



Développement d'une nouvelle méthode metaheuristique pour l'optimisation topologique des structures et des metamatériaux

Noëlie Di Cesare

► To cite this version:

Noëlie Di Cesare. Développement d'une nouvelle méthode metaheuristique pour l'optimisation topologique des structures et des metamatériaux. Mécanique des matériaux [physics.class-ph]. Université de Technologie de Belfort-Montbéliard, 2016. Français. NNT : 2016BELF0299 . tel-01576611v2

HAL Id: tel-01576611

<https://hal.science/tel-01576611v2>

Submitted on 24 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

Développement d'une nouvelle méthode metaheuristique pour l'optimisation topologique des structures et des metamatériaux

■ NOËLIE DI CESARE

SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

N° 2 | 9 | 9

THÈSE présentée par

NOËLIE DI CESARE

pour obtenir le

Grade de Docteur de

l'Université de Technologie de Belfort-Montbéliard

Spécialité : **Mécanique**

Développement d'une nouvelle méthode metaheuristique pour l'optimisation topologique des structures et des metamatériaux

Unité de Recherche :

Institut de Recherche sur les Transports l'Energie et la Société

Soutenue publiquement le 28/11/16 devant le Jury composé de :

PIERRE DUYSINX	Rapporteur	Professeur à l'Université de Liège, Belgique
ZHI-QIANG FENG	Rapporteur	Professeur à l'Université d'Evry-Val d'Essonne
GEORGES FADEL	Examineur	Professeur à l'Université de Clemson, SC, USA
NICOLE COUTRIS	Examineur	Professeure à l'Université de Clemson, SC, USA
SYLVIE RONEL	Examineur	Professeure à l'Université Lyon 1
CATHERINE VAYSSADE	Examineur	Ingénieure de recherche CNRS à l'Université de Technologie de Compiègne
MATTHIEU DOMASZEWSKI	Directeur de thèse	Professeur à l'Université de Technologie de Belfort-Montbéliard

REMERCIEMENTS

En premier lieu, je tiens à remercier mon directeur de thèse Professeur Matthieu DOMASZEWSKI pour tout ce qu'il m'a apporté sur les plans scientifique, professionnel et personnel. Il est - en toute objectivité - le meilleur directeur de thèse qu'un doctorant puisse avoir. Ses méthodes d'encadrement m'ont permis de progresser grandement et d'améliorer continuellement mes compétences scientifiques. Sa gentillesse m'a permis de m'épanouir pendant le marathon que représente les trois années de thèse.

Je remercie les membres du jury Mme Catherine VAYSSADE, M. Georges FADEL, Mme Nicole COUTRIS, Mme Sylvie RONEL et les rapporteurs M. Zhi-Qiang FENG et M. Pierre DUYSINX d'avoir accepté d'évaluer mon travail. En outre, un remerciement spécial va à Mme COUTRIS et M. FADEL pour m'avoir accueillie dans leur laboratoire de l'Université de Clemson aux Etats-Unis. Ce fût pour moi une expérience extraordinairement enrichissante tant sur le plan scientifique que sur le plan personnel.

Je tiens également à remercier M. Sébastien ROTH, chef de notre équipe de travail, pour ses encouragements, ses qualités pédagogiques et humaines, sa sympathie et l'aide qu'il m'a apportée tout au long de la réalisation de ce travail. Je remercie aussi M. François PEYRAUT et Mme Nadia LABED pour l'indispensable aide qu'ils m'ont apportée sur le plan scientifique. J'envoie aussi mes plus sincères remerciements à M. Kamel MAZOUZI, responsable du Mésocentre de calcul de Franche-Comté, pour l'aide précieuse qu'il m'a apportée, et la disponibilité à toute épreuve dont il a fait preuve.

Je remercie également tous les membres des laboratoires M3M et LERMPS de l'institut IRTES pour l'aide qu'ils m'ont apportée et leur soutien tout au long de ces trois années de travail. Un remerciement spécial va à Mme Béatrice ROSSEZ pour sa bonne humeur naturelle et la joie qu'elle nous apporte à tous au laboratoire.

Je remercie aussi ici toute la communauté des utilisateurs de Cast3m qui a répondu à mes questions et m'a beaucoup aidée dans la réalisation des travaux concernant la modélisation numérique des metamatériaux mécaniques.

Un immense merci à mes collègues et amis doctorants : Aristide AWOUKENG, Lorenzo TADDEI, Michèle BODO, Timothée ZUSSY, Renye CAI, Jean-Baptiste DEVILLERS, Homam ISSA, Anh-Tuan TA, Abdelhakim SETTAR et Caminde REZAI, pour leur soutien, leur amitié, les cafés partagés. Un remerciement spécial va à Lorenzo en particulier pour l'aide indispensable qu'il m'a apportée sur le plan scientifique.

Je remercie toute ma famille et mes amis, qui m'ont donné la chance d'arriver là, qui m'ont accueillie, soutenue, épaulée et aidée de manière absolument inconditionnelle. Un remerciement spécial va à mon frère Samuel, mon père, ma belle-mère Sylvie, ma mamie Marcelle, mes tantes Marie-Ange et Michelle, mes oncles Jean-Marie et Fred, tous mes cousins, cousines et petits cousins. Merci aussi à toute ma belle-famille, et en particulier Françoise, Gérard, Béa, Christophe et Célia.

Je remercie enfin les amis qui sont devenus des parties intégrantes de ma famille : Nico-

lou et toute sa famille, Poupi et ses enfants, Cath et ses enfants, Laureline, Nils, Bachelor et Audrey. Et bien sûr notre merveilleuse Fouzia.

Un immense merci à Bruno, pour m'avoir toujours encouragée et soutenue. Merci de rendre ma vie plus belle et plus heureuse chaque jour.

Cette thèse aura été réalisée avec une pensée pour mon grand-père Jean, mes grands-parents Pasqualino et Silvana, mon oncle Pierrot, et mon ami Rémi.

SOMMAIRE

1	Introduction générale	1
2	Revue de la littérature et bases théoriques	5
2.1	Optimisation	7
2.1.1	Optimisation difficile	7
2.1.1.1	Cas de l'optimisation difficile : définition	7
2.1.1.2	Classification des méthodes d'optimisation	9
2.1.2	Optimisation multiobjectif	9
2.1.2.1	Définition du problème	9
2.1.2.2	Notion de dominance	11
2.1.2.3	Méthodes d'optimisation multiobjectif	12
2.1.3	L'optimisation en mécanique des structures	17
2.1.3.1	Généralités	17
2.1.3.2	Optimisation topologique	19
2.2	Méthodes metaheuristiques	23
2.2.1	Généralités	23
2.2.1.1	Propriétés	24
2.2.1.2	Avantages et inconvénients	25
2.2.1.3	Classification des méthodes metaheuristiques	27
2.2.2	Quelques méthodes metaheuristiques	27
2.2.2.1	Ant Colony Optimization	27
2.2.2.2	Les algorithmes évolutionnaires	33
2.2.3	Focus on : Particle Swarm Optimization	42
2.2.3.1	Historique, principe de base	42
2.2.3.2	Topologies de la population	45
2.2.3.3	Principales variantes et améliorations	53
2.3	Notions concernant la théorie des graphes	70
2.3.1	Introduction	70
2.3.2	Généralités concernant la théorie des graphes	70

2.3.2.1	Définitions	72
2.3.2.2	Introduction aux chaînes de Markov	76
2.3.3	L'algorithme PageRank	77
2.3.3.1	Généralités, définition	77
2.3.3.2	Calcul du PageRank	78
2.3.3.3	Calcul matriciel	81
2.3.3.4	Utilisation originale et intéressante de PageRank dans la littérature	83
2.4	Metamatériaux et notions d'homogénéisation	84
2.4.1	Les metamatériaux	84
2.4.2	Homogénéisation	87
2.4.2.1	Principes de changement d'échelle : passage du milieu hétérogène au milieu homogène	87
2.4.2.2	Méthode des champs moyens	90
2.4.2.3	De l'importance des conditions aux limites	94
2.5	Conclusions de l'étude bibliographique	96
3	Développement d'un nouvel algorithme d'optimisation metaheuristique in- telligent	99
3.1	Introduction	100
3.2	Algorithme PageRank inverse	101
3.3	Utilisation de PageRank inverse dans PSO	103
3.3.1	Analogie avec les chaînes de Markov	103
3.3.2	Définition de l'algorithme I-PR-PSO	105
3.3.3	Exemples de calculs réalisés et problématiques rencontrées	108
3.4	Validation sur un benchmark de fonctions mathématiques	109
3.4.1	Paramètres de l'algorithme et fonctions testées	109
3.4.2	Résultats obtenus	111
3.4.3	Pour conclure	115
3.5	Validation en mécanique	115
3.5.1	Présentation du problème	115
3.5.2	Benchmark de treillis et paramètres du calcul	116
3.5.2.1	Treillis à 10 barres	116
3.5.2.2	Treillis à 22 barres	116
3.5.2.3	Paramètres mécaniques et numériques	118
3.5.3	Gestion des contraintes d'optimisation	118

3.5.3.1	Les bornes limites des variables d'optimisations	119
3.5.3.2	Les contraintes intrinsèques au problème	120
3.5.4	Résultats obtenus	120
3.5.4.1	Treillis à 10 barres	120
3.5.4.2	Treillis à 22 barres	122
3.5.5	Pour conclure	122
3.6	Conclusions du chapitre	123
4	Développement d'une méthode d'optimisation metaheuristique hybride : application à l'optimisation topologique en mécanique	125
4.1	Introduction	126
4.2	Définition du nouvel algorithme ESO-I-PR-PSO	126
4.2.1	Binarisation de l'algorithme I-PR-PSO	126
4.2.2	Analyse des damiers dans la topologie	128
4.2.3	Algorithme ESO-I-PR-PSO global	131
4.2.4	Définition des problèmes d'optimisation à résoudre	132
4.3	Choix des paramètres de I-PR-PSO	134
4.3.1	Analyse des paramètres de confiance c_1 et c_2	135
4.3.2	Analyse du paramètre d'inertie ω	137
4.3.3	Analyse du nombre de particules de l'essaim	140
4.3.4	Conclusions de l'étude paramétrique	142
4.4	Problèmes traités, résultats et discussion	142
4.5	Conclusions du chapitre	154
5	Optimisation des metamatériaux électromagnétiques : développement d'une cape d'invisibilité électromagnétique fréquentielle	155
5.1	Introduction	156
5.2	Définition du problème d'optimisation à résoudre	158
5.2.1	Définition du modèle numérique	158
5.2.2	Equations gouvernant le problème	160
5.2.3	Calcul de la perméabilité effective	161
5.3	Formulation du problème d'optimisation	163
5.4	Algorithme de résolution : Utilisation d'une version binaire de PSO	164
5.5	Implémentations numériques	166
5.5.1	Généralités concernant le modèle numérique et les unités employées	166
5.5.2	Exemples numériques	167

5.5.2.1	Minimisation de la perméabilité effective à 10 THz	167
5.5.2.2	Minimisation de la perméabilité effective à 15 THz	170
5.6	Conclusions du chapitre	172
6	Optimisation des metamatériaux mécaniques : développement d'un meta-matériau hyperélastique	175
6.1	Introduction	176
6.2	Présentation du problème	176
6.3	Considérations numériques	176
6.3.1	Définition du modèle numérique de la cellule de périodicité	177
6.3.2	Connectivité des cellules et continuité de la matière	179
6.3.3	Grands déplacements et notion d'hypoélasticité	181
6.3.3.1	Notions générales concernant la mécanique des milieux continus	181
6.3.3.2	Notion de référentiel, de repère	182
6.3.3.3	Repère corotationnel et dérivée de Jaumann	184
6.4	Définition du problème d'optimisation	185
6.5	Définition du processus d'optimisation	187
6.6	Investigations numériques et résultats obtenus	188
6.6.1	Etude du premier cas de calcul	188
6.6.2	Etude du second cas de calcul	194
6.6.3	Etude du troisième cas de calcul	200
6.6.4	Etude du quatrième cas de calcul	205
6.6.5	Recherche de la totalité de la courbe $\sigma - \epsilon$ en compression	211
6.7	Conclusions du chapitre	213
7	Conclusions de la thèse	215
7.1	Bilan des travaux réalisés	215
7.2	Perspectives et améliorations	217
A	Etats de sollicitations	253
B	Courbes de convergence obtenues avec l'algorithme I-PR-PSO	257
C	Résultats complémentaires : courbes de convergence des calculs d'optimisation topologique réalisés avec ESO-I-PR-PSO	265
C.1	Courbes de convergence de tous les problèmes d'optimisation traités . . .	265
C.2	Résultats d'optimisation topologique réalisés avec PSO seul	278

D Résultats complémentaires : Valeurs atteintes par l'algorithme ESO-I-PR-PSO	281
E Exemple de modèle numérique de VER défini avec Cast3m	283

INTRODUCTION GÉNÉRALE

L'optimisation offre la possibilité, dans de nombreux domaines, d'améliorer les performances d'un système donné, qu'il soit physique ou mathématique. Elle est d'ailleurs très utilisée dans l'industrie afin d'améliorer les performances de systèmes complexes, ou de minimiser les coûts, en améliorant soit la structure elle-même (minimisation de la masse de la structure), soit les chaînes de production (minimisation des temps de fabrication). La minimisation, aussi minime soit-elle, d'une fonction objectif bien choisie (au sens où le problème mathématique à résoudre reste bien posé) permet de faire de grosses économies, et c'est la raison pour laquelle l'optimisation est non seulement prisée par les entreprises industrielles, mais aussi un domaine de recherche très actif ces dernières années. Avec l'avènement des nanotechnologies, les innovations récentes découlent très souvent du développement de nouveaux matériaux. C'est ainsi que l'optimisation topologique permettra le développement de matériaux innovants, et plus particulièrement de metamatériaux, c'est à dire des matériaux dont les propriétés physiques exceptionnelles n'existent pas à l'état naturel.

Contexte général de la thèse et motivations

Cette thèse apporte une contribution aux travaux de recherche du laboratoire de mécanique de l'Université Bourgogne Franche-Comté, et a été réalisée à l'Université de Technologie de Belfort-Montbéliard.

Du grec *meta* signifiant "un niveau supérieur", les metaheuristiques, comme les metamatériaux, sont des innovations scientifiques dont les propriétés, respectivement numériques ou physiques, peuvent être qualifiées d'exceptionnelles, notamment parce qu'elles présentent des innovations majeures et/ou des améliorations de l'existant.

En effet, depuis plusieurs années, les méthodes metaheuristiques se sont imposées dans le domaine de l'optimisation, combinatoire ou continue, de par leur capacité à résoudre les problèmes d'optimisation dits difficiles. Leurs extraordinaires propriétés leur permettent d'être réellement compétitives quant à la résolution de ces problèmes. Plus récemment, ces méthodes ont eu vocation à évoluer, à se perfectionner, afin d'être toujours plus efficaces. Ainsi, de nombreuses méthodes, tout d'abord définies de manière immuable, ont évolué vers des méthodes adaptatives, voire intelligentes.

Cependant, les méthodes d'optimisation sont très souvent couplées à un calcul réalisé par la méthode des éléments finis, qui calculera la valeur de la fonction objectif pour chaque itération d'optimisation. Ces calculs sont souvent très coûteux en terme de temps

de calcul. Il est donc nécessaire de tenter de minimiser tout de même les temps de calculs globaux, en minimisant le temps de calcul de l'optimisation. Ainsi, de nouvelles méthodes sont proposées chaque année dans la littérature, permettant soit de trouver une meilleure solution à un problème d'optimisation donné, soit de minimiser le temps de calcul global, et donc le coût du calcul.

D'autre part, le domaine des metamatériaux est un axe de recherche important ces dernières années, tant dans le domaine de la mécanique, que dans les domaines de l'électromagnétisme, l'acoustique, ou encore la physique fondamentale.

Dans le domaine de l'électromagnétisme, alors que les propriétés physiques des metamatériaux électromagnétiques sont étudiées depuis des dizaines d'années, les recherches ont pris un nouveau tournant en 2015 avec le développement, et la fabrication, d'une véritable cape d'invisibilité par une équipe américaine de l'Université de Californie, et publiée dans la revue Science du 18 septembre 2015¹ [245].

Dans le domaine de la mécanique, les recherches concernant les metamatériaux sont relativement récentes. En effet, ce n'est qu'à la fin des années 80 que l'on voit apparaître le concept de matériau à coefficient de Poisson négatif dans la littérature. Depuis, les recherches ont beaucoup avancé dans ce domaine, puisque la communauté scientifique est aujourd'hui capable de fabriquer ce type de matériau, notamment sous la forme de mousses. En revanche, de nouvelles propriétés mécaniques peinent à voir le jour.

L'utilisation de l'optimisation, topologique en particulier, s'avère donc nécessaire au développement de nouveaux matériaux aux propriétés physiques inconnues jusqu'alors, ou n'existant pas dans la nature.

Objectifs des travaux

Développer de nouveaux metamatériaux, via l'utilisation de méthodes metaheuristiques innovantes. Le développement de nouveaux matériaux est un problème d'optimisation topologique complexe, qui doit permettre le développement de nouvelles propriétés physiques, jusqu'alors inconnues, ou meilleures que celles présentées précédemment dans la littérature. Dans ce domaine, l'utilisation des méthodes d'optimisation traditionnelles, c'est à dire utilisant le calcul de gradient pour les sensibilités de la fonction objectif aux variables d'optimisation, ne permet généralement pas de trouver le minimum global de la fonction objectif, notamment lorsque celle-ci présente de nombreux minima locaux ou est non différentiable. En outre, les calculs d'optimisation peuvent être très coûteux en terme de temps de calcul, notamment du fait du calcul de la fonction objectif qui est généralement réalisé via la méthode des éléments finis. C'est ainsi qu'il sera proposé une nouvelle méthode d'optimisation, capable non seulement de trouver le minimum global de fonctions objectifs complexes² non différentiables, mais aussi d'améliorer le temps de calcul, en convergeant plus rapidement que ses congénères.

Cet algorithme sera inspiré de l'algorithme d'optimisation par essaim particulaire (appelé [Particle Swarm Optimization](#) dans la littérature anglophone) PSO [29] [60] et de l'algorithme PageRank utilisé par le moteur de recherche Google ® pour classer les pages web, et couplé avec des calculs éléments finis (EF) permettant de calculer, pour chaque

1. Source : lemonde.fr

2. Le terme 'complexe' sera ici entendu en terme de difficulté, et non pas en terme de complexité mathématique

particule, la valeur de la fonction objectif.

Concernant les metamatériaux, le travail consistera notamment au développement de nouveaux matériaux, mécaniques et électromagnétiques, présentant de nouvelles propriétés physiques jusqu'alors inconnues, ou tout du moins inhabituelles.

Avant d'appliquer notre nouvel algorithme d'optimisation à l'étude des metamatériaux, celui-ci sera validé en mathématique sur des fonctions objectifs classiques proposées dans la littérature et en mécanique sur des exemples de treillis classiques.

Plan de présentation de la thèse

La suite de ce rapport de thèse est organisée en cinq chapitres de la manière suivante :

Le chapitre 2 sera consacré à la présentation d'une revue bibliographique et des bases théoriques nécessaires à l'élaboration de ce travail de thèse. Dans la première partie de ce chapitre, nous présenterons le domaine de l'optimisation numérique. Les concepts d'optimisation difficile, d'optimisation multiobjectif et d'optimisation topologique seront détaillés. La seconde partie proposera une étude bibliographique concernant les méthodes d'optimisation metaheuristiques. On présentera notamment brièvement l'algorithme d'optimisation par colonies de fourmis, ainsi que les algorithmes évolutionnaires, avant de détailler plus précisément le fonctionnement de l'algorithme d'optimisation par essais particuliers PSO, dont sera inspiré par la suite le développement de notre algorithme d'optimisation. La troisième partie proposera les bases mathématiques de la théorie des graphes et en particulier de l'algorithme PageRank utilisé dans ce travail de thèse. Finalement, la dernière partie proposera une étude bibliographique des méthodes d'homogénéisation utilisées dans le domaine des metamatériaux.

Le chapitre 3 détaillera le développement de notre algorithme d'optimisation Inverse-PageRank-PSO (I-PR-PSO), ainsi que sa validation dans les domaines des mathématiques et de la mécanique, après avoir déterminé les paramètres les plus adaptés à sa convergence rapide et efficace.

Le chapitre 4 proposera de développer une version binaire de I-PR-PSO couplée avec l'algorithme *Evolutionary Structural Optimization* ESO et de la valider en optimisation topologique sur des problèmes mécaniques classiques de la littérature. Ce travail nécessitera notamment le développement d'algorithmes spécifiques de vérification des topologies obtenues, afin de vérifier qu'elles ne présentent pas de damiers. Ce travail proposera de résoudre simultanément deux objectifs apparemment contradictoires, en les considérant de manière séquentielle au long du calcul.

