Co-tutelle
École Normale Supérieure de Cachan (ENS Cachan)
Université Catholique de Louvain (UCL)

THÈSE
Présentée pour obtenir le grade de
Docteur de l'École Normale Supérieure de Cachan
et
Docteur en sciences de l'ingénieur de l'École Polytechnique de Louvain

Métaheuristiques pour l'optimisation topologique

Application à la conception de dispositifs électromagnétiques

par
Jonathan DENIES
Soutenue publiquement le 10 septembre 2013

Membres du jury

Président :	Pr.	Francis Labrique	Pr. émérite (Belgique)
Rapporteurs :	Pr.	Christophe Espanet	UFC (France)
	Pr.	Patrick Siarry	UPEC (France)
Examinateurs :	Pr.	Bernard Multon	ENS (France)
	Pr.	Yves Perriard	EPFL (Suisse)
Dir. de thèse :	MCF-HDR	Hamid Ben Ahmed	ENS (France)
	Pr.	Bruno Dehez	UCL (Belgique)
	Pr.	François Glineur	UCL (Belgique)

Laboratoire SATIE, ENS-Cachan – antenne de Bretagne, Campus de Ker Lann,
avenue Robert Schuman, F-35170 Bruz, France
Laboratoire CEREM, École Polytechnique de Louvain,
2 place du Levant, B-1348 Louvain-la-Neuve, Belgique

«La nature fait toujours, selon les conditions dont elle dispose et autant que possible, les choses les plus belles et les meilleures.»

Aristote, 384-322 av J-C

Remerciements

Période charnière particulière entre les études et le monde du travail, une thèse nécessite un investissement personnel très important sur une période de plusieurs années. Heureusement, cette période ne se vit pas seul. C'est pourquoi, je réserve deux pages de ce manuscrit pour remercier les personnes qui ont partagé ce temps avec moi pour m'encourager lorsque je sentais une baisse de motivation et m'aider lorsque j'en avais le plus besoin.

Je remercie tout particulièrement mes directeurs de thèse, c'est-à-dire Hamid Ben Ahmed, Bruno Dehez et François Glineur, de m'avoir permis d'effectuer cette thèse à la fois en Belgique au laboratoire CEREM à l'École Polytechnique de Louvain mais aussi en France au laboratoire SATIE à l'ENS Cachan - antenne de Bretagne. Je les remercie aussi de m'avoir encadré tout au long de celle-ci et d'avoir surtout pris la patience de me corriger.

Je remercie Patrick Siarry et Christophe Espanet qui ont accepté d'être les rapporteurs de la thèse et je remercie aussi Francis Labrique pour avoir accepté la présidence du jury ainsi que pour avoir apporté un intérêt sincère à ce travail lors des différentes discussions qui ont précédées la défense.

Au niveau des laboratoires, puisque j'y ai passé une des plus grandes parties de mon temps ces dernières années, je remercie aussi les personnes qui ont travaillé autour de moi et avec moi, qui m'ont aidé ou qui ont créé une ambiance de recherche sérieuse enrichie de pauses régénératrices. Au sein de l'équipe, je remercie particulièrement Paul Sente du laboratoire CEREM sans qui mes nombreux problèmes informatiques liés à l'infrastructure n'auraient pu trouver de solution. Je remercie aussi Marie-Christine Vandingen, Antoinette Dupont et Ghislaine Gobert du laboratoire CEREM pour leur soutien non négligeable à tout ce qui tient

de la gestion administrative liée à ma thèse. Je n'oublie évidemment pas les moments passés avec ceux du laboratoire SATIE qui m'ont permis d'apprécier mes séjours en France.

Dans la vie quotidienne, je remercie aussi les personnes qui m'ont été très proches et qui ont surtout cru en moi. Par ces mots je veux parler de ma copine, de ma famille et de mes amis pour leur soutien plus que nécessaire. Je remercie particulièrement François Baudart pour avoir joué le double rôle de collègue de bureau et d'ami, et avec qui les pauses se sont souvent transformées en discussions enrichissantes à la fois pour ma thèse mais aussi pour mes différents projets. Je remercie aussi Caroline Jonckheere qui, avec François Baudart, a réussi à mettre une ambiance détendue dans notre bureau, tout en gardant un sérieux qui a nous permis à tous trois de terminer ce que l'on a commencé.

Finalement, je remercie toutes les personnes que j'ai pu rencontrer grâce à ma thèse et que je n'ai pas cité ici. Ces différentes rencontre ont aussi contribué de loin ou de près à la bonne continuité de ce travail qui s'achève.

Abstract

Topology optimization is a method of conception which is able to define the topology, the form and the dimensions of a device with the aim of responding optimally to given design criteria. Initially reserved to the sizing of mechanics parts, this method is directed today towards the conception of more complexes devices as those encountered in applied electromagnetic. It is in this context that our work was performed.

A topology optimization tool is made of the combination of an optimization algorithm and a material distribution formalism. In a first step, we compared different couplings of metaheuristic algorithms and material distribution formalisms. This comparison led us to choose as optimization tool for the problem under study, the combination of a genetic algorithm and a distribution of material by Voronoi cells. In a second step, we discussed how to improve the exploration and exploitation capabilities of this tool. We have, for this purpose, studied aspects related to the management of the size of the population and to the adaptation of the mechanisms of reproduction to the graphical nature of the problem.

After this second step, we builded our optimization tool that we tested on study cases whose complexity is similar to that encountered at industrial showing its potential of to design electromechanical devices.

Résumé

L'optimisation topologique est une méthode de conception qui permet de définir de manière autonome la topologie, les formes et les dimensions d'un dispositif en vue de répondre de manière optimale à des critères de design. Initialement réservée au dimensionnement de pièces mécaniques, elle s'oriente aujourd'hui vers la conception de dispositifs plus complexes comme ceux rencontrés dans le domaine de l'électromécanique. C'est dans ce cadre que se situe notre travail.

Un outil d'optimisation topologique étant formé de l'association d'un algorithme d'optimisation et d'un formalisme de distribution de matière, nous avons dans une première étape comparé différents couplages d'algorithmes métaheuristiques et de formalismes de distribution de matière en vue de choisir le couple qui semble le mieux adapté au problème traité. Cette comparaison nous a conduit à choisir comme outil d'optimisation l'association d'un algorithme génétique et d'une distribution de matière par cellules de Voronoï. Nous avons ensuite examiné comment améliorer les capacités d'exploration et d'exploitation de cet outil. Nous avons, à cet effet, étudié les aspects liés à la gestion de la taille de la population et à l'adaptation des mécanismes de reproduction au caractère graphique du problème.

A l'issue de cette deuxième étape, nous avons finalisé un outil d'optimisation que nous avons testé sur des cas d'étude dont la complexité se rapproche de celle rencontrée au niveau industriel. Nous avons ainsi montré le potentiel de notre outil d'optimisation au niveau de la conception dans le cadre de l'électromécanique.

Table des matières

Remerciements **i**

Abstract **iii**

Résumé **v**

Liste des principaux symboles **xi**

1 Introduction **1**
1.1 Formulation d'un problème d'optimisation 2
1.2 Méthodes de conception optimale 3
1.2.1 Optimisation dimensionnelle 4
1.2.2 Optimisation géométrique 5
1.2.3 Optimisation topologique 6
1.3 Structure d'un outil d'optimisation topologique 7
1.4 Classification des outils d'optimisation topologique 9
1.5 Cadre et objectifs de la thèse 10
1.6 Organisation du manuscrit 11

2 Etat de l'art **15**
2.1 Algorithmes d'optimisation 15
2.2 Formalismes de distribution de matière 21
2.3 Adaptations spécifiques 27
2.3.1 Évolution du nombre de variables de design 28
2.3.2 Diminution de l'espace des solutions par utilisation des contraintes 30
2.3.3 Suppression des éléments non connectés 31
2.4 Synthèse 33

3 Stratégie de développement des outils d'optimisation topologique métaheuristiques 35
3.1 Méthodologie 35
3.1.1 Phase de sélection (Chapitre 4) 36
3.1.2 Phase d'adaptation (Chapitre 5 et Chapitre 6) . . 36
3.1.3 Phase des tests (Chapitre 7) 37
3.2 Les cas d'étude 37
3.2.1 Phase 1 : Problème simple graphique 37
3.2.2 Phase 2 : Problème simple de type physique 39
3.2.3 Phase 3 : Problèmes complexes 43
3.3 Critères de comparaison 48
3.3.1 Courbe de convergence 49
3.3.2 Courbe de sensibilité sur un paramètre 49
3.4 Synthèse 51

4 Sélection de l'outil de référence 53
4.1 Algorithmes d'optimisation 53
4.1.1 Sélection 54
4.1.2 Implémentation 55
4.2 Formalismes de distribution de matière 69
4.2.1 Sélection 70
4.2.2 Implémentation 71
4.3 Comparaison des combinaisons 74
4.3.1 Sensibilité 74
4.3.2 Convergence 85
4.4 Synthèse 87

5 Adaptation de la gestion de la population 91
5.1 Evolution de la taille de la population 92
5.1.1 Décroissance préprogrammée à priori : $N_s = N_s(k)$ 93
5.1.2 Régulation par le rendement de chaque génération : $N_s = N_s(N_b)$ 97
5.1.3 Régulation par attribution d'un coefficient de reproduction : $N_{se} = N_{se}(\vec{c_r})$ 100
5.2 Initialisation de la population 104
5.2.1 Prédéfinir la population initiale par anticipation . 105
5.2.2 Recouvrement équilibré de l'espace de design par les matériaux 107

5.2.3 Influence des dimensions de l'espace de design . . . 109
5.3 Synthèse . . . 111

6 Adaptation de la reproduction 115
6.1 Croisements . . . 115
6.1.1 Description des opérateurs de croisements . . . 116
6.1.2 Comparaisons . . . 121
6.1.3 Cohérence du croisement . . . 122
6.2 Mutations . . . 123
6.2.1 Déplacements des centres de Voronoï . . . 124
6.2.2 Modification du nombre de cellules . . . 138
6.3 Synthèse . . . 164

7 Constitution de l'outil final et application aux cas tests 167
7.1 Constitution de l'outil final . . . 167
7.1.1 Population . . . 167
7.1.2 Reproduction . . . 169
7.2 Description des outils de référence . . . 171
7.2.1 Description de l'outil de Schoenauer . . . 172
7.2.2 Description de l'outil de Ishikawa . . . 172
7.3 Cas d'étude 1 : le moteur linéaire à réluctance variable . . . 174
7.3.1 Résultats . . . 174
7.3.2 Discussions . . . 179
7.4 Cas d'étude 2 : moteur synchrone . . . 180
7.4.1 Résultats . . . 181
7.4.2 Discussions . . . 183
7.5 Synthèse . . . 185

8 Perspectives 187
8.1 Population . . . 187
8.1.1 Description du parallélisme de la reproduction . . . 187
8.1.2 Variation de la taille des populations des enfants . . . 188
8.2 Croisement . . . 191
8.2.1 Croisement graphique non équilibré . . . 192
8.2.2 Croisement par superposition . . . 194
8.3 Mutations des matériaux . . . 194
8.3.1 Choix du taux de mutation en fonction du voisinage 195

8.3.2 Choix du taux de mutation en fonction de l'historique de la population . . . 196
8.3.3 Choix du taux de mutation en fonction de l'évolution de la surface de la cellule . . . 197
8.4 Hybridation . . . 197
8.5 Synthèse . . . 199

9 Conclusions **201**
9.1 Etape I : Sélection d'un outil d'optimisation topologique . 202
9.2 Etape II : Adaptation de l'algorithme génétique à l'aspect graphique de l'optimisation topologique . . . 204
9.2.1 Gestion de la population . . . 204
9.2.2 Gestion de la reproduction . . . 205
9.3 Etape III : Composition et test de l'outil final . . . 207
9.4 Perspectives et contributions supplémentaires . . . 208

Publications associées à la thèse **209**

A Choix d'implémentation des algorithmes **211**
A.1 Cas d'étude . . . 212
A.2 Algorithme génétique . . . 212
A.2.1 Sélection des parents et des descendants . . . 213
A.2.2 Croisement . . . 217
A.2.3 Mutation . . . 220
A.3 Algorithme du recuit simulé . . . 221
A.3.1 Description . . . 221
A.3.2 Résultats . . . 222

Références **224**

Liste des principaux symboles

GA	- Algorithme génétique
PSO	- Algorithme de l'essaim particulaire
SA	- Algorithme du recuit simulé
IA	- Algorithme immunitaire
k	- Numéro d'itération du processus d'optimisation
k_{max}	- Nombre maximum d'itérations alloué au processus d'optimisation
N_{ev}	- Nombre d'évaluations du processus d'optimisation
$N_{ev,max}$	- Nombre maximum d'évaluations alloué au processus d'optimisation
$\vec{f}$	- Vecteur d'évaluation associé à un problème d'optimisation
N_D	- Nombre de variables discrètes d'optimisation
N_p	- Nombre total de variables d'optimisation
N_c	- Nombre total de cellules de Voronoï
N_s	- Nombre de solution dans la population principale
N_{se}	- Nombre de solution dans la population des enfants
$\mathbf{X}$	- Matrice de variables d'optimisation décrivant une population
$\vec{x}$	- Vecteur de variables d'optimisation décrivant une solution/un individu
x	- Variable d'optimisation
p_c	- Probabilité de croisement de l'algorithme génétique
p_m	- Probabilité de mutation

N_{clone}	- Nombre de clones d'un individu pour l'algorithme immunitaire
N_{sel}	- Nombre d'individus sélectionnés pour le clonage pour l'algorithme immunitaire
β	- Coefficient de clonage
p_0	- Probabilité initiale de mutation pour le recuit simulé
ω	- Paramètre d'inertie de l'algorithme de l'essaim particulaire
α_0	- Paramètre de mémoire de l'algorithme de l'essaim particulaire
α_1	- Paramètre de groupe de l'algorithme de l'essaim particulaire
γ, Γ	- Paramètre de rebond de l'algorithme de l'essaim particulaire
p_v	- Probabilité de suppression d'une cellule de Voronoï par vieillissement cellulaire
p_p	- Probabilité de prolifération cellulaire d'une cellule de Voronoï
α	- Amplitude maximale de déplacement des centres de Voronoï exprimé en % sur les dimensions de l'espace de design
	- Age des cellules de Voronoï pour le vieillissement cellulaire et la prolifération cellulaire

Chapitre 1

Introduction

Dans le cadre de la conception optimale de dispositifs en ingénierie, le concepteur est amené à remplir un cahier des charges reprenant les fonctions, les contraintes et les performances que l'objet de conception doit respectivement réaliser, respecter et maximiser. Le concepteur détermine ensuite, sur la base de ce cahier des charges, une solution qu'il peut être amené à faire évoluer afin d'en améliorer les performances. Pour cela, il doit faire varier un certain nombre de paramètres pour trouver un optimum. Cette dernière étape peut être accélérée et améliorée par l'utilisation d'outils d'optimisation.

Dans un contexte où l'on cherche à produire de plus en plus vite des dispositifs de plus en plus performants, les ingénieurs et chercheurs se retrouvent face à une complexité de conception croissante avec des délais d'études de plus en plus courts. C'est pourquoi les outils d'optimisation prennent une place de plus en plus importante dans le processus de conception. Depuis les débuts de l'ère informatique, de nombreux outils d'optimisation automatique ont émergé. L'intérêt de développer de nouveaux outils plus performants faisant intervenir de moins en moins l'expérience du concepteur est soutenu par l'explosion de la puissance disponible du calcul numérique.

Ce chapitre introductif a pour objectif de présenter quelques points essentiels permettant de faciliter la lecture du manuscrit. Avant de présenter la recherche en elle-même, il convient de formaliser ce que l'on entend par problème d'optimisation, section 1.1, ainsi que de décrire sommairement les différentes approches utilisées dans le domaine de la conception optimale, section 1.2. La section 1.3 détaille la structure gé-

nérale des outils d'optimisation. La section 1.4 expose le cadre de travail ainsi que la méthodologie suivie qui va conditionner la recherche. Et finalement, la section 1.5 décrit l'organisation du manuscrit.

1.1 Formulation d'un problème d'optimisation

Dans l'objectif de trouver une solution optimale à un cas d'étude, il est nécessaire de traduire une description qualitative, par exemple maximiser le rendement d'une machine, par une modélisation mathématique quantitative [1]. Cette étape de quantification définit les variables d'optimisation, les fonctions objectifs ainsi que les contraintes.

Les variables d'optimisation, qui seront notées par une lettre minuscule italique x, sont les éléments descriptifs qui caractérisent chaque solution. Elles peuvent se classer en deux familles : les variables discrètes et les variables continues. Pour améliorer la lecture, l'ensemble des variables d'optimisation caractérisant une solution sont regroupées sous la forme vectorielle suivante :

$$\vec{x} = \left\{ \underbrace{x_1, x_2, \ldots, x_{N_D}}_{\text{Variables discrètes}}, \underbrace{x_{N_D+1}, x_{N_D+2}, \ldots, x_{N_p}}_{\text{Variables continues}} \right\}, \tag{1.1}$$

où N_D et N_p représentent respectivement le nombre de variables discrètes et le nombre total de variables d'optimisation. Puisqu'un vecteur représente une solution, une matrice où chaque ligne est associée à une solution représente un ensemble de N_s solutions. Pour les besoins des notations, l'indice des scalaires est doublé et on ajoute un indice pour les vecteurs afin d'identifier la solution. Ceci nous donne :

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N_p} \\ \vdots & \ddots & \vdots \\ x_{N_s,1} & \cdots & x_{N_s,N_p} \end{bmatrix} = \begin{bmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_{N_s} \end{bmatrix} \tag{1.2}$$

où $\{x_{i,k}\}$ avec $k \in [1, N_p]$ représente la solution i. Cette solution peut également être représentée par $\vec{x}_i$.

Les fonctions objectifs, $F_i(\vec{x})$, sont des liens quantitatifs entre le jeu de variables d'optimisation et les objectifs visés. Elles sont utilisées pour évaluer la solution :

$$f_i = F_i\left(\vec{x}\right), \tag{1.3}$$

ce qui nous donne le vecteur d'évaluation $\vec{f} = \left\{f_1, f_2, \ldots, f_{N_f}\right\}$ où N_f est le nombre de fonctions objectifs. Par principe et surtout par facilité d'écriture, dans le cadre de cette thèse les problèmes d'optimisation ont pour objectif de minimiser le vecteur d'évaluation, sauf mention spécifique de maximisation et ceci sans véritable perte de généralité.

Des fonctions de contraintes limitent ou sont utilisées pour limiter l'espace de recherche, ou d'un autre point de vue l'espace des solutions. On distingue deux types de contraintes : les contraintes d'égalité,

$$C_i\left(\vec{x}\right) = 0, \tag{1.4}$$

et les contraintes d'inégalité,

$$G_i\left(\vec{x}\right) \leq 0 \tag{1.5}$$

Étant donné que les outils d'optimisation sont généralement basés sur un calcul itératif et afin d'identifier la valeur d'une variable d'optimisation à une itération k donnée, les notations deviennent $\mathbf{X} \rightarrow \mathbf{X}^{(k)}$, $\vec{x}_i \rightarrow \vec{x}_i^{(k)}$ et $x_{i,j} \rightarrow x_{i,j}^{(k)}$:

$$\mathbf{X}^{(k)} = \begin{bmatrix} x_{1,1}^{(k)} & \cdots & x_{1,N_p}^{(k)} \\ \vdots & \ddots & \vdots \\ x_{N_s,1}^{(k)} & \cdots & x_{N_s,N_p}^{(k)} \end{bmatrix} = \begin{bmatrix} \vec{x}_1^{(k)} \\ \vdots \\ \vec{x}_{N_s}^{(k)} \end{bmatrix} \tag{1.6}$$

1.2 Méthodes de conception optimale

En fonction de l'approche suivie pour atteindre les objectifs d'optimisation fixés, les méthodes de conception optimale peuvent être différentes. On peut scinder en première approximation ces méthodes en trois grandes familles : l'optimisation dimensionnelle, l'optimisation géométrique et l'optimisation topologique. Cette distinction entre trois familles repose sur l'interprétation des paramètres et de ses conséquences sur le rôle à jouer par le concepteur.

Pour mieux distinguer les différences entre ces trois familles de méthodes, celles-ci sont illustrées par la suite à l'aide d'un exemple de

problème à optimiser. Le problème utilisé est l'optimisation par la minimisation de sa flèche d d'une poutre encastrée, dont la longueur L est fixée, sous la contrainte d'un volume maximal de matériaux à ne pas dépasser (Figure 1.1). Cette poutre est sollicitée, à son extrémité, par une force verticale donnée et dirigée vers le bas. Le problème se limite à l'utilisation d'un seul matériau de construction.

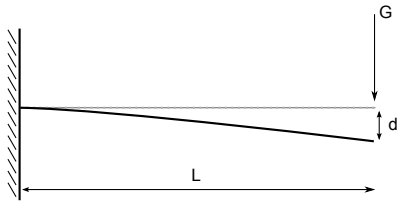

Figure 1.1 – Schéma du problème de minimisation de la flèche d'une poutre encastrée

Pour résoudre la question, il nous faut poser le problème d'optimisation, c'est-à-dire définir les variables d'optimisation, $\vec{x}$, qui vont permettre de décrire la solution, c'est-à-dire la poutre. En fonction du rôle de ces variables, l'outil d'optimisation prendra une place différente dans le processus de conception. Les trois sections suivantes décrivent les trois familles en fonction de cette interprétation des variables d'optimisation.

1.2.1 Optimisation dimensionnelle

L'optimisation dimensionnelle repose sur la modification directe des dimensions d'une solution. Le concepteur détermine la topologie et la forme de la solution initiale, ainsi que des paramètres dimensionnels (Figure 1.2). Des variables d'optimisation sont associées à ces paramètres et sont utilisées pour rechercher les meilleures dimensions de la solution, sans modifier la forme et la topologie prédéfinie par le concepteur.

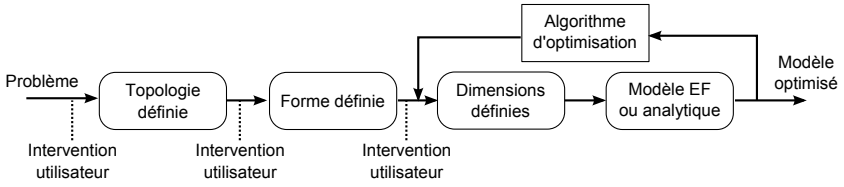

Figure 1.2 – Schéma fonctionnel de l'optimisation dimensionnelle

Pour le cas de la poutre encastrée, deux exemples de choix de variables d'optimisation dimensionnelle sont illustrés par la Figure 1.3. L'exemple de gauche présente une situation avec deux variables d'optimisation, X et Y, associées à la largeur de la poutre. L'exemple de droite représente une autre situation avec cinq variables d'optimisation : X, Y, Z, A et B.

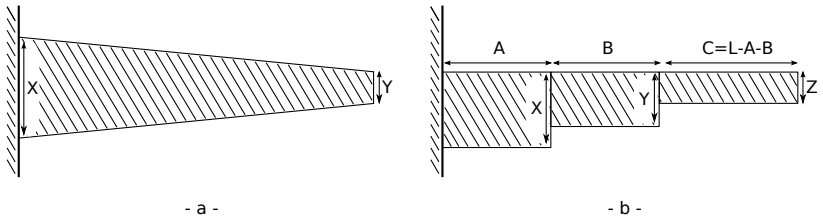

Figure 1.3 – Exemples de paramétrisation pour le problème de minimisation de la flèche d'une poutre encastrée : optimisation dimensionnelle

1.2.2 Optimisation géométrique

L'optimisation géométrique repose sur la modification de la géométrie d'une solution. Le concepteur détermine la topologie de la solution initiale, ainsi que des paramètres de description géométrique (Figure 1.4). Les variables d'optimisation correspondent aux coordonnées et sont modifiées par l'algorithme d'optimisation pour changer la géométrie de la solution initiale, généralement sans modifier sa topologie.

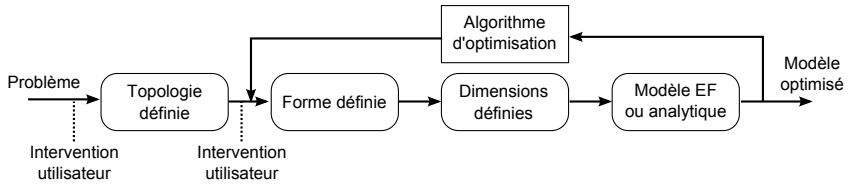

Figure 1.4 – Schéma fonctionnel de l'optimisation géométrique

Pour le cas de la poutre encastrée, deux exemples de choix de variables d'optimisation géométriques sont illustrés par la Figure 1.5. L'exemple de gauche présente une situation modifiant la courbure de la partie inférieure de la solution initiale et décrite par les points A, B, ..., M. L'exemple de droite permet de travailler avec un trou dans la

structure initiale décrit par les points N, O, ..., T. La création de ce trou peut permettre d'alléger la structure dans les espaces les moins sollicités afin de consolider d'autres parties de la structure plus sollicitées par la répartition des charges.

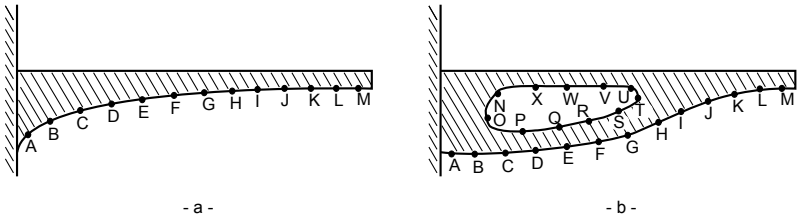

Figure 1.5 – Exemple de paramétrisation pour le problème de minimisation de la flèche d'une poutre encastrée : optimisation géométrique

1.2.3 Optimisation topologique

L'optimisation topologique repose sur la modification de la topologie d'une solution. Le concepteur intervient dans la définition des paramètres d'optimisation topologique (Figure 1.6). Ces paramètres, associés aux variables d'optimisation, permettent de décrire la distribution de matière au sein de l'espace de design. Par la modification des valeurs de ces variables d'optimisation, l'algorithme d'optimisation a la possibilité de tester différentes topologies, ainsi que différentes formes et dimensions.

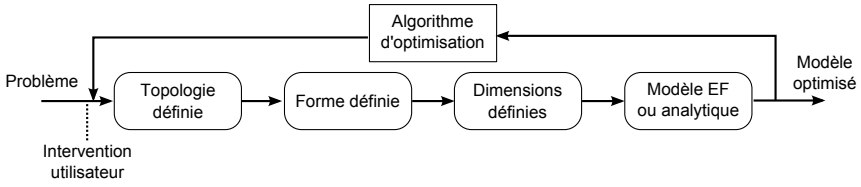

Figure 1.6 – Schéma fonctionnel de l'optimisation topologique

Cette interprétation des variables d'optimisation en distribution de matière nécessite de discrétiser l'espace en sous-domaines de matière de distribution homogène. Au sein de ces sous-domaines, il est possible de déterminer un ensemble de caractéristiques de la matière sous forme

de paramètres d'optimisation ou tout simplement de se limiter à des matériaux prédéfinis dans une bibliothèque de matériaux.

L'optimisation topologique est, sur la base du problème de la poutre encastrée, illustrée ici par la Figure 1.7. Un espace de design est d'abord défini (Figure 1.7 -a-) et est discrétisé en un quadrillage régulier fixe (Figure 1.7 -b-). Chacun de ces sous-domaines de matière est aussi appelé cellule ou élément. Les variables d'optimisation correspondent à la valeur A ou B que devra prendre la cellule associée.

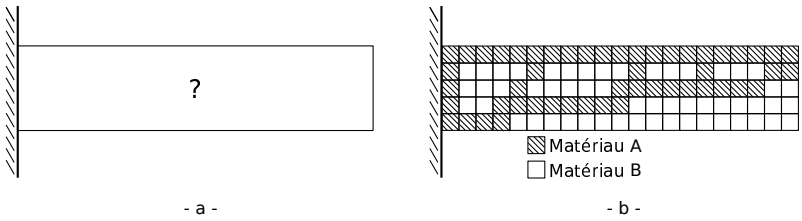

Figure 1.7 – Exemple de paramétrisation pour le problème de minimisation de la flèche d'une poutre encastrée : optimisation topologique

1.3 Structure d'un outil d'optimisation topologique

La structure d'un outil d'optimisation topologique peut généralement être décrite sous la forme d'une combinaison de trois modules complémentaires. Ces trois modules communiquent entre eux de la façon décrite par le schéma de la Figure 1.8. Ces trois modules sont :

- le module pour l'optimisation de la valeur des variables d'optimisation ;
- le module pour la description de la distribution de matière ;
- le module pour l'évaluation de la solution.

Le module d'optimisation, composé d'un algorithme d'optimisation, utilise différentes informations, telles que les valeurs des fonctions objectifs, les contraintes, le gradient, etc. pour modifier la valeur des variables d'optimisation avec pour objectif d'améliorer la ou les fonctions objectifs tout en essayant de respecter la ou les contraintes

Le module de description de la solution utilise un formalisme de distribution de matière pour déterminer la manière de décrire une solu-

tion au travers d'une distribution de matière à partir d'un ensemble de variables d'optimisation.

Le module d'évaluation, utilisant un logiciel éléments-finis, fonctionne en trois étapes. Premièrement, le module récupère la description de la distribution de matière pour créer un modèle éléments-finis. Deuxièmement, le modèle éléments-finis est résolu en fonction de la ou des « physiques » souhaitées. Et troisièmement, il évalue les différentes informations nécessaires à l'algorithme d'optimisation, telles que les valeurs des fonctions objectifs, les contraintes, etc.

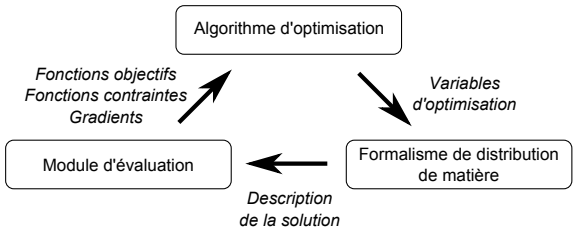

Figure 1.8 – Les trois modules composant un outil d'optimisation topologique

Cependant, bien que jouant des rôles distincts, ces trois modules ne sont pas toujours indépendants. Le choix de chacun peut influencer les autres et détermine aussi les capacités et le comportement de l'outil d'optimisation topologique.

C'est ainsi que le choix de l'algorithme d'optimisation va conditionner la capacité de l'outil d'optimisation topologique d'explorer les différentes topologies par de grandes modifications de la structure, lors d'une première étape, l'étape d'exploration, et la capacité de modifier la forme et les dimensions d'une géométrie par de petites modifications de la structure, lors d'une deuxième étape, l'étape d'exploitation. L'algorithme va aussi conditionner les types de variables possibles, la capacité de gérer un grand nombre de variables, ainsi que le nombre de fonctions objectifs et de contraintes.

Ensuite, le choix du formalisme de distribution de matière va conditionner la taille de l'espace des solutions, les types de variables d'optimisation, le nombre de variables d'optimisation nécessaire pour décrire une solution donnée avec un degré de précision acceptable, la capacité/facilité d'exploration de l'espace des solutions et la capacité d'exploitation de la recherche en intensifiant celle-ci dans le voisinage d'une solution.

Et finalement, le choix du module d'évaluation va conditionner la résolution par éléments finis, c'est-à-dire les physiques simulées, le degré de précision, la vitesse de la résolution numérique et la souplesse de représentation d'une solution en fonction de la distribution de matière décrite.

1.4 Classification des outils d'optimisation topologique

Il existe de nombreuses classifications possibles des outils d'optimisation topologique. Dans ce manuscrit, deux classifications sont utilisées pour caractériser les algorithmes d'optimisation utilisés. Ces deux classifications sont : les algorithmes déterministes vs les algorithmes non-déterministes, ainsi que les algorithmes exacts vs les algorithmes heuristiques.

La différence entre les algorithmes déterministes et les algorithmes non-déterministes repose sur la méthode de variation des paramètres d'optimisation. Les algorithmes déterministes utilisent l'état de la solution, ou de l'algorithme, à l'itération précédente pour définir la valeur des paramètres de la solution à la prochaine itération. Pour une même solution initiale, l'algorithme parcourt donc le même chemin dans l'espace des solutions. A l'inverse, les algorithmes non-déterministes utilisent un facteur aléatoire pour influencer la modification de la valeur des variables de la solution. Cette méthode permet de réduire la dépendance d'une solution aux états précédents. Pour une même solution initiale, l'algorithme peut parcourir des chemins différents dans l'espace des solutions.

Les algorithmes exacts sont des algorithmes développés pour permettre de trouver l'optimum global d'une solution, c'est-à-dire la meilleure solution du problème étudié, lorsque cela est possible. Ces algorithmes sont le plus souvent spécialisés pour des cas particuliers tel que l'algorithme de Dijkstra [2] pour le problème de la recherche du chemin le plus court à travers un ensemble de chemins reliant des nœuds. Dans certaines situations, la recherche de l'optimum global peut s'avérer être très longue. Dans ce cas, il est possible d'utiliser des algorithmes heuristiques à la place des algorithmes exacts. Pour un temps de calcul plus petit, les algorithmes heuristiques sont capable de donner de bonnes solutions mais ce gain de temps de calcul est obtenu au détriment de la

qualité de la solution finale, dont l'optimalité n'est plus garantie. Pour le cas de l'optimisation du problème de chemin le plus court, la version heuristique de l'algorithme de Dijkstra est l'algorithme A* [3]. Ce dernier permet de rechercher un chemin en un temps relativement court mais ne garantit pas que la première solution trouvée soit l'optimum.

1.5 Cadre et objectifs de la thèse

Actuellement, différents outils d'optimisation topologique sont utilisés par l'industrie avec de bons résultats dans les domaines de la conception optimale en mécanique, que ce soit dans la conception de pièces mécaniques ou sur l'architecture complète d'un pont, en prenant en compte l'élasticité linéaire du matériau qui est distribuée pour réaliser la structure à optimiser. On retrouve aussi des outils spécifiques développés dans les domaines de l'électromagnétisme et de la mécanique des fluides. Cependant, ces outils d'optimisation topologique utilisent généralement des algorithmes déterministes afin de garantir une bonne vitesse de convergence, au risque de ne pas trouver l'optimum du problème posé. De plus, ces algorithmes d'optimisation sont souvent limités à l'utilisation de variables d'optimisation continues ne permettant donc pas d'avoir un outil d'optimisation topologique générique.

Dans le cadre de l'utilisation des outils d'optimisation topologique en milieu industriel, il est souhaité qu'ils soient indépendants du problème traité, c'est-à-dire qu'ils doivent :

- pouvoir traiter des problèmes multiphysiques, multiobjectifs, multicontraintes et multivariables ;
- être indépendants du type des variables utilisées pour décrire une solution. Par exemple, on peut citer l'utilisation des variables discrètes et/ou continues.

De plus, afin de garantir un design de conception abouti, les outils d'optimisation topologique doivent non seulement être capables d'optimiser la topologie mais aussi la géométrie et les dimensions associées. Ceci implique que ces outils doivent posséder une étape d'exploration (topologie), une étape d'exploitation (géométrie et dimensions) et un mécanisme de transition entre ces deux étapes.

Suivant ces critères, le choix de l'algorithme d'optimisation s'est orienté sur l'utilisation des algorithmes métaheuristiques, c'est-à-dire des heuristiques génériques, couplés à des formalismes de distribution de

matière adéquats. Le choix de ce type d'algorithme garantit une grande ouverture sur les types de problèmes à optimiser mais souvent au détriment de la vitesse de convergence.

Afin d'augmenter la performance de l'algorithme d'optimisation, il est nécessaire d'accélérer le processus d'optimisation, tout en conservant un niveau élevé d'exploration de l'espace des solutions. Il convient donc de développer des méthodologies et des outils basés sur les algorithmes métaheuristiques compatibles avec une optimisation topologique et d'analyser les performances de ces méthodes quant à leur capacité de déterminer des géométries et des dimensions optimales avec un minimum d'a priori. Pour cela, il faut :

- définir les critères de performance ;
- définir des cas tests aussi proches que possible des cas applicatifs réels ;
- choisir et mettre en œuvre des algorithmes d'optimisation adéquats ;
- choisir et mettre en œuvre des formalismes de distribution de matière adéquats ;
- proposer et tester de nouveaux mécanismes d'optimisation permettant d'augmenter les performances des algorithmes d'optimisation topologiques ;
- appliquer les outils développés à quelques cas réalistes.

Finalement, l'évaluation des capacités des outils d'optimisation topologique se fait au travers de deux objectifs :

- avoir un processus d'optimisation qui converge vers l'optimum global le plus rapidement possible. Cette amélioration de la vitesse doit pouvoir inclure un équilibre efficace entre une recherche par exploration pour dégager la topologie optimale et une recherche par exploitation pour dégager la géométrie et les dimensions optimales ;
- obtenir des solutions directement utilisables. Ceci inclut le respect de contraintes mécaniques, mais surtout une description aboutie des frontières des sous-domaines de matière composant la solution.

1.6 Organisation du manuscrit

Le Chapitre 2 est un état de l'art des outils d'optimisation topologique basés sur des algorithmes métaheuristiques. Cette mise en contexte

discute des différents algorithmes utilisés, ainsi que des formalismes de distribution de matières utilisés à ce jour dans le cadre de l'optimisation topologique, ainsi que les différentes adaptations apportées aux outils développés.

Le Chapitre 3 discute les choix stratégiques à effectuer pour mener à bien notre recherche. Ces choix sont influencés par différents aspects pratiques. Le chapitre donne aussi une description des différents cas d'étude utilisés ainsi que les outils d'évaluation nécessaires.

Le Chapitre 4 est l'étape de sélection d'un outil prometteur qui pourra être adapté à l'optimisation topologique. Il est subdivisé en trois sections. Premièrement, il établit une présélection des outils d'optimisation topologique afin de limiter le nombre d'outils à étudier. Deuxièmement, il décrit l'implémentation des différents éléments constitutifs des outils d'optimisation topologique choisis, c'est-à-dire les différents algorithmes et formalismes. Troisièmement, il étudie et compare les outils d'optimisation topologique choisis sur la base d'un cas simple graphique, afin d'estimer leur efficacité. Le choix se porte finalement sur l'utilisation d'un algorithme génétique couplé à un formalisme de distribution de matière basé sur le diagramme de Voronoï.

Le Chapitre 5 est la première partie de l'étape d'adaptation des outils d'optimisation topologique. Il se focalise sur l'approche de gestion de la population de l'algorithme génétique. Le chapitre étudie différentes méthodes pour faire varier la taille de la population en fonction des besoins de l'algorithme génétique et étudie aussi l'influence de la population initiale sur le bon déroulement du processus d'optimisation.

Le Chapitre 6 est la seconde partie de l'étape d'adaptation des outils d'optimisation topologique. Il se focalise sur les méthodes de reproduction de l'algorithme génétique. Le chapitre décrit et compare différentes méthodes de croisement et de mutation.

Le Chapitre 7 est l'étape des tests de l'outil développé. Plusieurs cas d'applications complexes se rapprochant de la réalité industrielle sont utilisés afin de comparer le nouvel outil avec des outils d'optimisation topologique issus de la littérature.

Le Chapitre 8 propose une série de perspectives faisant suite à la recherche effectuée.

Le Chapitre 9 est le chapitre de conclusion sur les résultats obtenus dans ce manuscrit.

Le document se termine par une annexe qui décrit les choix pré-

liminaires effectués pour déterminer l'implémentation des algorithmes utilisés.

Chapitre 2

Etat de l'art

Conformément aux choix préliminaires opérés dans l'introduction, l'étude poursuivie tout au long de cette thèse se concentre sur le domaine des outils d'optimisation topologique par distribution de matière utilisant des algorithmes métaheuristiques. Depuis maintenant une trentaine d'années, de nombreux travaux ont été effectués sur ce sujet. Certains de ceux-ci ne visent que l'utilisation d'un algorithme métaheuristique particulier, l'exploitation de nouveaux formalismes de distribution de matière ou encore l'amélioration du couplage entre l'algorithme d'optimisation et le formalisme de distribution de matière.

Les trois sous-sections de ce chapitre présentent ainsi l'évolution des outils d'optimisation topologique dans le domaine ciblé. Il est difficile de retracer séquentiellement l'historique de l'évolution de ces outils, de nombreux travaux ayant été menés en parallèle. C'est pourquoi les deux premières sections se limitent à passer en revue des algorithmes d'optimisation et des formalismes de distribution de matière utilisés dans le cadre de l'optimisation topologique. La troisième section présente, par thème, différentes adaptations déjà effectuées avec pour objectif d'améliorer les performances des outils d'optimisation topologique.

2.1 Algorithmes d'optimisation

Les premiers travaux sur l'optimisation topologique remontent au début du XXème siècle, par une approche qui diffère de l'optimisation par distribution de la matière dans un domaine de design tel qu'on le

considère dans ce manuscrit, c'est-à-dire par une approche de l'optimisation de structure en treillis. Hajela [4], Grierson et Pak [5] et Wu [6] ne sont que quelques exemples de chercheurs parmi tant d'autres ayant travaillé sur l'optimisation topologique de treillis de barres.

Dès la fin des années 80, les premières approches par distribution de matière pour l'optimisation topologique ont été appliquées à des problèmes d'optimisation dans le domaine de l'élasticité linéaire en mécanique. Les pionniers furent G. Strang et R. Kohn en 1986 [7] suivis de M. Bendsøe et N. Kikuchi en 1988 [8] en développant une méthode d'optimisation par densité de matière. Suite à la nécessité de travailler avec des paramètres continus, cette méthode nécessite la construction d'un opérateur d'homogénéisation comme décrit par G. Allaire et R.V. Kohn [9]. Un tel opérateur permet d'établir une continuité entre chaque matériau disponible en définissant des matériaux intermédiaires. Cependant, cette méthode possède trois défauts. Premièrement, il est nécessaire de créer un nouvel opérateur d'homogénéisation lorsque la physique du problème étudié est modifiée. Deuxièmement, Bendsøe [10] montre non seulement que l'optimum obtenu dépend fortement du modèle de microstructure utilisé pour l'homogénéisation, mais aussi que l'algorithme est fortement dépendant de la distribution de matière initiale. Finalement la solution finale doit être dégradée pour supprimer les matériaux intermédiaires. Pour répondre à ce dernier problème, une autre version d'interpolation plus populaire des caractéristiques des matériaux est développée : la méthode d'optimisation avec pénalisation utilisant des matériaux solides isotropiques, mieux connu sous le nom SIMP (Solid Isotropic Material with Penalization) [8, 11, 12, 13]. La fonction objectif est pénalisée en fonction de l'existence de matériaux intermédiaires en vue de guider la solution vers une disparition de ce type de matériaux. Toujours dans le domaine de la mécanique, et pour éliminer le besoin d'utiliser des paramètres continus, une autre approche fut proposée en 1993 par Xie et Steven [14] : la méthode ESO (Evolutionary Structural Optimization). Cette approche utilise à chaque itération de calcul les contraintes mécaniques de la solution précédente pour chacun des éléments de la structure à optimiser. Ceux ne reprenant que très peu d'effort peuvent être considérés comme inutiles.

Afin de désensibiliser l'outil par rapport au choix de la solution initiale, une autre branche des algorithmes d'optimisation a été exploitée. Sachant que toutes les méthodes précédentes peuvent être considérées

comme déterministes, une autre direction de recherche se focalise sur l'utilisation d'algorithmes non-déterministes tels que des algorithmes métaheuristiques dont les avantages avec des problèmes combinatoires ne sont pas négligeables.

Il existe de nombreux algorithmes d'optimisation métaheuristiques dont certains furent déjà utilisés pour l'optimisation topologique. On retrouve dès 1992, une version d'algorithme du recuit simulé proposée par G. Anagnostou [15]. En novembre 1992 à l'université de Purdue aux États-Unis, E. Jensen a proposé dans sa thèse une version différente de l'outil d'optimisation topologique avec cette fois-ci un algorithme génétique [16].

En 1996, M. Schoenauer compare un algorithme génétique (GA) avec un algorithme à évolution stratégique (ES) sur un problème de poutre encastrée [17]. Les courbes moyennes de convergence sont reprises par le graphique de la Figure 2.1. La fonction objectif consiste à minimiser le volume de matériau utilisé ainsi que la déformation maximale de la structure. Les résultats obtenus montrent une plus grande efficacité de l'algorithme génétique comparé à l'algorithme à évolution stratégique pour plusieurs types de formalismes de distribution de matière.

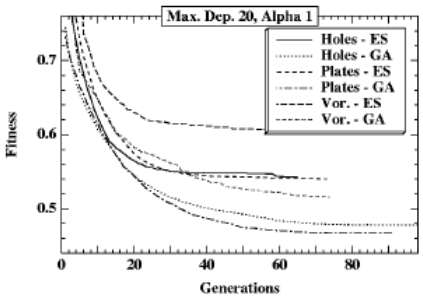

Figure 2.1 – Comparaison des résultats entre l'algorithme génétique et l'algorithme à évolution stratégique combinés à trois formalismes différents de distribution de matière (Source [17])

Globalement, l'algorithme génétique a le plus souvent été pris comme référence pour tester de nouvelles approches. En 2002, Wilfried Hansel a présenté une comparaison entre une méthode heuristique et un algorithme génétique pour l'optimisation d'une poutre encastrée avec des matériaux laminés [18]. Le fonctionnement de la méthode heuristique

repose sur deux actions. La première action supprime les éléments de matière ne subissant que très peu de contraintes mécaniques alors que la seconde action ajoute de la matière là où les contraintes mécaniques sont fortes. L'objectif du problème est de minimiser le poids de la structure pénalisé par le risque de non respect de résistance de la structure. Les résultats montrent que l'algorithme génétique donne de meilleurs résultats, de l'ordre de 15%, sur la masse totale à déformation équivalente, grâce à sa capacité d'exploration de l'espace des solutions. Toutefois, là où l'algorithme génétique nécessite environ 3500 évaluations, la méthode heuristique n'en utilise que 50 (Figure 2.2).

L'algorithme immunitaire fit son apparition en optimisation topologique en 2004. G.C. Luh proposa une version d'un outil d'optimisation topologique avec cet algorithme pour résoudre un problème multimodal d'optimisation de structure. Campelo compara en 2006 ce même algorithme [19] avec un réseau d'Hopfield [20] sur la conception d'un aimant MRI (Magnetic resonance imaging). L'article montre que l'algorithme immunitaire est plus efficace.

Au cours de l'année 2006, deux articles [21, 22] de S. Bureerat présentent une étude plus importante de comparaison entre plusieurs algorithmes d'optimisation sur un problème de type ON/OFF[1]. Ils comparèrent l'algorithme génétique avec une variante, le Stud-GA [23], l'algorithme du recuit simulé et la technique PBIL (Population-Based Incremental Learning). Plusieurs cas d'étude sont utilisés mais restent dans le domaine de la résistance des matériaux tel que la poutre encastrée. Ils consistent tous à minimiser la compliance, le volume de matière utilisé ainsi que l'aspect en damier des résultats. Le Stud-GA consiste à prendre à chaque itération la meilleure solution, appelée le *Stud*, et de la combiner avec toutes les autres solutions du groupe de solution pour former un nouveau groupe. Par contre, le PBIL utilise une analyse statistique de la population pour déterminer la valeur de chaque paramètre. A la fin de celle-ci, chaque paramètre de design a une probabilité d'être fixé à ON ou à OFF. Finalement, les résultats montrent que l'algorithme du recuit simulé est meilleur dans tous les cas, suivi de l'algorithme Stud-GA. S. Bureerat compara les résultats obtenus pour les quatre algorithmes avec ceux obtenus avec la méthode SIMP. Les résultats sont de même niveau

[1] L'appellation ON/OFF provient des problèmes d'élasticité linéaire en mécanique où l'absence de matière est remplacée par de l'air. Le problème se réduit à activer ou à désactiver le matériau en chacun des points de l'espace.

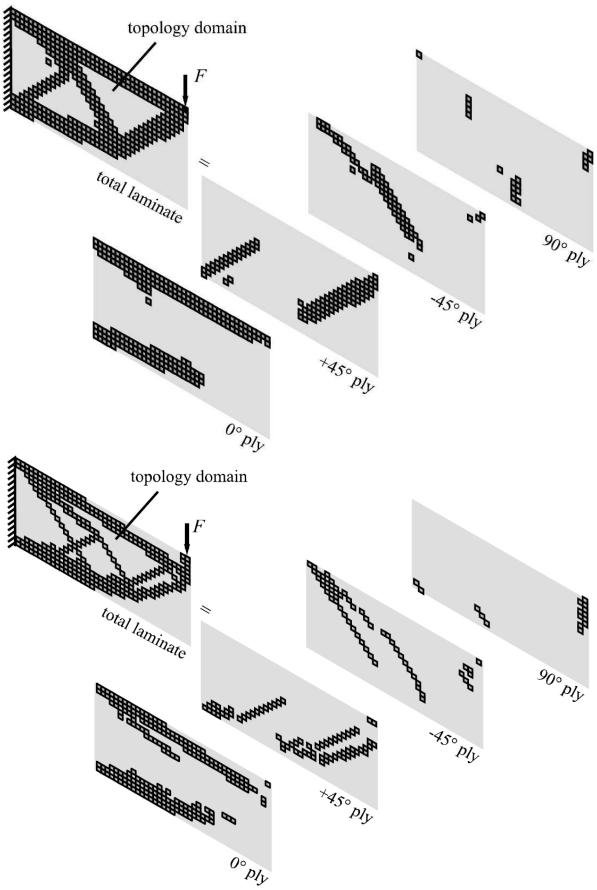

Figure 2.2 – Comparaison des distributions de matière obtenues avec un algorithme heuristique (haut) et avec un algorithme génétique (bas) pour un problème d'optimisation d'une poutre encastrée avec des matériaux laminés (Source [18])

mais cette dernière méthode génère des matériaux intermédiaires.

C.Y. Wu proposa en 2010 une méthode avec un algorithme à évolution différentielle [24] en se basant sur l'article [25] qui propose une version de l'algorithme à évolution différentielle travaillant avec des variables discrètes, . Les résultats montrent que l'algorithme donne de meilleurs résultats que la méthode SIMP et que la méthode par homogénéisation, mais avec un coût plus élevé lié à la nature de recherche globale de l'algorithme.

D'autres algorithmes évolutionnaires hybridés avec une recherche locale, nommés algorithmes mémétiques, continuent à compléter le tableau d'études. Ainsi, en 2008, A. Kaveh publia un article sur l'utilisation d'un algorithme par colonies de fourmis (ACO) sur le cas d'une optimisation topologique d'une structure mécanique [26]. Il montra que l'ACO est assez proche des résultats obtenus avec la méthode SIMP, comme le montrent par exemple les résultats de la Figure 2.3.

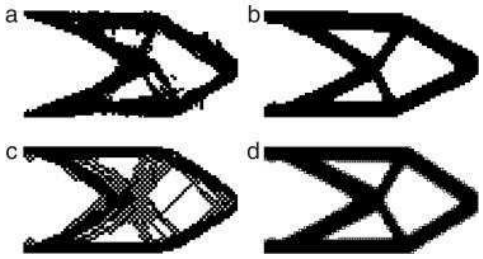

Figure 2.3 – Comparaison des distributions de matière pour un problème d'optimisation de minimisation du poids d'une poutre encastrée (Source [26]) : de haut en bas, les résultats obtenus avec un algorithme heuristique (haut) et avec la méthode SIMP (bas) ; de gauche à droite, les résultats obtenus sans techniques de nettoyage et avec technique de nettoyage pour limiter la présence de cellules de matière solitaires

Toutes ces études démontrent que chaque algorithme métaheuristique a ses avantages et ses défauts. Ils sont généralement développés pour faire de l'exploration mais aussi pour faire de l'exploitation. Il est possible de créer des algorithmes hybrides par combinaison avec d'autres algorithmes pour profiter des qualités de chacun et compenser leurs défauts. En 2001, K. Deb publia un article [27], revisité en 2008 [28], utilisant une version alliant un algorithme évolutionnaire avec un algorithme de recherche locale, l'algorithme du hill climbing, afin de profiter de la capacité d'exploration de l'algorithme génétique et de la capacité d'exploitation de l'algorithme du hill climbing. Cependant, cette version ne couple pas réellement les deux algorithmes, puisque ceux-ci restent entièrement indépendants, l'algorithme du hill climbing n'étant utilisé qu'après l'algorithme génétique.

En 2003, C.H. Im propose une hybridation par couplage de l'algorithme génétique et d'une méthode de test de sensibilité [29]. Au début de chaque génération, l'outil a la possibilité de faire appel à la méthode

de test de sensibilité avec une probabilité déterminée par l'utilisateur. Ce test de sensibilité consiste à changer la valeur de chaque paramètre et à mesurer le gain obtenu. La probabilité de mutation est calculée en fonction du gain obtenu. Cette méthode permet d'intensifier la recherche de meilleures solutions, aux alentours de celles obtenues, en effectuant de petites modifications. Après un critère d'arrêt lié au niveau du gain, l'outil retourne dans l'algorithme génétique pour continuer le processus. Les tests effectués sur trois problèmes électromagnétiques montrent qu'effectivement, il est plus qu'intéressant de travailler avec une approche hybride combinant exploration et exploitation.

Sur la période de 2007 à 2008, Wang et Tai publièrent trois articles [30, 31, 32] proposant un couplage entre l'algorithme génétique et la méthode de Hooke and Jeeves [33]. Après la création de chaque nouvelle population d'enfants, les N meilleures solutions de cette population sont soumises à une optimisation locale.

2.2 Formalismes de distribution de matière

La complexité d'un problème d'optimisation est non seulement liée au problème lui-même mais aussi à la définition des variables d'optimisation. Un choix judicieux des variables évite une complexification inutile du problème et en particulier de l'exploration de l'espace des solutions. La Figure 2.4 présente deux cas de choix de paramètres dimensionnels pour optimiser les rayons intérieur et extérieur d'un anneau.

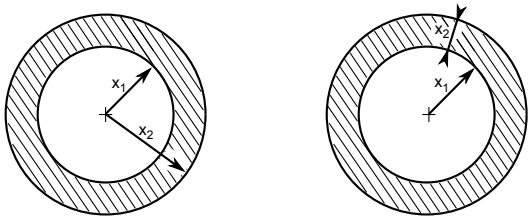

Figure 2.4 – Deux choix possibles de paramètres pour décrire un anneau

Le schéma de gauche présente le cas où les paramètres sont les rayons à optimiser alors que le schéma de droite présente un choix plus judicieux de paramètre où le rayon extérieur est exprimé en fonction du rayon intérieur et de l'épaisseur de l'anneau. Ce choix permet d'élimi-

ner la contrainte $x_1 < x_2$. De la même manière, pour l'optimisation topologique, un choix judicieux de variables de design peut aider l'algorithme dans son processus d'optimisation. C'est pourquoi plusieurs formalismes de distribution de matière furent proposés pour permettre à l'outil d'optimisation topologique d'exprimer une distribution de matière au travers d'un nombre optimal de variables de design puisque le choix d'un formalisme de distribution de matière influence à la fois le nombre de variables nécessaires pour décrire une solution donnée mais aussi la facilité de passer de l'exploration à l'exploitation.

Le premier formalisme de distribution de matière utilisé est le formalisme Bit-String. Introduit par Jensen en 1992, il associe un vecteur de variables de design à la distribution de matière décrite au travers d'un quadrillage régulier de l'espace de design. Le nom de Bit-String provient de la concaténation de variables de design en un vecteur représentant le codage de la solution. Jensen proposa de représenter ce vecteur par un Bit-Array [16] pour rappeler les liens entre le génotype et le phénotype[2] de chaque proposition de solution. Un tenseur d'ordre de même valeur que la dimension de l'espace de design est utilisé pour stocker les variables d'optimisation (Figure 2.5).

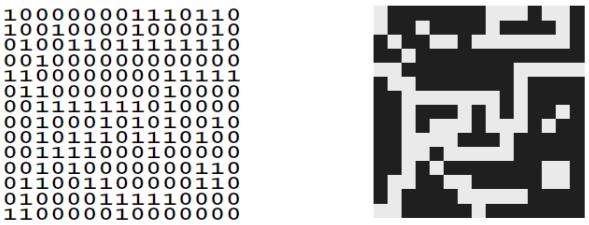

Figure 2.5 – Formalisme de distribution de matière pour une représentation 2D avec deux matériaux : Bit-Array (Source [34])

Afin de répondre à la limitation de la précision de description d'une solution liée par le niveau de discrétisation du Bit-Array ou Bit-String, Dejonghe proposa en 1993 la méthode par patchwork [35] pour décrire

[2]En biologie, le génotype désigne une partie de la composition génétique d'un individu. Quant au phénotype, il est la conséquence observable du génotype d'un individu. Le lien effectué ici avec l'optimisation topologique est obtenu en distinguant le génotype comme étant le vecteur de paramètres d'optimisation et le phénotype comme étant la distribution de matière résultante.

une solution au moyen de formes prédéfinies de dimension aléatoire (Figure 2.6). La solution est ensuite projetée sur un maillage indépendant de la solution pour permettre les calculs éléments finis.

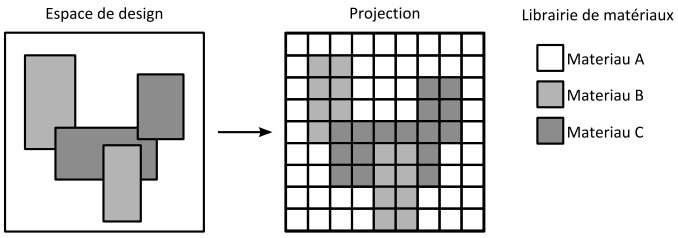

Figure 2.6 – Formalisme de distribution de matière : méthode du patchwork rectangulaire à trois matériaux

M. Schoenauer étudia aussi cette approche en 1996 [17, 36]. Le formalisme utilisé consiste à disposer des rectangles paramétrables en nombre variable pour décrire la distribution de matière au sein de l'espace de design. Pour un problème dit ON/OFF, le formalisme possède deux variantes. La première variante considère l'espace de design comme vide (OFF) et les rectangles comme remplis de matière (ON), la seconde variante reprend la situation inverse. Cette asymétrie peut être dérangeante pour l'optimisation comme le démontre Schoenauer [36]. Si le problème mécanique nécessite plus de matière que d'absence de matière, il est plus facile de partir d'un espace de design plein de matière (ON) et d'y opérer des trous [37] et inversement pour le cas où la présence de matière est faible dans l'optimum global.

Au lieu d'utiliser un découpage régulier de l'espace de design, Schoenauer, en 1996 [17], proposa l'utilisation du diagramme de Voronoï pour subdiviser l'espace de design en sous-espaces de matière homogène. Cette méthode (qui sera décrite en détail à la Section 4.2.2.2) consiste à créer la subdivision en distribuant un certain nombre de points dans l'espace de design (les centres de Voronoï), et en associant à chaque centre un sous-espace (une cellule) dont la forme et la taille est fonction de la distance qui sépare son centre des centres voisins. A chacun des sous-espaces ainsi créés, il associe un matériau (Figure 2.7). Cette méthode permet de diminuer le nombre de variables nécessaires pour décrire des formes complexes, puisqu'il est possible de décrire des segments par l'utilisation de deux cellules de Voronoï. Toutefois, dans certaines situations,

ce formalisme doit utiliser un très grand nombre de variables de design comme pour des représentations en damier.



