

École Doctorale Sciences pour l'Ingénieur et Microtechniques
Université de Technologie de Belfort-Montbéliard

THÈSE

Présentée pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBELIARD ET
L'UNIVERSITÉ DE FRANCHE-COMTÉ

Spécialité : Sciences pour l'ingénieur

OPTIMISATION TOPOLOGIQUE DE STRUCTURES PAR ALGORITHMES GENETIQUES

Par :

Belkacem SID

JURY :

Président : **Zhi-Qiang FENG**, Professeur, Université d'Evry-Val d'Essonne

Rapporteurs : **Eduardo SOUZA DE CURSI**, Professeur, INSA de Rouen
Ying Qiao GUO, Professeur, Université de Reims Champagne Ardennes

Examineurs : **Matthieu DOMASZEWSKI**, Professeur, UTBM
François PEYRAUT, Professeur, UTBM
Jean-Claude GELIN, Professeur, Université de Franche-Comté

Soutenue le : 05 décembre 2006

Remerciements

Cette thèse a été effectuée au sein du laboratoire Mécatronique 3M (Méthodes, Modèles et Métiers) de l'Université de Technologie de Belfort-Montbéliard.

Mes premiers remerciements vont particulièrement à mon directeur de thèse, Monsieur **Matthieu DOMASZEWSKI** et mon co-directeur de thèse Monsieur **François PEYRAUT**. Qu'ils trouvent ici l'expression de ma profonde gratitude pour m'avoir accueilli au sein de leur équipe de travail et pour m'avoir dirigé et soutenu tout au long de cette thèse.

Je tiens à remercier mes rapporteurs, Monsieur **Eduardo SOUZA DE CURSI**, Professeur à l'INSA de Rouen, et Monsieur **Ying Qiao GUO**, Professeur à l'université de Reims Champagne Ardenne, pour leur patience en révisant mon travail de thèse.

Mes remerciements vont aussi aux membres de jury, Monsieur **Zhi-Qiang FENG**, Professeur à l'université d'Evry-Val d'Essonne, et Monsieur **Jean-Claude GELIN**, Professeur à l'université de Franche-Comté, qui ont accepté d'examiner ce travail de thèse.

Je tiens à remercier aussi mes collègues de laboratoire et tous ceux qui m'ont soutenu et aidé pour réaliser cette thèse. Un remerciement particulier à notre secrétaire dévouée Mme Béatrice ROSSEZ.

Enfin, un grand merci à ma famille, en particulier ma mère, pour leur soutien moral tout au long des quatre années de thèse.

Résumé

L'intérêt que suscite l'optimisation topologique s'explique par sa capacité à offrir des solutions innovantes permettant un gain notable de performance. Pendant la dernière décennie, des efforts considérables en recherche fondamentale ont été consacrés au développement de nouvelles méthodes, fiables et efficaces, pour l'optimisation topologique des structures. Cette thèse apporte une contribution dans ce domaine en proposant l'utilisation des algorithmes génétiques. Deux méthodes ont été développées. La première propose une approche par algorithme génétique basée sur une représentation graphique par courbes de Bézier. Le principe consiste à construire la structure en utilisant d'abord un squelette que l'on complète ensuite avec de la chair. Le squelette est composé par l'ensemble des courbes cubiques de Bézier reliant les surfaces d'application des conditions aux limites entre elles. La chair, sous forme de couches d'épaisseurs variables, complète le squelette pour donner la forme finale de la structure. Avec cette technique, les éléments de la structure sont tous reliés entre eux. Il n'y a donc pas de matière déconnectée ni problème de damier. La deuxième méthode propose une approche, toujours par algorithmes génétiques mais basée cette fois ci sur une représentation par adjacence. Le principe de base de cette représentation est fondé sur le concept de connectivité des éléments finis, considérés comme cellules. Ce principe s'exprime par une matrice d'adjacence similaire à celle utilisée en théorie des graphes. Le codage des structures solutions utilise cette matrice en transformant cette dernière en une chaîne binaire unidimensionnelle. Les deux méthodes proposées ont fait l'objet d'une implémentation informatique dans l'environnement de programmation Matlab. Elles ont été validées par des applications numériques sur des benchmarks simples en élasticité linéaire bidimensionnelle.

Mots clés

optimisation topologique, algorithme génétique, représentation graphique, représentation par adjacence

Abstract

The interest of topological optimization is explained by its capacity to offer innovating solutions allowing a notable profit of performance. During the last decade, considerable efforts in fundamental research were devoted to the development of new methods, reliable and efficient, for the structural topology optimization. This thesis contributes in this field by using genetic algorithms. Two methods were developed. The first proposes a genetic algorithm approach based on a graphical representation by Bézier curves. The principle consists in building the structure by a skeleton and a flesh. The skeleton is composed by the set of the Bézier cubic curves connecting the boundary conditions surfaces. The flesh, in form of layers, completes the skeleton to give the final form of the structure, completely connected. The second approach is based on a representation by adjacency. The principle of this representation is founded on the concept of connectivity of finite elements, considered as cells. This principle is expressed by an adjacency matrix similar to that used in the graph theory. The encoding of the structure solutions uses this matrix by transforming it into a binary string. The two proposed methods are implemented inside Matlab programming environment. They are validated by numerical applications on simple two-dimensional linear elasticity benchmarks.

Key words

topology optimization, genetic algorithm, graphical representation, adjacency representation

SOMMAIRE

Introduction générale	1
Chapitre I : Etat de l'art	5
1. Introduction	7
2. Généralités	9
3. Evolution historique de l'optimisation structurale	11
4. Méthodes d'optimisation topologique	13
4.1 Méthodes déterministes	13
4.1.1 La méthode d'homogénéisation	13
4.1.2 Les méthodes évolutives	15
4.1.3 Autres Méthodes	16
4.1.4 Limitations des méthodes déterministes	20
4.2 Méthodes Stochastiques	21
4.2.1 Recuit simulé	21
4.2.2 Algorithmes génétiques	22
4.2.2.1 Principe d'un algorithme génétique	24
4.2.2.2 Représentation des individus et codage	25
4.2.2.3 Initialisation	30
4.2.2.4 Evaluation	31
4.2.2.5 Reproduction	33
4.2.2.6 Sélection	33
4.2.2.7 Croisement	36
4.2.2.8 Mutation	38
4.2.2.9 Partage (Shearing)	39
4.2.2.10 Avantages et désavantages des algorithmes génétiques	41
5. Conclusions	43
Chapitre II : Optimisation topologique par algorithme génétique - Représentation graphique par courbes de Bézier	45
1. Introduction	47
2. Formulation du problème d'optimisation topologique	49
3. Problème mécanique	50
4. Algorithme génétique	52
5. Représentation graphique	53
5.1 Principe	54
5.2 Notre contribution	55
5.3 Codage	60

5.4	Décodage	62
5.5	Fonction performance	64
5.6	Analyse structurale	66
5.7	Traitement des limitations	66
5.8	Opérateurs génétiques.....	68
6.	Croisement et Mutation adaptatifs	76
7.	Implémentation	80
8.	Résultats et discussions	82
8.1	Treillis de Michell.....	82
8.2	Poutre console (cantilever)	89
8.3	Poutre MBB.....	95
9.	Conclusions.....	104
Chapitre III : Représentation par adjacence		105
1.	Introduction.....	107
2.	Représentation par adjacence.....	108
2.1	Principe.....	108
2.2	Codage.....	110
3.	Initialisation	112
4.	Evaluation	113
5.	Mutation.....	114
6.	Croisement.....	117
7.	Implémentation	117
8.	Résultats et discussion	118
8.1	Treillis de Michell.....	118
8.2	Poutre console (cantilever)	126
8.3	Une structure de type Michell.....	128
9.	Conclusions.....	131
Conclusions générales		133
BIBLIOGRAPHIE		137
Annexe		145

Introduction générale

Contexte de la thèse

Le travail présenté dans ce mémoire de thèse est dédié à l'optimisation topologique par algorithmes génétiques de structures continues. Les approches que nous proposons sont générales mais, pour des raisons pratiques liées au temps de calcul, nous nous restreindrons dans ce mémoire aux structures bidimensionnelles en élasticité linéaire.

L'optimisation topologique est actuellement un domaine en pleine effervescence. En effet, depuis les ouvrages de référence cités dans les études bibliographiques menées par Saitou et al. [Sai05] et par Eschenauer et Olhoff [Esc01], on constate une recrudescence des publications dans le domaine de l'optimisation topologique. Cette recrudescence a été particulièrement marquée à partir du milieu et jusqu'à la fin des années 90. Dans cette période, des efforts considérables en recherche fondamentale ont été consacrés au développement de nouvelles méthodes basées sur des approches micro-macro [Ben02]. L'intérêt que suscite l'optimisation topologique s'explique par sa capacité à offrir des solutions innovantes permettant un gain de performance nettement supérieur à ce que l'on peut obtenir par optimisation dimensionnelle ou par optimisation de forme.

Les méthodes d'optimisation peuvent se diviser en deux classes : les approches déterministes et les approches stochastiques. Les méthodes déterministes utilisent des informations sur les valeurs aux points courants de la fonction à optimiser, sur les limitations, ainsi que sur leur gradient. A l'inverse, les méthodes stochastiques ne nécessitent aucune hypothèse de régularité (continuité, différentiabilité) portant sur la fonction à optimiser ou sur les limitations imposées. Les avantages et les inconvénients des deux méthodes sont bien connus : les méthodes déterministes garantissent une convergence robuste, mais souvent vers l'optimum local le plus proche du point de départ. Les méthodes stochastiques, en particulier les algorithmes génétiques, sont capables de localiser l'optimum global, en se basant sur une recherche par probabilités, mais avec une vitesse de convergence généralement très lente.

Les algorithmes génétiques [Gol89] ont prouvés leur efficacité en optimisation dans différents domaines comme la médecine [Pen99], l'économie, la gestion de trafic (optimisation de trajet [Vae95]) ou encore dans l'ingénierie [Haj92]. Par contre, cette efficacité est fortement liée à un choix judicieux des paramètres de l'algorithme. Parmi tous ces paramètres, le codage est généralement considéré comme étant le plus important. En effet, le codage est basé sur la représentation des solutions, représentation qui doit décrire au mieux le problème traité. En optimisation topologique des structures continues, la représentation est souvent binaire puisqu'elle s'appuie généralement sur la présence ou l'absence de matière. Ce principe a été adopté dès les premiers travaux par Sandergen et ses collaborateurs [San90], par Jensen [Jen92], par Jakiela et ses collaborateurs [Cha94, Dud97, Jak00], par Fanjoy et Crossley [Fan02a, Fan02b] et récemment par Wang et Tai [Wan04]. Cependant, l'étude bibliographique menée dans cette thèse, montre que cette représentation est la principale cause des difficultés rencontrées. En effet, la représentation binaire souffre de plusieurs limitations dont la plus importante concerne les défauts de connectivité et la dépendance par rapport au maillage. La réparation de ces défauts est coûteuse et elle participe à la dégénérescence des chromosomes [Kan96], ce qui conduit à l'inefficacité de l'algorithme dans la recherche de l'optimum global.

L'étude bibliographique, présentée dans le chapitre I, montre aussi la possibilité de surmonter ces difficultés en proposant d'autres représentations. Hamda et ses collaborateurs [Ham00] ont proposé une représentation par des cellules de Voronoï qui dépassent la limitation de la représentation binaire concernant la dépendance par rapport au maillage. Tai [Tai98] a proposé une représentation graphique par courbes de Bézier qui assure la connectivité dans les structures générées. Nos travaux de thèse s'inscrivent dans ce dernier cadre. Plus précisément, nous avons centré nos travaux sur la recherche d'une représentation visant à améliorer l'efficacité de l'algorithme génétique dans le cadre de l'optimisation topologique des structures continues. Nos travaux ont été orientés dans deux directions faisant chacune l'objet d'un chapitre (chapitres II et III). La première a consisté à étendre et généraliser les travaux de Tai. Le but visé est l'amélioration des performances de l'algorithme génétique tout en conservant l'idée force de Tai qui permet d'assurer la connectivité dans les structures générées. La deuxième orientation sur laquelle nous avons travaillé propose des idées originales inspirées partiellement des méthodes déterministes (Automates cellulaires [Abd04, Tat00, Tov04]). Cette deuxième orientation vise à proposer une représentation qui soit adaptée à la problématique de l'optimisation topologique des structures continues. En effet, malgré son efficacité, que nous avons établie sur plusieurs exemples numériques, l'approche de Tai reste pour une large part dédiée à l'étude des mécanismes. Pour préserver la continuité des éléments, nous avons choisi une

représentation par adjacence. Le terme ‘*adjacence*’ est emprunté de la terminologie de la théorie des graphes [Dou01].

Dans les deux directions de recherche que nous avons choisies, nos travaux ont été guidés par le souci de proposer les approches les plus générales possibles, en essayant de s’affranchir, autant que faire se peut, de l’intuition qui prévaut souvent dans le choix des paramètres des algorithmes génétiques.

Plan de la thèse

Le mémoire de cette thèse est divisé en trois chapitres. Le premier est consacré à une étude bibliographique dont la plus grande partie concerne l’optimisation topologique des structures par algorithmes génétiques. Le deuxième chapitre est dédié aux extensions que nous avons apportées à l’approche de Tai. Le troisième chapitre porte sur la méthode par adjacence.

Le premier chapitre commence par un bref historique sur l’optimisation structurale. L’approche déterministe est ensuite évoquée au travers de la présentation des méthodes les plus connues. Les algorithmes génétiques, qui sont au cœur de ce travail, sont présentés dans le dernier paragraphe de ce chapitre. Après un résumé sur les contributions par algorithmes génétiques en optimisation topologique, le principe, ainsi que les paramètres de l’algorithme, sont présentés. Parmi tout ces paramètres, le codage, considéré comme étant le plus important, prend la plus grande part du paragraphe. Enfin, le chapitre se termine par la présentation des techniques de partage et de regroupement appelées souvent paramètres évolués.

Le deuxième chapitre décrit notre contribution sur la représentation graphique par courbes de Bézier. Notre travail s’inscrit dans la continuité des travaux de Tai qui a proposé l’idée pour la première fois en 1998 [Tai98]. Les premiers paragraphes sont consacrés aux développements théoriques de notre approche pour l’optimisation topologique des structures continues. Sur ces bases théoriques, nous avons développé un code de calcul en langage Matlab. Le dernier paragraphe est dédié à la validation du code de calcul à l’aide d’applications simples en élasticité linéaire bidimensionnelle. Le chapitre se termine par des conclusions et des perspectives.

Le troisième chapitre a pour objectif de proposer une contribution originale, toujours dans le contexte de l’optimisation topologique par algorithmes génétiques des structures continues. Nous visons à proposer une méthode générale, exempté de toute intuition, et compatible avec le problème posé. La théorie de cette contribution, dont le principe est inspiré partiellement de la méthode déterministe appelée ‘Automates cellulaires’ [Abd04,

Tat00, Tov04], fait l'objet des premiers paragraphes. L'idée de base consiste à résoudre le problème d'optimisation topologique en se basant sur le concept de connectivité des éléments finis, considérés comme cellules. Cette connectivité est exprimée par une matrice binaire similaire à une matrice d'adjacence utilisée en théorie des graphes. La représentation adoptée en utilisant la matrice d'adjacence offre beaucoup de souplesse pour la mise en œuvre d'un algorithme génétique. En effet, les chromosomes codés en utilisant la matrice d'adjacence sont similaires à ceux obtenus par codage binaire. Cela simplifier largement les opérations génétiques. Des applications numériques sont présentées dans le dernier paragraphe de ce chapitre.

Des conclusions générales clôturent ce mémoire de thèse. Elles synthétisent nos principaux résultats et exposent quelques perspectives qui ouvrent des voies pour des développements et des recherches ultérieures.

Chapitre I

Etat de l'art

Chapitre I	5
Etat de l'art	5
1. Introduction	7
2. Généralités	9
3. Evolution historique de l'optimisation structurale	11
4. Méthodes d'optimisation topologique	13
4.1 Méthodes déterministes	13
4.1.1 La méthode d'homogénéisation	13
4.1.2 Les méthodes évolutives	15
4.1.3 Autres Méthodes	16
4.1.4 Limitations des méthodes déterministes	20
4.2 Méthodes Stochastiques	21
4.2.1 Recuit simulé	21
4.2.2 Algorithmes génétiques	22
4.2.2.1 Principe d'un algorithme génétique	24
4.2.2.2 Représentation des individus et codage	25
4.2.2.3 Initialisation	30
4.2.2.4 Evaluation	31
4.2.2.5 Reproduction	33
4.2.2.6 Sélection	33
4.2.2.7 Croisement	36
4.2.2.8 Mutation	38
4.2.2.9 Partage (Shearing)	39
4.2.2.10 Avantages et désavantages des algorithmes génétiques	41
5. Conclusions	43

1. Introduction

Dans ce chapitre, nous présentons l'état de l'art de l'optimisation topologique des structures continues. Nous commençons par la formulation générale d'un problème d'optimisation structurale suivi par un bref historique de l'optimisation de structures. L'optimisation topologique de structures est la plus récente des méthodes d'optimisation structurale. Les premiers travaux dans ce domaine remontent à la fin des années quatre-vingt [Ben88]. Aujourd'hui, deux approches concurrentielles sont utilisées : les approches déterministes d'une part et les approches stochastiques d'autre part.

Les méthodes déterministes, dont la méthode d'homogénéisation est la plus connue [All02a, Ben03], sont présentées dans le paragraphe 4.1. Les méthodes stochastiques, dont la plus populaire est l'approche par algorithmes génétiques [Gol89], font l'objet du paragraphe 4.2. Cette dernière méthode a connu un regain d'intérêt ces dernières années grâce au développement d'outils de calcul numérique, notamment en calcul parallèle. Les contributions majeures en optimisation topologique par algorithmes génétiques [San90, Jen92, Cha94, Kan96, Fan02], seront d'abord présentées avant le principe général d'un algorithme génétique.

Parmi les différents paramètres d'un algorithme génétique, le codage est le plus important. Il doit en effet être compatible avec l'algorithme génétique pour permettre la convergence de celui-ci vers l'optimum global mais doit aussi être compatible avec le problème traité pour être représentatif de ce dernier. Le codage le plus fréquemment utilisé est basé sur une représentation en tableau binaire. Le principe de ce codage est d'associer à chaque élément fini du maillage une variable binaire indiquant la présence ou l'absence de matière dans cet élément. Ce principe, qui ne tient pas compte de la connectivité, peut conduire à la génération de structures comprenant des éléments et des blocs complètement déconnectés. Ces défauts induisent une instabilité numérique lors de l'analyse structurale par éléments finis. De plus, la longueur du chromosome, basé sur ce codage, est égale à la taille du maillage. Cette dépendance par rapport au maillage contribue à l'augmentation du temps de calcul sachant que ce dernier est proportionnel à la longueur du chromosome. Afin de remédier aux limitations de la représentation en tableau binaire, d'autres représentations ont été proposées. Hamda et ses collaborateurs [Ham00] ont proposé une représentation par des cellules de Voronoï afin de remédier au problème de la dépendance par rapport au maillage. Tai [Tai98] a proposé une représentation graphique en utilisant les courbes de Bézier dans le but de dépasser la limitation de la représentation par tableau binaire concernant la connectivité des structures.

Les autres paramètres de l'algorithme génétiques (sélection, croisement et mutation) sont bien sûr également très importants bien qu'à un degré moindre que le codage. La convergence de l'algorithme dépend en effet d'un choix judicieux de la méthode de sélection ainsi que les paramètres qui contrôlent les opérateurs de croisement et de mutation. Selon des études comparatives [Gol91], le choix de la méthode de sélection est crucial pour la performance de l'algorithme génétique. Concernant les opérateurs de croisement et de mutation, l'efficacité de l'algorithme est liée aux probabilités qui contrôlent le taux de ces opérations [Eib99], ainsi qu'aux types des opérateurs utilisées [Kan96, Spe92].

Afin d'améliorer l'exploration du domaine de conception, une technique dite de partage est parfois utilisée [Gol87]. Cette dernière s'accompagne souvent d'une technique de regroupement, appelée '*Clustering*', dans le but d'augmenter l'efficacité des opérations de croisement [Dud97].

2. Généralités

Face aux exigences d'un marché toujours plus concurrentiel, les industriels doivent impérativement réduire leurs coûts de production tout en proposant des produits toujours plus innovants. Ce défi devient aujourd'hui, plus que jamais, la première préoccupation des ingénieurs. C'est pourquoi l'optimisation s'impose comme un outil incontournable pour augmenter les performances tout en réduisant la masse des structures mécaniques, entraînant ainsi de substantielles économies.

L'optimisation est actuellement un domaine de recherche en pleine effervescence grâce au développement d'outils de modélisation, d'algorithmes et de moyens de calcul de plus en plus rapides et performants. Ce progrès a facilité le développement des méthodes d'optimisation notamment dans le domaine mécanique.

L'optimisation structurale en mécanique peut intervenir, soit par l'amélioration de l'existant, en modifiant des paramètres vis à vis de nouveaux objectifs ou limitations, soit par l'émergence de nouvelles conceptions par modification de la topologie. Un problème d'optimisation peut prendre la forme générale suivante :

$$\begin{array}{ll} \text{Minimiser :} & f(x) \\ \text{Sous limitations :} & \begin{array}{ll} g_i(x) \leq 0 & i = 1, \dots, K \\ h_j(x) = 0 & j = 1, \dots, L \end{array} \end{array} \quad (1-1)$$

où f est la fonction objectif que nous voulons minimiser et g et h sont les limitations à respecter.

Le volume et la compliance sont les deux types de fonction objectif habituellement considérées en optimisation de structures mécaniques. Si le volume est sélectionné comme fonction objectif, on lui associe en général une limitation sur la compliance. Cela se traduit par la recherche du volume minimum tout en ne descendant pas en dessous d'un seuil de rigidité qui représente la limitation. A l'inverse, si la compliance est sélectionnée comme fonction objectif, la limitation portera sur le volume. La problématique consiste alors à rechercher la structure la plus rigide possible (minimiser la compliance ou maximiser la rigidité), tout en restant en dessous d'un volume limite qui représente la limitation imposée. En d'autres termes, le premier problème revient à chercher parmi toutes les solutions ayant une rigidité donnée celle qui a le volume minimum. Le second problème consiste à rechercher parmi toutes les solutions d'un volume donné celle qui est la plus rigide.

L'optimisation de structures est habituellement divisée en trois catégories, par ordre croissant de difficulté et de généralité (Fig. 1-1).

- *L'optimisation dimensionnelle* qui consiste à ne retenir, comme variables de conception, que les dimensions transversales des composants structuraux telles que les sections de barres d'un treillis, les épaisseurs de membranes ou de plaques, ... etc.
- *L'optimisation de forme* qui permet de contrôler la géométrie de la structure à topologie fixe. Les variables de conception sont des paramètres décrivant les frontières du domaine, par exemple des coordonnées de points, des rayons de congés de raccordements ou des alésages. Elle demande une gestion délicate du modèle géométrique issu de la CAO et son coût de réalisation est essentiellement lié à la nécessité de modifier le modèle éléments finis au cours du processus d'optimisation.
- *L'optimisation topologique*, appelée aussi optimisation de forme généralisée, permet de contrôler non seulement la géométrie, mais aussi la topologie de la structure. Elle consiste à déterminer la nature et la connectivité des éléments constitutifs d'une structure pour laquelle les conditions aux limites et le domaine spatial dans lequel elle se situe sont spécifiés. Les variables de conception prennent alors le sens de variables logiques : présence ou absence de la matière. L'optimisation se ramène donc à la recherche de la distribution optimale de matière dans un domaine donné. Elle permet de déterminer les endroits où on peut éliminer la matière. Optimiser la topologie conduit naturellement à déterminer, d'une certaine manière, la forme ou les dimensions transversales optimales de la structure, de sorte que certains auteurs [All02a] lui attribuent aussi le nom d'*optimisation de forme généralisée*. Elle est potentiellement beaucoup plus intéressante, car elle ouvre la voie à une plus grande variété de conceptions.

La flexibilité disponible avec une topologie variable permet des gains de performances très significatifs et nettement supérieurs à ceux obtenus par une méthode d'optimisation de forme ou l'optimisation dimensionnelle. Pour des problèmes de conception courants, on peut estimer que les gains moyens de performances vont, en optimisation dimensionnelle, de 5 à 10 % et de 10 à 30 % en optimisation de forme [Duy96]. Avec une optimisation topologique, les gains obtenus sur la performance peuvent aller de 40 à 100 %. Cela montre la potentialité de l'optimisation topologique par rapport aux autres méthodes d'optimisation de dimensionnement et de forme.

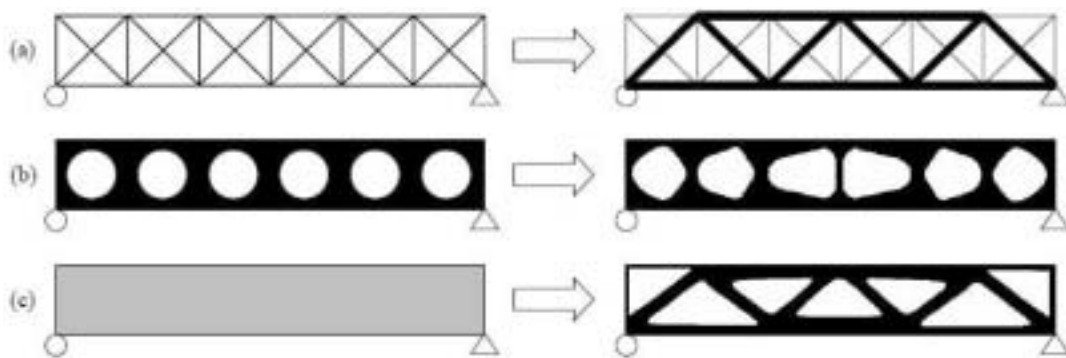


Figure 1-1. Les trois types d'optimisation structurale : (a) dimensionnelle (b) de forme (c) topologique. [Ben03]

3. Evolution historique de l'optimisation structurale

Historiquement, l'évolution de l'optimisation structurale remonte au début des années soixante. C'est à cette époque qu'ont été développées les bases théoriques modernes de l'optimisation des structures fondées sur les concepts de programmation mathématique et d'analyse de sensibilité [Sch63]. Paradoxalement, à cette époque, le '*Fully Stressed Design*' était la seule technique à être largement utilisée en pratique même s'il n'existait aucune justification théorique, en dehors de l'empirisme et de l'intuition des ingénieurs. Il a fallu attendre quelques années pour que les méthodes variationnelles et les conditions d'optimalité de Lagrange permettent de justifier les critères de '*Fully Stressed Design*' pour une classe de problèmes d'optimisation structurale [Pra68]. Les conditions d'optimalité du problème d'optimisation furent alors utilisées directement pour construire un algorithme de résolution itératif qu'on appelle la technique des critères d'optimalité [Sav85, Roz88].

A côté des méthodes de critères d'optimalité, le problème d'optimisation structurale fait des progrès rapides basés sur le développement d'analyse de sensibilité, le concept d'approximation structurale explicite et les méthodes de programmation mathématique [Sva87]. Le problème de dimensionnement automatique a bénéficié en premier lieu de ces recherches, de sorte que l'optimisation des dimensions transversales est actuellement un outil fiable. Outre les dimensions transversales des éléments structuraux, il est possible d'en faire varier la forme mais la topologie reste inchangée [Sva81].

Le problème d'optimisation de forme est plus complexe que le dimensionnement, puisque les modifications de la forme du contour extérieur entraînent des modifications du domaine intérieur, et dans ce cas il faut surmonter plusieurs difficultés. Le calcul des sensibilités est plus complexe qu'en dimensionnement. Un calcul mené de façon semi-analytique peut entraîner des erreurs importantes [Bar88]. D'autre part, pour maîtriser les

frontières du domaine on procède au remaillage, qui nécessite des mailleurs automatiques fiables. Malheureusement, lorsque les changements de forme deviennent très importants, il est parfois difficile de faire varier le modèle éléments finis sans introduire des éléments très distordus et, par conséquent, une erreur d'approximation très importante.

Pour généraliser l'optimisation de forme, il est nécessaire d'abandonner la représentation paramétrique pour décrire le domaine. Poser le problème sous forme d'une distribution de matière répond à ce critère. Mais une méthode de résolution générale restait inconnue jusqu'à ce que Bendsøe et Kikuchi [Ben88] suggèrent d'introduire la notion de microstructure poreuse et la théorie de homogénéisation dans le problème de distribution optimale de matière. La publication de Bendsøe et Kikuchi en 1988 constitue une date pivot et un tournant dans le domaine de la recherche en optimisation topologique.

Les premières techniques d'optimisation topologique remontent au début du vingtième siècle avec les travaux de Michell [Mic1904]. Les solutions intuitives proposées par ce chercheur australien concernent les structures naturellement discrètes (treillis de barres). Plus tard, et dans le même cadre, les premières formulations du problème ont été proposées dans le contexte de la programmation mathématique en utilisant le principe appelé univers structural ou '*Ground Structure*' [Kir93]. Dans cette approche, le domaine de conception est constitué d'un ensemble de nœuds, de coordonnées fixes, reliés entre eux par des barres. En faisant varier la section ou l'épaisseur de chaque barre entre zéro (dans ce cas l'élément disparaît) et certaines valeurs maximales, on génère une variété de solutions de topologies différentes (Fig. 1-2). Cependant, elle est inadaptée aux problèmes de grande taille car le processus d'optimisation topologique demeure indissociable du dimensionnement. La méthode a prouvé son efficacité pour résoudre les problèmes de petites tailles. Cependant, elle est inadaptée aux problèmes de grandes tailles.

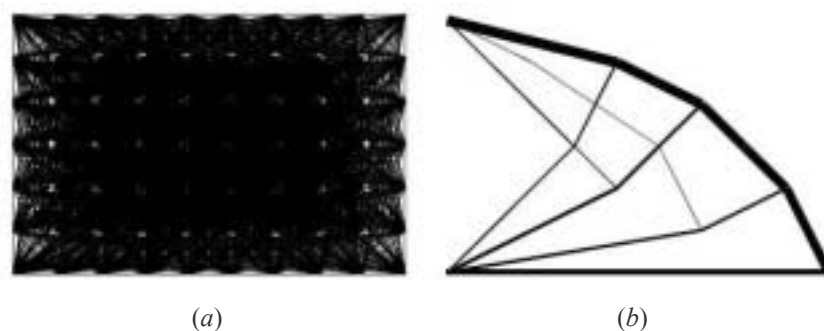


Figure 1-2. L'approche '*Ground Structure*' (a) univers structural (b) solution optimale. [Bek97]

En introduisant le concept de Bendsøe qui consiste à relaxer les propriétés physiques du matériau composant le domaine de conception, les recherches en optimisation topologique sont désormais tournées vers les structures continues (membranes, plaques et volumes). Les structures continues sont souvent discrétisées en éléments finis et les variables de conception, dont le nombre est égale à celui des éléments finis, sont de type logique (0 ou 1) exprimant l'absence ou la présence de matière.

4. Méthodes d'optimisation topologique

Deux catégories de méthodes d'optimisation topologique structurale, basées sur deux principes différents, sont présentées dans cette section. La première catégorie regroupe les méthodes classiques qui adoptent un principe déterministe, basé sur l'analyse de sensibilité. La deuxième catégorie utilise des principes stochastiques.

4.1 Méthodes déterministes

Dans ce paragraphe nous présentons chronologiquement les méthodes d'optimisation structurales les plus populaires, dites déterministes.

4.1.1 La méthode d'homogénéisation

L'optimisation topologique des milieux continus élastiques débute avec les travaux de Bendsøe et Kikuchi [Ben88]. La méthode qu'ils proposent pour résoudre les problèmes d'optimisation topologique, et qui est maintenant connue sous le nom de méthode d'homogénéisation, est fondée sur les idées de distribution optimale de matière et de relaxation. Pour cela, Bendsøe et Kikuchi [Ben88] introduisent une microstructure poreuse dont ils déterminent les propriétés macroscopiques en fonction de la densité grâce à la théorie de l'homogénéisation des microstructures [Ben78]. Une microstructure périodique est utilisée avec une porosité variable provenant d'une perforation rectangulaire au sein de la cellule de base (Fig. 1-3). La théorie de l'homogénéisation, valable pour tout matériau composite de microstructure périodique, est utilisée pour déterminer les propriétés macroscopiques équivalentes. Les variables de conception sont les dimensions de la perforation de la cellule de base au sein de chaque élément fini et, comme le matériau homogénéisé est orthotrope, l'orientation des microstructures locales est également une variable de conception du problème d'optimisation. Le problème numérique d'optimisation étant de très grande taille, on adopte le principe de minimisation du travail des charges extérieures appliquées (encore appelé compliance) pour l'aborder.

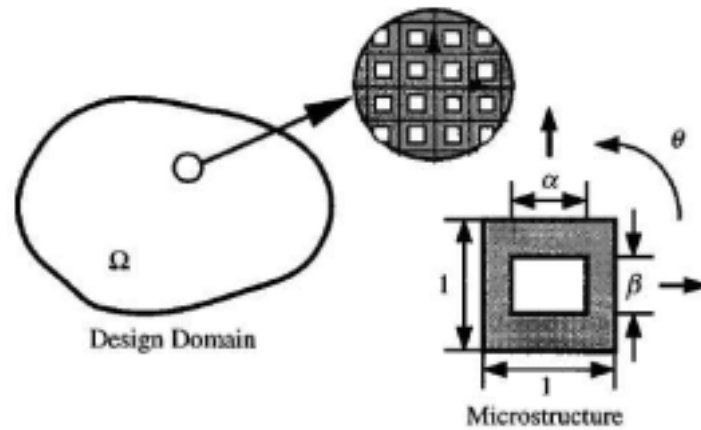


Figure 1-3. Principe d'homogénéisation avec microstructures.

Le choix de la microstructure poreuse a fait l'objet de nombreux travaux [Wan03]. Dans sa version primitive, la méthode d'homogénéisation s'est tournée vers les micro-perforations rectangulaires mais, très tôt, on a proposé de nouvelles microstructures et d'autres lois donnant la raideur en fonction de la densité relative. Malheureusement, la technique de microstructure conduit à des résultats qui sont souvent inutilisables pour la pratique de l'ingénieur. En effet, la structure solution est largement remplie de zones de matériaux composites de densité comprises entre le vide et le solide. Ces microstructures ne sont ni usinables, ni faciles à interpréter pour dégager un modèle CAO qui servirait à un autre type d'optimisation. Pour améliorer la séparation entre les régions occupées par le vide et le solide, et pour donner lieu à des solutions qui soient plus directement utilisables dans la pratique, on préfère recourir à des microstructures sub-optimales.

Simultanément à l'introduction des microstructures, Bendsøe [Ben89] a suggéré d'employer une loi dans laquelle les densités intermédiaires sont pénalisées par une loi de type puissance (*Power Law*). Le module d'élasticité E_i de chaque élément fini est exprimé en fonction de la densité relative $\bar{\rho}_i$ et de la valeur maximale E_0 :

$$E_i = \bar{\rho}_i^p E_0 \quad (p \geq 1) \quad (1-2a)$$

$$\rho_i = \bar{\rho}_i \rho_0 \quad (0 \leq \bar{\rho} \leq 1) \quad (1-2b)$$

ρ_0 et E_0 sont la densité et le module d'élasticité du matériau.

La méthode est appelée souvent *SIMP* (Solid Isotropic Material with Penalization). La méthode *SIMP* a été proposée, sous le nom 'Approche directe' ou 'Approche par densité artificielle', par Bendsøe en 1989 [Ben89]. Elle a été développée, et largement utilisée par

son groupe de recherche depuis 1990 (<http://www.topopt.dtu.dk>). Le nom *SIMP* a été proposé en 1992 par le même auteur.

Dès le début des années quatre-vingt dix, plusieurs chercheurs ont contribué au développement de la méthode d'homogénéisation. Parallèlement à la méthode *SIMP*, de nouvelles microstructures ont été proposées. Allaire [All02a], propose la méthode *OMP* (*Optimal Microstructures with Penalization*). Mais la méthode *SIMP* n'a pas cessé de se développer, et la contribution la plus importante est celle appelée 'Schémas d'interpolation' de Bendsøe et Sigmund [Ben99]. Depuis cette date, la méthode *SIMP* est devenue la plus utilisée en optimisation topologique. Ses avantages par rapport à d'autres méthodes sont présentés dans [Roz01a].

La méthode *SIMP* a été utilisée avec la méthode de critère d'optimalité ou avec la programmation mathématique. Par conséquent, les performances de cette méthode ont considérablement augmenté, et le domaine d'étude a été étendu à une grande variété de problèmes : *charges de pression, systèmes multi-physiques, nouvelle classe de composites, non-linéarité géométrique, interaction fluide/structure, ...* [Roz01a].

4.1.2 Les méthodes évolutives

En parallèle des méthodes d'homogénéisation, d'autres méthodes basées sur l'approche évolutive ont été proposées. Ces méthodes trouvent leur origine dans le principe intuitif des critères d'optimalité '*Fully Stressed Design*'. Le principe consiste à éliminer progressivement du domaine de conception la matière jugée inutile. Le processus se fait, soit par élimination des espaces occupés par les éléments de faibles valeurs de contraintes locales, soit par manipulation des propriétés physiques du matériau, en affectant de faibles valeurs aux modules d'élasticité de ces éléments. Plusieurs méthodes ont été développées sur ce principe telle la méthode *ESO* (*Evolutionary Structural Optimization*) [Xie93, Xie97, Lia00, Tan02] ou encore les méthodes appelées '*Hard kill*' et '*Soft kill*' [Pap96].

En 1993, Y.M. Xie et G.P. Steven ont proposé une approche appelée '*Evolutionary Structural Optimization*' (*ESO*). Cette méthode est basée sur l'élimination des éléments ayant un niveau de contrainte de Von Mises faible. Le niveau de contrainte de chaque élément est déterminé en comparant la contrainte de cet élément σ_e^{vm} à la contrainte de Von Mises maximale σ_{Max}^{vm} . Après chaque analyse élément fini, les éléments vérifiant la condition suivante sont éliminés de la structure :

$$\frac{\sigma_e^{vm}}{\sigma_{Max}^{vm}} < RR_i \quad (1-3)$$

où RR_i représente le taux d'élimination de l'itération en cours.

L'opération d'élimination continue avec le même taux RR_i jusqu'à l'élimination de tous les éléments vérifiant la condition (1-3). A cette étape, un paramètre ER dit 'Evolutionary Rate' est introduit et ajouté au taux d'élimination :

$$RR_{i+1} = RR_i + ER \quad (1-4)$$

Le processus évolutif se poursuit jusqu'à la convergence. En se basant sur ce critère, la solution optimale recherchée est une structure ayant une distribution quasi-uniforme des contraintes locales de Von Mises, ce qui est en réalité l'ancien concept intuitif de la méthode de critère d'optimalité '*Fully Stressed Design*'. Cette méthode évolutive requiert le choix de deux paramètres. Un taux d'élimination initial RR_0 dont la valeur est de l'ordre de 1 %, et le taux évolutif ER d'une valeur souvent égale à 1 %.

La méthode ESO semble avoir quelques avantages par rapport aux autres méthodes d'optimisation topologique. Elle est d'abord très simple à implémenter numériquement ou à coupler avec un logiciel d'analyse de structure basé sur la méthode des éléments finis. De plus, le coût du calcul structural devient de plus en plus faible en approchant de la solution finale en raison de l'élimination progressive des éléments du maillage. La simplicité de la méthode ESO a été prouvée dans plusieurs applications concernant des problèmes discrets ou continus. Cependant, la méthode ESO n'a pas de bases théoriques encore établies, et elle est considérée à l'heure actuelle comme une méthode intuitive. Enfin, dans certaines situations, Zhou et Rozvany [Zho01] ont prouvé que cette méthode ne converge pas toujours vers des solutions optimales.

4.1.3 Autres Méthodes

Depuis les premiers travaux en optimisation topologique par homogénéisation, l'optimisation topologique bénéficie de l'apport constant de nouvelles approches et méthodes proposées par les chercheurs. L'objectif de paragraphe est de présenter quelques unes de ces méthodes parmi les plus connues et les plus efficaces.

▪ *Approche duale*

Le problème dual consiste à maximiser une fonction qui ne dépend que des multiplicateurs lagrangiens associés aux contraintes. Il est continu et quasi non contraint, mais sa non-différentiabilité requiert l'utilisation de méthodes basées sur les sous-gradients. Dans ce contexte, M. Beckers [Bec97, Bec99] a réussi à résoudre en variables discrètes les problèmes d'optimisation topologique de très grande taille dans le domaine de l'élasticité linéaire. La méthodologie consiste à générer des sous-problèmes discrets

approchés par des schémas d'approximation convexes et séparables. Chacun des sous-problèmes est résolu dans un espace dual avec un algorithme basé sur le sous-gradient.

En optimisation structurale, un problème discret standard s'écrit sous la forme [Bec97] :

$$\begin{cases} \min_x g_0(x) \\ \text{avec} \begin{cases} g_j(x) \leq g_j^{\max} & j = 1, m \\ x_i \in X_i & i = 1, n \end{cases} \end{cases} \quad (1-5)$$

Le vecteur $x = (x_1, \dots, x_n)$ contient les n variables de conception. L'ensemble X_i représente la liste des valeurs admissibles associées à la $i^{\text{ème}}$ variable et g_0 la fonction objectif.

L'approximation convexe consiste à remplacer toutes les fonctions $g_h(x)$, au point x^k , par les approximations convexes et séparables $\hat{g}_h(x)$ données par l'expression :

$$\hat{g}_h(x) = g_h(x^k) + \sum_{\frac{\partial g_h(x^k)}{\partial x_i} > 0} \frac{\partial g_h(x^k)}{\partial x_i} (x_i - x_i^k) - \sum_{\frac{\partial g_h(x^k)}{\partial x_i} < 0} (x_i^k)^2 \frac{\partial g_h(x^k)}{\partial x_i} \left(\frac{1}{x_i} - \frac{1}{x_i^k} \right) \quad (1-6)$$

Les structures continues sont discrétisées en éléments finis et chaque élément fini est associé à une variable binaire dont la valeur '1' indique la présence de l'élément et '0' indique son absence.

▪ **Méthode de lignes de niveaux (level-set)**

G. Allaire, F. Jouve et A.M. Toader [All02b] étudient des problèmes d'optimisation de formes en utilisant une représentation des formes par la méthode 'level-set'. L'idée est de reprendre les résultats fondateurs d'Hadamard au début du 20^{ème} siècle et de Murat-Simon de la fin des années 70, cités dans [All02b], concernant l'optimisation de formes dite 'classique' (aussi appelées méthodes de variation de frontières ou de variation de domaine) et de les adapter à une représentation des formes de type 'courbe de niveau'. Les formes sont capturées sur un maillage fixe, et la dérivée de forme est utilisée comme une vitesse de transport dans une équation d'Hamilton-Jacobi. Bien que cette méthode ne soit pas spécifiquement conçue pour faire de l'optimisation topologique, elle permet très facilement les changements de topologie de la forme d'une structure pour des fonctions objectif très générales. Son coût en temps de calcul est modéré puisqu'il s'agit d'une méthode numérique de capture de formes sur maillage Eulérien fixe. On évite ainsi les inconvénients des méthodes classiques d'optimisation de forme dû aux problèmes de

remaillage successif du domaine. La méthode a été implémentée et testée en 2D et en 3D [All02b].

La contribution de Wang et *al.* [Wan03] combine la méthode de lignes de niveaux avec la programmation mathématique. Dans ce travail, les auteurs ont montré la flexibilité de la méthode avec le changement de la topologie dans des applications 3D.

Cependant, contrairement à la méthode d'homogénéisation, on n'a plus l'assurance de trouver un optimum global de la fonctionnelle à optimiser, et les solutions obtenues dépendent fortement de la configuration initiale (Fig. 1-4).

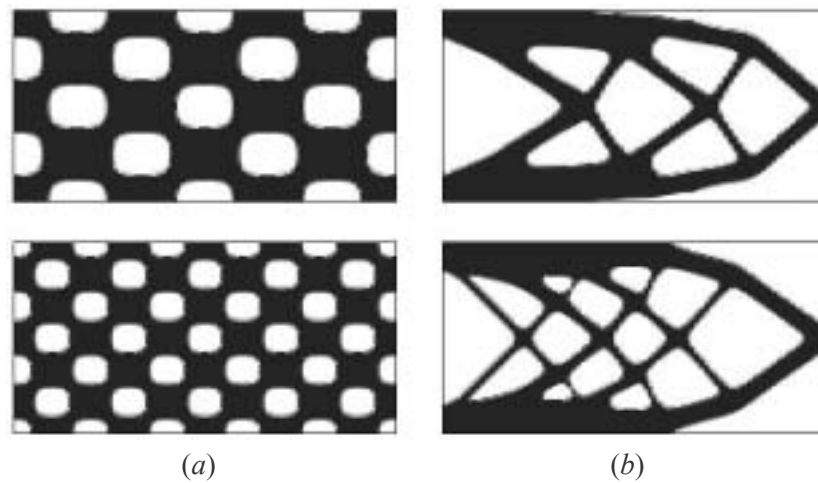


Figure 1-4. Méthode de lignes de niveaux (a) initialisations, (b) solutions optimales. [All02b]

▪ Méthode d'automates cellulaires (*Cellular Automata*)

Le paradigme des automates cellulaires, dont le principe est basé sur des règles de transitions locales entre une population de cellules, a été fondé et utilisé dans les années quarante. Cependant, ce paradigme n'est devenu populaire que récemment grâce au développement du calcul parallèle. Ce principe a été utilisé la première fois par Kita et Toyoda [Kit00] et par Tatting et Gürdal [Tat00] pour l'optimisation des structures continues. Le domaine de conception a été discrétisé en éléments finis, chaque élément représentant une cellule. La recherche de la solution optimale est basée sur la mise à jour des paramètres d'optimisation dans chaque cellule en fonction des paramètres des cellules de voisinage (Fig. 1-5). Chaque cellule, à la position r , est définie par un ensemble de variables d'état $S(S_1, \dots, S_n)$. L'évolution de chaque variable d'état, $S_i(r, t+1)$, est contrôlée par une règle locale :

$$S_i(r, t+1) = R_i(S(r, t), S(r+\Delta_1, t), \dots, S(r+\Delta_N, t)) \quad (1-7)$$

Où $r+\Delta_j$ désigne la position associée à une cellule voisine. L'ensemble des règles locales $R(R_1, \dots, R_n)$ sont identiques pour toutes les cellules et sont appliquées simultanément. Dans les premiers travaux utilisant cette approche [Kit00], une règle basée sur le principe FSD 'Fully Stressed Design' a été largement utilisée.