Le chapitre 5 proposera une investigation de l'optimisation des metamatériaux électromagnétiques avec l'algorithme PSO. On détaillera notamment les équations employées pour résoudre un tel problème électromagnétique, ainsi que les méthodes numériques employées, en particulier en terme de conditions aux limites. On montrera que l'algorithme PSO est adapté à l'optimisation topologique des metamatériaux électromagnétiques, et qu'il permet de développer des capes d'invisibilité électromagnétiques fréquentielles.

Le chapitre 6 sera consacré à l'application des méthodes d'optimisation précitées au développement des metamatériaux mécaniques. On essaiera notamment d'investiguer le possible développement de metamatériaux constitués d'acier, mais présentant des propriétés hyperélastiques - ou tout du moins non linéaires. Pour ce faire, on emploiera une

méthode d'homogénéisation par moyennation des champs afin de déduire des propriétés mécaniques microscopiques du Volume Élémentaire Représentatif les propriétés mécaniques macroscopiques du metamatériau ainsi développé.

Finalement, le dernier chapitre conclura ce travail de thèse de doctorat et proposera des perspectives de travail.

REVUE DE LA LITTÉRATURE ET BASES THÉORIQUES

Sommaire

2.1 Optimisation	7
2.1.1 Optimisation difficile	7
2.1.1.1 Cas de l'optimisation difficile : définition	7
2.1.1.2 Classification des méthodes d'optimisation	9
2.1.2 Optimisation multiobjectif	9
2.1.2.1 Définition du problème	9
2.1.2.2 Notion de dominance	11
2.1.2.3 Méthodes d'optimisation multiobjectif	12
2.1.3 L'optimisation en mécanique des structures	17
2.1.3.1 Généralités	17
2.1.3.2 Optimisation topologique	19
2.2 Méthodes metaheuristiques	23
2.2.1 Généralités	23
2.2.1.1 Propriétés	24
2.2.1.2 Avantages et inconvénients	25
2.2.1.3 Classification des méthodes metaheuristiques	27
2.2.2 Quelques méthodes metaheuristiques	27
2.2.2.1 Ant Colony Optimization	27
2.2.2.2 Les algorithmes évolutionnaires	33
2.2.3 Focus on : Particle Swarm Optimization	42
2.2.3.1 Historique, principe de base	42
2.2.3.2 Topologies de la population	45
2.2.3.3 Principales variantes et améliorations	53
2.3 Notions concernant la théorie des graphes	70
2.3.1 Introduction	70
2.3.2 Généralités concernant la théorie des graphes	70
2.3.2.1 Définitions	72
2.3.2.2 Introduction aux chaînes de Markov	76
2.3.3 L'algorithme PageRank	77
2.3.3.1 Généralités, définition	77
2.3.3.2 Calcul du PageRank	78
2.3.3.3 Calcul matriciel	81

2.3.3.4	Utilisation originale et intéressante de PageRank dans la littérature	83
2.4	Metamatériaux et notions d'homogénéisation	84
2.4.1	Les metamatériaux	84
2.4.2	Homogénéisation	87
2.4.2.1	Principes de changement d'échelle : passage du milieu hétérogène au milieu homogène	87
2.4.2.2	Méthode des champs moyens	90
2.4.2.3	De l'importance des conditions aux limites	94
2.5	Conclusions de l'étude bibliographique	96

2.1/ OPTIMISATION

2.1.1/ OPTIMISATION DIFFICILE

Dans de nombreux domaines scientifiques et techniques, les ingénieurs et chercheurs sont confrontés quotidiennement à des problèmes de complexité grandissante [121].

Le problème à résoudre peut alors être exprimé sous la forme d'un **problème d'optimisation**, c'est à dire sous la forme d'une fonction objectif (appelée aussi 'fonction coût'), que l'on cherche à minimiser ou maximiser, en modifiant un nombre de paramètres précédemment définis, qui sont **les variables d'optimisation**. De plus, ce problème est très souvent complété par des **contraintes**. En effet, tous les paramètres modifiés durant le processus d'optimisation doivent respecter un certain nombre de critères, sans quoi les solutions trouvées ne sont pas réalisables. Ces contraintes peuvent concerner tant des paramètres de conception mécanique, que des paramètres physiques que l'on sait d'ores et déjà inatteignables, ou encore des paramètres économiques.

De manière générale, on peut formaliser un problème d'optimisation classique à la manière de l'équation (2.1) suivante

$$\begin{cases} \text{Min (ou Max)} & f(\mathbf{x}) & \mathbf{x} = (x_1, x_2, \dots, x_D) \\ \text{soumis à} & g_i(\mathbf{x}) \geq 0 & \text{pour } i = 1, 2, \dots, n \end{cases} \quad (2.1)$$

D étant le nombre de variables d'optimisation du problème, $\mathbf{x}$ le vecteur des variables d'optimisation, f la fonction objectif, et les g_i sont les n fonctions contraintes.

2.1.1.1/ CAS DE L'OPTIMISATION DIFFICILE : DÉFINITION

Les problèmes d'optimisation peuvent être classés en différentes catégories. Tout d'abord, on distingue les problèmes à variables discrètes (ou problèmes d'optimisation combinatoire), et les problèmes à variables continues.

Dans la première catégorie, les variables d'optimisation ne peuvent pas prendre de valeurs réelles quelconques, mais doivent être choisies dans un panel de valeurs précédemment définies, ce qui rend cette catégorie de problèmes particulièrement difficile à résoudre. L'exemple utilisé de manière récurrente pour démontrer l'efficacité des algorithmes d'optimisation discrète est **le problème du voyageur de commerce**. Connaissant un ensemble de villes séparées entre elles par des distances données, ce problème consiste à trouver le plus court chemin qui relie toutes les villes (voir le chapitre 2.3.2 qui détaille plus précisément ce problème d'optimisation).

Dans la deuxième catégorie, les variables d'optimisation sont définies sur un intervalle donné, et peuvent prendre toutes les valeurs de cet intervalle. Un exemple de problème continu est la recherche des valeurs à affecter aux paramètres définissant une fonction dérivable et continue sur un intervalle donné, afin d'en trouver le minimum global.

En pratique, on rencontre aussi des problèmes mixtes qui comportent à la fois des variables continues et des variables mixtes [121], ainsi que - notamment dans le domaine de l'optimisation topologique - des problèmes dont les variables sont continues, puis transformées pour être contenues dans un domaine discret.

De nombreuses méthodes ont été développées pour résoudre ces deux types de pro-

blèmes. Dans le domaine de l'optimisation continue, les méthodes classiques dites d'**optimisation globale**, basées sur le calcul des sensibilités, peuvent être utilisées. Cependant, elles se révèlent souvent inefficaces du fait des problématiques engendrées par le calcul des sensibilités, par exemple lorsque la fonction objectif n'est pas convexe, ou qu'elle est non-différentiable. En effet, les algorithmes classiques de résolution sont basés sur le principe suivant : on part d'une solution initiale qui peut être soit choisie au hasard, soit définie par un concepteur. On applique alors au problème une modification aléatoire des variables d'optimisation, et l'on compare la valeur de la fonction objectif avant et après cette modification. Si la fonction objectif est plus adaptée au problème d'optimisation posé (c'est à dire qu'elle diminue dans le cas d'une minimisation, ou qu'elle augmente dans le cas d'une maximisation), on réutilise cet état comme un état initial et on réitère le processus jusqu'à ce que toute modification n'améliore plus la valeur de la fonction objectif. Si la fonction objectif est moins adaptée après la modification, on revient à la configuration précédente avant de faire une autre tentative. Comme on le voit sur la figure 2.1, ces méthodes d'optimisation itératives ne conduisent que très difficilement à l'optimum global de la fonction objectif. Si une solution trouvée par cette méthode est située dans un minimum local, l'algorithme ne réussira pas en sortir, et le minimum global ne sera pas trouvé. En effet, le calcul des sensibilités induit le calcul des **directions de descente**, qui pousse l'algorithme à rester bloqué dans les minima locaux de la fonction objectif.

Par exemple, dans le cas de la figure 2.1, si la solution choisie initialement de manière aléatoire est le point x_3 , la solution trouvée par un algorithme classique sera le point x_2 qui est un optimum local. Si la solution initiale est le point x_1 , la solution trouvée sera la solution optimale x^* . Ces méthodes ne sont donc pas toujours efficaces quant à leur capacité à résoudre correctement un problème d'optimisation donné, puisque leur réussite dépend fortement de l'état initial des variables d'optimisation, et de la présence et la quantité de minima locaux dans le domaine de recherche étudié.

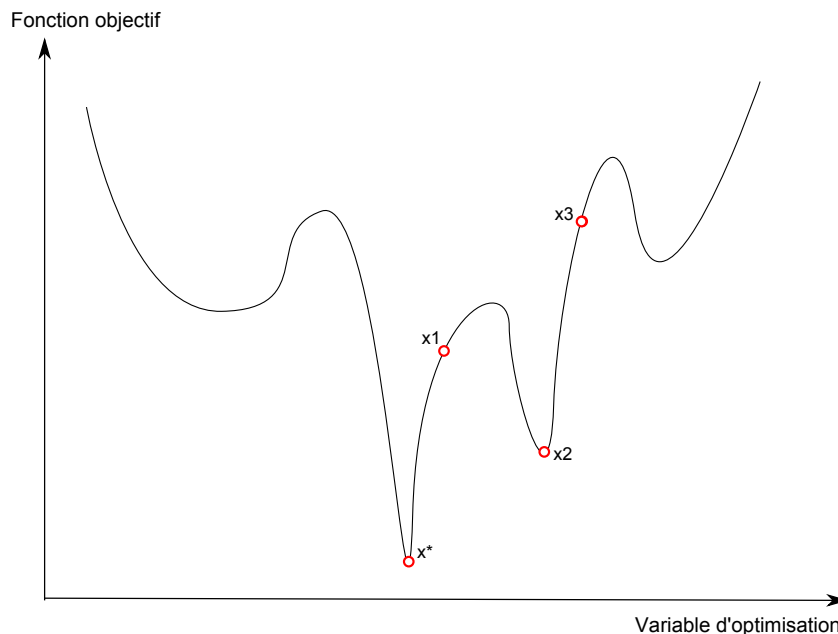


FIGURE 2.1 – Blocage d'un algorithme de résolution dans un optimum local

Dans le domaine de l'optimisation discrète, des méthodes heuristiques permettent d'obtenir des solutions approchées au problème d'optimisation posé, mais sont généralement

très spécialisées, et donc s'avèrent n'être aptes à la résolution que d'un nombre réduit de problèmes donnés. Les problèmes ne pouvant être résolus avec ces méthodes classiques peuvent être définis comme des **problèmes d'optimisation difficile**.

Il est donc nécessaire de différencier ces deux catégories, afin de cerner le domaine de l'optimisation difficile, et d'être en mesure de choisir la méthode adaptée pour résoudre ces problèmes. D'après [121], deux types de problèmes distincts peuvent être qualifiés de 'difficiles' :

- Certains problèmes d'optimisation discrète pour lesquels on ne connaît pas de méthode classique permettant de trouver de solution acceptable. C'est à dire que l'on ne connaît pas d'algorithme exact polynomial (c'est à dire dont le temps de calcul est proportionnel à N^n , où N désigne le nombre de variables d'optimisation du problème, et n est un nombre constant entier) permettant de résoudre efficacement le problème considéré. C'est le cas, en particulier, des problèmes N-P difficiles.
- Certains problèmes d'optimisation à variables continues pour lesquels on ne connaît pas d'algorithme permettant de repérer **systématiquement** un optimum global.

Définition 1 : Les classes de complexité

Les classes de complexité sont des ensembles de problèmes dont la complexité peut être assimilée identique selon un critère précédemment défini [228]. En théorie de la complexité, la classe des problèmes Non-déterministes Polynomiaux (classe NP) comprend tous les problèmes de décision qui peuvent être résolus sur une machine non-déterministe en un temps polynomial. Ainsi, un problème N-P complet est un problème de décision combinatoire dont les propriétés sont les suivantes :

- Il est possible de trouver une solution en un temps polynomial acceptable.
- Le problème est au moins aussi difficile que tous les autres problèmes de la classe N-P.

De fait, un problème **N-P difficile** est un problème qui possède au moins la seconde propriété [227].

2.1.1.2/ CLASSIFICATION DES MÉTHODES D'OPTIMISATION

Ainsi, pour répondre à la quantité de problèmes d'optimisation différents, incluant les problèmes d'optimisation difficile, Collette et al. [107] ont proposé une classification des différentes méthodes d'optimisation mono-objectif, en fonction de la catégorie du problème d'optimisation. Cette classification est explicitée à la figure 2.2.

2.1.2/ OPTIMISATION MULTIOBJECTIF

2.1.2.1/ DÉFINITION DU PROBLÈME

Alors que les problèmes d'optimisation mono-objectif ne cherchent à optimiser qu'une seule fonction objectif à la fois, les problèmes d'optimisation multiobjectif [107] cherchent, eux, à optimiser **simultanément** plusieurs objectifs, en fonction de nombreux paramètres. Ces fonctions objectif sont généralement contradictoires, c'est à dire que l'amélioration de l'une d'entre elles implique nécessairement l'altération d'une autre. Ainsi, il existe non

pas une seule, mais une multitude de solutions à trouver, qui représentent finalement des compromis entre les différentes fonctions objectif, c'est à dire des points pour lesquels l'amélioration d'une fonction impliquerait nécessairement la détérioration d'une autre [78].

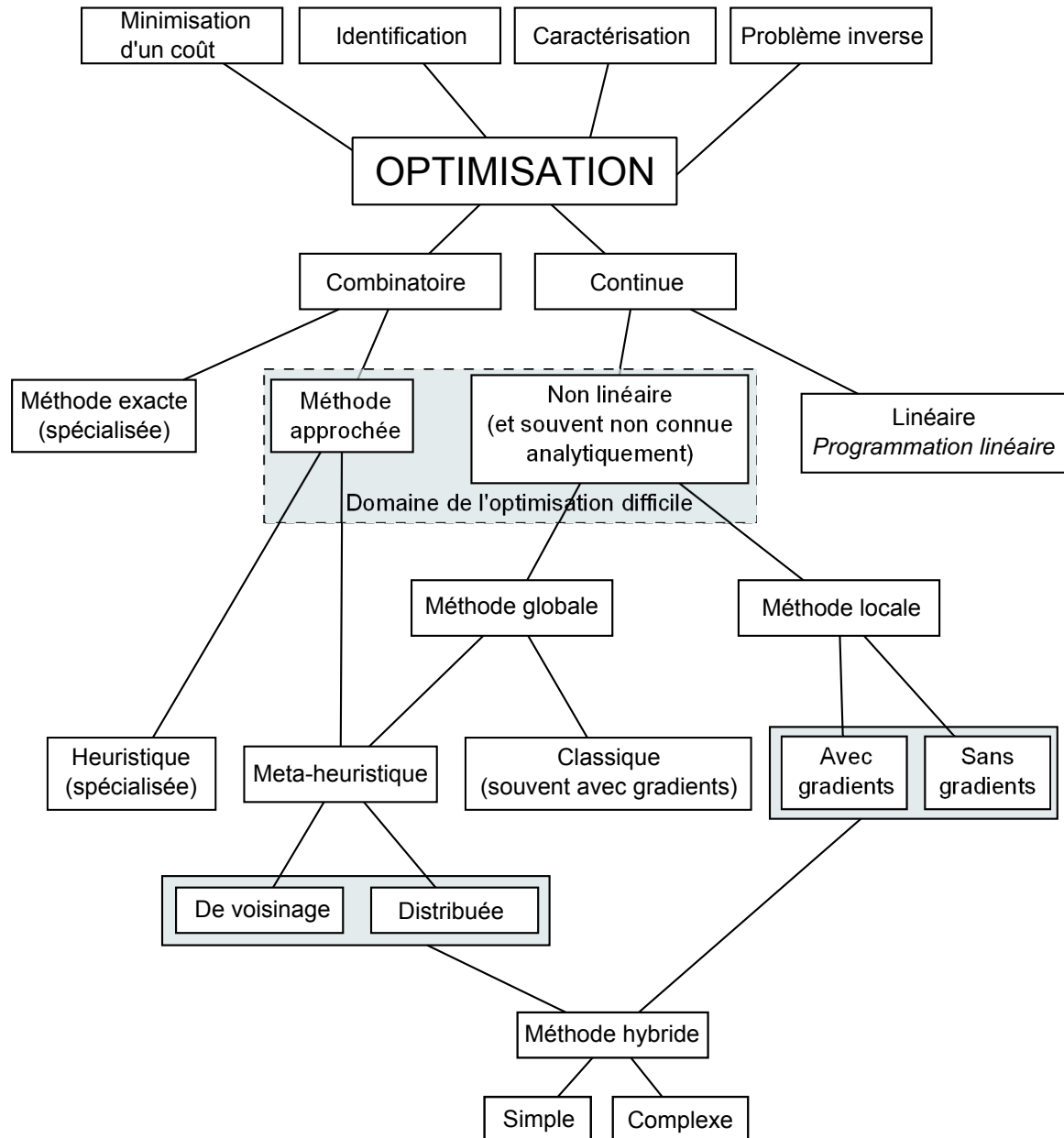


FIGURE 2.2 – Classification générale des méthodes d'optimisation mono-objectif

Un problème d'optimisation multiobjectif peut être exprimé de la manière suivante [117] :

Soient :

- X un domaine de recherche de dimension n ,
- $f_i(x)$ avec $i \in \{1, k\}$ k fonctions objectif définies sur X ,
- $g_i(x) \leq 0$ avec $i \in \{1, m\}$ m contraintes d'inégalité,
- $h_i(x) = 0$ avec $i \in \{1, p\}$ p contraintes d'égalité.

Le problème d'optimisation multi-objectif peut être établi comme la recherche d'un vecteur $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*) \in X$ qui satisfasse les $m + p$ contraintes et optimise le vecteur des fonctions objectifs $f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_k(\mathbf{x})]$. Pour plus de simplicité, on appellera ce problème **le problème P** dans la suite de l'étude.

2.1.2.2/ NOTION DE DOMINANCE

Comme expliqué précédemment, lorsque l'on résout un problème d'optimisation multiobjectif, on trouve finalement une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser : les solutions qui représentent des compromis entre les différentes fonctions objectif. C'est à dire qu'il faut qu'il existe une relation de **dominance** (au sens de Pareto) entre la solution en question, et les autres solutions de l'espace X .

Définition 2 : Dominance d'un vecteur

[107] : On dit qu'un vecteur $\mathbf{x}_1$ domine le vecteur $\mathbf{x}_2$ si et seulement si :

- $\mathbf{x}_1$ est au moins aussi bon que $\mathbf{x}_2$ dans tous les objectifs, et,
- $\mathbf{x}_1$ est strictement meilleur que $\mathbf{x}_2$ dans au moins un objectif.

Mathématiquement, cette relation de dominance peut être exprimée et vérifiée de la manière suivante :

Soient $\mathbf{u} = (u_1, \dots, u_k)$ et $\mathbf{v} = (v_1, \dots, v_k)$ deux vecteurs. On dit alors que $\mathbf{u}$ domine $\mathbf{v}$ si et seulement si $u_i \leq v_i \forall i \in [1, k]$, et $u_i < v_i$ pour au moins une composante.

On appelle cette propriété **la dominance au sens de Pareto**, et elle est utilisée pour définir **les points optimaux de Pareto**.

Ainsi, une solution $\mathbf{x}$ du problème d'optimisation multiobjectif posé est dite **optimale au sens de Pareto** si et seulement si il n'existe pas d'autre solution $\mathbf{y}$ telle que $f(\mathbf{y})$ domine $f(\mathbf{x})$. L'ensemble des solutions optimales au sens de Pareto est noté P^* .

Le front de Pareto est alors défini par

$$PF^* = \{(f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) / \mathbf{x} \in P^*\}.$$

Ainsi, on a la définition suivante : on appelle **front de Pareto** l'ensemble des points appartenant au graphique représentant la première fonction objectif en fonction de la deuxième, dont il est impossible d'augmenter la performance d'une fonction objectif sans diminuer les performances de la seconde. On appelle ces solutions, des **solutions non-dominées**.

Un front de Pareto peut être convexe ou concave, et cette propriété est définie à la manière décrite dans les équations (2.2 et 2.3) suivantes.

- Un front de Pareto est dit convexe si et seulement si

$$\forall (\mathbf{u}, \mathbf{v}) \in (PF^*) \times (PF^*), \forall \lambda \in (0, 1), \exists \omega \in (PF^*) | \lambda \|\mathbf{u}\| + (1 - \lambda) \|\mathbf{v}\| \geq \|\omega\| \quad (2.2)$$

- Un front de Pareto est dit concave si et seulement si

$$\forall (\mathbf{u}, \mathbf{v}) \in (PF^*) \times (PF^*), \forall \lambda \in (0, 1), \exists \omega \in (PF^*) | \lambda \|\mathbf{u}\| + (1 - \lambda) \|\mathbf{v}\| \leq \|\omega\| \quad (2.3)$$

En pratique, un front de Pareto peut être convexe, concave, ou partiellement concave et convexe. Il peut aussi être continu, ou discontinu.

