avons pas effectué de calculs parallèles, *a priori* possibles, ni stocké des résultats intermédiaires dans une mémoire qui pourrait être réutilisée. La complexité de calcul du problème est devenue quadratique ($\mathcal{O}(t^2)$) au lieu d'être éventuellement linéaire ($\mathcal{O}(t)$) où t correspond au nombre de pistes dans le mixage, mais, par contre, les besoins en ressources mémoire sont devenus linéaires ($\mathcal{O}(\xi)$) au lieu d'être quadratiques ($\mathcal{O}(\xi^2)$) où ξ correspond au nombre des résultats STFT (intermédiaires) à sauvegarder dans la mémoire.

5.9 Les modèles de profondeur et de largeur, distance entre les instruments

Au départ, nous avons commencé nos recherches uniquement avec les critères d'équilibre entre les canaux stéréophoniques et de minimisation des effets de masquage des pistes. Cependant, nous nous sommes rapidement rendus compte que l'utilisation de ces deux critères n'est pas suffisante, car ces critères conduisent à concentrer tous les instruments les plus proches du centre en un seul "point", ce qui est contraire aux attentes associées à un bon mixage.

Un bon mixage suppose également, outre sa clarté et sa transparence, un espace volumineux, c'est-à-dire à la fois une sensation de profondeur de l'espace et de largeur de celui-ci. Nous avons donc décidé d'introduire deux critères d'optimisation supplémentaires intervenant dans la construction de cet espace.

Afin de construire un espace mathématique quelconque, il est nécessaire d'introduire deux concepts : l'*objet* et la *distance* entre deux objets. Dans le cas que nous considérons, les objets sont des instruments positionnés dans la scène 3D virtuelle.

Idéalement, nous aurions pu construire une base tridimensionnelle (volume, panorama, fréquence) et positionner les instruments à partir de leurs coordonnées dans cette base. Mais nous avons choisi une voie plus simple et utilisé des *projections* de cet espace tridimensionnel sur des axes : l'axe du volume, qui correspond à la *profondeur* du mixage, et sur l'axe du panorama, qui correspond à la *largeur* du mixage. Ainsi, nous avons réduit le problème tridimensionnel, où la troisième coordonnée, la fréquence, n'a pas de représentation mathématique claire, à deux projections indépendantes.

Si on suppose que chaque instrument du mixage a un volume v , alors le critère de *profondeur* (*Depth*) pour le mixage s'écrit :

$$\mathbb{D} = \frac{1}{t} \sum_{i,j: i>j}^{t(t-1)/2} |v_i - v_j| , \quad (5.9)$$

où t correspond au nombre de pistes audio (*tracks*) prises en compte dans le mixage.

De la même manière, si chaque instrument du mixage a une valeur de panorama p , alors le critère de *largeur* (*Width*) pour le mixage s'écrit :

$$\mathbb{W} = \frac{1}{t} \sum_{i,j: i>j}^{t(t-1)/2} |p_i - p_j| . \quad (5.10)$$

5.10 Le problème d'optimisation multi critères

Au total, nous avons quatre critères d'optimisation indépendants, chacun travaillant dans son propre espace de recherche et fournissant généralement une évaluation de la qualité du mixage. Il s'agit notamment de l'équilibre \mathbb{B} entre les canaux droit et gauche de l'ensemble du mixage, du critère qui évalue le degré de masquage \mathbb{M} de l'ensemble du mixage, du critère qui est responsable de la profondeur \mathbb{D} du mixage et du critère qui est responsable de la largeur \mathbb{W} du mixage.

Nous aurions pu résoudre le problème d'optimisation multi critères pour ces quatre critères mais il faut tenir compte du fait que nous avons choisi des exigences plus élevées en matière de vitesse de calcul et qu'aucune exigence de convergence et de précision absolues pour la solution optimale. L'algorithme d'optimisation devra donc effectuer de nombreuses itérations de calcul de ces critères. Nous avons donc décidé de ne pas considérer les quatre critères mais plutôt leur somme pondérée, ce qui conduit à résoudre le problème d'optimisation pour ce seul critère complexe.

$$f_{obj} = \omega_B \cdot \mathbb{B} + \omega_M \cdot \mathbb{M} + \omega_D \cdot \mathbb{D} + \omega_W \cdot \mathbb{W}, \quad (5.11)$$

où $\mathbb{B}, \mathbb{M}, \mathbb{D}, \mathbb{W}$ sont les critères décrits en amont et ω_k à leurs poids relatifs, fonction de nos préférences personnelles. Pour des raisons de commodité, une contrainte additionnelle de normalisation peut être imposée aux poids relatifs : $\sum_k \omega_k = 1$.

Chapitre 6

Les résultats et les tests perceptifs

Dans ce chapitre, nous présentons les résultats fournis par le système de mixage automatique que nous avons imaginé. Puis, nous décrivons les tests perceptifs que nous avons développé et réalisé pour évaluer objectivement les mixages optimisés obtenus. Pour ce faire, nous avons opéré une sélection de morceaux de musique.

6.1 La sélection des projets musicaux à mixer

Toutes les musiques sont tirées du site web <https://www.cambridge-mt.com/ms/mtk/>. Ce site propose une liste de projets multipistes qui peuvent être téléchargés gratuitement pour s'entraîner au mixage. Tous ces projets sont présentés sous forme d'archives zip contenant des fichiers wav non compressés (résolution de 24 ou 16 bits et fréquence d'échantillonnage de 44,1 kHz). Afin de proposer un éventail aussi large que possible d'exercices de mixage, les contributeurs ont fait tout leur possible pour fournir des fichiers audio "bruts", c'est-à-dire sans effets ou traitements supplémentaires.

La sélection des œuvres musicales a été effectuée sur la base des critères suivants :

- nous voulions explorer autant de styles de musique différents que possible, allant du classique au jazz-rock en passant par le jazz et la musique pop ;
- nous voulions sélectionner des morceaux de musique par ordre de complexité croissante pour le mixage, donc par ordre croissant du nombre de pistes considérées dans le mixage ;
- nous avons choisi des morceaux qui nous semblaient agréables à l'oreille pour la plupart des auditeurs potentiels.

Nous avons donc choisi la liste de pièces présentée dans le Tableau 6.1.

6.2 La convergence et l'ensemble de solutions

Le processus d'optimisation est un processus itératif. À chaque itération, de nouvelles solutions sont créées et évaluées. Certaines des nouvelles solutions, selon les critères d'optimisation, s'avèrent meilleures que les solutions trouvées précédemment. Comme après un nombre suffisant d'itérations de l'algorithme, il devient impossible de trouver de nouvelles solutions qui améliorent les critères d'optimisation, il est alors habituel de parler de *stagnation* de l'algorithme, s'il bloque sur un optimum local, ou bien un *optimum global* dans les autres cas. Le processus dynamique

n°	Style	Nom	n° pistes
1	Classic	Rod Alexander - Canon	2
2	Folk	Leslie Mendelson - The Hardest Part 2	4
3	Ska	Big Stone Culture - Fragile Thoughts	7
4	Latino	Eddie Garrido - Una Semana Sin Ti	10
5	Country	Angela Thomas Wade - Milk Cow Blues	11
6	Choir	Don Camillo Choir - Marsh Marigolds Song	12
7	Jazz	Araujo - The Saga Of Harrison Crabfeathers	13
8	Ballad	MR 1001 - Jay Menon	16

Table 6.1 – La liste de morceaux musicaux (multi-pistes) utilisés.

d'amélioration de la solution courante jusqu'à l'optimum global est appelé la *convergence* de l'algorithme. S'il faut un petit nombre d'itérations pour atteindre l'optimum global, on dit que l'algorithme a une convergence rapide.

Chaque solution à un problème d'optimisation constitue un vecteur de valeurs. Nous traduisons ce vecteur en un ensemble de règles de mixage pour les pistes audio incluses dans le projet. Ainsi, dans la suite de ce mémoire, nous ferons référence aux règles de mixage comme à une solution optimale.

Voici un exemple de règles de mixage :

```
6th mix : fobj = 1.042
1: MixRules(vol=-13dB, pan=-2°, locut=196Hz, hicut=18886Hz) 01_Percussion
2: MixRules(vol=-15dB, pan=-75°, locut=223Hz, hicut=15162Hz) 02_Woodblock
3: MixRules(vol=-17dB, pan=-36°, locut=71Hz, hicut=19331Hz) 03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=36Hz, hicut=6490Hz) 04_Bass
5: MixRules(vol=-12dB, pan=73°, locut=169Hz, hicut=12808Hz) 05_Piano
6: MixRules(vol=-15dB, pan=-87°, locut=87Hz, hicut=9005Hz) 06_Vibes
7: MixRules(vol=-32dB, pan=56°, locut=146Hz, hicut=10987Hz) 07_Strings1
8: MixRules(vol=-22dB, pan=35°, locut=182Hz, hicut=13989Hz) 08_Strings2
9: MixRules(vol=-34dB, pan=-57°, locut=473Hz, hicut=15274Hz) 09_FrenchHorns
10: MixRules(vol=-6dB, pan=0°, locut=56Hz, hicut=18265Hz) 10_LeadVox
```

Ces règles de mixage doivent être interprétées comme suit. La première ligne nous indique le numéro de la solution parmi toutes les solutions optimales trouvées, qui peuvent être multiples. La première ligne indique également la valeur de la fonction objectif, qui nous permet de comparer deux ou plusieurs solutions et de voir si une valeur plus petite de la fonction objectif correspond à une meilleure qualité de mixage.

La série des lignes suivantes concerne les instruments, les pistes audio, impliquées dans le mixage. A droite, nous voyons le nom de cette piste, indiquant souvent le type de signal audio (instrument, voix, percussion).

Chaque règle comporte quatre parties : *vol*, *pan*, *locut*, *hicut*. Ce sont les quatre paramètres de la piste audio qui sont optimisés par l'algorithme et grâce auxquels nous créons le mixage spatial.

Plusieurs solutions optimales, correspondant souvent à plusieurs minima locaux, peuvent être obtenues au moyen d'un algorithme d'optimisation. Par conséquent, dans notre travail, pour chaque mixage, nous avons présenté tout un ensemble de solutions optimales, qui ne sont pas très différentes dans les valeurs du critère d'optimalité, mais, par contre, très différentes du point de vue des règles de mixage et, par conséquent, sensiblement différentes dans l'esthétique et la qualité du mixage. Toutes les solutions (*MixRules*) pour tous les morceaux musicaux de Table 6.1 sont rassemblées dans l'Annexe E.

Les fichiers audio de ces mixages peuvent être écoutés en bonne qualité sans compression. Ils sont disponibles sur la page internet pointant vers le *Questionnaire* [Vital, 2021c], questionnaire que nous avons développé pour évaluer subjectivement la qualité des critères d'optimisation et l'utilité du système de mixage dans son ensemble.

6.3 Le questionnaire : les tests perceptifs

Nous avons élaboré un questionnaire visant à réaliser des tests perceptifs. Nous avons identifié 4 groupes cibles de participants :

1. les ingénieurs du son professionnels ;
2. les étudiants, futurs ingénieurs du son ;
3. les musiciens professionnels ;
4. les mélomanes et audiophiles avertis.

Dans l'entête du questionnaire nous avons indiqué qu'il s'agit d'évaluer la création automatique de pré-mixages d'un morceau de musique, qui pourraient servir de point de départ à l'ingénieur du son pour continuer ensuite le mixage, et, qu'éventuellement, ce pré-mixage automatise une partie importante du travail de routine. Ce sont ces hypothèses que nous voulons vérifier à l'aide de tests perceptifs.

Il convient également de noter que l'éventail des opérations effectuées dans ce travail est restreint : volume, panorama, filtres de fréquences basses et aigues. De plus, l'optimisation (automatisation) ne se fait que pour 4 critères fixes (cf. Chapitre 5) : équilibre, masquage, profondeur et largeur.

Nous avons demandé aux participants d'adopter les meilleures conditions d'écoute possibles à savoir : une salle acoustiquement préparée si possible, des convertisseurs et un système de monitoring professionnel, ou au minimum un casque.

Pour chacune des pistes (cf. Table 6.1), nous leur avons proposé plusieurs pré-mixages à écouter, ordonnés de manière aléatoire, et, nous leur avons demandé de choisir le meilleur parmi toutes les pré-mixages proposés. Nous avons également demandé de justifier ce choix. Et nous avons ensuite demandé aux participants s'ils envisageraient d'utiliser le pré-mixage choisi comme point de départ pour une session de mixage et nous leur avons également demandé d'expliquer leur réponse de manière détaillée.

Que voulons-nous apprendre à travers ce questionnaire ?

- Est-ce que les pré-mixages obtenus peuvent être utilisés pour des sessions futures de mixages ?
- Quelle catégorie d'utilisateurs exprime le plus grand besoin de tels pré-mixages ?
- Dans quelle mesure les critères d'optimalité utilisés pour décrire la qualité du mixage se révèlent-ils efficaces ?
- Est-ce qu'on peut constater une tendance selon laquelle une diminution de la valeur de la fonction objectif correspond à une augmentation de la qualité (subjective) du mixage ?
- Est-il nécessaire de rechercher un optimum global, ou bien est-ce qu'il est suffisant de trouver, par exemple, les trois premières solutions optimales et est-ce qu'elles peuvent constituer une solution de départ suffisante pour continuer la session ?
- L'optimum global est-il le meilleur du point de vue de l'ingénieur du son ?
- Quelles sont les opérations supplémentaires dont les utilisateurs pourraient souhaiter l'automatisation ? Par exemple : quid du travail sur l'équilibre spectral, du travail avec l'équilibre musical, du travail sur la forme, la compression, le démasquage de piste par l'égaliseur, etc.
- Quelles pourraient être les autres opérations indispensables à automatiser pour obtenir un meilleur pré-mixage ?

6.4 L'interprétation des résultats

Plusieurs catégories d'utilisateurs potentiels ont pris part aux tests perceptifs : des ingénieurs du son professionnels expérimentés, des étudiants en formation d'ingénieur du son, des musiciens professionnels et des audiophiles amateurs. La répartition en pourcentage des utilisateurs par catégorie est présentée dans la figure ci-dessous (cf. Figure 6.1).

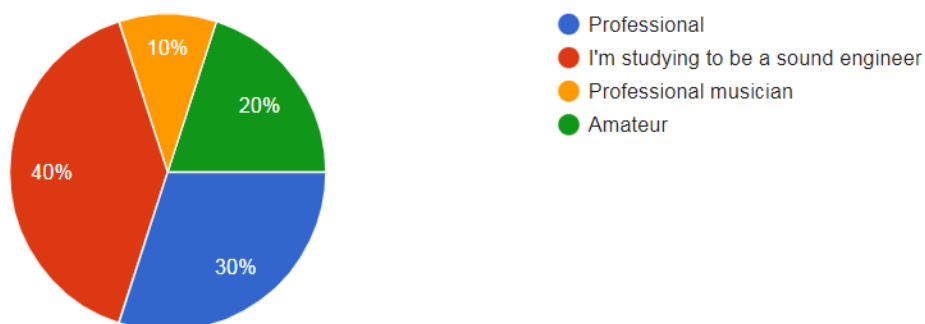


Figure 6.1 – La répartition en pourcentages des participants aux tests perceptifs.

Nous avons résumé les résultats dans Annexe E. Pour chaque composition du Tableau 6.1, nous avons fourni les données suivantes :

- l'ensemble de règles de mixage sous forme textuelle ;

- l’histogramme de préférence utilisateur pour un pré-mixage particulier ;
- la taux d’intention (en pourcentage) d’utiliser un pré-mixage sélectionné dans le travail de mixage ultérieur ;
- le graphe de convergence de la fonction objectif du modèle psychoacoustique étendu par des informations sur les préférences des différentes catégories de participants ;
- le graphe récapitulatif des paramètres du modèle psychoacoustique (volume, panorama, filtres d’écrtage passe-bas et passe-haut) ;
- les graphes individuels de la dépendance des paramètres du modèle de la fonction objectif pour chacun des instruments.

Nous allons maintenant examiner en détail un exemple d’analyse des informations pouvant être déduites des graphiques considérés.

6.4.1 Le graphe de convergence de la fonction objectif

Le graphique de convergence de la fonction objectif du modèle psychoacoustique est le reflet des performances de l’algorithme d’optimisation dans le temps. A chaque nouvelle itération de l’algorithme, la nouvelle solution est comparée à la meilleure solution trouvée lors des itérations précédentes, si la fonction objectif de la nouvelle solution est inférieure en valeur, alors la nouvelle solution se voit attribuer le statut de la meilleure solution actuelle, et, sur le graphique, cela se traduit par un point (de couleur bleue) dont l’abscisse correspond au nombre d’itérations et l’ordonnée à la valeur actuelle de la fonction objectif. Dans ce cas, de nouvelles règles de mixage correspondant à ce point sont générées et stockées sous forme de texte. De plus, en utilisant ces règles de mixage, un pré-mixage (*wav*) est créé, stocké et ensuite utilisé dans les tests perceptifs.

Le tout dernier point du graphique est créé artificiellement pour plus de clarté afin de montrer combien d’itérations ont été faites au total.

De cette façon, le graphique montre clairement la variation (diminution) de la fonction objectif au fil du temps (itérations de l’algorithme). Il montre la dynamique des changements, la gamme des changements de la fonction objectif, le nombre de solutions trouvées et l’effort nécessaire pour trouver chaque nouvelle solution.

Par *solution*, ici et ci-après, nous entendons chaque nouvel ensemble de règles de mixage qui améliore (diminue) la valeur actuelle de la fonction objectif du modèle psychoacoustique.

À des fins de démonstration, nous avons choisi l’un des graphiques, celui pour *Choir* (cf. Figure 6.2), car nous pensons qu’il peut illustrer la plupart des caractéristiques communes aux autres graphiques.

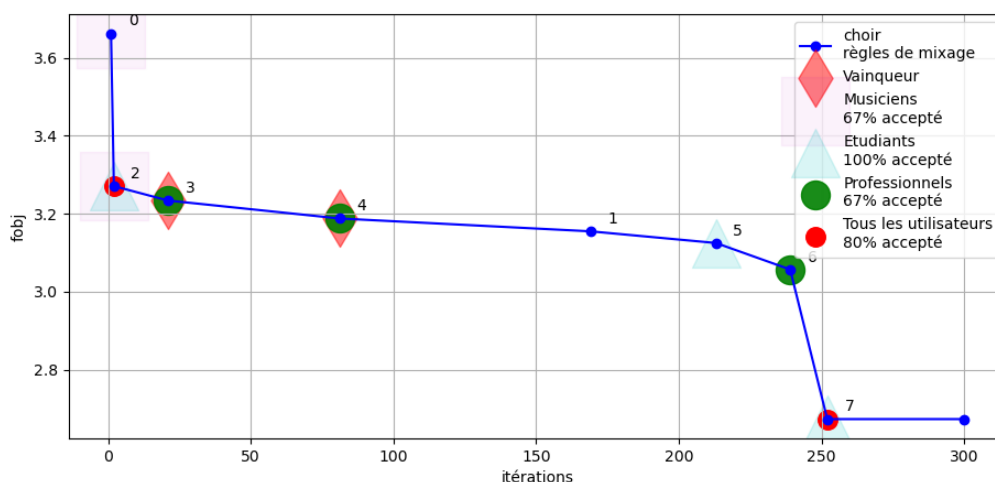


Figure 6.2 – Exemple : graphe de convergence du modèle psychoacoustique (Choir).

Nous avons décidé de compléter le graphique de convergence avec des informations supplémentaires obtenues grâce aux tests perceptifs.

Les chiffres à côté de chaque point bleu (la solution correspondant au pré-mixage) indiquent le numéro du pré-mixage, donné au hasard afin de mélanger ces pré-mixages pour les tests aléatoires en aveugle. Ce nombre peut être comparé au nombre figurant dans la version texte des règles de mixage (cf. Section E.6).

A chaque morceau de musique correspond un résumé global des statistiques de préférence qui inclut toutes les catégories de répondants. Ces statistiques sont affichées sous la forme d'un histogramme en Annexe E.55. Nous avons décidé de marquer visuellement la ou les solutions (pré-mixages) préférées sur le graphique de convergence sous forme de points rouges de petit diamètre.

L'Annexe E rassemble également les statistiques globales s'agissant de l'intention des personnes ayant répondu au test, nommés sujets dans la suite, d'utiliser le meilleur pré-mélange qu'ils ont choisi comme point de départ pour un travail ultérieur avec celui-ci. C'est ce que montre le graphique du camembert dans l'Annexe E.56. Nous avons également affiché cette information dans le quadrant supérieur droit, sur la dernière ligne, du graphique de convergence.

Nous ne nous sommes pas limités aux statistiques générales, mais sommes allés plus loin et avons décidé d'étudier les statistiques pour chaque catégorie de sujets. Ainsi, les cercles verts correspondent aux solutions choisies par les ingénieurs du son professionnels ayant une grande expérience, et, le pourcentage de leur intention d'utiliser des pré-mixages à l'avenir est également affiché sous forme de texte dans le coin supérieur droit du graphique.

Le test a été aussi effectué par des étudiants en dernière année de formations d'ingénieurs du son et leurs réponses sont indiquées par des triangles bleu clair transparents. Le pourcentage d'acceptation du pré-mixage pour une suite des travaux est également indiqué dans le coin supérieur droit en face du triangle bleu clair.

Pour la dernière catégorie (musiciens), nous avons combiné deux types de sujets : les musiciens professionnels et les amateurs audiophiles. Leurs préférences correspondent à des carrés transparents violets légèrement plus grands, et leur accord pour utiliser le pré-mixage à l'avenir est également reflété dans le coin supérieur opposé au carré violet. Il convient de noter d'emblée que de nombreux musiciens et amateurs n'ont pas été personnellement confrontés au processus d'ingénierie du son et du mixage en particulier, et jugent donc l'acceptation du mixage davantage en fonction de leur préférence pour un style ou un autre qu'en fonction du travail professionnel effectué sur le mixage. Par conséquent, leurs statistiques sur leur taux d'acceptation sont à étudier avec un peu de circonspection.

Il convient également de noter que nous avons utilisé des tailles différentes pour les figures sélectionnés (cercle, triangle, carré) pour les trois catégories (professionnels, étudiants et musiciens). Une taille plus grande et une plus grande transparence de la figure signifient moins de confiance et une décision floue. Nous avons pensé que cette approche sémantique permettrait de créer intuitivement les bonnes associations chez notre lecteur.

Le dernier élément important du graphique est une figure de diamant de couleur rouge, légèrement transparente. Il indique les pré-mixages qui ont été sélectionnés par les professionnels avec l'intention de continuer à travailler sur ces pré-mixages dans le futur, c'est-à-dire une acceptation de ces pré-mixages à 100% avec le désir de les utiliser comme point de départ pour le travail.

Parfois, il peut y avoir plusieurs diamants rouges si des ingénieurs du son différents ont choisi des pré-mixages différents comme pouvant être choisis pour continuer un travail de mixage. Parfois, le graphique ne comporte pas de diamant rouge si les professionnels n'ont pas eu envie d'utiliser ces pré-mixages pour une raison ou une autre.

Dans le contexte de notre travail, la présence du diamant rouge sur un graphique est un synonyme de succès, c'est-à-dire que l'algorithme a créé un mixage qui, premièrement, plaît et, deuxièmement, suscite l'envie de l'utiliser pour finaliser un mixage.

6.4.2 Le graphe récapitulatif des paramètres du modèle psychoacoustique

Ce graphique de synthèse (cf. Figure 6.3) permet d'évaluer la performance globale de l'algorithme d'optimisation et la dynamique des paramètres de mixage, sans entrer dans le détail au niveau des instruments impliqués dans le mixage. Le graphique permet d'évaluer la variabilité des changements de paramètres lors du passage d'un mixage à un autre, ainsi que, dans certains cas, de retracer l'évolution des paramètres lorsque la valeur de la fonction objectif diminue.

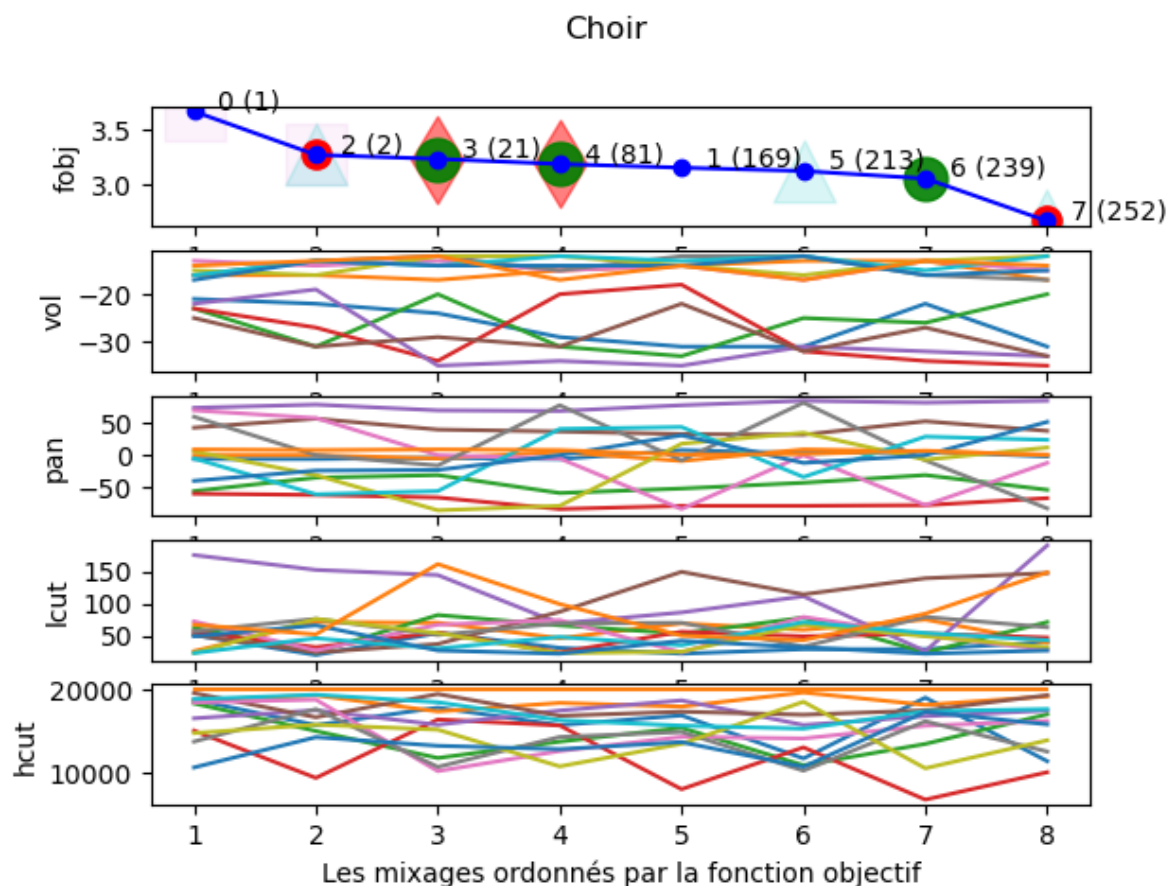


Figure 6.3 – Exemple : graphe récapitulatif des paramètres du modèle (Choir).

Ce graphique de synthèse est composé d'un ensemble de cinq sous-graphes. Le sous-graphe supérieur répète brièvement le graphe de convergence de la fonction objectif *fobj* (cf. Figure 6.2), dans sa version complétée. Cela a été fait pour faciliter l'analyse des autres graphiques, car le graphique supérieur reflète les choix des sujets par catégorie, ainsi que les pré-mixages choisis.

Les 4 sous-graphes inférieurs correspondent aux 4 paramètres à optimiser : volume (*vol*), panorama (*pan*), filtres passe-haut (*lcut*) et passe-bas (*hcut*). Sur l'axe des abscisses figurent les numéros de mixage, en fonction du nombre de mixages obtenus pour chaque morceau de musique. Tous les mixages sont classés en fonction de la valeur de leur fonction objectif, ce qui permet de retracer l'impact de la progression de l'optimisation sur la fluctuation des paramètres optimisés.

Pour conserver l'information sur les itérations, à savoir l'effort appliqué par l'algorithme d'optimisation pour trouver une nouvelle meilleure solution, nous avons étendu le graphique supérieur (*fobj*) et ajouté entre parenthèses le numéro d'itération de l'algorithme où ce mixage a été obtenu à la valeur numérique du numéro de mixage correspondant au nu-

méro de mixage randomisé utilisé dans les tests perceptifs et dans l'enregistrement textuel des règles de mixages.

Comme les 4 graphiques du bas comportent autant de lignes que le nombre d'instruments, ces graphiques sont un peu difficiles à étudier de manière précise, aussi, nous avons choisi de ne pas ajouter les informations sur les instruments sur ce graphique mais de créer plutôt des graphiques séparés pour chacun des instruments.

6.4.3 Le graphe individuel des paramètres pour chaque instrument

Ce graphique (Figure 6.4) reprend les mêmes 5 sous-graphiques (cf. Figure 6.3) mais en n'indiquant que les changements correspondant à un seul des instruments du mixage.

En plus du nom du morceau de musique (cf. Tableau 6.1), le nom de l'instrument a été ajouté au titre de la figure. De plus, sur les sous-graphes présentant les paramètres (règles de mixages), des valeurs numériques des paramètres ont été ajoutées à chaque point (mixage), uniquement pour la commodité de l'analyse de ces graphiques.

Il convient de noter que pour créer ces pré-mixages, nous avons très souvent, et lorsque cela n'est pas contraire au bon sens, essayé d'assouplir les restrictions imposées au positionnement des instruments (cf. Annexe C). Nous nous sommes demandé quelle était la stabilité de notre modèle psychoacoustique et si l'algorithme d'optimisation était capable de trouver une solution esthétiquement acceptable dans des conditions plus lourdes (plus grand espace de valeurs à explorer).

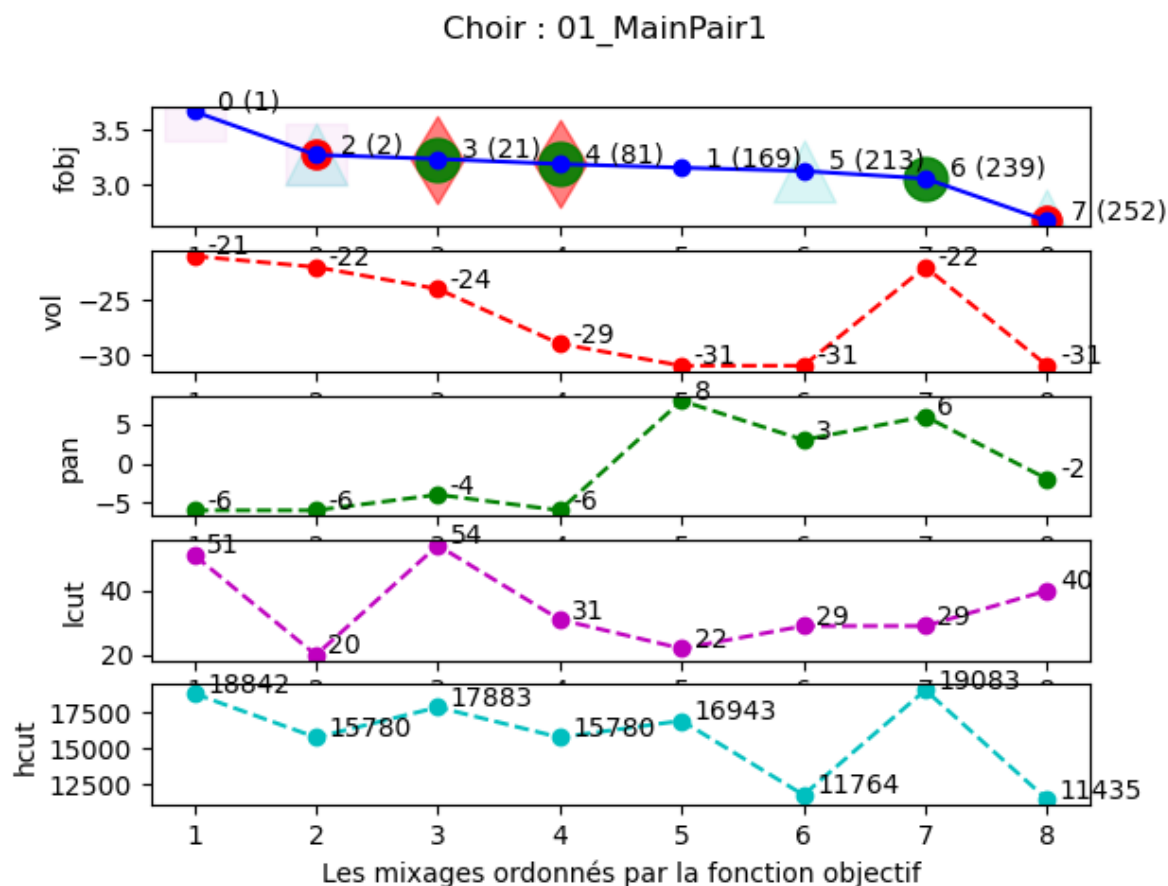


Figure 6.4 – Exemple : graphe des paramètres pour un instrument (Choir).

6.5 L'observation et la discussion des résultats

Si on étudie les graphiques de convergence, la première chose que l'on constate est le fait que des mixages choisis sont complètement différents, ce qui signifie que des personnes différentes préfèrent des pré-mixages différents et qu'il y a très rarement un consensus pour choisir un seul pré-mixage.

Il est également intéressant de noter la différence entre les choix effectués par les professionnels, les étudiants et les musiciens. Leurs choix sont également différents les uns des autres. Il convient de noter que les choix effectués par les professionnels sont généralement plus stables (par exemple, pour les valeurs de la fonction objectif), tandis que les choix effectués par les étudiants sont plus diversifiés et souvent assez différents de ceux des professionnels, même si les concordances sont nombreuses. Et les choix des musiciens, y compris ceux des audiophiles amateurs, peuvent être très éloignés de ceux des professionnels. C'est peut-être la raison pour laquelle il y a parfois des cas dans la pratique où l'ingénieur du son est fier de son travail mais le musicien, le client, n'est pas content et

suggère des changements, souvent contraires aux convictions ou à l'expérience du professionnel.

De plus, il convient de répéter qu'il est insensé de se fier aux intentions des musiciens et des amateurs d'utiliser ou pas les pré-mixages choisis à l'avenir, car leur jugement n'est pas fondé sur des connaissances techniques en matière d'ingénierie du son, mais sur des préférences personnelles. En d'autres termes, le taux d'acceptation global du mixage pour toutes les catégories ensemble sera supérieur aux chiffres indiqués.

Un pré-mixage réussi, donc indiqué par des professionnels comme pouvant être utilisé comme base pour finaliser un mixage, n'a pas toujours la préférence des musiciens. Est-ce que cela signifie que, le plus souvent, dans la pratique, les ingénieurs du son imposent leurs goûts esthétiques à leurs clients ? Où se situe la frontière entre le respect des souhaits du client et l'expression de ses propres opinions et préférences ? L'image du professionnel et la relation de confiance entre le professionnel et le client jouent peut-être un rôle très important.

Il y a eu quelques cas (Folk cf. Figure E.9, Ballade cf. Figure E.88) où aucun des mixages choisis n'était envisagé comme base pour finaliser un mixage. On peut supposer qu'il y a au moins deux raisons :

- soit il y a eu une erreur dans la définition des contraintes et/ou du style au stade préparatoire ; par exemple les contraintes n'ont pas permis à l'algorithme d'optimisation d'explorer des solutions potentiellement intéressantes pour l'œuvre musical, ou la coloration stylistique ne correspondait pas bien au genre musical ;
- soit le modèle psychoacoustique n'est pas assez précis, parce que, par exemple, un démasquage réalisé par égalisation a pu être indiqué comme souhaitable au stade du pré-mixage, ou par qu'il a été proposé qu'une compression grossière pour les instruments sur-dynamiques devrait être appliquée.

Il ne semble pas y avoir de relation claire entre les choix effectués (des pré-mixages) et la valeur de la fonction objectif. Il faudrait déterminer si cela constitue un indice selon lequel il existe une grande variété de goûts artistiques, ou, au contraire, le fait que le modèle psychoacoustique doit être encore affiné pour pouvoir prendre en compte autant que possible d'éléments communs de la perception humaine de la musique.

Le plus souvent, les gens choisissent des pré-mixages qui ne correspondent ni aux pré-mixages déterminés au début du cycle d'optimisation, ni aux pré-mixages calculés à la fin du cycle d'optimisation, lorsque le modèle a convergé vers une valeur minimale mais les pré-mixages choisis correspondent aux solutions trouvées au milieu du processus d'optimisation. Nous pourrions soulever ici la question philosophique de savoir ce qu'est la beauté ou ce qui est considéré comme beau. Et, cette question ne s'applique pas nécessairement qu'à la musique puisqu'elle peut concerner n'importe quel art ou phénomène. Peut-être qu'un petit déséquilibre, avec un désir prononcé d'équilibre sous forme d'indices et de repères, nous semble plus parfait que la beauté parfaite dans sa plus belle forme ? Cet effet

peut peut-être s'expliquer par le dispositif de notre pensée : en traçant les grandes lignes et en ne dessinant pas les détails, nous donnons au cerveau la liberté de créer sa propre image intérieure, individuellement parfaite, d'un chef-d'œuvre, et les indications iconiques lui facilitent le travail dans une telle construction. Et la beauté intérieure qui en résulte, subjective pour chacun, construite sur les lignes de référence, légèrement déséquilibrées, de la beauté extérieure, sera la solution esthétique la plus souhaitée. En termes techniques, nous devons probablement introduire des éléments de léger déséquilibre et d'imperfection dans le modèle psychoacoustique et il ne sert vraisemblablement à rien de chercher une solution mathématique globale universellement parfaite, il suffit juste de s'en approcher un peu : il faut peut-être juste faire la moitié du chemin parce que, s'agissant du reste du chemin, notre cerveau le fera tout seul.

Avec des contraintes initiales correctement formulées au stade préliminaire, il est possible de trouver un mixage réussi dans les premières itérations de l'optimisation (Jazz cf. Figure E.73, Latino cf. Figure E.28). Ce fait nous indique l'importance du pré-traitement. Dans ce travail de mémoire, nous n'avons pas prêté beaucoup d'attention au pré-traitement, mais, à l'avenir, grâce à l'intelligence artificielle et aux méthodes d'apprentissage profond il sera possible de construire déjà dans la phase de préparatoire des contraintes mathématiques très précises, pour le modèle d'optimisation, reflétant clairement le caractère musical et stylistique du projet considéré. Cette étape devrait augmenter considérablement les chances de créer un pré-mixage réussi et réduire le temps de calcul pour le produire, en réduisant le nombre d'itérations à réaliser pendant le processus d'optimisation. Avec des contraintes mal formulés, ou des contraintes trop lâches, même après de nombreuses itérations, le mixage peut ne pas converger vers une version esthétiquement attendue et/ou acceptable.

Je voudrais maintenant parler un peu de la difficulté à conduire les tests perceptifs, ou plutôt du fait que moins de personnes que souhaité y ont répondu, malgré le fait que nous ayons mené une très grande campagne de publicité.

Nous avons consacré beaucoup de temps à des mailings directs avec des demandes personnalisées pour inviter des personnes à consacrer un peu de temps pour des tests. Nous avons écrit des annonces d'invitations sur de nombreux forums et groupes d'ingénieurs du son dans différents pays (France, Angleterre, Russie, États-Unis). Avec l'aide des enseignants, des annonces ont été faites pour les étudiants de dernière année de départements d'ingénierie du son. Nous avons directement contacté le président (Jim Andreson) et le directeur (Brecht De Man) de l'Audio Engineering Society, en leur demandant de diffuser l'annonce sur des tests perceptifs auprès de leurs membres, y compris les étudiants. Nous avons personnellement rencontré la présidente de la Guilde des ingénieurs du son de la Russie (Maria Soboleva) et lui avons demandé de distribuer l'annonce aux membres de cette organisation, ainsi qu'aux étudiants de tous les établissements d'enseignement de Moscou où il existe un département d'ingénierie du son. Nous avons contacté Joshua Reiss, professeur à l'université Queen Mary de Londres, qui a consacré les dix dernières années de sa carrière aux systèmes de mixage automatique, pour lui demander de diffuser également notre annonce à son entourage et à ses étudiants. En d'autres termes, nous avons fait

de notre mieux pour maximiser le nombre de participants potentiels à ces tests.

Malgré de nombreux efforts, il y a eu au final peu de réponses. Nous avons essayé de découvrir pourquoi et avons posé même la question directement à certaines personnes.

- ◇ Les ingénieurs du son professionnels sont souvent très occupés et les tests prennent beaucoup de temps, généralement plus d'une heure.
- ◇ Il était difficile pour les étudiants d'avoir de bonnes conditions d'écoute, pour certains cela prenait beaucoup de temps et ils ont donc abandonné préférant d'autres activités, certains n'étaient pas sûrs de leurs propres réponses et ont fini par renoncer à finir le test.
- ◇ Les musiciens professionnels étaient souvent très occupés, beaucoup ne comprenaient pas l'essence du questionnaire car ils n'avaient jamais fait de mixage musical et ne savaient pas comment et selon quels critères évaluer ces mixages. En plus, les musiciens ne disposaient souvent pas d'un équipement d'écoute de bonne qualité.
- ◇ Les amateurs et les audiophiles, dans la plupart des cas, n'avaient pas non plus d'expérience en matière de mixage, aussi il était très difficile pour eux d'utiliser des critères propres aux ingénieurs du son pour émettre un jugement, et, de plus, ils ne comprenaient pas forcément suffisamment bien ces critères.

Nous avons essayé d'éliminer au maximum les difficultés afin de maximiser le nombre de participants. Plus précisément, nous avons entièrement revu l'introduction de l'entête des tests et la description des tests à faire est devenue plus détaillée. Initialement, les tests ont été conçus en anglais, une partie du texte a été traduit en russe et en français, avec une indication d'une éventuelle traduction automatique (Google Translate) pour supprimer la barrière linguistique. Nous avons également ajouté une référence aux critères internationaux d'évaluation du matériel musical selon les normes approuvées *EBU Tech 3286*, avec un lien vers le document¹ lui-même. Ce document contient des instructions détaillées sur la façon dont les tests doivent être effectués, avec une indication détaillée des critères de qualité des mixages et de l'échelle d'évaluation de la qualité. Nous n'avons pas imposé ce système d'évaluation à tous les participants du test, afin de ne pas perdre les quelques volontaires, chacun était libre de donner son appréciation et ses commentaires de manière totalement libre.

Pour détendre l'atmosphère, je conclurai cette section par une anecdote qui m'est arrivée. Afin d'étudier le fonctionnement du modèle, je ne me suis pas limité aux petits projets de 10 pistes, mais j'ai essayé de faire converger automatiquement un vrai projet *pop* de plus de 50 pistes. Bien qu'il s'agisse d'un prototype, le code est néanmoins écrit de manière suffisamment efficace pour utiliser tous les processeurs de l'ordinateur à 100%. Les tests ont eu lieu sur mon vieil ordinateur portable, et les calculs ont duré toute la soirée, toute la nuit et toute la matinée jusqu'à ce que je sente l'odeur de brûlé dans la pièce. Heureusement que je n'étais pas loin à ce moment-là, car je me suis vite rendu compte que mon ordinateur préféré était en feu. Je l'ai immédiatement arrêté et, comme c'était l'hiver

1. <https://tech.ebu.ch/docs/tech/tech3286.pdf>

et que le balcon était gelé, je l'ai tout de suite sorti et posé sur une dalle de béton froide pour qu'il se refroidisse. Heureusement, le processeur a survécu et j'ai réussi à sauver mon ami fidèle et fiable. Depuis, je n'ai plus expérimenté de grands projets à la maison, mon ordinateur m'étant trop précieux !

6.6 Quelques citations

Dans cette section, nous citerons certains des commentaires laissés lors des tests perceptifs que nous avons choisis car ils nous semblent essentiels. Tous les commentaires et réponses seront disponibles dans leur intégralité à partir d'un site internet [VITAL, 2022].

Nous avons décidé de fournir des citations sous forme de tableau pour chaque chanson séparément. Le tableau comporte deux colonnes : 1) les réactions ou commentaires positifs et 2) les réactions ou commentaires négatifs. Le tableau comporte également trois lignes, chaque ligne ayant un code couleur différent : 1) le vert clair est destiné aux professionnels, 2) le jaune clair est aux étudiants et 3) le bleu clair est aux musiciens et aux amateurs. Nous avons trouvé que cette présentation était la plus efficace. Au début de chaque citation, nous avons mis entre parenthèses le nombre (chiffre) des règles de mixage que le sujet cité préférait. La plupart des réponses étaient en français ou en anglais ; nous avons pris la liberté de ne pas traduire de l'anglais au français dans ces cas.

6.6.1 Classic : Rod Alexander – Canon

positif	négatif
<ul style="list-style-type: none"> ✓ (4) l'image est assez stable ✓ (3) Le son est le plus naturel et c'est appréciable pour un instrument acoustique, surtout en solo 	<ul style="list-style-type: none"> ✓ (4) Il faudrait retravailler les haut-mediums, mais la base est là
<ul style="list-style-type: none"> ✓ (3) on ne perd pas la sensation d'intimité du morceau 	
<ul style="list-style-type: none"> ✓ (3) les 2 guitares se distinguent bien, l'équilibre stéréo est agréable 	
<ul style="list-style-type: none"> ✓ (4) well balanced, no distortion, no muddiness, clear and transparent 	
<ul style="list-style-type: none"> ✓ (1) The track has nice balance between instruments 	

Table 6.2 – Citations sur Classic : Rod Alexander - Canon.

6.6.2 Folk : Leslie Mendelson – The Hardest Part 2

positif	négatif
<p>✓ (3) Clarté des guitares et présence des voix</p> <p>✓ (2) I think that this would be a good premix</p> <p>✓ (3) J'ai bien aimé les rapports de niveau</p> <p>✓ (1) Le 1 est mon préféré car les deux voix sont équilibrées, les guitares sont intelligibles et pas trop en-dessous. C'est un fonctionnement du mix assez symétrique</p> <p>✓ (2) deep dynamic range and no low mids overloaded</p> <p>✓ (1) good balance between low and high frequencies</p>	<p>✓ (2) weird vocal panning</p> <p>✓ (2) guitare à filtrer dans les graves</p> <p>✓ (0) je compresserais les guitares</p> <p>✓ (1) Je pense qu'on peut faire mieux pour la clarté des guitares notamment</p>

Table 6.3 – Citations sur Folk : Leslie Mendelson – The Hardest Part 2.

6.6.3 Ska : Big Stone Culture – Fragile Thoughts

positif	négatif
<p>✓ (3) Le jeu basse/grosse caisse fonctionne, la basse imposante n'écrase pas tout. La batterie guitare et voix trouvé alors leur place</p> <p>✓ (4) La 4 a le plus d'énergie</p> <p>✓ (5) La voix est très devant par rapport aux autres instruments (un peu trop) mais c'est un choix qui me plait personnellement, la partie basse du spectre est moins chargée et brouillonne ce qui rend pré-mix mieux défini, plus agressif. La dynamique de la guitare du début me paraît aussi plus tenue.</p> <p>✓ (5) good balance between voice, instruments and drums, better than others, also much more clear</p> <p>✓ (5) This is the best version it has some depth and balance in it</p>	<p>✓ (0) the balance is all wrong</p> <p>✓ (4) la réverb est à revoir, souvent trop forte et trop décalée avec le son direct (artificielle)</p> <p>✓ (1) Je trouve que tout semble très éloigné et peu précis, même si tout est audible</p> <p>✓ (3) le fait de faire l'équilibre des niveau et pan est ce qui va provoquer en moi les autres considérations de mix (eq/comp/reverb etc...)</p> <p>✓ (5) I would like to mix it from scratch</p> <p>✓ (5) I had to reduce lower frequencies artificially, because they are killing the sound of guitar and take some depth from the vocals</p>

Table 6.4 – Citations sur Ska : Big Stone Culture – Fragile Thoughts.

6.6.4 Latino : Eddie Garrido – Una Semana Sin Ti

positif	négatif
<p>✓ (3) touching)</p> <p>✓ (5) meilleur équilibre</p> <p>✓ (1) Le mélange des sons percussifs est OK. l'équilibre des niveaux, textures, reverb et pan des percussions OK. La (contre ?) basse sonne naturel et l'accompagnement harmonique est équilibré. La voix trouvé alors sa place. Sur d'autres mix le rendu rappelle même des bandes sons avec des sons midi basse qualité.</p> <p>✓ (1) J'aime bien le placement du piano à gauche, la contre-basse n'est pas trop proche et suffisamment tenue. Je trouve la voix bien rentrée dans le mix, pas trop proche, avec une dynamique moins étendue que dans la version 2 par exemple. J'aime aussi le traitement de la cymbale qui n'a pas trop d'attaque (ce qui est agaçant par exemple dans la version 3) mais qui a une jolie résonance dans les extrêmes aigus.</p> <p>✓ (1) J'aime bien la 1 car elle met une bonne quantité de réverb sur la percussion dans l'oreille gauche, pour un peu casser son côté omniprésent et qui peut être focalisant et agaçant. J'aime bien aussi la place donnée à la seconde mélodie de clavier plus grave. Et la voix pas trop au premier plan.</p> <p>✓ (1) It's already a pretty good premix.</p> <p>✓ (6) J'ai préféré les niveaux ! Ce prémix donne une base de réflexion. Par exemple : avoir mis les cordes et le piano du début au même endroit, je pense que je ne ferais pas ça, mais l'équilibre marche déjà bien.</p> <p>✓ (4) sounds natural and well balanced</p> <p>✓ (3) vocal plus propre</p> <p>✓ (1) Guitar in this one sounded vibrant and more distinguished than in other mixes</p>	<p>✓ (5) Possible de partir de ce prémix, mais il faut faire du suivi de voix, séparer un peu le piano du vibraphone, tout en refermant un peu l'espace...</p> <p>✓ (1) je trouve les violons trop présents, trop forts, je préfère qu'ils apportent du relief comme dans la version 4 par exemple</p>

Table 6.5 – Citations sur Latino : Eddie Garrido – Una Semana Sin Ti.

6.6.5 Country : Angela Thomas Wade – Milk Cow Blues

positif	négatif
<p>✓ (2) Le mix respire bien. Surtout le jeu entre le piano, les guitares, le dobro et les balai de la batterie est clair et agréable à écouter...</p> <p>✓ (4) Je trouve que cette version a le meilleur équilibre entre les niveaux des différents instruments, j'aime aussi la répartition latérale.</p> <p>✓ (3) The place of the violin in the stereo in 3 makes more sense, and doesn't cover up other instruments.</p> <p>✓ (2) Je trouve qu'il est celui qui permet de distinguer le plus d'instrument</p> <p>✓ (2) C'est surtout le niveau de la voix et la snare qui a motivé mon choix.</p> <p>✓ (3) This premix is already good.</p> <p>✓ (5) For my ears here was the best balance between the singer's voice and a backing track.</p>	<p>✓ (3) 6 et 7 : batterie trop en avant. La voix est ok. Peut-être la version 3 est la plus homogène, malgré tous les autres défauts (piano, slide-guitare...)</p> <p>✓ (4) Je trouve que cette version manque un peu de relief et des effets de masque persistent, il faudrait améliorer ces points dans la suite du mix.</p> <p>✓ (2) Je trouve que toutes les propositions ont le même défaut : elles placent les instruments comme un amas timide sous la voix</p>

Table 6.6 – Citations sur Country : Angela Thomas Wade – Milk Cow Blues.

6.6.6 Choir : Don Camillo Choir – Marsh Marigolds Song

positif	négatif
<p>✓ (3) il y a un certain équilibre entre les voix d'hommes et de femmes, ainsi qu'une homogénéité dans cette version</p> <p>✓ (4) L'harmonie entre les différentes partie du chœur fonctionne bien... J'ai la sensation d'entendre un tout. L'équilibre entre les différentes partie du cœur est OK. Dans les autres mix les basses ou les médiums ou le chant féminin sólo sont soit trop présent soit trop lointain. Le 4 trouve un bel équilibre</p> <p>✓ (7) J'aime bien la répartition latérale des voix qui donne une sensation enveloppante.</p> <p>✓ (7) This is the one where I can discriminate the most every voice and that is the most overall satisfying</p> <p>✓ (5) J'aime beaucoup la proposition de profondeur qui est faite, avec la voix la plus aiguë très loin en niveau et dans la reverb. Je trouve que la rondeur de la voix des hommes apporte quelque chose d'important et est enveloppant. La voix des femmes soutient dans cette version. J'aime aussi que les voix du contrepoint soient suffisamment présentes.</p> <p>✓ (0) For the low frequencies were more prominent in this one. It had more power and depth</p>	<p>✓ (6) doesn't sound like a choir</p> <p>✓ (3) manque de suivi sur la voix de tête</p> <p>✓ (5) Je trouve qu'il y a des choses à améliorer, notamment sur les voix des hommes mais que c'est un très bon point de départ technique et artistique.</p> <p>✓ (7) This is a good premix, even if a little bit unbalanced</p>

Table 6.7 – Citations sur Choir : Don Camillo Choir – Marsh Marigolds Song.

6.6.7 Jazz : Araujo – The Saga Of Harrison Crabfeathers

positif	négatif
<p>✓ (0) Le meilleur équilibre bass batterie piano.</p>	<p>✓ (1) Parce qu'il y a beaucoup de pistes cela peut être intéressant de reprendre la version 1, même s'il faut resserrer le piano et le déplacer à droite pour mieux occuper l'espace. La guitare un peu surmixée peut être corrigée.</p>
<p>✓ (2) Elle est pas mal. ✓ (2) Dans la version 2, les différents instruments sont clairs, et distinguables, la contrebasse n'est pas trop présente mais est dans même ronde. J'aime bien la reverb sur la guitare et qu'elle ne soit pas trop prépondérante. ✓ (7) Je trouve qu'il y a un bon équilibre entre les différents éléments de la batterie et que la cymbale a une dynamique plutôt bien maîtrisée (même si c'est encore à améliorer). J'aime aussi l'équilibre entre la basse et la guitare de droite. ✓ (5) 5 was where the guitar fitted the most in the mix.</p>	<p>✓ (7) je trouve la guitare de gauche trop faible par rapport à l'autre guitare et à la basse, la basse est peut-être encore un peu forte et surtout les guitare et la basse se masquent entre elle, les bas médium sont trop chargés ce qui donne un côté boomie au pré-mix. ✓ (5) When it's paned (guitar) on the same side than the piano, it covers the piano. I would still want to put the snare a little bit to the left to make some space to put the guitar more in the center of the stereo.</p>
<p>✓ (6) In my opinion here was reached the perfect balance between low/high. Bass and rythm section in general sounded great without dominating the mix.</p>	<p>✓ (0) no one sounds balanced</p>

Table 6.8 – Citations sur Jazz : Araujo – The Saga Of Harrison Crabfeathers.

6.6.8 Ballad : MR 1001 – Jay Menon.

positif	négatif
	<p>✓ (3) Le moins pire</p> <p>✓ (3) 5, 6 et 7 déséquilibre total entre voix et guitare ; 2 et 4 réverb horrible qui noie tout ; 3 tout est éclaté de gauche à droite, mais le voix toujours dans les choux ; 1 et 2 totalement inhomogènes entre réverb et instruments collés au premier plan. A devoir choisir la 3 donne une ambiance de concert live à peu près homogène, mais pas pour un disque studio !</p>
<p>✓ (7) Je trouve que cette version a le meilleure équilibre entre les parties hautes et basses du spectre</p>	<p>✓ (7) La partie bas-médium est un peu trop chargée, il y a des problèmes de masquage dans cette bande de fréquences, mais moins que dans la version 2 par exemple. Il y a néanmoins des problèmes de dynamique et un côté très métallique dans le timbre de certains instruments mais je trouve que c'est la version la plus maîtrisée et la mieux équilibrée.</p> <p>✓ (0) I think that the guitars are a bit too high in volume, and that the voice is far too low. I would correct it a little bit more before starting mixing.</p> <p>✓ (6) Les instruments sont tous à aller repêcher et ce n'est pas un bon point de départ. C'est peut-être seulement une question de matière initiale.</p> <p>✓ (2) J'ai trouvé les guitares trop fortes à chaque fois, et la voix pas assez. Et avec l'extrait 2 je pense que c'est celui où la voix est la plus forte.</p>
<p>✓ (7) Here was reched the better mix between all instruments and the vocals.</p>	<p>✓ (7) I don't really like how vocals sound in any of this mixes, I like it to hear more pushed forward and build instruments around it.</p>

Table 6.9 – Citations sur Ballad : MR 1001 – Jay Menon

6.6.9 Quelques notes complémentaires

russe: «почти все примеры были очень неудачно сбалансированы, причём проблемы баланса могли решиться с помощью эквализации а не громкости, это как раз тот момент когда алгоритм который ограничен в действиях (например только фильтры высоких и низких, громкость, панорама) не может принять правильного решения, потому что только этих четырёх параметров очень часто просто не хватает чтобы сделать качественную отправную точку»

français: “presque tous les exemples étaient très mal équilibrés, et les problèmes d'équilibre pourraient être résolus avec l'égalisation et non avec la spatialisation, c'est exactement le point où un algorithme qui a des actions limitées (par exemple seulement les filtres haut et bas, le volume, le panorama) ne peut pas prendre la bonne décision, parce que seulement ces quatre paramètres très souvent ne suffisent pas.”

“Expérience très intéressante. Malheureusement, on aimerait changer des paramètres qui ne bougent pas d'une version à l'autre et c'est très frustrant de ne pas pouvoir régler d'abord certains problèmes (réverbères pas très belles, positionnement de sources, suivi de niveau de voix ou guitares, basse à recalculer, etc.)”

6.7 Quelles conclusions pouvons-nous tirer ?

Résumons brièvement certaines conclusions importantes que nous avons pu tirer de l'analyse des réponses aux tests perceptifs, ainsi que nos propres suggestions pour améliorer le modèle de mixage actuel.

- ◇ La précision de l'optimisation n'est pas importante, le fait d'avoir de nombreuses opinions subjectives et divergentes prédispose à de multiples solutions locales plutôt qu'à une seule solution globale.
- ◇ Il faut ajouter le démasquage par égalisation, car tous les cas ne peuvent être résolus par la disposition spatiale des instruments.
- ◇ Il faut ajouter au moins une compression grossière sur les instruments ayant une grande plage dynamique et selon le style de musique.
- ◇ Un critère évaluant la nécessité d'un traitement dynamique d'un instrument doit être considéré.
- ◇ Adapter les critères au nombre de pistes dans le mixage : plus le nombre de pistes augmente, plus la densité des instruments dans le mixage augmente et par conséquent le mixage doit répondre à des exigences supplémentaires.
- ◇ Un critère et un algorithme de correction de fréquence des instruments devrait être ajouté, au moins pour lisser les résonances parasites.
- ◇ Il serait utile d'ajouter une formule pour recalculer les valeurs de l'égaliseur et du compresseur (via un modèle mécaniste) en fonction de la distance d'un instrument

par rapport à l'auditeur : avec l'augmentation de la distance on obtient des hautes fréquences plus faibles et une compression plus forte.

- ◇ Il est hautement souhaitable d'avoir une version interactive avec l'utilisateur.

Chapitre 7

Comment améliorer le système de mixage automatique à l'avenir ?

Dans le prototype présenté dans ce mémoire, nous avons considéré un ensemble minimal d'outils de base pour créer un mixage spatial : volume, panorama, filtres de basses et d'aigus. L'objectif principal consistait à déterminer la pertinence, l'utilité et la viabilité d'un tel système, ainsi que son potentiel pour les ingénieurs du son et les musiciens.

Nous avons réalisé deux questionnaires pour mieux comprendre les méthodes de travail des ingénieurs du son, leurs besoins, ainsi que pour obtenir un retour sur notre prototype VitaMIX. Sur la base des informations recueillies, nous suggérerons dans ce chapitre comment améliorer le prototype actuel d'un point de vue technique, quelles sont les fonctionnalités qu'il serait pertinent d'ajouter, à quoi ressemblerait le lieu de travail de l'ingénieur du son avec un tel produit potentiellement créé (sur la base de ce prototype), et comment il serait possible de diversifier cette technologie de mixage automatique dans des domaines connexes de l'ingénierie du son.

7.1 Quelles sont les limites en termes de vitesse de calcul et de mémoire ?

Le travail d'un ingénieur du son est un travail en temps réel, où il faut écouter la musique et changer très souvent les réglages des outils qu'il utilise. Le plus souvent, lorsque l'on doit travailler sur un flux conséquents de projets, les sessions sont programmées à la minute, et tout temps d'arrêt est indésirable.

Sur la base de ces hypothèses, il est peu probable qu'un ingénieur du son soit prêt à attendre longtemps (10 minutes ou plus) pour obtenir un pré-mixage de qualité inconnue, s'il peut obtenir des résultats de meilleure qualité par lui-même dans cette intervalle de temps. Par conséquent, l'algorithme de pré-mixage du produit réel devrait être capable de proposer plusieurs pré-mixages en un minimum de temps (environ une ou plusieurs minutes, pendant que l'ingénieur du son se détend, écoute de la musique, prend une tasse de thé ou de café, prend connaissances de références, etc.).

Comme nous le savons par expérience, les performances et l'efficacité d'un algorithme peuvent être directement ou indirectement liées à la quantité de mémoire (RAM) nécessaire pour exécuter cet algorithme. En d'autres termes, il est très souvent possible d'obtenir de meilleures performances au prix d'une utilisation accrue de la mémoire. La recherche d'un équilibre optimal entre les exigences du processeur et de la mémoire fera l'objet d'une optimisation du code. Pour l'instant, nous pouvons supposer que la configuration de la station de travail minimale doit embarquer un processeurs Intel i7 de 12ème génération de 10 à 14 cœurs à 4.60 – 5.00 GHz et que la capacité de mémoire doit varier de 32 à 64 Go.

7.2 Comment et par quelles méthodes peut-on améliorer la performance ?

7.2.1 Plus de mémoire, moins de calculs

Avec plus de mémoire, nous pouvons stocker les résultats intermédiaires et les réutiliser pour de futurs calculs. Par exemple, lorsque on calcule le critère de masquage pour toutes les combinaisons de pistes, on peut d'abord convertir toutes les pistes dans le domaine fréquentiel, stocker ces données en mémoire et les réutiliser pour chaque paire de pistes.

7.2.2 Le changement du langage et l'optimisation du code

Alors que nous avons proposé ci-dessus des méthodes d'optimisation au niveau algorithmique en réduisant la complexité des algorithmes, nous proposons ici une optimisation au niveau du codage et de l'écriture d'un code rapide.

Un aspect important est le choix du langage de programmation. Pour le prototype, nous avons utilisé le langage Python car il est suffisamment flexible et il existe de nombreux projets open-source qui accélèrent la vitesse de prototypage. Mais pour un véritable produit à part entière, ce langage n'est pas adapté, car il se révèle trop lent pour le travail envisagé. Un bon compromis pourrait être l'utilisation du langage *C++* car ce langage est très proche de Python en termes de flexibilité, notamment dans ses dernières déclinaisons, mais, grâce à la compilation native, il peut être utilisé pour écrire du code très performant et le compiler pour différentes plateformes (Windows/PC, iOS/Mac, etc.).

Un autre aspect positif de l'utilisation d'un langage de programmation de bas niveau, et par conséquent de ses compilateurs, est la possibilité de vectoriser les opérations afin d'exploiter les dernières avancées techniques des processeurs. Nous parlons ici de l'utilisation de registres larges¹ pour effectuer plusieurs opérations simultanément dans un cycle de calcul.

7.2.3 Les calculs parallèles et distribués

Un autre moyen efficace d'améliorer les performances consiste à écrire du code parallèle. Un processeur moderne contient de l'ordre de 8 à 16 cœurs (sous-processeurs) et, sur

1. AVX, AVX2, AVX512, Single Instruction on Multiple pieces of Data (SIMD)

chacun de ces cœurs, des opérations différentes peuvent être exécutées simultanément, ce qui peut augmenter considérablement la vitesse du programme. Un programme peut être parallélisé à différents niveaux : au niveau du calcul des critères d'optimisation et ces éléments ; au niveau du fonctionnement de l'algorithme d'optimisation ; au niveau du traitement parallèle des pistes lors du mixage ; etc.

Il existe plusieurs technologies de programmation parallèle : on peut simplement décrire manuellement les *threads* (calculs indépendants avec accès à la mémoire partagée) et les interactions entre eux, ou, au contraire, on peut utiliser des frameworks comme *OpenMP*² ou *TBB*³.

Si nous envisageons des méthodes d'optimisation supplémentaires au sein d'une seule station de travail, il convient de noter l'utilisation très répandue des *GPU*⁴ pour la parallélisation des calculs scientifiques au cours des dix dernières années. Par exemple, pour notre cas, nous pouvons supposer que la station de travail est équipée d'une ou plusieurs cartes graphiques, et que les transformations de Fourier sont effectuées sur les cartes graphiques. Selon l'un des développeurs de Microsoft⁵, il est possible d'obtenir une accélération jusqu'à 40 fois supérieure, par rapport à la version optimisée de *MKL*⁶ adapté pour les processeurs Intel (*CPU*).