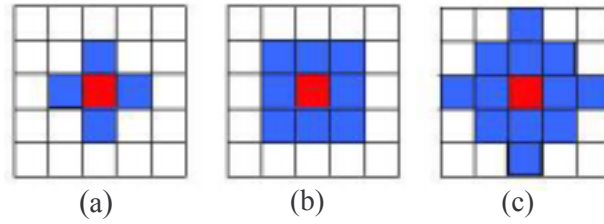


Figure 1-5. Voisinage d'une cellule (a) Von Neumann ($N = 4$), (b) Moor ($N = 8$), (c) MvonN ($N = 12$). [Tov04]

La méthode des automates cellulaires a été adoptée par plusieurs auteurs. Abdalla et Gürdal [Abd04] ont utilisé une règle locale basée sur le critère d'optimalité interprétée par la condition de Kuhn-Tucker. Une technique hybride a été proposée par Tovar et al. [Tov04], où la règle locale est inspirée de la modélisation des os. La variable de conception est la densité dont la valeur est calculée à chaque itération par la règle locale. Le principe 'Power Law' de la méthode SIMP a été utilisé ensuite pour calculer le module d'élasticité en fonction de la densité. Dans ce cas, l'état de chaque cellule $S_i(S_{Di}, \dots, S_{Fi})$ est défini par des variables de conception S_{Di} (propriétés géométriques et physiques par exemple) et des variables de champ S_{Fi} (contraintes mécaniques, déformations). La règle locale modifie les variables de conception en conduisant les variables de champ vers des valeurs cibles S_{Fi}^* . Le calcul s'effectue selon l'algorithme itératif présenté sur la figure 1-6.

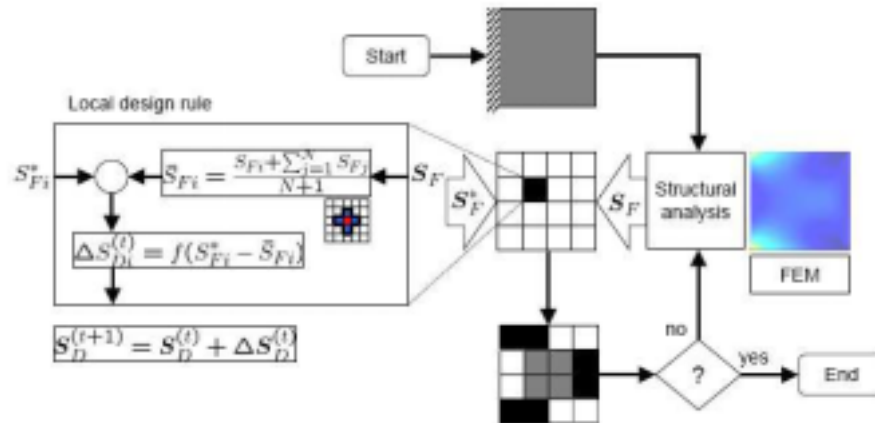


Figure 1-6. Algorithme de la méthode 'Hybrid Cellular Automata' [Tov04]

L'avantage de cette méthode est l'indépendance par rapport au maillage ainsi que la stabilité numérique grâce à l'absence de damiers (la notion de damiers, checkerboarded en anglais, est présentée dans le paragraphe suivant). En revanche, la convergence vers la solution optimale nécessite des milliers d'itérations à cause de la forte non-linéarité du problème (Fig. 1-6).

4.1.4 Limitations des méthodes déterministes

Les méthodes classiques d'optimisation par approche déterministe, sont confrontées à de sérieuses difficultés malgré les progrès de ces dernières années. L'une d'entre elles concerne la stabilité numérique. Dans les méthodes adoptant le principe de relaxation des propriétés physiques, la méthode d'homogénéisation par exemple, la difficulté est liée à la présence de damiers (Fig. 1-7). Le phénomène de damier consiste en une distribution de densité du matériau, dans les éléments finis, de manière périodique à la manière des cases noires et blanches dans le jeu de dames. Enfin, on résout les problèmes discrets par les méthodes déterministes sans tenir compte du caractère discret des variables, puis on arrondit les résultats aux valeurs discrètes admissibles les plus proches. Cela peut conduire à une solution éloignée de l'optimum et même parfois inadmissible. Beckers [Bek97] a réussi à résoudre les problèmes discrets en utilisant l'approche duale, mais sa méthode reste limitée au traitement des fonctions objectif dérivables. En fait, la plupart des méthodes déterministes sont limitées aux traitements des fonctions objectif différentiables.



Figure 1-7. Problème de damiers.

Zhou et Rozvany [Zho01] ont prouvé dans des applications simples que la méthode appelée 'ESO' donne des solutions non optimales. Le principe intuitif des critères d'optimalité '*Fully Stressed Design*', sur laquelle cette dernière méthode est basée, a été critiqué dans un autre travail de Rozvany [Roz01b]. Dans ce dernier article, l'auteur critique également les méthodes d'optimisation basées sur la minimisation de la compliance qui peuvent aboutir à des solutions non optimales dans certains problèmes. De plus, la stabilité numérique de la méthode évolutive ESO dépend fortement de ses paramètres RR et ER représentant les taux d'élimination et d'évolution respectivement.

Les méthodes déterministes utilisent souvent l'analyse de sensibilité. La technique de calcul de sensibilité (gradient), n'est pas une tâche facile car les fonctions objectifs ne sont pas toujours différentiables, et un calcul semi-analytique ou par différences finies peut aboutir à des approximations importantes, sans parler du coût de ce calcul qui s'ajoute au coût des analyses structurales. C'est pourquoi le développement de méthodes non basées sur le gradient présente un intérêt certain. Il s'agit des méthodes d'ordre zéro, capables de travailler sur n'importe quel domaine physique et permettent d'optimiser des fonctions non différentiables, voire discontinues.

4.2 Méthodes Stochastiques

Les phénomènes physiques ou biologiques ont été la source d'inspiration de nombreux algorithmes de calcul numérique pour la modélisation, et particulièrement pour l'optimisation. Ainsi l'algorithme de recuit simulé est inspiré de la thermodynamique et les algorithmes évolutionnaires (AEs) (dont les plus connus sont les algorithmes génétiques) sont inspirés de l'évolution Darwinienne des populations biologiques.

Les algorithmes évolutionnaires sont des méthodes stochastiques. Le caractère global d'optimisation, ainsi que la souplesse de ces algorithmes à traiter des fonctions objectif non régulières ou des fonctions définies sur des espaces de recherche non standards (espaces discrets,...etc.), permet de les appliquer aux problèmes d'optimisation structurales qui ne peuvent être résolus par des méthodes déterministes.

4.2.1 Recuit simulé

La méthode de recuit simulé a été développée dès le début des années quatre-vingt par Kirkpatrick et al. [Kir83]. Elle est basée sur la technique du recuit, utilisée en métallurgie, qui consiste à chauffer un solide métallique jusqu'à sa température de fusion, puis à le refroidir lentement jusqu'à la température ambiante, afin que les atomes aient le temps de prendre des positions qui minimisent l'énergie. L'algorithme simulant ce processus consiste, à une certaine température, de perturber d'une manière aléatoire la position des atomes de manière à avoir un nouvel état d'énergie. Dans le cas d'une augmentation d'énergie, une décision probabiliste aléatoire juge l'acceptation ou le rejet de cet état d'autant plus sévèrement que la température est faible. D'autres perturbations sont alors générées, jusqu'à obtention d'un état d'énergie minimale. L'application à l'optimisation topologique de structures repose sur l'analogie suivante : la fonction objectif est assimilée à l'énergie et les variables de conception aux positions des atomes. La température devient un paramètre de contrôle de la convergence de l'algorithme.

Le domaine de conception est discrétisé en éléments finis où chaque élément contient de la matière ou du vide. Il n'y a pas de densités intermédiaires. La méthode de recuit simulé travaille donc sur un domaine binaire $\{0, 1\}$.

La caractéristique principale de cette méthode est d'accepter, avec une certaine probabilité, l'augmentation de la fonction objectif. On permet ainsi de se dégager des optima locaux et de continuer la recherche. Mais le problème principal pour atteindre le minimum global dans un temps fini est la détermination de l'ensemble des paramètres qui gouvernent la convergence de l'algorithme. C'est ce qu'on appelle le 'schéma de refroidissement'. Ces paramètres sont la valeur initiale du paramètre de contrôle (la température initiale), le facteur de réduction de température, le nombre d'itérations et enfin le critère d'arrêt.

4.2.2 Algorithmes génétiques

Les algorithmes génétiques sont des méthodes d'optimisation stochastiques maintenant bien connues, depuis le travail de J. Holland [Hol75], consolidé par le livre de D. Goldberg [Gol89]. Plus récemment, le cadre initial strict des chaînes de bits de longueur fixe a été étendu à des espaces de recherche quelconques, montrant clairement que les algorithmes de base peuvent être modifiés pour tenir compte des particularités du problème à traiter. Il est aujourd'hui admis que le terme *algorithmes évolutionnaires* recouvre toutes ces variantes.

De telles méthodes peuvent permettre, en calcul scientifique comme dans bien d'autres domaines, de dépasser certaines limitations des méthodes classiques. Elles sont en effet capables de travailler sur des espaces discrets aussi bien que continus (voir des espaces mixtes, comme dans le domaine de l'optimisation topologique des structures en treillis [Ohs95]). Ce sont des méthodes d'ordre zéro particulièrement efficaces lorsque la fonction objectif possède de nombreux optimaux locaux qui pénalisent des méthodes classiques telles que les algorithmes de type gradient.

La plupart des problèmes d'optimisation sont de nature multi-objectif. L'optimisation multi-objectif cherche à optimiser plusieurs composantes d'un vecteur de fonction coût. Contrairement à un problème mono-objectif, la solution n'est pas unique mais constituée d'un ensemble de solutions dites de Pareto optimales (Fig. 1-8). Sur cette figure, on a représenté la surface de Pareto d'une fonction coût bijectif associée à un problème de minimisation de section droite de poutre, les sections droites sont symbolisées par les carrés. Cependant, Φ_1 et Φ_2 sont les composantes du vecteur de la fonction coût. La difficulté principale d'un problème multi-objectif ne consiste pas à chercher une solution

optimale mais à déterminer l'ensemble des solutions optimales pour lequel on ne pourra pas effectuer une opération de classement. Les méthodes d'optimisation multi-objectif sont donc des méthodes d'aide à la décision pour lesquelles les algorithmes génétiques semblent particulièrement adaptés.

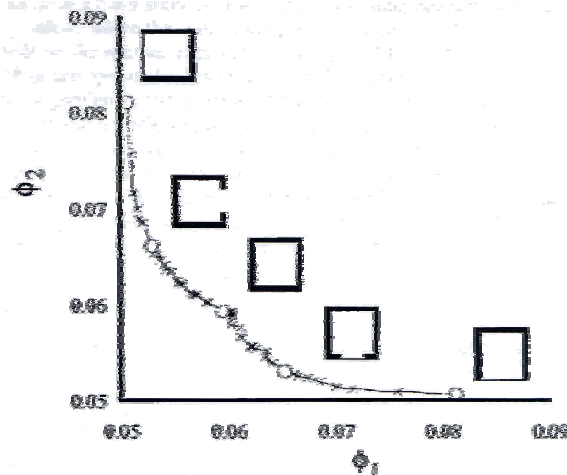


Figure 1-8. Exemple d'un problème multi-objectif. [Fan02b]

Les premiers travaux d'optimisation topologique par algorithmes génétiques datent du début des années quatre-vingt dix avec les travaux de Sandergen et de ses collaborateurs [San90]. Jensen [Jen92] a exploité les algorithmes génétiques pour l'optimisation de la section d'une poutre. Plus tard, son travail a été étendu pour l'optimisation des structures continues : Chapman and al. [Cha94] ont optimisé une plaque console, Chapman et Jakiela [Cha96] ont montré la souplesse des algorithmes génétiques pour travailler avec différentes fonctions objectifs, Duda et Jakiela [Dud97] ont utilisé les techniques de partage et de regroupement dans le but d'améliorer l'efficacité de l'algorithme. D.W. Fanjoy et W.A. Crossley [Fan02] ont optimisé la section d'une poutre soumise à des charges multiples de flexion et de torsion. En France, le problème d'optimisation topologique des structures continues a été abordé par C. Kane et ses collaborateurs [Kan95, Kan96], et par H. Hamada et ses collaborateurs en proposant une nouvelle représentation basée sur les sites de Voronoï [Ham00]. Tai et ses collaborateurs ont proposé une nouvelle représentation graphique qui s'affranchit du codage binaire traditionnel [Tai98, Tai00]. Enfin, toujours dans le cadre d'optimisation des structures continues, les algorithmes génétiques ont été combinés avec d'autres méthodes déterministes telle que la méthode des automates cellulaires [Haj99].

A cause de leur nature combinatoire, les problèmes discrets peuvent être résolus avec efficacité par les algorithmes génétiques. Les travaux d'optimisation de structures en treillis constituent une part importante des travaux d'optimisation par algorithmes

génétiques à cause de la nature discrète de ces problèmes. Citons le travail de Ohsaki [Ohs95] pour l'optimisation d'un système de barres sur un domaine de conception mixte (continu, discret). Wu et Chow [Wu95] ont proposé une stratégie basée sur les algorithmes génétiques pour un problème de treillis mixte sous contraintes non linéaires. Hajela et Lee [Haj95] ont utilisé une technique de deux niveaux pour un problème de treillis sous contraintes de stabilité cinématique. Coello et Christiansen [Coe00] ont résolu un problème d'optimisation multi-objectif et Azid and al. [Azi02] ont proposé une technique pour résoudre un problème sous limitations de déplacement et de contraintes.

Une étude bibliographique très riche a été présentée par R. Kicinger et al. [Kic05]. Cette étude regroupe des travaux de recherche récents dans le domaine de l'optimisation structurale par des méthodes évolutionnaires, en particulier par algorithmes génétiques.

4.2.2.1 Principe d'un algorithme génétique

Les algorithmes génétiques sont inspirés des mécanismes de la sélection naturelle et de la génétique. Ils utilisent les principes de la survie des individus les mieux adaptés. C'est Holand [Hol75] qui a posé les fondements théoriques des algorithmes génétiques, passant du paradigme Darwinien de l'évolution naturelle à celui de l'évolution artificielle. Une nouvelle étape est franchie lorsque les travaux de Goldberg [Gol89], vers le milieu des années quatre-vingt, donnent aux algorithmes génétiques leurs lettres de noblesse en tant que méthode d'optimisation viable, efficace et non spécifique.

Pour optimiser une fonction objectif donnée f dans un espace de recherche Ω , on définit une fonction performance 'fitness' reliée à cette fonction objectif. Les algorithmes génétiques maximisent souvent la fonction performance afin de s'adapter au principe Darwinien. Les étapes principales d'un algorithme génétique sont les suivantes :

La première étape d'optimisation par algorithme génétique consiste à initialiser une population avec une méthode complètement aléatoire. La population se compose d'un ensemble d'individus (points dans l'espace de recherche), solutions possible du problème. Un individu, appelé généralement 'chromosome', est constitué de gènes en chaîne de bits codés souvent en binaire '0-1'. A chaque individu, on attribue une fonction performance évaluant le mérite de cet individu en tant que solution possible du problème. La population évolue en une succession de générations en respectant le principe que les individus les plus adaptés (en terme de valeurs de la fonction performance) survivent et se reproduisent.

La deuxième étape est la reproduction, dont la phase la plus importante est la sélection. Les opérateurs de sélection s'appliquent sur une population pour créer une population intermédiaire. Dans le but d'explorer le domaine de recherche, les opérateurs de

croisement et de mutation s'appliquent ensuite sur la population intermédiaire pour créer une nouvelle population. Le croisement est un opérateur de reproduction qui donne naissance à deux nouveaux individus en échangeant des parties de deux individus qualifiés de parents. L'opérateur de mutation consiste pour sa part à perturber les valeurs de quelques gènes du chromosome dans le but d'augmenter les capacités d'exploration du domaine de recherche. Une fois la nouvelle population constituée, a priori plus performante que la précédente, on examine une condition d'arrêt. Si la condition d'arrêt, définie par l'utilisateur, est satisfaite on met fin au processus de calcul. Dans le cas contraire, on retourne à la deuxième étape, c'est-à-dire la reproduction, et le processus se poursuit jusqu'à la satisfaction de la condition d'arrêt.

4.2.2.2 Représentation des individus et codage

Une méthode évolutionnaire, spécifiquement basée sur un algorithme génétique, nécessite une représentation des individus qui sont les solutions du problème que l'on cherche à résoudre. Le choix de cette représentation est crucial.

Une représentation d'un système mécanique est une description de ce système exprimée en fonctions d'attributs. Dans la plupart des représentations, chaque gène correspond à un attribut qui représente une variable dans l'espace de recherche. Chaque variable, associée à une des caractéristiques de l'espace, peut prendre certaines valeurs (discrètes ou continues). Une représentation simple, souvent utilisée en algorithmes génétiques, adopte le principe des gènes binaires exprimant la présence ou l'absence de la caractéristique associée. Dans de telles représentations, chaque individu est représenté par une chaîne de longueur généralement fixe.

Aujourd'hui, il est unanimement admis qu'il faut considérer deux espaces : l'espace où est posé le problème, sur lequel on peut calculer la fonction objectif, ou espace phénotypique, et l'espace où travaillent les opérateurs génétiques, ou espace génotypique. Le codage est le passage du phénotype au génotype. Un bon codage est celui qui favorise la formation des briques élémentaires qui seront recombinaées par l'opérateur de croisement.

- *Codage en tableau binaire (Bit-array)*

Malgré toutes les critiques qui lui ont été adressées, le codage le plus utilisé reste le codage binaire. L'idée est de partir d'un maillage donné pour définir une représentation binaire appelée tableau de bits ou '*Bit-array*'. Elle est associée au maillage du domaine de conception qui est utilisé pour calculer le comportement mécanique de toutes les

structures. A chaque élément du maillage on attribue une valeur '0' qui représente le vide ou '1' pour la représentation de la matière (Fig. 1-9).

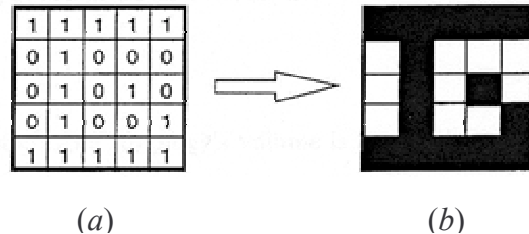


Figure 1-9. Représentation en tableau binaire (a) génotype (b) phénotype. [Jak00]

A priori, il existe une équivalence formelle entre le génotype, représenté par un tableau de bits, et une chaîne de bits. Mais il a été montré par C. Kane [Kan96], en utilisant le théorème des schémas et l'expérimentation, que ces deux types de représentations sont loin d'être équivalentes quand on utilise les opérateurs de croisement standard de la théorie des algorithmes génétiques (Fig. 1-10).

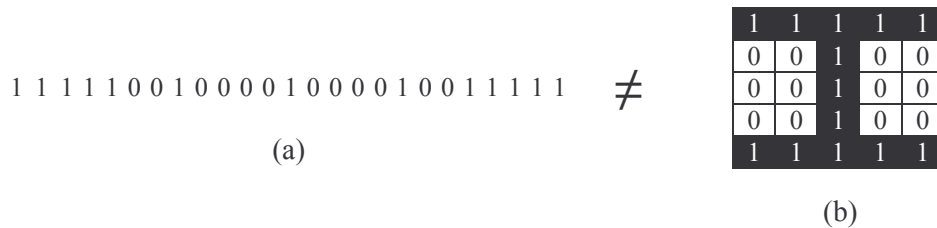


Figure 1-10. (a) représentation en une chaîne de bits, (b) représentation en tableau de bits.

Ce type de représentation a été largement adopté dans les travaux d'optimisation topologique des structures continues. Cependant, il souffre d'une profonde limitation liée à la dépendance du maillage et aux défauts de connectivité des structures générées (Fig. 1-11). La dépendance par rapport au maillage est problématique car des résultats théoriques [Cer96], ainsi que des constatations expérimentales [Gol92], indiquent que la taille critique de la population nécessaire pour atteindre la convergence augmente au moins linéairement avec la taille des individus. Or, la taille d'un individu (le nombre de bits nécessaire pour le décrire) est égale à la taille du maillage. De plus, les populations de grandes tailles nécessitent souvent un grand nombre de générations pour converger. Il est donc clair que cette approche doit restreindre son domaine d'application à des maillages grossiers bidimensionnels, alors que les ingénieurs ont souvent besoin de maillages fins tridimensionnels.

1	1	1	1	1
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
1	1	1	1	1

Figure 1-11. Défaut de connectivité.

Pour réparer les défauts de connectivité, on a recours à des méthodes coûteuses dites d'analyse de connectivité (Fig. 1-12). Dans les travaux de Jakiela et de ses collaborateurs [Jak00, Cha96, Cha94], les éléments ou bloc d'éléments finis qui ne sont pas connectés, d'une manière directe ou indirecte, aux zones dites de support, sont éliminés de la structure. Mais ces composantes contiennent du matériel génétique qui peut être utile lors des opérations génétiques. Les supprimer augmente donc considérablement le risque de converger vers un optimum local [Kan96]. Dans le travail de Fanjoy et Crossley [Fan02], une technique appelée 'chromosome mask' a été utilisée. Elle consiste à supprimer les parties déconnectées de l'espace phénotypique mais à les conserver dans le génotype afin de ne pas détériorer la diversité génétique. Ces parties déconnectées peuvent alors être transmises par les opérations de croisement.

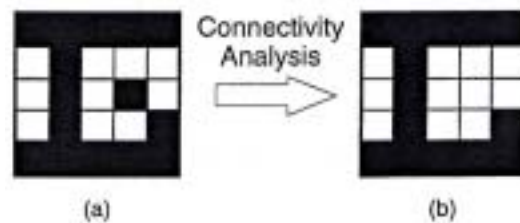


Figure 1-12. Analyse de connectivité, (a) avant réparation, (b) après réparation. [Jak00]

M. Schoenauer [Sch95] a proposé une technique de pénalisation. Le principe est de modifier la fonction performance en introduisant le volume déconnecté sous forme pénalisée. L'objectif est de favoriser la disparition progressive de la matière déconnectée. Pour une minimisation de la masse, par exemple, la fonction mérite prend la forme :

$$Fitness = \frac{1}{V_C + \alpha \cdot V_{NC}} \quad (1-8)$$

V_C est le volume de la partie connectée et V_{NC} celui de la partie non connectée pénalisée par le coefficient α . La difficulté d'utilisation de cette technique est liée à l'ajustement du paramètre de pénalisation. Wang et Tai [Wan04] ont proposé une technique de pénalisation basée sur un principe de traitement d'images.

Fanjoy et Crossley [Fan02] ont mené une étude comparative entre les différentes techniques d'analyse de connectivité proposées dans la littérature. L'étude a montré que leur technique, appelée '*Chromosome Mask*', est la moins coûteuse.

Ces considérations ont conduit à la recherche de représentations plus compactes, dont la complexité ne dépend pas de celle de la discrétisation. Pour obtenir une représentation indépendante de la complexité une ultime étape consiste à la faire évoluer elle-même en l'ajustant par algorithmes génétiques.

- *Représentation de Voronoï*

La représentation de *Voronoï* est un premier pas vers les représentations non structurées pour l'optimisation topologique. Elle a été proposée pour la première fois dans [Sho95] et a été utilisée essentiellement pour des problèmes d'identification. Cette représentation surmonte les limitations de la représentation par tableau binaire '*Bit-array*'.

Dans ce type de représentation, le domaine de travail est divisé en sous-ensembles de formes polyédriques appelées '*cellules de Voronoï*' ou '*sites de Voronoï*'. Le *diagramme de Voronoï* est la partition du domaine défini par cellules de Voronoï. Le génotype est défini en affectant à chaque cellule l'étiquette '0', indiquant l'absence de la matière, ou l'étiquette '1' pour la présence de la matière (Fig. 1-13).

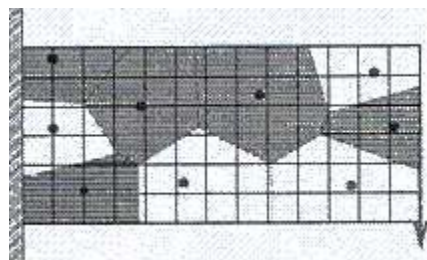


Figure 1-13. Représentation d'une plaque console en diagrammes de Voronoï. [Sch95]

D'un point de vue pratique, la performance de toutes les structures est évaluée en utilisant un maillage de taille fixe. Une partition décrite par un diagramme de Voronoï peut facilement se projeter sur un maillage donné. Un élément appartient à l'une ou l'autre des catégories (matériau ou vide) en fonction du label de la cellule de Voronoï dans laquelle se trouve son centre de gravité, mais il y a indépendance entre le nombre de site et la taille du maillage.

Un point important en optimisation topologique est l'ajustement des frontières de la structure solution. Une forme optimale ne peut être atteinte en un temps de calcul raisonnable que si l'algorithme est capable de contrôler précisément les frontières. Malheureusement, la représentation Voronoï ne permet qu'un contrôle indirect de la frontière. De plus, par sa structure même, cette représentation rend difficile la modification d'une portion de la frontière sans toucher aux portions adjacentes. La représentation par dipôles est une tentative pour palier cette difficulté [Ham00]. Un troisième type de représentation est celui en barres de Voronoï qui a été proposé afin de résoudre les problèmes dont les solutions ont tendance à évoluer vers des formes en treillis.

Les représentations de Voronoï ont été testées par H. Hamda and al. [Ham00] sur des problèmes-test simples d'optimisation topologique, afin de vérifier l'indépendance par rapport à la complexité du maillage. Les résultats semblent montrer que les trois représentations sont adaptées à la résolution de ce type de problèmes. Elles demandent grossièrement le même temps de calcul pour atteindre des solutions comparables, avec un léger avantage pour la représentation par barres de Voronoï. Pourtant, après examen de la complexité du codage des solutions obtenues, on trouve un avantage net pour la représentation par barres de Voronoï, dont les solutions utilisent moins de gènes que les deux autres.

- *Représentation graphique*

L'utilisation de représentation graphique pour l'optimisation topologique a été récemment proposée par Tai [Tai98]. Le principe consiste à relier les zones d'application des conditions aux limites (chargement et déplacements) par des courbes de Bézier. Ces zones sont qualifiées de zones de support. Dans une première étape, les zones de supports sont définies sur l'espace de conception (Fig. 1-14a). Dans une deuxième étape, le domaine de conception est discrétisé, chacun des éléments en noir sur la figure 1-14b représentant une zone de support. Chaque couple de zones de support est ensuite relié par une courbe de Bézier. Dans une troisième étape, entité géométrique appelée squelette est définie en suivant l'abscisse curviligne des courbes de Bézier (Fig. 1-14c). Enfin, dans une quatrième et dernière étape, une structure, dont tous les éléments sont complètement connectés, est construite en complétant le squelette par la chair (Fig. 1-14d). Chaque courbe, reliant deux zones de support, est divisée en trois segments de longueur paramétrique fixe égale à $1/3$. Les éléments de la chair sont disposés par couches autour des éléments constituant chaque segment du squelette. La quantité de chair est contrôlée par le nombre de couches ajoutées.

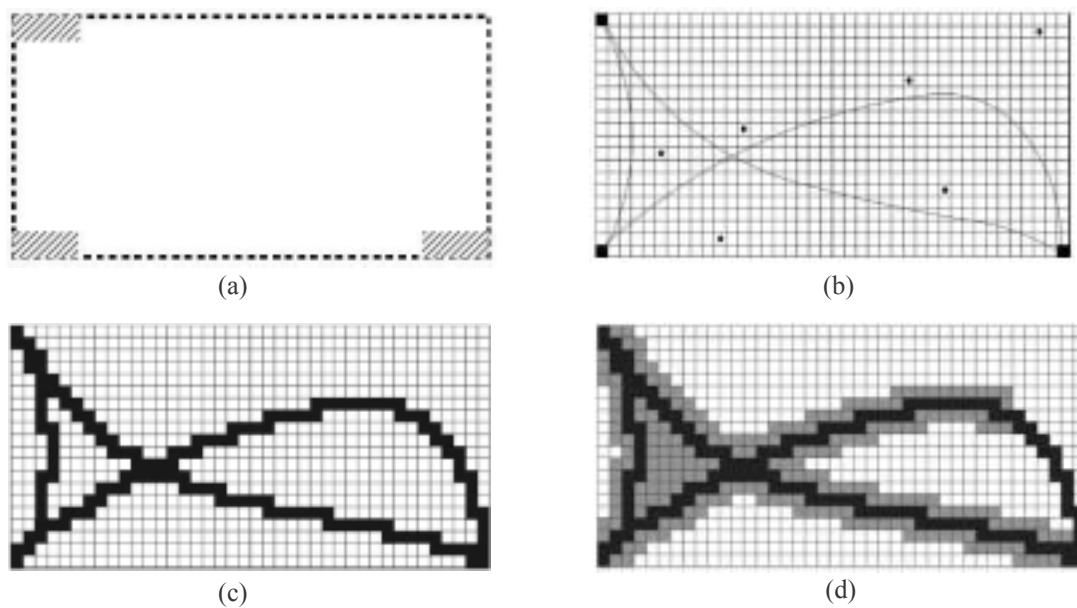


Figure 1-14. Principe de la représentation géométrique, (a) définition des zones de support, (b) connexion des zones de support par des courbes, (c) construction du squelette, (d) ajout des éléments de la chair. [Tai98]

Le premier avantage de cette nouvelle représentation est qu'elle assure la connectivité des structures solutions générées. D'autre part, le codage basé sur cette représentation surmonte la limitation du codage 'Bit-array' concernant la dépendance par rapport au maillage. Cependant, cette représentation souffre de quelques limitations. Les zones de support fixes pénalisent l'exploration du domaine de conception et peuvent conduire à une convergence prématurée. La disposition d'éléments de la chair, enrobant complètement les éléments d'un segment de squelette, sur une longueur fixe, pénalise la répartition de la matière (c'est-à-dire la répartition des éléments).

Dans notre travail, présenté dans le chapitre qui suit, nous avons adopté le même principe. Nous avons cependant apporté des modifications afin de dépasser les limitations exposées ci-dessus et afin de pouvoir résoudre une plus grande variété de problèmes d'optimisation topologique que les seuls mécanismes auxquels l'approche de Tai était à l'origine destinée.

4.2.2.3 Initialisation

L'initialisation sert à constituer la population initiale. C'est une étape à ne pas négliger car si la population n'est pas uniformément répartie au départ, l'évolution risque de se concentrer sur un optimum local duquel il peut être difficile de sortir. On essaie donc de créer une population initiale aussi diversifiée que possible. Classiquement, pour chaque

individu de la population, le mécanisme d'initialisation consiste à effectuer un tirage aléatoire de chacun des bits dans une chaîne. Dans le cas d'une représentation binaire par exemple, le tirage s'effectue dans l'ensemble $\{0, 1\}$.

4.2.2.4 Evaluation

A chaque solution, on associe une fonction performance '*fitness*' reliée à la valeur de la fonction objectif. Cette fonction performance décrit le mérite de l'individu qui est représenté par un chromosome. L'évaluation des individus en optimisation topologique des structures se fait par une méthode d'analyse numérique des structures, généralement la méthode des éléments finis.

La fonction performance est très importante pour un algorithme génétique au même titre que le codage. En effet, pour que les algorithmes génétiques se comportent bien, nous devons trouver une manière de formuler des fonctions performance ne comportant pas trop de maxima locaux et ne présentant pas de maximum local isolé.

La construction de la fonction performance est évidente pour certains problèmes. Pour les problèmes de maximisation par exemple, la fonction mérite peut être égale à la fonction objectif. Par contre, pour les problèmes de minimisation, l'objectif est de trouver des solutions pour lesquelles la fonction objectif atteint des valeurs minimales. Dans ce cas, la fonction performance choisie est la réciproque de la fonction objectif. Dans tous les cas, l'algorithme génétique cherche à maximiser la fonction performance qui, dans le cadre d'un problème de minimisation, prend la forme suivante :

$$Fitness(x_i) = \frac{1}{f(x_i)} \quad (1-9)$$

Où $f(x_i)$ représente la fonction objectif évaluée pour l'individu x_i .

Un choix classique de fonction objectif est la compliance. Ce choix se justifie pleinement dans le cadre d'une approche déterministe où il est nécessaire de dériver pour pouvoir procéder à une analyse de sensibilité. Dans un algorithme génétique, où l'approche stochastique ne nécessite pas de dérivation, le choix de la compliance comme fonction objectif n'est pas aussi vital. Jakiela et ses collaborateurs [Jak00, Cha94] ont posé le problème d'optimisation sous forme de maximisation de la raideur de la structure en supposant que la raideur est inversement proportionnelle au déplacement maximal ' $\delta_{\max}$ ' de la structure :

$$Fitness = \frac{1}{|\delta_{Max}|} \quad (1-10)$$

La raideur n'étant pas une grandeur différentiable, les méthodes déterministes ne sont opérationnelles pour maximiser mérite définie par (1-10). En revanche, ce n'est pas le cas pour les méthodes stochastiques, telles que les algorithmes génétiques, qui sont exemptées de l'analyse de sensibilité.

- *Problème sous limitations*

Les algorithmes génétiques, comme toutes autres méthodes classiques d'optimisation, trouvent des difficultés pour traiter les limitations. Ces limitations représentent, dans le cas d'un problème mécanique par exemple, les conditions technologiques ou fonctionnelles qui doivent être satisfaites. Un problème sous limitation est souvent formulé comme suit :

$$\begin{aligned} \text{Maximiser : } Fitness(x) &= \frac{1}{f(x)} \\ \text{avec : } g_j(x) &\geq 0 \quad j = 1, \dots, J \\ h_k(x) &= 0 \quad k = 1, \dots, K \end{aligned} \quad (1-11)$$

Où g représente des limitations de type inégalité et h des limitations de type égalité.

Plusieurs méthodes ont été proposées dans la littérature, les plus utilisées étant basées sur le principe de pénalité [Coe02]. Avec ce principe, la formulation d'un problème sous limitations prend s'exprime généralement de la manière suivante :

$$Fitness(x) = \begin{cases} \frac{1}{f(x)} & \text{si } x \in \hat{E} \\ \frac{1}{f(x) + violation(x)} & \text{si non} \end{cases} \quad (1-12)$$

$\hat{E}$ est l'espace de solutions admissibles et $violation(x)$ la fonction de pénalisation exprimée en fonction des limitations :

$$Violation(x) = \sum_{j=1}^J (\alpha_g)_j \cdot (G_j(x))^\beta + \sum_{k=1}^K (\alpha_h)_k \cdot (H_k(x))^\beta \quad (1-13)$$

G et H sont les degrés de violation correspondant aux limitations du problème (1-11) et α et β sont des coefficients de pénalisation dont les valeurs doivent être judicieusement choisies.

Les coefficients de pénalisation peuvent prendre des valeurs constantes au cours des générations et, dans ce cas, on parle de méthodes de pénalité statique. L'étude de Kane [Kan96], ainsi que d'autres études cités dans [Coe02], ont montré que la recherche des valeurs adéquates est entièrement empirique et dépend du problème posé.

L'ajustement de ces paramètres de pénalisation peut être dynamique [Mic95]. Jakiela [Jak00] a utilisé une stratégie où la pénalisation est une fonction linéaire du nombre de générations. Cependant, la stratégie dynamique requiert une bonne intuition pour être efficace.

Les paramètres de pénalisation dépendent généralement du problème traité et doivent être donc ajustés en fonction des paramètres du problème lui-même. C'est la stratégie adaptative dont le principe est de mettre à jour les coefficients de pénalisation conformément aux informations sur la population en cours. Hamda et *al.* [Ham00] ont proposé une technique adaptative basée sur des statistiques concernant le nombre d'individus admissibles et le nombre de ceux qui ne le sont pas.

Dans le cadre général d'un algorithme génétique classique, Deb [Deb00] a proposé une technique de pénalisation simple. La technique est inspirée du principe de l'algorithme génétique qui consiste à distinguer les individus, les uns par rapport aux autres, selon leurs performances. Cette technique a été largement utilisée dans notre travail et sera présentée dans le deuxième chapitre de ce mémoire.

4.2.2.5 Reproduction

La reproduction joue un rôle fondamental pour le passage d'une génération à une autre. Elle représente un cycle de l'algorithme génétique. Elle se fait en appliquant des opérations génétiques qui simulent les processus naturels appliqués par la nature aux espèces biologiques vivantes. Ces opérations génétiques sont la sélection, le croisement, la mutation et le partage. Elles sont présentées en détail dans les quatre paragraphes qui suivent.

4.2.2.6 Sélection

Les parents sont choisis aléatoirement suivant une méthode qui favorise les meilleurs d'entre eux. C'est la phase dite de sélection basée sur le principe Darwinien de l'évolution.

Certains individus peuvent être choisis plusieurs fois tandis que d'autres ne le seront jamais.

La sélection des individus s'effectue sur la base de leur fonction d'évaluation. Plusieurs opérateurs de sélection existent parmi lesquels la sélection par probabilité, par rang ou par tournoi ainsi que d'autres techniques faisant intervenir des notions de voisinage entre individus [Gol89].

- Sélection à la roulette : c'est la méthode la plus connue des sélections stochastiques. Elle consiste à copier les individus proportionnellement à leur performance. Ainsi un individu x_i a la probabilité suivante d'être sélectionné :

$$P_{\text{Sél}}(x_i) = \frac{\text{Fitness}(x_i)}{\sum_{j=1}^{\text{PopSz}} \text{Fitness}(x_j)} \quad (1-14)$$

Où PopSz représente le nombre d'individus de la population.

Pratiquement, on procède à un tirage sur une roulette représentée par la figure 1-15. La roulette est un disque constitué de secteurs angulaires. Chaque angle est proportionnel à la valeur de la fonction performance de l'individu associé à ce secteur. On détermine l'individu sélectionné en tirant un réel de façon aléatoire sur l'intervalle $[0 ; 2\pi]$. L'individu sélectionné est celui qui correspond au secteur de la roulette auquel le réel tiré aléatoirement appartient. Le tirage par roulette présente toutefois de nombreux inconvénients. Le plus important est lié à l'échelle de la fonction performance. En effet, dans le cas d'un écart considérable entre les performances des individus, les plus performants sont les plus favorisés par la sélection. Cela conduit à la stagnation de l'algorithme puis à une convergence prématurée. Afin de remédier à ce problème, on utilise souvent une technique de mise à l'échelle des fonctions performances [Gol89]. Mais cette technique de mise à l'échelle n'est pas toujours efficace car elle dépend fortement des performances des individus. C'est pourquoi ce mode de sélection est presque totalement abandonné aujourd'hui.

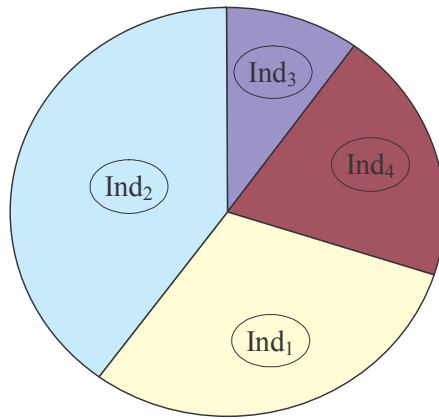


Figure 1-15. Principe de sélection à la roulette.

- Sélection par rang : cette méthode de sélection est divisée en deux étapes. Tout d'abord, il faut ranger les individus par ordre croissant (ou décroissant) de performance. Ensuite, une procédure de sélection similaire à la sélection par roulette est utilisée. Cette procédure permet d'attribuer une probabilité de sélection en fonction du rang. Cette probabilité est proportionnelle à une performance artificielle calculée en fonction du rang. A titre d'exemple, on donne la variante linéaire de cette performance artificielle :

$$Fit_r(i) = PS - (ri - 1) \times \frac{2 \times PS - 2}{PopSz - 1} \quad (1-15)$$

Où ri est le rang de l'individu i , $PopSz$ le nombre d'individus et PS est la 'pression de sélection'. Dans la littérature, on trouve d'autre variante de classement des individus (exponentielle par exemple [Law91]). Le choix d'une variante ou d'une autre dépend généralement du problème traité.

L'avantage de cette méthode de sélection est qu'elle n'est pas soumise à la limitation de la méthode à la roulette concernant la mise à l'échelle. En effet, les probabilités de sélection ne dépendent que de la variante utilisée pour le calcul de la performance artificielle.

- Sélection par Tournoi : Cette méthode, qui utilise des comparaisons entre individus, au même titre que la sélection par rang, ne nécessite en revanche aucun tri de la population. Le principe consiste à choisir uniformément T individus et à sélectionner le meilleur de ces T individus. Le choix de T , qui représente la taille du tournoi, permet de faire varier la pression sélective. De cette manière, on contrôle les chances de sélection des individus les plus performants par rapport aux plus

faibles. Il est à noter que le cas $T = 2$ correspond, en espérance et en premier ordre, à la sélection par rang linéaire décrite par la formule (1-15).

Des études comparatives entre les différentes méthodes de sélection ont montré que la sélection par rang et la sélection par tournoi sont les meilleures [Gol91]. Ces méthodes ont été comparées sur des critères concernant la qualité des résultats obtenus et la vitesse de convergence.

4.2.2.7 Croisement

Le croisement est l'échange d'un certain nombre de bits entre deux chromosomes représentant deux individus de la population. En choisissant aléatoirement deux individus parents, l'opération de croisement donne naissance à deux nouveaux individus. Il existe différents types de croisement pour un algorithme génétique classique. Sur la figure 1-16 nous présentons les trois principaux types de croisement (à un point, à deux points et uniforme). Cette opération est contrôlée par une probabilité qui varie généralement dans l'intervalle [0.6, 0.8].

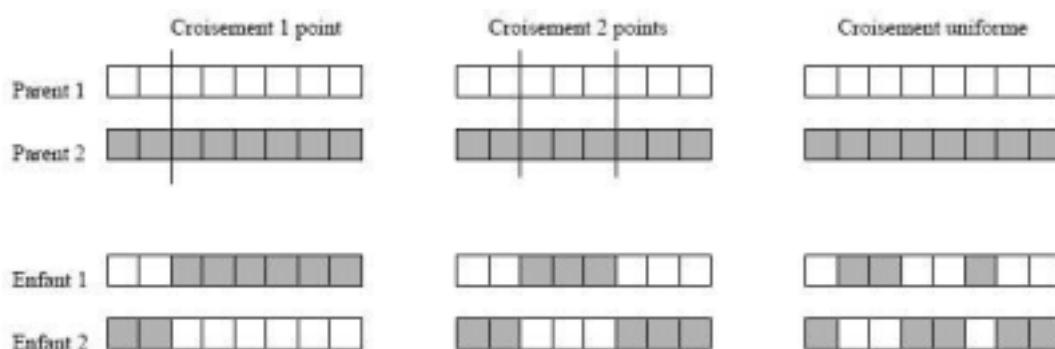


Figure 1-16. Exemples d'opérations de croisement.

Les opérateurs de croisement en optimisation topologique sont similaires aux opérateurs classiques. Cependant, ils sont appliqués souvent sur l'espace phénotypique [Kan96, Fan02]. Cela est dû à l'incompatibilité entre la représentation en chaîne binaire et le phénotype. Nous présentons dans ce paragraphe les opérateurs de croisement les plus utilisés et les plus adaptés aux problèmes de l'optimisation topologique de structures.

- *Croisement à deux points* : c'est un croisement unidimensionnel dont le principe consiste à échanger les informations des bits comprises entre deux points de

coupure. Les deux points sont choisis aléatoirement sur la chaîne représentant un chromosome (Fig. 1-17b).

- *Croisement uniforme* : les points de croisement peuvent être multiples et choisis aléatoirement. Dans ce cas, on parle de croisement uniforme. Ce type de croisement peut être appliqué sur une chaîne unidimensionnelle ou bidimensionnelle.
- *Croisement diagonal* : l'idée de base de ce croisement est de généraliser en deux dimensions le très populaire croisement à un point (Fig. 1-17a). On choisit aléatoirement deux points distincts sur le chromosome. Ces deux points vont générer une droite ou une diagonale (Fig. 1-17d). Les enfants sont le résultat d'échange des parties délimitées par la diagonale.
- *Croisement par blocs* : chaque parent est divisé en blocs en utilisant des droites horizontales et verticales, définies aléatoirement (Fig. 1-17c). L'échange s'effectue sur les blocs définis par ces droites. Cette technique a été utilisée par Jensen [Jen93] et par Kane [Kan96].

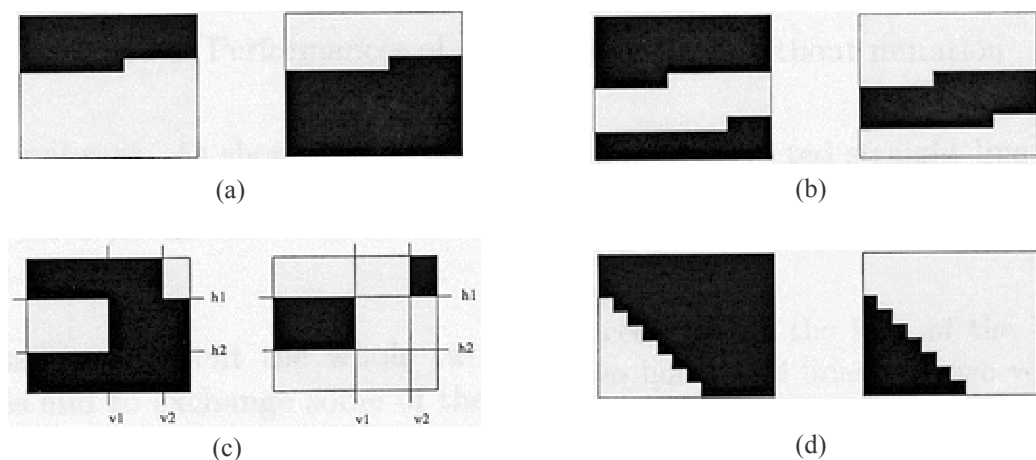


Figure 1-17. Opérateurs de croisement : (a) à un point, (b) à deux points, (c) par blocs, (d) diagonal. [Kan96]

Des études comparatives entre les différents types de croisement ont été faites. Les études de Kane [Kan96] ont montré que les opérateurs bidimensionnels sont plus performants que les opérateurs unidimensionnels excepté le croisement uniforme. Cela confirme que la chaîne de bits unidimensionnelle n'est pas équivalente au tableau de bits comme nous l'avons déjà signalé dans le paragraphe 4.2.2.2.

Dans les travaux de Schoenauer et de ses collaborateurs [Sch95, Ham00], ainsi que dans le travail de Fanjoy et Crossley [Fan02], le croisement a été appliqué sur l'espace

phénotypique (Fig. 1-18). Les techniques de coupure ont été purement géométriques en simulant les techniques de croisement bidimensionnelles utilisées dans le travail de Kane.