Graphiquement, on peut représenter les deux types de fronts de Pareto comme on le voit sur la figure 2.3 [102].

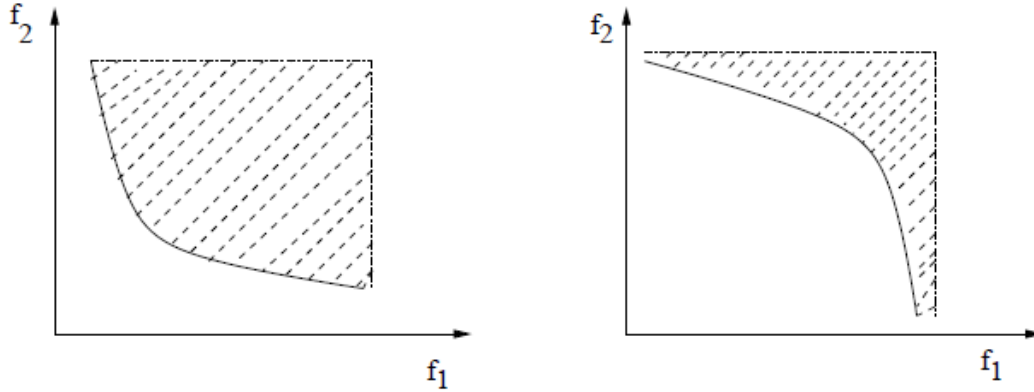


FIGURE 2.3 – Les deux types de fronts de Pareto : Convexe et concave

2.1.2.3/ MÉTHODES D'OPTIMISATION MULTIOBJECTIF

Du fait de la complexité des problèmes d'optimisation multi-objectif, de nombreuses méthodes ont été développées ces dernières années pour pallier à ces problèmes et proposer des méthodes efficaces.

2.1.2.3.1/ Les méthodes scalaires

Les méthodes scalaires permettent généralement de fusionner tous les objectifs du problème en un seul par des méthodes mathématiques diverses. On parle aussi de méthode d'agrégation des fonctions objectifs.

La méthode de pondération des fonctions objectif

Cette méthode propose de ramener le problème à un problème d'optimisation mono-objectif, dont les méthodes de résolution sont nombreuses [86]. Ainsi, la manière de procéder consiste à effectuer une combinaison linéaire de toutes les fonctions objectif, en choisissant judicieusement les coefficients de pondération appliqués à chacune d'entre elles.

Le problème est alors exprimé comme on le voit dans l'équation (2.4) suivante

$$\begin{cases} \min & f_{eq}(\mathbf{x}) = \sum_{i=1}^k \omega_i f_i(\mathbf{x}) \\ \text{soumis à} & \mathbf{g}(\mathbf{x}) \leq 0 \\ \text{et} & \mathbf{h}(\mathbf{x}) = 0 \end{cases} \quad (2.4)$$

où $\mathbf{x}$ est le vecteur des variables d'optimisation, f_{eq} est la fonction objectif équivalente et les paramètres ω_i sont les pondérations des différentes fonctions objectif f_i . On a aussi $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^m$ et $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^p$.

Pour définir les coefficients de pondération, on utilise généralement la relation décrite dans l'équation (2.5) suivante

$$\sum_{i=1}^k w_i = 1 \quad (2.5)$$

La méthode de Keeney-Raiffa

Assez proche de la méthode précédemment explicitée, cette dernière propose de créer une fonction objectif en utilisant le produit des fonctions objectifs précédentes. Ainsi, le problème devient équivalent à un problème d'optimisation mono-objectif plus simple à résoudre [107].

Dans ce cas, la nouvelle fonction objectif est définie de la manière suivante

$$f_{eq}(\mathbf{x}) = K \prod_{i=1}^k (k_i u_i(f_i(\mathbf{x})) + 1) \quad (2.6)$$

K étant un coefficient de normalisation compris entre 0 et 1, et $u_i(y)$ une fonction strictement non-décroissante, qui peut incorporer des linéarités.

La méthode de la distance à un objectif de référence

Dans ce cas, la somme que l'on va utiliser pour transformer les fonctions objectifs en un seul objectif prendra la forme d'une distance.

On considère le vecteur $\mathbf{F} = (F_1, \dots, F_k)$ dont les coordonnées correspondent à celle d'un objectif idéal, choisi par l'utilisateur. On définit alors la distance $L(f(\mathbf{x}))$ entre les points de l'espace de recherche et ce vecteur dans l'espace des solutions par la relation (2.7) suivante

$$\begin{cases} L(f(\mathbf{x})) = [\sum_{i=1}^k |F_i - f_i(\mathbf{x})|^r]^{1/r} \\ \text{avec } 0 \leq r < \infty \end{cases} \quad (2.7)$$

Dans cette méthode, on cherche donc tous les points de l'espace de recherche les plus proches du point objectif de référence, et c'est ce qui nous permet de définir le front de Pareto, comme on le voit dans la figure 2.4.

La méthode du compromis

Cette méthode permet de transformer un problème d'optimisation multi-objectif en un problème d'optimisation mono-objectif comportant quelques contraintes supplémentaires.

La démarche est la suivante :

- On choisit un objectif prioritaire à optimiser,
- On choisit un vecteur de contraintes initial,
- On transforme le problème en conservant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalité.

Cette méthode est aussi appelée [la méthode de la contrainte \$\epsilon\$](#) .

La méthode du but à atteindre

Contrairement aux méthodes de résolution précédentes qui n'étaient efficaces que sur

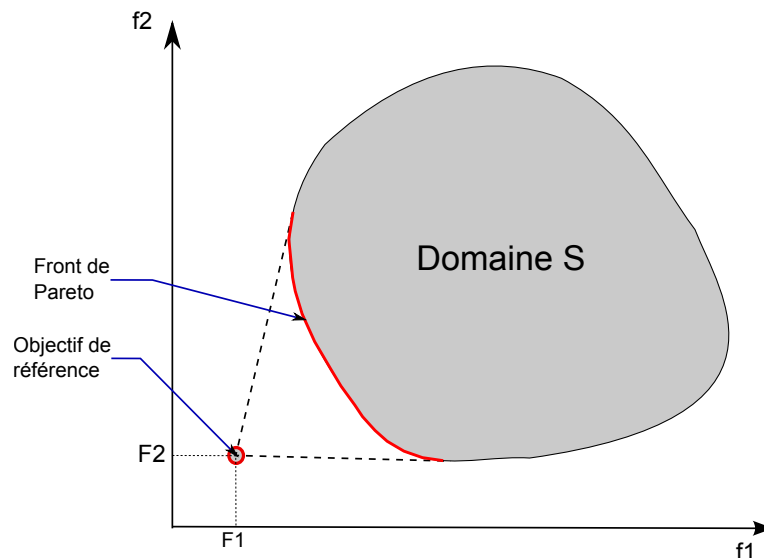


FIGURE 2.4 – Principe de la méthode de la distance à un objectif de référence

les domaines convexes, cette méthode est aussi efficace pour déterminer les fronts de Pareto convexes que les front de Pareto concaves.

La démarche est la suivante :

- On choisit un vecteur de fonctions objectif initial F ,
- On choisit une direction de recherche (c'est à dire que l'on propose, en quelque sorte, des coefficients de pondération, comme pour la méthode de pondération des fonctions objectifs [107]) ω ,
- On cherche ensuite à minimiser un coefficient scalaire λ qui représente l'écart par rapport à l'objectif initial que l'on s'était fixé.

La méthode dite “du but programmé”

Cette méthode est identique à la méthode du but à atteindre, si ce n'est que les fonctions objectif sont utilisées comme des contraintes d'égalité, et non pas comme des contraintes d'inégalité.

La démarche est donc la suivante :

- On choisit un vecteur de fonctions objectif initial F ,
- A chaque objectif, on associe deux nouvelles variables que l'on appelle les déviations par rapport au vecteur de fonction objectif fixé,
- On minimise ensuite une des deux variables d'optimisation. Le choix de la variable se fait en fonction du type de dépassement que l'on veut (au dessus ou au dessous de l'objectif précédemment fixé).

L'ordonnement lexicographique

Cette méthode propose autant d'étapes de calcul qu'il y a de fonctions objectifs [20]. En effet, elle consiste à considérer les fonctions objectif les unes après les autres, en considérant les autres comme des contraintes [86].

Par contre, le choix de l'ordre dans lequel les fonctions objectif sont choisies pour être minimisées est généralement fait de manière arbitraire, et deux ordonnancements différents peuvent ne pas donner la même solution finale.

De nombreuses autres méthodes ont été proposées dans la littérature, mais celles présentées ci-avant constituent l'essentiel des méthodes utilisées. Elles sont ainsi été déclinaées, améliorées, transformées, mais la grande majorités des méthodes sont basées sur celles-ci [107].

La méthode des contraintes d'égalité propres

Cette méthode, appelée aussi *Proper Equality Constraints* dans la littérature [13] ne suppose aucune hypothèse quant à la convexité du front de Pareto à trouver.

On part du [problème P](#) précédemment défini. On le transforme de la manière suivante

$$\left\{ \begin{array}{ll} \min & f_k(\mathbf{x}) \\ \text{soumis à} & f_1(\mathbf{x}) = \alpha_1 \\ & \vdots \\ & f_{k-1}(\mathbf{x}) = \alpha_{k-1} \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \end{array} \right. \quad (2.8)$$

Les quelques notations suivantes sont nécessaires à la bonne compréhension de la méthode :

- On note $\alpha = \{\alpha_1, \dots, \alpha_{k-1}\}$ le vecteur des bornes des contraintes d'optimisation.
 - On note $X_\alpha = \{\mathbf{x} \in X | f_1(\mathbf{x}) = \alpha_1, \dots, \alpha_{k-1}\}$.
 - On note $D = \{\alpha \in \mathbb{R}^{k-1} | X_\alpha \neq \emptyset\}$ l'ensemble des valeurs de α telles que les contraintes additionnelles (obtenues après avoir transformé le problème) ne soient pas trop restrictives, et qu'il existe des valeurs dans X_α .
 - On note $\phi(\alpha) = \inf \{f_k(\mathbf{x}) | \mathbf{x} \in X_\alpha\}$. C'est la valeur minimale de la fonction objectif en prenant en compte toutes les contraintes.
 - On note $B = \{\alpha \in D | \phi(\alpha) < \infty \text{ et } f_k(\hat{\mathbf{x}}) = \phi(\alpha), \hat{\mathbf{x}} \in X_\alpha\}$. B est donc une restriction de D.
- Cette méthode est réalisée en 5 étapes :

1. Transformer tous les objectifs, sauf un, en contraintes d'égalité paramétriques. On obtient alors le problème suivant

$$\left\{ \begin{array}{ll} \min & f_k(\mathbf{x}) \\ \text{avec} & f_1(\mathbf{x}) = \alpha_1, \dots, f_{k-1}(\mathbf{x}) = \alpha_{k-1} \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \end{array} \right. \quad (2.9)$$

2. Résoudre le problème et déterminer $\phi(\alpha)$, D , et $\hat{\mathbf{x}}(\alpha)$. Sachant qu'à chaque $\alpha \in D$ correspond un $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\alpha)$ qui n'est pas nécessairement dans X ni dans X_α . Ce $\hat{\mathbf{x}}$ est tel que $f_k(\hat{\mathbf{x}}(\alpha))$. On appelle cette valeur une solution "supremum" du problème associé.
3. Déterminer l'ensemble B et la solution optimale du problème associé $\hat{\mathbf{x}}(\alpha)$ avec $\alpha \in B$.
4. Supprimer de B tous les vecteurs de contraintes α qui ne soient pas des vecteurs propres.
5. Déterminer le reste final de B que l'on notera B^* .

Proposition : Un vecteur α_0 est propre si l'une des assertions suivantes est vérifiée :

- ϕ croît sur D et $\phi(\alpha) \neq \phi(\alpha_0) \forall \alpha \in B | \alpha \geq \alpha_0$.
- ϕ croît sur D et est absolument croissante à droite de α_0 sur B .
- ϕ croît sur D et est absolument croissante sur B .
- ϕ est absolument croissante à droite de α_0 sur D .
- ϕ est absolument croissante sur le sous-ensemble $D(\alpha_0) = \{\alpha \in D | \alpha \geq \alpha_0\}$.
- ϕ est absolument croissante sur D .
- ϕ est absolument croissante sur $D(\alpha_0)$ et est localement absolument croissante à droite de α_0 .

Proposition : Un vecteur $\alpha_0 \in B$ n'est pas localement propre si $\frac{\partial \phi(\alpha_0)}{\partial \alpha_i} < 0$ pour au moins un i tel que $1 \leq i \leq k - 1$.

Cette méthode est une méthode littérale, et n'est donc pas évidente à implémenter. Deux algorithmes ont toutefois été implémentés afin d'y parvenir : L'algorithme de Lin-Tabak [19] et l'algorithme de Lin-Giesy [16].

2.1.2.3.2/ Les méthodes interactives

Les méthodes interactives permettent de chercher une et une seule solution au problème d'optimisation multiobjectif posé. Elles forment la famille des méthodes progressives et permettent à l'utilisateur de déterminer des préférences vis à vis d'un compromis entre les différentes fonctions objectif. Ces méthodes ne proposant pas de calculer tous les points constituant le front de Pareto, elles ne seront pas développées dans cette étude.

La méthode du compromis par substitution

Cette méthode, appelée aussi *Surrogate Worth tradeoff* [11] dans la littérature est basée sur la méthode du compromis, à laquelle on ajoute un processus itératif, pour que la méthode converge vers la solution qui satisfasse au mieux l'utilisateur.

La méthode de Fandel

Cette méthode [107] permet de guider l'utilisateur dans le choix des coefficients de pondération. Une méthode mathématique permet d'automatiser le choix de ces coefficients en fonction des résultats précédemment trouvés. Cette automatisation mathématique permet de donner un vecteur représentant un objectif idéal pour l'utilisateur. Les coefficients sont ensuite calculés, à chaque instant, en fonction de la position actuelle de l'algorithme, par rapport à ce point idéal. Cette méthode peut être rapprochée de la méthode de la distance à un objectif de référence précédemment explicitée, et aux méthodes intelligentes que nous verrons par la suite dans la partie 2.2.

La méthode STEP

Cette méthode [36] est similaire à la méthode de Fandel. Ici, les informations tirées de la préférence de l'utilisateur permettent de restreindre le domaine petit à petit, au cours des itérations, afin de se rapprocher de la solution considérée comme idéale au sens défini par l'utilisateur.

La méthode de Jahn

Ici, on commence par choisir un point de départ (qui peut être choisi de manière aléatoire

ou sur indications de l'utilisateur) où le problème est résolu au cours des itérations à travers l'espace de recherche en se déplaçant en direction de la solution optimale au sens de Pareto [36]. Cette méthode est basée sur la méthode d'optimisation scalaire dans les directions réalisables présentée par Zoutendjik [4].

La méthode du simplexe

Cette méthode n'a rien à voir avec la méthode du simplexe utilisée en programmation linéaire [5]. Il s'agit d'une méthode de recherche séquentielle (c'est à dire fonctionnant pas à pas) [6] de l'optimum global de la fonction objectif d'un problème d'optimisation.

Cette méthode utilise des essais pour déterminer la direction d'amélioration des fonctions objectif.

On commence par choisir de manière aléatoire $k + 1$ points (k étant la dimension du vecteur de décision x). L'algorithme de résolution évalue alors tous ces points par rapport aux fonctions objectif testées, et supprime le point le moins adapté. Il crée alors un nouveau point à partir du point supprimé (la nouvelle position est calculée par réflexion du point rejeté par rapport à deux autres points) et reitère l'opération. Afin d'être effectif, cet algorithme comporte deux règles qui permettent d'éviter de tourner autour des mauvaises solutions, et de se déplacer suffisamment rapidement dans l'espace des solutions :

1. Rejeter les pires solutions.
2. Ne jamais revenir sur un point qui vient juste d'être rejeté.

2.1.3/ L'OPTIMISATION EN MÉCANIQUE DES STRUCTURES

2.1.3.1/ GÉNÉRALITÉS

Le processus de conception d'une structure peut être divisé en quatre étapes :

- **La formulation fonctionnelle du problème**, et les considérations technologiques. Cette étape consiste à faire des choix, ou des hypothèses mécaniques, avant même que l'ingénieur concepteur ne conçoive sa structure. On choisira par exemple le nombre de barres d'un treillis, les dimensions globales d'une structure. Ces considérations sont souvent données par d'autres contraintes, comme des contraintes de fabrication (on connaît la quantité de matière disponible), d'encombrement (on connaît les dimensions maximales d'un moteur pour qu'il puisse s'insérer sous le capot d'une voiture), ou de montage (on connaît les dimensions des roulements utilisés, et donc les dimensions de leurs logements dans la structure). Ces considérations seront à nouveau prises en compte lors de la phase d'optimisation.
- **La phase de conception**, caractérisée à la fois par la créativité, mais aussi les compétences d'ingénierie du concepteur. Dans cette phase, des règles d'ingénierie précises doivent être appliquées, notamment afin que le système final soit fonctionnel, fabriquant et montable. Le concepteur devra aussi tenir compte de la phase d'optimisation qui suit, en concevant son système de manière à ce qu'il puisse être optimisé. C'est à dire qu'il devra identifier clairement les paramètres d'optimisation, et réaliser une conception clairement paramétrée.
- **La phase d'optimisation**. Tout d'abord le problème d'optimisation est défini. On choisit la fonction objectif, les variables d'optimisation, ainsi que les contraintes d'optimisation à considérer. Ensuite, un algorithme d'optimisation est choisi, et va modifier le système de manière itérative jusqu'à ce qu'un critère d'arrêt soit atteint.

En pratique, ce critère d'arrêt est donné soit par le nombre d'itérations maximal, soit par une valeur de la fonction objectif à atteindre. Souvent, le nombre maximum d'itérations est atteint avant que le calcul ne trouve la configuration qui donne la valeur *exacte* de la fonction objectif.

- **L'analyse des résultats.** Quel que soit le type d'optimisation choisi, les algorithmes donnent souvent des résultats qu'il faut retravailler par la suite. Les solutions sont donc renvoyées au concepteur, qui va essayer de faire un compromis entre les solutions proposées par l'algorithme, et les règles de conception qu'il connaît.

Quel que soit le type d'optimisation choisi, le but d'un processus d'optimisation en mécanique est d'exploiter les ressources données à leur maximum, afin de maximiser l'utilité de la structure [50]. Ainsi, dans le domaine de la mécanique, la meilleure conception sera celle qui minimisera la masse et/ou le coût de la structure, que ce soit en terme de quantité de matière, ou en terme de temps de fabrication.

Dans le domaine de la mécanique des structures, les problèmes d'optimisation peuvent être classés dans les trois catégories suivantes [208] comme on le voit dans la figure 2.5 :

- L'optimisation paramétrique, où les variables d'optimisation sont généralement des paramètres géométriques du système mécanique à optimiser ;
- L'optimisation de forme où l'on fait varier la position des frontières de la structure mécanique étudiée ;
- L'optimisation topologique où l'on modifie la topologie, c'est à dire la répartition de matière, de la structure étudiée.

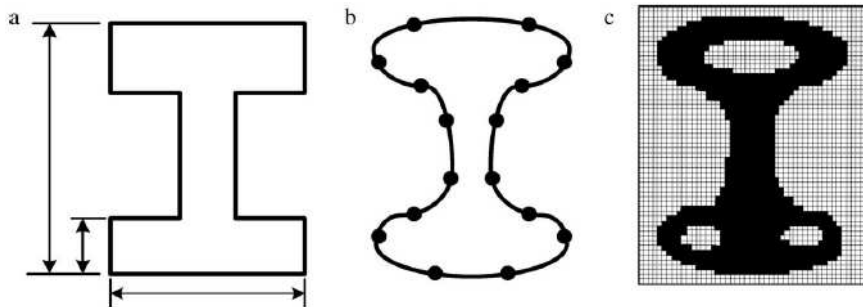


FIGURE 2.5 – Les trois types d'optimisation structurelle [208] : (a) L'optimisation paramétrique, (b) l'optimisation de forme, et (c) l'optimisation topologique

Concernant l'optimisation paramétrique, elle est souvent assez simple, et proche de l'optimisation classique continue, comme nous l'avons vu dans la partie 2.1.1.1. Dans le cas d'un treillis par exemple, on souhaite minimiser la masse de ce treillis, en considérant une contrainte d'optimisation sur les contraintes mécaniques maximales admissibles dans les barres qui le composent. Dans ce cas, le vecteur des variables d'optimisation sera composé des valeurs des sections des barres du treillis. Ce vecteur sera modifié de manière itérative, jusqu'à ce que le problème d'optimisation soit résolu, ou tout d'une moins que l'on tende vers une solution acceptable, au sens où elle satisfait les contraintes d'optimisation, tout en minimisant la valeur de la fonction objectif.