Si la performance maximale est déjà atteinte au sein d'un nœud, mais que le programme ne répond pas encore aux exigences spécifiées (dans notre cas, par exemple, des pré-mixages optimaux disponibles en 1 – 3 minutes pour un projet de taille moyenne), l'étape suivante de l'accélération du code est le calcul distribué sur plusieurs nœuds (ordinateurs, stations de travail, serveurs). Dans cette approche, chaque nœud de calcul crée son propre *processus* de calcul qui dispose de sa propre mémoire locale. Ce processus peut contenir de nombreux *threads* parallèles et les nœuds communiquent entre eux à l'aide du protocole *MPI*⁷.

7.2.4 Le principe de séparation pour la réduction de la dimensionnalité

Une autre approche importante et efficace pour réduire la complexité algorithmique d'un problème d'optimisation consiste à décomposer un grand problème en plusieurs tâches plus petites qui peuvent être résolues en parallèle et en beaucoup moins de temps.

Puisque notre problème appartient à la classe des problèmes lourds et exponentiels⁸, le temps requis pour les calculs mathématiques croît à une vitesse énorme avec la taille

2. <https://www.openmp.org/>

3. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onetbb.html>

4. Graphical Processor Unit <https://www.nvidia.com/fr-fr/studio/compare-gpus/>

5. https://mc.stanford.edu/cgi-bin/images/7/75/SC08_FFT_on_GPUs.pdf

6. Math Kernel Library <https://www.intel.com/content/www/us/en/develop/documentation/oneapi-programming-guide/top/api-based-programming/intel-oneapi-math-kernel-library-onemkl.html>

7. A High Performance Message Passing Library <https://www.open-mpi.org/>

8. Le terme mathématique est *NP-hard* en anglais

du problème (nombre de pistes). Aussi, en divisant le problème en sous-tâches nous pourrions diminuer significativement le temps de calcul, surtout si beaucoup de ces sous-tâches peuvent être exécutées en parallèle, puisqu'elles sont indépendantes les unes des autres.

Dans le cadre de notre projet, nous pouvons prévoir deux types de division du problème en sous-tâches :

- **L'espace** : Il s'agirait de procéder au regroupement des pistes en groupes et à travailler sur l'obtention de l'équilibre fréquentiel de ces groupes. On procéderait ainsi à la division de toutes les pistes en groupes et à la résolution des problèmes d'optimisation au sein de chaque groupe en parallèle et indépendamment. Et, à l'étape suivante, les tiges (stems) mixées de ces groupes seraient traitées comme des pistes pour une optimisation finale et un réglage fin.
- **Le temps** : On opérerait pour la décomposition d'un morceau de musique par forme (refrain, couplet, pont, intro...) et la résolution de sous-tâches pour chaque tranche de temps individuelle, puis à l'intégration du résultat dans l'ensemble de l'œuvre.

7.3 Les options pour la mise en œuvre et l'utilisation du programme VitaMIX

Si l'on prend comme point de départ le projet *Ballad* (cf. Annexe C.8) comportant 16 pistes, le prototype VitaMIX a calculé une itération en environ 360 secondes, soit 6 minutes. Il n'a pas fallu plus de 80 itérations pour obtenir 7 solutions différentes. Si on arrondit à 100 itérations pour plus de simplicité, le cycle de calcul total ne prend pas plus de 600 minutes, soit 10 heures.

7.3.1 L'accélération du calcul

Comme nous ne disposons pas encore d'une image complète du produit final et de la liste de ses fonctionnalités, nous nous baserons sur les fonctionnalités actuelles pour estimer la diminution possible du temps de calcul.

- En utilisant la technologie *SIMD*, nous pouvons espérer une accélération d'au moins un facteur 4.
- En utilisant la technologie *OpenMP* avec, disons, 10-12 cœurs en moyenne par processeur, nous pouvons compter sur une accélération d'un facteur 10.
- Compte tenu du fait que les tests ont été effectués sur un vieil ordinateur portable avec une fréquence de 2.40 GHz pour leur processeur, nous pouvons sans risque diviser le temps de calcul par deux, car les processeurs modernes tournent systématiquement à 4.90 – 5.00 GHz.
- Si nous utilisons une carte graphique pour les calculs de transformée de Fourier, nous pouvons encore obtenir un gain de vitesse de facteur 40.
- Grâce à l'optimisation algorithmique, nous pouvons réduire la complexité du calcul des fonctions objectif de quadratique en $\mathcal{O}(n^2)$ à linéaire en $\mathcal{O}(n)$.

- Et par une gestion intelligente de la mémoire (Il n’y a pas de gestion manuelle de la mémoire en Python) nous pouvons gagner un facteur 10 au minimum sur les sections du code responsables de la copie de grands tableaux en mémoire (wav). Pour de nombreuses opérations, on peut imaginer des fonctions de traitement des données sur place (in-place) sans copie intermédiaire.
- On peut également re-écrire ou utiliser un moteur de mixage optimisé déjà disponible, par exemple Cubase, Pro Tools,... au lieu d’un projet Python de Pedalboard réalisé par Spotify. Nous estimons qu’il est possible sans risque d’accélérer le code d’un facteur 10.
- Si l’on souhaite traiter l’audio de façon plus performante, on peut encore utiliser des processeurs analogiques spécialisés comme les processeurs UAD-2 OCTO PCIe Analog Device SHARC. Cela permettra de libérer une grande partie de la capacité du processeur CPU pour d’autres calculs parallèles. Et l’utilisation de plug-ins UAD populaires pourrait nous apporter une flexibilité dans le contrôle des paramètres et la qualité du mixage final.

Au total, nous avons une accélération totale pour un projet typique, selon une estimation approximative, de pas moins de $4 \times 10 \times 40 \times 2 \times 10 \times 4 = 128\,000$ fois, même si l’utilisateur n’a pas de carte graphique, le gain de vitesse sera plus de 3200 fois. Appliqué au projet actuel de 16 pistes : 600 min / 3200 fois \simeq 12 sec.

Cela donne une marge de temps importante ($3 \times 60/12 = 15x$) pour construire des modèles psycho-acoustiques plus complexes, ajouter plus de critères d’optimisation, étendre les fonctionnalités, par exemple en ajoutant l’égalisation, la compression et d’autres traitements et effets.

7.3.2 Les serveurs et le calcul distribué

Si, pour une raison quelconque, la capacité d’une seule station de travail s’avère insuffisante, il est possible d’avoir plusieurs unités de calcul ou serveurs connectés à travers le réseau et, en utilisant la technologie *MPI*, d’obtenir une accélération d’environ un multiple du nombre de nœuds, sauf pour les temps nécessaires à la communication entre ces nœuds.

Comme solution alternative à *MPI*, une solution peut-être plus compacte consisterait à utiliser un serveur de calcul avec plusieurs, par exemple 4 processeurs à 16 cœurs chacun. Généralement sur de tels serveurs on peut installer jusqu’à 1 TB de RAM, ce qui devrait permettre la réalisation des calculs pour des projets de presque n’importe quelle complexité.

Il faut dire un mot ou deux sur le refroidissement et l’isolation sonore nécessaire pour un tel poste de travail. Et, comme on peut s’attendre à ce que tous les cœurs de processeur ainsi que toutes les cartes graphiques soient utilisés à presque 100%, grâce à un bon équilibre de charge et une optimisation du code, une telle station sera bruyante et chauffera de manière conséquente. Il serait donc plus confortable de déplacer la station dans un

couloir ou une pièce dédiée avec une bonne circulation d'air. En général, comme les serveurs modernes ne nécessitent pas d'unités de réfrigération, il suffit de fournir une nouvelle alimentation en air à température ambiante.

7.3.3 Les services en ligne

Ces dernières années, la tendance consiste à opter pour de nombreux outils bureautiques en ligne. L'exemple emblématique est celui de Microsoft Office. Ce n'est peut-être pas le cas pour les principaux acteurs des *DAW* (Cubase, ProTools), mais il faut noter que des homologues en ligne ont déjà fait leur apparition⁹. Une mention spéciale doit être faite au projet de démarrage (startup) *Faders*¹⁰, car ce projet tente d'utiliser le mixage automatique.

Bien que le passage au mixage en ligne pour les professionnels soit techniquement et stratégiquement discutable, l'approche en ligne présente un certain nombre d'avantages, notamment :

- l'adaptabilité aisée des ressources informatiques en fonction des tâches et de la taille du projet ;
- l'accès quasi illimité aux ressources informatiques sans avoir à entretenir et à gérer un parc de serveurs ;
- la possibilité de travailler à distance avec des musiciens de session ;
- une mise à jour des nouvelles versions imperceptible pour l'utilisateur ;
- le stockage fiable des données dans le *Cloud* avec des sauvegardes.

7.3.4 L'estimation de prix

Nous voulons proposer une estimation des prix approximatifs pour ces différentes solutions.

Selon la configuration, le prix moyen d'une station de travail se situe entre 2 et 10 k€. Il peut s'agir d'un ordinateur portable ou de bureau.

Si l'on considère un serveur de calcul, son prix variera de 30 à 100 k€ en fonction du nombre de processeurs, de cartes graphiques et de mémoire vive.

En cas de calcul distribué avec plusieurs nœuds, il faut multiplier la somme de la solution de bureau ou du serveur par le nombre de nœuds nécessaires à son fonctionnement.

Dans le cas de la version en ligne (cloud), le modèle commercial est différent : il s'agit d'acheter du temps de serveur : des minutes, des heures, des mois et des années de calcul. Si vous envisagez des opérations au jour le jour, un service de cloud en ligne sera peut-être moins rentable, car les frais actuels sont assez élevés, surtout lorsque des cartes graphiques sont utilisées. On pourrait facilement payer environ 5 k€ par mois ou plus.

9. <https://blog.landr.com/best-online-daw/>

10. <https://faders.io/>

Un autre inconvénient est que les services en ligne utilisent généralement des couches de *virtualisation*, ce qui réduit la vitesse de calcul de 5 à 20%. L'utilisateur paie donc un prix élevé et n'a pas un accès total aux ressources.

Ce modèle conviendrait peut-être mieux aux musiciens amateurs qui ont tendance à utiliser les services quelques heures par semaine au maximum.

7.4 Est-il nécessaire de complexifier le modèle de démasquage ?

Après avoir effectué des tests perceptifs (cf. Chapitre 6), il est apparu que :

- la solution préférée et sélectionnée par des ingénieurs du son n'est pas nécessairement l'optimum global ;
- il n'est pas nécessaire de faire beaucoup d'itérations et 100 à 200 itérations sont suffisantes pour la plupart des cas ;
- le plus pratique consiste à proposer 2 à 3 pré-mixages, correspondant à des solutions locales, à l'ingénieur du son ;
- il s'avère qu'il n'est pas nécessaire de rechercher une grande précision numérique pour la solution obtenue car une précision suffisamment grossière est suffisante vu que la solution peut être affinée ultérieurement.

Dans ce mémoire, nous avons proposé un modèle très simple et réalisable pour le démasquage et il se pourrait bien que ce modèle, pourtant grossier, soit suffisant pour créer la plupart des pré-mixages. Son principal avantage est la rapidité et la facilité de calcul.

Néanmoins, on pourrait essayer d'utilisations des modèles psycho-acoustiques plus complexes [Glasberg and Moore, 2005]. Dans certains articles, des modèles de codage *Mpeg* potentiellement adaptés au problème ont été utilisés [Tom et al., 2019], mais ils ont considéré la résolution d'un problème de spatialisation diffuse des objets dans l'espace [Pestana and Reiss, 2014], de sorte qu'il est difficile de juger de son efficacité à partir des résultats fournis dans les articles. Comme de tels modèles nécessitent plus de ressources de calcul, nous devrions donc immédiatement penser à écrire une version du code haute performance pour pouvoir les utiliser.

Il convient également de réfléchir à la méthode de travail avec un signal stéréophonique car la plupart des modèles considèrent le cas de signaux monophoniques. Dans ce travail, nous avons également réduit le signal stéréophonique à 2 signaux monophoniques (mid et side). Mais il est probablement plus correct de partir à l'origine d'un modèle stéréophonique pour la perception psycho-acoustique des sons.

7.5 L'analyse préliminaire des pistes par des méthodes d'apprentissage profond

Dans Chapitre 2.2 nous avons examiné de nombreuses opérations de routine qui peuvent être automatisées avec succès à l'aide de modèles d'apprentissage profond ou d'approches traditionnelles.

Le but de cette automatisation est de créer 2 fichiers : un fichier de *projet* et un fichier de *style*. Plus le projet et la spécification artistique sont décrits avec précision dans ces fichiers, plus l'algorithme d'optimisation trouvera facilement et précisément des règles de mixage de qualité.

Cette partie du travail futur a un impact critique sur la qualité du mixage final, cela vaut donc la peine d'accorder beaucoup d'attention à la création de modèles de réseaux neuronaux fiables, précis et robustes pour tout type de matériel musical.

7.6 Les extensions de fonctionnalité

Lors des tests perceptifs, il a été constaté que certains ingénieurs du son percevaient ces pré-mixages comme des mixages finaux, mais n'y trouvaient pas les types de traitement souhaités, tels que la correction spectrale ou le démasquage par l'égalisation, ou au moins un traitement dynamique grossier des instruments. C'est pourquoi nous pensons que ce serait une bonne décision pour le futur modèle d'étendre la fonctionnalité pour inclure au moins la compression, l'égalisation et un traitement spatial plus fin.

7.6.1 L'extension à l'égalisation

La plupart des ingénieurs du son préfèrent résoudre les problèmes de masquage en utilisant l'égalisation à une position fixe (souhaitée) de l'instrument sur la scène virtuelle plutôt que d'utiliser le déplacement spatial de l'instrument. Pour ce faire, nous pouvons nous inspirer de la méthode proposée dans l'article [Hafezi and Reiss, 2015].

Il est également important, au stade initial, d'effectuer une correction spectrale grossière de l'instrument : lisser les fréquences indésirables saillantes et nettoyer les grandes résonances.

7.6.2 L'extension à la compression

Si un instrument a une plage dynamique extrêmement élevée, la réaction naturelle d'un ingénieur du son consiste à appliquer immédiatement un compresseur pour "stabiliser" le son dans l'espace.

Une autre application importante de la compression qui mérite d'être mise en œuvre consiste à considérer la force de la compression (*ratio*) en fonction de la distance de l'instrument par rapport à l'auditeur : plus l'instrument est éloigné, plus de la compression est nécessaire.

Dans le cadre de cette application, on pourrait même envisager d'utiliser un modèle de propagation des ondes sonores dans l'espace qui, en fonction de la distance du son, ajusterait à la fois les paramètres du compresseur et de l'égaliseur.

7.7 L'extension des styles avec des plug-ins externes

7.7.1 Les *inserts* stylistiques

Même aux premiers stades d'écoute de pré-mixages, les ingénieurs du son et les amateurs voulaient entendre des traitements d'instruments et de voix spécifiques à un style de musique particulier.

Cette tâche peut être résolue assez facilement par des méthodes d'apprentissage profond, ou plus précisément par l'identification d'un style ou d'un sous-style [Hsu et al., 2021] par l'analyse de la piste actuelle ainsi que des références si les références sont disponibles. En fonction des résultats de la classification, on peut associer les patterns des chaînes de traitements stylistiques sur les inserts à chaque instrument individuellement, ainsi qu'aux groupes d'instruments et aux bus maître.

7.7.2 Les *sends* et la création de plans de réverbération

Dans ce prototype, nous avons implémenté un plan très primitif de 3 réverbérations avec des présélections de presets et des réglages fixes. Bien que la réverbération n'ait pas été prise en compte dans le cadre de ce travail, nous l'avons mise en œuvre pour des raisons purement pratiques et en pensant à l'avenir. Néanmoins, cette approche simple a semé la confusion chez certains ingénieurs du son.

Par conséquent, si nous décidons d'ajouter des plans de réverbération à un pré-mixage, il est impératif de construire un modèle plus sophistiqué qui tienne compte de la disposition spatiale du son, de sa distance par rapport à l'auditeur, de sa fonction dans l'œuvre musicale et du style général de la composition.

De même, pour certains styles, un seul espace de réverbération peut être suffisant, alors que pour certains autres styles, cinq espaces de réverbération ou plus peuvent être nécessaires. En d'autres termes, les paramètres et les capacités de plans de la réverbération doivent être plus flexibles, pris en compte par l'algorithme d'optimisation et corrigés à l'aide d'un modèle mécaniste de propagation du son dans l'espace.

7.8 La redondance des solutions complique l'optimisation

Comme nous l'avons noté précédemment dans ce chapitre, la précision de la solution numérique ne constitue pas le facteur le plus important pour trouver un ensemble réussi de règles de mixage à l'aide de l'algorithme d'optimisation.

En exécutant le programme VitaMIX de nombreuses fois avec différents paramètres, nous avons remarqué qu'il existe d'innombrables zones locales dans l'espace de recherche qui ne sont pas très différentes les unes des autres ce qui a pour conséquence le fait que l'algorithme perd beaucoup de temps en essayant de les trouver. Par conséquent, nous pouvons recommander d'introduire un espace de recherche discret (une grille de valeurs fixes) afin d'augmenter l'efficacité de l'optimisation, il semble fort probable que cette approche réduira le nombre d'évaluations inutiles pour atteindre les solutions optimales. Mais l'inconvénient pourrait être la stagnation prématurée de l'algorithme et, par conséquent, une réduction du nombre de solutions potentielles trouvées. Cette recommandation devrait donc être testée empiriquement à l'avenir.

7.9 En guise de conclusion

Nous avons brièvement présenté dans ce chapitre les principaux axes d'amélioration du système de pré-mixage automatique VitaMIX. Mais chacun de ces domaines implique un grand nombre de détails techniques qui dépassent le cadre du présent travail.

Il convient également de noter que dans ce mémoire, nous considérons le système de mixage automatique comme un assistant virtuel de l'ingénieur du son. Toutefois, d'autres applications de ce concept sont également possibles. Pour les ingénieurs du son professionnels, ce système peut sembler trop simple et peut-être pas toujours efficace et utile. Mais par exemple, les musiciens débutants, les amateurs et les créateurs du contenu web pourraient tirer un grand profit d'un tel système, car ces personnes ne sont pas compétentes en matière de mixage musical.

Ce système conviendrait également à toute musique qui n'a pas de valeur artistique particulière, par exemple les publicités, la musique d'ambiance, la musique de jeux vidéo simples, etc.

Récemment, le recours à la musique générée automatiquement par ordinateur, plus précisément par réseau neuronal, par exemple *OpenAI*¹¹, s'est également développé. Cette musique a rarement une grande valeur artistique et ne nécessite pas d'analyse approfondie donc elle peut être mixée de manière entièrement automatique. On pourrait aussi envisager également des applications plus modestes¹² dans le domaine de la radio, du mixage de podcasts, etc.

11. <https://openai.com/blog/jukebox/>

12. <https://www.roexaudio.com/>

Chapitre 8

Conclusion

Dans ce mémoire, nous avons effectué un important travail de recherche. L'objectif principal consistait à étudier les possibilités de mixage automatique de morceaux musicaux.

À cette fin, nous avons mené une petite enquête auprès d'ingénieurs du son de différents pays ayant une expérience moyenne de 20 ans ou plus. Nous avons essayé de découvrir leurs habitudes de mixage, ainsi que l'importance qu'ils accordent à une bonne spatialisation du mixage ainsi que les outils et approches qu'ils utilisent pour le faire.

De plus, nous n'avons pas simplement décidé d'automatiser la création d'espace (spatialisation), car outre l'importance absolue de cet aspect, l'analyse statistique du nombre de clics montre qu'un ingénieur du son consacre près de la moitié de son temps à la création et à la correction de la répartition spatiale des instruments. En d'autres termes, si nous étions en mesure d'automatiser les opérations de routine, nous pourrions réduire considérablement le temps consacré au mixage et l'ingénieur du son pourrait se concentrer davantage sur les aspects esthétiques et artistiques de son travail.

Nous avons inventé et développé le premier prototype d'un système de mixage automatique (VitaMIX). Mathématiquement, nous avons modélisé le problème de la spatialisation dans le mixage comme un problème d'optimisation globale.

Souvent, les ingénieurs du son résolvent le problème du positionnement des instruments dans l'espace (balance, panorama) et du démasquage (égaliseur) en séquence. Habituellement, la balance est réglée en monophonique, puis les instruments sont répartis dans l'espace stéréophonique, mais il est rare que les ingénieurs du son règlent la balance et le panorama en même temps. Ils résolvent souvent le problème du démasquage à l'étape suivante et, très souvent, par égalisation lorsque les instruments sont déjà fixés dans l'espace.

Nous avons essayé de résoudre ces problèmes simultanément grâce à des techniques d'optimisation globale, en espérant que cette approche produirait des pré-mixages originaux et uniques d'une qualité suffisante pour les finaliser ultérieurement. Afin de simplifier la tâche, nous avons utilisé un ensemble minimal d'utilitaires dans ce travail, à savoir le

volume, le panorama, et les filtres de haute et basse fréquence. Comme critères d'optimisation, nous avons proposé d'améliorer 4 critères différents : 1) l'équilibre de l'ensemble du mixage ; 2) le masquage des fréquences ; 3) la profondeur et 4) la largeur du mixage.

En utilisant le prototype que nous avons développé, nous avons réalisé le pré-mixage de morceaux de musique. Nous avons pris plusieurs genres différents et augmenté le nombre de pistes à mixer à chaque nouveau morceau pour tester la scalabilité du système proposé, d'un morceau plus simple au plus complexe.

Sur la base de ces résultats, nous avons mis au point des tests de perception afin d'obtenir un retour des ingénieurs du son et des musiciens sur la qualité des mixages automatiques. Il s'est avéré qu'il y avait des mixages appropriés et d'autres non appropriés pour la suite des travaux. Et il est évident que ce système ne peut pas être utilisé dans sa forme actuelle et nécessite un certain nombre d'améliorations.

- Nous sommes arrivés à la conclusion que la précision de l'optimisation ne joue pas un rôle important et que les solutions locales peuvent être suffisantes et satisfaisantes du point de vue esthétique pour leur élaboration ultérieure.
- Nous avons réalisé qu'une meilleure mise à l'échelle nécessite d'adapter les critères d'optimisation et de les rendre dépendants du nombre de pistes à mixer.
- Il nous a semblé juste de tester empiriquement des modèles psycho-acoustiques plus sophistiqués pour l'estimation du masquage.
- Nous avons réalisé qu'une plus grande palette d'utilitaires dans la phase d'optimisation était absolument nécessaire et qu'il fallait introduire l'égalisation et au moins une compression grossière.
- Nous avons réalisé l'importance de construire des plans de réverbération plus précis, ou de les éliminer complètement à ce stade de la création du pré-mixage.

Nous avons également proposé toute une série de solutions techniques et technologiques pour transformer ce prototype en un produit complet. Un ajout important serait l'utilisation de techniques d'apprentissage profond pour le pré-traitement et l'analyse des pistes mixées, afin de façonner plus précisément la tâche d'optimisation ainsi que les réglages du mixeur.

Nous avons examiné plusieurs possibilités d'amélioration de l'environnement de travail d'un ingénieur du son, comme un plug-in intégré à la station audio-numérique (DAW) qui pourrait être interactif, ou une solution en ligne à distance. Nous avons procédé à une évaluation des coûts de la nouvelle configuration de l'environnement de travail, et avons tenté d'en identifier les avantages et les inconvénients.

Nous sommes conscients qu'en raison de la complexité et de l'ambiguïté du travail d'un ingénieur du son, ainsi que des nombreuses caractéristiques esthétiques qui ne peuvent être exprimées sous forme mathématique, il sera difficile, voire impossible, de créer un système de mixage automatique parfait.

Par conséquent, nous avons proposé de se concentrer sur les marchés alternatifs pour l'utilisation d'un tel système. Pour les musiciens amateurs et les personnes qui n'ont pas une grande expérience de l'ingénierie du son, un tel système serait probablement utile pour créer des ébauches d'œuvres musicales.

En outre, dans un certain nombre de domaines où la valeur esthétique de la musique n'est pas forcément le premier critère (comme dans la publicité, les podcasts, les jeux vidéo, la musique générée automatiquement par l'intelligence artificielle, etc.), il n'y a pas d'exigence excessive pour la qualité du produit musical final et l'utilisation d'un système de mixage automatique devient alors potentiellement possible.

Nous conseillons également de faire évoluer le projet vers la conception d'un assistant virtuel interactif de l'ingénieur du son, tel qu'une sorte d'élément d'intelligence artificielle qui faciliterait le travail sur les activités de routine, alors que le projet musical resterait sous le contrôle constant de l'ingénieur du son. Mais ce sujet nécessite une étude séparée et approfondie.

Nous espérons que ce travail sera le point de départ de solutions plus avancées et innovantes, qui aideront l'ingénieur du son à se concentrer entièrement sur les questions esthétiques et à développer pleinement son talent de créateur lorsqu'il travaille sur les aspects artistiques d'un morceau de musique.

Tout commentaire est le bienvenu et nous sommes toujours prêts à discuter de ce sujet et d'autres sujets liés à l'ingénierie du son. Tous ceux qui apprécient nos idées, peuvent nous contacter à l'adresse électronique suivante : jean.vital@gmail.com.

Bibliographie

- Agrawal, M. and Nandy, A. (2020). A novel multimodal music genre classifier using hierarchical attention and convolutional neural network.
- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies - a comprehensive introduction. *Natural Computing*, 1 :3–52.
- Binford, T., Ruby, J., D, P., Nedumaan, J., Lepika, J., and Tisa, J. (2019). *Modern Deep Learning and Advanced Computer Vision*.
- Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Elsevier.
- Brucker, P. (2010). *Scheduling Algorithms*. Springer Publishing Company, Incorporated, 5th edition.
- Case, A. (2012). *Mix Smart*. Focal Press.
- Castelfranchi, C. and Falcone, R. (2010). *Trust Theory : A Socio-Cognitive and Computational Model*. Wiley Publishing, 1st edition.
- Chai, J., Zeng, H., Li, A., and Ngai, E. W. (2021). Deep learning in computer vision : A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6 :100134.
- Das, S. and Kolya, A. K. (2020). Detecting generic music features with single layer feedforward network using unsupervised hebbian computation.
- De Man, B. (2017). *Towards a better understanding of mix engineering*. PhD thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom.
- Decherf, P. (22 février 2021). Apprenez le vocabulaire de la production musicale. <https://handsupelectro.fr/dossier/apprenez-le-vocabulaire-de-la-production-musicale/>.
- Dorigo, M. and Blum, C. (2005). Ant colony optimization theory : A survey. *Theor. Comput. Sci.*, 344 :243–278.
- Egiazaryan, G. and Sudakov, K. (2007). Theory of functional systems in the scientific school of p.k. anokhin. *Journal of the history of the neurosciences*, 16 :194–205.

- Engelbrecht, A. (2007). *Computational Intelligence : An Introduction*. Wiley.
- Feoktistov, V. (2006). *Differential Evolution : In Search of Solutions*, volume 5. Springer USA.
- Fogel, D. (1995). *Evolutionary Computation : Toward a New Philosophy of Machine Intelligence*, volume 1.
- Fuhrmann, F. (2012). *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra.
- Gao, F. and Han, L. (2012). Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51 :259–277.
- Giannoulis, D., Massberg, M., and Reiss, J. D. (2012). Digital dynamic range compressor design - A tutorial and analysis. *AES : Journal of the Audio Engineering Society*.
- Giannoulis, D., Massberg, M., and Reiss, J. D. (2013). Parameter automation in a dynamic range compressor. *AES : Journal of the Audio Engineering Society*.
- Gibson, D. (2019). *The Art of Mixing*. Routledge.
- Gigerenzer, G. (2002). *Adaptive thinking rationality in the real world*. Oxford University Press,, New York ;.
- Glasberg, B. and Moore, B. (2005). Development and evaluation of a model for predicting the audibility of time-varying sounds in the presence of background sounds. *AES : Journal of the Audio Engineering Society*, 53 :906–918.
- Glover, F. and Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA.
- Hafezi, S. and Reiss, J. D. (2015). Autonomous multitrack equalization based on masking reduction. *AES : Journal of the Audio Engineering Society*.
- Hart, E., McEwan, C., Timmis, J., and Hone, A. (2011). Advances in artificial immune systems. *Evolutionary Intelligence*, 4 :67–68.
- Hauschild, M. and Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1 :111–128.
- He, T., Liu, W., Gong, C., Yan, J., and Zhang, N. (2021). Music plagiarism detection via bipartite graph matching.
- Herre, J. and Dick, S. (2019). Psychoacoustic models for perceptual audio coding—a tutorial review. *Applied Sciences*, 9(14).
- Holland, J. H. (1992). Genetic algorithms. *Scientific American*.
- Hsu, W.-H., Chen, B.-Y., and Yang, Y.-H. (2021). Deep learning based edm subgenre classification using mel-spectrogram and tempogram features.

-
- Hu, H., Zhang, X., Yan, X., Wang, L., and Xu, Y. (2017). Solving a new 3d bin packing problem with deep reinforcement learning method.
- Izhaki, R. (2017). *Mixing Audio*. Focal Press.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning : with Applications in R*. Springer.
- Jillings, N. and Stables, R. (2017). An Intelligent audio workstation in the browser. In *Proceedings of the International Web Audio Conference*.
- Johnson, A. and Jacobson, S. (2002a). A class of convergent generalized hill climbing algorithms. *Applied Mathematics and Computation*, 125 :359–373.
- Johnson, A. and Jacobson, S. (2002b). On the convergence of generalized hill climbing algorithms. *Discrete Applied Mathematics*, 119(1) :37–57. Special Issue devoted to Foundation of Heuristics in Combinatorial Optimization.
- Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948.
- Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science (New York, N.Y.)*, 220 :671–80.
- Lauriola, I., Lavelli, A., and Aiolli, F. (2022). An introduction to deep learning in natural language processing : Models, techniques, and tools. *Neurocomputing*, 470 :443–456.
- Luber, S. (2011). Cognitive science artificial intelligence : Simulating the human mind to achieve goals. pages 207 – 210.
- Ma, Z., De Man, B., Pestana, P. D., Black, D. A., and Reiss, J. D. (2015). Intelligent multitrack dynamic range compression. *AES : Journal of the Audio Engineering Society*.
- Mahanta, S., Khilji, A., and Pakray, D. P. (2021). Deep neural network for musical instrument recognition using mfccs.
- Mansbridge, S., Finn, S., and Reiss, J. D. (2012). An autonomous system for multi-track stereo pan positioning. In *133rd Audio Engineering Society Convention 2012, AES 2012*.
- Molina, D., Latorre, A., and Herrera, F. (2018). Shade with iterative local search for large-scale global optimization. pages 1–8.
- Owsinski, B. (1999). *Mixing Engineer’s Handbook*.
- Pandey, R. and Rathi, R. (2009). Theoretical brain model using artificial neural networks.
- Perez-Gonzalez, E. and Reiss, J. (2009). Automatic equalization of multi-channel audio using cross-adaptive methods. In *127th Audio Engineering Society Convention 2009*.
- Perez-Gonzalez, E. and Reiss, J. D. (2010). A real-time semiautonomous audio panning system for music mixing. *Eurasip Journal on Advances in Signal Processing*.

- Perlovsky, L. (2006). Toward physics of the mind : Concepts, emotions, consciousness, and symbols. *Physics of Life Reviews*, 3 :23–55.
- Pestana, P. D. (February 2013). *Automatic mixing systems using adaptive digital audio effects*. Phd thesis, Universidade Catolica Portuguesa.
- Pestana, P. D. and Reiss, J. D. (2014). A cross-adaptive dynamic spectral panning technique. In *DAFx 2014 - Proceedings of the 17th International Conference on Digital Audio Effects*.
- Pimenta-Zanon, M. H., Bressan, G. M., and Lopes, F. M. (2021). Complex network-based approach for feature extraction and classification of musical genres.
- Pinedo, M. L. (2016). *Scheduling : Theory, Algorithms, and Systems*. Springer, New York, NY, 5th edition.
- Schindler, A., Lidy, T., and Böck, S. (2020). Deep learning for mir tutorial.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press.
- Spyridon, S. (2019). *Audio equalisation using natural language*. PhD thesis, Digital Media Technology Lab, Faculty of Computing, Engineering and the Built Environment. Birmingham City University.
- Steinmetz, C. (2020). *Learning to mix with neural audio effects in the waveform domain*. PhD thesis, Universitat Pompeu Fabra, Barcelona.
- Steinmetz, C. J., Pons, J., Pascual, S., and Serrà, J. (2021). Automatic multitrack mixing with a differentiable mixing console of neural audio effects. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*.
- Sudakov, K. (2015). Theory of functional systems : A keystone of integrative biology. *Cognitive Systems Monographs*, 25 :153–173.
- Tan, K., Lee, T., and Khor, E. (2002). Evolutionary algorithms for multi-objective optimization : Performance assessments and comparisons. *Artif. Intell. Rev.*, 17 :251–290.
- Tom, A., Reiss, J., and Depalle, P. (2019). An automatic mixing system for multitrack spatialization for stereo based on unmasking and best panning practices. In *AES 146th International Convention*.
- Tot, J. (March 2018). *Multitrack mixing : An investigation into music mixing practices*. Master's thesis, MSc Music Technology, Staffordshire University.
- van Harmelen, F., Lifschitz, V., and Porter, B., editors (2008). *Handbook of Knowledge Representation*. Foundations of Artificial Intelligence. Elsevier Science, 1 edition.
- Venkatalakshmi (2020). Artificial intelligence, does it has ability to mimic human intelligence. *Frontiers in Robotics and AI*, 8 :1.
- Vickers, E. (2001). Automatic long-term loudness and dynamics matching. *Journal of the audio engineering society*.

-
- Vital, J.-A. (2021a). Questionnaire : Mixage musical (français). <https://forms.gle/WDnk2ZiXVYL5vGXW6>.
- Vital, J.-A. (2021b). Questionnaire : Mixage musical (russe). <https://forms.gle/6QYk4fy7CHPvW7BZA>.
- Vital, J.-A. (2021c). Survey : Perceptual tests (english). <https://forms.gle/npiHG1zNh8CTU1fM6>.
- VITAL, J.-A. (2022). Surveys on musical mixing and perception tests. <https://drive.google.com/drive/folders/17RC9N8qiDqqtJDEbpQcMH8FMZYhUoEyh?usp=sharing>.
- Wilson, A. (2017). *Evaluation and Modelling of Perceived Audio Quality in Popular Music, towards Intelligent Music Production*. PhD thesis, School of Computing, Science and Engineering. Salford Innovation Research Centre.
- Yang, G. and Wang, X.-J. (2020). Artificial neural networks for neuroscientists : A primer. *Neuron*, 107 :1048–1070.
- Zeng, D. D. and Mao, W. (2014). Supporting global collective intelligence via artificial intelligence. *Intelligent Systems, IEEE*, 29 :2–4.
- Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). Dive into deep learning.

Table des figures

3.1	3 plans pour le volume.	54
3.2	Les plans panoramiques : secteurs, sous-secteurs et valeurs fixes.	55
3.3	Les paramètres passe-haut (locut) et passe-bas (hicut).	56
3.4	Les regroupements de pistes et les bus.	58
3.5	Un exemple de description de projet.	59
3.6	L'architecture de VitaMIX.	62
5.1	Deux découpages en bandes de fréquences : 1) classique ; 2) recommandé par Pascal Spitz.	89
6.1	La répartition en pourcentages des participants aux tests perceptifs.	96
6.2	Exemple : graphe de convergence du modèle psychoacoustique (Choir).	98
6.3	Exemple : graphe récapitulatif des paramètres du modèle (Choir).	100
6.4	Exemple : graphe des paramètres pour un instrument (Choir).	102
E.1	2guitars : Le pré-mixage préféré.	180
E.2	2guitars : L'intention d'utiliser le pré-mixage préféré.	180
E.3	2guitars : Graphe de convergence – la fonction objectif.	181
E.4	2guitars : Graphe de convergence – tous les instruments ensemble.	182
E.5	2guitars : Graphe de convergence – <i>01_MainPair</i>	183
E.6	2guitars : Graphe de convergence – <i>02_DI</i>	184
E.7	Folk : Le pré-mixage préféré.	185
E.8	Folk : L'intention d'utiliser le pré-mixage préféré.	185
E.9	Folk : Graphe de convergence – la fonction objectif.	186
E.10	Folk : Graphe de convergence – tous les instruments ensemble.	187
E.11	Folk : Graphe de convergence – <i>260</i>	188
E.12	Folk : Graphe de convergence – <i>Guitar-Steve-CU29</i>	189
E.13	Folk : Graphe de convergence – <i>251</i>	190
E.14	Folk : Graphe de convergence – <i>Vocals-Steve-U47</i>	191
E.15	Ska : Le pré-mixage préféré.	193
E.16	Ska : L'intention d'utiliser le pré-mixage préféré.	193
E.17	Ska : Graphe de convergence – la fonction objectif.	194
E.18	Ska : Graphe de convergence – tous les instruments ensemble.	195
E.19	Ska : Graphe de convergence – <i>01_Kick</i>	196
E.20	Ska : Graphe de convergence – <i>02_Snare</i>	197
E.21	Ska : Graphe de convergence – <i>03_Overhead</i>	198

E.22Ska : Graphe de convergence – 04_BassDI.	199
E.23Ska : Graphe de convergence – 05_Gtr.	200
E.24Ska : Graphe de convergence – 06_Percussion.	201
E.25Ska : Graphe de convergence – 07_LeadVox.	202
E.26Latino : Le pré-mixage préféré.	204
E.27Latino : L'intention d'utiliser le pré-mixage préféré.	204
E.28Latino : Graphe de convergence – la fonction objectif.	205
E.29Latino : Graphe de convergence – tous les instruments ensemble.	206
E.30Latino : Graphe de convergence – 01_Percussion.	207
E.31Latino : Graphe de convergence – 02_Woodblock.	208
E.32Latino : Graphe de convergence – 03_TomsCymbals.	209
E.33Latino : Graphe de convergence – 04_Bass.	210
E.34Latino : Graphe de convergence – 05_Piano.	211
E.35Latino : Graphe de convergence – 06_Vibes.	212
E.36Latino : Graphe de convergence – 07_Strings1.	213
E.37Latino : Graphe de convergence – 08_Strings2.	214
E.38Latino : Graphe de convergence – 09_FrenchHorns.	215
E.39Latino : Graphe de convergence – 10_LeadVox.	216
E.40Country : Le pré-mixage préféré.	218
E.41Country : L'intention d'utiliser le pré-mixage préféré.	219
E.42Country : Graphe de convergence – la fonction objectif.	219
E.43Country : Graphe de convergence – tous les instruments ensemble.	220
E.44Country : Graphe de convergence – 01_Kick.	221
E.45Country : Graphe de convergence – 02_Snare.	222
E.46Country : Graphe de convergence – 03_Tom.	223
E.47Country : Graphe de convergence – 04_Overheads.	224
E.48Country : Graphe de convergence – 05_Bass.	225
E.49Country : Graphe de convergence – 06_AcGtr.	226
E.50Country : Graphe de convergence – 07_Dobro.	227
E.51Country : Graphe de convergence – 08_ElecGtr.	228
E.52Country : Graphe de convergence – 09_Fiddle.	229
E.53Country : Graphe de convergence – 10_Piano.	230
E.54Country : Graphe de convergence – 11_LeadVox.	231
E.55Choir : Le pré-mixage préféré.	234
E.56Choir : L'intention d'utiliser le pré-mixage préféré.	234
E.57Choir : Graphe de convergence – la fonction objectif.	235
E.58Choir : Graphe de convergence – tous les instruments ensemble.	236
E.59Choir : Graphe de convergence – 01_MainPair1.	237
E.60Choir : Graphe de convergence – 02_MainPair2.	238
E.61Choir : Graphe de convergence – 03_SpotMic1_Tenors.	239
E.62Choir : Graphe de convergence – 04_SpotMic2_Basses.	240
E.63Choir : Graphe de convergence – 05_SpotMic3_Sopranos.	241
E.64Choir : Graphe de convergence – 06_SpotMic4_Altos.	242
E.65Choir : Graphe de convergence – 07_BassCloseMic1.	243
E.66Choir : Graphe de convergence – 08_BassCloseMic2.	244

E.67	Choir : Graphe de convergence – 09_BassCloseMic3.	245
E.68	Choir : Graphe de convergence – 10_BassCloseMic4.	246
E.69	Choir : Graphe de convergence – 11_BassCloseMic5.	247
E.70	Choir : Graphe de convergence – 12_LeadVox.	248
E.71	Jazz : Le pré-mixage préféré.	250
E.72	Jazz : L'intention d'utiliser le pré-mixage préféré.	251
E.73	Jazz : Graphe de convergence – la fonction objectif.	251
E.74	Jazz : Graphe de convergence – tous les instruments ensemble.	252
E.75	Jazz : Graphe de convergence – 01_KickIn.	253
E.76	Jazz : Graphe de convergence – 02_KickOut.	254
E.77	Jazz : Graphe de convergence – 03_Snare.	255
E.78	Jazz : Graphe de convergence – 04_Tom1.	256
E.79	Jazz : Graphe de convergence – 05_Tom2.	257
E.80	Jazz : Graphe de convergence – 06_Overhead1.	258
E.81	Jazz : Graphe de convergence – 07_Overhead2.	259
E.82	Jazz : Graphe de convergence – 08_BassDI.	260
E.83	Jazz : Graphe de convergence – 10_ElecGtrMic1.	261
E.84	Jazz : Graphe de convergence – 11_ElecGtrMic2.	262
E.85	Jazz : Graphe de convergence – 13_Piano.	263
E.86	Ballad : Le pré-mixage préféré.	266
E.87	Ballad : L'intention d'utiliser le pré-mixage préféré.	266
E.88	Ballad : Graphe de convergence – la fonction objectif.	267
E.89	Ballad : Graphe de convergence – tous les instruments ensemble.	268
E.90	Ballad : Graphe de convergence – 01_Kick.	269
E.91	Ballad : Graphe de convergence – 02_Snare.	270
E.92	Ballad : Graphe de convergence – 03_HiHat.	271
E.93	Ballad : Graphe de convergence – 04_Overheads.	272
E.94	Ballad : Graphe de convergence – 05_CymbalRoll1.	273
E.95	Ballad : Graphe de convergence – 06_CymbalRoll2.	274
E.96	Ballad : Graphe de convergence – 07_Bass.	275
E.97	Ballad : Graphe de convergence – 08_AcGuitar1.	276
E.98	Ballad : Graphe de convergence – 09_AcGuitar2.	277
E.99	Ballad : Graphe de convergence – 10_AcGuitar3.	278
E.100	Ballad : Graphe de convergence – 11_AcGuitar4.	279
E.101	Ballad : Graphe de convergence – 12_AcGuitar5.	280
E.102	Ballad : Graphe de convergence – 13_Piano.	281
E.103	Ballad : Graphe de convergence – 14_Pad1.	282
E.104	Ballad : Graphe de convergence – 15_Pad2.	283
E.105	Ballad : Graphe de convergence – 16_LeadVox.	284

Annexe A

Liste des personnes participant aux questionnaires

Seuls les noms des participants qui en ont autorisé la divulgation sont présentés ici.

A.1 Le mixage musical

- Zemlinskiy Lev Ilich
- Kozyrev Nikolai Nikolaevich
- Dima Kovalentchick
- Zviagin Oleg Viacheslavovich
- Dmitry Hamkalov
- Mikhail Kidt
- Anton Vovk ("ShamanTon")
- Zimin Dmitriy Iurievich
- Onischenko Euvgeniy Viktorovich
- Denis Vareha
- Benoît Fleury

A.2 Les tests perceptifs

- Ilya Lukashev
- Modan Tailleur
- Anna
- Antoine
- Guy-Marc Guede
- Ana Psuturi

Annexe B

L'intelligence artificielle

B.1 Le contexte philosophique

L'idée du concept d'"intelligence artificielle" a été grandement influencée par la naissance du matérialisme mécaniste, qui commence avec le "Discours de la méthode" de René Descartes (1637) et immédiatement après avec la "Nature humaine" de Thomas Hobbes (1640). En effet, René Descartes a supposé que l'animal est une sorte de mécanisme complexe, formulant ainsi la théorie mécaniste.

B.2 Les conditions technologiques

Wilhelm Schickard a construit le premier ordinateur numérique mécanique en 1623, suivi par les machines de Blaise Pascal (1643) et de Leibniz (1671). Leibniz a également été le premier à décrire le système moderne des nombres binaires bien, qu'avant lui, de nombreux grands scientifiques aient été périodiquement fascinés par ce système. En 1832, le conseiller collégial S. N. Korsakov a mis en avant le principe de l'élaboration de méthodes et de dispositifs scientifiques destinés à améliorer les capacités de l'esprit et a proposé une série de "machines intelligentes", dont la conception, pour la première fois dans l'histoire de l'informatique, utilise des cartes perforées. Au XIXe siècle, Charles Babbage et Ada Lovelace ont travaillé sur une machine à calculer mécanique programmable.

B.3 La naissance de la science : "Intelligence artificielle"

Entre 1910 et 1913, Bertrand Russell et A. N. Whitehead publient "Principles of Mathematics", qui révolutionne la logique formelle. En 1941, Konrad Zuse a construit le premier ordinateur commandé par programme. Warren McCulloch et Walter Pitts ont publié en 1943 "A Logical Calculus of the Ideas Immanent in Nervous Activity", qui a jeté les bases des réseaux neuronaux. Ils ont notamment proposé un modèle de neurones artificiels. D. Hebb dans "The Organization of Behavior" en 1949 a décrit les principes de base de l'apprentissage neuronal. Ces idées ont été développées quelques années plus tard par le neurophysiologiste américain Frank Rosenblatt qui a proposé un schéma d'un dispositif qui simule le processus perceptif humain et l'a appelé le "perceptron".

B.4 L'intelligence artificielle moderne

De nombreux domaines scientifiques et méthodes de résolution de problèmes sont désormais regroupés sous l'égide de l'intelligence artificielle. Parmi celles-ci figurent la modélisation du cerveau humain [Perlovsky, 2006; Pandey and Rathi, 2009; Yang and Wang, 2020] et la modélisation de la manière dont les humains résolvent les problèmes [Venkata-lakshmi, 2020; Lubner, 2011], l'apprentissage statistique [James et al., 2013] et automatique [Shalev-Shwartz and Ben-David, 2014; Zhang et al., 2021], la représentation des connaissances [Brachman and Levesque, 2004; van Harmelen et al., 2008], la planification [Pinedo, 2016; Brucker, 2010], le traitement du langage naturel [Lauriola et al., 2022], la vision et la perception par ordinateur [Binford et al., 2019; Chai et al., 2021], l'intelligence computationnelle [Engelbrecht, 2007], l'intelligence sociale [Gigerenzer, 2002; Castelfranchi and Falcone, 2010] et globale [Zeng and Mao, 2014].

Il est intéressant de noter la classification des notions d'intelligence artificielle :

- au *sens étroit*, où l'intelligence artificielle vise à résoudre une tâche étroitement ciblée et n'est pas capable de dépasser un contexte fixe ;
- au *sens large*, où l'intelligence artificielle est capable de résoudre une large classe de tâches et possède des propriétés d'adaptation et d'auto-adaptation à la tâche.

Aujourd'hui, les scientifiques ont appris à résoudre de nombreux problèmes d'intelligence artificielle dans un sens étroit, et dans des domaines très restreints, comme par exemple la vision par ordinateur, où l'intelligence construite artificiellement dépasse de loin les capacités et les aptitudes humaines. Mais, au sens large, l'intelligence artificielle moderne s'adapte encore très mal à un environnement en constante mutation. Toutes les recherches et tendances modernes visent à étendre les capacités d'auto-apprentissage et d'auto-adaptation de l'intelligence artificielle à autant de conditions que possible, la rapprochant ainsi de l'intelligence humaine.

L'apogée de l'intelligence artificielle sera sa capacité à se fixer ses propres objectifs et à les résoudre.

Il y a plus de 50 ans, les scientifiques réfléchissaient déjà à la manière d'organiser un tel système basé sur la théorie biologique des systèmes fonctionnels d'Anokhin [Egiazaryan and Sudakov, 2007; Sudakov, 2015]. Les éléments de base de l'orientation vers un but et de la capacité de survie des systèmes biologiques ont servi d'inspiration pour développer des systèmes de contrôle automatique intelligents pendant l'ère soviétique. Peut-être que dans les décennies à venir, ces idées seront réinterprétées, complétées et étendues, en utilisant les ressources informatiques et puissances de calcul modernes, pour construire des systèmes intelligents au sens large.

B.5 L'apprentissage automatique

Le groupe de méthodes d'apprentissage automatique, qui est devenu très populaire au cours des dernières décennies, est particulièrement intéressant. Parmi toutes les approches

d'apprentissage automatique, il existe l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Les méthodes les plus courantes sont les méthodes de vecteurs de support, les arbres aléatoires ("random forest") et l'apprentissage profond ("deep learning").

Le concept de base ici est que les caractéristiques ("features") sont extraites de grandes quantités de données, à la discrétion de l'utilisateur, et que des modèles sont construits sur la base de ces caractéristiques. Les exemples très connus sont la régression linéaire ou la régression logistique. Il convient de noter que les méthodes sont très sensibles à différents ensembles de caractéristiques, de sorte que la sélection des bonnes caractéristiques n'est pas une tâche triviale en soi, et, qu'elle est entièrement déterminée par l'utilisateur d'un tel système.

B.6 L'apprentissage profond

Dans le cas de l'apprentissage profond, il n'est plus nécessaire de passer par une étape d'extraction de caractéristiques, cette partie du travail est effectuée automatiquement par la technologie elle-même. La principale caractéristique des méthodes d'apprentissage profond est la nécessité de disposer de grandes quantités de données ("big data") afin que l'apprentissage à partir d'exemples de ces données puisse se dérouler de la meilleure façon possible pour obtenir une solution de qualité.

Les réseaux neuronaux eux-mêmes existent depuis plusieurs décennies. Mais c'est grâce aux progrès technologiques des nouvelles architectures de calcul parallèle qu'il est devenu possible de construire de très grands réseaux neuronaux (d'où vient le mot "profond") qui dépendent de millions de paramètres et de plusieurs milliers de fonctions non linéaires. Nous parlons ici des cartes graphiques, des nouvelles façons de les utiliser et de leurs nouvelles architectures spécifiquement adaptées à la résolution des problèmes d'apprentissage profond. Alors que les puissants serveurs contiennent 10 à 12 cœurs par processeur, une carte graphique moderne¹ en 2021 compte jusqu'aux 7 000 cœurs. Cela permet d'obtenir une énorme puissance de calcul (19.5 téraflops) pour une faible consommation d'énergie (400 W).

B.7 L'intelligence computationnelle

Un autre domaine important qu'il convient de souligner ici est l'intelligence computationnelle. Elle comprend trois groupes de méthodes : la logique floue, les réseaux neuronaux et le calcul évolutif ("evolutionary computation").

Les logiques floues ont permis de remplacer les modèles de probabilité, pour les cas où il est impossible de deviner à l'avance quelle loi de distribution suit une variable aléatoire. Cette approche est devenue très populaire pour les systèmes de contrôle automatique et se caractérise par sa stabilité et la robustesse, même dans le cas de conditions imprévisibles.

1. Le GPU NVIDIA A100 Tensor Core

Le calcul évolutif est devenu populaire au moment de l'émergence des algorithmes génétiques et de la programmation génétique. Les algorithmes génétiques modélisaient les processus de sélection naturelle inhérents à l'évolution et étaient utilisés pour résoudre des problèmes techniques. Alors que la programmation génétique permettait l'écriture automatique d'un code informatique optimal.

B.8 Le changement du paradigme

Il convient de noter que les technologies d'apprentissage profond et de calcul évolutif, également connu sous le nom d'intelligence des foules ("swarm intelligence"), sont officiellement reconnues comme des technologies de rupture de la dernière décennie, des technologies qui ont fondamentalement changé la façon dont nous résolvons les problèmes.

Alors qu'auparavant, les problèmes complexes exigeaient des mathématiciens et des programmeurs qu'ils conçoivent les heuristiques et les algorithmes les plus complexes afin de s'approcher au moins de leur solution, les nouvelles approches révolutionnaires éliminent la nécessité d'écrire des milliers de lignes de code. En effet, l'algorithme se construit automatiquement; on pourrait même dire qu'il écrit automatiquement du code. Le paradigme de la programmation et de la résolution des tâches a donc complètement changé.

B.9 L'optimisation numérique

Qu'ont donc en commun ces deux technologies révolutionnaires que sont l'apprentissage profond et l'intelligence des foules? Le dénominateur commun est constitué par les techniques d'optimisation qui sous-tendent ces technologies. Nous arrêterons ici notre très brève introduction à l'intelligence artificielle, et renverrons notre lecteur au chapitre consacré à une introduction plus détaillée aux méthodes de l'optimisation (Chapitre 4).

Annexe C

Les projets

C.1 Classic : Rod Alexander - Canon : 2 tracks

```
1 {
2     "project_name": "Rod Alexander Canon - 2 guitars",
3     "project_path": "D:\\RodAlexander_Canon_Full\\",
4     "sample_rate": 44100,
5     "start_time": 0,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_MainPair.wav",
10            "track_vol": "mid",
11            "track_pan": "any",
12            "track_locut": "80",
13            "track_hicut": "20000",
14            "track_eq": "normal",
15            "track_comp": "none",
16            "track_group": "music"
17        },
18        {
19            "track_name": "02_DI.wav",
20            "track_vol": "mid",
21            "track_pan": "any",
22            "track_locut": "80",
23            "track_hicut": "10000",
24            "track_eq": "agressive",
25            "track_comp": "attack",
26            "track_group": "music"
27        }
28    ]
29 }
```

C.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks

```
1 {
2     "project_name": "Leslie Mendelson - The Hardest Part 2",
3     "project_path": "D:\\Leslie Mendelson_The Hardest Part 2\\",
4     "sample_rate": 48000,
5     "start_time": 122,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "Acoustic Guitar-Leslie-Ela M 260.wav"
10            ,
11            "track_vol": "mid",
12            "track_pan": "left",
13            "track_locut": "80",
14            "track_hicut": "20000",
15            "track_eq": "none",
16            "track_comp": "none",
17            "track_group": "music"
18        },
19        {
20            "track_name": "Acoustic Guitar-Steve-CU29.wav",
21            "track_vol": "mid",
22            "track_pan": "right",
23            "track_locut": "80",
24            "track_hicut": "20000",
25            "track_eq": "none",
26            "track_comp": "none",
27            "track_group": "music"
28        },
29        {
30            "track_name": "Vocals-Leslie- Ela M 251.wav",
31            "track_vol": "front",
32            "track_pan": "left",
33            "track_locut": "200",
34            "track_hicut": "20000",
35            "track_eq": "none",
36            "track_comp": "none",
37            "track_group": "vocal"
38        },
39        {
40            "track_name": "Vocals-Steve-U47.wav",
41            "track_vol": "mid",
42            "track_pan": "right",
43            "track_locut": "80",
```

```
43         "track_hicut": "10000",
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "vocal"
47     }
48 ]
49 }
```

C.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks

```
1 {
2     "project_name": "Big Stone Culture - Fragile Thoughts",
3     "project_path": "D:\\BigStoneCulture_FragileThoughts_Full\\",
4     "sample_rate": 44100,
5     "start_time": 44,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_Kick.wav",
10            "track_vol": "front",
11            "track_pan": "C",
12            "track_locut": "100",
13            "track_hicut": "6000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "drums"
17        },
18        {
19            "track_name": "02_Snare.wav",
20            "track_vol": "front",
21            "track_pan": "C",
22            "track_locut": "300",
23            "track_hicut": "20000",
24            "track_eq": "none",
25            "track_comp": "none",
26            "track_group": "drums"
27        },
28        {
29            "track_name": "03_Overhead.wav",
30            "track_vol": "mid",
31            "track_pan": "C",
32            "track_locut": "400",
33            "track_hicut": "16000",
34            "track_eq": "none",
```

```
35         "track_comp": "none",
36         "track_group": "drums"
37     },
38     {
39         "track_name": "04_BassDI.wav",
40         "track_vol": "front",
41         "track_pan": "C",
42         "track_locut": "60",
43         "track_hicut": "5000",
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "music"
47     },
48     {
49         "track_name": "05_Gtr.wav",
50         "track_vol": "front",
51         "track_pan": "C",
52         "track_locut": "300",
53         "track_hicut": "6000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "music"
57     },
58     {
59         "track_name": "06_Percussion.wav",
60         "track_vol": "mid",
61         "track_pan": "mid",
62         "track_locut": "500",
63         "track_hicut": "10000",
64         "track_eq": "none",
65         "track_comp": "none",
66         "track_group": "drums"
67     },
68     {
69         "track_name": "07_LeadVox.wav",
70         "track_vol": "front",
71         "track_pan": "C",
72         "track_locut": "200",
73         "track_hicut": "20000",
74         "track_eq": "none",
75         "track_comp": "none",
76         "track_group": "vocal"
77     }
78 ]
79 }
```

C.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks

```
1 {
2     "project_name": "Eddie Garrido - Una Semana Sin Ti",
3     "project_path": "D:\\EddieGarrido_UnaSemanaSinTi_Full\\",
4     "sample_rate": 44100,
5     "start_time": 130,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_Percussion.wav",
10            "track_vol": "mid",
11            "track_pan": "mid",
12            "track_locut": "300",
13            "track_hicut": "10000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "drums"
17        },
18        {
19            "track_name": "02_Woodblock.wav",
20            "track_vol": "mid",
21            "track_pan": "left",
22            "track_locut": "300",
23            "track_hicut": "10000",
24            "track_eq": "none",
25            "track_comp": "none",
26            "track_group": "drums"
27        },
28        {
29            "track_name": "03_TomsCymbals.wav",
30            "track_vol": "mid",
31            "track_pan": "any",
32            "track_locut": "100",
33            "track_hicut": "15000",
34            "track_eq": "none",
35            "track_comp": "none",
36            "track_group": "drums"
37        },
38        {
39            "track_name": "04_Bass.wav",
40            "track_vol": "front",
41            "track_pan": "C",
42            "track_locut": "60",
43            "track_hicut": "6000",
```

```
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "music"
47     },
48     {
49         "track_name": "05_Piano.wav",
50         "track_vol": "mid",
51         "track_pan": "any",
52         "track_locut": "200",
53         "track_hicut": "9000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "music"
57     },
58     {
59         "track_name": "06_Vibes.wav",
60         "track_vol": "mid",
61         "track_pan": "any",
62         "track_locut": "200",
63         "track_hicut": "7000",
64         "track_eq": "none",
65         "track_comp": "none",
66         "track_group": "music"
67     },
68     {
69         "track_name": "07_Strings1.wav",
70         "track_vol": "any",
71         "track_pan": "any",
72         "track_locut": "200",
73         "track_hicut": "8000",
74         "track_eq": "none",
75         "track_comp": "none",
76         "track_group": "music"
77     },
78     {
79         "track_name": "08_Strings2.wav",
80         "track_vol": "any",
81         "track_pan": "any",
82         "track_locut": "200",
83         "track_hicut": "8000",
84         "track_eq": "none",
85         "track_comp": "none",
86         "track_group": "music"
87     },
88     {
```



```

89         "track_name": "09_FrenchHorns.wav",
90         "track_vol": "far",
91         "track_pan": "any",
92         "track_locut": "500",
93         "track_hicut": "15000",
94         "track_eq": "none",
95         "track_comp": "none",
96         "track_group": "music"
97     },
98     {
99         "track_name": "10_LeadVox.wav",
100        "track_vol": "front",
101        "track_pan": "C",
102        "track_locut": "150",
103        "track_hicut": "18000",
104        "track_eq": "none",
105        "track_comp": "none",
106        "track_group": "vocal"
107    }
108 ]
109 }

```

C.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks

```

1 {
2     "project_name": "Angela Thomas Wade - Milk Cow Blues",
3     "project_path": "D:\\AngelaThomasWade_MilkCowBlues_Full\\",
4     "sample_rate": 44100,
5     "start_time": 59,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_Kick.wav",
10            "track_vol": "front",
11            "track_pan": "C",
12            "track_locut": "40",
13            "track_hicut": "4000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "drums"
17        },
18        {
19            "track_name": "02_Snare.wav",

```

```
20         "track_vol": "front",
21         "track_pan": "C",
22         "track_locut": "300",
23         "track_hicut": "10000",
24         "track_eq": "none",
25         "track_comp": "none",
26         "track_group": "drums"
27     },
28     {
29         "track_name": "03_Tom.wav",
30         "track_vol": "mid",
31         "track_pan": "any",
32         "track_locut": "80",
33         "track_hicut": "7000",
34         "track_eq": "none",
35         "track_comp": "none",
36         "track_group": "drums"
37     },
38     {
39         "track_name": "04_Overheads.wav",
40         "track_vol": "mid",
41         "track_pan": "mid",
42         "track_locut": "250",
43         "track_hicut": "18000",
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "drums"
47     },
48     {
49         "track_name": "05_Bass.wav",
50         "track_vol": "front",
51         "track_pan": "C",
52         "track_locut": "50",
53         "track_hicut": "6000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "music"
57     },
58     {
59         "track_name": "06_AcGtr.wav",
60         "track_vol": "mid",
61         "track_pan": "any",
62         "track_locut": "200",
63         "track_hicut": "8000",
64         "track_eq": "none",
```

```
65         "track_comp": "none",
66         "track_group": "music"
67     },
68     {
69         "track_name": "07_Dobro.wav",
70         "track_vol": "mid",
71         "track_pan": "right",
72         "track_locut": "200",
73         "track_hicut": "6000",
74         "track_eq": "none",
75         "track_comp": "none",
76         "track_group": "music"
77     },
78     {
79         "track_name": "08_ElecGtr.wav",
80         "track_vol": "mid",
81         "track_pan": "any",
82         "track_locut": "250",
83         "track_hicut": "8000",
84         "track_eq": "none",
85         "track_comp": "none",
86         "track_group": "music"
87     },
88     {
89         "track_name": "09_Fiddle.wav",
90         "track_vol": "front",
91         "track_pan": "any",
92         "track_locut": "220",
93         "track_hicut": "15000",
94         "track_eq": "none",
95         "track_comp": "none",
96         "track_group": "music"
97     },
98     {
99         "track_name": "10_Piano.wav",
100        "track_vol": "mid",
101        "track_pan": "left",
102        "track_locut": "100",
103        "track_hicut": "10000",
104        "track_eq": "none",
105        "track_comp": "none",
106        "track_group": "vocal"
107    },
108    {
109        "track_name": "11_LeadVox.wav",
```

```
110         "track_vol": "front",
111         "track_pan": "C",
112         "track_locut": "200",
113         "track_hicut": "20000",
114         "track_eq": "none",
115         "track_comp": "none",
116         "track_group": "vocal"
117     }
118 ]
119 }
```

C.6 Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks

```
1 {
2     "project_name": "Don Camillo Choir - Marsh Marigolds Song",
3     "project_path": "D:\\DonCamilloChoir_MarshMarigoldsSong_Full\\",
4     "sample_rate": 44100,
5     "start_time": 35,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_MainPair1.wav",
10            "track_vol": "far",
11            "track_pan": "mid",
12            "track_locut": "80",
13            "track_hicut": "10000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "music"
17        },
18        {
19            "track_name": "02_MainPair2.wav",
20            "track_vol": "mid",
21            "track_pan": "mid",
22            "track_locut": "80",
23            "track_hicut": "17000",
24            "track_eq": "none",
25            "track_comp": "none",
26            "track_group": "music"
27        },
28        {
29            "track_name": "03_SpotMic1_Tenors.wav",
30            "track_vol": "far",
```

```

31         "track_pan": "mleft",
32         "track_locut": "100",
33         "track_hicut": "8000",
34         "track_eq": "none",
35         "track_comp": "none",
36         "track_group": "music"
37     },
38     {
39         "track_name": "04_SpotMic2_Basses.wav",
40         "track_vol": "far",
41         "track_pan": "eleft",
42         "track_locut": "60",
43         "track_hicut": "5000",
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "music"
47     },
48     {
49         "track_name": "05_SpotMic3_Sopranos.wav",
50         "track_vol": "far",
51         "track_pan": "eright",
52         "track_locut": "200",
53         "track_hicut": "15000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "music"
57     },
58     {
59         "track_name": "06_SpotMic4_Altos.wav",
60         "track_vol": "far",
61         "track_pan": "mright",
62         "track_locut": "150",
63         "track_hicut": "15000",
64         "track_eq": "none",
65         "track_comp": "none",
66         "track_group": "music"
67     },
68     {
69         "track_name": "07_BassCloseMic1.wav",
70         "track_vol": "mid",
71         "track_pan": "any",
72         "track_locut": "80",
73         "track_hicut": "10000",
74         "track_eq": "none",
75         "track_comp": "none",