Figure 2.7 – Formalisme de distribution de matière : diagramme de Voronoï avec en gris clair le matériau A et en gris foncé le matériau B (Source [17])

Les différents formalismes de distribution de matière peuvent être comparés sur base d'un cas d'étude de problème graphique. L'objectif de l'optimisation est de retrouver une forme prédéfinie. Les résultats montrent que le formalisme patchwork est plus efficace que le formalisme Voronoï lorsque la topologie à obtenir peut facilement être décrite avec des formes prédéfinies [36]. Mais l'avantage revient au formalisme Voronoï pour sa souplesse de description de formes plus complexes.

Cependant, lors de l'évaluation, M. Schoenauer utilise une projection des cellules de Voronoï sur un espace discrétisé de même dimension que pour les formalismes Bit-Array et Bit-String pour établir une comparaison [17, 36, 34], dégradant la description de la distribution de matière de la solution obtenue.

De plus, un défaut de ce formalisme est le haut niveau d'épistasie[3]. La position d'un centre de Voronoï influence la distribution de matière pour les cellules voisines, la frontière entre deux cellules étant définie par la position des deux centres.

Toujours avec pour objectif de travailler avec de nouveaux formalismes plus aptes à décrire des formes spécifiques, K. Tai [38, 39, 40, 41, 42], dès 1998, travailla sur un formalisme dont les variables de design décrivent la forme de courbes de Bézier et définissent leur épaisseur dans un espace de design afin d'y décrire une distribution de matière. La principale motivation d'utiliser cette approche est de garantir la connectivité de la matière dans l'espace de design, ce qui a souvent été mis en défaut

[3]Le degré d'épistasie en génétique indique le degré d'interaction entre deux ou plusieurs gènes pour la description du phénotype

pour les formalismes tels que le Bit-String (Figure 2.8).

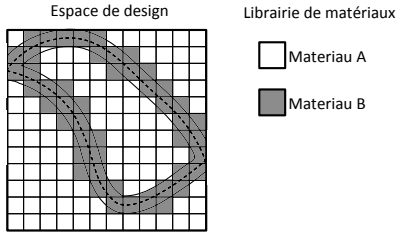

Figure 2.8 – Formalisme de distribution de matière : courbes de Bézier

Les courbes de Bézier ont besoin de la définition d'un point de départ et d'un point d'arrivée ainsi que de points de contrôle. Pour obtenir une distribution de matière à partir du squelette ainsi formé, un paramètre d'épaisseur de peau est introduit entre chaque point définissant la courbe (Figure 2.9). Les opérateurs de modification de la solution ne permettent pas de modifier la taille du vecteur de variables afin de permettre des combinaisons cohérentes entre les différentes solutions.

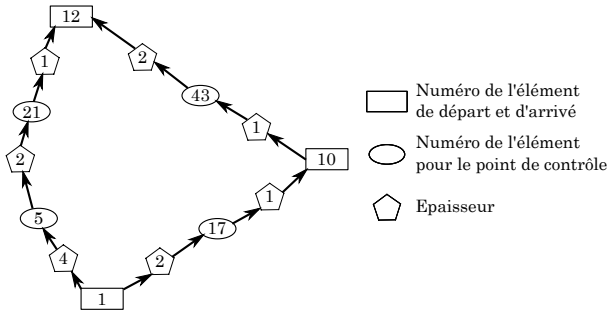

Figure 2.9 – Représentation de la structure des variables pour décrire le formalisme à base de courbes de Bézier

K. Tai proposa une comparaison de ce nouveau formalisme avec le formalisme Bit-String [39]. L'outil est complété avec un algorithme d'optimisation et l'espace de design est discrétisé en un quadrillage de 50×50 cellules. Le cas d'étude consiste à retrouver une géométrie définie. Dans les deux cas, la population est composée de 400 individus évoluant sur 100 générations. Le nombre de courbes de Bézier, N_{cB}, est de 4 et le nombre de points de contrôle, N_{pc}, est de 2 pour chaque courbe, soit

20 variables de design, ce qui est peu comparé aux 2500 variables discrètes nécessaires pour le Bit-String. Le niveau de complexité est choisi par l'utilisateur en fixant N_{pc} et N_{cB}. Les résultats montrent que les solutions sont proches, mais celles du formalisme Bit-String possèdent des cellules non-connectées. Les courbes de convergence montrent que le formalisme Bit-String est moins rapide à converger que le formalisme à base de courbe de Bézier. Ces résultats sont nuancés par l'obligation de définir à l'avance le nombre de courbes de Bézier nécessaires pour décrire une solution, ce qui est difficile. Avec une géométrie plus complexe et les mêmes configurations des outils, le formalisme Bit-Array est plus apte à trouver la meilleure solution.

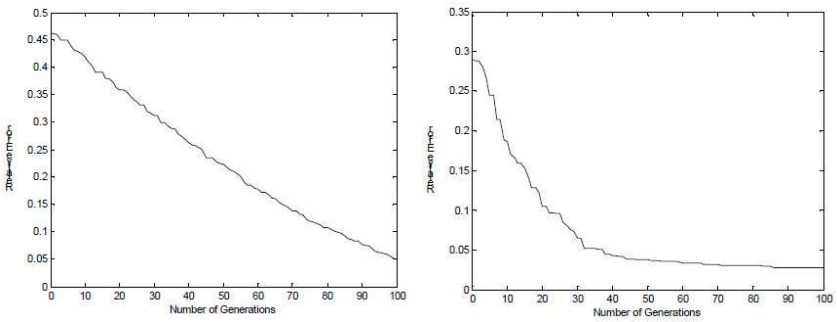

Figure 2.10 – Comparaison des courbes de convergence pour des tests avec le formalisme Bit-Array (à gauche) et le formalisme avec des courbes de Bézier (à droite), couplés avec un algorithme génétique (Source [39])

Suite au problème du nombre de courbes de Bézier à définir à l'avance, K. Tai et N. Wang proposèrent en 2007 [42] une version adaptée du formalisme permettant à l'algorithme d'optimisation de modifier lui-même le nombre de variables de design avec un nombre maximum fixé. Cette évolution permet à l'algorithme d'élargir l'espace des solutions accessibles.

En 2009, J. K. Guest et L. C. Smith Genut présentèrent un nouveau formalisme [43] afin de garantir une indépendance du formalisme de distribution de matière vis-à-vis du maillage à éléments finis. Ce formalisme fonctionne sur la projection de Heaviside [44] pour décrire la distribution de matière au sein de l'espace de design (Figure 2.11).

Cette méthode se base sur l'utilisation de points de référence $\vec{x}_i$ dans l'espace de design, aussi appelés nœuds, et de fonctions $\phi_i(r)$ associées à

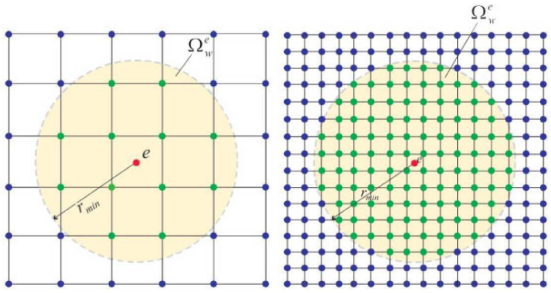

Figure 2.11 – Formalisme de distribution de matière : méthode de la projection d'Heaviside (Source [43])

$\vec{x}_i$ avec le paramètre r représentant la distance entre un point de matière e de l'espace de design et le nœud $\vec{x}_i$. Un rayon r_{min} est fixé pour définir le domaine d'influence Ω_W^e lié au point de matière. Pour tous r supérieur à ce rayon r_{min}, la fonction $\phi(r)$ retourne la valeur 0. Ce rayon est fixé dès le début du processus d'optimisation et la position des nœuds peut être définie arbitrairement.

Pour attribuer la présence d'un matériau au point de matière e de l'espace de design, c'est-à-dire ρ_e, la somme des fonctions ϕ_i superposées associées aux nœuds $\vec{x}_i$ proche du point de matière doit être supérieure à une borne K. Nous avons donc :

$$\rho_e = \begin{cases} 1 \text{ si } \sum_{i \in N^e} \phi_i > K \\ 0 \text{ si } \sum_{i \in N^e} \phi_i \leq K \end{cases} \tag{2.1}$$

où N^e représente l'ensemble des indices i des nœuds à l'intérieur du cercle défini par la position de e et $r_{\min}$:

$$\vec{x}_i \in \Omega_W^e \text{ si } \|\vec{x}_i - \vec{x}_e\| < r_{\min} \tag{2.2}$$

2.3 Adaptations spécifiques

Lors d'un processus d'optimisation, afin de maximiser les chances de trouver l'optimum global, il est nécessaire d'avoir une phase d'exploration suffisante pour identifier la topologie optimale, suivie par une phase d'exploitation afin de mieux définir les formes et les dimensions

de la solution. Or, généralement, les algorithmes métaheuristiques présentent une faiblesse non négligeable, soit dans l'étape d'exploitation ou soit dans l'étape d'exploration. Il est possible d'aider l'outil dans sa phase la plus faible au prix d'un grand nombre d'évaluations, ce qui peut impliquer un temps élevé du processus complet d'optimisation.

En fin de processus d'optimisation, deux cas différents non souhaités peuvent se présenter (Figure 2.12), qu'il convient de résoudre en favorisant certains mécanismes de l'optimisation. Ces trois cas sont :

- La solution finale « convergée » n'est pas l'optimum : la topologie trouvée n'est pas l'optimum et la forme et ses dimensions ont été optimisées pour cette topologie ;
- La solution finale est une solution intermédiaire : la topologie n'est pas la topologie optimale ou la forme et ses dimensions ne sont pas optimisées.

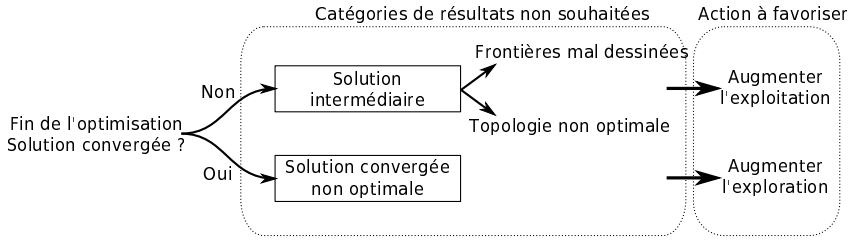

Figure 2.12 – Distinction des résultats non souhaités en fin de processus d'optimisation et du type d'action à favoriser pour éviter le cas

A nombre d'évaluations fixé, deux approches sont possibles pour trouver le bon niveau d'exploration et d'exploitation. La première consiste à trouver les paramètres de configuration adéquats de l'algorithme d'optimisation lorsque cela est possible. Ceci requiert une certaine compréhension de l'outil d'optimisation. La seconde approche consiste à adapter l'outil d'optimisation topologique de manière à l'aider dans son processus d'optimisation.

2.3.1 Évolution du nombre de variables de design

La finesse de la discrétisation est un élément non négligeable pour l'évolution du processus d'optimisation. Un plus grand nombre de variables contraint l'algorithme à travailler sur un plus grand espace de

recherche. Afin de faciliter l'exploration, tout en obtenant un nombre suffisant de variables de design pour représenter la meilleure solution, C.D. Chapman [45] utilisa en 1994 un raffinement programmé de la discrétisation de l'espace avec le formalisme Bit-Array au cours de l'optimisation. Ce raffinement est forcé à partir d'un certain nombre d'évaluations, par exemple en passant de 10×16 à 20×32 cellules au bout de 250 générations et de 20×32 à 40×34 cellules après quelques 50 nouvelles générations, l'outil terminant sur ce raffinement pour encore 10 nouvelles générations. A chaque nouvelle discrétisation, la taille du vecteur de variables d'optimisation augmente. Pour éviter tout problème lié à cette augmentation, il proposa en même temps de subdiviser l'espace de design en quatre, puis en seize. A ces nouveaux espaces de design, l'algorithme crée de nouvelles populations liées à chacun d'entre-eux.

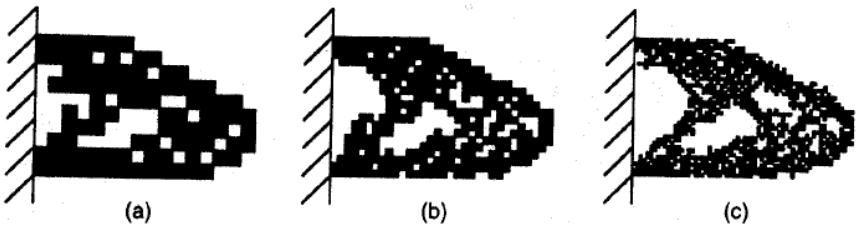

Figure 2.13 – Exemple d'une solution décrite au travers d'une discrétisation de 10×16, 20×32 et 40×64 cellules [45]

En 2004, I.Y. Kim et O.L. de Weck présentèrent une autre méthode de raffinement de la discrétisation de l'espace de design [46, 47]. En travaillant sur différentes tailles du chromosome avec l'algorithme génétique, ils proposent de transmettre, étape par étape, les meilleures solutions sans altération du chromosome par simple projection d'un niveau de maillage sur un autre, mais aussi de transférer les autres solutions en altérant une partie de leur patrimoine génétique. Lors de l'altération, le matériau de chacune des cellules a une chance d'être modifié. Cette chance est exprimée par la relation suivante :

$$\rho_k^{new} = round\left(\alpha + \beta\rho_k^{previous} + \gamma\Gamma\right) \tag{2.3}$$

où α, β et γ sont des paramètres fixés par l'utilisateur, Γ est une variable aléatoire de distribution normale, ρ_k représente la matière au sein du k^{ieme} élément et la fonction *round* arrondit à l'entier son argument.

Les résultats de cette nouvelle méthode sont comparés à ceux d'un algorithme d'optimisation par recherche exhaustive. La méthode de raffinement donne de meilleurs résultats (Figure 2.14).

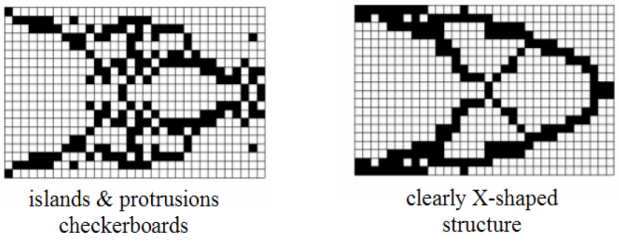

Figure 2.14 – Résultats obtenus avec une discrétisation fixe (à gauche) et une discrétisation variable (à droite) de l'espace de design [46]

2.3.2 Diminution de l'espace des solutions par utilisation des contraintes

Dans de nombreux problèmes ON/OFF à optimiser, le volume des structures obtenues ne doit pas dépasser une valeur limite. De nombreuses solutions sont ainsi automatiquement éliminées au cours du processus d'optimisation afin de respecter cette contrainte. Pour faciliter la tâche de l'outil d'optimisation avec un quadrillage régulier, tel que le Bit-String ou le Bit-Array, J.F. Madeira proposa en 2002 de contrôler le nombre de paramètres mis à 1 pour chaque solution pour obliger l'outil à n'utiliser que des solutions admissibles [48]. Cette manière d'exploiter la contrainte permet de diminuer la complexité du problème. Un vecteur de paramètres de longueur L possède a priori 2^L possibilités tandis qu'imposant un nombre maximal K de 1 pour avoir un volume désiré, le nombre de valeurs différentes possibles d'un vecteur de paramètres passe à :

$$C_K^L = \frac{L!}{K!\,(L-K)!} \tag{2.4}$$

Les opérateurs de transformation des solutions doivent être adaptés pour que les solutions enfants respectent la contrainte.

2.3.3 Suppression des éléments non connectés

La présence d'éléments non connectés est un des phénomènes que l'on retrouve dans la plupart des formalismes. On dit qu'un élément est non connecté lorsqu'il ne partage pas une arête commune avec un autre élément du même matériau. On peut toutefois utiliser une règle plus souple admettant que les éléments n'ayant qu'un nœud en commun sont aussi connectés (Figure 2.15).

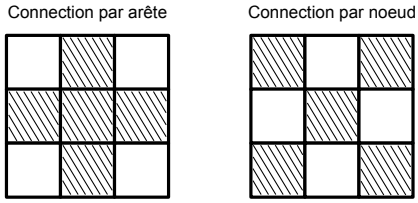

Figure 2.15 – Différents types de connectivité des éléments discrets

Les éléments non connectés sont le plus souvent considérés comme indésirables dans les solutions optimales si l'on suppose que les domaines de matière doivent être le plus homogènes possibles. Il est difficile pour les métaheuristiques de les éliminer sans adapter leurs méthodes de modifications des variables d'optimisation.

Bien que la présence d'un grand nombre d'éléments non connectés dans l'optimum global, sous forme de damier par exemple, puisse cacher un autre problème plus spécifique de choix de matériau, il est possible de modifier ou d'influencer l'outil d'optimisation topologique pour éliminer ces éléments non désirés.

2.3.3.1 Modification de la fonction d'évaluation

Pour aider l'outil d'optimisation topologique à supprimer naturellement les éléments non connectés, il est possible d'intervenir sur la fonction objectif pour pénaliser les solutions ayant plus d'éléments non connectés que d'autres. Puisque chaque élément non connecté augmente le périmètre total de la structure, C.D. Chapman [45] proposa en 1994 d'utiliser le périmètre de toute la structure ajouté à celui des trous, afin de forcer l'algorithme à préférer les solutions avec peu d'éléments non connectés. La fonction objectif à minimiser devient donc :

$$f' = -\frac{f}{\delta} \tag{2.5}$$

où δ est le périmètre de l'ensemble de la structure. Cette méthode force l'algorithme à sélectionner les individus qui présentent un périmètre plus petit et peu de trous. Toutefois, cette méthode peut éliminer des solutions avec peu d'éléments non connectés mais ayant un plus grand périmètre.

Dès 2004, dans le cadre d'un problème de déformation de matériaux avec une contrainte de poids à respecter, I.Y. Kim et O.L. de Weck proposèrent [46, 47] d'utiliser l'énergie de déformation pour distinguer les éléments non connectés et les saillies du reste du design. Par calcul, les éléments non désirables ont une énergie inférieure à celle des autres éléments de la structure. Ces éléments non désirables sont triés en fonction de leur énergie. Les rangs inférieurs sont éliminés rang par rang jusqu'à ce que la contrainte de poids soit respectée.

Dans le cadre de la conservation de volume déjà présentée dans [48], il est possible de permuter les éléments flottants de matière avec des éléments vides proches du bloc de matière le plus important, afin de regrouper la matière en un bloc et d'en conserver le volume total. Cette idée fut émise en 2006 par J.A. Madeira et son équipe [49]. L'algorithme repère en premier lieu les différents blocs déconnectés entre eux et compte le nombre d'éléments de chacun des blocs. En second lieu, l'algorithme calcule la distance entre chacun des blocs déconnectés par rapport au bloc dominant (en volume). Chacun à son tour, les blocs les plus éloignés sont raccordés au bloc le plus important. Ce raccord utilise le calcul de degré d'attraction de chaque cellule vide. Ce degré d'attraction est calculé en tenant compte du nombre de cellules contenant de la matière dans son voisinage ainsi que du contact avec les blocs déconnectés. Une fois le degré d'attraction calculé pour chacune des cellules vides, l'intervention s'effectue en remplissant les cellules les plus attractives.

2.3.3.2 Filtrage graphique des solutions

Les méthodes de filtrage telles que des opérations de post-traitement appliquées aux enfants créés dans l'algorithme permettent d'éliminer les éléments non connectés sans faire intervenir les opérateurs de reproduction. Cependant, supprimer des éléments trop hâtivement peu bloquer

le processus d'optimisation trop rapidement [50]. Kane et Schoenauer [51] présentèrent deux directions différentes de filtrage.

La première consiste à supprimer les éléments flottants lors de l'évaluation de la solution, le phénotype seul étant modifié. Cela introduit un niveau de dégénérescence dans la représentation des solutions, c'est-à-dire que plusieurs génotypes peuvent correspondre au même phénotype. Montré en 1994 par Radcliff et Surry [52], et confirmé numériquement en 1995 par Kane [50], cette dégénérescence n'est pas désirable pour l'optimisation.

La seconde direction consiste à supprimer dans le génotype, et donc dans le phénotype, les éléments non connectés. Cela consiste à utiliser une méthode dite de réparation.

2.4 Synthèse

Dans ce chapitre, sur base d'un étude bibliographique, nous avons effectué une revue des principaux algorithmes d'optimisation de type métaheuristique et des formalismes de distribution de matière utilisés dans le cadre de l'optimisation topologique par distribution de matière. C'est sur base de cette étude que nous choisissons au Chapitre 4 les algorithmes d'optimisation et les formalismes de distribution de matière qui seront comparés pour sélectionner une seule combinaison.

Chapitre 3

Stratégie de développement des outils d'optimisation topologique métaheuristiques

Pour atteindre les objectifs donnés en section 1.5, une stratégie globale a été mise en place. La section 3.1 présente la méthodologie de travail suivie pour permettre de comparer différents outils d'optimisation topologique de type métaheuristique. La section 3.2 présente les cas d'études sur lesquels s'appliquent les essais d'optimisation aux différentes phases de la comparaison. La section 3.3 décrit les outils d'évaluation des critères qui servent à comparer les résultats obtenus par les différents outils.

3.1 Méthodologie

Le choix de la méthodologie suivie est influencé par trois points. Premièrement, il existe de nombreux outils d'optimisation topologique de type métaheuristique à comparer. Deuxièmement, de par leur nature aléatoire ces outils impliquent le besoin d'effectuer un grand nombre d'évaluations. Et troisièmement, chaque évaluation fait intervenir une analyse par éléments finis, ce qui entraîne des temps de calcul parfois très élevés. La combinaison de ces trois remarques fait qu'il n'est pas

envisageable de travailler sur l'ensemble des algorithmes métaheuristiques avec un cas d'étude de type industriel. Toutes les combinaisons et adaptations ne peuvent donc pas être envisagées et comparées exhaustivement sur base d'un cas d'étude aussi complexe que souhaité. Il est nécessaire de travailler en plusieurs étapes afin de répartir les charges de travail en faisant varier la complexité des cas d'étude de manière adéquate.

3.1.1 Phase de sélection (Chapitre 4)

La phase de sélection explore l'espace des outils d'optimisation topologique. Pour y parvenir, elle se base sur quatre critères :

- La sensibilité des paramètres de contrôle ;
- La capacité à effectuer efficacement une optimisation topologique ;
- Le potentiel d'adaptation permettant une facilité d'adaptation ;
- La facilité à choisir les paramètres de l'algorithme d'optimisation.

Pour ces critères, il n'est pas nécessaire d'utiliser un cas d'étude complexe et lourd en temps de calcul. L'exploitation d'un problème simple de type graphique, section 3.2.1, convient tout à fait à cette première phase. Ce choix présente deux avantages et un inconvénient. Le premier avantage est que la solution optimale peut être imposée par l'utilisateur. Le second avantage est que le problème est rapide à évaluer puisqu'il ne demande aucun calcul de nature physique. L'inconvénient est qu'il est peu représentatif de la réalité des problèmes industriels d'optimisation topologique. A la fin de cette première phase, on sélectionne l'algorithme d'optimisation et le formalisme de distribution de matière qui feront l'objet de la phase d'adaptation.

3.1.2 Phase d'adaptation (Chapitre 5 et Chapitre 6)

Cette phase applique une série d'adaptations sur l'outil d'optimisation topologique de référence afin d'en améliorer les performances. Une adaptation consiste à améliorer la liaison existante entre l'algorithme d'optimisation et le formalisme de distribution de matière. Sans adaptation, l'algorithme modifie les paramètres d'optimisation sans prendre en compte leur caractère graphique.

Puisqu'il n'y a pas qu'un outil d'optimisation à étudier dans cette phase, on peut complexifier le cas d'étude. Le problème utilisé, décrit en section 3.2.2, fait intervenir l'électromagnétisme.

3.1.3 Phase des tests (Chapitre 7)

La dernière phase, la phase des tests, compose l'outil final à partir des différentes adaptations proposées dans la section précédente. Elle évalue ensuite cet outil sur la base des critères d'évaluation :

– La vitesse de convergence,
– La qualité de la solution finale,

Les résultats obtenus sont finalement comparés avec ceux obtenus avec des outils de référence présentés dans la littérature.

3.2 Les cas d'étude

Les cas d'étude utilisés aux différentes phases de la construction de notre outil d'optimisation topologique sont présentés dans cette section ainsi que pour chacun d'eux l'implémentation du module d'évaluation.

3.2.1 Phase 1 : Problème simple graphique

Le problème simple graphique est le cas d'étude utilisé lors de la phase de sélection. De par sa simplicité et de sa facilité d'évaluation, il permet des comparaisons de résultats avec un grand nombre d'évaluations. Le problème étudié entre dans la catégorie des problèmes inverses puisque l'optimum global, supposé unique, est imposé et doit être retrouvé au moyen de l'évaluation de la différence entre l'optimum et les solutions. Dans le cas du problème inverse étudié, la fonction objectif calcule un résultat reflétant les différences graphiques entre la cible et les solutions proposées par l'algorithme d'optimisation.

3.2.1.1 Description du cas d'étude

La topologie de la solution à retrouver est définie de manière à ce qu'elle possède des caractéristiques graphiques particulières afin de tester la capacité de l'outil à l'optimisation topologique, de forme et dimensionnelle. La topologie choisie, illustrée par la Figure 3.1, représente une géométrie à deux matériaux (A, B) dont l'un (B) domine l'espace de design et le second (A) enrichit cet espace de design de deux sous-domaines.

Le sous-domaine de gauche représente une entaille avec une forme complexe et le sous-domaine de droite représente un trou ovale. Cette

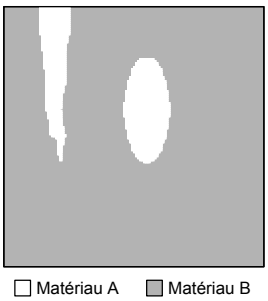

Figure 3.1 – Solution globale à retrouver composée de deux sous-domaines de matériau A inclus dans un domaine de matériau B avec un maillage de 200×200

topologie particulière a été choisie afin d'obliger l'outil d'optimisation topologique à rechercher simultanément une forme non régulière représentée par l'entaille et une forme difficile à approcher exactement représentée par le trou ovale.

3.2.1.2 Implémentation du module d'évaluation

Le calcul de la fonction objectif se fait via la différence de répartition de matière entre la solution cible et la solution proposée et passe par une projection des domaines de matière décrivant la topologie de la solution sur une structure fixe (Figure 3.2) ce qui permet de simplifier la création du modèle à évaluer. Toutefois, lors de l'évaluation de la solution, cette approche peut dégrader la topologie si la structure d'accueil, sous-domaines ou maillage, n'est pas suffisamment fine.

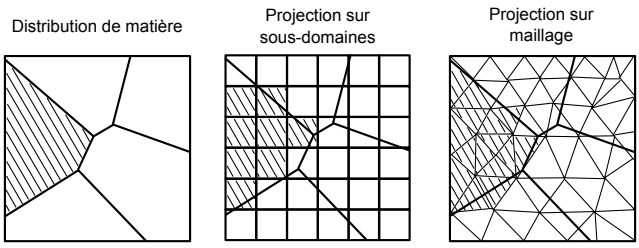

Figure 3.2 – Exemple de méthodes de projection : projection sur sous-domaines et projection sur maillage

3.2.1.3 Fonction objectif

Dans l'exemple traité, on a choisi une projection sur une structure en sous-domaines comportant 200×200 éléments. Si l'on définit la matrice $\mathbf{C} \in \mathbb{1}^{200\times200}$ comme étant la distribution de matière cible, avec $c_{ij} = 0$ et $c_{ij} = 1$ pour représenter respectivement le matériau A et le matériau B, et $\mathbf{P} \in \mathbb{1}^{200\times200}$ la solution proposée à évaluer, la fonction d'évaluation est définie comme étant la somme de l'opération XOR[1] sur chacun des termes des deux matrices :

$$f_{obj} = \sum_{i=0}^{200}\sum_{j=0}^{200} c_{i,j} \oplus p_{i,j} \tag{3.1}$$

La valeur de la fonction objectif est bornée par la valeur minimale qui est l'optimum global $f_{obj} = 0$, et la valeur maximale, $f_{obj} = 40.000$. Une variante de la fonction objectif est de l'exprimer en pourcentage d'erreur sachant que pour une erreur de 40.000, la fonction objectif vaudrait 100% et 0 pour une erreur de 0%.

Ce problème est simple à résoudre. Puisque chaque terme de (3.1) est indépendant des autres termes et est toujours positif, pour chaque solution, il est possible de l'améliorer au moyen d'un seul changement. Minimiser f_{obj} revient à minimiser chacun des termes ou d'une autre manière à trouver indépendamment la meilleure valeur de chaque variable d'optimisation. Toutefois, le problème est représentatif d'un problème topologique avec des formes complexes.

3.2.2 Phase 2 : Problème simple de type physique

Le problème simple physique se présente comme étant une évolution du cas simple graphique précédent. La fonction d'évaluation n'est plus une comparaison directe géométrique. Il est question ici de retrouver indirectement la distribution de matière au travers d'une fonction objectif évaluant la différence physique entre deux topologies. Afin de rester dans la continuité du problème simple graphique, il est choisi de conserver la topologie de la solution cible. Le problème d'optimisation reste un problème inverse. Ce choix permet de résoudre un problème physique tout en connaissant l'optimum global.

[1] L'opération XOR, dont le symbole mathématique est $\oplus$, est un ou exclusif. Il ne retourne 1 que lorsque les deux valeurs sont différentes et 0 autrement

3.2.2.1 Description du cas d'étude

Le thème choisi du problème est le domaine de la vérification de la qualité des matériaux. Un capteur électromagnétique (Fig. 3.3) est utilisé afin de mesurer les variations de flux électromagnétique lorsque ce capteur est déplacé le long d'une barre de fer ayant des défauts de fabrication c'est-à-dire des trous ou des fissures dans la matière. Cette variation de flux est exploitée pour détecter la présence de défauts et retrouver leur topologie.

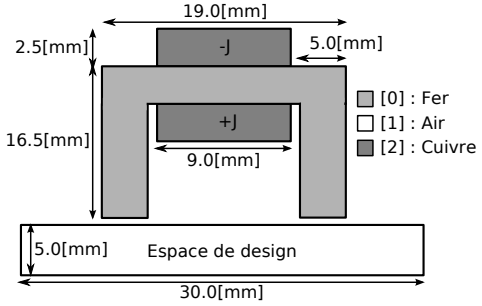

Figure 3.3 – Caractérisation du capteur électromagnétique

Afin de diminuer le temps de calcul, quatre hypothèses simplificatrices sont utilisées :

- Le problème est un problème plan ;
- L'induction magnétique dans l'espace de design est connue ;
- L'alimention électrique s'effectue par un courant DC ;
- Le capteur reste immobile.

Le problème est de type ON/OFF, ce qui implique que les matériaux dans l'espace de design représentent la présence ou l'absence de fer. Dans les zones d'absence de fer, les propriétés de l'air sont utilisés. Les défauts choisis conservent la topologie du cas simple graphique : un trou ovale au centre de la barre et une fissure située sur la gauche (Fig. 3.4). Les dimensions de la topologie subissent une projection paramétrique pour correspondre aux dimensions de l'espace de design réel, c'est-à-dire $[0; 0.03] \times [0; 0.005]$.

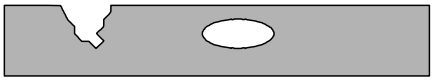

Figure 3.4 – Topologie à rechercher : description des défauts de la barre de fer

3.2.2.2 Implémentation du module d'évaluation

Pour calculer les champs magnétiques des différentes solutions, il est nécessaire d'utiliser les éléments finis. Un script Matlab© est utilisé comme interface entre l'algorithme d'optimisation et le logiciel éléments finis Comsol© (Figure 3.5) afin de créer le modèle élément finis et de l'évaluer.

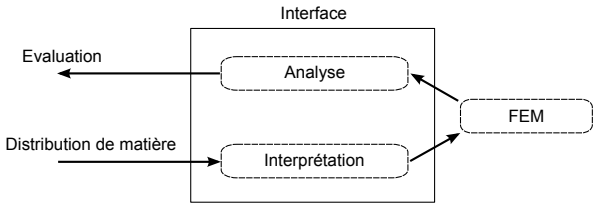

Figure 3.5 – Décomposition de l'outil d'évaluation en trois outils fonctionnels

La méthode utilisée pour générer le modèle élément finis crée autant de sous-domaines que de cellules et maille chacun d'entre-eux (Figure 3.6). Cette approche se justifie si le temps de calcul d'une évaluation est plus long que le temps nécessaire à la génération du maillage, ce qui est le cas ici.

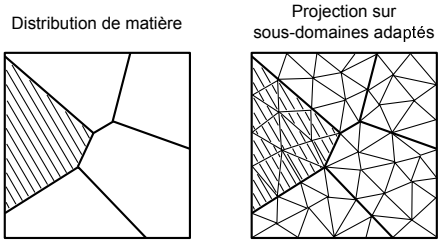

Figure 3.6 – Exemple de la projection sur sous-domaines adaptés

Cette méthode permet aux maillages de s'adapter aux solutions et évite ainsi de dégrader la solution décrite. Toutefois, elle présente des

formes aléatoires pouvant mener le mailleur à générer des maillages plus fins dans certaines parties de l'espace de design afin de garantir des dimensions équilibrées pour chaque mailles.

Il est donc possible que dans certaines situations, l'outil présente des difficultés à mailler une solution, pouvant ainsi mener à une erreur critique, à un blocage du logiciel dans l'algorithme de maillage ou à fausser les valeurs obtenues. Ces cas peuvent se présenter un certains nombre de fois lors d'un processus d'optimisation et pour parer à ce type de problème, il convient de détecter l'erreur et de donner une valeur infinie à la fonction objectif de telle sorte que la solution sera nécessairement éliminée par les processus de sélection.

3.2.2.3 Fonction objectif

Le problème à optimiser est défini par une seule fonction d'évaluation sans aucune contrainte. Cette fonction d'évaluation consiste à minimiser la différence entre l'induction magnétique cible $\vec{B}(x,y)$ correspondant à la topologie cible et l'induction magnétique de chacune des solutions proposées par l'outil d'optimisation, $\vec{B}'(x,y)$. La fonction objectif est :

$$f = \int_0^{0.005} \int_0^{0.03} \left| \frac{\vec{B} - \vec{B}'}{\vec{B}} \right|^2 dxdy \tag{3.2}$$

et peut être approchée par une version discrétisée en $n \times m$ points de mesures dans l'espace de design. Si on choisit $n = m = 50$, les points de mesures dans l'espace de design (Figure 3.3) sont définis par les relations suivantes :

$$\begin{aligned} x'_\psi &= k_1 \psi \\ y'_\phi &= k_2 \phi \\ &\text{avec } \psi, \phi \in [1, 50] \end{aligned} \tag{3.3}$$

et k_1 et k_2 valant respectivement $^{0.03}/_{50}$ et $^{0.005}/_{50}$. Ce qui nous donne la nouvelle fonction à optimiser :

$$\sum_{\psi=1}^{50} \sum_{\phi=1}^{50} \left| \frac{\vec{B}\left(x'_\psi, y'_\phi\right) - \vec{B}'\left(x'_\psi, y'_\phi\right)}{\vec{B}\left(x'_\psi, y'_\phi\right)} \right|^2 \tag{3.4}$$

3.2.3 Phase 3 : Problèmes complexes

Plusieurs problèmes complexes sont utilisés afin d'élargir le champ d'application des tests dans le domaine de l'électromagnétisme. Pour chaque cas d'étude, l'outil adapté obtenu est comparé à un outil d'optimisation topologique de référence défini en section 7.2 et le module d'évaluation utilisé est identique à celui décrit en section 3.2.2.2. La section 3.2.3.1 présente un problème de conception d'un actionneur linéaire à réluctance variable. La section 3.2.3.2 présente un problème de conception d'une machine synchrone à aimants permanents et enroulements concentrés.

3.2.3.1 Actionneur linéaire à réluctance variable

L'actionneur linéaire à réluctance variable étudié est composé d'une partie fixe, le stator, et d'une partie mobile, le mouveur, séparées de par un entrefer. Le déplacement du mouveur est provoqué par des forces réluctantes créées par des variations des champs magnétiques induits par les courants électriques dans les enroulements du stator.

Pour simplifier le problème, on suppose que le moteur présente une symétrie axiale et transversale (Figure 3.7). L'espace de design est limité à la description d'une moitié d'une encoche du stator et d'une moitié du mouver.

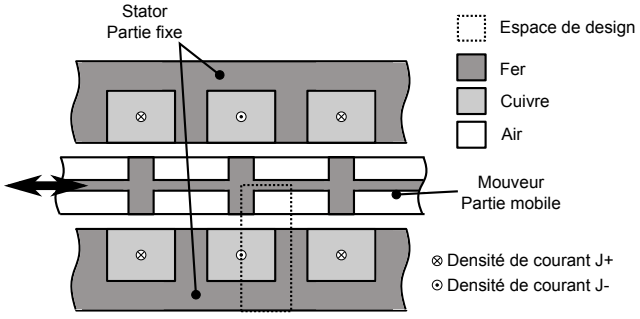

Figure 3.7 – Exemple de structure de l'actionneur

Cet espace de design principal est subdivisé en deux sous-espaces délimitant le stator et le mouveur. Un entrefer d'une épaisseur fixée est ajouté pour désolidariser mécaniquement les deux parties. Le schéma de la Figure 3.8 donne les dimensions utilisées des espaces de design.

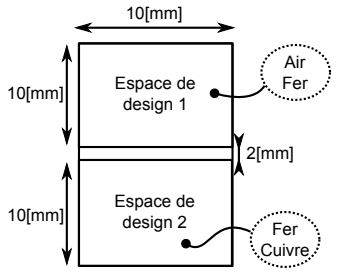

Figure 3.8 – Définition des dimensions, de l'espace de design et du contexte du cas d'étude de l'actionneur linéaire à réluctance variable

Les conditions aux frontières à appliquer sur l'espace de design complet sont : un potentiel magnétique fixé à 0 pour les bords inférieur et gauche, et $\vec{n} \times \vec{B} = 0$ pour les bords supérieur et droit.

Les deux espaces de design n'utilisent pas la même librairie de matériaux. Pour le stator, les matériaux sont le fer et le cuivre qui est traversé par une densité de courant fixée à $J = 10^6 Am^{-2}$. Pour le mouveur, les matériaux sont le fer et l'air. L'entrefer est composé d'air afin de garantir le déplacement de la partie mobile. La saturation magnétique du fer est aussi prise en compte et la caractéristique de cette saturation est donnée par le graphique de la Figure 3.9.

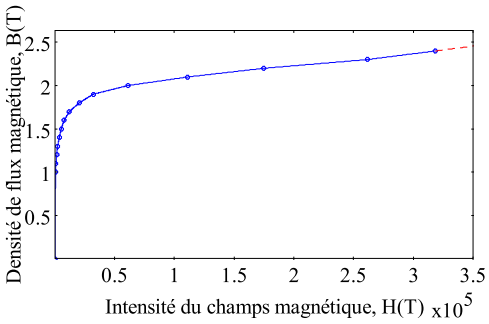

Figure 3.9 – Caractéristique magnétique du fer (Source *Comsol*©)

La Table 3.1 donne les valeurs de la densité de flux magnétique et de l'intensité du champs magnétique qui permettent de tracer la courbe

du graphique de la Figure 3.9.

Table 3.1 – Caractéristiques magnétiques du fer

B	H
0	0
1	663.14
1.1	1067.5
1.2	1705.23
1.3	2463.11
1.4	3841.67
1.5	5425.74
1.6	7957.75
1.7	12298.3
1.8	20462.8
1.9	32169.8
2.0	61213.4
2.1	111408
2.2	175070
2.3	261469
2.4	318310

La force en une position donnée du système est obtenue par la dérivée de la coénergie magnétique par rapport au déplacement :

$$F = \frac{\partial W_{cmag}}{\partial x} \tag{3.5}$$

Le design étant reproduit périodiquement, pour maximiser la force de l'actionneur, il suffit de maximiser la différence entre la coénergie magnétique du système en position de conjonction et la coénergie magnétique du système en position d'opposition :

$$f = W_{cmag,conj} - W_{cmag,opp} \tag{3.6}$$

Les deux positions sont illustrées par les schémas de la Figure 3.10. Puisque le pas de déplacement Δx est constant, il n'est pas pris en compte dans le calcul de la fonction à optimiser.

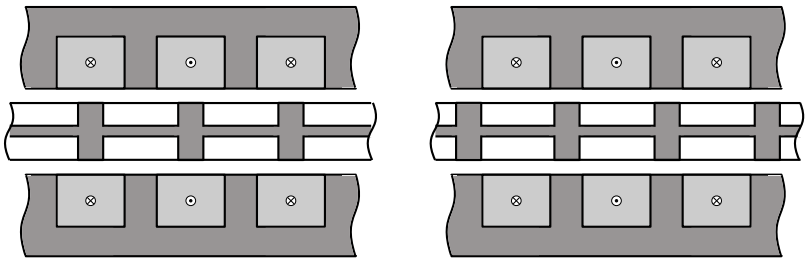

Figure 3.10 – Représentation de la position d'opposition (gauche) et de conjonction (droite) du mouveur de l'actionneur linéaire

La coénergie magnétique est calculée à partir de l'intégrale du flux magnétique ψ intercepté par le bobinage du stator :

$$W_{cmag} = \int_0^I \psi di \tag{3.7}$$

avec I le courant électrique passant dans le bobinage du stator. Afin de limiter le temps de calcul, l'intégrale (3.7) est calculée numériquement via la formule :

$$W_{cmag} = \frac{1}{2} \sum_{k=1}^{10} \left(\psi\left(i_{k-1}\right) + \psi\left(i_k\right)\right)\left(i_k - i_{k-1}\right) \tag{3.8}$$

Le problème étant un problème plan, nous pouvons écrire :

$$\psi\left(i_k\right) = \int_V A\left(\vec{i}_k\right) \cdot \vec{N} dV \propto \int_S A\left(i_k\right) dS \tag{3.9}$$

où S est la surface du conducteur , $\vec{N}$ est le vecteur densité du conducteur et $\vec{A}$ est le vecteur potentiel.

3.2.3.2 Machine synchrone à aimants permanents

Le moteur synchrone à aimants permanents étudié est représenté à la Figure 3.11. Ce problème a été proposé par Ishikawa [53]. Au stator on trouve trois phases, chacune formée de deux bobines diamétralement opposées entourant chacune une dent du stator. Le rotor comporte quatre paires de pôles. Le mouvement de la machine est dû aux couples électrodynamiques et réluctants créés par les variations du courant dans les

phases du stator. Le schéma de la Figure 3.11 présente l'orientation du champs rémanent des aimants utilisés et l'ordre de succession de phase.

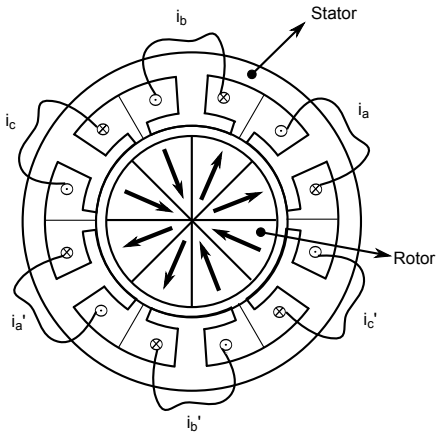

Figure 3.11 – Schéma de la machine synchrone étudiée

L'objectif de design est limité à l'optimisation topologique du rotor. Comme ce dernier est composé de 4 paires de pôle, l'espace de design est restreint à un huitième du rotor et est obtenue en appliquant des règles de symétrie de périodicité comme illustrées à la Figure 3.12.

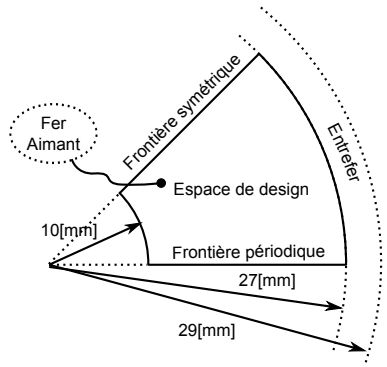

Figure 3.12 – Définition des dimensions et de l'espace de design.

La machine est alimentée au stator par des courants, i_a, i_b et i_c formant un système triphasé de grandeurs sinusoïdales avec pour densité maximale $J = {}^{10^6 A}/_{m^2}$. On suppose que quelle que soit la topologie du

rotor, il n'y a pas de phénomène de saturation au stator. Cette hypothèse permet de simplifier le modèle éléments finis en remplaçant le stator par des courants de surface. La librairie des matériaux utilisés est composée de fer et d'aimants avec un champs rémanent radial d'amplitude de 1.19T.

La fonction objectif vise à maximiser le couple moyen, τ_{moy}, de la machine tout en limitant le volume utilisé des aimants V_{aim} :

$$f = \frac{\tau_{moy}}{k\frac{V_{aim}}{V_{rot}} + 1} \tag{3.10}$$

où le facteur k permet de paramétrer l'importance du volume des aimants et V_{rot} est le volume total du rotor. Le couple moyen est obtenu par moyenne des couples pour 10 positions différentes du rotor. Entre chaque position, le rotor tourne de 3°. Chaque couple est calculé par l'intégrale, le long de la surface interne du stator, du produit de l'induction magnétique avec le courant électrique traversé, le tout multiplié par le rayon interne du stator. Pour faciliter les calculs par éléments finis, seul les courant de surface se déplacent, simulant ainsi le déplacement du rotor par rapport au stator.

3.3 Critères de comparaison

La possibilité de combiner différents algorithmes d'optimisation avec différents formalismes de distribution de matière, conduit à de nombreuses variantes d'outils d'optimisation topologique. Leur comparaison nécessite d'établir des critères qui permettront de les évaluer. Les outils d'optimisation topologique étudiés étant de nature stochastique, la tendance générale est de comparer la courbe de convergence obtenue soit après un nombre fixe d'évaluations, soit lorsqu'intervient une condition d'arrêt en fonction de l'évolution de la meilleure solution.

Cette section a pour objectif de présenter les différents critères sélectionnés pour caractériser une solution, soit qualitativement, soit quantitativement. La section 3.3.1 décrit le critère de comparaison classique, l'évolution de la convergence, à partir de laquelle on extrait de nombreuses informations. Ensuite la section 3.3.2 discute de la sensibilité des outils à leurs paramètres de configuration.

3.3.1 Courbe de convergence

Une courbe de convergence est une courbe représentant l'évolution de la meilleur solution obtenue à chaque itération du processus d'optimisation en fonction du nombre d'évaluations déjà réalisé. Cette courbe permet une analyse de la rapidité de convergence vers l'optimum pour un jeu de paramètres de contrôle choisi, associé à l'algorithme d'optimisation et au formalisme de distribution de matière. Étant donné le caractère stochastique des algorithmes utilisés, il est nécessaire d'exécuter le processus d'optimisation un nombre suffisant de fois avec le même jeu de paramètres et des conditions initiales différentes afin d'en observer la robustesse. Sur base de ces différents essais, on peut observer le taux de convergence du processus d'optimisation ainsi que la valeur finale convergée. Pour tracer les courbes de convergence de la Figure 3.13, les évaluations sont exprimées en fonction du pourcentage de l'évolution du processus d'optimisation.

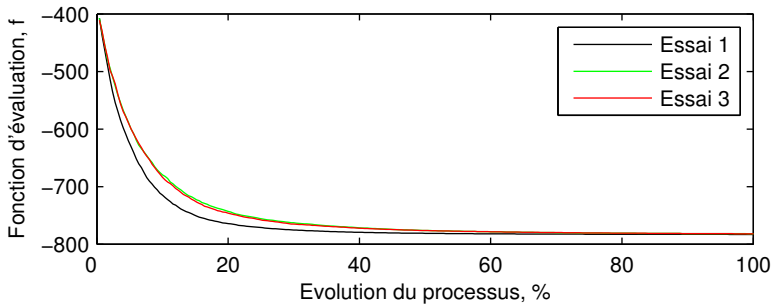

Figure 3.13 – Exemple de courbes de convergence

En plaçant dans un même graphique les courbes de convergence obtenues pour différents essais, il est possible de comparer facilement les valeurs obtenues f_{min}, f_{max} et f_{med} de chacun des outils.

3.3.2 Courbe de sensibilité sur un paramètre

Les outils d'optimisation topologique basés sur des algorithmes d'optimisation stochastique peuvent reposer sur le choix de plusieurs paramètres de contrôle. A priori, la valeur de ces paramètres peut être déterminante sur les capacité d'optimisation des algorithmes. Pourtant, aucune règle stricte ne permet de leur donner une valeur optimale, même

s'il existe des règles de bonne pratique. Ces règles n'indiquent qu'un ordre de grandeur des valeurs à donner.

Avec cette difficulté ajoutée au coût élevé de chaque évaluation de solution dans le cadre de l'optimisation topologique, il est très peu concevable de lancer une étude paramétrique avant chaque optimisation pour distinguer le meilleur jeu de paramètres. En supposant que chaque paramètre de contrôle a une influence différente sur les capacités d'optimisation des outils d'optimisation topologique, on peut supposer qu'il est possible de trouver l'outil ayant une sensibilité plus faible quant aux choix de sa paramétrisation, réduisant aussi l'influence de l'utilisateur.

La courbe de sensibilité (Fig. 3.14) est le résultat visuel de cette sensibilité. Cette courbe permet de déterminer la sensibilité du résultat convergé au paramètre étudié sur une gamme de valeurs.

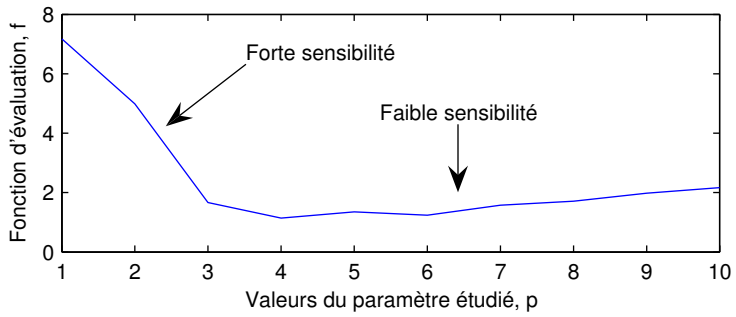

Figure 3.14 – Exemple de courbe de sensibilité

Pour obtenir cette courbe, il faut fixer tous les paramètres de contrôle sauf celui que l'on veut étudier. On fait alors varier celui-ci sur une gamme de valeurs. Pour chaque nouveau jeu de paramètres, il est nécessaire d'exécuter N_e fois le processus d'optimisation et de calculer la valeur médiane des valeurs convergées obtenues de la fonction objectif. Dans ce manuscrit, les graphes sont construits avec $N_e = 10$ sauf si N_e est mentionné.

L'outil d'optimisation topologique sera d'autant moins sensible à un paramètre sur une gamme de valeurs que la dérivée de la courbe aux valeurs considérées sera proche de 0. Cependant, cette étude se base sur un nombre élevé d'exécution du processus d'optimisation pour étudier la moyenne des résultats pour chaque jeu de paramètres de contrôle, elle sera donc réservée à la phase 1.

3.4 Synthèse

Dans ce chapitre, nous proposons une stratégie globale, divisée en trois phases, qui sera mise en œuvre pour comparer plusieurs outils d'optimisation topologique et développer notre outil d'optimisation topologique (Section 3.1). Ces trois phases, qui utiliseront des cas d'étude différents de complexité croissante afin de se rapprocher de la complexité de problèmes industriels, sont :

- la phase de sélection (Section 3.2.1) qui comparera différents outils d'optimisation topologique afin de déterminer l'outil le plus prometteur pour l'optimisation topologique. Cette comparaison est faite sur base d'un problème inverse simple graphique (Section 3.2.1) ;
- la phase d'adaptation (Section 3.2.2) qui modifiera l'outil sélectionné pour en améliorer les performances à l'optimisation topologique en adaptant l'algorithme d'optimisation. Le cas d'étude utilisé dans cette phase est à nouveau un problème inverse mais de complexité accrue par la prise en compte de l'électromagnétisme dans le calcul par éléments finis (Section 3.2.2) ;
- la phase de tests (Section 3.2.3) qui permettra de comparer les résultats obtenus par l'outil final à ceux obtenus par deux outils d'optimisation topologique provenant de la littérature. Deux cas d'étude sont utilisés, l'actionneur linéaire à réluctance variable et le moteur synchrone à aimants permanents.

Afin de comparer ces différents outils sur base des cas d'étude considérés, deux outils de comparaisons sont utilisés (Section 3.3) : les courbes de convergence qui donnent la valeur finale obtenue et permettent de déterminer le taux de convergence, et les courbes de sensibilités qui permettent d'observer la sensibilité des outils aux paramètres de contrôle.

Chapitre 4

Sélection de l'outil de référence

L'objectif de ce chapitre est de choisir un outil d'optimisation topologique de type métaheuristique qui présente une bonne combinaison d'algorithme d'optimisation et de formalisme de distribution de matière.

La section 4.1 décrit quels sont les algorithmes d'optimisation pris en compte dans l'étude ainsi que le choix de leur implémentation, puisque ceux-ci peuvent exister en différentes variantes sous le même nom. La section 4.2 reprend la même procédure pour les formalismes de distribution de matière. La section 4.3 compare les différentes combinaisons d'algorithmes d'optimisation et de formalismes de distribution de matière choisis aux sections précédentes afin d'observer leurs capacités à l'optimisation topologique et de sélectionner une seule combinaison pour la phase suivante de la thèse. La section 4.4 conclut ce chapitre.

4.1 Algorithmes d'optimisation

Les algorithmes d'optimisation tiennent un rôle primordial dans tout outil d'optimisation topologique. Ils permettent de faire évoluer une solution, ou un ensemble de solutions, en modifiant les valeurs des variables d'optimisation qui les caractérisent et ce en fonction de certaines informations disponibles. Comme annoncé en section 1.5, les algorithmes ciblés sont des algorithmes de type métaheuristique.

4.1.1 Sélection

Il existe actuellement de nombreux algorithmes métaheuristiques dont la plupart ont été créés en tentant de reproduire des phénomènes observés dans la nature. On recense principalement trois sources d'inspiration : la physique, la biologie de l'évolution et l'éthologie. Le graphique de la Figure 4.1 montre un classement des algorithmes les plus souvent utilisés.

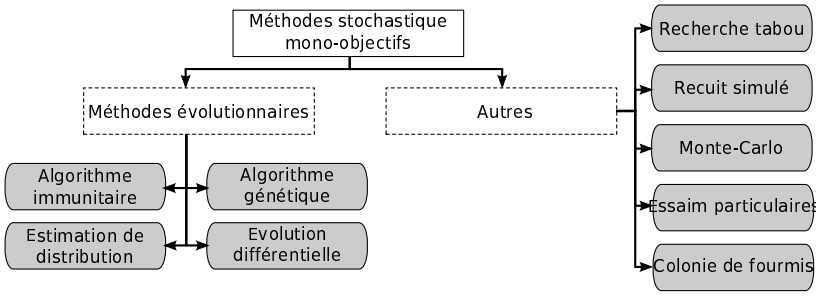

Figure 4.1 – Classement des méthodes stochastiques les plus souvent utilisées

Une classe particulière se distingue, celle des algorithmes évolutionnaires. Ces algorithmes se basent sur la théorie de l'évolution. Ils utilisent itérativement une méthode de sélection et une méthode de variation de la composition de leur population au travers de générateurs et de mutateurs. Un générateur est un opérateur permettant de créer une nouvelle solution soit à partir de rien, soit à partir d'une base de données, alors qu'un mutateur est un opérateur permettant de modifier une solution existante pour obtenir une nouvelle version. Par exemple, dans le cas d'un algorithme génétique, l'opérateur de croisement est un générateur et l'opérateur de mutation est un mutateur.

Parmi le grand nombre d'algorithmes métaheuristiques et leurs variantes, quatre ont retenu notre attention et ont été présélectionnés comme candidats pour la suite de ce chapitre.

– L'algorithme génétique : cet algorithme est sans conteste un choix incontournable pour l'optimisation topologique. Son approche de recherche globale et sa souplesse d'utilisation le rendent très puissant pour l'exploration ; il souffre néanmoins d'une faible vitesse de convergence, c'est-à-dire que le coût en évaluation est élevé pour améliorer la population ;

– L'algorithme du recuit simulé : au contraire de l'algorithme génétique, cet algorithme montre de grandes performances d'exploitation. Sa capacité d'exploration peut être garantie grâce à la possibilité qu'il offre d'accepter de moins bonnes solutions ;
– L'algorithme immunitaire : cet algorithme est une variante de l'algorithme génétique destinée à palier aux faiblesses de ce dernier dans sa phase d'exploitation. Récemment mis en place, il est aussi modulable que l'algorithme génétique bien qu'il se base sur des paramètres de contrôle plus délicats à définir ;
– L'algorithme de l'essaim particulaire : cet algorithme, basé sur une relation établie entre les solutions à chaque itération, possède une capacité d'exploitation très intéressante tout en conservant une faculté d'exploration dépendant du nombre de particules et des coefficients utilisés. Cet algorithme est aussi fortement utilisé mais offre le désavantage de ne pouvoir travailler que sur des variables continues et non des variables discrètes.

Le choix de ces algorithmes repose sur leur popularité dans la littérature mais prend aussi en compte les deux mécanismes principaux d'optimisation : à base de populations ou non et théorie de l'évolution ou non. D'autres algorithmes n'ont pas été pris en repris, tel que la recherche tabou ou l'algorithme par estimation de distribution. La recherche tabou est un algorithme de recherche local, or ce que l'on souhaite c'est de travailler avec des algorithmes de recherche globale afin de garantir une bonne exploration de l'espace des solutions. De plus, l'algorithme par estimation de distribution et l'algorithme tabou ne peuvent travailler directement qu'avec les formalismes de distribution de matière utilisant un nombre fixe de variables d'optimisation.

4.1.2 Implémentation

Sachant qu'un algorithme d'optimisation métaheuristique est une suite itérative d'utilisation d'opérateurs de sélection, de générateurs et de mutateurs, il s'agit maintenant d'expliquer la manière dont ces différents opérateurs ont été implémentés en fonction du type d'algorithme. Plusieurs implémentations ont été considérées et comparées notamment sur base de tests dont les détails se trouvent dans l'annexe A.

4.1.2.1 Algorithme génétique : GA

Les algorithmes génétiques sont des algorithmes s'inspirant de l'évolution des êtres vivant dans la nature selon les lois d'évolution de Darwin. Ces lois stipulent que chaque individu peut transmettre son patrimoine génétique à une nouvelle génération au travers d'un mécanisme de reproduction. Pour créer de la diversité, le patrimoine génétique peut être légèrement altéré par des mutations. Tous les individus composant une population se retrouvent en compétition avec les autres pour survivre et faire partie de la génération suivante dans un contexte de ressources vitales limitées. Les premiers développements de l'algorithme génétique remontent à 1975 et sont du fait de John Holland [54]. Ce dernier proposa une méthode axée sur trois étapes (Figure 4.2) recréant schématiquement l'évolution selon Darwin :

- une sélection de parents pour la reproduction(étape 1) ;
- une reproduction pour créer de nouveaux individus (étape 2) ;
- une sélection de descendants pour recomposer une population principale (étape 3).