En revanche, le domaine de l'optimisation topologique s'avère être beaucoup plus complexe. Là encore, il existe deux sortes d'optimisation topologique des structures :

- L'optimisation topologique continue. Dans ce cas, on modifie la densité ou la masse volumique du matériau dans la structure.

- L'optimisation topologique discrète (elle est souvent binaire), dans laquelle on choisit une répartition de matière dans les éléments finis discrétisant la structure, le choix des matériaux possibles étant donné dans une liste prédéfinie.

Dans tous les cas, en mécanique, l'optimisation topologique nécessite la discrétisation de la pièce ou de la structure par éléments finis. Un algorithme modifiera ensuite la répartition de matière dans ces éléments. Comme dans le cas de l'optimisation classique, non nécessairement appliquée à la mécanique, des méthodes déterministes ou non-déterministes peuvent être utilisées. Notons que les gains moyens de performances vont de 5 à 10% en optimisation paramétrique, de 10 à 30 % en optimisation de forme, et de 40 à 100% en optimisation topologique [68, 163]. Ces augmentations de performances respectives montrent tout l'intérêt de l'optimisation topologique, que nous présenterons dans la suite de notre travail.

2.1.3.2/ OPTIMISATION TOPOLOGIQUE

2.1.3.2.1/ Définition du problème d'optimisation

En optimisation des structures, le problème d'optimisation peut être décrit de deux manières différentes :

- La minimisation de la compliance, avec une contrainte d'optimisation sur le volume,
- La minimisation du volume, avec une contrainte d'optimisation sur la compliance.

Dans les faits, la première solution est souvent choisie. Dans ce cas, le problème d'optimisation peut être formulé dans sa forme discrète à la manière de l'équation (2.10) suivante [120]

$$\begin{array}{ll} \min & \mathbf{f}^T \mathbf{u} \\ \text{avec} & \mathbf{K} \mathbf{u} = \mathbf{f} \end{array} \quad (2.10)$$

où l'on a :

- $\mathbf{u}$ est le vecteur déplacement,
- $\mathbf{f}$ est le vecteur définissant le chargement,
- $\mathbf{K}$ est la matrice de rigidité du système considéré.

La rigidité $\mathbf{K}$ dépend bien sûr de la rigidité K_e de chaque élément e du maillage. On peut donc exprimer $\mathbf{K}$ de la manière suivante

$$\mathbf{K} = \sum_{e=1}^N \mathbf{K}_e \quad (2.11)$$

où $\mathbf{K}_e$ représente la matrice de rigidité élémentaire de chaque élément $e = 1, 2, \dots, n$, et le signe $=$ représente l'assemble par la méthode des éléments finis des différentes matrices de rigidité de chacun des éléments du maillage.

Deux types de méthodes d'optimisation topologiques des structures peuvent être employées : les méthodes déterministes et les méthodes stochastiques et metaheuristiques. Dans cette partie, nous ne présenterons que les méthodes déterministes. Les méthodes metaheuristiques, quant à elles, seront présentées en détail dans la partie 2.2 suivante, puisqu'elles constituent le coeur de notre travail.

2.1.3.2.2/ Les méthodes déterministes

Dans ce paragraphe, nous présentons brièvement les méthodes d'optimisation déterministes communément utilisées dans la littérature.

La méthode d'homogénéisation

Proposée initialement par Bendsoe et al.[31, 56] et développée ensuite par Allaire et al.[48, 63, 64, 71, 72], la méthode d'homogénéisation est basée sur les concepts de distribution optimale de matière dans un domaine donné, et de relaxation. Ainsi, grâce à la théorie de l'homogénéisation des structures, Bendsoe et al. proposent de déterminer les propriétés macroscopiques d'une microstructure poreuse en fonction de la densité de matière dans le domaine [15, 31]. Comme dit par Allaire et al.¹, "d'un point de vue mécanique, il est plus souvent avantageux de creuser, pour un même volume de matériau, un nombre infiniment grand de trous infiniment petits, plutôt que quelques gros trous". Ainsi, les variables d'optimisation sont les dimensions des perforations de la cellule de base. On comprend aisément que dans ce cas, les résultats seront très dépendants du maillage initial de la cellule de base. Dans ce cas, la fonction objectif traitée est toujours la minimisation de la compliance[163]. En d'autres termes, le problème revient à chercher la répartition optimale de matière dans le domaine de la structure, afin de maximiser sa rigidité. Ainsi, la structure est déterminée, en chacun des points la composant, par une densité de matériau et par les propriétés effectives du composite considéré dans sa globalité. C'est d'ailleurs à cette étape que la théorie de l'homogénéisation - que nous verrons plus en détails par la suite - permet de calculer ces propriétés effectives. Le problème ici est que la topologie finale est souvent non fabriquer, puisque chacun des éléments contient un matériau différent, dont la densité peut varier de manière continue entre le vide et le solide. Pour résoudre ce problème, on recourra à des structures sub-optimales. Le concepteur aura alors en charge de modifier manuellement la structure obtenue, afin de déterminer quels éléments garder et quels éléments supprimer. Ces décisions seront prises en fonction de plusieurs paramètres. On choisira par exemple de ne garder que les éléments dont la densité est supérieure à un certain seuil précédemment défini, et l'on essaiera aussi de supprimer les problèmes de damiers dans la topologie.

Dans le registre des méthodes déterministes, on notera aussi la méthode **power law** présentée initialement par Bendsoe et al. [84], c'est à dire une loi dans laquelle les densités intermédiaires sont pénalisées. Ainsi, pour chaque élément fini i , son module d'élasticité E_i est donné en fonction de la densité relative $\bar{\rho}_i$, de la valeur maximale admissible du module d'élasticité E_0 et de la densité ρ_0 du matériau utilisé, comme on le voit dans l'équation suivante

$$E_i = \bar{\rho}_i^p E_0 \quad (2.12)$$

où l'on a

$$\rho_i = \bar{\rho}_i \rho_0 \quad (2.13)$$

Souvent appelée SIMP (Solid Isotropic Material with Penalization), cette méthode a été

1. http://www.cmap.polytechnique.fr/allaire/cray_plaq.html, dernière connexion le 12 avril 2016.

largement utilisée dans la littérature. Dans ce cas là, le problème - discrétisé par éléments finis - peut être formulé de la manière suivante

$$\begin{aligned} S_e(P) &= \sum_{e=1}^N (\rho_e)^p [\mathbf{u}_e]^T [\mathbf{k}_e] [\mathbf{u}_e] \\ \text{avec} \quad V^* - \sum_{e=1}^N V_e \rho_e &= 0 \\ 0 < \rho_{\min} \leq \rho_e &\leq 1 \end{aligned} \quad (2.14)$$

où $S_e(P)$ est l'énergie de déformation, $[\mathbf{u}_e]$ est le vecteur des déplacements nœuds de l'élément e en particulier, $[\mathbf{k}_e]$ est la matrice de rigidité de l'élément e , $\rho_{\min}$ est la densité de matière minimale acceptable, V^* est le volume à atteindre pour optimiser la structure, V_e et ρ_e sont respectivement le volume et la densité de l'élément e . Le facteur p est justement le facteur de pénalisation permettant la modification de la densité de matière dans chacun des éléments du maillage au cours des itérations. Ces méthodes ne nous concernant pas directement, nous renvoyons le lecteur aux articles et ouvrages [47, 119, 120, 141, 147, 167, 237, 240]. Notons tout de même que des adaptations de ces méthodes de pénalisation ont été proposées dans la littérature [143] afin d'obtenir des structures binaires.

Les méthodes évolutives

A ne pas confondre avec les méthodes évolutionnaires, les méthodes évolutives consistent à supprimer, à chaque itération, la matière jugée inutile, c'est à dire la matière la moins contrainte. Ainsi, les éléments finis dont le rapport de contrainte vis à vis de la contrainte équivalente de Von Mises maximale est inférieure à un certain critère choisi sont éliminés de la topologie, comme on le voit dans l'équation suivante

$$\frac{\sigma_e^{VM}}{\sigma_{\max}^{VM}} < RR_i \quad (2.15)$$

où σ_e^{VM} représente la contrainte équivalente de Von Mises de l'élément e , $\sigma_{\max}^{VM}$ est la contrainte équivalente de Von Mises maximale de la structure, et RR_i est le critère de suppression des éléments à l'itération i . Le critère RR_i reste identique jusqu'à ce que tous les éléments vérifient la condition de l'équation (2.15). Dès lors que le calcul a convergé, c'est à dire que tous les éléments jugés comme inutiles sont supprimés, le critère de suppression évolue comme on le voit dans l'équation suivante

$$RR_{i+1} = RR_i + ERR \quad (2.16)$$

L'évolution du paramètre RR_i se poursuit jusqu'à la convergence, c'est à dire jusqu'à ce que les critères d'arrêt soient vérifiés.

Dans le cas de la méthode évolutive ESO, le problème d'optimisation est souvent formulé de la manière suivante [200]

$$\begin{aligned} \text{Minimize} \quad & C = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} \\ \text{Subject to} \quad & V^* - \sum_{i=1}^N V_i x_i = 0 \\ & x_i = x_{\min} \text{ or } 1 \end{aligned} \quad (2.17)$$

où C est généralement la compliance moyenne ($\frac{1}{2} \mathbf{f}^T \mathbf{u}$) plutôt que la compliance ($\mathbf{f}^T \mathbf{u}$) utilisée généralement avec la méthode SIMP, $\mathbf{K}$ est la matrice de rigidité globale, $\mathbf{u}$ le

vecteur déplacement, V_i le volume de chaque élément i , V^* le volume à atteindre avec le processus d'optimisation, N le nombre total d'éléments dans le modèle, et x_i est la densité du $i^{\text{ème}}$ élément du domaine. On définit alors α_i comme le changement de l'énergie de déformation dû à la suppression de l'élément i du domaine par [66]

$$\alpha_i = \frac{1}{2} u_i^T K_i u_i \quad (2.18)$$

où u_i et K_i sont respectivement le vecteur déplacement et la matrice de rigidité de l'élément i . Dans la méthode ESO originale, les éléments sont complètement supprimés du domaine, comme on le voit dans l'algorithme 1 suivant [66].

Algorithm 1 L'algorithme ESO global

- 1) Discrétiser la structure en utilisant un maillage fin d'éléments finis.
 - 2) Calcul par éléments finis de la structure en considérant les efforts et les conditions aux limites du problème,
 - 3) Calculer la sensibilité α_i de chaque élément,
 - 4) Supprimer les éléments dont la sensibilité α_i est la plus petite,
 - 5) Répéter les étapes 2 à 4 jusqu'à ce qu'un des critères d'arrêt soit vérifié.
-

Notons que Hardjasaputra et al. [244] remarquent que l'exactitude ou la qualité de la solution sont améliorées en choisissant une valeur assez petite du paramètre ERR . Pour obtenir une solution acceptable, il est noté que ERR devrait être choisi entre 1 et 2 %.

Basée sur la méthode ESO, la méthode BESO [241] est très similaire, si ce n'est qu'elle considère la possibilité d'ajouter de nouveaux éléments aux abords des éléments les plus contraints du domaine. Yang et al. [92] ont proposé d'extrapoler linéairement le champ des déplacements afin d'estimer l'énergie de déformation des éléments considérés comme vides. Les deux méthodes ESO et BESO ont été utilisées dans de nombreux domaines d'application, comme la mécanique [83, 106, 191] et le génie civil [1, 2]. Ces deux méthodes ont été très largement utilisées dans la littérature, notamment parce qu'elles sont simples à implémenter, et que le concept est assez simple à appréhender [241].

2.1.3.2.3/ Optimisation par lignes de niveaux (level set) Développée initialement par Allaire et al. [109], la méthode des courbes de niveaux, ou méthode level-set, est basée sur les méthodes d'optimisation de forme. Ici, et comme dit dans [139], les formes sont capturées sur un maillage fixe et représentées par la courbe de niveau d'un champ scalaire. On évite ainsi les problèmes habituellement rencontrés dans le domaine de l'optimisation topologique, notamment les instabilités numériques et les méthodes de re-maillage extrêmement coûteuses. La vitesse du front de propagation est calculée en utilisant la dérivée de forme de l'optimisation de frontière classique. Allaire et al. insistent ici sur le fait que cette méthode n'est pas une procédure d'optimisation topologique, mais plutôt une méthode d'optimisation de forme. Ceci dit, le couplage de level-set avec une méthode de gradient topologique autorise la création de trous dans la matière, ce qui en fait donc une méthode d'optimisation topologique.

Cette méthode présente notamment les avantages suivants :

- La topologie peut être grandement modifiée au cours des itérations. Il n'est pas nécessaire de restreindre les modifications à chaque itération.

- Son coût en terme de temps de calcul CPU est assez faible,
- Elle permet de traiter différentes sortes de fonctions objectif, ainsi que des modèles physiques très généraux. Elle a notamment été grandement utilisée dans le domaine de la mécanique [105] et de l'électromagnétisme [210].
- Elle s'avère aussi efficace que la méthode d'homogénéisation. Notons tout de même qu'elle donne de meilleurs résultats, dans le sens où les topologies optimales sont très souvent fabriquables.

Cependant, contrairement à la méthode d'homogénéisation précédemment citée, il n'existe pas de preuve que cette méthode converge vers l'optimum global de la fonction objectif considérée [163]. Notons aussi que cette méthode est très dépendante de la configuration initiale [109], comme on le voit dans la figure 2.6 suivante.

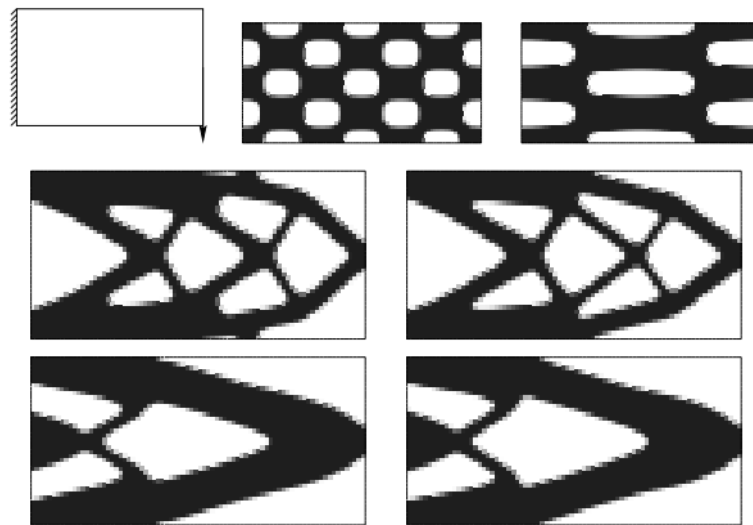


FIGURE 2.6 – (haut) Conditions aux limites et deux initialisations différentes de la topologie. (milieu) Solutions à la 50ème et la 100ème itération pour la première initialisation. (bas) Solutions à la 50ème et la 100ème itération pour la seconde initialisation.

2.2/ MÉTHODES METAHEURISTIQUES

2.2.1/ GÉNÉRALITÉS

De nombreuses méthodes metaheuristiques ont été développées ces dernières années. En effet, comme nous l'avons vu dans la partie 2.1.1.1, les méthodes de résolution classiques que nous avons étudiées précédemment ont tendance à rester piégées dans les optima locaux que présente la fonction objectif. On pourrait bien sûr utiliser ces méthodes classiques de nombreuses fois en changeant les conditions initiales (c'est à dire la position d'un premier point dans l'espace des solutions) jusqu'à ce que l'on trouve une solution qui peut être considérée comme étant la meilleure. Cependant, il faudrait effectuer ce processus une infinité de fois avant de pouvoir affirmer que la solution trouvée est effectivement la meilleure.

Afin de pallier à ce problème, deux idées novatrices se sont imposées. La première est d'accepter de temps en temps une dégradation temporaire de la situation, afin de se déplacer dans l'espace des solutions à la recherche d'une meilleure solution. La seconde

est l'utilisation d'un nombre de solutions conséquent qui se "déplacent" dans l'espace des solutions à la recherche d'un optimum local. Ainsi, en bouclant sur toutes ces solutions (appelées "individus" dans certains algorithmes, "particules" dans d'autres), on effectue, à chaque itération, autant d'évaluations de la fonction objectif qu'il y a de potentielles solutions disponibles. Cela permet d'étudier le domaine plus en détails, et d'obtenir plus rapidement des solutions. De plus, elles tiennent compte de paramètres aléatoires, heuristiques donc, qui permettent une certaine intelligence - ou tout du moins une liberté certaine - à l'algorithme qui peut tout à fait trouver des solutions inattendues *a priori* dans le domaine de recherche.

Définition 3 : Méthode heuristique

En optimisation combinatoire, en théorie des graphes, en théorie de la complexité des algorithmes, et en intelligence artificielle, une heuristique est une méthode de calcul qui fournit de manière relativement rapide (c'est à dire dans un temps que l'on peut considérer comme polynomial) une solution réalisable, c'est à dire assez proche de l'optimum cherché, pour un problème d'optimisation NP-difficile [226].

Cependant, nombre de ces méthodes ont été développées pour un type de problème en particulier, et ne sont donc pas suffisamment généralistes pour être efficaces à coup sûr.

L'arrivée des méthodes metaheuristiques permet ainsi de réconcilier les problèmes continus et les problèmes discrets, puisqu'elles sont efficaces sur ces deux types de domaines, comme nous le verrons notamment dans la partie 2.2.3. De par leur capacité à résoudre des problèmes d'optimisation dits difficiles pour lesquels on ne connaît pas de méthode classique ou directe plus efficace, elles se sont rapidement imposées dans le domaine de l'optimisation. Du fait qu'elles sont très facilement adaptables à tous types de problèmes, il existe un très grand nombre de metaheuristiques différentes, allant de la simple recherche probabiliste à des algorithmes complexes de recherche globale.

Du grec *meta* signifiant "au-delà" ou "un niveau au dessus" et de *heuriskein* qui signifie "trouver", les metaheuristiques sont généralement des algorithmes stochastiques itératifs qui progressent vers un optimum global, c'est à dire l'extremum global d'une fonction, par échantillonnage de la fonction objectif, sans nécessiter de calcul des sensibilités. Elles se comportent donc comme des algorithmes de recherche, tentant d'étudier l'ensemble d'un domaine afin d'en trouver efficacement les extrema.

Les méthodes metaheuristiques les plus utilisées conjointement dans les domaines de l'industrie et de la recherche sont la méthode du recuit simulé, la recherche avec tabous, l'algorithme d'Optimisation par Essaim Particulaire (Particle Swarm Optimization : PSO), les Algorithmes Evolutionnaires (AEs) et l'algorithme d'optimisation par colonie de fourmis (Ant Colony Optimization : ACO).

2.2.1.1/ PROPRIÉTÉS

Il est important de noter que l'utilisation des méthodes metaheuristiques ne nécessite pas de connaître la forme du domaine étudié, ni l'évolution de la fonction objectif dans son domaine, pour fonctionner.

Du fait qu'il existe de très nombreuses méthodes metaheuristiques différentes, il est très difficile de trouver une définition claire et précise de ces méthodes. Cependant, l'en-

semble des méthodes metaheuristiques partagent les caractéristiques suivantes :

- Elles sont, pour la plupart, stochastiques, ce qui leur permet de faire face à l'explosion combinatoire des possibilités de solution.
- Elles considèrent généralement une population de solutions au lieu d'une seule - en tout cas pour ce qui concerne les méthodes metaheuristiques basées sur le fonctionnement de populations² - ce qui leur permet d'une part d'analyser simultanément l'ensemble du domaine des solutions, et donc de ne pas rester bloqué dans un optimum local, mais aussi de résoudre simplement les problèmes d'optimisation multi-objectifs.
- Généralement développées pour des problèmes discrets et adaptées par la suite à des problèmes continus, elles ont l'avantage d'être directes, c'est à dire qu'elles ne recourent pas au calcul, souvent problématique, des sensibilités.
- Elles sont inspirées d'analogies avec la physique, la biologie, ou l'éthologie, ce qui leur permet d'une part d'être facilement accessibles et utilisables, notamment dans le monde de l'industrie, puisque l'efficacité est un critère déterminant dans le choix d'une méthode de calcul, mais aussi d'être relativement simples à implémenter.
- Les algorithmes définissant ces méthodes sont généralement donnés en fonction de paramètres, certains constants, d'autres heuristiques, difficiles à borner et à définir. En effet, l'efficacité de ces méthodes est très fortement dépendante des paramètres choisis, et il est nécessaire, avant d'utiliser une metaheuristique, de connaître parfaitement quels sont les effets de chaque paramètre sur le fonctionnement de la méthode en question [187].
- Il n'existe aucune démonstration mathématique de leur capacité à converger vers le minimum global de la fonction objectif. Seule l'expérience a montré qu'elles en sont capables.