```

```
76         "track_group": "music"
77     },
78     {
79         "track_name": "08_BassCloseMic2.wav",
80         "track_vol": "mid",
81         "track_pan": "any",
82         "track_locut": "80",
83         "track_hicut": "10000",
84         "track_eq": "none",
85         "track_comp": "none",
86         "track_group": "music"
87     },
88     {
89         "track_name": "09_BassCloseMic3.wav",
90         "track_vol": "mid",
91         "track_pan": "any",
92         "track_locut": "80",
93         "track_hicut": "10000",
94         "track_eq": "none",
95         "track_comp": "none",
96         "track_group": "music"
97     },
98     {
99         "track_name": "10_BassCloseMic4.wav",
100        "track_vol": "mid",
101        "track_pan": "any",
102        "track_locut": "80",
103        "track_hicut": "10000",
104        "track_eq": "none",
105        "track_comp": "none",
106        "track_group": "music"
107    },
108    {
109        "track_name": "11_BassCloseMic5.wav",
110        "track_vol": "mid",
111        "track_pan": "any",
112        "track_locut": "80",
113        "track_hicut": "10000",
114        "track_eq": "none",
115        "track_comp": "none",
116        "track_group": "music"
117    },
118    {
119        "track_name": "12_LeadVox.wav",
120        "track_vol": "mid",
```

```
121         "track_pan": "mid",
122         "track_locut": "200",
123         "track_hicut": "20000",
124         "track_eq": "none",
125         "track_comp": "none",
126         "track_group": "vocal"
127     }
128 ]
129 }
```

C.7 Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks

```
1 {
2     "project_name": "Araujo - The Saga Of Harrison Crabfeathers",
3     "project_path": "D:\\Araujo_TheSagaOfHarrisonCrabfeathers_Full\\",
4     "sample_rate": 44100,
5     "start_time": 139,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_KickIn.wav",
10            "track_vol": "front",
11            "track_pan": "C",
12            "track_locut": "40",
13            "track_hicut": "4000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "drums"
17        },
18        {
19            "track_name": "02_KickOut.wav",
20            "track_vol": "front",
21            "track_pan": "C",
22            "track_locut": "40",
23            "track_hicut": "6000",
24            "track_eq": "none",
25            "track_comp": "none",
26            "track_group": "drums"
27        },
28        {
29            "track_name": "03_Snare.wav",
30            "track_vol": "front",
31            "track_pan": "C",
```

```
32         "track_locut": "200",
33         "track_hicut": "7000",
34         "track_eq": "none",
35         "track_comp": "none",
36         "track_group": "drums"
37     },
38     {
39         "track_name": "04_Tom1.wav",
40         "track_vol": "mid",
41         "track_pan": "mright",
42         "track_locut": "100",
43         "track_hicut": "10000",
44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "drums"
47     },
48     {
49         "track_name": "05_Tom2.wav",
50         "track_vol": "mid",
51         "track_pan": "eright",
52         "track_locut": "50",
53         "track_hicut": "6000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "drums"
57     },
58     {
59         "track_name": "06_Overhead1.wav",
60         "track_vol": "mid",
61         "track_pan": "eleft",
62         "track_locut": "400",
63         "track_hicut": "12000",
64         "track_eq": "none",
65         "track_comp": "none",
66         "track_group": "drums"
67     },
68     {
69         "track_name": "07_Overhead2.wav",
70         "track_vol": "mid",
71         "track_pan": "eright",
72         "track_locut": "400",
73         "track_hicut": "12000",
74         "track_eq": "none",
75         "track_comp": "none",
76         "track_group": "drums"
```



```
77     },
78     {
79         "track_name": "08_BassDI.wav",
80         "track_vol": "front",
81         "track_pan": "C",
82         "track_locut": "50",
83         "track_hicut": "8000",
84         "track_eq": "none",
85         "track_comp": "none",
86         "track_group": "music"
87     },
88     {
89         "track_name": "10_ElecGtrMic1.wav",
90         "track_vol": "mid",
91         "track_pan": "any",
92         "track_locut": "250",
93         "track_hicut": "8000",
94         "track_eq": "none",
95         "track_comp": "none",
96         "track_group": "music"
97     },
98     {
99         "track_name": "11_ElecGtrMic2.wav",
100        "track_vol": "mid",
101        "track_pan": "any",
102        "track_locut": "250",
103        "track_hicut": "8000",
104        "track_eq": "none",
105        "track_comp": "none",
106        "track_group": "music"
107    },
108    {
109        "track_name": "13_Piano.wav",
110        "track_vol": "mid",
111        "track_pan": "any",
112        "track_locut": "100",
113        "track_hicut": "10000",
114        "track_eq": "none",
115        "track_comp": "none",
116        "track_group": "music"
117    }
118 ]
119 }
```

C.8 Ballad : MR 1001 - Jay Menon : 16 tracks

```
1 {
2     "project_name": "MR 1001 - Jay Menon",
3     "project_path": "D:\\MR1001_JayMenon\\",
4     "sample_rate": 44100,
5     "start_time": 0,
6     "duration": 30,
7     "tracks": [
8         {
9             "track_name": "01_Kick.wav",
10            "track_vol": "front",
11            "track_pan": "C",
12            "track_locut": "90",
13            "track_hicut": "7000",
14            "track_eq": "none",
15            "track_comp": "none",
16            "track_group": "drums"
17        },
18        {
19            "track_name": "02_Snare.wav",
20            "track_vol": "mid",
21            "track_pan": "C",
22            "track_locut": "200",
23            "track_hicut": "9000",
24            "track_eq": "none",
25            "track_comp": "none",
26            "track_group": "drums"
27        },
28        {
29            "track_name": "03_HiHat.wav",
30            "track_vol": "mid",
31            "track_pan": "cleft",
32            "track_locut": "500",
33            "track_hicut": "20000",
34            "track_eq": "none",
35            "track_comp": "none",
36            "track_group": "drums"
37        },
38        {
39            "track_name": "04_Overheads.wav",
40            "track_vol": "mid",
41            "track_pan": "mid",
42            "track_locut": "400",
43            "track_hicut": "20000",
```

```

44         "track_eq": "none",
45         "track_comp": "none",
46         "track_group": "drums"
47     },
48     {
49         "track_name": "05_CymbalRoll1.wav",
50         "track_vol": "far",
51         "track_pan": "mleft",
52         "track_locut": "200",
53         "track_hicut": "7000",
54         "track_eq": "none",
55         "track_comp": "none",
56         "track_group": "drums"
57     },
58     {
59         "track_name": "06_CymbalRoll2.wav",
60         "track_vol": "far",
61         "track_pan": "mright",
62         "track_locut": "200",
63         "track_hicut": "7000",
64         "track_eq": "none",
65         "track_comp": "none",
66         "track_group": "drums"
67     },
68     {
69         "track_name": "07_Bass.wav",
70         "track_vol": "front",
71         "track_pan": "C",
72         "track_locut": "40",
73         "track_hicut": "6000",
74         "track_eq": "none",
75         "track_comp": "none",
76         "track_group": "music"
77     },
78     {
79         "track_name": "08_AcGuitar1.wav",
80         "track_vol": "mid",
81         "track_pan": "mid",
82         "track_locut": "80",
83         "track_hicut": "2000",
84         "track_eq": "none",
85         "track_comp": "none",
86         "track_group": "music"
87     },
88     {

```

```
89         "track_name": "09_AcGuitar2.wav",
90         "track_vol": "mid",
91         "track_pan": "eleft",
92         "track_locut": "500",
93         "track_hicut": "12000",
94         "track_eq": "none",
95         "track_comp": "none",
96         "track_group": "music"
97     },
98     {
99         "track_name": "10_AcGuitar3.wav",
100        "track_vol": "mid",
101        "track_pan": "eright",
102        "track_locut": "500",
103        "track_hicut": "12000",
104        "track_eq": "none",
105        "track_comp": "none",
106        "track_group": "music"
107    },
108    {
109        "track_name": "11_AcGuitar4.wav",
110        "track_vol": "mid",
111        "track_pan": "mid",
112        "track_locut": "500",
113        "track_hicut": "12000",
114        "track_eq": "none",
115        "track_comp": "none",
116        "track_group": "music"
117    },
118    {
119        "track_name": "12_AcGuitar5.wav",
120        "track_vol": "mid",
121        "track_pan": "any",
122        "track_locut": "200",
123        "track_hicut": "7000",
124        "track_eq": "none",
125        "track_comp": "none",
126        "track_group": "music"
127    },
128    {
129        "track_name": "13_Piano.wav",
130        "track_vol": "mid",
131        "track_pan": "any",
132        "track_locut": "300",
133        "track_hicut": "10000",
```

```
134         "track_eq": "none",
135         "track_comp": "none",
136         "track_group": "music"
137     },
138     {
139         "track_name": "14_Pad1.wav",
140         "track_vol": "far",
141         "track_pan": "any",
142         "track_locut": "200",
143         "track_hicut": "8000",
144         "track_eq": "none",
145         "track_comp": "none",
146         "track_group": "music"
147     },
148     {
149         "track_name": "15_Pad2.wav",
150         "track_vol": "far",
151         "track_pan": "any",
152         "track_locut": "200",
153         "track_hicut": "8000",
154         "track_eq": "none",
155         "track_comp": "none",
156         "track_group": "music"
157     },
158     {
159         "track_name": "16_LeadVox.wav",
160         "track_vol": "front",
161         "track_pan": "C",
162         "track_locut": "100",
163         "track_hicut": "20000",
164         "track_eq": "none",
165         "track_comp": "none",
166         "track_group": "vocal"
167     }
168 ]
169 }
```


Annexe D

Les styles

D.1 Classic : Rod Alexander - Canon : 2 tracks

```
1 {
2     "drums": [
3
4     ],
5
6
7     "vocal": [
8
9     ],
10
11    "music": [
12
13    ],
14
15    "master": [
16
17    ],
18
19    "sends": [
20        {
21            "type": "vst3",
22            "name": "room",
23            "parameters": {
24                "path" : "C:\\Program Files\\Common Files\\
25                    VST3\\TC Electronic\\VSS3.vst3",
26                "program": "Clear Room"
27            }
28        },
29        {
30            "type": "vst3",
```

```
30         "name": "plate",
31         "parameters": {
32             "path" : "C:\\Program Files\\Common Files\\
33                 VST3\\TC Electronic\\VSS3.vst3",
34             "program": "Bright Plate"
35         },
36         {
37             "type": "vst3",
38             "name": "hall",
39             "parameters": {
40                 "path" : "C:\\Program Files\\Common Files\\
41                 VST3\\TC Electronic\\VSS3.vst3",
42                 "program": "Large Hall Clear"
43             }
44         }
45     ]
}
```

D.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks

```
1 {
2     "drums": [
3
4     ],
5
6     "vocal": [
7         {
8             "type": "vst3",
9             "name": "buscomp",
10            "parameters": {
11                "path" : "C:/Program Files/Common Files/VST3
12                    /Plugin Alliance/Brainworx/bx_townhouse
13                    Buss Compressor.vst3",
14                "program": null,
15                "thresh_db": "-3.0",
16                "mix": "33.3"
17            }
18        }
19    ],
20
21    "music": [
22
23    ],
24
25 }
```



```

22
23     "master": [
24
25 ],
26
27     "sends": [
28         {
29             "type": "vst3",
30             "name": "room",
31             "parameters": {
32                 "path" : "C:\\Program Files\\Common Files\\
33                     VST3\\TC Electronic\\VSS3.vst3",
34                 "program": "Clear Room"
35             }
36         },
37         {
38             "type": "vst3",
39             "name": "plate",
40             "parameters": {
41                 "path" : "C:\\Program Files\\Common Files\\
42                     VST3\\TC Electronic\\VSS3.vst3",
43                 "program": "Bright Plate"
44             }
45         },
46         {
47             "type": "vst3",
48             "name": "hall",
49             "parameters": {
50                 "path" : "C:\\Program Files\\Common Files\\
51                     VST3\\TC Electronic\\VSS3.vst3",
52                 "program": "Large Hall Clear"
53             }
54         }
55     ]
56 }

```

D.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks

```

1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {

```

```
7         "path" : "C:/Program Files/Common Files/VST3
            /Plugin Alliance/Brainworx/bx_townhouse
            Buss Compressor.vst3",
8         "program": null,
9         "thresh_db": "-6.0",
10        "mix": "20"
11    }
12    }
13    ],
14    "vocal": [
15        {
16            "type": "vst3",
17            "name": "vocal_bus",
18            "parameters": {
19                "path" : "C:/Program Files/Common Files/VST3
20                    /Plugin Alliance/Brainworx/bx_townhouse
21                    Buss Compressor.vst3",
22                "program": null,
23                "thresh_db": "-3.0",
24                "mix": "33.3"
25            }
26        },
27    ],
28    "music": [
29    ],
30    "master": [
31    ],
32    "sends": [
33        {
34            "type": "vst3",
35            "name": "room",
36            "parameters": {
37                "path" : "C:\\Program Files\\Common Files\\
38                    VST3\\TC Electronic\\VSS3.vst3",
39                "program": "Clear Room"
40            }
41        },
42    ],
43    {
44        "type": "vst3",
```

```

47         "name": "plate",
48         "parameters": {
49             "path" : "C:\\Program Files\\Common Files\\
                    VST3\\TC Electronic\\VSS3.vst3",
50             "program": "Bright Plate"
51         }
52     },
53     {
54         "type": "vst3",
55         "name": "hall",
56         "parameters": {
57             "path" : "C:\\Program Files\\Common Files\\
                    VST3\\TC Electronic\\VSS3.vst3",
58             "program": "Large Hall Clear"
59         }
60     }
61 ]
62 }

```

D.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks

```

1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {
7                 "path" : "C:/Program Files/Common Files/VST3
                        /Plugin Alliance/Brainworx/bx_townhouse
                        Buss Compressor.vst3",
8                 "program": null,
9                 "thresh_db": "-6.0",
10                "mix": "20"
11            }
12        }
13    ],
14
15    "vocal": [
16        {
17            "type": "vst3",
18            "name": "vocal_bus",
19            "parameters": {
20

```

```
21         "path" : "C:/Program Files/Common Files/VST3
                /Plugin Alliance/Brainworx/bx_townhouse
                Buss Compressor.vst3",
22         "program": null,
23         "thresh_db": "-3.0",
24         "mix": "20"
25     }
26 }
27 ],
28
29 "music": [
30
31 ],
32
33 "master": [
34
35 ],
36
37 "sends": [
38     {
39         "type": "vst3",
40         "name": "room",
41         "parameters": {
42             "path" : "C:\\Program Files\\Common Files\\
                    VST3\\TC Electronic\\VSS3.vst3",
43             "program": "Clear Room"
44         }
45     },
46     {
47         "type": "vst3",
48         "name": "plate",
49         "parameters": {
50             "path" : "C:\\Program Files\\Common Files\\
                    VST3\\TC Electronic\\VSS3.vst3",
51             "program": "Bright Plate"
52         }
53     },
54     {
55         "type": "vst3",
56         "name": "hall",
57         "parameters": {
58             "path" : "C:\\Program Files\\Common Files\\
                    VST3\\TC Electronic\\VSS3.vst3",
59             "program": "Large Hall Clear"
60         }
61     }
62 ]
```

```

61         }
62     ]
63 }

```

D.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks

```

1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {
7                 "path" : "C:/Program Files/Common Files/VST3
8                     /Plugin Alliance/Brainworx/bx_townhouse
9                     Buss Compressor.vst3",
10                "program": null,
11                "thresh_db": "-6.0",
12                "mix": "20"
13            }
14        },
15        "vocal": [
16            {
17                "type": "vst3",
18                "name": "vocal_bus",
19                "parameters": {
20                    "path" : "C:/Program Files/Common Files/VST3
21                        /Plugin Alliance/Brainworx/bx_townhouse
22                        Buss Compressor.vst3",
23                    "program": null,
24                    "thresh_db": "-3.0",
25                    "mix": "33.3"
26                }
27            },
28            "music": [
29
30        ],
31        "master": [
32
33    ]

```

```
34     ],
35
36     "sends": [
37         {
38             "type": "vst3",
39             "name": "room",
40             "parameters": {
41                 "path" : "C:\\Program Files\\Common Files\\
42                     VST3\\TC Electronic\\VSS3.vst3",
43                 "program": "Clear Room"
44             }
45         },
46         {
47             "type": "vst3",
48             "name": "plate",
49             "parameters": {
50                 "path" : "C:\\Program Files\\Common Files\\
51                     VST3\\TC Electronic\\VSS3.vst3",
52                 "program": "Bright Plate"
53             }
54         },
55         {
56             "type": "vst3",
57             "name": "hall",
58             "parameters": {
59                 "path" : "C:\\Program Files\\Common Files\\
60                     VST3\\TC Electronic\\VSS3.vst3",
61                 "program": "Large Hall Clear"
62             }
63         }
64     ]
65 }
```

D.6 Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks

```
1 {
2     "drums": [
3
4     ],
5
6     "vocal": [
7         {
8             "type": "vst3",
```

```

9         "name": "vocal_bus",
10        "parameters": {
11            "path" : "C:/Program Files/Common Files/VST3
12                /Plugin Alliance/Brainworx/bx_townhouse
13                Buss Compressor.vst3",
14            "program": null,
15            "thresh_db": "-6.0",
16            "mix": "100.0"
17        }
18    },
19    "music": [
20
21    ],
22
23    "master": [
24
25    ],
26
27    "sends": [
28        {
29            "type": "vst3",
30            "name": "room",
31            "parameters": {
32                "path" : "C:\\Program Files\\Common Files\\
33                VST3\\TC Electronic\\VSS3.vst3",
34                "program": "Clear Room"
35            }
36        },
37        {
38            "type": "vst3",
39            "name": "plate",
40            "parameters": {
41                "path" : "C:\\Program Files\\Common Files\\
42                VST3\\TC Electronic\\VSS3.vst3",
43                "program": "Bright Plate"
44            }
45        },
46        {
47            "type": "vst3",
48            "name": "hall",
49            "parameters": {
50                "path" : "C:\\Program Files\\Common Files\\
51                VST3\\TC Electronic\\VSS3.vst3",

```

```
49         "program": "Large Hall Clear"
50     }
51 }
52 ]
53 }
```

D.7 Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks

```
1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {
7                 "path" : "C:/Program Files/Common Files/VST3
8                     /Plugin Alliance/Brainworx/bx_townhouse
9                     Buss Compressor.vst3",
10                "program": null,
11                "thresh_db": "-3.0",
12                "mix": "50"
13            }
14        ],
15
16        "vocal": [
17
18        ],
19
20        "music": [
21
22        ],
23
24        "master": [
25            {
26                "type": "vst3",
27                "name": "master_bus",
28                "parameters": {
29                    "path" : "C:/Program Files/Common Files/VST3
30                        /Plugin Alliance/Brainworx/bx_townhouse
31                        Buss Compressor.vst3",
32                    "program": null,
33                    "thresh_db": "-3.0",
```



```

32         "mix": "15"
33     }
34 }
35 ],
36
37 "sends": [
38     {
39         "type": "vst3",
40         "name": "room",
41         "parameters": {
42             "path" : "C:\\Program Files\\Common Files\\
43                 VST3\\TC Electronic\\VSS3.vst3",
44             "program": "Clear Room"
45         }
46     },
47     {
48         "type": "vst3",
49         "name": "plate",
50         "parameters": {
51             "path" : "C:\\Program Files\\Common Files\\
52                 VST3\\TC Electronic\\VSS3.vst3",
53             "program": "Bright Plate"
54         }
55     },
56     {
57         "type": "vst3",
58         "name": "hall",
59         "parameters": {
60             "path" : "C:\\Program Files\\Common Files\\
61                 VST3\\TC Electronic\\VSS3.vst3",
62             "program": "Large Hall Clear"
63         }
64     }
65 ]
66 }

```

D.8 Ballad : MR 1001 - Jay Menon : 16 tracks

```

1 {
2     "drums": [
3         {
4             "type": "vst3",
5             "name": "drums_bus",
6             "parameters": {

```

```
7         "path" : "C:/Program Files/Common Files/VST3
           /Plugin Alliance/Brainworx/bx_townhouse
           Buss Compressor.vst3",
8         "program": null,
9         "thresh_db": "-6.0",
10        "mix": "20"
11    }
12 }
13 ],
14
15 "vocal": [
16     {
17         "type": "vst3",
18         "name": "vocal_bus",
19         "parameters": {
20             "path" : "C:/Program Files/Common Files/VST3
                       /Plugin Alliance/Brainworx/bx_townhouse
                       Buss Compressor.vst3",
21             "program": null,
22             "thresh_db": "-3.0",
23             "mix": "33.3"
24         }
25     }
26 ],
27
28 "music": [
29
30 ],
31
32 "master": [
33
34 ],
35
36 "sends": [
37     {
38         "type": "vst3",
39         "name": "room",
40         "parameters": {
41             "path" : "C:\\Program Files\\Common Files\\
                       VST3\\TC Electronic\\VSS3.vst3",
42             "program": "Clear Room"
43         }
44     },
45     {
46         "type": "vst3",
```

```
47         "name": "plate",
48         "parameters": {
49             "path" : "C:\\Program Files\\Common Files\\
50                 VST3\\TC Electronic\\VSS3.vst3",
51             "program": "Bright Plate"
52         },
53     {
54         "type": "vst3",
55         "name": "hall",
56         "parameters": {
57             "path" : "C:\\Program Files\\Common Files\\
58                 VST3\\TC Electronic\\VSS3.vst3",
59             "program": "Large Hall Clear"
60         }
61     ]
62 }
```


Annexe E

Les règles de mixage

Dans cette annexe, nous avons compilé toutes les règles de mixage déterminées par l'algorithme d'optimisation de notre système de pré-mixage automatique. Nous avons classé les règles pour chacune des chansons à mixer séparément (les sections), ce qui permet d'analyser facilement les décisions analytiques de l'algorithme tout en écoutant un mixage final fait selon ces règles.

Le questionnaire des tests perceptifs peut être trouvé par le lien <https://forms.gle/npiHG1zNh8CTU1fM6> [Vital, 2021c], ainsi que l'accès aux fichiers audio originaux de ces mixages en format non compressé wav.

E.1 Classic : Rod Alexander - Canon : 2 tracks

E.1.1 Les règles de mixage

0th mix : fobj = 5.789

1: MixRules(vol=-8dB, pan=-45°, locut=31Hz, hicut=20000Hz)	01_MainPair
2: MixRules(vol=-20dB, pan=-23°, locut=23Hz, hicut=16640Hz)	02_DI

1th mix : fobj = 4.894

1: MixRules(vol=-7dB, pan=-61°, locut=32Hz, hicut=20000Hz)	01_MainPair
2: MixRules(vol=-27dB, pan=-22°, locut=25Hz, hicut=11674Hz)	02_DI

2th mix : fobj = 4.727

1: MixRules(vol=-11dB, pan=-30°, locut=63Hz, hicut=20000Hz)	01_MainPair
2: MixRules(vol=-25dB, pan=69°, locut=78Hz, hicut=14585Hz)	02_DI

3th mix : fobj = 4.838

1: MixRules(vol=-9dB, pan=59°, locut=65Hz, hicut=20000Hz)	01_MainPair
2: MixRules(vol=-19dB, pan=-81°, locut=61Hz, hicut=15811Hz)	02_DI

4th mix : fobj = 4.992

1: MixRules(vol=-9dB, pan=13°, locut=25Hz, hicut=20000Hz)	01_MainPair
2: MixRules(vol=-34dB, pan=0°, locut=63Hz, hicut=19333Hz)	02_DI

5th mix : fobj = 4.114

1: MixRules(vol=-8dB, pan=45°, locut=25Hz, hicut=20000Hz)

01_MainPair

2: MixRules(vol=-30dB, pan=0°, locut=63Hz, hicut=19870Hz)

02_DI

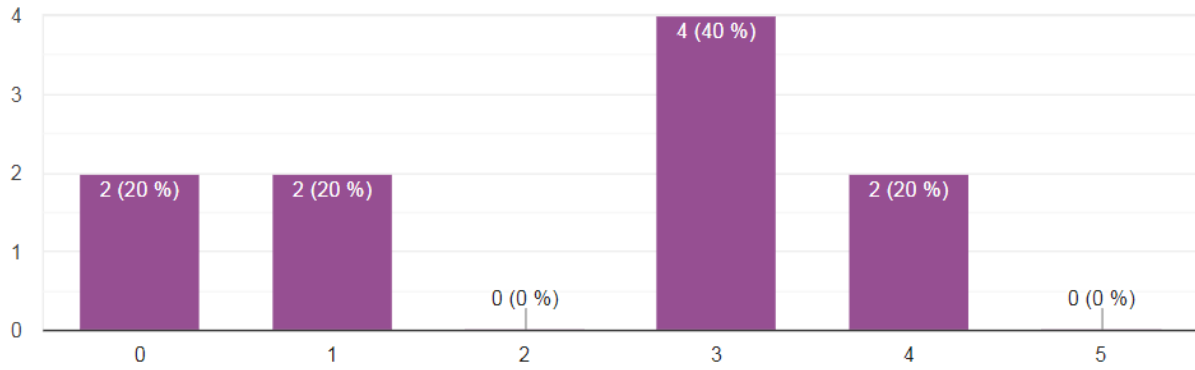


Figure E.1 – 2guitars : Le pré-mixage préféré.

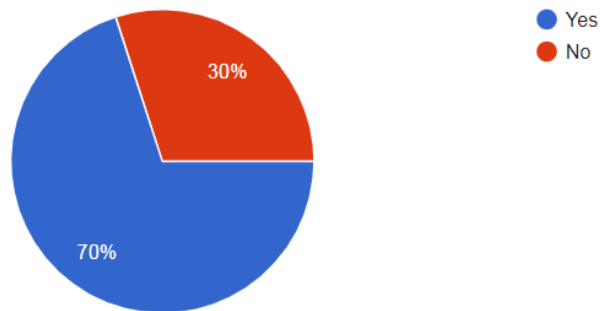


Figure E.2 – 2guitars : L'intention d'utiliser le pré-mixage préféré.

E.1.2 Les graphes de convergence

La fonction objectif

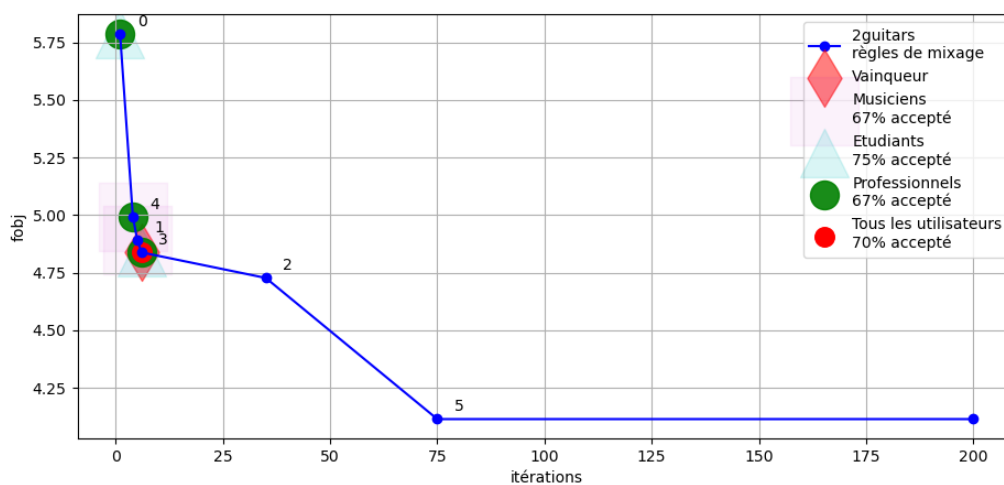


Figure E.3 – 2guitars : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

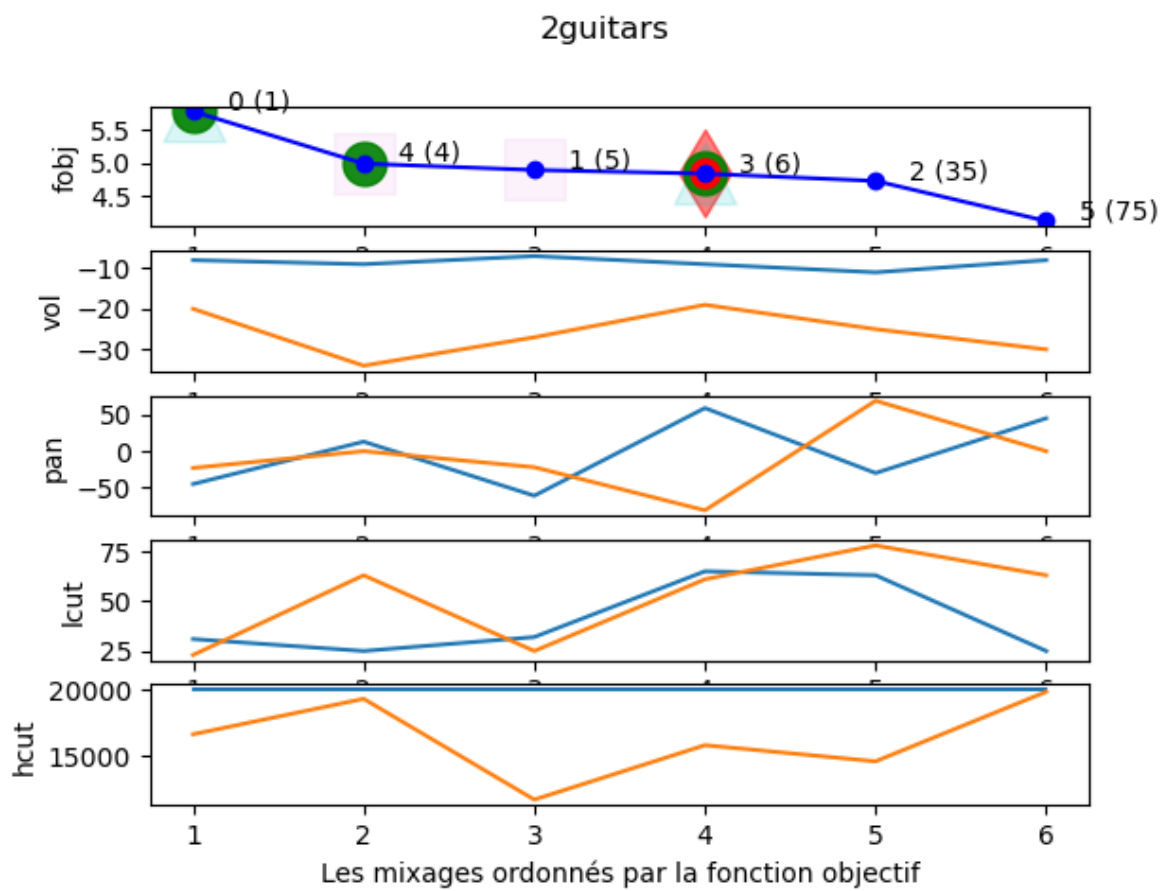


Figure E.4 – 2guitars : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

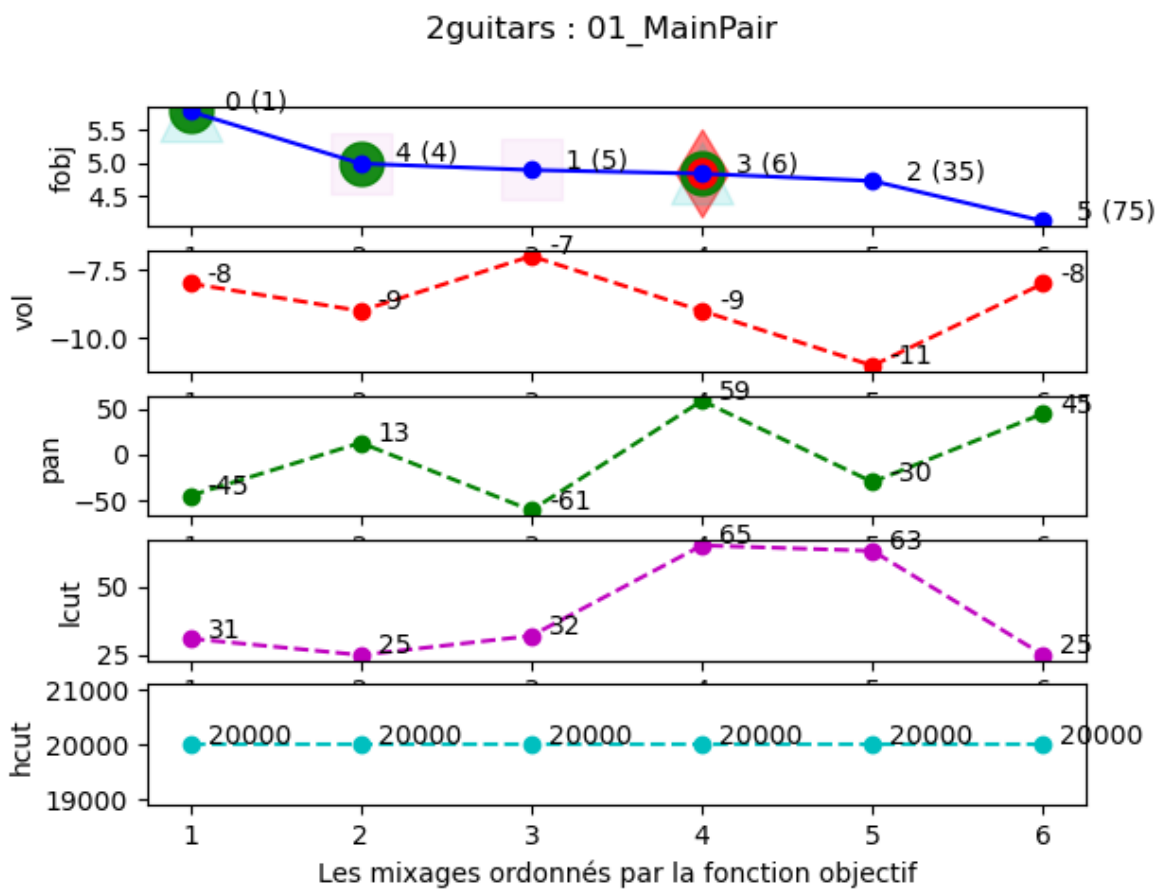


Figure E.5 – 2guitars : Graphe de convergence – 01_MainPair.

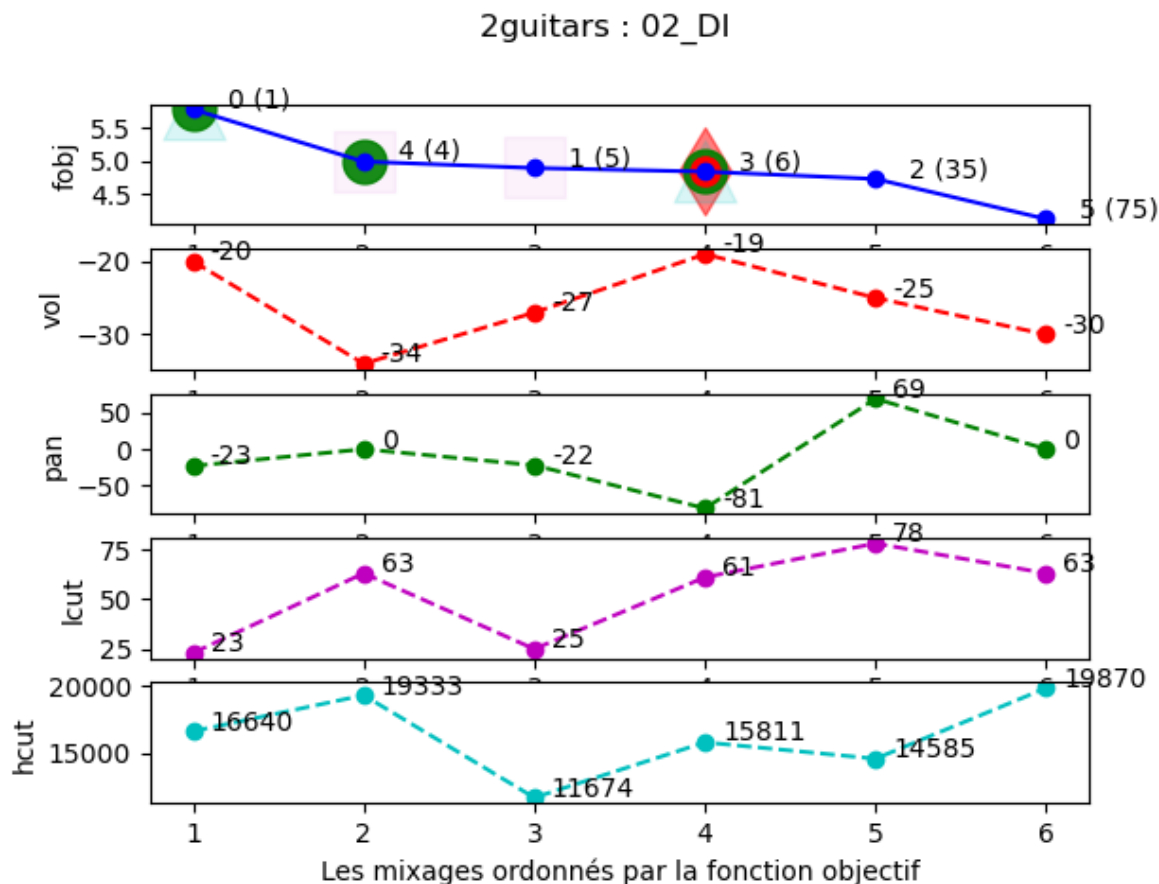


Figure E.6 – 2guitars : Graphe de convergence – 02_DI.

E.2 Folk : Leslie Mendelson - The Hardest Part 2 : 4 tracks

E.2.1 Les règles de mixage

0th mix : fobj = 5.093

- 1: MixRules(vol=-16dB, pan=-33°, locut=77Hz, hicut=20000Hz)
- 2: MixRules(vol=-14dB, pan=49°, locut=28Hz, hicut=20000Hz)
- 3: MixRules(vol=-9dB, pan=-80°, locut=54Hz, hicut=20000Hz)
- 4: MixRules(vol=-12dB, pan=63°, locut=20Hz, hicut=14073Hz)

Acoustic Guitar-Leslie-Ela M 260
 Acoustic Guitar-Steve-CU29
 Vocals-Leslie- Ela M 251
 Vocals-Steve-U47

1th mix : fobj = 5.192

- 1: MixRules(vol=-16dB, pan=-9°, locut=65Hz, hicut=20000Hz)
- 2: MixRules(vol=-12dB, pan=4°, locut=77Hz, hicut=20000Hz)
- 3: MixRules(vol=-6dB, pan=-78°, locut=31Hz, hicut=20000Hz)
- 4: MixRules(vol=-16dB, pan=82°, locut=52Hz, hicut=14965Hz)

Acoustic Guitar-Leslie-Ela M 260
 Acoustic Guitar-Steve-CU29
 Vocals-Leslie- Ela M 251
 Vocals-Steve-U47

2th mix : fobj = 4.197

- 1: MixRules(vol=-17dB, pan=-88°, locut=41Hz, hicut=20000Hz)

Acoustic Guitar-Leslie-Ela M 260

2: MixRules(vol=-13dB, pan=0°, locut=77Hz, hicut=20000Hz)
3: MixRules(vol=-8dB, pan=-7°, locut=94Hz, hicut=20000Hz)
4: MixRules(vol=-14dB, pan=84°, locut=61Hz, hicut=16526Hz)

Acoustic Guitar-Steve-CU29
Vocals-Leslie- Ela M 251
Vocals-Steve-U47

3th mix : fobj = 3.760

1: MixRules(vol=-17dB, pan=-88°, locut=41Hz, hicut=20000Hz)
2: MixRules(vol=-13dB, pan=0°, locut=77Hz, hicut=20000Hz)
3: MixRules(vol=-8dB, pan=-7°, locut=74Hz, hicut=20000Hz)
4: MixRules(vol=-14dB, pan=84°, locut=52Hz, hicut=13649Hz)

Acoustic Guitar-Leslie-Ela M 260
Acoustic Guitar-Steve-CU29
Vocals-Leslie- Ela M 251
Vocals-Steve-U47

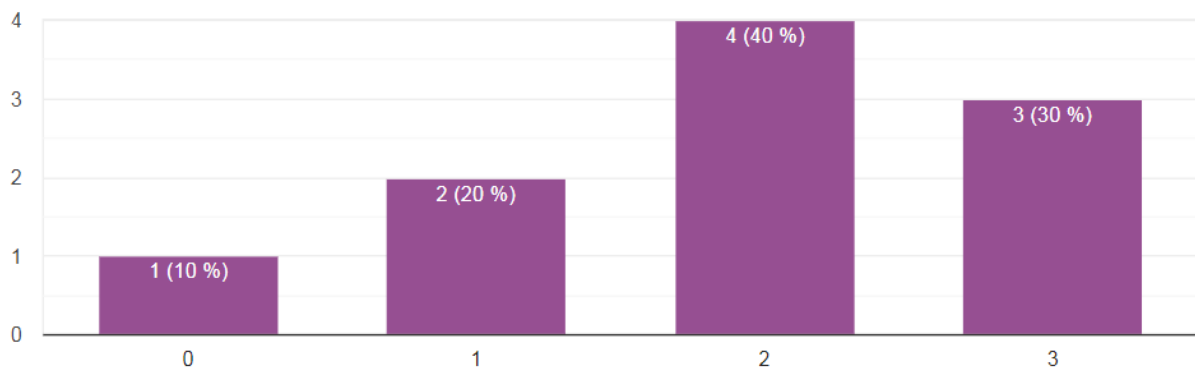


Figure E.7 – Folk : Le pré-mixage préféré.

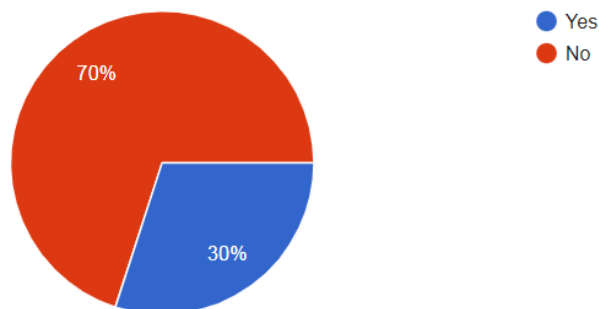


Figure E.8 – Folk : L'intention d'utiliser le pré-mixage préféré.

E.2.2 Les graphes de convergence

La fonction objectif

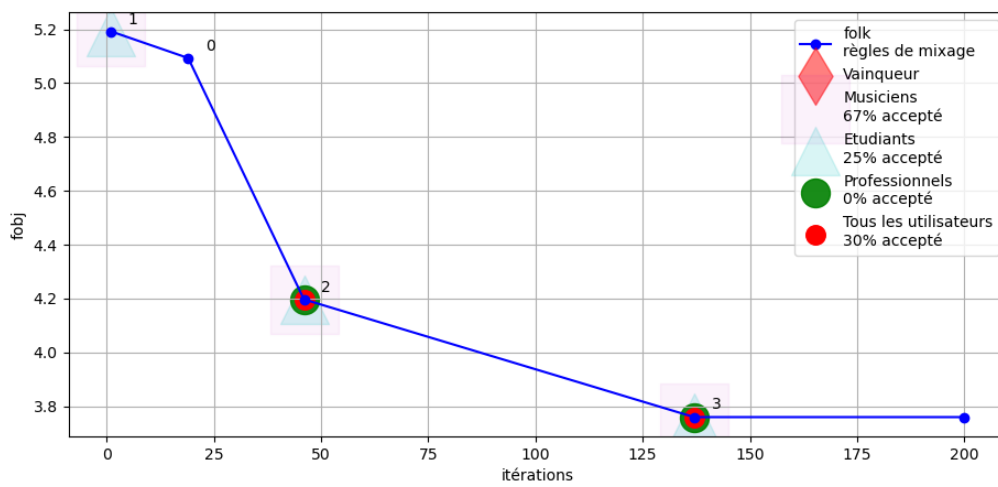


Figure E.9 – Folk : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

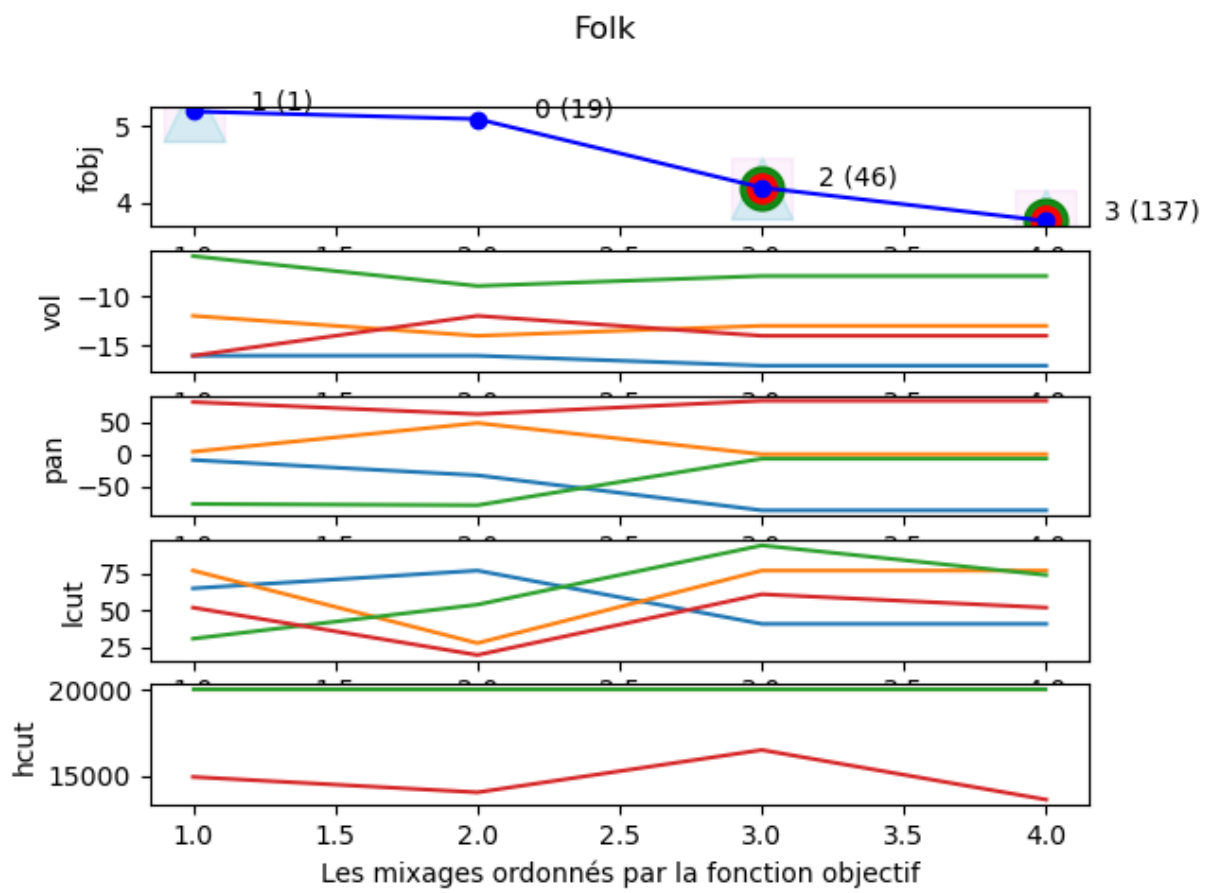


Figure E.10 – Folk : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

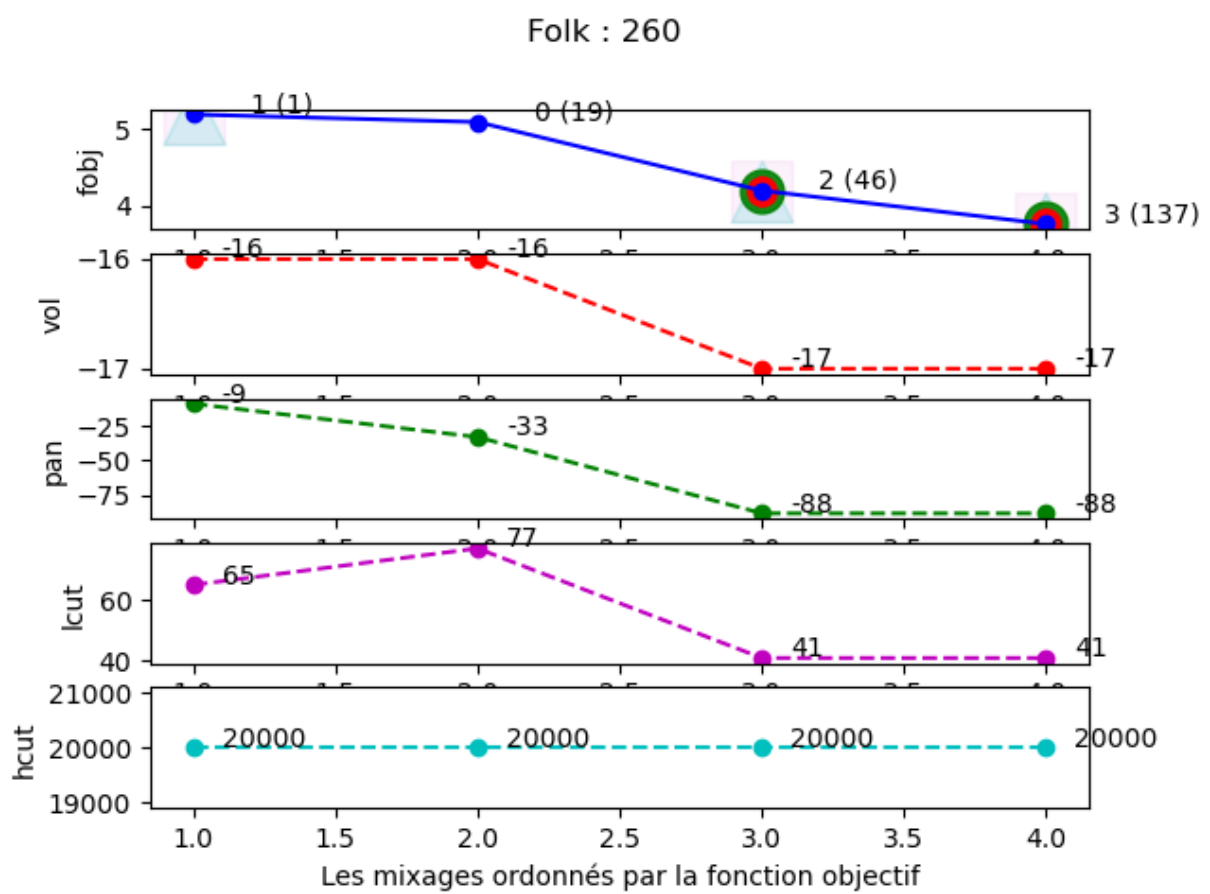


Figure E.11 – Folk : Graphe de convergence – 260.

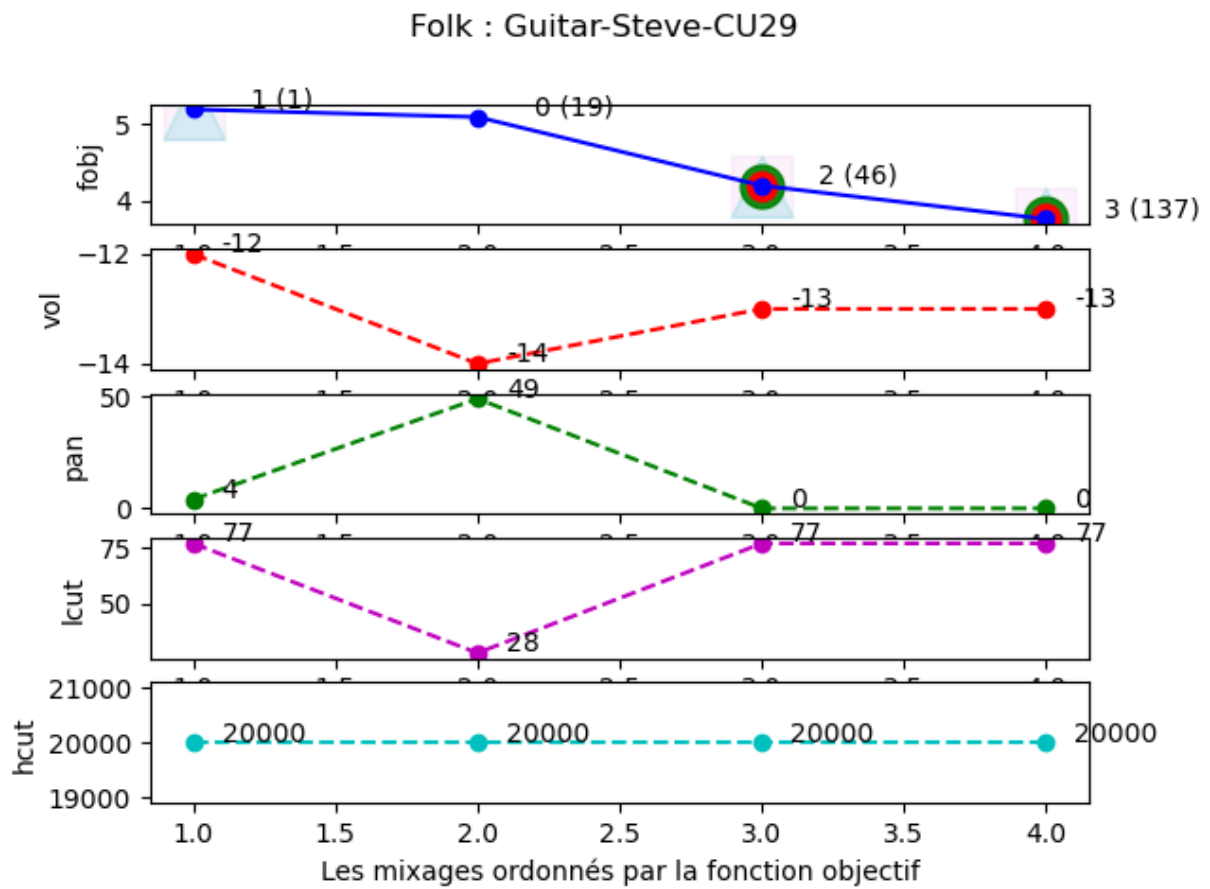


Figure E.12 – Folk : Graphe de convergence – *Guitar-Steve-CU29*.

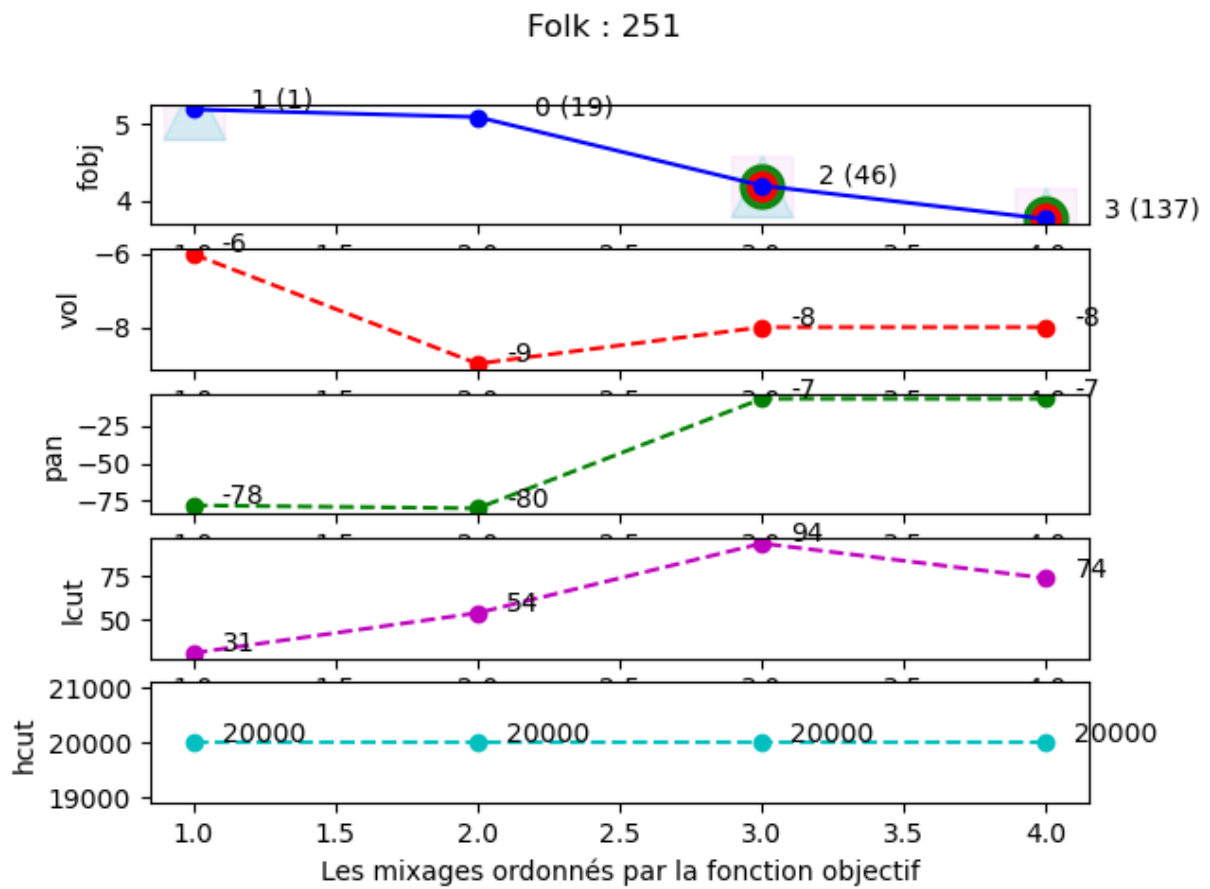


Figure E.13 – Folk : Graphe de convergence – 251.

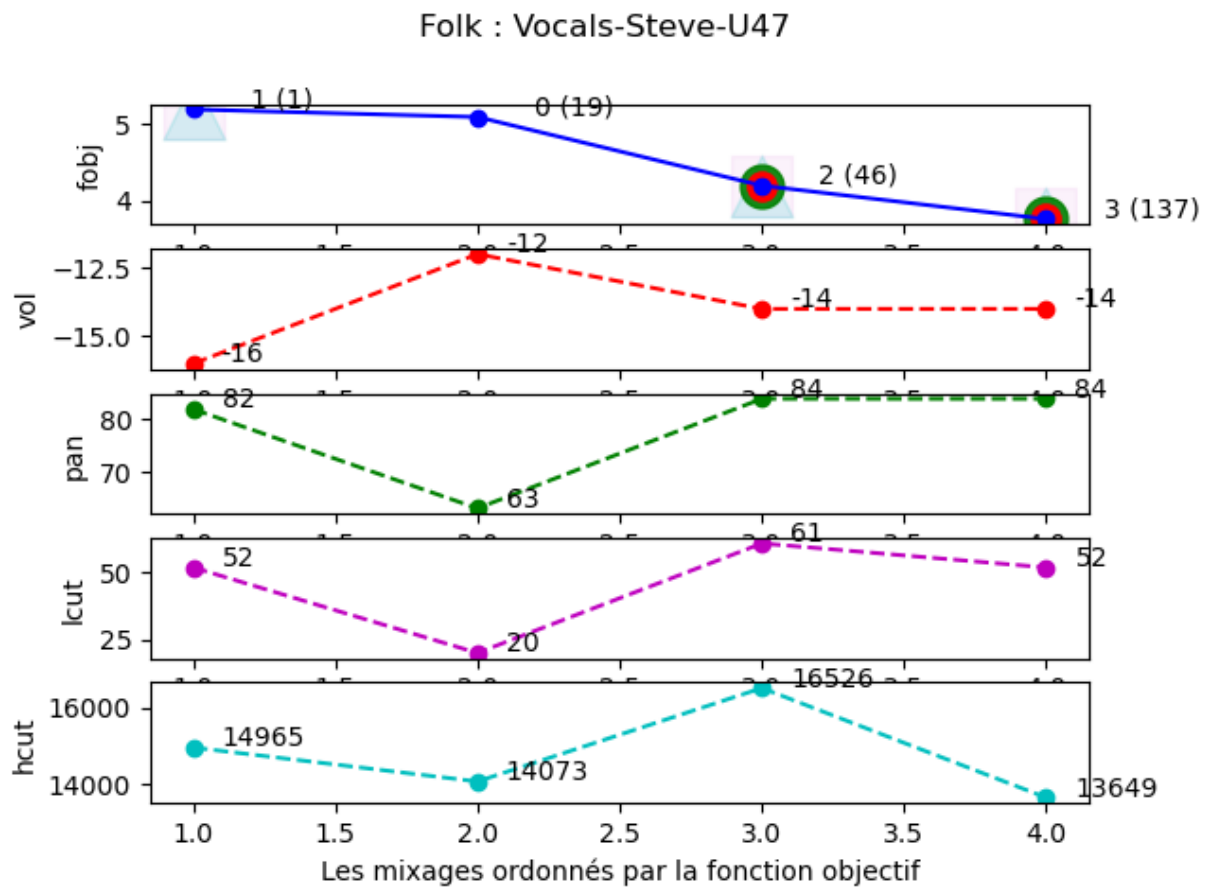


Figure E.14 – Folk : Graphe de convergence – *Vocals-Steve-U47*.

E.3 Ska : Big Stone Culture - Fragile Thoughts : 7 tracks

E.3.1 Les règles de mixage

0th mix : fobj = 10.061

1: MixRules(vol=-9dB, pan=0°, locut=67Hz, hicut=12262Hz)	01_Kick
2: MixRules(vol=-7dB, pan=0°, locut=182Hz, hicut=20000Hz)	02_Snare
3: MixRules(vol=-13dB, pan=0°, locut=98Hz, hicut=18051Hz)	03_Overhead
4: MixRules(vol=-6dB, pan=0°, locut=50Hz, hicut=6993Hz)	04_BassDI
5: MixRules(vol=-7dB, pan=0°, locut=220Hz, hicut=16747Hz)	05_Gtr
6: MixRules(vol=-15dB, pan=9°, locut=281Hz, hicut=17963Hz)	06_Percussion
7: MixRules(vol=-6dB, pan=0°, locut=138Hz, hicut=20000Hz)	07_LeadVox

1th mix : fobj = 9.757

1: MixRules(vol=-10dB, pan=0°, locut=40Hz, hicut=6009Hz)	01_Kick
2: MixRules(vol=-11dB, pan=0°, locut=180Hz, hicut=20000Hz)	02_Snare
3: MixRules(vol=-17dB, pan=0°, locut=396Hz, hicut=18122Hz)	03_Overhead
4: MixRules(vol=-8dB, pan=0°, locut=20Hz, hicut=17431Hz)	04_BassDI

5: MixRules(vol=-7dB, pan=0°, locut=255Hz, hicut=18513Hz) 05_Gtr
6: MixRules(vol=-12dB, pan=9°, locut=482Hz, hicut=10845Hz) 06_Percussion
7: MixRules(vol=-11dB, pan=0°, locut=181Hz, hicut=20000Hz) 07_LeadVox

2th mix : fobj = 9.644

1: MixRules(vol=-10dB, pan=0°, locut=40Hz, hicut=6009Hz) 01_Kick
2: MixRules(vol=-11dB, pan=0°, locut=84Hz, hicut=20000Hz) 02_Snare
3: MixRules(vol=-17dB, pan=0°, locut=396Hz, hicut=19622Hz) 03_Overhead
4: MixRules(vol=-8dB, pan=0°, locut=20Hz, hicut=17431Hz) 04_BassDI
5: MixRules(vol=-7dB, pan=0°, locut=255Hz, hicut=11150Hz) 05_Gtr
6: MixRules(vol=-15dB, pan=9°, locut=482Hz, hicut=13010Hz) 06_Percussion
7: MixRules(vol=-11dB, pan=0°, locut=181Hz, hicut=20000Hz) 07_LeadVox

3th mix : fobj = 9.808

1: MixRules(vol=-6dB, pan=0°, locut=45Hz, hicut=6001Hz) 01_Kick
2: MixRules(vol=-7dB, pan=0°, locut=145Hz, hicut=20000Hz) 02_Snare
3: MixRules(vol=-15dB, pan=0°, locut=314Hz, hicut=19238Hz) 03_Overhead
4: MixRules(vol=-11dB, pan=0°, locut=35Hz, hicut=17267Hz) 04_BassDI
5: MixRules(vol=-8dB, pan=0°, locut=201Hz, hicut=18208Hz) 05_Gtr
6: MixRules(vol=-16dB, pan=9°, locut=90Hz, hicut=13450Hz) 06_Percussion
7: MixRules(vol=-10dB, pan=0°, locut=189Hz, hicut=20000Hz) 07_LeadVox

4th mix : fobj = 9.823

1: MixRules(vol=-6dB, pan=0°, locut=55Hz, hicut=6001Hz) 01_Kick
2: MixRules(vol=-7dB, pan=0°, locut=131Hz, hicut=20000Hz) 02_Snare
3: MixRules(vol=-15dB, pan=0°, locut=317Hz, hicut=19238Hz) 03_Overhead
4: MixRules(vol=-9dB, pan=0°, locut=30Hz, hicut=17267Hz) 04_BassDI
5: MixRules(vol=-8dB, pan=0°, locut=160Hz, hicut=18208Hz) 05_Gtr
6: MixRules(vol=-15dB, pan=9°, locut=90Hz, hicut=12160Hz) 06_Percussion
7: MixRules(vol=-10dB, pan=0°, locut=189Hz, hicut=20000Hz) 07_LeadVox

5th mix : fobj = 9.858

1: MixRules(vol=-6dB, pan=0°, locut=42Hz, hicut=8566Hz) 01_Kick
2: MixRules(vol=-6dB, pan=0°, locut=66Hz, hicut=20000Hz) 02_Snare
3: MixRules(vol=-16dB, pan=0°, locut=365Hz, hicut=18353Hz) 03_Overhead
4: MixRules(vol=-11dB, pan=0°, locut=42Hz, hicut=17306Hz) 04_BassDI
5: MixRules(vol=-11dB, pan=0°, locut=149Hz, hicut=18762Hz) 05_Gtr
6: MixRules(vol=-16dB, pan=9°, locut=70Hz, hicut=14166Hz) 06_Percussion
7: MixRules(vol=-8dB, pan=0°, locut=112Hz, hicut=20000Hz) 07_LeadVox

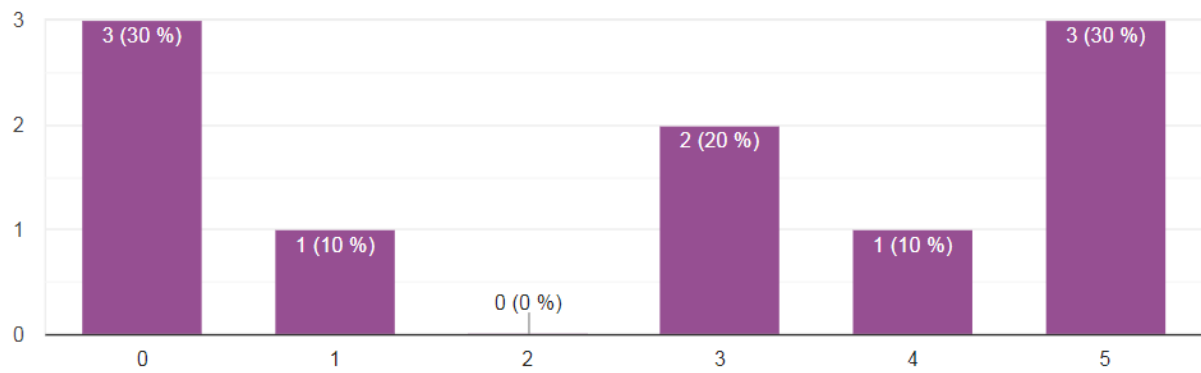


Figure E.15 – Ska : Le pré-mixage préféré.