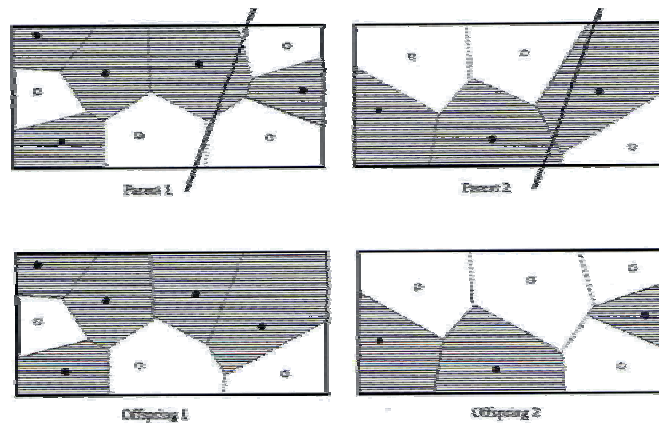


Figure 1-18. Croisement basé sur un principe géométrique. [Had00]

4.2.2.8 Mutation

De façon aléatoire, un gène peut, au sein d'un chromosome, être substitué à un autre. La mutation la plus simple sur un chromosome change un bit de façon aléatoire (Fig. 1-19). De la même manière que pour le croisement, on définit une probabilité qui contrôle l'opération de mutation. Ce paramètre peut être associé à chaque gène du chromosome, c'est-à-dire que chaque bit a une probabilité de muter. En affectant à chaque individu une probabilité, la valeur de cette dernière représente le taux maximal possible d'individus mutant. Ce taux est généralement faible dans un algorithme génétique à codage binaire classique et peut être très élevé avec un algorithme génétique à codage réel [Tai98]. Il existe plusieurs variantes de mutation qui ont été adaptées aux problèmes d'optimisation topologique [Kan96]. La mutation peut être aussi spécifique selon la représentation utilisée [Sch95].

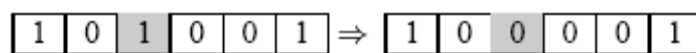


Figure 1-19. Un exemple d'une opération de mutation.

- *Mutation uniforme* : c'est la mutation souvent utilisée dans un algorithme génétique classique. Le principe est d'altérer le contenu d'un certain nombre de gènes choisis aléatoirement.

- *Mutation générale* : ce type de mutation prend en compte les caractéristiques de la population dans son ensemble, généralement la diversité génétique. On affecte à chaque bit une probabilité de mutation inversement proportionnelle à sa diversité. Ce type de mutation est très efficace pour empêcher la convergence prématurée mais il ralentit aussi la convergence [Kan96].
- *Mutation des composants* : il s'agit d'un opérateur local conçu spécifiquement pour les problèmes d'optimisation de forme. Le principe consiste à faire des petites modifications localisées sur les contours du domaine de conception. On utilise ce type de mutation à la fin de l'évolution afin de raffiner la forme.

Le choix des probabilités de croisement et de mutation est crucial. Ce choix doit être judicieux afin de maintenir un équilibre entre l'exploration et l'exploitation dans l'algorithme génétique. Cependant, il est difficile de maintenir cet équilibre en choisissant des probabilités fixes au cours des générations. C'est pourquoi la gestion des opérateurs de croisement et de mutation peut être dynamique ou adaptative [Spe92, Sri94].

4.2.2.9 Partage (Shearing)

Le partage est un paramètre évolué des algorithmes génétiques [Gol87]. Il est utilisé pour éviter le regroupement d'individus performants, et assurer une certaine diversité génétique dans la population. Le principe consiste à pénaliser les individus qui ont beaucoup de voisins proches en divisant leurs performances sur une fonction dite de partage. Cette dernière, dont la valeur est comprise entre 0 et 1, est calculée en fonction d'un paramètre qui mesure le degré de similarité entre les individus. La performance modifiée pour un individu x_i s'écrit de la manière suivante [Deb89] :

$$Fitness_{Sh}(x_i) = \frac{Fitness(x_i)}{\sum_{j=1}^n Sh(d(x_i, x_j))} \quad (1-16)$$

Sh (Shearing) est la fonction de partage et d est la distance entre les individus qui exprime le degré de similarité entre ces individus.

La fonction de partage dépend du codage adopté. Duda et Jakiela [Dud97] ont utilisé une fonction appelée coefficient de Jaccard et l'ont appliqué sur le génotype :

$$C_j = \frac{na}{na + nb} \quad (1-17)$$

où na est le nombre d'éléments finis contenant de la matière simultanément dans les deux structures testées. nb représente le nombre d'éléments finis qui ne contiennent pas de la matière simultanément dans les deux structures testées. Le degré de similarité, mesuré par un coefficient compris entre 0 et 1, est fixé par l'utilisateur. Dans le travail de Duda et Jakiela [Dud97], on considère que deux individus sont similaires si le coefficient de Jaccard est supérieur à 0.85.

La technique de partage est souvent accompagnée par une technique de regroupement appelé '*Clustering*' [Rom84]. L'information fournie par la fonction de partage peut être utilisée pour éviter le croisement, inutile, entre les individus similaires. Considérons la figure 1-20a, qui schématise le domaine de conception où les solutions sont représentées par les points ronds numérotés de 1 à 6. La première étape de regroupement consiste à mesurer le degré de similarité entre un couple de solutions en utilisant le coefficient de Jaccard de l'équation 1-17. La figure 1-20b montre le regroupement des solutions 1 et 5, qui peuvent être considérées donc comme une seule entité. On poursuit ensuite le même processus : regroupement des solutions 2 et 3 (Fig. 1-20c) et regroupement des solutions 1-5 et 4 (Fig. 1-20d). Le tirage de deux individus pour le croisement s'effectue donc à partir de deux groupes différents.

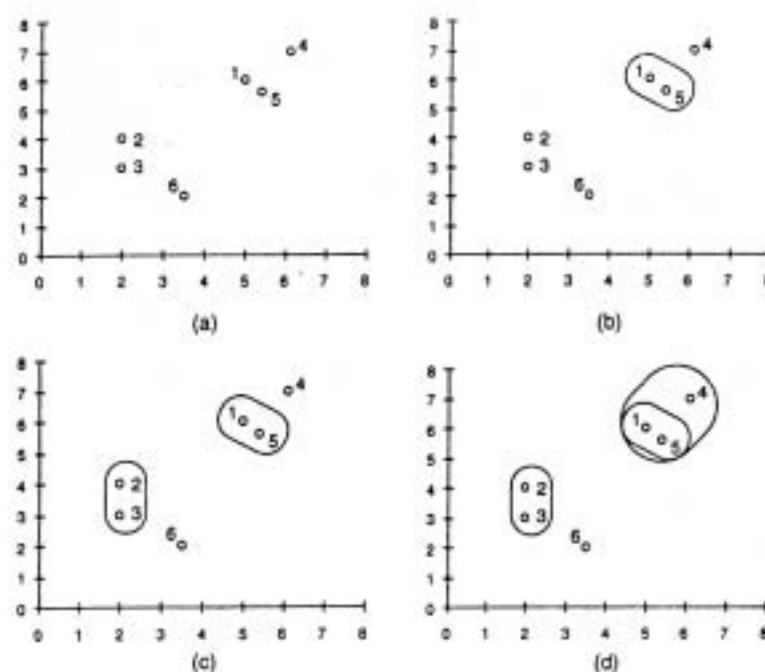


Figure 1-20. Processus de regroupement. [Dud97]

Sur la figure 1-21, nous présentons un exemple de regroupement des individus de la population selon leurs degrés de similarité.

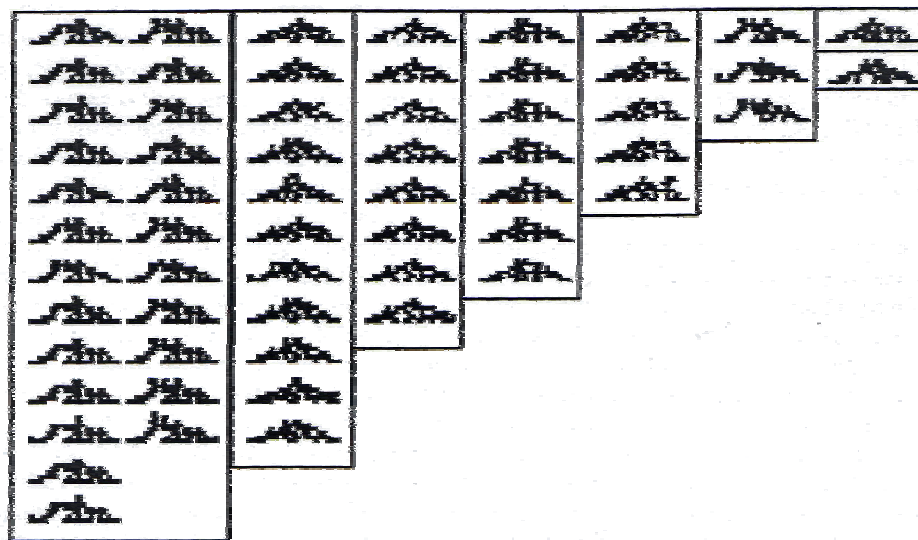


Figure 1-21. Les solutions finales regroupées selon leur degré de similarité. [Jak00]

L'exemple de regroupement de la figure 1-21, concerne l'optimisation topologique d'une structure continue réalisé par Jakiela and al. [Jak00].

4.2.2.10 Avantages et désavantages des algorithmes génétiques

Les algorithmes génétiques, méthodes d'optimisation d'ordre zéro, présentent l'avantage de ne pas avoir besoin de calcul de sensibilités. Ils ne nécessitent que la connaissance des valeurs prises par ma fonction à optimiser. Ils sont donc efficaces lorsque cette fonction possède plusieurs optima locaux, cas où les méthodes d'optimisation déterministes classiques sont défaillantes. Cette situation se présente pour les problèmes complexes, problèmes impliquant par exemple des non linéarités ou problèmes avec interaction entre variables de conception. De plus, les algorithmes génétiques sont capables de travailler sur des espaces quelconques : espaces continu, discret ou produit cartésien d'espaces continu et discret par exemple.

Pour bien expliquer l'avantage des algorithmes génétiques par rapport aux méthodes déterministes, prenons par exemple le problème de l'optimisation de structures de treillis de barres. Ce type de problème est aujourd'hui bien connu. Dans sa version la plus simple, les variables de conception sont les sections des différentes barres. Les méthodes classiques de type gradient permettent alors de trouver la solution dans le cas où les sections prennent des valeurs dans un intervalle réel donné. Ces méthodes peuvent également prendre en compte différents types de limitations imposées à la solution tels que des déplacements maximaux ou des valeurs de contraintes mécaniques en certains nœuds.

Mais le problème concret qui est posé à l'ingénieur de bureau d'étude est généralement limité par le catalogue des différentes barres disponibles. Le nombre de sections possibles est alors fini. De plus, l'un des degrés de liberté du problème peut être le matériau constituant les barres, paramètre discret par nature. L'ensemble des variables de conception peut être ainsi constitué de variables continues et de variables discrètes, certaines même purement qualitatives. Or, les méthodes classiques d'optimisation numérique ne peuvent plus s'appliquer dès que l'une au moins des variables de conception prend des valeurs discrètes.

Enfin, du point de vue de la mise en œuvre informatique, les algorithmes génétiques sont très faciles à implémenter dans un code de calcul numérique.

Cependant, le talon d'Achille des algorithmes génétiques est bien connu. Il s'agit du coût en temps de calcul, qui est élevé et parfois même prohibitif. En effet, pour un problème d'optimisation structurale par exemple, l'évaluation nécessite l'analyse, souvent par éléments finis, d'une population de structures solutions. Autrement dit, il est nécessaire de procéder à autant d'analyses éléments finis que d'individus dans la population. Heureusement, les progrès réalisés en calcul parallèle, ainsi qu'en algorithmique (en restreignant par exemple l'analyse éléments finis à une classe d'individus de la population – cf par exemple paragraphe 7, chapitre II à ce sujet), permettent de réduire significativement le coût de l'évaluation. Si bien qu'aujourd'hui, le coût informatique des algorithmes génétiques incombe essentiellement à la lenteur de la convergence dans la mesure où il est nécessaire de générer un grand nombre de populations avant d'obtenir la solution cherchée. Enfin, les algorithmes génétiques utilisent un grand nombre de paramètres qui doivent être fixés par l'utilisateur (taille de la population, probabilités d'application des opérateurs génétiques,...) et dont la mise au point de ces paramètres ne peut se faire qu'empiriquement. Pour effectuer cette mise au point, il est parfois nécessaire de procéder à de nombreux tests numériques, ce qui augmente d'autant le coût du temps de calcul.

5. Conclusions

Le premier chapitre de ce mémoire a été consacré à un état de l'art détaillé de l'optimisation topologique, considéré aujourd'hui comme la plus puissante des méthodes d'optimisation. L'étude bibliographique qui a été réalisée montre que ce domaine de recherche fait l'objet de nombreux travaux.

Les méthodes déterministes, malgré un certain nombre de limitations, sont les plus utilisées. Ces méthodes ont connues un grand succès depuis le travail de Bendsøe et Kikuchi [Ben88] qui ont suggéré d'introduire la notion de microstructure poreuse dans le problème de distribution optimale de la matière. La recherche bibliographique menée dans ce chapitre montre par ailleurs que les contributions en optimisation topologique de structures continues par algorithmes génétiques restent très modestes. La majorité de ces travaux a en effet été consacrée aux problèmes discrets. Cela s'explique par le fait que ces problèmes sont les mieux formulés pour être résolus efficacement par un algorithme génétique.

Pour évaluer les performances des solutions avec un algorithme génétique, dans le cadre de l'optimisation topologique, la méthode des éléments finis est largement utilisée, aussi bien pour les structures continues que pour les structures discrètes. En effet, une représentation binaire peut être utilisée, chaque élément étant codé par un 0 ou par un 1 suivant qu'il contient de la matière ou non. On se ramène ainsi dans tous les cas à une problématique discrète qui convient parfaitement aux algorithmes génétiques. Cependant, en se basant sur cette représentation, il s'avère à posteriori que le codage adopté souffre de sérieuses limitations (défauts de connectivité des structures, dépendance par rapport au maillage). De plus, la réparation des défauts de connectivité, par des méthodes d'analyse géométrique, est coûteuse. Ce coût s'ajoute au coût de l'analyse structurale par éléments finis pour l'évaluation des solutions. Pour peu que la convergence soit lente, ce qui est souvent le cas et ce qui nécessite de nombreuses évaluations, le cumul de toutes les opérations numériques peut s'avérer au final rédhibitoire pour une utilisation efficace des algorithmes génétiques.

Dans le but de palier aux limitations de la représentation par tableau binaire, d'autres représentations ont été proposées dans la littérature. Hamda et al. [Ham00] ont proposé une représentation, basée sur les sites de Voronoï, qui dépasse la limitation concernant la dépendance par rapport au maillage mais qui n'assure pas la connectivité. Tai [Tai98] a proposé une représentation graphique qui assure la connectivité des structures générées, ce qui permet un gain notable du coût de calcul. En revanche, cette représentation n'offre pas

toute l'efficacité souhaitée en raison du caractère trop figé de certains paramètres dans la gestion de l'ajout de matière, ce qui pénalise l'exploration du domaine de recherche.

Malgré leur efficacité démontrée pour la résolution des problèmes d'optimisation dans différents domaines, les algorithmes génétiques n'ont pas encore totalement prouvé aujourd'hui leur capacité à traiter les problèmes d'optimisation topologique des structures continues. Cela est essentiellement dû à la difficulté d'adopter une représentation conforme au problème traité et adaptée aux opérations génétiques. Même si tous les paramètres génétiques (sélection, croisement, mutation) sont importants, l'efficacité de l'algorithme génétique sera principalement conditionnée par le codage utilisé.

Vu l'importance du codage dans un algorithme génétique, nous avons consacré la plus grande partie de ce travail de thèse à la recherche d'un bon codage. La représentation proposée par Tai [Tai98], malgré ses limitations, se distingue des autres représentations par le fait qu'elle assure la connectivité des structures générées. C'est cette particularité qui nous a poussé à l'adopter. Les adaptations et améliorations que nous lui avons apportées sont exposées dans le chapitre suivant.

Chapitre II

Optimisation topologique par algorithme génétique Représentation graphique par courbes de Bézier

1. Introduction	47
2. Formulation du problème d'optimisation topologique	49
3. Problème mécanique	50
4. Algorithme génétique	52
5. Représentation géométrique	53
5.1 Principe	54
5.2 Notre contribution	55
5.3 Codage	60
5.4 Décodage	62
5.5 Fonction performance	64
5.6 Analyse structurale	66
5.7 Traitement des limitations	66
5.8 Opérateurs génétiques	68
6. Croisement et Mutation adaptatifs	76
7. Implémentation	80
8. Résultats et discussions	82
8.1 Treillis de Michell	82
8.2 Poutre console (cantilever)	89
8.3 Poutre MBB	95
9. Conclusions	104

1. Introduction

Dans ce chapitre, nous développons une approche par algorithme génétique pour l'optimisation topologique de structures continues. L'algorithme génétique est basé sur un codage géométrique qui consiste à utiliser les courbes de Bézier comme un élément moteur pour représenter la morphologie des structures solutions.

A l'origine, l'idée a été proposée par Tai [Tai98] (Chapitre I : § 4.2.2.2). L'auteur a proposé cette nouvelle technique pour remédier à un problème majeur en optimisation topologique par algorithmes génétiques. Il s'agit de la connectivité des structures solutions lorsque la représentation en tableau binaire est utilisée. En effet, avec la nouvelle représentation, la connectivité est assurée grâce à un schéma graphique basé sur l'utilisation de courbes de Bézier (Fig. 2-8). Cependant, la technique de Tai comporte plusieurs limitations qui empêchent son utilisation dans un contexte d'optimisation topologique générale. La première limitation concerne le choix des zones de support qui doivent, d'une part, se cantonner au niveau des surfaces d'application des conditions aux limites et, d'autre part, rester fixes dans toutes les structures générées. Notre contribution sur ce point consiste à élargir la définition d'une zone de support, qui dans notre approche ne contient pas de la matière. Les éléments finis, résultant de l'intersection des surfaces d'application de conditions aux limites et des surfaces de haute concentration de contraintes, forment une zone de support (Fig. 2-7). Une partie du squelette peut aussi servir comme zone de support (Fig. 2-8a). La deuxième limitation de Tai concerne la longueur fixe, égale à $1/3$, des segments du squelette sur lesquels se disposent les éléments de la chair (Fig. 2-5). Notre apport sur ce point est l'introduction de la notion de segment de longueur variable. Enfin, nous avons introduit deux nouveaux types de chair où les éléments de cette dernière peuvent être disposés à droite ou à gauche du squelette (Fig. 2-8b). Sur ce point, notre approche se démarque de l'approche de Tai où les éléments de la chair sont disposés simultanément des deux côtés (Fig. 2-6). Ces différents apports ont pour objectif de conférer plus de souplesse au schéma graphique afin d'améliorer l'exploration du domaine de conception.

Le principe de cette nouvelle représentation sera présenté en premier lieu, suivi par l'exploitation de cette technique dans le cadre d'un algorithme génétique. Cette exploitation se traduit par le passage au codage, c'est-à-dire la représentation des structures solutions par des chromosomes. Chaque chromosome, solution possible du problème, est évalué par une fonction performance. Cette fonction est reliée à la fonction objectif, qui permet de mesurer la qualité de la solution, et aux limitations imposées (Chapitre I : § 4.2.2.4).

Le codage est le paramètre le plus important en algorithmes génétiques, car il représente la position du problème, mais il y a aussi d'autres paramètres importants. Ce sont les opérateurs génétiques de sélection, de croisement et de mutation qui assurent les fonctions d'exploitation et d'exploration dans un algorithme génétique. La convergence vers un optimum global nécessite un choix judicieux de ces opérateurs, ainsi qu'une bonne stratégie de gestion d'une génération à l'autre. Dans le but de conserver la diversité génétique de la population, notre choix de l'opérateur de sélection a porté sur la méthode dite '*Stochastic Universal Sampling*' [Bak94]. Pour la gestion des opérations de croisement et de mutation, nous proposons une stratégie adaptative, partiellement inspirée des travaux de Jakobovic [Jak97].

Sur la base des développements théoriques que nous allons présenter dans ce chapitre, nous avons développé un code de calcul avec le langage de programmation Matlab. En assurant la connectivité, nous pouvons rechercher la solution optimale en ajoutant progressivement de la matière à l'inverse des toutes les autres méthodes d'optimisation qui procèdent à l'élimination de la matière. Cela peut induire comme avantage un gain notable du coût de l'analyse par éléments finis. Nous proposons aussi une technique de calcul en deux étapes. Dans la première, nous recherchons le squelette optimal. Dès la convergence vers ce dernier, la deuxième étape, dont le but est la recherche de la distribution des éléments de la chair, commence. Cette technique permet, d'une part, de réduire le temps du calcul en diminuant le nombre de variables dans chaque étape, et de minimiser le coût de l'analyse éléments finis d'autre part. Dans le paragraphe 8 nous présentons les résultats de plusieurs applications numériques réalisés à l'aide de ce code. Ces résultats seront comparés avec ceux de la méthode d'homogénéisation [Ben03, Sig01] qui est bien connue dans la littérature. Le chapitre se termine par des conclusions et des perspectives.

2. Formulation du problème d'optimisation topologique

Considérons un domaine Ω (sur $\mathbb{R}^2$ ou $\mathbb{R}^3$) dont les conditions aux limites sont spécifiées (Fig. 2-1). Le but de l'optimisation topologique est de déterminer le sous domaine Ω_m qui doit être occupé par la matière en satisfaisant certains critères prédéfinis. Cela équivaut à la recherche des zones vides Ω_v .

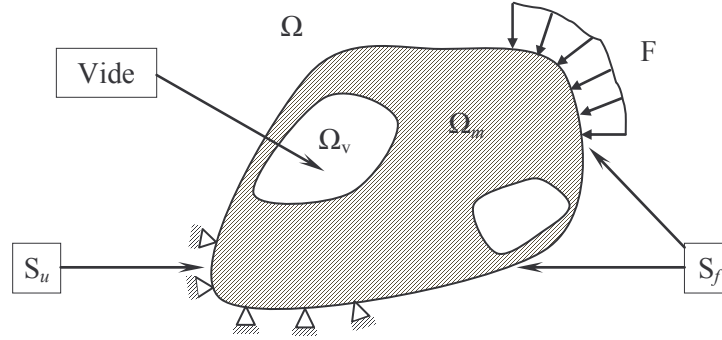


Figure 2-1. Domaine de conception.

Déterminer les zones de présence ou d'absence de matière implique la discrétisation de l'espace de conception. Si x décrit la position spatiale d'un point sur le domaine Ω , la fonction indicatrice $\mu(x)$ indique la présence ou l'absence de la matière :

$$\mu(x) = \begin{cases} 1 & \text{si : } x \in \Omega_m \\ 0 & \text{si : } x \in \Omega_v \end{cases} \quad (2-1)$$

Toute intégrale sur le volume de matière Ω_m d'une fonction quelconque $g(x)$ peut être réécrite comme une intégrale sur le volume total Ω en multipliant l'élément d'intégration $d\Omega$ par $\mu(x)$:

$$\int_{\Omega_m} g(x) d\Omega = \int_{\Omega} g(x) \mu(x) d\Omega \quad (2-2)$$

En particulier, en choisissant $g(x) = 1$, on peut définir le volume V par :

$$V = \int_{\Omega_m} d\Omega = \int_{\Omega} \mu(x) d\Omega \quad (2-3)$$

La formulation du problème d'optimisation topologique requiert le choix d'une fonction mesurant la qualité de la solution, ou fonction objectif. Un choix classique est la

compliance qui représente la souplesse de la structure. Le problème fondamental de la topologie s'intéresse au problème de la détermination de la topologie d'une structure soumise à un chargement statique unique.

En choisissant la compliance comme fonction objectif, le problème d'optimisation consiste à rechercher la structure la plus rigide en imposant une limitation sur le volume. Le problème s'écrit alors :

$$\begin{aligned} & \text{Minimiser : } \ell(U) \\ & \text{avec : } V = \int_{\Omega} \mu(x) d\Omega < V_{Max} \end{aligned} \quad (2-4)$$

Où V_{Max} est le volume limite, U le champ de déplacement obtenu en résolvant le problème mécanique et $\ell(U)$ la compliance.

3. Problème mécanique

La méthode que nous allons développer est générale car elle peut s'appliquer à des problèmes en 3D ainsi qu'à des problèmes non linéaires. Cependant, dans le cadre de la thèse, nous nous sommes limités à des cas simples (problèmes d'élasticité linéaire 2D). Ces cas simples constituent autant de benchmarks permettant de valider l'approche proposée. Le domaine de conception (structure initiale avant optimisation) est continu, bidimensionnel et composé d'un matériau élastique, homogène et isotrope. Il représente une structure soumise à des conditions de chargements et de déplacements statiques. La procédure d'optimisation consiste à définir la distribution optimale de la matière sur le domaine de conception. Cette distribution se traduit par l'absence ou la présence de matière dans chaque élément fini du maillage. La figure 2-2 montre un exemple d'un modèle mécanique représentant un domaine de conception avant optimisation sous certaines conditions de chargements et de déplacements. La frontière S_u représente les surfaces où les déplacements sont connus et la frontière S_f symbolise les surfaces d'application des sollicitations.

Le problème mécanique consiste à rechercher le champ de déplacement U sur le domaine Ω . Ce champ de déplacement s'obtient en résolvant l'équation d'équilibre qui s'écrit en tenant compte des charges surfaciques [Bat90] :

$$\begin{aligned} & \int_{\Omega} H_{ijkl} \epsilon_{ij}(U) \epsilon_{kl}(U^*) d\Omega = \int_{S_f} F U^* ds, \quad \forall U^* \\ & \text{avec : } U^* = 0 \quad \text{et} \quad U = \bar{U} \quad \text{sur } S_u \end{aligned} \quad (2-5)$$

où H_{ijkl} représente les composantes d'un tenseur du 4^{ème} ordre faisant intervenir les caractéristiques physiques du matériau et U^* est un champ de déplacement cinématiquement admissible.

Pour un matériau élastique linéaire, les contraintes sont des fonctions linéaires des déformations :

$$\sigma_{ij} = H_{ijkl} \varepsilon_{kl} \quad (2-6)$$

Et les déformations ε_{ij} sont des fonctions linéaires du déplacement :

$$\varepsilon_{ij}(U) = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (2-7)$$

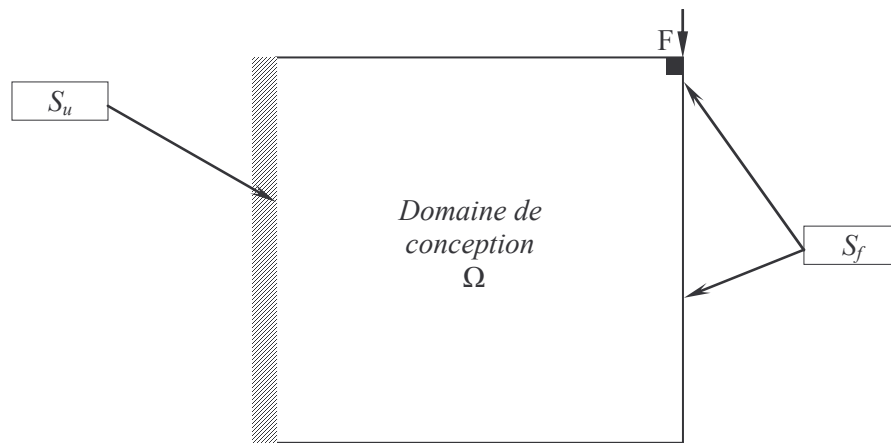


Figure 2-2. Modèle mécanique.

La méthode des éléments finis classique a été utilisée dans ce travail. Le domaine géométrique Ω est alors discrétisé en un nombre $nelt$ éléments finis (Fig. 2-3). L'introduction des polynômes d'interpolation associés à ces éléments dans la formulation variationnelle (2-5) nous ramène à la résolution du système matriciel :

$$K_G \cdot U = F \quad (2-8)$$

Où U est le vecteur des déplacements, F la charge appliquée et K_G la matrice de rigidité globale obtenue par l'assemblage des matrices élémentaires :

$$K_G = \sum_{ei}^{nelt} \int_{\Omega_{ie}} B_{ie}^T H B_{ie} d\Omega_{ie} \quad (2-9)$$

B_{ie} est une matrice reliant les déformations aux déplacements nodaux d'un élément fini ie et H est la matrice de Hooke qui sous l'hypothèse d'état plan de contraintes et en tenant compte d'un comportement isotrope, s'écrit [Bat90] :

$$H = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (2-10)$$

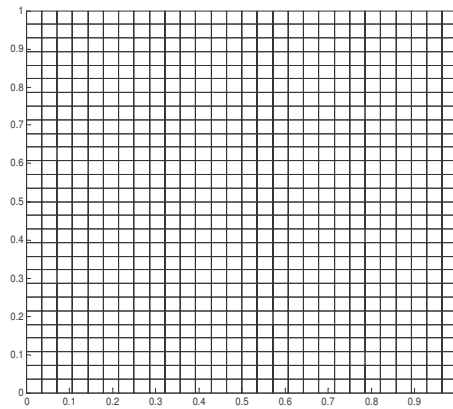


Figure 2-3. Modèle discret.

4. Algorithme génétique

Les algorithmes génétiques [Gol89] adoptent le principe de l'évolution darwinienne des populations biologiques. Une population de taille $PopSze$ où chaque individu est codé en une chaîne de caractères (chromosome), est initialisée aléatoirement au début de l'algorithme. La population évolue au cours des générations en appliquant à chaque génération un processus analogue au processus de sélection naturelle. Les individus les plus adaptés gagnent la compétition de la reproduction tandis que les moins adaptés meurent avant la reproduction. Les opérateurs génétiques de croisement et de mutation sont appliqués ensuite sur la population sélectionnée.

Lors de l'opération de croisement, deux chromosomes s'échangent des parties de leurs chaînes pour donner de nouveaux chromosomes. Ce croisement peut être simple ou multiple. Le taux maximal d'individus qui peuvent se croiser est contrôlé par une probabilité de croisement. En règle générale, on fixe la proportion d'apparition à 0.8 dans un algorithme génétique classique [Gol89].

De façon aléatoire, un gène peut, au sein d'un chromosome, être substitué à un autre. C'est ce qu'il est convenu d'appeler le processus de mutation. De la même manière que pour le croisement, on définit ici un paramètre (probabilité) qui contrôle le taux de mutation. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche complètement aléatoire qui diluerait les effets bénéfiques des principes de sélection et d'évolution.

Les opérateurs génétiques, dont la tâche est l'exploration du domaine de conception dans le but de la recherche de l'optimum global, génèrent une nouvelle population plus performante que la population de la génération précédente. Cette exploration se termine en vérifiant un test de convergence dont la nature sera précisée ultérieurement. L'architecture classique synthétisant les grandes étapes mises en œuvre dans un algorithme génétique est présentée sur la figure 2-4.

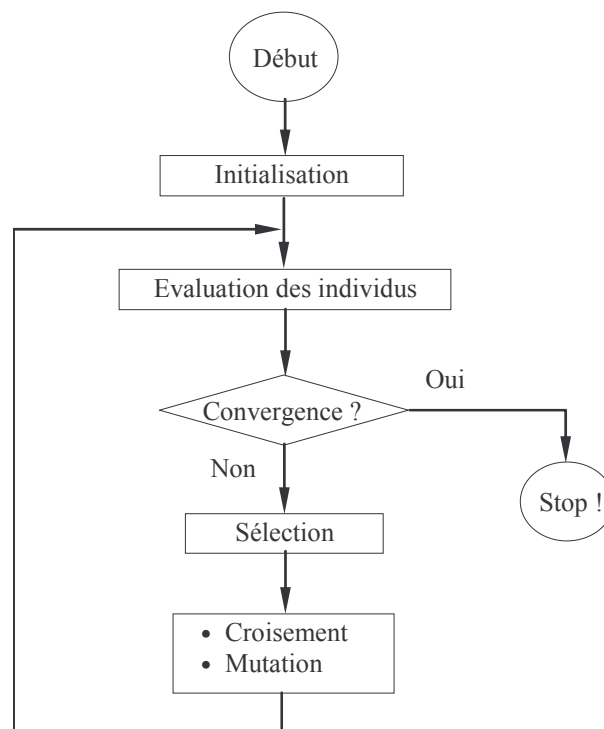


Figure 2-4. Organigramme d'un algorithme génétique.

5. Représentation graphique

La topologie d'une structure mécanique peut être classiquement décrite par un tableau binaire indiquant la présence (1) ou non (0) de la matière [Jen92, Jac00, Fan02]. Cependant, outre la dépendance qu'elle induit par rapport au maillage, cette représentation présente l'inconvénient de ne pas assurer de manière automatique la connectivité des

éléments de la structure. On peut alors être confronté à l'apparition de solutions techniquement irréparables avec des blocs de matière déconnectés les uns des autres. Pour pallier à ce défaut, il est possible d'avoir recours à des techniques de réparation de connectivité, appelées méthodes d'analyse géométrique [Jak00, Fan02]. Cependant, ces méthodes sont extrêmement coûteuses car toutes les structures doivent être examinées et réparées. Dans le cas de problème de grande taille, le coût de cette opération d'analyse géométrique devient prohibitif. C'est pourquoi nous proposons une nouvelle approche qui intègre complètement dès le départ la notion de connectivité. Cette approche dont le principe sera présenté dans le paragraphe qui suit, s'inspire de celle proposée par Tai [Tai98] (Chapitre I : § 4.2.2.2).

5.1 Principe

Une nouvelle représentation graphique, récemment proposée par Tai [Tai98], présente l'avantage de générer des structures dont les éléments sont complètement connectés. Le principe consiste à connecter, d'une manière directe ou indirecte, des régions prédéfinies dans la structure. Ces régions, qu'on qualifiera par la suite de zones de support, seront en général choisies au niveau des surfaces d'application des conditions aux limites (encastrement, chargement surfacique,...). La connexion est assurée par un schéma graphique composé d'un certain nombre de courbes de Bézier où chaque courbe est divisée paramétriquement en trois segments (Fig. 2-5). Chacun de ces trois segments est identifié par la coordonnée paramétrique u . Pour le premier, u varie de 0 à $1/3$, de $1/3$ à $2/3$ pour le second et de $2/3$ à 1 pour le troisième. Le rôle exact joué par ce paramètre ainsi que l'équation paramétrique générale d'une courbe de Bézier sont décrits par l'équation 2-11.

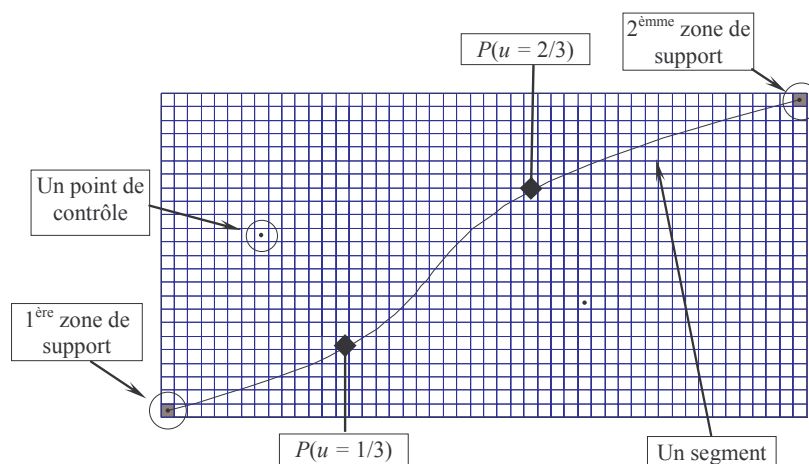


Figure 2-5. Principe de la représentation graphique.

La connexion des zones de support entre elles est assurée par une entité géométrique appelée squelette. Ce squelette est généré par un ensemble d'éléments connectés entre eux par leurs côtés et disposés en suivant l'abscisse curviligne d'une courbe de Bézier. Le squelette, qui constitue l'ossature et les prémisses de la topologie finale, est ensuite complété par une autre entité géométrique qu'on appelle la chair. Les éléments composant la chair sont disposés par couches autour des éléments du squelette (Fig. 2-6). La quantité de chair ajoutée est contrôlée par un paramètre de longueur et par un paramètre d'épaisseur. Le paramètre de longueur correspond à la longueur du segment du squelette que l'on veut enrober. Le paramètre d'épaisseur précise ensuite le nombre de couches d'éléments de chair autour de chaque élément du squelette de ce segment. Au final, le squelette et la chair forment une structure dont les éléments sont complètement connectés, ne contenant ni éléments flottants, ni damiers.

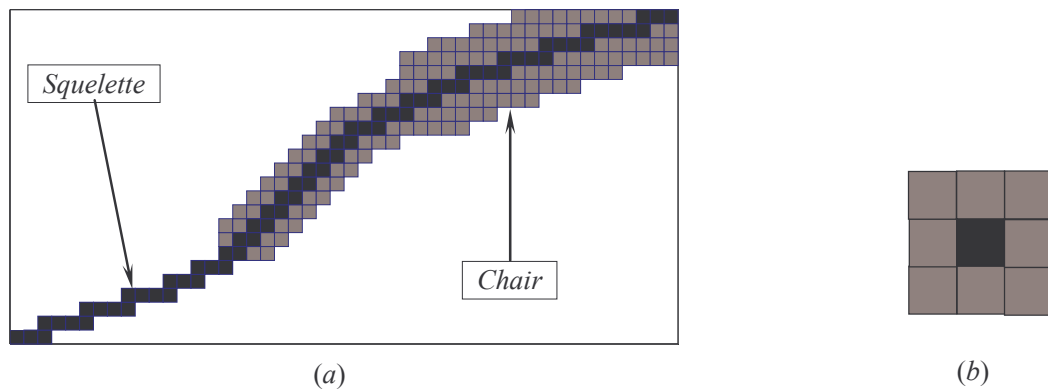


Figure 2-6. (a) squelette + chair, (b) élément du squelette enrobé d'une couche de chair.

5.2 Notre contribution

Dans notre travail, nous adoptons une approche similaire à celle proposée par Tai [Tai98]. Cependant, cette dernière, qui a été développée pour l'étude des mécanismes, comporte plusieurs limitations qui empêchent son utilisation dans un contexte d'optimisation topologique général. La première limitation concerne le choix des zones de support qui doivent, d'une part, se cantonner au niveau des surfaces d'application des conditions aux limites et, d'autre part, rester fixes dans toutes les structures générées par l'algorithme génétique. Ces limitations peuvent conduire l'algorithme à une convergence prématurée en limitant la zone d'exploration. En effet, en réduisant les possibilités d'exploration, on dégrade la capacité d'adaptation des courbes de Bézier à des géométries complexes. Or, bien souvent, les solutions définissant une distribution optimale de la matière sont justement caractérisées par des géométries complexes. La deuxième limitation

de l'approche de Tai touche à la distribution des éléments de la chair. Cette distribution doit se faire de manière symétrique de part et d'autre du squelette. Il n'est donc pas possible d'avoir de la chair d'un côté et pas de l'autre, ce qui nuit encore une fois, à la capacité d'exploration de l'algorithme génétique. Enfin, la troisième limitation est à nouveau relative à la répartition de la chair qui doit s'opérer sur des segments de longueur fixe égale à $1/3$.

La technique consistant à exploiter des zones dans le domaine de conception, dites de support dans le travail de Tai, n'est pas une nouveauté. L'utilisation de cette technique remonte aux premiers travaux de Jakiela et de ses collaborateurs [Jak00, Cha96, Cha94] qui ont appelé ces zones '*Seed Points*'. Dans tous les travaux qui ont adopté cette technique, ces zones contiennent toujours de la matière et restent fixes dans toutes les structures solutions. Le rôle de ces zones est de servir comme supports aux niveaux des surfaces d'application des conditions aux limites. La localisation de ces zones sur les surfaces d'application des conditions aux limites est intuitive et ne se base sur aucun critère. Par conséquent, une mauvaise localisation conduit à pénaliser l'algorithme en limitant la zone d'exploration, ce qui entraîne une convergence prématurée. Kane [Kan96(a), Kan96(b), Kan95], en traitant plusieurs types de problèmes, a constaté les limitations induites par une telle technique.

Dans la présente approche, nous avons modifié sur l'approche de Tai dans le but de pouvoir traiter une catégorie de problèmes plus large que les mécanismes.

Pour déterminer les zones de support, le domaine de conception est tout d'abord discrétisé en éléments finis (Fig. 2-3). Avant de procéder à l'optimisation, une analyse de la structure complète par la méthode des éléments finis permet ensuite de localiser les zones de forte concentration de contraintes mécaniques dans le domaine de conception (Fig. 2-7). L'intersection entre les surfaces d'application des conditions aux limites (Fig. 2-7b) et les surfaces représentant les zones de forte concentration de contraintes (Fig. 2-7a), forme les zones de support (Fig. 2-7c). Dans le but de conférer plus de flexibilité au schéma graphique nous avons donc donné une définition plus large aux zones de support que celle proposée par Tai. Une autre différence notable concerne la présence ou non de matière dans les zones de support. Avec l'approche de Tai, tous les éléments de ces zones contiennent de la matière alors que seuls les éléments représentant un point de départ ou un point d'arrivée d'une courbe de Bézier sont concernés dans notre contribution. De plus, dans le cas d'une zone composée de plusieurs éléments, les points de départ et d'arrivée peuvent être sélectionnés parmi tous ces éléments. En d'autres termes, les courbes de Bézier sont d'extrémités mobiles, ce qui, là encore, participe à la flexibilité du schéma graphique. Enfin, toujours dans l'esprit de fournir le plus de flexibilité possible au schéma

graphique, les courbes de Bézier elles mêmes peuvent servir de zone de support (courbe N°1 Fig. 2-8a), ce qui n'est pas possible dans l'approche de Tai.

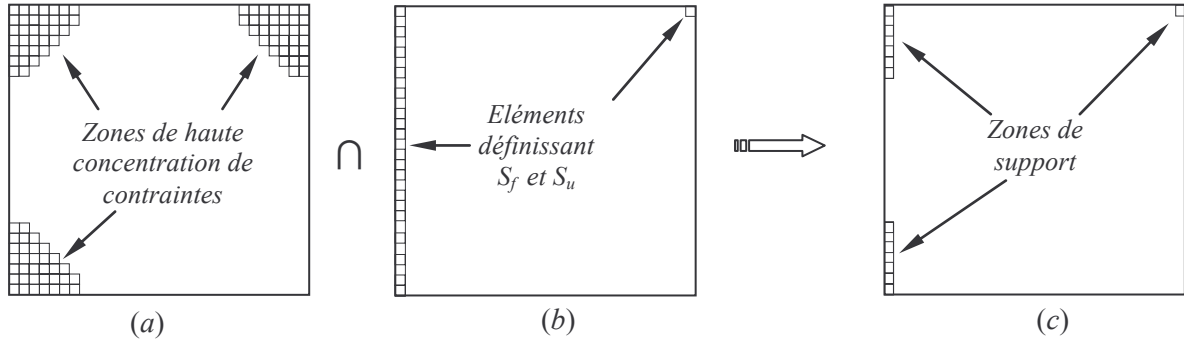


Figure 2-7. (a) zones de haute concentration des contraintes. (b) conditions aux limites (c) zones supports.

L'étape suivante consiste à établir le schéma graphique, composé de courbes de Bézier, connectant toutes les zones de support. Les courbes de Bézier sont définies de manière classique par un point de départ, un point d'arrivée et plusieurs points de contrôle. L'équation paramétrique générale définissant une courbe de Bézier de degré n s'écrit comme suit [Yam88] :

$$P(u) = \sum_{k=0}^n \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k} Q_k \quad (2-11)$$

Où $u \in [0,1]$ représente la coordonnée paramétrique de la courbe, $n+1$ est le nombre de points de contrôle, Q_k ($k = 0, \dots, n$) représente un point de contrôle, Q_0 étant le point de départ et Q_n le point d'arrivée.

Dans l'approche que nous proposons, il est possible d'utiliser des courbes de Bézier de degré quelconque. De manière pratique, nous nous sommes néanmoins limités à des courbes de Bézier cubiques ($n=3$). Elles possèdent en effet un minimum de points de contrôle, ce qui réduit le nombre de variables, tout en assurant une bonne flexibilité. Il s'agit donc d'un bon compromis entre la flexibilité et le coût du calcul. De plus, ces courbes présentent l'avantage de ne posséder qu'un minimum de points de croisement, ou points double (Fig. 2-12a), et un minimum de points d'inflexion. Il s'agit d'un avantage supplémentaire car les points doubles posent des problèmes algorithmiques dans la construction du squelette. Ce sont donc des points indésirables dont il faut limiter au

maximum la présence. Pour empêcher leur apparition, un traitement spécifique, basé sur des limitations, a été mis en place.

Le point de départ de chaque courbe se trouve au centre d'un élément fini dans la première zone support, et le point d'arrivée se trouve au centre d'un élément dans la deuxième zone support que l'on souhaite connecter avec la première. Les éléments, connectés entre eux par leurs côtés et disposés le long de l'abscisse curviligne des courbes de Bézier forment le *squelette* (Fig. 2-8b). Chaque courbe est divisée en trois segments de longueurs paramétriques éventuellement différentes (par exemple 5/10, 2/10 et 3/10) à la différence de Tai où chaque segment est de longueur fixe égale à 1/3. Cette nouvelle notion introduite dans notre approche concernant la longueur variable des segments va donner plus de souplesse à la distribution de la matière autour du squelette.

Une certaine quantité d'éléments est ensuite ajoutée au squelette afin de compléter la structure et lui donner sa forme finale. Cette quantité, représentant la *chair*, est ajoutée au squelette par couches où le nombre de couche détermine la quantité ajoutée. Ici aussi nous dépassons la définition classique de la chair proposée par Tai. En effet, les éléments de la chair peuvent être ajoutés en entourant complètement les éléments du squelette, selon la définition classique introduite par, c'est-à-dire simultanément des deux côtés du squelette, ou en ajoutant des éléments soit du côté droit du segment concerné soit du côté gauche. La figure 2-8b montre les trois types de chair proposés dans notre approche. Avec le premier type, la chair est disposée sur la droite du squelette. Un exemple en est donné avec la courbe N°1 de la figure 2-8a. Son squelette, et la chair qui est ajoutée, uniforme et d'épaisseur égale à 1, sont présentés sur la figure 2-8b. la droite du squelette est identifiée en partant du point de départ de la courbe, de coordonnées (0, 0), et en se déplaçant vers le point d'arrivée, de coordonnées (1, 1).