2.2.1.2/ AVANTAGES ET INCONVÉNIENTS

Bien que très efficaces sur un très grand nombre de problèmes différents, il est aujourd'hui très difficile de prévoir l'efficacité d'une méthode en particulier sur un problème donné. Ces méthodes ne s'excluent pas mutuellement [121], et la tendance actuelle est donc à l'émergence de méthodes hybrides, qui s'efforcent de tirer parti des qualités de chacune des méthodes en les combinant, soit de manière parallèle, soit de manière séquentielle.

Un autre avantage de ces méthodes est qu'elles sont facilement adaptables à tous types de problèmes. De nouvelles versions de chacune des méthodes metaheuristiques ont été développées, afin qu'elles puissent à la fois être efficaces non seulement sur les problèmes continus et les problèmes discrets, mais aussi sur les problèmes multi-objectifs, dynamiques, ou binaires comme nous le verrons lors de l'application de ces méthodes à l'optimisation topologique.

De par leurs coefficients heuristiques, les metaheuristiques ont la capacité à s'extraire des minima locaux qui peuvent être présents sur le domaine de recherche. Comme nous l'avons vu sur la figure 2.1, les méthodes classiques restent très souvent bloquées dans les minima locaux, parce qu'elles ne considèrent qu'un seul point du domaine et qu'elles ne se déplacent qu'en cherchant des directions de descente dans le domaine. Au contraire, les méthodes metaheuristiques analysent simultanément l'ensemble du do-

2. Dans la littérature anglophone, elles sont appelées *population-based metaheuristic methods*.

maine de recherche. De plus, alors que certaines méthodes classiques refusent systématiquement tout point du domaine qui serait moins adapté à la fonction objectif que le précédent, les méthodes metaheuristiques dites "de voisinage" (comme la méthode du recuit simulé, et la méthode de recherche avec tabou, par exemple) acceptent de temps en temps un mouvement de recul. En effet, les solutions trouvées au cours des itérations peuvent tout à fait être moins adaptées que les précédentes du fait des coefficients aléatoires qui définissent le fonctionnement des algorithmes. Dès lors, il devient possible de s'extraire du piège que représente un minimum local, pour partir explorer une nouvelle zone du domaine plus prometteuse, bien qu'elle ne donne pas, à première vue, des résultats très satisfaisants.

De la même manière, les méthodes dites "distribuées" (voir parties 2.2.2.1, 2.2.2.2 ainsi que toute la partie 2.2.3) proposent des mécanismes permettant la sortie d'un piège local. Ces mécanismes, généralement appliqués de manière aléatoire - et que l'on peut appeler au mécanisme de mutation utilisé par les algorithmes évolutionnaires - permettent d'explorer une nouvelle région du domaine si celle étudiée n'est pas suffisamment satisfaisante.

D'autre part, l'ensemble des méthodes metaheuristiques fonctionne avec un système de mémoire qui leur permet, tout en explorant de nouvelles régions du domaine, de conserver en mémoire les meilleurs points trouvés jusqu'à présent. Cette mémoire, bien qu'assez lourde du point de vue logiciel, permet aux algorithmes d'être efficaces, tant lors des phases d'exploration du domaine, que lors des phases d'exploitation des zones prometteuses.

Par contre, il est admis qu'aucune metaheuristique ne peut être considérée comme étant meilleure qu'une autre, dans l'absolu. En effet, le théorème du [no free lunch](#) développé par Wolpert et al. [75] démontre qu'une méthode metaheuristique ne peut pas prétendre être efficace sur tous les problèmes, bien que certaines configurations puissent apparaître comme étant plus efficaces sur certaines classes de problèmes en particulier. Ainsi, dans les parties suivantes nous montrerons que chacune des metaheuristiques développées a évolué au cours du temps, que des recherches les ont modifiées afin qu'elles deviennent plus performantes.

2.2.1.3/ CLASSIFICATION DES MÉTHODES METAHEURISTIQUES

De nombreux classements différents ont été proposés dans la littérature, afin de situer la méthode metaheuristique la plus adaptée à un problème d'optimisation donné, ou au moins une classe de problèmes donnée. La classification présentée dans figure 2.7 nous a permis notamment de choisir judicieusement la méthode à employer en fonction de la classe de problèmes à résoudre.

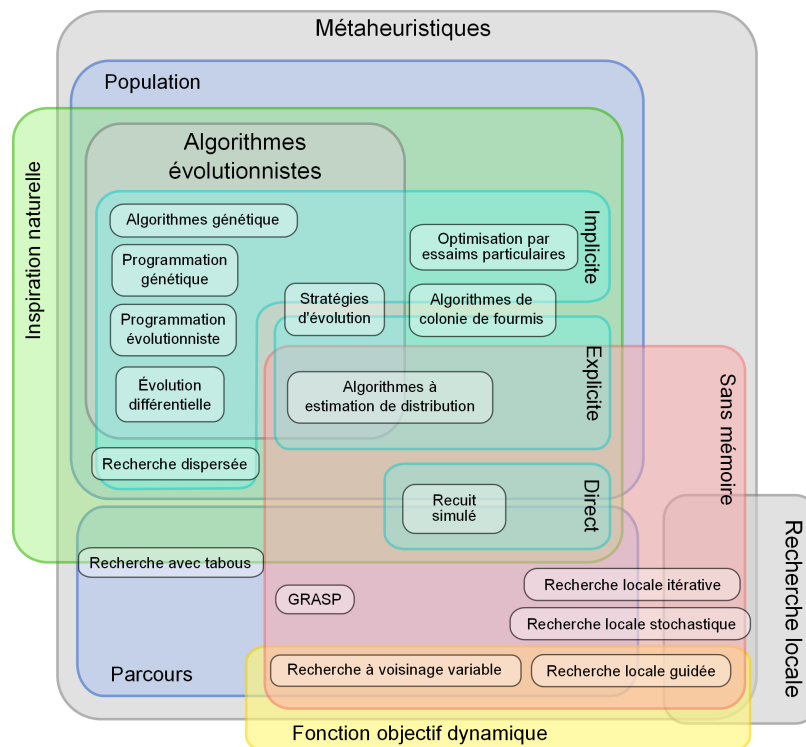


FIGURE 2.7 – Classification des méthodes metaheuristiques en fonction de la classe du problème à résoudre

2.2.2/ QUELQUES MÉTHODES METAHEURISTIQUES

2.2.2.1/ ANT COLONY OPTIMIZATION

2.2.2.1.1/ Généralités Développé pour la première fois par M. Dorigo [67], l'algorithme d'optimisation par colonies de fourmis (appelé *Ant Colony Optimization* (ACO) dans la littérature anglophone) est aujourd'hui l'un des plus utilisés pour les problèmes d'optimisation combinatoire difficiles, et notamment dans le domaine de la théorie des graphes (voir partie 2.3.2). En effet, il a tout d'abord été développé pour résoudre efficacement le problème du voyageur de commerce, c'est à dire pour trouver le chemin le plus court dans un graphe, orienté ou non. Cet algorithme s'inspire du comportement social des fourmis dans la nature l'on peut qualifier d'**auto-organisé**.

Définition 4 : L'auto-organisation

Décrit dans les domaines de la physique, de la biologie et de la sociologie, **l'auto-organisation** est un processus dans lequel un modèle de niveau global émerge uniquement d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre les composants du système sont suivies en utilisant uniquement des informations locales, sans référence au modèle global [93] [121]. Une propriété **émergente** est une propriété qui peut apparaître de manière imprévue, sans avoir été explicitement déterminée au préalable. On peut par exemple considérer que de nombreux phénomènes de sociétés sont résultants de propriétés émergentes de cette même société, puisque c'est finalement le mouvement local de chaque individu de la société qui construit un modèle global, une manière de fonctionner globale qui définit les grands phénomènes sociétaux.

2.2.2.1.2/ Origines de l'algorithme L'idée originale de cet algorithme provient donc de l'observation de la manière dont les fourmis s'organisent pour exploiter les ressources de nourriture alentour. En effet, bien que chacune des fourmis ait des capacités cognitives limitées, l'ensemble de leur société est capable de trouver le chemin le plus court entre une source de nourriture et la fourmilière. Les observations faites par les biologistes Dorigo et Maniezzo dans [67] ont permis de déterminer qu'une colonie de fourmis ayant le choix entre plusieurs chemins de longueurs différentes menant à la même source de nourriture avait tendance à utiliser le chemin le plus court.

Le problème crucial auquel se sont confrontés les chercheurs qui ont développé cet algorithme est donc le processus de communication de composants locaux au sein d'un système global. C'est à dire la manière dont les composants interagissent entre eux pour produire un modèle complexe, que chacun des composants ne pourrait pas reproduire lui-même. Dans le cadre de la biologie du comportement, des individus peuvent communiquer de plusieurs manières différentes. La première est l'utilisation de signaux que l'on pourrait qualifier de **directs**, c'est à dire l'utilisation de moyens spécifiques pour transmettre une information. La parole, l'écriture, sont des exemples de signaux de communication directs. Au contraire, la deuxième manière de communiquer est l'utilisation de signaux que l'on pourrait qualifier d'**indirects**, c'est à dire l'utilisation, intentionnelle ou non, d'indices qui modifient l'environnement. De la même manière, l'information peut provenir directement d'autres individus, ou bien provenir de l'état d'un travail en cours. Ainsi, les systèmes auto-organisés sont particulièrement intéressants dans le sens où la **stigmergie** s'auto-organise elle-même et peut même être adaptative.

Définition 5 : La stigmergie

A l'origine de la création de l'algorithme d'optimisation par colonie de fourmis, **la stigmergie** est précisément définie comme une forme de communication passant par le biais de modifications de l'environnement. Les biologistes différencient la stigmergie quantitative de son homologue qualitative, mais les concepts sont toutefois assez similaires.

Un autre aspect très important de la définition de l'algorithme par colonie de fourmis est le concept de **contrôle décentralisé**. En effet, dans un système auto-organisé, il n'y a pas de prise de décision collective, suivie d'ordres et d'actions prédéterminées, et c'est

d'ailleurs en ce sens que tout système organisé peut être qualifié d'intelligent (artificiel s'il s'agit d'un algorithme, naturel s'il s'agit d'un groupe d'êtres vivants). De la même manière, dans un système décentralisé, que l'on peut considérer de manière générale comme un sous-ensemble de l'ensemble des systèmes auto-organisés, chaque individu dispose d'une vision *locale* de son environnement, et ne connaît donc ni le domaine dans sa globalité, ni la topologie de la population. Les avantages des systèmes décentralisés sont :

- La robustesse, du fait que l'ensemble du système peut continuer à fonctionner, même si l'une des parties individuelles est en panne ;
- La flexibilité, car ils sont tout à fait efficaces tant sur des domaines statiques que sur des domaines dynamiques (c'est à dire des domaines où la fonction objectif évolue au cours du temps et où, donc, la valeur de la fonction objectif pour une position donnée peut changer).

2.2.2.1.3/ Définition du modèle biologique Ainsi, basé sur les informations précédemment explicitées, l'algorithme par colonie de fourmis est basé sur le modèle suivant (voir figure 2.8) :

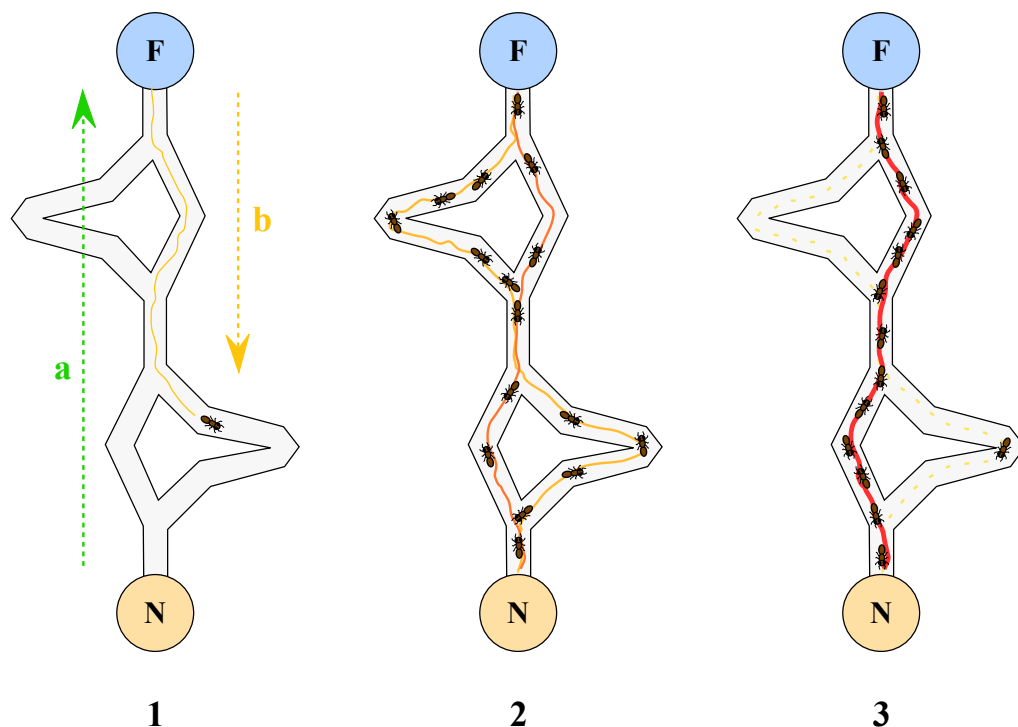


FIGURE 2.8 – (1) La première fourmi trouve la source de nourriture F, via un chemin quelconque (a), puis dépose une piste de phéromones en revenant au nid par le chemin le plus court (b). (2) Les fourmis empruntent indifféremment l'un des quatre chemins, mais renforcent tout de même la piste du chemin le plus court lors de leur retour au nid. (3) Les fourmis empruntent le chemin le plus court. Les chemins les plus longs perdent leur pistes de phéromones qui s'évaporent au court du temps.

1. Une fourmi parcourt au hasard l'environnement autour de la colonie.
2. Si cette fourmi trouve une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones (elle modifie

donc l'environnement de son chemin de retour).

3. Les phéromones étant attractives, les fourmis qui passent à proximité de la piste précédemment déposée vont avoir tendance à suivre, de façon plus ou moins précise, cette piste.
4. En revenant au nid, ces nouvelles fourmis vont elles aussi laisser sur leur chemin une piste de phéromones. La piste est donc **renforcée**, elle comporte plus de phéromones, et est donc encore plus attractive.
5. Si deux pistes sont possibles pour atteindre la même source de nourriture, la plus courte sera parcourue par plus de fourmis que la plus longue. Cette piste sera donc de plus en plus renforcée, et donc de plus en plus attractive pour de nouvelles fourmis susceptibles de l'emprunter et de la renforcer à nouveau.
6. Les phéromones sont volatiles. Au cours du temps, la longue piste finira par disparaître et ne comportera plus du tout de phéromones.
7. A terme, l'ensemble des fourmis de la colonie a donc déterminé et "choisi" (au sens où la colonie est un système décentralisé, comme expliqué dans la partie précédente) la piste la plus courte.

Les systèmes naturels auto organisés tels que les colonies de fourmis reposent donc aussi sur des systèmes de rétroactions. En effet, le dépôt de phéromones attirant de nouvelles fourmis susceptibles elles-mêmes de déposer de nouvelles phéromones sur cette même piste constitue une **rétroaction positive**. De la même manière, la dissipation des pistes les moins empruntées par évaporation empêchant le système de dégénérer au cours des itérations constitue une **rétroaction négative**.

2.2.2.1.4/ Application au problème du voyageur de commerce Basées sur les observations biologiques présentées dans le paragraphe précédent, on obtient des règles de fonctionnement qui bornent les mouvements autorisés par l'algorithme. Développé pour résoudre efficacement le problème du voyageur de commerce, cet algorithme est donc défini par les règles suivantes :

1. Chaque fourmi ne peut visiter une ville qu'une seule fois ;
2. Plus une ville est loin de la fourmi, moins elle a de chances d'être visitée. C'est à dire que les fourmis choisissent en priorité les villes qui sont visibles **à leur échelle**.
3. Plus l'intensité de la piste de phéromones disposée sur l'arête entre deux villes est grande, plus le trajet aura de chances d'être choisi ; Une fois son trajet terminé, la fourmi dépose sur chacune des arêtes parcourues une quantité de phéromones dépendant de la longueur du trajet. Plus le trajet est court, plus grande est la quantité de phéromones déposée, et inversement.
4. Les pistes de phéromones s'évaporent à chaque itération.

Le fonctionnement de l'algorithme sur le problème du voyageur de commerce peut être illustré par la figure 2.9 suivante.

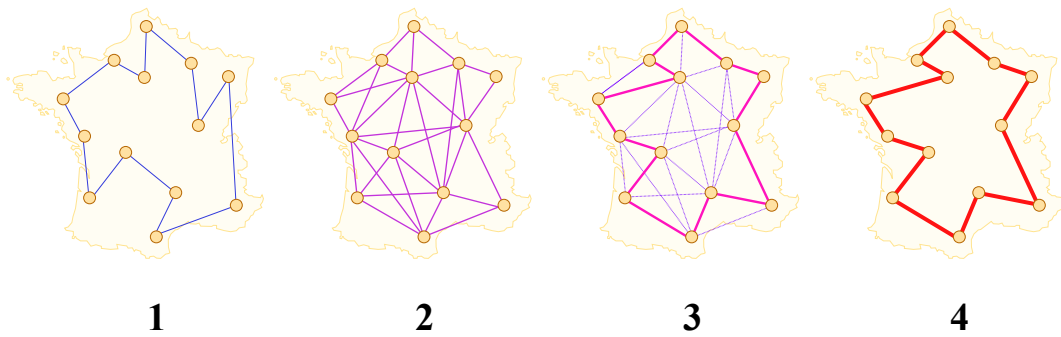


FIGURE 2.9 – (1) Une fourmi choisit un trajet au hasard et trace une piste de phéromones. (2) L'ensemble des fourmis parcourt un certain nombre de trajets, chaque fourmi déposant une certaine quantité de phéromones sur son trajet dépendant de la longueur du trajet qu'elle a emprunté. (3) Alors que les pistes les moins empruntées s'évaporent, les pistes les plus empruntées, et donc les plus courtes, s'intensifient. (4) Le meilleur chemin est finalement le plus court qui a été choisi de manière non intentionnelle par la colonie.

2.2.2.1.5/ Description formelle et algorithmique Soit une colonie de m fourmis et N un ensemble de villes avec $\text{card}(N) = |N|$. Pour chaque fourmi k , le trajet entre une ville i et une ville j dépend de :

1. **L'ensemble des villes** déjà visitées, qui définit donc les villes que la fourmi k peut possiblement visiter lorsqu'elle est sur la ville $i : J_i^k$.
2. **La visibilité** de la fourmi qui est l'inverse de la distance entre la ville i et la ville j et qui est définie par $\eta_{ij} = \frac{1}{d_{ij}}$. Ce paramètre permet à la fourmi de choisir, dans la mesure du possible, les villes les plus proches d'elle à chaque instant.
3. **L'intensité de la piste**, c'est à dire la quantité de phéromones présente à chaque instant sur les pistes que la fourmi peut potentiellement emprunter à cet instant. C'est en quelque sorte une mémoire globale des trajets qui ont d'ores et déjà été effectués par les fourmis, et qui évolue presque de manière continue au cours de l'algorithme [121].

La règle de déplacement des fourmis de ville en ville dépend donc de tous ces paramètres. On l'appelle la "règle aléatoire de transition proportionnelle" et elle est définie par

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha (\eta_{ij})^\beta}{\sum_{l \in J_i^k} (\tau_{il}(t))^\alpha (\eta_{il})^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (2.19)$$

avec α et β deux paramètres contrôlant l'importance relative de l'intensité de la piste τ_{ij} et de la visibilité η_{ij} . Si $\alpha = 0$, seule la visibilité de la piste est prise en compte dans le choix de déplacement des fourmis. Au contraire, si $\beta = 0$ seules les phéromones ont une influence sur le choix de déplacement des fourmis. Pour éviter qu'un trajet ne soit sélectionné trop tôt lors de l'exécution de l'algorithme, il est important de sélectionner correctement ces deux paramètres. On joue ainsi sur le choix entre un comportement qui favorise la diversification et un comportement qui favorise l'intensification des pistes déjà sélectionnées. Comme on le verra dans la partie 2.2.3, ces paramètres correspondent à l'équivalent des paramètres c_1 et c_2 de l'algorithme d'optimisation par essaim particulaire PSO, qui jaugent le comportement de l'algorithme entre ses capacités d'exploration de

nouvelles contrées du domaine, et ses capacités d'exploitation des zones prometteuses déjà trouvées.