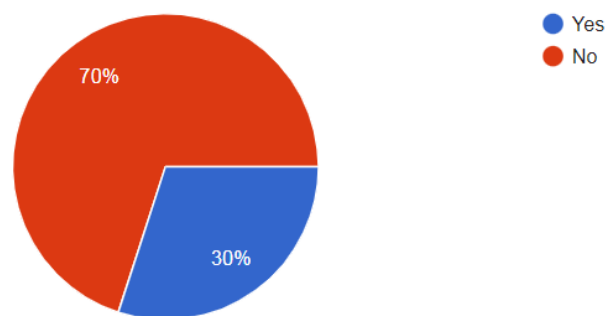


Figure E.16 – Ska : L'intention d'utiliser le pré-mixage préféré.

E.3.2 Les graphes de convergence

La fonction objectif

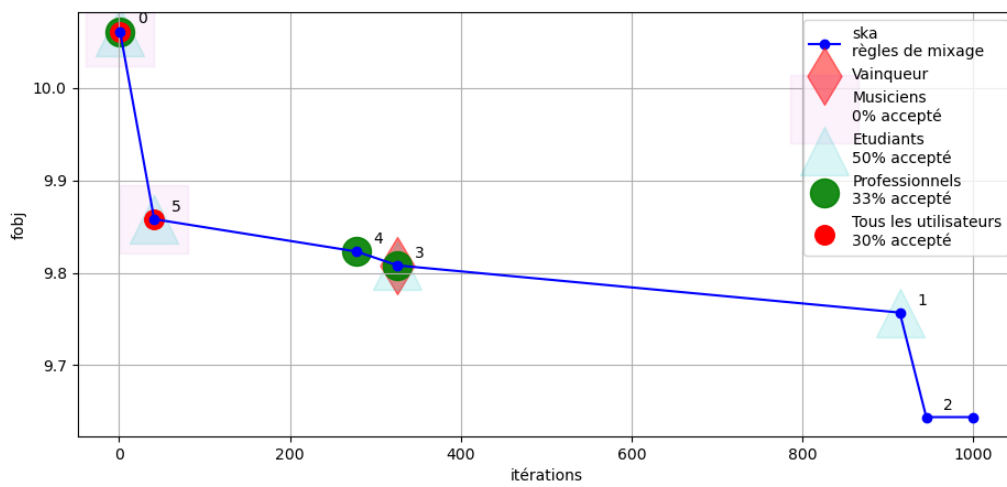


Figure E.17 – Ska : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

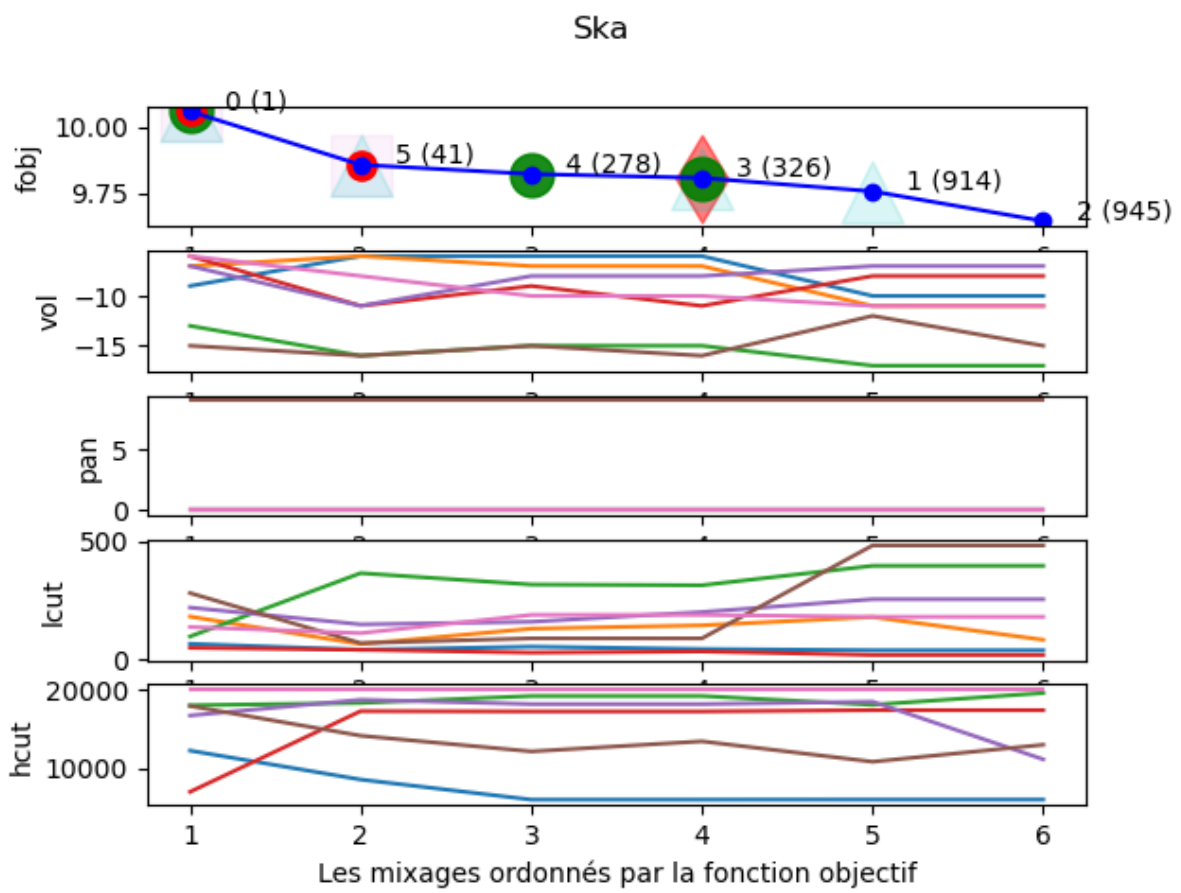


Figure E.18 – Ska : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

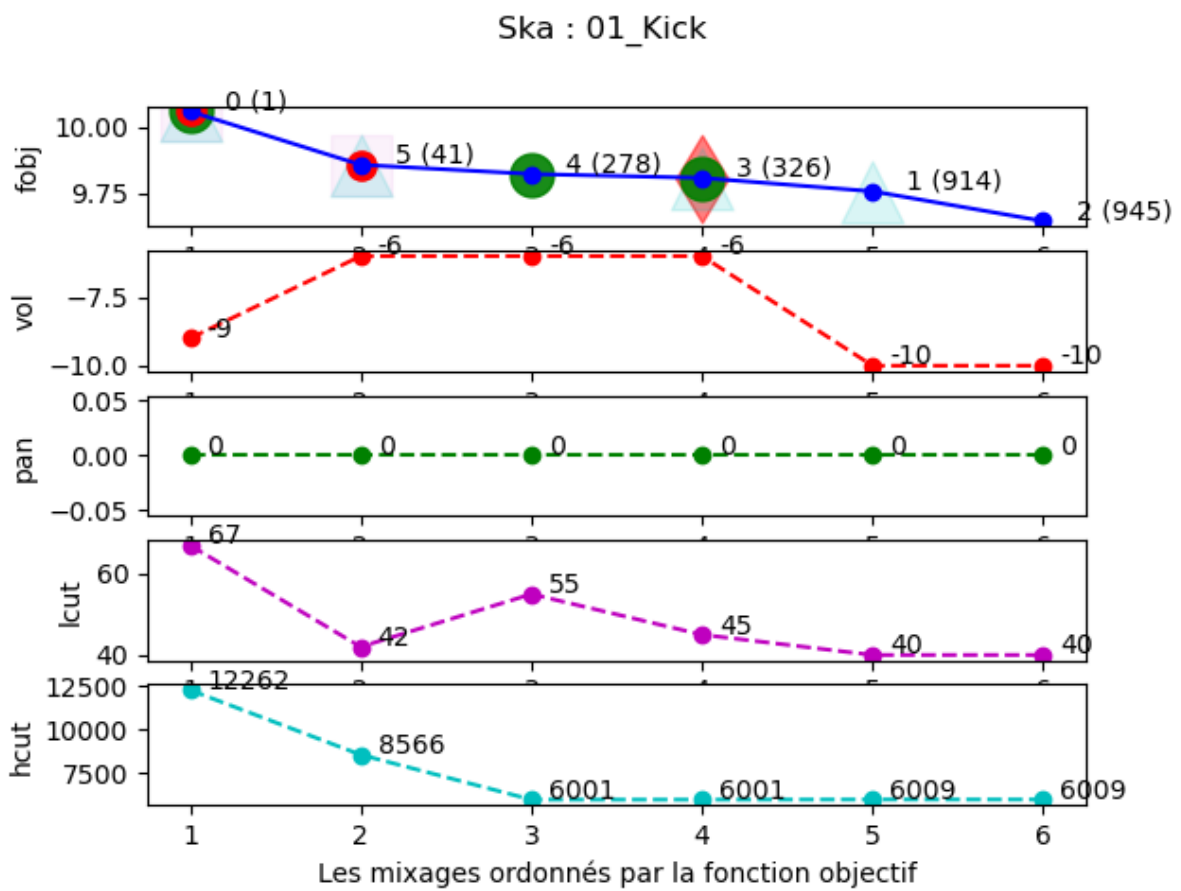


Figure E.19 – Ska : Graphe de convergence – 01_Kick.

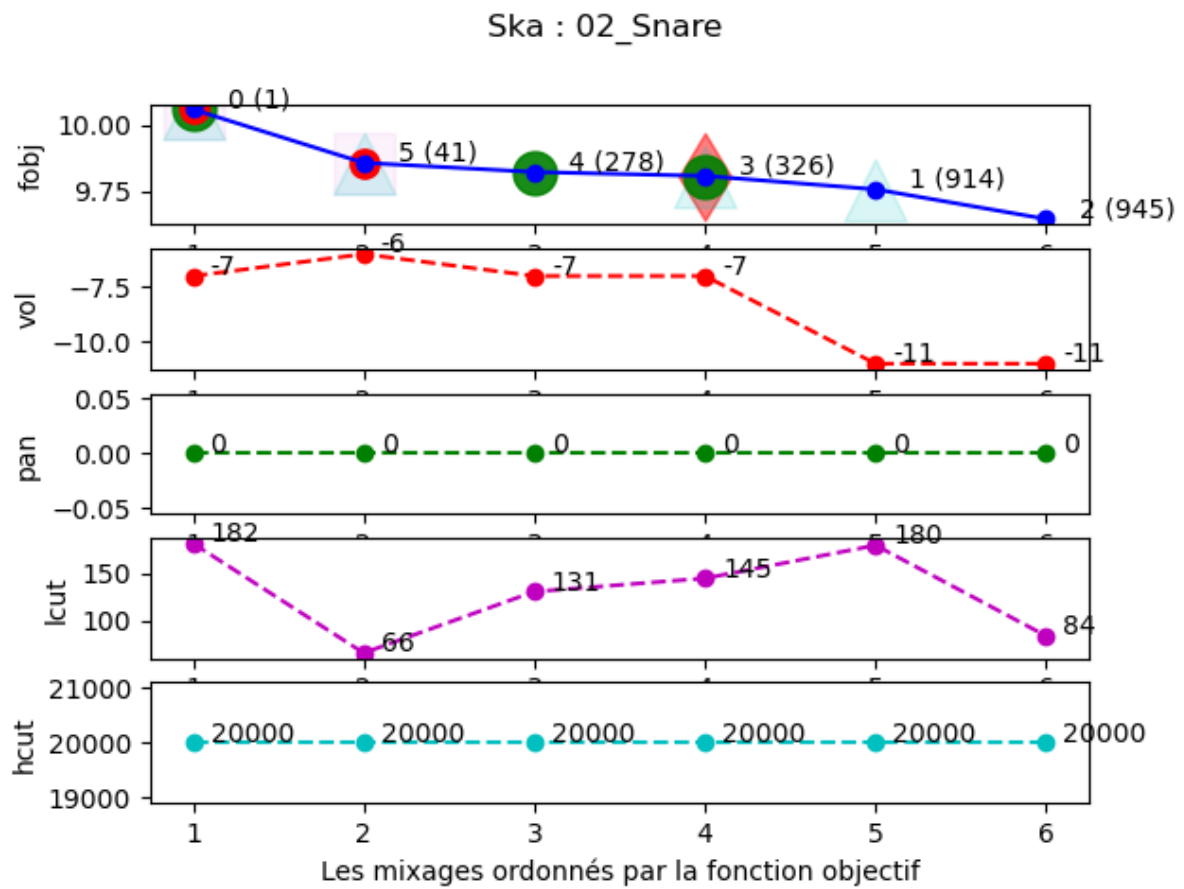


Figure E.20 – Ska : Graphe de convergence – 02_Snare.

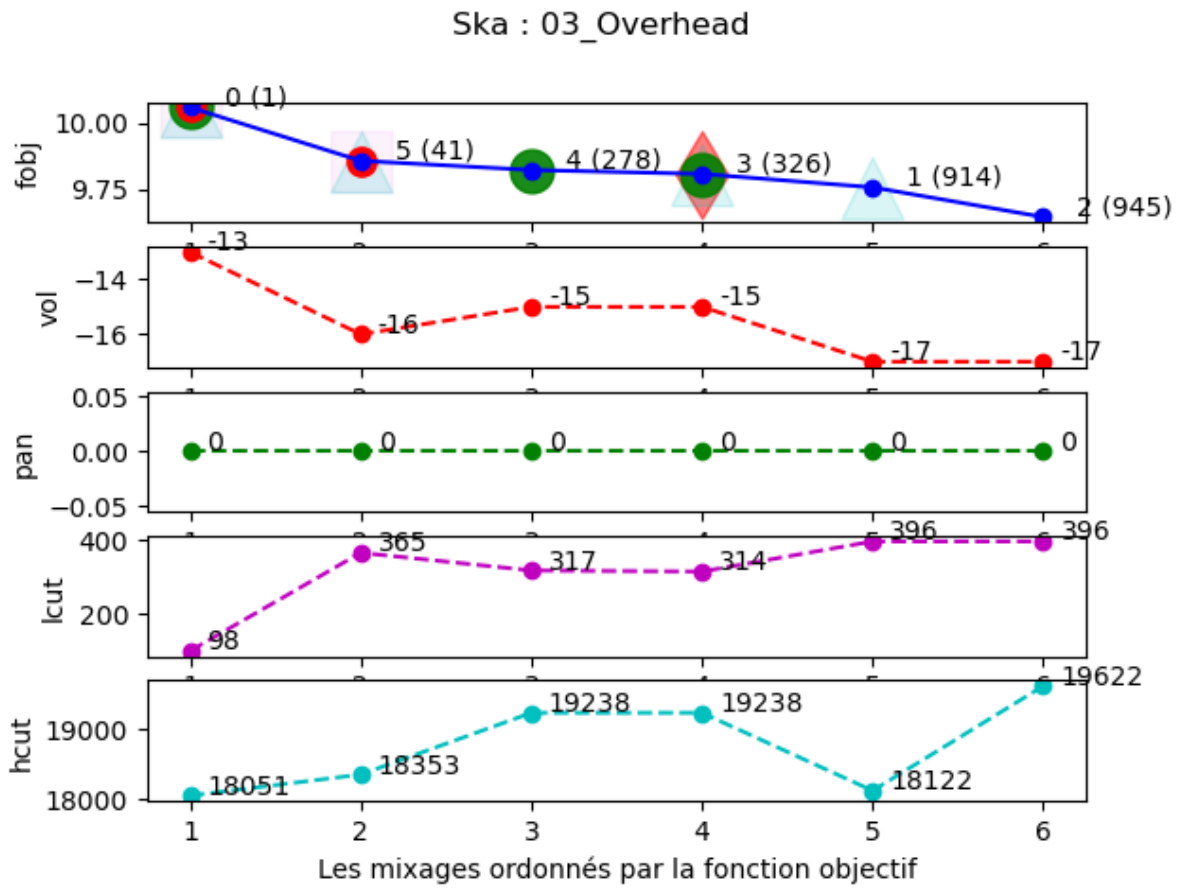


Figure E.21 – Ska : Graphe de convergence – 03_Overhead.

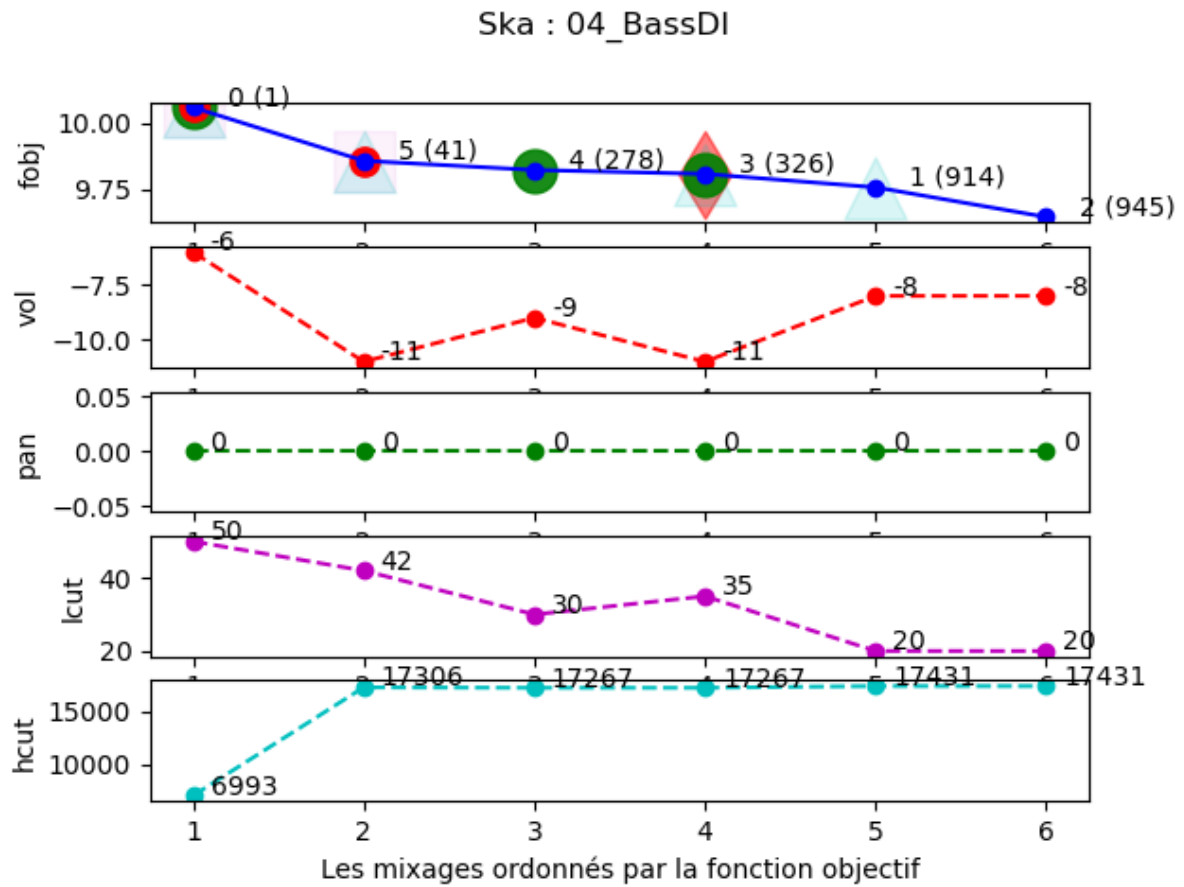


Figure E.22 – Ska : Graphe de convergence – 04_BassDI.

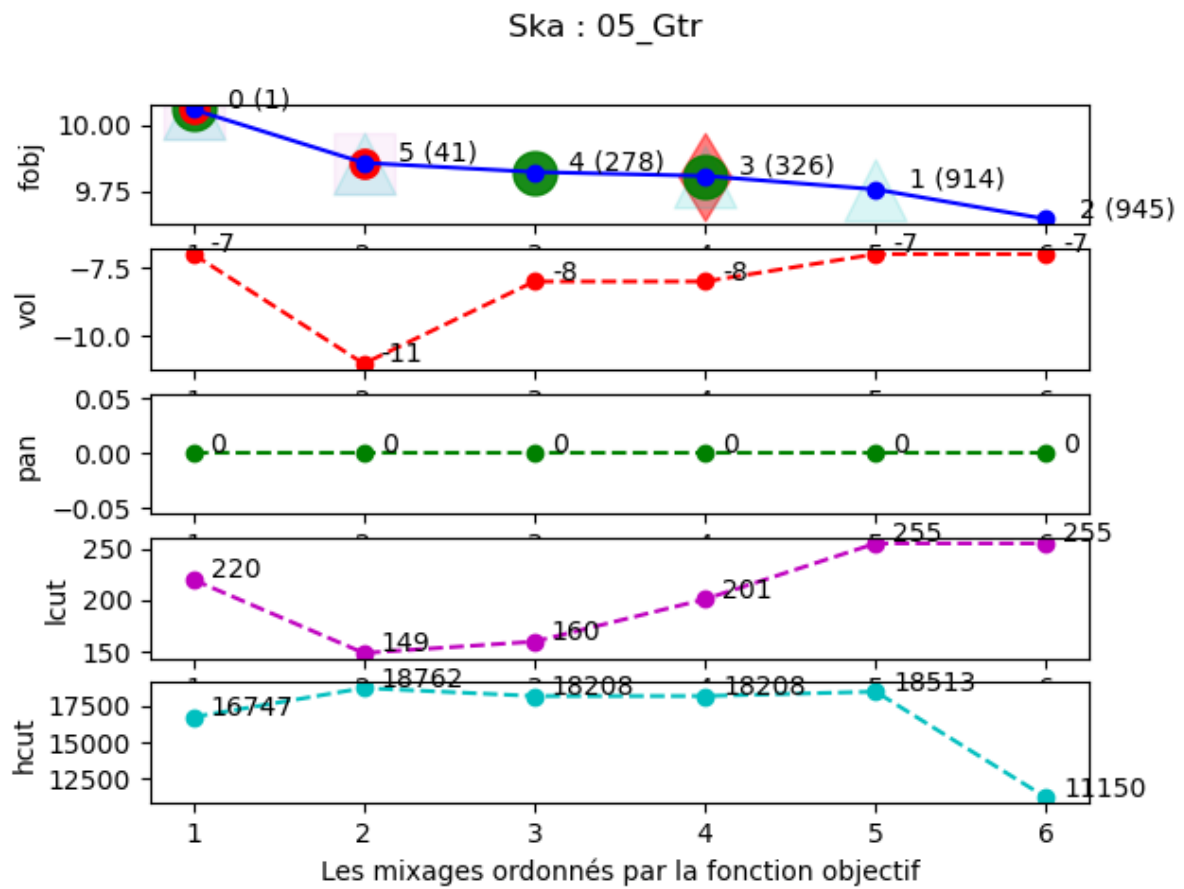


Figure E.23 – Ska : Graphe de convergence – 05_Gtr.

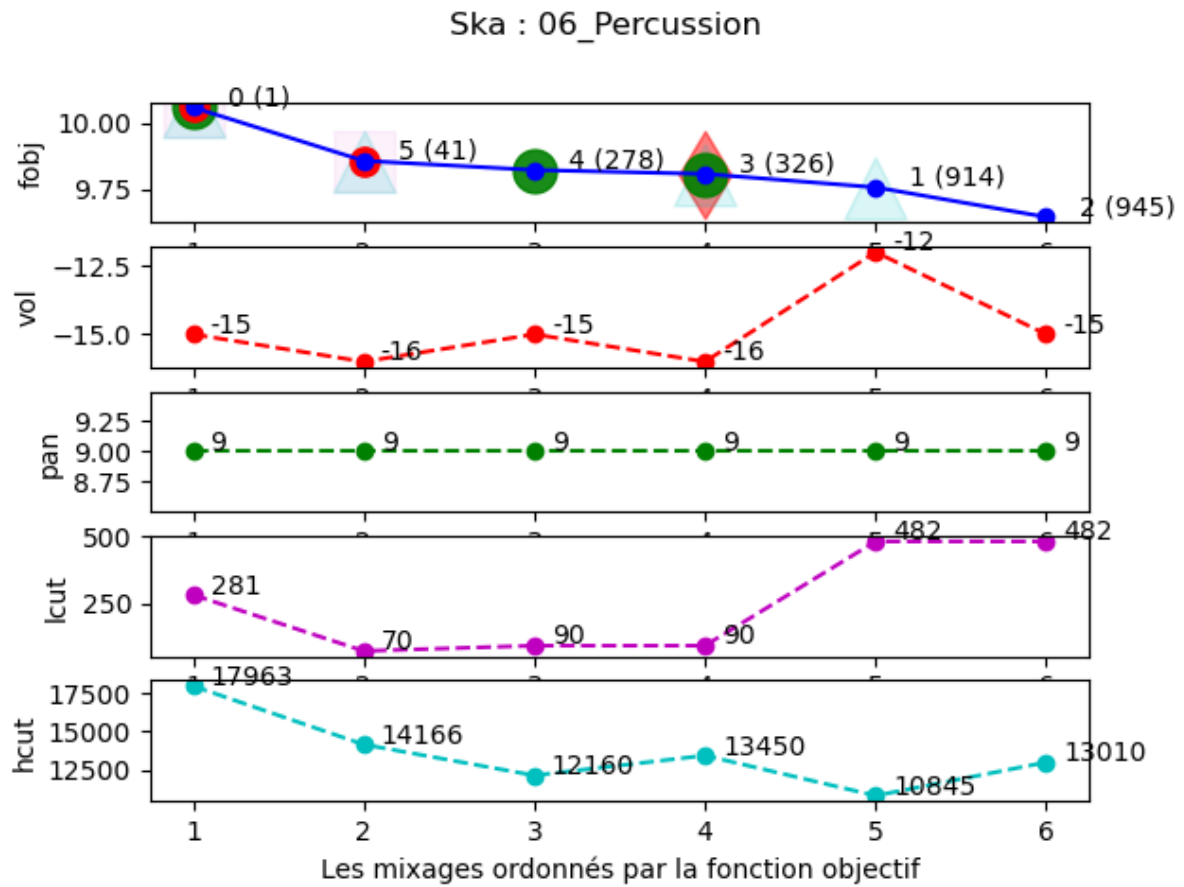


Figure E.24 – Ska : Graphe de convergence – 06_Percussion.

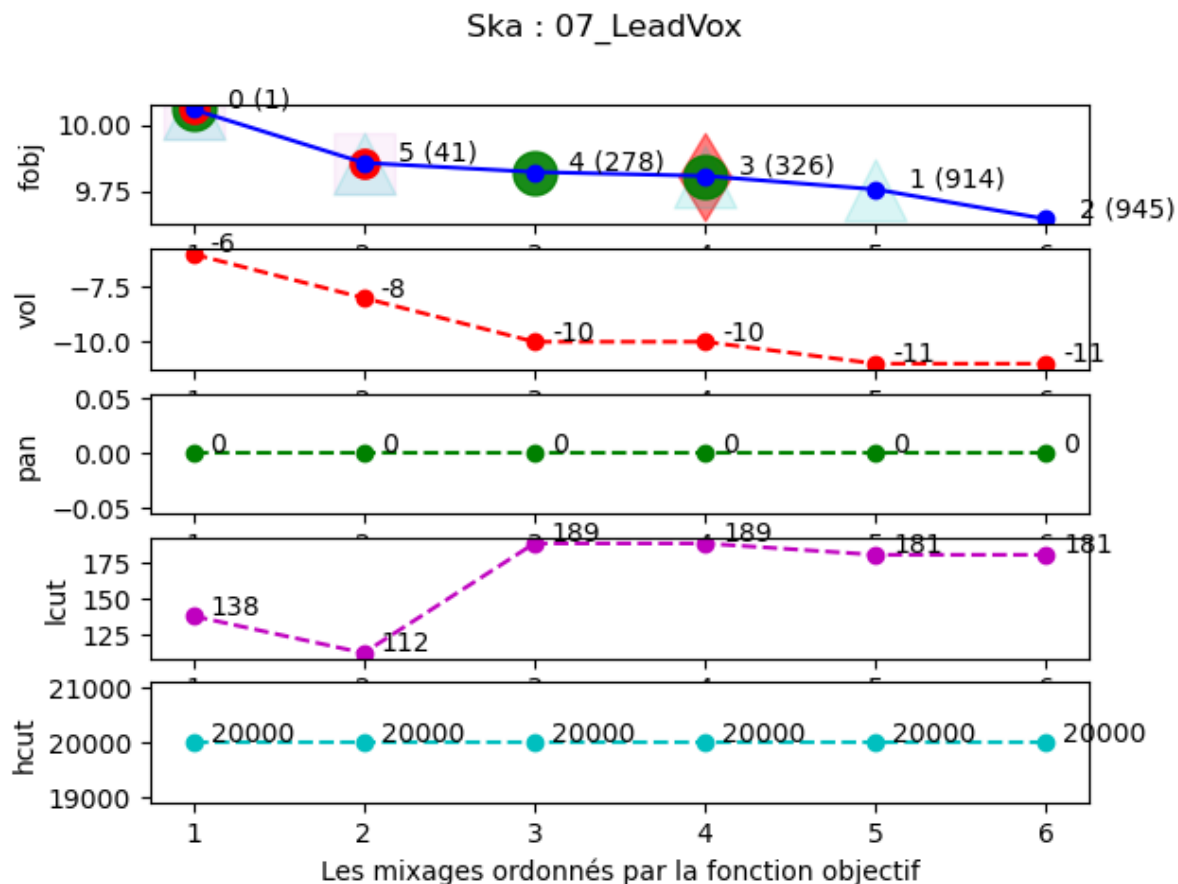


Figure E.25 – Ska : Graphe de convergence – 07_LeadVox.

E.4 Latino : Eddie Garrido - Una Semana Sin Ti : 10 tracks

E.4.1 Les règles de mixage

0th mix : fobj = 1.541

1: MixRules(vol=-17dB, pan=3°, locut=272Hz, hicut=17131Hz)	01_Percussion
2: MixRules(vol=-13dB, pan=-3°, locut=39Hz, hicut=19465Hz)	02_Woodblock
3: MixRules(vol=-13dB, pan=-76°, locut=82Hz, hicut=17894Hz)	03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=32Hz, hicut=11883Hz)	04_Bass
5: MixRules(vol=-16dB, pan=-74°, locut=175Hz, hicut=12326Hz)	05_Piano
6: MixRules(vol=-13dB, pan=-36°, locut=58Hz, hicut=8581Hz)	06_Vibes
7: MixRules(vol=-12dB, pan=44°, locut=54Hz, hicut=11550Hz)	07_Strings1
8: MixRules(vol=-18dB, pan=-9°, locut=187Hz, hicut=8990Hz)	08_Strings2
9: MixRules(vol=-29dB, pan=-10°, locut=373Hz, hicut=17117Hz)	09_FrenchHorns
10: MixRules(vol=-9dB, pan=0°, locut=116Hz, hicut=18118Hz)	10_LeadVox

1th mix : fobj = 1.406

1: MixRules(vol=-12dB, pan=-9°, locut=105Hz, hicut=10689Hz)	01_Percussion
---	---------------

2: MixRules(vol=-13dB, pan=-1°, locut=165Hz, hicut=10728Hz)	02_Woodblock
3: MixRules(vol=-13dB, pan=85°, locut=95Hz, hicut=15517Hz)	03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=51Hz, hicut=15809Hz)	04_Bass
5: MixRules(vol=-12dB, pan=-39°, locut=188Hz, hicut=13628Hz)	05_Piano
6: MixRules(vol=-14dB, pan=67°, locut=192Hz, hicut=19013Hz)	06_Vibes
7: MixRules(vol=-24dB, pan=-8°, locut=106Hz, hicut=16750Hz)	07_Strings1
8: MixRules(vol=-12dB, pan=-47°, locut=166Hz, hicut=9933Hz)	08_Strings2
9: MixRules(vol=-23dB, pan=-81°, locut=186Hz, hicut=16058Hz)	09_FrenchHorns
10: MixRules(vol=-11dB, pan=0°, locut=99Hz, hicut=18628Hz)	10_LeadVox
2th mix : fobj = 1.234	
1: MixRules(vol=-14dB, pan=-5°, locut=96Hz, hicut=18528Hz)	01_Percussion
2: MixRules(vol=-12dB, pan=-48°, locut=139Hz, hicut=17325Hz)	02_Woodblock
3: MixRules(vol=-14dB, pan=39°, locut=43Hz, hicut=16785Hz)	03_TomsCymbals
4: MixRules(vol=-8dB, pan=0°, locut=21Hz, hicut=15368Hz)	04_Bass
5: MixRules(vol=-13dB, pan=58°, locut=105Hz, hicut=12370Hz)	05_Piano
6: MixRules(vol=-17dB, pan=56°, locut=60Hz, hicut=15313Hz)	06_Vibes
7: MixRules(vol=-23dB, pan=61°, locut=181Hz, hicut=8060Hz)	07_Strings1
8: MixRules(vol=-24dB, pan=-74°, locut=50Hz, hicut=16373Hz)	08_Strings2
9: MixRules(vol=-31dB, pan=29°, locut=184Hz, hicut=15717Hz)	09_FrenchHorns
10: MixRules(vol=-9dB, pan=0°, locut=88Hz, hicut=19976Hz)	10_LeadVox
3th mix : fobj = 1.158	
1: MixRules(vol=-12dB, pan=2°, locut=148Hz, hicut=16985Hz)	01_Percussion
2: MixRules(vol=-13dB, pan=-71°, locut=267Hz, hicut=10684Hz)	02_Woodblock
3: MixRules(vol=-12dB, pan=60°, locut=96Hz, hicut=19532Hz)	03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=50Hz, hicut=7232Hz)	04_Bass
5: MixRules(vol=-14dB, pan=38°, locut=131Hz, hicut=10769Hz)	05_Piano
6: MixRules(vol=-14dB, pan=46°, locut=136Hz, hicut=14791Hz)	06_Vibes
7: MixRules(vol=-17dB, pan=-26°, locut=126Hz, hicut=15146Hz)	07_Strings1
8: MixRules(vol=-34dB, pan=-32°, locut=196Hz, hicut=14756Hz)	08_Strings2
9: MixRules(vol=-31dB, pan=-49°, locut=423Hz, hicut=19897Hz)	09_FrenchHorns
10: MixRules(vol=-9dB, pan=0°, locut=31Hz, hicut=19248Hz)	10_LeadVox
4th mix : fobj = 1.105	
1: MixRules(vol=-13dB, pan=4°, locut=190Hz, hicut=16330Hz)	01_Percussion
2: MixRules(vol=-12dB, pan=-88°, locut=117Hz, hicut=15971Hz)	02_Woodblock
3: MixRules(vol=-12dB, pan=58°, locut=33Hz, hicut=17578Hz)	03_TomsCymbals
4: MixRules(vol=-9dB, pan=0°, locut=49Hz, hicut=12434Hz)	04_Bass
5: MixRules(vol=-16dB, pan=75°, locut=190Hz, hicut=10598Hz)	05_Piano
6: MixRules(vol=-13dB, pan=-19°, locut=57Hz, hicut=18461Hz)	06_Vibes
7: MixRules(vol=-34dB, pan=37°, locut=138Hz, hicut=14128Hz)	07_Strings1
8: MixRules(vol=-23dB, pan=19°, locut=44Hz, hicut=19260Hz)	08_Strings2
9: MixRules(vol=-31dB, pan=-64°, locut=315Hz, hicut=19489Hz)	09_FrenchHorns
10: MixRules(vol=-10dB, pan=0°, locut=76Hz, hicut=19904Hz)	10_LeadVox
5th mix : fobj = 1.062	
1: MixRules(vol=-15dB, pan=0°, locut=149Hz, hicut=11981Hz)	01_Percussion
2: MixRules(vol=-13dB, pan=-33°, locut=264Hz, hicut=17201Hz)	02_Woodblock
3: MixRules(vol=-14dB, pan=56°, locut=67Hz, hicut=16492Hz)	03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=35Hz, hicut=11120Hz)	04_Bass
5: MixRules(vol=-16dB, pan=31°, locut=144Hz, hicut=13984Hz)	05_Piano
6: MixRules(vol=-14dB, pan=68°, locut=67Hz, hicut=18280Hz)	06_Vibes
7: MixRules(vol=-33dB, pan=65°, locut=39Hz, hicut=9325Hz)	07_Strings1
8: MixRules(vol=-34dB, pan=-56°, locut=170Hz, hicut=16148Hz)	08_Strings2

9: MixRules(vol=-27dB, pan=-61°, locut=314Hz, hicut=15329Hz)	09_FrenchHorns
10: MixRules(vol=-8dB, pan=0°, locut=88Hz, hicut=19290Hz)	10_LeadVox

6th mix : fobj = 1.042

1: MixRules(vol=-13dB, pan=-2°, locut=196Hz, hicut=18886Hz)	01_Percussion
2: MixRules(vol=-15dB, pan=-75°, locut=223Hz, hicut=15162Hz)	02_Woodblock
3: MixRules(vol=-17dB, pan=-36°, locut=71Hz, hicut=19331Hz)	03_TomsCymbals
4: MixRules(vol=-6dB, pan=0°, locut=36Hz, hicut=6490Hz)	04_Bass
5: MixRules(vol=-12dB, pan=73°, locut=169Hz, hicut=12808Hz)	05_Piano
6: MixRules(vol=-15dB, pan=-87°, locut=87Hz, hicut=9005Hz)	06_Vibes
7: MixRules(vol=-32dB, pan=56°, locut=146Hz, hicut=10987Hz)	07_Strings1
8: MixRules(vol=-22dB, pan=35°, locut=182Hz, hicut=13989Hz)	08_Strings2
9: MixRules(vol=-34dB, pan=-57°, locut=473Hz, hicut=15274Hz)	09_FrenchHorns
10: MixRules(vol=-6dB, pan=0°, locut=56Hz, hicut=18265Hz)	10_LeadVox

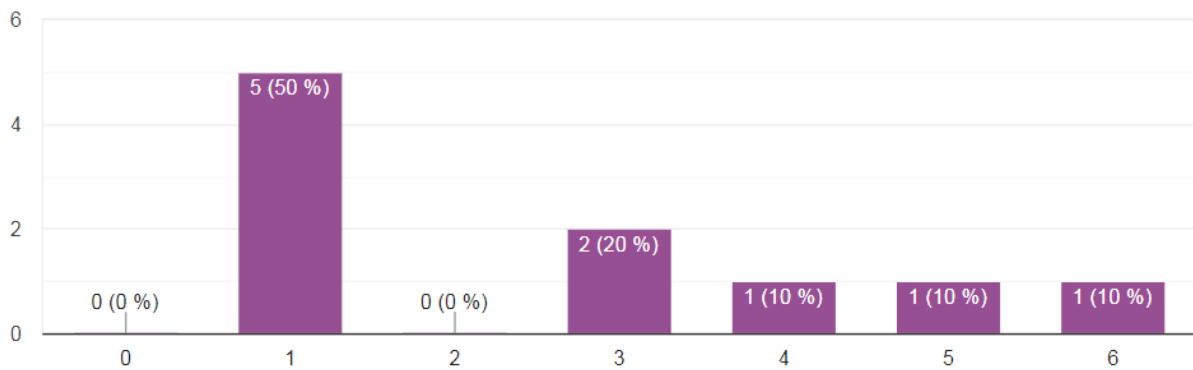


Figure E.26 – Latino : Le pré-mixage préféré.

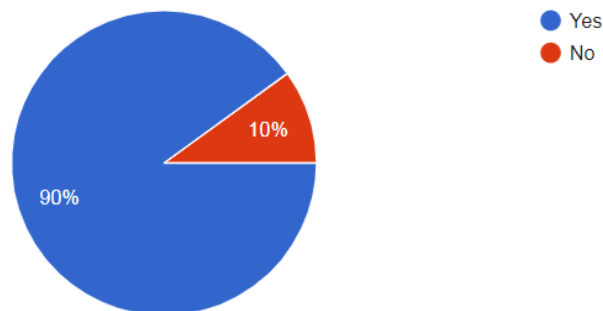


Figure E.27 – Latino : L'intention d'utiliser le pré-mixage préféré.

E.4.2 Les graphes de convergence

La fonction objectif

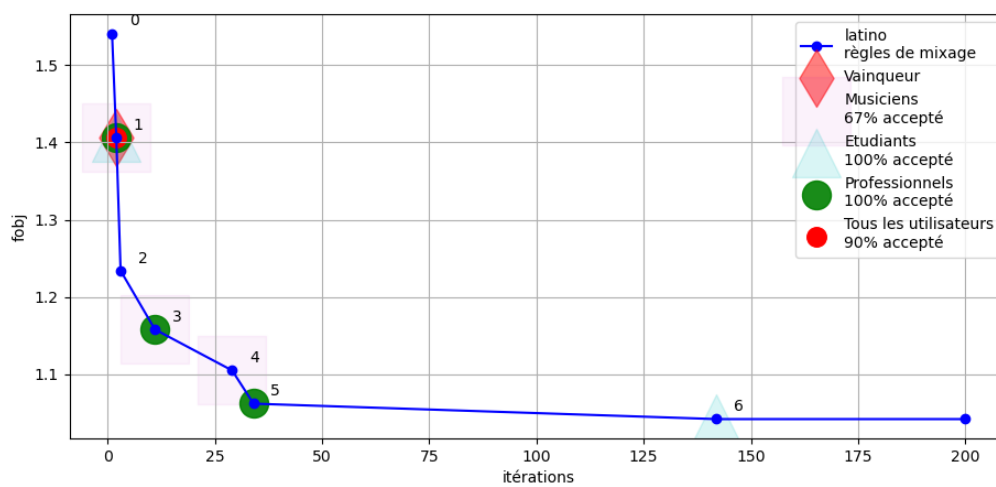


Figure E.28 – Latino : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

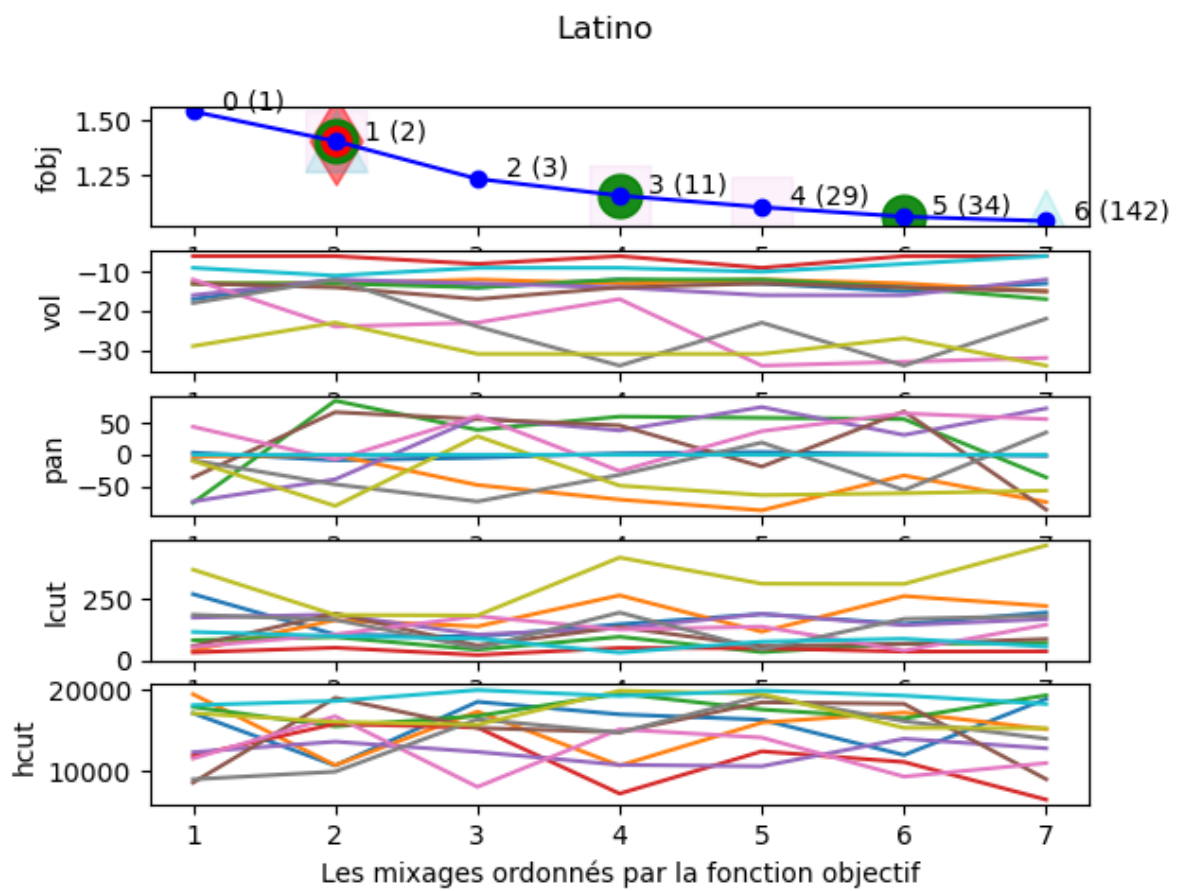


Figure E.29 – Latino : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

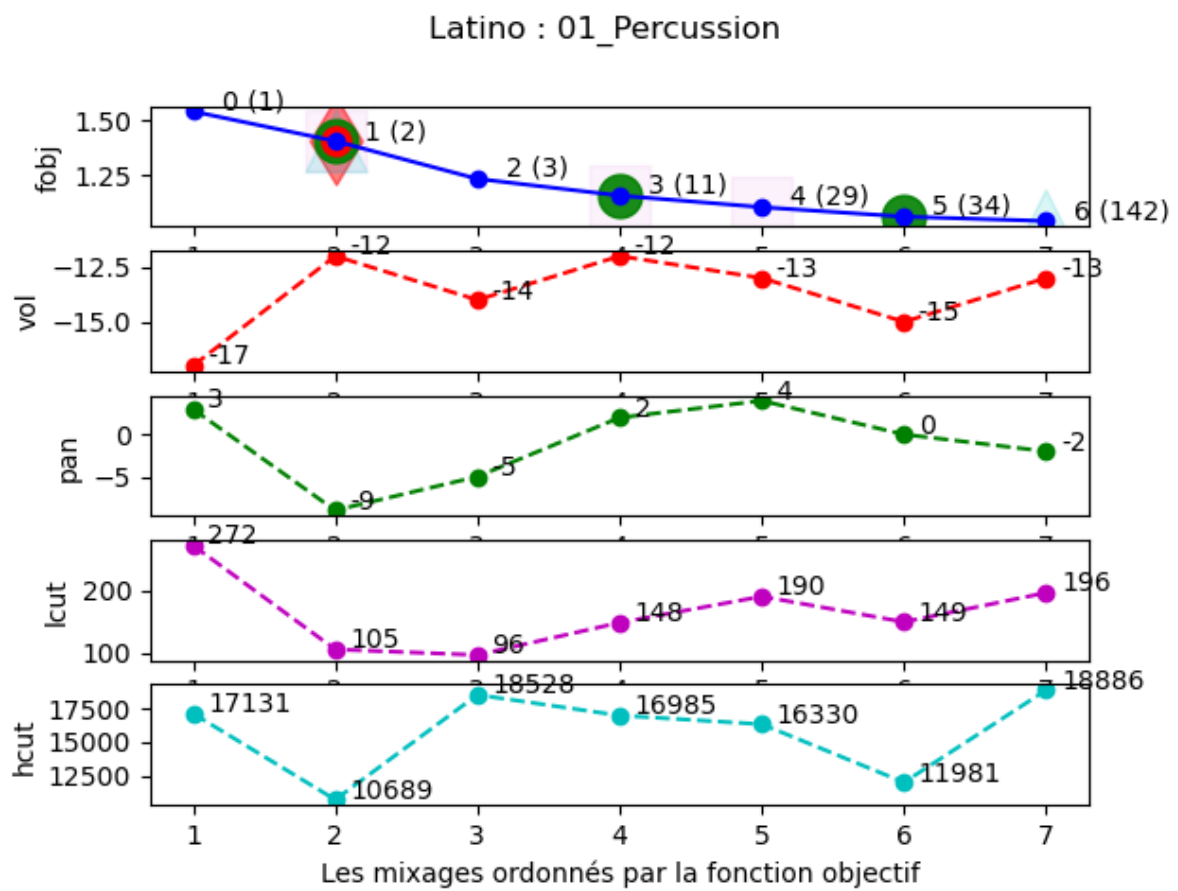


Figure E.30 – Latino : Graphe de convergence – 01_Percussion.

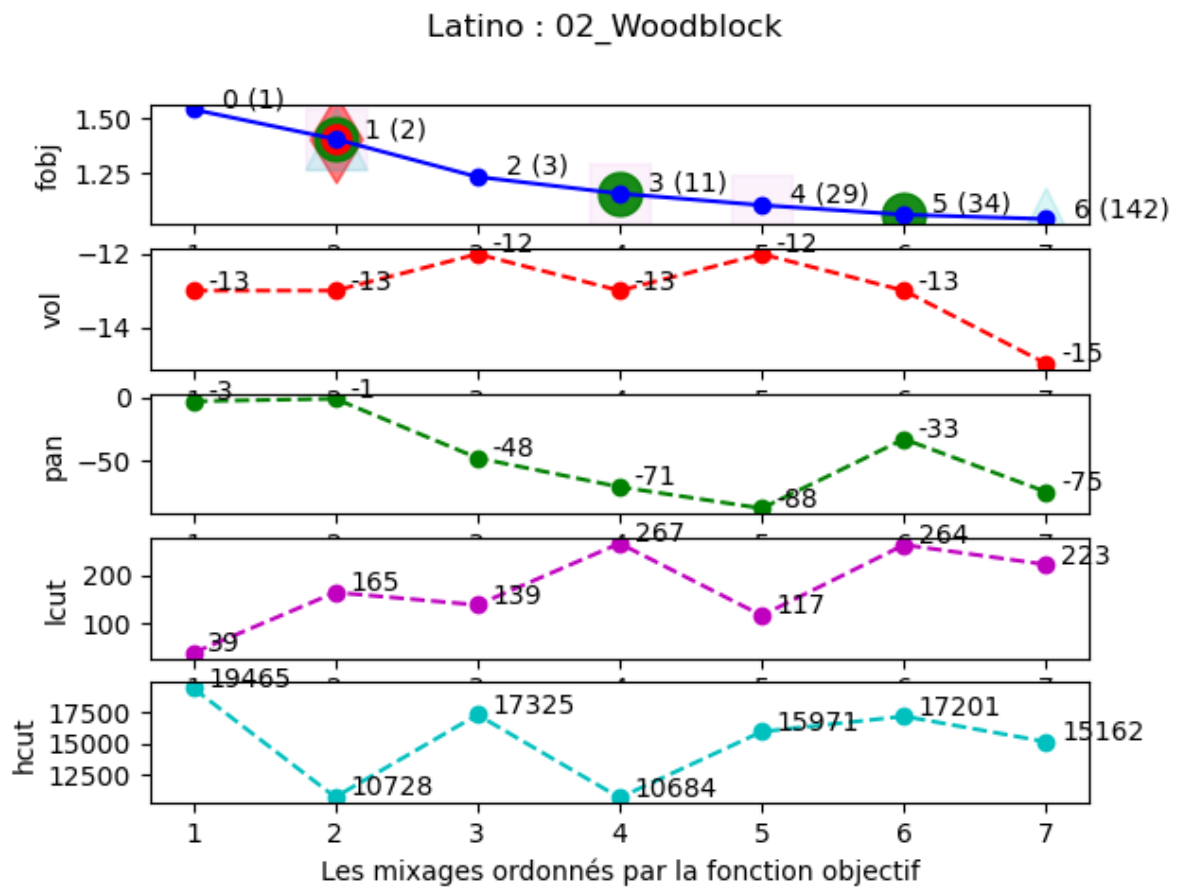


Figure E.31 – Latino : Graphe de convergence – 02_Woodblock.

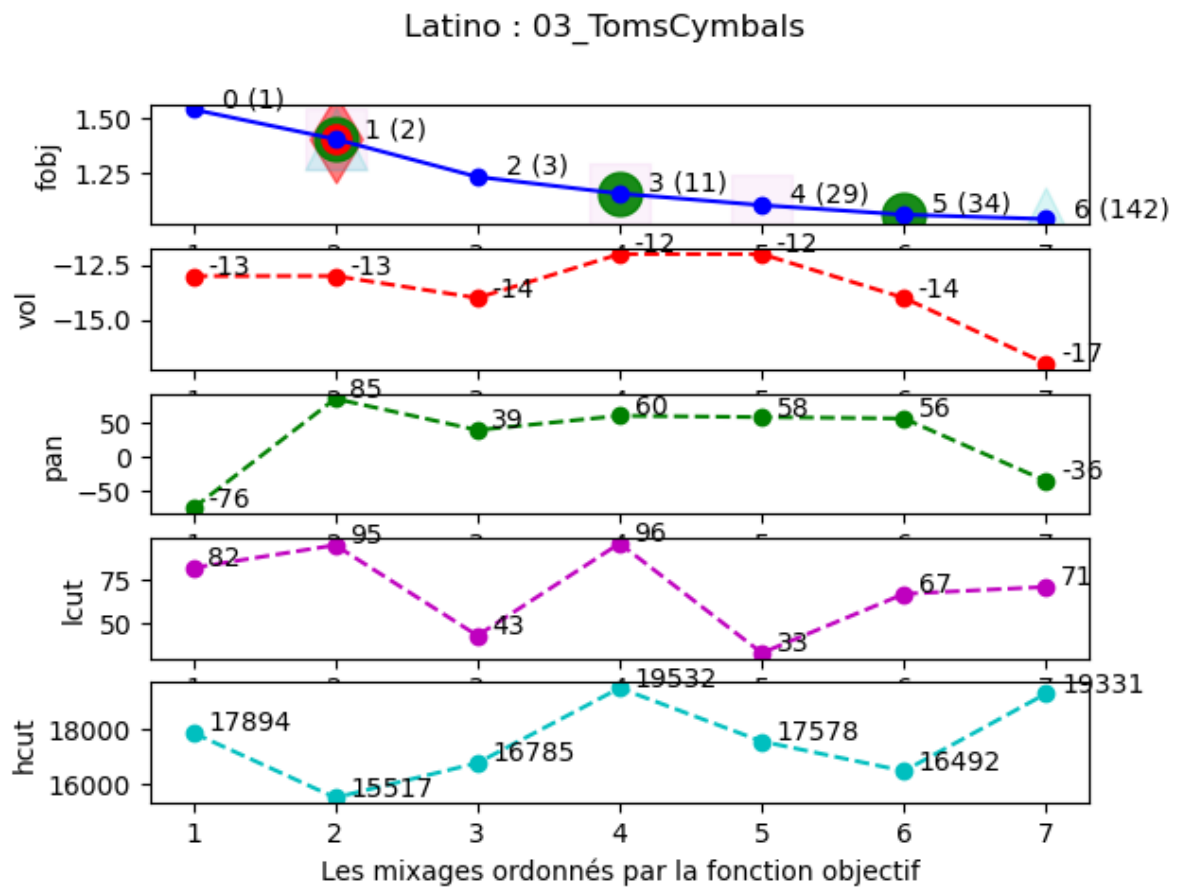


Figure E.32 – Latino : Graphe de convergence – 03_TomsCymbals.

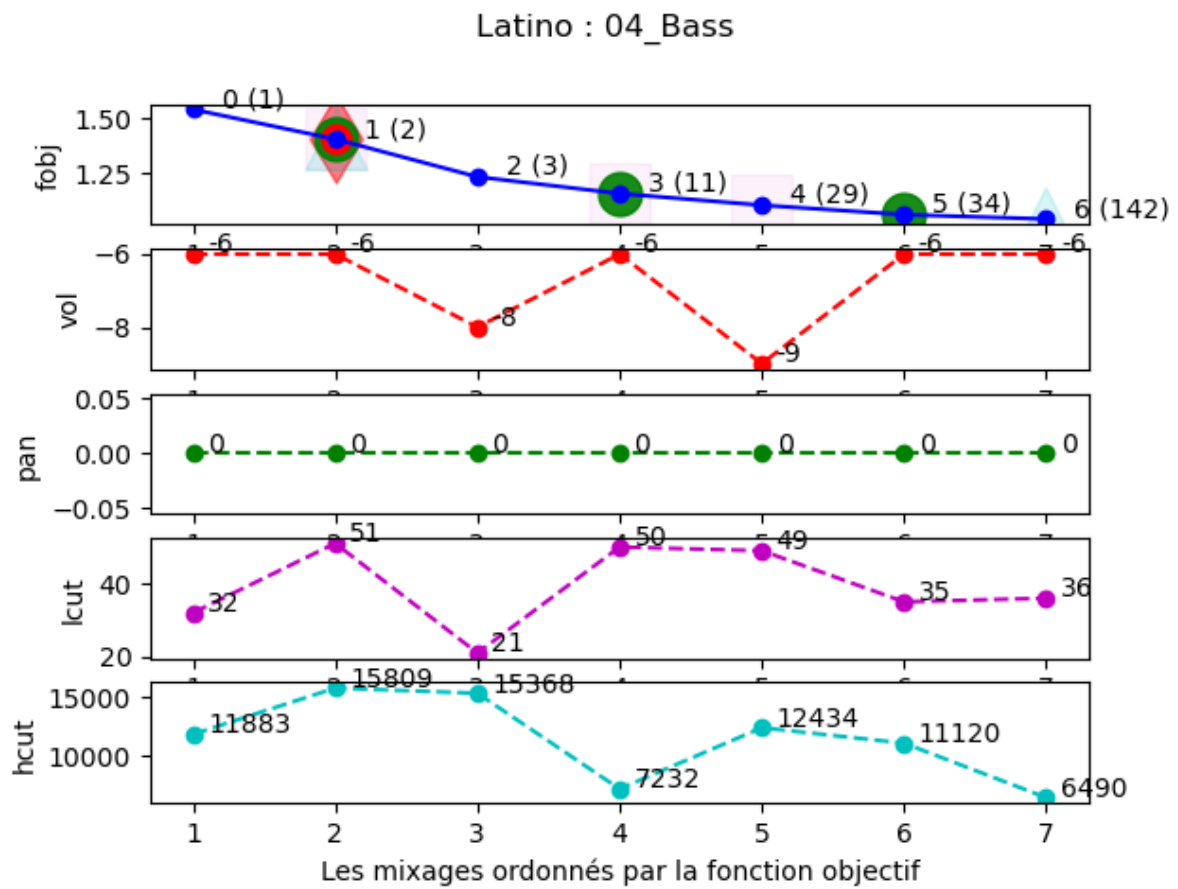


Figure E.33 – Latino : Graphe de convergence – 04_Bass.

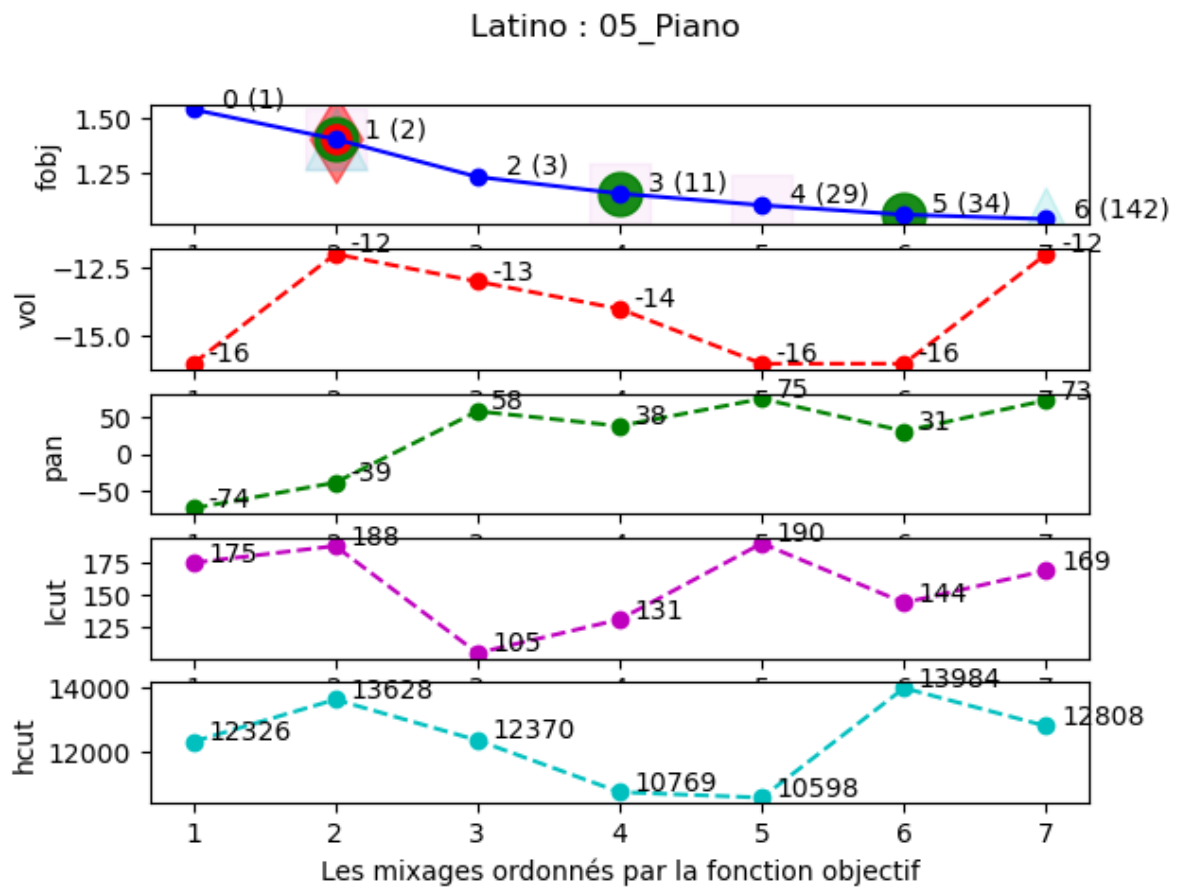


Figure E.34 – Latino : Graphe de convergence – 05_Piano.