Avec le deuxième type, la chair est disposée sur la gauche du squelette. Une illustration est fournie par la courbe N°2 de la figure 2-8a. Le squelette et la chair qui lui sont associés sont présentés sur la figure 2-8b. Pour repérer la gauche du squelette, on adopte la même convention qu'avec le premier type de chair, en partant du point de départ de la courbe N°2, de coordonnées (1, 1), il est en effet clair, en examinant la figure 2-8b, que la chair est bien disposée à gauche du squelette. Enfin, le troisième type correspond à la définition utilisée par Tai. Il consiste à enrober le squelette simultanément à droite et à gauche. C'est la courbe N°3 de la figure 2-8a qui correspond à cette situation ainsi qu'on peut le constater à l'examen de la figure 2-8b.

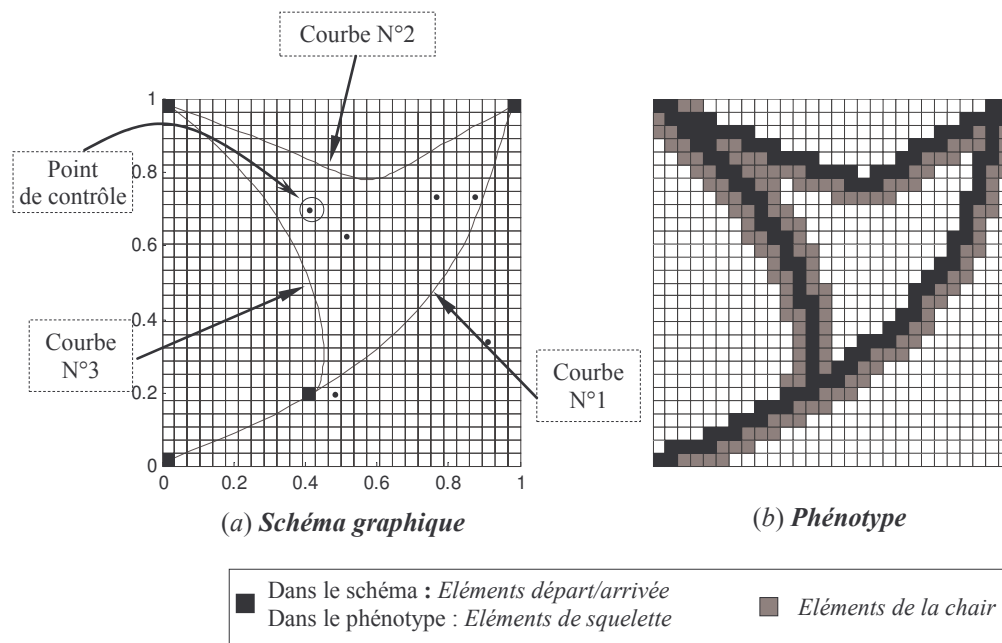


Figure 2-8. Représentation graphique.

La structure de la figure 2-9 illustre un exemple où les trois types de chair sont disposés avec différentes valeurs d'épaisseurs sur des segments de longueurs différentes. La disposition des éléments de chair, sur des segments de longueur variable, à droite et/ou à gauche du squelette et avec des quantités différentes permet la génération de topologies complexes qui ne restent pas uniquement dépendantes de la capacité des courbes de Bézier à s'adapter à des formes complexes. Sur ce point, notre approche se démarque de l'approche de Tai. Enfin, on note que l'ensemble squelette et chair forme une seule structure où les éléments sont complètement connectés ce qui est était l'objectif affiché. De ce point de vue, notre approche reste par contre similaire aux travaux de Tai.

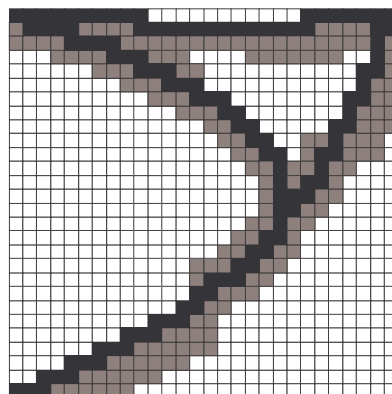


Figure 2-9. Structure d'éléments complètement connectés.

5.3 Codage

L'exploitation de la représentation géométrique présentée dans la section précédente dans le cadre d'un algorithme génétique nécessite le codage des paramètres de cette représentation. Avant de présenter le codage, nous commençons par exposer toutes les variables sous forme de graphes orientés. La structure de la figure 2-8b est considérée à titre d'exemple. Le schéma des graphes orientés associés à cette structure est détaillé sur la figure 2-10. A chaque sommet du graphe, une valeur réelle ou un nombre entier désigne une information affectée à chaque variable de la représentation géométrique. Les sommets sont reliés par des arêtes (arcs), désignés par des flèches. Les sommets et les arcs relèvent de la terminologie de la théorie des graphes.

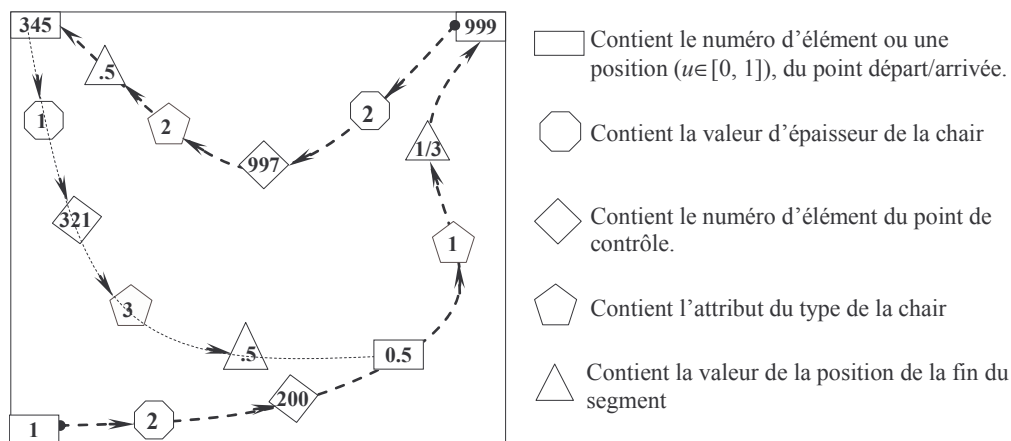


Figure 2-10. Représentation graphique du codage.

Pour ne pas surcharger la figure 2-10, nous avons présenté seulement les informations concernant un segment de courbe. Pour le premier graphe, par exemple, nous avons présenté le point de départ, l'épaisseur de chair du premier segment, un point de contrôle, l'attribut de la chair du premier segment, la position du fin de ce premier segment et enfin le point d'arrivée de la courbe.

Les points de départ et d'arrivée de chaque courbe de Bézier se trouvent au centre des éléments des zones de support, et dans ce cas le numéro de l'élément correspondant est affecté au sommet représentant cette variable. Dans le cas d'un point de départ ou d'arrivée glissant sur une autre courbe, le numéro d'élément sera remplacé par une valeur réelle u comprise entre zéro et un. D'une manière pratique, et dans le but de limiter le nombre de variables de l'algorithme génétique, la variable paramétrique u prend des valeurs dans un

tableau où le nombre des composants est limité. Par exemple, u peut être égale à 0, 1/10, 2/10, ..., ou 1, ce qui limite le choix de la variable continue u à 11 valeurs discrètes.

Les points de contrôle se situent au centre des éléments finis. Par conséquent, les numéros de ces éléments finis sont affectés aux variables qui correspondent aux points de contrôle. Les segments de chaque courbe de Bézier sont de longueur variable. Une variable paramétrique contrôle les longueurs de ces segments dont les valeurs sont stockées dans un tableau. La chair disposée autour d'un segment est contrôlée par deux variables. La première variable concerne la quantité d'éléments ou de matière ajoutée. Cette quantification est exprimée par des nombres entiers. Le nombre entier '1', par exemple, désigne une seule couche d'éléments (fig. 2-8b). La deuxième variable concerne le type de la chair. Un attribut est affecté à cette variable précisant le type de disposition de ces éléments. L'attribut '1' désigne une disposition sur la droite du segment, l'attribut '2' une disposition sur le côté gauche et l'attribut '3' correspond à la définition classique de Tai, c'est-à-dire des deux côtés.

Le codage consiste à mettre toutes les variables de la représentation géométrique en une chaîne de bits appelée souvent chromosome. Le génotype ou chromosome doit être compatible avec les opérateurs génétiques de croisement et de mutation. Le codage que nous proposons, sous forme d'un tableau, semble le plus adéquat car il facilite les opérations de croisement. Un exemple de codage, associé à la représentation graphique de la figure 2-10 est donné dans le tableau 2-1.

	Point départ	1 ^{ère} valeur chair	1 ^{er} point de contrôle	Type 1 ^{ère} chair	1 ^{er} segment	...	Point d'arrivée
Graphe N°1	1	2	200	1	1/3	...	999
Graphe N°2	999	2	997	2	0.5	...	345
Graphe N°3	345	1	321	3	0.5	...	0.5

Tableau 2-1. Chromosome.

Douze variables sont associées à chaque courbe dans notre approche. Quatre variables représentent les points de contrôle, y compris le point de départ et le point d'arrivée. Trois variables sont associées aux deux premiers segments et deux au troisième : la valeur de l'épaisseur de la chair, l'attribut du type de la chair et la position de la fin du segment (sauf pour le troisième segment qui arrive sur le point d'arrivée). Ce nombre relativement faible permet de limiter le coût du calcul de l'algorithme génétique. Mais il s'agit également d'une bonne solution de compromis car on a constaté que les géométries générées étaient suffisamment riches pour promouvoir une large variété de topologies.

5.4 Décodage

Les informations stockées dans le tableau 2-1 seront utilisées pour la construction du phénotype. Ce passage du génotype au phénotype s'appelle décodage. Pour construire la structure, nous commençons par le squelette avant de traiter la chair. Afin de maintenir la stabilité de la structure, le squelette sera composé d'éléments connectés entre eux par les côtés. Ces éléments sont disposés en suivant l'abscisse curviligne de la courbe paramétrique cubique de Bézier (équation 2-11). Connaissant l'élément de départ, l'élément suivant est déterminé en trouvant le côté qui remplit la condition d'intersection avec la courbe de Bézier. Cela se traduit par la résolution ou la vérification de l'existence des solutions d'une équation cubique. Dans notre travail nous avons utilisé les formules analytiques de Cardan (Annexe A) pour calculer les racines de cette équation, ce qui permet d'accélérer notablement la construction informatique du squelette. Cette opération se répète pour chaque élément jusqu'à atteindre le dernier élément d'arrivée qui est connu. Cette technique assure la connectivité et la stabilité du squelette qui lui-même assure la connectivité de la structure complète. La figure 2-11 montre le principe de la technique adoptée pour la construction du squelette.

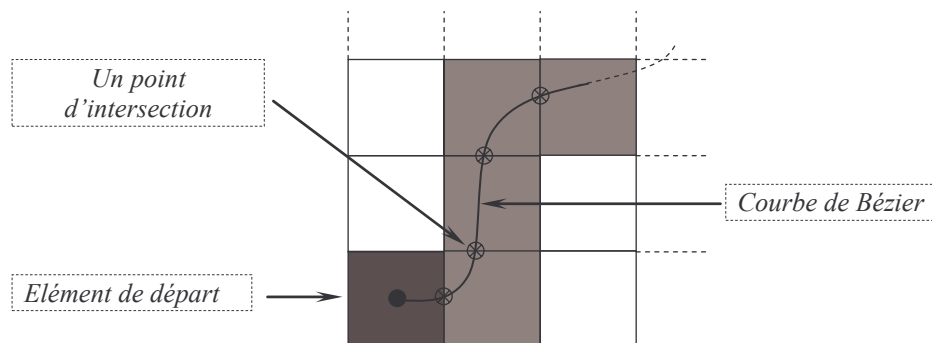


Figure 2-11. Construction du Squelette.

Les points de contrôle qui gèrent la courbure des courbes de Bézier sont initialisés aléatoirement au début de l'algorithme. Ces points, localisés aux centres d'éléments du maillage, changent de coordonnées au cours des générations par l'application des opérateurs génétiques. Bien que les courbes planes cubiques paramétrées soient très flexibles, la gestion aléatoire des points de contrôle peut conduire à l'intersection de la courbe avec elle-même en formant ainsi une boucle (Fig. 2-12a). Les courbes possédant des boucles posent des difficultés pendant la construction du squelette et peuvent provoquer un bug dans notre code de calcul. De plus, ce phénomène de boucles, bien qu'il montre le degré de souplesse des courbes de Bézier, peut engendrer des formes

indésirables. Ce problème de boucle peut être traité sous forme de limitations. Pour cela, il est nécessaire d'étudier la courbe de Bézier sous sa forme paramétrique [Yam88] :

$$P(u) = (X(u), Y(u)) \quad (2-12)$$

où $X(u)$ et $Y(u)$ sont des polynômes cubiques.

La fonction P admet comme fonctions dérivées première et seconde : $P'(u) = (X'(u), Y'(u))$ $P''(u) = (X''(u), Y''(u))$. Les courbes paramétrées sont définies pour des valeurs restreintes du paramètre, prises généralement dans l'intervalle $[0,1]$. Wang, Su et Liu [Wan81, Su83] ont démontré que la présence d'une boucle peut être déterminée en examinant la fonction :

$$\Phi(u) = \text{Det}(P'(u), P''(u)) = X'(u) \cdot Y''(u) - X''(u) \cdot Y'(u) \quad (2-13)$$

Cette fonction est proportionnelle au signe de la courbure de la courbe au point $P(u)$. L'équation $\Phi(u) = 0$ indique la présence d'un point d'inflexion, puisque, pour un tel point, les vecteurs de la dérivée première et de la dérivée seconde sont linéairement dépendants. Cette dépendance montre que $\Phi(u)$ est une fonction cubique. Les auteurs montrent par la suite que le terme cubique est nul et c'est ainsi que $\Phi(u)$ devient une fonction quadratique de la forme :

$$\Phi(u) = p u^2 - 2q u + 2r \quad (2-14)$$

Le discriminant réduit $\Delta = q^2 - 2pr$ joue alors un rôle prépondérant. Wang, Su et Liu [Wan81, Su83] ont en effet montré que les caractéristiques de la courbe sont entièrement déterminées à partir des valeurs p , q et r . En particulier, la courbe contient une boucle si :

$$\Delta < 0 \quad (2-15a)$$

La coordonnée paramétrique u_0 associée au point de croisement $\Phi(u_0)$ de cette boucle est donnée par :

$$u_0 = \frac{q \pm \sqrt{-3\Delta}}{p} \quad (2-15b)$$

Si la condition (2-15a) est satisfaite, la structure concernée sera considérée comme inadmissible. Dans ce cas, une limitation artificielle sera ajoutée aux limitations associées à cette structure solution. Nous attribuons à cette limitation une valeur égale à la solution maximale u_0 (équation 2-15b) car cette valeur est toujours bornée dans l'intervalle $]0, 1[$.

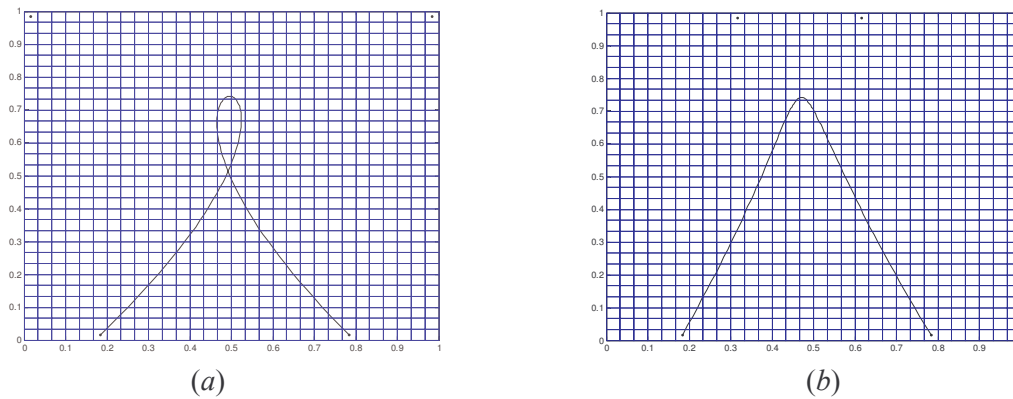


Figure 2-12. Courbe de Bézier cubique (a) avec boucle (b) sans boucle.

5.5 Fonction performance

Le principe des algorithmes génétiques est la recherche de l'individu le plus performant. Ce dernier est évalué par une fonction performance qualifiée souvent de fonction *Fitness* ou fonction mérite. L'individu le plus performant sera celui pour lequel la fonction *Fitness* est maximum. Pour un problème d'optimisation avec recherche de maximum, la valeur de la fonction performance affectée à cet individu peut donc être égale à la valeur de la fonction objectif. Par contre, pour un problème d'optimisation avec recherche de minimum, la fonction performance *Fitness* devra être considérée comme la réciproque de la fonction objectif f :

$$Fitness(x_i) = \frac{1}{f(x_i)} \quad (2-16)$$

où x_i représente le $i^{\text{ème}}$ individu de la population.

Dans les problèmes concrets d'optimisation structurale, rencontrés dans l'industrie, la solution optimale finale doit satisfaire certaines conditions technologiques ou fonctionnelles. Ces conditions se traduisent par des limitations de type égalité ou inégalité et la formulation du problème prend la forme suivante :

$$\text{Maximiser : } Fitness(x) = \frac{1}{f(x)} \quad (2-17)$$

$$\begin{aligned} \text{sous limitations : } & g_j(x) \geq 0. & j=1, \dots, J \\ & h_k(x) = 0. & k=1, \dots, K \end{aligned}$$

Dans ce problème, il y a J limitations sous forme d'inégalité et K limitations sous forme d'égalité. La fonction objectif f est supposée positive et non nulle.

La fonction performance joue un rôle très important dans un algorithme génétique au même titre que le codage. En effet, pour que l'algorithme génétique converge, il est préférable que cette fonction ne comporte pas trop de maxima locaux et ne présente pas de maximum local isolé.

Un choix classique de fonction objectif est la compliance [Ben03, All02, Jak00, Kan96(a)]. L'intérêt principal de ce choix est le caractère différentiable de cette grandeur. Ce choix se justifie pleinement dans le cadre d'une approche déterministe où il est nécessaire de dériver pour faire l'analyse de sensibilité. Dans notre cas, où l'approche stochastique ne nécessite pas de dérivation, le choix de la compliance comme fonction objectif n'est pas aussi vital. Du point de vue de l'ingénieur, il peut même être problématique. La compliance de la structure n'est pas toujours une grandeur physique simple à appréhender physiquement même si elle représente le travail des forces extérieures. L'approche adoptée par Jakiela et ses collaborateurs [Cha94] s'inscrit dans un contexte où la fonction objectif n'est pas différentiable. Le problème d'optimisation est posé sous la forme de la maximisation de la raideur de la structure en supposant que la raideur R est inversement proportionnelle au déplacement maximal $\delta_{\max}$ de la structure :

$$R = \frac{1}{\delta_{\max}} \quad (2-18)$$

Ces derniers auteurs ont proposé aussi de maximiser le rapport de la raideur au volume :

$$Fitness = \frac{R}{V} = \frac{1}{\delta_{\max} \cdot V} \quad (2-19)$$

Dans cette formulation, deux quantités de grandeurs différentes sont associées pour exprimer un seul objectif. Mais une telle approche augmente considérablement le risque de converger vers des optima locaux. Le problème peut alors être formulé autrement en prenant en considération les deux objectifs par la sommation des deux quantités multipliées par des coefficients de pondération. On parle dans ce cas d'un problème d'optimisation multi-objectif, ce qui ne fait pas l'objet des travaux de recherche présentés dans ce mémoire.

Dans notre contribution, nous nous sommes limités à la minimisation de la compliance avec une limitation sur le volume et à la minimisation du volume avec une limitation sur la compliance. Ces choix classiques vont nous permettre de valider notre approche en comparant nos résultats avec ceux des approches déterministes.

5.6 Analyse structurale

La méthode d'analyse structurale utilisée dans notre approche est la méthode des éléments finis classique. Chaque structure, solution du problème, est analysée par éléments finis si cela est nécessaire. Le résultat de cette analyse sera exploité par la suite pour évaluer l'individu associé à cette structure. Cette analyse est l'opération la plus coûteuse dans un algorithme génétique car elle concerne tous les individus admissibles, et peut être aussi quelques individus inadmissibles. En effet, pour une population de grande taille (quelques centaines d'individus), le coût de l'analyse structurale est considérable. Ce coût devient critique dans certains problèmes d'optimisation tels que les calculs 3D et les problèmes non linéaires.

L'approche souvent adoptée dans de nombreux travaux d'optimisation topologique des structures continues [Ben88, Cha94, Cap03, Sig01] est basée sur la relaxation des propriétés physiques du matériau. Le principe consiste à établir une relation entre le module de Young et un paramètre dit de densité [Ben88]. Ce dernier représente généralement le volume relatif d'un élément fini. Une telle approche est très utile dans un algorithme génétique avec un codage binaire, puisqu'elle permet de remédier au problème de connectivité. Par contre, l'analyse devient plus coûteuse car tous les éléments finis sont présents. Ainsi tous les éléments du maillage sont concernés par les phases classiques du maillage et de la résolution. Même les éléments associés à une faible densité doivent être pris en considération bien qu'ils participent peu à la tenue mécanique de la structure. Avec l'approche par densité, ces éléments alourdissent donc inutilement le calcul. De plus, dans le cadre non-linéaire des grands déplacements et des grandes déformations, ces éléments subissent des déplacements de très forte amplitude. Ce phénomène induit des difficultés numériques dues au mauvais conditionnement qu'ils provoquent dans la matrice de raideur [Kim04].

Dans notre travail, la connectivité est assurée, ce qui nous permet de ne prendre en compte que les éléments contenant de la matière. Les éléments finis qui ne contiennent pas de matière sont éliminés. Par conséquent, la taille du problème mécanique se réduit d'une manière considérable et l'évaluation devient beaucoup moins coûteuse.

5.7 Traitement des limitations

Les problèmes d'optimisation avec limitations se présente généralement sous la forme de l'équation (2-17) lorsqu'on les traite avec des algorithmes génétiques. Les méthodes de pénalité sont les plus utilisées pour le traitement des limitations [Coe02]. Les limitations

peuvent se présenter sous formes d'égalités ou d'inégalités. Les limitations sous forme d'égalité sont, souvent, transformées sous forme d'inégalité comme suit [Deb00] :

$$g_{k+J}(x) \equiv \varepsilon - |h_k(x)| \geq 0 \quad k = 1, \dots, K \quad (2-20)$$

où ε est une valeur positive très faible. Cela augmentera le nombre de contraintes sous forme d'inégalité à : $m = J+K$.

En adoptant la transformation (2-20), les limitations sous forme d'égalité et les limitations sous forme d'inégalité peuvent être additionnées et introduites dans la formulation de la fonction performance. On obtient ainsi la forme pénalisée de la fonction de performance d'un problème de maximisation sous limitations d'inégalités [Coe02] :

$$Fitness(x) = \frac{1}{f(x) + \sum_{j=1}^m R_j \langle g_j(x) \rangle^2} \quad (2-21)$$

où $\langle g_j(x) \rangle$ est la valeur absolue de la quantité de violation des limitations dans le cas où l'opérande est négatif, et zéro dans le cas contraire. Le paramètre de pénalité R_j , correspondant à la $j^{ème}$ limitation, sert à mettre la valeur de la quantité de violation de la limitation $g_j(x)$ à la même échelle de grandeur que la fonction objectif $f(x)$. Il est clair qu'avec cette définition, les individus violant le plus de limitations dégraderont le plus leur performance. La formule (2-21) permet ainsi de pondérer la performance pure en terme de fonction objectif f par le degré de violation des limitations.

L'exploitation de la fonction performance telle qu'elle est donnée par l'équation (2-21) exige l'évaluation de tous les individus. Cette évaluation nécessite au préalable une analyse par la méthode des éléments finis dans notre travail, ce qui demeure très coûteux même si l'élimination des éléments de densité nulle permet d'en amortir le coût comme cela a été expliqué au paragraphe 5.6. D'après Deb [Deb00], le fait que le principe des algorithmes génétique soit de rechercher le meilleur individu n'exige pas l'évaluation de chacun d'entre eux. Il suffit seulement de distinguer les plus performants des moins performants. Suivant ce principe, les individus représentant des solutions admissibles sont les plus favorisés, et leur évaluation est obligatoire afin de pouvoir les comparer. Les autres individus sont d'autant plus défavorisés que leurs degrés de violation de limitations sont élevés. Leur évaluation par éléments finis n'est alors plus nécessaire puisque c'est ce degré de violation qui permet de définir leur position dans la population. Notons enfin qu'il est difficile d'ajuster le paramètre de pénalité R_j (voir le chapitre I § 4.2.2.4). Afin de remédier à ce problème, Deb [Deb00] propose d'introduire dans la formulation de la fonction

performance la quantité de violation de contraintes sans le coefficient de pénalité. La fonction performance pénalisée prend alors la forme suivante :

$$Fitness(x) = \begin{cases} \frac{1}{f(x)} & \text{si } x \in \tilde{E} \\ \frac{1}{f_{\max} + \sum_{j=1}^m G_j(x)} & \text{si non} \end{cases} \quad (2-22a)$$

$$\text{avec : } G_j(x) = \max[0, -g_j(x)], \quad (2-22b)$$

$\tilde{E}$ représente l'espace des solutions admissibles et $f_{\max}$ l'évaluation de la fonction objectif de l'individu le moins bon parmi les individus admissibles.

Cette formulation, telle quelle est proposée par Deb [Deb00], ne tient pas compte des grandeurs d'échelles entre $f_{\max}$ et la somme des quantités de violation des limitations $\sum_{j=1}^m G_j$.

Pour mettre ces deux quantités à la même échelle, nous avons normalisé ces limitations de telle sorte qu'elles soient comprises entre 0 et 1 :

$$\hat{G}_j = |G_j - G_{\min}| / (G_{\max} - G_{\min}) \quad (2-23)$$

En introduisant (2-23) dans (2-22a), et afin de mettre la valeur $f_{\max}$ et la somme des quantités de violation des limitations $\sum_{j=1}^m G_j$, la fonction performance prend la forme finale suivante :

$$Fitness(x) = \begin{cases} \frac{1}{f(x)} & \text{si } x \in \tilde{E} \\ \frac{1}{f_{\max} \times \left(1 + \sum_{j=1}^m \hat{G}_j(x)\right)} & \text{si non} \end{cases} \quad (2-24)$$

5.8 Opérateurs génétiques

5.8.1 Sélection

La diversité génétique, variété des génotypes dans la population, et la pression sélective sont les deux facteurs importants dans un processus de recherche stochastique tel que les algorithmes génétiques. Ces deux facteurs sont fortement dépendants puisqu'une

augmentation de la pression sélective diminue la diversité dans la population, et vice versa. Autrement dit, une forte pression sélective rend la population très homogène, ce qui conduit à une convergence prématurée de l'algorithme. Par contre, une faible pression sélective rend la recherche inefficace car le caractère aléatoire qu'elle privilégie ne permet pas l'émergence des individus les plus performants. De ce fait, il est important de trouver un compromis entre ces deux facteurs, ce que les méthodes dites '*Sampling Mechanisms*' peuvent réaliser.

La méthode de sélection dite '*Stochastic Universal Sampling*' (SUS), proposée par Baker [Bak94], fait partie de la famille de méthodes de type '*Sampling Mechanisms*'. Nous l'avons utilisé dans ce travail car elle permet d'établir un compromis entre la diversité génétique et la pression sélective. Dans cette méthode, chaque individu est représenté par un secteur rectangulaire dont la longueur correspond à sa probabilité de sélection (Fig. 2-13). Le secteur total, de longueur 1, est marqué par des indices, désignés par des flèches sur la figure 2-13. Ces indices sont régulièrement espacés. Considérons N indices, nombre d'individu que nous souhaitons sélectionner. La distance entre chaque indice est égale à $1/N$. La position du premier est défini aléatoirement dans l'intervalle $[0, 1/N]$ (*Indice aléatoire* sur la figure 2-13). Si le premier indice, noté α sur la figure 2-13, est strictement plus petit que $\frac{1}{N}$ ($\alpha < \frac{1}{N}$), le dernier indice, égal à $\alpha + \frac{N-1}{N}$, sera strictement plus petit que 1. Si α est égal à $\frac{1}{N}$, le dernier indice sera égal à $\frac{1}{N} + \frac{N-1}{N} = 1$. La méthode consiste à introduire dans la nouvelle population tous les individus marqués par les indices.

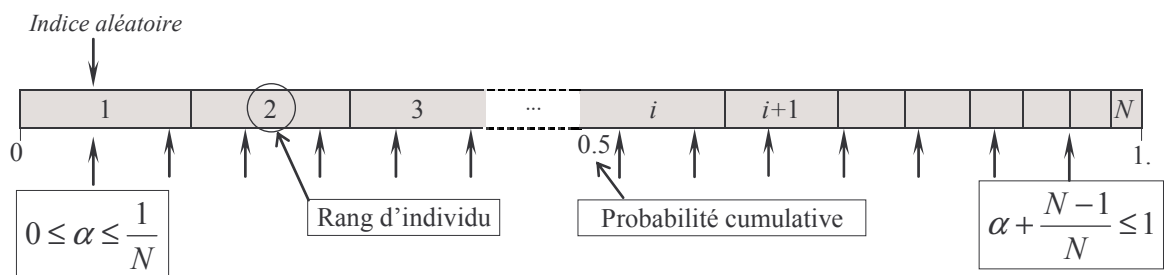


Figure 2-13. Principe de la méthode de sélection SUS.

Dans une population de taille N , les individus sont classés par rang de 1 à N . Le rang 1 est attribué au meilleur individu et le rang N à l'individu le moins performant.

La probabilité de sélection d'un individu est proportionnelle à une performance artificielle calculée en fonction du rang de cet individu. Il existe plusieurs variantes dans la littérature qui permettent le calcul de la performance artificielle en fonction du rang. La variante linéaire [Bak94] est donnée par la formule suivante :

$$Fitness_{Rank}(x_i) = (2 - P_s) - \frac{2(P_s - 1) \cdot (r_i - N)}{N - 1} \quad (2-25)$$

où P_s est la pression sélective, comprise entre 1 et 2, et r_i le rang de l'individu x_i .

Il est facile de voir que, P_s , $P_s - 1$ et $2 - P_s$ étant positifs, la fonction performance artificielle est une fonction positive qui décroît linéairement de P_s à $2 - P_s$ (Fig. 2-14).

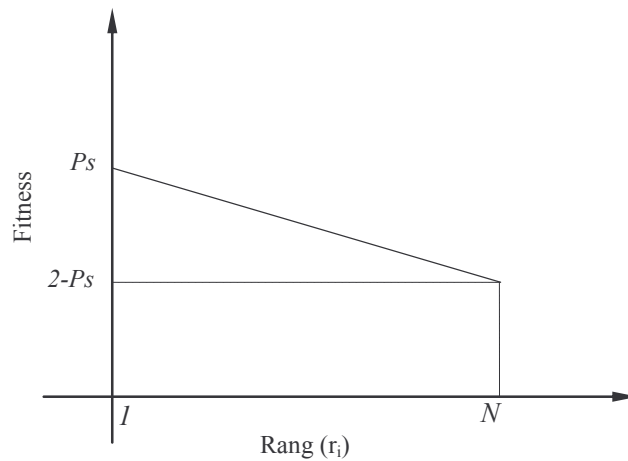


Figure 2-14. Fonction performance artificielle – variante linéaire.

On remarque que le cas extrême $P_s = 1$, qui correspond à la pression sélective la plus faible possible, donne une fonction performance égale à 1 et identique pour toute la population (Fig. 2-15). Ce cas extrême est bien entendu à éviter car aucun individu, et en particulier les plus performants, n'est alors favorisé par rapport aux autres. La recherche du meilleur individu devient alors inefficace.

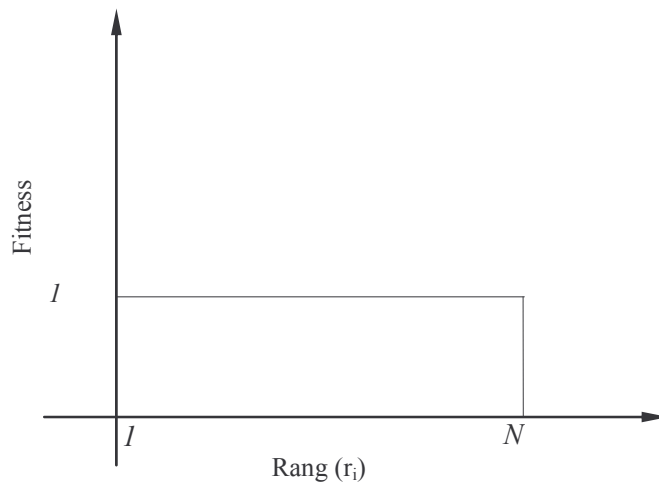


Figure 2-15. Fonction performance artificielle – $P_s = 1$.

Enfin, on constate que l'autre cas extrême, correspondant à $P_s = 2$, donne la décroissance linéaire la plus forte avec une pente égale à $-\frac{2}{N-1}$. En effet, on a :

$$\frac{2}{N-1} > \frac{2(P_s-1)}{N-1} \quad \forall P_s \in [1, 2]$$

En pratique nous avons souvent opté pour la pression sélective la plus forte, c'est-à-dire $P_s = 2$. Les nombreux tests numériques que nous avons réalisés semblent en effet montrer qu'il s'agit du choix le plus pertinent pour converger dans les meilleures conditions.

En adoptant la méthode de sélection '*Stochastic Universal Sampling*', combinée avec la variante linéaire de classement des individus par rang, nous cumulon les avantages des méthodes de sélection par roulette, par rang et par tournoi. C'est une très bonne technique qui permet de maintenir la diversité génétique en choisissant seulement le schéma de classement par rang des individus ou en faisant varier les paramètres de la pression sélective.

5.8.2 Croisement

Le croisement est l'un des paramètres les plus importants en algorithme génétique. Le principe de cet opérateur est de produire deux nouveaux individus, les enfants, en échangeant des informations entre deux autres individus appelés parents. Le croisement s'effectue en deux étapes. Dans la première étape, deux individus sont choisis aléatoirement dans la population. Dans la deuxième étape, en localisant d'une manière aléatoire des points sur les deux chromosomes parents, l'opération de croisement échange le contenu (informations) de ces points entre les deux chromosomes. L'information peut être une valeur d'épaisseur de la chair, un point de contrôle, un attribut d'un type de chair ou la longueur d'un segment, etc. Cette opération produit deux individus modifiés génétiquement qui font partie de la génération ultérieure. Ce sont les enfants. Le taux maximal de croisement est contrôlé par la probabilité de croisement P_c .

L'opérateur de croisement doit participer à la convergence de l'algorithme en produisant une population plus performante. Cet objectif ne peut être atteint en croisant aléatoirement les individus de la population. Le principe des algorithmes génétiques, inspiré de la biologie, consiste en effet à transmettre les caractéristiques des bons individus aux générations ultérieures. Afin de respecter ce principe, on doit favoriser le croisement des individus les plus performants. Cela peut être réalisé en gérant la probabilité de croiser deux parents Pr_1 et Pr_2 . Nous proposons dans notre approche la formule suivante :

$$Pc(Pr_1, Pr_2) = Coef_c \cdot \frac{(PopSz - mean(r_1, r_2) + 1)}{PopSz} \quad (2-26)$$

Le paramètre $Coef_c$ est une constante de l'ordre de 0.85, $PopSz$ est la taille de la population et $mean(r_1, r_2)$ la moyenne des rangs r_1 et r_2 des parents Pr_1 et Pr_2 respectivement.

La formule (2-26) constitue une contribution originale par rapport aux distributions aléatoires de probabilités utilisées classiquement. Dans cette équation, la probabilité de croiser deux parents est une fonction linéaire de la moyenne de leurs rangs. Si cette moyenne est faible, la probabilité de croisement sera forte. Cela signifie que deux parents performants, c'est-à-dire placés aux rangs les plus faibles, seront favorisés par l'opération de croisement. A l'inverse, deux parents peu performants posséderont une moyenne de rang forte et auront donc une probabilité de croisement faible. Dans une situation intermédiaire, un parent performant et l'autre pas, l'opérateur de croisement permet d'attribuer une probabilité de croisement moyenne, ce qui contribue à entretenir la diversité génétique en n'écartant pas de manière définitive le croisement entre des individus performants et d'autres moins.

La probabilité de croisement Pc calculée par la formule (2-26) est toujours dans l'intervalle $]0, 0.85[$, et cela quelques soient les rangs r_1 et r_2 . En effet, les rangs r_1 et r_2 prennent des valeurs entières comprises entre 1 et $PopSz$. Le quotient de l'équation est alors supérieur à 0 et inférieur à 1 et, par conséquent, une probabilité Pc comprise entre 0 et $Coef_c$. La valeur de $Coef_c$ a été choisie égale à 0.85, dans notre approche, car nous avons constaté que c'est la valeur la plus appropriée selon les tests numériques que nous avons effectués.

Dans les algorithmes génétiques classiques, les opérateurs de croisement sont souvent à un ou à deux points [Mic96]. Mais plusieurs études récentes ont montré que les croisements avec plusieurs points sont beaucoup plus avantageux [Kan96a]. Certaines études ont porté sur le croisement uniforme qui s'applique sur $L/2$ points en moyenne pour des chaînes de longueurs L [Spe92]. Cependant, il est difficile de prévoir le type de croisement optimal pour un problème donné. En général, le choix est basé sur des critères empiriques en observant l'effet de différents types de croisement.

Deux types de croisement ont été utilisés dans notre travail. Le premier est à deux points. Son principe est d'échanger les informations de deux chromosomes compris entre deux points de coupures. Choissant aléatoirement une ou plusieurs courbes de Bézier, la technique consiste, dans une première étape, à découper les deux chromosomes par deux

points (Fig. 2-16). L'étape suivante consiste à permuter le contenu des zones (les composants du tableau) situées entre les points de coupure.

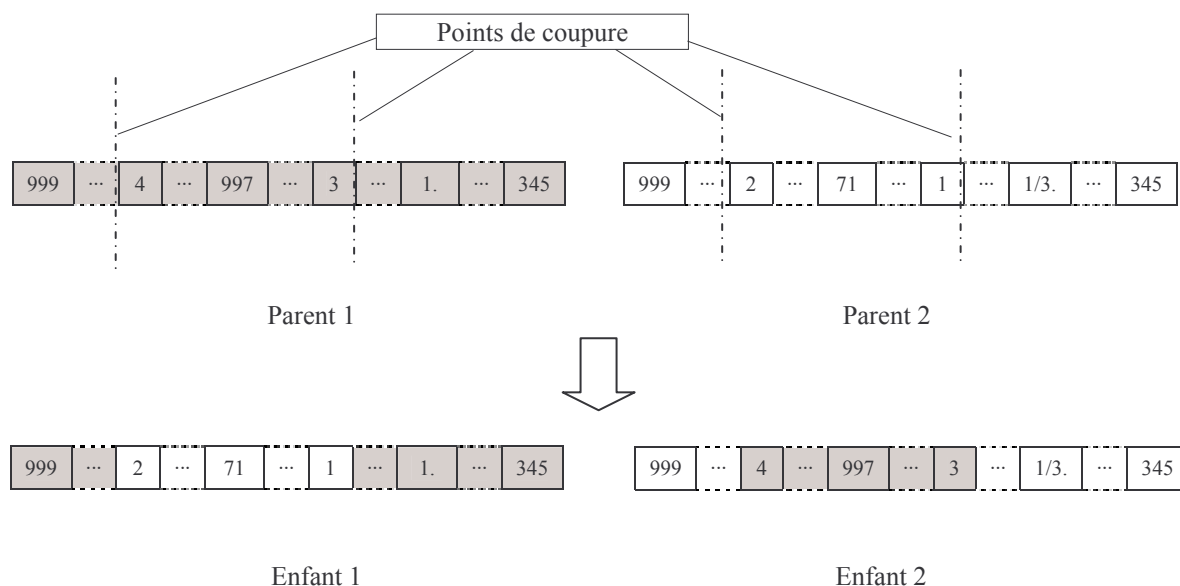


Figure 2-16. Opération de croisement (deux points de coupure).

Le deuxième type de croisement est le croisement uniforme. Nous avons calqué son principe sur ce qui est déjà pratiqué par les algorithmes génétiques à codage binaire (chapitre I : § 4.2.2.7). Ce principe consiste à choisir aléatoirement un certain nombre de positions dans le chromosome. Le contenu des gènes, correspondants à ces positions, est alors échangé entre les deux chromosomes parents.

Le premier type de croisement est utilisé au début du calcul pour la recherche du squelette optimal afin d'éviter un phénomène de dominance de certaines parties du squelette par rapport aux autres, ce qui conduit à une convergence vers un optimum local. Nous avons en effet constaté que le croisement uniforme favorisait ce phénomène lorsqu'il était utilisé au début du calcul. En revanche, il s'avère très efficace pour l'échange des informations concernant la distribution des éléments relatifs à la chair. Sur la base de ces constatations, nous avons adopté une stratégie de changement du type d'opérateur de croisement au cours du calcul. C'est une technique qui complète une stratégie adaptative adoptée dans notre travail et dont les grandes lignes sont décrites dans le paragraphe 6 de ce chapitre.

5.8.3 Mutation

Le but de la mutation est d'explorer, dans le domaine de conception, de nouvelles zones de solutions admissibles ou d'explorer l'une de ces zones dans le but de localiser l'optimum global. Le principe consiste à altérer une information contenue dans le tableau représentant le chromosome. Pour mettre en oeuvre ce principe de manière pratique, plusieurs variantes sont possibles : locale, uniforme ou classique [Kan96]. Toutes ces variantes diffèrent les unes des autres mais ont en commun d'utiliser une probabilité de mutation qui peut être utilisée de manière aléatoire ou adaptative [Mic96, Cap03]. L'objectif de ce paragraphe est de justifier les choix que nous avons réalisés dans notre approche vis-à-vis de ces différentes variantes.

Les individus de la population peuvent être classés en trois catégories : les performants, les moins performants et les intermédiaires. En conséquence, ils ne doivent pas être exposés au même type de mutation. Les plus performants représentent des optimaux locaux et une altération aléatoire des informations dont ils sont porteurs peut détériorer leurs caractéristiques. Pour cette catégorie d'individus, nous proposons une mutation de type locale. Cependant, ce type de mutation ne permet pas d'explorer tout le domaine de conception. La mutation uniforme, appliquée sur les individus les moins performants, peut être utilisée pour remplir cette tâche. Mais l'aspect aléatoire de ce dernier type de mutation ne peut garantir une diversité génétique qui représente un facteur important dans un algorithme génétique. Un troisième type de mutation est alors utilisé dans notre approche. Ce troisième type, que nous avons appelé mutation classique, a pour objectif de maintenir la diversité génétique. Il est appliqué aux individus intermédiaires situés entre les plus et les moins performants. En résumé, les individus les plus performants sont exposés à une mutation locale, les moins performants à une mutation uniforme et les "intermédiaires" à une mutation classique. Nous allons préciser maintenant comment fonctionne en détail chacun de ces trois types de mutation.

Si la mutation est uniforme, le choix du point de mutation est aléatoire. Dans ce type de mutation, tous les points ont la même probabilité de muter alors que chaque point est lié à une variable différente. L'objectif de la mutation est d'explorer de nouvelles zones. Les espaces de recherche des variables sont de tailles différentes donc l'exploration ne peut être efficace en exposant tous les points, gène dans le codage, à la même probabilité de mutation.

Un autre type de mutation appelé mutation classique est basé sur la diversité génétique de chaque gène dans le chromosome. Le principe consiste à exposer les points de faible diversité génétique à une forte probabilité de mutation. En affectant à chaque point une probabilité de sélection, le principe de la roulette est appliqué pour choisir le point de

mutation. Ce type de mutation améliore l'exploration et permet de maintenir la diversité génétique, mais il peut ralentir la convergence.

Dans ces deux types de mutation, uniforme et classique, la valeur au point de mutation est remplacée par une valeur choisie aléatoirement dans l'espace de recherche de la variable liée à ce point. L'objectif de ces deux types de mutation est de se rapprocher de l'optimum global en explorant le domaine de recherche. Mais elles permettent difficilement de converger précisément vers cet optimum lorsque des optima locaux coexistent dans son voisinage. Dans ce cas, il est plus efficace de provoquer une légère variation de la longueur d'un segment, de l'épaisseur de la chair ou de la courbure d'une courbe de Bézier puisque les différences entre les optima locaux et l'optimum global sont naturellement localisées, les grandes tendances définissant la topologie étant identiques. C'est le rôle de la mutation locale qui, de provoquer ces altérations locales permettant de se dégager des optima locaux pour converger vers l'optimum global. Ce dernier type de mutation joue aussi un rôle de lissage des frontières quand l'algorithme converge vers l'optimum global.

De manière pratique, l'opération de mutation consiste à altérer une information dans le tableau représentant le chromosome. Cette information peut être la position d'un point de départ/arrivée, la position d'un point de contrôle, une valeur d'épaisseur, l'attribut du type de la chair ou la longueur d'un segment. Un individu x_i , choisi aléatoirement dans la population, mute si sa probabilité de mutation $Pm(x_i)$ vérifie la condition [Mic96] :

$$rand \leq Pm(x_i) \quad (2-27)$$

où $rand$ est une valeur générée aléatoirement dans l'intervalle $[0, 1]$.

L'exposition de tous les individus à une probabilité de mutation affectée aléatoirement risque d'altérer les bonnes caractéristiques des individus performants. Or, ces bonnes caractéristiques doivent être transmises aux générations ultérieures par croisement. Pour éviter de les altérer par mutation, nous proposons dans notre approche une distribution linéaire de probabilité qui favorise la mutation des individus les moins performants :

$$Pm(x_i) = Coef_m \cdot \left(\frac{r_i}{PopSz} \right) \quad (2-28)$$

où $Coef_m$ est une constante de l'ordre de 0.2. Ce choix de 0.2 résulte des nombreux tests numériques que nous avons effectué afin de déterminer la valeur la plus appropriée. r_i est le rang de l'individu x_i et $PopSz$ la taille de la population.

Cette approche, exprimant la probabilité de mutation d'un individu en fonction de son rang, constitue une contribution originale par rapport aux distributions aléatoires de probabilités utilisées classiquement. Dans l'équation (2-28), les individus performants sont protégés alors que les individus les moins bons sont perturbés.