A chaque itération, chaque fourmi k a effectué un tour complet des villes de l'ensemble N , et a déposé sur chacun des chemins empruntés lors de son parcours une quantité de phéromones $\Delta\tau_{ij}^k(t)$ qui dépend, comme vu précédemment, de la longueur du trajet effectué (c'est à dire de la qualité de la solution proposée par la fourmi, puisque l'on cherche le chemin le plus court possible, plus court est le chemin, plus la solution est adaptée), et qui est donnée par la formule suivante

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases} \quad (2.20)$$

avec $T^k(t)$ le trajet déjà effectué par la fourmi k à l'itération t , $L^k(t)$ la longueur du tour effectué par la fourmi, et Q un paramètre fixé.

Comme expliqué précédemment, pour éviter que le processus ne s'emballe, et pour que l'algorithme de sélection du plus court chemin soit efficace, il est nécessaire que les pistes de phéromones s'évaporent à chaque itération. La règle de mise à jour des pistes est donc la suivante

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2.21)$$

avec $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$ et m est le nombre de fourmis.

La quantité initiale de phéromones sur les arêtes est déterminée grâce à une distribution uniforme d'une petite quantité $\tau_0 \geq 0$.

L'algorithme simplifié de résolution du problème du voyageur de commerce est défini dans l'algorithme 2.

Algorithm 2 L'algorithme "Ant System"

```

for  $t = 1, \dots, t_{max}$  do
  for chaque fourmi  $k = 1, \dots, m$  do
    Choisir une ville au hasard
    for chaque ville non visitée  $i$  do
      Choisir une ville  $j$ , dans la liste  $J_i^k$  des villes restantes, selon la formule (2.19)
    end for
    Déposer une piste  $\Delta\tau_{ij}^k(t)$  sur le trajet  $T^k(t)$  conformément à l'équation (2.20).
  end for
  Evaporer les pistes selon la formule (2.21).
end for

```

2.2.2.1.6/ Quelques applications et améliorations Comme il l'a été dit dans la partie 2.2.1.2, bien que les méthodes metaheuristiques soient très efficaces, elles peuvent n'être efficaces que sur certains types de problèmes d'optimisation. C'est la raison pour laquelle de nombreux algorithmes hybrides sont proposés afin de combiner les avantages d'algorithmes différents, tout en essayant d'éviter leurs inconvénients respectifs. Par exemple, Holden et al. [145, 170, 181] ont proposé un algorithme hybride tirant profit à la fois des capacités de l'algorithme d'optimisation par essaim particulaire (PSO) et de celles de l'algorithme par colonies de fourmis (ACO) pour effectuer des classements de données et du data mining. De la même manière, Iqbal et al. [158] ont proposé un

algorithme hybride PSO/ACO. En effet, dans l'algorithme PSO classique, les particules n'exploitent pas la mémoire collective de l'essaim, mais seulement leur mémoire personnelle, et éventuellement celle de leur voisinage. Ceci implique que l'algorithme peut explorer à nouveau des zones qui ont d'ores et déjà été exploitées auparavant. Dans cet algorithme, des principes tirés de l'algorithme ACO sont utilisés afin d'utiliser au mieux la mémoire collective de l'essaim, en prenant en compte les zones déjà explorées lors du déplacement de l'essaim.

En outre, la méthode développée par Romdhane et al. [233] propose de partitionner un graphe (cf. partie 2.3.2), par exemple une population ou un essaim (cf. partie 2.2.3) en plusieurs sous graphes. Ainsi, chacun des sous-graphes peut explorer une partie du domaine, avant que les meilleures particules de chaque sous-graphe ne soient récupérées pour créer un nouvel essaim très efficace, puisqu'hétérogène et réparti aux quatre coins du domaine. Dans cette méthode, l'algorithme ACO définit les sous graphes de la même manière qu'il résout le problème du voyageur de commerce (voir algorithme 2), c'est à dire en se déplaçant de particule en particule, pour déterminer quels sont les sous-ensembles qui émergent du modèle global. Notons ici que Romdhane et al. [233] insistent sur la nécessité de bien considérer un graphe orienté en tant que tel. En effet, la création de groupes ou de communautés peut être faussée si l'on ne considère pas correctement l'orientation du graphe.

2.2.2.2/ LES ALGORITHMES ÉVOLUTIONNAIRES

Selon la théorie Darwinienne, les mécanismes qui régissent l'évolution des êtres vivants sur Terre consistent essentiellement en une [sélection naturelle](#) basée sur la compétition, qui engendre l'extinction des espèces les moins adaptées à leur environnement (notons d'ailleurs, en référence à la partie précédente que, d'après des recherches récentes, les fourmis sont présentes sur Terre depuis plus de 140 millions d'années, et qu'elles font ainsi partie des êtres vivants les plus compétitifs et qui se sont le mieux adaptés aux cours de ces millions d'années). Cependant, les recherches récentes en génétique ont montré que les seuls mécanismes de sélection ne peuvent expliquer complètement l'évolution des espèces sur Terre.

Les Algorithmes Evolutionnaires (appelés Evolutionary Algorithms et notés EAs dans la littérature anglophone) sont donc des techniques inspirées par les découvertes concernant l'évolution biologique des espèces apparues à la fin des années 50 [3]. Parmi plusieurs approches différentes, les algorithmes génétiques (appelés Genetic Algorithms et notés GAs dans la littérature anglophone) en constituent l'exemple le plus probant, à la suite notamment de la parution en 1989 de l'ouvrage de D.E. Goldberg [35].

Remarque : De nombreux chercheurs utilisent encore le terme d'algorithmes génétiques pour définir les algorithmes évolutionnaires, bien que ces derniers n'aient en réalité que peu de points communs avec la méthode des algorithmes génétiques développée initialement par Holland [12] et Goldberg [35]. Ici, nous ferons bien la distinction entre les algorithmes génétiques qui ne représentent qu'un sous-ensemble des algorithmes évolutionnaires.

Ces algorithmes ont tout d'abord suscité un intérêt relativement limité, du fait de leur complexité qui engendrait un important coût de calcul. Avec le développement des calculateurs, ils ont connu un essor considérable, notamment grâce à l'apparition des architectures massivement parallèles qui ont permis la parallélisation de nombreux calculs

évolutionnaires, et donc l'obtention de résultats en un temps polynomial [121].

2.2.2.2.1/ Généralités

Dans la théorie de la génétique, on considère plusieurs mécanismes à l'origine de la sélection naturelle développée par Darwin [225] :

- Le phénotype d'un organisme est codé en grande partie dans son génotype. Depuis peu, on commence à considérer aussi que l'environnement peut avoir des effets sur le phénotype.
- Chaque population d'individus est constituée d'individus différents.
- Ce sont justement ces différences entre les individus qui leur confèrent une certaine capacité d'adaptation à leur environnement.
- Chaque organisme transmet une partie de ses caractéristiques à sa descendance, via la génétique.
- Les individus les plus adaptés se reproduisent plus facilement (notamment parce qu'ils sont plus adaptés à l'environnement, et que donc ils réunissent toutes les conditions nécessaires à la reproduction), et ont plus de chances de transmettre leurs caractéristiques à la population.

Basés sur ces observations, les algorithmes génétiques considèrent donc une population de potentielles solutions (les individus) au problème d'optimisation posé. Les individus sont donc caractérisés par leurs **génotypes** respectifs, qui leur confèrent, via leur **phénotype**, une certaine adaptation au domaine considéré (la valeur de la fonction objectif appelée aussi **fitness**). Afin de promouvoir l'adaptation au domaine, les individus les plus adaptés ont plus de chances (au sens probabiliste du terme) de se reproduire au cours des **générations** (les itérations de l'algorithme), afin de transmettre leur génotype au sein de la population.

La manipulation des individus est réalisée grâce à deux types d'opérateurs qui permettent de mettre en œuvre les considérations précédentes : les opérateurs de **sélection**, et les opérateurs de **variation**.

Les opérateurs de sélection

A chaque génération, les individus peuvent se reproduire, survivre ou disparaître, du fait des deux opérateurs de sélection suivant [121] :

- La sélection pour la reproduction, appelée aussi **sélection**, qui détermine le nombre de descendants de chaque individu à chaque génération. Les individus les plus adaptés ont donc plus de chance d'être **sélectionnés** que les autres.
- La sélection pour le remplacement, appelée aussi **remplacement**, qui détermine quels individus devront disparaître de la population parce qu'ils ne sont pas suffisamment adaptés à l'environnement. On remarque que, grâce à ce processus, la taille de la population est contrôlée. Elle peut soit restée constante, soit évoluer selon une politique précédemment définie.

Pour déterminer les meilleurs individus de la population, ils sont en quelque sorte classés en fonction de leur performance, c'est à dire de la valeur de la fonction objectif (leur phénotype) induite par leur génotype. Ceci implique que la valeur de la fonction objectif de chaque enfant doit être calculée à chaque itération, ce qui peut être assez lourd en terme de temps de calcul.

Les opérateurs de variation

Afin que l'algorithme puisse faire évoluer les populations dans le but de trouver de meilleures solutions au cours des générations, il est nécessaire d'appliquer des opérateurs de variation, dont le but est de transformer les populations. Ils sont généralement classés en deux catégories :

- La **mutation**, qui modifie le génotype d'un individu pour en obtenir un autre (on pourrait apparenter cela à un processus d'évolution des espèces qui serait extrêmement rapide).
- Le **croisement** (appelée aussi "crossover" dans la littérature anglophone), qui engendre des enfants à partir du croisement du génotype de deux parents.

Ces opérateurs sont inspirés de la reproduction sexuée, et ils peuvent être étendus de manière numérique. On peut par exemple imaginer que de nombreux parents (dont le nombre est supérieur à deux) génèrent, de par le croisement de leurs génotypes respectifs, un enfant.

Le problème de ces algorithmes réside dans la représentation de la population. En effet, elle est intimement liée au type de problème d'optimisation à résoudre. Ainsi, si l'on veut résoudre un problème d'optimisation continu dans un domaine de $\mathbb{R}^n$, chaque individu sera représenté par un vecteur de $\mathbb{R}^n$. Au contraire, si l'on souhaite résoudre un problème d'optimisation combinatoire ou séquentielle, comme le problème du voyageur de commerce, on pourra considérer que les individus sont des vecteurs d'entiers, dont chaque composante est le numéro d'une ville. Ainsi, chaque vecteur représente une tournée, c'est à dire une potentielle solution au problème d'optimisation. Il est donc important, avant d'utiliser un algorithme évolutionnaire, de bien spécifier le type de problème d'optimisation à résoudre, et donc la représentation des individus dans l'espace des solutions, afin que les opérateurs soient adaptés.

2.2.2.2.2/ Algorithme Les algorithmes évolutionnaires sont tous basés sur l'algorithme 3 suivant, au demeurant assez simple, bien qu'il puisse se compliquer suivant le domaine sur lequel il est appliqué.

Algorithm 3 L'algorithme évolutionnaire

Construction d'une population initiale
Evaluation des individus de cette population
while Le critère d'arrêt n'est pas vérifié **do**
 Sélection d'une partie de la population,
 Reproduction des individus sélectionnés,
 Mutation des descendants ainsi créés,
 Evaluation du degré d'adaptabilité de chaque individu,
 Remplacement de la population initiale par une nouvelle population
end while

L'algorithme peut aussi être représenté de manière géométrique, comme on le voit sur la figure 2.10 suivante.

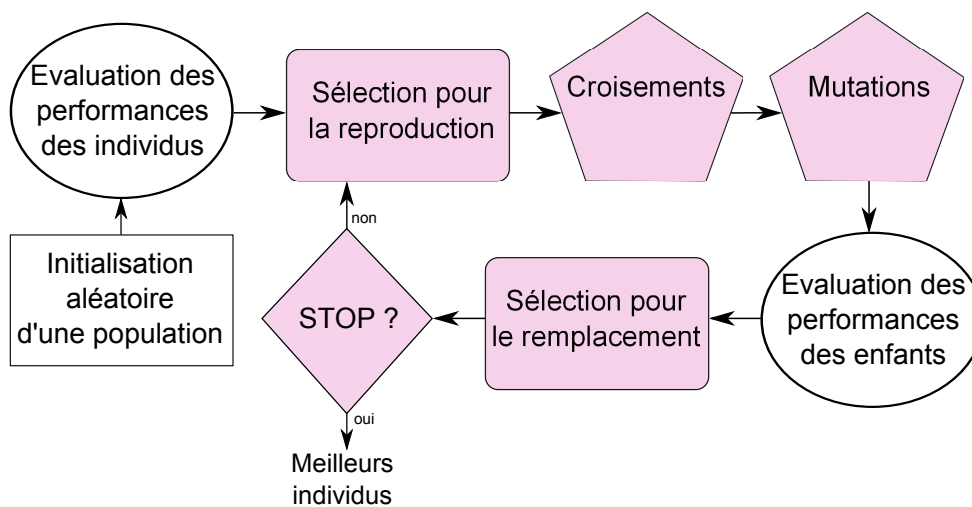


FIGURE 2.10 – Pseudo code d'un algorithme évolutionnaire

Sur ce pseudo-code, les blocs roses représentent les parties de l'algorithme relevant effectivement d'un algorithme évolutionnaire.

2.2.2.2.3/ Application à un domaine binaire Bien que les opérateurs dépendent de la représentation de la population, la meilleure manière de les représenter est de considérer un domaine binaire, qui correspond en tout point à la représentation génotype/phénotype qui existe dans la nature. Ainsi, le génotype de chaque individu est représenté par une chaîne de caractères binaires, ou plus généralement de symboles d'un alphabet à faible cardinalité [121] alors que le phénotype est représenté par la *fitness* de l'individu par rapport à la fonction objectif considérée.

Le croisement

Dans un domaine binaire, il existe trois variantes du croisement :

- Le croisement "un point",
- Le croisement "deux points",
- Le croisement uniforme.

Dans un domaine binaire, le croisement "un point" peut être représenté à la manière de la figure 2.11.

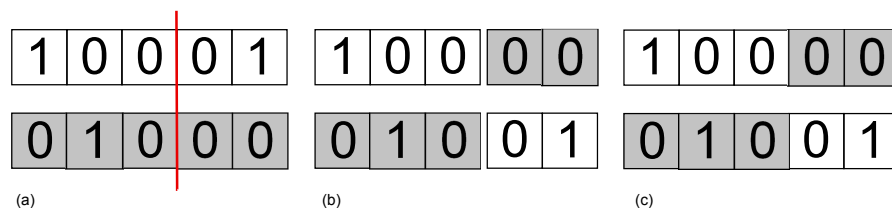


FIGURE 2.11 – Croisement "1 point" de deux génotypes composés de 5 bits chacun : (a) Un point de coupure est choisi. (b) On réalise l'échange des parties coupées. (c) Résultat du croisement.

Le croisement deux points est réalisé exactement de la même manière, si ce n'est que deux points de coupure sont choisis dans chaque génome.

Le croisement uniforme peut être vu comme un croisement multipoints dont le nombre de coupures est indéterminé a priori, il peut même être choisi de manière aléatoire pour favoriser l'hétérogénéité des populations [27]. Pour cela, on utilise un "masque de croisement" qui détermine où auront lieu les coupures. Le masque est un mot binaire de même longueur que les individus : un "0" à la n^{e} position du masque implique qu'il n'y ait aucun changement à la n^{e} position des deux génomes ; un "1" à la n^{e} position du masque implique un échange des n^{es} positions des deux génomes, comme on le voit sur la figure 2.12 suivante

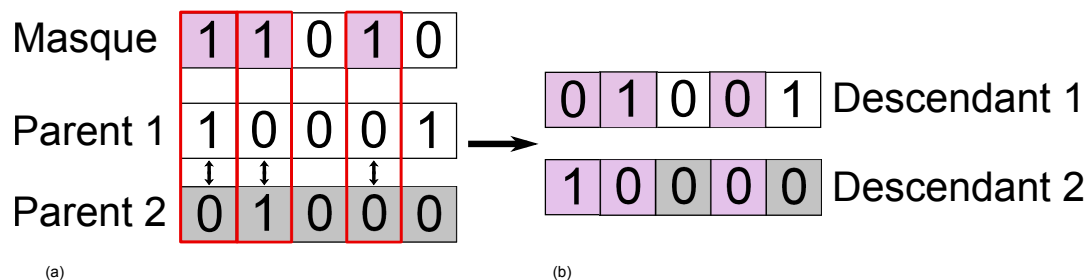


FIGURE 2.12 – Croisement uniforme. (a) Choix des symboles à échanger en fonction du masque. (b) Résultat du croisement uniforme.

La mutation

L'opérateur de mutation sur des individus binaires opère simplement en modifiant, avec une probabilité assez faible généralement comprise entre 0.01 et 0.1 par individu, des bits choisis au hasard dans les individus. Lorsque la mutation est opérée, les bits choisis changent simplement de valeur, les "1" deviennent des "0" et les "0" deviennent des "1".

2.2.2.2.4/ Quelques applications et améliorations

Méthodes hybrides

De nombreuses méthodes hybrides ont été développées afin de tirer profit notamment des opérateurs de croisement et de mutation qui ont une très nette tendance à accélérer la convergence des algorithmes d'optimisation. Lovbjerg et al. [103] ont par exemple amélioré l'algorithme PSO en ajoutant aux formules basiques (cf. chapitre 2.2.3) une notion de "mariage" entre les particules qui permet de créer des sous-populations, et ainsi de se rapprocher des méthodes qui ont précédemment été développées dans l'article de Romdhane et al. [233]. Hsieh et al. [171] ont développé un nouvel opérateur basé sur les observations précédentes concernant PSO et GA. Cet opérateur de "mutation guidée" permet de choisir de manière judicieuse, en fonction du déplacement des particules avec PSO, les particules sélectionnées pour muter.

De la même manière, Krink et al. ont développé un algorithme hybride adaptatif qui tire lui aussi parti des intérêts apportés par les algorithmes évolutionnaires [114]. Cet algorithme, appelé "Life Cycle", s'inspire des différentes étapes qui jalonnent la vie de tout être humain, à savoir l'enfance, l'adolescence, et enfin la phase adulte. Chaque individu n'a pas les mêmes capacités suivant la phase de vie dans laquelle il se situe, mais toujours est-il qu'il reste généralement adapté à son environnement tout au long de sa vie. Chacun des individus peut choisir librement d'être soit une particule PSO, soit un individu génétique, soit un individu Hill Climbing. Ce choix vient du fait que ces trois algorithmes

sont performants, mais pas dans les mêmes situations. Alors que l'un aurait tendance à être plus performant en phase d'exploration du domaine, l'autre pourrait être plus performant en phase d'exploitation locale.

Ainsi, dès lors qu'une particule n'est manifestement plus adaptée à son environnement puisque la valeur de la fonction objectif n'a pas évolué depuis un certain nombre d'itérations (50 dans [114]), la particule change de type, comme on le voit dans l'algorithme 4.

Algorithm 4 L'algorithme "life cycle"

```

Initialiser la population
while Le critère d'arrêt n'est pas vérifié do
  for tous les individus de la population do
    Evaluer la fonction objectif (fitness)
    if la valeur de la fonction objectif n'a pas changé depuis  $n$  itérations then
      Changer le type de l'individu :
      if L'individu est une particule PSO then
        L'individu devient un individu génétique
      else if L'individu est un individu génétique then
        L'individu devient un individu Hill Climbing
      else
        L'individu devient une particule PSO
      end if
    end if
    for Toutes les particules PSO do
      Calculer la nouvelle vitesse de la particule
      Déplacer les particules dans l'espace des solutions
    end for
    for Les individus génétiques do
      Sélectionner des individus dans la population
      Effectuer les opérateurs de croisement
      Effectuer les opérateurs de mutation
    end for
    for Les individus Hill climbing do
      Trouver des nouvelles solutions de voisinage
      Evaluer la fonction objectif de la nouvelle solution
      Accepter la nouvelle solution avec la probabilité  $p$ 
    end for
  end for
end while
  
```

Les particules changent de type dans l'ordre suivant : particule PSO – individu génétique – individu Hill Climbing – particule PSO – ...

Les résultats numériques obtenus dans cet article sont véritablement passionnants, puisqu'assez différents des prédictions effectuées par les auteurs au début de l'article. La figure 2.13 présente l'évolution de la population au cours des itérations. Prédissant que les individus génétiques seraient plus efficaces dans la phase d'exploration du domaine, les auteurs ont initialisé la population avec des individus génétiques uniquement. Ce-

pendant, on remarque que la tendance s'inverse très rapidement, et que la quasi-totalité des individus deviennent des individus Hill Climbing. Ils sont alors très performants dans cette phase d'exploration du domaine. Plus tard, lors de la phase d'exploitation locale, ils ne sont plus très efficaces, et changent alors petits à petits de type. Finalement, les individus sont partagés entre des individus génétiques et des particules PSO lors de cette dernière phase de calcul. Cependant, bien que ces résultats apportent une étude considérable - bien qu'empirique - concernant l'efficacité des différentes méthodes d'optimisation testées suivant la phase de recherche dans laquelle l'algorithme se situe, les résultats concernant la convergence ne sont, selon moi, pas probants. En effet, bien que l'algorithme LifeCycle converge systématiquement vers le minimum global de toutes les fonctions testées, il le fait de manière générale de manière plus lente que les autres méthodes. En effet, de manière générale, et sur une grande majorité des fonctions testées, l'algorithme PSO utilisé seul est bien plus rapide que la méthode LifeCycle.