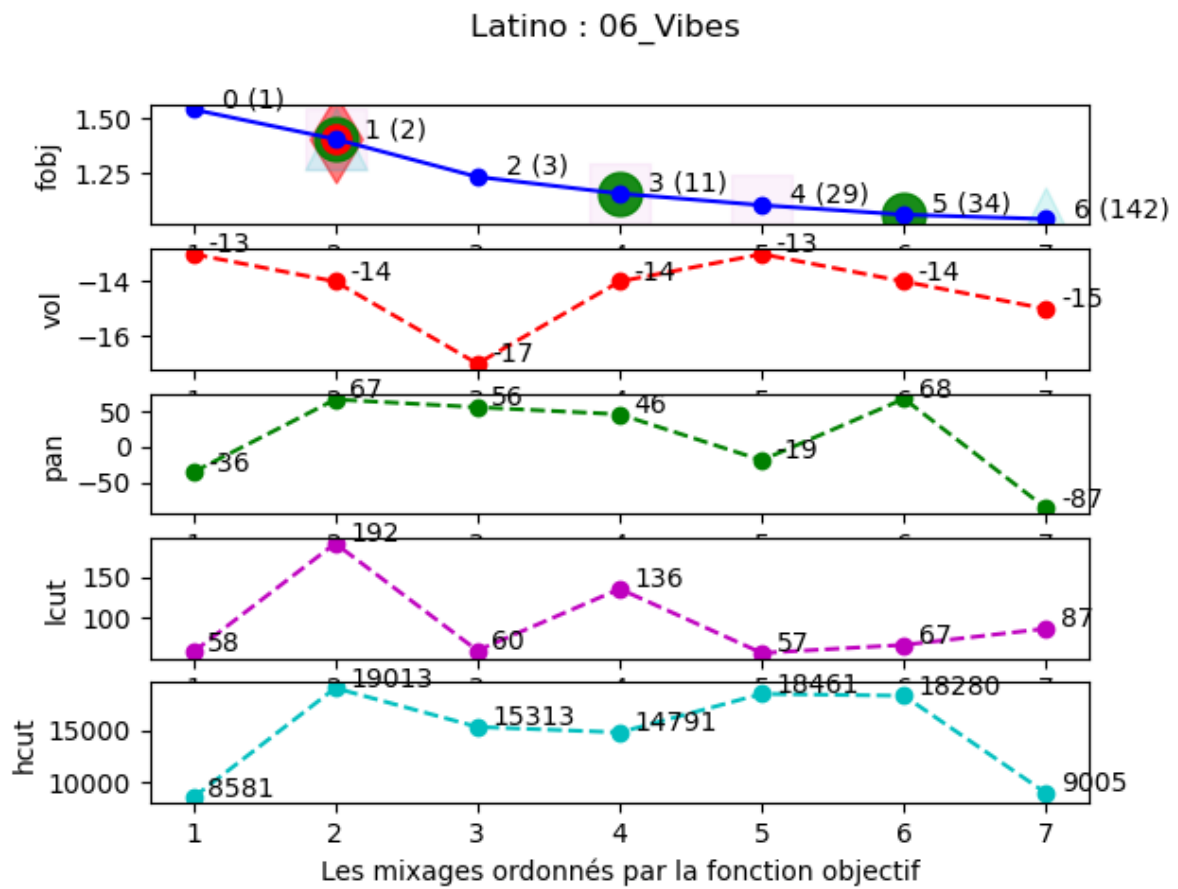


Figure E.35 – Latino : Graphe de convergence – 06_Vibes.

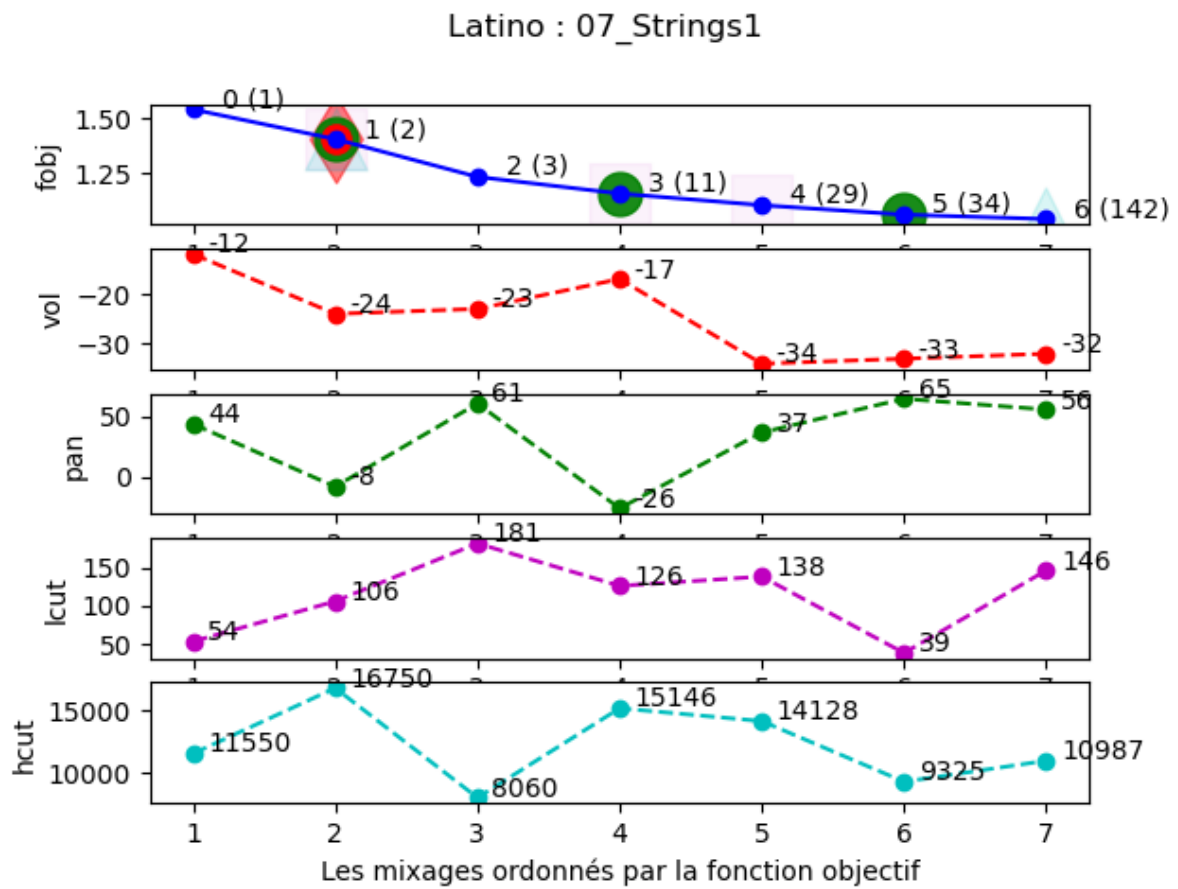


Figure E.36 – Latino : Graphe de convergence – 07_Strings1.

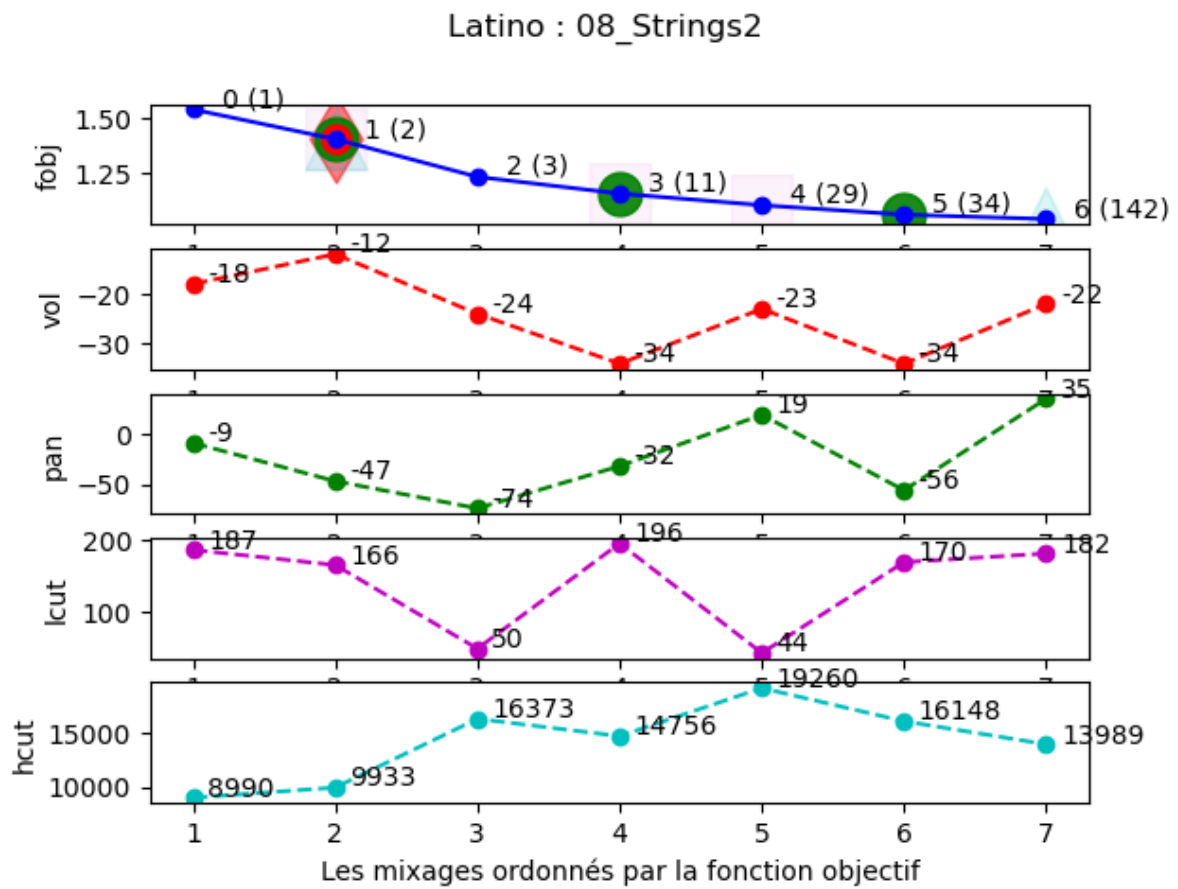


Figure E.37 – Latino : Graphe de convergence – 08_Strings2.

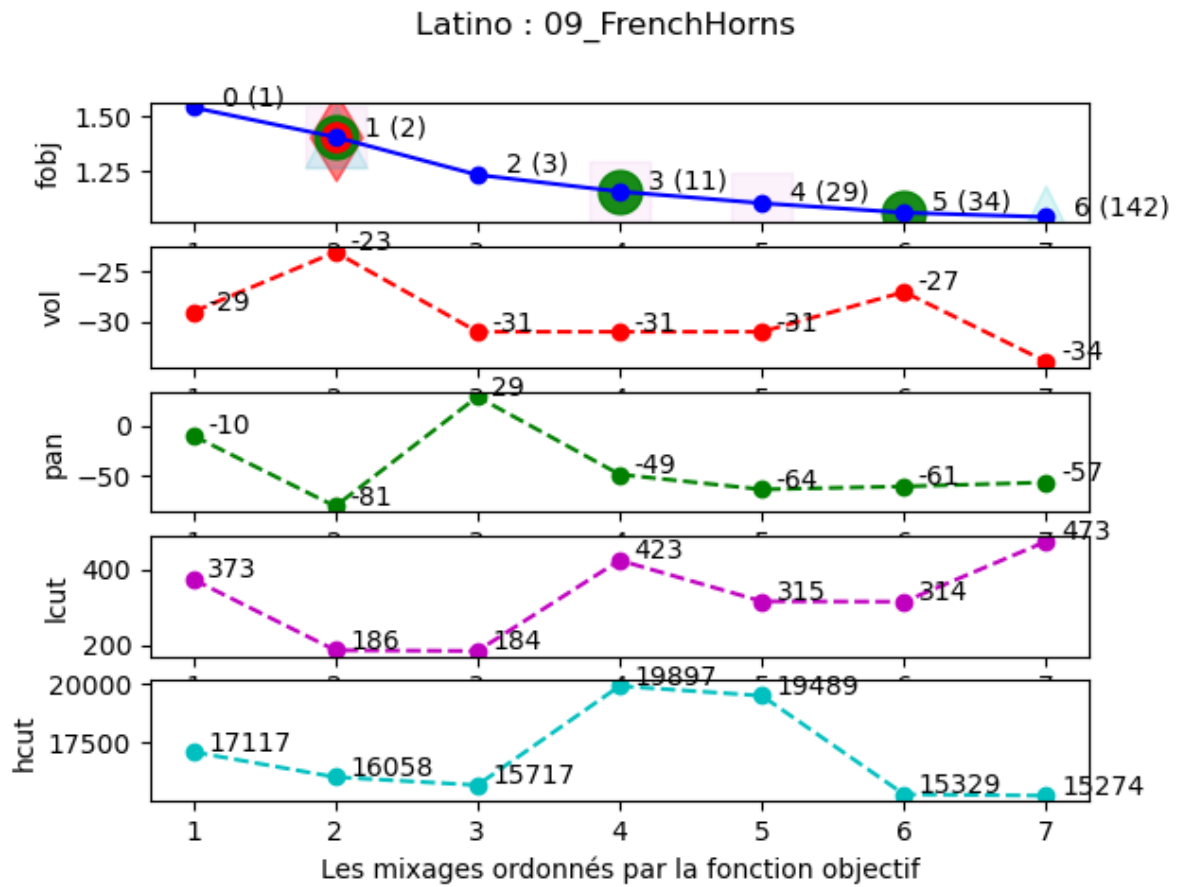


Figure E.38 – Latino : Graphe de convergence – 09_FrenchHorns.

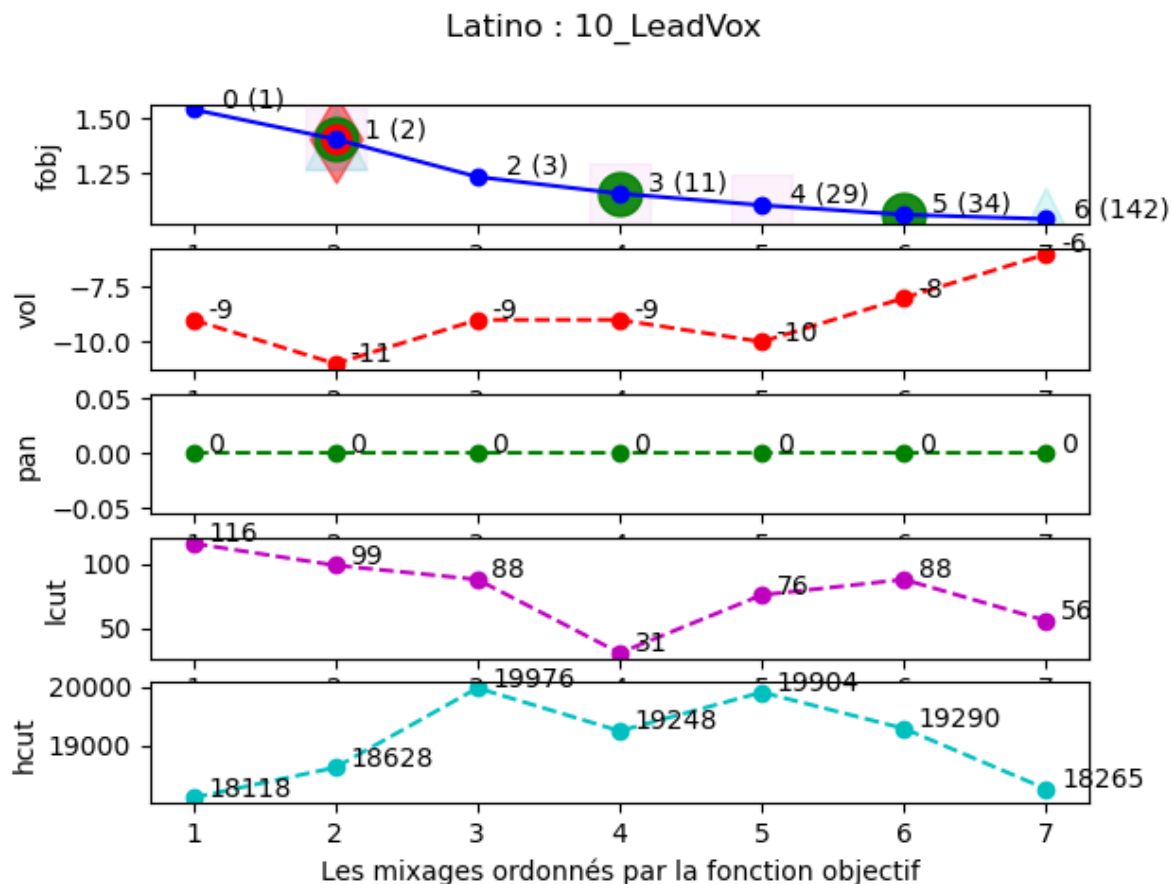


Figure E.39 – Latino : Graphe de convergence – 10_LeadVox.

E.5 Country : Angela Thomas Wade - Milk Cow Blues : 11 tracks

E.5.1 Les règles de mixage

0th mix : fobj = 1.805

- | | |
|--|--------------|
| 1: MixRules(vol=-10dB, pan=0°, locut=26Hz, hicut=13466Hz) | 01_Kick |
| 2: MixRules(vol=-11dB, pan=0°, locut=76Hz, hicut=17182Hz) | 02_Snare |
| 3: MixRules(vol=-16dB, pan=-66°, locut=50Hz, hicut=11914Hz) | 03_Tom |
| 4: MixRules(vol=-15dB, pan=-1°, locut=149Hz, hicut=19256Hz) | 04_Overheads |
| 5: MixRules(vol=-6dB, pan=0°, locut=30Hz, hicut=17199Hz) | 05_Bass |
| 6: MixRules(vol=-12dB, pan=34°, locut=133Hz, hicut=10836Hz) | 06_AcGtr |
| 7: MixRules(vol=-12dB, pan=27°, locut=141Hz, hicut=8890Hz) | 07_Dobro |
| 8: MixRules(vol=-12dB, pan=68°, locut=163Hz, hicut=11502Hz) | 08_ElecGtr |
| 9: MixRules(vol=-10dB, pan=-13°, locut=32Hz, hicut=17320Hz) | 09_Fiddle |
| 10: MixRules(vol=-17dB, pan=-81°, locut=35Hz, hicut=13056Hz) | 10_Piano |
| 11: MixRules(vol=-9dB, pan=0°, locut=125Hz, hicut=20000Hz) | 11_LeadVox |

1th mix : fobj = 1.706

1: MixRules(vol=-7dB, pan=0°, locut=37Hz, hicut=17899Hz)	01_Kick
2: MixRules(vol=-9dB, pan=0°, locut=167Hz, hicut=15632Hz)	02_Snare
3: MixRules(vol=-13dB, pan=-37°, locut=60Hz, hicut=10053Hz)	03_Tom
4: MixRules(vol=-17dB, pan=-3°, locut=158Hz, hicut=18382Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=22Hz, hicut=6797Hz)	05_Bass
6: MixRules(vol=-17dB, pan=9°, locut=125Hz, hicut=14866Hz)	06_AcGtr
7: MixRules(vol=-16dB, pan=30°, locut=138Hz, hicut=10427Hz)	07_Dobro
8: MixRules(vol=-14dB, pan=-46°, locut=196Hz, hicut=17220Hz)	08_ElecGtr
9: MixRules(vol=-7dB, pan=-88°, locut=197Hz, hicut=18877Hz)	09_Fiddle
10: MixRules(vol=-12dB, pan=-47°, locut=50Hz, hicut=10142Hz)	10_Piano
11: MixRules(vol=-11dB, pan=0°, locut=124Hz, hicut=20000Hz)	11_LeadVox

2th mix : fobj = 1.715

1: MixRules(vol=-10dB, pan=0°, locut=33Hz, hicut=16583Hz)	01_Kick
2: MixRules(vol=-6dB, pan=0°, locut=188Hz, hicut=10310Hz)	02_Snare
3: MixRules(vol=-16dB, pan=7°, locut=54Hz, hicut=7140Hz)	03_Tom
4: MixRules(vol=-14dB, pan=7°, locut=209Hz, hicut=18419Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=33Hz, hicut=11370Hz)	05_Bass
6: MixRules(vol=-15dB, pan=16°, locut=27Hz, hicut=13535Hz)	06_AcGtr
7: MixRules(vol=-16dB, pan=87°, locut=110Hz, hicut=16615Hz)	07_Dobro
8: MixRules(vol=-12dB, pan=20°, locut=151Hz, hicut=12919Hz)	08_ElecGtr
9: MixRules(vol=-11dB, pan=59°, locut=113Hz, hicut=16309Hz)	09_Fiddle
10: MixRules(vol=-12dB, pan=-85°, locut=55Hz, hicut=18370Hz)	10_Piano
11: MixRules(vol=-9dB, pan=0°, locut=124Hz, hicut=20000Hz)	11_LeadVox

3th mix : fobj = 1.667

1: MixRules(vol=-6dB, pan=0°, locut=25Hz, hicut=13047Hz)	01_Kick
2: MixRules(vol=-11dB, pan=0°, locut=220Hz, hicut=18121Hz)	02_Snare
3: MixRules(vol=-14dB, pan=-42°, locut=79Hz, hicut=10919Hz)	03_Tom
4: MixRules(vol=-14dB, pan=-4°, locut=69Hz, hicut=18352Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=31Hz, hicut=18780Hz)	05_Bass
6: MixRules(vol=-15dB, pan=80°, locut=156Hz, hicut=16002Hz)	06_AcGtr
7: MixRules(vol=-12dB, pan=70°, locut=108Hz, hicut=12116Hz)	07_Dobro
8: MixRules(vol=-12dB, pan=-70°, locut=232Hz, hicut=11424Hz)	08_ElecGtr
9: MixRules(vol=-10dB, pan=42°, locut=85Hz, hicut=17466Hz)	09_Fiddle
10: MixRules(vol=-17dB, pan=-1°, locut=78Hz, hicut=10814Hz)	10_Piano
11: MixRules(vol=-11dB, pan=0°, locut=157Hz, hicut=20000Hz)	11_LeadVox

4th mix : fobj = 1.661

1: MixRules(vol=-8dB, pan=0°, locut=34Hz, hicut=18208Hz)	01_Kick
2: MixRules(vol=-6dB, pan=0°, locut=250Hz, hicut=10885Hz)	02_Snare
3: MixRules(vol=-13dB, pan=-27°, locut=39Hz, hicut=10174Hz)	03_Tom
4: MixRules(vol=-13dB, pan=9°, locut=94Hz, hicut=18010Hz)	04_Overheads
5: MixRules(vol=-7dB, pan=0°, locut=30Hz, hicut=11197Hz)	05_Bass
6: MixRules(vol=-16dB, pan=87°, locut=107Hz, hicut=12551Hz)	06_AcGtr
7: MixRules(vol=-15dB, pan=72°, locut=134Hz, hicut=19409Hz)	07_Dobro
8: MixRules(vol=-16dB, pan=87°, locut=194Hz, hicut=18667Hz)	08_ElecGtr
9: MixRules(vol=-8dB, pan=-17°, locut=41Hz, hicut=19735Hz)	09_Fiddle
10: MixRules(vol=-14dB, pan=-22°, locut=36Hz, hicut=18768Hz)	10_Piano
11: MixRules(vol=-11dB, pan=0°, locut=47Hz, hicut=20000Hz)	11_LeadVox

5th mix : fobj = 1.590

1: MixRules(vol=-8dB, pan=0°, locut=27Hz, hicut=9721Hz)	01_Kick
---	---------

2: MixRules(vol=-11dB, pan=0°, locut=131Hz, hicut=16323Hz)	02_Snare
3: MixRules(vol=-15dB, pan=-30°, locut=32Hz, hicut=17909Hz)	03_Tom
4: MixRules(vol=-13dB, pan=0°, locut=54Hz, hicut=19441Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=38Hz, hicut=17470Hz)	05_Bass
6: MixRules(vol=-17dB, pan=58°, locut=194Hz, hicut=12147Hz)	06_AcGtr
7: MixRules(vol=-14dB, pan=33°, locut=105Hz, hicut=16292Hz)	07_Dobro
8: MixRules(vol=-16dB, pan=-33°, locut=69Hz, hicut=17490Hz)	08_ElecGtr
9: MixRules(vol=-8dB, pan=79°, locut=177Hz, hicut=19478Hz)	09_Fiddle
10: MixRules(vol=-16dB, pan=-64°, locut=78Hz, hicut=11715Hz)	10_Piano
11: MixRules(vol=-10dB, pan=0°, locut=189Hz, hicut=20000Hz)	11_LeadVox

6th mix : fobj = 1.544

1: MixRules(vol=-10dB, pan=0°, locut=21Hz, hicut=4389Hz)	01_Kick
2: MixRules(vol=-6dB, pan=0°, locut=266Hz, hicut=16477Hz)	02_Snare
3: MixRules(vol=-15dB, pan=77°, locut=67Hz, hicut=14889Hz)	03_Tom
4: MixRules(vol=-17dB, pan=4°, locut=155Hz, hicut=19014Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=44Hz, hicut=16132Hz)	05_Bass
6: MixRules(vol=-16dB, pan=-78°, locut=42Hz, hicut=8004Hz)	06_AcGtr
7: MixRules(vol=-14dB, pan=36°, locut=95Hz, hicut=6567Hz)	07_Dobro
8: MixRules(vol=-14dB, pan=-78°, locut=56Hz, hicut=17865Hz)	08_ElecGtr
9: MixRules(vol=-11dB, pan=83°, locut=74Hz, hicut=18167Hz)	09_Fiddle
10: MixRules(vol=-14dB, pan=-66°, locut=50Hz, hicut=17521Hz)	10_Piano
11: MixRules(vol=-8dB, pan=0°, locut=131Hz, hicut=20000Hz)	11_LeadVox

7th mix : fobj = 1.520

1: MixRules(vol=-8dB, pan=0°, locut=27Hz, hicut=12628Hz)	01_Kick
2: MixRules(vol=-7dB, pan=0°, locut=216Hz, hicut=15925Hz)	02_Snare
3: MixRules(vol=-17dB, pan=26°, locut=77Hz, hicut=8947Hz)	03_Tom
4: MixRules(vol=-12dB, pan=7°, locut=90Hz, hicut=18157Hz)	04_Overheads
5: MixRules(vol=-6dB, pan=0°, locut=41Hz, hicut=17931Hz)	05_Bass
6: MixRules(vol=-12dB, pan=51°, locut=62Hz, hicut=10050Hz)	06_AcGtr
7: MixRules(vol=-15dB, pan=69°, locut=143Hz, hicut=6192Hz)	07_Dobro
8: MixRules(vol=-17dB, pan=54°, locut=41Hz, hicut=10789Hz)	08_ElecGtr
9: MixRules(vol=-10dB, pan=-79°, locut=186Hz, hicut=16260Hz)	09_Fiddle
10: MixRules(vol=-17dB, pan=-83°, locut=71Hz, hicut=16969Hz)	10_Piano
11: MixRules(vol=-8dB, pan=0°, locut=66Hz, hicut=20000Hz)	11_LeadVox

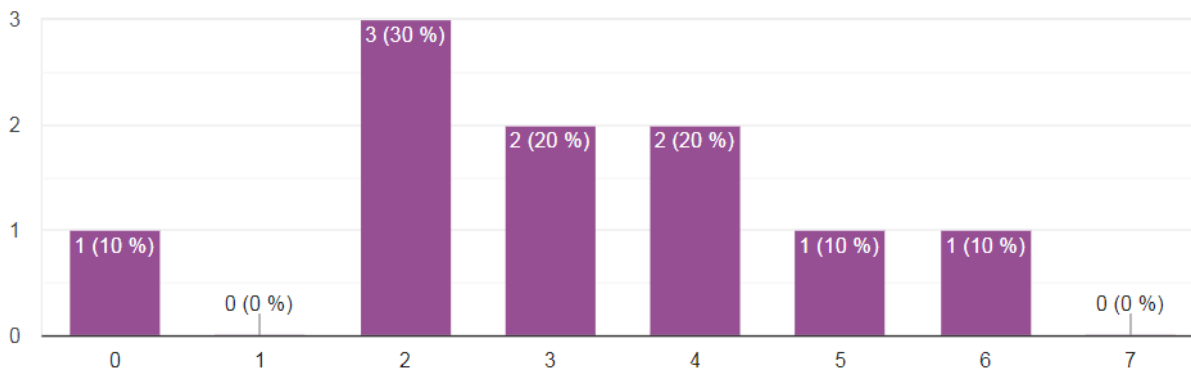


Figure E.40 – Country : Le pré-mixage préféré.

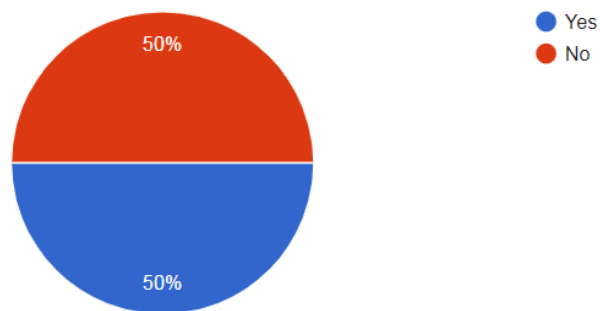


Figure E.41 – Country : L'intention d'utiliser le pré-mixage préféré.

E.5.2 Les graphes de convergence

La fonction objectif

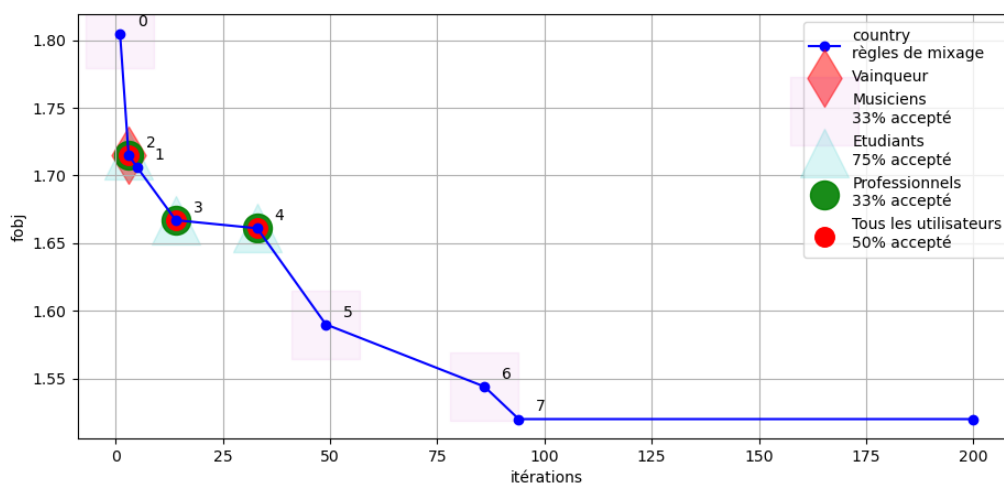


Figure E.42 – Country : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

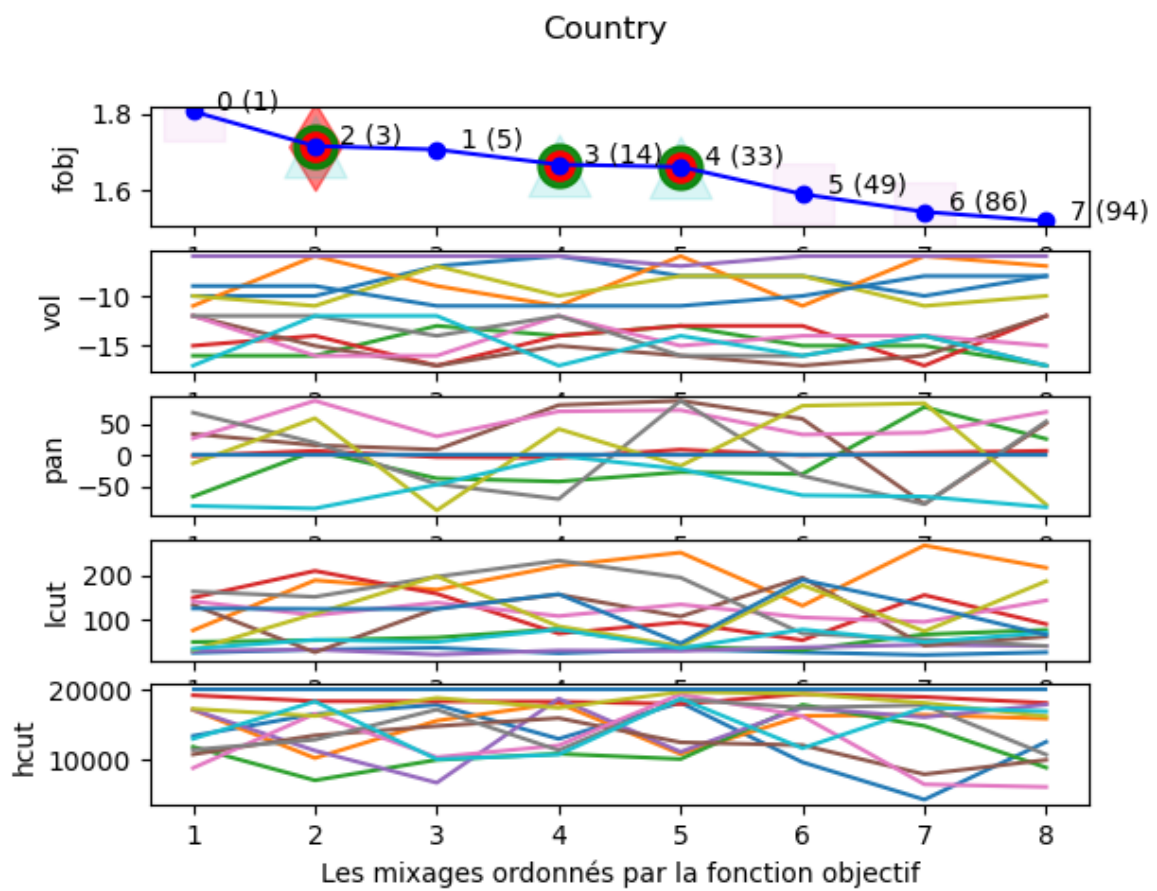


Figure E.43 – Country : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

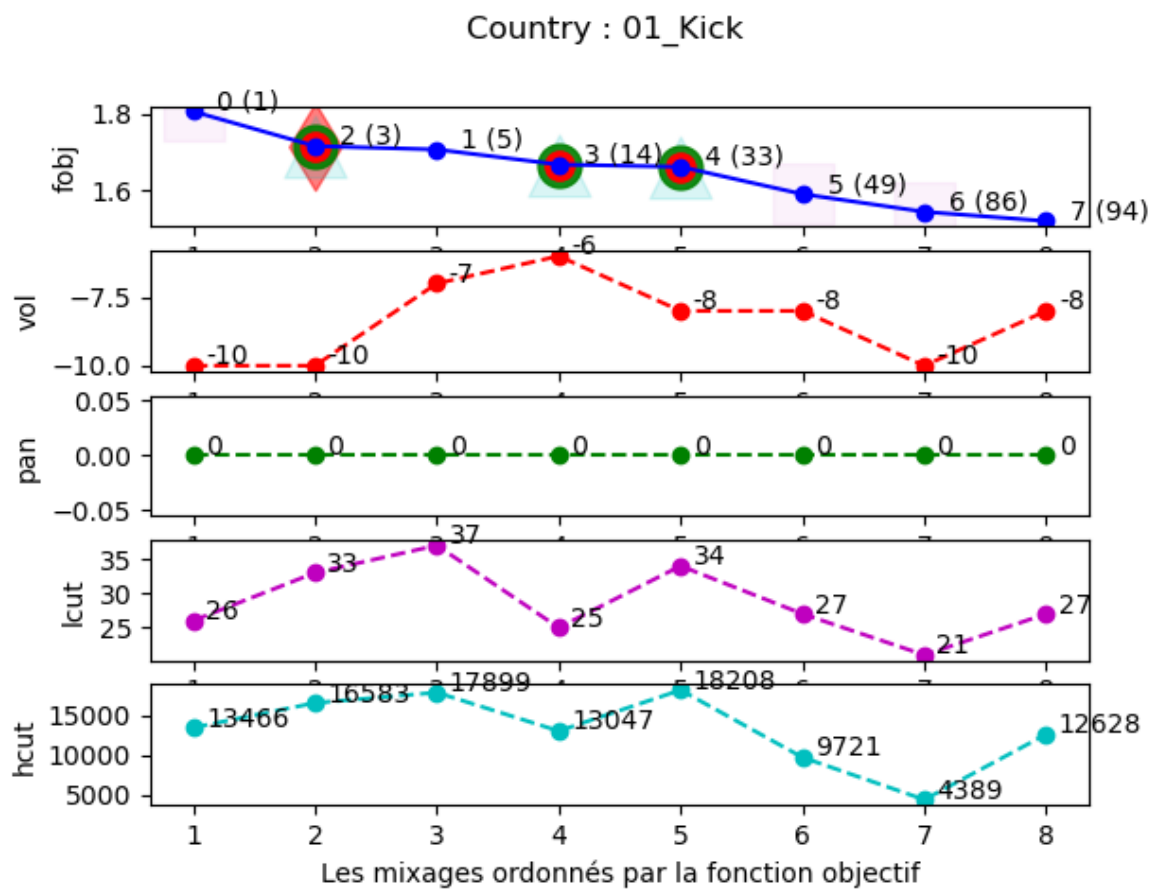


Figure E.44 – Country : Graphe de convergence – 01_Kick.

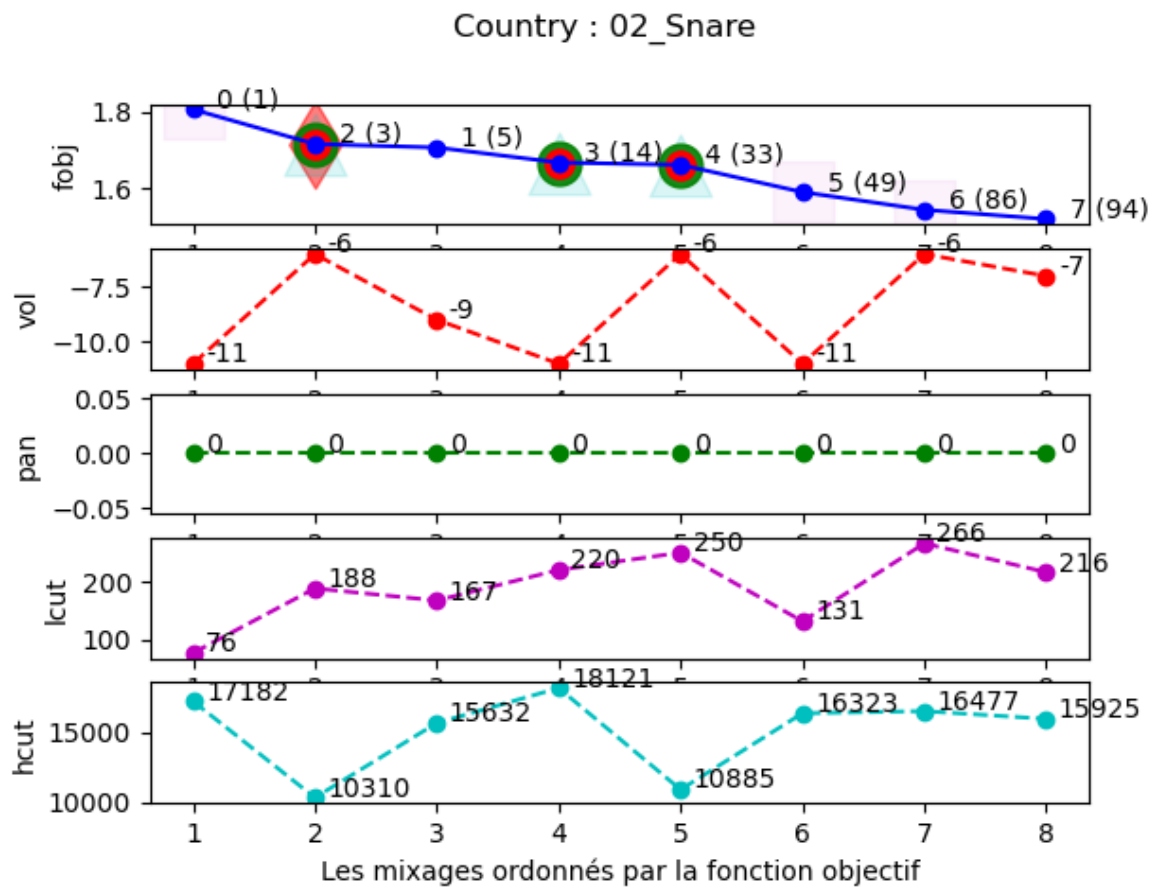


Figure E.45 – Country : Graphe de convergence – 02_Snare.

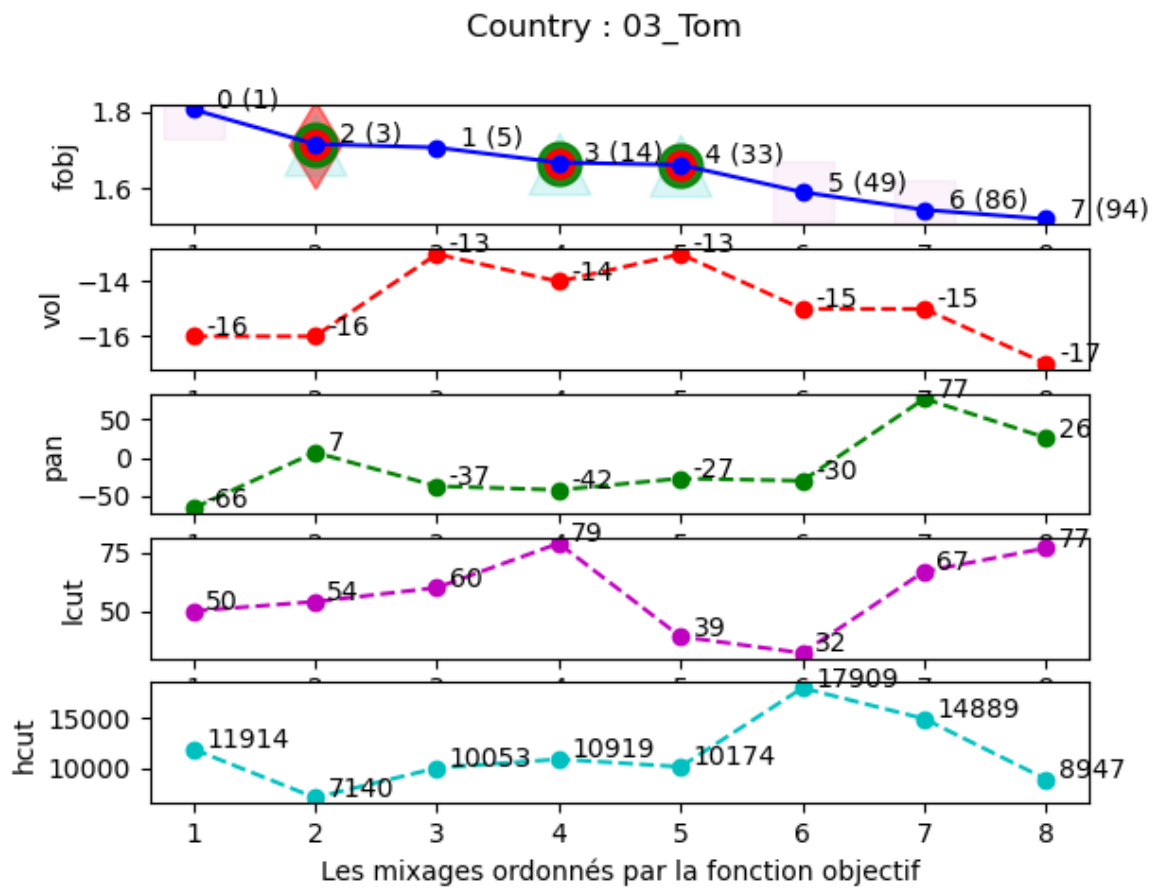


Figure E.46 – Country : Graphe de convergence – 03_Tom.

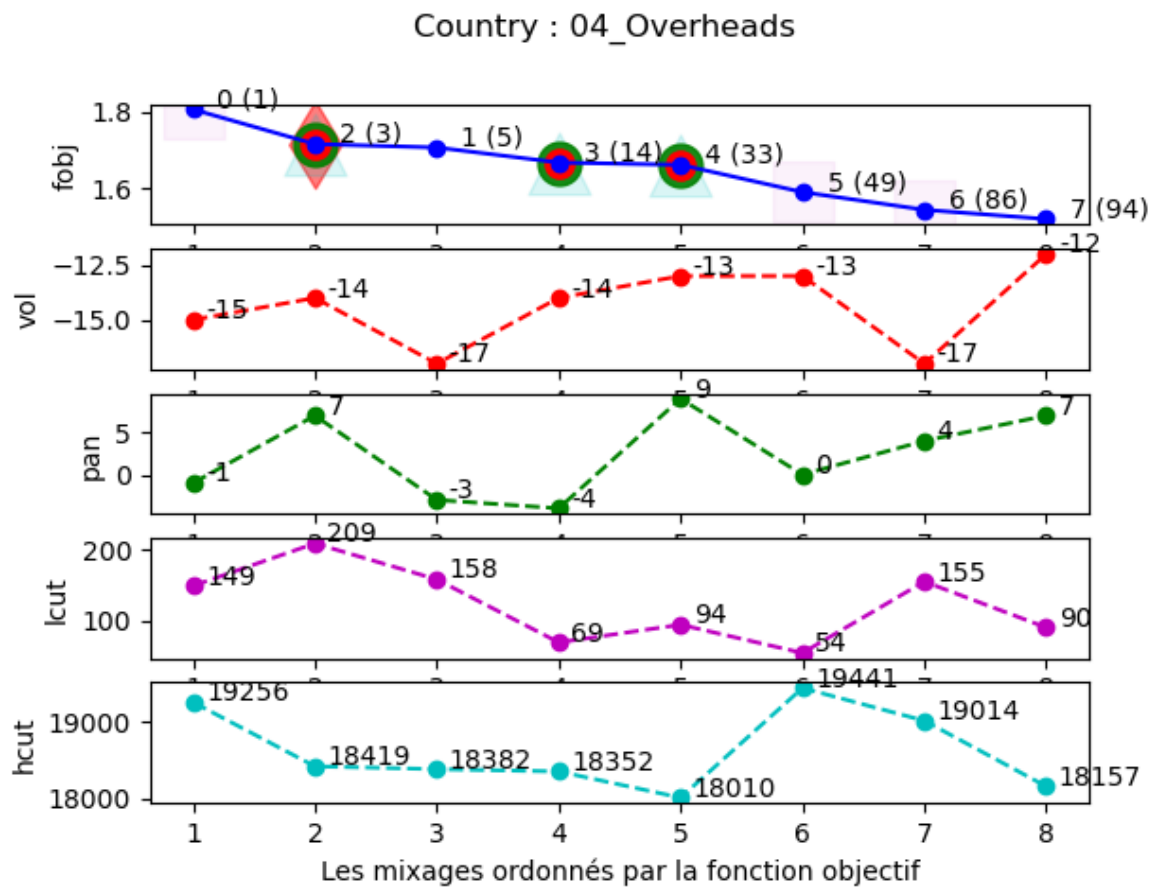


Figure E.47 – Country : Graphe de convergence – 04_Overheads.

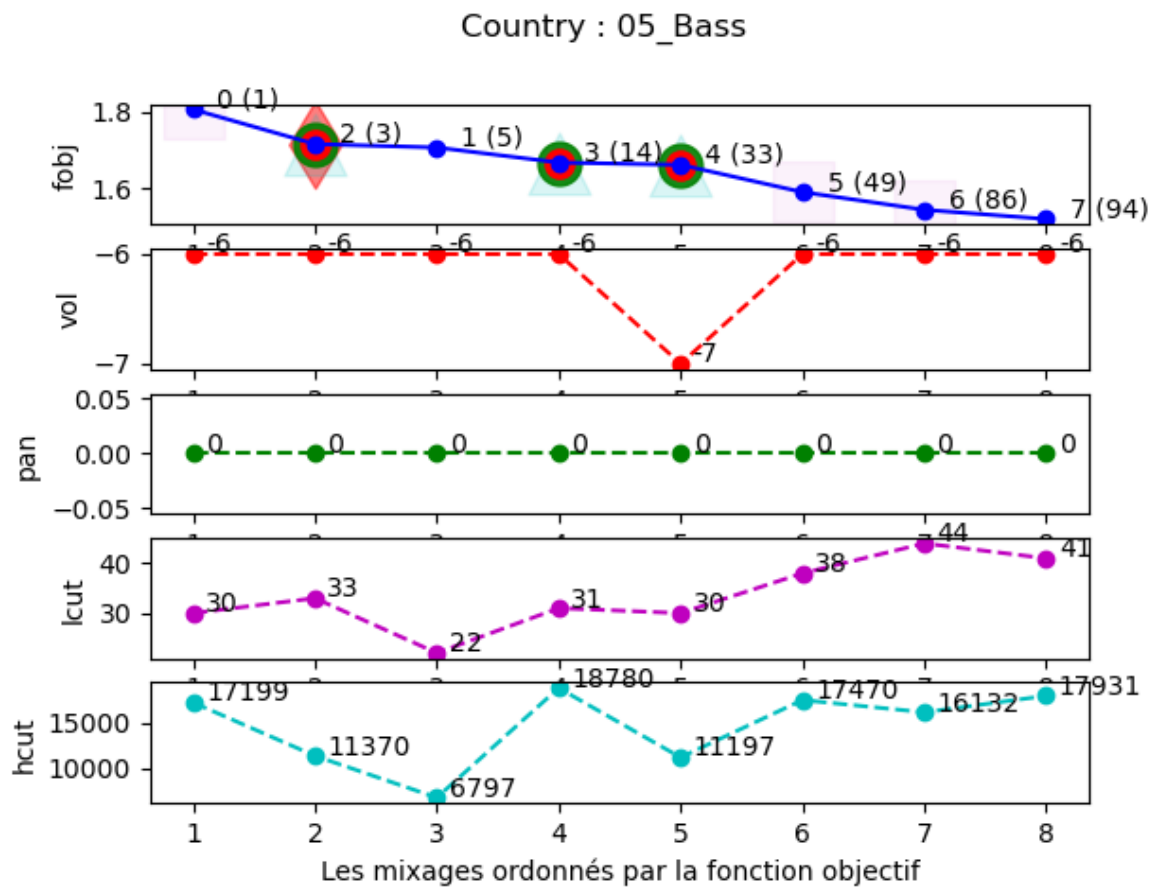


Figure E.48 – Country : Graphe de convergence – 05_Bass.

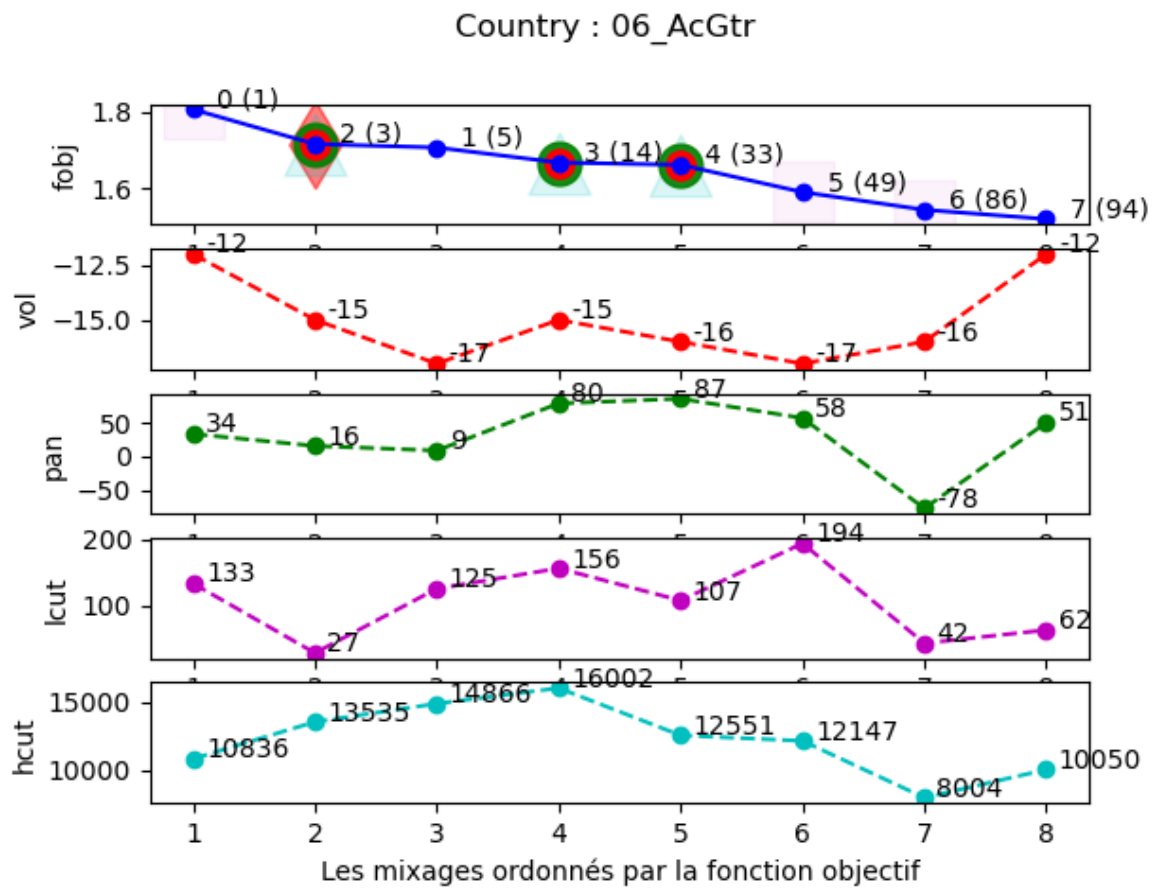


Figure E.49 – Country : Graphe de convergence – 06_AcGtr.

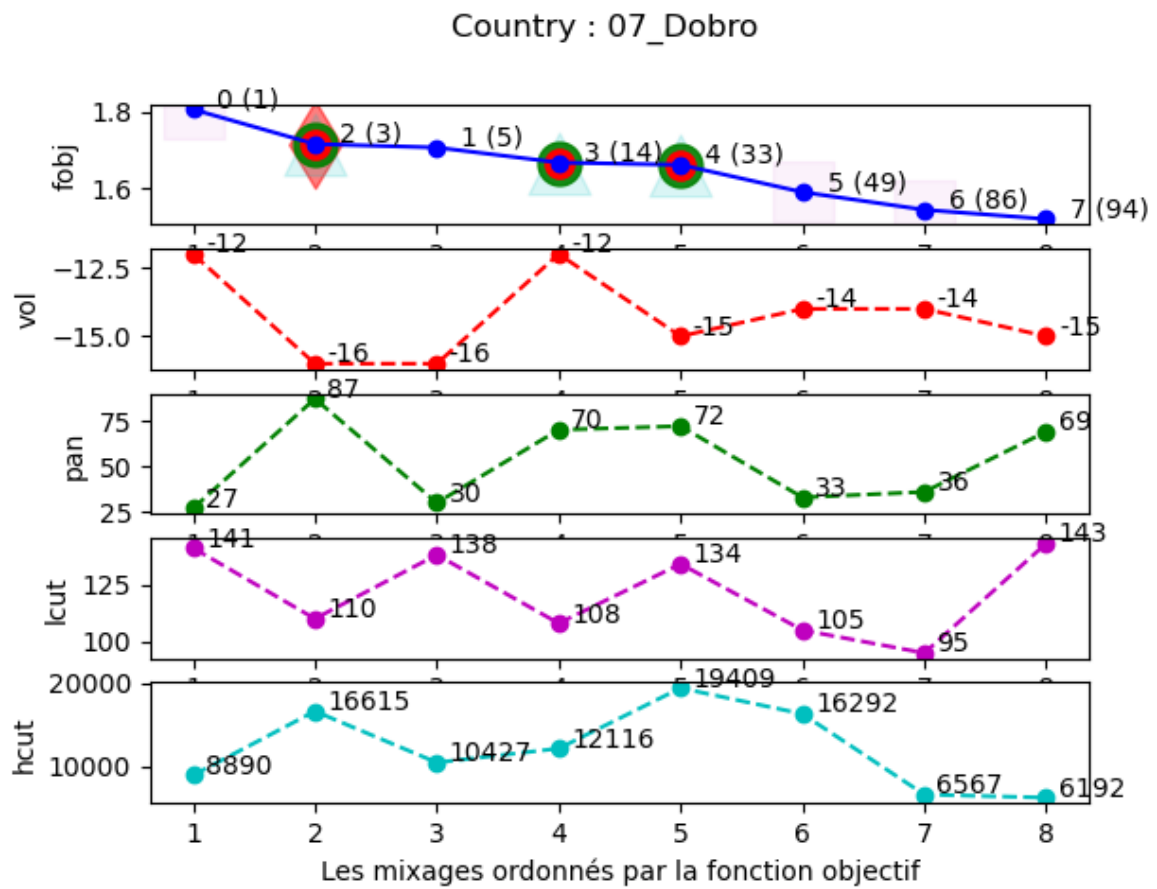


Figure E.50 – Country : Graphe de convergence – 07_Dobro.

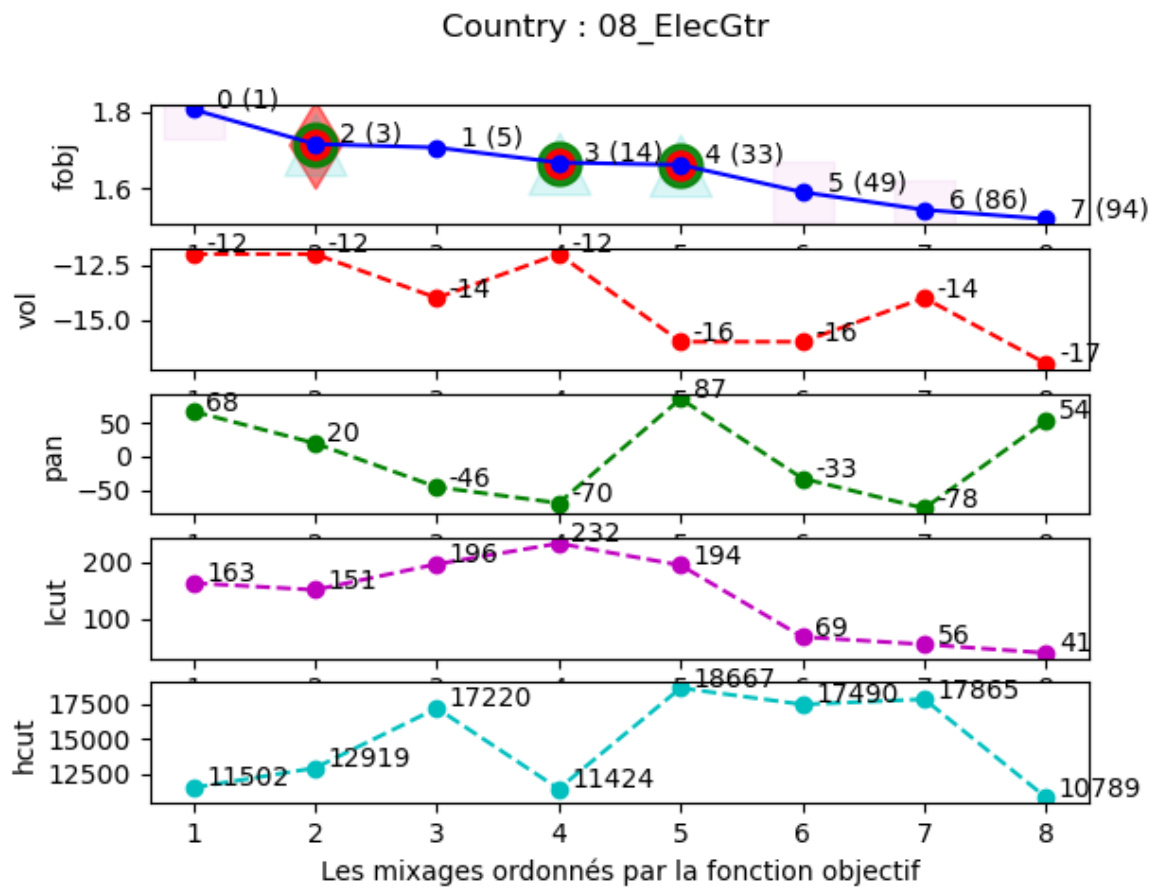


Figure E.51 – Country : Graphe de convergence – 08_ElecGtr.

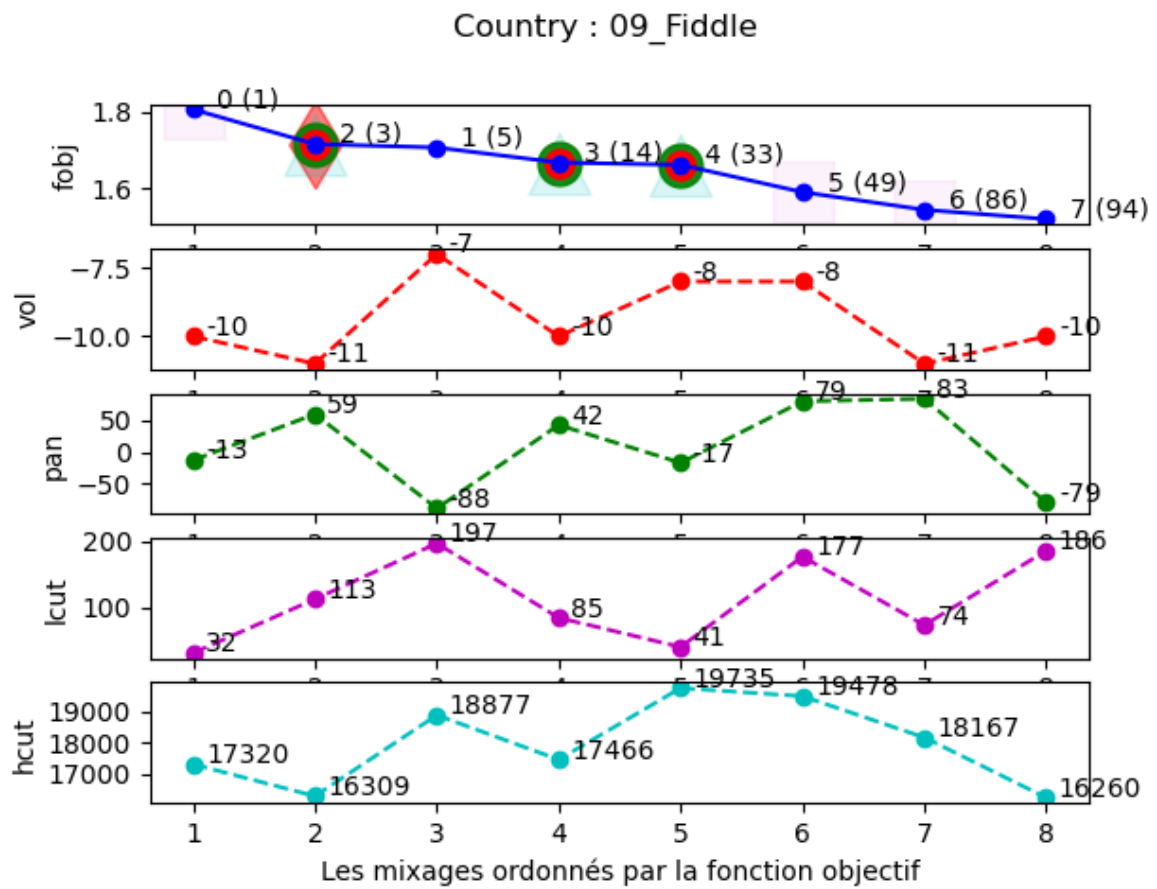


Figure E.52 – Country : Graphe de convergence – 09_Fiddle.