La figure 2-17 illustre une opération de mutation où le point de mutation représenté par la couleur grise dans le tableau est altéré :

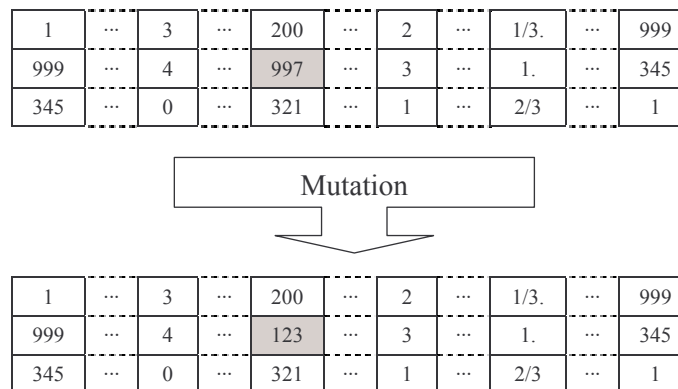


Figure 2-17. Opération de mutation.

Les taux de croisement et de mutation sont contrôlés par les probabilités P_c et P_m respectivement. Ces probabilités sont définies par les relations (2-26) et (2-28) au cours de l'évolution de la population. Néanmoins, la convergence des algorithmes génétiques étant très sensible vis à vis de ces paramètres, il est préférable de les ajuster de manière adaptative, en fonction de l'évolution de la population, plutôt que de les laisser fixes. C'est ce que nous proposons d'expliquer au paragraphe suivant.

6. Croisement et Mutation adaptatifs

Deux caractéristiques s'avèrent essentielles en optimisation de fonctions multimodales par algorithmes génétiques. La première est l'aptitude de converger vers un optimum, qu'il soit local ou global, après localisation de la zone contenant cet optimum. La deuxième est la capacité d'explorer de nouvelles zones dans l'espace des solutions admissibles afin de localiser l'optimum global. La balance entre ces deux caractéristiques est déterminée par les valeurs de probabilités P_c et P_m et le type d'opérateur utilisé.

En combinant les individus les plus performants, le croisement joue un rôle très important pour la convergence. Cependant, le croisement des solutions les plus performantes peut accélérer la convergence de l'algorithme vers l'optimum local le plus

proche. Autrement dit, quand la population devient très homogène, la sélection des individus performants devient difficile, ce qui conduit l'algorithme à une convergence prématurée vers un optimum local et à une stagnation.

L'opérateur de mutation a pour rôle d'empêcher l'algorithme de stagner. Si la population converge vers un optimum local, l'opérateur de mutation intervient avec un taux élevé, c'est-à-dire avec une probabilité P_m élevée, ce qui a pour effet de disperser une partie importante de cette population sur l'espace de recherche. Néanmoins, le choix de la valeur de P_m est critique car il induit des effets importants sur les performances de l'algorithme génétique. En effet, un taux de mutation très élevé transforme l'algorithme génétique en un algorithme de recherche purement aléatoire, alors que seulement quelques opérations de mutation pourraient suffire pour empêcher la convergence prématurée.

La recherche des valeurs optimales de P_c et P_m , afin d'améliorer les performances de l'algorithme génétique, est un problème récurrent qui a fait l'objet de nombreux travaux [Sri89, Dav89, Jak97].

L'idée principale d'un algorithme génétique adaptatif est d'adapter les probabilités des opérateurs de croisement et de mutation sur la base de statistiques concernant les performances de tous les individus de la population et ce à chaque génération. La technique adoptée dans notre travail est partiellement inspirée de la stratégie de Jakobovic [Jak97].

L'observation montre que l'allure de la population au cours des générations dépend de la performance maximale $F_{\max}$ et de la performance moyenne $\bar{F}$ de la population. La différence $F_{\max} - \bar{F}$ pour une population convergente est plus faible que la différence pour une population plus dispersée. Cette quantité joue un rôle primordial dans un pilotage adaptatif car sa valeur normalisée représente le degré de diversité de la population :

$$Dvrst = \frac{F_{\max} - \bar{F}}{F_{\max}} \quad (2-29)$$

Avant d'évaluer $Dvrst$, la performance de l'individu le moins performant doit être soustrait du calcul de $\bar{F}$ afin de trouver une valeur toujours supérieur à zéro. En pratique, et pour la plupart des applications, la valeur de $Dvrst$ est généralement comprise entre 0.05 et 0.85. Si cette quantité est faible la population converge vers un optimum local. Si elle augmente, la population est plus diversifiée. Cependant, il a été observé que la variation de la valeur de $Dvrst$ dépend du problème traité, c'est-à-dire de la fonction objectif employée [Jak97]. Cette valeur peut, dans certains cas, dépasser les valeurs des générations antérieures, ce qui pose une difficulté pour la mise en place de la stratégie adaptative.

Concrètement, pour palier à ce problème de dépassement de valeur, la variable *Dvrst* est évaluée à chaque génération et comparée avec une ancienne valeur stockée dans une variable statique notée *Val*. Si la nouvelle valeur est supérieure à l'ancienne, signe que la diversité s'est améliorée, aucune action de nature adaptative n'est entreprise. On actualise simplement la variable statique *Val* par la nouvelle valeur de *Dvrst*. En revanche, si la nouvelle valeur est inférieure à l'ancienne, ce qui est symptomatique d'un appauvrissement de la diversité, un mécanisme adaptatif est mis en œuvre afin de contrer l'appauvrissement. Pour cela, on utilise un coefficient, noté *ConvCoef*, qui représente le degré d'appauvrissement de la diversité de la population d'une génération à l'autre :

$$ConvCoef = \frac{Dvrst}{Val} \quad (2-30)$$

Ce coefficient positif, et plus petit que 1 lorsque la diversité régresse, va nous permettre d'adapter les probabilités de croisement *Pc* et de mutation *Pm* en fonction de la diversité de la population.

En ce qui concerne le croisement, la probabilité (2-26), fonction linéaire des rangs des parents, est remplacée par une dépendance non linéaire :

$$Pc(Pr_1, Pr_2) = Coef_c \cdot \left(\frac{(PopSz - mean(r_1, r_2) + 1)}{PopSz} \right)^{ConvCoef} \quad (2-31)$$

Cette dépendance non-linéaire traduit une action adaptative sur le croisement lorsque la diversité génétique régresse de manière importante avec un ratio *ConvCoef* proche de zéro. Dans ce cas, la probabilité définie par (2-31) tend vers une valeur constante *Coef_c*, imposant de ce fait une probabilité de croisement quasiment égale à tous les individus. Cela a donc pour effet de réintroduire de la diversité au sein de la population en retardant l'émergence et l'hégémonie des individus les plus performants.

Une autre manière d'interpréter la formule (2-31) consiste à remarquer que le coefficient *ConvCoef* est aussi un indicateur de convergence. En effet, lorsque ce coefficient tend vers zéro, la performance moyenne est proche de la performance maximale en vertu de (2-29). La distance entre les individus est alors faible et la proximité entre les individus les moins performants est réduite. Dans ces conditions, il apparaît naturel d'imposer une même probabilité de croisement à tous les individus, ce que reflète bien la formule (2-31) lorsqu'elle est déclinée avec un coefficient *ConvCoef* proche de zéro.

Toujours dans l'esprit de prendre en considération de manière adaptative la diversité de la population dans les opérations de croisement, le tirage des individus s'effectue en tenant

compte de la diversité dans le calcul du rang. En rappelant que les individus étaient auparavant classés par rang, où le rang 1 est attribué au meilleur individu et le rang $PopSz$ au moins bon, le calcul adaptatif du rang est maintenant réalisé à l'aide de la formule (2-32) où la diversité intervient par le biais du paramètre Pwr :

$$r_i = round((PopSz - 1) \cdot rand^{Pwr}) + 1 \quad (2-32)$$

$$Pwr = 1 + 2 \cdot ConvCoef \quad (2-33)$$

Où $PopSz$ est la taille de la population, $rand$ une valeur générée aléatoirement dans l'intervalle $[0, 1]$ et $round$ est une fonction d'arrondi.

La valeur de la variable Pwr varie de 1 à 3 lorsque $ConvCoef$ varie de 0 à 1. Plus la diversité est faible, plus Pwr est faible et plus le coefficient $rand^{Pwr}$ intervenant dans (2-32) est grand puisque $rand$ est plus petit que 1. Grâce à ce coefficient, un individu aux performances médiocres pourra plus facilement bénéficier d'un bon rang. L'introduction d'un facteur non linéaire de diversité de type puissance rend donc plus équitables les chances de tirage de chaque individu au sein d'une population homogène en diminuant la prépondérance des plus performants.

La mutation est adaptée différemment. Le nombre d'individus mutant doit varier en fonction de la diversité génétique de la population. Si le taux de diversité génétique est élevé, le nombre d'individus mutant doit baisser afin de favoriser la reproduction. L'expression de la probabilité de mutation de l'équation (2-28) prend donc la forme :

$$Pm(x_i) = Coef_m \cdot (1 - Dvrst) \cdot \left(\frac{r_i^{ConvCoef}}{PopSz} \right) \quad (2-34)$$

La probabilité de mutation s'exprime en fonction de la diversité génétique par le produit d'une partie linéaire et d'une partie non-linéaire. Il s'agit, pour la partie linéaire, d'un coefficient de pondération $1 - Dvrst$ permettant d'augmenter le nombre d'individus mutant lorsque la diversité est faible et de le diminuer si la diversité est élevée. Il s'exerce de manière adaptative au cours des générations. En convergeant, la probabilité de mutation tend vers une distribution uniforme $Coef_m / PopSz$. En effet, s'il y a convergence, la diversité $Dvrst$ et le ratio $ConvCoef$ tendent vers zéro. La probabilité de mutation, dans ce cas, ne dépend plus du rang. Chaque individu a alors la même probabilité de muter.

7. Implémentation

Sur la base des développements théoriques présentés dans les paragraphes précédents, nous avons développé un code de calcul avec le langage de programmation Matlab [Hig05]. Ce code de calcul inclut un solveur éléments finis en élasticité linéaire isotrope, avec l'hypothèse de contraintes planes, ainsi qu'un module d'optimisation basé sur les algorithmes génétiques. L'organigramme de ce module est présenté sur la figure 2-18. Par ailleurs, il est à noter que nous n'avons pas utilisé les toolbox dédiées du logiciel Matlab mais seulement le langage de programmation qu'il propose. Notre choix s'est porté sur cette plate-forme de développement en raison des nombreuses facilités de programmation qu'elle procure au stade de l'avant-projet. Cependant, en vue d'améliorer les performances en terme de temps de calcul, le portage de nos travaux en C++ fait partie des perspectives envisagées.

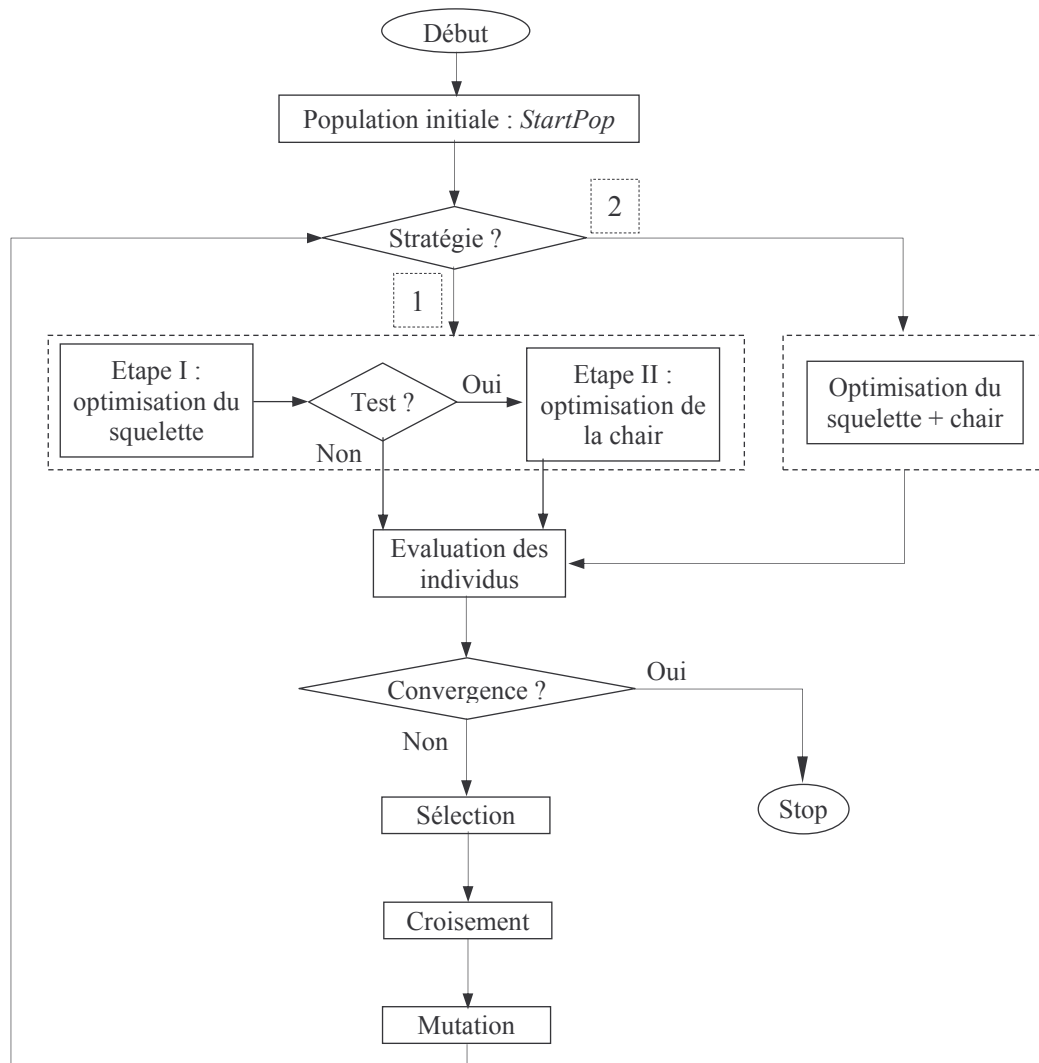


Figure 2-18. Organigramme de l'approche proposée.

Deux stratégies ont été développées dans notre algorithme. Avec la première stratégie, le calcul s'effectue en deux étapes. La première étape consiste à rechercher le squelette optimal sans faire intervenir la chair. Une fois cette première étape acquise, la seconde étape consiste à compléter le squelette par de la chair. A l'inverse, avec la deuxième stratégie, l'optimisation est réalisée en considérant simultanément le squelette et la chair.

L'intérêt de la première stratégie est de diminuer le coût du calcul en analysant par éléments finis des structures ne comportant qu'un nombre réduit d'éléments puisque la chair n'est pas représentée dans la 1^{ère} étape. D'autre part, le nombre de variables gérées par l'algorithme génétique est très réduit, ce qui a également pour conséquence de réduire le coût du calcul. En effet, dans la première étape, seules les variables contrôlant le squelette, en l'occurrence les points de contrôle, sont pris en considération. Et dans la deuxième étape, seules les variables qui contrôlent la chair, en l'occurrence le type de la chair, l'épaisseur de la chair et la longueur des segments, sont prises en compte. Cependant, l'inconvénient de cette stratégie est que la convergence vers l'optimum dépend beaucoup de la convergence du squelette. Le squelette optimal peut aussi limiter la distribution des éléments de la chair, ce qui entraîne une convergence prématurée.

Dans la deuxième stratégie, le squelette et la chair évoluent simultanément. Le problème est plus coûteux dans ce cas puisque toutes les variables du problème sont gérées à la fois et que les structures sont plus volumineuses. En revanche, l'évolution du squelette offre plus de la souplesse pour distribuer la matière par l'intermédiaire de la chair.

Par ailleurs, il est possible que des individus des générations antérieures réapparaissent dans la population actuelle. La fréquence de cette réapparition augmente avec la convergence lorsque la population devient quasiment homogène. Afin de minimiser le coût de l'analyse structurale et du décodage de ces individus, nous avons développé une méthode basée sur une copie de la population. Pratiquement, avant d'appliquer les opérateurs génétiques, une copie de la population courante est sauvegardée. Seuls les individus n'existant pas dans cette copie sont évalués dans la génération suivante.

Le test de convergence est basé sur deux critères : le nombre maximal de génération et la comparaison de la solution en cours avec la solution des générations antérieures. Nous avons fixé à 2000 le nombre maximal de génération. Pour ce qui concerne le critère de comparaison, celui-ci sera vérifié si la solution ne change pas pendant vingt-cinq générations successives.

8. Résultats et discussions

Dans ce paragraphe nous allons évaluer notre approche à l'aide d'exemples numériques. Ces exemples numériques sont des benchmarks classiques en optimisation topologique. Il s'agit du treillis de Michell [Mic1904], de la poutre cantilever [Xie97] et de la poutre MBB [Oih91].

8.1 Treillis de Michell

Cette application est un benchmark classique d'optimisation topologique des structures. Le domaine de conception est une plaque plane, de dimensions $L \times 2.4L$ (avec $L=10$ m) et d'épaisseur égale à 1 mm, encastree du coté gauche et sollicitée par une force F concentrée au milieu du côté droit (Fig. 2-19a). Le domaine, discrétisé en 25×60 éléments finis (Fig. 2-19b), est composé d'un matériau élastique isotrope avec un module de Young $E = 100$ GPa et un coefficient de poisson $\nu = 0.3$. Enfin, on considère l'hypothèse de contraintes planes.

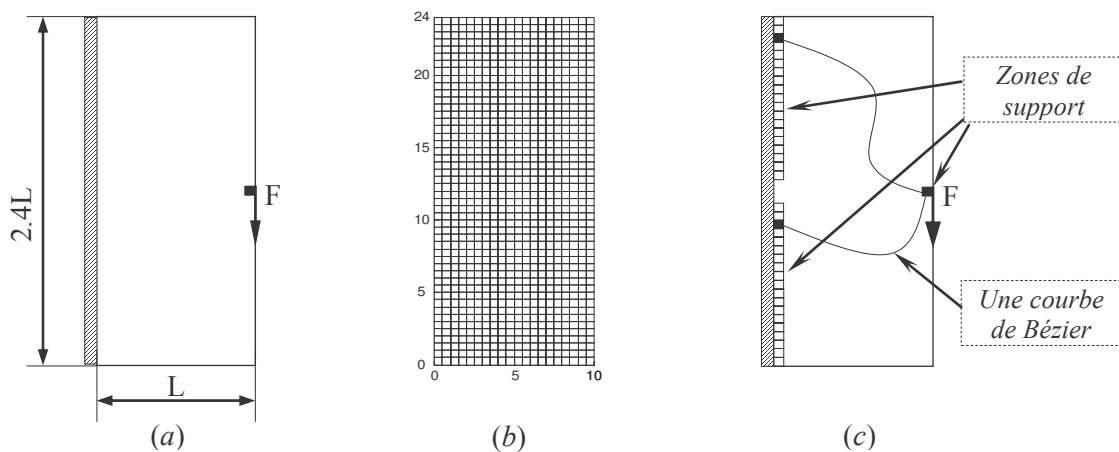


Figure 2-19. (a) Domaine de conception (b) maillage (c) schéma graphique proposé.

Le problème d'optimisation consiste à minimiser la compliance en limitant le volume à moins de 20 % du volume initial. La première étape consiste à établir un schéma graphique. Pour cet exemple, un schéma simple de deux graphes a été proposé (Fig. 2-19c). Les deux graphes, dont l'une des deux extrémités est mobile, assurent la connexion des trois zones de support définies pour ce problème. La population est composée de 60 individus initialisés aléatoirement au début de calcul.

La solution de référence proposée par Michell [Mic1904] est constituée de deux barres formant un angle droit entre elles (Fig. 2-20). Cette solution va nous permettre de valider l'approche proposée.

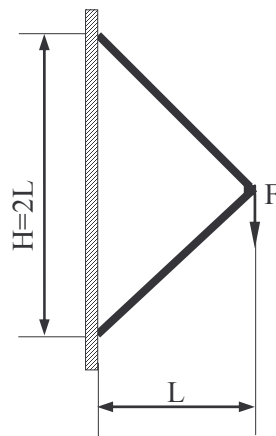


Figure 2-20. Solution de référence du treillis de Michell.

La solution trouvée après une centaine de génération (Fig. 2-21a) correspond à la solution de référence de la figure 2-20. Cette solution est également conforme à la solution obtenue par la méthode d'homogénéisation (Fig. 2-21b) [Sig01] ainsi qu'à la solution obtenue par le logiciel commercial ANSYS (Fig. 2-21c) [Law03].

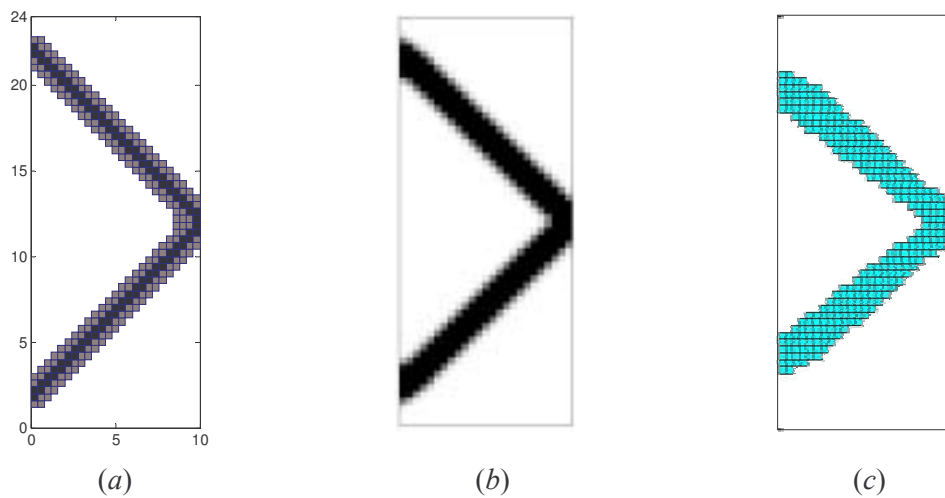


Figure 2-21. Solution par : (a) Algorithme génétique (b) homogénéisation (c) ANSYS

Les résultats numériques des trois solutions, en terme de compliance, sont donnés dans le tableau suivant :

	AG	SIMP	ANSYS
Compliance	0.0167	0.01575	0.015638
Ecart	-	5.69 %	6.36 %

Tableau 2-2. Treillis de Michell - Comparaison de la compliance.

La solution obtenue par notre approche est d'un volume égale à 19.6 % du volume initial. La limitation de 20 % que nous avons imposé sur le volume limite est donc respectée de manière satisfaisante. Par contre, nous avons constaté que la compliance de notre solution est légèrement supérieure, de 5 à 6 %, à celles fournies par Sigmund [Sig01] et ANSYS (tableau 2-2). Ceci s'explique par le fait que les solutions obtenues par [Sig01] et ANSYS résultent d'une approche par relaxation de densité. Cela signifie que tous les éléments, mêmes ceux de faible densité, sont concernés, ce qui diminue marginalement la compliance, alors que ces éléments ne sont pas pris en compte dans notre approche.

L'évolution des performances, maximale et moyenne, ainsi que l'évolution de la fonction objectif (compliance) au cours des générations sont présentées sur la figure 2-22. Cette figure montre que l'algorithme a convergé en 130 générations. Kane [Kan96], qui a résolu un problème similaire par un algorithme génétique avec un codage en tableau binaire, n'a obtenu la convergence qu'en 1500 générations.

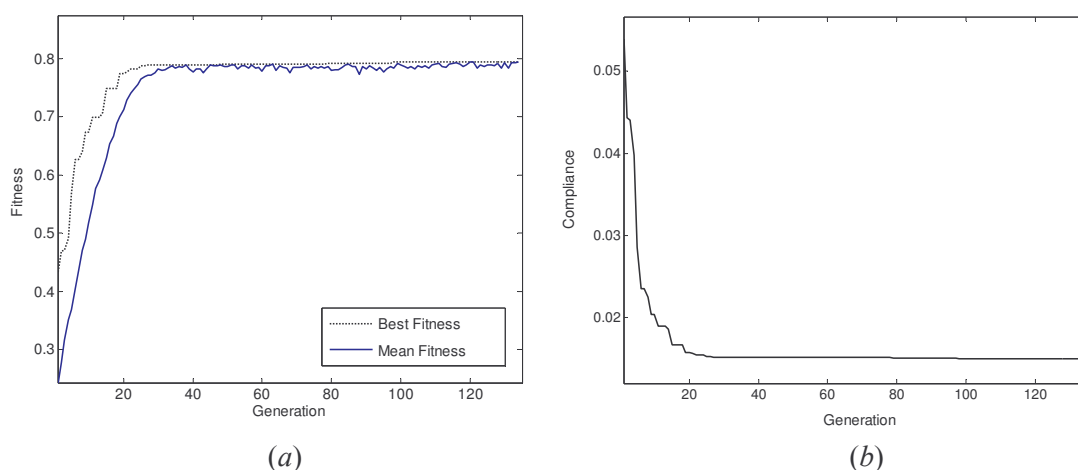


Figure 2-22. (a) évolution des performances (b) évolution de la compliance.

Influence de la courbe d'extrémité mobile

Dans ce sous-paragraphe, nous allons illustrer l'intérêt d'utiliser des courbes de Bézier à extrémité mobile. Rappelons pour cela que nous avons introduit une technique permettant la mobilité d'extrémités de courbes de Bézier sur les zones de support (§ 5.2 de ce chapitre). A l'opposé, Tai [Tai98] a fait le choix de courbes à extrémité fixe. Ce choix peut limiter la zone d'exploration et conduire à la pénalisation de la solution, comme nous allons le constater sur l'exemple du treillis de Michell traité dans ce sous-paragraphe. La figure 2-23a montre le squelette optimal obtenu avec une courbe d'extrémité fixe. On observe que ce squelette optimal conduit à une solution optimale (Fig. 2-23b) qui n'est pas

conforme à la solution de référence (Fig. 2-20). Par contre, avec un schéma de courbes mobiles (Fig. 2-23c), la solution trouvée est similaire à celle attendue.

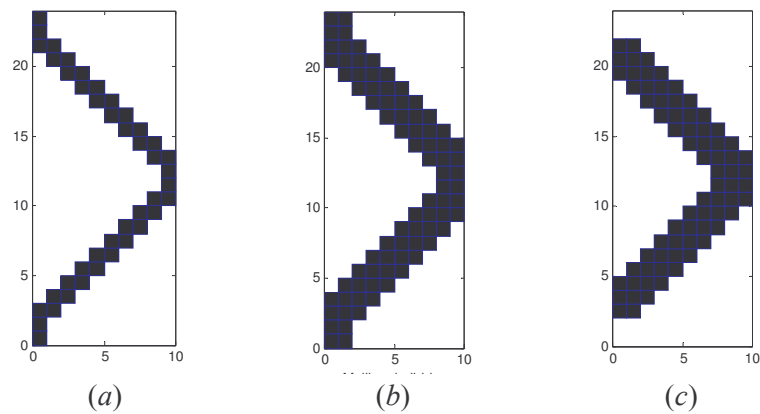


Figure 2-23. Influence de la courbe d'extrémité mobile (a) squelette optimal avec extrémités fixes (b) solution optimale avec extrémités fixes (c) solution optimale avec extrémités mobiles.

Influence du type de la chair

Le but de ce sous-paragraphe est de montrer l'importance du type de la disposition des éléments de la chair autour du squelette. Rappelons que dans notre approche, nous avons enrichi cette variable en introduisant deux nouveaux types (§ 5.2 de ce chapitre). Nous allons montrer avec un exemple simple que notre contribution sur ce point apporte une amélioration sensible. Cet exemple est toujours le treillis de Michell mais avec une limitation de 30 % sur le volume initial cette fois à la place des 20 % précédemment imposé. Les solutions obtenues avec différents types de chair sont représentées sur la figure 2-24. La première (Fig. 2-24a), est obtenue en imposant une chair de type '1' (disposition de la matière à droite de chaque courbe en considérant l'extrémité mobile comme point de départ). La deuxième (Fig. 2-24a), est obtenue avec une chair de type '2' (disposition à gauche des courbes de Bézier). La troisième (Fig. 2-24c), est obtenue en imposant la chair du type classique dont l'attribut est '3'. Enfin, la quatrième solution (Fig. 2-24d) a été obtenue en combinant tous les types de chair. On remarque que c'est cette configuration qui approche le mieux la solution optimale de référence donnée par la figure 2-20.

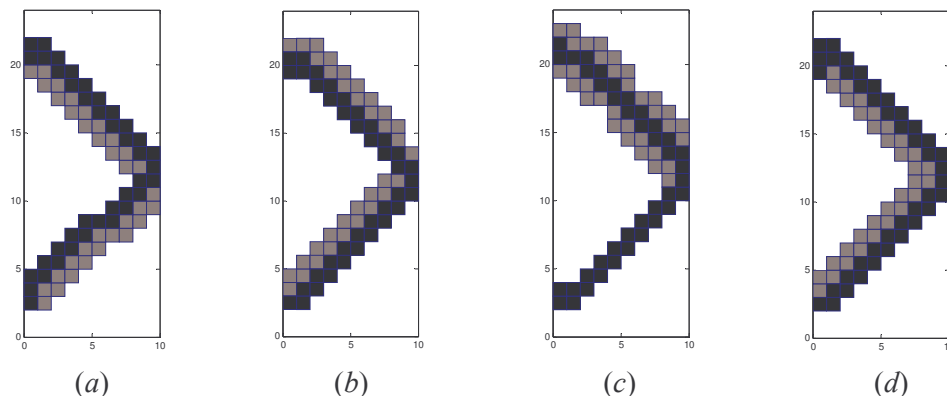


Figure 2-24. Solution : (a) avec chair de type '1', (b) avec chair de type '2', (c) avec chair de type '3', (d) de référence.

En imposant un seul type de chair, la distribution de matière n'est pas réalisée de manière idéale, toutes les solutions obtenues ((a), (b) et (c)) s'écartant de la solution de référence (d). Si on observe les compliances des différentes solutions (tableau 2-3), on remarque néanmoins une plus grande proximité des solutions (a) et (b) avec (d), la plus éloignée étant la solution (c). Cela s'explique par le fait qu'avec la solution (c), la quantité de chair disposée autour d'un segment est le double de celle utilisée dans les solutions (a) et (b), ce qui confère moins de souplesse à l'algorithme génétique.

	<i>Solution (a)</i>	<i>Solution (b)</i>	<i>Solution (c)</i>	<i>Solution (d)</i>
Volume	30 %	30 %	30 %	30 %
Compliance	1.01924×10^{-2}	1.033948×10^{-2}	1.217851×10^{-2}	1.0008×10^{-2}

Tableau 2-3. Influence du type de la chair.

Deux stratégies de calcul :

Nous présentons dans ce sous-paragraphe la résolution du treillis de Michell par les deux stratégies qui ont été présentées au paragraphe 7. Avec la première stratégie, le calcul s'effectue en deux étapes. Dans la première étape, la structure est composée seulement du squelette. Le nombre de variable se réduit alors au nombre de points de contrôle de chaque courbe. Une fois la convergence du squelette obtenue, la deuxième étape commence en activant les variables qui contrôlent la chair. Avec la deuxième stratégie, toutes les variables de la représentation sont prises en compte. La structure est composée du squelette et de la chair tout au long du calcul.

Sur la figure 2-25, nous présentons l'évolution de la meilleure solution au cours des générations pour la première stratégie.

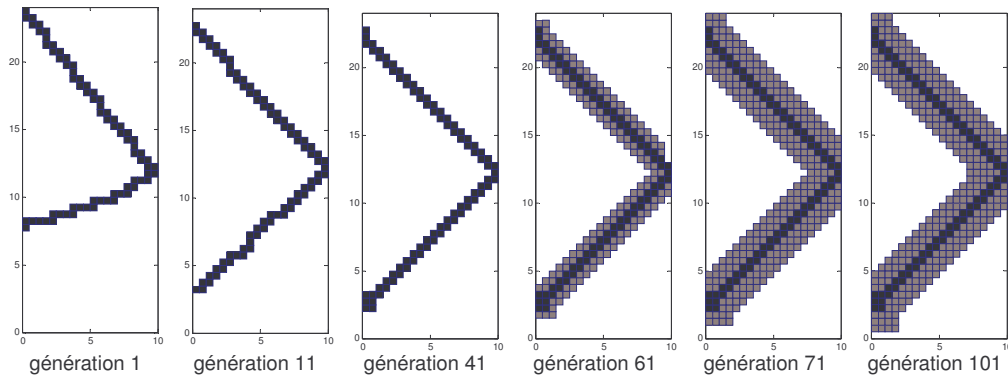


Figure 2-25. Évolution de la meilleure solution (première stratégie).

L'historique des performances maximale ainsi que l'historique du temps de calcul sont présentés sur la figure 2-26.

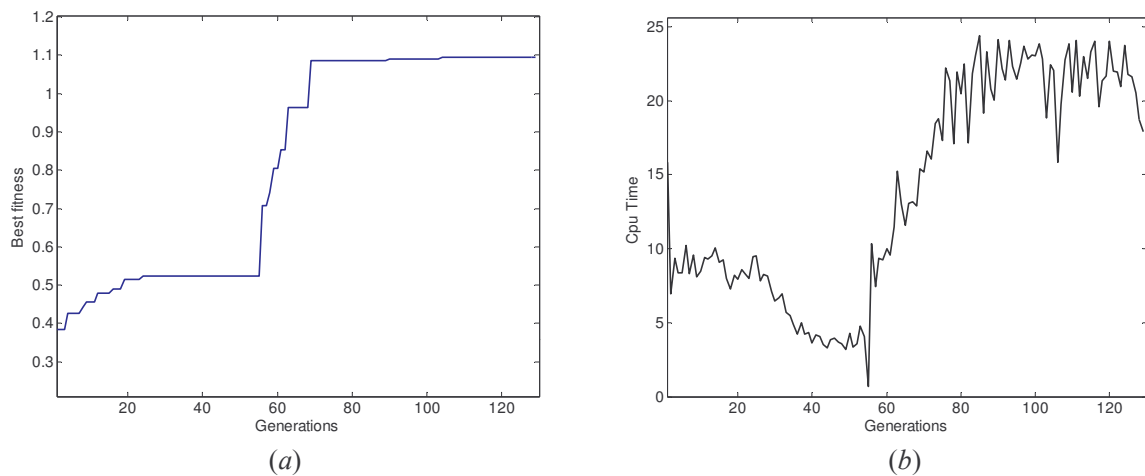


Figure 2-26. Première stratégie, historique : (a) des performances (b) du temps de calcul.

La figure 2-26 montre clairement les étapes du calcul. En effet, dans la première étape, les performances (Fig. 2-26a) ainsi que le temps de calcul sont faibles (Fig. 2-26b). Le temps de calcul continue à diminuer légèrement en convergeant vers le squelette optimal. Dès la convergence du squelette, le coût du calcul, ainsi que les performances, augmentent considérablement. Cela s'explique par l'ajout de la chair, absente dans la première étape, qui augmente notablement la taille du maillage et donc le temps de l'évaluation.

Les résultats de la deuxième stratégie sont présentés sur les figures 2-27 et 2-28. Sur la figure 2-27 nous présentons l'évolution de la solution la plus performante. On remarque que la meilleure solution à la première génération est plus prometteuse avec la deuxième stratégie (Fig. 2-27) qu'avec la première (Fig. 2-25). Néanmoins, on constate ainsi que les générations suivantes apportent rapidement les corrections nécessaires, les deux stratégies fournissent au final des solutions analogues.

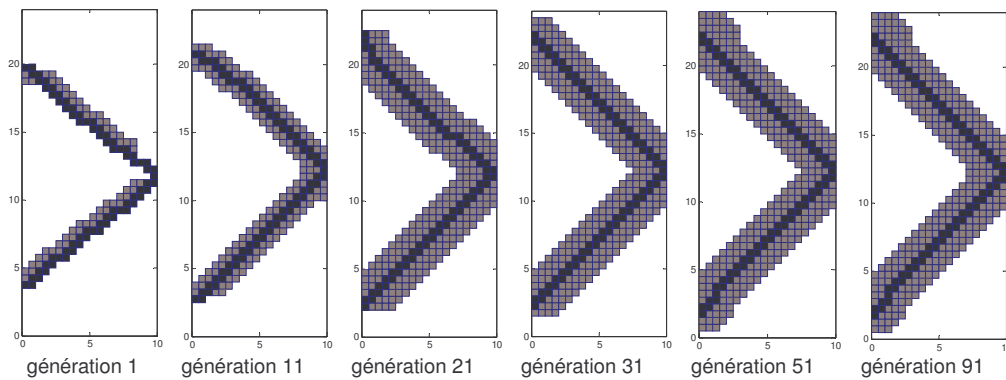


Figure 2-27. Évolution de la meilleure solution (deuxième stratégie).

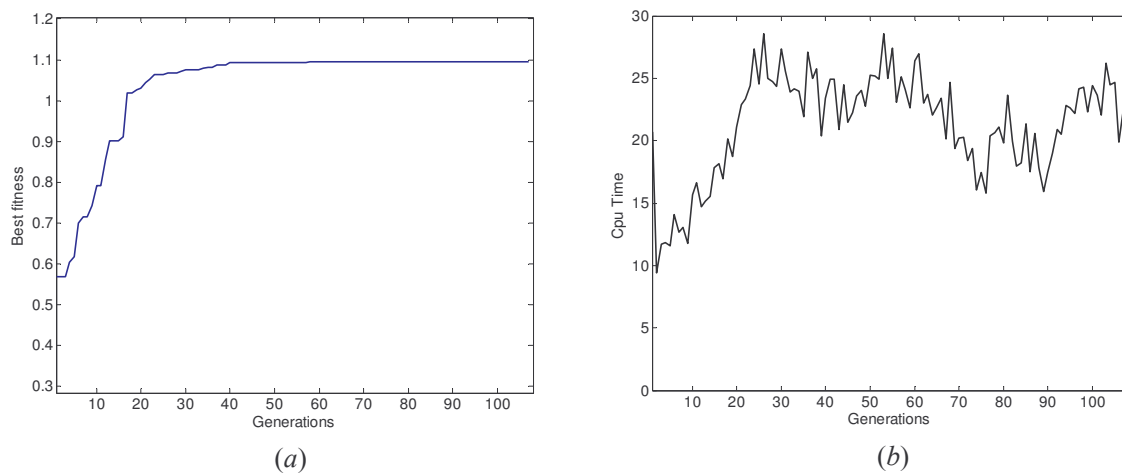


Figure 2-28. Deuxième stratégie, historique : (a) des performances (b) du temps de calcul.

A l'inverse de la première stratégie, le coût du calcul augmente d'une manière linéaire au cours des premières générations (Fig. 2-28b). Après environ 25 générations, on atteint un niveau de 25 à 30 (s) qui reste homogène par la suite. Une diminution légère a été constatée dans les dernières générations à cause de l'augmentation d'individus redondants qui ne sont pas concernés par l'analyse éléments finis.

Les deux stratégies aboutissent à des solutions similaires (solutions optimales finales). Par contre, le temps de calcul de la deuxième stratégie est plus élevé par rapport à la

première. Cela s'explique par le coût de l'analyse par éléments finis, plus important dans la deuxième stratégie puisqu'elle implique dès le départ un nombre plus élevé d'éléments. Nous préconisons donc l'utilisation de la première stratégie puisqu'elle aboutit à des solutions optimales de la même qualité que la seconde stratégie tout en étant moins consommatrice de temps de calcul.

8.2 Poutre console (cantilever)

L'objectif de ce paragraphe est double. Le premier objectif est de valider l'ensemble de notre approche à l'aide d'un benchmark classique. Le deuxième objectif est de tester plus particulièrement l'extension que nous avons proposée au sujet de la notion de zone de support, extension dans laquelle une partie du squelette peut être considérée comme zone de support (§ 5.2 de ce chapitre).

Un benchmark très populaire pour l'optimisation topologique est celui de la poutre console appelée souvent cantilever [Ben03]. Le domaine de conception est un rectangle de longueur L , de largeur H et d'épaisseur th (Fig. 2-30a). La structure est encastree sur le côté gauche et sollicitée par une force au milieu du côté droit. Le matériau est élastique linéaire et isotrope avec comme propriétés physiques : $E = 207 \text{ GPa}$ et $\nu = 0.3$. Une solution analytique de ce problème, proposée par Michell [Bec97], est donnée par la figure 2-29.

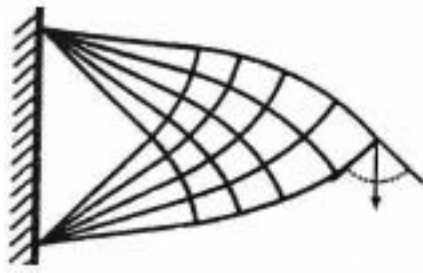


Figure 2-29. Solution analytique du problème de la poutre console. [Bec97]

Le schéma approprié pour traiter ce problème est composé de quatre graphes dont deux connectent les zones de support d'une manière directe. Les deux autres courbes sont connectées par une extrémité à une zone de support et par l'autre à une des deux premières courbes (Fig. 2-30b).

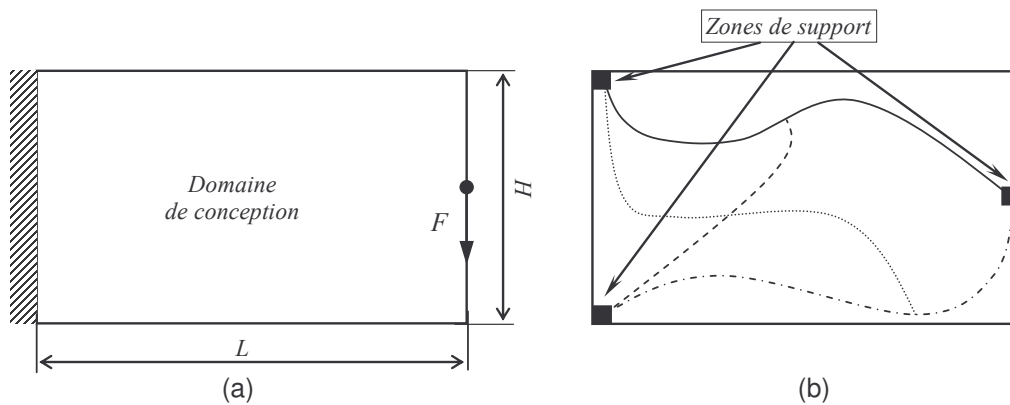


Figure 2-30. Poutre console. (a) modèle mécanique (b) schéma graphique adopté.

Le problème d'optimisation consiste à minimiser la compliance sous limitation de volume de 40 % du volume initial. Ce choix classique de la compliance a été adopté par plusieurs auteurs [All02, Ben03, Bec97, Cha94]. La topologie optimale obtenue par la méthode SIMP est présentée sur la figure 2-31 pour deux dimensions L et H différentes. La plupart des topologies optimales fournies par d'autres approches sont similaires à celle de la figure 2-31. On peut donc les considérer comme des solutions de référence. Ces deux solutions sont obtenues à l'aide du code de calcul en langage Matlab proposé par Sigmund [Sig01] et téléchargeable gratuitement sur le site internet : <http://www.topopt.dtu.dk>.

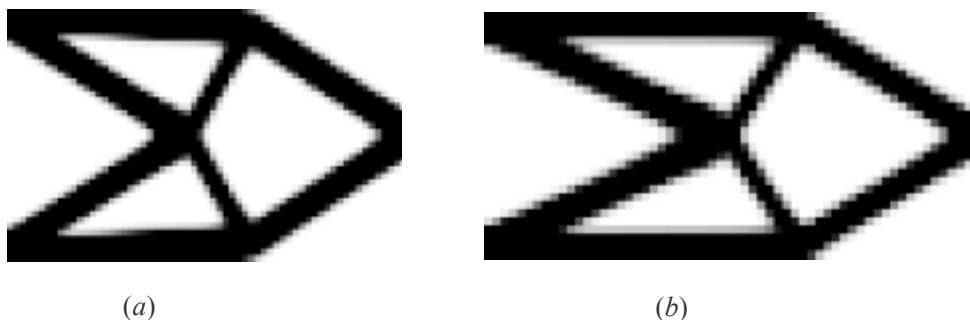


Figure 2-31. Solutions par la méthode SIMP pour les dimensions : (a) $L \times H = 1.6 \times 1$ (b) $L \times H = 2 \times 1$

Le problème a été résolu pour deux dimensions différentes de la poutre. Pour la première application, le domaine de conception de dimension $L \times H = 1.6 \times 1$ est discrétisé en 64×40 éléments finis. La population est de taille fixe et composée de 100 individus. La figure 2-32 montre les résultats de deux exécutions différentes pour une limitation de volume de 40 % du volume initial. La première stratégie (paragraphe 7), qui consiste à effectuer le calcul en deux étapes (squelette et chair séparément), a été utilisée dans les deux exécutions. Dans la première application (figures 2-32 (a) et (b) en haut) nous avons baissé le taux de croisement afin de limiter un phénomène indésirable que nous avons

constaté. Il s'agit de la dominance de certaines parties du squelette par rapport aux autres. En effet, avec une technique standard telle que celle expliquée au paragraphe 5.8.2, le résultat obtenu (Fig. 2-32b en bas) ne correspond pas à la solution optimale de la figure 2-31a.

Cet exemple illustre bien la difficulté du choix et du réglage des paramètres des algorithmes génétiques. En abaissant le taux de croisement, on a en effet obtenu une solution satisfaisante alors que les paramètres standard que nous avons initialement fixés ne le permettaient pas. Il s'agit probablement de l'un des défis majeurs que devront relever les algorithmes génétiques dans le futur que de proposer des paramètres fiables permettant la convergence sans connaissance a priori de la solution optimale recherchée.

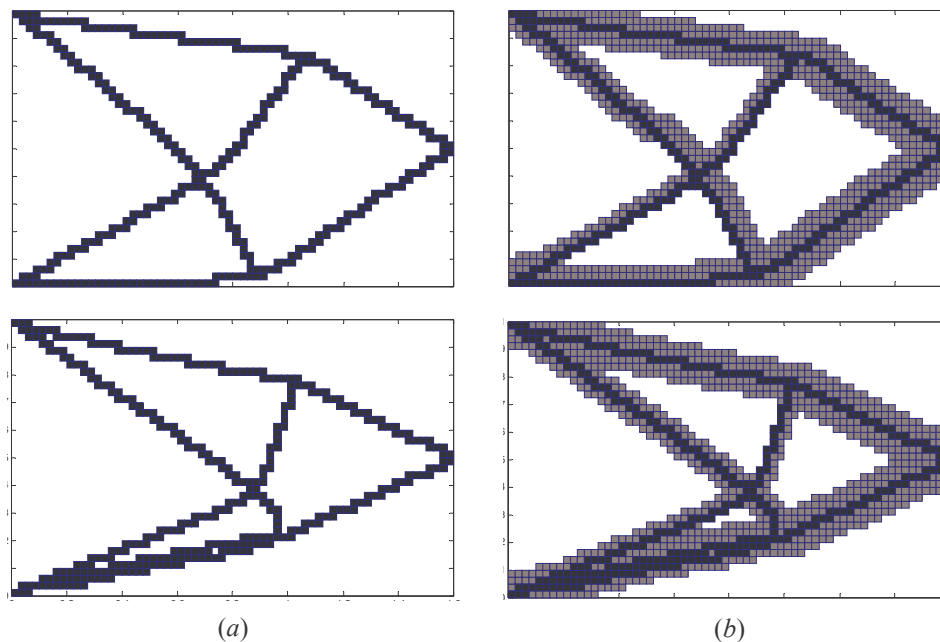


Figure 2-32. Solutions par AG (a) squelette optimal (b) structure optimale.