Le fondement des mécanismes de sélection des parents et de sélection des descendants diffère en un seul point. La sélection des parents utilise un tirage aléatoire de chaque individu avec remise dans le lot. Ceux sélectionnés sont copiés dans une population dites de parents et chaque individu a la possibilité d'être sélectionné plusieurs fois. Alors que la sélection des descendants utilise un tirage aléatoire sans remise dans le lot. Chaque individu ne peut donc être sélectionné qu'une seule fois.

L'analogie avec l'évolution naturelle d'une population permet de considérer qu'un individu de la population est une solution et une population de N_s individus un ensemble de N_s solutions.

Au cours de chaque étape d'une génération de l'algorithme génétique, plusieurs populations temporaires sont créées. Afin de les distinguer, la population principale, la population des parents, la population des enfants et la population des descendants sont respectivement notées $\mathbf{X}^{(k)}$, $\mathbf{X}^{(k)'}$, $\mathbf{X}^{(k)''}$ et $\mathbf{X}^{(k+1)}$. Pour la population d'enfants, on distingue trois états différents : les clonés ${}_{cl}\mathbf{X}^{(k)''}$, les croisés ${}_{cr}\mathbf{X}^{(k)''}$ et les mutés ${}_{mu}\mathbf{X}^{(k)''}$. Le schéma de la figure 4.3 illustre l'évolution des matrices au sein de la $k^{ème}$ génération. Par la suite, pour simplifier les notations, l'exposant (k) est toujours sous-entendu.

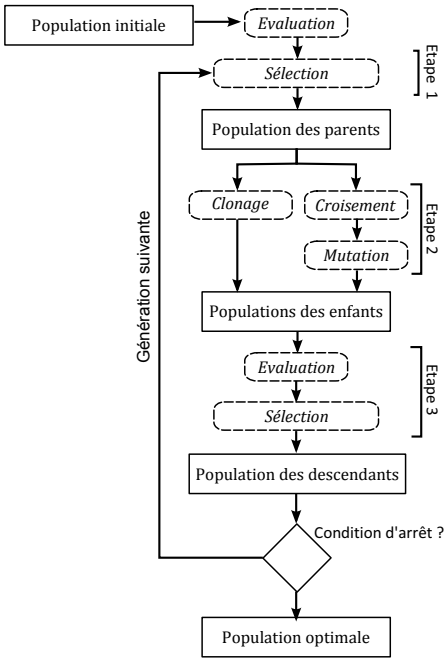

Figure 4.2 – Schéma d'évolution des processus de l'algorithme génétique

Etape 1 : Sélection des parents. Dans l'objectif de créer une population de N_s enfants, l'algorithme a besoin de définir quels sont les individus de la population principale qui auront le rôle de parents. L'étape de sélection des parents doit favoriser les individus ayant une meilleure évaluation aux objectifs afin qu'ils puissent transmettre, plus facilement que les autres, leur patrimoine génétique. Toutefois, une méthode ne sélectionnant que les meilleurs individus à cette étape limitera la diversité de la population, ce qui implique une convergence rapide vers une solution au détriment de l'exploration de l'espace des solutions. Pour éviter une convergence trop hâtive, la sélection doit pouvoir fournir une chance à de moins bons individus.

Trois méthodes de sélection de parents, décrites en Annexe A, sont étudiées : la copie, le tournoi et la roulette. La première méthode ne sélectionne aucun parent et permet à toute la population de se reproduire, alors que les deux méthodes suivantes mettent en confrontation les individus. La deuxième méthode imite la compétition entre les individus

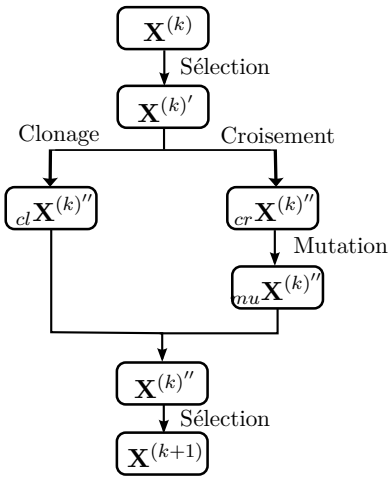

Figure 4.3 – Évolution des matrices des populations sur la génération k

lors de la reproduction en confrontant ceux-ci par duel pour déterminer ceux qui auront le mérite de se reproduire. La troisième méthode sélectionne aléatoirement chaque individu en fonction d'une probabilité associée à chacun. Cette probabilité est calculée par une relation liant la performance de l'individu et celles des autres.

Les résultats obtenus, repris dans l'annexe A, montrent que les trois méthodes sont efficaces mais présentent un niveau d'élitisme différent. Comme nous souhaitons privilégier un compromis entre la vitesse de convergence et la qualité d'exploration de l'espace des solutions, il convient de choisir une méthode semi-élitiste, c'est-à-dire la méthode de sélection par tournoi.

Une manière d'implémenter la méthode de sélection par tournoi consiste tout d'abord à utiliser 2 vecteurs, $\vec{u}$ et $\vec{v}$, de longueur N_s dont les valeurs font références aux individus de la population principale. Ces vecteurs sont initialisés en attribuant à chacun des éléments une valeur aléatoire selon une loi de distribution discrète uniforme :

$$u_i \text{ et } v_i \sim U_d\,(1, N_s) \tag{4.1}$$

Ensuite pour construire la population de parents, on compare l'évaluation de paires d'individus (u_i, v_i) et on copie le meilleur individu :

$$\vec{x}'_i = \begin{cases} \vec{x}_{u_i} & \text{si } F(\vec{x}_{u_i}) \text{ meilleur que } F(\vec{x}_{v_i}) \\ \vec{x}_{v_i} & \text{si } F(\vec{x}_{v_i}) \text{ meilleur que } F(\vec{x}_{ui}) \end{cases} \tag{4.2}$$

Le choix des vecteurs $\vec{u}$ et $\vec{v}$ est entièrement aléatoire, un individu peut être sélectionné plusieurs fois et donc être copié plusieurs fois dans la population de parents. En outre, tout individu possède une chance d'être sélectionnée puisqu'il peut toujours tomber en duel avec un autre individu de moins bonne performance.

Etape 2 : Procédure de reproduction La reproduction est l'étape primordiale de l'algorithme génétique. Elle permet de brasser le patrimoine génétique pour créer de nouveaux individus grâce au générateur par croisement génétique. Une augmentation de la diversification du patrimoine génétique est garantie grâce aux mutateurs par des variations localisées dans le chromosome.

Avant de passer à l'étape de la création de la population d'enfants, il est nécessaire de générer les couples de parents. L'implémentation utilisée consiste à choisir aléatoirement deux vecteurs de taille N_s pour représenter les parents 1 et les parents 2 : $\vec{u}$ et $\vec{v}$.

La création de la population d'enfants à partir de la population de parents utilise deux opérations différentes : un générateur, le croisement génétique, et un mutateur, la mutation génétique. La reproduction débute par le croisement génétique de deux parents avec une probabilité p_c pour que cela arrive. Dans le cas contraire, l'enfant est simplement obtenu par copie de l'un ou de l'autre parent. L'implémentation de cette méthode utilise une valeur générée aléatoirement selon une loi uniforme continue, $r_i \sim U_c(0,1)$, et compare celle-ci au paramètre de contrôle p_c :

$${}_{cr}\vec{x}''_i = \begin{cases} Croisement\left(\vec{x}'_{u_i}, \vec{x}'_{v_i}\right) & \text{Si } r_i < p_c \\ Copie\left(\vec{x}'_{u_i} \text{ ou } \vec{x}'_{v_i}\right) & \text{Si } r_i \geq p_c \end{cases} \tag{4.3}$$

La copie d'un des deux parents est déterminée aléatoirement entre les deux. Pour effectuer l'opération de croisement, un vecteur masque, $\vec{m} \in \mathbb{1}^{N_p}$, est généré aléatoirement afin de déterminer quelles parties du parent 1 et du parent 2 seront combinées pour former un nouvel enfant suivant la règle :

$$Croisement :_{cr} x''_{i,j} \quad = \begin{cases} x'_{u_i,j} & \text{Si } m_j = 0 \\ x'_{v_i,j} & \text{Si } m_j = 1 \end{cases} \quad \text{avec } j = 1, 2, \ldots, N_p \tag{4.4}$$

Le masque $\vec{m}$ peut être généré de plusieurs manières. Trois méthodes sont comparées : la méthode à 1 pivot, la méthode à 2 pivots et la méthode uniforme. La méthode à 1 pivot génère un vecteur de 0 et choisit aléatoirement un des éléments de ce vecteur. A partir de cet élément, tous les éléments suivant sont fixés à 1, les autres sont maintenus à 0. La méthode à 2 pivots fonctionne de la même manière mais il défini deux éléments dans le vecteur. A partir du premier et jusqu'au deuxième, les valeurs sont fixés à 1 et les autres restent 0. La méthode de croisement uniforme est différente des deux précédentes. Un choix aléatoire entre 0 et 1 est effectué pour chaque valeur du vecteur masque indépendamment de la performance des deux parents et selon une loi uniforme continue U_c définie entre 0 et 1 :

$$m_j \quad = \begin{cases} 1 & \text{Si } U_c\,(0,1) > 0.5 \\ 0 & \text{Si } U_c\,(0,1) \leq 0.5 \end{cases} \quad \text{avec } j = 1, 2, \ldots, N_p \tag{4.5}$$

Les résultats de l'étude, donnés en annexe A, montrent qu'il existe une faible différence entre ces méthodes. Cependant, sur base des résultats obtenus dans d'autres études [51], notre choix se porte sur l'utilisation de la méthode de croisement uniforme.

Une fois l'opération de croisement effectuée, l'opération de mutation est utilisée pour injecter un peu de diversité dans le patrimoine génétique de la population. Pour chaque individu ${}_{cr}\vec{x}''_i$ issu d'un croisement, chacune des variables ${}_{cr}x''_{i,j}$ a une probabilité p_m de voir sa valeur changer :

$$Mutation :_{mu} x_{i,j} \quad = \begin{cases} {}_{cr}x_{i,j} & \text{Si } r_j < p_m \\ \alpha_j & \text{Si } r_j > p_m \end{cases} \quad \text{avec } j = 1, 2, \ldots, N_p \tag{4.6}$$

où α_j et r_j sont des valeurs aléatoires uniformes. Si $j \leq N_D$, où N_D correspond au nombre de variable discrètes dans le vecteur $\vec{x}_i$, alors $\alpha_j = U_d\,(\alpha_{D,\min}, \alpha_{D,\max})$ ou si $j > N_D$ alors $\alpha_j = U_c\,(\alpha_{C,min}, \alpha_{C_max})$

où les valeurs de $\alpha_{C,min}$ et $\alpha_{C,max}$ sont respectivement fixées à 0 et à 1 dans la suite de la thèse.

Trois méthodes de mutation ont été étudiées et notées par la suite (A), (B) et (C). Les trois mutations diffèrent sur l'application de la probabilité de mutation et/ou du choix de la nouvelle valeur α_j. D'une part, les mutations (A) et (C) utilisent une probabilité de mutation p_m à appliquer pour chaque variable d'optimisation, alors que la mutation (B) modifie N variables d'optimisation avec $N = p_m N_p$. D'autre part, les mutations (A) et (B) choisissent une valeur pour α_j sans tenir compte de la valeur de la variable d'optimisation ${}_{cr}x_{i,j}$ alors que la mutation (C) modifie la valeur de ${}_{cr}x_{i,j}$ par la relation suivante :

$$ {}_{cr}x_{i,j}^{(k+1)} = {}_{cr}x_{i,j}^{(k)} \pm \theta_j \tag{4.7} $$

avec $\theta_j = \beta_j \left|\epsilon_j\right|$, $\epsilon_j \sim \mathcal{N}\left(0, \frac{1}{3}\right)$ et β_j la distance minimale avec l'extrémité du domaine de définition de la variable d'optimisation. Si la valeur de la variable dépasse les bornes, elle est modifiée en la remplaçant par sa valeur symétrique par rapport au bord.

Les résultats, donnés en annexe A, montrent que la mutation (B) est plus performante, suivie de la mutation (A) et de la mutation (C). Toutefois, la mutation (B) ne peut pas toujours être utilisée pour des variables discrètes. De plus, pour l'optimisation topologique, le nombre de variables d'optimisation est généralement élevé et les mutations (A) et (C) ont le même comportement. Finalement, le choix se porte sur la mutation (A) pour sa simplicité d'implémentation et pour conserver un facteur aléatoire sur le nombre de variables modifiées. Effectivement, comme nous ne connaissons pas le nombre idéal de mutations à effectuer à chaque génération, il est mieux de considérer un nombre oscillant autour d'une valeur fixée pour donner plus de souplesse à la méthode.

En parallèle de la branche de reproduction croisement-mutation, le clonage permet de créer une population d'enfants en copiant la population principale. Cette méthode permet de conserver le patrimoine génétique et de le comparer au patrimoine génétique de la population d'enfants lors de la sélection des descendants. Cette méthode consiste à copier chaque variable de chaque individu :

$$ {}_{cl}\mathbf{X}'' = \mathbf{X} \text{ ou } \mathbf{X}' \tag{4.8} $$

Pour éviter la perte du patrimoine génétique, on a retenu le clonage de l'entièreté de la population principale.

Etape 3 : Sélection des descendants Suite au clonage, au croisement et à la mutation, la population d'enfants est composée de $2N_s$ individus. Cette étape permet de réduire la population en sélectionnant N_s individus pour former la population des descendants $\mathbf{X}^{(k+1)}$. Deux méthodes sont étudiées : la sélection par élitisme et la sélection par roulette. La sélection par élitisme consiste à ne sélectionner que les N meilleurs individus de la population des enfants en les comparants sur leur évaluation. La sélection par roulette est identique à celle décrite pour la sélection des parents.

Pour éviter de perdre les bons individus créés par l'outil d'optimisation, le choix de la méthode de sélection se porte sur la sélection par élitisme :

$$\textit{Sélection} : \mathbf{X}^{(k+1)} = N_s \text{ meilleurs } \left(\mathbf{X}''^{(k)}\right) \tag{4.9}$$

4.1.2.2 Algorithme du recuit simulé : SA

S'appuyant sur les travaux d'Hastings, Kirkpatrick et son équipe ont conçu l'algorithme du recuit simulé [55]. Le recuit est un procédé métallurgique de refroidissement d'une pièce métallique par paliers d'énergie puisque lors d'un abaissement rapide de la température, les cristaux peuvent ne pas se mettre en configuration optimale. En faisant l'analogie entre l'évolution cristalline de la pièce métallique lors de cette opération et l'évolution d'une solution lors du processus d'optimisation, l'évolution de la température du système par paliers dégressifs est simulée par un paramètre de contrôle, T, et le niveau d'énergie d'une solution est identifié à l'évaluation de celle-ci. Suivant la méthode du recuit, la température est abaissée par paliers afin d'amener la solution vers un minimum d'énergie.

L'algorithme du recuit simulé est une suite logique de la méthode classique de l'algorithme de Monte-Carlo dans sa variante de *marche aléatoire* sur les chaînes de Markov ou algorithme de Metropolis-Hastings [56, 57]. Comme dans l'algorithme de Monte-Carlo, une solution $\vec{x}^{(k)}$ choisie est modifiée tout au long du processus d'optimisation (Figure 4.4). A chaque itération k, $\vec{x}^{(k)}$ subit une modification pour former un

nouvelle solution, $\vec{x}^{(k+1)}$. Des règles de comparaisons sont ensuite définies afin de déterminer si elle ou sa précédente, $\vec{x}^{(k)}$ ou $\vec{x}^{(k+1)}$, est conservée comme référence pour l'itération suivante.

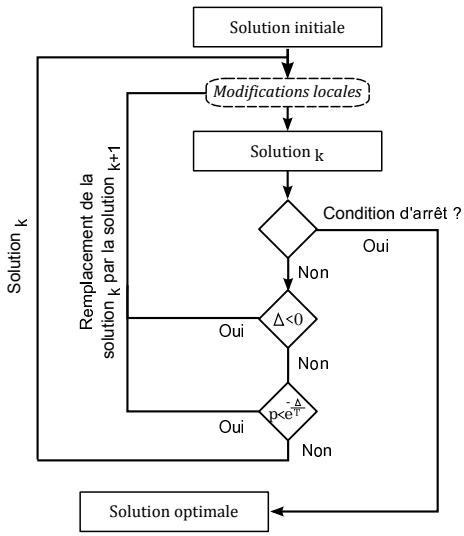

Figure 4.4 – Schéma d'évolution des processus de l'algorithme du recuit simulé

La méthode de comparaison de l'algorithme favorise ses capacités d'exploration. Pour chaque solution $\vec{x}^{(k+1)}$ ayant de moins bonnes performances que la solution $\vec{x}^{(k)}$, l'algorithme détermine une probabilité d'accepter cette nouvelle solution ou non. Cette probabilité, p, dépend à la fois de la différence d'évaluation, Δ, de la solution avec sa précédente, et à la fois d'un paramètre de contrôle qui est la température du système T. L'implémentation de cette loi est la suivante :

$$p = exp\left(\frac{\Delta}{T_k}\right) \text{ avec } \begin{cases} \Delta = F\left(\vec{x}^{(k+1)}\right) - F\left(\vec{x}^{(k)}\right) \\ T_k = g\left(k, k_{max}, T_0\right) \end{cases} \tag{4.10}$$

où $g\left(k, k_{max}, T_0\right)$ est la loi d'évolution de la température à déterminer. Suivant la relation donnée pour T_k, le choix de la valeur de T_0 détermine la probabilité d'accepter la première solution. Choisissant la méthode neutre, c'est-à-dire une probabilité de 50%, on pose $p = 0.5$ pour T_0. Ceci nous donne

$$0.5 = exp\left(\frac{-\Delta}{T_0}\right). \tag{4.11}$$

Toutefois, la valeur de Δ dépend fortement de l'évolution des solutions par rapport à la solution de référence. Lors des premiers essais, la valeur de Δ peut fortement varier selon les cas. On peut cependant utiliser la valeur moyenne de Δ en effectuant N_{essais} essais de perturbations sur la solution de référence.

$$0.5 = exp\left(\frac{-\Delta_{moy}}{T_0}\right) \text{ avec } \Delta_{moy} = \sum_{n=1}^{N_{essais}} \frac{(f_0 - f_n)}{N_e} \tag{4.12}$$

Ave ces N_{essais} essais, on peut déduire un estimateur T_0 pour avoir une probabilité de 50% d'accepter une mauvaise solution.

La configuration de l'algorithme du recuit simulé repose sur deux éléments : le choix de la méthode de perturbation de la solution $\vec{x}^{(k)}$ et le choix de la stratégie de refroidissement.

La perturbation de la solution est obtenue au travers de l'utilisation d'un mutateur. Pour la suite du projet, nous utilisons le même mutateur que celui choisi pour l'algorithme génétique (4.6).

Trois méthodes de refroidissement sont étudiées et diffèrent sur la relation entre la température et le numéro k de l'itération : évolution de la température soit par palier, soit linéaire, soit quadratique. Les résultats obtenus en annexe A montrent que la méthode quadratique surpasse les deux autres méthodes et est donc utilisée pour la suite de la recherche. L'évolution quadratique de la température est exprimée par la relation suivante :

$$T_k = \frac{T_0}{k_{max}} (k_{max} - k)^2 \tag{4.13}$$

où k_{max} est le nombre maximum d'itération du processus d'optimisation.

4.1.2.3 L'algorithme immunitaire : IA

L'algorithme immunitaire copie les principes fondamentaux du comportement de notre système immunitaire face à une infection. Ce dernier, pour s'adapter aux nouvelles infections, active les lymphocytes répondant le mieux aux attaques pour les multiplier. Cette multiplication

génère des mutations ce qui permet d'insérer dans la population de nouveaux lymphocytes proches de la version originale et potentiellement mieux adaptés pour répondre aux infections.

Par analogie à la situation biologique, les infections sont représentées par les fonctions d'évaluation du système à optimiser et les lymphocytes sont représentés par les solutions proposées par l'algorithme. Le schéma d'évolution de l'algorithme est présenté à la Figure 4.5.

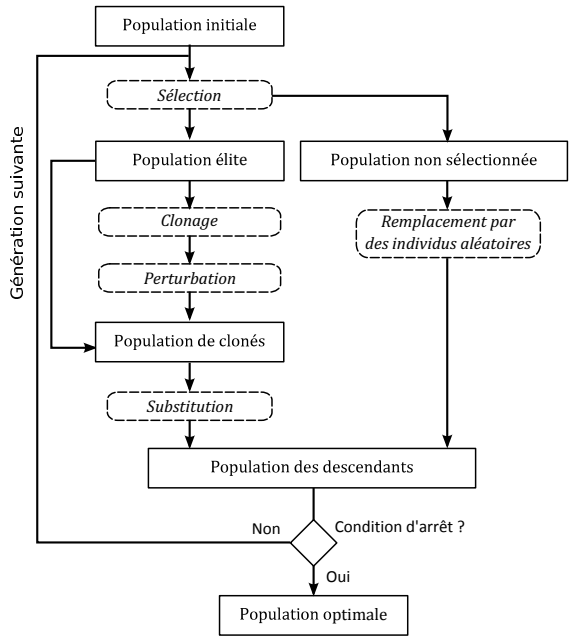

Figure 4.5 – Schéma d'évolution des processus de l'algorithme immunitaire

Le processus démarre à partir d'une population initiale de lymphocytes, $\mathbf{X}$, qui est évaluée et triée en fonction des évaluations de chacun. Ensuite, il y a prolifération des meilleures réponses à l'infection, les N_{sel} meilleurs lymphocytes sont sélectionnés pour le clonage. Le nombre de clones, $N_{clone,i}$, est calculé en fonction du rang de la solution i :

$$N_{clone,i} = \beta \frac{N_s}{rang, i}, \tag{4.14}$$

avec N_s, la taille de la population totale, $rang$ le classement en performance de l'individu dans la population et β un coefficient défini par

l'utilisateur. Afin de créer de la diversité, de légères modifications sont appliquées à la population de clones, $\mathbf{X}'$, grâce à un mutateur avec une probabilité p_m pour modifier les solutions clones :

$$\begin{aligned} Perturbation : x''_{i,j} &= \begin{cases} x'_{i,j} & \text{Si } r_j < p_m \\ \alpha_j & \text{Si } r_j > p_m \end{cases} \\ &\text{avec } j = 1, 2, \ldots, N_{clone} \\ &\text{avec } i = 1, 2, \ldots, N_s \end{aligned} \tag{4.15}$$

où α_j et r_j sont des valeurs aléatoires uniformes et N_{clone} le nombre total de clones. Si $j \leq D$, où D correspond au nombre de variable discrètes dans le vecteur $\vec{x}_i$ alors $\alpha_j = U_d(\alpha_{\min}, \alpha_{\max})$ ou si $j > D$ alors $\alpha_j = U_c(0, 1)$. Le mutateur utilisé correspond à celui de l'algorithme génétique pour les mêmes raisons.

Après avoir évalué la nouvelle population, le meilleur clone de chaque groupe j de clones, ${}_jx''_{best}$, est récupéré. Chaque meilleur clone plus performant que son original remplace ce dernier :

$$\begin{aligned} Sélection : x_j &= \begin{cases} {}_jx''_{best} & \text{Si } F({}_jx''_{best}) \leq F(x_j) \\ x_j & \text{Si } F({}_jx''_{best}) > F(x_j) \end{cases} \\ &\text{avec } j = 1, 2, \ldots, N_{sel} \end{aligned} \tag{4.16}$$

Finalement, toutes les solutions non sélectionnées au début de la génération sont remplacées par un nouveau tirage aléatoire.

4.1.2.4 L'algorithme des essaims particulaires : PSO

En se basant sur des populations d'individus évoluant en essaim tels que les groupes d'oiseaux en vol ou les bancs de poissons, J. Kennedy et R.C. Eberhart mirent au point un nouvel algorithme mémétique : l'essaim particulaire [58, 59]. Par observation, ils décrivirent une relation entre les individus d'une population lors de déplacements et reportèrent ces descriptions sous forme algorithmique. Ils en conclurent que l'effet de groupe permet d'améliorer les performances générales pour obtenir un effet de réaction intelligente face à un environnement hostile. Les conditions d'émergence d'une intelligence collective sont :

- Une information locale et limitée. Chaque individu n'a qu'une connaissance partielle de son environnement et n'a pas conscience de la globalité de celui-ci ;

– Un ensemble de règles simples. Chaque individu obéit à un ensemble restreint de règles simples ;
– Les interactions sont multiples. Chaque individu doit pouvoir communiquer avec plusieurs autres individus afin de partager son savoir ;
– La solution émergente est utile à la communauté. Chaque individu peut tirer parti de cette collaboration afin d'améliorer sa condition.

Le principe de l'algorithme d'optimisation par essaim particulaire, illustré par l'organigramme de la Figure 4.6, exploite la même notion d'influence entre les individus dans une population. Le déplacement de chaque individu est influencé par l'historique de son parcours et de celui des autres individus qui se retrouve dans son voisinage. La mémoire de chacun des individus est sollicitée et permet d'influencer le parcours de l'ensemble de la population.

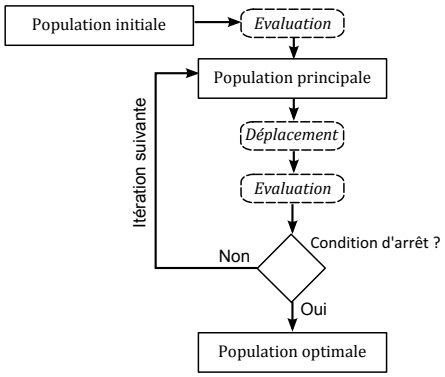

Figure 4.6 – Schéma d'évolution des processus de l'algorithme de l'essaim particulaire

Le processus d'optimisation est divisé en plusieurs étapes mais n'utilise aucune méthode de sélection. Lors de l'étape de déplacement, l'évolution d'une solution est définie par une direction et une amplitude, la vitesse de déplacement, $\vec{v}_i^{(k)}$:

$$\vec{x}_i^{(k+1)} = \vec{x}_i^{(k)} + \vec{v}_i^{(k+1)}\delta t \tag{4.17}$$

où $\vec{x}_i^{(k)}$ et $\vec{v}_i^{(k)}$ représentent respectivement la position et la vitesse dans l'espace des variables d'optimisation à la $k^{\text{ième}}$ itération de l'individu i, et δt, assimilé à 1, est la différence de temps entre deux itérations.

La valeur de $\vec{v}_i^{(k)}$ est la résultante entre plusieurs sources d'influence. La première source est celle de son inertie. La seconde source fait référence aux souvenirs de la meilleur position qu'il ait eu l'occasion de visiter au cours du processus. Et finalement, chaque individu est informé de la meilleur position obtenue par un individu du groupe auquel il appartient (Figure 4.7).

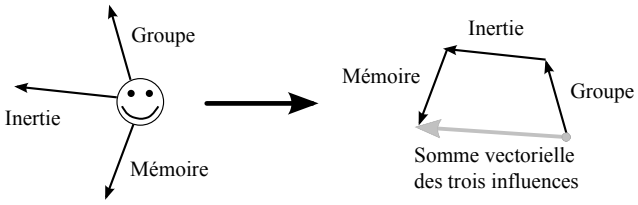

Figure 4.7 – Représentation des influences des individus de l'essaim particulaire

La notion de groupe dépend de la définition du voisinage. Les individus considérés comme voisins à la solution et la solution elle-même forme un groupe commun.

On peut donc maintenant implémenter $\vec{v}_i^{(k+1)}$:

$$\vec{v}_i^{(k+1)} = \overbrace{\omega r_0 \vec{v}_i^{(k)}}^{\text{Inertie}} + \overbrace{\alpha_1 r_1 \left(\vec{x}_i^{(k)} - \vec{b}\right)}^{\text{Mémoire}} + \overbrace{\alpha_2 r_2 \left(\vec{x}_i^{(k)} - \vec{g}\right)}^{\text{Groupe}} \tag{4.18}$$

où ω, α_1 et α_2 sont des coefficients fixés par l'utilisateur, r_0, r_1 et r_2 sont des valeurs aléatoire uniformes $U_c(0,1)$. Les vecteurs $\vec{b}$ et $\vec{g}$ représentent respectivement la meilleure position obtenue de l'individu et la meilleure position du groupe au cours des k itérations exécutées.

La notion de voisinage peut s'interpréter de différentes manières. Ce degré de liberté permet de déterminer les individus utilisés pour rechercher la meilleure solution du voisinage. Pour cela, trois méthodes particulières sont proposées :

- voisinage en anneau ;
- voisinage en étoile ;
- voisinage en rayon.

Le voisinage en anneau permet de relier un nombre m d'individus situés autour de chaque individu. Le voisinage en étoile permet de relier tous les individus entre-eux et le voisinage en rayon relie tous les

individus avec un seul individu qui sert alors de point de référence, de préférence le meilleur en première itération (Fig. 4.8).

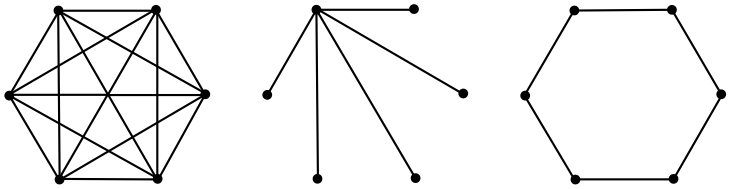

Figure 4.8 – Voisinages, de gauche à droite : étoile, rayon et anneau ($m = 2$)

Notre choix du type de voisinage repose sur l'étude [60] mettant en comparaison les trois méthodes sur un cas d'étude de type graphique simple. Cette étude montre que le voisinage en étoile est plus performant que les autres méthodes, avec une très légère avance.

Afin d'éviter d'obtenir des valeurs divergentes pour les variables d'optimisation, celles-ci sont bornées au domaine unité. La gestion du contact entre les individus et les bords de l'espace de recherche ajoute un degré de liberté. La collision d'un individu avec le bord de définition des variables d'optimisation est de type amorti. L'amortissement dissipe une partie de l'énergie et la vitesse du rebond est donc inférieure à la vitesse incidente. Cette loi d'amortissement remplace la relation (4.17) et est décrite suivant la loi :

$$x_i^{(k+1)} = \begin{cases} 1 - \Gamma\left(x_i^{(k)} + v_i^{(k+1)} - 1\right) & \text{Si } i = b \text{ et } x_b^{(k)} + v_b^{(k+1)} > 1 \\ -\Gamma\left(x_i^{(k)} + v_i^{(k+1)}\right) & \text{Si } i = b \text{ et } x_b^{(k)} + v_b^{(k+1)} < 0 \\ x_i^{(k)} + \Gamma v_i^{(k+1)} & \text{Si } i \neq b \end{cases} \quad \text{avec } i = 1, 2, \ldots, N_p \tag{4.19}$$

où b est l'indice de la variable en collision.

4.2 Formalismes de distribution de matière

Les formalismes de distribution de matière permettent de faire le lien entre les variables d'optimisation et la répartition de matière au sein d'un espace de design. Cette répartition est effectuée selon des règles décrites

par le formalisme choisi. C'est au travers de ces règles que l'espace de design est discrétisé en cellules. Les formalismes de distribution de matière peuvent être classés en deux familles : formalisme statique et formalisme dynamique [61].

La première famille de formalisme utilise une discrétisation fixée à priori et qui ne peut évoluer en cours d'optimisation autrement qu'au travers d'une subdivision des cellules (Fig. 4.9). Cette approche reste limitée et est contrainte par la présence des frontières des cellules de l'ancienne discrétisation.

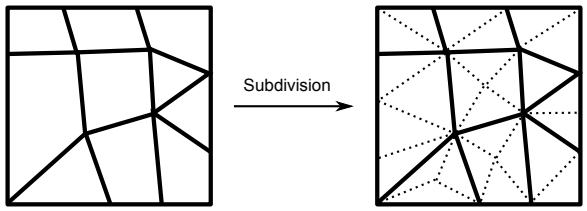

Figure 4.9 – Exemple de raffinement de la discrétisation d'un formalisme statique

La seconde famille de formalisme de distribution de matière utilise une discrétisation de l'espace de design qui peut évoluer en cours d'optimisation. Certaines variables d'optimisation sont utilisées pour caractériser cette discrétisation. Cette approche permet de faire évoluer la discrétisation afin de l'adapter au problème traité.

4.2.1 Sélection

Au chapitre 2, l'état de l'art a présenté cinq formalismes différents de distribution de matière. Deux formalismes nous semblent les plus intéressants à comparer : le formalisme Bit-Array comme formalisme statique et le formalisme à base d'un diagramme de Voronoï comme formalisme dynamique. Ces deux formalismes sont décrits dans les deux sous-sections suivantes et seront nommés dans la suite du texte sous la forme abrégée : formalisme Bit-Array et formalisme Voronoï.

Les formalismes à base de courbes de Bézier, de patchworks ou selon la méthode d'Heaviside n'ont pas été retenus pour l'étude. Le premier, bien qu'il présente de nombreux avantages pour l'étude de structures mécaniques, est très peu adapté pour une distribution de matière au sein de l'espace de design. La méthode ne permet pas de décrire toutes les

topologies. Le second, par son approche asymétrique dans la distribution de matière dans l'espace de design (Figure 4.10), n'est pas suffisamment générique [36]. Nous aurions été confronté au problème de la définition du matériau de fond. Ceci implique une probabilité différente pour chaque matériau de participer à la description de la topologie. Le troisième formalisme présente le même problème que la méthode patchwork concernant le manque de symétrie.

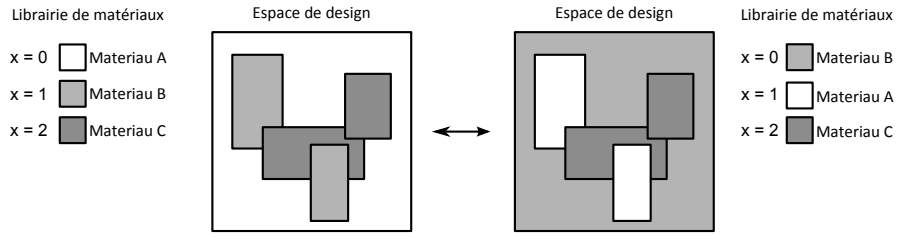

Figure 4.10 – Exemple de problème de symétrie lié à l'utilisation du formalisme à base de patchworks. Dans l'exemple de gauche, le Matériau A représente 70% de la matière après quatre ajouts de rectangle de matière alors que dans l'exemple de droite, c'est le Matériau B

4.2.2 Implémentation

Les formalismes de distribution fonctionnent sur la définition de règles d'interprétation des variables d'optimisation pour décrire une distribution de matière. Dans cette section, ces règles sont décrites pour les deux formalismes retenus, Bit-Array et Voronoï.

4.2.2.1 Formalisme Bit-Array

Méthode proposée en 1992 par Jensen [16], le formalisme Bit-Array est le formalisme statique le plus largement utilisé pour décrire une topologie à partir d'un jeu de variables de design.

Le formalisme consiste à discrétiser l'espace de design au moyen d'un quadrillage formant un damier de cellules de matière (Figure 4.11). Dans le cadre de l'optimisation topologique utilisant des algorithmes métaheuristiques, les variables de design, qui caractérisent la matière dans chaque cellule, sont le plus souvent des variables discrètes faisant référence à des matériaux choisis et disponibles pour l'optimisation.

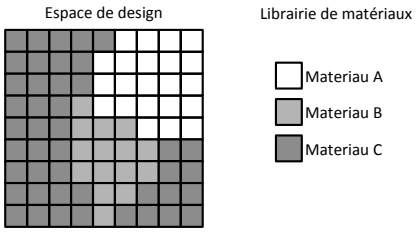

Figure 4.11 – Exemple d'application du formalisme Bit-Array dans un espace de design avec trois matériaux dans la librairie

Les variables de design dans le vecteur $\vec{x}^{(k)}$ sont ordonnées de manière à correspondre aux cellules de matière en parcourant de gauche à droite et de haut en bas :

$$\mathbf{x}^{(k)} = \left\{ x_{1,1}^{(k)}, \cdots, x_{1,n_L}^{(k)}, \cdots, x_{n_H,1}^{(k)}, \cdots, x_{n_H,n_L}^{(k)} \right\} \tag{4.20}$$

où n_H et n_L sont respectivement la résolution de discrétisation en hauteur et en largeur de l'espace de design.

4.2.2.2 Formalisme à base de cellules de Voronoï

Le formalisme à base d'un diagramme de Voronoï a été utilisé pour la première fois par Schoenauer [17] en 1996. Ce formalisme est un formalisme dynamique qui détermine la distribution de matières dans l'espace de design au travers d'une construction en deux étapes. La première étape consiste à créer un diagramme de Voronoï qui subdivise l'espace de design Ω en cellules à partir d'un ensemble de S points de références appelés les centres de Voronoï (Figure 4.12).

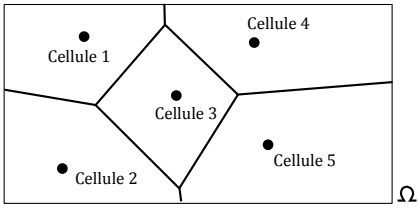

Figure 4.12 – Exemple d'application du diagramme de Voronoï

La cellule $V_S(p)$ associée au centre de Voronoï $p \in S$ est composée

par l'ensemble des points de l'espace de design qui obéissent à la règle suivante :

$$V_S(p) = \{x \in \Omega | \forall q \in S, d(x,p) \leq d(x,q)\} \tag{4.21}$$

On obtient les frontières entre cellules simplement en traçant des segments de droite représentant les points équidistants entre centres de Voronoï voisins.

La deuxième étape consiste à associer un matériau homogène à chacune des cellules (Figure 4.13).

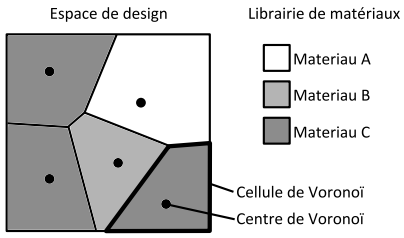

Figure 4.13 – Exemple d'application du diagramme de Voronoï dans un espace de design avec trois matériaux dans la librairie

Le vecteur de paramètres de design nécessaire est subdivisé en deux parties (4.22) : la position des points de références définie dans un système d'axes bien choisi et la référence des matériaux associés aux cellules :

$$\vec{x} = \left[\underbrace{x_0, x_1, ..., x_{N_c}}_{\text{Matériaux}}, \overbrace{x_{N_c+1}, ..., x_{3N_c-1}, x_{3N_c}}^{\text{Coordonnées}} \right] \tag{4.22}$$

avec N_c le nombre de cellules de la solution. De cette manière, l'algorithme utilise deux paramètres continus de position pour une représentation 2D et 3 pour une représentation 3D, afin de situer le centre d'une cellule de Voronoï dans l'espace de design et un paramètre discret pour déterminer le matériau utilisé.

L'optimisation s'effectuant sur un domaine fermé, il a été choisi de reproduire l'espace de design quatre fois en appliquant des symétries axiales sur la position des centres de Voronoï pour générer les bords de l'espace de design (Figure 4.14).

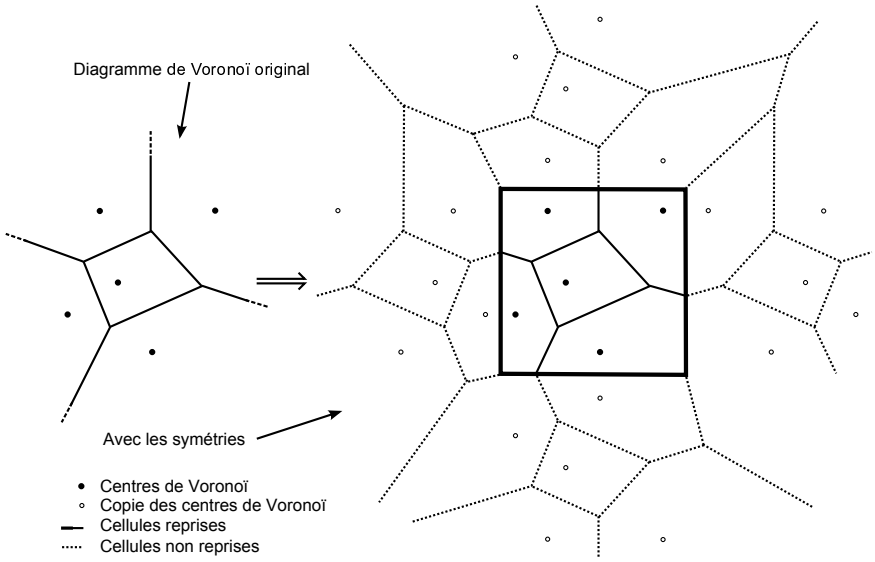

Figure 4.14 – Exemple d'application de quatre symétries au diagramme de Voronoï pour déterminer un domaine bornée

4.3 Comparaison des combinaisons

Les quatre algorithmes d'optimisation et les deux formalismes de distribution de matière choisis dans la section précédente permettent de former huit combinaisons. La sélection d'une seule combinaison se fera au travers de comparaisons sur base de deux critères étudiés dans les deux sous-section suivantes et visant à cerner les besoins minimaux pour effectuer une optimisation topologique :

- La sensibilité de l'outil par rapport aux choix des différents paramètres de contrôle ;
- La capacité de l'outil à effectuer une recherche topologique.

4.3.1 Sensibilité

Chaque outil d'optimisation topologique possède ses propres paramètres de contrôles. Ceux-ci dépendent du choix de l'algorithme d'optimisation et du formalisme de distribution de matière.

Suite aux mécanismes internes des algorithmes, certains des paramètres sont communs aux différents algorithmes alors que d'autres non.

Les différentes analyses et comparaisons sont regroupées en plusieurs sections. Les trois premières sections regroupent les paramètres qui sont communs aux différents outils d'optimisation topologique. On y retrouve le nombre de variables d'optimisation N_p associées à la discrétisation de l'espace de design, le nombre d'individus N_s composant les populations des algorithmes à population, et la probabilité p_m de mutation lié au mutateur commun. Les quatre dernières sections regroupent les paramètres propres à chacun des quatre algorithmes étudiés.

Pour tous les essais, sauf mention contraire, les paramètres de contrôle sont fixés et les valeurs choisies sont données dans la Table 4.1. Pour l'ensemble des algorithmes, nous avons fixé le nombre maximum d'évaluations à $k_{max} = 5000$, le nombre de variables initial à $N_p = 399$ dans le cas du formalisme Voronoï (133 cellules) et à $N_p = 400$ dans le cas du formalisme Bit-Array (matrice de 20×20), et les taux de mutations à $1/N_p$. Pour l'algorithme génétique, nous avons fixé la taille de la population à $N_s = 50$ et le taux de croisement à $p_{cr} = 90\%$. Pour l'algorithme immunitaire, nous avons fixé la taille de la population à $N_s = 5$, le coefficient de clonage à $\beta = 1$ et le coefficient de sélection pour le clonage à $N_{sel} = 50\%$. Pour l'algorithme du recuit simulé, nous avons fixé l'estimateur de la probabilité d'acceptation à $p_0 = 50\%$. Et pour l'algorithme de l'essaim particulaire, nous avons fixé les coefficients d'influence de déplacement à $\omega = 0.9$, $\alpha_1 = 1.7$ et $\alpha_2 = 1.7$, le coefficient de rebond à $\Gamma = 40\%$ et nous avons considéré une relation de voisinage en rayon.

Table 4.1 – Résumé des paramètres de contrôle autres que N_p fixés pour l'ensemble des outils d'optimisation topologique étudié.

	N_s	p_m	p_{cr}	p_0	β	N_{sel}	ω	α_1	α_2	Γ
GA	50	$\frac{1}{N_p}$	90%							
IA	5	$\frac{1}{N_p}$			1	50%				
SA	1	$\frac{1}{N_p}$		50%						
PSO	50						0.9	1.7	1.7	40%

4.3.1.1 Nombre N_p de variables d'optimisation (GA-IA-PSO-SA)

Le nombre N_c de cellules utilisées pour la description d'une solution est déterminant sur les performances de l'outil d'optimisation. Par son choix, on influence le niveau de précision dans la description d'une solution mais aussi la vitesse de convergence de l'algorithme d'optimisation :

- Pour un grand nombre de cellules discrétisant l'espace de design, il est possible de décrire avec plus de précision une topologie pour mieux approcher la solution optimale mais ceci implique plus de difficultés pour converger vers l'optimum ;
- Pour un petit nombre de cellules, l'algorithme converge plus rapidement, mais les solutions peuvent être moins bien définies en terme de dimension, de géométrie ou de topologie.

Le nombre de variables d'optimisation dépend du nombre N_c de cellules dans la représentation graphique de l'espace de design. Pour le formalisme Bit-Array, nous avons $N_p = N_c$ pour décrire le matériau associé à chaque cellule. Pour le formalisme Voronoï dans un espace 2D, nous avons $N_p = 3N_c$ pour décrire la position des centres de Voronoï et les matériaux associés aux cellules.

Les deux graphiques de la Figure 4.15 montrent pour le problème simple graphique présenté en section 3.2.1 la sensibilité des outils en fonction du nombre de paramètres disponibles pour décrire une solution.

Les mêmes résultats s'observent aussi bien pour le formalisme Bit-Array que pour le formalisme Voronoï. Dans les deux cas, l'essaim particulaires est le moins performant et la taille de la population joue un rôle important dans les performances des résultats. L'algorithme génétique dans le cas du formalisme Bit-Array est meilleur que l'algorithme immunitaire à partir de $N_p = 1225$. Le cas d'étude étant linéaire, les méthodes favorisant le nombre d'itérations au détriment de la taille de la population ont généralement un avantage sur les autres.

4.3.1.2 Nombre N_s d'individus (GA-IA-PSO)

Le rôle joué par une population composée de N_s individus est de pouvoir conserver une diversité dans les solutions obtenues afin d'aider l'outil à ne pas converger trop rapidement dans un optimum local. La taille optimale de la population peut varier en fonction de deux facteurs : le cas d'étude à optimiser et le déroulement du processus d'optimisation.

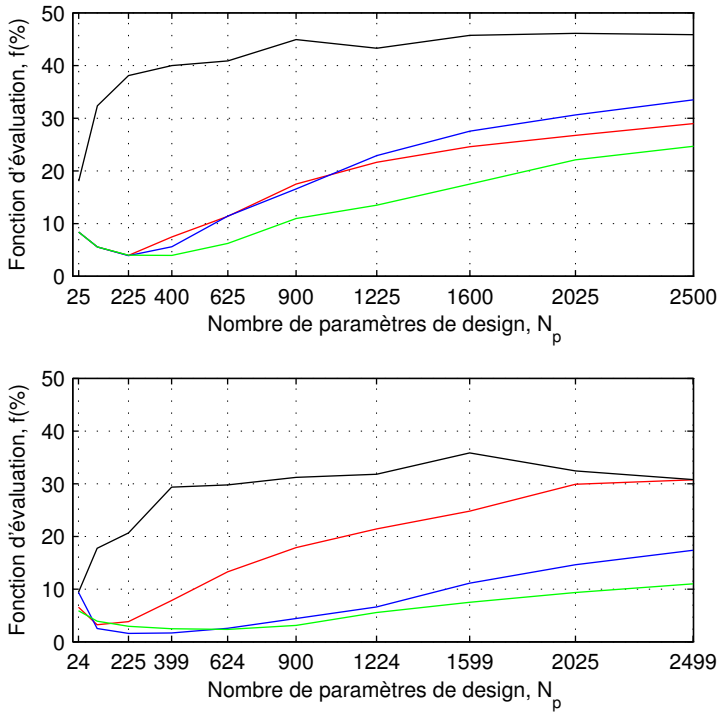

Figure 4.15 – **Bit-Array (haut) et Voronoï (bas) :** Graphique de sensibilité en fonction du nombre de cellules pour GA (rouge), IA (bleu), SA (vert) et PSO (noir)

Les graphiques de la Figure 4.16 montrent, dans le cas du problème simple graphique, que les outils sont peu sensibles à la taille de la population dans le cas d'un formalisme Bit-Array avec 20×20 cellules. On retrouve les mêmes résultats en remplaçant le formalisme Bit-Array par le formalisme Voronoï avec 133 cellules.

L'algorithme génétique avec peu d'individu a de meilleures performances que l'algorithme immunitaire. A partir de 10 individus, l'algorithme immunitaire est plus efficace. Son processus étant plus élitiste que l'algorithme génétique, il converge plus rapidement vers l'optimum. Cependant, le cas d'étude étant convexe, il n'est pas nécessaire d'utiliser de grandes populations pour explorer l'espace des solutions à chaque itération sauf dans le cas de l'essaim particulaires qui exploite cette plus grande diversification pour améliorer son exploration.

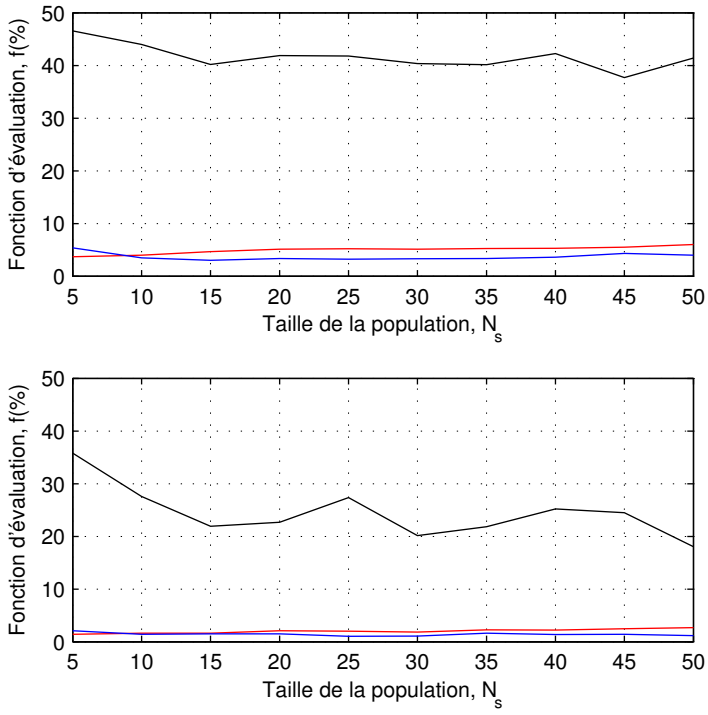

Figure 4.16 – **Bit-Array (haut) et Voronoï (bas) :** Graphique de sensibilité en fonction de la taille de la population pour GA (rouge), IA (bleu) et PSO (noir)

4.3.1.3 Probabilité p_m de mutation (GA-IA-SA)

Le mutateur fonctionne grâce au paramètre p_m qui détermine la probabilité de mutation d'une variable de design. Au travers du choix de celui-ci il est possible de créer de grandes perturbations ou de petites perturbations. Les résultats obtenus sont illustrés par les courbes de sensibilités à la Figure 4.17. À la fois pour le formalisme Bit-Array et le formalisme Voronoï, ces résultats montrent que la sensibilité de l'algorithme génétique est plus faible que pour les autres algorithmes pour des probabilités de mutation inférieures à $^{10}/_{N_p}$. Au delà de cette valeur, l'évolution de la sensibilité de l'algorithme génétique est similaire à celle des autres algorithmes. Les résultats sont en faveur du formalisme Voronoï.

Cette différence de résultat est la conséquence de l'utilisation d'un

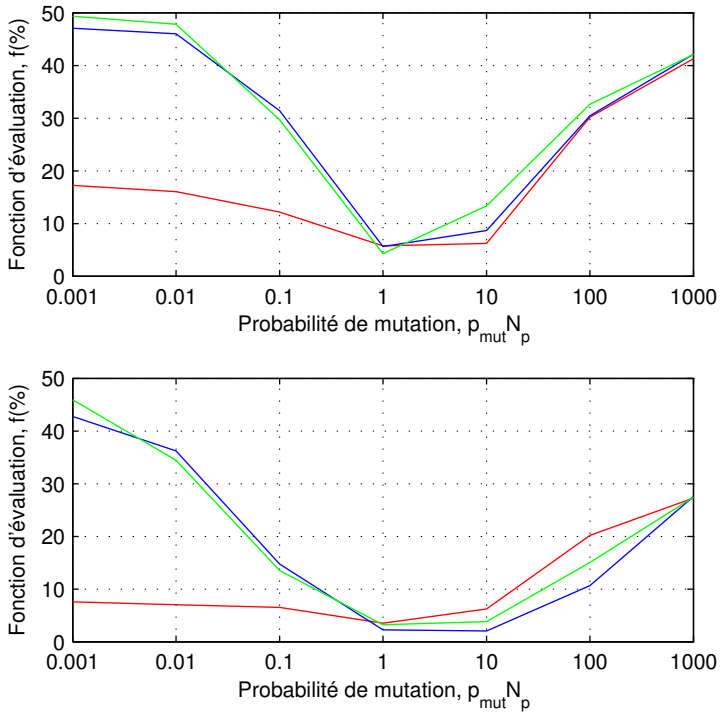

Figure 4.17 – **Bit-Array (haut) et Voronoï (bas) :** Graphique de sensibilité en fonction de la probabilité de mutation pour AG (rouge), IA (bleu) et SA (vert)

mécanisme de croisement pour générer de nouvelles solutions. Cependant, avec un taux de mutation élevé, au-delà de $^{10}/_{N_p}$, les perturbations sont telles que l'algorithme ne peut améliorer les solutions.

Les résultats montrent que les algorithmes couplés au formalisme Voronoï sont moins sensibles que couplés avec le formalisme Bit-Array.

4.3.1.4 Probabilité p_{cr} de croisement (AG)

L'outil de croisement est un des opérateurs les plus importants de l'algorithme génétique et lui est propre. Il permet de brasser un patrimoine génétique entre différentes solutions afin de tirer le meilleur parti de chaque individu. Les graphiques de la Figure 4.18 montrent la situation avec le formalisme Voronoï et avec le formalisme Bit-Array. Pour le cas du formalisme Bit-Array, les performances croissent avec le taux de

croisement. Alors que pour le formalisme Voronoï, les résultats montrent une insensibilité de l'outil à la probabilité de croisement. L'outil avec le formalisme Voronoï converge plus rapidement vers un optimum et ce pour toutes les valeurs de probabilité de croisement, ce qui n'est pas le cas pour la version avec le formalisme Bit-Array où le choix de taux de croisement est important.

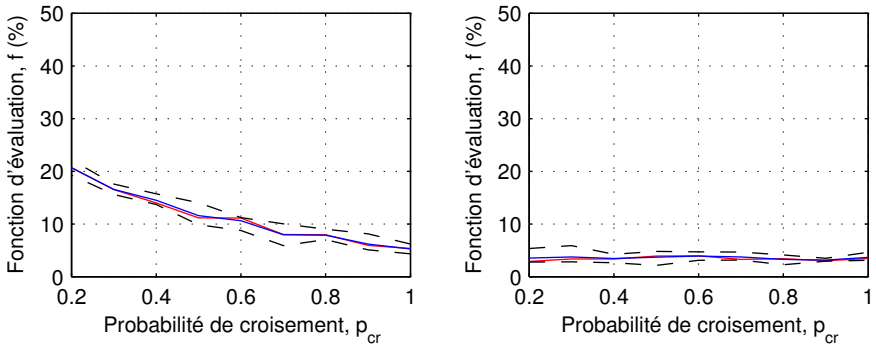

Figure 4.18 – Graphique de sensibilité pour le paramètre de croisement avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

Les courbes de sensibilités de la Figure 4.19 sont tracées à 50% du processus d'optimisation pour le cas du formalisme Voronoï et montrent que la sensibilité au paramètre de croisement est en permanence très faible lorsque l'algorithme est couplé à ce formalisme.

4.3.1.5 Nombre N_{sel} de sélections pour le clonage et coefficient β de clonage (IA)

L'algorithme immunitaire utilise deux paramètres de contrôle supplémentaires : le nombre N_{sel} de solutions sélectionnées pour être clonées et le coefficient de clonage β.

Le nombre de solutions sélectionnées influence l'algorithme sur sa capacité d'exploration. Un nombre élevé permet à l'algorithme d'exploiter des solutions de moindre performance mais diminue le nombre d'itérations de l'algorithme pour maintenir constant le nombre total disponible d'évaluations. Au contraire, à faible taux de sélection, l'algorithme se concentre sur les meilleurs lymphocytes et augmente ainsi sa vitesse de convergence tout en diminuant son exploration.

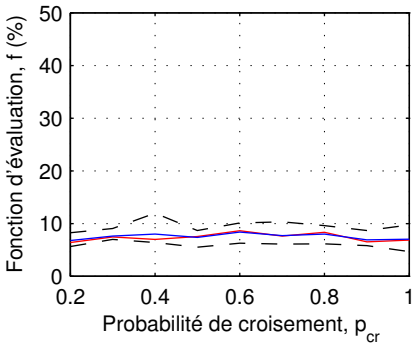

Figure 4.19 – Graphique de sensibilité pour le paramètre de croisement avec un formalisme Voronoï à 50% du processus d'optimisation

Les courbes de sensibilité pour le nombre de solutions sélectionnées sont montrées à la Figure 4.20. Pour les deux formalismes étudiés, les meilleurs résultats sont obtenus avec des taux faibles de sélection et les courbes sont croissantes dans les deux cas. La comparaison des résultats montre que ceux-ci sont meilleurs dans le cas du formalisme Voronoï et sont moins sensibles qu'avec le formalisme Bit-Array.

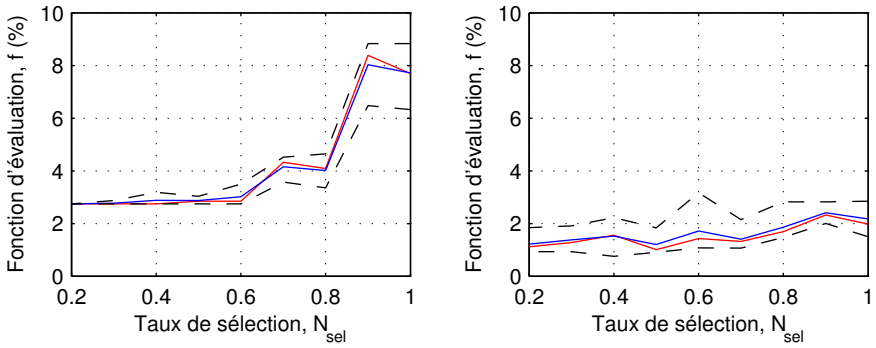

Figure 4.20 – Graphique de sensibilité pour le paramètre du taux de sélection avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

Pour le coefficient de clonage (Figure 4.21), les résultats varient beaucoup pour le formalisme Bit-Array et sont moins bons que pour le formalisme Voronoï où la sensibilité est aussi plus faible.