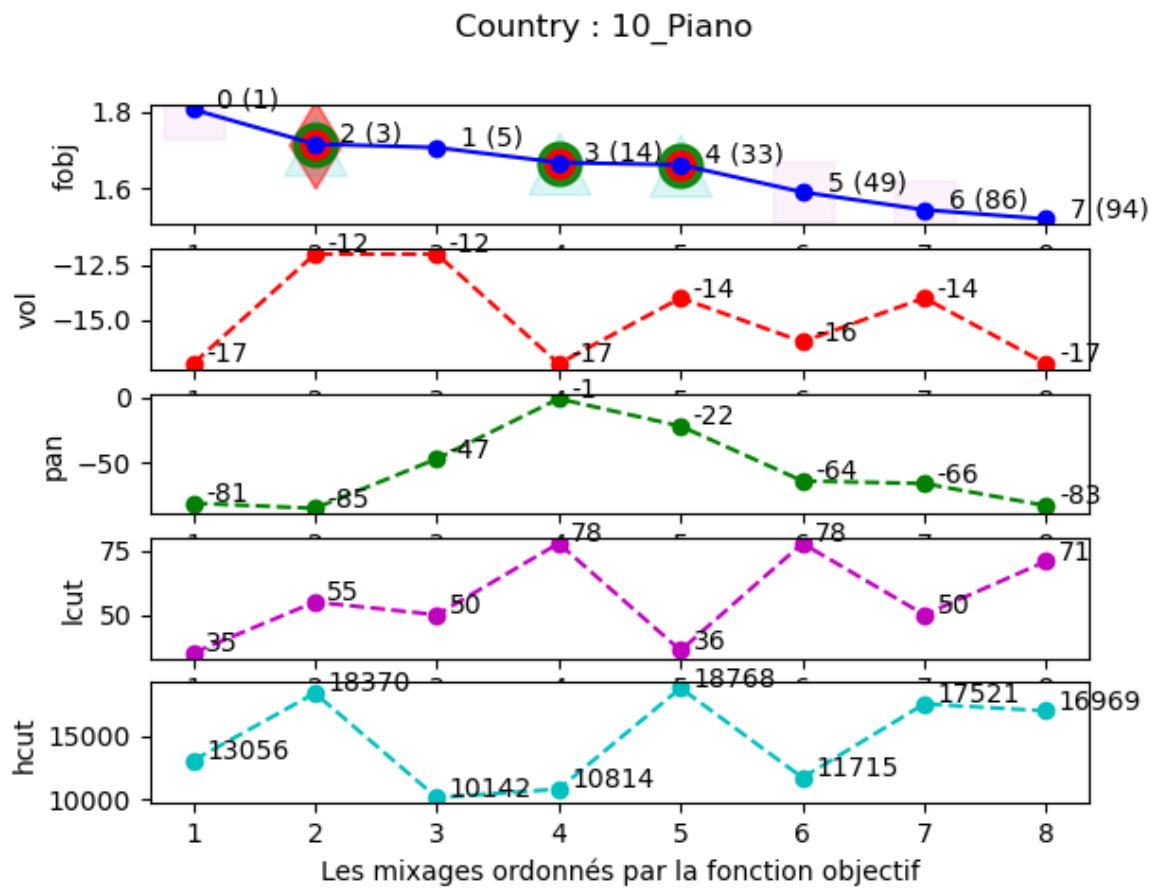


Figure E.53 – Country : Graphe de convergence – 10_Piano.

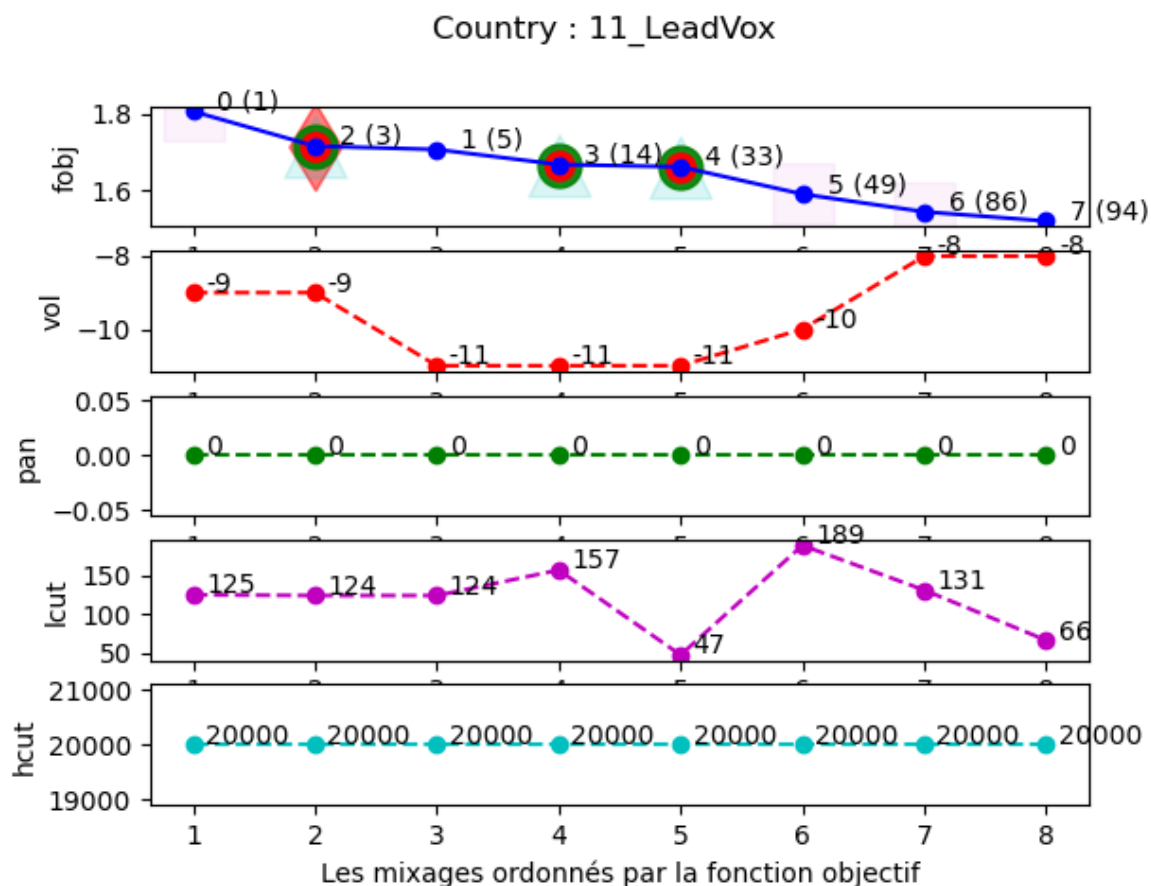


Figure E.54 – Country : Graphe de convergence – 11_LeadVox.

E.6 Choir : Don Camillo Choir - Marsh Marigolds Song : 12 tracks

E.6.1 Les règles de mixage

0th mix : fobj = 3.663

- | | |
|--|----------------------|
| 1: MixRules(vol=-21dB, pan=-6°, locut=51Hz, hicut=18842Hz) | 01_MainPair1 |
| 2: MixRules(vol=-16dB, pan=0°, locut=26Hz, hicut=18706Hz) | 02_MainPair2 |
| 3: MixRules(vol=-23dB, pan=-56°, locut=62Hz, hicut=18393Hz) | 03_SpotMic1_Tenors |
| 4: MixRules(vol=-23dB, pan=-60°, locut=56Hz, hicut=15106Hz) | 04_SpotMic2_Basses |
| 5: MixRules(vol=-22dB, pan=74°, locut=175Hz, hicut=16595Hz) | 05_SpotMic3_Sopranos |
| 6: MixRules(vol=-25dB, pan=43°, locut=51Hz, hicut=19640Hz) | 06_SpotMic4_Altos |
| 7: MixRules(vol=-13dB, pan=70°, locut=72Hz, hicut=18508Hz) | 07_BassCloseMic1 |
| 8: MixRules(vol=-16dB, pan=60°, locut=57Hz, hicut=13793Hz) | 08_BassCloseMic2 |
| 9: MixRules(vol=-15dB, pan=6°, locut=23Hz, hicut=14815Hz) | 09_BassCloseMic3 |
| 10: MixRules(vol=-16dB, pan=-5°, locut=23Hz, hicut=18897Hz) | 10_BassCloseMic4 |
| 11: MixRules(vol=-17dB, pan=-40°, locut=49Hz, hicut=10679Hz) | 11_BassCloseMic5 |

12: MixRules(vol=-14dB, pan=9°, locut=67Hz, hicut=20000Hz)	12_LeadVox
1th mix : fobj = 3.155	
1: MixRules(vol=-31dB, pan=8°, locut=22Hz, hicut=16943Hz)	01_MainPair1
2: MixRules(vol=-14dB, pan=-9°, locut=69Hz, hicut=18009Hz)	02_MainPair2
3: MixRules(vol=-33dB, pan=-52°, locut=53Hz, hicut=15450Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-18dB, pan=-79°, locut=56Hz, hicut=8061Hz)	04_SpotMic2_Basses
5: MixRules(vol=-35dB, pan=78°, locut=86Hz, hicut=18753Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-22dB, pan=33°, locut=149Hz, hicut=17399Hz)	06_SpotMic4_Altos
7: MixRules(vol=-14dB, pan=-84°, locut=24Hz, hicut=14362Hz)	07_BassCloseMic1
8: MixRules(vol=-12dB, pan=-8°, locut=70Hz, hicut=14953Hz)	08_BassCloseMic2
9: MixRules(vol=-14dB, pan=18°, locut=25Hz, hicut=13527Hz)	09_BassCloseMic3
10: MixRules(vol=-13dB, pan=44°, locut=35Hz, hicut=15714Hz)	10_BassCloseMic4
11: MixRules(vol=-14dB, pan=31°, locut=42Hz, hicut=13727Hz)	11_BassCloseMic5
12: MixRules(vol=-14dB, pan=1°, locut=50Hz, hicut=20000Hz)	12_LeadVox
2th mix : fobj = 3.271	
1: MixRules(vol=-22dB, pan=-6°, locut=20Hz, hicut=15780Hz)	01_MainPair1
2: MixRules(vol=-16dB, pan=0°, locut=71Hz, hicut=19359Hz)	02_MainPair2
3: MixRules(vol=-31dB, pan=-35°, locut=24Hz, hicut=15064Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-27dB, pan=-62°, locut=32Hz, hicut=9419Hz)	04_SpotMic2_Basses
5: MixRules(vol=-19dB, pan=79°, locut=152Hz, hicut=17569Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-31dB, pan=57°, locut=24Hz, hicut=16724Hz)	06_SpotMic4_Altos
7: MixRules(vol=-14dB, pan=58°, locut=27Hz, hicut=18842Hz)	07_BassCloseMic1
8: MixRules(vol=-13dB, pan=0°, locut=76Hz, hicut=17704Hz)	08_BassCloseMic2
9: MixRules(vol=-16dB, pan=-31°, locut=76Hz, hicut=15835Hz)	09_BassCloseMic3
10: MixRules(vol=-13dB, pan=-61°, locut=46Hz, hicut=19437Hz)	10_BassCloseMic4
11: MixRules(vol=-13dB, pan=-24°, locut=66Hz, hicut=14328Hz)	11_BassCloseMic5
12: MixRules(vol=-13dB, pan=8°, locut=52Hz, hicut=20000Hz)	12_LeadVox
3th mix : fobj = 3.234	
1: MixRules(vol=-24dB, pan=-4°, locut=54Hz, hicut=17883Hz)	01_MainPair1
2: MixRules(vol=-17dB, pan=-5°, locut=70Hz, hicut=17424Hz)	02_MainPair2
3: MixRules(vol=-20dB, pan=-31°, locut=82Hz, hicut=11818Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-34dB, pan=-66°, locut=55Hz, hicut=16455Hz)	04_SpotMic2_Basses
5: MixRules(vol=-35dB, pan=70°, locut=144Hz, hicut=15820Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-29dB, pan=40°, locut=37Hz, hicut=19528Hz)	06_SpotMic4_Altos
7: MixRules(vol=-13dB, pan=0°, locut=67Hz, hicut=10239Hz)	07_BassCloseMic1
8: MixRules(vol=-12dB, pan=-16°, locut=51Hz, hicut=10745Hz)	08_BassCloseMic2
9: MixRules(vol=-12dB, pan=-86°, locut=54Hz, hicut=15233Hz)	09_BassCloseMic3
10: MixRules(vol=-14dB, pan=-56°, locut=30Hz, hicut=18531Hz)	10_BassCloseMic4
11: MixRules(vol=-14dB, pan=-23°, locut=27Hz, hicut=13289Hz)	11_BassCloseMic5
12: MixRules(vol=-12dB, pan=9°, locut=161Hz, hicut=20000Hz)	12_LeadVox
4th mix : fobj = 3.188	
1: MixRules(vol=-29dB, pan=-6°, locut=31Hz, hicut=15780Hz)	01_MainPair1
2: MixRules(vol=-15dB, pan=5°, locut=46Hz, hicut=18451Hz)	02_MainPair2
3: MixRules(vol=-31dB, pan=-59°, locut=66Hz, hicut=13788Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-20dB, pan=-84°, locut=24Hz, hicut=15737Hz)	04_SpotMic2_Basses
5: MixRules(vol=-34dB, pan=69°, locut=69Hz, hicut=17533Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-31dB, pan=37°, locut=87Hz, hicut=16899Hz)	06_SpotMic4_Altos
7: MixRules(vol=-15dB, pan=-5°, locut=74Hz, hicut=12653Hz)	07_BassCloseMic1
8: MixRules(vol=-15dB, pan=78°, locut=69Hz, hicut=14347Hz)	08_BassCloseMic2
9: MixRules(vol=-12dB, pan=-79°, locut=23Hz, hicut=10807Hz)	09_BassCloseMic3
10: MixRules(vol=-12dB, pan=41°, locut=48Hz, hicut=16370Hz)	10_BassCloseMic4

11: MixRules(vol=-14dB, pan=0°, locut=22Hz, hicut=12883Hz)	11_BassCloseMic5
12: MixRules(vol=-17dB, pan=8°, locut=99Hz, hicut=20000Hz)	12_LeadVox
5th mix : fobj = 3.125	
1: MixRules(vol=-31dB, pan=3°, locut=29Hz, hicut=11764Hz)	01_MainPair1
2: MixRules(vol=-13dB, pan=9°, locut=59Hz, hicut=19709Hz)	02_MainPair2
3: MixRules(vol=-25dB, pan=-43°, locut=78Hz, hicut=10940Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-32dB, pan=-79°, locut=49Hz, hicut=13100Hz)	04_SpotMic2_Basses
5: MixRules(vol=-31dB, pan=85°, locut=111Hz, hicut=15790Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-32dB, pan=32°, locut=114Hz, hicut=17027Hz)	06_SpotMic4_Altos
7: MixRules(vol=-17dB, pan=0°, locut=79Hz, hicut=14174Hz)	07_BassCloseMic1
8: MixRules(vol=-12dB, pan=82°, locut=33Hz, hicut=10293Hz)	08_BassCloseMic2
9: MixRules(vol=-16dB, pan=35°, locut=67Hz, hicut=18592Hz)	09_BassCloseMic3
10: MixRules(vol=-12dB, pan=-34°, locut=71Hz, hicut=15354Hz)	10_BassCloseMic4
11: MixRules(vol=-12dB, pan=-12°, locut=32Hz, hicut=10763Hz)	11_BassCloseMic5
12: MixRules(vol=-17dB, pan=2°, locut=43Hz, hicut=20000Hz)	12_LeadVox
6th mix : fobj = 3.057	
1: MixRules(vol=-22dB, pan=6°, locut=29Hz, hicut=19083Hz)	01_MainPair1
2: MixRules(vol=-13dB, pan=6°, locut=75Hz, hicut=18208Hz)	02_MainPair2
3: MixRules(vol=-26dB, pan=-31°, locut=24Hz, hicut=13515Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-34dB, pan=-78°, locut=53Hz, hicut=6806Hz)	04_SpotMic2_Basses
5: MixRules(vol=-32dB, pan=82°, locut=26Hz, hicut=16850Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-27dB, pan=53°, locut=139Hz, hicut=17550Hz)	06_SpotMic4_Altos
7: MixRules(vol=-13dB, pan=-78°, locut=50Hz, hicut=15708Hz)	07_BassCloseMic1
8: MixRules(vol=-16dB, pan=-8°, locut=79Hz, hicut=16239Hz)	08_BassCloseMic2
9: MixRules(vol=-13dB, pan=-7°, locut=49Hz, hicut=10580Hz)	09_BassCloseMic3
10: MixRules(vol=-15dB, pan=29°, locut=54Hz, hicut=17442Hz)	10_BassCloseMic4
11: MixRules(vol=-16dB, pan=0°, locut=22Hz, hicut=17372Hz)	11_BassCloseMic5
12: MixRules(vol=-13dB, pan=7°, locut=84Hz, hicut=20000Hz)	12_LeadVox
7th mix : fobj = 2.673	
1: MixRules(vol=-31dB, pan=-2°, locut=40Hz, hicut=11435Hz)	01_MainPair1
2: MixRules(vol=-17dB, pan=1°, locut=39Hz, hicut=19238Hz)	02_MainPair2
3: MixRules(vol=-20dB, pan=-54°, locut=71Hz, hicut=17082Hz)	03_SpotMic1_Tenors
4: MixRules(vol=-35dB, pan=-67°, locut=47Hz, hicut=10104Hz)	04_SpotMic2_Basses
5: MixRules(vol=-33dB, pan=85°, locut=190Hz, hicut=17527Hz)	05_SpotMic3_Sopranos
6: MixRules(vol=-33dB, pan=38°, locut=147Hz, hicut=19375Hz)	06_SpotMic4_Altos
7: MixRules(vol=-15dB, pan=-12°, locut=28Hz, hicut=16283Hz)	07_BassCloseMic1
8: MixRules(vol=-17dB, pan=-83°, locut=63Hz, hicut=12596Hz)	08_BassCloseMic2
9: MixRules(vol=-12dB, pan=12°, locut=32Hz, hicut=13950Hz)	09_BassCloseMic3
10: MixRules(vol=-12dB, pan=24°, locut=43Hz, hicut=17710Hz)	10_BassCloseMic4
11: MixRules(vol=-15dB, pan=52°, locut=27Hz, hicut=15872Hz)	11_BassCloseMic5
12: MixRules(vol=-14dB, pan=0°, locut=148Hz, hicut=20000Hz)	12_LeadVox

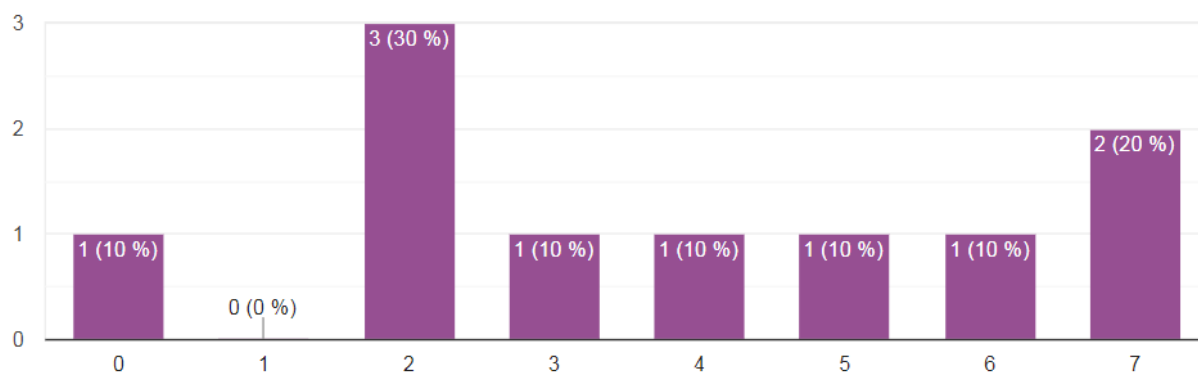


Figure E.55 – Choir : Le pré-mixage préféré.

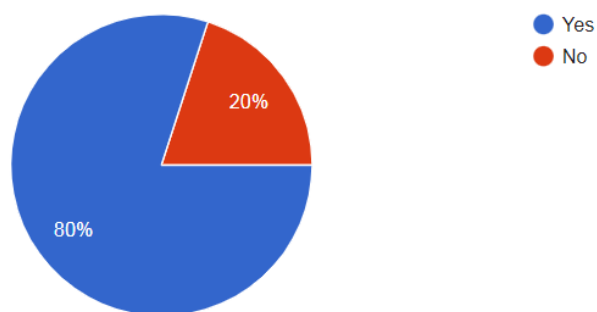


Figure E.56 – Choir : L'intention d'utiliser le pré-mixage préféré.

E.6.2 Les graphes de convergence

La fonction objectif

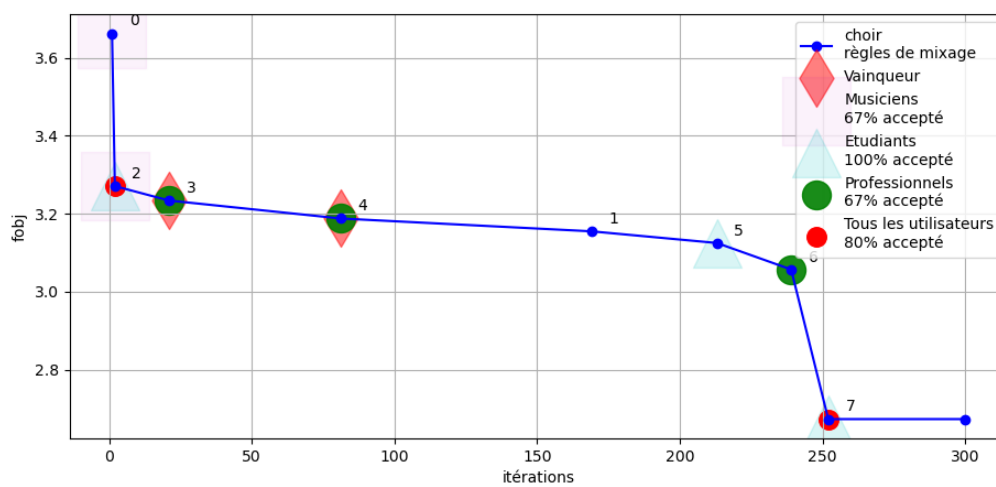


Figure E.57 – Choir : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

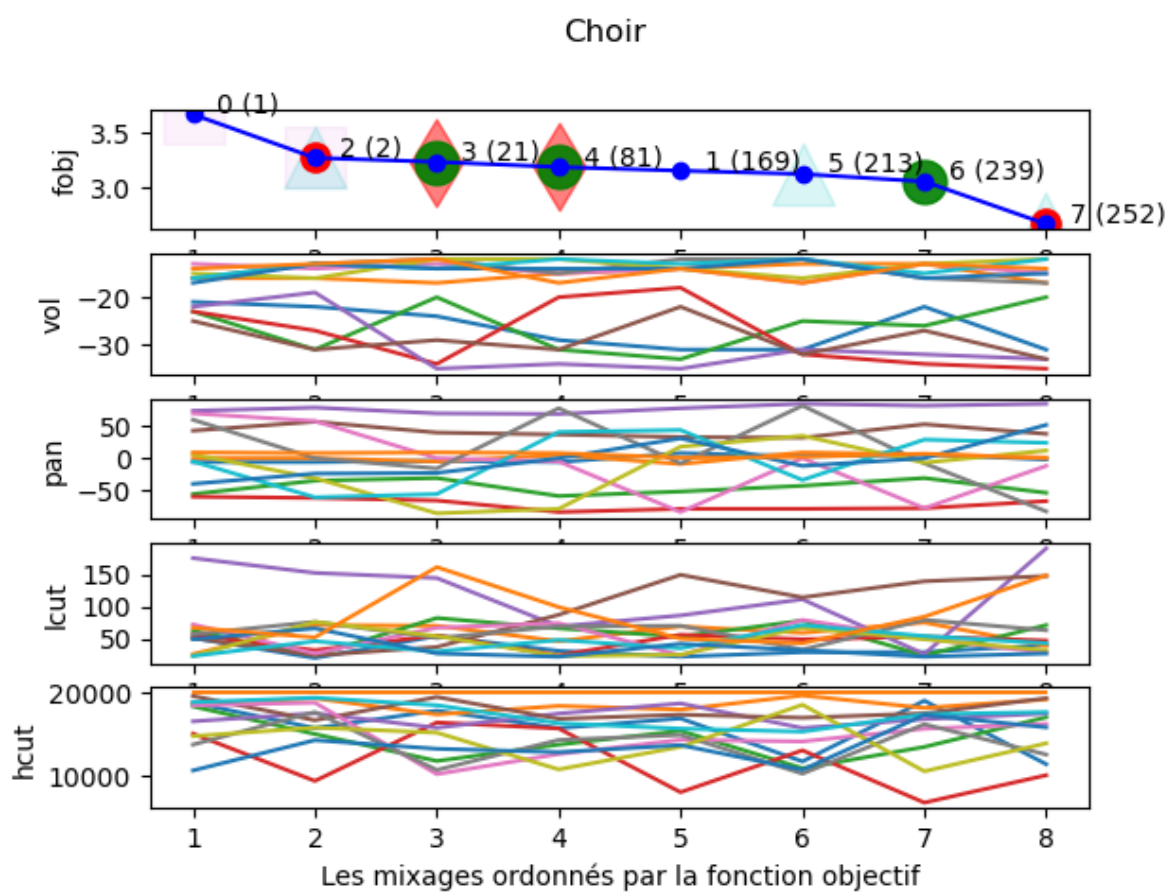


Figure E.58 – Choir : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

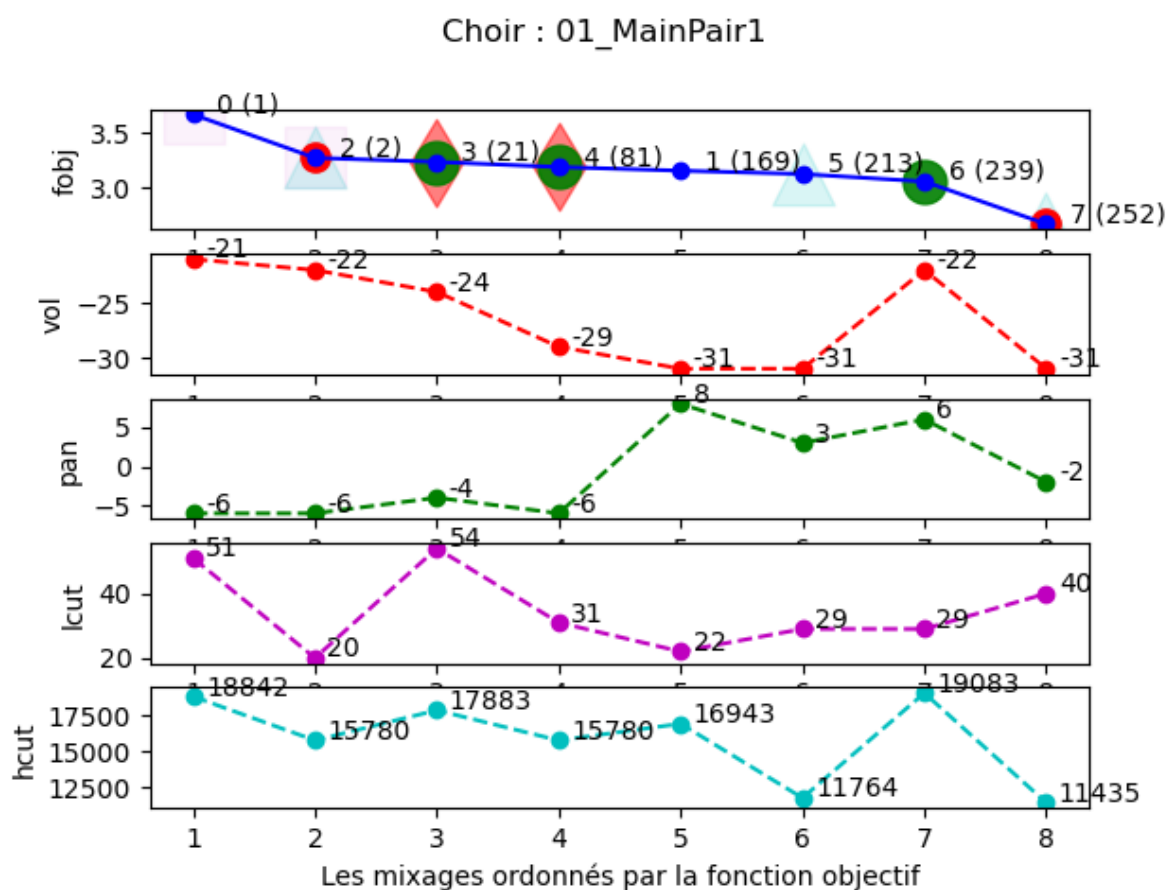


Figure E.59 – Choir : Graphe de convergence – 01_MainPair1.

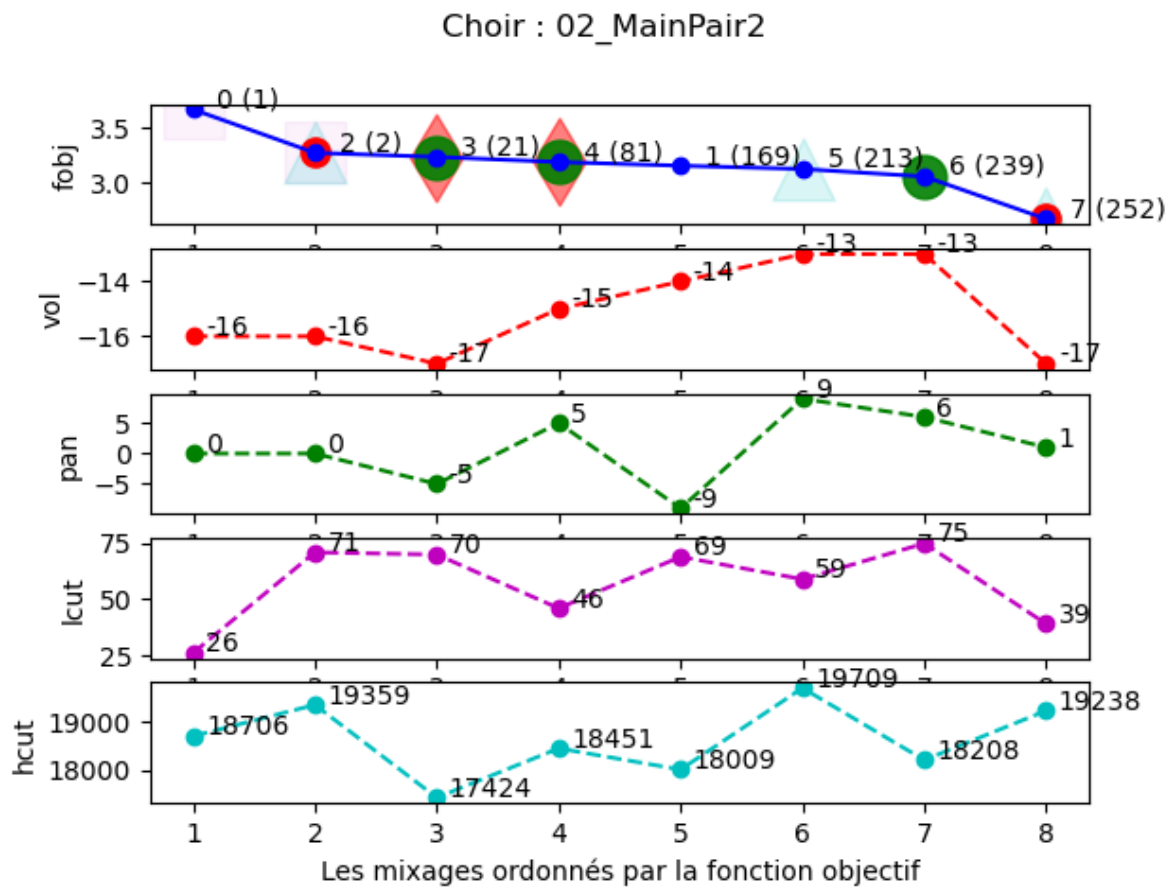


Figure E.60 – Choir : Graphe de convergence – 02_MainPair2.

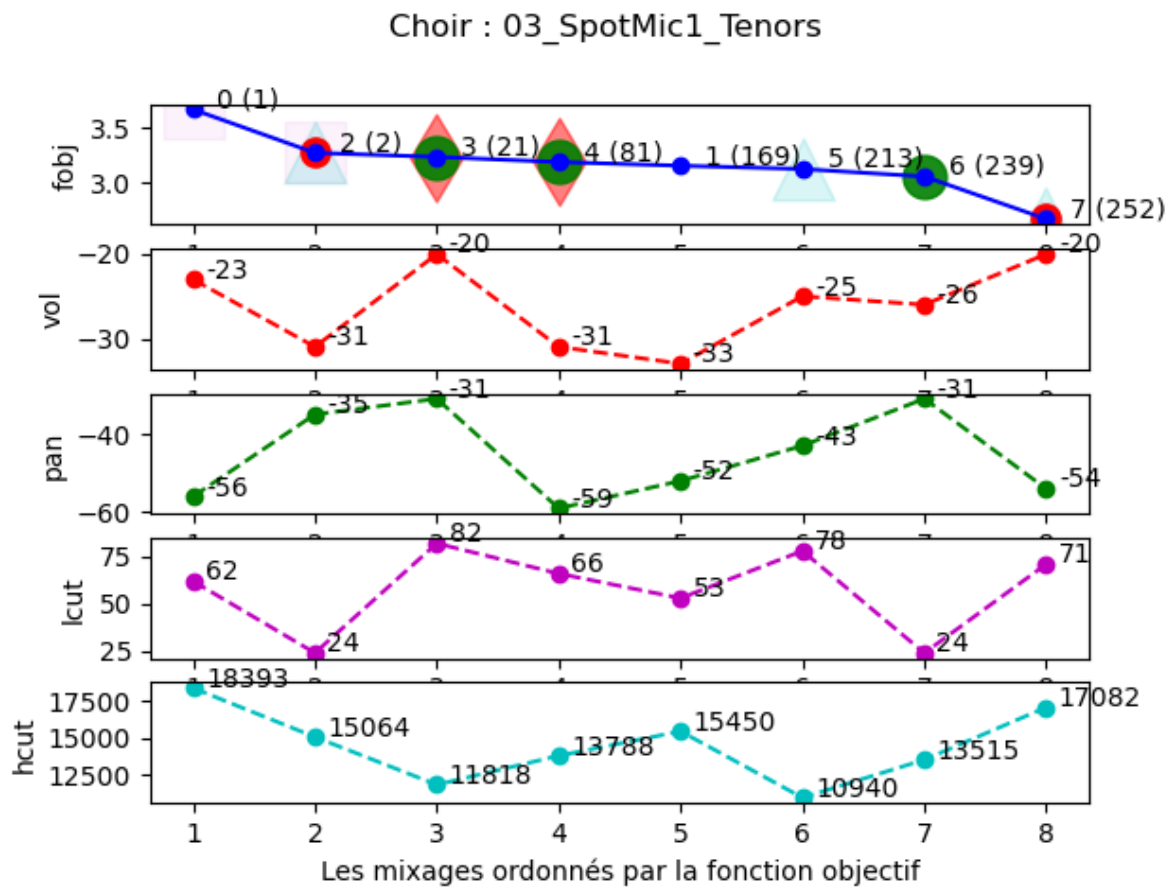


Figure E.61 – Choir : Graphe de convergence – 03_SpotMic1_Tenors.

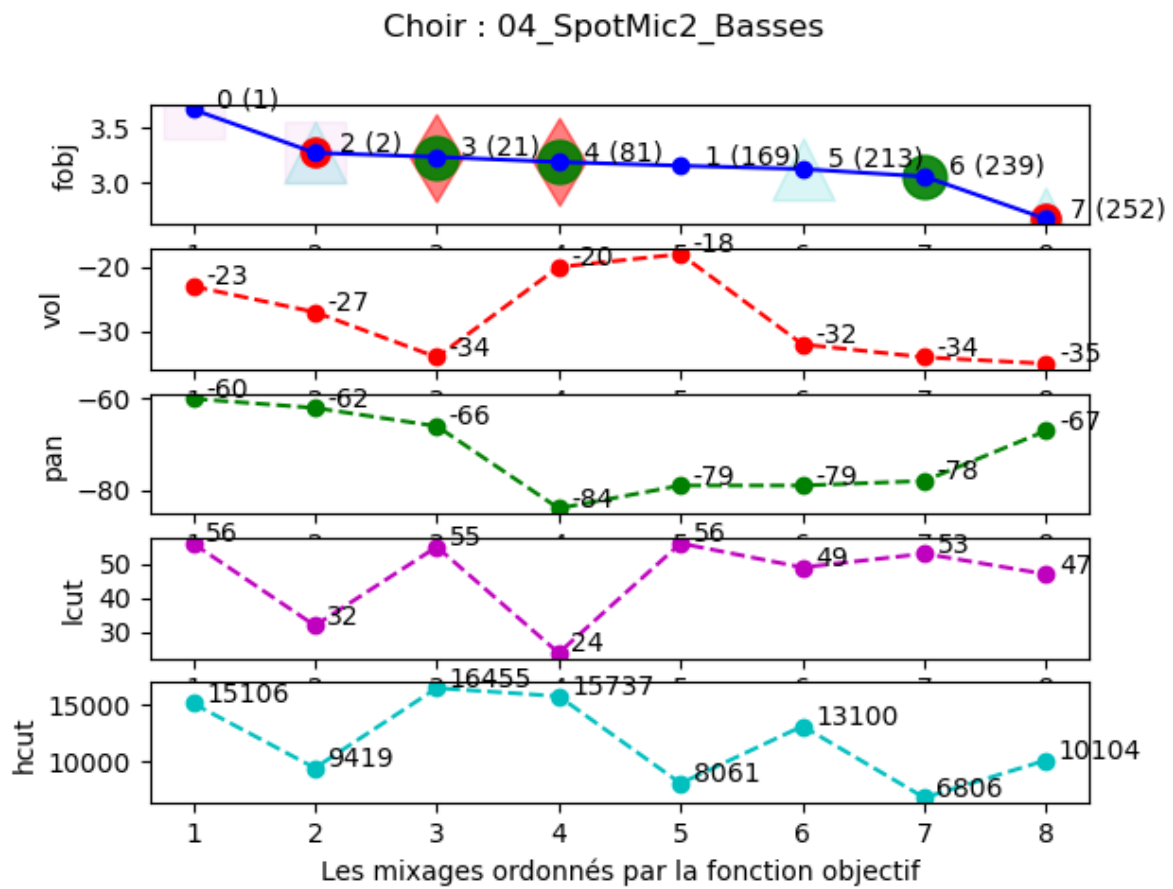


Figure E.62 – Choir : Graphe de convergence – 04_SpotMic2_Basses.

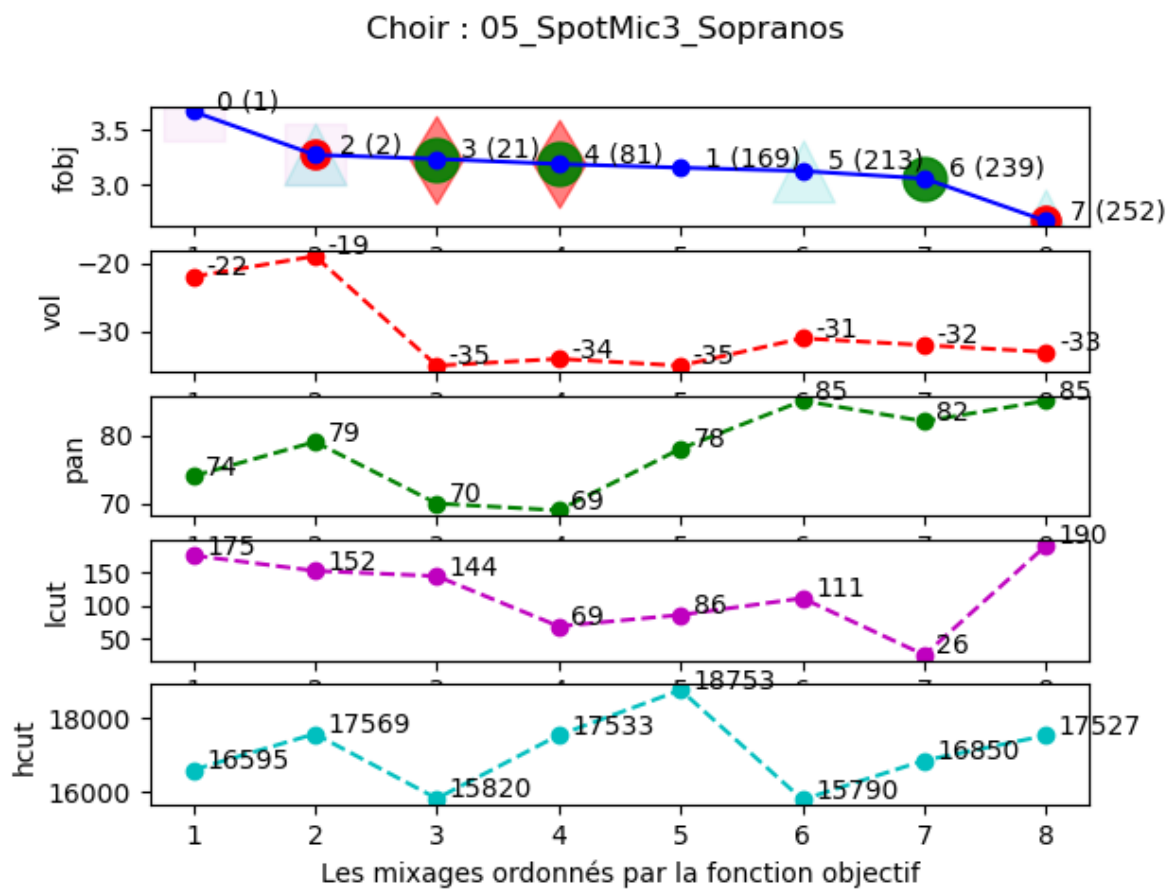


Figure E.63 – Choir : Graphe de convergence – 05_SpotMic3_Sopranos.

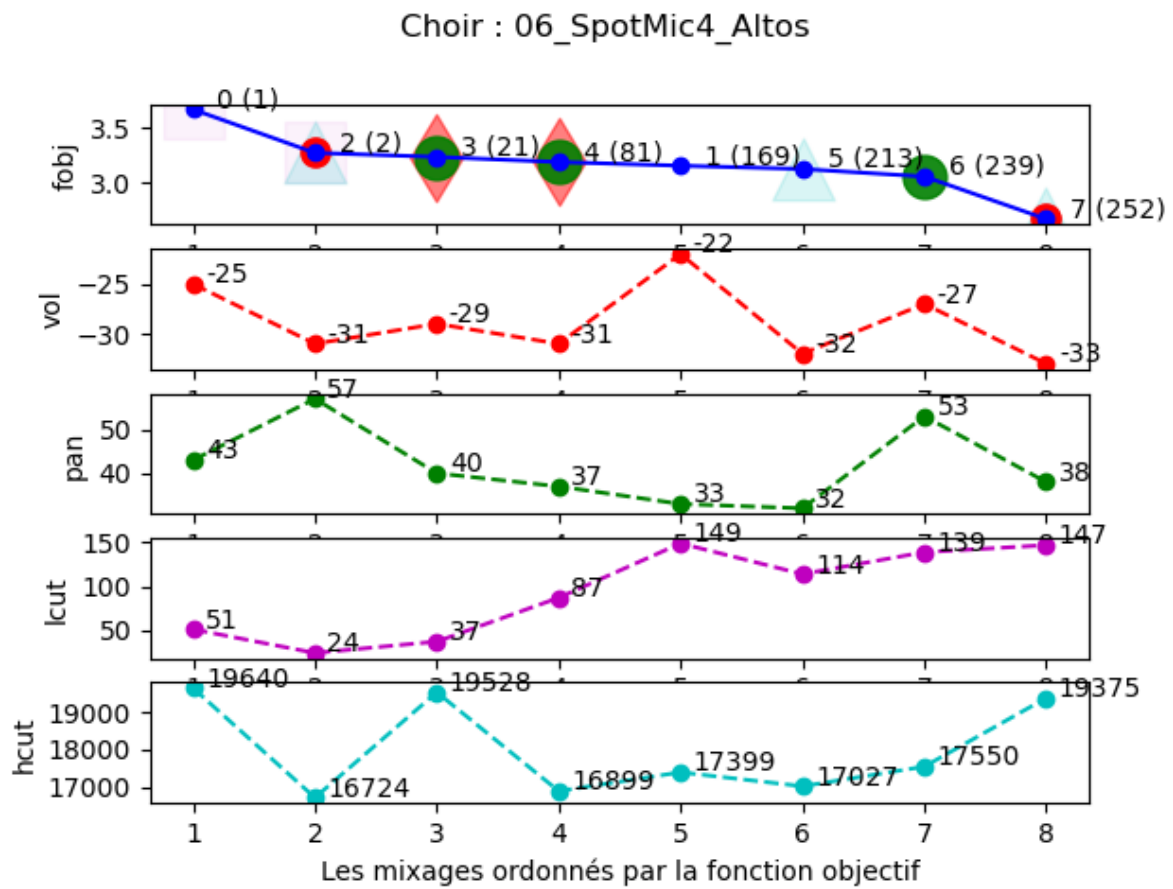


Figure E.64 – Choir : Graphe de convergence – 06_SpotMic4_Altos.

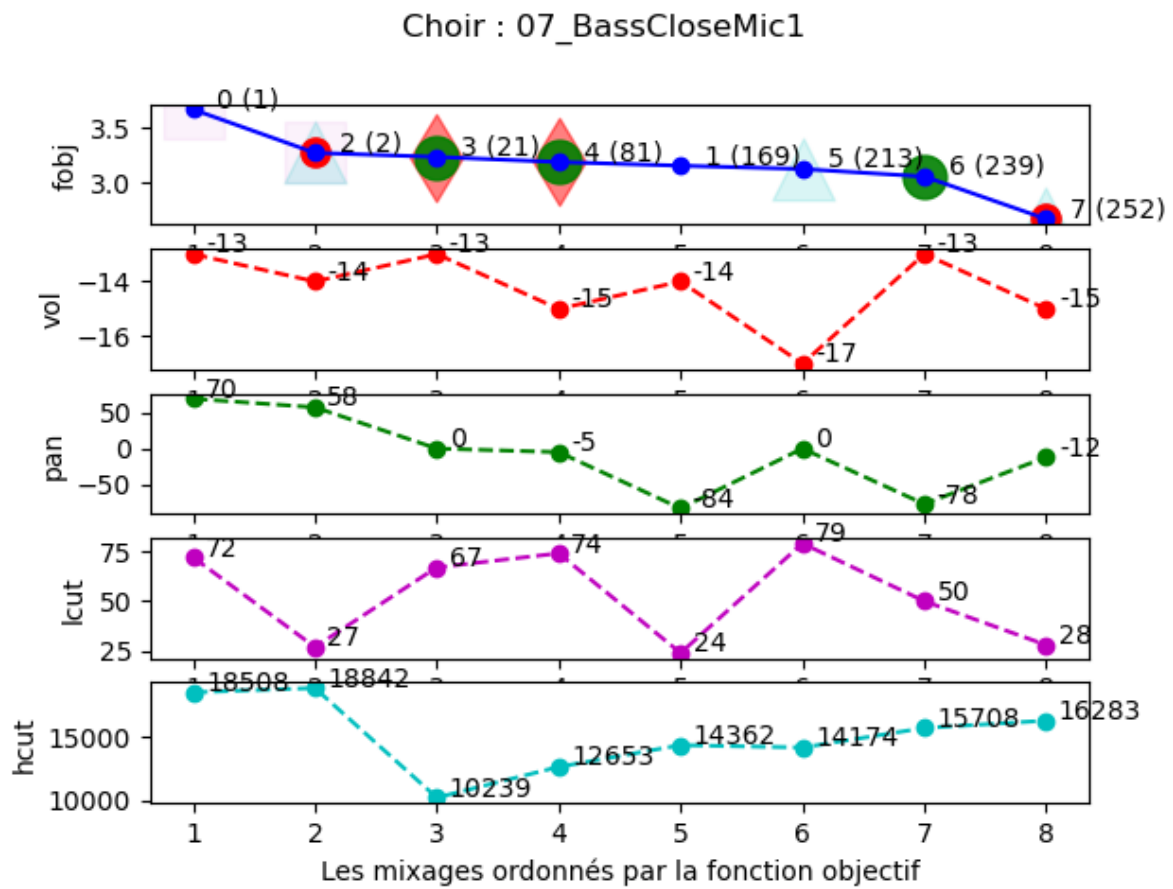


Figure E.65 – Choir : Graphe de convergence – 07_BassCloseMic1.

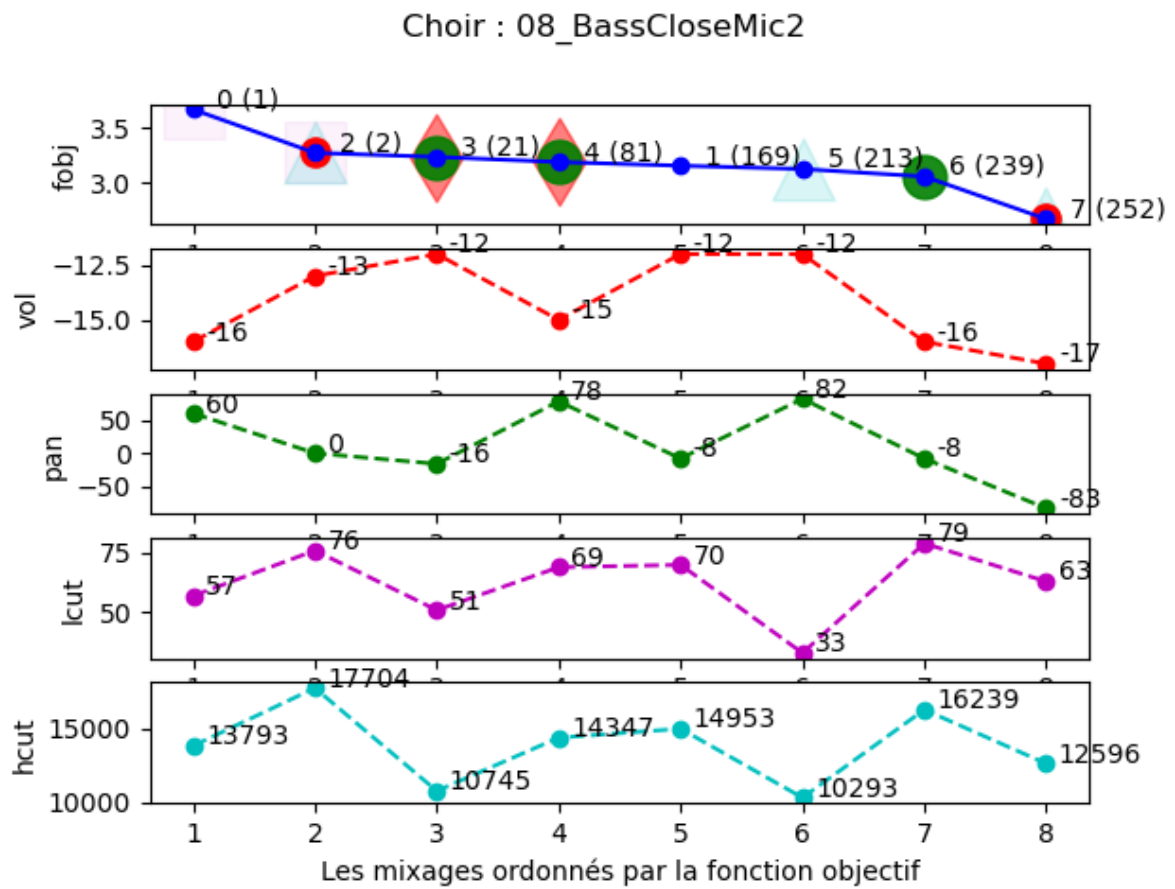


Figure E.66 – Choir : Graphe de convergence – 08_BassCloseMic2.

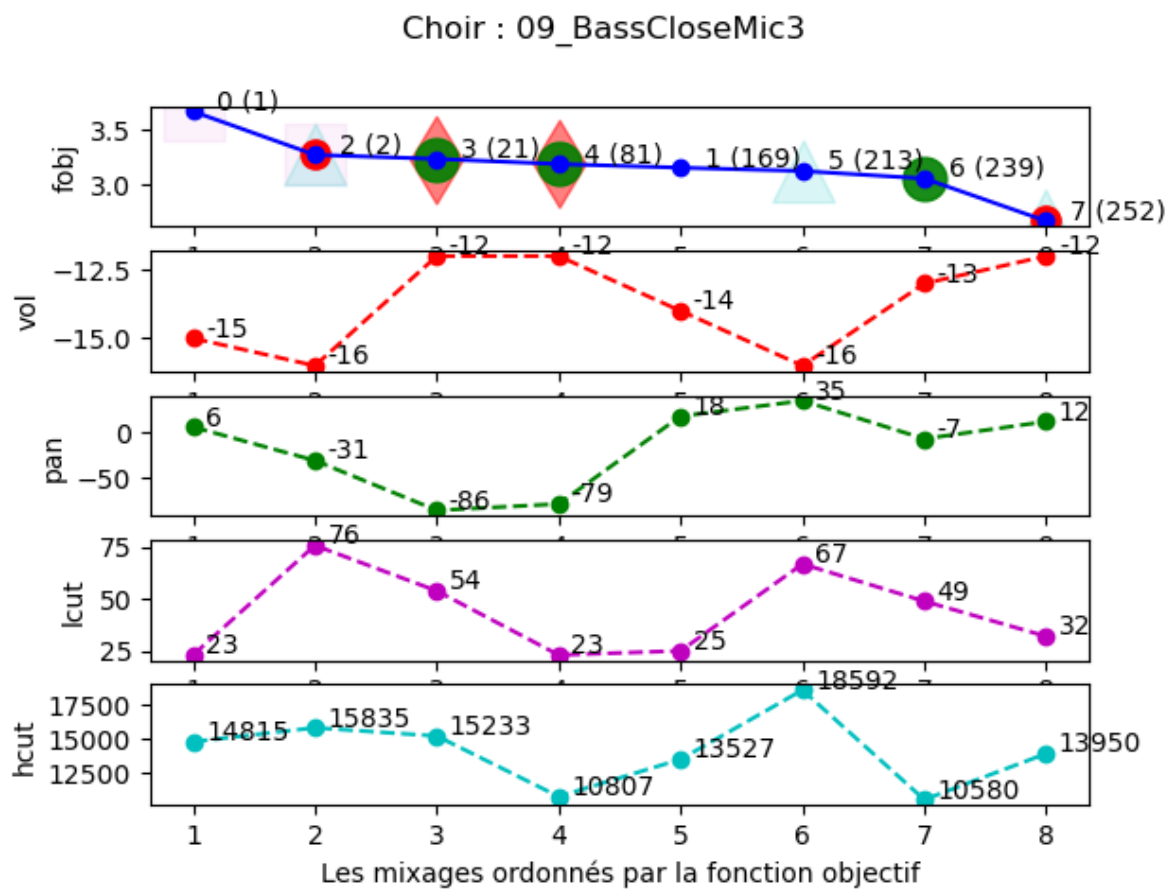


Figure E.67 – Choir : Graphe de convergence – 09_BassCloseMic3.

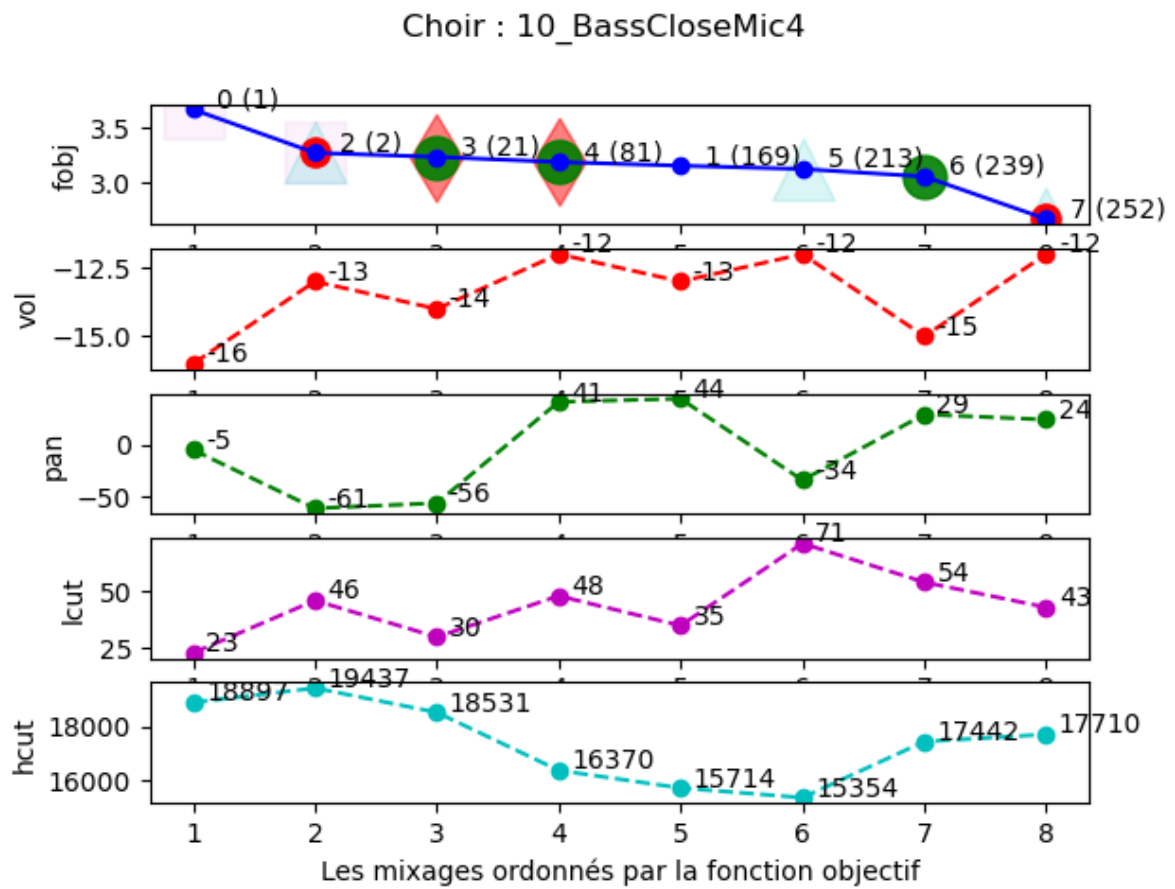


Figure E.68 – Choir : Graphe de convergence – 10_BassCloseMic4.

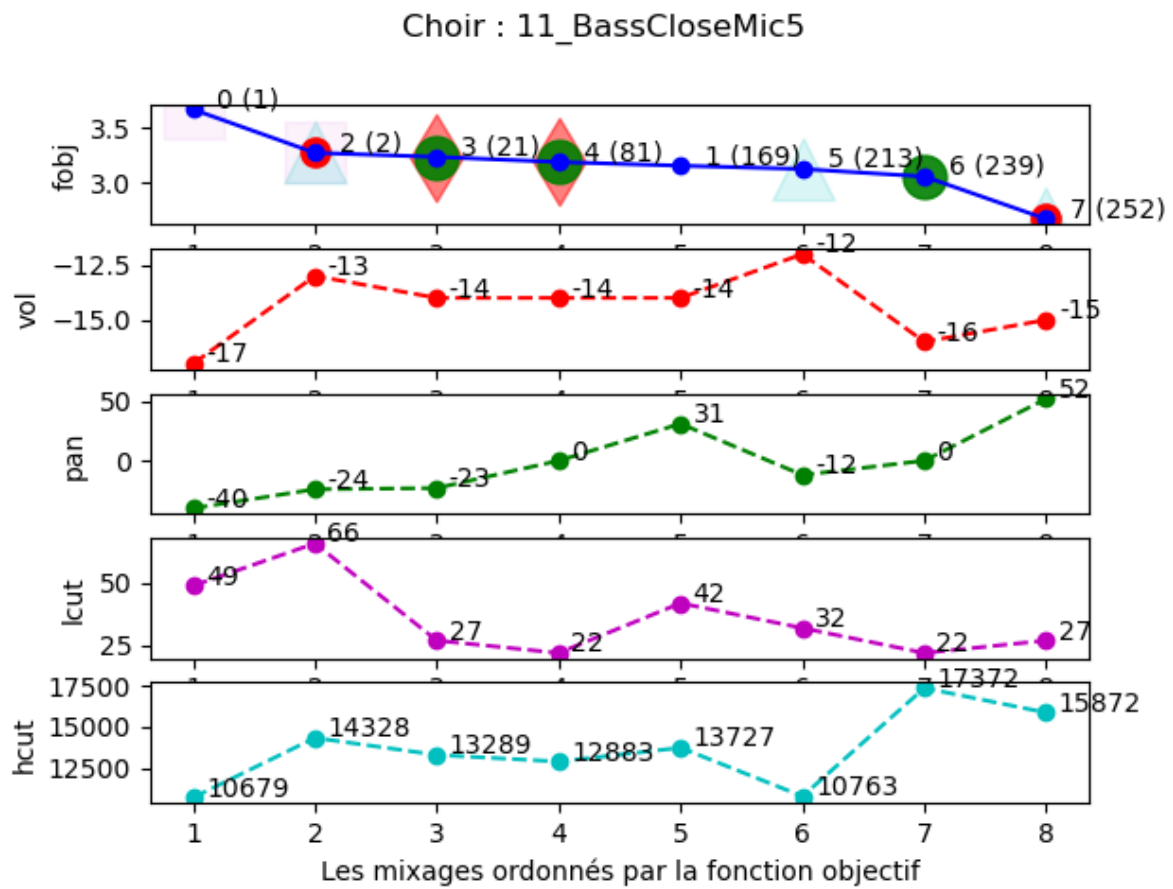


Figure E.69 – Choir : Graphe de convergence – 11_BassCloseMic5.

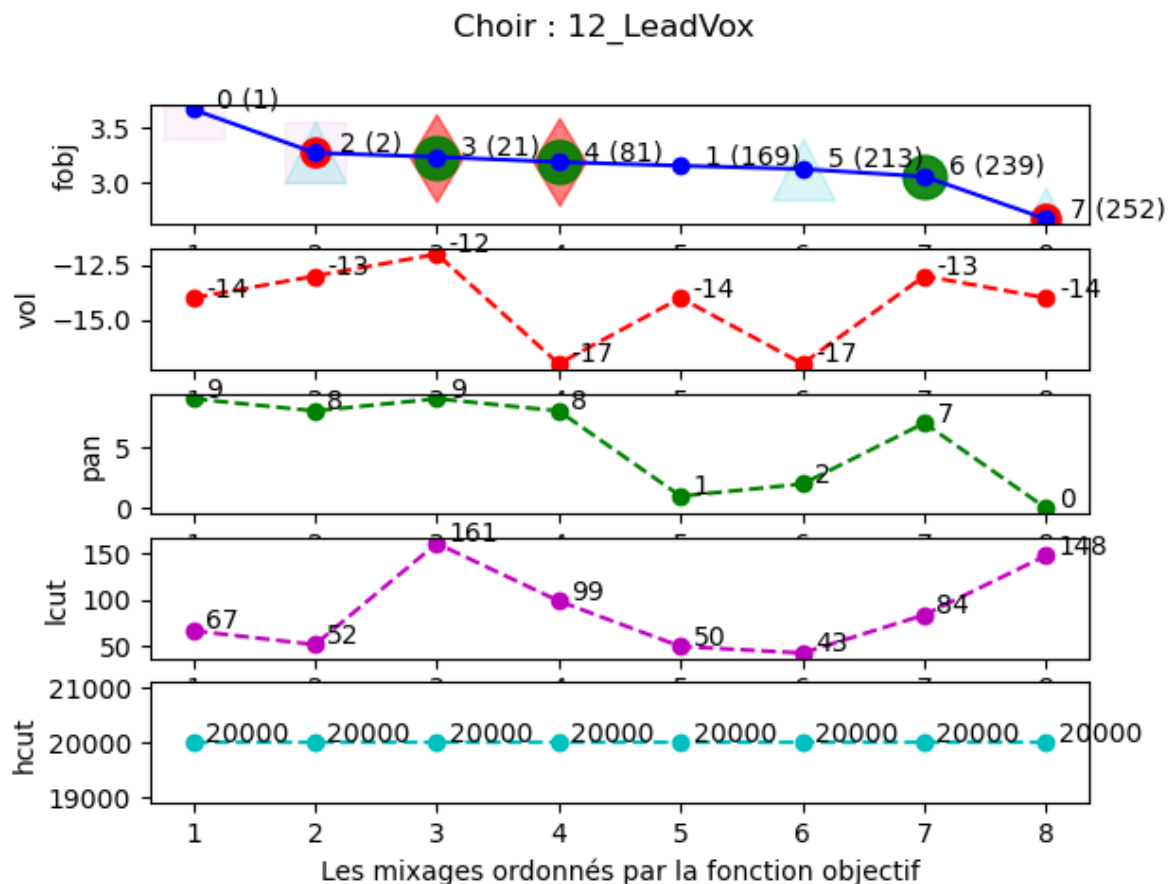


Figure E.70 – Choir : Graphe de convergence – 12_LeadVox.