Le tableau 2-4 montre les valeurs de la fonction objectif (compliance) des solutions optimales obtenues par algorithme génétique ainsi que celle obtenue par la méthode SIMP (Fig. 2-31a). On remarque que, par rapport à la solution obtenue par SIMP, notre approche surévalue d'environ 5 % la compliance. Ceci s'explique par le fait que la méthode SIMP relève de la catégorie des techniques d'homogénéisation. En utilisant la relaxation des propriétés physiques du matériau, aucun élément fini n'est rejeté de la structure, contrairement à notre approche. En conséquence, la solution optimale est plus rigide. La compliance est donc plus faible.

		Solution N°1 par AG	Solution N°2 par AG	Solution par SIMP
Compliance	Squelette	10.04863	9.825672	-
	Structure	2.194928	2.188824	2.0973

Tableau 2-4. Poutre console (résultats numériques - application 1).

Une deuxième application pour le même problème a été réalisée en prenant les dimensions : $L \times H = 2 \times 1$. Le domaine est discrétisé en 60×30 éléments finis. Les solutions sont présentées sur la figure 2-33. Comme pour la première application présentée précédemment, la première solution a été obtenue en abaissant le taux de croisement.

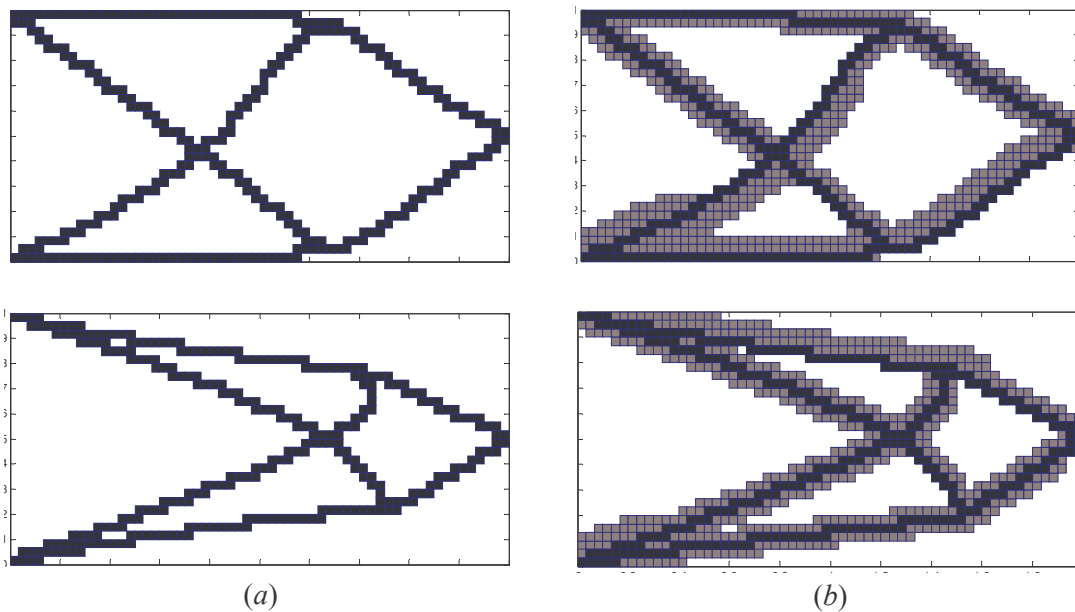


Figure 2-33. Solutions par AG (poutre console) (a) squelette optimal (b) structure optimale.

Les résultats numériques concernant la fonction objectif sont montrés dans le tableau 2-5. On remarque une surévaluation d'environ 7 % de la compliance fournie par notre approche par rapport à la méthode SIMP. Une partie de l'explication est similaire à celle que nous avons fournie pour la 1^{ère} application, à savoir la conservation des éléments de faible densité avec la méthode SIMP, ce qui a pour effet de diminuer artificiellement la compliance. Mais, pour cette seconde application, si on compare les figures 2-33b avec la figure 2-31b, on constate que les algorithmes génétiques fournissent une solution optimale qui ne respecte pas la symétrie, et ce bien que le squelette optimal soit satisfaisant. Cela

signifie que la chair n'a pas été correctement distribuée malgré les améliorations que nous avons apportées sur ce point par rapport à la méthode de Tai. Cette non symétrie introduit certainement un biais dans le calcul de la compliance, ce qui contribue à creuser l'écart entre notre approche et la méthode SIMP. On peut donc raisonnablement penser qu'en introduisant encore plus de souplesse dans la gestion de la chair, par des techniques qui restent à définir, l'écart pourrait diminuer.

		Solution N°1 par AG	Solution N°2 par AG	Solution par SIMP
Compliance	Squelette	12.53976	13.15292	-
	Structure	3.792897	3.915260	3.5502

Tableau 2-5. Poutre console (résultats numériques - application 2).

A l'examen des figures 2-32 d'une part et 2-33 d'autre part, il est clair que le squelette influence fortement le résultat. Grâce aux tableaux 2-4 et 2-5, on constate même un lien direct entre la compliance finale et la compliance du squelette optimal. En effet, pour la première application (tableau 2-4), le ratio entre ces deux compliances est quasiment invariant :

$$\text{Solution N°1 : } \frac{C_s}{C_f} = \frac{10.04863}{2.194928} = 4.578113724 \quad (2-35)$$

$$\text{Solution N°2 : } \frac{C_s}{C_f} = \frac{9.825672}{2.188824} = 4.489018761 \quad (2-36)$$

Où C_s représente la compliance du squelette et C_f la compliance de la structure finale.

L'écart relatif entre ces deux ratios n'est en effet que de 1.95 %. On obtient une conclusion analogue avec la deuxième application (tableau 2-5) pour laquelle l'écart relatif entre les deux ratios est de 1.59 %.

$$\text{Solution N°1 : } \frac{C_s}{C_f} = \frac{12.53976}{3.792897} = 3.30611667 \quad (2-37)$$

$$\text{Solution N°2 : } \frac{C_s}{C_f} = \frac{13.15292}{3.91526} = 3.359398865 \quad (2-38)$$

Le lien que nous avons établi entre la compliance du squelette et la compliance finale montre que, pour converger vers la meilleure topologie finale, il est nécessaire d'obtenir préalablement le meilleur squelette possible. Nous avons pu obtenir ce squelette optimal en

abaissant le taux de croisement. Mais il s'agit d'un pis-aller car, avec un taux de croisement faible, l'efficacité d'exploration diminue et la solution dépend fortement de la première population. Or, la diversité de la première population, générée aléatoirement au début de calcul, dépend de la taille de cette dernière et de la complexité du problème. Dans la partie qui suit, nous allons introduire la notion de squelette renforcé qui permet l'obtention du squelette optimal sans modification arbitraire du taux de croisement.

Influence du schéma

Le squelette est composé de plusieurs portions dont chacune correspond à une courbe de Bézier dans le schéma proposé. C'est la combinaison de toutes ces portions qui donne le squelette optimal.

Le schéma de la figure 2-30b est composé de quatre courbes dont deux connectent deux zones de support entre elles d'une manière directe. Ces deux courbes constituent la partie du squelette sollicitée par la force appliquée. Les deux autres courbes ont pour rôle de soutenir les deux premières. Elles peuvent être qualifiées de secondaire puisqu'elles représentent la partie du squelette qui n'est pas sollicitée. En conséquence, les deux premières courbes peuvent dominer les deux autres en définissant la topologie finale de la structure et l'algorithme converge vers un optimum local. Une illustration en est donnée par la figure 2-34 qui montre une solution où la topologie est influencée par les deux courbes principales. La solution converge vers la bonne topologie pendant les premières dizaines de génération puis l'une des deux courbes secondaires tend à se superposer sur une courbe principale ce qui conduit à la convergence vers un optimum local.

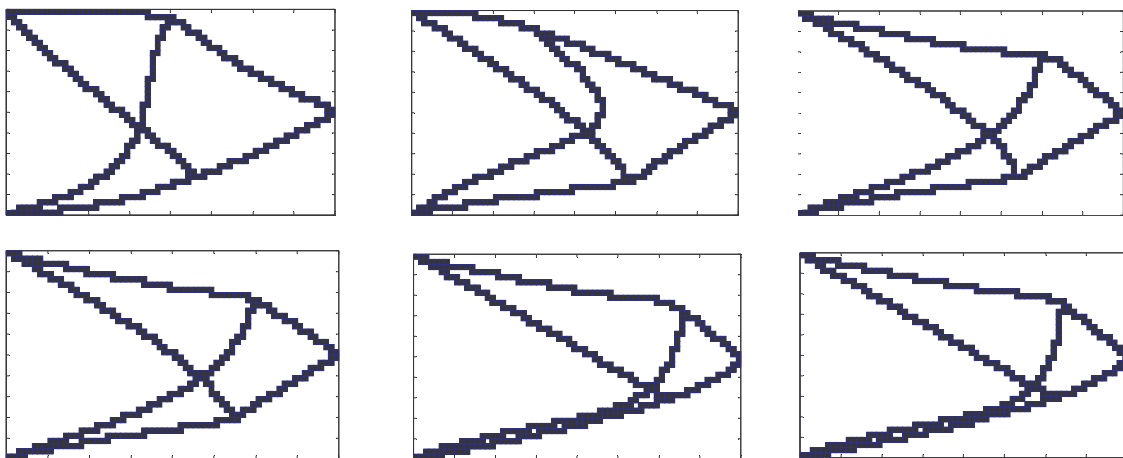


Figure 2-34. Squelette nu - évolution au cours des générations.

Afin de tester l'influence des parties secondaires du squelette nous avons initialisé la première population avec une chair uniforme de type '1' et d'épaisseur égale à 1. L'historique de convergence des squelettes montre que les parties secondaires dans le squelette ont la même influence que les parties principales (Fig. 2-35) et le phénomène de superposition disparaît.

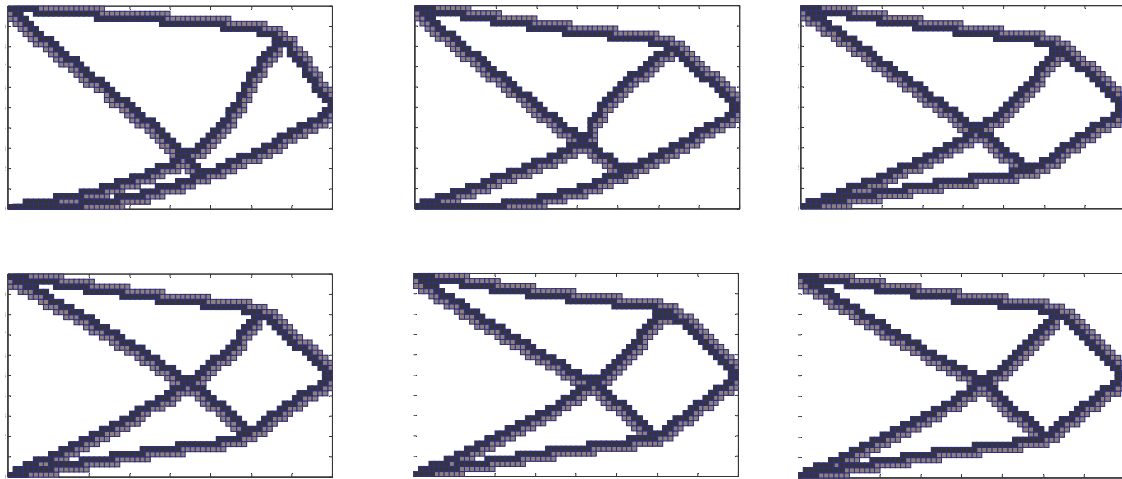


Figure 2-35. Squelette renforcé – évolution au cours des générations.

Nous proposons donc de renforcer le squelette par une couche de chair, ce qui permet de maintenir un équilibre entre les différentes portions de ce squelette. En effet, lorsqu'on travaille avec un squelette nu, les portions sollicitées directement par la force risquent de dominer les autres, ce qui peut conduire à une convergence prématurée vers un optimum local.

8.3 Poutre MBB

Un benchmark classique d'optimisation topologique de structures est celui de la poutre MBB (Messerschmitt-Bölkow-Blohm) dont le modèle mécanique est présenté sur la figure 2-36. Le problème consiste à minimiser la masse d'une poutre de support dans un avion de transport civil de type Airbus. La solution optimale recherchée doit respecter certaines limitations de conception, notamment le nombre et les dimensions des trous qui doivent faciliter l'installation des câbles hydrauliques et électriques. La déflexion maximale au centre de la poutre ne doit pas excéder 9.4 mm pour une charge de 20 kN appliquée au milieu du côté haut de la poutre. Le domaine de conception est de dimension $L \times H = 2.4 \times 0.4$ (m²) et d'épaisseur égale à 4 mm. Le matériau est élastique isotrope de caractéristiques physiques : $E = 70$ Gpa et $\nu = 0.3$.

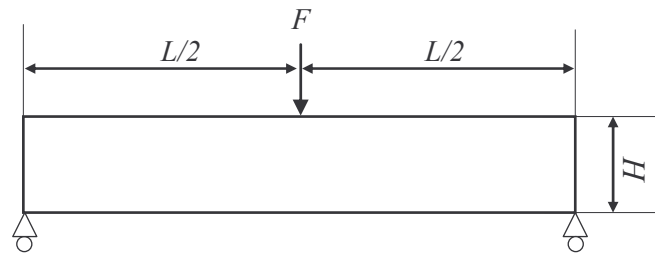


Figure 2-36. Modèle mécanique de la poutre MBB.

A l'origine, le problème a été résolu par des techniques d'optimisation de forme (Fig. 2-37) [Olh91]. Dans ce problème, d'optimisation de forme, seuls les contours des perforations étaient optimisés et il n'était pas possible d'en modifier le nombre. Le modèle avec trois trous circulaires (Fig. 2-37a) est d'une déflexion maximale égale à 10.1 mm. La solution obtenue par optimisation de forme (Fig. 2-37b), en utilisant l'interpolation *B-spline*, n'apporte qu'un gain de 5.2 % de la masse pour une déflexion maximale égale à 9.4 mm. Pour améliorer sensiblement la solution, une nouvelle topologie est nécessaire.

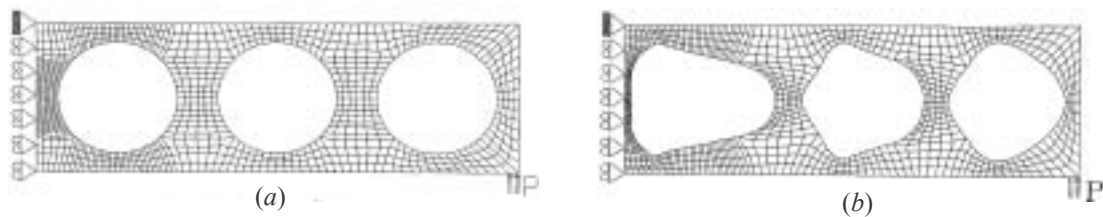


Figure 2-37. (a) Enoncé du problème (b) et solution par optimisation de forme [Olh91].

Compte tenu de ces limitations, les auteurs ont procédé à une optimisation topologique par homogénéisation en utilisant un modèle avec microstructures. Le problème d'optimisation a consisté à minimiser le volume sous une limitation sur le déplacement maximal. La solution obtenue (Fig. 2-38a) ne peut être utilisée comme base pour un modèle CAO de la structure à réaliser. Olhoff et al. [Olh91] proposent alors une conception finale en interprétant les zones vides et pleines suggérées par la solution de l'optimisation topologique puis en optimisant la forme (Fig. 2-38b). Le volume a été réduit de 42 % par rapport à la première solution (Fig. 2-37) pour une déflexion maximale de 9.4 mm.

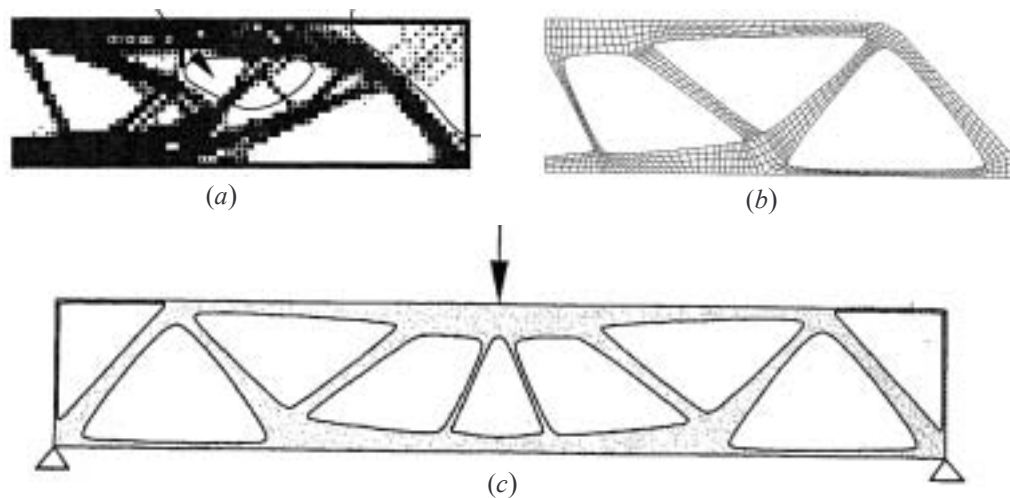


Figure 2-38. Solutions proposées par Olhof et al. [Olh91] : (a) par optimisation topologique (b) solution finale (demi poutre) (c) solution finale (poutre complète).

Zhou & Rozvany [Zho91] ont proposé une solution théorique composée d'un réseau de courbes orthogonales (Fig. 2-39). Cette solution ne peut être retenue à cause des trous dont les dimensions et le nombre ne remplissent pas le cahier de charge proposé.

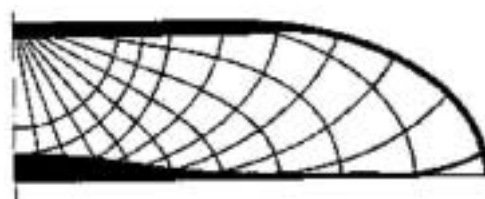


Figure 2-39. Solution théorique proposée par Zhou et Rozvany [Zho91].

Le problème de la poutre MBB a également été résolu par des méthodes d'optimisation topologique en variable continues [All02, Ben03, Duy96] et discrètes [Bec97]. Le point commun entre ces solutions est la distribution de la matière qui est conforme avec la solution théorique de Zhou et Rozvany (Fig. 2-39). Un exemple de résolution en variables continues par la méthode SIMP [Sig01] est présenté sur la figure 2-40. Mais on voit sur la figure 2-40a que le nombre de perforations pose un problème. Ce problème peut être résolu en utilisant la méthode SIMP avec filtre. La solution est alors améliorée (Fig. 2-40b). Cependant, toutes les méthodes que nous venons de présenter ont été confrontées au problème de damier (voir chapitre I : § 4.1.4). C'est l'un des avantages de la méthode que nous proposons que d'être exemptée de ce problème.



Figure 2-40. Problème de la poutre MBB - Solutions par la méthode SIMP (a) sans filtre (b) avec filtre.

Pour réaliser une application numérique avec notre méthode, nous avons discrétisé le domaine de conception en 150×25 élément finis. Pour établir le schéma graphique, qui doit satisfaire les conditions de conception, une analyse préliminaire par éléments finis a été réalisée. Cette analyse fait apparaître trois zones de concentration de contraintes mécaniques au niveau des surfaces d'application des conditions aux limites (Fig. 2-41). Ces zones de concentration feront donc partie des zones de support des courbes de Bézier (Fig. 2-7, § 5.2). Par ailleurs, le nombre de courbes de Bézier doit être limité afin de contrôler le nombre de perforations. Enfin, la poutre étant large, la solution optimale ne peut être obtenue si on impose uniquement des connexions directes entre les zones de support. Sur la base de ces arguments, nous proposons un schéma constitué de huit courbes dont quatre seulement assurent des connexions directes. Les autres courbes assurent des connexions indirectes, en se servant des premières courbes comme zones de support (Fig. 2-42).

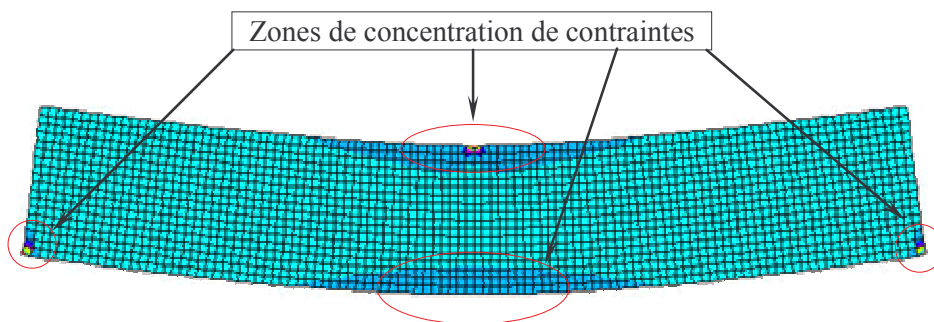


Figure 2-41. Distribution des contraintes de Von Mises sur la déformé de poutre avant optimisation.

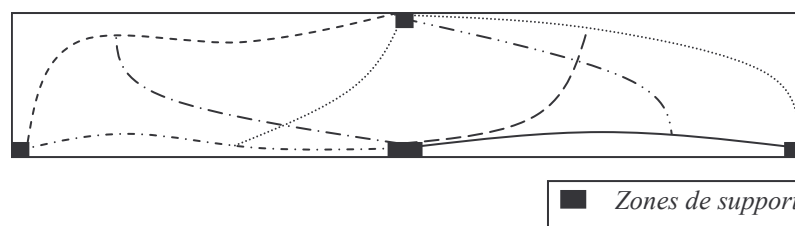


Figure 2-42. Problème de la poutre MBB - schéma proposé pour la résolution par AG.

Compte tenu de la symétrie du modèle mécanique, nous n'avons pris en considération que la moitié du domaine (Fig. 2-43a). Les dimensions se réduisent à 1.2×0.4 (m²) et le maillage est réduit de moitié avec 75×25 éléments finis. Nous avons posé le problème d'optimisation de deux manières différentes. Dans la première, nous recherchons la solution la plus légère avec une limitation sur la déflexion maximale. Dans la deuxième, la solution recherchée est la plus raide possible en posant une limitation sur le volume.

La population, dans une première application, est composée de soixante individus générés aléatoirement au début du calcul. Le problème d'optimisation consiste à minimiser le volume en se limitant à un déplacement maximal égale à 8.5 mm sachant que le cahier de charge impose 9.4 mm comme limite. Nous avons choisi cette valeur pour tenir compte des erreurs numériques possibles provenant de l'analyse structurale. La source principale de ces erreurs vient du modèle éléments finis proposé (type d'éléments, taille du maillage,...). C'est une simple mesure de sécurité traduite par la multiplication de la limitation imposée (9.4 mm) par un coefficient de sécurité d'environ 0.9. La recherche de la solution optimale est réalisée en deux étapes, la première concernant le squelette exclusivement et la seconde impliquant la chair avec le squelette optimal trouvé dans la 1^{ère} étape (voir § 7). La figure 2-43 montre la solution obtenue après 272 générations. La solution optimale que nous avons trouvée présente un volume égale à 49,86 % du volume initial de la structure pleine et une déflexion égale à 8.467837 mm, ce qui est très proche de la limitation imposée. De plus, la réduction du volume est de 51.14 % du volume initial total, ce qui constitue un excellent résultat. De plus on remarque que l'allure générale de la distribution de la matière est très proche de la solution théorique (Fig. 2-39) et de celle fournie par l'approche SIMP (Fig. 2-40b).

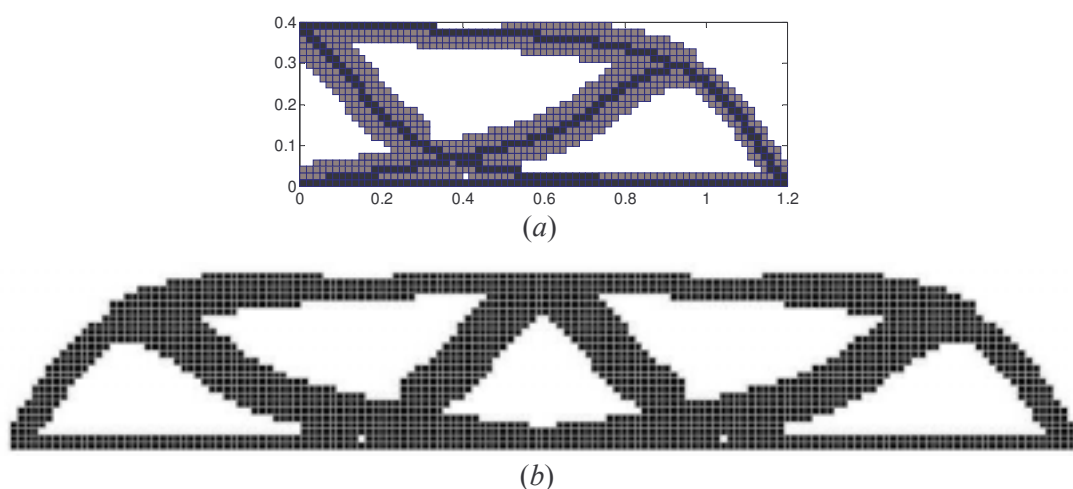


Figure 2-43. Solution optimale de la poutre MBB par algorithme génétique (application N°1)
(a) demi poutre, (b) poutre complète.

Afin de tester l'influence de la taille de la population, nous avons fait une autre application avec une population composée de 75 individus. La limitation dans cette application est égale à 9 mm. Dans le même esprit de l'application précédente, nous appliquons maintenant un coefficient de sécurité de 0.95. La solution obtenue après 353 générations est présentée sur la figure 2-44. Le volume de cette solution est égale à 42,98 % du volume initial et la déflexion maximale est égale à 8.982748 mm. On constate un gain de volume d'environ 7 % contre une surélévation de la déflexion d'environ 0.5 mm par rapport à la solution précédente, ce qui est logique puisque nous avons été moins contraignant sur la limitation imposée à la déflexion maximale. Le temps de calcul, d'autre part, est plus élevé dans cette dernière application. Là aussi, le constat est logique puisque, par rapport à l'application précédente, nous avons augmenté la taille de la population de 15 individus. En revanche, comme nous avons été moins contraignant sur la limitation, les structures sont moins volumineuses, ce qui rend leur analyse par éléments finis moins coûteux. Rappelons à ce sujet que, dans notre approche, on procède à l'ajout progressif de la matière.

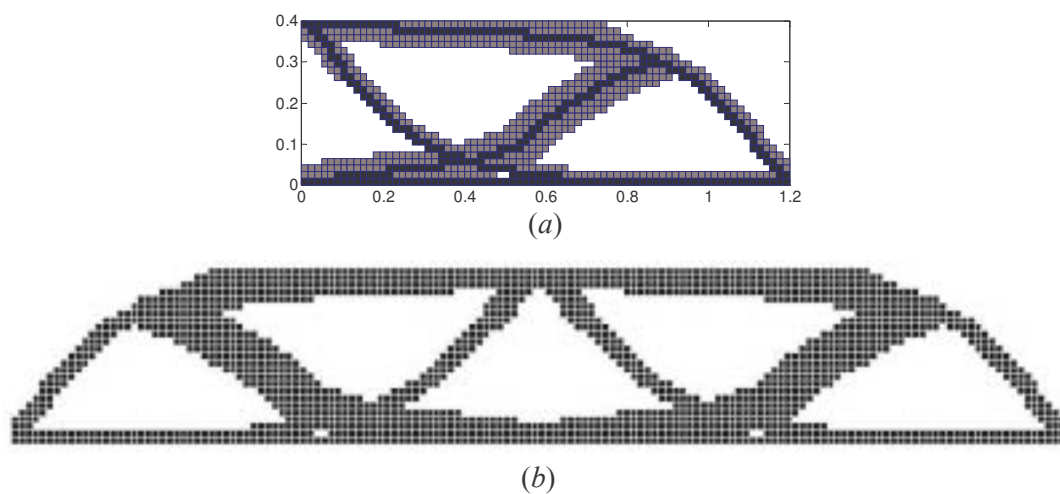


Figure 2-44. Solution optimale de la poutre MBB par algorithme génétique (application N°2)
(a) demi poutre, (b) poutre complète.

Nous nous intéressons maintenant à la deuxième façon de poser le problème de l'optimisation topologique de la poutre MBB en minimisant la compliance sous limitation sur le volume. Il est classique de fixer cette limitation à 50 % du volume initial [Duy96, Bec97]. Par contre, nous avons remplacé la compliance par le déplacement maximal comme fonction objectif. En effet, bien qu'il représente une grandeur non différentiable, le déplacement maximal est une donnée plus significative que la compliance de point de vue de l'ingénieur. La fonction performance que nous avons utilisée est donc définie par

l'équation (2-18). Notre solution la plus raide sous limitation d'un volume maximal de 50 % est représentée sur la figure 2-45. La solution est obtenue après 350 générations avec une population composée de 60 individus. Le déplacement maximal est égal à 8.50567 mm et le volume est égale à 49.92 % du volume initial. On observe que la topologie de cette solution est légèrement différente de celles obtenues lorsque la fonction objectif portait sur le volume (figures 2-43b et 2-44b). Ce résultat est bien entendu logique puisque les deux problèmes ne sont pas formulés de la même manière. Pour le premier problème, le volume est considéré comme la fonction objectif, la limitation portent sur la déflexion maximale. Pour le second problème, c'est l'inverse. Ce que montre en tout état de cause la figure 2-45, c'est que le schéma graphique que nous avons proposé, bien qu'il ne soit composé que de quatre courbes, est capable de traiter des situations très diverses en générant une grande variété de topologie.

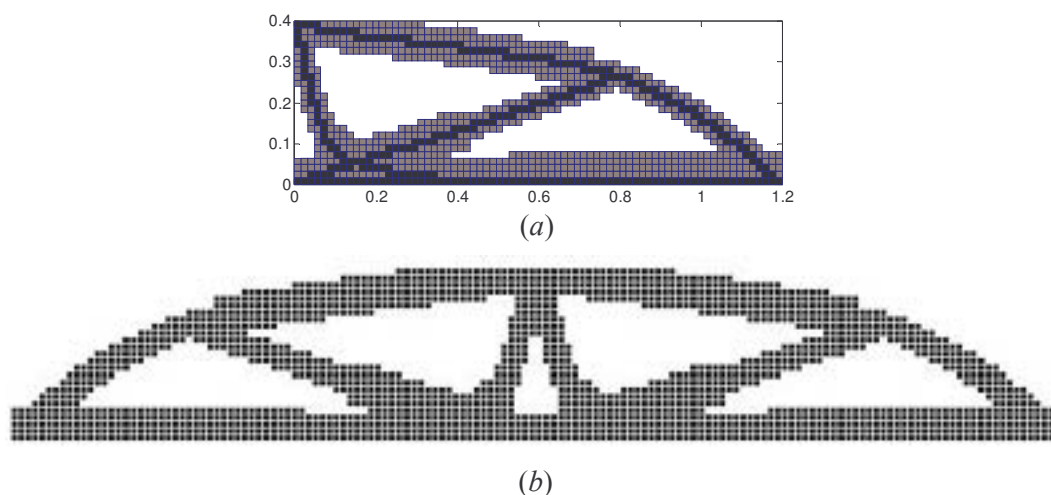


Figure 2-45. Solution optimale de la poutre MBB par algorithme génétique (application N°3)
(a) demi poutre, (b) poutre complète.

Pour ce dernier exemple, nous avons fixé un nombre de 40 générations successives comme critère d'arrêt. L'historique de convergence est présenté sur la figure 2-46. Sur la figure 2-46a nous avons présenté l'évolution de la fonction objectif. Sur cette figure on observe deux zones distinctes. La première, qui représente la première étape du calcul correspondant à la recherche du squelette optimal, s'étend sur environ 150 générations. Cette évolution du squelette est clairement visible sur la figure 2-46b, où le volume est très faible. Dès la convergence du squelette, la deuxième étape commence en ajoutant des éléments de la chair. Cette deuxième étape est visible sur la figure 2-46b où le volume augmente de la 150^{ème} à la 300^{ème} génération environ pour se stabiliser sur une valeur

proche de la limitation imposée. En parallèle de l'évolution du volume, le déplacement maximal (Fig. 2-46a) continue à baisser puis se stabilise à 8.50567 mm.

Par ailleurs, en procédant à des tests numériques, nous avons constaté que la compliance (Fig. 2-47) suit une évolution identique à celle du déplacement maximal, représentée sur la figure 2-46a. Ce constat est logique puisque la déflexion maximale se situe au point d'application de la force. Comme la compliance est par définition le produit de cette force par la déflexion du point d'application, on voit que, sur cet exemple de la poutre MBB, compliance et déflexion maximale sont proportionnelles, le coefficient de proportionnalité étant donné par la force. Cela signifie que l'optimisation de la poutre en minimisant la compliance équivaut à la minimisation du déplacement maximal.

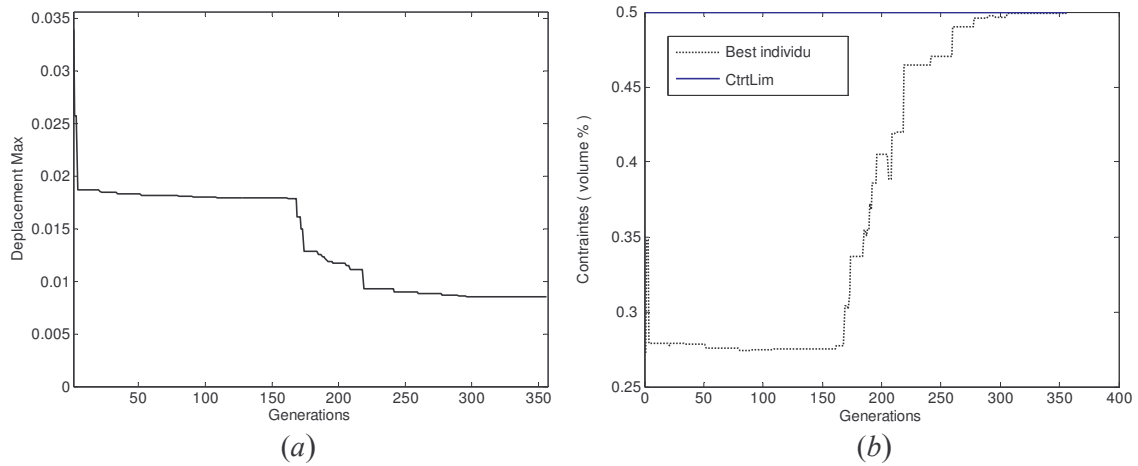


Figure 2-46. Évolution : (a) de la fonction objectif (δ_{Max}), (b) des limitations (volume).

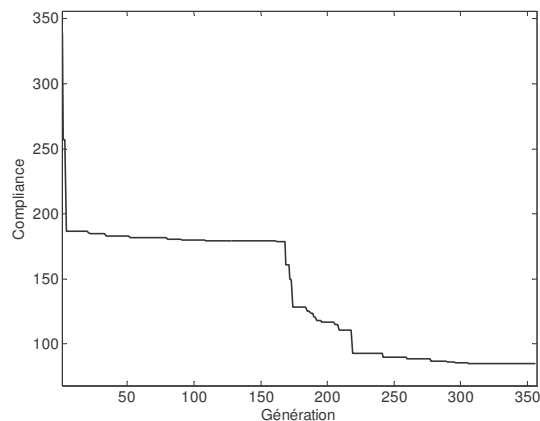


Figure 2-47. Évolution de la compliance.

En conclusion de ce paragraphe, on peut noter que les solutions trouées avec notre approche sont cohérentes avec les solutions de référence connues dans la littérature. De plus, nos solutions remplissent parfaitement le cahier de charge, notamment en terme de déflexion maximale et de nombre de perforations imposées. Enfin, elles sont exemptées des problèmes de damier. Elles ont donc pu être conçues sans qu'il soit nécessaire de procéder à une optimisation de forme.

9. Conclusions

Une approche d'optimisation topologique de structure par algorithme génétique basée sur une nouvelle représentation a été présentée dans ce chapitre. Le principe de l'approche est de considérer la structure comme composée de deux entités géométriques : le squelette et la chair. Le squelette est basé sur un schéma graphique composé d'un certain nombre de courbes de Bézier de degré trois. Chaque courbe connecte d'une manière directe ou indirecte deux zones prédéfinies sur le domaine de conception. Ces zones sont appelées zones de support. Les éléments finis connectés entre eux par leurs côtés et disposés le long de l'abscisse curviligne des courbes forment le squelette. La chair, composée des éléments finis enrobant le squelette, complète la structure sans générer de damiers.

Outre l'avantage de la connectivité des structures générées, notre approche permet un gain notable du temps de calcul. La recherche de la solution optimale peut en effet être réalisée en deux étapes. Dans la première, l'optimisation porte sur le squelette seulement. Lorsque le squelette optimal est obtenu, la deuxième étape commence avec pour objectif de trouver la distribution optimale de la chair. Sur ce point, notre approche, qui procède à l'ajout progressif de la matière, se démarque des méthodes d'optimisation topologique qui procèdent souvent à l'élimination de la matière pour la recherche de la solution optimale.

Les résultats obtenus avec l'approche proposée ont été validés et comparés avec ceux de la méthode déterministe d'homogénéisation SIMP. La comparaison a montré un écart faible sur les compliances. Cet écart s'explique par la présence de tous les éléments finis dans les structures optimales trouvées par la méthode SIMP. La présence de tous les éléments, et en particulier ceux de faible densité, conduit en effet à une légère surestimation de la compliance.

Parmi tous les exemples que nous avons traité, le problème de la poutre MBB illustre clairement les avantages de l'optimisation topologique par rapport à l'optimisation de forme en montrant qu'elle conduit à des topologies originales. Les solutions que nous avons obtenues satisfont le cahier des charges et notamment le nombre de perforations souhaitées. Cependant, ce nombre est contrôlé par le choix du schéma graphique, choix qui, dans notre approche, reste pour une large part intuitif. Il n'est pas toujours facile de trouver le plus adapté au problème traité, en particulier dans le cas de géométries complexes. Aujourd'hui, il s'agit probablement de la principale limitation de la méthode que nous proposons. C'est pourquoi le chapitre III de ce mémoire s'attache à développer une méthode, toujours basée sur les algorithmes génétiques, mais qui a pour ambition de s'affranchir des schémas graphiques.

Chapitre III

Représentation par adjacence

1. Introduction	107
2. Représentation par adjacence	108
2.1 Principe	108
2.2 Codage	110
3. Initialisation.....	112
4. Evaluation	113
5. Mutation.....	114
6. Croisement	117
7. Implémentation	117
8. Résultats et discussion.....	118
8.1 Treillis de Michell	118
8.2 Poutre console (cantilever).....	126
8.3 Une structure de type Michell	128
9. Conclusions	131

1. Introduction

Dans ce chapitre nous présentons une approche pour l'optimisation topologique de structures continues basée sur une représentation par adjacence. Le principe de base de cette représentation est fondé sur le concept de connectivité qui est exprimé par une matrice binaire dite d'adjacence. Par rapport à la représentation par courbes de Bézier, qui fait l'objet du chapitre II, cela permet de traiter tout les problèmes d'optimisation topologique des structures. Cette approche par adjacence est une contribution originale dont les concepts de base sont inspirés des méthodes déterministes [Abd04, Tat00, Tov04].

La représentation en tableau binaire (bit-array [San90, Jen92, Jak00]), qui est souvent adoptée dans le cadre des algorithmes génétiques, est basée sur le principe d'absence ou de présence de matière. Ce principe, qui se traduit par un codage en chaîne binaire, est bien adapté à la mise en œuvre des algorithmes génétiques. Il présente néanmoins l'inconvénient de ne pas tenir compte de la connectivité, ce qui est un frein à son utilisation pour traiter les problèmes d'optimisation topologique. De plus C. Kane [Kan96] a montré qu'il n'y a pas équivalence entre la représentation en tableau binaire et la représentation en chaîne binaire si on utilise les opérateurs de croisement standard de la théorie des algorithmes génétiques (Fig. 1-10, chapitre I).

Pour l'essentiel, l'optimisation topologique d'une structure se résume à déterminer la nature et la connectivité des éléments qui la constituent. L'approche que nous présentons s'appuie sur cette considération. Elle consiste à considérer chaque élément fini comme une cellule et à définir une représentation basée sur la connectivité de ces cellules entre elles. Cette connectivité s'exprime par une matrice binaire creuse, dite d'adjacence. Le codage des structures solutions utilise cette matrice en transformant cette dernière en une chaîne binaire unidimensionnelle. Ce codage est bien adapté au problème d'optimisation topologique puisque il est basé sur la connectivité. De plus, poser le problème en variables binaires augmente considérablement l'efficacité de l'algorithme génétique. Le principe de cette nouvelle approche est présenté en détail dans le paragraphe qui suit cette introduction.

La recherche de la solution optimale, qui correspond à une distribution optimale de matière, s'effectue en déterminant les connectivités des éléments cellules. La mutation joue le rôle le plus important pour réaliser cette tâche. L'opération de mutation est inspirée partiellement de la méthode des automates cellulaires (Chapitre I : § 4.1.3). Nous affectons à chaque cellule une variable de densité artificielle dont la valeur est altérée en mutant. La technique de mutation est étudiée en s'inspirant des travaux de Tovar and al. [Tov04]. La valeur de la densité artificielle est utilisée ensuite pour la recherche des éléments liaisons

associés à la cellule mutante. De par l'utilisation d'une variable de densité, l'approche que nous proposons présente une certaine proximité avec les techniques d'homogénéisation [Ben03] même si, par ailleurs, les principes mis en œuvre sont très différents.

Sur la base des développements théoriques présentés dans ce chapitre, nous avons développé un code de calcul en langage Matlab [Hig05]. Des applications numériques sont présentées à la fin de ce chapitre afin de valider l'approche proposée.

2. Représentation par adjacence

Dans ce paragraphe nous présentons le principe, ainsi que la mise en œuvre sous forme de chromosome, de la représentation par adjacence.

Dans le cadre de la thèse, nous nous sommes limités aux problèmes bidimensionnels sachant que l'approche proposée peut être étendue facilement aux problèmes en 3D.

2.1 Principe

Le principe de cette nouvelle représentation s'appuie sur le concept de voisinage ou d'adjacence. Chaque élément fini est considéré comme une cellule entourée par d'autres éléments finis qui la relient avec d'autres cellules (Fig. 3-1). En considérant les éléments finis, en gris sur la figure 3-1, comme cellule, les éléments qui remplissent la tâche de connexion des cellules entre elles sont considérés comme éléments liaisons (en blanc). La suppression d'une liaison entraîne l'élimination de l'élément fini associé et par conséquent l'élimination de matière occupant l'espace géométrique occupé. Par contre, l'ajout de matière s'opère par le renforcement de la connexion d'une cellule avec ses cellules voisines en ajoutant un ou plusieurs éléments liaisons. Pratiquement, l'élément considéré comme cellule dans la génération actuelle sera une liaison dans la génération qui suit, et vice versa pour un élément liaison. Chaque élément fini, en tant que cellule ou liaison, est donc alterné d'une génération à l'autre. Cette alternance permet de donner le maximum de souplesse à la représentation, sans quoi l'exploration de l'espace des solutions risque d'être bridée si des éléments restent figés dans un rôle particulier.

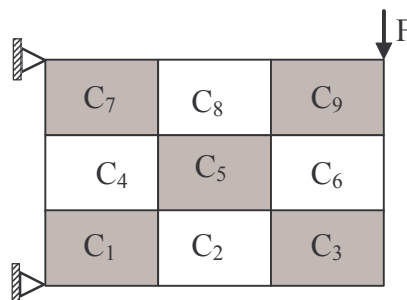


Figure 3-1. Principe de la représentation par adjacence.

Eléments de frontières

Un cas particulier concerne les éléments cellules qui se trouvent sur les surfaces d'application des conditions aux limites. Pour ces éléments, nous ajoutons des cellules fictives voisines qui représentent les surfaces d'application des conditions aux limites (Fig. 3-2). Les cellules C_1 , C_7 et C_9 ont les cellules fictives voisines C_{1f} , C_{7f} et C_{9f} respectivement.

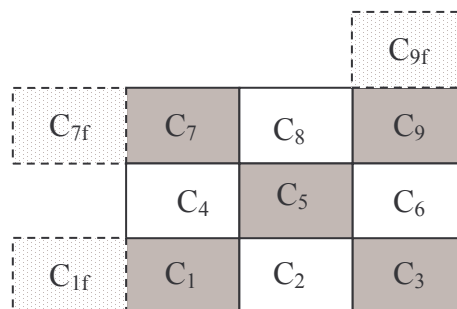


Figure 3-2. Cellules fictives sur les frontières.

Avant de passer au codage, nous illustrons le principe des relations d'adjacence entre les cellules à l'aide d'un exemple (Fig. 3-3). Cet exemple concerne les éléments cellule et liaison de la figure 3-2. La ligne d'entrée de la matrice booléenne de la figure 3-3 est formée par les éléments cellules (C_1 , C_3 , C_5 , C_7 et C_9) et la colonne d'entrée par les éléments liaisons (C_2 , C_4 , C_6 et C_8). Les relations entre les éléments cellules et les éléments liaisons sont exprimées par une matrice booléenne. La valeur 1 indique la présence de l'élément liaison alors que la valeur 0 indique l'absence de cet élément. Un trait symbolise le fait qu'un élément cellule donné ne peut être mis en relation avec un élément liaison donné. Seules les valeurs 0 et 1 sont donc prises en considération par l'algorithme génétique.