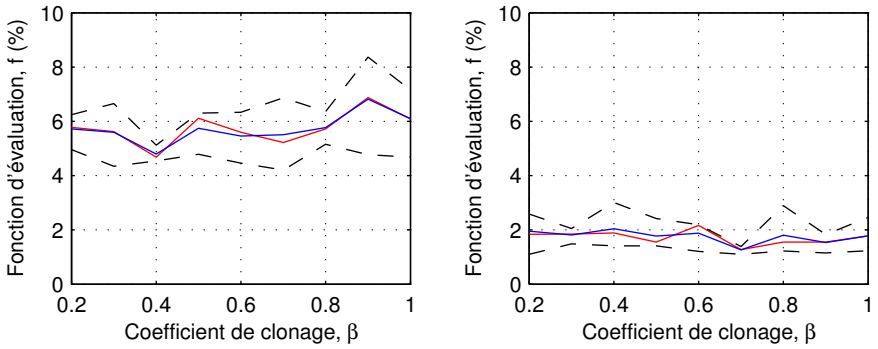

Figure 4.21 – Graphique de sensibilité pour le paramètre du coefficient de clonage avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

4.3.1.6 Probabilité p_0 d'acceptation d'une solution (SA)

L'estimateur p_0 représente la probabilité initiale d'accepter une solution moins performante. Son estimation est faite en calculant la moyenne obtenue en effectuant 100 essais autour de la solution initiale. On peut voir que, à la fois pour le formalisme Bit-Array (Figure 4.22-gauche) et pour le formalisme Voronoï (Figure 4.22-droite), il est préférable de prendre une probabilité faible ($p_0 < 40\%$).

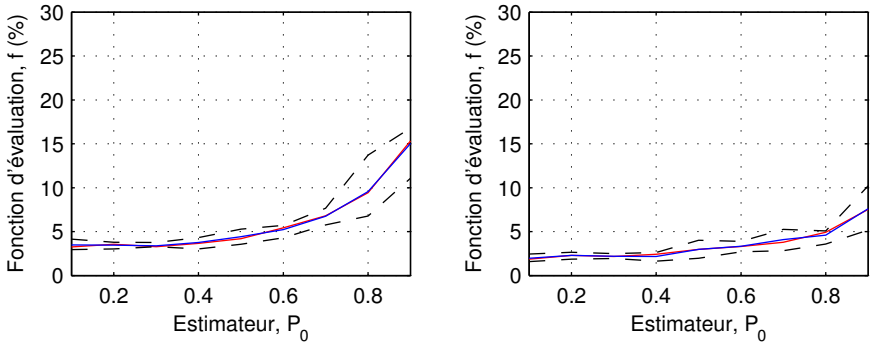

Figure 4.22 – Graphique de sensibilité pour le paramètre de l'estimateur avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

4.3.1.7 Coefficients α_0, α_1, ω d'influence du déplacement et coefficient Γ d'amortissement (PSO)

L'algorithme de l'essaim particulaire utilise quatre paramètres : trois pour l'évolution d'un individu (α_0, α_1 et ω) et un pour la gestion du rebond d'une variable aux bornes de l'espace de définition (Γ).

Les courbes de sensibilités des trois paramètres, α_0, α_1 et ω, sont représentées dans les Figure 4.23 à 4.25 et montrent de meilleures performances et une plus faible sensibilité avec le formalisme Voronoï. Les meilleurs résultats sont obtenus avec des valeurs élevées des paramètres.

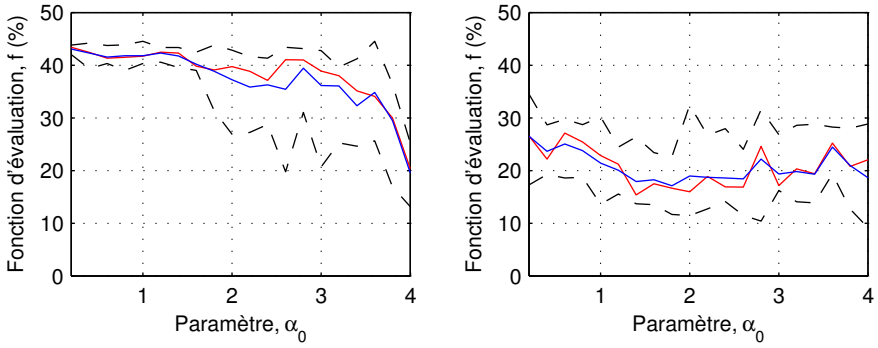

Figure 4.23 – Graphique de sensibilité pour les paramètres de comportement α_0 avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

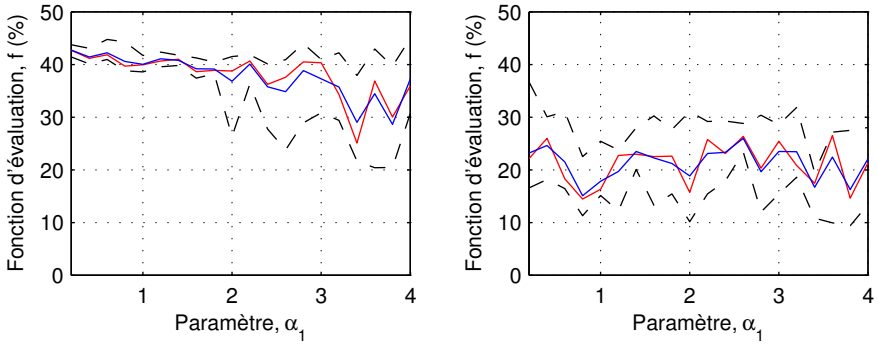

Figure 4.24 – Graphique de sensibilité pour les paramètres de comportement α_1 avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

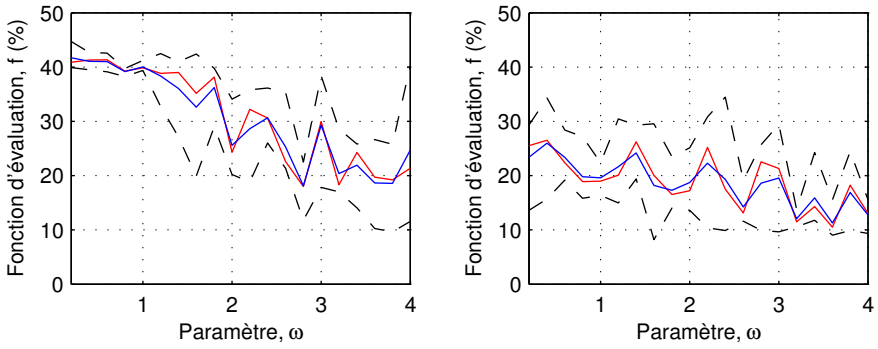

Figure 4.25 – Graphique de sensibilité pour le paramètre de comportement ω avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

Pour la gestion du rebond des individus sur les frontières de l'espace de design, on observe (Figure 4.26) que les résultats diffèrent entre le formalisme Bit-Array et le formalisme Voronoï. Pour le premier, les performances sont plus élevées avec une valeur de Γ proche de 1 alors que pour le second, le paramètre doit plutôt se rapproche de 0.5. La médiane montre que c'est effectivement pour cette valeur que la majorité des solutions se trouvent proche du meilleur résultat.

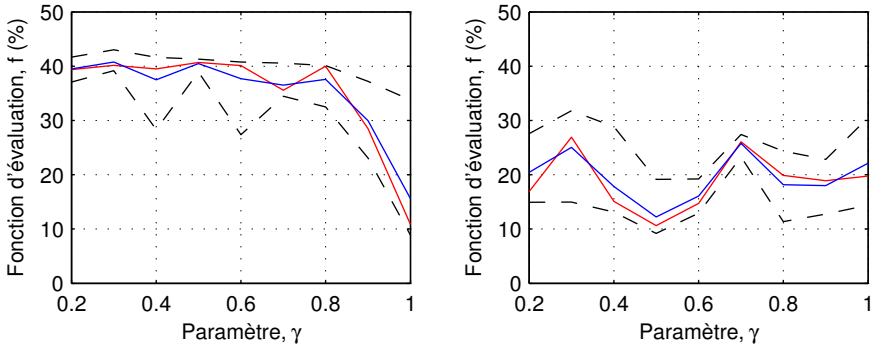

Figure 4.26 – Graphique de sensibilité pour le paramètre de rebond Γ avec un formalisme Bit-Array (gauche) et un formalisme Voronoï (droite)

Globalement, les performances obtenues sont en faveur du formalisme Voronoï avec un gain allant jusqu'à 50% dans le meilleur des cas. Seul dans le choix du paramètre d'amortissement Γ, les résultats pour

les deux formalismes s'intervertissent pour des valeurs proches de 1.

4.3.2 Convergence

On distingue deux types d'algorithmes utilisés pour composer les huit outils. D'une part, nous avons les algorithme génétique, immunitaire et du recuit simulé et d'autre part, l'algorithmes de l'essaim particulaire. Cette différenciation est faite sur base de la méthode de l'opérateur de modification des paramètres de design. Les trois premiers algorithmes, grâce à leur mutateur, peuvent atteindre toutes les solutions possibles, ce qui n'est pas le cas de l'algorithme d'essaim particulaire. Ce dernier, utilise une relation entre chacune des solutions pour changer les variables d'optimisation, c'est-à-dire la position des individus dans l'espace des solutions.

Afin d'illustrer les capacités des outils à atteindre l'optimum global dans le cas du Bit-Array, la Figure 4.27 illustre les convergences sur 100.000 évaluations pour un cas de 100×100 variables d'optimisation.

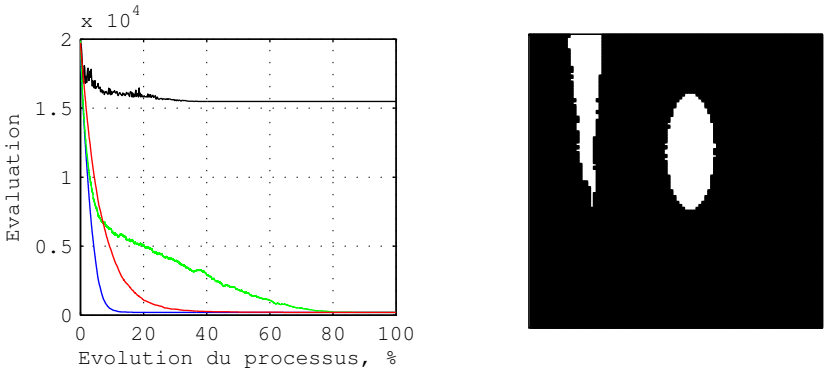

Figure 4.27 – Graphique de convergence des quatre algorithmes utilisées (gauche) et exemple d'optimum global obtenue (droite) pour AG (rouge), IA (bleu), SA (vert) et PSO (noir) et représentation de la meilleure solution obtenue

Les courbes de gauche montrent que l'algorithme génétique, l'algorithme immunitaire et l'algorithme du recuit simulé ont atteint la meilleure solution accessible avec les 10.000 variables d'optimisation. Ce qui n'est pas le cas de l'algorithme de l'essaim particulaire qui converge vers une mauvaise solution située aux alentours de $f_{obj} = 15000$ dans cet exemple.

Dans le cas de l'algorithme de l'essaim particulaire, la valeur des variables d'optimisation peut converger vers une valeur non souhaitée sans possibilité de la modifier avec les mécanismes classiques de l'algorithme. En effet, dans un cas extrême si la variable x_i tend vers 0 pour tous les individus, il n'est plus possible pour l'algorithme de changer cette valeur et ce pour toutes les configurations possible de paramètres de contrôles. On dit que cette variable est gelée et ne pourra plus être modifiée par la suite.

Plusieurs essais (Figure 4.28) sont effectués avec de plus grandes populations afin d'aider l'algorithme de l'essaim particulaire à explorer l'espace des solutions et de limiter le nombre de variables d'optimisation gelées. Ces nouveaux tests sont effectuées avec un exemple simplifié composé d'un maillage de 10×10 paramètres d'optimisation. Cet exemple montre la difficulté des essaims particulaires à éviter des paramètres gelés à de mauvaises valeurs, et donc à s'engouffrer dans un optimum local. Cependant, au plus la population est importante, au plus l'algorithme améliore le meilleur individu obtenu.

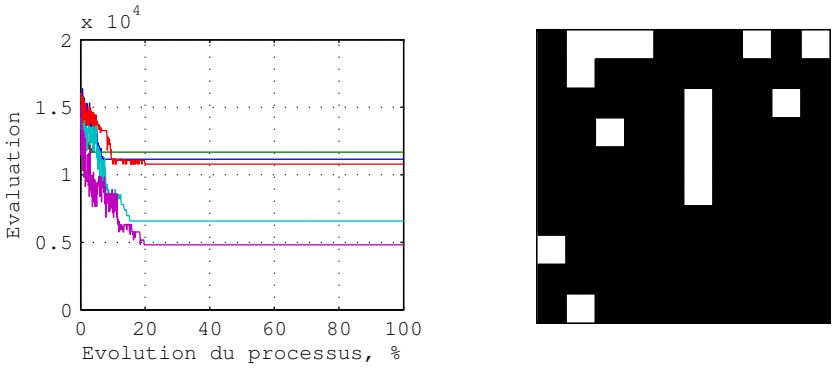

Figure 4.28 – Graphique de convergence pour le PSO avec 100, 200, 400, 800 et 1000 individus (gauche) et exemple d'optimum obtenu par le PSO avec 1000 individus (droite)

Les résultats de ces deux cas d'étude montrent qu'il est possible que l'algorithme de l'essaim particulaire ne retrouve pas la topologie et s'en écarte de manière significative.

Des tests identiques avec le formalismes Voronoï montrent les mêmes résultats (Figure 4.29). Les trois algorithmes GA-IA-SA arrivent non

seulement à obtenir une topologie équivalente à la solution cible mais de plus, peuvent approcher plus finement la forme finale. L'algorithme par essaim particulaire, en rouge sur le graphique, obtient à nouveau de faibles résultats.

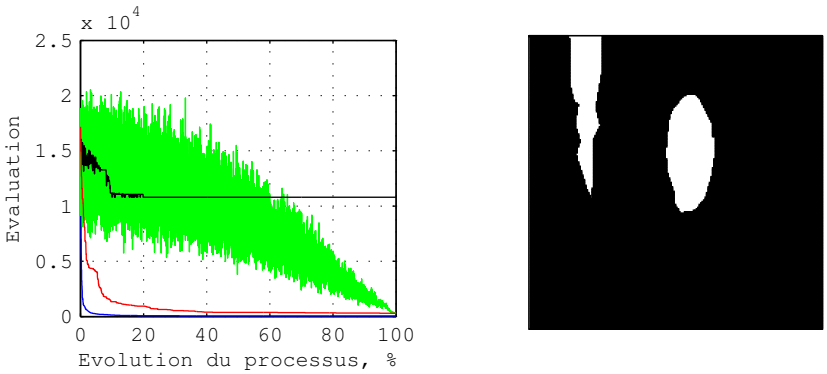

Figure 4.29 – Graphique de convergence des quatre algorithmes utilisées (gauche) et exemple d'optimum global obtenue (droite) pour AG (rouge), IA (bleu), SA (vert) et PSO (noir)

On peut toutefois observer sur cette dernière figure que la courbe de l'algorithme du recuit simulé présente une évolution chaotique non présente pour le cas avec le formalisme Bit-Array. Cette évolution s'explique par une plus grande tolérance de l'algorithme pour de mauvaises solutions avec le formalisme Voronoï. Sachant que le calcul de l'estimateur T_0 est effectué à l'itération $k = 0$, il est normal d'obtenir une grande tolérance aux perturbations puisque la perturbation peut être plus élevée dans le cas avec le formalisme Voronoï que dans le cas avec le formalisme Bit-Array lors d'une mutation.

4.4 Synthèse

Dans ce chapitre nous avons effectué une première étude visant à sélectionner, entre quatre algorithmes d'optimisation et deux formalismes de distribution de matière, la combinaison la plus prometteuse pour la phase suivante sur base d'une comparaison effectuée à l'aide des deux outils de comparaison. Les quatre algorithmes d'optimisation sont l'algorithme génétique, l'algorithme immunitaire, l'algorithme du recuit si-

mulé et l'algorithme de l'essaim particulaire, et les deux formalismes de distribution de matière sont le formalismes Bit-Array et le formalisme Voronoï.

Pour la comparaison avec les courbes de sensibilité (Section 4.3.1), nous avons deux parties distinctes : sensibilité sur les paramètres communs aux différents algorithmes d'optimisation et sensibilité sur les paramètres propres aux algorithmes d'optimisation.

Les Table 4.2 et Table 4.3 résument les résultats obtenus pour la comparaison sur base de la sensibilité aux paramètres communs. Quatre points sont observables :

- le classement des outils d'optimisation topologique dépend peu du formalisme de distribution de matière utilisé ;
- l'algorithme de l'essaim particulaire se retrouve toujours en dernière place du classement pour chaque paramètre étudié ;
- l'algorithme génétique est le seul à ne jamais se retrouver en dernière place du classement pour chaque paramètre étudié ;
- l'algorithme génétique et l'algorithme immunitaire ont des résultats forts similaires.

Table 4.2 – Résumé des résultats obtenus pour les paramètres de contrôle communs pour les 4 outils d'optimisation topologique étudiés avec le formalisme Bit-Array.

Paramètres	Outils			
	Formalisme Bit-Array			
	GA	SA	IA	PSO
Nombre de variables	+	++	0	-
Taille de la population	0		+	-
Probabilité de mutation	+	-	0	
	++	+	0	- -

La Table 4.4 résume les résultats obtenus pour l'étude des sensibilités aux paramètres propres. Les résultats sont comparés entre les deux formalismes et que le formalisme Voronoï donne de meilleurs résultats.

Pour la comparaison des courbes de convergence, on observe trois points :

- les conclusions sont identiques pour les deux formalismes étudiés ;
- l'algorithme de l'essaim particulaire à beaucoup de difficulté pour

Table 4.3 – Résumé des résultats obtenus pour les paramètres de contrôle pour les 4 outils d'optimisation topologique étudiés avec le formalisme Voronoï.

Paramètres	Outils			
	Formalisme Bit-Array			
	GA	SA	IA	PSO
Nombre de variables	0	++	+	-
Taille de la population	0		+	-
Probabilité de mutation	+	-	0	
	+	++	0	- -

Table 4.4 – Résumé des paramètres de contrôle propres pour les 4 outils d'optimisation.

Algorithmes	Paramètres	Bit-Array	Voronoï
GA	Probabilité de croisement	-	+
IA	Nombre de sélection	-	+
IA	Coefficient de clonage	-	+
SA	Probabilité de mutation init.	-	+
PSO	Coefficient de mémoire	-	+
PSO	Coefficient de groupe	-	+
PSO	Coefficient d'inertie	-	+
PSO	Coefficient de rebond	-	+

obtenir l'optimum topologique ;

– l'algorithme génétique, l'algorithme immunitaire et l'algorithme du recuit simulé convergent vers la solution optimale mais avec des vitesses différentes. Dans l'ordre de performance, nous avons l'algorithme immunitaire, l'algorithme génétique et finalement l'algorithme du recuit simulé.

L'algorithme génétique couplé au formalisme de Voronoï semble donc être un bon point de départ offrant de nombreuses possibilités d'adaptation et c'est donc cette solution qui fera l'objet de la phase de développement suivante de l'outil d'optimisation topologique.

Chapitre 5

Adaptation de la gestion de la population

Comme nous l'avons vu au chapitre précédent, l'outil choisi pour la suite de l'étude est composé d'un algorithme génétique et d'un formalisme Voronoï. Les avantages de cet outil en font un bon candidat aux adaptations graphiques pour l'optimisation topologique : la modularité de l'algorithme d'optimisation permet une grande liberté d'adaptations et le formalisme Voronoï permet de modifier dynamiquement la discrétisation de l'espace de design pour décrire précisément des solutions géométriquement complexes.

Ce chapitre se focalise particulièrement sur les adaptations apportées à l'algorithme génétique liées à la gestion de la population en cours d'optimisation. Les caractéristiques de la population influencent de manière significative l'évolution du processus d'optimisation. Sa taille et la diversité des individus au sein de cette population agissent sur les capacités de l'algorithme à explorer et à exploiter les recherches dans l'espace des solutions.

Deux aspects de la gestion de la population sont étudiés au sein de ce chapitre. La section 5.1 concerne les méthodes de gestion de la taille de la population principale au cours du processus d'optimisation. Différentes méthodes pour réguler la taille de la population à partir de différents facteurs sont proposées et comparées. La section 5.2 se focalise quant à elle sur la gestion de l'initialisation de la population de l'algorithme génétique. L'étude porte sur la diversification et la prise en compte de

la nature du problème, à savoir l'optimisation d'une topologie, pour créer des individus dans la population initiale. La section 5.3 fait la synthèse des résultats obtenus aux sections 5.1 et 5.2. Les comparaisons sont effectuées sur le problème physique de type simple décrit en section 3.2.2.

5.1 Evolution de la taille de la population

L'évaluation des performances des individus d'une population donnée représente généralement l'étape la plus coûteuse d'une itération. Par conséquent, nous travaillerons en considérant que le nombre total d'individus qui seront évalués est une constante (nous dirons plus brièvement le nombre d'évaluations) de telle sorte qu'une population comportant un petit nombre d'individus permettra un grand nombre d'itérations et inversement.

Dès lors, à nombre constant d'évaluations, le choix d'une trop grande population augmente le nombre d'évaluations par génération, ce qui implique une diminution de la capacité d'exploitation de l'outil en limitant le nombre d'itérations qu'il pourra effectuer, mais augmente sa capacité d'exploration. Au contraire, le choix d'une trop petite population limite les capacités d'exploration mais peut augmenter les capacités d'exploitation en permettant un plus grand nombre d'itérations.

Il est donc important d'utiliser une stratégie déterminant l'évolution de la taille de la population au cours du processus d'optimisation afin de libérer l'utilisateur de la responsabilité du choix de ce paramètre.

On peut préprogrammer la manière dont la taille de la population va évoluer en la faisant décroître de manière prédéterminée à mesure que le nombre d'itérations réalisées augmente. Bien que l'évolution de la taille de la population soit définie à l'avance, les résultats de cette approche peuvent être efficaces pour une loi bien choisie, comme on le verra à la section 5.1.1.

On peut aussi faire dépendre la taille de la population à chaque itération des caractéristiques de la population. Deux méthodes basées sur cette approche sont présentées aux sections 5.1.2 et 5.1.3.

5.1.1 Décroissance préprogrammée à priori : $N_s = N_s(k)$

La fonction programmée utilise directement le numéro d'itération associé à l'évolution du processus d'optimisation afin de paramétrer à l'avance l'évolution de la taille de la population. Cette paramétrisation est choisie par l'utilisateur. Ceci nous permet d'écrire la relation :

$$N_s = N_s(k) \tag{5.1}$$

Choix de la loi d'évolution

Le choix de l'évolution de la population se calque sur les besoins en exploration et en exploitation de l'outil pour maximiser ses performances. La taille initiale de la population est choisie de l'ordre de grandeur du nombre de variables d'optimisation, afin de garantir une bonne exploration. Ensuite, cette taille est réduite régulièrement pour augmenter le niveau d'exploitation, mais est limité à 4 individus afin de réduire les risques de convergence prématurée. Finalement, sur la fin du processus, nous choisissons de la réduire à une taille minimale.

Pour modifier la taille de la population en fonction de la vitesse de décroissance souhaitée, nous choisissons un profil de type :

$$N_s = N_s(k) = max\left(4, round\left(\frac{b}{k+a}\right)\right) \tag{5.2}$$

où a et b sont des paramètres à fixer en fonction des stratégies d'évolution souhaitées et la valeur 4 garantit une taille minimale (Figure 5.1).

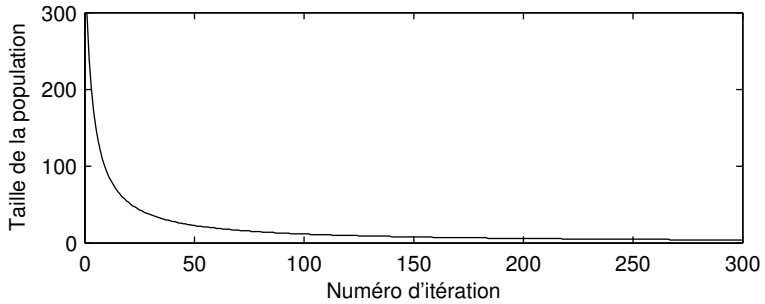

Figure 5.1 – Exemple d'évolution de la taille de la population $N_s = N_s(k)$

Comparaisons

Pour initialiser les individus, 100 cellules sont utilisées afin de décrire la distribution de matières de chacun. Ces 100 cellules représentent 300 variables d'optimisation. La taille de la population de l'algorithme génétique étant généralement choisie a priori proportionnellement au nombre de variables d'optimisation utilisées pour décrire une solution [62, 63], nous choisissons d'utiliser au départ 300 individus. Ce choix est fait afin de couvrir l'espace des solutions correctement au début du processus. Les paramètres de la loi d'évolution de la population sont choisis pour obtenir 300 individus pour $k = 1$ et 4 individus pour $k = 300$. Ces conditions nous donne le jeu de paramètres suivant : $(a = 3.04; b = 1212)$.

La Figure 5.2 permet de comparer pour 5 essais les résultats obtenus avec une population constante et ceux obtenus avec la loi de décroissance proposée, avec un nombre maximal d'évaluation fixé à $N_{ev,max} = 21000$.

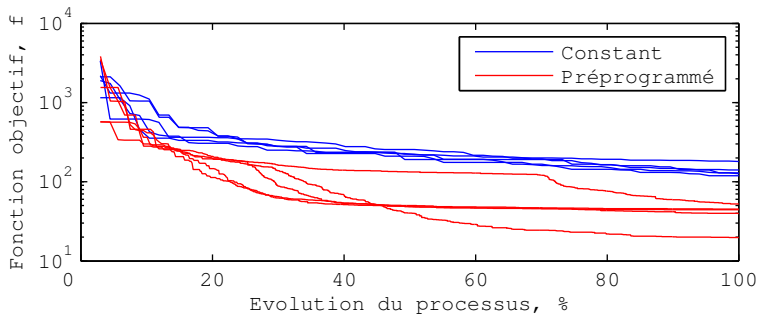

Figure 5.2 – Comparaison des courbes de convergence obtenues pour le cas $N_s = C^{st}$ et pour le cas $N_s = N_s(k)$ – (21000 évaluations ; 300 individus initiaux)

On remarque sur le graphique qu'à partir d'une même population initiale, on obtient de meilleurs résultats avec la loi d'évolution de la population proposée, bien que la variabilité entre les individus y soit plus marquée. La cause de ce problème est peut être le fait que l'on a trop rapidement diminué la population avant qu'elle ne soit en condition suffisante pour l'étape d'exploitation. Cependant, il est difficile de prévoir le moment du passage de l'étape d'exploration à l'étape d'exploitation sans avoir préalablement fait quelques essais.

Les résultats des essais sont résumés dans le Tableau 5.1. La variance de ceux-ci, pour le cas de la taille variable de population, est plus élevée

que pour le cas de la taille constante. Les gains obtenus sont largement favorables à l'utilisation d'une méthode variant la taille de la population.

Table 5.1 – Résumé des résultats obtenus avec 5 essais : $N_s = C^{st}$ et $N_s = N_s(k)$

Fonct. Objectif	$N_s = 300$	$N_s = N_s(k)\|_{a=3.04;b=1212}$
Minimum	119.57	19.83 (-83.4%)
Médiane	129.33	44.34 (-65.7%)
Maximum	180.55	52.38 (-70.9%)

Les résultats graphiques sont représentés par la Figure 5.3. On observe de grandes différences entre la méthode $N_s = N_s(k)$ et la méthode $N_s = 300$.

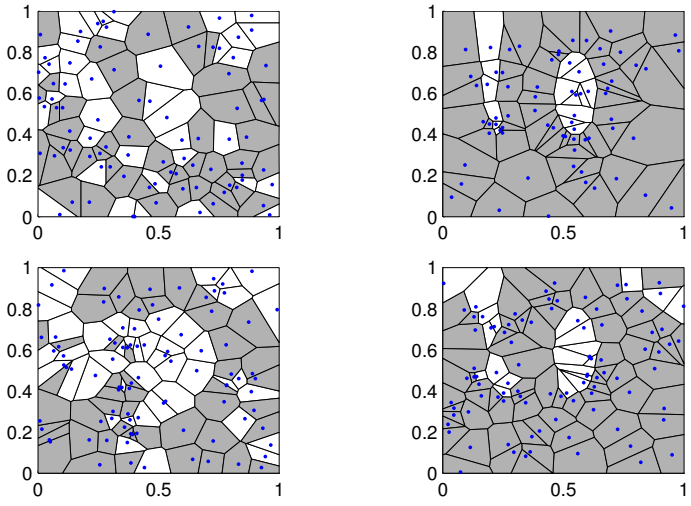

Figure 5.3 – Représentations des meilleures solutions (haut) et des moins bonnes (bas) pour la stratégie avec une taille de population constante $N_s = 300$ (gauche) et avec une taille variable $N_s = N_s(k)$ (droite) – (20.000 évaluations)

Le choix d'utiliser durant tout le processus une population de 300 individus n'a pas aidé à fournir de bons résultats du fait que le nombre d'évaluations autorisé est atteint après un petit nombre d'itération. Une réduction progressive de la taille de la population permet à l'outil de

réaliser un plus grand nombre d'itérations et donc de mener une phase d'exploitation avant d'avoir atteint le nombre d'évaluations autorisées et ainsi améliorer la forme du meilleur individu de la population.

Discussion

Bien que les résultats obtenus soient encourageants, il ne faut pas perdre de vue que cette méthode n'est fonctionnelle qu'avec un choix judicieux des paramètres de contrôle visant à modifier l'allure de la loi d'évolution de la taille de la population.

Le graphique de la Figure 5.4 montre un cas où l'on a relaxé la loi d'évolution de la taille de la population ($a = 1.25; b = 4050$) afin que celle-ci n'arrive à 4 individus qu'après 900 itérations. Les courbes de convergence montrent qu'une plus longue période d'exploration diminue la variance dans les résultats. Mais ce coût supplémentaire en évaluations réduit la phase d'exploitation si l'on conserve constant le nombre d'évaluations durant le processus d'optimisation. Les individus finaux sont donc moins variables mais aussi moins performants pour les meilleurs d'entre-eux.

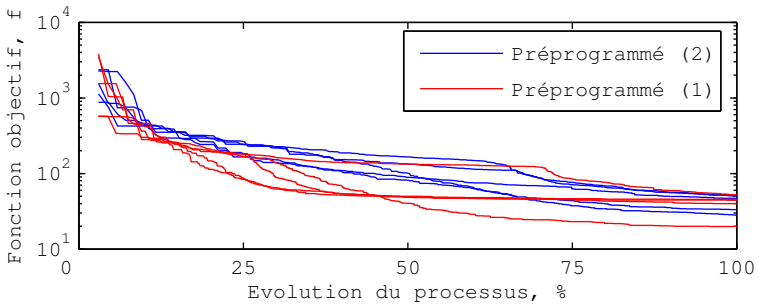

Figure 5.4 – Comparaison de convergences pour l'évolution variable de la taille de population avec le jeu de paramètres 1 ($a = 3.04; b = 1212$) et avec le jeu de paramètres 2 ($a = 12.5; b = 4050$)

5.1.2 Régulation par le rendement de chaque génération : $N_s = N_s(N_b)$

La méthode de régulation proposée repose entièrement sur le rendement de chaque génération afin de déterminer le coefficient correcteur adaptant la taille de la population aux besoins de l'outil.

Description

A chaque génération, un nombre variable N_b d'individus enfants ont une meilleure performance que le meilleur individu de la population principale. On dit que ces individus améliorent la population. Le rendement d'une génération peut donc être défini comme étant le ratio du nombre d'individus améliorant la population, sur le nombre total d'individus dans la population :

$$\eta = \frac{N_b}{N_s} \tag{5.3}$$

Si le rendement d'une génération est élevé, le nombre d'individus N_s de cette génération capables de faire évoluer positivement la population à la génération suivante est important, il n'est donc pas nécessaire de garder une population grande puisqu'un petit nombre d'individu suffit à la faire évoluer. Inversement, si le rendement est faible, le nombre d'individus capables de faire évoluer positivement la population est faible et il faut augmenter la taille de la population pour permettre une meilleure exploration localement autour du meilleur individu.

Deux situations sont possibles. Soit $N_b > 0$, on adopte alors la loi d'évolution suivante :

$$N_s^{(k+1)} = \frac{N_s^{(k)}}{N_b^{(k)}} \tag{5.4}$$

ce qui implique une diminution de la taille de la population principale, soit $N_b = 0$, alors on adopte la loi d'évolution suivante :

$$N_s^{(k+1)} = \rho N_s^{(k)} \tag{5.5}$$

où $\rho > 1$ est le coefficient d'augmentation défini par l'utilisateur.

Résultats

Des tests ont été effectués avec une population initiale composée de 300 individus avec un maximum de 21 000 évaluations. Les résultats de ces tests sont comparés au cas de référence sur base des courbes de convergence (Figure 5.5). On y observe une amélioration des résultats avec une régulation de la population.

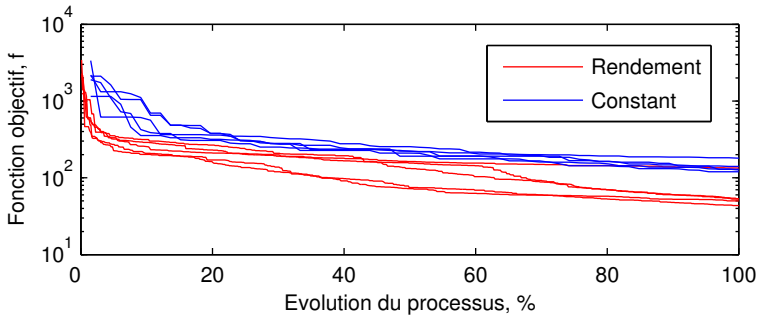

Figure 5.5 – Comparaison des courbes de convergence obtenues pour le cas $N_s = C^{st}$ constant et pour le cas $N_s = N_s\,(N_b)$ – (21 000 évaluations ; 300 individus initiaux)

Le Tableau 5.2 résume les résultats des différents essais. La différence de performance est significative pour la méthode $N_s = N_s\,(N_b)$ avec un gain de plus de 60% pour le minimum et la médiane et qui tombe à environ 25% pour le maximum.

Table 5.2 – Résumé des résultats obtenus avec 5 essais pour chaque méthode : $N_s = C^{st}$ et $N_s = N_s\,(N_b)$

Fonct. Objectif	$N_s = 300$	$N_s = N_s\,(N_b)$
Minimum	119.57	43.54 (-63.5%)
Médiane	129.33	52.69 (-59.2%)
Maximum	180.55	133.35 (-26.1%)

Toutefois, la variance des résultats reste élevée pour cette nouvelle approche. La méthode appauvrit la population en diminuant sa diversité génétique. La taille de la population (Figure 5.6) est fortement réduite dès le début de l'optimisation. Après seulement une dizaine d'ité-

rations, la population n'est composée que de quelques individus. Sur le graphique, on observe que tout au long de l'optimisation, la taille de la population subit quelques légères variations autour de sa valeur minimale. Et finalement en fin d'optimisation, la taille de la population connaît des variations importantes. Ceci veut dire qu'il est de plus en plus difficile de trouver des enfants améliorant la population. C'est le signe que l'outil converge vers une solution et rencontre dès lors de plus en plus de difficultés pour faire évoluer la population.

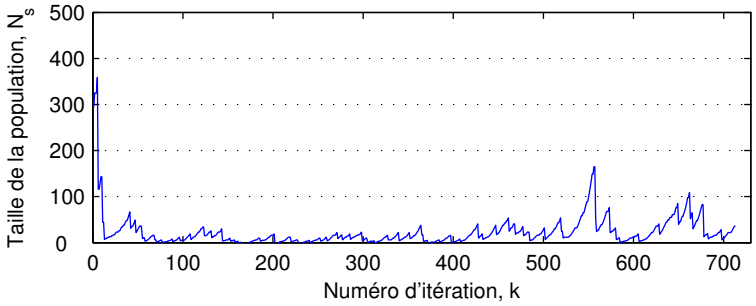

Figure 5.6 – Exemple d'évolution de la taille de population obtenue pour le meilleur des cinq essais avec $N_s = N_s\,(N_b)$

Discussion

La méthode présentée ne maintient pas suffisamment longtemps la taille de la population en début de processus d'optimisation. Elle ne tient pas compte du besoin de diversité pour améliorer l'exploration suffisamment longtemps de l'espace des solutions, et donc du besoin de conserver une taille de population élevée. Une population de grande taille permet à l'outil de conserver un plus grand nombre d'individus différents et de faire évoluer un plus grand nombre de direction de recherche. En limitant trop vite la taille de la population en début de processus d'optimisation, on augmente les chances de converger vers une solution non optimale.

5.1.3 Régulation par attribution d'un coefficient de reproduction : $N_{se} = N_{se}(\vec{c_r})$

Chaque individu possède un potentiel propre pour créer de meilleurs individus. Ce potentiel est quantifié et utilisé pour déterminer le taux de leur participation lors de la création de la population des enfants. Si l'ensemble des parents a des facilités pour proposer des enfants améliorant la population, la taille de la population des enfants sera augmentée pour permettre d'étendre l'exploration de l'espace des solutions. Au contraire, si les parents montrent des difficultés à proposer des enfants améliorant la population, le choix est de réduire la taille de la population des enfants afin de permettre un plus grand nombre de générations et laisser plus de temps à la phase d'exploitation.

Description

Un coefficient de reproduction, $c_{r,i} \in \mathbb{N}$, est calculé à chaque itération pour chaque individu afin d'estimer son potentiel de reproduction. Au plus un parent est efficace pour créer de bons enfants, au plus ce coefficient lui donne la possibilité de créer de nouveaux enfants. Et inversement, au moins les enfants d'un parent sont efficaces, au moins il pourra se reproduire.

Lors du croisement entre deux parents, la somme de leurs coefficients de reproduction détermine le nombre de leurs enfants. Ce mécanisme permet de supprimer l'étape de sélection de parents puisque c'est au travers du coefficient de reproduction que les parents vont pouvoir se reproduire ou non. Chaque individu dans la population principale est utilisé une fois dans la population de parents.

Les couples de parents sont créés par une sélection aléatoirement basée sur leur classement, r_i, dans la population par ordre décroissant de performance. La probabilité p_i de sélectionner un parent est calculée suivant :

$$p_i = 100 \frac{t_i}{\sum_{j=1}^{N_p} t_j} \% \tag{5.6}$$

$$t_i = \frac{1}{r_i} \tag{5.7}$$

Puisque l'on additionne les coefficients des deux parents pour déterminer le nombre d'enfants, la taille totale de la population, N_{se}, des enfants est la somme de tous les coefficients de reproduction des parents :

$$N_{se} = N_{se}\left(\vec{c_r}\right) = \sum c_{r,i} \text{ avec } i \in 1, 2, \ldots, N_s \tag{5.8}$$

Pour améliorer la population principale, un enfant doit être plus performant que 50% des individus de celle-ci. Pour chaque enfant d'un couple de parents améliorant la population, les coefficients de reproduction des deux parents sont incrémentés de un. Inversement, lorsque aucun des enfants d'un couple de parents ne parvient à améliorer la population sur une période de deux générations consécutives, les coefficients de reproduction des parents sont décrémentés de un.

La valeur du coefficient de reproduction de chaque individu est soumise à deux règles. Premièrement, le coefficient de reproduction du meilleur individu a une valeur minimale de 2. Secondement, chaque nouvel individu a un coefficient de reproduction initiale fixé à 1.

Résultats

Des tests ont été effectués pour des populations initiales de 10 et de 100 individus évoluant avec un maximum de 20000 évaluations [64]. Le choix de ces deux tailles de population est important. Nous savons par expérience sur le problème étudié qu'une population de 10 individus donne de meilleurs résultats qu'une population de 100 individus. Les graphiques de la Figure 5.7 montrent une évolution positive des résultats.

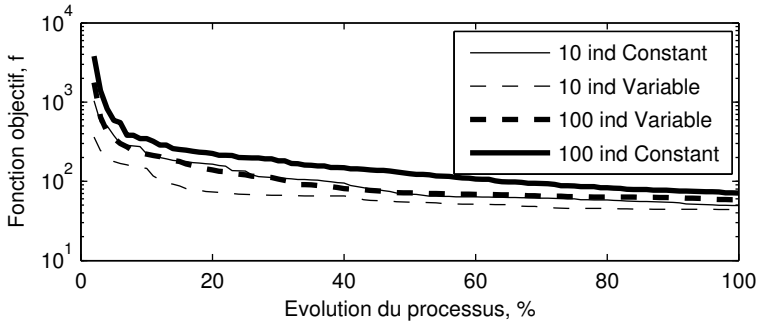

Figure 5.7 – Comparaison de courbes de convergence médiane obtenues pour la méthode $N_s = C^{st}$ et la méthode $N_s = N_s\left(\vec{c_r}\right)$ – (20000 évaluations ; 10 et 100 individus)

Bien que les résultats sont toujours dépendants du choix de la taille de la population, ceux-ci sont presque équivalents pour une population de taille constante avec 10 individus ou avec une population de taille variable avec 100 individus au départ.

Plus de détails des résultats sont donnés dans le Tableau 5.3 où l'on compare les quatre configurations.

Table 5.3 – Résumé des résultats obtenus avec 5 essais pour chaque test : avec la méthode $N_s = C^{st}$ et la méthode $N_s = N_s\,(\vec{c_r})$

Fonct. Objectif	Constant	Variable	Constant	Variable
N_{se}	10	10	100	100
Minimum	34	39 (+14%)	62	41 (-33.9%)
Médiane	49	44 (-10.3%)	71	57 (-19.8%)
Maximum	115	60 (-47.8%)	117	127 (+8.5%)

Avec le mauvais choix de taille de population initiale, $N_s = 100$ dans notre cas, l'outil arrive à atteindre des valeurs proches des résultats obtenus à N_s constant avec un meilleur choix de la taille de la population, $N_s = 10$ dans notre cas. L'outil a donc été capable d'adapter la taille de la population pour augmenter les chances d'évolution d'une population.

Bien qu'il y ait des variations de la taille de la population des enfants tout au long du processus d'optimisation (Figure 5.8), la stratégie proposée en fonction du mérite des différents individus de la population principale tend à faire évoluer cette taille jusqu'à une faible valeur, ce qui est favorable à l'amélioration des résultats grâce aux gains du nombre d'itérations.

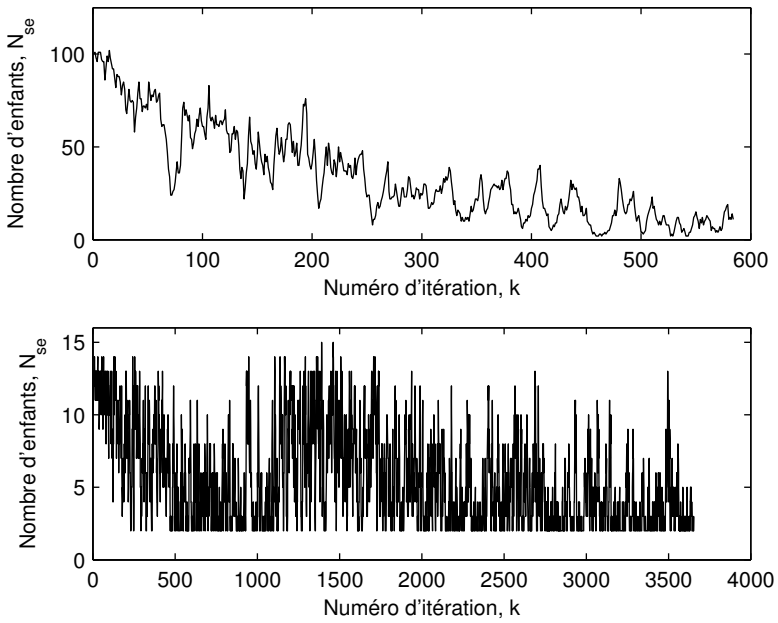

Figure 5.8 – Exemple d'évolution de la taille de population des enfants obtenue pour le meilleur des cinq essais – (20000 évaluations ; 100 individus (haut) et 10 individus (bas))

Discussion

La stratégie proposée ne concerne pas la taille de la population principale mais bien celle de la population des enfants. C'est cette dernière qui détermine le nombre d'évaluations nécessaires par génération. En modifiant la taille de cette population, l'outil est donc capable de varier le nombre d'itérations. Toutefois, l'étape de reproduction reste directement influencée par la taille de la population principale. Avec une petite population principale, les meilleurs individus ont plus de chance de se reproduire entre-eux. En reprenant le calcul de la probabilité de sélection (5.6) pour créer des paires de parents, avec $N_s = 10$, la probabilité que le meilleur individu se reproduise avec le second est de 18% alors qu'elle n'est que de 5.6% pour $N_s = 100$.

En suivant ce raisonnement et en considérant les dix premiers individus équivalents pour $N_s = 10$ et pour $N_s = 100$, le meilleur parent

n'a qu'environ 54% de chance de ne sélectionner que les neuf meilleurs solutions suivantes dans le cas de $N_s = 100$. L'outil perd donc du temps pour l'évaluation des enfants provenant de couples de parents ayant potentiellement moins de chances d'améliorer la population.

5.2 Initialisation de la population

Comme on le sait maintenant, au début d'un processus d'optimisation, l'algorithme génétique s'appuie principalement sur l'utilisation de l'opérateur de croisement afin de brasser le patrimoine génétique (c'est-à-dire les vecteurs de variables de design) de la population, pour en tirer le meilleur parti. Ce brassage est d'autant plus efficace si la variété du patrimoine est grande et augmente ainsi les chances d'exploration. La notion de variété y est donc très importante. Chaque individu, aussi différent soit-il, apporte un potentiel d'évolution de la population entière à court ou à long terme. Il est reconnu que, lors de l'optimisation d'un problème difficile, il faut conserver une grande diversité le plus longtemps possible. Ceci permet d'accroître l'efficacité de l'outil au cours de l'exploration de l'espace des solutions. Au cours de la phase suivante, la phase d'exploitation, chaque individu tend à ressembler aux autres, limitant ainsi l'exploration, et à n'apporter qu'une légère variation par rapport à la meilleure solution.

Cette courte introduction nous rappelle finalement l'intérêt de travailler sur l'initialisation du processus d'optimisation, source de diversité dans la population. Il est nécessaire de maximiser la diversité de cette population initiale afin d'augmenter au maximum non seulement la richesse du patrimoine mais aussi pour couvrir un plus grand espace dans l'espace des solutions avec le même nombre d'individus.

Plusieurs approches sont proposées pour donner le meilleur départ possible à l'outil d'optimisation. En section 5.2.1, on étudie les conséquences de l'initialisation de la population par anticipation. En section 5.2.2, on essaie de maximiser la zone couverte par la population dans l'espace des solutions. Finalement en section 5.2.3, on fait intervenir le changement des dimensions entre l'espace de design et l'espace réel.

5.2.1 Prédéfinir la population initiale par anticipation

L'idée de cette première méthode est d'inclure dans la population initiale une ou des solutions supposées a priori topologiquement proches de l'optimum. Cependant, ceci implique une bonne connaissance du problème de la part du concepteur. Son choix va biaiser l'étape d'exploration de l'algorithme. Ceci implique aussi un risque de bloquer la population dans un optimum local si le concepteur choisit une mauvaise topologie initiale. Et, finalement, il est difficile d'appliquer cette approche pour des problèmes dont la topologie optimale est peu connue.

Pour notre cas d'application, puisqu'un individu de meilleur performance en début d'optimisation va propager rapidement son patrimoine aux autres individus, nous choisissons d'initialiser directement tous les individus de la population avec la même topologie composée d'un seul domaine composé de fer. Cette situation représente le cas où il n'y a pas de défaut présent. Le soin de la recherche de ces défauts est laissé à l'outil d'optimisation. Celui-ci va ajouter des domaines d'absence de fer dans l'espace de design pour tester différents défauts afin de retrouver ceux du barreau de référence.

Les courbes de convergence de la Figure 5.9, obtenues au terme de 50 000 évaluations, confirment l'existence de risques lors de l'anticipation de la solution optimale.

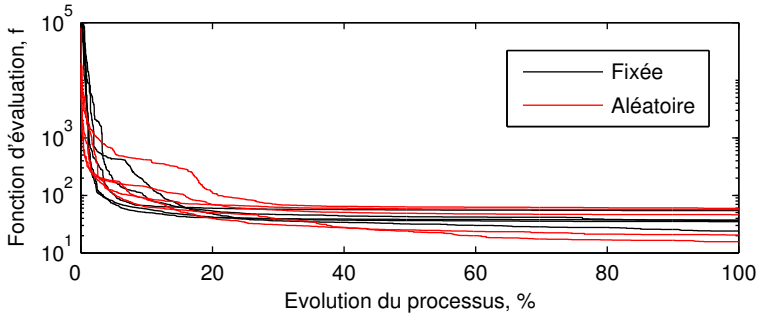

Figure 5.9 – Comparaison des courbes de convergence obtenues avec une population initiale non-aléatoire et avec une population initiale aléatoire – (50 000 évaluations ; 10 individus)

La méthode avec une population initiale fixée évolue plus rapidement dès le début du processus d'optimisation, mais perd ensuite son avance

en faveur de la méthode avec une population initiale aléatoire. Celle-ci met plus de temps pour évoluer mais donne les meilleurs résultats.

Les résultats graphiques (Figure 5.10) montrent finalement une très faible différence entre les deux méthodes d'initialisation. Les deux méthodes convergent vers la même topologie. La seule différence entre les deux méthodes réside dans la précision de description de la forme en faveur de la méthode d'initialisation aléatoire de la population.

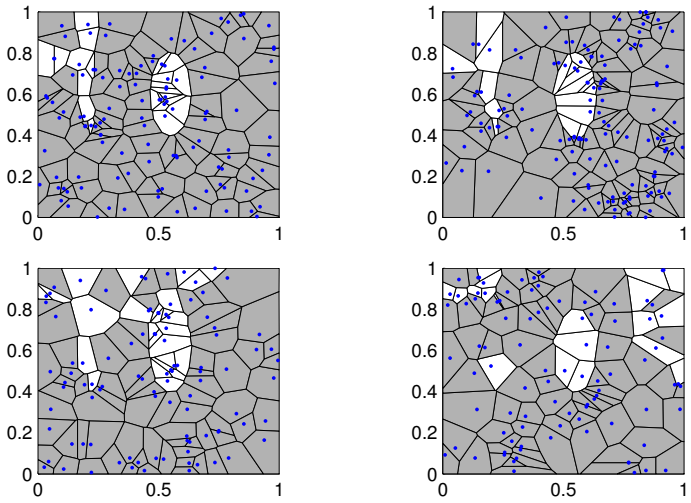

Figure 5.10 – Représentations de l'individu le plus performant (haut) et du moins performant (bas) pour la population initiale aléatoire (gauche) et la population initiale non aléatoire (droite) avec 50 000 évaluations

Bien que la meilleure solution soit obtenue avec la méthode aléatoire, la variance sur les résultats y est aussi plus élevée (Table 5.4).

Table 5.4 – Résumé des résultats obtenus avec 5 essais pour chaque test : population initiale non-aléatoire et population initiale aléatoire.

	Non-aléatoire	Aléatoire
Minimum	23.94	15.61 (-34.79%)
Médiane	36.73	46.11 (+25.53%)
Maximum	54.17	59.32 (+10.95%)

L'intérêt d'utiliser une population diversifiée se retrouve lorsqu'on dispose d'un plus grand nombre d'évaluations à disposition puisque la méthode a besoin d'une période d'exploration plus longue au départ de l'optimisation. Pour un plus petit nombre d'évaluations, une topologie fixée dès le départ permet d'atteindre des résultats rapidement, avec toutefois le risque de biaiser la solution finale. Il n'est évidemment pas facile d'estimer à quelle occasion l'une ou l'autre approche serait plus avantageuse. Cependant, l'utilisation d'un outil d'optimisation topologique est généralement utilisé pour rechercher l'optimum au problème étudié. La recherche d'une diversification maximale de la population initiale est donc plus que conseillée.

5.2.2 Recouvrement équilibré de l'espace de design par les matériaux

Afin de maximiser l'efficacité de la phase d'exploration de l'algorithme génétique, il est souhaitable de maximiser la diversité de la population initiale en terme de répartition des matériaux sur l'espace de design. On peut maximiser la diversité de matériau pour chaque point de matière au sein de l'espace de design en offrant une chance équitable à chacun des matériaux d'être attribués à chacun des points de matières, ce que l'on appelle le recouvrement équilibré. Effectivement, lors de l'initialisation d'une population, c'est-à-dire lors de l'attribution de valeurs aléatoires aux variables d'optimisation, il est possible d'obtenir une distribution de matière non équitablement répartie. Pour assurer une répartition équitablement répartie, il est nécessaire que la relation suivante soit vérifiée :

$$\bar{x}_i = \frac{\sum_{j=1}^{N_s} x_{j,i}}{N_s} = \frac{\sum_{j=1}^{N_d} j}{N_d} \tag{5.9}$$

où N_d est le nombre de matériaux utilisés.

Pour qu'une population regroupe toutes les combinaisons possibles, sa taille doit être égale à $N_d^{N_c}$. Parmi tous les individus se trouve l'optimum. Comme ce nombre est fort élevé, il est impossible de générer et d'évaluer toutes les possibilités. Pour une taille limité de la population, on peut décomposer le vecteur de variables en N_{sv} sous-vecteurs et associer la même valeur à chaque sous-vecteur. En changent les indices du vecteur pour regrouper par sous-vecteur, nous avons :

$$\vec{x} = \left\{ \underbrace{x_1, \ldots, x_{s_1}}_{sous-vecteur\ 1}, \underbrace{x_{s_1+1}, \ldots, x_{s_2}}_{sous-vecteur\ 2}, \underbrace{x_{s_{Nsv-1}+1}, \ldots, x_{s_{Nsv}}}_{sous-vecteur\ N_{sv}} \right\} \tag{5.10}$$

ce qui donne :

$$\vec{x} = \{sv_1, sv_2, \ldots, sv_{Nsv}\} \tag{5.11}$$

Par exemple, pour 100 cellules et 2 matériaux différents, il existe au total 2^{100} combinaisons. Pour une population proche de 15 individus, on peut exploiter 4 sous-vecteurs de 25 variables d'optimisation, puisque $2^4 = 16$ est proche de $N_s = 15$. Lors de l'initialisation des 16 individus, une des 16 combinaisons possibles est attribuée à chaque individu.

Les résultats obtenus sont illustrés par le graphique de la Figure 5.11. Au début de l'optimisation, les courbes montrent que la version sans recouvrement équilibré a une variance plus élevée. Au cours du processus d'optimisation, la version sans recouvrement récupère son retard. Et finalement, en fin d'optimisation, la version avec recouvrement équilibré obtient la plus performante mais aussi la moins performante des solutions.

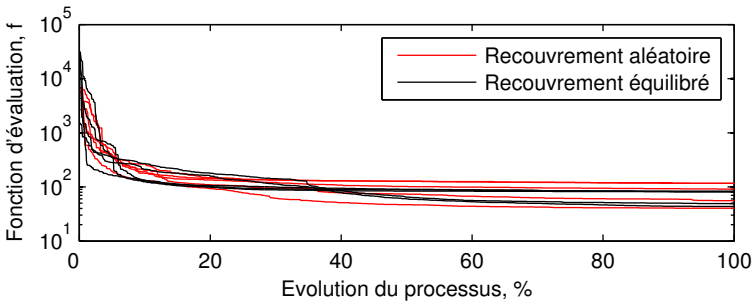

Figure 5.11 – Comparaison des courbes de convergence obtenues avec une population initiale non-aléatoire et avec une population initiale aléatoire (20 000 évaluations - 16 individus)

Les valeurs des minima, maxima et médianes obtenus sur l'ensemble des essais repris dans le Tableau 5.5. Ceux-ci montrent que la moins

bonne performance de la version avec recouvrement équilibré est largement moins efficace que celle avec recouvrement aléatoire. Travailler en maximisant le recouvrement de l'espace des solutions en bridant le formalisme de Voronoï en début d'optimisation ne permet pas d'obtenir une diversité suffisante de discrétisation de l'espace de design.

Table 5.5 – Résumé des résultats obtenus avec 5 essais pour chaque test : population initiale avec recouvrement aléatoire et population initiale avec recouvrement équilibré.

	Avec recouvrement aléatoire	Avec recouvrement équilibré
Min.	43.5	40.1 (-7.8%)
Méd.	80.5	91.3 (+13.4%)
Max.	83.2	116.5 (+40%)

5.2.3 Influence des dimensions de l'espace de design

L'algorithme d'optimisation est configuré de telle manière que les variables d'optimisation liées à la position des centres de voronoï évoluent au sein d'un espace confiné normalisé, $\vec{x}_C \in [0,1]^{(N_p - N_D)}$. Ceci implique une différence entre l'espace de design propre à l'évolution des variables d'optimisation et l'espace de design réel entrainant une déformation des formes et des dimensions. En effet, l'espace de design réel possède une dimension de type $[0;a] \times [0;b]$ où a et b sont la hauteur et la longueur définis à l'avance. Les cellules de matière subissent la même déformation pour s'adapter aux dimensions de l'espace de design réel.

Pour obtenir une distribution homogène des centres de Voronoï dans l'espace de design réel, il faut une distribution non-homogène des centres de Voronoï dans l'espace de design virtuel. L'espace de design réel est décomposé en un quadrillage de $n_H \times n_L$ sous-domaines carrés (Figure 5.12). Le nombre de sous-domaines dépend du nombre de cellules ainsi que des dimensions de l'espace de design réel. On initialise la distribution de matière avec N'_c cellules dans chacun de ces sous-domaines, N'_c étant un nombre à définir. Pour le cas du problème simple de type physique, les dimensions de l'espace de design ont un rapport de 6 ($n_H = 0.05, n_L = 0.3$). Nous obtenons $n_L = 6n_H$, ce qui fait $6n_H^2$ sous-domaines. Pour 100 cellules et en fixant $N'_c = 1$, nous obtenons $n_H = 4.08$. En arrondissant

la valeur à 4, le nombre total de cellules dans le domaine de design vaut 96.

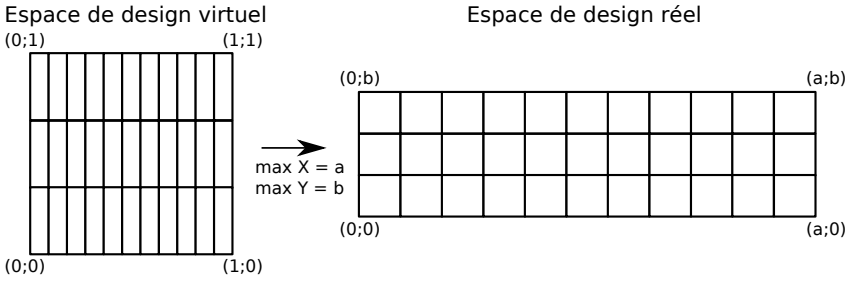

Figure 5.12 – Exemple de transformation entre l'espace de design virtuel et l'espace de design réel avec $n_H = 3$ et $n_L = 11$

Les résultats obtenus sont illustrés par le graphique de la Figure 5.13. Bien que les courbes de convergence soient proches pour les deux approches, à partir des 40 derniers pourcents du processus d'optimisation, il y a une légère amélioration des résultats en faveur de l'approche avec correction de la distribution des cellules de Voronoï.