E.7 Jazz : Araujo - The Saga Of Harrison Crabfeathers : 13 tracks

E.7.1 Les règles de mixage

0th mix : fobj = 2.336

- | | |
|--|----------------|
| 1: MixRules(vol=-11dB, pan=0°, locut=39Hz, hicut=10559Hz) | 01_KickIn |
| 2: MixRules(vol=-10dB, pan=0°, locut=36Hz, hicut=15439Hz) | 02_KickOut |
| 3: MixRules(vol=-6dB, pan=0°, locut=62Hz, hicut=19108Hz) | 03_Snare |
| 4: MixRules(vol=-16dB, pan=40°, locut=78Hz, hicut=18689Hz) | 04_Tom1 |
| 5: MixRules(vol=-12dB, pan=73°, locut=29Hz, hicut=16849Hz) | 05_Tom2 |
| 6: MixRules(vol=-16dB, pan=-84°, locut=192Hz, hicut=18376Hz) | 06_Overhead1 |
| 7: MixRules(vol=-13dB, pan=86°, locut=288Hz, hicut=19201Hz) | 07_Overhead2 |
| 8: MixRules(vol=-8dB, pan=0°, locut=49Hz, hicut=13702Hz) | 08_BassDI |
| 9: MixRules(vol=-14dB, pan=-63°, locut=36Hz, hicut=12261Hz) | 10_ElecGtrMic1 |
| 10: MixRules(vol=-13dB, pan=30°, locut=54Hz, hicut=14657Hz) | 11_ElecGtrMic2 |
| 11: MixRules(vol=-12dB, pan=-9°, locut=24Hz, hicut=12130Hz) | 13_Piano |

1th mix : fobj = 1.963

1: MixRules(vol=-6dB, pan=0°, locut=23Hz, hicut=12827Hz)	01_KickIn
2: MixRules(vol=-7dB, pan=0°, locut=32Hz, hicut=17155Hz)	02_KickOut
3: MixRules(vol=-9dB, pan=0°, locut=67Hz, hicut=13619Hz)	03_Snare
4: MixRules(vol=-17dB, pan=51°, locut=83Hz, hicut=15311Hz)	04_Tom1
5: MixRules(vol=-14dB, pan=87°, locut=33Hz, hicut=18168Hz)	05_Tom2
6: MixRules(vol=-12dB, pan=-64°, locut=265Hz, hicut=16967Hz)	06_Overhead1
7: MixRules(vol=-17dB, pan=65°, locut=380Hz, hicut=19096Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=35Hz, hicut=11534Hz)	08_BassDI
9: MixRules(vol=-17dB, pan=19°, locut=175Hz, hicut=17638Hz)	10_ElecGtrMic1
10: MixRules(vol=-13dB, pan=-82°, locut=126Hz, hicut=8632Hz)	11_ElecGtrMic2
11: MixRules(vol=-14dB, pan=28°, locut=59Hz, hicut=18785Hz)	13_Piano

2th mix : fobj = 1.827

1: MixRules(vol=-7dB, pan=0°, locut=21Hz, hicut=6321Hz)	01_KickIn
2: MixRules(vol=-11dB, pan=0°, locut=24Hz, hicut=8489Hz)	02_KickOut
3: MixRules(vol=-7dB, pan=0°, locut=111Hz, hicut=12977Hz)	03_Snare
4: MixRules(vol=-12dB, pan=51°, locut=48Hz, hicut=10131Hz)	04_Tom1
5: MixRules(vol=-17dB, pan=86°, locut=36Hz, hicut=10203Hz)	05_Tom2
6: MixRules(vol=-12dB, pan=-65°, locut=182Hz, hicut=12222Hz)	06_Overhead1
7: MixRules(vol=-12dB, pan=84°, locut=74Hz, hicut=17537Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=49Hz, hicut=13032Hz)	08_BassDI
9: MixRules(vol=-14dB, pan=-86°, locut=152Hz, hicut=11339Hz)	10_ElecGtrMic1
10: MixRules(vol=-16dB, pan=62°, locut=61Hz, hicut=15207Hz)	11_ElecGtrMic2
11: MixRules(vol=-16dB, pan=31°, locut=34Hz, hicut=15283Hz)	13_Piano

3th mix : fobj = 1.759

1: MixRules(vol=-8dB, pan=0°, locut=22Hz, hicut=9003Hz)	01_KickIn
2: MixRules(vol=-7dB, pan=0°, locut=30Hz, hicut=9394Hz)	02_KickOut
3: MixRules(vol=-6dB, pan=0°, locut=95Hz, hicut=15581Hz)	03_Snare
4: MixRules(vol=-15dB, pan=32°, locut=97Hz, hicut=16785Hz)	04_Tom1
5: MixRules(vol=-12dB, pan=81°, locut=34Hz, hicut=14036Hz)	05_Tom2
6: MixRules(vol=-16dB, pan=-79°, locut=132Hz, hicut=19596Hz)	06_Overhead1
7: MixRules(vol=-17dB, pan=65°, locut=170Hz, hicut=12771Hz)	07_Overhead2
8: MixRules(vol=-9dB, pan=0°, locut=20Hz, hicut=10316Hz)	08_BassDI
9: MixRules(vol=-17dB, pan=-66°, locut=210Hz, hicut=12075Hz)	10_ElecGtrMic1
10: MixRules(vol=-15dB, pan=79°, locut=90Hz, hicut=8853Hz)	11_ElecGtrMic2
11: MixRules(vol=-14dB, pan=-70°, locut=97Hz, hicut=13548Hz)	13_Piano

4th mix : fobj = 1.782

1: MixRules(vol=-7dB, pan=0°, locut=35Hz, hicut=4871Hz)	01_KickIn
2: MixRules(vol=-9dB, pan=0°, locut=34Hz, hicut=13964Hz)	02_KickOut
3: MixRules(vol=-8dB, pan=0°, locut=64Hz, hicut=18182Hz)	03_Snare
4: MixRules(vol=-17dB, pan=59°, locut=40Hz, hicut=11990Hz)	04_Tom1
5: MixRules(vol=-17dB, pan=78°, locut=44Hz, hicut=14421Hz)	05_Tom2
6: MixRules(vol=-13dB, pan=-63°, locut=215Hz, hicut=12330Hz)	06_Overhead1
7: MixRules(vol=-17dB, pan=81°, locut=128Hz, hicut=15634Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=42Hz, hicut=10963Hz)	08_BassDI
9: MixRules(vol=-17dB, pan=-80°, locut=70Hz, hicut=10774Hz)	10_ElecGtrMic1
10: MixRules(vol=-12dB, pan=35°, locut=44Hz, hicut=8219Hz)	11_ElecGtrMic2
11: MixRules(vol=-13dB, pan=-63°, locut=30Hz, hicut=16430Hz)	13_Piano

5th mix : fobj = 1.762

1: MixRules(vol=-8dB, pan=0°, locut=22Hz, hicut=10789Hz)	01_KickIn
--	-----------

2: MixRules(vol=-6dB, pan=0°, locut=28Hz, hicut=17711Hz)	02_KickOut
3: MixRules(vol=-6dB, pan=0°, locut=52Hz, hicut=16954Hz)	03_Snare
4: MixRules(vol=-16dB, pan=51°, locut=31Hz, hicut=12090Hz)	04_Tom1
5: MixRules(vol=-16dB, pan=61°, locut=46Hz, hicut=11500Hz)	05_Tom2
6: MixRules(vol=-14dB, pan=-66°, locut=74Hz, hicut=16249Hz)	06_Overhead1
7: MixRules(vol=-12dB, pan=64°, locut=38Hz, hicut=19237Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=20Hz, hicut=10570Hz)	08_BassDI
9: MixRules(vol=-14dB, pan=77°, locut=178Hz, hicut=14964Hz)	10_ElecGtrMic1
10: MixRules(vol=-15dB, pan=-19°, locut=240Hz, hicut=18375Hz)	11_ElecGtrMic2
11: MixRules(vol=-16dB, pan=-77°, locut=41Hz, hicut=17310Hz)	13_Piano
6th mix : fobj = 1.713	
1: MixRules(vol=-7dB, pan=0°, locut=28Hz, hicut=7437Hz)	01_KickIn
2: MixRules(vol=-6dB, pan=0°, locut=28Hz, hicut=7011Hz)	02_KickOut
3: MixRules(vol=-8dB, pan=0°, locut=139Hz, hicut=18182Hz)	03_Snare
4: MixRules(vol=-17dB, pan=49°, locut=40Hz, hicut=13436Hz)	04_Tom1
5: MixRules(vol=-17dB, pan=78°, locut=44Hz, hicut=14421Hz)	05_Tom2
6: MixRules(vol=-14dB, pan=-63°, locut=82Hz, hicut=14721Hz)	06_Overhead1
7: MixRules(vol=-17dB, pan=71°, locut=128Hz, hicut=15949Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=42Hz, hicut=10963Hz)	08_BassDI
9: MixRules(vol=-12dB, pan=-86°, locut=114Hz, hicut=12698Hz)	10_ElecGtrMic1
10: MixRules(vol=-14dB, pan=29°, locut=127Hz, hicut=8219Hz)	11_ElecGtrMic2
11: MixRules(vol=-13dB, pan=-29°, locut=30Hz, hicut=17542Hz)	13_Piano
7th mix : fobj = 1.806	
1: MixRules(vol=-8dB, pan=0°, locut=22Hz, hicut=10789Hz)	01_KickIn
2: MixRules(vol=-9dB, pan=0°, locut=35Hz, hicut=12244Hz)	02_KickOut
3: MixRules(vol=-6dB, pan=0°, locut=52Hz, hicut=16954Hz)	03_Snare
4: MixRules(vol=-16dB, pan=50°, locut=31Hz, hicut=15543Hz)	04_Tom1
5: MixRules(vol=-16dB, pan=69°, locut=46Hz, hicut=7877Hz)	05_Tom2
6: MixRules(vol=-14dB, pan=-66°, locut=87Hz, hicut=16249Hz)	06_Overhead1
7: MixRules(vol=-12dB, pan=64°, locut=119Hz, hicut=14998Hz)	07_Overhead2
8: MixRules(vol=-6dB, pan=0°, locut=20Hz, hicut=10018Hz)	08_BassDI
9: MixRules(vol=-14dB, pan=46°, locut=106Hz, hicut=14964Hz)	10_ElecGtrMic1
10: MixRules(vol=-15dB, pan=-19°, locut=226Hz, hicut=18375Hz)	11_ElecGtrMic2
11: MixRules(vol=-16dB, pan=-77°, locut=36Hz, hicut=17310Hz)	13_Piano

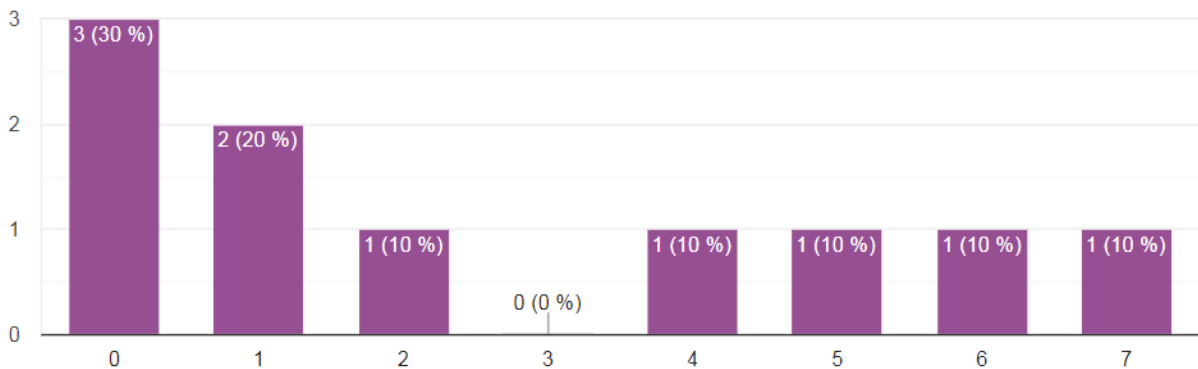


Figure E.71 – Jazz : Le pré-mixage préféré.

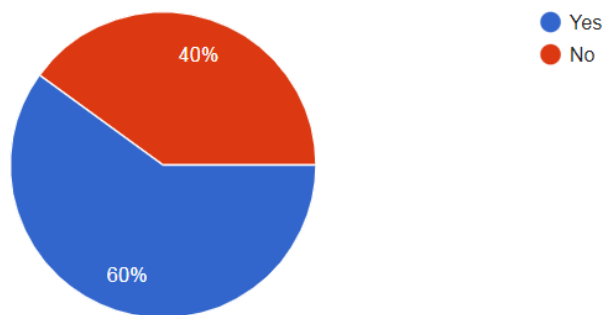


Figure E.72 – Jazz : L'intention d'utiliser le pré-mixage préféré.

E.7.2 Les graphes de convergence

La fonction objectif

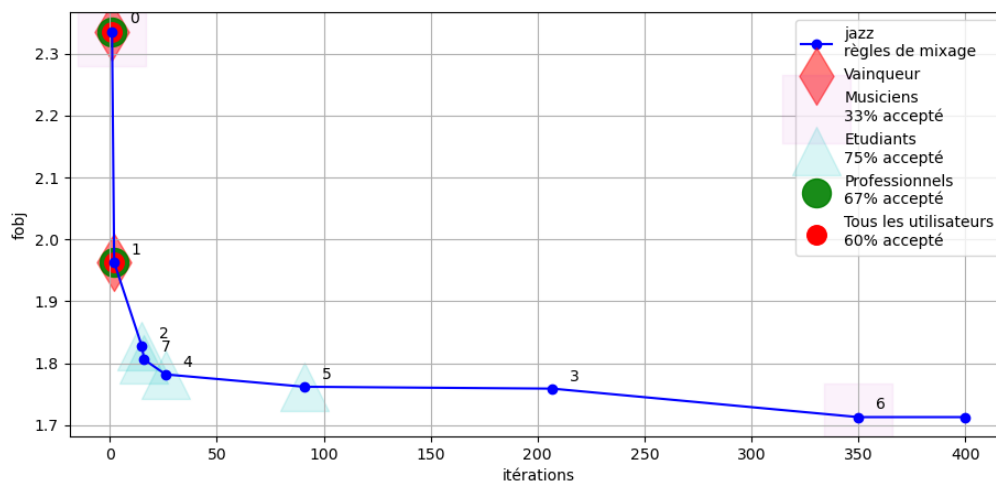


Figure E.73 – Jazz : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

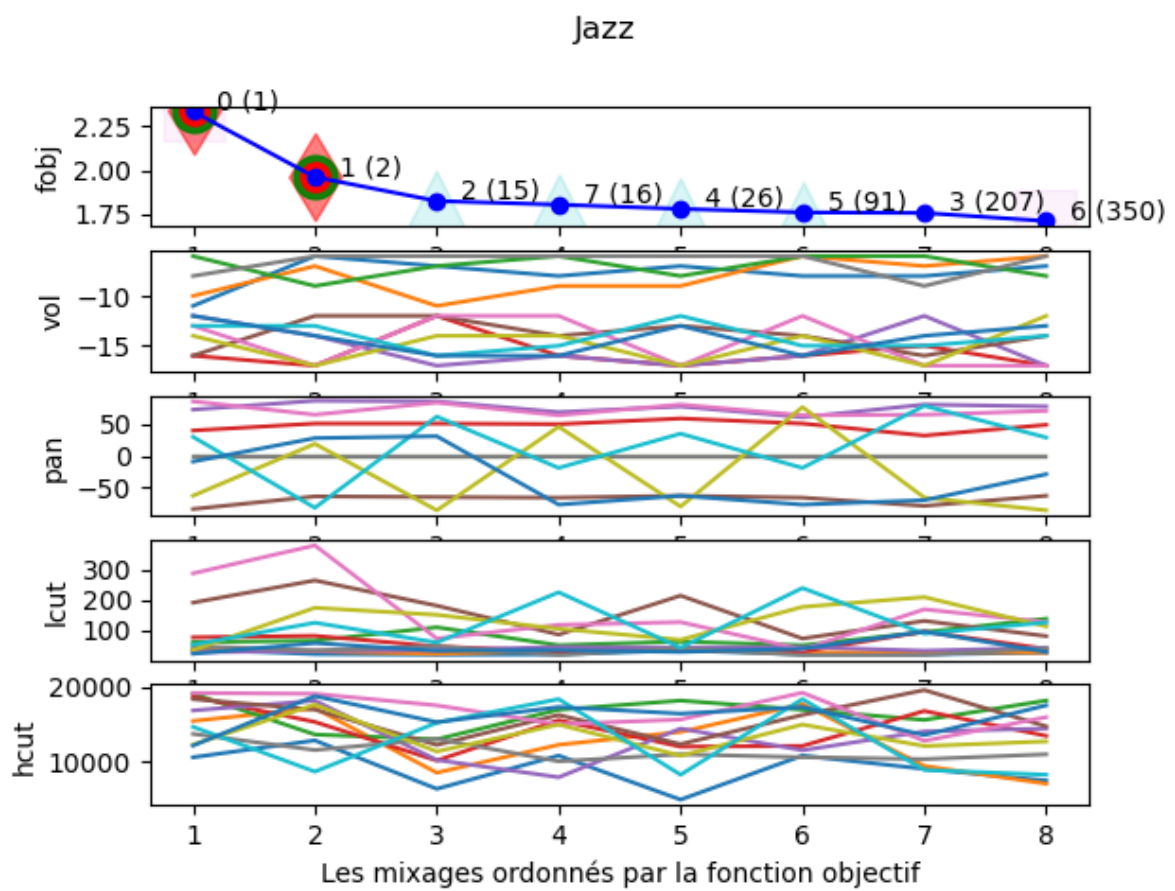


Figure E.74 – Jazz : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

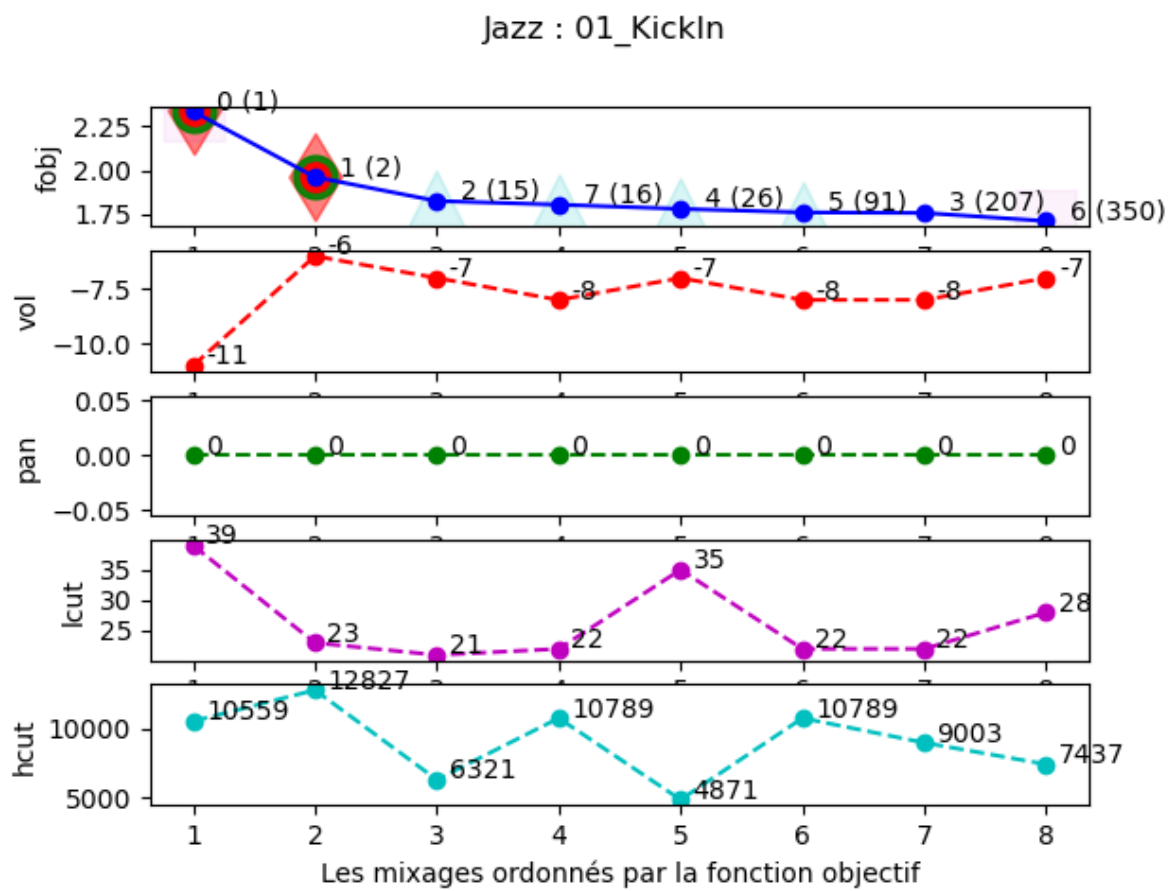


Figure E.75 – Jazz : Graphe de convergence – 01_KickIn.

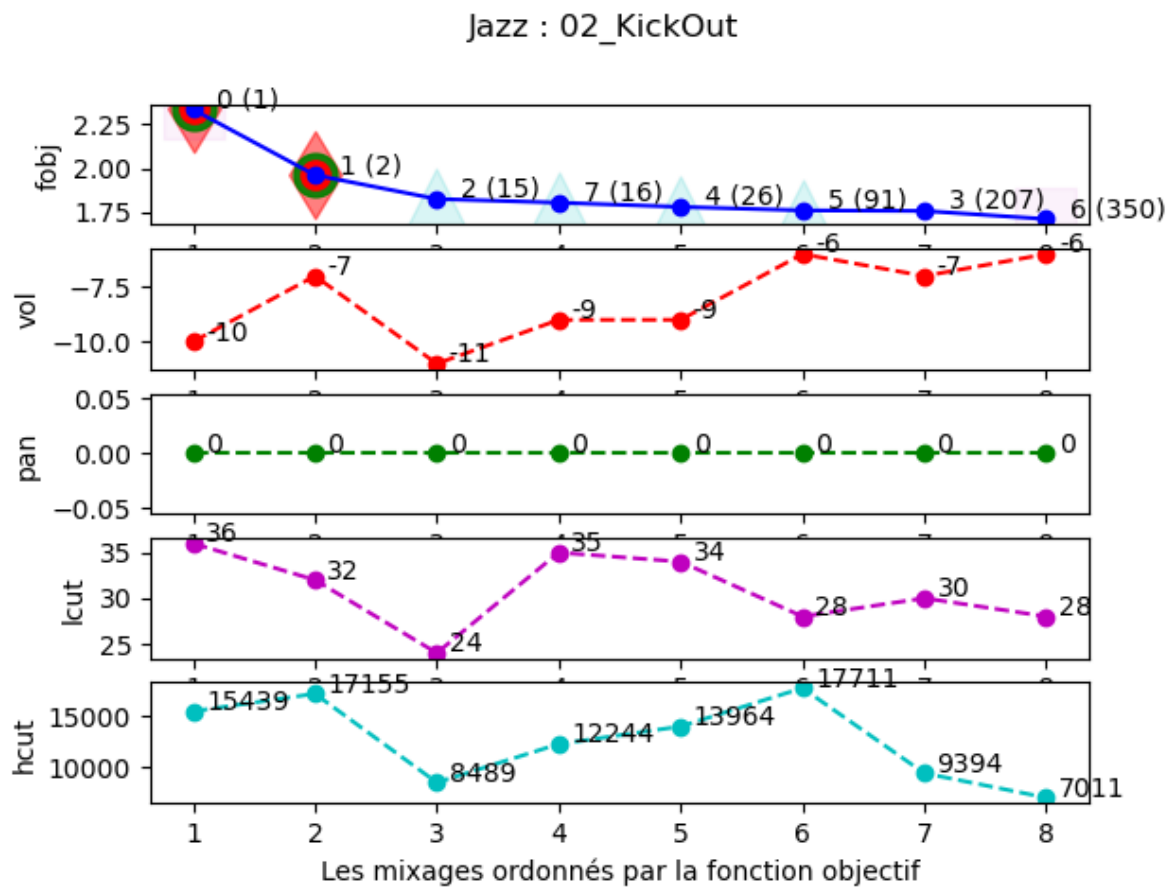


Figure E.76 – Jazz : Graphe de convergence – 02_KickOut.

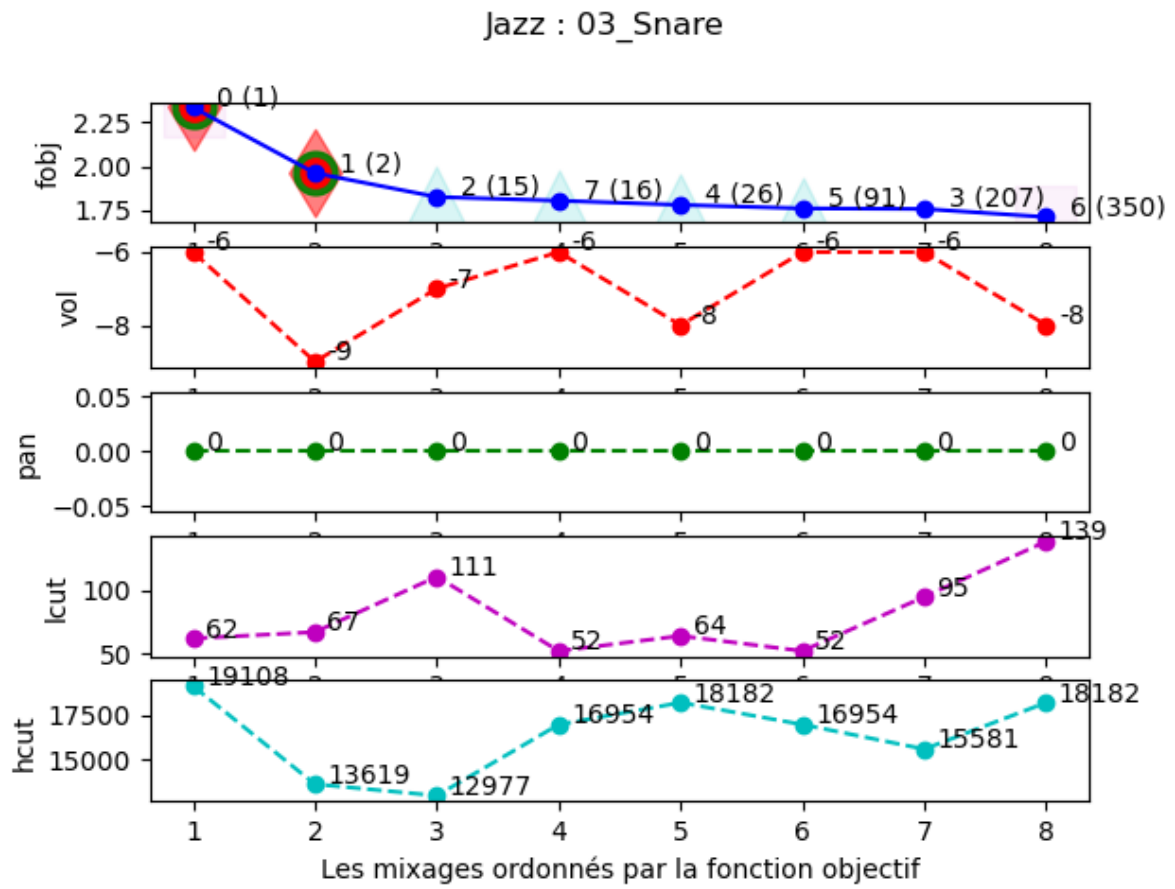


Figure E.77 – Jazz : Graphe de convergence – 03_Snare.

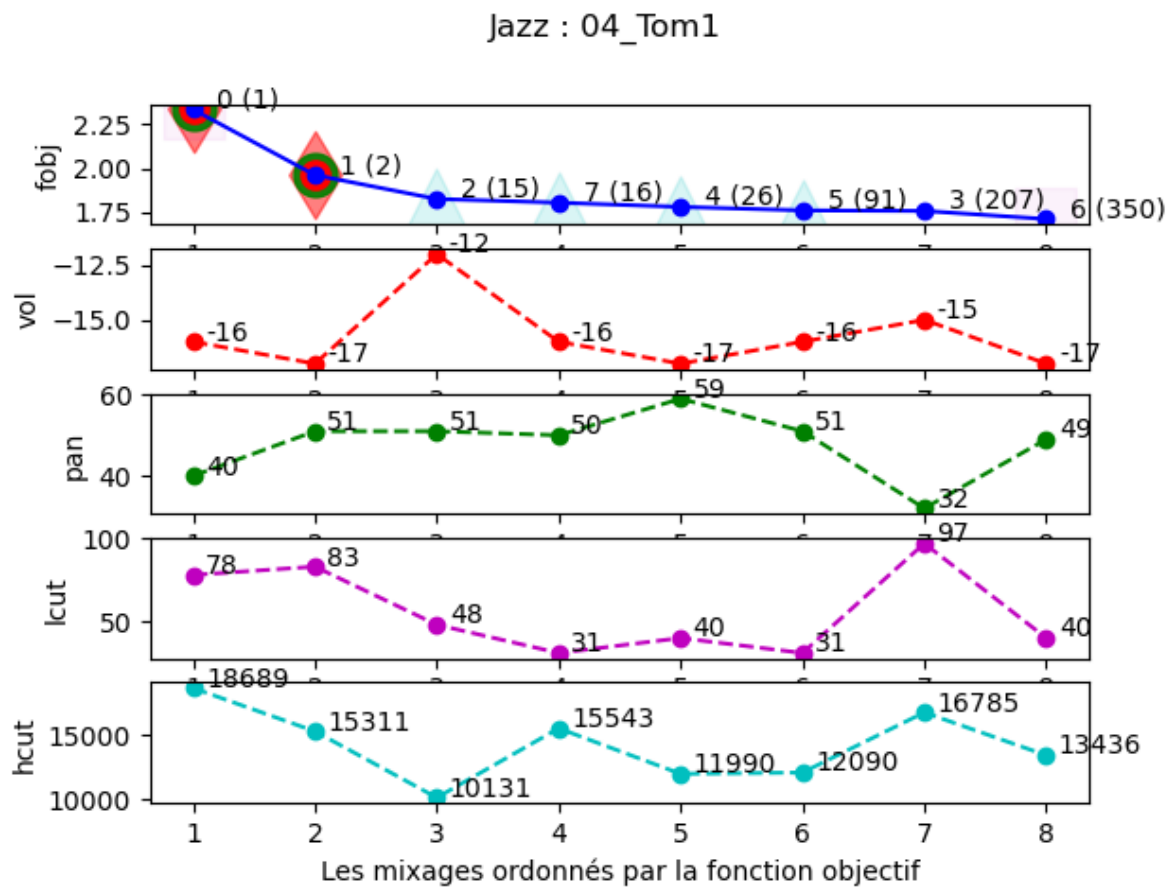


Figure E.78 – Jazz : Graphe de convergence – 04_Tom1.

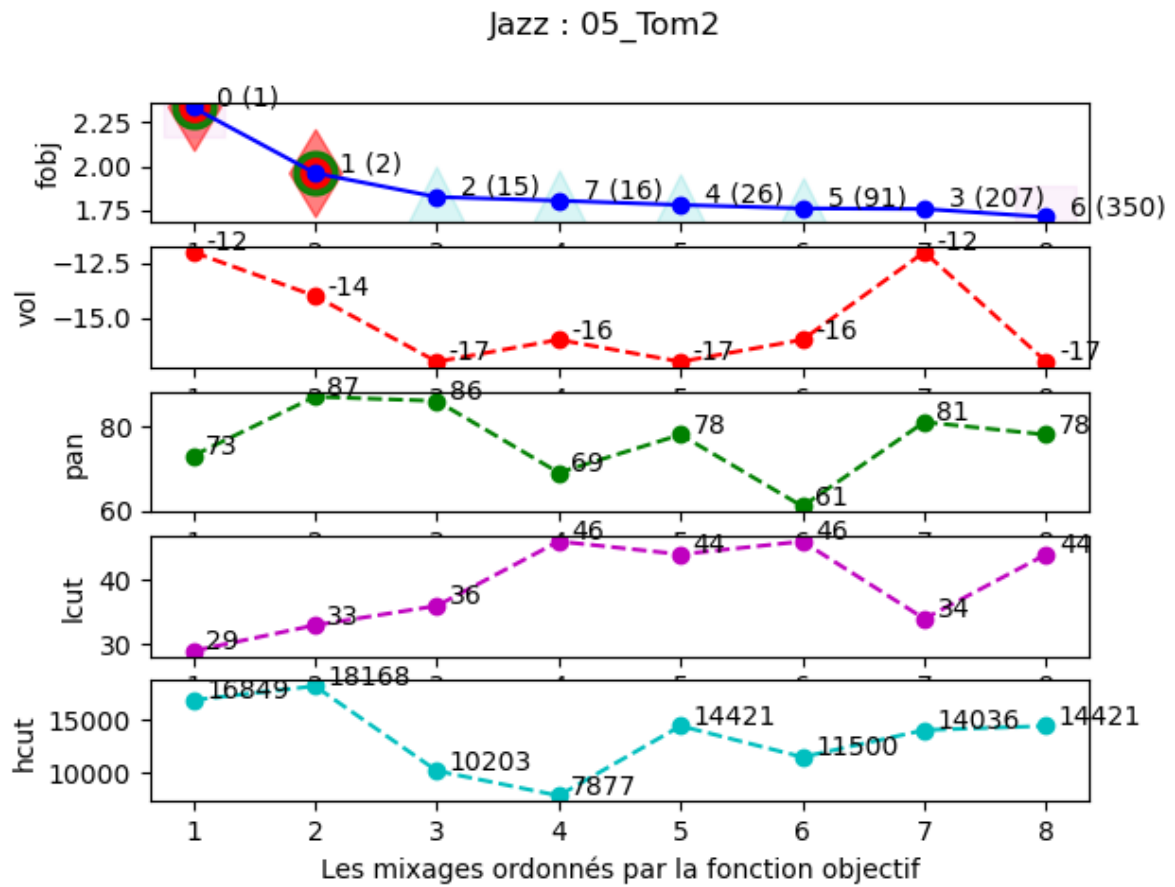


Figure E.79 – Jazz : Graphe de convergence – 05_Tom2.

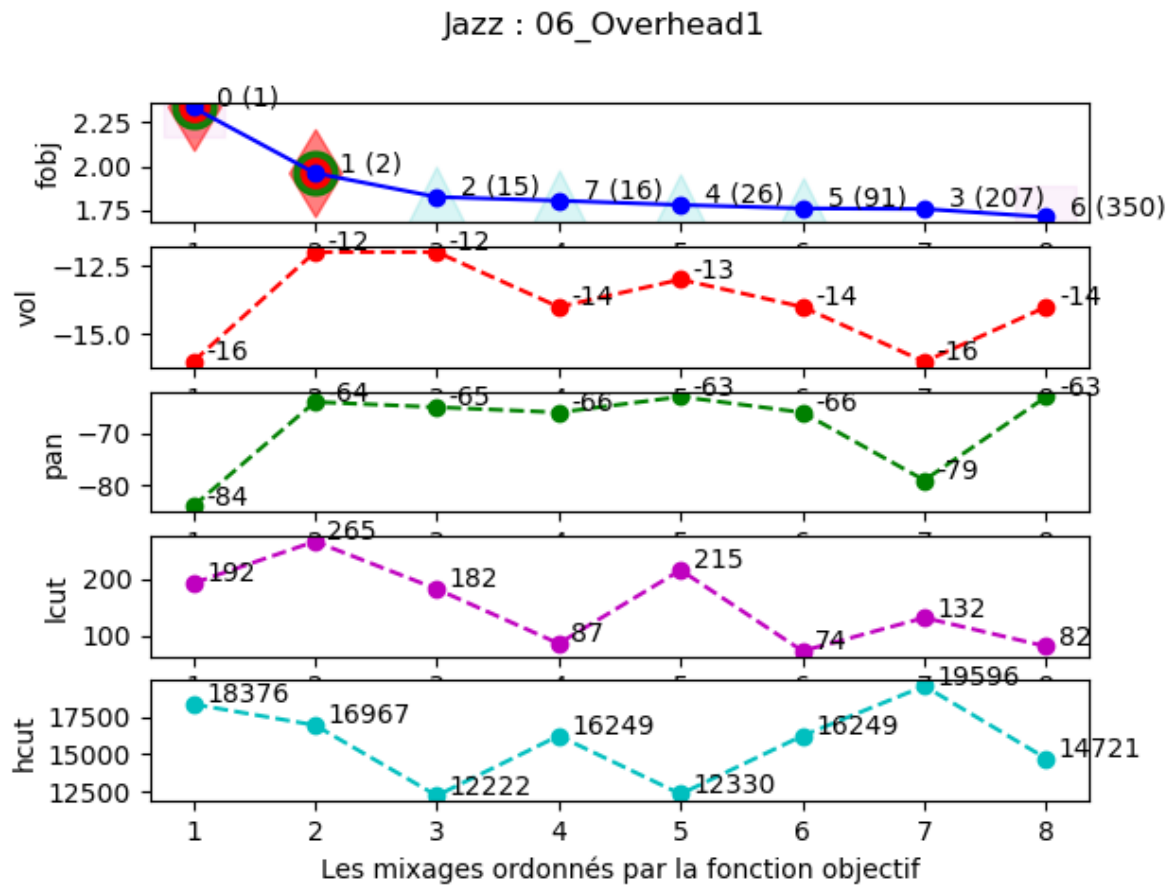


Figure E.80 – Jazz : Graphe de convergence – 06_Overhead1.

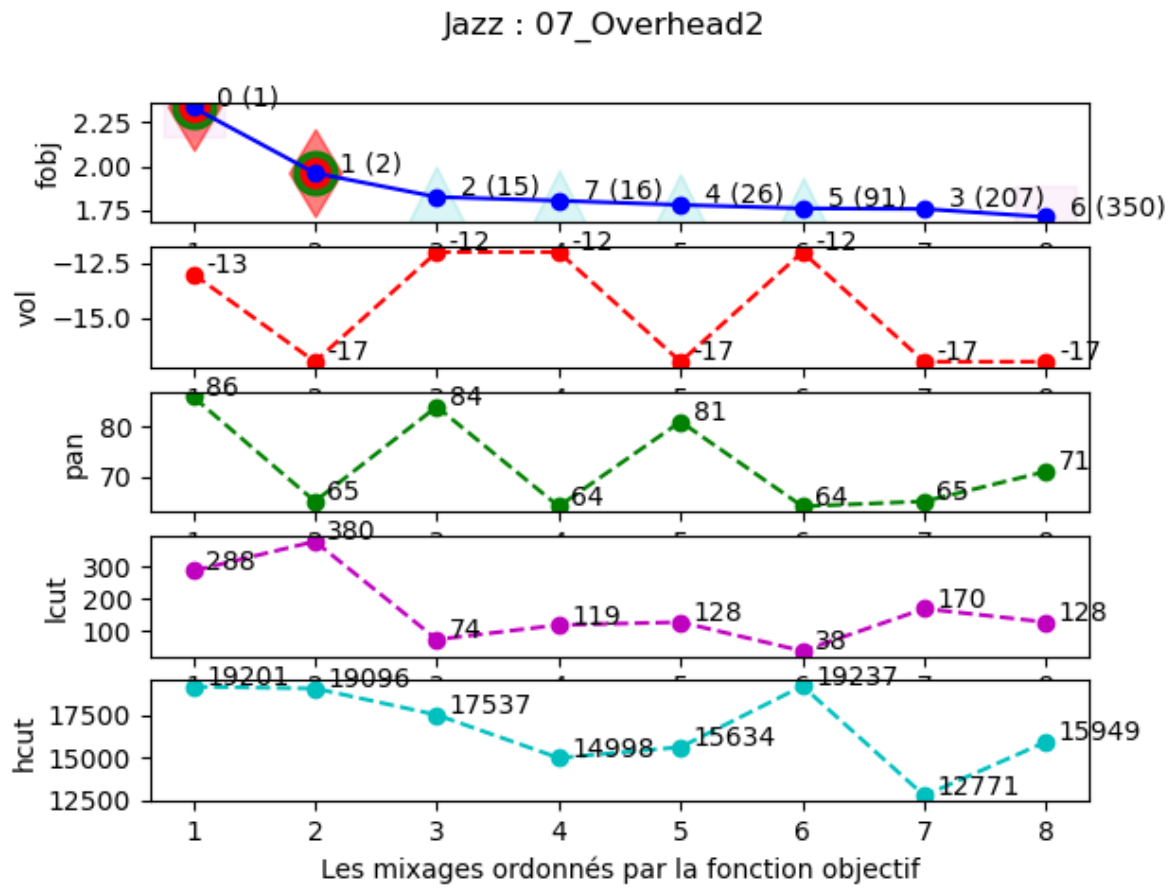


Figure E.81 – Jazz : Graphe de convergence – 07_Overhead2.

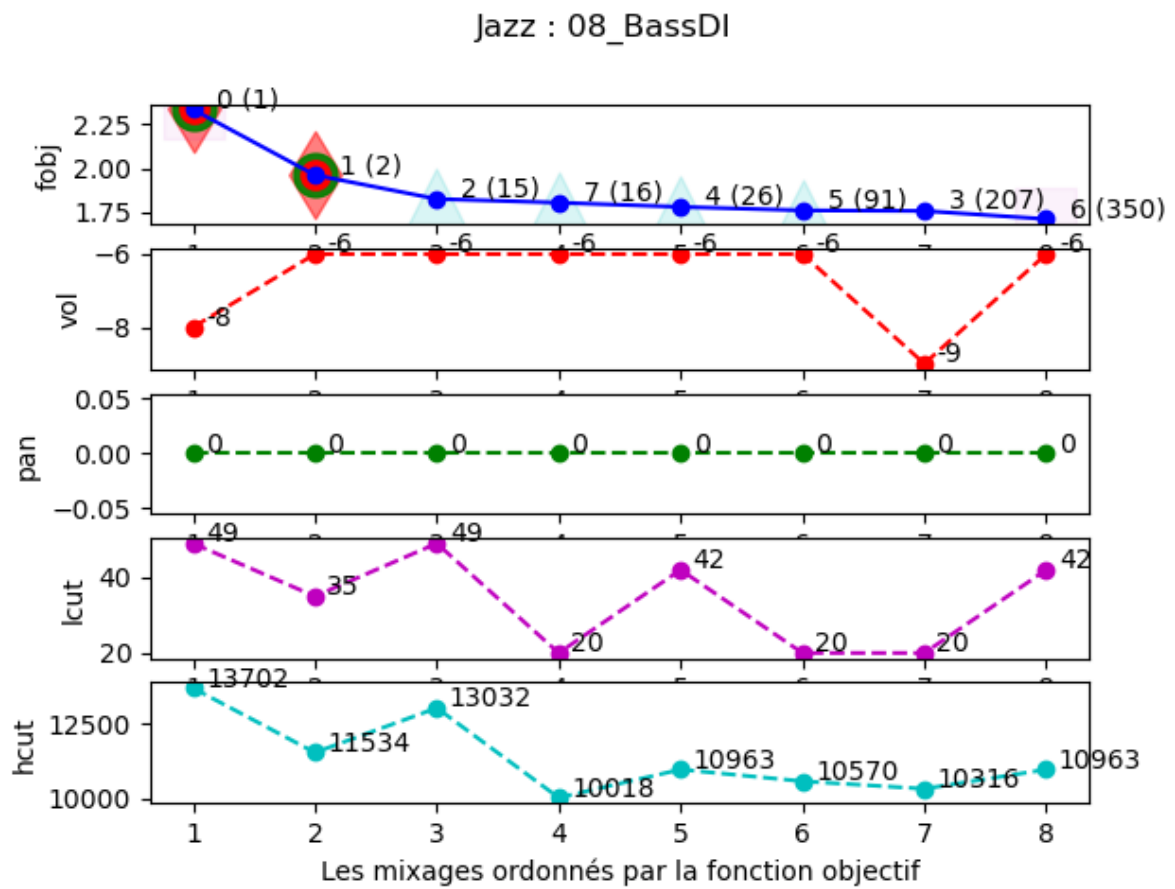


Figure E.82 – Jazz : Graphe de convergence – 08_BassDI.

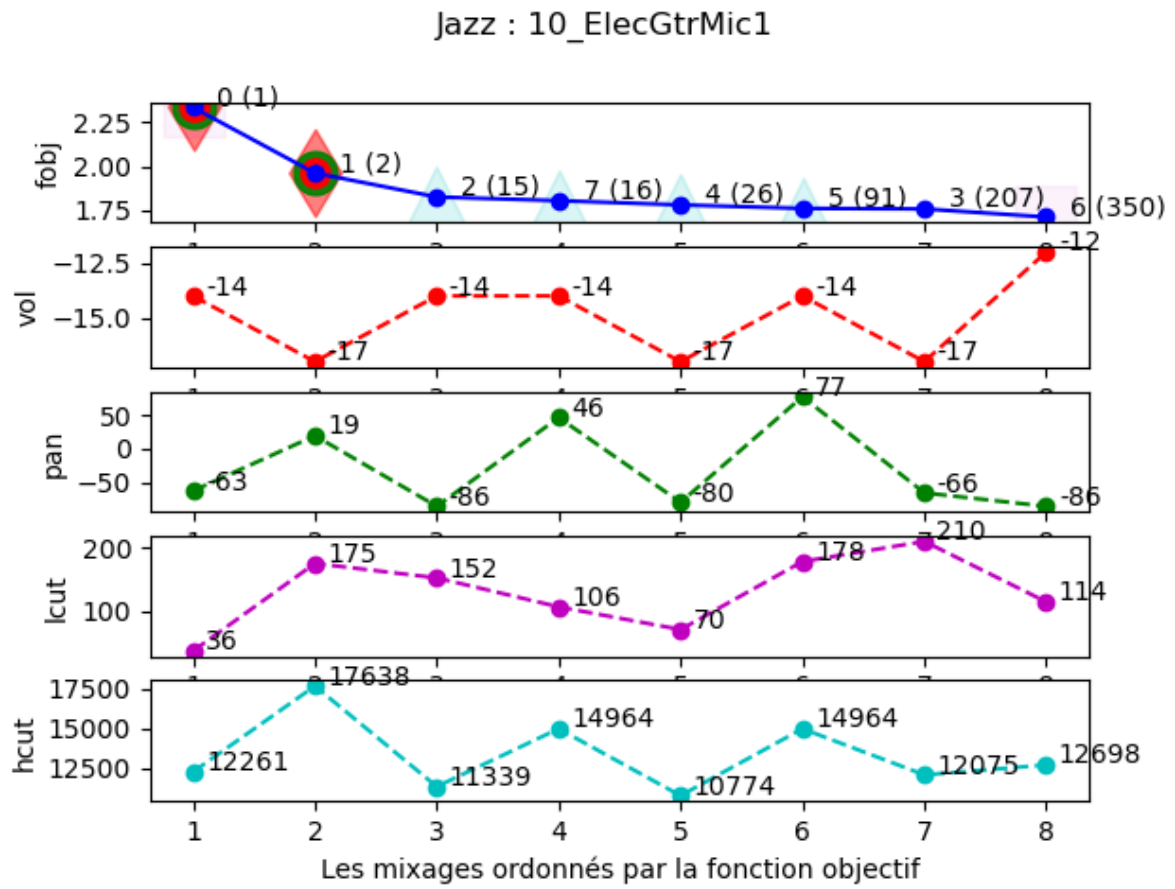


Figure E.83 – Jazz : Graphe de convergence – 10_ElecGtrMic1.

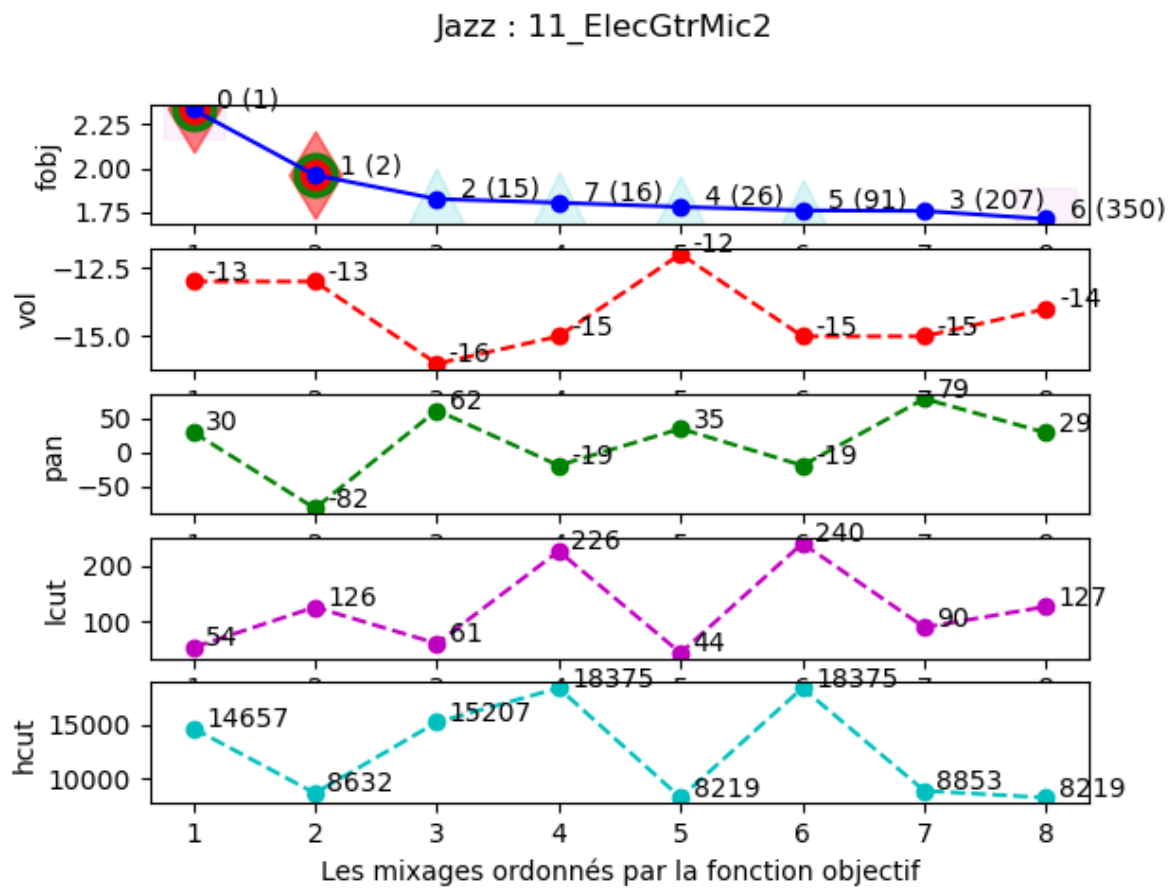


Figure E.84 – Jazz : Graphe de convergence – 11_ElecGtrMic2.

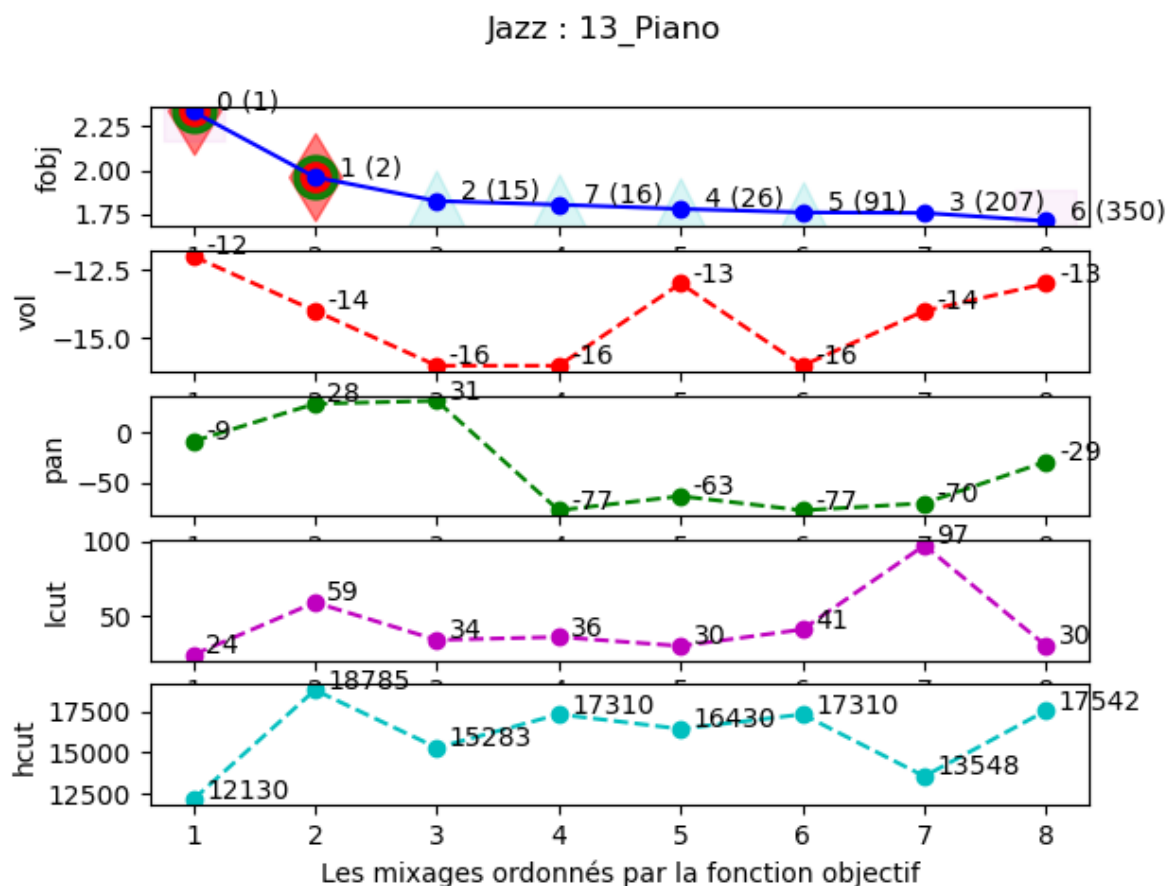


Figure E.85 – Jazz : Graphe de convergence – 13_Piano.

E.8 Ballad : MR 1001 - Jay Menon : 16 tracks

E.8.1 Les règles de mixage

0th mix : fobj = 1.378

- | | |
|--|----------------|
| 1: MixRules(vol=-6dB, pan=0°, locut=86Hz, hicut=18058Hz) | 01_Kick |
| 2: MixRules(vol=-13dB, pan=0°, locut=81Hz, hicut=12605Hz) | 02_Snare |
| 3: MixRules(vol=-14dB, pan=-3°, locut=318Hz, hicut=20000Hz) | 03_HiHat |
| 4: MixRules(vol=-17dB, pan=-7°, locut=180Hz, hicut=20000Hz) | 04_Overheads |
| 5: MixRules(vol=-19dB, pan=-33°, locut=186Hz, hicut=19052Hz) | 05_CymbalRoll1 |
| 6: MixRules(vol=-31dB, pan=55°, locut=193Hz, hicut=7965Hz) | 06_CymbalRoll2 |
| 7: MixRules(vol=-10dB, pan=0°, locut=28Hz, hicut=7614Hz) | 07_Bass |
| 8: MixRules(vol=-14dB, pan=-9°, locut=54Hz, hicut=3305Hz) | 08_AcGuitar1 |
| 9: MixRules(vol=-16dB, pan=-69°, locut=493Hz, hicut=16312Hz) | 09_AcGuitar2 |
| 10: MixRules(vol=-12dB, pan=68°, locut=59Hz, hicut=16638Hz) | 10_AcGuitar3 |
| 11: MixRules(vol=-14dB, pan=0°, locut=275Hz, hicut=18495Hz) | 11_AcGuitar4 |
| 12: MixRules(vol=-17dB, pan=30°, locut=116Hz, hicut=17442Hz) | 12_AcGuitar5 |
| 13: MixRules(vol=-12dB, pan=58°, locut=63Hz, hicut=12889Hz) | 13_Piano |

14: MixRules(vol=-29dB, pan=64°, locut=86Hz, hicut=9094Hz)	14_Pad1
15: MixRules(vol=-30dB, pan=73°, locut=61Hz, hicut=15634Hz)	15_Pad2
16: MixRules(vol=-8dB, pan=0°, locut=70Hz, hicut=20000Hz)	16_LeadVox

1th mix : fobj = 1.372

1: MixRules(vol=-8dB, pan=0°, locut=77Hz, hicut=14694Hz)	01_Kick
2: MixRules(vol=-15dB, pan=0°, locut=194Hz, hicut=11650Hz)	02_Snare
3: MixRules(vol=-14dB, pan=-21°, locut=30Hz, hicut=20000Hz)	03_HiHat
4: MixRules(vol=-15dB, pan=-6°, locut=224Hz, hicut=20000Hz)	04_Overheads
5: MixRules(vol=-26dB, pan=-59°, locut=147Hz, hicut=8800Hz)	05_CymbalRoll1
6: MixRules(vol=-21dB, pan=52°, locut=46Hz, hicut=12709Hz)	06_CymbalRoll2
7: MixRules(vol=-9dB, pan=0°, locut=34Hz, hicut=19268Hz)	07_Bass
8: MixRules(vol=-12dB, pan=1°, locut=64Hz, hicut=9731Hz)	08_AcGuitar1
9: MixRules(vol=-17dB, pan=-67°, locut=457Hz, hicut=18188Hz)	09_AcGuitar2
10: MixRules(vol=-17dB, pan=73°, locut=386Hz, hicut=17991Hz)	10_AcGuitar3
11: MixRules(vol=-12dB, pan=-6°, locut=77Hz, hicut=14433Hz)	11_AcGuitar4
12: MixRules(vol=-13dB, pan=-79°, locut=38Hz, hicut=12049Hz)	12_AcGuitar5
13: MixRules(vol=-13dB, pan=-84°, locut=157Hz, hicut=16466Hz)	13_Piano
14: MixRules(vol=-31dB, pan=-26°, locut=133Hz, hicut=15857Hz)	14_Pad1
15: MixRules(vol=-24dB, pan=63°, locut=151Hz, hicut=17806Hz)	15_Pad2
16: MixRules(vol=-8dB, pan=0°, locut=82Hz, hicut=20000Hz)	16_LeadVox

2th mix : fobj = 1.307

1: MixRules(vol=-9dB, pan=0°, locut=71Hz, hicut=8425Hz)	01_Kick
2: MixRules(vol=-17dB, pan=0°, locut=159Hz, hicut=9645Hz)	02_Snare
3: MixRules(vol=-16dB, pan=-16°, locut=401Hz, hicut=20000Hz)	03_HiHat
4: MixRules(vol=-16dB, pan=0°, locut=133Hz, hicut=20000Hz)	04_Overheads
5: MixRules(vol=-28dB, pan=-32°, locut=130Hz, hicut=14877Hz)	05_CymbalRoll1
6: MixRules(vol=-31dB, pan=31°, locut=73Hz, hicut=19476Hz)	06_CymbalRoll2
7: MixRules(vol=-11dB, pan=0°, locut=20Hz, hicut=17235Hz)	07_Bass
8: MixRules(vol=-14dB, pan=0°, locut=56Hz, hicut=6406Hz)	08_AcGuitar1
9: MixRules(vol=-13dB, pan=-65°, locut=447Hz, hicut=15568Hz)	09_AcGuitar2
10: MixRules(vol=-12dB, pan=74°, locut=155Hz, hicut=16523Hz)	10_AcGuitar3
11: MixRules(vol=-14dB, pan=2°, locut=355Hz, hicut=15021Hz)	11_AcGuitar4
12: MixRules(vol=-13dB, pan=12°, locut=116Hz, hicut=18025Hz)	12_AcGuitar5
13: MixRules(vol=-16dB, pan=-83°, locut=132Hz, hicut=18896Hz)	13_Piano
14: MixRules(vol=-25dB, pan=65°, locut=119Hz, hicut=17419Hz)	14_Pad1
15: MixRules(vol=-20dB, pan=-84°, locut=70Hz, hicut=13007Hz)	15_Pad2
16: MixRules(vol=-8dB, pan=0°, locut=66Hz, hicut=20000Hz)	16_LeadVox

3th mix : fobj = 1.229

1: MixRules(vol=-11dB, pan=0°, locut=37Hz, hicut=8991Hz)	01_Kick
2: MixRules(vol=-14dB, pan=0°, locut=160Hz, hicut=9014Hz)	02_Snare
3: MixRules(vol=-12dB, pan=-27°, locut=152Hz, hicut=20000Hz)	03_HiHat
4: MixRules(vol=-16dB, pan=-2°, locut=280Hz, hicut=20000Hz)	04_Overheads
5: MixRules(vol=-26dB, pan=-35°, locut=109Hz, hicut=8734Hz)	05_CymbalRoll1
6: MixRules(vol=-26dB, pan=32°, locut=159Hz, hicut=9694Hz)	06_CymbalRoll2
7: MixRules(vol=-6dB, pan=0°, locut=22Hz, hicut=14608Hz)	07_Bass
8: MixRules(vol=-13dB, pan=0°, locut=61Hz, hicut=15666Hz)	08_AcGuitar1
9: MixRules(vol=-15dB, pan=-77°, locut=106Hz, hicut=19831Hz)	09_AcGuitar2
10: MixRules(vol=-16dB, pan=87°, locut=183Hz, hicut=18709Hz)	10_AcGuitar3
11: MixRules(vol=-13dB, pan=-5°, locut=101Hz, hicut=13649Hz)	11_AcGuitar4
12: MixRules(vol=-12dB, pan=29°, locut=175Hz, hicut=14171Hz)	12_AcGuitar5
13: MixRules(vol=-15dB, pan=-75°, locut=85Hz, hicut=11467Hz)	13_Piano
14: MixRules(vol=-34dB, pan=58°, locut=121Hz, hicut=13567Hz)	14_Pad1

15: MixRules(vol=-23dB, pan=-57°, locut=124Hz, hicut=13755Hz)
 16: MixRules(vol=-7dB, pan=0°, locut=49Hz, hicut=20000Hz)

15_Pad2
 16_LeadVox

4th mix : fobj = 1.235

1: MixRules(vol=-8dB, pan=0°, locut=83Hz, hicut=9413Hz)
 2: MixRules(vol=-15dB, pan=0°, locut=178Hz, hicut=15254Hz)
 3: MixRules(vol=-12dB, pan=-24°, locut=100Hz, hicut=20000Hz)
 4: MixRules(vol=-13dB, pan=-1°, locut=119Hz, hicut=20000Hz)
 5: MixRules(vol=-23dB, pan=-43°, locut=152Hz, hicut=16437Hz)
 6: MixRules(vol=-28dB, pan=33°, locut=110Hz, hicut=14194Hz)
 7: MixRules(vol=-9dB, pan=0°, locut=23Hz, hicut=7534Hz)
 8: MixRules(vol=-15dB, pan=7°, locut=75Hz, hicut=17004Hz)
 9: MixRules(vol=-15dB, pan=-77°, locut=374Hz, hicut=15930Hz)
 10: MixRules(vol=-14dB, pan=68°, locut=210Hz, hicut=13401Hz)
 11: MixRules(vol=-14dB, pan=4°, locut=214Hz, hicut=17563Hz)
 12: MixRules(vol=-17dB, pan=-50°, locut=88Hz, hicut=9818Hz)
 13: MixRules(vol=-14dB, pan=-44°, locut=264Hz, hicut=16442Hz)
 14: MixRules(vol=-31dB, pan=-84°, locut=37Hz, hicut=11175Hz)
 15: MixRules(vol=-35dB, pan=54°, locut=139Hz, hicut=12931Hz)
 16: MixRules(vol=-10dB, pan=0°, locut=25Hz, hicut=20000Hz)

01_Kick
 02_Snare
 03_HiHat
 04_Overheads
 05_CymbalRoll1
 06_CymbalRoll2
 07_Bass
 08_AcGuitar1
 09_AcGuitar2
 10_AcGuitar3
 11_AcGuitar4
 12_AcGuitar5
 13_Piano
 14_Pad1
 15_Pad2
 16_LeadVox

5th mix : fobj = 1.178

1: MixRules(vol=-10dB, pan=0°, locut=41Hz, hicut=19897Hz)
 2: MixRules(vol=-15dB, pan=0°, locut=30Hz, hicut=10216Hz)
 3: MixRules(vol=-13dB, pan=-8°, locut=71Hz, hicut=20000Hz)
 4: MixRules(vol=-14dB, pan=5°, locut=104Hz, hicut=20000Hz)
 5: MixRules(vol=-33dB, pan=-54°, locut=97Hz, hicut=10783Hz)
 6: MixRules(vol=-28dB, pan=37°, locut=186Hz, hicut=19954Hz)
 7: MixRules(vol=-11dB, pan=0°, locut=30Hz, hicut=6041Hz)
 8: MixRules(vol=-14dB, pan=-3°, locut=34Hz, hicut=19511Hz)
 9: MixRules(vol=-15dB, pan=-86°, locut=399Hz, hicut=17777Hz)
 10: MixRules(vol=-16dB, pan=74°, locut=32Hz, hicut=18818Hz)
 11: MixRules(vol=-13dB, pan=-4°, locut=342Hz, hicut=19122Hz)
 12: MixRules(vol=-15dB, pan=50°, locut=194Hz, hicut=9654Hz)
 13: MixRules(vol=-17dB, pan=41°, locut=172Hz, hicut=13641Hz)
 14: MixRules(vol=-29dB, pan=-89°, locut=172Hz, hicut=8093Hz)
 15: MixRules(vol=-33dB, pan=-48°, locut=184Hz, hicut=13015Hz)
 16: MixRules(vol=-11dB, pan=0°, locut=32Hz, hicut=20000Hz)

01_Kick
 02_Snare
 03_HiHat
 04_Overheads
 05_CymbalRoll1
 06_CymbalRoll2
 07_Bass
 08_AcGuitar1
 09_AcGuitar2
 10_AcGuitar3
 11_AcGuitar4
 12_AcGuitar5
 13_Piano
 14_Pad1
 15_Pad2
 16_LeadVox

6th mix : fobj = 1.180

1: MixRules(vol=-6dB, pan=0°, locut=60Hz, hicut=19072Hz)
 2: MixRules(vol=-17dB, pan=0°, locut=74Hz, hicut=11755Hz)
 3: MixRules(vol=-17dB, pan=-10°, locut=382Hz, hicut=20000Hz)
 4: MixRules(vol=-16dB, pan=-4°, locut=117Hz, hicut=20000Hz)
 5: MixRules(vol=-28dB, pan=-57°, locut=114Hz, hicut=12113Hz)
 6: MixRules(vol=-29dB, pan=37°, locut=111Hz, hicut=17895Hz)
 7: MixRules(vol=-10dB, pan=0°, locut=32Hz, hicut=16065Hz)
 8: MixRules(vol=-15dB, pan=7°, locut=63Hz, hicut=19346Hz)
 9: MixRules(vol=-13dB, pan=-65°, locut=92Hz, hicut=12855Hz)
 10: MixRules(vol=-12dB, pan=70°, locut=150Hz, hicut=15786Hz)
 11: MixRules(vol=-12dB, pan=8°, locut=201Hz, hicut=19791Hz)
 12: MixRules(vol=-17dB, pan=81°, locut=138Hz, hicut=14695Hz)
 13: MixRules(vol=-15dB, pan=-40°, locut=208Hz, hicut=17600Hz)
 14: MixRules(vol=-35dB, pan=-18°, locut=194Hz, hicut=11833Hz)
 15: MixRules(vol=-32dB, pan=-85°, locut=194Hz, hicut=15177Hz)

01_Kick
 02_Snare
 03_HiHat
 04_Overheads
 05_CymbalRoll1
 06_CymbalRoll2
 07_Bass
 08_AcGuitar1
 09_AcGuitar2
 10_AcGuitar3
 11_AcGuitar4
 12_AcGuitar5
 13_Piano
 14_Pad1
 15_Pad2

16: MixRules(vol=-10dB, pan=0°, locut=84Hz, hicut=20000Hz)	16_LeadVox
7th mix : fobj = 1.177	
1: MixRules(vol=-8dB, pan=0°, locut=22Hz, hicut=8230Hz)	01_Kick
2: MixRules(vol=-14dB, pan=0°, locut=143Hz, hicut=13472Hz)	02_Snare
3: MixRules(vol=-16dB, pan=-28°, locut=223Hz, hicut=20000Hz)	03_HiHat
4: MixRules(vol=-16dB, pan=-1°, locut=124Hz, hicut=20000Hz)	04_Overheads
5: MixRules(vol=-30dB, pan=-58°, locut=117Hz, hicut=13595Hz)	05_CymbalRoll1
6: MixRules(vol=-21dB, pan=58°, locut=136Hz, hicut=15049Hz)	06_CymbalRoll2
7: MixRules(vol=-6dB, pan=0°, locut=22Hz, hicut=14163Hz)	07_Bass
8: MixRules(vol=-12dB, pan=-8°, locut=62Hz, hicut=3083Hz)	08_AcGuitar1
9: MixRules(vol=-16dB, pan=-80°, locut=150Hz, hicut=13674Hz)	09_AcGuitar2
10: MixRules(vol=-12dB, pan=82°, locut=191Hz, hicut=12487Hz)	10_AcGuitar3
11: MixRules(vol=-16dB, pan=6°, locut=336Hz, hicut=14785Hz)	11_AcGuitar4
12: MixRules(vol=-13dB, pan=41°, locut=38Hz, hicut=10064Hz)	12_AcGuitar5
13: MixRules(vol=-13dB, pan=65°, locut=108Hz, hicut=12731Hz)	13_Piano
14: MixRules(vol=-21dB, pan=-19°, locut=191Hz, hicut=11604Hz)	14_Pad1
15: MixRules(vol=-32dB, pan=-75°, locut=118Hz, hicut=8891Hz)	15_Pad2
16: MixRules(vol=-6dB, pan=0°, locut=89Hz, hicut=20000Hz)	16_LeadVox

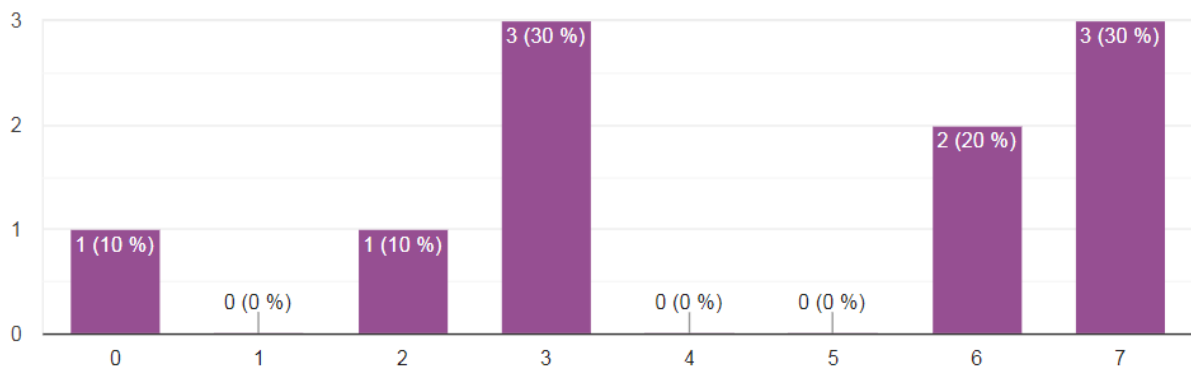


Figure E.86 – Ballad : Le pré-mixage préféré.