Pour expliquer pratiquement, sur un exemple, comment s'opère le remplissage de la matrice d'adjacence, prenons le cas de la cellule C_1 . Cette dernière est entourée par les éléments liaisons C_2 , C_4 et un élément fictif C_{1f} . La connectivité de cette cellule C_1 s'exprime dans notre exemple par l'absence de la liaison C_2 (valeur 0 dans le tableau), et par la présence des liaisons C_4 et C_{1f} (valeurs 1 dans le tableau). Autrement dit, l'élément cellule C_1 possède un potentiel maximum de trois liaisons (C_2 , C_4 et C_{1f}) qui s'activent (1) ou se désactivent (0) au gré des itérations de l'algorithme génétique.

	C ₁	C ₃	C ₅	C ₇	C ₉
C ₂	0	1	1	-	-
C ₄	1	-	1	1	-
C ₆	-	1	1	-	1
C ₈	-	-	1	0	1
C _{1f}	1	-	-	-	-
C _{7f}	-	-	-	1	-
C _{9f}	-	-	-	-	1

Figure 3-3. Représentation par une matrice d'adjacence.

La représentation de la structure de la figure 3-2 est la partie de la matrice d'adjacence (fig. 3-3) qui exprime la connectivité des éléments entre eux. Cette partie, indiquée par la couleur grise, est celle que nous utilisons pour construire le génotype, c'est-à-dire le codage.

2.2 Codage

Le codage consiste à transformer la matrice d'adjacence donnée par la figure 3-3 sous forme d'une chaîne unidimensionnelle afin de faciliter les opérations génétiques, notamment celles du croisement. A l'inverse du codage basé sur la représentation 'Bit-array', où chaque élément fini est représenté par un bit, les éléments sont représentés par des gènes dans notre approche. La colonne, correspondant à un élément cellule dans la matrice d'adjacence, représente le gène associé à cet élément fini. Par contre, le gène associé à un élément liaison correspond à une ligne dans la matrice d'adjacence. Le codage consiste à disposer ces gènes les uns à côté des autres en formant ainsi une chaîne binaire. Le chromosome résultant, basé sur la représentation de la figure 3-1, est donné par :

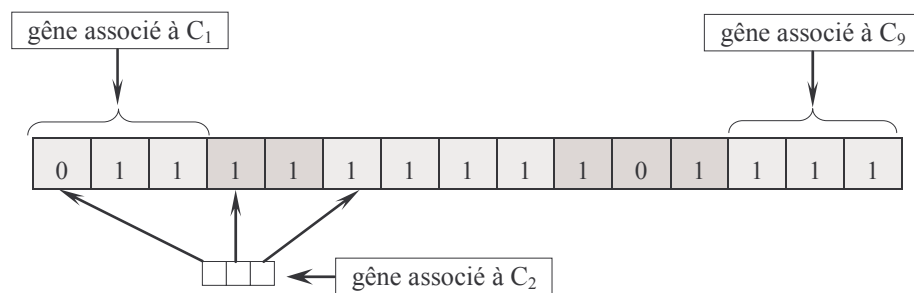


Figure 3-4. Codage basé sur la représentation par adjacence.

Chaque gène du chromosome est composé d'un certain nombre de bits où chaque bit est associé à un élément de type liaison. Concrètement, chaque élément fini est une cellule entourée par des éléments liaisons. Dans ce cas, les gènes des éléments cellules et les gènes des éléments liaisons partagent entre eux des bits.

Le codage ne consiste pas seulement à stocker les paramètres de la représentation sous forme de chaîne. La disposition des informations génétiques, qui représente le chromosome, doit être aussi compatible avec les opérateurs de croisement. Or, cette compatibilité dépend de la technique utilisée pour ce stockage.

Nous avons comparé entre elles deux techniques différentes de codage. La première correspond à celle que nous venons de présenter (Fig. 3-4). Dans cette technique, la disposition des gènes les uns à côté des autres dans le but de construire le chromosome est aléatoire. Les tests numériques que nous avons réalisés dans ce cas montrent que le génotype résultant n'est pas compatible avec les opérateurs de croisement basé sur les points de coupure.

La deuxième technique est similaire à la première. Cependant, l'ordre des éléments cellules sur la chaîne (chromosome) est basé sur le lien de connectivité entre les éléments exprimé par l'arborescence de la figure 3-5. Le principe des relations entre cellules est indiqué par des flèches sur la figure 3-5. Si on commence par exemple par l'élément cellule C_1 , les éléments liaisons qui le jouxtent sont C_2 et C_4 . On poursuit ensuite par les éléments cellules associés à C_2 et C_4 , soit C_3 , C_5 et C_7 . Les éléments liaisons associés à ces trois derniers sont C_6 et C_8 . On termine enfin par l'élément cellule C_9 et on obtient ainsi une disposition arborescente complète de la figure 3-1.

Cette deuxième technique de codage est celle que nous avons adoptée. Les tests numériques que nous avons réalisés ont en effet montré qu'elle est plus efficace que la première technique de codage.

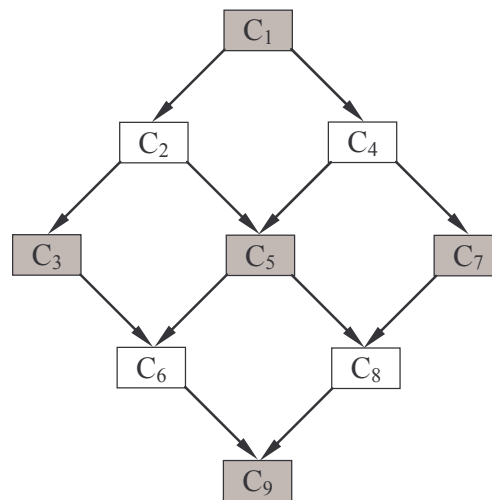


Figure 3-5. Connectivités des éléments finis en arborescence.

3. Initialisation

La première étape dans un algorithme génétique est l'initialisation. Elle consiste à générer aléatoirement des individus qui représentent des structures solutions. Cette première étape est très importante car la première population doit être suffisamment diversifiée pour permettre l'exploration de tout le domaine de recherche.

Un individu est initialisé en sélectionnant aléatoirement, en première étape, un certain nombre d'éléments finis. Les éléments sélectionnés sont du même type, c'est-à-dire qu'ils sont tous de type cellule ou de type liaison. L'initialisation consiste à affecter au bit du gène associé à chaque élément la valeur 1 qui indique la présence de l'élément liaison associé à ce bit ou la valeur 0 pour indiquer son absence. Chaque gène doit comporter un nombre de liaison minimal assurant la connectivité de chaque élément cellule avec au moins deux cellules voisines. A l'inverse de la représentation en tableau binaire '*Bit-array*', notre approche assure la connectivité de toutes les structures initialisées. La figure 3-6 montre quelques individus initialisés aléatoirement. Chaque cellule est représentée par une couleur dont le niveau de gris est proportionnel au nombre d'éléments liaisons entourant la cellule. La couleur blanche indique l'absence de matière alors que la couleur noire indique une présence maximale. La couleur intermédiaire indique la présence de matière mais avec une densité plus faible.

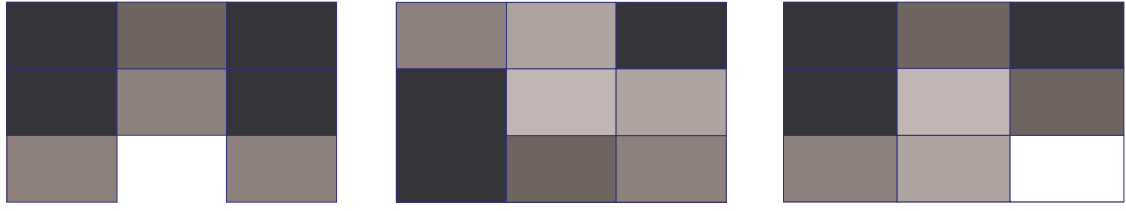


Figure 3-6. Des individus initialisés aléatoirement.

Les structures, ou individus, générés dans cette première étape de l'algorithme, ne comportent pas de défauts de connectivité. Néanmoins, il est possible que de tels défauts apparaissent au niveau des surfaces d'application des conditions aux limites. Dans ce cas, une opération de correction de ces défauts est mise en œuvre après l'initialisation. Cette opération de correction consiste à vérifier la connectivité des cellules, sur les surfaces des conditions aux limites, avec au moins une cellule fictive.

4. Evaluation

L'évaluation est effectuée à l'aide de la méthode des éléments finis. La formulation de la fonction performance est la même que celle utilisée dans le chapitre II (§ 5-5). Cependant, nous utilisons dans cette approche une technique de relaxation des propriétés physiques du matériau semblable à celle utilisée dans la méthode d'homogénéisatrice SIMP [Ben03]. Le module de Young est en effet fonction d'un paramètre de densité relative ρ et du module de Young réel du matériau :

$$E_i(ie) = \rho_i^p E_0 \quad (p > 1) \quad (3-1)$$

où E_0 est le module d'Young réel du matériau de l'élément ie , p est un coefficient de pénalisation et ρ_i est la densité relative donnée par l'expression :

$$\rho_i(ie) = \frac{\sum_{j=1}^{Nbts(ie)} bit_j(ie)}{Nbts(ie)} \quad (3-2)$$

où $Nbts(ie)$ est la longueur du gène associé à l'élément cellule ie et bit_j le $j^{ème}$ bit de ce gène.

L'expression (3-2) donne la densité de l'élément cellule ie en fonction de la moyenne des valeurs de bits. Sachant que ces valeurs sont binaires, la densité devient nulle dans le cas d'un élément fini ne possédant aucun élément liaison. La densité est égale à 1 dans le cas où tous les bits ont la valeur 1. Pratiquement, nous affectons une valeur faible, de

l'ordre de 10^{-4} , aux densités nulles pour éviter la singularité de la matrice de rigidité. Cette dernière est calculée en assemblant les matrices élémentaires associées aux éléments du maillage :

$$K_G = \sum_{ie=1}^{nelt} \rho_{ie}^p k(ie) \quad (3-3)$$

où K_G est la matrice de rigidité globale, $k(ie)$ est la matrice élémentaire associée à l'élément ie et $nelt$ est le nombre d'éléments total.

La matrice de rigidité élémentaire $k(ie)$ s'exprime, entre autres, en fonction du module d'Young réel E_0 du matériau. Cela permet de calculer les matrices de rigidité élémentaires indépendamment des densités artificielles ρ_i qui sont affectées aux éléments.

5. Mutation

L'opération de mutation consiste à modifier un certain nombre de gènes du chromosome de l'individu mutant. Ce nombre de gène est l'un des paramètres qui doivent être judicieusement choisis afin d'assurer la convergence de l'algorithme. Dans ce troisième chapitre, la mutation que nous utilisons est la mutation uniforme, souvent utilisée dans un algorithme génétique classique (chapitre I § : 4.2.2.8). Un individu sélectionné aléatoirement dans la population mutera donc si sa probabilité de mutation vérifie la condition 2-27.

Le choix de la cellule qui doit muter s'opère en deux étapes afin que l'opération de mutation n'affecte l'ajout éventuel de matière. Dans la première étape, nous tirons aléatoirement une cellule de densité non nulle, c'est-à-dire une cellule dont le gène associé contient au moins un bit non nul. La deuxième étape consiste à effectuer un deuxième tirage aléatoire parmi toutes les cellules voisines de celle retirée à la première étape. C'est cette deuxième étape qui permet de définir la cellule mutante. Afin de mieux expliquer le fonctionnement de ce tirage, considérons les cellules présentées sur la figure 3-7. Supposons que la cellule C_1 soit sélectionnée aléatoirement à la première étape. Le deuxième tirage s'effectue sur l'ensemble $\{C_1, C_2, C_3, C_4, C_5\}$. On rappelle que le niveau de gris représente la densité et que le blanc indique une densité nulle. Contrairement à la 1^{ère} étape, les cellules de densités nulles (C_3 et C_4) peuvent être choisies pour muter. On permet ainsi l'ajout de matière sur les frontières.

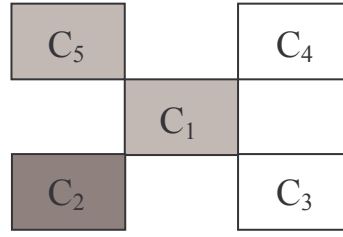


Figure 3-7. Technique pour le tirage de la cellule mutante.

Cette technique de mutation en deux étapes permet de garantir la propagation de matière. En effet, une cellule tirée à la première étape peut se trouver sur une frontière d'une perforation. Le choix d'une cellule voisine pour muter favorise donc la propagation de la matière dans le domaine de conception.

En choisissant aléatoirement une cellule, l'opération de mutation consiste à modifier les informations du gène associé. Il s'agit de la modification des valeurs binaires des bits de ce gène qui entraîne, en conséquence, la suppression ou l'ajout des éléments liaisons. Pratiquement, le nombre des éléments liaisons est proportionnel à la densité de l'élément cellule. Par conséquent, la modification de la densité de l'élément cellule mutant conduit à la modification du nombre des éléments liaisons. Nous avons proposé une stratégie basée sur l'adaptation de chaque cellule avec son voisinage. Cette stratégie s'inspire de la méthode des automates cellulaire 'Cellular automata' [Tov04]. Le principe consiste à interpoler la densité de la cellule mutante en fonction des densités des cellules voisines, puis à corriger sa valeur selon l'expression suivante :

$$x_{ie} = \hat{x}_{ie} + \Delta x_{ie} \quad (3-4a)$$

$$\text{avec : } \hat{x}_{ie} = \frac{\rho_{ie} + \sum_{je=1}^{Nc} \rho_{je}}{Nc + 1} \quad (3-4b)$$

$$\text{et : } \Delta x_{ie} = \frac{rand \cdot (-1)^{round(rand)}}{length(gen_{ie})} \quad (3-4c)$$

x_{ie} représente la densité artificielle de l'élément mutant ie , $\hat{x}_{ie}$ la moyenne des densités des cellules voisines y compris la cellule mutante, Nc le nombre de cellule voisines et ρ_{je} les densités données par l'expression (3-2). Δx_{ie} est la correction donnée par la formule 3-4c, $rand$ est une valeur, générée aléatoirement, comprise entre 0 et 1 et $round$ est une fonction d'arrondi vers l'entier le plus proche.

L'expression (3-4a) pouvant donner une valeur supérieure à 1 ou inférieure à 0, on apporte la correction suivante :

$$x_{ie} = \begin{cases} 0 & \text{si } x_{ie} < 0 \\ 1 & \text{si } x_{ie} > 1 \\ x_{ie} & \text{si } 0 < x_{ie} < 1 \end{cases} \quad (3-5)$$

Le nombre entier N_c dépend du schéma choisi. Dans notre approche, nous utilisons l'un ou l'autre des schémas de Von Neumann et de Moor (Chapitre I : § 4.1.3) (Fig. 3-8).

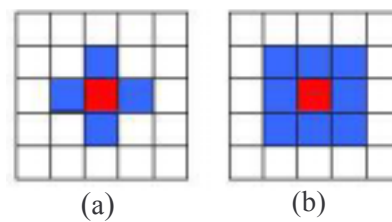


Figure 3-8. Schéma de type Von Neumann ($N_c = 4$), (b) Moor ($N_c = 8$).

L'expression (3-4a) est composée de deux parties. La première exprime une interpolation de la densité. Cette partie, équivalente à un filtre, participe efficacement à la l'élimination de damiers et des éléments flottants. La deuxième partie est la correction qui permet soit d'accroître soit de diminuer la densité.

La densité artificielle x_{ie} est utilisée ensuite pour calculer le nombre de liaisons qui doivent être maintenues. Les bits correspondants à ces éléments liaisons prennent la valeur 1 tandis que le reste des bits ont des valeurs nulles. Le nombre de ces liaisons est calculé par la formule :

$$Nbr = round(x_{ie} \times Nbit) \quad (3-6)$$

où Nbr est un nombre entier qui correspond au taux des éléments liaisons. Le nombre maximal $Nbit$ est la taille du gène associé à l'élément mutant.

L'affectation de la valeur 1 à Nbr bits, permettant de définir les éléments liaisons, s'effectue aléatoirement parmi les bits du gène mutant. Cependant, les bits correspondants aux éléments liaisons assurant la continuité de la matière sont choisis prioritairement. Ce choix est effectué en testant les éléments liaisons qui permettent la continuité de la matière. Concrètement, on repère ceux qui possèdent des cellules voisines de densité non nulle.

6. Croisement

Nous avons testé plusieurs types de croisement. Les résultats ont montré que le croisement uniforme et le croisement basé sur des points de coupure multiples sont les plus efficaces. Rappelons que le croisement uniforme consiste à échanger un nombre aléatoire de points entre les deux chromosomes parents. Dans notre approche, les points échangés sont des parties d'un certain nombre de gènes choisis aléatoirement. Le croisement à points de coupure multiples consiste à échanger des parties des deux chromosomes, incluses entre des points de coupures pris aléatoirement (Fig. 3-9). Le nombre de points, en croisant deux individus très proches, est élevé afin de permettre un échange maximal d'informations entre les deux parents.

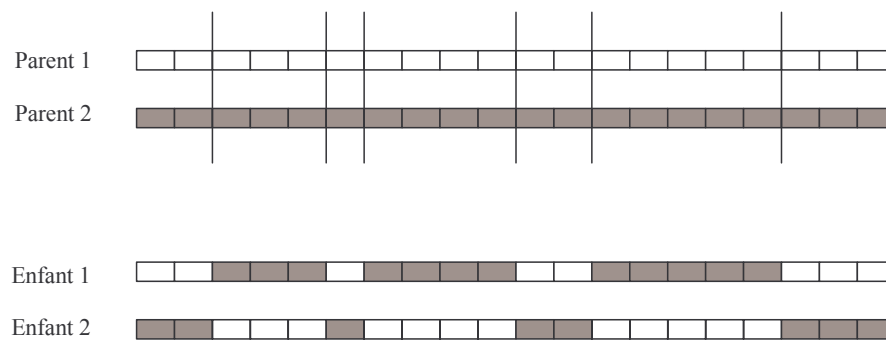


Figure 3-9. Principe du croisement à points de coupures multiples.

7. Implémentation

Sur la base de la théorie présentée dans les paragraphes précédents, nous avons développé un code de calcul en langage Matlab selon l'organigramme de la figure 3-10.

Le code de calcul est composé de deux parties. La première est celle où l'utilisateur introduit les paramètres du problème traité (maillage, conditions aux limites, propriétés physiques, paramètre de l'algorithme génétique, etc.). La deuxième partie se décompose en deux sous parties. La première concerne la détermination des paramètres du problème, notamment ceux du codage en se basant sur les données fournies par l'utilisateur. La deuxième partie est un algorithme génétique classique.

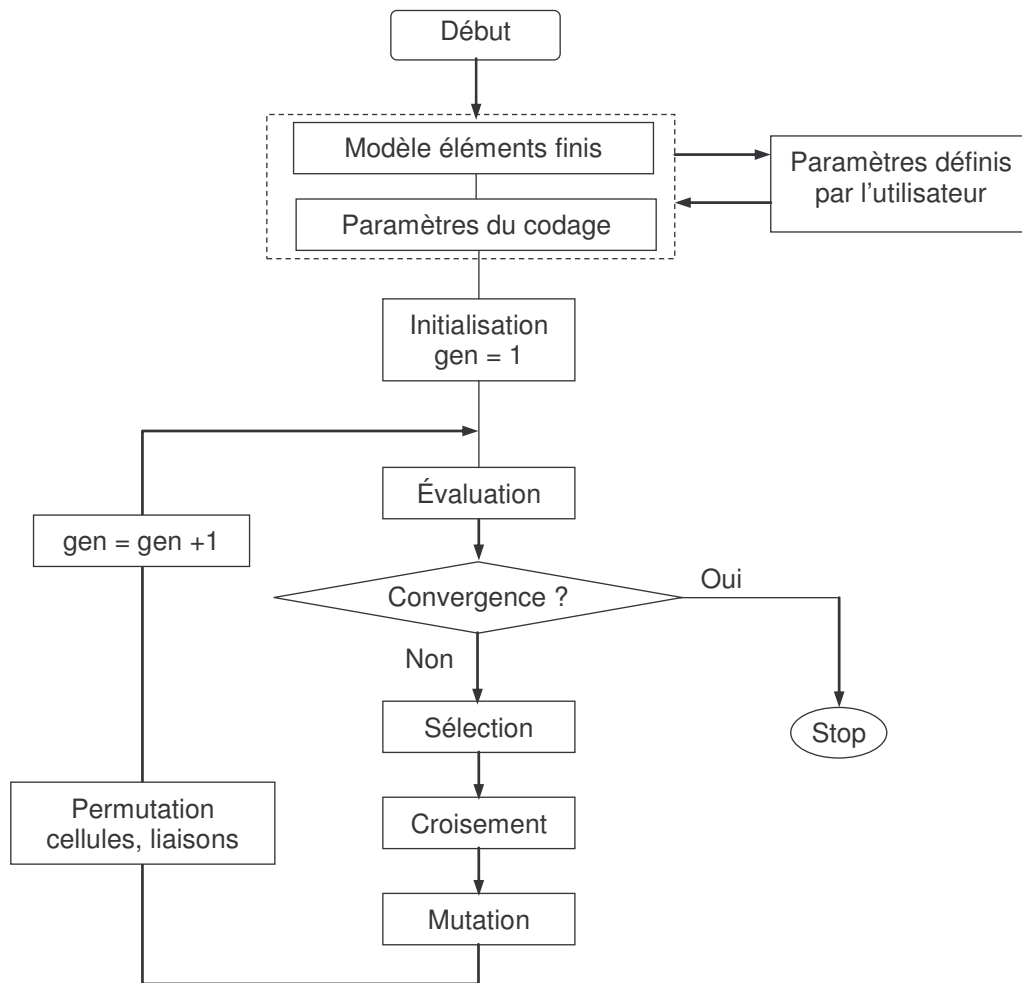


Figure 3-10. Organigramme de l'approche proposée.

8. Résultats et discussion

Nous avons testé notre code de calcul sur quelques benchmarks simples en nous limitant au cadre de l'élasticité linéaire bidimensionnelle sachant que la méthode peut être étendue facilement aux problèmes 3D.

8.1 Treillis de Michell

Cette première application est destinée à l'étude du problème de la poutre courte traité dans le chapitre II (§ : 8.1). Il s'agit d'une poutre courte de forme rectangulaire de dimensions $10 \times 24 \text{ m}^2$ et d'épaisseur égale à 1 mm. La structure est encastree sur le côté gauche et sollicitée par une force concentrée au milieu du côté droit (fig. 3-11). Le domaine de conception, composé d'un matériau homogène isotrope élastique et linéaire, est discrétisé en 10×24 élément finis rectangulaires.

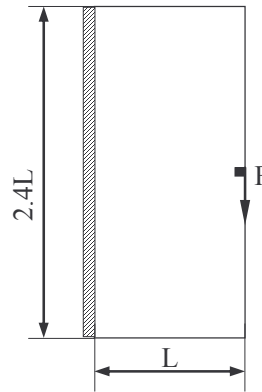


Figure 3-11. Domaine de conception (treillis de Michell).

Le problème d'optimisation consiste à minimiser la compliance sous limitation sur le volume. Le problème se formule de la manière suivante :

$$\begin{aligned} & \underset{x_i}{\text{Maximiser}} \quad \frac{1}{F^T U(x_i)} \\ & \text{avec : } \sum_{ie=1}^{nelt} \rho_{ie} V_{ie} \leq V_{Lim} \end{aligned} \quad (3-7)$$

où F est le vecteur force appliqué, $U(x_i)$ est le déplacement de l'individu x_i , ρ_{ie} et V_{ie} sont respectivement la densité artificielle et le volume de l'élément ie et V_{Lim} le volume limite fixé dans cette application à 50 % du volume initial.

L'algorithme génétique utilise comme paramètres un schéma de voisinage pour les cellules de type Von Neumann, une population composée de 40 individus, la méthode de sélection SUS, un croisement uniforme et une mutation uniforme. Les coefficients de croisement $Coef_c$ et de mutation $Coef_m$ prennent les valeurs 0.9 et 0.35 dans les équations (2-26) et (2-28) que nous avons utilisées dans cet exemple. Les paramètres, notamment les coefficients de croisement et de mutation, sont choisis sur la base de nombreux tests que nous avons réalisés. Selon ces tests, le coefficient $Coef_c$ doit prendre ses valeurs dans l'intervalle [0.8, 0.9] alors que le coefficient de mutation $Coef_m$ se situe généralement dans l'intervalle [0.1, 0.4]. Concernant le schéma que nous avons utilisé, les tests numériques que nous avons réalisés montrent que le schéma de type Von Neumann est le mieux adapté pour cet exemple. En effet, le schéma de type Moor a tendance à favoriser les défauts de discontinuité de la matière.

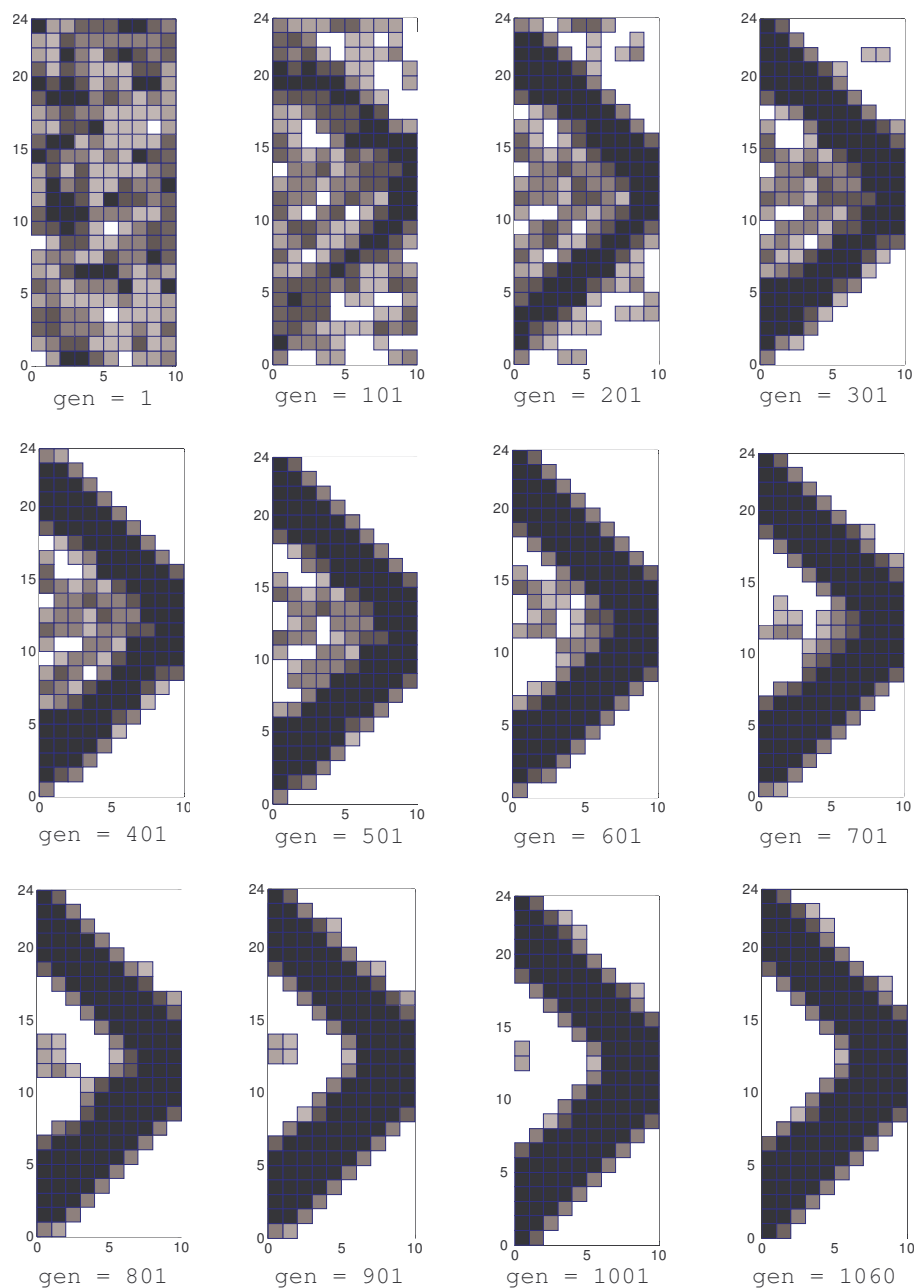


Figure 3-12. Évolution de la meilleure solution (treillis de Michell).

La figure 3-12 présente l'évolution de la meilleure solution au cours des générations. L'objectif est de faire apparaître des zones de densité nulle, où la matière est absente, et des zones de densité élevée égale ou proche de 1. Au cours des premières générations (gen = 1 par exemple), la matière est répartie sur tout le domaine de conception avec des densités différentes. En revanche, au cours des générations suivantes, deux zones, l'une de forte densité et l'autre de faible densité, se forment et se distinguent nettement. Par ailleurs, on remarque que des éléments flottants peuvent apparaître au cours des générations (gen =

201 par exemple) mais qu'ils disparaissent ensuite grâce à la technique de filtrage que nous avons implémenté (formule 3-4b). Enfin, on constate que la solution finale (gen = 1060) est similaire à la solution théorique présentée sur la figure 2-20 (chapitre II).

La figure 3-13 présente l'historique de la fonction objectif (compliance) ainsi que l'historique des contraintes (volume) au cours des générations. La compliance atteint un minimum avec une décroissance continue au cours des quatre cent premières générations. A l'issue de ces quatre cent générations, la topologie finale de la solution est quasiment acquise (gen = 401, Fig. 3-12), la pente de décroissance de la compliance étant alors très faible. En revanche, pour ce qui concerne la limitation imposée sur le volume, on constate des oscillations importantes autour du volume imposé de 50 % (Fig. 3-13b). Ces oscillations ont cependant tendance à s'estomper au cours des générations pour se stabiliser à 49.96 % du volume initial à la fin du calcul, ce qui est très proche de la limitation imposée de 50 %.

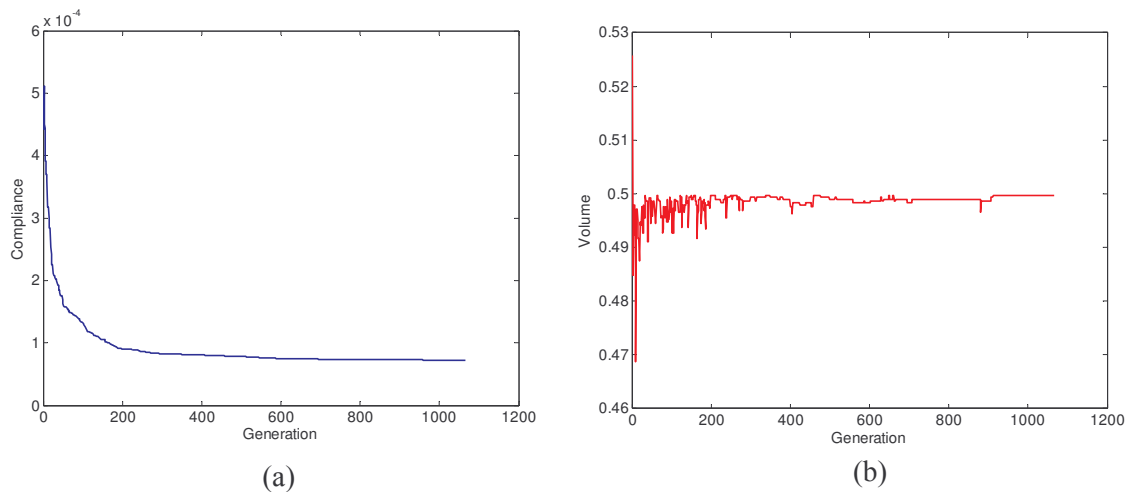


Figure 3-13. Historique : (a) de la fonction objectif (b) des contraintes.

Sur la figure 3-14 nous représentons le nombre d'individus évalués et le temps de calcul consommé au cours des générations.

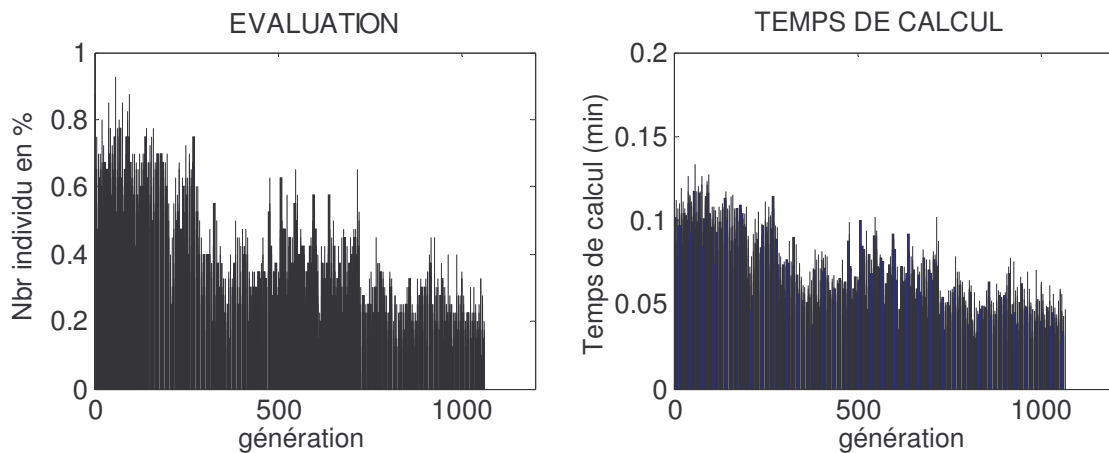


Figure 3-14. Historique du nombre d'individus évalués et du temps consommé.

On constate sur la figure 3-14 que le temps total consommé au cours des générations, temps dédié aux opérations génétiques inclus, est proportionnel au nombre d'individus évalués. Cela signifie que l'évaluation est l'opération la plus coûteuse dans un algorithme génétique. La figure montre aussi que le coût est élevé au début du calcul puis qu'il diminue en convergeant. Cela est dû à la technique que nous avons utilisée et qui consiste à ne pas évaluer les individus qui ont été copiés, sans modifications, à partir de la génération précédente. Les premières générations étant plus diversifiées et les dernières plus homogènes, cet effet est logiquement plus marqué lorsque l'on converge.

Influence des taux de mutation

Les taux de mutation, le nombre d'individus et le nombre de cellules mutantes influencent beaucoup l'algorithme. C'est pourquoi nous nous proposons d'étudier numériquement l'influence de ces paramètres dans la suite – deux exemples sont proposés. Le premier traite d'un cas où la mutation est faible et le second d'un cas où la mutation est forte.

Mutation faible

En affectant au coefficient de mutation $Coef_m$ (équation 2-28) une valeur faible égale à 0.05 et un taux de cellules mutantes égale à 0.001, on constate sur la figure 3-15, qui présente l'historique de la meilleure solution au cours des générations, que l'on ne converge pas vers la solution théorique de référence donnée par la figure 2-20.

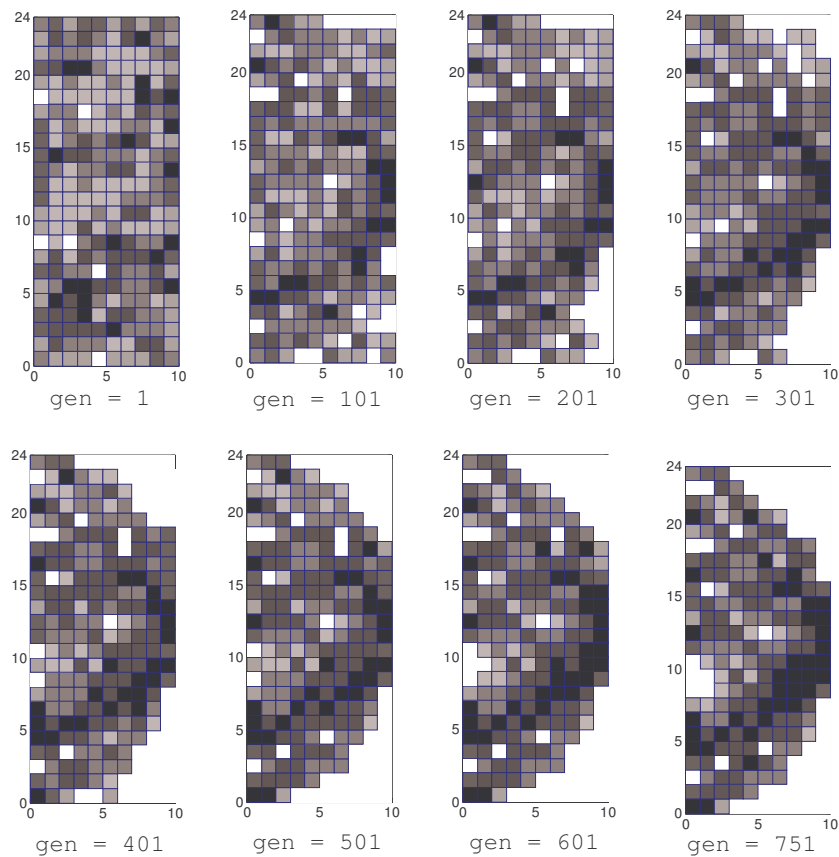


Figure 3-15. Historique de la meilleure solution (mutation faible).

Par rapport au calcul où il y avait convergence (Fig. 3-12), la figure 3-15 montre que la topologie évolue très faiblement au cours des générations. A partir de la 600^{ème} génération (gen = 601), on observe même une quasi stagnation. Cela s'explique par la chute de la diversité génétique, ce qui conduit à une population homogène puis à la stagnation de l'algorithme.

Par rapport au calcul précédent (Fig. 3-13), on constate sur la figure 3-16a que la décroissance de la compliance présente un profil moins régulier, signe d'une convergence plus difficile. En outre, au cours des générations, le volume de la meilleure solution reste souvent au dessous de la limitation imposée (Fig. 3-16b). Lors des dernières générations, le volume se stabilise à 49.7 % du volume initial, soit une performance moindre que le précédent calcul (Fig. 3-13b) où la stabilisation s'opérait sur une valeur de 49.96 %.

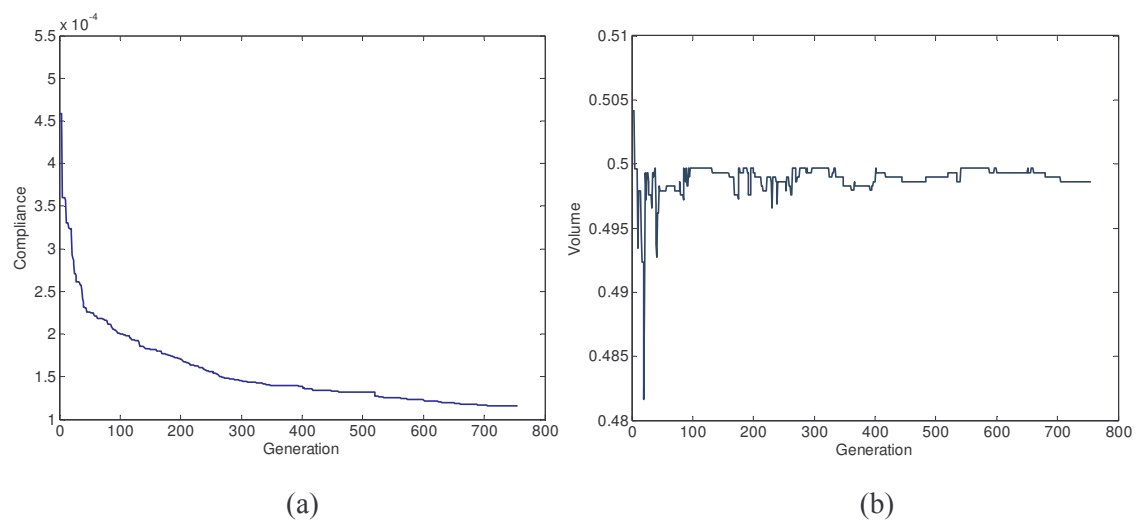


Figure 3-16. Mutation faible - Historique : (a) de la fonction objectif, (b) de la limitation.

Mutation forte

Dans ce deuxième test, le taux de mutation est élevé avec un coefficient de mutation $Coef_m$ égal à 0.5 et un taux de cellules mutantes égal à 0.1. L'historique de la meilleure solution au cours des générations est présenté sur la figure 3-17.

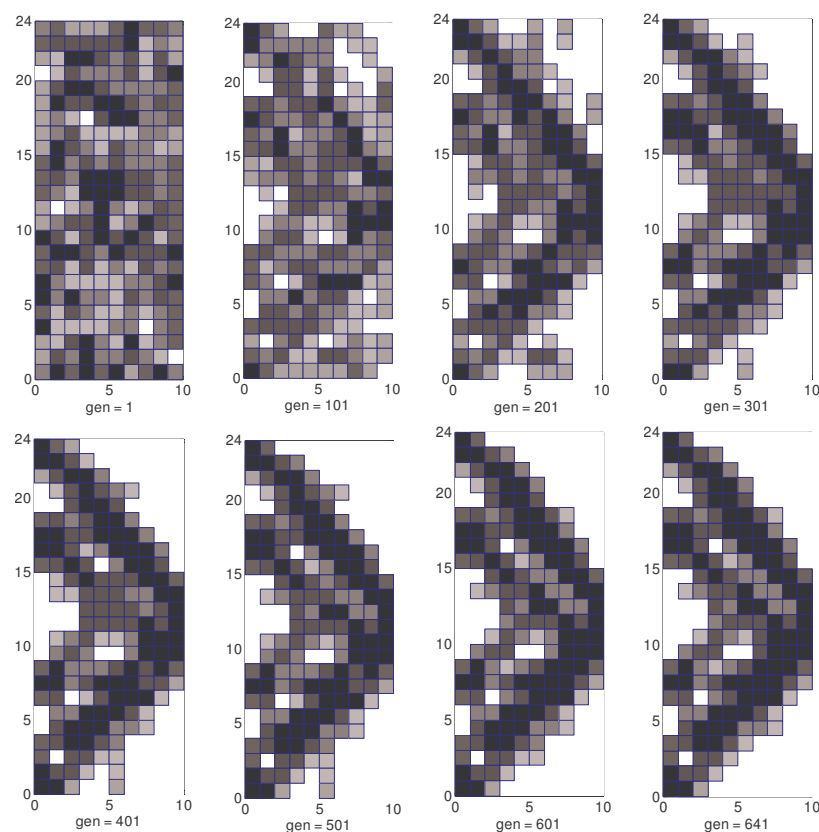


Figure 3-17. Historique de la meilleure solution (mutation forte).

La topologie de la solution, présentée sur la figure 3-17, évolue, au cours des générations, plus rapidement que celle de la figure 3-15 et celle sur la figure 3-12. Cependant, cette évolution stagne à partir de la 300^{ème} génération. Cela en effet est à cause de la mutation élevée qui provoque une dégradation d'informations génétiques et, par conséquence, une décadence des performances des individus. La solution finale, trouvée après 600 générations environ, n'est pas similaire à la solution théorique de la figure 2-20. De plus, la distribution de matière de la solution finale n'est pas homogène en raison de la présence de perforations provoquées par une mutation trop forte.

La figure 3-18a montre une décroissance de la compliance moins régulière par rapport au calcul présenté sur la figure 3-13. A partir de la 350^{ème} génération environ, la compliance devient quasiment stable. Par contre, l'évolution du volume (limitation), présentée sur la figure 3-18b, montre moins d'oscillations par rapport au calcul précédent. Cela est caractéristique d'une population dispersée mais possédant des caractéristiques génétiques dégradées.

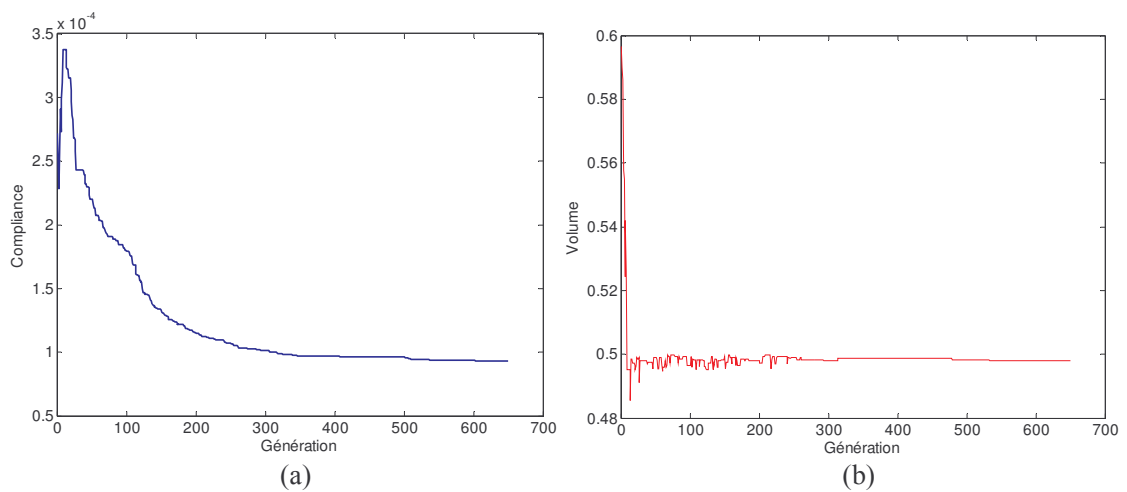


Figure 3-18. Mutation forte - Historique : (a) de la fonction objectif, (b) de la limitation.

Pratiquement, l'étude de sensibilité réalisée dans ce paragraphe fournit des indications pratiques sur le choix du nombre de cellules mutantes. En effet, il est clair qu'un nombre trop faible dégrade la convergence de l'algorithme génétique. De plus, un taux trop faible peut conduire, dans certain cas de maillage grossier, à un nombre de cellules mutantes nul. C'est la raison pour laquelle nous fixons un nombre minimal de cellules mutantes égal à 1.

8.2 Poutre console (cantilever)

Le problème traité ici est similaire à celui du chapitre II (§ : 8.2). C'est une poutre console de longueur $L = 1.6$ m et de hauteur H égale à 1 m, encastrée sur le côté gauche et sollicitée par une force concentrée au milieu du côté droit (Fig. 3-19). Le domaine de conception est composé d'un matériau homogène élastique, isotrope et linéaire.

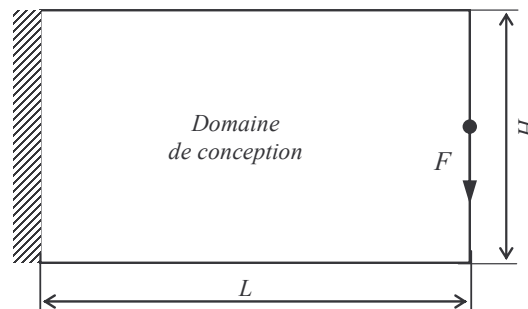


Figure 3-19. Domaine de conception de la poutre console.