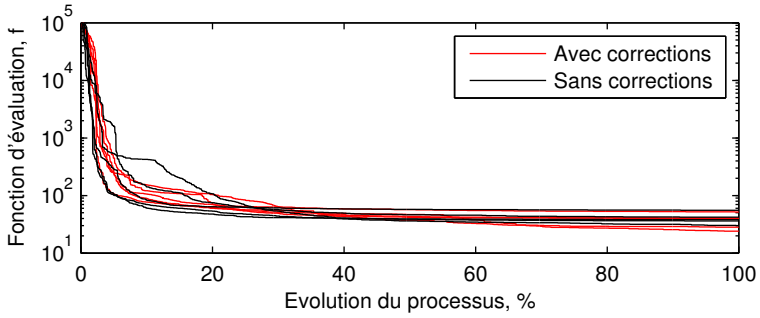

Figure 5.13 – Comparaison des courbes de convergence obtenues avec et sans correction des dimensions lors de l'initialisation – (30000 évaluations ; 10 individus)

Le Tableau 5.6 résume les valeurs obtenus pour chaque méthode. On constate un gain important sur la valeur minimale et très peu de variation pour la médiane et la valeur maximale.

Table 5.6 – Résumé des résultats obtenus avec 5 essais pour chaque test : population initiale avec recouvrement aléatoire et population initiale avec recouvrement équilibré.

	Sans corrections	Avec corrections
Minimum	30.10	23.93 (-20.49%)
Médiane	38.08	40.16 (+5.46%)
Maximum	55.42	51.43 (-7.20%)

5.3 Synthèse

Au travers de ce chapitre, nous avons étudié deux approches de la gestion de la population, c'est-à-dire la gestion de la taille de la population en cours de processus d'optimisation et la gestion de l'initialisation de la population.

La gestion de la taille de la population permet de partir d'une taille de population importante en début de processus d'optimisation afin de permettre une exploration satisfaisante de l'espace des solutions puis de réduire la taille de la population lorsque la topologie se précise et qu'il convient de passer à une phase d'exploitation pour fixer la géométrie et les dimensions de la solution optimale puisque la réduction de la taille permet, à nombre total d'évaluations fixé, d'augmenter le nombre d'itérations. Trois méthodes de gestion de la taille de la population ont été étudiées. Comme le montre la Table 5.7 qui reprend les résultats obtenus avec les trois méthodes :

- la décroissance préprogrammée (Section 5.1.1) permet d'obtenir les meilleurs résultats mais ses performances sont fortement liées à un choix judicieux des paramètres utilisés pour configurer la courbe d'évolution de la taille de la population. Par ailleurs, elle ne tient pas compte de la performance des individus pour agir sur la loi d'évolution ;
- la régulation par le rendement de chaque génération (Section 5.1.2) qui prend en compte le nombre d'individus qui à chaque génération améliore la population, présente le défaut de faire décroître trop rapidement la population en début d'optimisation et donc de ne pas tenir compte suffisamment longtemps du besoin de diversité nécessaire pour l'exploration de l'espace des solutions. Elle

augmente le risque de converger vers une solution non optimale comme le traduit la valeur maximale atteinte par la fonction objectif. Elle autorise également une croissance de la taille de la population en fin de processus lorsqu'il devient difficile de trouver un individu améliorant la population.
– la régulation par attribution d'un coefficient de reproduction (Section 5.1.3) qui utilise les performances des enfants pour déterminer le taux de reproduction des parents donne de bons résultats sans toutefois atteindre les performances d'une décroissance programmée lorsque les paramètres qui fixent la loi de décroissance sont bien choisis.

Table 5.7 – Résumé des résultats obtenus pour la gestion de la taille de la population

	$N_s\,(k)$	$N_s\,(N_b)$	$N_s\,(\vec{c_r})$
Minimum	19.83	43.54	41
Médiane	44.34	52.69	57
Maximum	52.38	133.35	127

La gestion de l'initialisation de la population permet de générer une population de manière plus adaptée. Trois méthodes différentes, dont les résultats sont résumés dans la Table 5.8, sont proposées :
– l'anticipation de la solution finale (Section 5.2.1) qui détermine la topologie initiale des individus de la population dépend fortement de l'expérience de l'utilisateur. Les résultats obtenus pour cette méthode montrent qu'un choix de la topologie initiale, même cohérent avec le problème, peut ne pas garantir un gain de performance significatif voire même biaiser l'évolution du processus. La médiane et le maximum augmentent fortement et le minimum diminue ;
– le recouvrement équilibré (Section 5.2.2) qui permet une distribution équitable de la matière dans l'ensemble de la population, n'obtient pas toujours de bons résultats et augmente considérablement le maximum obtenu ;
– la prise en compte de l'influence de la différence entre les dimensions de l'espace de design réel et l'espace de design virtuel (Section 5.2.3) donne de bons résultats en diminuant à la fois le minimum

et le maximum obtenu tout en ayant une faible augmentation pour la médiane.

Table 5.8 – Résumé des résultats obtenus pour la gestion de l'initialisation de la population

	Anticipation	Recouvrement équilibré	Projection
Minimum	15.61	40.1	23.93
Médiane	46.11	91.3	40.16
Maximum	59.32	116.5	51.43

Chapitre 6

Adaptation de la reproduction

Ce chapitre porte sur l'étude de différentes adaptations des mécanismes de reproduction de l'algorithme génétique. Ces mécanismes de reproduction sont liés à une étape importante dans le processus d'optimisation. C'est le moment où les croisements et les mutations vont permettre de créer de nouvelles solutions à partir de la population principale. Cette étape est décomposée en deux sous-étapes qui sont étudiées indépendamment : le croisement et la mutation.

Dans la section 6.1, différentes méthodes de croisements, graphiques et non graphiques, sont comparées et étudiées. La section 6.2 concerne l'étude des mutations. Les mutateurs proposés et étudiés se focalisent sur la mutation des centres des cellules de Voronoï ainsi que sur la variation du nombre de cellules de Voronoï pour décrire une solution. La section 6.3 présente une synthèse de l'étude effectuée.

Tous les résultats présentés sont obtenus en appliquant les méthodes de croisement et de mutation étudiées au problème physique simple décrit à la section 3.2.2 du Chapitre 3.

6.1 Croisements

L'opérateur de croisement de l'algorithme génétique est un générateur dont la vocation est de créer de nouveaux individus à partir d'un ensemble d'individus que constitue la population. De manière générale,

les croisements sont réalisés sur base de pivots graphique et non graphique. Un pivot graphique est une référence géométrique qui permet de subdiviser l'espace de design en plusieurs parties alors qu'un pivot non graphique est une référence indicielle qui permet de subdiviser le vecteur de variables d'optimisation en plusieurs sous-vecteurs.

C. Kane a comparé plusieurs méthodes de croisement [51] de type graphique (pivot linéaire et pivot par bloc) et non graphique (1-pivot, 2-pivots et uniforme) et les résultats obtenus montrent de meilleurs résultats pour les méthodes graphiques. En effet, les approches graphiques ont l'avantage de conserver la notion de voisinage entre les différentes cellules : on n'échange plus des variables d'optimisation, mais bien des blocs de matière.

Dans la même ligne de recherche, cette section présente et compare trois nouvelles méthodes graphiques ou non graphiques, adaptées au formalisme Voronoï, en plus du croisement graphique avec pivot linéaire proposé par C. Kane.

6.1.1 Description des opérateurs de croisements

Dans le cas de l'utilisation de cellules de Voronoï, la position des variables d'optimisation dans le vecteur de variables ne représente pas une grande importance. Ceci implique que l'échange d'informations provenant d'une cellule décrivant une partie de l'espace de design d'un premier individu avec des informations d'une autre cellule décrivant une autre partie de l'espace de design d'un second individu, ne donne pas toujours l'effet voulu.

La Figure 6.1 présente le cas d'un croisement uniforme de deux parents chacun décrits par trois cellules dont les variables d'optimisation sont reprises par les vecteurs de variables suivants :

$$\begin{array}{lcl} \text{Parent 1} & : & (x_1, x_2, x_3, y_1, y_2, y_3, m_1, m_2, m_3) \\ \text{Parent 2} & : & (x'_1, x'_2, x'_3, y'_1, y'_2, y'_3, m'_1, m'_2, m'_3) \end{array} \tag{6.1}$$

avec x_i, y_i les coordonnées des centres de Voronoï C_i et m_i le matériau de la cellules associée. La distribution de matière obtenue en croisant les variables d'optimisation x_1, y_2, x_3 et m_1 du premier parent avec les variables d'optimisation y'_1, x'_2, y'_3, m'_2 et m'_3 du deuxième parent ne possède aucun lien graphique avec les parents.

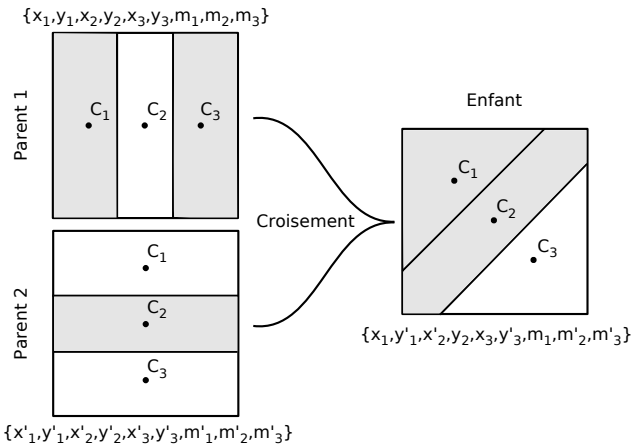

Figure 6.1 – Exemple de croisement uniforme appliqué au formalisme Voronoï

Pour éviter ceci, pour les quatre méthodes de croisement étudiées dans cette section, toutes les variables d'optimisation décrivant une cellule seront toujours transmises ensemble. Le vecteur de variables d'optimisation peut donc s'écrire comme ceci :

$$\vec{x} = \{c_1, c_2, \ldots, c_{N_c}\} \tag{6.2}$$

où c_i correspond à la cellule i.

Pour la suite de cette section, le croisement graphique par pivot linéaire proposé par C. Kane nous servira de référence. Nous proposons aussi deux nouvelles méthodes de croisement non graphiques inspirées du croisement uniforme [65], ainsi qu'une nouvelle méthode de croisement graphique inspirée de la méthode de croisement par bloc étudiée par C. Kane [51]. Ces quatre méthodes décrites par la suite sont :

- Le croisement non graphique équilibré ;
- Le croisement non graphique non-équilibré ;
- Le croisement graphique à pivot linéaire ;
- Le croisement graphique à pivot circulaire ;

6.1.1.1 Croisement non graphique équilibré

Le croisement non graphique équilibré attribue une probabilité de 50% à chaque cellule des deux parents d'être transmises lors du croisement. Chaque cellule subit un test de probabilité et est transmise si le

test est positif. Le croisement est donc dit équilibré puisque le test de transfert effectué est identique pour chacune des cellules. Ceci implique que le vecteurs de variables des parents et des enfants peuvent avoir une taille différentes.

En pratique, deux vecteurs masques, $\vec{m} \in \mathbb{1}^{N_c}$ et $\vec{m}' \in \mathbb{1}^{N'_c}$ avec N_c le nombre de cellules du parent 1 et N'_c celui du parent 2, sont créés en attribuant une valeur aléatoire à chaque élément selon une loi binomiale : $m_i \sim B(2, 0.5)$ et $m'_j \sim B(2, 0.5)$. Ensuite, les cellules sont ajoutées au nouvel individu en fonction de la valeur associée du masque :

$$\begin{aligned} Croisement : & \begin{cases} Ajouter(c_i) & \text{Si } m_i = 1 \\ Ajouter\left(c'_j\right) & \text{Si } m'_j = 1 \end{cases} \\ & \text{avec } i = 1, 2, \ldots, N_c \\ & \text{avec } j = 1, 2, \ldots, N'_c \end{aligned} \tag{6.3}$$

6.1.1.2 Croisement non graphique non-équilibré

Le croisement non-équilibré est défini pour profiter d'avantage des parents plus performants lors de la reproduction. Il reprend le même concept que le croisement équilibré mais la probabilité d'ajouter une cellule dépend cette fois-ci de la performance de chaque parent.

En pratique, les valeurs des éléments des deux vecteurs masques ne sont plus choisies en fonction d'une loi binomiale mais en fonction d'une probabilité p_j attribuée à chacun des deux parents :

$$\begin{aligned} m_i &= \begin{cases} 1 & \text{Si } r_i < p_1 \\ 0 & \text{Si } r_i \geq p_1 \end{cases} \\ m'_i &= \begin{cases} 1 & \text{Si } r'_i < p_2 \\ 0 & \text{Si } r'_i \geq p_2 \end{cases} \end{aligned} \tag{6.4}$$

où r_i et r'_i sont des valeurs aléatoires définies selon une loi uniforme continue : $r_i, r'_i \sim U_c(0, 1)$.

Si les parents sont renumérotés par réagencement des indices sur base des valeurs de leur fonction objectif f_i de façon à avoir $f_1 \leq f_2$, on choisit d'écrire la probabilité, p_j, associée à chaque individu par :

$$\begin{aligned} p_1 &= \frac{\ln(f_2)}{\sum_{i=1}^{2} \ln(f_i)} \\ p_2 &= \frac{\ln(f_1)}{\sum_{i=1}^{2} \ln(f_i)} \end{aligned} \tag{6.5}$$

avec $f_i > 0$ et où l'utilisation de le logarithme naturel permet de diminuer l'influence de la différence entre les performances de deux parents ayant de grandes valeurs de f_i.

6.1.1.3 Croisement graphique à pivot linéaire

Le croisement au moyen d'un pivot linéaire agit directement sur l'espace de design en utilisant pour chaque couple d'individus une droite choisie aléatoirement afin de subdiviser l'espace de design en deux parties. Tous les centres de Voronoï d'un des deux parents appartenant à une des deux parties sont copiés pour le nouvel individu et tous les centres de Voronoï de l'autre parent appartenant à l'autre partie sont ajoutés pour compléter le nouvel individu. Les modifications obtenues à partir de cette subdivision sont ensuite traduites dans le vecteur de variables d'optimisation.

La Figure 6.2 illustre un exemple de croisement graphique à pivot linéaire.

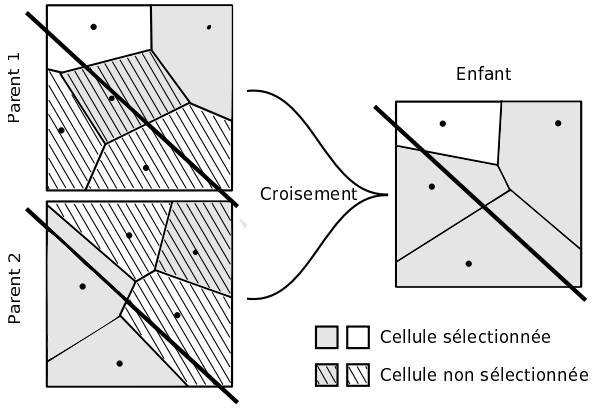

Figure 6.2 – Exemple de croisement graphique au moyen d'un pivot linéaire

Pour générer une droite $y = ax + b$ dans l'espace de design virtuel, deux points (x_A, y_A) et (x_B, y_B) servant à la définir sont choisis aléatoirement dans cet espace. La sélection des centres de Voronoï des deux parents dépend de leur position par rapport à la droite pivot :

$$m_i = \begin{cases} 1 & \text{Si } y_i > ax_i + b \\ 0 & \text{Si } y_i \leq ax_i + b \end{cases} \tag{6.6}$$

$$m'_j = \begin{cases} 1 & \text{Si } y'_j \leq ax'_j + b \\ 0 & \text{Si } y'_j > ax'_j + b \end{cases} \tag{6.7}$$

avec $i = 1, 2, \ldots, N_c$ et $j = 1, 2, \ldots, N'_c$.

6.1.1.4 Croisement graphique à pivot circulaire

Le mécanisme du croisement graphique avec un pivot circulaire est identique à celui du croisement avec un pivot linéaire [66], mais un avec cercle choisi aléatoirement pour jouer le rôle de pivot. La Figure 6.3 illustre un exemple de croisement graphique à pivot circulaire.

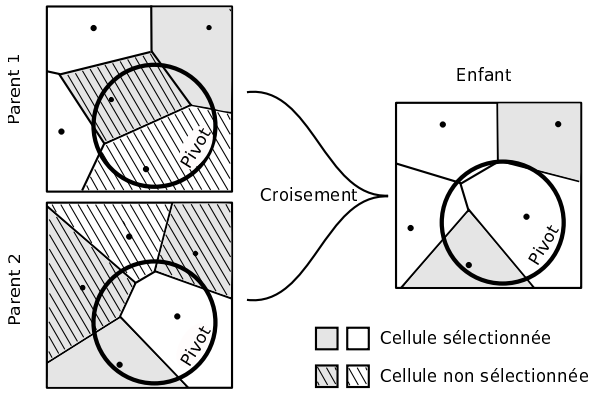

Figure 6.3 – Exemple de croisement graphique au moyen d'un pivot circulaire

La génération du cercle dans l'espace de design virtuel est obtenue par le choix aléatoire de trois variables (x_c, y_c, d_c) définissant respectivement la position du centre du cercle et le diamètre de ce dernier :

$$x_c, y_c, d_c \sim U_c\,(0, 1) \tag{6.8}$$

De même que pour le croisement graphique à pivot linéaire, la sélection des centres de Voronoï dépend de leur position par rapport au pivot :

$$m_i = \begin{cases} 1 & \text{Si } \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} > \frac{d_c}{2} \\ 0 & \text{Si } \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \leq \frac{d_c}{2} \end{cases} \tag{6.9}$$

$$m'_j = \begin{cases} 1 & \text{Si } \sqrt{\left(x'_j - x_c\right)^2 + \left(y'_j - y_c\right)^2} \leq \frac{d_c}{2} \\ 0 & \text{Si } \sqrt{\left(x'_j - x_c\right)^2 + \left(y'_j - y_c\right)^2} > \frac{d_c}{2} \end{cases} \tag{6.10}$$

avec $i = 1, 2, \ldots, N_c$ et $j = 1, 2, \ldots, N'_c$.

L'intérêt d'utiliser ce type de pivot est de favoriser l'échange de bloc de matière en se focalisant sur une zone d'un individu pour l'intégrer dans un autre individu.

6.1.2 Comparaisons

Les comparaisons mettent en jeu les quatre générateurs de croisements présentés ci-avant avec comme référence le croisement graphique par pivot linéaire puisque déjà étudié par C. Kane. Dans ces quatre générateurs, deux utilisent des opérations graphiques alors que les deux autres restent plus classiques en travaillant directement sur les variables d'optimisation.

Le graphique de la Figure 6.4 montre la courbe médiane obtenue avec 5 essais pour chaque méthode. Pour chacun des essais, seul le mécanisme de croisement est remplacé par ceux étudiés. Dans l'ordre de performance décroissant nous avons le croisement à pivot linéaire, le croisement non équilibré, le croisement équilibré et le croisement à pivot circulaire.

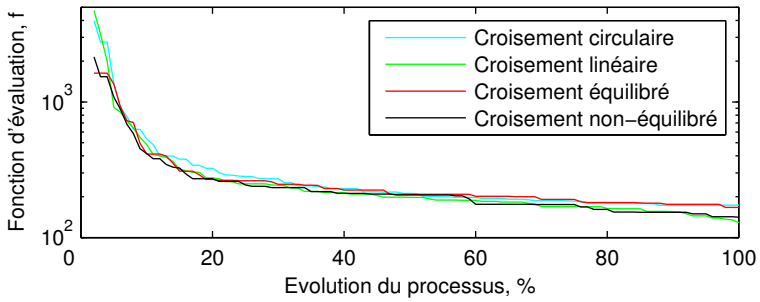

Figure 6.4 – Valeurs médianes obtenues pour les méthodes de croisements linéaire, circulaire, équilibrée et non équilibrée – (10000 évaluations ; 100 individus)

Un résumé des résultats est donné dans le Tableau 6.1 et montre que la méthode de croisement non équilibré, qui est une méthode élitiste, limite l'amplitude entre la performance du meilleur individu et du moins bon en diminuant la valeur maximale obtenue. Cette diminution est due à l'aspect élitiste de l'approche de la méthode. Les individus ayant une meilleure performance vont influencer plus rapidement l'évolution du processus d'optimisation. La méthode de croisement avec un pivot circulaire obtient des valeurs minimales assez faibles mais possède une grande différence de performance entre le meilleur individu et le moins bon.

Table 6.1 – Résumé des résultats obtenus avec 5 essais pour chaque test : croisement avec pivot linéaire, pivot circulaire, équilibré et non équilibré

	Droite	Cercle	Equilibré	Non équilibré
Min.	74.10	93.42 (+26%)	123.24 (+66%)	104.10 (+40%)
Méd.	128.92	172.55 (+33%)	167.01 (+29%)	141.05 (+9%)
Max.	161.62	184.98 (+14%)	184.31 (+14%)	142.24 (-12%)

6.1.3 Cohérence du croisement

Le meilleur résultat obtenu provient d'une méthode de croisement graphique. Celle-ci permet de conserver une logique dans la génération d'un nouvel individu. Avec une méthode non graphique, il est possible que cette génération ne soit pas cohérente.

Dans le cas d'un croisement équilibré entre deux parents[1], l'algorithme se focalise sur les vecteurs de variables d'optimisation. Observons ce qu'il se passe lorsque pour le premier vecteur nous sélectionnons la première cellule et la dernière cellule et sélectionnons les trois premières cellules pour le second vecteur :

$$\left\{ \underset{\triangle}{c_1}, c_2, c_3, c_4, \underset{\triangle}{c_5} \right\} \text{ et } \left\{ \underset{\triangle}{c'_1}, \underset{\triangle}{c'_2}, \underset{\triangle}{c'_3}, c'_4, c'_5 \right\} \tag{6.11}$$

où le symbole $\triangle$ indique que la cellule a été sélectionnée. Nous construi-

[1]Les conclusions sont identiques pour le croisement non équilibré

sons notre nouvel individu à partir des cellules sélectionnées de ses parents :

$$\{c_1, c_5, c'_1, c'_2, c'_3\} \tag{6.12}$$

Avec ce croisement, on obtient une solution dont la discrétisation dans l'espace de design est fort éloignée de celles des parents (Figure 6.5).

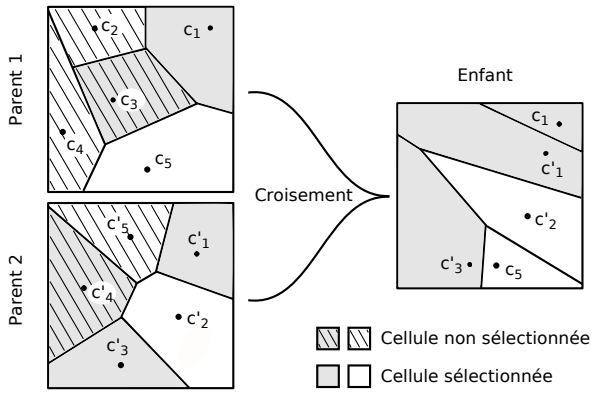

Figure 6.5 – Exemple de croisement non graphique

6.2 Mutations

Les opérateurs de mutation (les mutateurs) de l'algorithme génétique permettent de modifier un individu au travers de perturbations, c'est-à-dire de modifications des variables de design. L'amplitude de la perturbation va influencer le type de recherche. Pour une faible amplitude, l'algorithme effectue une phase d'exploitation et pour une grande amplitude, l'algorithme effectue une phase d'exploration.

On distingue plusieurs types de mutations jouant sur les variables de design : la mutation de matière, la mutation des positions des centres de Voronoï et la mutation du nombre de cellules de Voronoï. Les deux dernières mutations sont étudiés dans cette section.

6.2.1 Déplacements des centres de Voronoï

Le déplacement d'un centre de Voronoï est décrit au travers de quatre paramètres : la cellule ciblée, le type de déplacement, l'amplitude du déplacement et l'orientation du déplacement. Ces quatre paramètres sont étudiés dans les sections suivantes en étant toujours attentif à ce que le début de l'optimisation soit dirigé par la phase d'exploration et la fin de l'optimisation par la phase d'exploitation afin de garantir une meilleure forme finale.

6.2.1.1 Distinction entre les cellules

La topologie découlant de l'association d'un diagramme de Voronoï et d'une distribution de matière partitionne l'espace de design en une série de domaines homogènes de matière. Ceux-ci sont des regroupements de cellules possédant le même matériau et ayant entre-elles une connexité. (Figure 6.6). La forme de chaque domaine de matières est déterminée au travers de ces regroupements de cellules. Cette forme est définie par un ensemble de segments qui sont les médiatrices de couples de centres de Voronoï n'appartenant pas au même domaine de matière.

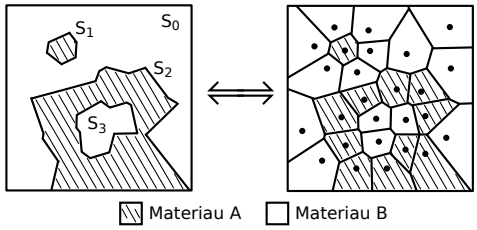

Figure 6.6 – Exemple de topologie décrite au moyen du formalisme Voronoï

Les cellules ne partageant pas de médiatrice avec une cellule d'un autre domaine ne participent donc pas à la description de la forme de la solution. La suppression de ces cellules ne change nullement la description de la distribution de matières au sein de l'espace puisque la surface couverte par la cellule supprimée se retrouve partagée entre ses cellules voisines (Figure 6.7).

On nomme par la suite « cellule interne » à un domaine une cellule de Voronoï qui ne partage aucune médiatrice avec des cellules d'un autre domaine et qui ne joue donc aucun rôle dans la description de la forme

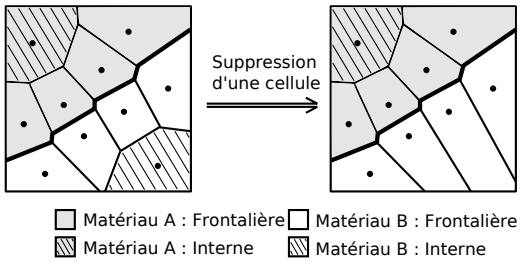

Figure 6.7 – Exemple de suppression d'une cellule interne

de la solution. Par contre, une cellule qui partage une ou plusieurs médiatrices avec des cellules d'un autre domaine de matière, est nommée « cellule frontalière » et contribue à la description de la forme et de ses dimensions.

Cette distinction entre cellules frontalières et cellules internes est très importante. On la retrouvera aux sections 6.2.2.5 et 6.2.2.6 relatives à la suppression ou à l'ajout de cellules. Néanmoins, pour cette section, nous nous limitons à leur rôle joué lors d'un déplacement.

Dans l'hypothèse de faibles déplacements, c'est-à-dire de déplacement dont l'amplitude est suffisamment faible pour ne pas changer le type de la cellule, les cellules internes et les cellules frontalières ont un apport différent dans l'évolution d'une solution. Le déplacement d'une cellule interne ne change en rien la distribution de matières au sein de l'espace de design mais seulement la discrétisation de celui-ci (Figure 6.8). Alors que le déplacement d'une cellule frontalière va impliquer la modification de la forme voir même de la topologie par changement des règles de connexités entre les domaines de matières.

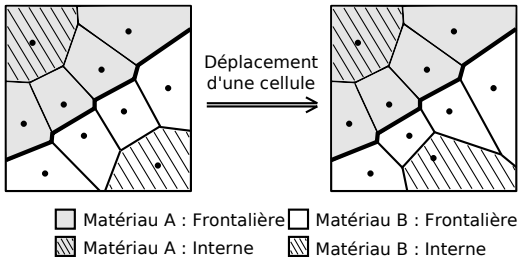

Figure 6.8 – Exemple de déplacement de cellule interne

Tenir compte des différences du rôle joué par les cellules frontalières et par les cellules internes permet de concentrer la mutation sur l'optimisation de la forme d'une solution. Lorsqu'on se trouve avec N_c cellules réparties en N_{cf} frontalières et en N_{ci} internes, la probabilité $p'_{p,int}$ de perdre une évaluation[2] en modifiant la position d'un centre de Voronoï d'une cellule interne dépend de la probabilité p_p de modifier la position d'une cellule et du pourcentage de cellules internes :

$$p'_{p,int} = p_p \frac{N_{ci}}{N_c}. \tag{6.13}$$

En appliquant la mutation seulement aux cellules frontalières, on diminue les pertes d'évaluations. Ces résultats sont confirmés et illustrés par le graphique de la Figure 6.9, où l'on voit, dans la mesure où on admet que le déplacement d'un centre de Voronoï d'une cellule interne ne produit aucune modification des frontières entre domaines de matière, que tenir compte du type de la cellule peut jouer un grand rôle pour de faibles amplitudes de déplacement.

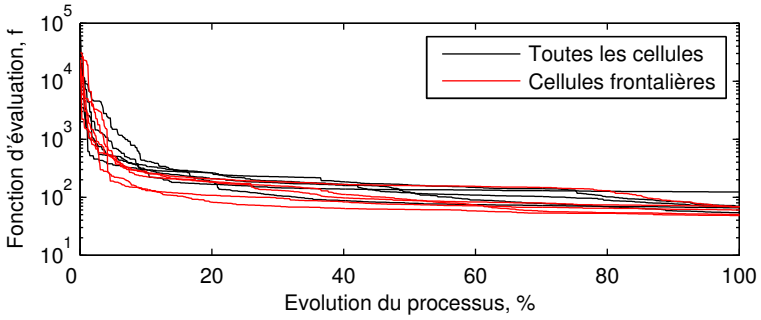

Figure 6.9 – Comparaison entre les résultats obtenus avec une mutation appliquée sur toutes les cellules et sur seulement les cellules frontalières – (10000 évaluations ; 10 individus)

La figure a été tracée pour des déplacements dont l'amplitude est choisie aléatoirement entre 0 et une valeur maximale α valant 10% des

[2]C'est à dire d'effectuer une évaluation inutile lorsqu'on l'applique à un individu qui n'a pas été modifié au niveau des domaines qui le constituent et possède donc toujours la même topologie, la même forme et les mêmes dimensions.

dimensions de l'espace de design[3]. Les valeurs élevées de déplacement de la position des centres de Voronoï donnent un risque plus élevée de modifier la topologie de la solution, ce qui n'est que très peu souhaité sur une grande partie du processus d'optimisation.

Cette méthode permet aussi d'améliorer considérablement la variance des résultats. Le Tableau 6.2 montre un gain pour la valeur minimale, la valeur médiane et la valeur maximale au bout des 1000 itérations. Seul la valeur minimale reste assez proche et légèrement en retrait.

Table 6.2 – Résumé des résultats obtenus avec 5 essais pour chaque test : Mutation sur toutes les cellules ou sur les cellules frontalières.

	Toutes les cellules	Cellules frontalières
Minimum	52.4	48.3 (-7.1%)
Médiane	67.8	60.1 (-10.9%)
Maximum	122.2	67.4 (-45.1%)

6.2.1.2 Type de déplacement de la cellule

Chaque variable de position a donc une probabilité p_p d'être modifiée. Toutefois, la probabilité que les deux variables décrivant la position d'une cellule soient modifiées simultanément est de p_p^2, (Figure 6.10).

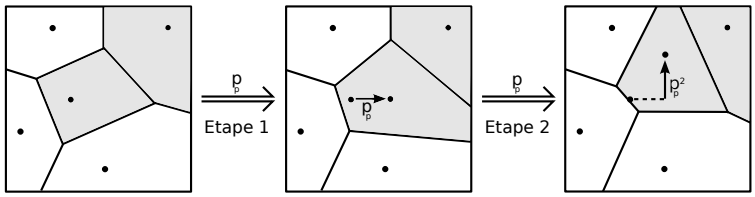

Figure 6.10 – Exemple où la probabilité de modifier la position d'un centre selon les deux axes est plus faible que sur un seul axe avec une mutation classique

[3]La valeur de 10% est une valeur proche de la distance moyenne entre les cellules dans le cas où les 100 cellules qui discrétisent l'espace de design dans l'exemple considérés sont réparties uniformément dans cet espace

Les déplacements effectués au travers de la mutation classique de position des centres de Voronoï se font donc préférentiellement suivant un des axes définissant l'espace de design, ce qui limite la zone accessible au centre de Voronoï touché par la mutation. Le schéma de la Figure 6.11 montre les déplacements possibles des centres des cellules avec une faible amplitude de déplacement au maximum de $\alpha = 10\%$ des dimensions de l'espace de design ainsi que l'aspect contraignant de devoir suivre les axes principaux.

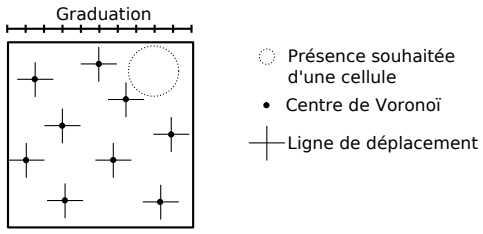

Figure 6.11 – Exemple de déplacement possible pour des amplitudes faibles de déplacement, $\alpha = 10\%$

Afin de libérer le déplacement des centres de Voronoï de cette contrainte, une autre méthode consiste à appliquer le taux de mutation p_p non pas sur chacune des variables mais bien sur chacune des cellules. Si le test est réussi, la cellule est déplacée et ses variables de position se voient attribuer une nouvelle valeur. Une mutation polaire est implémentée et les variables aléatoires sont r pour l'amplitude et θ pour l'orientation, tel que :

$$\begin{cases} x_i^{(k+1)} = x_i^{(k)} + r\cos(\theta) \\ y_i^{(k+1)} = y_i^{(k)} + r\sin(\theta) \end{cases} \tag{6.14}$$

Les courbes de convergence des résultats obtenus sont présentées à la Figure 6.12 pour le cas où r est choisi aléatoirement entre 0 et $\alpha = 10\%$ et θ entre 0 et 2π. Au début du processus d'optimisation, on peut observer que les courbes des deux approches ont une évolution identique. Au delà des 10% du processus d'optimisation, la meilleure valeur est obtenue avec la mutation polaire.

Le Tableau 6.3 donne la synthèse chiffrée numérique des résultats obtenus sur l'ensemble des tests. Les résultats montrent que pour les valeurs minimales, médianes et maximales, la mutation polaire est plus efficace.

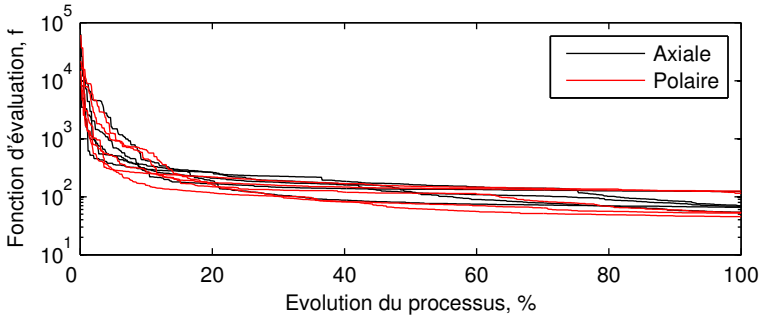

Figure 6.12 – Comparaison des courbes de convergence entre avec mutation polaire et avec mutation axiale pour des amplitudes de déplacement variant aléatoirement entre 0 et $\alpha = 10\%$ – (10000 évaluations ; 10 individus)

Les valeurs minimales et les valeurs médianes des deux cas étudiés sont très éloignées mais les valeurs maximales se rapprochent fortement.

Table 6.3 – Résumé des résultats obtenus avec 5 essais pour chaque test : Mutation axiale et mutation polaire.

	Mutation axiale	Mutation polaire
Minimum	53.3	45.5 (-14.6%)
Médiane	68.2	53.4 (-21.7%)
Maximum	122.9	125.4 (+0.02%)

6.2.1.3 Amplitude maximale de déplacement : α

L'amplitude maximale de la mutation de déplacement est un facteur influençant la qualité de la phase d'exploration ou d'exploitation lorsque cette amplitude est fixée à une valeur constante au début du processus d'optimisation. L'utilisateur est par conséquent chargé de choisir l'amplitude α des déplacements des centres de Voronoï qui apporterait le meilleur résultat. Ce choix dépend à la fois du cas d'étude mais aussi du nombre de cellules. Dès lors, il est impossible pour l'utilisateur de choisir la valeur idéale sans réaliser au préalable une étude paramétrique.

En début d'optimisation, afin d'explorer au mieux l'espace des solu-

tions, des mutations à forte perturbation sont nécessaires. De grandes amplitudes de déplacement des centres de Voronoï sont tout à fait adéquates pour répondre à cette exigence[4]. Le graphique de gauche de la Figure 6.13 montre qu'effectivement, au plus l'amplitude est élevée au plus la population a la possibilité d'évoluer rapidement. Les graphiques du milieu et de droite de la Figure 6.13 sont obtenus en initialisant la population à partir de la meilleure solution obtenue dans le graphique précédent. On remarque qu'au cours du processus d'optimisation, les grandes perturbations apportent de moins en moins de bonnes évolutions. Il est alors nécessaire de réduire l'amplitude des déplacements pour diminuer les perturbations dans la population. La transition entre le graphique de gauche et le graphique de droite en passant par celui du milieu illustre l'influence de l'amplitude de mutation au cours du processus d'optimisation.

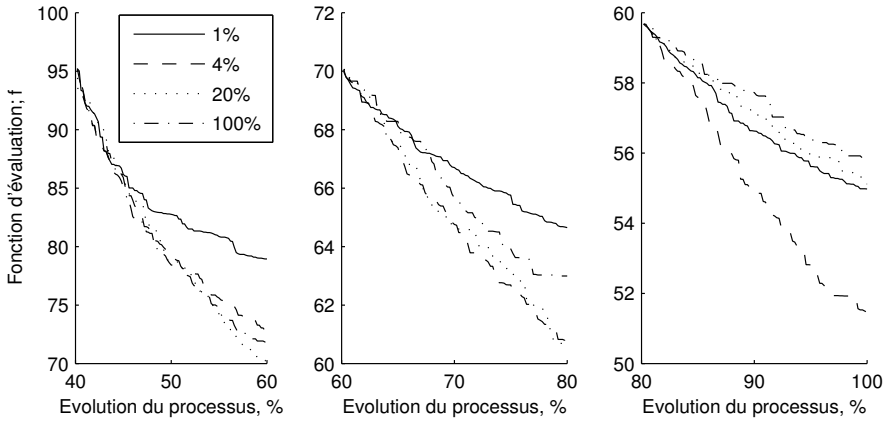

Figure 6.13 – Exemple de l'influence de l'amplitude de déplacement des cellules de Voronoï sur trois étapes successives du processus d'optimisation

Afin de ne pas devoir choisir un paramètre qui va conditionner le processus d'optimisation, il est nécessaire de mettre en place une stratégie de modification des amplitudes des déplacements maximales possibles des centres de Voronoï. Trois stratégies sont étudiées. La première force

[4]Lorsque les déplacements sont importants, le point d'arrivé peut se situer hors de l'espace de design. Dans ce cas, on le ramène dans l'espace de design en appliquant un effet miroir à sa position par rapport à la frontière de l'espace de design qu'il a franchi

la valeur α à diminuer au cours du processus d'optimisation, les deux autres méthodes prennent par contre en compte la configuration de la discrétisation de l'espace de design pour calculer l'amplitude α.

Méthode cinématique

La méthode cinématique consiste à forcer l'amplitude de déplacement en fonction du nombre d'itérations déjà effectuées, k, au cours du processus d'optimisation. Cependant, cette valeur ne donne pas d'information concernant l'évolution de la performance de la population.

Elle permet en quelque sorte de forcer l'outil d'optimisation à passer par des étapes prédéfinies pour réduire α. Dans notre cas, nous avons utilisé cette information afin de diminuer linéairement la valeur de α d'itération en itération en passant de 1 à 0, avec une pente en fonction du nombre maximal d'itérations alloués à l'optimisation :

$$\alpha\left(k\right) = 1 - \frac{k}{k_{max}} \tag{6.15}$$

Méthodes dynamiques

Contrairement à la méthode cinématique, les deux autres méthodes utilisent une information interne à la structure du diagramme de Voronoï pour calculer la valeur de l'amplitude maximale des déplacements des centres de cellules. La première version, nommée méthode dynamique globale, se base sur la distance minimale moyenne entre chaque cellule et ses cellules voisines :

$$\alpha^{(k)}\left(\vec{\delta}\right) = \frac{\sum \delta_{i,min}^{(k)}}{2N_c^{(k)}} \tag{6.16}$$

avec $\delta_{i,min}^{(k)}$ la distance minimale de la cellule i avec ses cellules voisines. Cette distance est calculée à chaque génération. La seconde méthode, nommée dynamique locale, se base sur la distance minimale entre les cellules voisines et attribue cette valeur personnalisée de déplacement à chacune des cellules :

$$\alpha_i^{(k)}\left(\vec{\delta_i}\right) = \frac{\delta_{i,min}^{(k)}}{2} \tag{6.17}$$

Résultats

Les graphiques de la Figure 6.14 montrent la médiane des résultats obtenus pour l'ensemble des essais effectués pour chaque méthode. La méthode cinématique et la méthode dynamique globale donnent les meilleurs résultats. Cependant, cette seconde mutation nécessite plus d'itérations pour marquer une évolution supérieure aux autres méthodes.

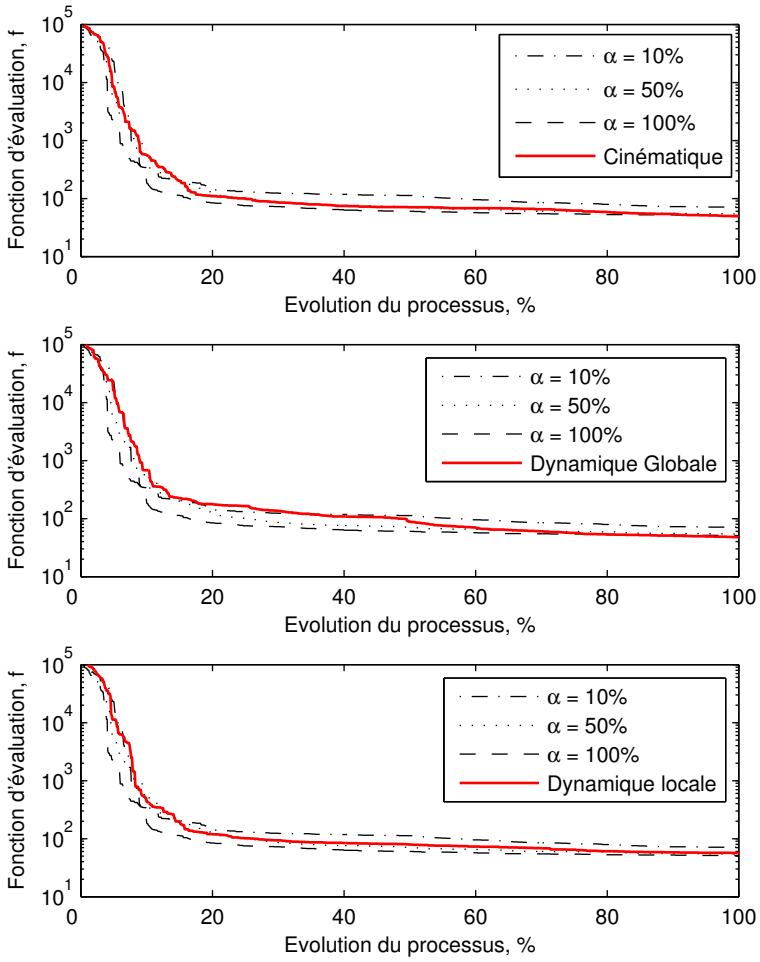

Figure 6.14 – Comparaison des courbes médianes de convergence pour la mutation T cinématique, la mutation dynamique globale D_G et la mutation dynamique locale D_L – (10000 évaluations ; 10 individus)

Le Tableau 7.2 complète la synthèse des résultats obtenus pour les quatre méthodes étudiées : statique, cinématique, dynamique locale et dynamique globale.

	k=500			k=1000		
	min.	med.	max.	min.	med.	max.
$\alpha = 10\%$	54.08	112.2	168.53	47.90	70.81	128.22
$\alpha = 50\%$	61.05	70.33	**86.51**	49.55	53.63	**63.18**
$\alpha = 100\%$	56.76	**60.16**	104.46	49.65	51.13	98.94
Cin.	60.70	70.36	162.7	46.58	49.58	65.77
Dyn. loc.	**45.92**	79.38	104.50	**37.88**	56.62	65.57
Dyn. glob.	52.35	87.65	208.57	40.10	**47.86**	136.23

Table 6.4 – Valeurs minimales, médianes et maximales des résultats à $k = 500$ and $k = 1000$ pour les configurations suivantes : $\alpha = 10\%$, $\alpha = 50\%$, $\alpha = 100\%$, $\alpha = \alpha(k)$, $\alpha = \alpha\left(\vec{\delta}\right)$ et $\alpha_i = \alpha_i\left(\vec{\delta}_i\right)$

Les valeurs numériques obtenues sur deux intervalles différents y sont reprises. Le premier intervalle, de $k = 0$ à $k = 500$, permet d'observer les résultats à court terme et ainsi évaluer la capacité d'exploration. Le second intervalle, de $k = 500$ à $k = 1000$, permet d'observer les résultats à long terme et ainsi évaluer les capacités d'exploitation.

Les performances de la méthodes cinématique dépendent du choix de la vitesse d'évolution de l'amplitude des déplacements. Une évolution trop rapide peut entraîner une exploration insuffisante de l'espace des solutions et une évolution trop lente peut entraîner une exploitation insuffisante.

Le principal inconvénient des méthodes dynamiques locale et globale réside dans le fait qu'elles ne prennent pas en compte du besoin de grands déplacements en phase d'exploration.

6.2.1.4 Orientation du déplacement des cellules

Description

L'objectif de la méthode proposée, la méthode du gradient local, est d'utiliser les déplacements effectués par les cellules à chaque génération combinés à l'évolution de la fonction objectif pour en déduire une proposition d'évolution pour la prochaine génération.

Si l'on note $(x,y)_i^{(k)}$ la position du centre de la cellule i à l'itération k, sa variation de position peut s'écrire

$$\vec{v}_i^{(k)} = (x,y)_i^{(k-1)} - (x,y)_i^{(k)} \tag{6.18}$$

et la variation de l'évaluation obtenue pour l'ensemble des transformations opérées est $\Delta_f^{(k)} = f^{(k-1)} - f^{(k)}$.

Trois cas possibles de $\Delta_f^{(k)}$ sont à traiter. Si $\Delta_f^{(k)}$ est nulle, la reproduction n'a apporté aucune modification. Si $\Delta_f^{(k)}$ est positive, la $k^{ème}$ solution est meilleure que la $(k-1)^{ème}$ solution. A la prochaine itération, $k+1$, il peut être intéressant d'utiliser toutes les informations obtenues, $\vec{v}_i^{(k)}$, pour influencer le mouvement des centre de Voronoï i. Si $\Delta_f^{(k)}$ est négatif alors la performance de la solution est inférieure à celle de la solution $k-1$. On peut exploiter l'information $\vec{v}_i^{(k)}$ pour orienter dans le sens inverse le mouvement des cellules. Ces trois cas s'écrivent :

$$(x,y)_i^{(k+1)} = \begin{cases} (x,y)_i^{(k)} - \vec{\beta}_i^{(k)} \cdot \vec{v}_i^{(k)} & \text{si } \Delta_f^{(k)} > 0 \\ (x,y)_i^{(k)} + \vec{\beta}_i^{(k)} \cdot \vec{v}_i^{(k)} & \text{si } \Delta_f^{(k)} < 0 \\ (x,y)_i^{(k)} & \text{si } \Delta_f^{(k)} = 0 \end{cases} \tag{6.19}$$

où le terme $\vec{\beta}_i^{(k)} \cdot \vec{v}_i^{(k)}$ représente le produit terme à terme entre un vecteur aléatoire et le vecteur de direction.

L'introduction du terme $\vec{\beta}_i^{(k)}$ permet de donner une approche aléatoire de la trajectoire du centre de Voronoï i. Son produit avec le vecteur de direction donne une nouvelle direction $\vec{u}_i^{(k)}$. Cette direction est caractérisée par une longueur $\ell_i^{(k)} \in \left[0, \ell_{i,max}^{(k)}\right]$ et par un angle $\theta_i^{(k)} \in [-\pi, \pi]$. La Figure 6.15 illustre la relation entre $\vec{u}_i^{(k)}$ et $\vec{v}_i^{(k)}$.

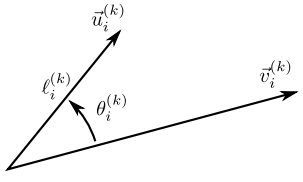

Figure 6.15 – Représentation des paramètres $\theta_i^{(k)}$ et $\ell_i^{(k)}$ du vecteur $\vec{u}_i^{(k)}$ en fonction du vecteur $\vec{v}_i^{(k)}$

Le choix de ces deux paramètres est fonction de deux objectifs :

- Il est souhaité d'avoir une plus forte probabilité d'obtenir un $\theta_i^{(k)}$ proche de 0 que proche de π ou de $-\pi$ afin d'être le plus proche possible de la direction $v_i^{(k)}$;
- La longueur maximale $\ell_{i,max}^{(k)}$ doit être fonction de l'angle d'écartement $\theta_i^{(k)}$. Au plus l'angle est proche de 0, au plus $\ell_{i,max}^{(k)}$ est proche de $\left\|\vec{v}_i^{(k)}\right\|$. Au plus l'angle s'approche de π ou $-\pi$, au plus $\ell_{i,max}^{(k)}$ s'approche de 0.

Afin de respecter ces deux objectifs fixés, une variable aléatoire X_i, définie selon une loi de distribution normale centrée,

$$X_i \sim \mathcal{N}(0,1), \tag{6.20}$$

est utilisée dans deux fonctions, $\phi(X_i)$ et $\Theta(X_i)$, pour déterminer la longueur maximale $\ell_{i,max}^{(k)}$ et l'angle d'écartement $\theta_i^{(k)}$ respectivement. La distance $\ell_{i,max}^{(k)}$ est calculée au moyen de la fonction $\phi(X_i)$ définie sur l'intervalle $[-B;B]$ avec $B=4$ pour convrir plus de 99% des possibilités :

$$\begin{aligned} \ell_{i,max}^{(k)} &= \phi(X_i) \\ \phi(X_i) &= \left(u(X_i+B)(X_i+B) - 2X_i u(X_i)\, u(B-X_i)\right)\frac{d_{max}}{B} \end{aligned} \tag{6.21}$$

où $u(x)$ est la fonction échelon[5] et $d_{max} = \left\|\vec{v}_i^{(k)}\right\|$. La fonction $\phi(X_i)$ est illustrée sur la Figure 6.16.

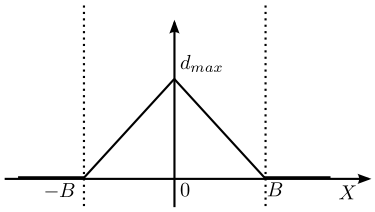

Figure 6.16 – Représentation de la fonction $\phi(X_i)$

Il ne reste plus que l'angle $\theta_i^{(k)}$ à calculer. Celui-ci est obtenu en

[5]Pour toutes valeurs de x supérieures ou égales à 0, la fonction échelon $u(x)$ vaut 1 sinon elle vaut 0

considérant une fonction linéaire $\theta_i^{(k)} = \Theta(X_i)$ illustrée par la Figure 6.17.

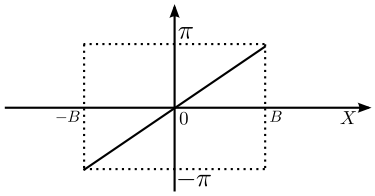

Figure 6.17 – Représentation de la fonction $\Theta(X_i)$

Résultats

La mutation par gradient local, G_L, est introduite directement après les mutations classiques comme le montre le schéma de la Figure 6.18.

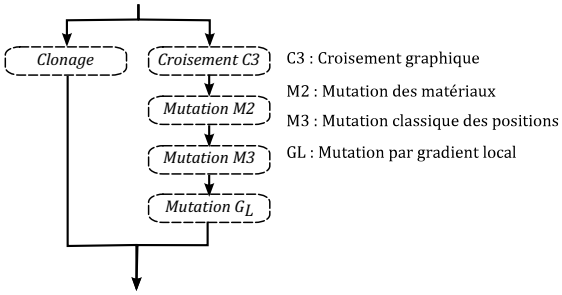

Figure 6.18 – Schéma de la configuration utilisée de la reproduction

Nous fixons la probabilité de mutation par gradient local à $p_g = {}^{10}/_{N_c}$. Cette valeur est dix fois supérieure aux valeurs habituelles pour une mutation. Ayant une information plus forte sur la manière dont il convient de déplacer les centres de Voronoï, il est intéressant d'exploiter une valeur plus élevée.

Les résultats obtenus avec le cas d'une amplitude de déplace fixé à $\alpha = 10\%$ et avec le cas du gradient, illustrés par la Figure 6.19, montrent que l'outil est plus performant dans le cas d'une mutation avec gradient. On observe qu'avant les 10% du processus d'optimisation, la population converge plus rapidement. A partir de 10% et jusqu'à 20% du processus d'optimisation, l'outil de référence est plus performant. De 20% à 35%,

les deux approches se valent. Et finalement, après 35%, la mutation avec le gradient donne de meilleurs résultats.

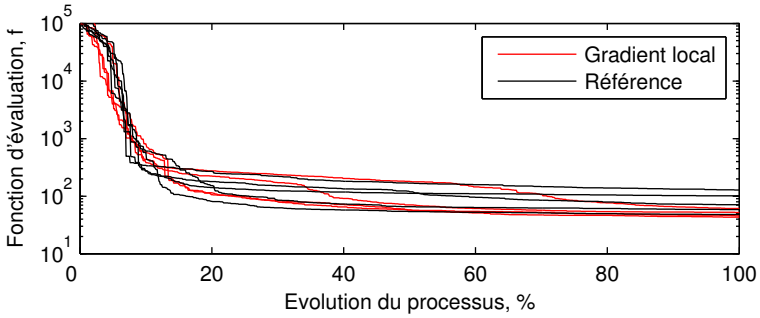

Figure 6.19 – Comparaison de convergence entre la référence et avec la mutation G_L – (10.000 évaluations ; 10 individus)

Les résultats sont résumés dans le Tableau 6.5. Pour les trois catégories (minimum, maximum et médiane), les résultats sont en faveur de la mutation G_L avec des différences supérieures à 14%.

	Minimum	Médiane	Maximum
Référence	49.8	68.2	105.9
G_L	43.07 (-14%)	52.04 (-23%)	71.9 (-33%)

Table 6.5 – Comparaisons des valeurs minimales, médianes et maximales obtenues pour 5 essais avec et sans la mutation G_L

Le choix d'une probabilité de mutation élevée pour le mutateur G_L, par gradient local, entraîne au début du processus d'optimisation de fortes variations supplémentaires de position des centres de Voronoï améliorant la phase d'exploration. Cependant, ce supplément de variation devient néfaste à l'optimisation lorsque de plus faibles variations deviennent souhaitables. Il est nécessaire d'attendre que le gradient n'effectue plus que de petits déplacements de cellules pour que la mutation G_L redeviennent plus performante. Ceci est obtenu au cours du processus d'optimisation lorsque les meilleurs individus sont obtenus avec de légers déplacements des centres de Voronoï par la mutation classique des positions des centres, et donc de plus petits $\ell_{i,max}^{(k)}$. Pour la suite de l'optimisation le gradient local aide l'outil à obtenir de meilleurs résultats.

6.2.2 Modification du nombre de cellules

En optimisation, il est souvent difficile de quantifier ou de qualifier la complexité d'un problème à optimiser. Plusieurs facteurs interviennent, mais l'un des plus importants est le nombre de variables d'optimisation.

Dans le domaine de l'optimisation topologique, le nombre de variables définissant l'espace de recherche est le plus souvent très élevé et directement lié à la précision de description de chaque topologie. Ce nombre de variables est un paramètre de contrôle que l'utilisateur doit choisir avec soin sur base d'un compromis entre la complexité de l'algorithme et le niveau de détail graphique de la topologie. Plus il est élevé, plus l'optimisation sera longue et complexe, mais plus la solution produite au terme de l'optimisation sera susceptible d'être graphiquement précise. Le choix du nombre de cellules de Voronoï est difficile à priori car fortement dépendant du problème. Un mauvais choix du nombre de cellules peut conduire soit à résoudre un problème dont l'optimum est inaccessible lorsque le nombre de cellules de Voronoï est insuffisant, ou soit à devoir travailler sur un plus grand nombre d'itération pour converger vers l'optimum lors d'un nombre trop élevé de cellules de Voronoï. Il est donc intéressant de pouvoir travailler sur ce paramètre afin de faciliter la tâche à l'outil lorsque cela est nécessaire sans pour autant diminuer la qualité de la solution finale.

6.2.2.1 Compromis entre la qualité de la solution et la vitesse de convergence

Le graphique de la Figure 6.20 présente les conséquences du choix du nombre initiale de cellules, sur la vitesse de convergence du processus d'optimisation. Il montre les courbes de convergence de cinq essais réalisés avec $N_c = 10$, $N_c = 20$, $N_c = 50$, $N_c = 100$ ou $N_c = 200$. On observe une nette différence de vitesse de convergence entre ces différents essais. Un grand nombre de cellules conduit à une convergence plus lente qu'avec un petit nombre de cellules, mais permet de donner un plus grand nombre de degrés de liberté pour décrire la solution. La courbe avec 10 cellules initiales montre une convergence rapide vers un optimum local alors que les autres courbes continuent à évoluer.

Les résultats graphiques des distributions de matière obtenues sont présentés à la Figure 6.21. On observe deux tendances : la qualité finale de la solution et la vitesse de convergence.

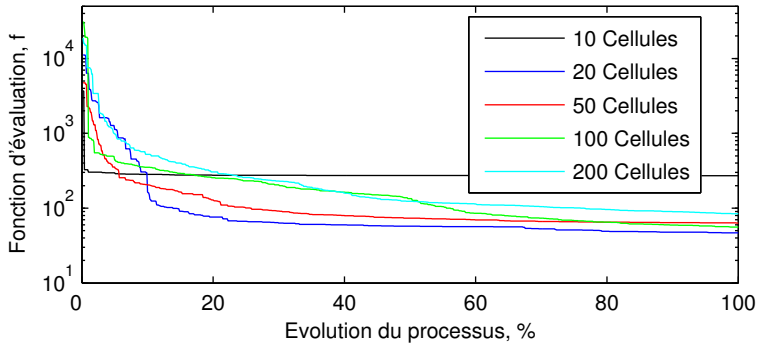

Figure 6.20 – Comparaison de convergence pour différents nombres de cellules initiales – (10.000 évaluations ; 10 individus)

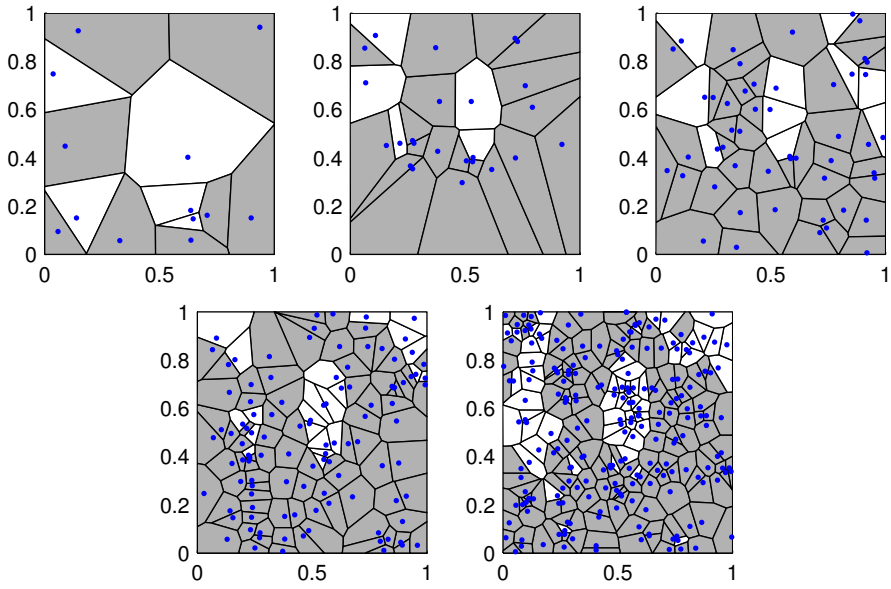

Figure 6.21 – Représentation des topologie trouvées dans les cas de 10 cellules, 20 cellules, 50 cellules, 100 cellules et 200 cellules – (10.000 évaluations ; 10 individus)

En premier lieu, avec un nombre de cellules trop faible, le niveau de discrétisation de l'espace de design n'est pas suffisamment fin que pour permettre une approche correcte de la topologie. Au plus on augmente

le nombre de cellules et au plus on observe qu'il est possible d'approcher l'optimum global réel.