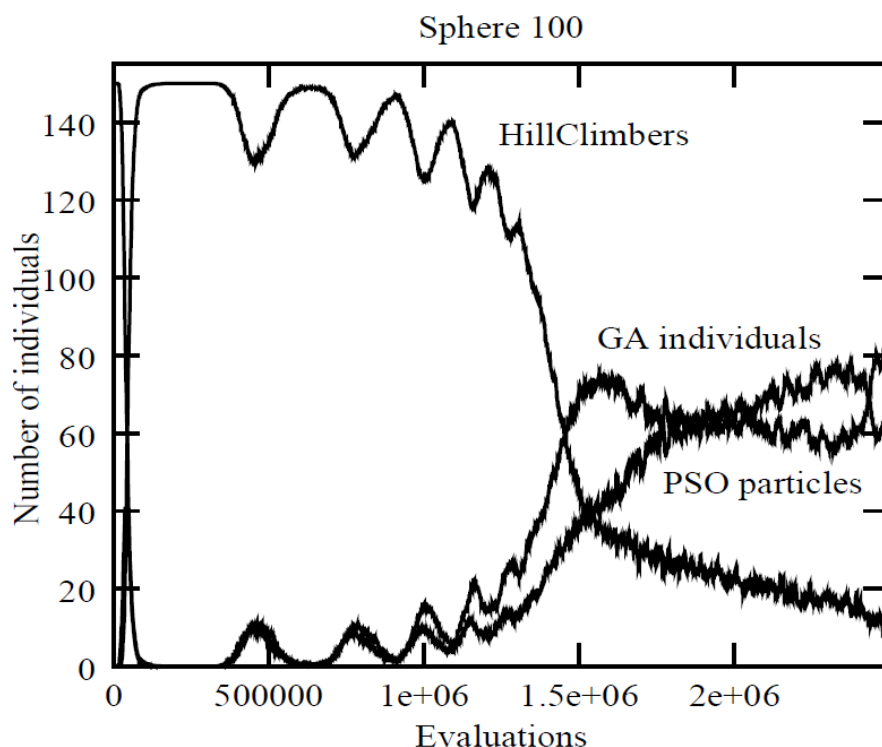


FIGURE 2.13 – Evolution de la population dans l'algorithme LifeCycle

En outre, dans le travail de Shi et al. [128], les auteurs proposent deux algorithmes hybrides combinant les algorithmes PSO et GAs de deux manières différentes. La première propose d'interfacer ces deux algorithmes de manière parallèle, comme on le voit dans l'algorithme 5.

Ainsi, les individus des deux populations peuvent être sélectionnés pour être des particules PSO ou des individus GAs en fonction de leur adaptation à l'environnement. Cette parallélisation des algorithmes permet une diminution significative du temps de calcul et est en outre très efficace à converger vers le minimum global de la fonction objectif considérée.

La deuxième méthode consiste à intégrer les algorithmes PSO et GAs en série, dans

Algorithm 5 L'algorithme PSO-GA parallèle

```

Initialiser deux sous-populations GA et PSO
while Le critère d'arrêt n'est pas vérifié do
    Exécuter les algorithmes PSO et GA simultanément sur chacune de leurs popula-
    tions respectives
    Mémoriser la meilleure solution de chacune des sous-populations
    if L'un des meilleurs individus de chaque population satisfait le critère d'arrêt then
        Stop
    end if
    if Le nombre de générations créées peut être divisible par le numéro de l'itération en
    cours then
        Sélectionner  $P$  individus dans chacun des sous systèmes
    end if
end while

```

un même algorithme. Tout d'abord, M populations différentes sont traitées avec PSO simultanément. Après un nombre M_1 d'itérations, les meilleures particules de chaque population sont récupérées pour coder des individus GAs afin de créer une population de N individus qui sera traitée avec un algorithme évolutionnaire. Après M_2 itérations, les meilleures solutions obtenues avec GAs sont réinjectées dans les populations PSO, comme on le voit dans l'algorithme 6 suivant.

Dans le domaine de l'optimisation multi-objectif

De nombreux auteurs [34, 44, 49] ont utilisé des méthodes d'agrégation pondérée que l'on a d'ores et déjà étudiées dans la partie 2.1.2.3.1 pour résoudre des problèmes d'optimisation multiobjectifs.

Des approches multi-objectif non-Pareto ont aussi été proposées pour pallier à tous les problèmes rencontrés basiquement dans le domaine de l'optimisation multi-objectif présentée dans la partie 2.1.2. Schaffer [26] est certainement le premier à présenter des méthodes d'optimisation multi-objectif qui cherchent les solutions non dominées basées sur les algorithmes génétiques. Dans cette approche, appelée Vector Evaluated Genetic Algorithm (VEGA), l'algorithme évalue un **vecteur objectif** au lieu d'une valeur scalaire (*fitness*), chaque élément de ce vecteur représentant l'une des fonctions objectif considérées [99]. Si l'on considère M fonctions objectif à optimiser simultanément, Schaffer propose de diviser, à chaque itération, la population en M sous-populations de manière aléatoire. A chaque sous-population est assignée une fonction objectif, et donc une valeur scalaire correspondant à la *fitness* de la fonction objectif considérée. Ainsi, chacune des fonctions objectif est évaluée par une sous-population. Afin que les individus ne soient pas partitionnés en fonction de leur position dans l'espace des solutions, ils sont "mélangés" avant que les sous-populations ne soient créées. Ainsi, il y a, dans chaque sous population, des individus dispersés dans l'ensemble du domaine des solutions. Suite à l'évaluation de la fonction objectif assignée à chaque sous-population, les opérateurs de sélection sont appliqués, puis la population est à nouveau fusionnée avant que les opérateurs de croisement et de mutation ne soient appliqués à l'ensemble de la nouvelle population sélectionnée précédemment. La nouvelle population ainsi créée ne contient donc que des individus performants concernant une fonction objectif en particulier. Du fait que les sous-populations sont créées de manière aléatoire, et que les objectifs sont

Algorithm 6 L'algorithme PSO-GA en série

```

Initialiser  $N$  populations de particules PSO
Gap=0
while Gap différent de GapMAX do
  GapPSO=0
  while GapPSO différent de  $(M_1 + 1)$  do
    for Toutes les populations do
      for Tous les individus do
        Appliquer les opérateurs PSO
      end for
    end for
    Récupérer la meilleure solution
    if Le critère d'arrêt est rempli then
      Stop
    end if
    GapPSO=GapPSO+1
  end while
  for Toutes les populations do
    Sélectionner la meilleure particule
    Transformer cette particule en individu GA
  end for
  while GapGA différent de  $M_2 + 1$  do
    GapGA=0
    Appliquer les opérateurs GA
    Récupérer la meilleure solution
    if Le critère d'arrêt est rempli then
      Stop
    end if
    GapGA=GapGA+1
  end while
  if Gap différent de GapMAX then
    Récupérer la meilleure solution
    Stop
  else
    Gap=Gap+1
    Transmettre les meilleurs individus GA aux populations PSO
  end if
end while

```

assignés à chaque sous-population de manière aléatoire, les individus sont testés sur des objectifs différents à chaque itération. Finalement, on obtient des individus performants sur toutes les fonctions objectif considérées.

Cet algorithme a été remanié plusieurs fois, afin de pallier au fait que les individus ne sont testés que sur une seule fonction objectif à la fois. Des méthodes permettant de classer les particules en fonction de leur adaptabilité aux fonctions objectif considérées ont été développées, appelées méthodes de classement lexicographique [20].

Finalement, de très nombreux algorithmes basés sur les méthodes évolutionnaires ont été développés pour résoudre les problèmes d'optimisation multi-objectif, et font même l'objet d'ouvrages entiers [99, 154].

2.2.3/ FOCUS ON : PARTICLE SWARM OPTIMIZATION

2.2.3.1/ HISTORIQUE, PRINCIPE DE BASE

Inspirée par l'observation des groupes d'oiseaux et des bancs de poissons réalisée par Craig Reynolds à la fin des années 80 [29], la méthode d'optimisation par essaims particulaires (appelée Particle Swarm Optimization (PSO) dans la littérature anglophone) est ainsi issue d'une analogie avec les comportements collectifs et sociétaux des groupes d'animaux. En effet, chez certains groupes d'animaux comme les bancs de poissons, on peut observer des dynamiques globales relativement complexes, alors que chacun des individus, dont les capacités cognitives sont relativement limitées, n'a accès qu'à un nombre restreint d'informations **locales**. Par exemple, comme souligné par Dreco et al. [121], on observe qu'un banc de poissons est capable d'éviter un prédateur - d'abord en divisant le groupe en deux, puis en reformant le banc originel - tout en maintenant la cohésion du banc, et ce de manière très rapide. Ces dynamiques globales s'inscrivent parfaitement dans la définition de l'auto-organisation, décrite dans la partie 2.2.2.1.

Basé sur ces informations, l'algorithme PSO a tout d'abord été développé par Kennedy et al. en 1995 [60], et s'est très rapidement situé comme l'une des meilleures méthodes métaheuristiques dans le domaine de l'optimisation difficile. Fait remarquable, cette méthode a tout d'abord été développée pour être appliquée à des domaines continus, contrairement aux autres méthodes (ACO, méthodes évolutionnaires) qui ont été développées pour résoudre des problèmes discrets et combinatoires.

Le principe de cet algorithme est le suivant :

On définit des particules qui correspondent chacune à une **potentielle solution** au problème d'optimisation posé, c'est à dire à une position dans l'espace des solutions, à laquelle est associée une valeur de la fonction objectif pour cette position en particulier. Chaque particule n'a accès qu'à un nombre restreint d'informations locales, c'est à dire des renseignements concernant les particules de son voisinage (voir partie 2.2.3.2), et est dotée de plusieurs renseignements qui la définissent dans l'espace des solutions :

- Sa position dans l'espace des solutions,
- Sa vitesse,
- La valeur de la fonction objectif pour sa position actuelle.

Ainsi, à la manière des bancs de poissons ou des groupes d'oiseaux, les particules vont avoir tendance à se suivre les unes les autres, c'est à dire à agir comme un groupe, pour se déplacer ensemble dans l'espace des solutions et converger ensemble vers l'optimum global de la fonction objectif considérée. Plus précisément, à chaque itération PSO, chaque particule recalcule sa vitesse et sa nouvelle position dans l'espace des solutions, en fonction à la fois de ses données personnelles, mais aussi des données qui lui sont transmises par son voisinage. Ainsi, chaque particule peut "choisir" entre les différentes possibilités suivantes, comme on le voit dans la figure 2.14 :

- Suivre sa meilleure performance personnelle,
- Suivre la meilleure performance de son voisinage,

- Suivre sa vitesse actuelle (comportement qualifié d'"égoïste" dans la littérature)

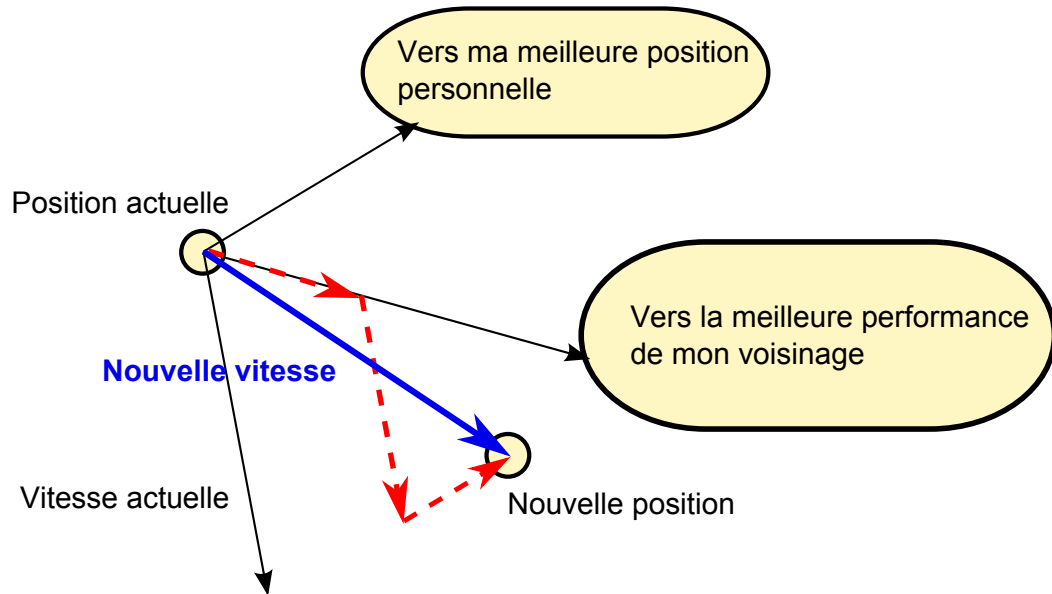


FIGURE 2.14 – Mode de déplacement des particules dans l'espace des solutions

En réalité, chaque particule n'effectue par réellement ce choix, puisqu'elle combine ces trois possibilités grâce à une combinaison linéaire (représentée en rouge sur la figure 2.14 par une somme vectorielle) de ces trois potentielles directions.

Ces remarques sont cohérentes avec les études socio-cognitives à la base de cet algorithme. En effet, il est établi que le comportement de chaque individu est influencé à la fois par son passé, mais aussi par le comportement de ses voisins (c'est à dire par les individus qui lui sont proches en terme de réseau social, sans être nécessairement proches dans l'espace).

Ainsi, la position des particules dans l'espace des solutions à un instant t est fonction de la direction actuelle de leur mouvement personnel, de leur vitesse actuelle, de leur meilleure performance personnelle et de la meilleure position des particules qui composent le voisinage de cette particule, comme on le voit dans l'équation suivante

$$\mathbf{X}_i^k = f(\mathbf{X}_i^{k-1}, \mathbf{v}_i^{k-1}, \mathbf{P}_{i,best}^k, \mathbf{G}_{i,best}^k) \quad (2.22)$$

Les variables associées à chaque particule sont donc les suivantes :

- $\mathbf{X}_i^k$: La position de la particule n^oi à l'itération k .
- $\mathbf{V}_i^k$: La vitesse de la particule n^oi à l'itération k .
- $\mathbf{G}_{i,best}^k$: La meilleure position atteinte par les particules du voisinage de la particule i , à l'itération k .
- $\mathbf{P}_{i,best}^k$: La meilleure position personnelle atteinte par la particule n^oi à l'itération k .

Les paramètres (constants dans les premières versions de PSO, puis évolutifs dans des versions plus récentes, comme nous le verrons dans la partie 2.2.3.3) qui définissent le comportement de l'algorithme sont les suivants :

- V_{min} : la vitesse minimale que peut atteindre une composante d'une particule.
- V_{max} : La vitesse maximale que peut atteindre une composante d'une particule.

- Le nombre de particules.
- Les paramètres c_1 et c_2 qui définissent respectivement les capacités d'exploration et d'exploitation de l'essaim, comme définis dans l'équation (2.23).

Ainsi, le changement de position s'effectue donc comme suit

$$\begin{cases} V_i^k = V_i^{k-1} + c_1(P_{i,best}^k - X_i^{k-1}) + c_2(G_{i,best}^k - X_i^{k-1}) \\ X_i^k = X_i^{k-1} + V_i^k \end{cases} \quad (2.23)$$

Les constantes c_1 et c_2 sont des paramètres de confiance. Ils représentent la confiance de la particule en son propre chemin et en l'essaim. Ces constantes sont des termes d'accélération stochastique qui déplacent chaque particule vers sa meilleure position personnelle, ou vers la meilleure position atteinte par les particules de son voisinage.

Les paramètres V_{max} et V_{min} permettent d'éviter que l'algorithme n'oscille trop, en limitant les vitesses limites des particules. En effet, des extrema trop grands de la vitesse auraient pour conséquence de déplacer les particules trop rapidement dans l'espace des solutions, et donc potentiellement de rater des zones optimales.

Le pseudo-code de la version la plus générale de PSO est donné dans l'algorithme 7.

Algorithm 7 L'algorithme PSO en variables continues

```

n = nombre de particules
D = Dimension du problème
while Le critère d'arrêt n'est pas vérifié do
  for i=1,n do
    if  $F(X_i) > F(P_i)$  then
      for d=1,D do
         $p_{id} = k_{id} : p_{id}$  est donc le meilleur individu trouvé
      end for
    end if
    g = i
    for Tous les voisins de la particule i do
      if  $F(P_j) > F(P_g)$  then
        g = j : g est donc le meilleur individu du voisinage
      end if
    end for
    for d=1,D do
      Mise à jour des vitesses et des positions avec les équations (2.23)
    end for
  end for
end while
  
```

Plus tard, une autre version PSO s'est très rapidement imposée comme la version la plus efficace, et a été utilisée comme une base pour toutes les modifications suivantes. En effet, dans l'article de Shi et al. [81], les auteurs introduisent la notion d'inertie dans leur algorithme. En effet, il est noté que les particules, tout en combinant leur vitesse actuelle avec d'autres informations auxquelles ils ont accès, devraient avoir une certaine inertie. C'est à dire que leur précédente vitesse devrait être pondérée dans la formule de mise à jour de la vitesse (2.23). Ainsi, les auteurs proposent les nouvelles équations (2.24) suivantes

$$\begin{cases} \mathbf{V}_i^k = \omega \times \mathbf{V}_i^{k-1} + c_1 \times rand_1 \times (\mathbf{P}_{i,best}^k - \mathbf{X}_i^{k-1}) + c_2 \times rand_2 \times (\mathbf{G}_{i,best}^k - \mathbf{X}_i^{k-1}) \\ \mathbf{X}_i^k = \mathbf{X}_i^{k-1} + \mathbf{V}_i^k \end{cases} \quad (2.24)$$

Comme on le voit dans les équations précédentes, il est aussi noté dans cet article que les paramètres c_1 et c_2 doivent être choisis judicieusement, puis multipliés par des nombres aléatoires $rand_1$ et $rand_2$ distribués selon une loi normale, afin que l'algorithme soit plus performant. Dans les premiers essais numériques effectués, le paramètre d'inertie est fixé et est compris dans l'intervalle $[0.9, 1.2]$. Finalement, afin d'améliorer les performances de l'algorithme, les auteurs établissent que le paramètre d'inertie ω doit décroître au cours du temps, et il s'avère justement que l'algorithme est beaucoup performant dans ce cas. En effet, plus le paramètre d'inertie est élevé, plus les particules auront tendance à se déplacer loin de leur position actuelle, et donc à aller découvrir de nouvelles contrées. Dans ce cas, la [capacité d'exploration](#) de l'essaim est favorisée. De la même manière, plus l'inertie est petite, plus les particules se déplacent de manière locale. Dans ce cas, la [capacité d'exploitation de zones prometteuses locales](#) de l'essaim est favorisée. Effectivement, en diminuant l'inertie au cours des itérations, on considère que l'on souhaite [explorer le domaine](#) au début du calcul, puis, en fin de calcul, [exploiter les zones prometteuses](#) en cherchant de manière plus précise le minimum d'une zone du domaine en particulier. La convergence de l'algorithme est ainsi contrôlée. Comme on le verra dans la partie 2.2.3.3, bien que cette première proposition consistant à diminuer le paramètre d'inertie au cours du temps donne des résultats intéressants, elle fut aussi et surtout l'initiatrice de nombreuses autres modifications de l'algorithme. De nombreuses recherches ont ainsi proposé de modifier les paramètres de PSO au cours du temps, de différentes manières, afin d'améliorer ses performances.

2.2.3.2/ TOPOLOGIES DE LA POPULATION

Une dimension extrêmement intéressante de PSO est la [topologie](#) de la population, c'est à dire la manière dont sont choisies les particules qui définissent le voisinage de chacune d'entre elles, et donc en quelle mesure l'information circule dans l'essaim. En effet, comme il l'a été souligné et détaillé dans le travail de Kennedy et al. [113], plus la connectivité entre les particules est élevée, plus la convergence est rapide, puisque l'information circule d'autant plus rapidement dans l'essaim qu'il y a de connections. Cependant, cette caractéristique ne favorise pas l'exploration de nouvelles contrées du domaine. De plus, lorsque la distance entre les particules (définie comme le nombre de liens entre deux particules i et j , et non pas comme la distance géographique entre ces particules) est trop petite, et que donc la communication entre les particules est vraiment très rapide, la population a tendance à se déplacer très rapidement dans le domaine des solutions, et donc à rater des solutions pourtant pressenties lors de la découverte de zones prometteuses. Au contraire, une connectivité trop faible aurait pour effet d'inhiber la gestion de zones prometteuses par l'essaim, en ne dirigeant pas les particules dans ces zones suffisamment rapidement. Finalement, cet article souligne surtout l'[influence](#) de la topologie des essaims sur la convergence de l'algorithme PSO.