L'objectif est de minimiser la compliance sous limitation d'un volume à 50 % du volume initial de la structure initiale. La structure est discrétisée en 29×18 éléments finis rectangulaires à quatre nœuds. Les paramètres de l'algorithme génétique sont une population de taille fixe égale à 80, un schéma de cellule de type Moor, la méthode de sélection SUS, le coefficient de croisement $Coef_c$ égal à 0.9 et de mutation $Coef_m$ égal à 0.25. Par rapport au benchmark traité dans le paragraphe précédent, le nombre d'éléments du maillage est plus important (522 éléments contre 210 précédemment, soit plus du double). La taille de la population a donc été augmentée en conséquence (80 individus contre 40 précédemment). Pour ce qui concerne le schéma utilisé, nous avons comparé les résultats obtenus à l'aide des schémas de Von Neumann et de Moor. Pour cet exemple, c'est le schéma de Moor qui a paru le plus adéquat.

La figure 3-20 montre l'évolution de la topologie de la meilleure solution au cours des générations. On fait un constat similaire, concernant l'évolution de la topologie, que celui qui a été établi pour l'exemple traité au paragraphe précédent. En effet, deux zones, l'une de faible et l'autre de forte densité, apparaissent progressivement au cours des générations. On constate également que l'on converge vers la topologie obtenue à l'aide de méthodes déterministes (Fig. 2-31a), et ce au moins dans les grandes lignes. En effet, dans la partie centrale du domaine de conception, l'algorithme que nous avons développé ne permet pas d'obtenir la structure en croix de la figure 2-31a car, à partir de 800^{ème} génération, on observe une stagnation. Cela peut s'expliquer par le fait que la topologie recherchée est plus complexe que celle de l'exemple traité au paragraphe précédent. Cela montre aussi

que des améliorations doivent être apportées à l'algorithme pour le rendre plus performant. L'amélioration pourrait par exemple être réalisée en introduisant dans notre algorithme des techniques de filtrage utilisées par des méthodes déterministes. Des limitations sur le périmètre [Bec97, Duy96] pourrait aussi être introduites dans notre algorithme afin d'améliorer les résultats.

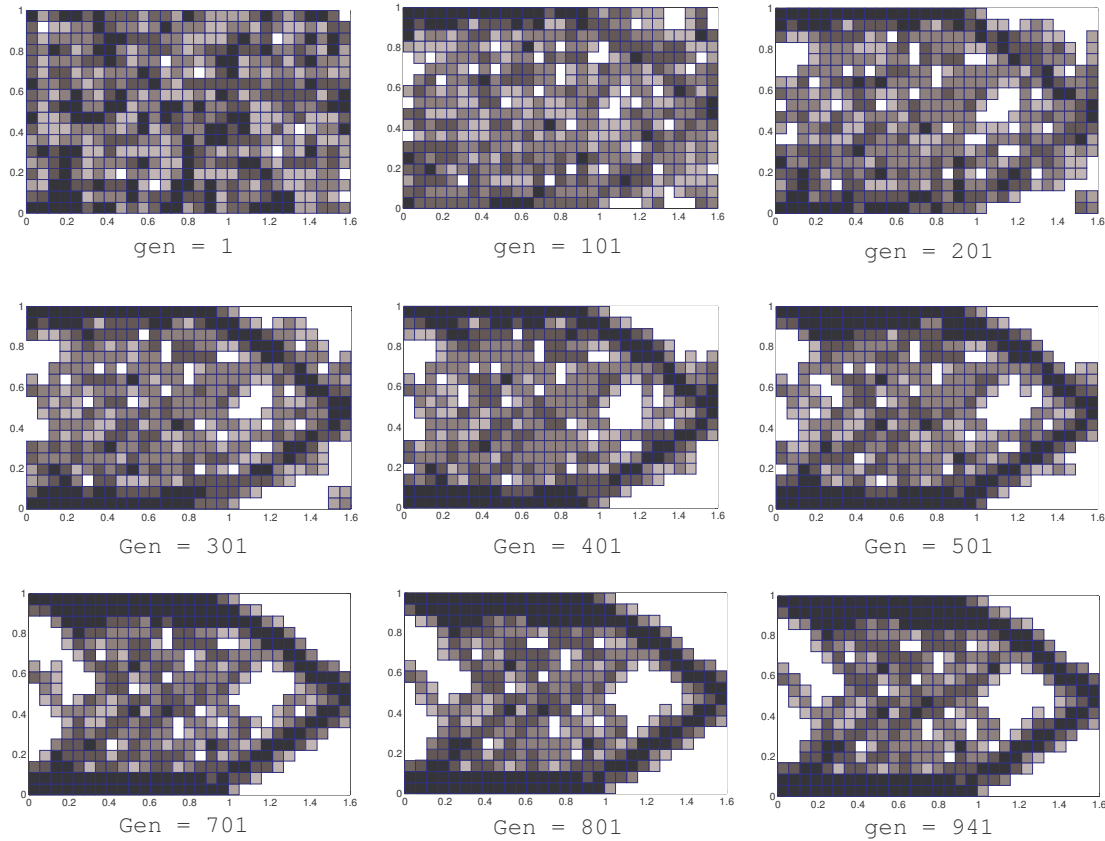


Figure 3-20. Évolution de la meilleure solution (Cantilever).

Sur la figure 3-21 nous présentons les historiques de la fonction objectif et de la contrainte sur le volume. La figure 3-21a montre une augmentation de la compliance dans les premières dizaines de générations. Cela s'explique par le fait que tous les individus sont inadmissibles. La compliance diminue ensuite au cours des générations pour se stabiliser à partir de la 800^{ème} génération. Le volume, qui représente la limitation, chute rapidement dans les premières générations. Sa valeur oscille ensuite autour de la limitation imposée et, en fin de calcul, se fixe à 49.97 %.

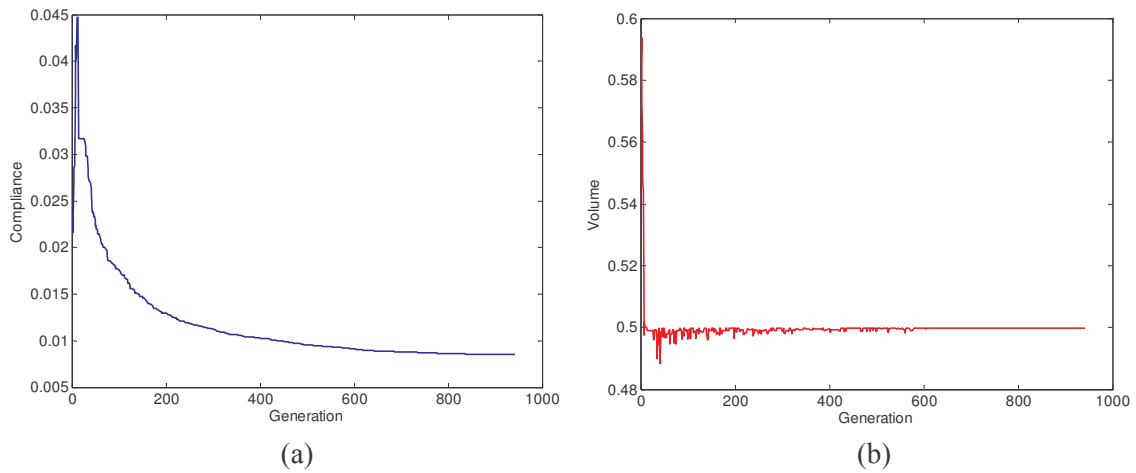


Figure 3-21. Poutre console – Historique : (a) de la fonction objectif, (b) des limitations.

8.3 Une structure de type Michell

Une application bien connue dans la littérature est la structure de type Michell [Xie97], ou treillis de Michell [Abd04]. Le domaine de conception est de forme rectangulaire, de dimensions $1 \times 0.5\text{m}^2$ ($L = 0.5\text{m}$). La structure est articulée sur le coin bas gauche et en appui simple sur le coin bas droit (Fig. 3-22). Une force ponctuelle égale à $4 \times 10^3 \text{ N}$ est appliquée au milieu du côté inférieur. Le domaine de conception est composé d'un matériau homogène élastique, isotrope et linéaire ayant un module d'Young E égal à 100 GPa. Le problème possède une solution théorique représentée par le treillis de la figure 3-22.

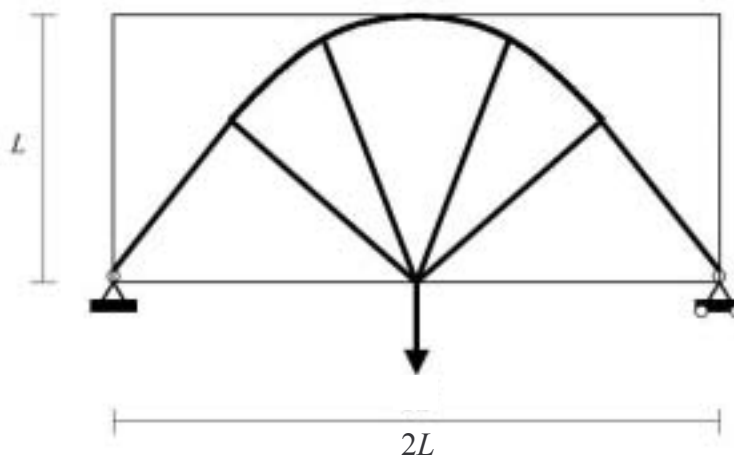


Figure 3-22. Une structure de type Michell (le domaine de conception et une solution théorique). [Abd04]

Le domaine de conception est discrétisé en 36×18 éléments finis. Le problème consiste à minimiser la compliance sous une limitation du volume égale à 50 % du volume initial.

La population, de taille fixe, est composée de 80 individus. Nous utilisons pour cet exemple un schéma de type Moor (Fig. 3-8b), la méthode de sélection SUS, un croisement à points de coupure multiples et une mutation uniforme de coefficients $Coef_c$ égal à 0.9 et de mutation $Coef_m$ égal à 0.2 respectivement.

L'historique de la meilleure solution est présenté sur la figure 3-23 qui montre une évolution de topologie similaire à celles des deux précédentes applications. Au début du calcul, la matière est répartie sur tout le domaine de conception mais avec une densité différente dans chaque cellule. En convergeant, deux zones commencent à apparaître, l'une de forte densité et l'autre de faible densité. La solution tend vers la solution théorique (Fig. 3-22) sous forme d'une demi-roue. On note cependant qu'il n'y a pas une correspondance exacte, les rayons de la roue n'étant pas parfaitement dessinés et l'écartement entre les rayons n'étant pas à l'identique de ceux de la solution théorique. Il s'agit néanmoins d'une solution satisfaisante dans la mesure où celle obtenue par une méthode déterministe avec ANSYS (Fig. 3-24) s'éloigne encore plus de la référence.

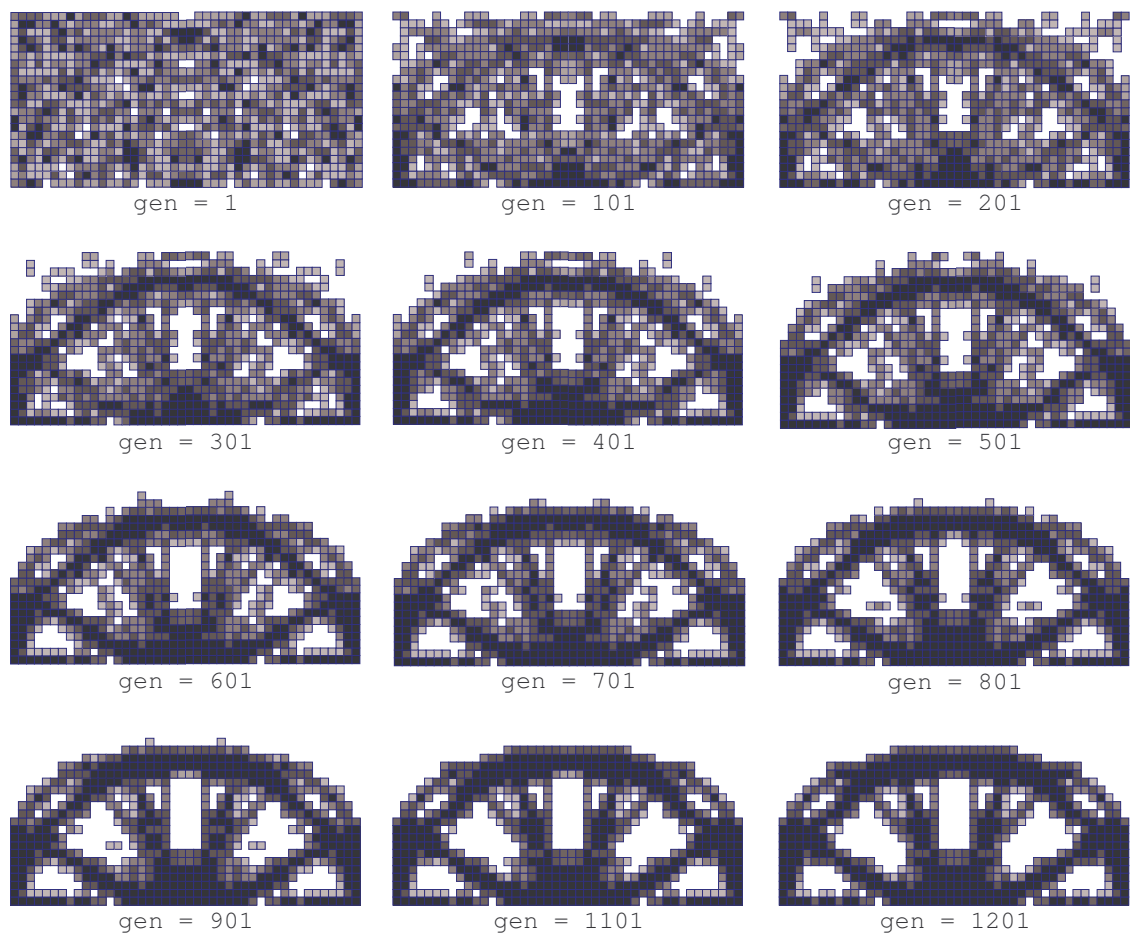


Figure 3-23. Évolution de la meilleure solution (structure de type Michell).

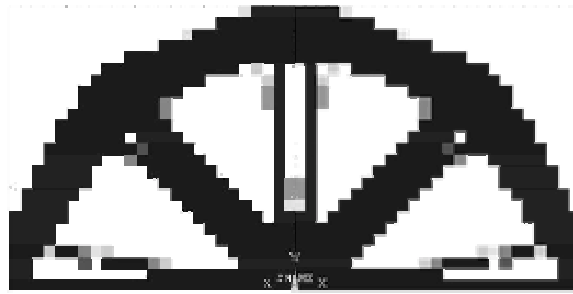


Figure 3-24. Solution obtenue par ANSYS (structure de type Michell).

La figure 3-25a montre l'historique de la compliance au cours des générations. La variation est significative lors des 600 premières générations. Cependant, les limitations, présentées sur la figure 3-25b montrent des oscillations importantes au début de calcul. Ces oscillations s'interprètent par le fait que l'algorithme recherche la distribution optimale de matière en testant de nombreuses possibilités. Ces oscillations, qui caractérisent l'ajout et l'élimination de la matière, diminuent dans les dernières générations, signe que la recherche s'est stabilisée sur une distribution optimale de matière.

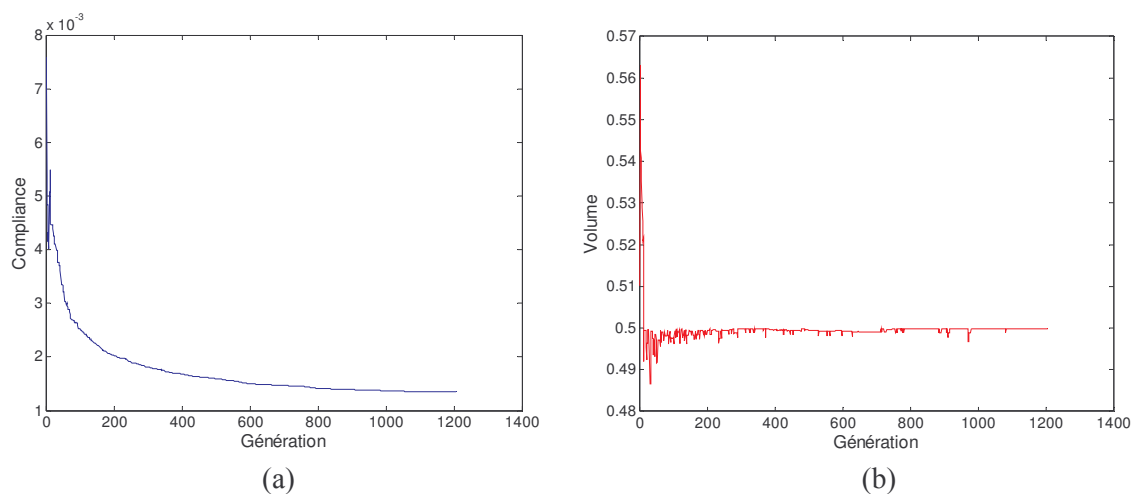


Figure 3-25. Structure de type Michell - Historique : (a) de la fonction objectif, (b) des limitations.

9. Conclusions

Dans ce chapitre, nous avons présenté une contribution originale qui propose une nouvelle représentation pour l'optimisation topologique des structures continues par algorithmes génétiques.

L'idée est inspirée partiellement de la méthode déterministe 'Automates cellulaires' [Kit00, Tat00, Tov04]. Les structures continues sont discrétisées par la méthode des éléments finis, où chaque élément est considéré comme cellule. Le codage des structures solution est basé sur une représentation par matrice binaire similaire à la matrice d'adjacence utilisée dans la théorie des graphes. La recherche de la distribution de la matière est basée sur le principe de la relaxation de densité souvent utilisé par la méthode d'homogénéisation SIMP [Sig01]. La variable de densité est altérée par l'opérateur de mutation en utilisant une technique inspirée de la méthode 'Automates cellulaires' [Tov04]. Cette technique joue un rôle important pour éliminer les damiers et les éléments flottants. La densité est utilisée ensuite pour altérer les bits du gène associé à la cellule mutante.

Des trois exemples que nous avons traités, deux ont fourni la solution attendue, ce qui valide notre approche. Dans l'exemple où la solution idéale n'a pas été obtenue, même si nous nous en sommes approchées, les densités intermédiaires empêchent l'algorithme de converger vers une distribution de matière avec une densité proche ou égale à 1. Cette inefficacité peut être expliquée par l'incompatibilité des opérateurs génétiques, notamment l'opération de mutation, avec l'algorithme génétique. En effet, l'opération de mutation est basée sur un concept qui provient de la méthode déterministe 'Automates cellulaires', ce qui nuit à cette efficacité.

L'approche que nous avons proposée partage des concepts avec les méthodes déterministes. Cela peut être un avantage comme on l'a vu avec la gestion des connexions entre éléments. Mais cela peut aussi constituer un inconvénient, notamment pour ce qui concerne la mise en œuvre de la mutation. C'est probablement dans cette direction, l'amélioration de la compatibilité de l'opération de mutation avec un concept issu de méthodes déterministes, qu'il faudrait œuvrer pour améliorer les performances de l'approche que nous avons présentée dans ce chapitre.

Conclusions générales

Les travaux présentés dans ce mémoire de thèse sont consacrés à la recherche d'algorithmes dédiés à l'optimisation topologique de structures continues. Les approches proposées sont basées sur un principe stochastique utilisant les algorithmes génétiques.

Le premier chapitre est consacré essentiellement à une étude bibliographique sur les méthodes d'optimisation topologique de structures. Après un bref historique sur l'optimisation structurale, les méthodes déterministes les plus connues sont présentées. La plus grande partie de ce premier chapitre est ensuite consacrée à l'étude de l'optimisation topologique des structures par algorithmes génétiques. Cette étude bibliographique montre qu'un nombre très modeste de travaux a été consacré aux algorithmes génétiques appliqués à l'optimisation topologique de structures continues. Cela peut s'expliquer par le fait que les algorithmes génétiques, malgré leur popularité, constituent un champ de recherche encore relativement jeune. Leur utilisation dans le contexte de l'optimisation topologique de structures reste donc encore marginal, contrairement aux méthodes déterministes dont l'exemple le plus mature est la méthode par homogénéisation.

En se basant sur le principe de présence ou d'absence de matière dans chaque élément fini, la majorité des travaux par algorithmes génétiques utilisent une représentation par tableau binaire. Mais le codage basé sur cette représentation souffre de sérieuses limitations (défaut de connectivité, dépendance par rapport au maillage,...). Afin de corriger les défauts de connectivité, on a souvent recours à des techniques dites d'analyse de connectivité [Jak00]. Cependant, en plus de leur coût élevé, ces techniques de réparation induisent souvent une dégénérescence des informations génétiques des chromosomes.

En raison de l'importance du codage, constaté en menant cette étude bibliographique, nous avons proposé une approche basée sur une représentation graphique par courbes de Bézier. Cette approche, proposée pour la première fois par Tai [Tai98], est exposée dans le deuxième chapitre. L'avantage de la représentation graphique est qu'elle assure la connectivité des éléments du maillage. Nous avons exploité cette caractéristique pour améliorer la performance de l'algorithme, notamment la minimisation du coût de calcul.

Cela a été réalisé en adoptant des techniques qui consistent à rechercher la solution optimale en rajoutant progressivement de la matière. En adoptant une stratégie de calcul en deux étapes (construction du squelette d'abord, ajout de chair ensuite) le gain du coût de l'analyse structurale par éléments finis est notable. Cependant, la représentation reste dépendante du schéma graphique et cela malgré toutes les améliorations que nous avons apportées : nouvelle définition des zones de support, segment de longueur variable et épaisseur de chair non uniforme de part et d'autre du squelette.

Dans le but de trouver une représentation générale, compatible avec n'importe quel problème d'optimisation topologique, nous avons proposé une deuxième approche dont les principes fondateurs diffèrent de ceux qui sont utilisés pour la représentation graphique. Cette deuxième approche, qui constitue une contribution originale, est présentée dans le troisième chapitre de ce mémoire. Elle est basée sur le principe de la connectivité des éléments entre eux et permet d'éviter l'emploi des techniques de réparation qui sont coûteuses. En considérant chaque élément fini comme cellule, la représentation des structures solutions est exprimée par une matrice d'adjacence binaire. La matrice est exploitée ensuite pour définir le codage, c'est-à-dire pour construire le chromosome. Chaque cellule est représentée par un gène dont la longueur est proportionnelle au nombre d'éléments de type liaison voisins à cette cellule. Le principe de la relaxation du matériau, à l'aide d'une densité artificielle, est utilisé ensuite pour la recherche de la distribution optimale de matière. L'opérateur de mutation, qui joue le rôle le plus important dans cette recherche, utilise une technique inspirée de la méthode déterministe appelée 'Automates cellulaires' [Tov04].

Cette deuxième approche a été testée sur plusieurs benchmarks simples en élasticité linéaire bidimensionnelle. Deux de ces benchmarks ont abouti, avec succès, à la solution attendue, ce qui valide l'approche proposée. Par contre, l'un des benchmarks testés n'a pas fourni la solution de référence bien qu'il ait permis d'améliorer la solution initiale. Cela s'explique sans doute par l'utilisation de concepts déterministes sans tenir compte des propriétés des algorithmes génétiques. En effet, l'opérateur de mutation, qui est inspiré de la méthode des automates cellulaires, est peu compatible avec l'algorithme génétique. En revanche, ce point commun, qui concerne le partage de quelques concepts entre les approches déterministes et notre approche, peut être exploité pour améliorer l'efficacité de cette dernière. Cela pourrait par exemple être réalisé en introduisant dans notre algorithme des techniques de filtrage utilisées par des méthodes déterministes. Des limitations sur le périmètre [Bec97, Duy96] pourraient aussi être introduites dans notre algorithme afin d'améliorer les résultats.

Suites à ces travaux nous avons tiré les conclusions suivantes :

- Les algorithmes génétiques constituent une voie prometteuse pour résoudre les problèmes d'optimisation topologiques de structures continues. Nous avons en effet résolu avec succès plusieurs benchmarks avec cette approche.
- Par rapport aux approches proposées dans la littérature, le codage par représentation graphique est l'un des moins coûteux. De plus, les modifications que nous lui avons apportées, notamment pour ce qui concerne la gestion de l'ajout de matière, permettent d'économiser encore plus de temps de calcul.
- Le codage par adjacence permet d'aborder de manière systématique n'importe quelle topologie, contrairement à la représentation graphique qui nécessite parfois l'élaboration d'un schéma graphique adéquat pour une convergence efficace. Le prix à payer pour ce caractère systématique est le coût élevé du temps de calcul.

En termes de perspectives, la continuité des travaux de recherche présentés dans ce mémoire pourrait être orientée dans les directions suivantes :

- Le codage par représentation graphique, présenté dans le deuxième chapitre, pourrait être étendu à des problèmes plus complexes que ceux traités dans ce mémoire : problèmes 3D et problèmes non-linéaires par exemple.
- Nous avons montré que la dépendance de l'algorithme par rapport au schéma graphique pouvait, dans certains cas, constituer une limitation dans la recherche de la solution. Surmonter cette limitation doit donc être une priorité pour de futurs travaux utilisant le codage par représentation graphique.
- Avec la représentation par adjacence, présentée dans le troisième chapitre, nous avons établi que le manque de compatibilité entre l'algorithme génétique d'une part et l'opérateur de mutation génétique défini sur la base d'approches déterministes d'autre part, nuisait à l'efficacité de l'approche. C'est donc prioritairement sur ce point que devront se focaliser de futurs développements.

BIBLIOGRAPHIE

- [Abd04] **M.M. Abdalla** and **Z. Gürdal**, Structural design using cellular automata for eigenvalue problems. *Structural and Multidisciplinary Optimization*, 26(3-4), 200–208, 2004.
- [Akh02] **S. Akhtar**, **K. Tai** and **J. Prasad**, Topology optimization of compliant mechanisms using evolutionary algorithm with design geometry encoded as a graph. *Proceeding of DETC'02, ASME 2002 Design Engineering Technical Conferences and computer and information in Engineering Conference, Montreal, Canada, September 29-October 2, 2002*.
- [All02a] **G. Allaire**, Shape optimization by homogenization method. *Springer, New York* (2002).
- [All02b] **G. Allaire**, **F. Jouve** and **A.M. Toader**, A level-set method for shape optimization. *C.R. Acad. Sci. Paris, série 1*, 2002.
- [Azi02] **I.A. Azid**, **A.S.K. Kwan** and **K.N. Seetharamu**, A AG-based technique for layout optimization of truss with stress and displacement constraints. *Int. J. Numer. Meth. Engng.*, 53 (2002), pp. 1641-1674.
- [Bak94] **J. E. Baker**, Reducing Bias and Inefficiency in the Selection Algorithms. in *Proceeding of the Second International Conference on Genetic Algorithms and their Applications, New Jersey, USA*, pp. 14-21, 1987.
- [Bar88] **B. Barthelemy** and **R.T. Haftka**, Accuracy Analysis of the Semi-Analytical Method for Shape Sensitivity Analysis. *Mechanics of Structures and Machines*, vol. 18, n°3, pp. 407-432, 1990.
- [Bat90] **J.-L. Batoz** et **G. Dhatt**, Modélisation des structures par éléments finis, volume 1 : solides élastiques. *Hermès, Paris*, 1990.
- [Bec97] **M. Beckers**, Optimisation de structures en variable discrètes. *Thèse de doctorat*, 1997, *Université de Liège, Belgique*.
- [Bec99] **M. Beckers**, Topology optimization using a dual method with discrete variables. *Struct. Optim.*, Vol. 17, pp. 14-24, 1999.
- [Ben78] **A. Bensoussan**, **J.L. Lions**, and **G. Papanicolaou**, Asymptotic analysis for periodic structures. *North-Holland*, 1978.
- [Ben88] **M.P. Bendsøe** and **N. Kikuchi**, Generating optimal topologies in structural design using a homogenization method. *Comp. Meth. Appl. Mech. Engng.*, vol. 71, (1988), pp. 197-224.
- [Ben89] **M.P. Bendsøe**, Optimal shape optimization as a material distribution problem. *Structural Optimization*, vol. 1, pp. 193-202, 1989.
- [Ben99] **M.P. Bendsøe** and **O. Sigmund**, Material interpolations in topology optimization. *Arch. Appl. Mech.*, vol. 69, pp. 635-654, 1999.

- [Ben02] **M.P. Bendsøe**, Recent developments in topology design of materials and mechanisms. ESAIM: Proceedings, Vol. 11, pp. 41-60, 2002.
- [Ben03] **M.P. Bendsøe** and **O. Sigmund**, Topology optimization: Theory, methods and applications. Springer-Verlag, New York (2003).
- [Cap03] **F. Cappello** and **A. Mancuso**, A genetic algorithm for combined topology and shape optimizations. *Computer-Aided Design*, vol. 35, issue 8, pp. 761-769.
- [Cer96] **R. Cerf**, An asymptotic theory of genetic algorithms. In: *J.-M. Aillot, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, editors, Artificial Evolution*, Vol. 1063 of *LNCIS*, pp. 37-53, Springer Verlag (1996).
- [Cha94] **C. Chapman, K. Saitou and M. Jakiela**, Genetic Algorithms as an approach to configuration and topology design. *ASME Journal of Mechanical Design*, 116(16), pp. 1005-1012, (1994).
- [Cha96] **C. Chapman and M. Jakiela**, Genetic algorithm-based structural topology design with compliance and topology simplification considerations. *ASME Journal of Mechanical Design*, 118(1), pp. 89-98, 1996.
- [Coe00] **C.A. Coello Coello and A.D. Christiansen**, Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, vol. 75, pp. 647-660, 2000.
- [Coe02] **C.A. Coello Coello**, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, vol. 191, N° 11-12, pp. 1245-1287, January 2002.
- [Dav89] **L. Davis**, Adapting operator probabilities in genetic algorithms. In *Proceeding of the 3rd international Conference on Genetic Algorithms*, Fairfax, VA, June 1989, pp. 61-69.
- [Deb89] **K. Deb and D.E. Goldberg**, An investigation of Niche and Species Formation in Genetic Function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Inc., San Mateo California, pp. 42-50.
- [Deb00] **K. Deb**, An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4) (2000) 311-338.
- [Dou01] **B.W. Douglas**, Introduction to Graph Theory. Second edition, Prentice Hall, 2001.
- [Dud97] **J. Duda and M. Jakiela**, Generation and classification of structural topologies with genetic algorithm speciation. *ASME Journal of Mechanical Design* 119 (1) (1997) 127-131.
- [Duy96] **P. Duysinx**, Optimisation Topologique : Du Milieu Continu à la Structure Elastique. *PhD dissertation, University of Liege, Belgium, Applied Sciences Faculty*, LTAS Report OF-37, (1996).

- [Eib99] **A.E. Eiben, R. Hinterding, and Z. Michalewicz**, Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, IEEE, 3(2), pp. 124-141, 1999.
- [Esc01] **H. A. Eschenauer and N. Olhoff**, Topology optimization of continuum structures: A review. *Applied Mechanics Reviews*, Vol. 54, No. 4, pp. 331–390, July 2001.
- [Fan02a] **D.W. Fanjoy and W.A. Crossley**, Topology design of planar Cross-sections with a genetic algorithm: Part 1- Overcoming the obstacles. *Engineering Optimization* Vol. 34(1) (2002) pp. 33-48.
- [Fan02b] **D.W. Fanjoy and W.A. Crossley**, Topology design of planar Cross-sections with a Genetic Algorithm: Part 2- Bending, Torsion and combined loading applications, *Engineering Optimization* Vol. 34(1) (2002) pp.49-64.
- [Gol87] **D.E. Goldberg and J. Richardson**, Genetic algorithm with shearing for multi-modal function optimization. In *J.J. Proceeding of the 2nd international Conference on genetic Algorithms*, pages 41-49, Lawrence Erlbaum Associates, 1987.
- [Gol89] **D.E. Goldberg**, Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley Reading (1989) Massachusetts*.
- [Gol91] **D.E. Goldberg**, A comparative analysis of selection schemes used in genetic algorithms. In *Gregory Rawlins, editor. Foundations of Genetic Algorithms*, pages 69-93, San Mateo, CA: Morgan Kaufmann Publishers, (1991).
- [Gol92] **D.E. Goldberg, K. Deb and J.H. Clark**, Genetic Algorithms, noise and the sizing of populations. *Complex systems*, 6, pp. 333-362, 1992.
- [Ham00] **H. Hamda, F. Jouve, E. Lutton, M. Schoenauer et M. Sebag**, Représentation non-structurée en optimisation topologique de forme par Algorithmes Evolutionnaires. *ESAIM Proceedings actes du 32^{ème} Congrès d'analyse Numérique CNUM 2000* Vol. 8, septembre 2000, 1-3.
- [Haj92] **P. Hajela**, Stochastic search in structural optimization: Genetic algorithms and simulated annealing. In *Structural Optimization: Status and Promise*, volume 150, pp. 611-637. Progress in Astronautics and Aeronautics, 1992.
- [Haj95] **P. Hajela and E. Lee**, Genetic Algorithms in Truss Topological optimization. *Int. J. Solids Structures* Vol. 32, N° 22 (1995), pp. 3341-3357.
- [Haj99] **P. Hajela and B. Kim**, GA based learning in cellular automata models for structural analysis. In *Proceedings of the 3rd Word Congress on Structural and Multidisciplinary Optimization*, Nigara Falls, NY.
- [Hig05] **D.J. Higham and N.J. Higham**, MATLAB guide. *Society for Industrial and Applied Mathematics*, 2nd edition, 2005, Philadelphia.
- [Hin95] **E. Hinton and J. Sienz**, Fully stressed topological design of structures using an evolutionary procedure. *Engineering computations*, 3, Vol. 12, pp. 229-244, 1995.
- [Hol75] **J. Holland**, Adaptation in natural and artificial systems, *University of Michigan Press, Ann Arbor*, 1975.

- [Jak97] **D. Jakobovic**, Adaptive genetic operators in elimination genetic algorithm. *Proc. 19th Conference ITI'97, Pula*, pp. 351-356, June 17-20, 1997,
- [Jak00] **M. Jakiela, C. Chapman, J. Duda, A. Adewuya and K. Saitou**, Continuum Structural topology design with Genetic Algorithms. *Comput. Methods Appl. Mech. Engrg* 186 (2000) 339-356.
- [Jen92] **E. Jensen**, Topological Structural Design using Genetic Algorithms. *PhD. Thesis* (1992) *Purdue University*.
- [Kan95] **C. Kane, F. Jouve and M. Schoenauer**, Structural Topology Optimization in linear and nonlinear elasticity using genetic algorithms. *21st ASME Design Automatic Conference, Boston MA, Sept. 1995*.
- [Kan96a] **C. Kane**, Algorithmes génétiques et optimisation topologique. *Ph D. Thesis, Université de Paris VI*, 1996.
- [Kan96b] **C. Kane and M. Schoenauer**, Topological Optimum Design using Genetic Algorithms. *Control and Cybernetics*, 25(5), pp. 1059-1088, (1996).
- [Kic05] **R. Kicinger, T. Arciszewski and K.A. De Jong**, Evolutionary computation and structural design: a survey of the state of the art. *Computers & structures*, 83 (23-24), 1943-1978, (2005).
- [Kim04] **Y.Y. Kim**, Topology optimization by element connectivity parameterization: why and what? Sixth World Congress on Computational Mechanics (WCCM VI) in conjunction with the Second Asian-Pacific Congress on Computational Mechanics (APCOM 04), Sept. 5-10, 2004, Beijing, China, Tsinghua University
- [Kir83] **S. Kirkpatrick, C. Gelatt and M. Vecchi**, Optimization by simulated annealing. *Science* 220 (1983) 671-680.
- [Kir93] **U. Kirsch**, Structural optimization: fundamentals and applications. *Springer-Verlag*, 1993, New York.
- [Kit00] **E. Kita and T. Toyoda**, Structural design using cellular automata. *Journal of Structural and Multidisciplinary Optimization*, (19) 64-73, 2000.
- [Lia00] **Q.Q. Liang, Y.M. Xie and G.P. Steven**, Optimal topology selection of continuum structures with displacement constraints. *Computers and Structures* 77(2000) 635-64.
- [Law91] **D. Lawrence**, Handbook of genetic algorithms. Van Nostrand Reinhold, New York, 1991.
- [Law03] **K.L. Lawrence**, ANSYS tutorial. Mission. KS 2003.
- [Mic1904] **A.G.M. Michell**, The limits of economy of material in frame-structures. *Philosophical Magazine*, s. 6, vol. 8, N°47, pp. 589-597, 1904.
- [Mic95] **Z. Michalewicz**, Genetic Algorithms, numerical optimization and constraints. In: *Eshelman L.J, editor, Proceedings of the 6th international conference on Genetic Algorithms. Los Atlos, CA: Morgan Kaufmann; 1995*, pp. 151-158.
- [Mic96] **Z. Michalewicz**, Genetic algorithms + data structures = evolution programs. *Springer-Verlag, Berlin* (1996).

- [Mle03] **H.P. Mlejnek** and **R. Schirrmacher**, An engineering approach to optimal material distribution and shape finding. *Comp. Meth. Appl. Mech. Engng.* 106 (1993) 1-26.
- [Ohs95] **M. Ohsaki**, Genetic algorithm for topology optimization of trusses. *Computers & Structures* Vol. 57 N° 2, (1995) pp. 219-225.
- [Olh91] **N. Olhoff**, **M.P. Bendsoe** and **J. Rasmussen**, On CAD-integrated structural topology and design optimization. *Computer Methods in Applied Mechanics and Engineering*, 89 (1991), pp. 259-279.
- [Pap96] **M. Papadrakakis**, **Y. Tsompanakis**, **E. Honton** and **J. Sienz**, Advanced solution methods in topology optimization and shape sensitivity analysis. *Eng. Comp.*, vol. 13, pp. 97-90, 1996.
- [Pen99] **C.A. Pena-Reyes** and **M. Sipper**, Evolutionary computation in medicine: an overview. *Artificial Intelligence in Medicine*, vol. 19, pp. 1–23, 2000.
- [Pra68] **W. Prager** and **J.E. Taylor**, Problems of Optimal Structural Design. *Journal of Applied Mechanics*, vol. 35, pp.102-106, 1968.
- [Rom84] **C. Romesburg**, Cluster analysis for researches, Lifetime Learning Publications (Wadsworth), Belmont California, 1984.
- [Roz88] **G.I.N. Rozvany** and **B.L. Karihaloo**, Structural optimization. Kluwer Academic Publishers, London, 1988.
- [Roz01a] **G.I.N. Rozvany**, Aims, scope, methods, history unified terminology of computer-aided topology optimization in structural mechanics. *Struct. Multidisc. Optim.* Vol. 21, pp. 90-108, 2001.
- [Roz01b] **G.I.N. Rozvany**, Stress ratio and compliance based methods in topology optimization - a critical review, *Struct. Multidisc. Optim.* 21, 109-119 (2001).
- [Sai05] **K. Saitou**, **K. Izui**, **S. Nishiwaki** and **P. Papalambros**, A survey of Structural Optimization in Mechanical Product development. *Journal of computing and information Science in Engineering*, Vol. 5(3), pp. 169-263, 2005.
- [San90] **E. Sanderger**, **E.D. Jensen** and **J. Welton**, Topological design of structural components using genetic optimization methods. *In Sensitivity Analysis and optimization with Numerical Methods, AMD*, vol. 115, *Proceeding of Winter Annual Meeting of the American Society of Mechanical Engineers*. Dallas, TX, 31-43, 1990.
- [Sav85] **M. Save** and **W. Prager**, Structural optimization: v.1. Optimality criteria, v.2. Mathematical programming. *Plenum Press*, New York, 1985.
- [Sch63] **L.A. Schmit** and **R.A. Malett**, Structural synthesis and design parameter hierarchy. *Journal of the Structural Division*, ASCE, vol 89, ST4, pp 269-299, 1963.
- [Sch95] **M. Schoenauer**, Shape representation for evolutionary optimization and identification in structural mechanisms. *in Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galain and P. Cuesta, Eds. Chichester John wiley, (1995) pp. 443-464.

- [Su83] **B. Su and D. Lu**, An affine invariant and its application in computational geometry. *Scientia Sinica (Series A)* 24, 3 (Mar. 1983), 259-267.
- [Sva87] **K. Svanberg**, The method of moving asymptotes - A New Method for *Structural Optimization*. *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 359-373, 1987.
- [Sva87] **K. Svanberg**, Optimization of geometry in truss design. *Computer Methods in Applied Mechanics and Engineering*, vol. 28, pp. 63-80, 1981.
- [Sig01] **O. Sigmund**, A 99 line topology optimization code written in Matlab. *Struct. Multidisc. Optim.*, 21 (2001) 120-127.
- [Spe92] **W.M. Spears**, Adapting crossover in a genetic algorithm. *Technical Report AIC-92-025, DC: Novel Research Laboratory, Navy Center for applied Research in Artificial Intelligence, Washington*, 1992.
- [Sri94] **M. Srinivas and L.M. Patnaik**, Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transaction son System, Man and Cybernetics*, 24(4) (1994) 656-667.
- [Su83] **B.Q. Su, D.Y. Liu**, Computational geometry, curve and surface modeling. *Academic Press*, 1983.
- [Tat00] **B. Tatting and Z. Gürdal**, Cellular automata for design of tow-dimensional continuum structures. *In proceeding of the 8th AIAA/USAF/NASA/ISSMO Symposium of Multidisciplinary Analysis and Optimization*, pages 2000-4832, Long Beach, CA, 2000, AIAA Paper.
- [Tai98] **K. Tai and T.H. Chee**, Genetic algorithm with structural morphology representation for topology design optimization. *in Gentle, C.R. and Hull, J.B., eds., Mechanics in Design Proceedings of the 2nd International Conference on Mechanics in Design*, Nottingham Trent University, pp. 827-836, 1998.
- [Tai00] **K. Tai and T.H. Chee**, Design of structures and compliant mechanisms by evolutionary optimization of morphological representations of topology. *ASME Journal of Mechanical Design*, 122 (2000) 560-566.
- [Tai02] **K. Tai, G.Y. Cui, and T. Ray**, Design synthesis of path generating compliant mechanisms by evolutionary optimization of topology and shape. *ASME Journal of Mechanical Design*, 124 (2002) 492-500.
- [Tan02] **P. Tanskanen**, The evolutionary structural optimization method - theoretical aspects. *Comput. Methods Appl. Mech. Engrg.*, 191 (2002) 5485-5498.
- [Tov04] **A. Tovar, N. Patel, A.K. Kaushik, G.A. Letona and J.E. Renaud**, "Hybrid cellular automata: a biologically-inspired structural optimization technique", 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 30 August – 1 September 2004, Albany, New York.
- [Val95] **C.L. Valenzuela**, Evolutionary Divide and Conquer: a novel genetic approach to the TSP. PhD thesis, Imperial College, University of London, London, England, 1995.

- [Wan81] **C.Y. Wang**, Shape classification of the parametric cubic curve and parametric B-Spline cubic curve. *Computer Aided Design*, Vol. 13, N°4, pp. 199-206, 1981.
- [Wan03] **Y. Wang**, A study on microstructures of homogenization for topology optimization. Coursework Master Thesis, Victoria University, 2003.
- [Wan03] **M.Y. Wang, X. Wang and D. Guo**, A level set method for structural topology optimization. *Compt. Methods Appl. Mech. Engrg.* 192 (2003) 227-246.
- [Wan04] **S.Y. Wang and K. Tai**, structural topology optimization using genetic algorithms with bit-array representation. *Computer Methods in Applied Mechanics and Engineering*, 2004, in press.
- [Wu95] **S.-J. Wu and P.-T. Chow**, Integrated discrete and configuration optimization of trusses using genetic algorithms. *Computers & Structures*, Vol. 55, N° 4, (1995), pp. 695-702.
- [Xie93] **Y.M. Xie and G.P. Steven**, A simple evolutionary procedure for structural optimization. *Computers & Structures*, vol. 49, pp. 885-896, 1993.
- [Xie97] **Y.M. Xie and G.P. Steven**, Evolutionary Structural Optimization. *Springer, New York*, 1997.
- [Yam88] **F. Yamaguchi**, Curves and Surfaces in Computer Aided Geometric Design. *Springer-Verlag, New York* 1988.
- [Yan03] **R.J. Yang and C.H. Chuang**, Optimal topology design using linear programming. *Comput. Struct.*, 52(2) (1994) 265-275.
- [Zho01] **M. Zhou and G.I.N. Rozvany**, On the Validity of ESO type methods in topology optimization, *Struct. Multidisc. Optim.* 21 (2001) 80-83.

Annexe

Résolution d'équations polynomiales cubiques par les formules de Cardan

Nous présentons dans cette annexe les célèbres formules de Cardan pour la résolution de l'équation cubique (A-1). Si Cardan était un mathématicien, c'était aussi un médecin et un mécanicien célèbre puisqu'il inventa la pièce mécanique qui porte son nom. Les formules de Cardan, publiées dans l'ouvrage mathématique l'*Ars Magna* en 1545, ont été utilisées dans ce travail de thèse (chapitre II) pour la construction du squelette.

Soit l'équation cubique :

$$\begin{aligned} ax^3 + bx^2 + cx + d &= 0 \\ \text{avec } a, b, c, d &\in \mathbb{R} \end{aligned} \quad (\text{A-1})$$

En posant :

$$x = X - \left(\frac{b}{3a} \right) \quad (\text{A-2})$$

On ramène l'équation (A-1) à la forme :

$$X^3 + pX + q = 0 \quad (\text{A-3})$$

où les nouveaux paramètres p et q sont définis en fonction des racines a , b , c et d par :

$$p = b - \frac{a^2}{3} ; q = \frac{2a^3}{27} - \frac{ab}{3} - c \quad (\text{A-4})$$

Les racines de l'équation (A-3) sont données par les formules de Cardan :

$$X_1 = A + B \quad (\text{A-5a})$$

$$X_2 = \alpha_1 A + \alpha_2 B \quad (\text{A-5b})$$

$$X_3 = \alpha_2 A + \alpha_1 B \quad (\text{A-5c})$$

Dans lesquelles :

$$A = \sqrt[3]{-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}, \quad B = \sqrt[3]{-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}; \quad (\text{A-6})$$

$$\alpha_1 = \frac{-1 + i\sqrt{3}}{2}, \quad \alpha_2 = \frac{-1 - i\sqrt{3}}{2}; \quad (\text{A-7})$$

$$\Delta = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 \quad (\text{A-8})$$

Si $\Delta < 0$, il y a trois racines réelles ;

Si $\Delta = 0$, il y a trois racines réelles, dont deux sont confondues (égales à $-\frac{3q}{2p}$), la troisième étant distincte des deux premières (égale à $\frac{3q}{p}$) ;

Si $\Delta > 0$, il y a une racine réelle et deux racines complexes conjuguées.