En second lieu, le niveau de discrétisation de l'espace de design joue sur la vitesse de convergence. Utiliser un faible nombre de cellules de Voronoï permet d'accélérer la convergence, alors que l'utilisation d'un trop grand nombre de cellules diminue la vitesse de convergence.

Suite à ces deux tendances, le nombre de cellules utilisées pour la discrétisation de l'espace de design est un critère fort important jouant directement sur la complexité du problème et la qualité de la solution comme on l'a déjà mentionné. Il semble donc intéressant de jouer sur ce nombre de cellules au cours du processus d'optimisation.

6.2.2.2 Temps de l'évaluation en fonction du nombre de cellules

Le temps d'évaluation d'une solution est directement lié au nombre de cellules de Voronoï. La résolution numérique du modèle élément-finis dépend du nombre de sous-domaines directement lié au nombre de cellules (Section 3.2.1.2). En variant ce nombre de sous-domaines et en utilisant un maillage automatique proposé par $Comsol^{©}$, la taille des éléments du maillages va s'adapter. Il s'avère qu'au plus les sous-domaines sont petits, au plus les éléments du maillage vont se réduire, augmentant ainsi le nombre total de mailles. Les résultats du Tableau 6.6 reprend les temps moyens nécessaires à l'évaluation des solutions initiales des essais présentés à la Figure 6.20. Les valeurs montrent que le temps d'évaluation est fort dépendant du nombre de cellules. A cela s'ajoute le temps nécessaire pour la création dynamique du modèle éléments-finis au moyen du diagramme de Voronoï.

Table 6.6 – Temps moyen de 50 évaluations pour 10, 20, 50, 100 et 200 cellules

Nombre de Cellules	10	20	50	100	200
Temps moyen (sec.)	1	1.2	1.7	2.25	3.6

6.2.2.3 Évolution naturelle du nombre de cellules en mode graphique

Naturellement, avec la méthode de croisement graphique, l'algorithme peut indirectement faire évoluer le nombre de cellules pour décrire une topologie, comme on peut l'observer sur la représentation de l'évolution du nombre de cellules du meilleur individu au cours du processus d'optimisation sur la Figure 6.22.

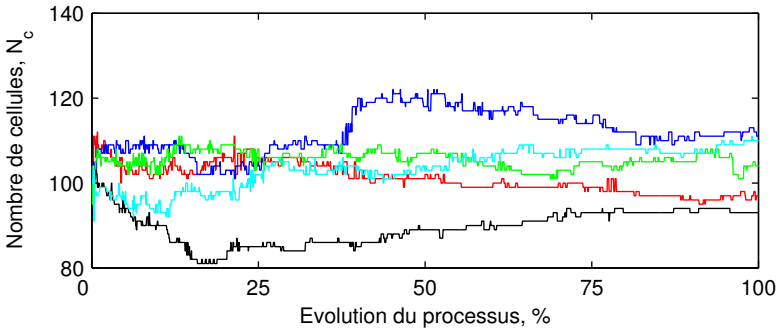

Figure 6.22 – Évolution du nombre de cellules pour 5 essais du cas de référence avec 100 cellules initiales – (20.000 évaluations ; 10 individus)

En combinant les valeurs obtenues pour les fonctions d'évaluation et le nombre de cellules, les meilleurs résultats sont pour les individus ayant le plus de cellules. Effectivement, grâce au gain de cellules, la forme de la solution globale est mieux approchée par une augmentation de degrés de liberté.

Les deux images de la Figure 6.23 nous montrent la distribution des centres de référence obtenue pour le plus performant des individus et le moins performant des individus obtenus avec 100 cellules initiales. L'individu de droite a pris une mauvaise direction en éliminant les centres situés au niveau de la fissure. Par manque de degrés de liberté, la fissure n'a pu être dessinée. La probabilité est très faible de modifier la discrétisation de l'espace de design en une seule génération pour sortir de cet optimum local. C'est une situation où toute légère variation des variables d'optimisation tend à dégrader la performance de l'individu. Pour la meilleure solution, le nombre de cellules est minimisé pour les deux imperfections du matériau, mais on trouve encore de nombreuses

cellules inutiles se trouvant dans la matière même et n'intervenant pas dans la définition des frontières des domaines de matière. On notera cependant que le faible nombre d'individus utilisé pour résoudre ce problème et donc le faible niveau d'exploration, est la cause de la grande différence entre les résultats.

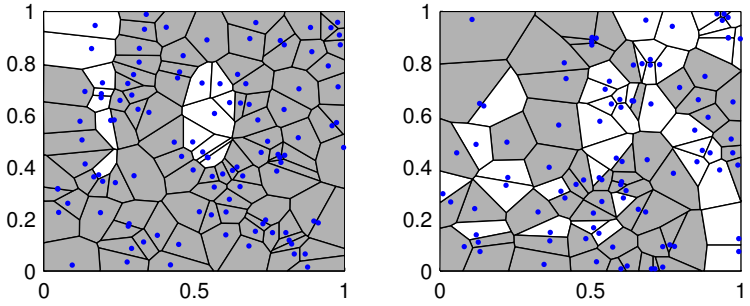

Figure 6.23 – Distribution des centres de Voronoï pour la meilleure (gauche) et la moins bonne solution (droite) – (20.000 évaluations ; 10 individus)

On observe donc dans cette section qu'il est nécessaire de garder un grand nombre de cellules pour permettre à l'outil d'explorer l'espace des solutions mais que ce nombre doit d'une part diminuer dans certaines régions du design et d'autre part augmenter dans d'autres régions en fonction de la nature des cellules, à savoir les cellules internes et les cellules frontalières. Ce transfert de centres de Voronoï augmente le nombre de degrés de liberté dans les zones sensibles pour décrire la topologie ainsi que la forme de celle-ci et ses dimensions, et permet de diminuer des degrés de liberté dans les zones peu sensibles.

6.2.2.4 Ajout et suppression de cellules suivant un processus aléatoire

Suite à l'analyse faite sur l'influence du nombre de cellules de Voronoï sur le résultat final, on peut conclure qu'il est intéressant de maîtriser l'évolution du nombre de cellules dans l'objectif d'adapter le niveau de discrétisation en fonction du stade de l'optimisation (exploration ou exploitation). La méthode la plus simple pour modifier le nombre de cellules au sein du diagramme de Voronoï est d'ajouter une mutation enlevant ou ajoutant des cellules aléatoirement dans l'espace de design.

Description

Modifier le nombre de cellules n'est pas chose aisée. Pour le cas de la mutation par ajout de cellules dans sa variante la plus simple, deux paramètres de contrôle doivent être définis : la probabilité d'ajout de nouvelles cellules de Voronoï dans l'espace de design et le nombre de cellules concernées. La position de chaque nouvelle cellule ajoutée est choisi aléatoirement dans le domaine de design. Pour le cas de la mutation par suppression de cellules, deux paramètres de contrôle doivent aussi être définis : la probabilité de suppression de cellules et le nombre de cellules concernées. Respectant le principe qu'une mutation ne sert qu'à diversifier une population, le nombre de cellules supprimées ou ajoutées est fixé à 1 et la probabilité est de 10% par individu pour ne pas modifier toute la population.

Lorsque le test de probabilité de l'ajout d'une cellule est vérifié, une cellule est créée et ajoutée aléatoirement dans la structure.

Lorsque le test de probabilité de la suppression d'une cellule est vérifiée, une cellule est choisie aléatoirement dans la solution pour être ensuite supprimée.

Résultats pour l'ajout de cellules

Comme on peut le voir sur le graphique de la Figure 6.24, on observe sur le courbes de convergence une légère amélioration des valeurs minimales et maximales obtenues en fin de processus d'optimisation pour le cas de l'ajout aléatoire d'une seule cellule par génération.

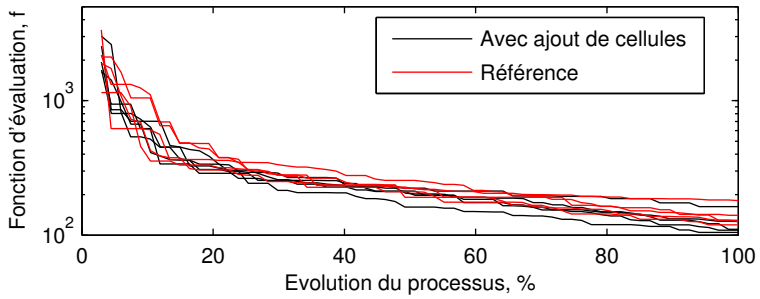

Figure 6.24 – Comparaison de convergence entre la méthode avec un ajout de cellules et celle sans – (20.000 évaluations : 300 individus)

Visuellement, les résultats graphiques (Figure 6.25), montrent une différence dans le nombre de cellules. Dans le cas de référence pour la meilleure solution obtenue, la densité de centres de Voronoï est plus faible que dans le cas de la meilleure solution obtenue avec la mutation d'ajout de cellule. Mais on ne constate pas d'amélioration sensible de la solution suite à l'ajout de cellules.

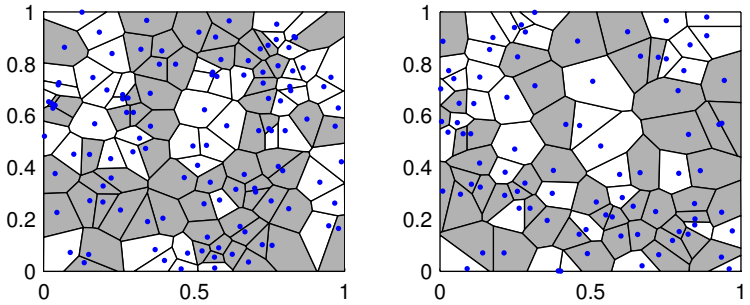

Figure 6.25 – Distribution des centres de Voronoï pour la meilleure solution avec ajout de cellules (gauche) et sans ajout (droite) – (20.000 évaluations : 300 individus)

Résultats pour la suppression de cellules

Les résultats sont similaires pour la suppression d'une cellule par itération. La mutation n'apporte ici aussi aucune aide pour améliorer le résultat (Figure 6.26).

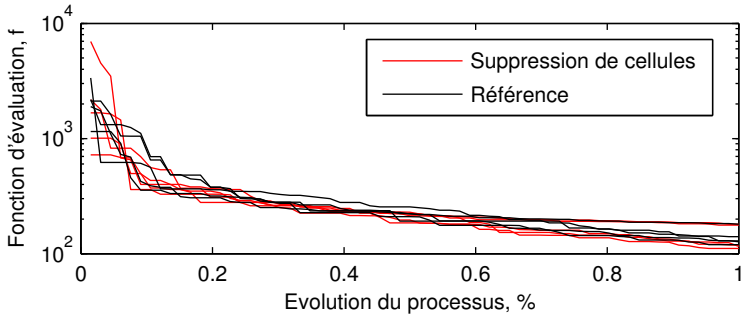

Figure 6.26 – Comparaison de convergence entre la méthode avec la suppression de cellules et celle sans – (20.000 évaluations ; 300 individus)

Limites de la méthode

L'ajout ou la suppression aléatoire d'une cellule par itération ne s'avère pas très efficace, comme on vient de le voir. Il convient donc d'établir des règles appropriés pour effectuer de manière adéquate les opérations d'ajout et de suppression de cellules. Ces règles font l'objet des sections suivantes.

6.2.2.5 Élimination des cellules internes basée sur le vieillissement cellulaire

Principe

Comme on a pu l'observer dans la section précédente, la mutation de suppression aléatoire d'une cellule par itération ne supprime pas nécessairement les cellules adéquates. Dans cette section, une autre approche est proposée.

Comme indiqué en section 6.2.1.1, chaque cellule joue un rôle différent dans la description de la topologie. D'un côté, nous avons les cellules frontalières qui sont responsables de la description des frontières des domaines de matière et d'un autre côté, nous avons les cellules internes, qui ne participent en rien à la description d'une topologie.

Toutefois, les cellules internes ne sont pas réellement inutiles car elles peuvent contribuer à améliorer la description d'une frontière en faisant migrer leur centre de Voronoï de telle manière qu'elles deviennent une cellule frontalière ou encore modifier la topologie de la solution en changeant de matériaux. Il est donc important de n'attribuer une probabilité élevée de suppression à une cellule que lorsque la probabilité qu'elle puisse modifier la topologie est quasi nulle.

C'est sur base de ce principe que le vieillissement cellulaire fonctionne. Chaque cellule composant une solution se voit attribuer un âge. Cet âge varie en fonction de son rôle joué dans la description d'une topologie. Lorsque celle-ci se retrouve en frontière, son âge est fixé à 0. Lorsqu'une cellule se retrouve en interne, cet âge est incrémenté à chaque nouvelle génération et est par la suite utilisé par l'outil pour déterminer si la cellule est nécessaire ou non. On peut considérer qu'à chaque génération, au plus une cellule sera âgée, au plus la probabilité qu'elle puisse contribuer à faire évoluer la solution sera faible et au plus la probabilité qu'elle puisse être supprimée de la structure pourra être

élevée.

La configuration de la reproduction utilisée est illustrée par le schéma de la Figure 6.27. Le vieillissement est positionné après le clonage. Les cellules n'apportant aucune amélioration dans la population principale sont éliminées au fur et à mesure pour permettre à l'algorithme de se concentrer sur les paramètres les plus sensibles qui se situent généralement sur les frontières.

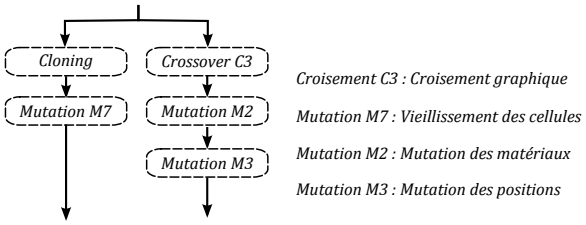

Figure 6.27 – Schéma de reproduction suivi par l'outil d'optimisation topologique étudié

Justification

L'intérêt de l'élimination de cellules sur base du vieillissement cellulaire est double.

En premier lieu, une diminution du nombre de cellules permet de diminuer le temps d'évaluation d'une solution. En effet, lors de la résolution par éléments-finis avec $Comsol^{©}$, chaque cellule représente un sous-domaine qui entraîne une dépendance du temps de calcul avec le nombre de cellules.

En second lieu, en fin de convergence, lorsque la topologie des solutions converge vers une seule topologie, il est plus intéressant de travailler sur le positionnement des cellules que sur la matière qui les constitue. A ce stade, si le déplacement des cellules est faible, empêchant une cellule de changer son voisinage, les cellules dites internes n'apportent aucune amélioration lors d'un déplacement. Or, dans le processus d'optimisation, une partie de la population n'est modifiée que par le déplacement de cellules internes, ce qui fait que l'outil d'optimisation perd à chaque fois une évaluation pour évaluer un individu qui en soi n'apporte directement aucun changement.

Pour vérifier cela, nous avons créé 500 individus enfants à partir d'un individu performant provenant d'un processus d'optimisation avancé et ayant déjà fortement convergé vers une solution. Ces 500 nouveaux individus sont créés par copie et ont subi le test de la mutation des déplacements. La topologie concernée contient 216 cellules dont 127 frontalières et 89 internes (Figure 6.28 - gauche).

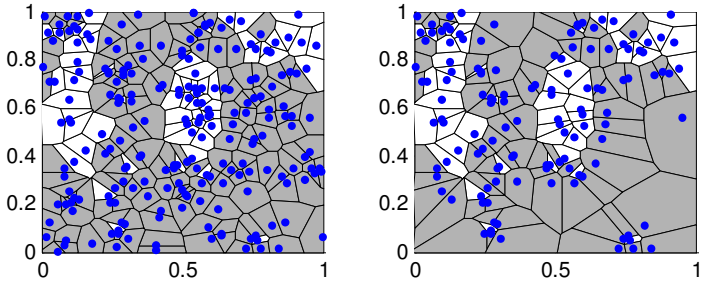

Figure 6.28 – Représentation de la distribution cellulaire au sein de l'espace de design d'une solution avec (gauche) et sans (droite) les cellules internes

Les 500 résultats obtenus sont triés en quatre catégories :

- les pertes : chaque enfant créé et subissant une mutation ne modifiant pas la topologie est considérée comme une perte ;
- les non-modifiés : chaque enfant ne subissant aucune mutation est un non-modifié ;
- les bons résultats : chaque enfant subissant une mutation améliorant sa performance est un bon résultat ;
- les mauvais résultats : chaque enfant subissant une mutation diminuant sa performance est un mauvais résultat.

Les résultats de l'essai de référence sont donnés dans le Tableau 6.7. L'outil perd effectivement 90 évaluations et a eu 76 individus non-modifiées. Concernant la réussite, on a 133 bons résultats et donc 201 solutions négatives.

La même procédure a été effectuée sur la même solution mais sans les cellules internes (Figure 6.28 - droite). Les résultats sont également donnés dans le Tableau 6.7. Deux situations sont étudiées soit le taux de mutation des positions des centres de Voronoï est conservé (1) malgré une diminution du nombre de cellules, soit ce taux est adapté (2) au nouveau nombre de cellules en adaptant la relation (6.13). Pour le pre-

	Référence	Sans cellules int. (1)	Sans cellules int. (2)
Pertes	90	0	0
Non-modifiés	76	162	78 (-84)
Bons rés.	133	123	173 (+50)
Mauvais rés.	201	215	249 (+34)
Temps (sec.)	968	580	602

Table 6.7 – Résumé de 500 essais de mutation sur un individu avec mutation de position des centres de Voronoï non adaptée (1) et adaptée (2) au nombre de cellules sans la mutation des matériaux

mier cas, on observe un nombre élevé de 162 individus non modifiés. Ce nombre est réduit à une valeur proche de la référence pour le cas avec adaptation du taux de mutation. Dans les deux cas, le nombre de non réussites est plus élevé que le cas de référence et le temps d'évaluation globale est plus faible. Toutefois, le nombre de réussites n'est plus élevé par rapport au cas de référence que pour le cas avec adaptation du taux de mutation des positions des centres de Voronoï.

En ajoutant la mutation des matériaux des cellules, avec une probabilité de $1/N_c$ initialement fixée et en considérant que chaque mutation de matériau est néfaste pour la solution vu que l'on a convergé topologiquement vers une solution, le taux de réussite va diminuer. En effet, les résultats du Tableau 6.8 confirment les tests.

	Référence	Sans cellules int. (1)	Sans cellules int. (2)
Pertes	36	0	0
Non-modifiés	32	32	39
Bons rés.	128	142	143
Mauvais rés.	304	326	318

Table 6.8 – Résumé de 500 essais de mutation sur un individu avec mutation de position des centres de Voronoï non adaptée (1) et adaptée (2) au nombre de cellules avec la mutation des matériaux

Les résultats nous montrent qu'il est donc nécessaire de diminuer le nombre de cellules internes pour diminuer les pertes d'évaluations mais aussi pour diminuer le temps d'évaluation. De plus, il faut surtout que la

mutation des matériaux des cellules soit faible. Lorsque l'on observe la distribution des cellules frontalières (Figure 6.28 - droite) on remarque qu'effectivement, dans ce cas là, il est généralement fortement déconseillé de changer de matériau d'une des cellules.

On préconise donc d'utiliser une faible évolution du vieillissement pour laisser l'outil explorer l'espace des solutions, d'utiliser un grand nombre de cellules, puis lors de la convergence, de laisser le taux de mutation des matériaux constant mais d'adapter le taux de mutation des déplacements des centres.

Lien entre le vieillissement et la probabilité de suppression d'une cellule

Au cours du processus d'optimisation, à chaque génération, chaque cellule interne i se voit attribuer une probabilité $p_{v,i}^{(k)}$ d'être supprimée. Cette probabilité doit être choisie de manière à ce que l'outil d'optimisation ait la possibilité de tester dans quelle mesure cette cellule est utile ou non dans la description de la distribution de matière. Pour cela, on fera croître la probabilité de suppression de chaque cellule interne proportionnellement à son âge suivant une loi de la forme :

$$p_{v,i} = h\left(\alpha_i, K_i\right) = \alpha_i K_i, \tag{6.22}$$

où α est le nombre de générations qui s'est écoulé depuis le début de son appartenance au groupe des cellules internes[6] et on choisit le coefficient de croissance K_i de manière à ce que $p_{v,i}$ atteigne une probabilité de 95% au bout d'un nombre N_g générations tel que la probabilité que la cellule n'ait pas subi une mutation de matière soit de 5%.

Si p_m est la probabilité qu'une cellule change de matière à une génération, la probabilité qu'elle n'ait pas changé au bout de k génération est de $(1-p_m)^k$. Le nombre N_g de générations nécessaires pour que la probabilité pour une cellule de ne pas avoir changé de matière soit de 5% est solution de :

$$(1-p_m)^{N_g} = 0.05 \tag{6.23}$$

Ce qui nous donne finalement :

[6]Ce début peut correspondre à l'initialisation du processus d'optimisation ou résulter d'une mutation à la $k^{ème}$ génération

$$N_g = \frac{\log 0.05}{\log (1 - p_m)} \tag{6.24}$$

Pour déterminer le coefficient K_i qui conduira à une probabilité d'élimination d'une cellule de 95% au bout de N_g générations, il faut considérer la chaine de Markov associée au processus de suppression d'une cellule. On peut calculer indirectement la probabilité $P_i\,[1]$ que la cellule i soit supprimée atteigne 95% au bout de N_g générations considérées au moyen de la probabilité que la cellule ne soit pas supprimée puisque les deux états sont complémentaires. Nous avons donc :

$$P\,[1] = 1 - P\,[0] = 1 - \prod_{j=1}^{N_g} (1 - jK_i) = 0.95 \tag{6.25}$$

Afin d'obtenir une valeur de K_i qui satisfasse à cette relation, nous utilisons la méthode de Newton-Raphson. L'amplitude des variations de la courbe autour de zéro croissant très rapidement, nous choisissons la valeur initiale $K_i = 0$ comme point de départ afin d'éviter des problèmes numériques.

Afin d'améliorer la méthode, nous allons considérer le passage entre la phase d'exploration et la phase d'exploitation. Au cours de son évolution, tous les individus de la population ont tendance à converger vers une seule topologie avec de légères différences en formes et en dimensions. La topologie est identique et les centres de Voronoï sont similaires à quelques variantes près d'un individu à un autre, ce qui fait que les variables d'optimisation des individus tendent à se ressembler de plus en plus au cours du processus d'optimisation. Ceci implique qu'en fin de convergence, la modification d'une cellule n'est donc plus testée par un seul individu mais par l'ensemble des individus de la population. Nous pouvons donc réduire le nombre de générations N_g, ce qui nous permet d'écrire :

$$N'_g = \frac{N_g}{N_s} \tag{6.26}$$

L'évolution de la population va transiter entre ces deux états de diversification de la population en fonction de sa convergence. On peut donc faire un lien entre les deux par une loi de transition :

$$N''_g = \frac{N_{ev,max} - N_{ev}}{N_{ev,max}} N_g + \frac{N_{ev}}{N_{ev,max}} N'_g \tag{6.27}$$

où $N_{ev,max}$ et N_{ev} sont respectivement le nombre d'évaluations maximum et le nombre d'évaluations effectuées.

Une relation non linéaire pourrait être plus adéquate à la situation, mais ne connaissant pas à l'avance l'évolution de la population principale, nous ne pouvons pas en dire plus sur la loi idéale pour relier les deux taux de vieillissement.

Résultats

Dans un cas idéal, on souhaiterait utiliser un grand nombre de cellules afin d'avoir un grand nombre de degrés de liberté. Bien qu'il n'y ait véritablement aucun moyen de qualifier de « grand » ou de « petit » un nombre de cellules, on considérera par la suite que 200 cellules est un grand nombre puisque dans les essais (Figure 6.22) en partant de 100 cellules, l'outil tend à varier le nombre de cellules avec une borne supérieure à 130 cellules.

On observe deux tendance dans les courbes de convergence illustrées à la Figure 6.29. Tout d'abord, les résultats des deux outils se ressemblent jusqu'à 60% avec quelques avantages pour le cas du vieillissement cellulaire. Au-delà de cette valeur, la pente d'évolution de la meilleure solution augmente. Les meilleurs résultats sont obtenus par l'outil avec le mécanisme de vieillissement cellulaire.

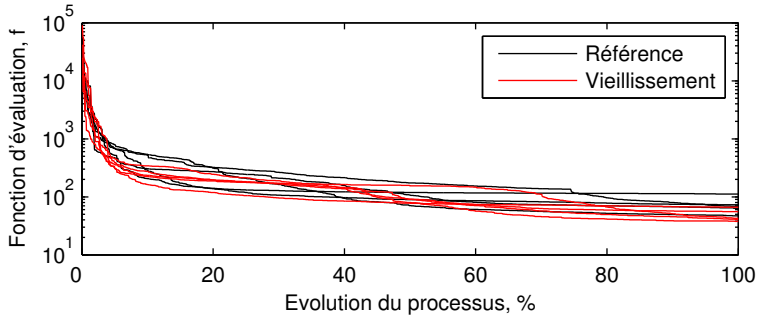

Figure 6.29 – Comparaison des courbes de convergence entre le cas sans et le cas avec le vieillissement cellulaire – (20000 évaluations ; 10 individus)

Les résultats numériques sont donnés dans le Tableau 6.9. Les valeurs minimales, médianes et maximales sont meilleures pour le cas avec le vieillissement cellulaire que sans ce mécanisme.

	Minimum	Médiane	Maximum
Référence	46.9	68.6	111.94
Vieillissement	37.8	42.5	66.3

Table 6.9 – Résultats numériques de cinq essais sans vieillissement cellulaire et de cinq essais avec vieillissement cellulaire

Les courbes d'évolution du nombre de cellules de Voronoï de la Figure 6.30 présentent l'évolution du nombre de cellules pour cinq essais dans le cas sans le mécanisme de vieillissement cellulaire et pour cinq essais dans le cas avec le mécanisme de vieillissement cellulaire.

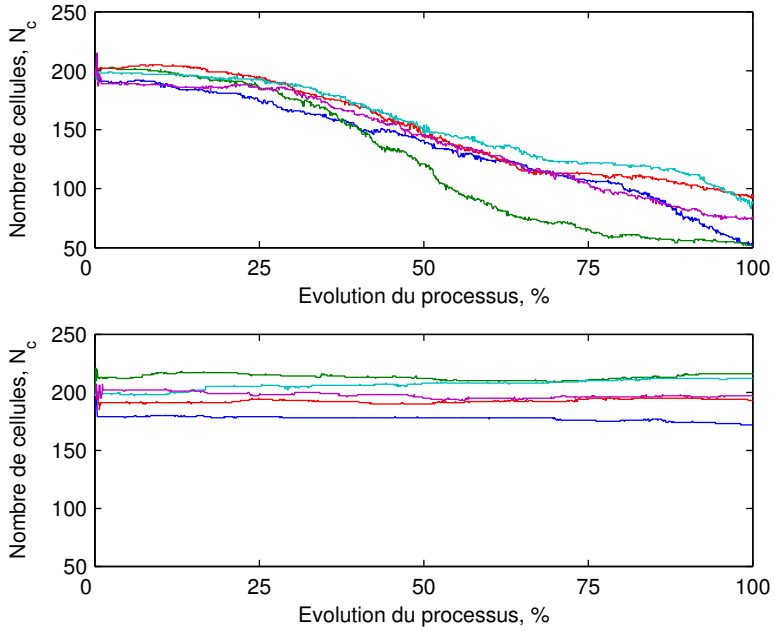

Figure 6.30 – Évolution du nombre de cellules au cours du processus d'optimisation avec le vieillissement cellulaire (haut) et sans le vieillissement cellulaire (bas) pour les cinq essais (20000 évaluations – 10 individus)

Pour le premier cas, le nombre de cellules de Voronoï reste quasi

constant tout au long du processus d'optimisation. Dans le second cas, le nombre de cellules de Voronoï reste approximativement constant jusqu'à 20% du processus d'optimisation, le mécanisme de vieillissement cellulaire n'étant pas encore activé par le jeu des probabilités. Ensuite le nombre de cellules diminue jusqu'à des valeurs allant de 50 à 90 pour l'ensemble des essais.

Le graphique de la Figure 6.31 montre en détail l'évolution des cellules internes et frontalières pour un cas avec vieillissement cellulaire (Figure 6.31 - haut) et un cas sans vieillissement cellulaire (Figure 6.31 - bas).

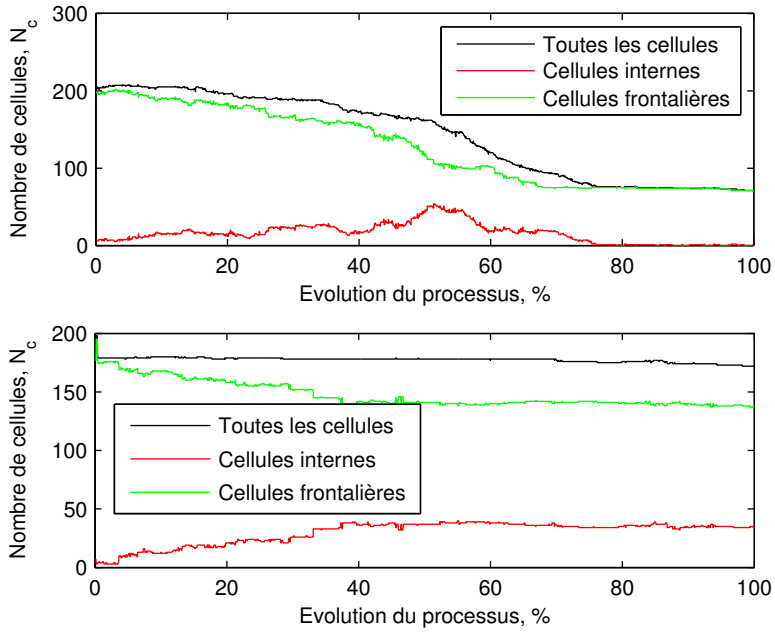

Figure 6.31 – Exemple d'évolution du nombre de cellules au cours du processus d'optimisation avec le vieillissement cellulaire (haut) et sans vieillissement cellulaire (bas) – (20000 évaluations ; 10 individus)

Pour le cas avec le mécanisme de vieillissement cellulaire, on observe que le nombre de cellules internes augmente légèrement pour ensuite tendre vers 0 et que le nombre de cellules frontalières diminue pour ensuite tendre vers une valeur non nulle. Cette forte diminution du nombre total de cellules de Voronoï est la combinaison de deux actions. La pre-

mière concerne le nombre de cellules frontalières qui se transforment en cellules internes. Effectivement, la distribution initiale de matière étant générée aléatoirement, presque l'ensemble des cellules de Voronoï sont frontalières au début du processus d'optimisation. Secondement, le vieillissement cellulaire intervient peu à peu pour éliminer les cellules internes qui présentent aucune utilité vu leur position dans la distribution de matière.

Pour le cas sans mécanisme de vieillissement cellulaire, l'évolution du nombre de cellules est illustrée par la Figure 6.31. Le nombre de cellules internes augmente et le nombre de cellules frontalières diminue, mais le nombre total de cellules reste approximativement constant. Cette stabilisation du nombre de cellules internes et de cellules frontalières implique qu'il y a peu de cellules internes qui viennent augmenter le nombre de degrés de liberté pour la forme et ses dimensions par suite du déplacement de leur centre de Voronoï.

On peut vérifier la différence du nombre de cellules sur les distributions de matière obtenues illustrées par les graphiques de la Figure 6.32. Pour le cas de l'outil avec le vieillissement cellulaire, le nombre de degrés de liberté est concentré autour des frontières. Il n'y a que très peu de cellules internes. Au contraire, le cas de référence possède de nombreuses cellules internes qu'il est évidemment inutile de bouger par de petits déplacements.

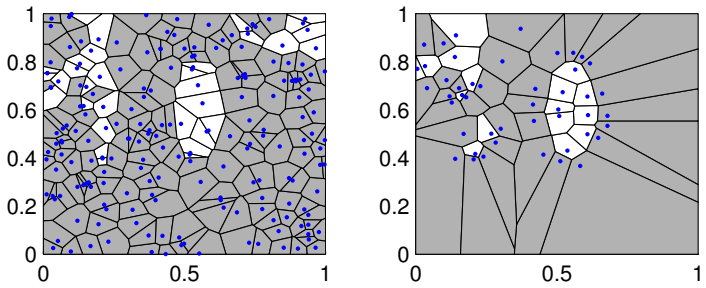

Figure 6.32 – Distribution de matières de la meilleure solution obtenue pour le cas de référence (gauche) et le cas avec le vieillissement cellulaire (droite)

Au-delà de l'avantage d'améliorer la convergence du processus d'optimisation, la diminution du nombre de cellules par vieillissement cellulaire implique aussi une diminution du temps d'évaluation d'un indi-

vidu. Pour les résultats obtenus, on mesure jusqu'à un gain de 50% pour les dernières centaines de générations du processus d'optimisation. Les temps d'exécution des essais, obtenus avec quatre cœurs d'un PC 3.Ghz 8-cœurs, sont repris à la Table 6.10

N° essai	Sans	Avec
1	$22H05$	$16H13$
2	$23H22$	$12H12$
3	$24H13$	$13H38$
4	$17H08$	$13H19$
5	$22H59$	$12H35$

Table 6.10 – Temps d'exécution des cinq essais sans vieillissement cellulaire et des cinq essais avec vieillissement cellulaire

6.2.2.6 Prolifération cellulaire

Tout au long du processus d'optimisation topologique, les distributions de matières sont décrites au travers des cellules frontalières et pas des cellules internes. Or, il est possible que les cellules frontalières ne soient pas en nombre suffisant pour décrire assez finement la forme d'une topologie.

Les mécanismes classiques de reproduction permettent d'optimiser la forme et les dimensions de celle-ci non seulement en agençant correctement les centres de Voronoï mais aussi en migrant des centres de Voronoï vers les zones frontalières. Afin d'aider cette évolution, nous proposons un mécanisme de prolifération cellulaire consistant à ajouter des cellules dans les zones frontalières des domaines de matières.

Description

A l'opposé du vieillissement cellulaire, ce sont cette fois les cellules frontalières qui se voient dotés d'un âge représentatif de leur temps de présence sur une frontière. Pour la suite du manuscrit, ce coefficient est noté α_i pour l'âge de la cellule i.

La méthode proposée permet à l'outil d'optimisation d'ajouter des cellules le long de la frontière entre sous-domaines de matériau différent. L'ajout d'une cellule est faite en fonction des liens de voisinage

des cellules frontalières. Chaque nouvelle cellule ajoutée est insérée aléatoirement sur le segment reliant les centres de deux cellules frontalières adjacentes (Figure 6.33) et chaque couple a la possibilité de générer une nouvelle cellule, ce qui implique que chaque cellule frontalière peut générer autant de cellules qu'elle possède de cellules voisines frontalières.

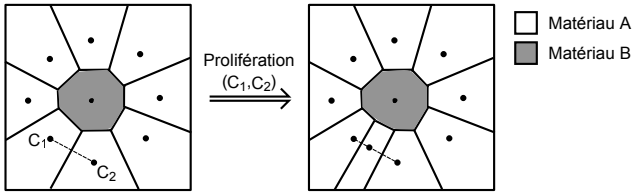

Figure 6.33 – Exemple de mutation par prolifération cellulaire pour deux cellules frontalières adjacentes A–B

La configuration de la reproduction utilisée est illustrée par le schéma de la Figure 6.34. La prolifération est positionnée après le croisement afin d'ajouter des cellules avant que les mutations de position et de matériaux changent la configuration des cellules.

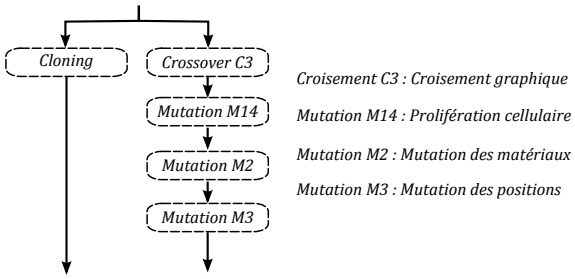

Figure 6.34 – Schéma de reproduction suivi par l'outil d'optimisation topologique étudié

Par la suite, la notation (i, j) fait référence au couple composé de la cellule i et de la cellule j et appartenant à l'ensemble $\mathscr{V}$ des couples de cellules frontalières adjacentes. Une probabilité $p_p^{(i,j)}$ est calculée afin d'ajouter un nombre contrôlé de cellules par génération pour l'ensemble des couples de cellules frontalières. Sachant que l'espérance d'une somme est égale à la somme des espérances,

$$E(X) = E(X_0 + \ldots + X_{N-1}) = E(X_0) + \ldots + E(X_{N-1}), \quad (6.28)$$

pour obtenir une seule cellule ajoutée, la somme des espérances doit être égale à 1. L'espérance d'un état d'une variable est le produit de la probabilité d'atteindre cet état de la variable avec la valeur de l'état de la variable. Ce qui nous donne pour l'espérance de chaque couple :

$$E\left(X^{(i,j)}\right) = \sum_{k=1}^{N_{état}} x_k p_{p,k}^{(i,j)'} \quad (6.29)$$

Puisque chaque couple n'a que deux états controlés par une seule probabilité, on a $\vec{x} = \{1, 0\}$ et $\vec{p_p}^{(i,j)'} = \left(p_p^{(i,j)}, 1 - p_p^{(i,j)}\right)$. Ceci permet d'écrire que

$$E\left(X^{(i,j)}\right) = 1 p_p^{(i,j)} + 0\left(1 - p_p^{(i,j)}\right) = p_p^{(i,j)} \quad (6.30)$$

Ainsi donc l'espérance totale est donnée par la somme des probabilités de mutation de chaque couple de cellules voisine,

$$E(X) = \sum_{(i,j)\in\mathscr{V}} p_p^{(i,j)} = \delta(k) \quad (6.31)$$

où $\delta(k)$ est définie comme une fonction dépendant du nombre d'évaluations que l'on choisit de faire varier entre 0 et 1 dans notre cas. Cette fonction permet de contrôler le taux de mutation par prolifération. Dans notre cas, nous choisissons la loi suivante pour tous les essais afin de minimiser l'effet de prolifération au début du processus d'optimisation :

$$\delta(k) = \left(\frac{k}{k_{max}}\right)^4 \quad (6.32)$$

Pour tenir compte de l'ancienneté de chaque cellule sur la frontière, la probabilité $p_p^{(i,j)}$ de chaque couple est proportionnelle à la somme de l'âge des deux cellules du couple :

$$p_p^{(i,j)} = K(\alpha_i + \alpha_j) \quad (6.33)$$

où K est un facteur de correction ajouté pour les besoins d'adaptation de l'espérance. Avec (6.31) et (6.33), on obtient la relation suivante :

$$\sum_{(i,j)\in\mathscr{V}} p_p^{(i,j)} = \sum_{(i,j)\in\mathscr{V}} K\left(\alpha_{p,i} + \alpha_{p,j}\right) = \delta\left(k\right), \tag{6.34}$$

ce qui permet de déterminer la valeur du coefficient de correction :

$$K = \delta\left(k\right) \frac{1}{\sum_{(i,j)\in\mathscr{V}} \left(\alpha_{p,i} + \alpha_{p,j}\right)} \tag{6.35}$$

et finalement, le calcul de la probabilité de chaque couple de cellule frontalière devient :

$$p_p^{(i,j)} = \delta\left(k\right) \frac{\left(\alpha_{p,i} + \alpha_{p,j}\right)}{\sum_{(m,n)\in\mathscr{V}} \left(\alpha_{p,m} + \alpha_{p,n}\right)}. \tag{6.36}$$

Description des essais

Trois cas différents sont étudiés afin de marquer les gains que peuvent apporter cette mutation par prolifération cellulaire.

Pour le premier cas, la prolifération cellulaire est appliquée dès le début du processus d'optimisation. La population est initialisé avec 10 cellules afin d'observer l'influence du mécanisme de prolifération cellulaire lorsque le nombre de cellules est trop petit.

Pour le deuxième cas, la prolifération cellulaire n'est appliquée qu'à un stade avancé de l'optimisation avec 10 cellules initiales (Figure 6.35), lorsque la topologie optimale peut être supposée avoir été découverte et que l'on veut apporter des degrés de liberté pour continuer l'optimisation de forme et de dimension.

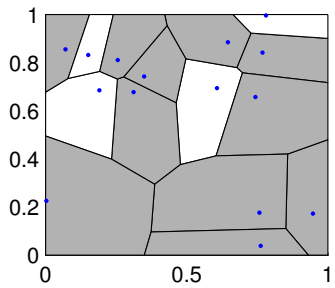

Figure 6.35 – Distribution de matière du meilleur individu de la population initiale – (10.000 évaluation ; 10 individus)

Et finalement, pour le troisième cas, la prolifération cellulaire est appliquée dès le début du processus d'optimisation avec une population initialisée avec 50 cellules.

Résultats cas 1

Pour ce cas, la prolifération est activée dès le début du processus d'optimisation afin d'observer l'influence de cette mutation particulière sur toute la durée du processus d'optimisation. Les essais ont été effectués avec 10.000 évaluations. La population est composée de 10 individus qui sont initialement décrits par 10 cellules. L'ensemble est comparé au cas de référence ne reprenant pas la mutation par prolifération cellulaire.

La Figure 6.36 qui représente les courbes de convergence des essais pour l'outil de référence pour l'outil avec la prolifération montre une grande variation dans les résultats pour les essais avec la prolifération. En début d'optimisation, n'étant pas influencé par le mécanisme de prolifération grâce au coefficient $\delta(k)$ qui a à ce moment une très faible valeur en raison de la loi choisie pour son évolution (7.1), la population évolue à l'identique de celle avec l'outil de référence. Cependant, à partir de 25%, la population évolue plus vite par palier pour le cas avec la prolifération.

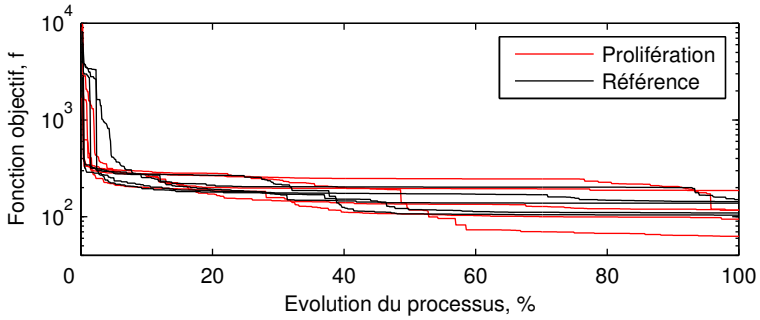

Figure 6.36 – Comparaison de courbes de convergence des essais avec prolifération cellulaire et sans prolifération cellulaire initialisés avec 10 cellules – (10.000 évaluation ; 10 individus)

La variation dans les résultats est mise en évidence dans le Tableau 6.11 où l'on voit que la meilleure solution est obtenue avec la prolifération mais aussi la moins bonne solution.

	Minimum	Médiane	Maximum
Référence	104	137	150
Prolifération	62.75	116.81	187

Table 6.11 – Résultats numériques de cinq essais sans prolifération cellulaire et de cinq essais avec prolifération cellulaire

La différence entre la meilleure performance et la moins bonne performance obtenue passe de 46 pour la méthode de référence à 124.25 pour la méthode avec prolifération, ce qui représente le triple de ce qu'on obtient avec la méthode de référence. Le gain obtenu pour la valeur minimale est proche des 40% et de 15% pour la valeur médiane mais on obtient une augmentation de 24% pour la valeur maximale.

Les deux distributions de matière illustrées à la Figure 6.37 représentent la meilleure solution obtenue pour le cas de l'outil de référence (gauche) et la meilleure solution obtenue pour le cas de l'outil avec la prolifération cellulaire (droite). On observe un plus grand nombre de cellules de Voronoï pour le cas avec la prolifération cellulaire. Cette augmentation du nombre de cellules permet d'obtenir une meilleure description de la topologie, de la forme et des dimensions pour le cas avec la prolifération cellulaire.

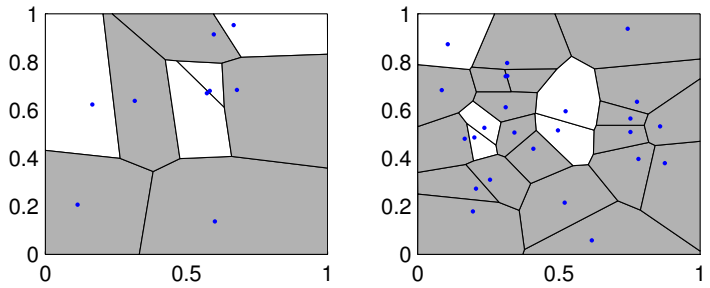

Figure 6.37 – Distribution de matière de la meilleure solution obtenue pour l'outil de référence (gauche) et pour l'outil avec la prolifération cellulaire (droite) – (10.000 évaluation ; 10 individus)

L'augmentation au cours du processus d'optimisation du nombre de cellules de Voronoï pour les 5 essais avec le mécanisme de prolifération cellulaire est illustrée par le graphique de la Figure 6.38. La meilleure

distribution de matière obtenue correspond à la courbe ayant la valeur la plus élevée du nombre de cellules.

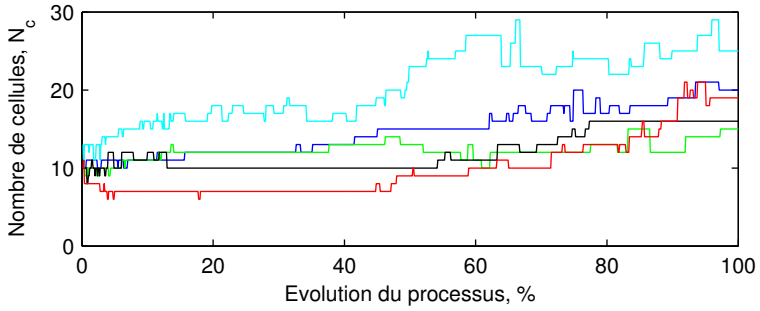

Figure 6.38 – Evolution du nombre de cellules pour 5 essais du cas avec prolifération cellulaires initialisés avec 10 cellules – (10.000 évaluation ; 10 individus)

Résultats cas 2

Pour ce deuxième cas, cinq essais avec prolifération et cinq essais de référence sont comparés en partant chaque fois d'une solution initiale déjà supposée proche de la topologie optimale et obtenue à partir d'un premier processus d'optimisation. Le graphique de la Figure 6.39 compare les courbes de convergence obtenus pour l'ensemble des 10 essais avec 10.000 évaluations et une population de 10 individus. En début d'optimisation, nous observons une situation semblable pour les deux outils, cependant, à partir de 20% du processus d'optimisation, l'approche avec la prolifération cellulaire améliore l'évolution de la population pour les cinq essais. Le meilleur résultat obtenu pour la méthode de référence est moins performant que le moins bon résultat obtenu avec la prolifération cellulaire.

Les données numériques sont résumées dans le Tableau 6.12. La prolifération cellulaire permet d'augmenter légèrement les performances de l'outil comme on a pu l'observer sur le graphique des courbes de convergence.

La Figure 6.40 montre les meilleures distributions de matière obtenues pour les deux outils. La solution de gauche représente la meilleure distribution de matière obtenue pour l'outil de référence. La solution de droite représente la meilleure distribution de matière obtenue pour

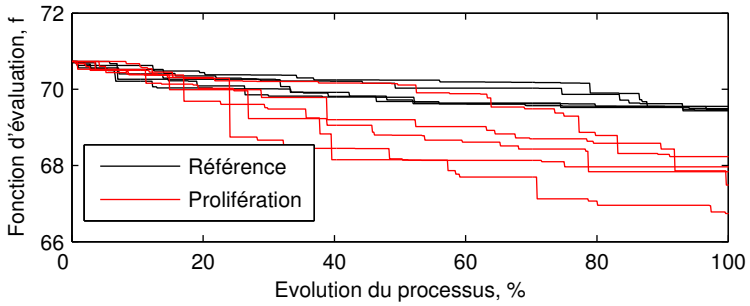

Figure 6.39 – Comparaison de courbes de convergence des essais avec prolifération cellulaire et sans prolifération cellulaire – (10.000 évaluation ; 10 individus)

	Minimum	Médiane	Maximum
Référence	69.43	69.46	69.55
Prolifération	66.73	67.48	68.23

Table 6.12 – Résultats numériques de cinq essais sans prolifération cellulaire et de cinq essais avec prolifération cellulaire avec une performance initiale de 70.73

l'outil avec le mécanisme de prolifération cellulaire. On remarque pour le cas de la prolifération que quelques centres de Voronoï ont été ajoutés au cours du processus d'optimisation pour améliorer la description de la forme du trou ovale.

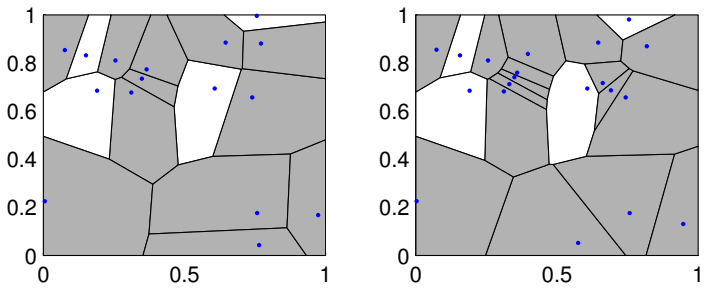

Figure 6.40 – Distribution de matière de la meilleure solution obtenue pour l'outil de référence (gauche), l'outil avec le mécanisme de prolifération (droite) – (10.000 évaluation ; 10 individus)

Résultats cas 3

Cette situation met en jeu des optimisations dont les solutions initiales ont été générées avec 50 cellules de Voronoï. Le processus d'optimisation s'est déroulé sur 20000 évaluations, utilisant 10 individus évoluant sur 2000 générations afin d'observer à plus long terme l'influence de la prolifération avec un plus grand nombre de cellules de Voronoï.

Le graphique de la Figure 6.41 présente les courbes de convergence obtenues pour l'ensemble des essais effectués pour l'outil de référence et pour l'outil avec la prolifération.

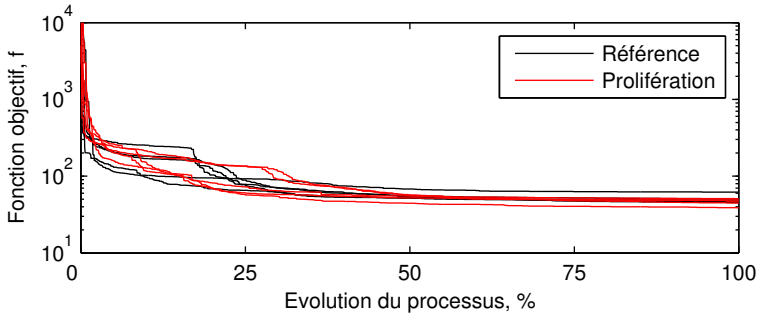

Figure 6.41 – Comparaison de courbes de convergence des essais avec prolifération cellulaire et sans prolifération cellulaire initialisés avec 50 cellules – (20.000 évaluation ; 10 individus)

Les résultats obtenus sont résumées dans le Tableau 6.13. La différence entre la meilleure performance et la moins bonne performance diminue fortement dans le cas de la prolifération cellulaire. De plus, la moins bonne performance obtenue est proche de la valeur médiane obtenue pour le cas de référence. La prolifération cellulaire permet donc d'augmenter les performances de l'outil comme on a pu l'observer sur le graphique des convergences mais avec un coût supplémentaire en temps de calcul pour l'exécution de l'algorithme de la prolifération et pour l'augmentation du nombre de cellules.

Les deux distributions de matière de la Figure 6.42 montrent les meilleures distributions de matière obtenues pour le cas de référence (gauche) et le cas avec la prolifération cellulaire (droite). On observe un plus grand nombre de cellules pour le cas utilisant le mécanisme de prolifération cellulaire que pour l'outil de référence. La topologie obtenue

	Minimum	Médiane	Maximum
Référence	45.28	48.66	61.97
Prolifération	38.62	47.62	50.83

Table 6.13 – Résultats numériques de cinq essais sans prolifération cellulaire et de cinq essais avec prolifération cellulaire

est plus éloignée de la topologie optimale pour le cas de référence que pour le cas avec la prolifération.

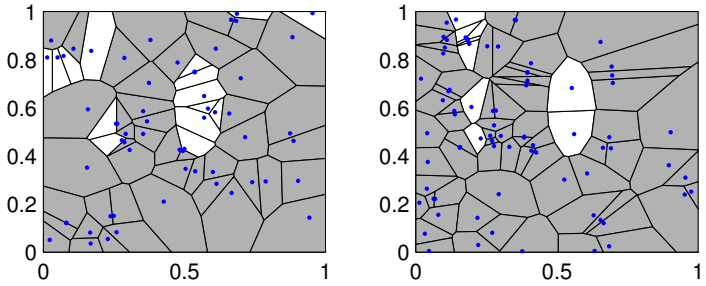

Figure 6.42 – Distribution de matière de la meilleure solution obtenue pour l'outil de référence (gauche) et l'outil avec le mécanisme de prolifération(droite) – (10.000 évaluation ; 10 individus)

La courbes de la Figure 6.43 présentent l'évolution du nombre de cellules de Voronoï au cours du processus d'optimisation pour le cas avec le mécanisme de prolifération cellulaire.

On observe sur le graphique que le nombre de cellules ne varie que très légèrement jusqu'à 40%. A partir de 40% et ce jusqu'à la fin du processus d'optimisation, le nombre de cellules augmente plus considérablement pour quatre essais sur cinq.

6.3 Synthèse

L'outil d'optimisation topologique par distribution de matière basé sur la combinaison de l'algorithme génétique et du formalisme Voronoï offre de nombreuses possibilités d'adaptations au niveau de la gestion de la reproduction. Les différentes évolutions étudiées dans ce chapitre ciblent deux domaines particuliers :

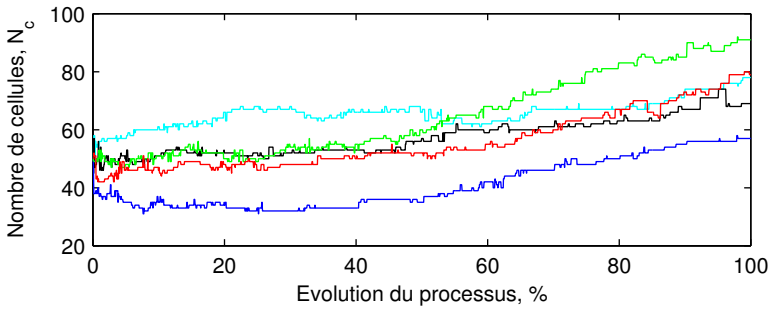

Figure 6.43 – Evolution du nombre de cellules pour 5 essais du cas avec prolifération cellulaire initialisés avec 50 cellules – (20.000 évaluation ; 10 individus)

- Les croisements ;
- Les mutations par modification de la position des centres de Voronoï et par modification du nombre de cellules.

En premier lieu, quatre méthodes de croisement ont été étudiées en Section 6.1. Deux d'entre-elles sont de type graphiques et les deux autres de type non-graphiques. Les résultats obtenus montrent que le croisement graphique à pivot linéaire donne les meilleures solutions par rapport aux autres méthodes, cependant elle est suivie de près par la méthode non graphique non équilibrée.

En deuxième lieu, les déplacements des centres de Voronoï ont été étudiés au travers de trois paramètres : le type de déplacement (Section 6.2.1.2), l'amplitude de déplacement (Section 6.2.1.3) et la direction de déplacement (Section 6.2.1.4).

Pour le type de déplacement, nous avons distingué le déplacement classique du déplacement polaire. Le déplacement classique attribue une probabilité de déplacement à chaque paramètre de position (x, y) de chaque centre de Voronoï, alors que le déplacement polaire attribue deux paramètres qui sont l'angle et l'amplitude, (r, θ). Les résultats finaux sont assez proches entre la méthode classique et la méthode polaire.

Pour l'amplitude de déplacement des centres de Voronoï, nous montrons qu'elle doit varier au cours du processus d'optimisation : des grandes variations au début pour favoriser la phase d'exploration puis des petites dans la phase d'exploitation afin d'éviter de dégrader la solution en modifiant la topologie par de brusques déplacements de cellules.

L'utilisation, pour fixer les déplacements à la génération k, d'un gradient local indiquant dans quelle mesure les déplacements effectués en passant de la génération $k-1$ à la génération k ont amélioré ou non la population, s'avère particulièrement performant.

On a également montré qu'il y avait intérêt à limiter les déplacements des centres de Voronoï aux cellules frontalières entre zones de matière différentes et donc contribuant à la description de la forme et des dimensions d'une solution.

Finalement, le nombre de cellules de Voronoï utilisées pour décrire une solution influence directement le temps d'évaluation d'une solution et la vitesse de convergence du processus d'optimisation. Plus ce nombre est élevé, plus le temps d'évaluation est important et la vitesse de convergence lente.

En début du processus d'optimisation, il faut un nombre suffisant de cellules pour permettre une bonne exploration de l'espace des solutions et obtenir la topologie optimale. A mesure que la solution s'affine, seules les cellules situées aux frontières entre les différents domaines de matières gardent de l'importance et une élimination des cellules internes permet de réduire le nombre de cellules et donc le temps de calcul et la vitesse de convergence. Le vieillissement cellulaire (Section 6.2.2.5) constitue un outil efficace pour effectuer cette élimination.

En phase d'exploration, l'ajout au sein d'un domaine de matière, de cellules comportant un matériau différent permet de créer de nouvelles topologies. En phase d'exploitation, augmenter le nombre de cellules frontalières permet de mieux définir la géométrie des solutions. L'approche utilisée pour gérer la prolifération cellulaire (Section 6.2.2.6) n'a pas encore donné des résultats aussi efficaces que l'élimination par le vieillissement des cellules et devrait être approfondie avant son implémentation.