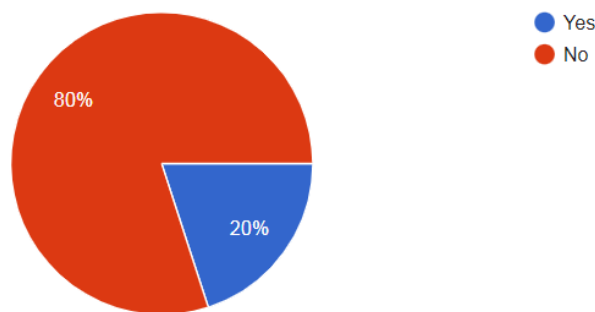


Figure E.87 – Ballad : L'intention d'utiliser le pré-mixage préféré.

E.8.2 Les graphes de convergence

La fonction objectif

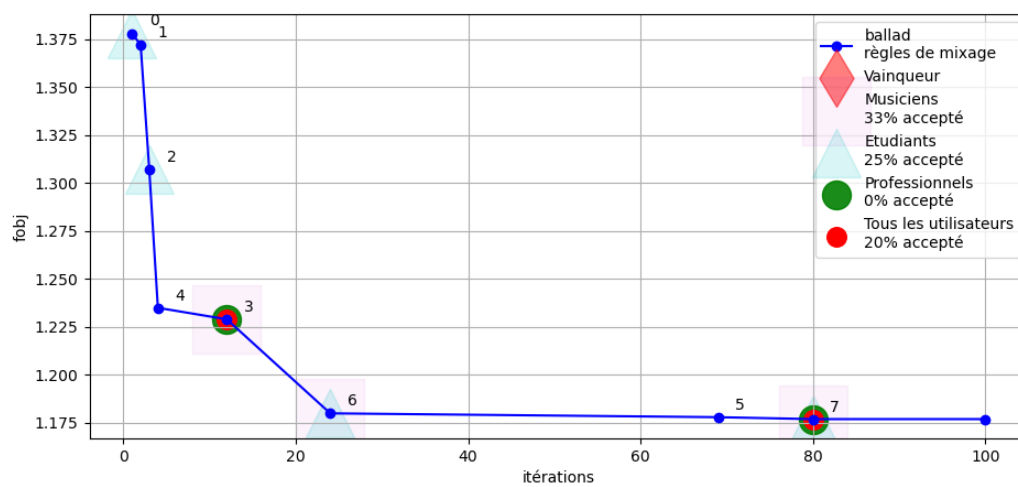


Figure E.88 – Ballad : Graphe de convergence – la fonction objectif.

Tous les instruments ensemble

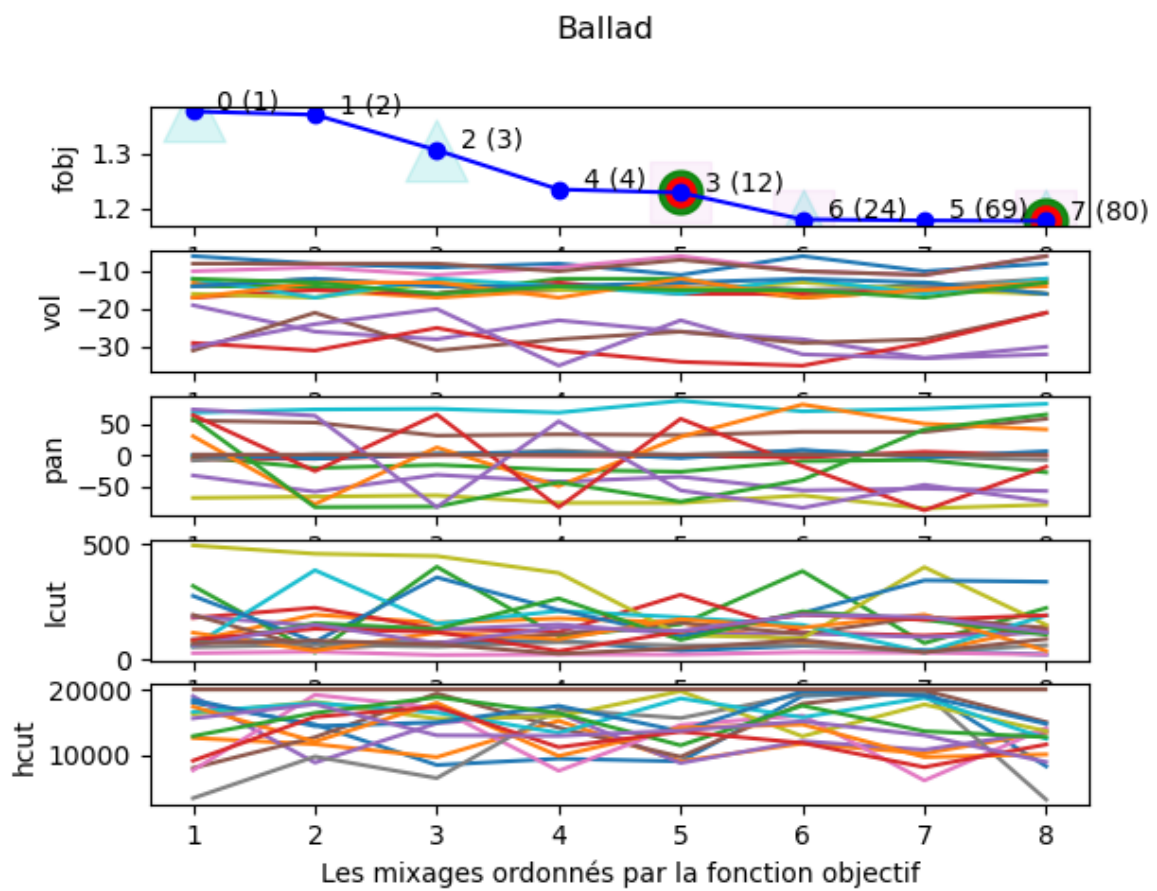


Figure E.89 – Ballad : Graphe de convergence – tous les instruments ensemble.

Chaque instrument séparément

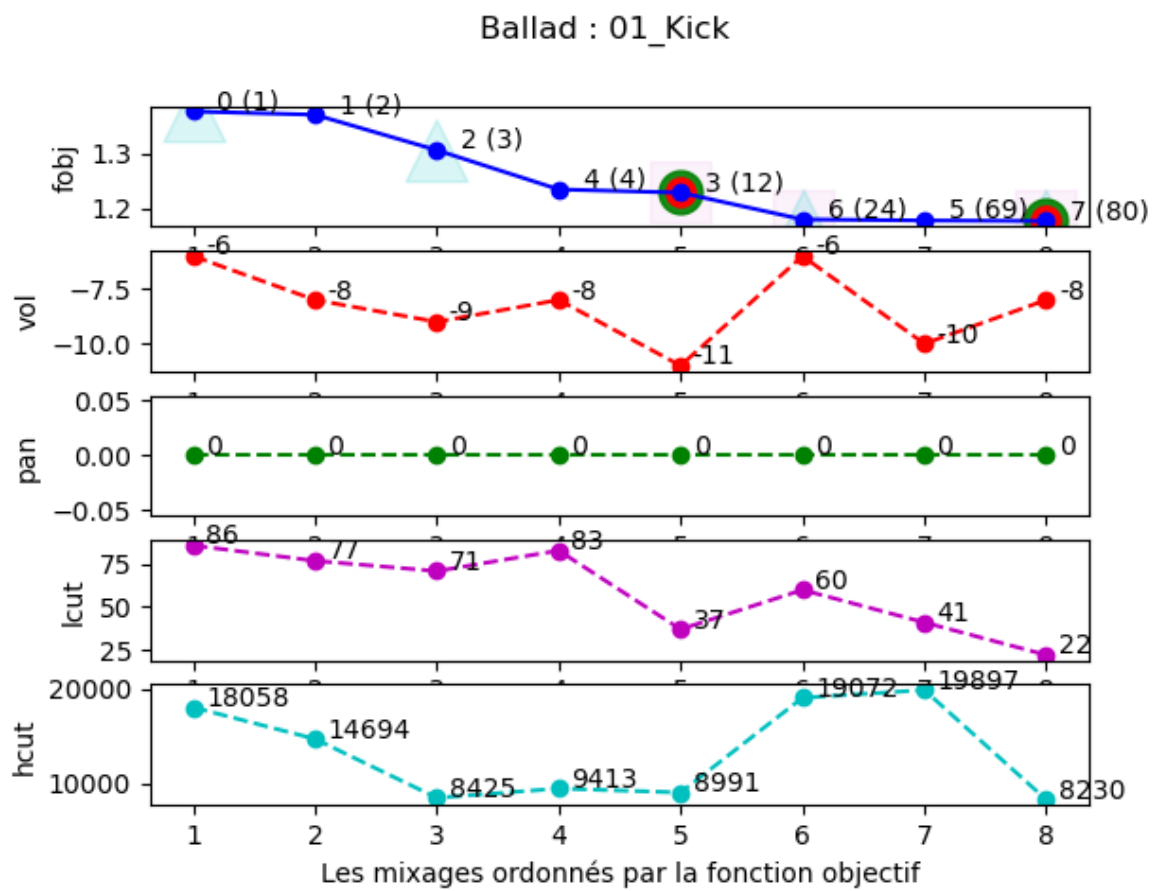


Figure E.90 – Ballad : Graphe de convergence – 01_Kick.

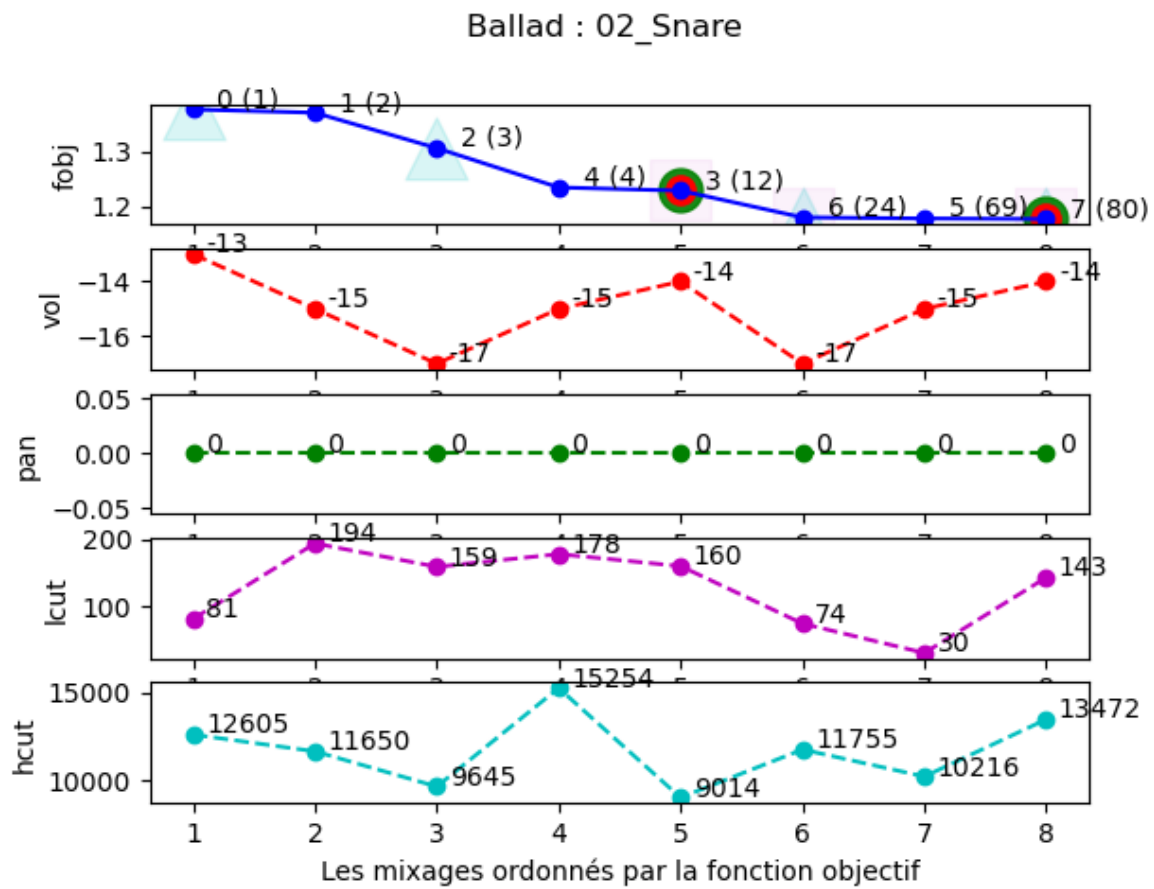


Figure E.91 – Ballad : Graphe de convergence – 02_Snare.

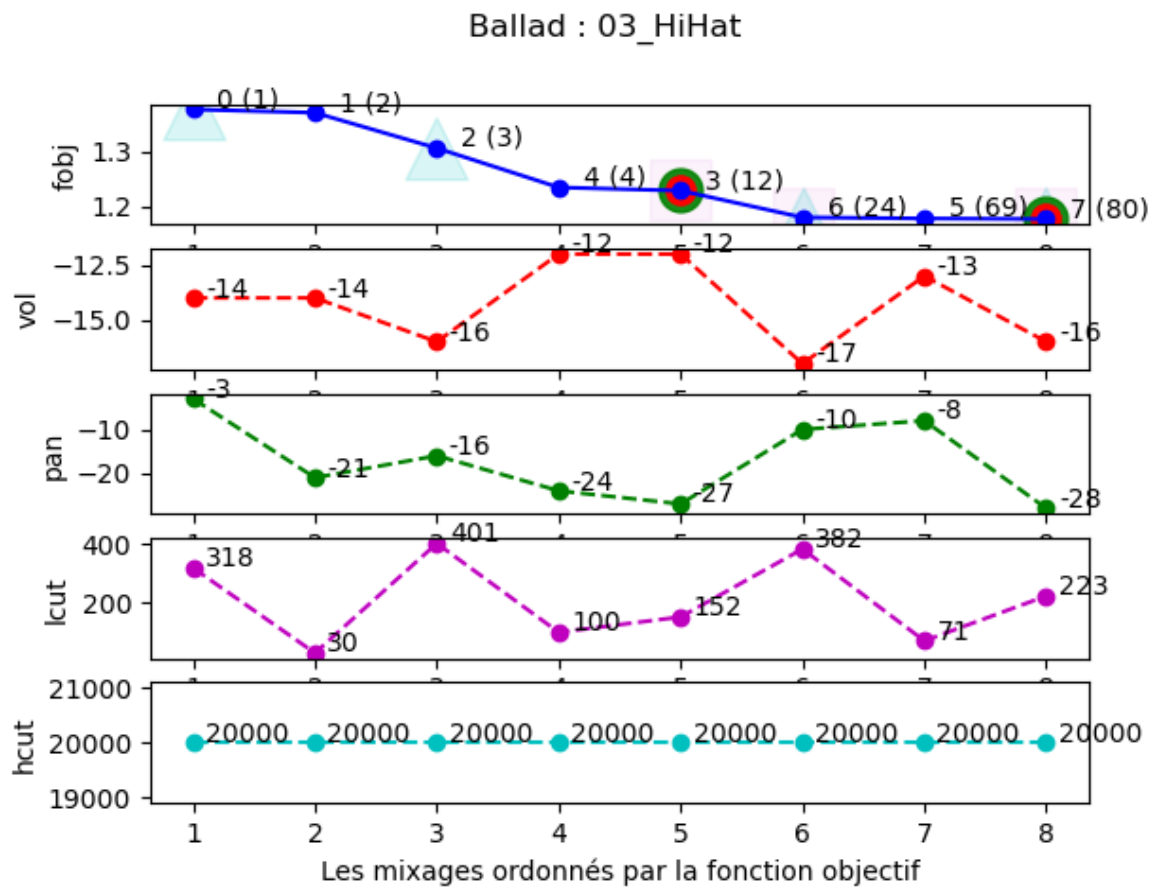


Figure E.92 – Ballad : Graphe de convergence – 03_HiHat.

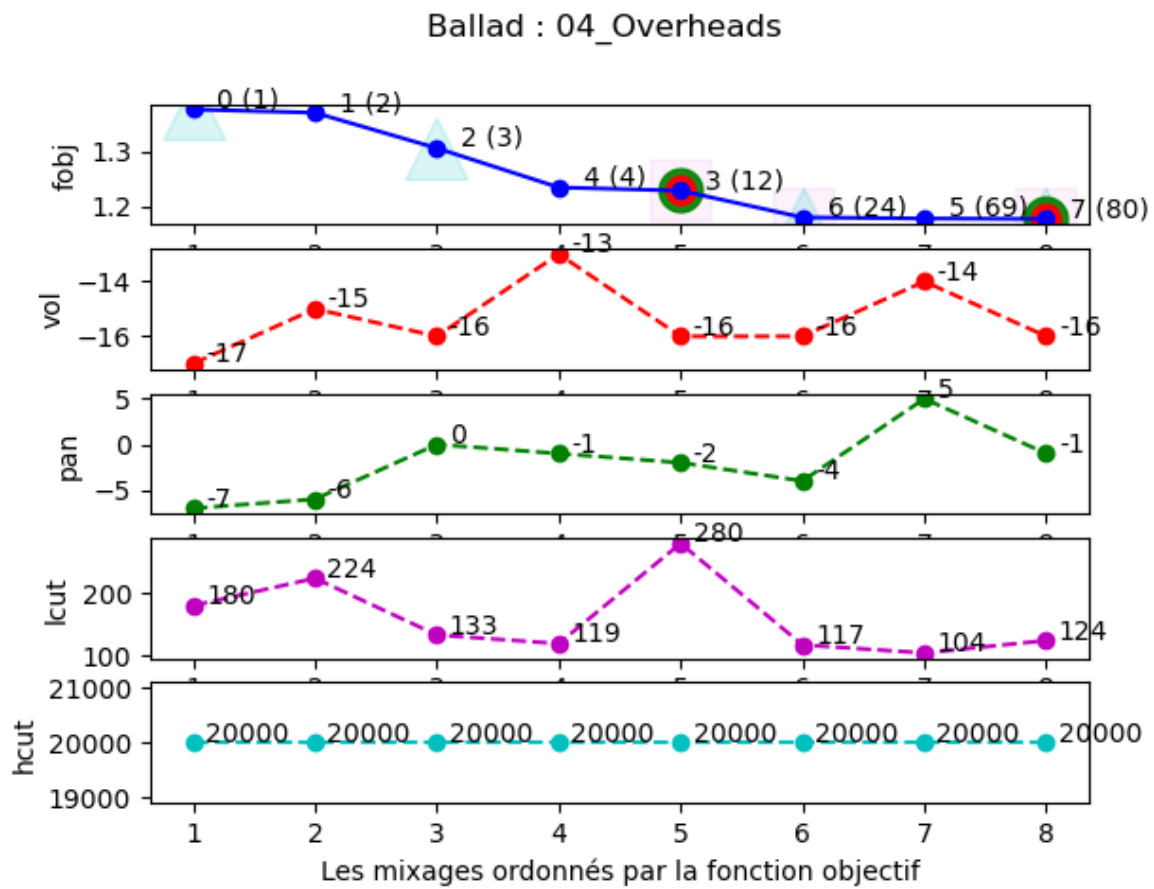


Figure E.93 – Ballad : Graphe de convergence – 04_Overheads.

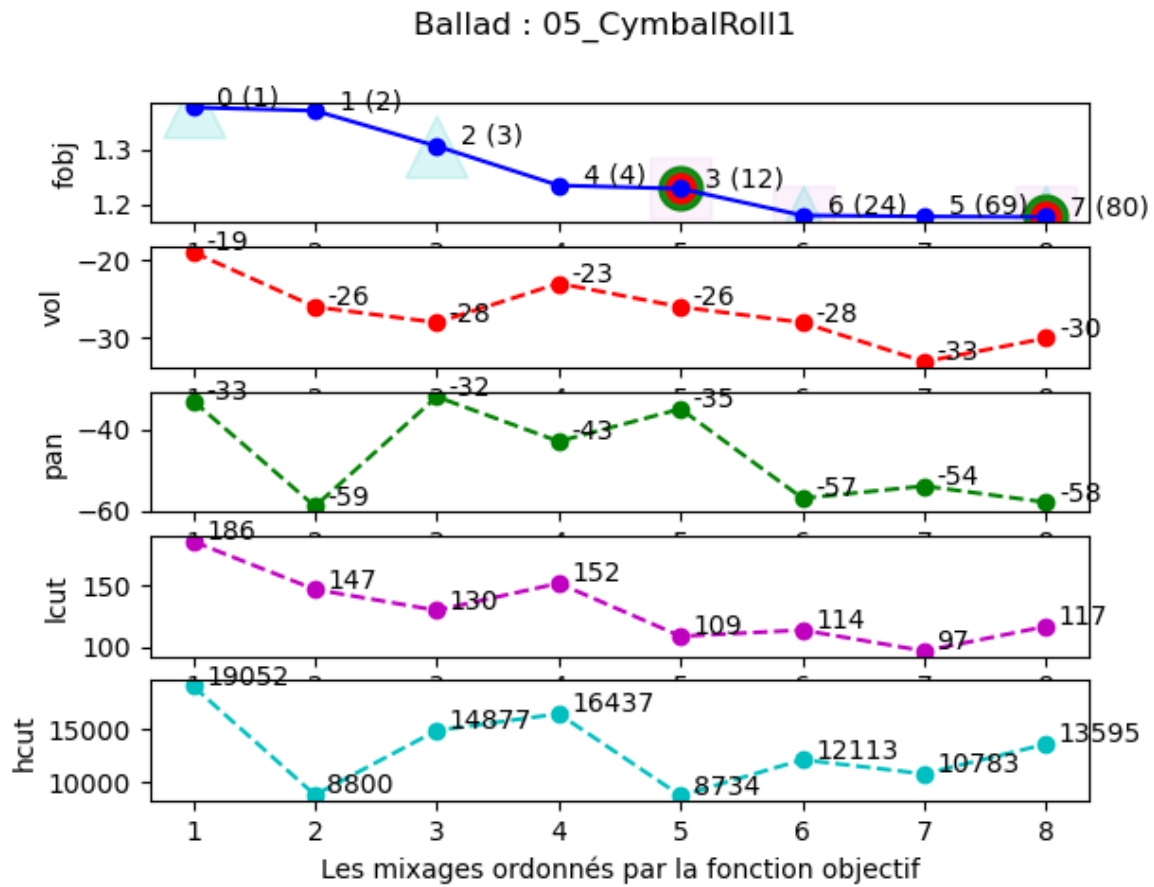


Figure E.94 – Ballad : Graphe de convergence – 05_CymbalRoll1.

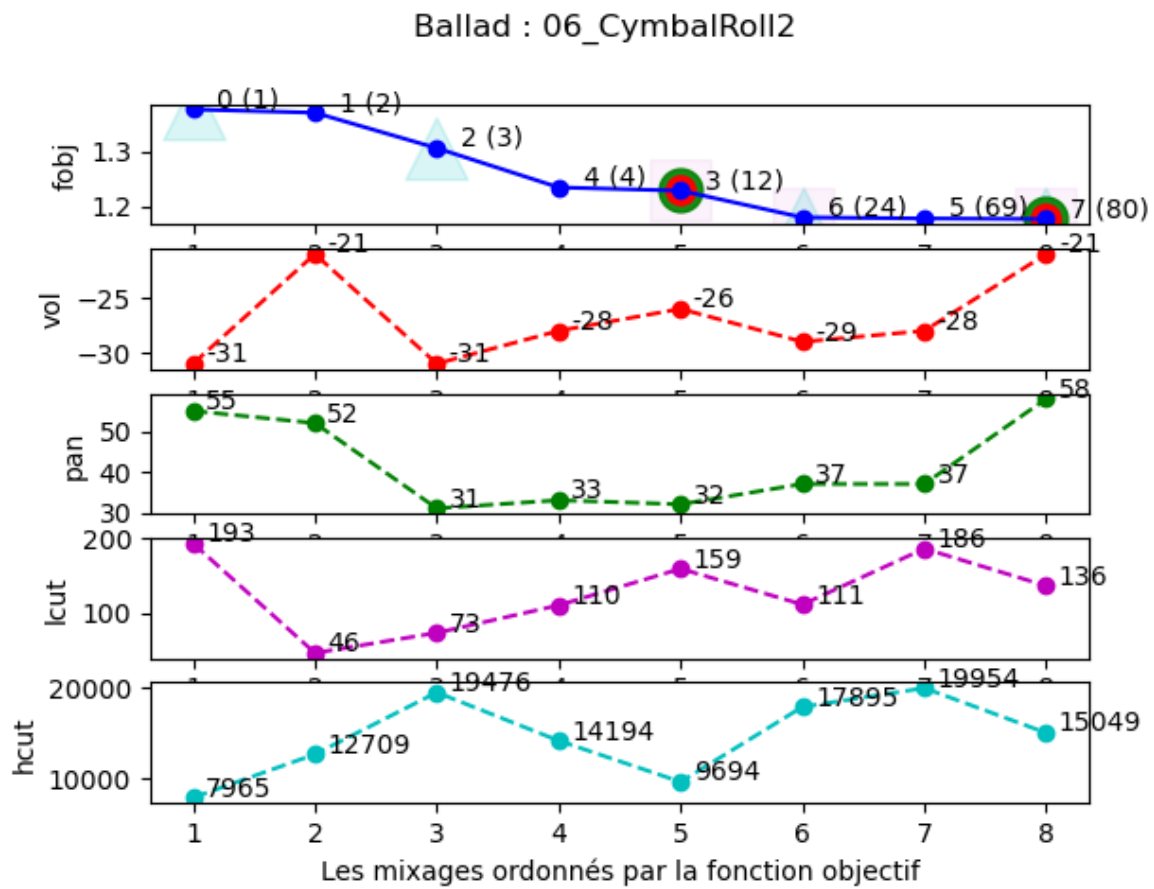


Figure E.95 – Ballad : Graphe de convergence – 06_CymbalRoll2.

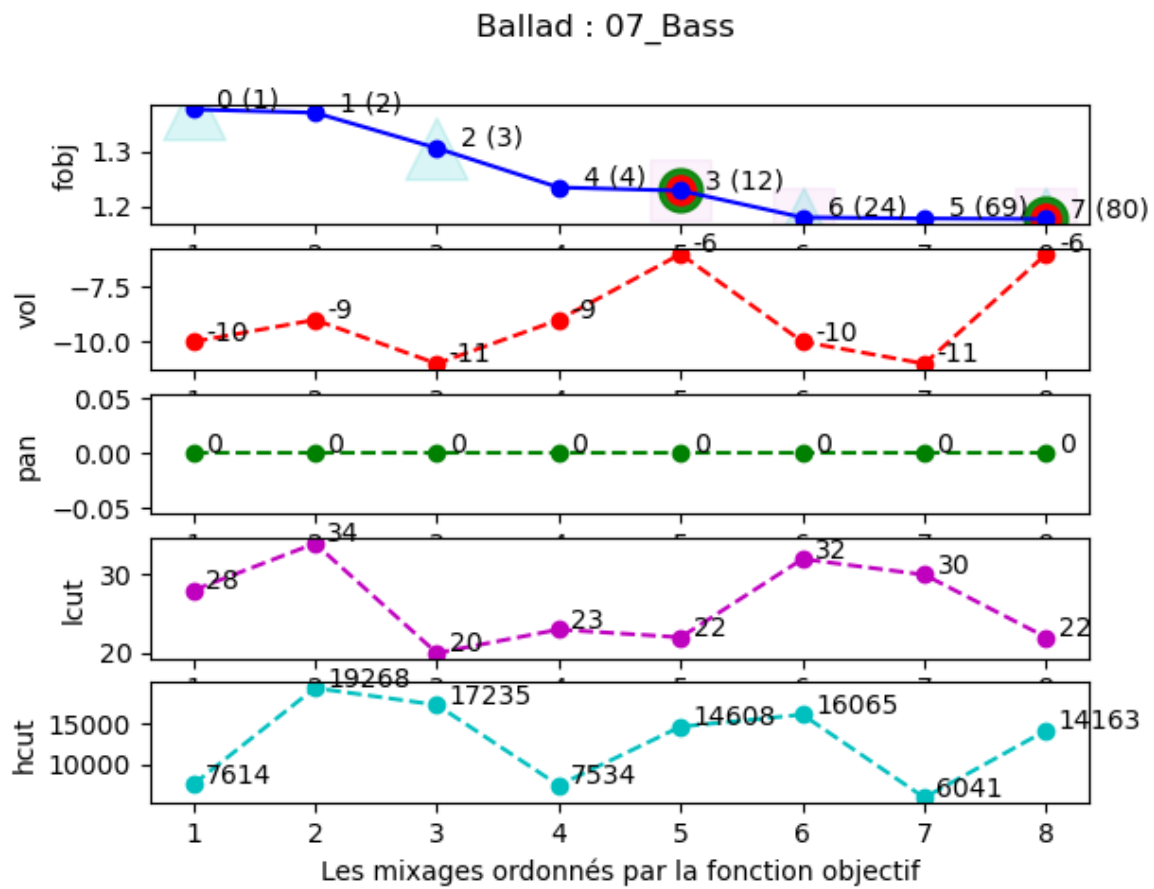


Figure E.96 – Ballad : Graphe de convergence – 07_Bass.

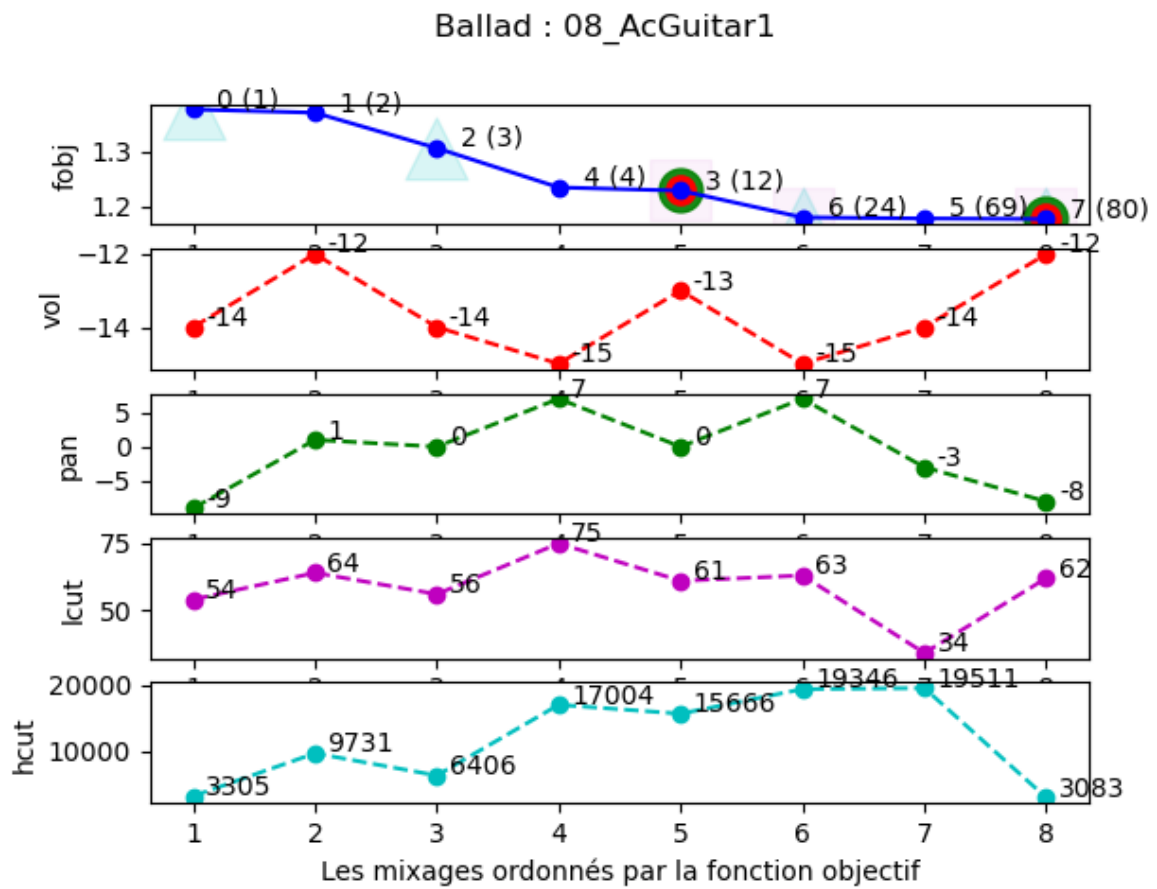


Figure E.97 – Ballad : Graphe de convergence – 08_AcGuitar1.

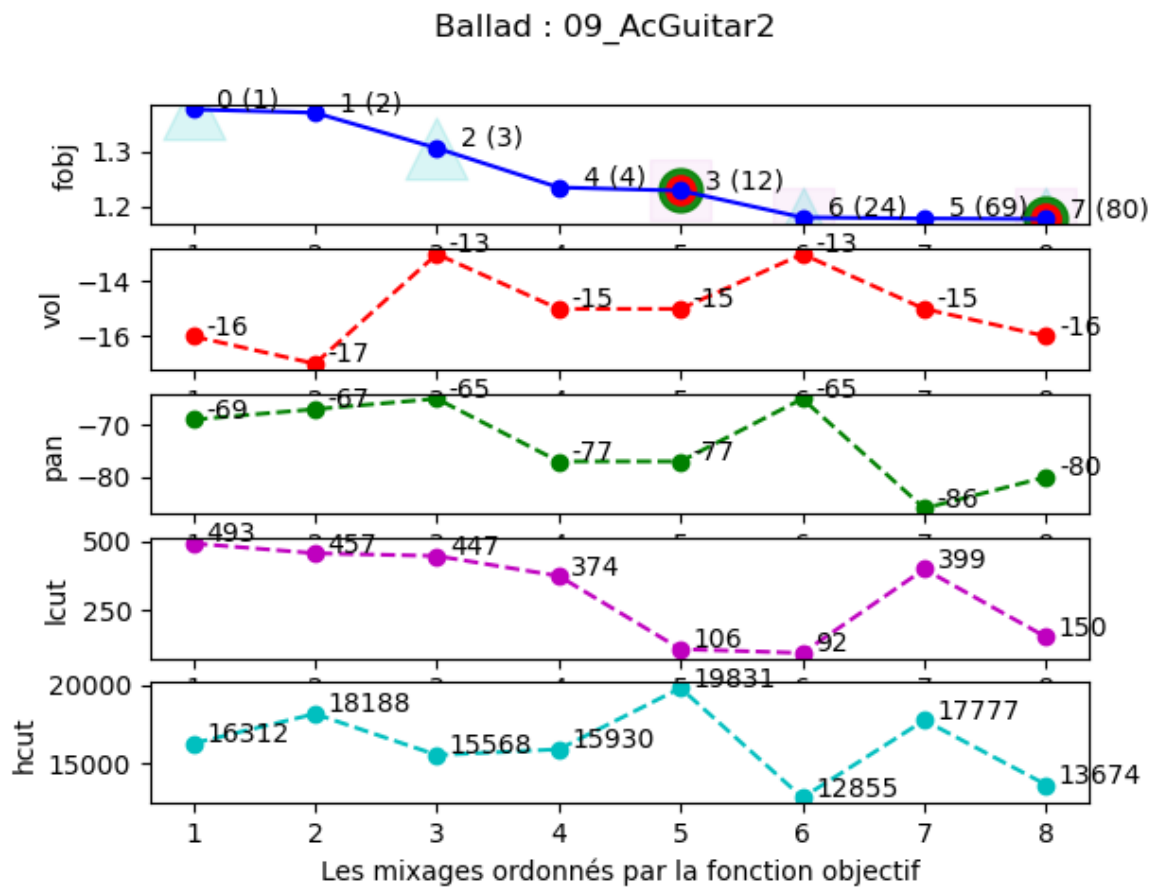


Figure E.98 – Ballad : Graphe de convergence – 09_AcGuitar2.

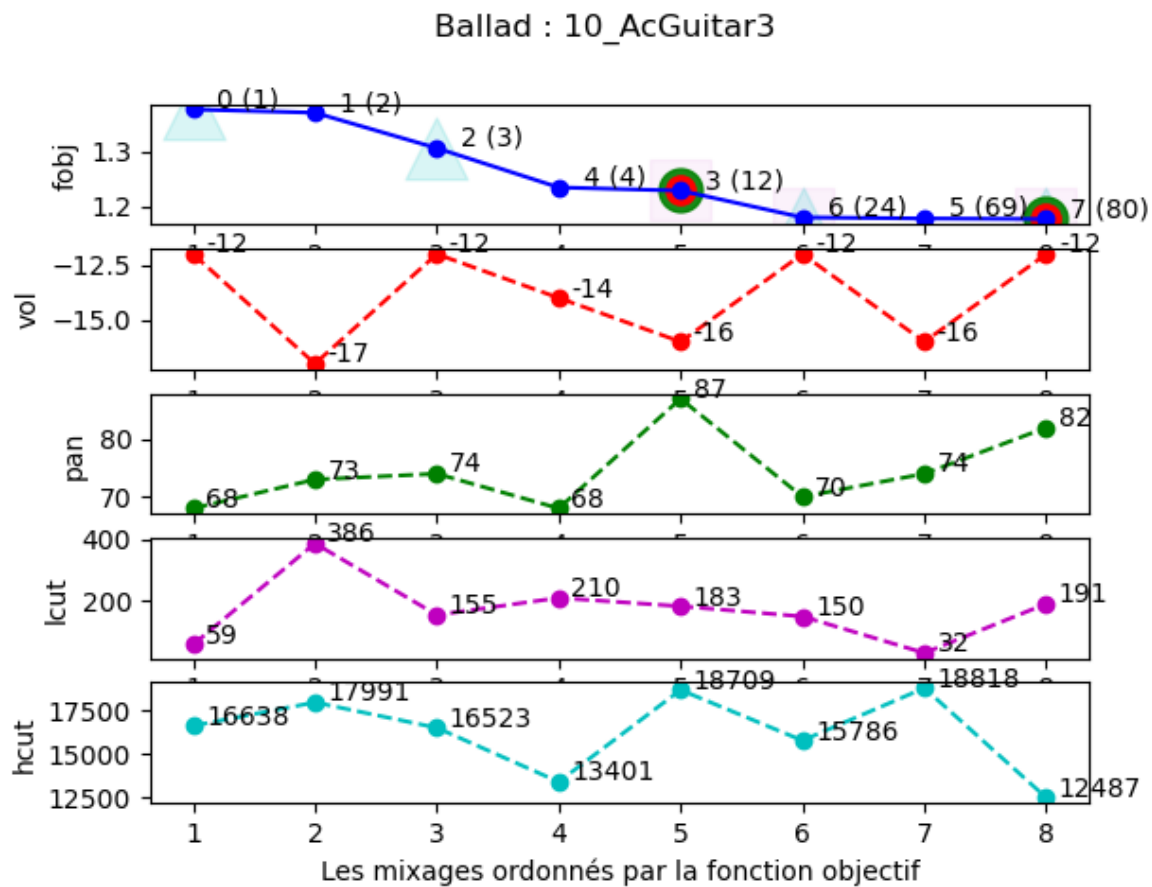


Figure E.99 – Ballad : Graphe de convergence – 10_AcGuitar3.

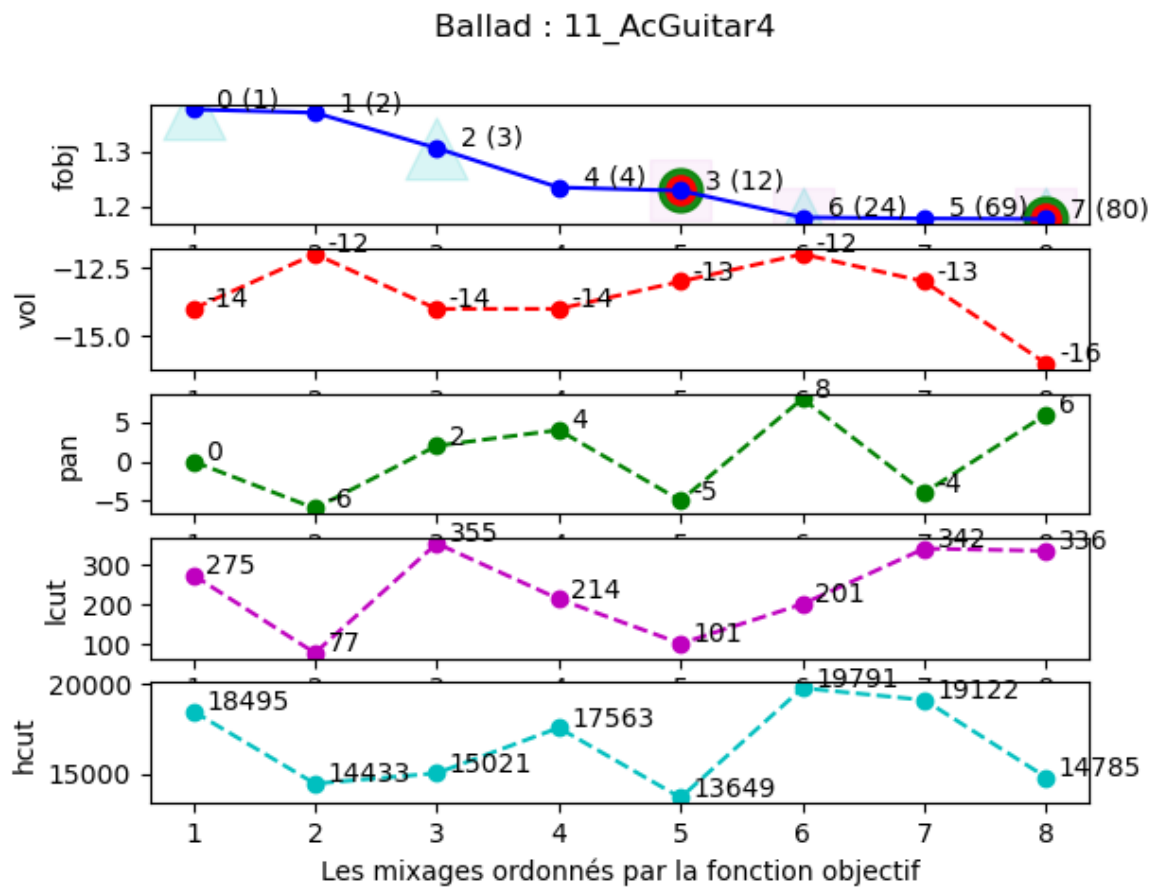


Figure E.100 – Ballad : Graphe de convergence – 11_AcGuitar4.

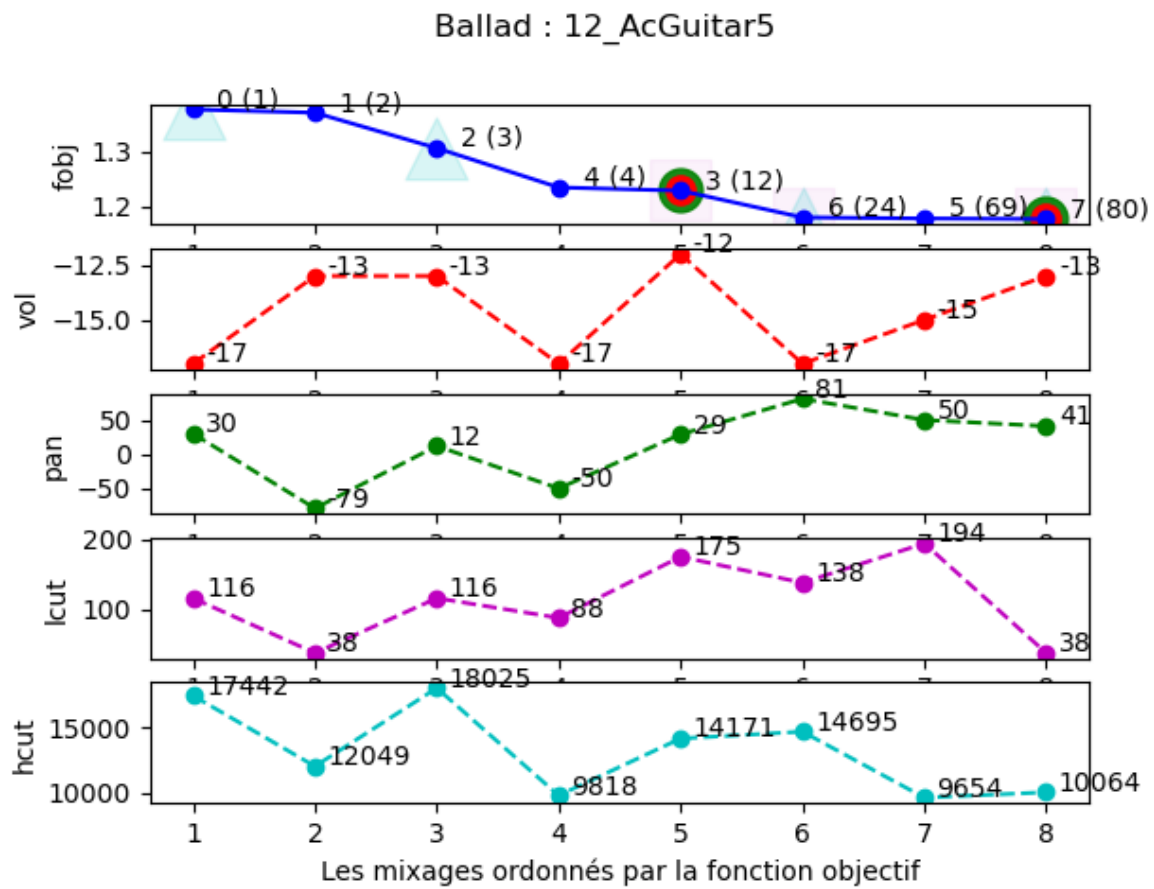


Figure E.101 – Ballad : Graphe de convergence – 12_AcGuitar5.

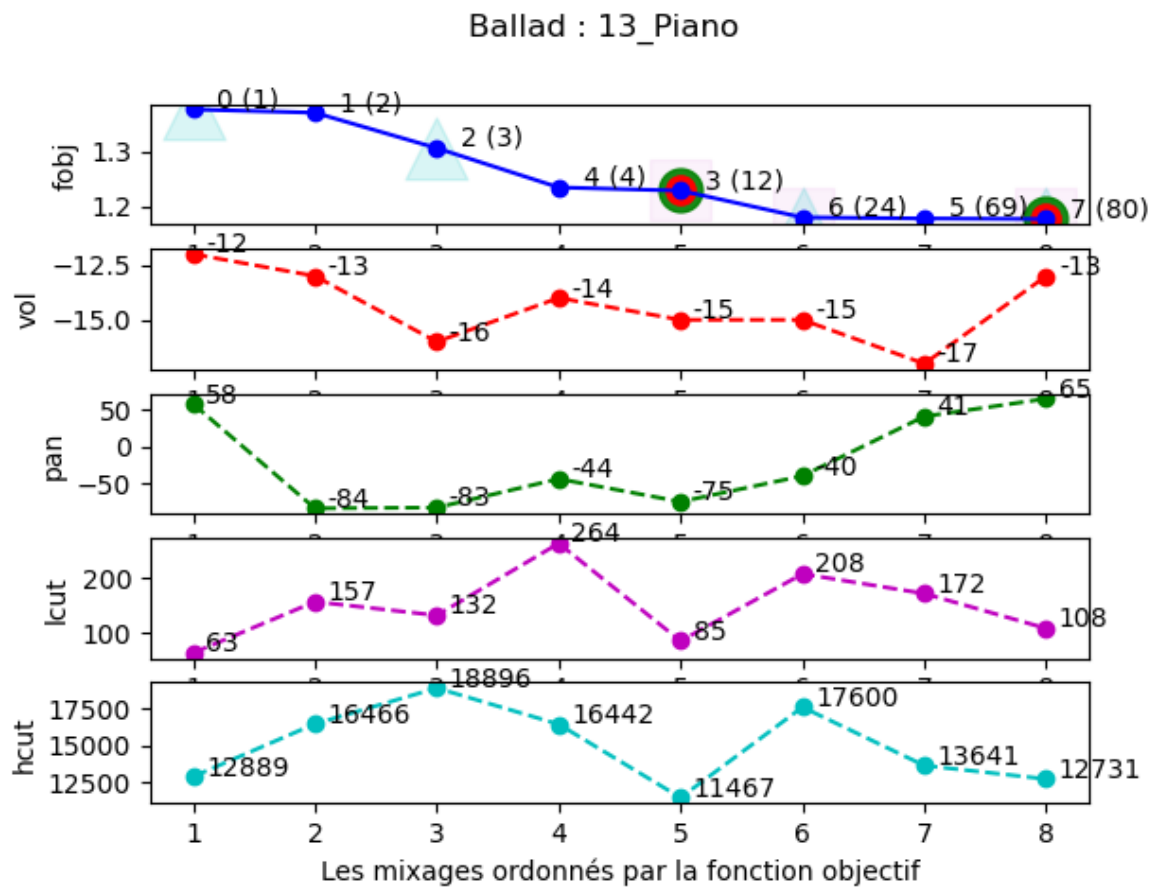


Figure E.102 – Ballad : Graphe de convergence – 13_Piano.

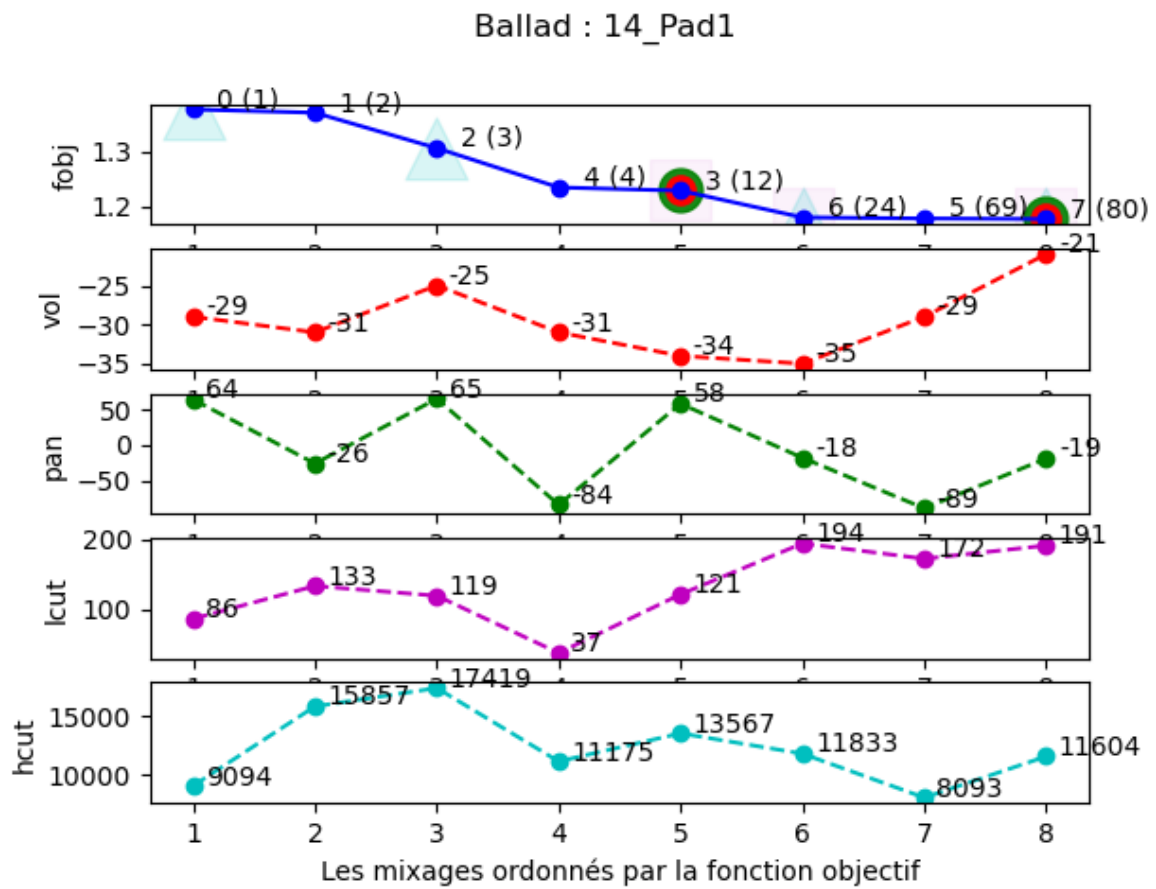


Figure E.103 – Ballad : Graphe de convergence – 14_Pad1.

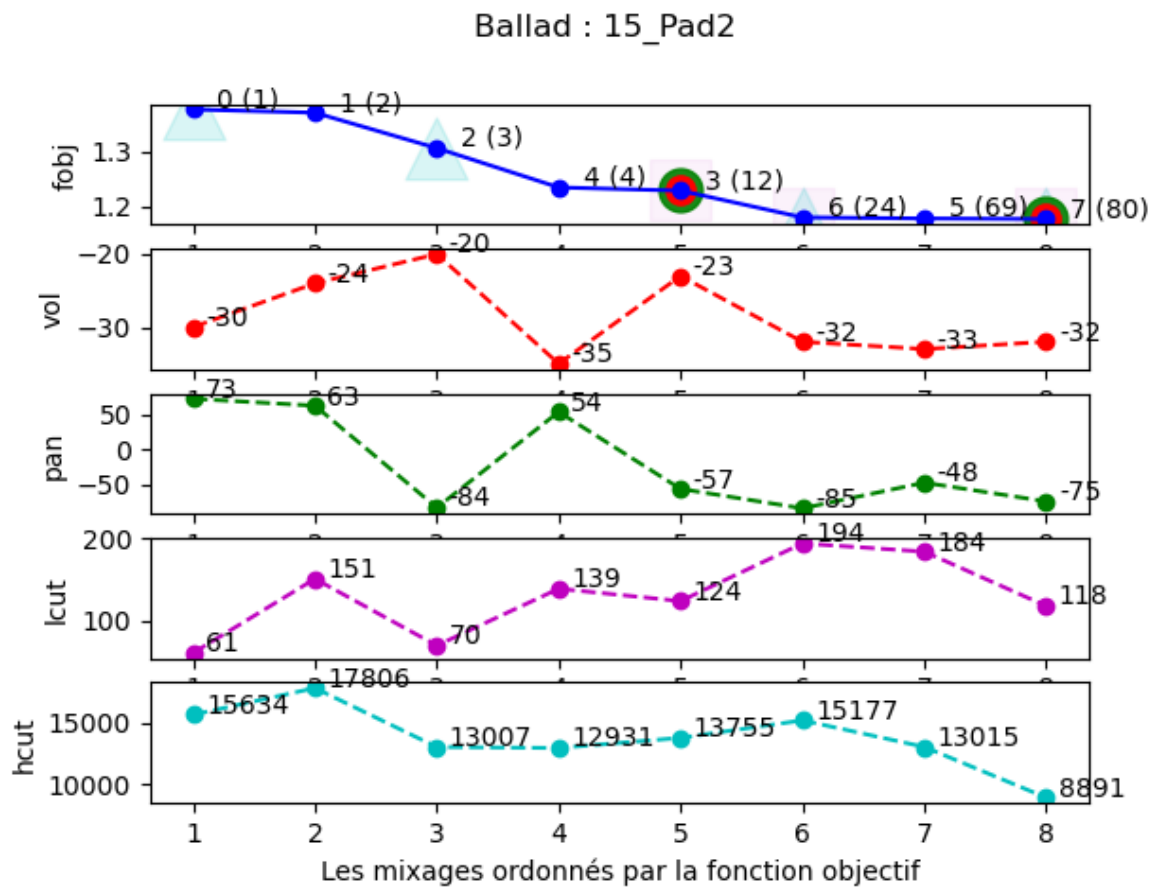


Figure E.104 – Ballad : Graphe de convergence – 15_Pad2.

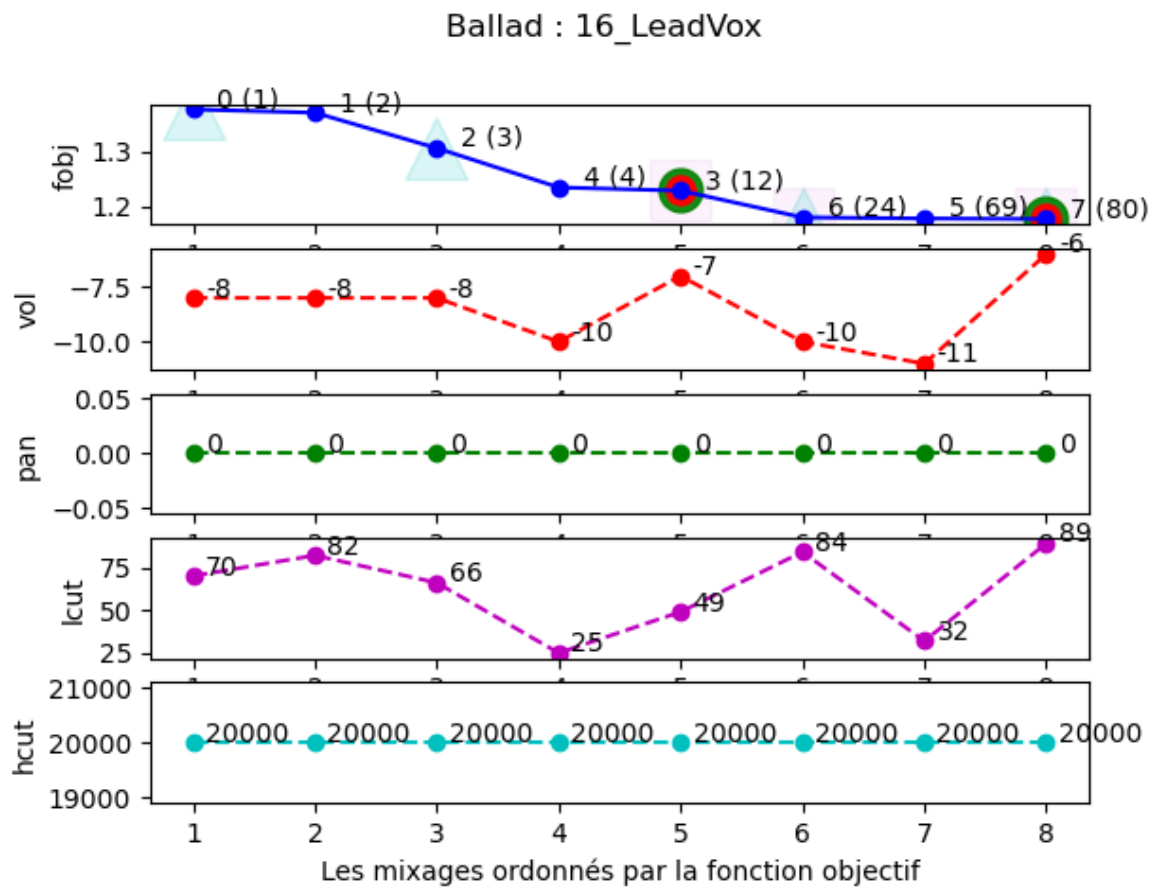


Figure E.105 – Ballad : Graphe de convergence – 16_LeadVox.