Chapitre 7

Constitution de l'outil final et application aux cas tests

Le premier objectif de ce chapitre est de mettre à profit les études effectuées aux chapitres 5 et 6 au niveau de la gestion de la population et de la reproduction de l'algorithme génétique pour finaliser l'outil d'optimisation topologique métaheuristique que nous proposons dans le cadre de cette thèse.

Le deuxième objectif du chapitre est de tester et comparer l'outil ainsi considéré à deux autres outils d'optimisation topologique sur base de deux cas d'étude de complexité industrielle en vue de mettre en évidence ses performances.

7.1 Constitution de l'outil final

Le choix des différents composants de l'outil d'optimisation topologique est décrit au travers des deux sous-sections suivantes. La première sous-section propose une stratégie de gestion de la population et la seconde sous-section discute de la manière d'utiliser au mieux les générateurs et les mutateurs.

7.1.1 Population

Dans le Chapitre 5, nous avons montré que le choix de la stratégie de gestion de la population est un facteur important, que ce soit pour l'initialisation de celle-ci ou pour l'évolution de sa taille.

Le schéma du fonctionnement de l'algorithme génétique est illustré à la Figure 7.1. Pour l'initialisation de la population, afin de garantir une exploration de l'espace des solutions, la population est définie par une génération aléatoire des variables d'optimisation de la population. Afin d'accentuer les capacités d'exploration de l'espace de design, il est nécessaire d'utiliser une taille de population importante. A nombre donné d'évaluations, plus la taille de la population augmente, plus le nombre de générations diminue, ce qui réduit la phase d'exploitation. Pour pallier à ce problème, la taille de la population est régulée grâce à la méthode de régulation par rendement.

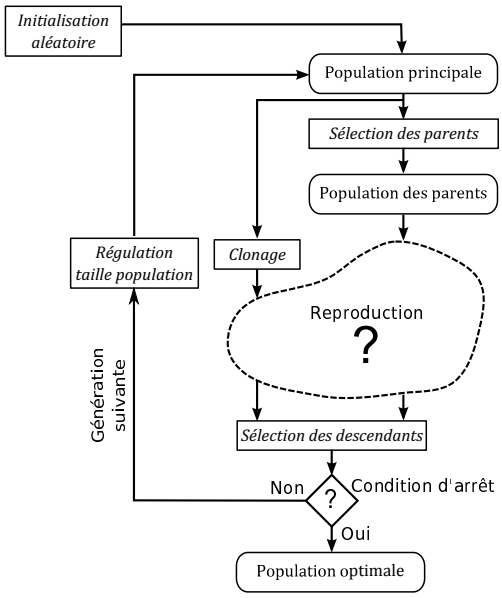

Figure 7.1 – Description partielle du schéma de fonctionnement de l'algorithme génétique intégrant l'initialisation de la population (Section 5.2), le clonage de la population principale (Section 4.1.2.1), la sélection des parents et des descendants (Section 4.1.2.1) ainsi que le module de régulation de la taille de la population (Section 5.1.2)

La régulation de la taille de la population permet de prendre en compte la capacité de la population à être améliorée. La taille de la population est réduite en fonction du nombre d'enfants présentant une meilleure performance que le meilleur individu de la population principale. Lorsqu'aucune amélioration n'est observée, la taille de la popula-

tion est réévaluée à la hausse en fonction d'un paramètre de croissance déterminé par l'utilisateur.

7.1.2 Reproduction

Mis à part le choix initial de cloner la population principale afin de préserver les meilleurs individus et d'aider l'algorithme à converger, la reproduction se résume à un générateur suivi de plusieurs mutateurs. Le schéma de la Figure 7.2 présente la structure utilisée pour la reproduction. Le choix et la description de chacun des opérateurs utilisés sont donnés dans les sous-sections suivantes en fonction de la branche de reproduction.

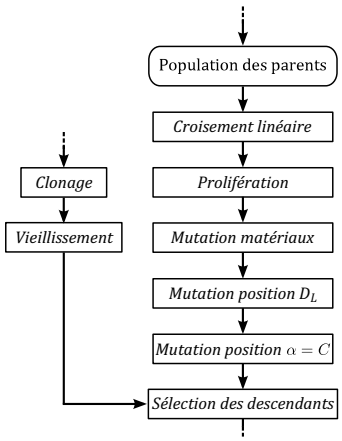

Figure 7.2 – Schéma de la reproduction divisée en deux branches de reproduction utilisant le vieillissement cellulaire (Section 6.2.2.5), le croisement linéaire (Section 6.1.1.3), la prolifération (Section 6.2.2.6), la mutation des matériaux (Section 4.1.2.1) et la mutation de la position des centres de Voronoï de type dynamique locale et à amplitude constante (Section 6.2.1.3)

7.1.2.1 Branche de reproduction par clonage

La branche de reproduction par clonage est la branche qui va conserver et mettre à jour certaines caractéristiques de chaque individu de la population. Le clonage consiste à copier chacun des membres de la population principale pour créer une nouvelle population d'enfants. Le

vieillissement cellulaire est ensuite appliqué afin de forcer la réduction du nombre de cellules inutiles à la description de la solution.

7.1.2.2 Branche de reproduction par croisement

La branche de reproduction par croisement est la branche active de l'étape de la reproduction. Suite aux résultats obtenus en section 6.1, la première action consiste à créer N_s nouveaux individus grâce au croisement graphique à pivot linéaire avec un taux de croisement fixé à 90%.

Quatre mutateurs sont ensuite appliqués pour modifier les individus obtenus.

Le premier mutateur est la mutation par prolifération cellulaire. Il permet d'augmenter le nombre de cellules frontalières afin d'aider à mieux décrire la forme d'un individu. En fonction du taux de mutation p_p, les cellules frontalières auront tendance ou non à générer de nouvelles cellules. Le taux de mutation de la prolifération cellulaire doit tenir compte de l'importance du rôle joué par ce mutateur. Celui-ci ne doit pas intervenir en début de processus d'optimisation, le risque étant de voir un renforcement de sous-domaines de matière devant être normalement éliminés. Afin de limiter ce risque, le taux de mutation est limité en fonction de la loi suivante :

$$p'_p = p_p \left(\frac{k}{k_{max}} \right)^4 . \tag{7.1}$$

Le deuxième mutateur est la mutation des matériaux. Le taux de mutation p_m permet de déterminer la probabilité qu'à chaque cellule de modifier le matériau lui étant associé. Le taux de mutation des matériaux est initialement fixé. Puisque le nombre de cellules devrait diminuer grâce au vieillissement, son influence devrait décroître au cours du processus d'optimisation. Un taux de mutation légèrement élevé, $p_m = {}^3/_{N_c}$ avec N_c le nombre initial de cellules, permet d'augmenter la phase d'exploration.

Les troisième et quatrième mutateurs sont la mutation de la position des centres de Voronoï de type dynamique locale et la mutation à amplitude constante. La combinaison des deux permet de travailler avec de fortes amplitudes en début de processus d'optimisation pour la phase d'exploration puis de travailler avec des amplitudes maximales calculées

pour chaque cellule en fonction de la distance avec ses cellules voisines. Un taux de mutation de déplacement p_d, identique pour les deux mutations, détermine la probabilité que chaque centre de Voronoï change de position.

Puisque il est possible que les deux mutateurs de position effectuent plusieurs mutations sur les mêmes variables d'optimisation, il est dès lors utile de limiter la probabilité de mutation et ainsi éviter l'un des deux effets suivants :

- Soit les perturbations s'additionnent et accentuent l'amplitude de la perturbation ;
- Soit les perturbations s'annihilent annulant la mutation et diminuant virtuellement le taux de mutation.

L'application des deux mutations de la position des centres de Voronoï va prendre en compte le fait qu'il est nécessaire de les faire intervenir sur des périodes différentes du processus d'optimisation. La mutation avec une amplitude constante (section 6.2.1.3) est utilisée avec un coefficient $\alpha = 100\%$ afin de forcer l'outil à explorer l'espace des solutions. Cette phase sera courte, 30% du nombre d'évaluations, et passera ensuite le relais à la mutation polaire dynamique locale (section 6.2.1.3).

7.2 Description des outils de référence

Deux outils d'optimisation topologique de référence sont utilisés pour tester l'outil final d'optimisation topologique constitué dans cette dernière étape. Le premier outil, utilisé lors du premier cas d'étude de ce chapitre (décrit en Section 3.2.3.1), est l'outil proposé par M. Schoenauer [36] et correspond à l'outil de référence utilisé comme base de comparaison en deuxième étape avec, en plus, un mécanisme de croisement graphique. Cet outil est l'un des derniers à avoir été publié en proposant d'utiliser le formalisme de Voronoï comme formalisme de distribution de matière. Le deuxième outil, utilisé lors du second cas d'étude de ce chapitre (décrit en Section 3.2.3.2), correspond à celui proposé par T. Ishikawa [67]. Les résultats récents obtenus par cet outil sur un problème électromagnétique en font un bon exemple de comparaison.

7.2.1 Description de l'outil de Schoenauer

L'outil de Schoenauer est basé sur la combinaison de l'algorithme génétique et du formalisme Voronoï. La sélection des parents et la sélection des descendants sont respectivement effectuées par la méthode du tournoi et par la méthode élitiste. Le mécanisme de reproduction de l'outil est illustré par le diagramme de la Figure 7.3. Chaque enfant est créé par le croisement graphique à pivot linéaire présenté au Chapitre 4. Les individus sont ensuite modifiés par une mutation classique consistant à changer aléatoirement indépendamment chaque variable décrivant la position des centres de Voronoï ou le matériau associé en fonction d'un taux de mutation.

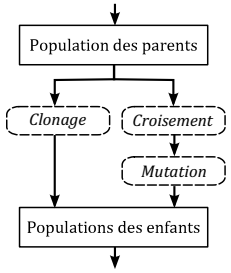

Figure 7.3 – Diagramme de reproduction de l'outil de Schoenauer

7.2.2 Description de l'outil de Ishikawa

L'outil de Ishikawa est composé de l'algorithme génétique et du formalisme Bit-Array. Au travers de cet outil, T. Ishikawa propose deux modifications majeures de l'algorithme génétique afin de prendre en compte l'aspect graphique des problèmes d'optimisation topologique. La première adaptation permet de modifier la taille du vecteur de variables d'optimisation. La seconde adaptation permet de filtrer les groupes de cellules représentant des sous-domaines dont la surface est plus petite qu'un paramètre choisi par l'utilisateur. Le diagramme de fonctionnement de cet algorithme est montré à la Figure 7.4.

Pour réaliser la modification de la taille du vecteur de variables d'optimisation, T. Ishikawa propose de travailler sur plusieurs cycles. A la fin de chaque cycle, chaque cellule est décomposée en quatre nouvelles cellules ayant une probabilité p_r de récupérer le matériau de l'ancienne

cellule. Pour reconstituer la nouvelle population, seule la distribution de matière du meilleur individu est utilisée. Après avoir généré cette nouvelle population initiale, le processus d'optimisation de l'algorithme génétique est relancé. Cette opération est effectuée N_{cycle} fois où N_{cycle} est un paramètre de contrôle que l'utilisateur doit déterminer.

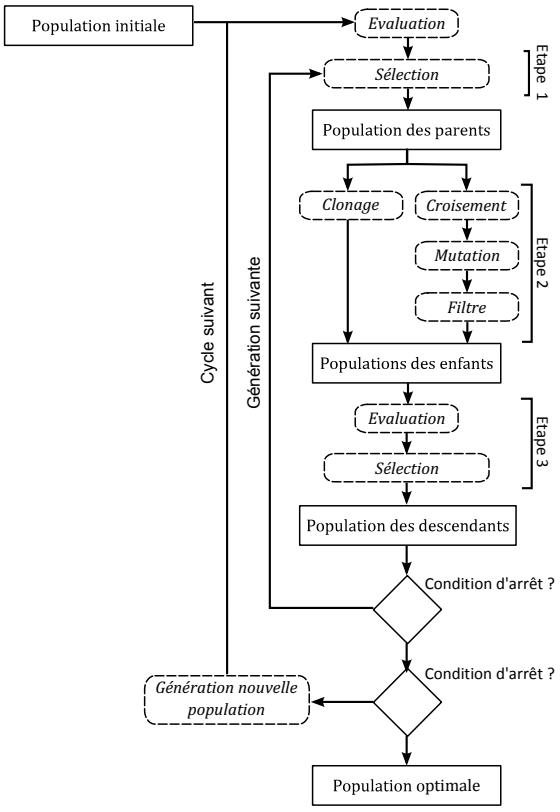

Figure 7.4 – Diagramme de l'outil de Schoenauer

Pour filtrer les sous-domaines ayant une trop petite surface, T. Ishikawa propose d'utiliser une mutation ayant une probabilité p_f de supprimer les sous-domaines de matière en fonction de leur taille.

7.3 Cas d'étude 1 : le moteur linéaire à réluctance variable

Le premier cas d'étude est un actionneur linéaire plan à réluctance variable décrit plus en détail en section 3.2.3.1. L'actionneur est composé d'un stator et d'un mouveur. La distribution de matière du stator est obtenue avec deux matériaux : le cuivre et le fer. Celle du mouveur est aussi obtenue avec deux matériaux : le fer et l'air. L'espace de design est dissocié en deux parties correspondant au stator et au mouveur. Un espace d'air entre les deux parties permet le déplacement mécanique du mouveur. Le calcul éléments-finis prend en compte la saturation magnétique du fer.

7.3.1 Résultats

Les paramètres utilisés pour l'outil d'optimisation de référence et l'outil final sont identiques. Ces deux tests sont effectués avec 50000 évaluations et la population est initialement composée de 300 individus, chacun décrit par 200 cellules de Voronoï. Ces 200 cellules sont réparties équitablement entre les deux espaces de design. La comparaison des résultats se fait au travers de la qualité de la topologie, de la forme et de ses dimensions, ainsi que de l'évolution de la courbe de convergence et finalement du temps d'exécution.

7.3.1.1 Distribution de matière

Les résultats obtenus par les deux outils d'optimisation topologique sont assez proches l'un de l'autre. Ils présentent tout d'abord une qualité de solution assez élevée, montrant une conception fort aboutie. La Figure 7.5 montre, pour les deux outils, les distributions de matière obtenues en position de conjonction et en position d'opposition. Dans les deux cas, la topologie et la forme obtenues au niveau du stator permettent de maximiser le courant dans l'encoche. L'encoche possède une fermeture de forme triangulaire. Cette fermeture permet de diminuer la saturation magnétique localement afin d'éviter des pertes de flux magnétique en position de conjonction. La largeur de cette fermeture permet de créer un chemin de fermeture des lignes de flux magnétique en position d'opposition.

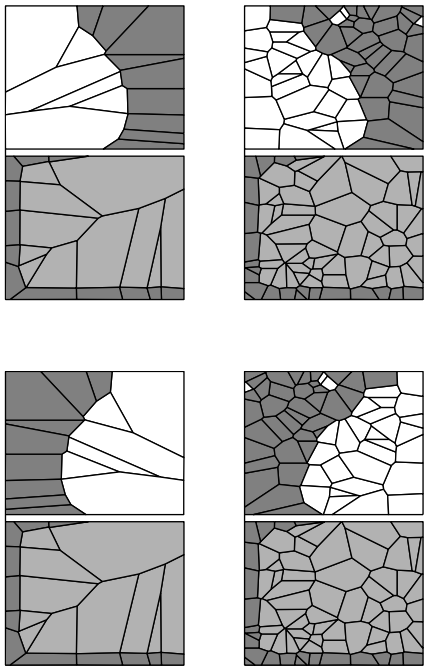

Figure 7.5 – Distributions de matière obtenues pour l'outil de référence (droite) et pour l'outil final (gauche) dans le cas de la position d'opposition (haut) et de conjonction (bas)

En comparant les résultats entre les deux cas, on peut tout d'abord observer que la topologie finale des deux solutions obtenues est très proche que ce soit pour le stator ou pour le mouveur. Cependant, il se trouve deux petits sous-domaines d'air dans le mouveur dans le cas de l'outil de référence qui ne se trouvent pas dans la solution obtenue avec l'outil final.

Ensuite, la forme diffère légèrement entre les deux solutions. Le pied de la dent statorique possède une excroissance fort prononcée dans le cas de l'outil de référence. La description de la frontière est plus aboutie pour le cas de l'outil final.

Finalement, les dimensions entre les deux solutions diffèrent à nouveau légèrement. L'ouverture de l'encoche du stator, l'épaisseur moyenne

de la carcasse en fer et la largeur du corps de la dent du mouveur sont quasi-identiques, sauf pour la largeur du pied de la dent du mouveur. Celle-ci est plus large pour le cas de l'outil de référence.

7.3.1.2 Courbe de convergence

Le graphique de la Figure 7.6 compare les deux courbes de convergence. Cette comparaison montre une différence significative en faveur de l'outil d'optimisation topologique final. La vitesse de convergence pour l'outil final est telle qu'en 7500 évaluations, la fonction objectif dépasse celle obtenue avec 50000 évaluations pour l'outil de référence. Les deux valeurs finales obtenues sont $f = -0.026$ et $f = -0.0256$. Avec beaucoup plus d'évaluations à disposition, l'outil de référence semble capable d'atteindre quasiment le même résultat.

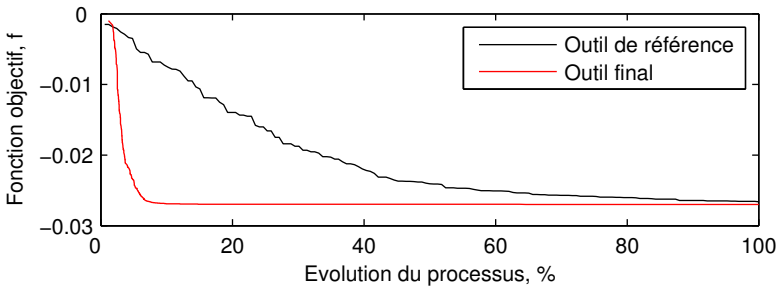

Figure 7.6 – Courbes de convergence pour l'outil d'optimisation topologique de référence et l'outil final d'optimisation topologique

Les schémas de la Figure 7.7 reprennent les distributions de matière obtenues après 25%, 50% et 75% du processus d'optimisation. Celles des résultats de l'outil final ont rapidement évolué vers l'optimum. A 25%, la topologie est déjà très proche de la topologie finale, excepté un trou, alors que celle produite par l'outil de référence en est encore très loin. Ce n'est qu'au-delà de 75% que l'outil de référence semble avoir « identifié » la topologie finale, mais avec encore de nombreux trous de matière qui ne seront pas totalement comblés à 100% du processus d'optimisation.

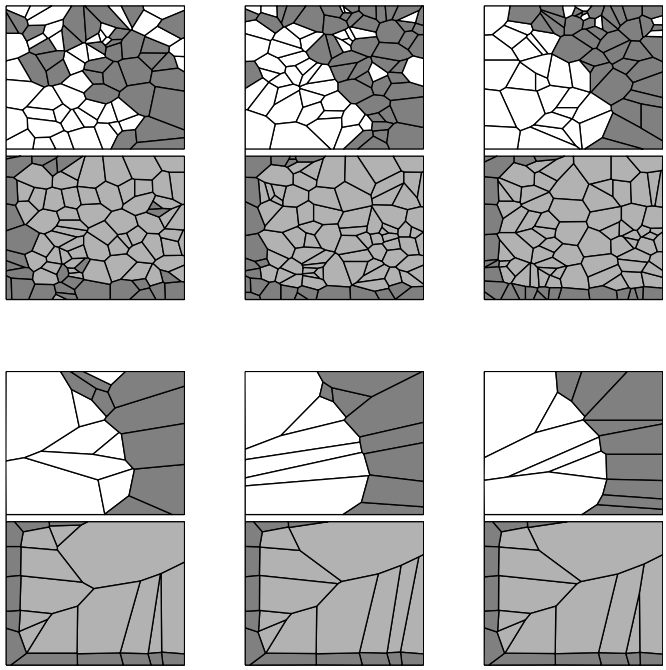

Figure 7.7 – Distributions de matière obtenues après 25% (gauche), 50% (milieu) et 75% (droite) du processus d'optimisation pour l'outil de référence (haut) et l'outil final (bas)

7.3.1.3 Temps d'exécution

Comme on peut l'observer sur les distributions de matière obtenues présentées à la Figure 7.7, le résultat obtenu avec l'outil final ne possède plus de cellules internes. L'outil est parvenu à éliminer celles-ci pour ne conserver que les cellules frontalières. Cette suppression de cellules, et donc la diminution du nombre de degrés de liberté résultante, n'a pas empêché l'outil final de retrouver le même optimum que l'outil de référence. De plus, la diminution du nombre de cellules n'a pas empêché l'outil de décrire avec suffisamment de précision la solution. Cette évolution du nombre de cellules est représentée par les courbes de la Figure 7.8. Le nombre de cellules pour le cas de l'outil final diminue régulièrement pour atteindre une valeur minimale.

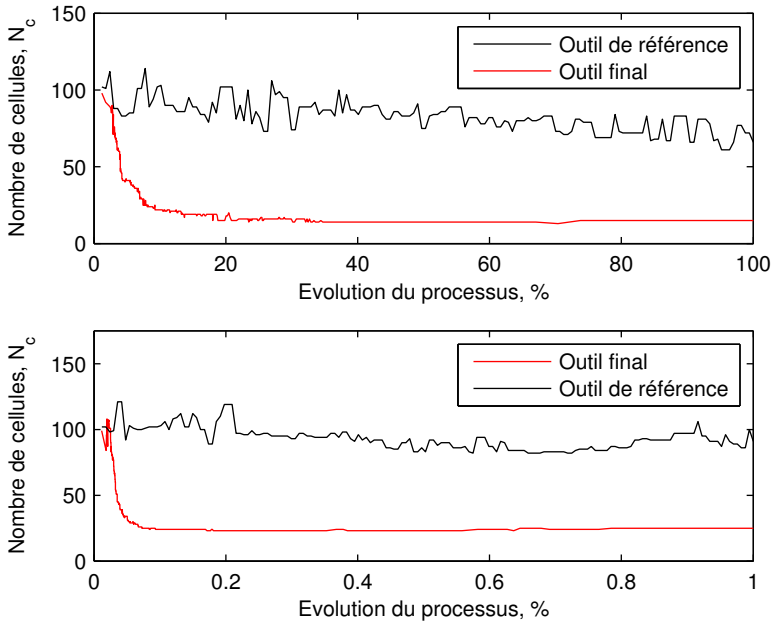

Figure 7.8 – Évolution du nombre de cellules dans les espaces de design 1 (haut) et 2 (bas)

L'évolution du nombre de cellules, et donc principalement sa diminution par l'élimination des cellules internes, permet à l'outil d'augmenter son temps de travail sur l'optimisation de la forme et de diminuer le temps d'évaluations des individus. Le tableau 7.1 reprends les temps moyens des évaluations pour le cas de l'outil de référence et le cas de l'outil final. On remarque immédiatement que le temps d'évaluation moyen est de 19.0 secondes contre 25.0 secondes pour l'outil de référence. Ce gain de temps considérable est un des points forts de l'utilisation du vieillissement cellulaire. Toutefois, les valeurs obtenues dépendent aussi d'autres facteurs importants. Par exemple, pour résoudre un problème magnétique avec possibilité de saturation des matériaux, il est nécessaire d'utiliser des méthodes itératives. Or ces méthodes itératives ne garantissent pas un temps de résolution fixe ou linéaire. La variation du temps d'évaluation d'une solution dépend donc aussi de l'état de la solution même (saturation des matériaux, complexité du maillage, ...).

Table 7.1 – Temps moyen, exprimé en secondes, de l'évaluation de l'individu le plus performant à différentes périodes du processus d'optimisation sur un PC 3.3GHz comprenant la construction du modèle élément finis et de l'évaluation de ce dernier.

Période	25%	50%	75%	100%
Référence	30.1	23.2	23.8	24.3
Final	18.6	18.9	19.0	19.0

7.3.2 Discussions

Bien que les résultats soient assez nettement en faveur de l'utilisation de l'outil d'optimisation topologique final, il est nécessaire de faire une analyse pour distinguer parmi les différents opérateurs utilisés ceux qui ont plus ou moins bien rempli leur fonction. Comme signalé en introduction sur la constitution de l'algorithme d'optimisation, chaque opérateur évolué a été étudié indépendamment des autres. Le comportement de chacun en présence des autres n'est dès lors pas connu.

La différence de vitesse de convergence entre l'outil de référence et l'outil évolué est fortement due à la régulation de la taille de la population. Comme on peut le voir à la Figure 7.9, la population décroit en effet rapidement pour atteindre des valeurs très faibles. L'outil ayant rapidement convergé vers la topologie finale, il nécessite moins de diversité dans la population. Il peut donc diminuer la taille de la population et ainsi augmenter le nombre d'itérations pour permettre une plus grande phase d'exploitation.

L'outil final est non seulement plus efficace en exploration, puisque la topologie finale est très rapidement trouvée pour ce cas d'application, mais est aussi plus rapide sur base du temps. Le nombre de cellules diminuant rapidement pour permettre à l'outil de se concentrer sur les cellules frontalières, l'évaluation de chaque solution tombe jusqu'à 18.6 secondes.

Cependant, on remarque que la méthode de régulation peut aussi avoir le défaut de ne pas limiter l'évolution de la taille de la population en fin de processus. A ce moment, en effet, l'outil ayant plus de difficultés pour faire évoluer le meilleur individu, la taille de la population est régulièrement augmentée de 10% pour atteindre des valeurs élevées. Cette augmentation peut amener à une diminution de performance de

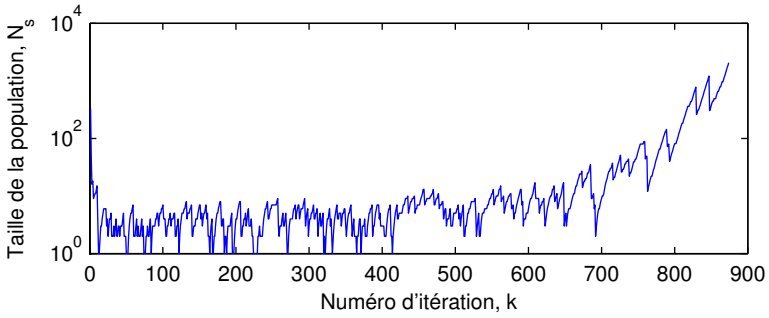

Figure 7.9 – Évolution de la taille de la population au cours du processus d'optimisation

certains opérateurs, comme le croisement, ce qui entraîne une diminution de l'efficacité de l'exploitation. On pourrait dès lors être tenté de trouver un mécanisme différent.

La prolifération semble ne pas avoir fonctionné correctement, et peut aussi avoir limité les performances de l'outil. En effet, en observant les distributions matière à des stades différents des processus d'optimisation, on remarque une très faible évolution du nombre de cellules frontalières. Il semble donc que la mutation par prolifération cellulaire utilise un mécanisme trop rigide pour permettre l'ajout de nouvelles cellules. L'ajout de cellules modifie l'individu de manière trop brutale, apportant une grande probabilité de réduire la performance de celui-ci. N'étant pas sélectionné par la suite, l'ajout de la cellule ne peut pas être répercuté dans les générations suivantes.

7.4 Cas d'étude 2 : moteur synchrone

Le cas d'étude repris ici, décrit en détail en section 3.2.3.2, est identique à celui étudié par T. Ishikawa [67]. Les résultats vont donc être confrontés à ceux déjà obtenus avec un algorithme génétique couplé à un formalisme Bit-Array.

Le cas d'étude est une machine synchrone à aimant permanent au rotor, décrite plus en détail en Section 3.2.3.2. Seule la topologie de la partie rotorique est optimisée. Les matériaux utilisés sont le fer et des aimants à aimantation radiale. La partie statorique est assimilée comme étant idéale, c'est-à-dire sans saturation magnétique.

7.4.1 Résultats

Les paramètres utilisés pour ce cas d'étude sont identiques lorsque cela est possible à ceux de l'article [67] sauf pour la taille de la population et le nombre de variables que l'article fixe respectivement à 45 individus et 360 variables maximum. L'optimisation est effectuée avec 40500 évaluations. Pour notre algorithme final, c'est une population initiale de 500 individus qui va être utilisé. Le nombre de cellules est initialement fixé à une valeur élevée, c'est-à-dire 200 cellules soit 600 variables d'optimisation, puisque nous pouvons compter sur le vieillissement cellulaire pour diminuer ce nombre.

Les trois sous-sections montrent la distribution de la matière obtenue avec l'outil final, la courbe de convergence et l'évolution du nombre de cellules.

7.4.1.1 Distribution de matière

Le résultat obtenu diffère de celui obtenu par T. Ishikawa (Figure 7.10-gauche). Comme on peut l'observer sur la Figure 7.10-droite, un bloc de fer s'est intégré dans l'aimant. Toutefois, le résultat présente une bonne qualité de solution, montrant une conception fort aboutie. L'aimant couvre presque l'ensemble de la surface du rotor, laissant une ouverture d'angle proche de 5° au niveau de la jonction avec le pôle à aimantation de sens opposé. A ce niveau, l'épaisseur de l'aimant augmente rapidement pour vite atteindre une épaisseur équivalente à l'entrefer. La topologie se complexifie ensuite pour intégrer le bloc de fer dans l'aimant.

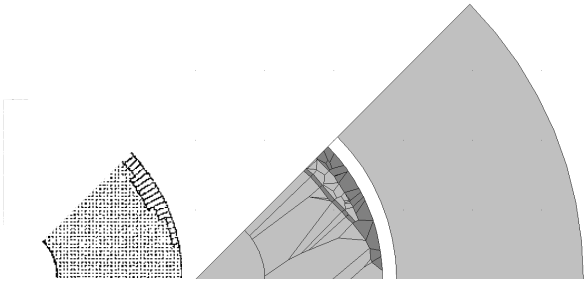

Figure 7.10 – Distribution de matière du rotor du moteur synchrone obtenue par l'outil de Ishikawa [67] (gauche) et l'outil final (droite)

7.4.1.2 Courbe de convergence

Le graphique de la Figure 7.11 représente la courbe de convergence du cas étudié. La population double ses performances sur les 15 premiers pour cents du processus d'optimisation. Au-delà, la population évolue très faiblement pour atteindre l'évaluation finale $f = -4.65$.

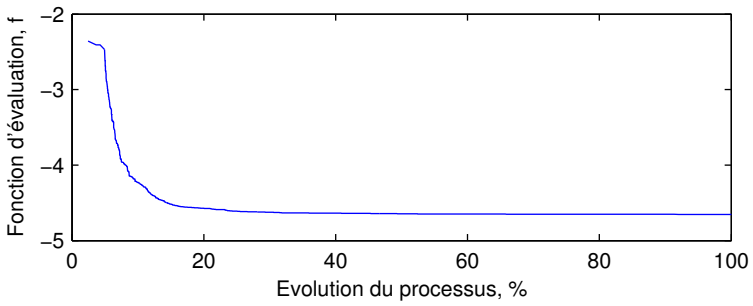

Figure 7.11 – Courbe de convergence pour l'outil d'optimisation topologique final

Les schémas de la Figure 7.12 reprennent les distributions de matière obtenues après 25%, 50% et 75% du processus d'optimisation. Ces schémas montrent que l'outil converge rapidement vers une topologie et est déjà en phase d'exploitation à 25% du processus d'optimisation. Comme on a pu l'observer sur la courbe de convergence en section précédente, la performance de la meilleure solution évolue peu, ce qui se retrouve dans l'évolution de la solution sur plus de 75% du processus d'optimisation.

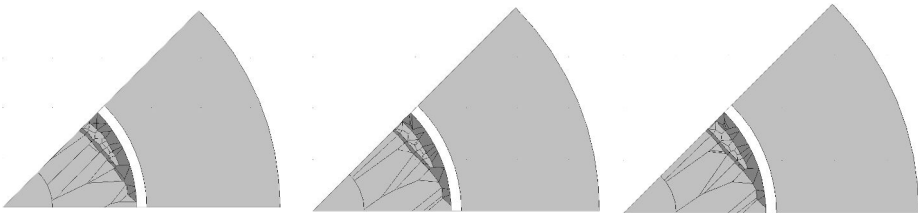

Figure 7.12 – Distributions de matière obtenues après 25% (gauche), 50% (milieu) et 75% du processus d'optimisation (droite)

7.4.1.3 Évolution du nombre de cellules

La Table 7.2 donne le temps moyen en secondes de l'évaluation d'une solution. Le temps d'évaluation diminue de 40% dès les 25 premiers pour cents du processus d'optimisation pour ensuite se stabiliser aux alentours des 3 secondes.

Table 7.2 – Temps moyen, exprimé en secondes, de l'évaluation de l'individu le plus performant à différentes périodes du processus d'optimisation sur un PC 3.3GHz comprenant la construction du modèle élément finis et de son évaluation.

Période	0%	25%	50%	75%
Final	5.48	2.95	3.04	3.05

Comme on peut le remarquer sur les distributions de matière obtenues, l'algorithme a supprimé toutes les cellules internes de l'espace de design. La courbe d'évolution du nombre de cellules au cours du processus d'optimisation donnée à la Figure 7.13 présente une légère augmentation aux alentours des 5% suivie ensuite d'une diminution rapide du nombre de cellules jusqu'à se stabiliser aux alentours de 50 cellules.

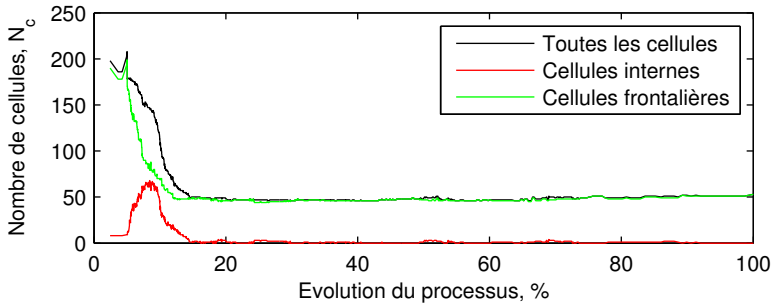

Figure 7.13 – Évolution du nombre de cellules au cours du processus d'optimisation

7.4.2 Discussions

Pour ce cas d'application, la topologie finale diffère du cas de référence obtenu par T. Ishikawa et reprise à la Figure 7.10. Afin de comparer les valeurs numériques, nous avons remplacé le bloc de fer dans

l'aimant de notre solution par de l'aimant. Cette nouvelle solution a ensuite été utilisée comme solution initiale pour un nouvel essai de l'algorithme. Les résultats obtenus montrent que la nouvelle topologie donne de meilleurs résultats que la topologie trouvée avec l'outil final. La Figure 7.14 présente la nouvelle solution avec une évaluation de $f = -4.74$ et qui se rapproche de celle de T. Ishikawa.

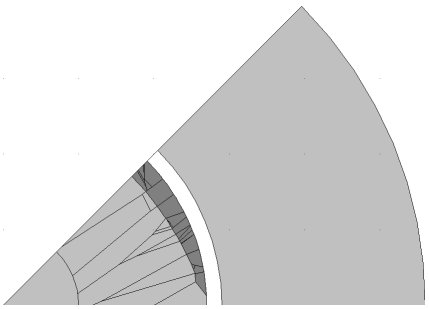

Figure 7.14 – Distribution de matière pour la solution corrigée

La forme obtenue présente une marche d'escalier sur l'extrémité de la forme de l'aimant et l'épaisseur varie ensuite très faiblement.

En réitérant l'opération d'optimisation en utilisant les mêmes paramètres de contrôle que T. Ishikawa, c'est-à-dire avec une population de 45 individus et 360 variables d'optimisation, nous obtenons la distribution de matière présentée à la Figure 7.15 avec une évaluation de $f = -4.75$. La forme est légèrement différente de celle de référence. L'épaisseur de l'aimant évolue régulièrement sur tout l'espace de design.

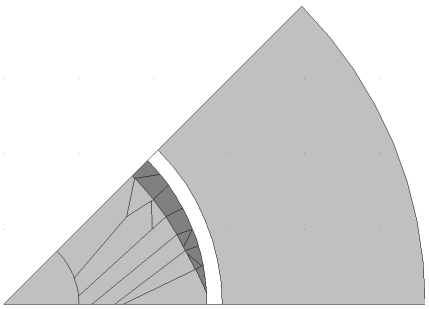

Figure 7.15 – Distribution de matière obtenue par l'outil final avec 45 individus et 120 cellules

Les solutions obtenues ne permettent pas d'éliminer la pulsation du couple. La fonction objectif étant de maximiser le couple du moteur, l'outil d'optimisation n'a pas pris en compte l'amplitude des oscillations du couple dans le cas d'une alimentation des phases par des courants formant un système équilibré de grandeurs sinusoïdale en fonction de la position du rotor. La Figure 7.16 présente le couple pour la meilleure solution obtenue avec la configuration initiale de l'étude de ce cas d'application.

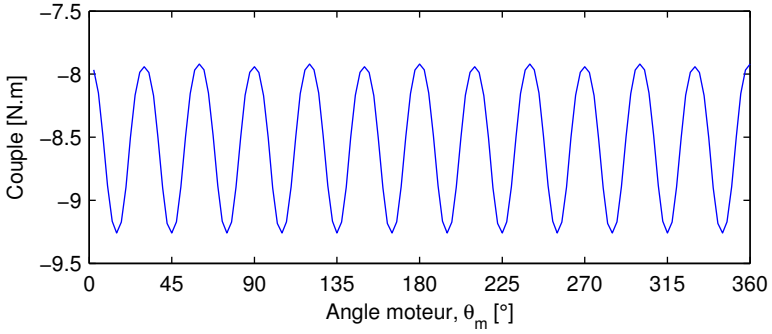

Figure 7.16 – Couple caractéristique du moteur avec la topologie obtenue

Finalement, le choix de la fonction objectif a fortement influencé l'évolution de la population. L'intervention du volume des aimants dans le calcul de la fonction objectif a diminué les possibilités de l'outil de sortir de son optimum local. Le fait de remplacer la partie en fer située dans l'aimant augmente le volume des aimants et augmente un peu le couple de la machine, mais a aussi pour conséquence de diminuer la performance de la solution. L'outil est donc contraint d'améliorer la forme et les dimensions de la solution trouvée sans pouvoir modifier la topologie.

7.5 Synthèse

Dans ce chapitre, nous avons construit, en section 7.1, la composition d'un outil final en fonction des résultats obtenus lors de la phase précédente présentée aux Chapitre 5 et Chapitre 6. Cet outil final a été ensuite utilisé pour optimiser le design de deux dispositifs électromagnétiques :

un actionneur linéaire à réluctance variable et une machine synchrone à aimant permanent. Pour les deux cas, nous avons confronté les résultats obtenus avec ceux obtenus avec des outils de référence, c'est-à-dire l'outil de Schoenauer pour le premier cas d'étude et l'outil de Ishikawa pour le second cas d'étude.

Pour le premier cas d'étude, les résultats obtenus sont favorables à l'emploi de l'outil final (Section 7.3). La topologie ainsi que la forme et les dimensions de la solution obtenue sont à la fois mieux aboutis tout en ayant une meilleure évaluation qu'avec l'outil de Schoenauer avec le même nombre d'évaluations. De plus, la vitesse de convergence est fortement accrue tout en ayant un temps plus petit d'évaluation des solutions. Cependant, on remarque que tous les composants de l'outil final n'ont pas participé à l'amélioration de la performance de l'outil d'optimisation.

Pour le second cas d'étude, les résultats obtenus sont plus mitigés (Section 7.4). En effet, la topologie finale obtenue avec un grand nombre de cellules n'est pas optimale. Bien que nous ayons une forme et des dimensions bien définies, l'outil a convergé vers un optimum local. En effectuant à nouveau l'optimisation avec un nombre plus faible de cellules, les résultats obtenus sont largement meilleurs et la topologie légèrement différente. Cette différence, fortement liée au nombre de cellules et donc au nombre de degrés de liberté, remet en question la robustesse de l'outil vis-à-vis des paramètres et en particulier la taille de la population. Avec un grand nombre de degrés de liberté, la phase d'exploration, insuffisante, se termine sur une topologie non optimale que la phase d'exploitation améliore ensuite autant que possible. Cette amélioration permet d'obtenir une solution aboutie, ce qui n'est pas le cas des résultats obtenus par T. Ishikawa, où l'on a une description de la topologie ayant un crénelage fortement prononcé dû au formalisme choisi.

Au vu des résultats obtenus, à la fois pour le premier cas d'étude et pour le second cas d'étude, l'outil final présente une bonne capacité à finaliser la description d'une solution grâce à sa phase d'exploitation. La phase d'exploration a cependant échoué dans le cas de la seconde application en ne retrouvant pas l'optimum au problème posé. Une évolution trop rapide de la population semble être une des causes principales à ce résultat mitigé.

Chapitre 8

Perspectives

Dans le cadre de cette thèse après, avoir justifié le choix d'un outil d'optimisation combinant un algorithme génétique et une distribution de matière utilisant le formalisme des cellules de Voronoï, nous avons examiné aux chapitres 5, 6 et 7 comment améliorer les performances de cet outil en proposant de nouveaux générateurs et nouveaux mutateurs.

Ce chapitre a pour objectif de rassembler quelques pistes d'améliorations supplémentaires et d'en donner une description, ainsi que quelques résultats le cas échéant. On envisage successivement des compléments relatifs à la génération de la population, aux croisements et aux mutations, aux critères d'évaluation et finalement une hybridation consistant à combiner l'algorithme génétique avec un autre algorithme.

8.1 Population

Lors de l'étude de l'algorithme génétique, le besoin de souplesse dans sa structure a conduit à utiliser le concept de reproductions parallèles. Au travers de cette structure, il est possible d'étudier différents modèles pouvant permettre d'augmenter les performances de l'outil.

8.1.1 Description du parallélisme de la reproduction

Lors de la création des enfants, ceux-ci peuvent être regroupés en plusieurs populations en fonction de l'origine de leur création. Sur base du schéma de la Figure 4.2, la génération de plusieurs populations d'enfants revient à exploiter des branches parallèles de reproduction. Cha-

cune d'elles doit être initialisée par un générateur, suivi le cas échéant par les mutateurs tel que représenté par le schéma de la Figure 8.1.

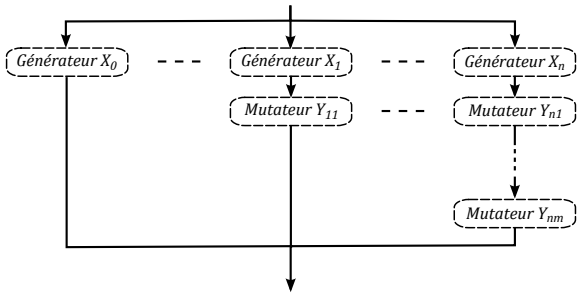

Figure 8.1 – Schéma général des branches de la reproduction

Par exemple, pour un algorithme génétique classique, nous pouvons associer le générateur X_0 à un opérateur de clonage, le générateur X_1 à un opérateur de croisement et le mutateur Y_{11} à un opérateur de mutation classique sur les variables. Ceci donne deux populations d'enfants qui sont ensuite regroupées pour ne former qu'une seule population.

Le principal avantage à ramener la reproduction sous cette structure est de varier le nombre de branches de reproduction pour créer des enfants de manières différentes.

8.1.2 Variation de la taille des populations des enfants

En travaillant avec des branches de reproduction parallèles, il est possible de répartir les ressources de calcul en fonction des différentes branches. Cette répartition se fait au travers des nombres d'enfants $N_{se,i}^{(k)}$ associés à la branche i de reproduction. La contraction de ce vecteur sur l'indice i nous donne la taille totale de la population des enfants :

$$N_{se}^{(k)} = \sum_i N_{se,i}^{(k)} \tag{8.1}$$

Parmi ces $N_{se}^{(k)}$ enfants, $N_s^{(k)}$ sont sélectionnés pour former la nouvelle population principale. La relation (8.1) permet d'écrire la relation réciproque :

$$N_{se,i}^{(k)} = \chi_i^{(k)} N_{se}^{(k)} \tag{8.2}$$

Avec les relations (8.1) et (8.2), nous obtenons la contrainte suivante sur les coefficients :

$$\sum_i \chi_i^{(k)} = 1 \tag{8.3}$$

La répartition entre les différentes branches, au travers du choix des coefficients χ_i, peut soit être fixée par l'utilisateur, ou soit être modifiée au cours du processus d'optimisation.

8.1.2.1 Population à taille constante pour chaque branche

Cette approche triviale permet d'imposer la taille $N_{se,i}^{(k)}$ pour chaque branche et de fixer par la même occasion la taille totale de la population des enfants $N_{se}^{(k)}$. Dans certaines situations, il est nécessaire de fixer la taille d'une branche de reproduction. Par exemple, après avoir développé un mutateur particulier analysant la population des parents pour générer un seul individu, il est intéressant de faire savoir à l'outil que cette branche ne produira qu'un seul individu par génération.

8.1.2.2 Population à taille variable par branche

Pour simplifier le raisonnement de cette approche, nous faisons l'hypothèse de travailler avec une taille $N_{se}^{(k)}$ constante tout au long du processus d'optimisation, l'objectif étant ici de répartir les ressources de calcul en fonction des performances de chacune des branches de reproduction. Deux approches sont possibles :

- évaluation des paramètres χ_i sans anticipation ;
- évaluation des paramètres χ_i avec anticipation.

Évaluation sans anticipation des paramètres $\chi_i^{(k)}$

A chaque itération, les coefficients $\chi_i^{(k)}$ sont évalués en fonction des résultats obtenus à l'itération précédente. Pour chaque nouvelle itération, nous avons :

$$\chi_i^{(k)} = \frac{N_{b,i}^{(k-1)}}{\sum_j N_{b,j}^{(k-1)}} \tag{8.4}$$

avec $N_{b,i}^{(k-1)}$, le nombre d'individus sélectionnés parmi les $N_{se,i}^{(k)}$ enfants provenant de la i^{eme} branche dans la nouvelle population $N_s^{(k)}$.

Cette méthode a pour défaut d'être sensible aux variations des résultats. Une approche pour lisser les variations des résultats permet d'éviter des saturations trop rapide des paramètres $\chi_i^{(k)}$. Effectivement, imaginons la situation où, à l'itération k, une branche fournit 0 bons individus, sachant que juste à l'itération précédente elle avait fournie 90% de la population des descendants. La probabilité de cette situation est normalement faible mais ne vaut pas 0%. Si cela se produit, le paramètre $\chi_i^{(k)}$ tombe à zéro et la branche ne produira plus d'enfants.

Pour éviter ce comportement de saturation, nous proposons la relation suivante :

$$\chi_i^{(k)} = \underbrace{\frac{N_{se}^{(k)} - \sum_j N_{b,j}^{(k-1)}}{N_{br} N_{se}^{(k)}}}_{Offset} + \underbrace{\frac{N_{b,i}^{(k-1)}}{N_{se}^{(k)}}}_{Variation} \tag{8.5}$$

où N_{br} correspond au nombre de branches de reproduction. Le premier terme de cette relation correspond au nombre de parents repris dans la population suivante et le second terme reprend la proportionnalité des performances pour faire varier le coefficient $\chi_i^{(k)}$. Cette nouvelle relation respecte toujours la relation (8.3) :

$$\sum \chi_i^{(k)} = \underbrace{\left(1 - \frac{N_b^{(k)}}{N_{se}^{(k)}}\right)}_{Offset} + \underbrace{\frac{N_b^{(k-1)}}{N_{se}^{(k)}}}_{Variation} = 1 \tag{8.6}$$

Le premier terme de la relation (8.5), l'offset, permet d'éviter une saturation inutile des coefficients.

Évaluation avec anticipation des paramètres $\chi_i^{(k)}$

A chaque itération, les coefficients $\chi_i^{(k)}$ sont calculés en fonction de l'évolution des performances de chaque branche. L'idée est d'accentuer l'anticipation des besoins de l'outil en favorisant les meilleures branches.

La loi proposée ici est une relation linéaire. La variation des deux derniers résultats $N_{se,i}^{(k)}$ est reportée à la génération suivante afin d'es-

timer la valeur optimale de la population de la branche. On calcule la variation :

$$\Delta N_{b,i}^{(k)} = N_{b,i}^{(k-2)} - N_{b,i}^{(k-1)} \tag{8.7}$$

Cette variation est utilisée pour calculer le coefficient $\chi_i^{(k)}$ selon la relation suivante :

$$\chi_i^{(k)} = K^{(k)} \Delta N_{b,i}^{(k)} \tag{8.8}$$

où le coefficient d'ajustement K est utilisé pour respecter la contrainte (8.3). Pour respecter celle-ci, il faut que la relation suivante soit respectée :

$$\sum_i K^{(k)} \Delta N_{b,i}^{(k)} = 1, \tag{8.9}$$

A partir de cette relation nous pouvons calculer $K^{(k)}$. Effectivement, nous avons :

$$\sum_i K^{(k)} \left(N_{b,i}^{(k-2)} - N_{b,i}^{(k-1)} \right) = 1 \tag{8.10}$$

donc

$$K^{(k)} \left(N_b^{(k-2)} - N_b^{(k-1)} \right) = 1 \tag{8.11}$$

ce qui nous donne

$$K^{(k)} = \frac{1}{\left(N_b^{(k-1)} - N_b^{(k-2)} \right)} \tag{8.12}$$

Le principal défaut de cette méthode est d'être sensible au bruit. L'évolution des performances de chaque branche peut être légèrement chaotique. D'autres lois d'évolution de $\chi_i^{(k)}$ peuvent être implémentées pour éviter d'être trop sensible aux variations des performances.

8.2 Croisement

L'aspect graphique de l'optimisation topologique permet d'introduire de nouvelles méthodes de croisement afin d'exploiter au mieux les informations de voisinage ou de distribution de matière. Dans cette section,

deux approches sont proposées. La première méthode concerne une mise en commun de ce qui a été étudié en section 6.1. La seconde méthode apporte une approche différente basée sur une étude de la structure permettant de combiner les zones sensibles de la topologie.

8.2.1 Croisement graphique non équilibré

Lors de l'étude de quatre opérateurs de croisement à la Section 6.1, il s'est avéré que la méthode graphique à croisement linéaire était plus performante que les autres méthodes, et les méthodes élitistes étaient plus performantes que les méthodes non élitistes. Il est possible de combiner l'ensemble de ces caractéristiques pour mettre en place un opérateur de croisement graphique élitiste exploitant la notion de voisinage entre les cellules.

Le croisement graphique non équilibré est une évolution graphique du croisement non équilibré. Deux versions sont proposées ici :

- La première version utilise les liens directs de voisinage entre les cellules pour conserver une partie de la structure d'un individu, qui est ensuite insérée dans un autre individu ;
- La seconde version utilise la notion de proportion de surfaces pour rendre élitiste la sélection graphique.

8.2.1.1 Liens de voisinage

Le choix des centres de Voronoï provenant des deux parents afin de composer un nouvel individu suit une règle de voisinage. Un centre de Voronoï est choisi aléatoirement comme référence dans l'individu le plus performant. On trie ensuite les centres de Voronoï par ordre croissant de distance par rapport à la référence. Les centres de Voronoï les plus proches appartenant au meilleur individu vont contribuer à la génération du nouvel individu. A l'inverse, pour l'individu le moins performant du couple, ce sont les cellules les plus éloignées des coordonnées de la référence du premier parent qui participent à la génération du nouvel individu.

Le nombre de cellules provenant de l'un ou l'autre parent dépend d'une relation à établir entre les performances des deux parents et de leur nombre de cellules :

$$(N_1, N_2) = h_1 \left(f_{(parent\ 1)}, f_{(parent\ 2)}, N_{c,(parent\ 1)}, N_{c,(parent\ 2)} \right) \quad (8.13)$$

où N_1 et N_2 correspondent respectivement au nombre de cellules à sélectionner dans le parent le plus performant et dans le parent le moins performant. La Figure 8.2 présente un exemple de croisement avec $N_1 = 8$ et $N_2 = 7$.

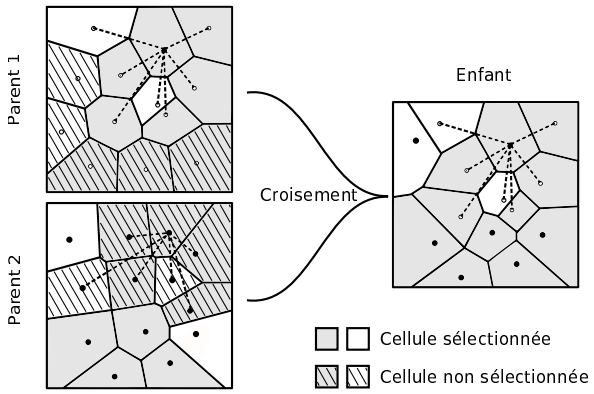

Figure 8.2 – Exemple de croisement graphique non équilibré en tenant compte des liens de voisinage avec $N_1 = 8$ et $N_2 = 7$

8.2.1.2 Proportion de surfaces

De la même manière que le croisement graphique circulaire, le croisement graphique non équilibré à proportion de surfaces utilise un cercle comme pivot. La différence réside dans le calcul du rayon du cercle. Ce dernier n'est plus un paramètre aléatoire mais devient fonction des évaluations des parents. Avec r le rayon du cercle, nous avons :

$$r = h_2 \left(f_{(parent\ 1)}, f_{(parent\ 2)} \right). \quad (8.14)$$

La fonction h_2 calcule le rapport de surface sachant que la surface du parent 2, pour $f_{(parent\ 2)}$ meilleur que $f_{(parent\ 1)}$, équivaut à $1 - \Omega_D$, où Ω_D est la surface du disque délimité par le cercle de sélection et 1 la surface de l'espace de design.

Le défaut de cette méthode est de ne pas prendre en compte la partie externe potentielle du disque. Effectivement, avec un mauvais choix de centre du cercle ou une mauvaise valeur du rayon, une partie du disque peut se retrouver en dehors de l'espace de design (Figure 8.3). Pour éviter cela, il peut être nécessaire d'effectuer des découpages géométriques de l'espace de design au moyen de segments de droites horizontales et verticales comme on peut le voir dans [51] tout en imposant la taille des surfaces pour prendre en compte la supériorité d'une solution sur une autre en terme d'évaluation.

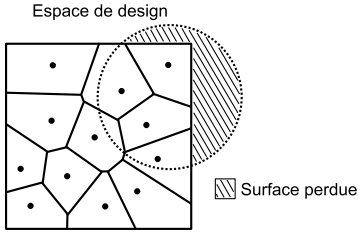

Figure 8.3 – Exemple de pertes de surfaces lors d'un croisement circulaire avec respect des surfaces de croisement

8.2.2 Croisement par superposition

En fin de convergence, les formes et topologies des solutions se ressemblent mais diffèrent par quelques variantes de position des centres de Voronoï. Par croisement, il est possible de combiner les cellules frontalières pour augmenter localement le nombre de degrés de liberté décrivant la forme de la solution (Figure 8.4).

Si l'on ne considère que les cellules frontalières lors du croisement, on peut récupérer les centres de Voronoï frontaliers du moins performant des parents pour les intégrer dans la structure du second parent. Un traitement supplémentaire est nécessaire pour éviter les doublons lors de la copie des centres.

8.3 Mutations des matériaux

L'optimisation de la manière dont il convient d'effectuer la mutation des matériaux n'a pas été abordée au cours de cette thèse. Or, cette mu-

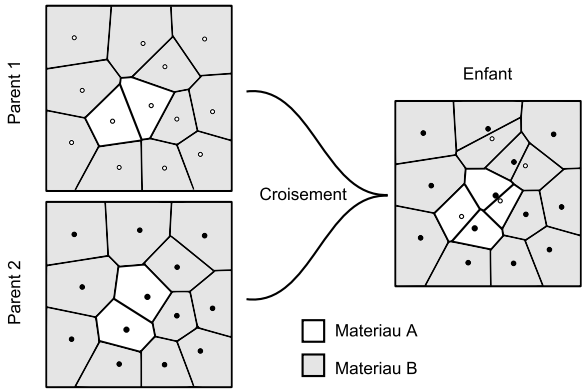

Figure 8.4 – Exemple de croisement graphique par superposition des centres de Voronoï

tation peut permettre d'accélérer la recherche de la meilleure topologie. On distingue différentes méthodes pour y arriver, dont la forme généralisée consiste à donner un taux de mutation, $p_{m,i}$, propre à chacune des cellules du diagramme de Voronoï. Dans cette section, différentes approches du choix des taux de mutation sont explorées.

8.3.1 Choix du taux de mutation en fonction du voisinage

En travaillant avec l'aspect graphique du problème, les cellules voisines d'une cellule peuvent influencer le matériau de cette dernière. On peut modifier le taux de mutation des cellules non-connectées (Figure 8.5). Cette approche a déjà été exploitée avec succès pour le formalisme Bit-Array [67].

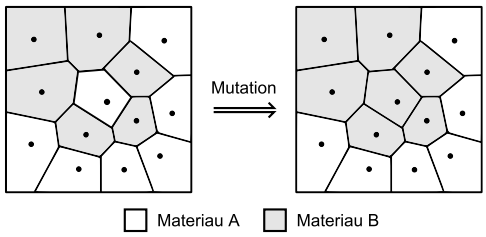

Figure 8.5 – Exemple de mutation de matériau avec un taux de 100%

La valeur du taux de mutation peut soit être radicale, c'est-à-dire

avec une valeur de 100% pour supprimer les cellules flottantes, soit une valeur intermédiaire fixe, soit une valeur variable.

8.3.2 Choix du taux de mutation en fonction de l'historique de la population

La population est une base de données. Si l'on sauvegarde dans une population secondaire tous les individus créés tout au long du processus d'optimisation, on peut établir des statistiques sur la distribution de matière au sein de l'espace de design. La Figure 8.6 illustre ce cas après 1 et après 20 générations pour une population de 100 individus dans le cas du problème simple physique. Bien que l'exemple soit donné avec 2 matériaux, la méthode est facilement adaptable pour plus de 2 matériaux.

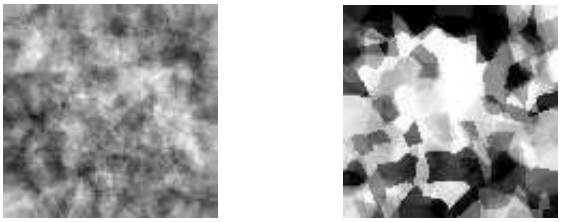

Figure 8.6 – Exemple de statistiques sur la distribution de matières après 1 et après 20 générations pour une population de 100 individus

Sur l'exemple donné, on peut apercevoir en blanc la naissance du trou de la topologie cible. La carcasse en fer, en nuances de gris, de la structure est aussi dessinée. Il ne reste que la fissure à découvrir.

Cette distribution statistique peut influencer la mutation d'une cellule de plusieurs manières, dont voici une liste non exhaustive :

- Le taux de mutation de chaque cellule peut être calculé en fonction de sa différence avec la référence statistique de distribution de matière ;
- Lorsqu'une cellule subit une mutation, la probabilité de choisir un matériau peut être définie en fonction de la référence statistique de distribution de matière ;
- On peut forcer la mutation d'une cellule, comme pour les cellules non connectées, en choisissant une valeur aléatoire influencée par la référence statistique de distribution de matière.

– ...

8.3.3 Choix du taux de mutation en fonction de l'évolution de la surface de la cellule

La suppression ou le changement de matériau d'une cellule peuvent être effectués suite à la pression interne au sein de la cellule au cours de l'optimisation. On entend par pression un indicateur de l'évolution de la surface de la cellule. Si une cellule voit sa surface réduite avec succès au cours du processus d'optimisation, il est possible que celle-ci doive être supprimée ou bien associée à un autre matériau déterminé en fonction du voisinage de la cellule.

Une variable, γ_i, est associée à chacune des cellules pour estimer le niveau de pression que chacune subit au cours du processus d'optimisation. Avant la reproduction, la pression de la cellule est mise à jour à chaque changement de surface suivant la règle suivante :

$$\gamma_i^{(k)} = \frac{S_i^{(k-1)}}{S_i^{(k)}} \gamma_i^{(k-1)} \tag{8.15}$$

avec $S_i^{(k)}$ la surface de la i^{eme} cellule à l'itération k, et $\gamma_i^{(0)} = 1$. Si la mutation de suppression ou de modification de matériau échoue, la pression de la cellule diminue légèrement. Si l'opération réussit, alors la pression est réinitialisée à 1.

Le calcul de la probabilité de suppression d'une cellule ou de son changement de matériau se base sur la valeur de la pression de toutes les cellules, afin de contrôler le nombre de mutations à chaque itération. Une version possible est de diviser la pression de chaque cellule par la somme des pressions de chaque cellule :

$$p_{m,i} = \frac{\gamma_i}{\sum_{i=1}^{N_c} \gamma_i} \tag{8.16}$$

8.4 Hybridation

De nombreuses modifications ont pour objectif d'apporter une aide à l'algorithme génétique dans sa phase d'exploitation. Or, il est tout à fait concevable d'utiliser un autre algorithme plus efficace dans cette tâche. Des essais déterminants ont été effectués durant cette thèse [68] en

combinant un algorithme génétique avec un algorithme de Nelder-Mead [69]. Le passage d'un algorithme à l'autre a été programmé de manière à effectuer un transit par étapes.

La combinaison des deux algorithmes demande des modifications :

- L'algorithme de Nelder-Mead travaille avec la meilleure solution de la population. A chaque génération de l'algorithme génétique, un nombre $N_{g,NE}$ de cycles de l'algorithme de Nelder-Mead sont exécutés pour améliorer cet individu ;
- Pour éviter la redondance dans les outils, les mutateurs de l'algorithme génétique ne s'occupent que des matériaux et l'algorithme de Nelder-Mead ne s'occupe que de la position des centres de Voronoï ;
- La contribution des deux algorithmes est changée au cours du processus d'optimisation via une évolution croissante du nombre $N_{g,NE}$. Au début, l'algorithme génétique fournit la contribution totale, alors qu'à la fin du processus d'optimisation, c'est l'algorithme de Nelder-Mead qui fournit la contribution totale ;
- L'amplitude de déplacement des centres de Voronoï évolue au cours des générations, la relation (8.19) donne un exemple de loi d'évolution de α :

$$\alpha = 0.2\left(1 - \frac{i-1}{k_{max}}\right) \text{ avec } i \in \{1, 2, \ldots, k_{max}\} \tag{8.17}$$

Des essais ont été effectués sur un problème d'électro-aimant cylindrique où l'objectif est de maximiser la force de maintien appliquée au plongeur ferromagnétique. Les courbes de convergences obtenues pour ce problème sont illustrées par la Figure 8.7.

On observe sur le graphique une nette dominance de l'algorithme hybride par rapport aux deux autres algorithmes d'optimisation. L'algorithme génétique met plus de 200% du temps pour approcher le résultat obtenu par l'algorithme hybride, alors que l'algorithme de Nelder-Mead n'arrive qu'85% de la fonction objectif de la meilleure solution obtenue par l'algorithme hybride au bout des 100% du processus d'optimisation.

Ces résultats sont confirmés par les représentations de distribution de matière illustrées à la Figure 8.8. L'algorithme génétique et l'algorithme hybride obtiennent la même topologie mais la forme est mieux définie pour la version hybride.

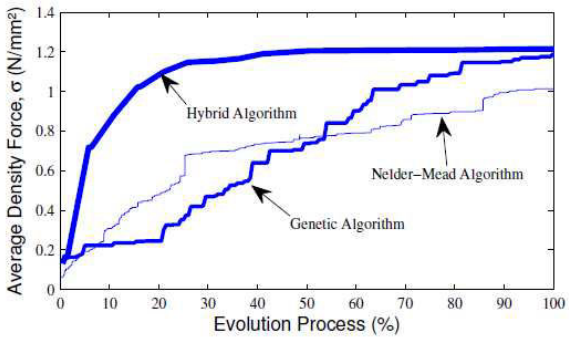

Figure 8.7 – Comparaison de convergence entre l'algorithme hybride, de Nelder-Mead (sur les positions) et de l'algorithme génétique (sur les positions et les matériaux) [68]

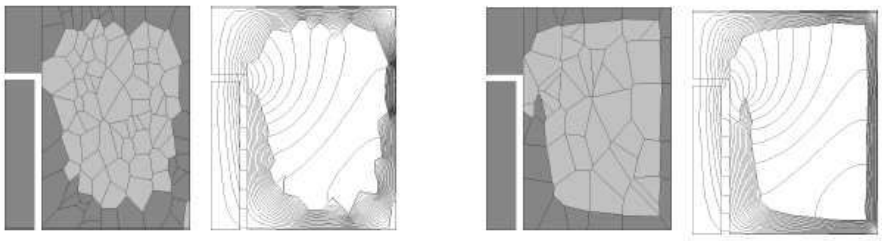

Figure 8.8 – Comparaison des distributions de matières avec en gris foncé le fer, en gris clair le cuivre et en blanc l'air, et des cartes de champs électromagnétiques obtenues entre l'algorithme génétique (gauche) et l'algorithme de de Nelder-Mead (droite) [68]

Les résultats montrent bien que l'hybridation pourrait être une solution intéressante pour l'optimisation topologique.

8.5 Synthèse

Au travers de ce chapitre, plusieurs idées issues de mes recherches au cours de ma thèse sont décrites. Certaines de ces idées sont le fruit de combinaisons entre plusieurs idées existantes, d'autres apportent réellement une nouveauté.

Sur l'ensemble des propositions, il semble que les méthodes d'hybri-

dation et de croisement soient les plus encourageantes. Une approche statistique pour attribuer un taux de mutation de matériaux est également prometteuse. Celle concernant la reproduction parallèle doit être soigneusement réglée pour éviter toute pertes d'évaluations si précieuses au processus d'optimisation.

Chapitre 9

Conclusions

Cette thèse apporte une contribution à l'étude des outils d'optimisation topologique basés sur des algorithmes métaheuristiques. Dans le cadre des travaux menés dans le domaine au sein du centre de recherche en mécatronique de l'École Polytechnique de Louvain en collaboration avec le laboratoire SATIE de l'ENS Cachan – antenne de Bretagne , elle a pour objectif de proposer une étude systématique afin de développer un outil d'optimisation topologique pour la conception de dispositifs électromagnétiques.

L'approche de la thèse s'est orientée vers une subdivision du travail en trois étapes, décrites à la Section 3.1 du Chapitre 3. Ces trois étapes sont : la sélection de l'outil candidat, l'étude d'adaptations de ce candidat à l'aspect graphique du problème d'optimisation topologique, et la composition de l'outil final par combinaison de différentes adaptations et son test sur deux applications liées à la conception de dispositifs industriels.

Afin de réduire les temps de calcul, des cas d'étude appropriés de complexité croissante, décrits en Section 3.2, ont été sélectionnés pour chacune de ces étapes. Pour la première étape de la thèse relative à la sélection de l'outil candidat, nous avons choisi un cas d'étude graphique simple consistant à retrouver une topologie déterminée sur base d'une fonction objectif à minimiser représentant directement la différence graphique existant entre la distribution de matière cible et les distributions de matière produites par l'outil d'optimisation. Pour la seconde étape de la thèse, qui étudie les adaptations permettant d'améliorer les performances de l'outil retenu, le cas d'étude sélectionné reprend la distribu-

tion de matière définie pour le cas d'étude graphique simple mais utilisé dans un contexte de mesure non invasive des défauts au sein d'un matériau ferromagnétique par une méthode basée sur le champ induit dans le matériau par un inducteur [vii]. L'évaluation de la fonction objectif nécessite donc le calcul de la répartition du champ magnétique au sein des distributions de matière produites par l'outil et est le résultat de leur comparaison avec la répartition de champ magnétique associé à la solution cible. Dans la troisième étape de la thèse (Section 3.2.3), l'outil final est composé et ses performances sont validées sur base de deux cas d'études représentatif d'un problème industriel. Ces deux cas d'étude concernent l'optimisation de la force moyenne d'un actionneur linéaire à réluctance variable et l'optimisation du couple moyen d'un moteur synchrone à aimant permanent. La fonction objectif revient à maximiser la variation de la co-énergie magnétique pour un déplacement donné du rotor.

Tout au long des étapes de sélection et d'adaptation, il a été nécessaire de définir des critères de comparaison des outils. Ces critères, décrits à la Section 3.3 du Chapitre 3, visent à mettre en évidence :

- La robustesse des outils vis-à-vis de leurs paramètres de contrôle pour limiter l'influence d'un mauvais choix de valeur par l'utilisateur ;
- Le taux de convergence du processus d'optimisation ;
- La valeur finale obtenue pour la solution convergée ;

9.1 Etape I : Sélection d'un outil d'optimisation topologique

Afin de choisir l'outil utilisé dans la deuxième étape de la thèse, il a été nécessaire d'effectuer une présélection d'algorithmes d'optimisation et de formalismes de distribution de matière afin de limiter le nombre d'outils à étudier. Quatre algorithmes d'optimisation topologique et deux formalismes de distribution de matière ont été retenus, formant ainsi 8 combinaisons différentes d'outils. Le Chapitre 4 décrit ces huit outils et compare ceux-ci entre eux afin de conserver le candidat le plus prometteur aux adaptations à l'optimisation topologique.

Les quatre algorithmes choisis sont l'algorithme génétique, l'algorithme du recuit simulé, l'algorithme immunitaire et l'algorithme à es-

saim particulaire. Ces quatre algorithmes représentent différentes approches de l'optimisation métaheuristique. Les deux formalismes de distribution de matière choisis sont le formalisme Bit-Array et le formalisme Voronoï. Chacun de ces deux formalismes représente une catégorie particulière, respectivement un formalisme statique et un formalisme dynamique.

Des résultats obtenus, trois conclusions importantes émergent concernant les algorithmes d'optimisation, ainsi qu'un enseignement concernant les formalismes de distribution de matière.

Au niveau des algorithmes d'optimisation, la première conclusion concerne la difficulté de l'algorithme à essaim particulaire de travailler dans un contexte d'optimisation topologique. Cet algorithme peut orienter l'évolution de ses individus vers un jeu de variables qui n'est pas optimal. Plusieurs essais ont été effectués en augmentant la taille de la population et le nombre d'évaluations pour limiter cet effet, mais sans obtenir de véritables bons résultats. La deuxième conclusion est que l'algorithme génétique est le seul algorithme à ne jamais être le dernier dans les différents classements obtenus, ce qui n'est pas le cas des trois autres algorithmes d'optimisation.

Au niveau des formalismes de distribution de matière, l'utilisation d'un formalisme dynamique permet à la discrétisation de l'espace de design de s'adapter à la solution optimale, diminuant ainsi la sensibilité aux paramètres de contrôle liés à la discrétisation de l'espace de design [iv]. Cette adaptation s'exprime de deux manières différentes : en modifiant les frontières des cellules et en modifiant le nombre de cellules. Ceci permet de discrétiser l'espace de manière à s'approcher au plus près de la topologie et de la forme de la solution optimale, tout en diminuant le nombre de cellules pour approcher le nombre idéal nécessaire, évidemment non connu à l'avance.

Les considérations qui précèdent montrent que le candidat le plus prometteur est le couplage de l'algorithme génétique avec le formalisme Voronoï. L'utilisation de cet outil permet un potentiel d'adaptation important pour y intégrer de nouveaux générateurs ou mutateurs afin d'accroitre son adaptation à l'optimisation topologique.

9.2 Etape II : Adaptation de l'algorithme génétique à l'aspect graphique de l'optimisation topologique

La seconde phase de la thèse a porté sur l'adaptation des opérateurs de l'algorithme génétique au problème de l'optimisation topologique. Les adaptations étudiées concernent deux domaines : la gestion de la population et la gestion de la reproduction.

9.2.1 Gestion de la population

Pour la gestion de la population, nous avons étudié au Chapitre 5 différentes méthodes de régulation de la taille de la population (Section 5.1) ainsi que différentes méthodes d'initialisation de la population (Section 5.2).

En premier lieu, nous avons montré que faire varier la taille de la population tout au long du processus d'optimisation joue un rôle important dans les performances de l'outil d'optimisation topologique [ix]. En effet, à nombre d'évaluations total constant, cette approche agit sur la taille de la population de génération en génération et permet de modifier le nombre d'évaluations nécessaires pour le calcul d'une génération donnée, et donc indirectement modifie le nombre d'itérations. Une bonne exploitation de ce paramètre permet d'augmenter l'efficacité de la phase d'exploration ainsi que celle de la phase d'exploitation. Les différentes méthodes de régulation de la taille de la population que nous avons étudiées sont la décroissance préprogrammée, la régulation par le rendement de chaque génération et la régulation par attribution d'un coefficient de reproduction. Les deux premières méthodes proposées donnent de bons résultats. La décroissance progressive de la population en fonction du nombre d'itérations réalisées permet de forcer l'algorithme à passer d'une phase d'exploration avec une grande population à une phase d'exploitation avec une petite population. Cependant, son efficacité dépend de la manière dont on programme la décroissance. La seconde méthode s'adapte à l'évolution de la performance de la population pour déterminer sa taille à chaque itération. Elle libère l'utilisateur du choix de la taille de la population.

En second lieu, nous avons montré que l'initialisation de la population influence légèrement les performances de l'outil d'optimisation topo-

logique. Le choix des premières solutions, donc des briques élémentaires composant chaque individu, va conditionner l'évolution de la population tout au long du processus d'optimisation. Des études ont été menées sur trois aspects de l'initialisation de la population afin d'identifier leur influence sur les performances de l'outil. Ces trois aspects sont la prise en compte des dimensions réelles de l'espace de design, la stratégie de distribution de la matière et l'anticipation de solution par une distribution de matière proche de la cible. En premier lieu, la méthode proposée afin de prendre en compte les dimensions de l'espace réel de design par une distribution aléatoire non-isotrope des centres de Voronoï au sein de l'espace de design normalisé permet une amélioration sensible du meilleur résultat obtenu mais a aussi pour conséquence d'augmenter la valeur médiane de la fonction objectif et la valeur de la fonction objectif du résultat le moins performant. Ensuite, lors de la génération aléatoire de la distribution de matière, la méthode proposée pour équilibrer la matière au sein de la population pour chaque point matériel de la solution permet à nouveau d'améliorer la meilleure solution mais au détriment de la valeur médiane de la fonction objectif et de la valeur de la fonction objectif de la moins bonne solution. Finalement, nous montrons qu'anticiper la solution permet seulement de conditionner l'évolution de la population, la conséquence étant une accélération de la convergence de la population vers une solution finale mais aussi une limitation de l'exploration de l'espace des solutions. Cette deuxième conséquence n'est évidemment pas souhaitable si la distribution de matière initiale n'est pas suffisamment bien choisie.

9.2.2 Gestion de la reproduction

Pour la gestion de la reproduction, nous avons étudié différentes approches de croisement ainsi que différentes approches de mutation de la position des centres de Voronoï et du nombre de cellules.

Pour le croisement, quatre méthodes ont été étudiées et comparées à la Section 6.1 du Chapitre 6. Deux d'entre elles travaillent dans le domaine graphique de la solution, alors que les deux autres travaillent dans le domaine des variables d'optimisation de la solution. Nous montrons que travailler directement dans le domaine graphique permet d'améliorer les résultats obtenus par l'outil d'optimisation topologique.

Pour les méthodes de mutation de position des centres de Voronoï,

quatre aspects ont été étudiés [x] : le type de cellule ciblée, le type de déplacement, l'amplitude du déplacement et l'orientation du déplacement. Tout d'abord, nous avons montré en Section 6.2.1.1 qu'il est intéressant de distinguer, au niveau des sous-domaines de matière qui définissent la topologie, les cellules internes des cellules frontalières, afin d'anticiper sur leur rôle futur pour la description de la distribution de matière. Une cellule interne a effectivement peu de chance de participer à la description des frontières des sous-domaines de la topologie, sauf en créant elle-même un nouveau sous-domaine de matière en changeant de matériau, devenant ainsi une cellule frontalière, alors que les variables de position des cellules frontalières sont les degrés de liberté pour décrire les frontières de la topologie et donc la forme de celle-ci. Il convient par conséquent de limiter préférentiellement le déplacement des centres de Voronoï aux cellules frontalières. Ensuite, nous avons montré à la Section 6.2.1.2, que le type de déplacement des centres de Voronoï influence la performance de l'outil d'optimisation topologique. Un déplacement classique, variable par variable, n'étant pas adapté à la situation graphique, nous avons proposé pour améliorer les performances de l'outil d'optimisation topologique une approche modifiant globalement la position (x, y) des centres de Voronoï lors de la mutation. Ensuite, nous avons montré en Section 6.2.1.3 que l'amplitude de déplacement des centres de Voronoï joue un rôle très important quant à l'évolution de la population. Effectivement, de grandes amplitudes permettent une bonne exploration de l'espace des solutions alors qu'une faible amplitude permet une bonne exploitation de chaque individu. Ainsi donc, il est important de faire varier cette amplitude au cours du processus d'optimisation. Bien qu'il soit difficile de déterminer avec précision comment il faut faire varier cette amplitude, trois méthodes ont été proposées et comparées avec l'approche classique utilisant une amplitude constante [viii]. Ces trois méthodes sont la méthode cinématique, la méthode dynamique locale et la méthode dynamique globale. Finalement en Section 6.2.1.4, nous avons introduit une méthode calculant l'orientation ainsi que l'amplitude de déplacement d'un centre de Voronoï en fonction de l'évolution de sa position au cours des deux itérations précédentes. Les résultats obtenus permettent de dire qu'en début d'optimisation, il est intéressant de favoriser les grands déplacements pour améliorer l'exploration de l'espace de design, puis de réduire leur amplitude et en fin d'optimisation de faire dépendre l'amplitude et l'orientation du déplacement des

centres en fonction des résultats obtenus aux itérations précédentes.

Nous avons proposé deux méthodes permettant de réguler le nombre de cellules : le vieillissement cellulaire (Section 6.2.2.5) et la prolifération cellulaire (Section 6.2.2.6). La première permet de supprimer des cellules et la seconde permet d'en ajouter de nouvelles. Ces deux méthodes se basent sur le type de chaque cellule, c'est-à-dire frontalière ou interne, afin de déterminer l'action à effectuer sur celles-ci. Nous avons montré qu'il est conseillé de supprimer les cellules internes afin de diminuer le nombre de variables d'optimisation et de laisser l'algorithme se focaliser sur les cellules frontalières décrivant la forme globale de la solution. Et finalement, bien que la stratégie de prolifération proposée ne soit pas entièrement efficace, nous avons constaté qu'il est intéressant d'augmenter le nombre de cellules frontalières pour améliorer la description de la forme de la solution.

9.3 Etape III : Composition et test de l'outil final

Finalement, après l'étude des différentes adaptations, la troisième phase a d'abord consisté à sélectionner les adaptations les plus intéressantes afin de composer l'outil d'optimisation topologique final (Section 7.1), et ensuite à comparer cet outil à des outils de référence sur les cas d'étude plus complexes décrits en Section 3.2.3.

Pour le premier cas d'étude, l'outil développé au sein de cette thèse dépasse largement les performances de l'outil de référence sur plusieurs points. En premier lieu, la vitesse de convergence est plus rapide. Au bout de 20% du processus avec l'outil final proposé, les résultats sont équivalents à ceux obtenus au bout de 100% du processus d'optimisation avec l'outil de référence. En second lieu, la précision des solutions produites au niveau de leur forme et de leurs dimensions est améliorée.

Pour le second cas d'étude, les résultats obtenus avec l'outil final restent mitigés. La topologie finale n'est pas la topologie optimale. Cependant, la qualité de la solution montre un résultat de meilleur qualité comme pour le premier cas d'étude au niveau de la forme et des dimensions. En utilisant par contre les mêmes paramètres que Ishikawa, c'est-à-dire avec 45 individus pour 120 cellules, nous obtenons aussi la même topologie. Les résultats montrent que le choix du nombre de cel-

lules et de la taille de la population joue encore un rôle quant à la qualité des résultats.

9.4 Perspectives et contributions supplémentaires

Pour finir, le Chapitre 8 propose des perspectives de poursuite du travail réalisé dans cette thèse. Ces perspectives touchent à la gestion de la reproduction ainsi qu'à la gestion de la population. Au-delà, ce chapitre propose d'étudier d'autres approches visant à modifier la composition de l'outil d'optimisation, comme par exemple l'hybridation de l'algorithme génétique avec l'algorithme de Nelder-Mead pour combiner les bénéfices de chacun, la capacité d'exploration du premier et la capacité d'exploitation du second, ce qui a conduit à des résultats positifs [vi].

Finalement, d'autres contributions ont été faites au cours de la thèse et publiées au travers d'articles de conférence [i][ii][iii][v] où nous appliquons des outils d'optimisation topologique, exploitant l'algorithme génétique ou l'algorithme du recuit simulé couplé au formalisme Voronoï ou au formalisme Bit-Array, à des cas d'application diverses tels qu'une micropompe à ferrofluide et un moteur à réluctance variable.

Publications associées à la thèse

[i] B. Dehez, **J. Denies** and H. Ben Ahmed, "Design of electromagnetic actuators using optimizing material distribution methods," *In : Proceedings of the 2008 International Conference on Electrical Machines (ICEM'2008)*, Vilamoura, Portugal, September 6-9, IEEE, 2008.

[ii] **J. Denies**, H. Ben Ahmed and B. Dehez, "Design of a Ferrofluid Micropump Using a Topology Optimization Method," *In : Proceedings of the ELECTROMOTION'2009 Conference*, Lille, France, July 1-3.

[iii] **J. Denies**, B. Dehez and H. Ben Ahmed, "Optimisation topologique par distribution de matière du design d'une micropompe à ferrofluide sans contrainte de structure fixe, *In : Proceedings de la conférence Electrontechnique du Futur 2009 (EF'2009)*," Compiègne, France, 24-25 septembre, 2009

[iv] **J. Denies**, B. Dehez, F. Glineur and H. Ben Ahmed, "Impact of the material distribution formalism on the efficiency of evolutionary methods for topology optimization," *Book chapter in : Recent Advances in Optimization and its Applications in Engineering*, Springer, 2010, pp.461-470, ISBN : 978-3-642-12597-3

[v] **J. Denies**, B. Dehez and H. Ben Ahmed, "Simulated annealing and genetic algorithms in topology optimization tools : a comparison through the design of a switched reluctance machine, *In : Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM'2010)*," Pise, Italie, June 14-16, IEEE Press, 2010, pp. 1247-1252.

[vi] **J. Denies**, H. Ben Ahmed, F. Glineur and B. Dehez, "Study of a topology optimization tool associating a hybrid algorithm with a Voronoi-based formalism for the design of electromagnetic devices," in *2010 XIX International Conference on Electrical Machines (ICEM'2010)*, pp. 1-6, Sept. 2010.

[vii] **J. Denies**, H. Ben Ahmed and B. Dehez, "Topology Optimization for Solving Electromagnetic Inverse Problem," *In : 1st International ICST Workshop on New Computational Methods for Inverse Problems (NCMIP'11)*, May 16, 2011, Cachan, France.

[viii] **J. Denies**, H. Ben Ahmed and B. Dehez, "Optimal Design of Electromagnetic Devices : Development of an Efficient Optimization Tool Based on Smart Mutation Operation Implemented in a Genetic Algorithm, " *ELECTRIMACS'11*, June 6-8, 2011, Cergy-Pontoise, France.

[ix] **J. Denies** and B. Dehez, "Genetic algorithm-based topology optimization : performance improvement through dynamic evolution of the population size," *In : Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM'2012)*, Sorrento, Italie, June 14-16, IEEE Press, 2012, pp. 1033-1038

[x] **J. Denies** and H. Ben Ahmed and B. Dehez, "Optimal design of electromagnetic devices : development of an efficient optimization tool based on smart mutation operations implemented in a genetic algorithm," *Mathematics and Computers in Simulation*, 2013, 10.1016/j.matcom.2013.01.002

Annexe A

Choix d'implémentation des algorithmes

Afin de comparer les différents algorithmes d'optimisation métaheuristiques entre-eux, il est nécessaire de déterminer les différents modules les composant, c'est-à-dire leurs générateurs et leurs mutateurs. Deux algorithmes sont ciblés : l'algorithme génétique et l'algorithme du recuit simulé. L'algorithme immunitaire n'est pas étudié ici puisqu'il partage les mêmes mutateurs que l'algorithme génétique. Quant à l'algorithme de l'essaim particulaire, il n'y a aucun mutateur ou générateur à étudier.

La section A.1 présente un nouveau cas d'étude rapide à évaluer pour permettre un grand nombre d'essais. Les sections A.2 et A.3 étudient les différentes configurations étudiées respectivement pour l'algorithme génétique et pour l'algorithme du recuit simulé.

A.1 Cas d'étude

Pour cette étude préliminaire, nous utilisons comme benchmark la fonction de Styblinski–Tang à minimiser :

$$f(\vec{x}) = \frac{\sum_{i=1}^{20} x_i^4 - 16x_i^2 + 5x_i}{2} \tag{A.1}$$

où les variables d'optimisation sont définies dans l'espace de travail suivant :

$$x_i \quad \in \quad [-5;5] \text{ avec } i \in 1, 2, \ldots, 20 \tag{A.2}$$

Afin d'étudier la variance des résultats, une cinquantaine d'essais sont ainsi effectués pour chaque configuration à étudier.

A.2 Algorithme génétique

Pour son fonctionnement, l'algorithme génétique utilise quatre modules :

- le module de sélection des parents ;
- le module de croisement ;
- le module de mutation ;
- le module de sélection des descendants.

Les deux modules de sélection sont étudiés simultanément en Section A.2.1 puisque, de par leur position dans la structure de l'algorithme génétique, leur interaction est forte. On peut toutefois rencontrer dans

la littérature une absence d'utilisation du module de sélection des descendants. Ceci a été le cas des premiers algorithmes génétiques qui se différenciaient ainsi des algorithme à stratégie d'évolution. Dans notre cas, nous étudierons un ensemble de combinaisons possibles de ces deux modules de sélection. Ensuite, la Section A.2.2 étudie différentes méthodes de croisement. Finalement, la Section A.2.3 étudie les méthodes de mutation.

A.2.1 Sélection des parents et des descendants

Le module de sélection des parents et le module de sélection des descendants peuvent utiliser des mécanismes identiques. Cependant, la seule différence entre les deux mécanismes est le choix de sélection aléatoire avec remise ou sans remise. La sélection des parents effectue une sélection avec remise pour donner plus de chances aux meilleurs individus de se reproduire par la suite puisqu'un individu sélectionné peut l'être à nouveau. La sélection des descendants a pour objectif de sélectionner un échantillon d'individus parmi les enfants pour composer une nouvelle population, chaque individu ne peut donc être sélectionné qu'une seule fois.

Quatre approches sont comparées :
- La sélection par copie de tous les individus (parents) ;
- La sélection par tournoi (parents et descendants) ;
- La sélection par roulette (parents et descendants) ;
- La sélection par élitisme (descendants).

Chacune de ces méthodes proposent des niveaux différents d'élitisme et de diversité dans la population. Les combinaisons de sélection parents/descendants sont : copie/élite, copie/roulette, copie/tournoi, tournoi/élite, tournoi/roulette, tournoi/tournoi, roulette/élite, roulette/roulette et roulette/tournoi.

A.2.1.1 Description

Sélection par copie :

La copie de tous les individus est la méthode la plus simple qui ne considère aucune intervention de la performance des individus lors de l'étape de reproduction. Tous les individus sont ainsi copiés dans la population de parents. Elle peut être facilement utilisée lors de problème

d'optimisation nécessitant un maximum d'exploration, c'est-à-dire évitant une convergence trop rapide par un élitisme abusif.

Sélection par tournoi :

La sélection par tournoi crée N_s couples de parents choisis aléatoirement, avec remise, dans les individus. Pour chaque couple, on ne sélectionne que le plus performant. Si il y a égalité, alors un individu du couple est tiré aléatoirement. Les N_s vainqueurs composent ensuite la population des parents.

Pour la sélection des descendants, le meilleur individu est directement copié dans la population des descendants afin de conserver la meilleure solution.

Sélection par roulette :

Cette troisième méthode attribue une probabilité de sélection de chaque individu proportionnellement à sa performance. Une méthode d'implémentation proposée par [70] consiste tout d'abord à trier les solutions par ordre décroissant de performances. Toutes solutions équivalentes ne seront pas mises à la même place. La première arrivée est considérée comme étant la première dans la liste.

$$f_i \overset{Tri}{\rightarrow} f'_i \tag{A.3}$$

L'ordre d'apparition dans la liste est ensuite utilisé pour déterminer la probabilité d'une solution d'être sélectionnée. Sachant que le $n^{ième}$ élément de la liste a n fois moins de chance d'être sélectionné que le premier, il faut diviser f'_i par la somme suivante :

$$S = \sum_{i=0}^{N_s} i \tag{A.4}$$

ce qui donne :

$$f''_i = \frac{f'_i}{S} \tag{A.5}$$

Ensuite, les performances de chaque solution sont empilées les unes sur les autres en calculant une borne minimale, $b_{min,i}$, et une borne maximale, $b_{max,i}$, pour chaque solution :

$$\begin{array}{lcll} b_{min,i} & = & 0 & \text{pour } i = 0 \\ b_{min,i} & = & b_{max,i-1} & \text{pour } i = 1, \ldots, N_s \\ b_{max,i} & = & b_{min,i} + f_i'' & \text{pour } i = 0, \ldots, N_s \end{array} \tag{A.6}$$

Finalement, l'algorithme choisit N_s fois une valeur aléatoire, r, entre 0 et 1. Un individu est sélectionné lorsqu'une des valeurs tombe entre sa borne minimale et sa borne maximale.

$$b_{min,i} \leq r < b_{max,i} \tag{A.7}$$

Pour la sélection des descendants, de même que pour la méthode de sélection par tournoi, le meilleur individu est directement copié dans la population des descendants afin de conserver la meilleure solution.

Sélection par élitisme :

L'élitisme est une méthode où le seul critère intervenant dans la sélection est la performance de la solution par rapport aux autres. Les N_s meilleures solutions sont sélectionnées pour composer la nouvelle population principale.

A.2.1.2 Résultats et discussion

Pour chaque combinaison de sélection étudiée, 50 essais ont été réalisés avec une population de 100 individus évoluant sur 250 générations pour un total de 25000 évaluations. Le taux de croisement est fixé à 90% et le taux de mutation à 5%. La méthode de croisement est une méthode à 1-pivot décrite en Section A.2.2 et la méthode de mutation est la méthode (A) décrite en Section A.2.3.

Les courbes médianes de convergence des résultats obtenus sont présentées sur les trois graphiques de la Figure A.1. On distingue deux groupes de courbes de convergence dépendant du choix de la sélection des descendants. Le groupe rassemblant les résultats les plus performants utilise une sélection des descendants par élitisme. Les combinaisons les moins performantes utilisent une sélection des descendants par roulette et par tournoi. Au sein du second groupe de courbes de convergence, bien que l'on distingue difficilement les courbes de convergence, les résultats obtenus avec la sélection des descendants par tournoi sont légèrement meilleurs que ceux obtenus avec la sélection des descendants par roulette.

On peut aussi observer qu'avant les 10% du processus d'optimisation, la population évolue à des vitesses différentes pour les trois cas de sélection de parents.

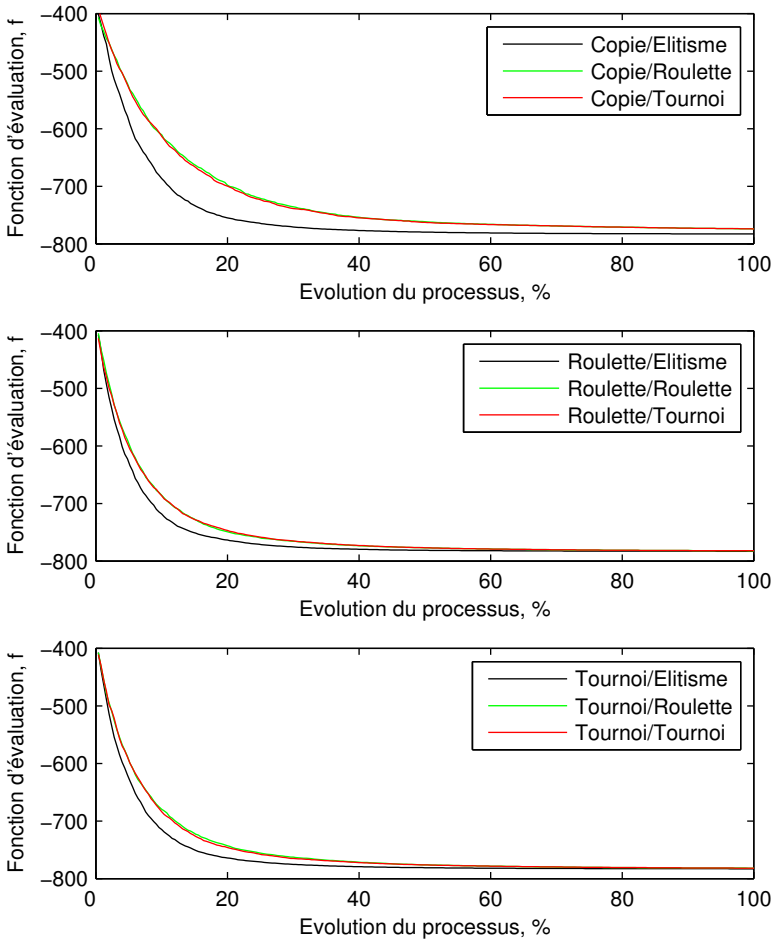

Figure A.1 – Comparaison des courbes médianes de convergence obtenues avec 50 essais pour chacune des 9 configurations de sélection parents/descendants

Le Tableau A.1 résume numériquement les résultats obtenus pour les 9 combinaisons de sélection de parents et de descendants. On observe à nouveau une différence significative entre les essais avec sélection des descendants par élitisme et les autres essais. Les meilleures résultats sont

obtenus avec la combinaison roulette/élitiste.

Combinaison	Minimum	Médiane	Maximum
Copie/Elitiste	−783.129	−782.870	−782.442
Copie/Roulette	−778.916	−774.105	−765.900
Copie/Tournoi	−778.514	−774.144	−765.342
Roulette/Elitiste	**-783.221**	**-783.050**	−782.72
Roulette/Roulette	−782.905	−782.278	−781.199
Roulette/Tournoi	−782.928	−782.207	−781.317
Tournoi/Elitiste	−783.217	−783.039	**-782.823**
Tournoi/Roulette	−782.907	−781.902	−780.432
Tournoi/Tournoi	−782.747	−781.952	−780.628

Table A.1 – Résumé des résultats obtenus avec 50 essais pour chaque combinaison de sélection de parents et de descendants

Les résultats obtenus nous montre qu'une approche élitiste pour la sélection des descendants permet d'accélérer la vitesse d'évolution de la population. Le choix entre la sélection des parents par tournoi ou par roulette est cependant difficile à faire. Les résultats des deux approches sont fort semblables.

A.2.2 Croisement

Quatre méthodes de croisement ont retenu notre attention dans les tests. Toutes se basent sur un croisement par transfert de variables d'optimisation et non pas sur un croisement par modification des valeurs des variables d'optimisation :

- Croisement à un pivot ;
- Croisement à deux pivots ;
- Croisement uniforme,
- Croisement non-uniforme.

Chacune de ces méthodes proposent des niveaux différents de mixages des variables d'optimisation des parents.

A.2.2.1 Descriptions

Croisement à 1 et à 2 pivots :

Le croisement à un pivot choisit aléatoirement un nombre r entre 1 et le nombre de variables. Les r premières variables du premier parent sont copiées dans l'enfant et les $N_p - r$ dernières variables du second parent sont aussi copiées pour compléter le chromosome.

Le croisement à deux pivots est identique au croisement à un pivot. La différence réside dans la sélection d'un second pivot. L'utilisation de deux pivots permet d'accentuer le brassage des variables d'optimisation. Nous avons r_1 et r_2, nos deux pivots avec $r_2 \geq r_1$. Les r_1 premières variables du premier parent sont copiées dans l'enfant, ensuite les $r_2 - r_1$ variables suivantes du parent 2 et finalement, les dernières variables, supérieures à r_2, sont copiées du parent 1.

Sélection uniforme et non uniforme :

Le croisement uniforme choisit aléatoirement pour chaque gène l'origine du donneur entre les deux parents avec une probabilité équivalente. Une variante est aussi proposée, le croisement non-uniforme. La probabilité de donner un gène est calculée en fonction des performances des parents. Cette probabilité est obtenue en calculant le rapport entre les évaluations des deux parents. En fixant les indices de manière à ce que l'évaluation f_1 est plus performante que l'évaluation f_2, le ratio devient :

$$a = \frac{f_2}{f_1} \tag{A.8}$$

La probabilité de choisir un gène provenant du meilleure des deux parents est de $a/(a+1)$ et est de $1/(a+1)$ pour l'autre parent.

A.2.2.2 Résultats et discussion

Pour chaque cas de croisement étudié, 50 essais ont été réalisés avec une population de 100 individus évoluant sur 250 générations, ce qui donne une maximum de 25000 évaluations. Le taux de croisement est fixé à 90% et le taux de mutation à 5%. La méthode de mutation est la méthode (A) décrite en Section A.2.3. La sélection des parents s'effectue par roulette et la sélection des descendants par élitisme.

Les courbes de convergence des résultats obtenus pour les quatre méthodes de croisement sont données par le graphique de la Figure A.2. On observe une dominance dans la méthode de croisement uniforme, suivit des méthodes de croisement à 2 pivots et non-uniforme.

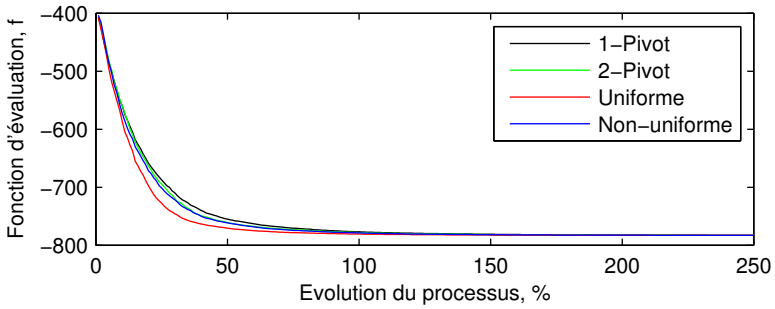

Figure A.2 – Comparaison des courbes médianes de convergence obtenues avec 50 essais pour les trois cas de croisement étudié

Le Tableau A.2 résume numériquement les résultats obtenus pour les 3 cas étudiés de croisement. On observe à nouveau une dominance de la méthode de croisement uniforme sur les valeurs minimales, médianes et maximales obtenues à partir de 50 essais pour chaque cas étudié.

Croisement	Minimum	Médiane	Maximum
1-Pivot	−783.091	−782.879	−782.567
2-Pivot	−783.173	−782.942	−782.582
Uniforme	**-783.262**	**-783.092**	**-782.945**
Non-uniforme	−783.156	−782.972	−782.687

Table A.2 – Résumé des résultats obtenus avec 50 essais pour les 4 cas de croisement étudiés

En observant les résultats obtenus, on peut avancer que la méthode de croisement doit pouvoir permettre un brassage le plus complet possible des variables d'optimisation. En effet, nous avons dans l'ordre croissant du niveau de brassage des variables d'optimisation : la méthode à 1 pivot, la méthode à 2 pivot, la méthode non-uniforme et la méthode uniforme. Ce classement est identique lorsqu'on observe les meilleures performances obtenues.

A.2.3 Mutation

Trois méthodes de mutation sont comparées dans cette section afin de déterminer laquelle sera implémentée pour l'algorithme génétique dans son étude tout au long de la recherche. Dans ces trois méthodes, nous faisons intervenir quelques petites variances.

A.2.3.1 Descriptions

Les trois méthodes sont pour la suite notées (A), (B) et (C) pour les distinguer. Leurs implémentations sont :

- Soit une probabilité de $\frac{1}{N_p}$ associé à chaque variable doit être vérifiée pour modifier sa valeur aléatoirement entre les bornes associées à la variable ;
- Soit N variables d'optimisation aléatoirement choisies dans le chromosome de chaque enfant pour être mutées. La valeur de N dépend du taux de mutation p et du nombre de paramètres N_p : $N = pN_p$;
- Soit une probabilité de $\frac{1}{N_p}$ associé à chaque variable mais dont la valeur, au lieu d'être changée pour une valeur aléatoire, change selon $x_i = x_i \pm \theta$ avec $\theta = \delta\,|\epsilon|$. La variable $\epsilon \sim \mathcal{N}\left(0, \frac{1}{3}\right)$ est une variable aléatoire et δ est le coefficient dépendant de la distance avec les bornes associées à la variable.

A.2.3.2 Résultats

Pour chaque cas de mutation, un ensemble de 50 essais a été effectué avec une population de 100 individus évoluant sur 250 générations, ce qui donne un maximum de 25000 évaluations. La probabilité de mutation, fixée à 5%, est identique aux trois méthodes étudiées. La probabilité de croisement est fixée à 90%. La méthode de croisement est une méthode à 1-pivot décrite en Section A.2.2. La sélection des parents est effectuée par roulette et la sélection des descendants est effectuée par élitisme.

Les courbes médianes de convergences obtenues pour chacun des cas sont donnés par le graphique de la Figure A.3. On observe peu de différence entre les trois courbes. La courbe médiane de convergence de la mutation (C) présente l'évolution de la population la moins efficace des trois mutations et la mutation (C) est légèrement plus .

Les résultats numériques résumés par la Table A.3 montrent que la mutation (C) obtient de meilleurs résultats pour la valeur minimale et la

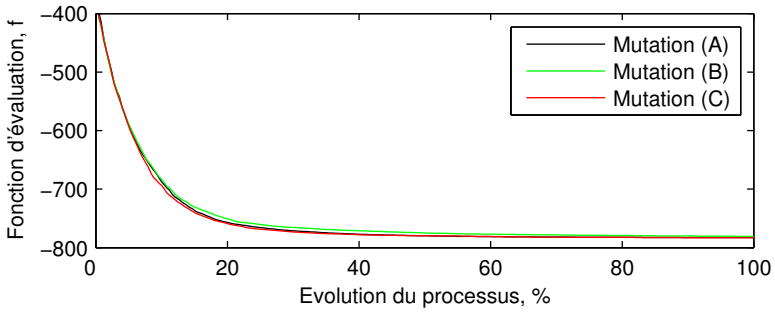

Figure A.3 – Graphiques des courbes médianes de convergence pour les trois cas de mutation étudiés.

valeur médiane. Mais la valeur maximale de cette mutation est la moins bonne pour l'ensemble des mutations. La mutation (A) présente donc des valeurs proches de la mutation (C) mais a aussi la meilleure valeur maximale.

Mutation	Minimum	Médiane	Maximum
Mutation (A)	−783.124	−782.873	**-782.538**
Mutation (B)	−781.793	−780.779	−779.606
Mutation (C)	**-783.296**	**-782.951**	−769.091

Table A.3 – Résumé des résultats obtenus avec 50 essais pour les 3 cas de mutation étudiés

A.3 Algorithme du recuit simulé

L'évolution d'une solution dans l'algorithme du recuit simulé est guidée par l'évolution de la température au cours du processus d'optimisation. Dans cette section, nous étudions l'influence de cette évolution de la température afin de déterminer une tendance du choix de la loi d'évolution.

A.3.1 Description

Trois méthodes d'évolution de la température sont comparées :
– Évolution par palier ;

– Évolution continue linéaire ;
– Évolution continue quadratique.

L'évolution par palier utilise 500 paliers de température par pas de 50 itérations. Cette règle est implémentée comme ceci :

$$T_1(k) = T_0 \left(\frac{20 - floor\left(\frac{k}{50}\right)}{20} \right) \tag{A.9}$$

où $floor$ est la commande matlab qui retourne l'entier le plus proche dans la direction de l'infini négatif.

Avec la méthode de l'évolution linéaire de la température, celle-ci évolue proportionnellement avec le nombre d'itérations effectués :

$$T_2(k) = T_0 \left(\frac{k_{max} - k}{k_{max}} \right) \tag{A.10}$$

Et l'évolution quadratique est semblable à la méthode linéaire à la seule différence que le coefficient multipliant T_0 est élevé au carré :

$$T_3(k) = T_0 \left(\frac{k_{max} - k}{k_{max}} \right)^2 \tag{A.11}$$

A.3.2 Résultats

Pour l'ensemble des essais, un individu évolue sur 25000 itérations avec un taux de mutation de 5%. La méthode de mutation permettant de modifier l'individu au cours du processus d'optimisation est la méthode (A) décrite en Section A.2.3.

Les courbes médianes de convergence sont données par le graphique de la Figure A.4. Les résultats montrent que la méthode quadratique est plus efficace que les deux autres méthodes qui ont des résultats semblables. La méthode quadratique permet de converger plus rapidement vers l'optimum en diminuant plus rapidement la température de contrôle du processus d'optimisation.

Les résultats numériques résumés par la Table A.4 montrent à nouveau que l'évolution quadratique de la température obtient de meilleurs résultats pour la valeur minimale, la valeur médiane et la valeur maximale obtenues. Les deux autres méthodes sont fortement en retrait et ont des valeurs proches.

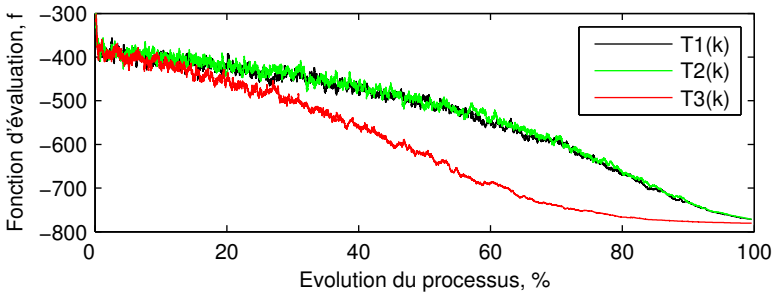

Figure A.4 – Comparaison des courbes médianes de convergence obtenues avec 50 essais pour les trois cas étudiés d'évolution de la température

Température	Minimum	Médiane	Maximum
$T_1\,(k)$	-777.804	-771.659	-760.878
$T_2\,(k)$	-778.199	-771.145	-754.484
$T_3\,(k)$	**-782.190**	**-780.578**	**-777.416**

Table A.4 – Résumé des résultats obtenus avec 50 essais pour les 3 cas étudiés d'évolution de température

Références

[1] J. S. Arora, *Introduction to optimum design*. Academic Press, May 2004.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, third edition ed., July 2009.

[3] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100 –107, July 1968.

[4] P. Hajela, E. Lee, and C. Lin, "Discrete optimal design using a genetic algorithm," *Topology Design of Structures*, pp. 117–133, 1993.

[5] D. Grierson and W. Pak, "Discrete optimal design using a genetic algorithm," *Topology Design of Structures*, pp. 89–102, 1993.

[6] Z. Wu, *Optimisations génétiques en Mécanique des Solides*. PhD thesis, Ecole Polytechnique de Paris, September 1995.

[7] G. Strang and R. V. Kohn, "Optimal design in elasticity and plasticity," *International Journal for Numerical Methods in Engineering*, vol. 22, pp. 183–188, Jan. 1986.

[8] M.P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 197–224, Nov. 1988.

[9] G. Allaire and R.V. Kohn, "Optimal design for minimum weight and compliance in plane stress using extremal microstructures," *European journal of mechanics. A. Solids*, vol. 12, no. 6, 1993.

[10] M. Bendsøe, A. Diaz, and N. Kikuchi, "Topology and generalized layout optimization of elastic structures," *Topology Design of Structures*, pp. 159–205, 1993.

[11] M. Bendsøe, "Optimal shape design as a material distribution problem," *Structural and Multidisciplinary Optimization*, vol. 1, pp. 193–202, Dec. 1989.

[12] G. I. N. Rozvany, M. Zhou, and T. Birker, "Generalized shape optimization without homogenization," *Structural Optimization*, vol. 4, pp. 250–252, Oct. 1992.

[13] O. Sigmund, "A 99 line topology optimization code written in matlab," *Structural and Multidisciplinary Optimization*, vol. 21, pp. 120–127, Apr. 2001.

[14] Y. Xie and G. Steven, "A simple evolutionary procedure for structural optimization," *Computers & Structures*, vol. 49, pp. 885–896, Dec. 1993.

[15] G. Anagnostou, E. M. Rønquist, and A. T. Patera, "A computational procedure for part design," *Computer Methods in Applied Mechanics and Engineering*, vol. 97, pp. 33–48, May 1992.

[16] E. Jensen, *Topological structural design using genetic algorithms*. PhD thesis, Purdue University, November 1992.

[17] M. Schoenauer, "Shape representations and evolution schemes," *Proceedings of the 5th annual conference on evolutionary programming*, pp. 121—129, 1996.

[18] W. Hansel, "A heuristic and a genetic topology optimization algorithm for weight-minimal laminate structures," *Composite Structures*, vol. 58, no. 2, pp. 287–294, 2002.

[19] F. Campelo, F. Guimaraes, H. Igarashi, K. Watanabe, and J. Ramirez, "An immune-based algorithm for topology optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 3204 –3211, 2006.

[20] D. Lowther, W. Mai, and D. Dyck, "A comparison of MRI magnet design using a hopfield network and the optimized material distribution method," *Magnetics, IEEE Transactions on*, vol. 34, pp. 2885 –2888, Sept. 1998.

[21] S. Bureerat and J. Limtragool, "Performance enhancement of evolutionary search for structural topology optimisation," *Finite Elements in Analysis and Design*, vol. 42, pp. 547–566, Mar. 2006.

[22] S. Bureerat and T. Kunakote, "Topological design of structures using population-based optimization methods," *Inverse Problems in Science and Engineering*, vol. 14, no. 6, pp. 589–607, 2006.

[23] W. Khatib and P. Fleming, "The stud GA : a mini revolution ?," in *Parallel Problem Solving from Nature — PPSN V* (A. Eiben, T. Bäck, M. Schoenauer, and H. Schwefel, eds.), vol. 1498 of *Lecture Notes in Computer Science*, pp. 683–691, Springer Berlin / Heidelberg, 1998.

[24] C.Y. Wu and K.Y. Tseng, "Topology optimization of structures using modified binary differential evolution," *Structural and Multidisciplinary Optimization*, vol. 42, pp. 939–953, July 2010.

[25] X. He and L. Han, "A novel binary differential evolution algorithm based on artificial immune system," in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pp. 2267–2272, IEEE, Sept. 2007.

[26] A. Kaveh, B. Hassani, S. Shojaee, and S. Tavakkoli, "Structural topology optimization using ant colony methodology," *Engineering Structures*, vol. 30, pp. 2559–2565, Sept. 2008.

[27] K. Deb and T. Goel, "A hybrid Multi-Objective evolutionary approach to engineering shape design," in *Proceedings of the First International Conference on Evolutionary MultiCriterion Optimization (EMO-2001*, p. 385–399, 2001.

[28] D. Sharma, K. Deb, and N. Kishore, "Towards generating diverse topologies of path tracing compliant mechanisms using a local search based multi-objective genetic algorithm procedure," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 2004 –2011, June 2008.

[29] C. Im, H. Jung, and Y. Kim, "Hybrid genetic algorithm for electromagnetic topology optimization," *Magnetics, IEEE Transactions on*, vol. 39, no. 5, pp. 2163–2169, 2003.

[30] N. Wang and K. Tai, "A hybrid genetic algorithm for multiobjective structural optimization," in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pp. 2948–2955, 2007.

[31] N. Wang and Y. Yang, "Target geometry matching problem for hybrid genetic algorithm used to design structures subjected to un-

certainty," in *IEEE Congress on Evolutionary Computation, 2009. CEC '09*, pp. 1644 –1651, May 2009.

[32] K. Tai, N. Wang, and Y. Yang, "Hybrid GA multiobjective optimization for the design of compliant micro-actuators," in *IEEE International Conference on Systems, Man and Cybernetics, 2008. SMC 2008*, pp. 559 –564, Oct. 2008.

[33] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. ACM*, vol. 8, p. 212–229, Apr. 1961.

[34] C. Kane and M. Schoenauer, "Topological optimum design using genetic algorithms," *Control and Cybernetics*, vol. 25, no. 5, 1996.

[35] J. Dejonghe, "Allègement de platines métalliques par algorithmes génétiques, rapport de stage d'option b2 de l'ecole polytechnique."

[36] M. Schoenauer, L. Kallel, and F. Jouve, "Mechanical inclusions identification by evolutionary computation," *Revue européenne des eléments finis*, vol. 5, no. 5-6, pp. 619–648, 1996.

[37] B. Lu, H. Ou, and Z. S. Cui, "Evolutionary topological optimisation for preform design," *AIP Conference Proceedings*, vol. 1353, pp. 327–332, May 2011.

[38] K. Tai and T.H. Chee, "Genetic algorithm with structural morphology representation for topology design optimization," in *Proceedings of the 2nd International Conference on Mechanics in Design, Notthingham, U.K.*, pp. 827–836, 1998.

[39] K. Tai, S. Wang, S. Akhtar, and J. Prasad, "Structural topology optimization using a genetic algorithm and a morphological representation of geometry," *High Performance Computation for Engineered Systems (HPCES)*, 2003. Singapore-MIT Alliance (SMA).

[40] K. Tai and S. Akhtar, "Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme," *Structural and Multidisciplinary Optimization*, vol. 30, no. 2, pp. 113–127, 2005.

[41] K. Tai and J. Prasad, "Multiobjective topology optimization using a genetic algorithm and a morphological representation of geometry," in *6th World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil*, June 2005.

[42] K. Tai. and N. Wang, "An enhanced chromosome encoding and morphological representation of geometry for structural topology

optimization using GA," in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pp. 4178–4185, 2007.

[43] J.K. Guest, J.H. Prévost, and T. Belytschko, "Achieving minimum length scale in topology optimization using nodal design variables and projection functions," *International journal for numerical methods in engineering*, pp. 238–254, 2004.

[44] J. K. Guest and L. C. Smith Genut, "Reducing dimensionality in topology optimization using adaptive design variable fields," *International Journal for Numerical Methods in Engineering*, vol. 81, pp. 1019–1045, Feb. 2010.

[45] C.D. Chapman, K. Saitou, and M.J. Jakiela, "Genetic algorithms as an approach to configuration and topology design," *Journal of Mechanical Design*, vol. 116, pp. 1005–1012, Dec. 1994.

[46] O. De Weck and I.Y. Kim, "Variable chromosome length genetic algorithm for structural topology design optimization," *Strain*, vol. AIAA 2004-, no. April, pp. 1–12, 2004.

[47] O. De Weck and I.Y Kim, "Variable chromosome length genetic algorithm for progressive refinement in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 29, no. 6, pp. 445–456, 2005.

[48] J. Madeira, "Multi-objective topology optimization of structures," in *Proceedings of the sixth conference on Computational structures technology*, (Edinburgh, UK, UK), pp. 351–352, Civil-Comp press, 2002.

[49] J. A. Madeira, H. C. Rodrigues, and H. Pina, "Multiobjective topology optimization of structures using genetic algorithms with chromosome repairing," *Structural and Multidisciplinary Optimization*, vol. 32, pp. 31–39, Apr. 2006.

[50] C. Kane, *Algotithmes génétiques et optimisation topologiques de formes*. PhD thesis, Ecole Polytechnique, September 1995.

[51] C. Kane and M. Schoenauer, "Genetic operators for two dimensional shape optimization," in *Selected Papers from the European conference on Artificial Evolution*, (London, UK), pp. 355–369, Springer-Verlag, 1996.

[52] N. J. Radcliffe and P. D. Surry, "Fitness variance of formae and performance prediction," in *FOGA - Proceedings of the Third Work-*

shop on Foundations of Genetic Algorithms (L.D. Whitley and M.D. Vose, eds.), pp. 51–72, Morgan Kaufmann, 1994.

[53] T. Ishikawa, K. Yonetake, and N. Kurita, "An optimal material distribution design of brushless DC motor by genetic algorithm considering a cluster of material," *IEEE Transactions on Magnetics*, vol. 47, pp. 1310 –1313, May 2011.

[54] J. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press, Ann Arbor*, 1975.

[55] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Sciences, volume 220, No 4598*, pp. 671–680, 1983.

[56] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, pp. 1087–1092, June 1953.

[57] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, Jan. 1970.

[58] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*, 1995.

[59] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*, 1995.

[60] F. M. P. da Graca Lobo, "Implémentation de l'algorithme « particle swarm optimization » au sein d'un outil d'optimisation topologique," 2009.

[61] D.-i. Seo and B.-r. Moon, "A survey on chromosomal structures and operators for exploiting topological linkages of genes," in *In Genetic and Evolutionary Computation Conference*, p. 1357–1368, 2003.

[62] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *COMPLEX SYSTEMS*, vol. 6, p. 333–362, 1991.

[63] R. Cerf, "An asymptotic theory for genetic algorithms," in *Artificial Evolution* (G. Goos, J. Hartmanis, J. Leeuwen, J.-M. Alliot, E. Lut-

ton, E. Ronald, M. Schoenauer, and D. Snyers, eds.), vol. 1063, pp. 35–53, Berlin, Heidelberg : Springer Berlin Heidelberg, 1996.

[64] J. Denies, B. Dehez, and H. Ben Ahmed, "Genetic algorithm-based topology optimization : performance improvement through dynamic evolution of the population size," in *In Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion - Speedam 2012*, (Sorrento, Italy), June 20-22 2012.

[65] G. Syswerda, "Uniform crossover in genetic algorithms," in *, Proceedings of the 3rd International Conference on Genetic Algorithms, 1989*, pp. 2–9, 1989.

[66] J. Denies, B. Dehez, F. Glineur, and H. Ben Ahmed, "Impact of the material distribution formalism on the efficiency of evolutionary methods for topology optimization," in *Springer Book : Recent Advances in Optimization and its Applications in Engineering*, 2009.

[67] T. Ishikawa and K. Nakayama, "Topology optimization of rotor structure in brushless DC motor with concentrated windings using genetic algorithm combined with cluster of material," *IEEE Transactions on Magnetics*, vol. 48, pp. 899 –902, Feb. 2012.

[68] J. Denies, H. Ben Ahmed, F. Glineur, and B. Dehez, "Study of a topology optimization tool associating a hybrid algorithm with a voronoi-based formalism for the design of electromagnetic devices," in *2010 XIX International Conference on Electrical Machines (ICEM)*, pp. 1 –6, Sept. 2010.

[69] J. N. et R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308 – 313, 1965.

[70] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, (Hillsdale, NJ, USA), pp. 101–111, L. Erlbaum Associates Inc., 1985.

«Quand on ne travaillera plus les lendemains des jours de repos, la fatigue sera vaincue.»

Allais Alphonse, 1855-1905