Les topologies peuvent être classées en deux parties distinctes, les topologies dites [géographiques](#) et les topologies dites [sociales](#). Dans le type de voisinage géographique, chaque particule est influencée par les particules qui lui sont le plus proches dans l'espace des solutions à un instant donné. Cette méthode nécessite donc de recalculer le

voisinage de chaque particule à chaque itération en fonction de sa distance avec les autres, ce qui peut être très long en terme de temps de calcul. Les méthodes de topologies sociales consistent à prédéterminer le voisinage de chaque particule, et à ne plus le modifier par la suite. C'est le voisinage le plus simple à implémenter, et il ne fait pas perdre de temps de calcul. Enfin, quel que soit le type de topologie choisi, au cours des itérations, un voisinage social tend à devenir un voisinage géographique, puisque les particules se déplacent et se suivent dans l'espace des solutions. Dans la grande majorité des méthodes basées sur PSO, ce sont des voisinages sociaux qui sont utilisés en priorité.

Ainsi, deux sortes de topologies sociales différentes ont été développées depuis les années 90 : tout d'abord les topologies statiques, puis les topologies dynamiques qui évoluent au cours du temps (mais pas nécessairement en fonction de la position géographique des particules dans l'espace des solutions, comme pour les topologies dites géographiques), et qui vont nous intéresser plus particulièrement dans le cadre de cette étude.

2.2.3.2.1/ Principes importants Dans son article concernant le développement de nouvelles topologies [58], Eberhart rappelle les cinq principes de l'intelligence des essaims développés par Millonas dans son article traitant de l'intelligence collective dans les essaims [46] qui sont les suivants :

- Le principe de proximité : la population devrait être capable de traiter des problèmes simples et des calculs temporels en se déplaçant au cours du temps dans l'espace des solutions.
- Le principe de qualité : la population devrait être en mesure de répondre à certains critères de qualité définis par l'environnement (c'est à dire le domaine) dans lequel elles évoluent.
- Le principe des réponses diverses : la population ne devrait pas avoir à se déplacer dans des domaines trop étroits du domaine, et rester plus ou moins groupée dans le domaine au cours des itérations.
- Le principe de stabilité : la population ne devrait pas changer son mode de fonctionnement à chaque fois que l'environnement change.
- Le principe d'adaptabilité : la population devrait être en mesure de changer son mode de fonctionnement dès que cela peut être bénéfique pour minimiser le temps de calcul.

Ces cinq principes définissent les "règles de vie" des populations, c'est à dire à la fois la manière dont elles doivent se comporter dans l'espace sur lequel elles sont définies, mais aussi les qualités requises afin qu'elles soient efficaces pour résoudre les problèmes d'optimisation.

2.2.3.2.2/ Topologies statiques Les deux topologies statiques les plus couramment utilisées dans la littérature spécialisée ont été développées par Eberhart et al. [58]. La première, appelée la topologie LBEST peut être décrite à la manière de la figure 2.15. Comme on le voit, chaque particule est liée avec deux autres particules de l'essaim. Ces liaisons n'évoluent pas au cours des itérations, et ce quelles que soient les positions des particules dans l'espace des solutions. Elles sont choisies arbitrairement au début du calcul et restent les mêmes au cours des itérations. Comme il l'est dit dans l'article [113], cette topologie peut ne pas être efficace si certains paramètres ne sont pas bien choisis. En effet, la faible connectivité des particules implique que les informations circulent très

lentement au sein de l'essaim (plus il y a de particules, ce qui est censé faire accélérer la convergence, plus la communication est lente !), et que donc la convergence reste limitée, comme il l'a été expliqué précédemment.

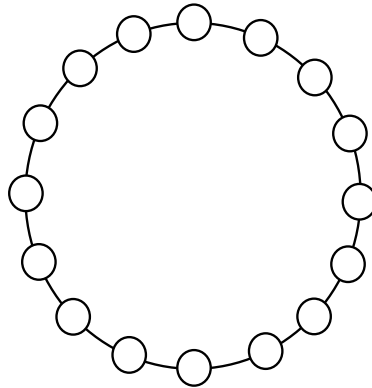


FIGURE 2.15 – Représentation géométrique de la topologie LBEST

La deuxième topologie employée très régulièrement est la topologie GBEST qui peut être décrite à la manière de la figure 2.16. La méthode PSO utilisant ce type de topologie est aussi appelée Fully Informed Particle Swarm (FIPS) dans [136]. Comme on le voit sur la figure 2.16, dans ce cas toutes les particules sont liées entre elles. Cette topologie peut être très efficace, car les informations circulent de manière extrêmement rapide dans l'essaim (toutes les particules ont accès à toutes les informations de l'essaim à chaque instant). Cependant, comme il l'a été dit précédemment, une connectivité trop élevée peut bloquer les capacités d'exploration de l'essaim en rapprochant toutes les particules de la meilleure à un instant donné (quand bien même ce serait dans un minimum local). De plus, il est de la définition de l'algorithme PSO, et de toute l'analogie qui est à son origine, que chacune des particules n'ait qu'une vision locale et se déplace dans l'espace des solutions en suivant plus ou moins certaines de ses congénères. C'est certainement cette propriété qui fait l'efficacité de la méthode, mais aussi la raison pour laquelle les groupes d'animaux ont choisi ce mode de fonctionnement (peut-on remettre en cause une organisation sociale qui a fait ses preuves depuis des millions d'années ?).

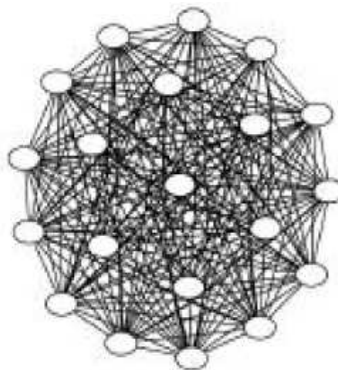


FIGURE 2.16 – Représentation géométrique de la topologie GBEST

Par la suite, dans l'article de Mendes et al. [127], deux nouvelles topologies sont utilisées et appliquées sur des fonctions d'optimisation test classiques. La première est la topologie "four clusters" que l'on voit sur la figure 2.17. Cette topologie peut s'avérer très

efficace car elle considère des sous-populations qui évoluent presque séparément dans l'espace des solutions. En effet, la connectivité est beaucoup plus élevée **dans** chaque cluster qu'**entre** les clusters. Ainsi, alors que la communication entre les particules est élevée dans chaque cluster et que donc, elles se suivent très rapidement dans l'espace des solutions, les informations circulent assez lentement entre chaque cluster, et les clusters ne se suivent donc que très lentement dans l'espace des solutions. Dans le même article est proposée une topologie pyramidale, comme on le voit sur la figure 2.18 (vue de dessus). Cette topologie peut être très efficace, puisque les informations circulent rapidement en son sein. Ceci dit, la particule située en haut de la pyramide a donc plus d'influence sur toutes les autres, puisqu'elle est située en amont de tous les liens de voisinage. Comme on le verra par la suite dans la partie 2.2.3.2.3, des topologies s'inspirant d'un système pyramidal ont été développées par Janson et al. dans [122, 133] dans lesquelles les particules sont organisées de manière hiérarchique pyramidale. Cependant, puisque ces topologies sont adaptatives, la pyramide est réorientée en fonction des résultats, afin que les particules les plus influentes de la pyramide soient celles qui ont les *fitness* les plus adaptées au problème.

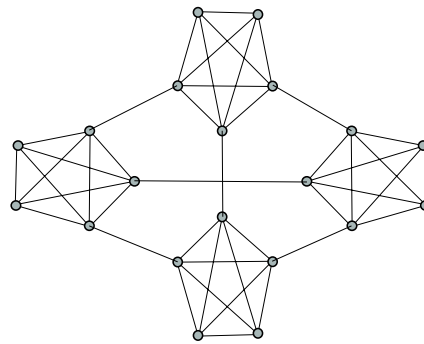


FIGURE 2.17 – Représentation géométrique de la topologie four clusters

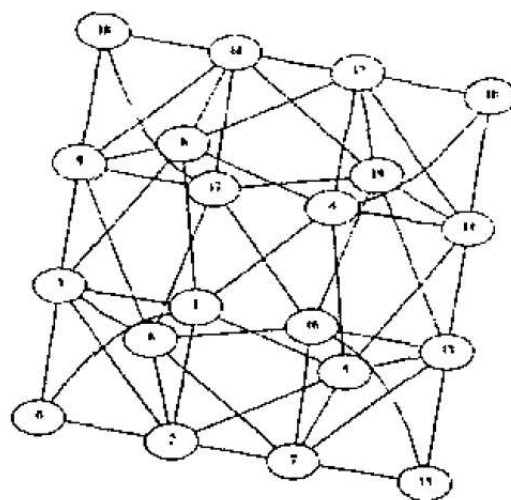


FIGURE 2.18 – Représentation géométrique de la topologie pyramidale

D'autre part, la topologie grégaire développée par Pasupuleti et al. [161] nous a semblée pertinente dans le cadre de cette étude. Dans ce mode de voisinage, les particules ne

sont attirées que par la meilleure particule de l'essaim uniquement. Cependant, dès lors qu'une particule se rapproche (géographiquement dans l'espace des solutions) trop de la meilleure particule de l'essaim (c'est à dire dès que la distance Euclidienne entre les deux particules devient plus petite qu'une constante ϵ), sa vitesse est réinitialisée, afin de favoriser l'exploration du domaine. La topologie peut donc être décrite à la manière de la figure 2.19 suivante. La réinitialisation de la vitesse est faite de manière aléatoire, en fonction d'un facteur γ qui détermine l'ordre de grandeur des nouvelles vitesses. Ainsi, une grande valeur de γ impliquera une grande vitesse de déplacement des particules, et donc une phase d'exploration du domaine, alors qu'une petite valeur de γ impliquera plutôt des petits déplacements, et donc une phase d'exploitation de zones prometteuses. Ainsi, dans cette méthode GPSO (Gregarious Particle Swarm Optimization), ce paramètre γ est réajusté au cours des itérations, en fonction de l'évolution de la meilleure particule trouvée de la manière suivante :

- Initialement, γ vaut 1.5.
- Si, au cours d'une itération, la meilleure valeur globale est améliorée, c'est à dire que l'algorithme a trouvé une position plus adaptée, alors γ diminue jusqu'à une valeur minimale de 2.
- Si, au cours d'une itération, la meilleure valeur globale n'est pas améliorée, alors γ augmente jusqu'à une valeur maximale de 4.

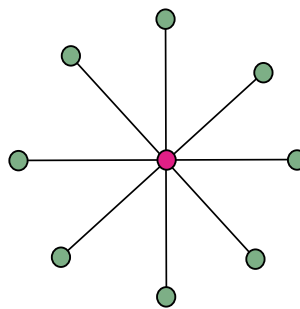


FIGURE 2.19 – Représentation géométrique de la topologie grégaire

Cette topologie, bien que pouvant présenter des contraintes du fait du manque de communication entre les particules, peut s'avérer intéressante si l'on souhaite trouver une réponse rapidement, sans qu'elle ne soit obligatoirement la meilleure possible.

Cette méthode pourrait être associée à l'analogie suivante : considérons un groupe d'oiseaux cherchant de la nourriture. La population se place à l'endroit où la nourriture est la plus abondante jusqu'à ce que cet endroit devienne pauvre en nourriture. Dès lors, les oiseaux changent d'endroit les uns après les autres, en suivant le premier oiseau qui aura trouvé une zone prometteuse. C'est la raison pour laquelle, dans cette méthode, les particules suivent uniquement la meilleure particule globale, mais aussi la raison pour laquelle les vitesses sont réinitialisées dès que les particules se rapprochent trop de la meilleure particule globale. Finalement, on peut dire que cet algorithme ne favorise pas les comportements sociaux, en ne développant pas de communication au sein de l'essaim. C'est la raison pour laquelle il est qualifié de "grégaire".

2.2.3.2.3/ Topologies dynamiques Par la suite, des topologies dynamiques et évolutives ont été développées afin que la population évolue en fonction des itérations pour s'adapter aux résultats courants et à l'évolution du domaine. En effet, comme nous

l'avons vu précédemment, il est intéressant pour un algorithme basé sur la méthode d'optimisation PSO de commencer par une phase d'exploration du domaine, pour essayer de trouver des zones prometteuses, dans lesquelles l'optimum local pourrait se trouver, puis de finir par une phase d'exploitation dans laquelle on exploite les zones prometteuses trouvées afin de converger vers l'optimum global. Ainsi, comme on l'a vu précédemment dans la partie 2.2.3.2.2, alors que la topologie LBEST ne présente pas beaucoup de liens entre les particules et est donc très adaptée à la phase d'exploration, la topologie GBEST, quant à elle, lie toutes les particules entre elles, et est donc très adaptée à la phase d'exploitation. Ainsi, Suganthan et al. ont proposé un algorithme PSO dans lequel la topologie évolue au cours des itérations [91]. En effet, le voisinage est défini par une topologie LBEST au début du calcul, puis, au cours des itérations, le voisinage de chaque particule est étendu, pour finalement terminer avec une topologie GBEST. Cette évolution est faite de manière linéaire, et est donnée en fonction du nombre maximal d'itérations à réaliser durant le calcul. Cette topologie est décrite comme avantageuse par rapport à la méthode PSO classique, mais les résultats ne sont pas décrits suffisamment clairement pour qu'une comparaison puisse être effectuée entre ce nouvel algorithme et la méthode PSO classique.

Dans l'article [243] que nous avons proposé, nous utilisons la topologie dynamique la plus simple possible. En effet, cette topologie est inspirée à la fois de la topologie LBEST [58] et de la topologie grégaire proposée dans [161]. D'une part les particules sont liées de manière circulaire comme on le voit dans la figure 2.15, et d'autre part, chaque particule est liée à la meilleure particule G_{best} trouvée depuis le début du calcul, comme on le voit dans la figure 2.19. cette topologie peut donc être qualifiée de "relativement" dynamique, dans le sens où la meilleure particule doit être mise à jour à chaque itération, si une meilleure particule est trouvée. La représentation dans l'espace de la topologie des particules n'est pas modifiée, c'est simplement la particule mise en exergue et liée à toutes les autres qui est modifiée à chaque itération.

Inspirée par la topologie pyramidale, une topologie hiérarchique a été développée par Janson et al. [122]. Cette topologie particulière permet de mettre en exergue la meilleure particule de l'essaim en la positionnant au sommet de la pyramide afin qu'elle ait le plus d'influence possible sur ses congénères. Ainsi, dès qu'une particule obtient de meilleurs résultats que la particule qui lui est directement supérieure dans la hiérarchie, elles échangent leurs places dans la pyramide, et la nouvelle meilleure particule a donc plus d'influence que sa congénère, comme on le voit dans la figure 2.20. Dans le même domaine, Ratnaweera et al. se sont inspirés de cette vision de la topologie hiérarchisée pour développer une hiérarchie intelligente, c'est à dire auto-organisée et dynamique au cours des itérations [137].

Plus tard, dans le but de vérifier si le concept même de topologie dynamique pouvait être adapté à l'optimisation par essais particuliers PSO, Mohais et al. ont proposé de réinitialiser les topologies de la population de manière aléatoire [140]. Dans cet article, on considère la topologie comme un graphe (et c'est cette assimilation qui a finalement amené au développement de topologies intelligentes). Ainsi, lors de l'initialisation, on génère des liens entre une particule u et une particule v , ce qui implique que la particule u fait partie du voisinage de la particule v (et donc qu'elle l'influence) mais pas l'inverse. Deux paramètres sont donc considérés pour déterminer des voisinages aléatoires orientés :

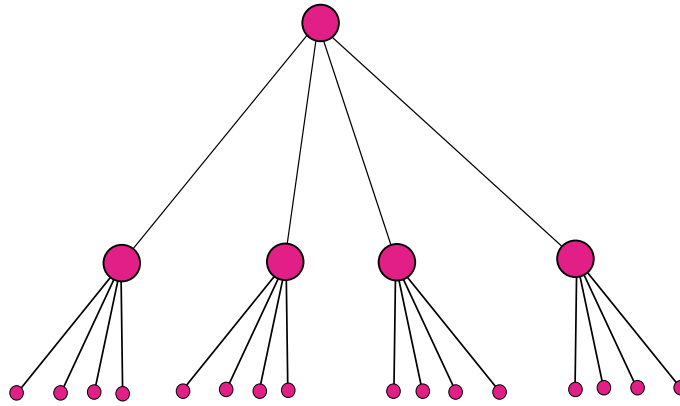


FIGURE 2.20 – Représentation géométrique de la topologie hiérarchique

- La taille n du voisinage de chaque particule.
- Le nombre m de liens sortant de chaque particule.

Ainsi, deux méthodes sont utilisées pour réinitialiser le voisinage des particules :

- *The random edge migration method* : L'algorithme sélectionne au hasard un nœud dont la taille du voisinage est supérieure à 1. Elle détache ensuite un de ses voisins et le rattache à un autre nœud dont le voisinage n'est pas d'ores et déjà complet, choisi de manière aléatoire. Cette méthode est appliquée avec une probabilité p_m .
- *The neighborhood restructuring method* : Cette approche réinitialise complètement la topologie de la population de manière aléatoire. Les paramètres de voisinage précédemment choisis, à savoir la taille n du voisinage de chaque particule, et le nombre m de liens sortant de chaque particule, sont conservés. Cette méthode est appliquée avec une probabilité p_r .

Finalement, les résultats présentés dans cet article montrent que ces deux méthodes sont assez efficaces, et que l'algorithme ainsi développé est performant. On peut donc en conclure qu'une topologie fixée et immuable n'est pas nécessaire pour obtenir une bonne convergence de l'algorithme.

C'est la raison pour laquelle nous avons décidé d'orienter les recherches en ce sens, afin de développer une topologie évolutive intelligente, qui optimiserait l'algorithme d'optimisation lui-même.

Jusqu'alors, les topologies dynamiques proposées étaient évolutives, mais pas nécessairement intelligentes. En effet, leurs évolutions, généralement effectuées de manière aléatoire - bien que très performante - n'étaient pas nécessairement inspirées de mécanismes naturels. Pourtant, les méthodes metaheuristiques sont très généralement inspirées du monde du vivant, et il serait cohérent que les populations utilisées en ce sens le soient aussi. Ainsi, Akbari et al. [203] ont proposé une nouvelle topologie dynamique dite "intelligente" inspirée d'observations sociologiques et psychologiques effectuées par Latané [22]. Ainsi, comme il l'a été vu précédemment, le partage de l'information parmi les particules de l'essaim (ou parmi les individus d'un groupe social) peut améliorer très fortement les capacités de convergence du groupe. Les particules peuvent ainsi utiliser les renseignements récupérés par leurs congénères, ou au moins par les congénères de leur voisinage. Comme dit par Kennedy concernant les études sociologiques de Latané : "Latané a proposé l'idée selon laquelle l'impact d'un groupe sur un individu est fonction des autres individus du groupe". C'est ainsi que Kennedy a développé que l'effi-

capacité d'un algorithme d'optimisation PSO vient de l'interaction des particules entre elles. Comme nous l'avons vu dans la partie 2.2.3.1, seule la meilleure particule de l'essaim est utilisée pour réactualiser la vitesse des particules à chaque itération. Les informations collectées par les particules ne sont donc pas utilisées pour mouvoir l'essaim dans l'espace des solutions ! Basé donc sur les observations de Latané, Akbari et. al proposent d'utiliser le principe de classement et de l'étendre à l'optimisation par essaim particulaire comme suit :

A chaque itération, après que la vitesse des particules ait été actualisée, les particules sont classées selon leur *fitness*, c'est à dire la valeur de la fonction objectif qu'elles obtiennent dans la position où elles sont actuellement. Ainsi, seules les γ meilleures particules sont utilisées pour actualiser la position des particules de l'essaim à l'itération suivante. Afin de gérer correctement la balance entre la phase d'exploration du domaine et la phase d'exploitation des zones prometteuses, la valeur de γ décroît linéairement au cours des itérations. Au début de l'optimisation, toutes les particules appartiennent au voisinage de chaque particule. On a donc affaire à la topologie GBEST qui permet de partager les informations dans l'intégralité de la population, et donc de déplacer l'essaim un peu partout dans le domaine des solutions. A mesure que γ décroît, on tend vers l'obtention d'une topologie grégaire, qui a tendance à favoriser l'exploitation des zones prometteuses, puisque les particules ne sont plus influencées que par la meilleure particule de l'essaim. La contribution de chaque particule est fonction de nombreux paramètres, définis par Latané dans ses recherches, qui définissent l'influence que cette particule devrait avoir dans le groupe. Finalement, la contribution d'un individu à l'actualisation de la vitesse d'une des particules qu'il influence est donnée en fonction de son paramètre de classement, de l'inverse de la distance avec cette particule influencée, et du nombre d'individus dans le voisinage considéré, comme nous le verrons dans les formules suivantes.

Ainsi, l'actualisation de la vitesse de la particule i à l'itération $(k + 1)$ est donnée par les équations (2.25).

$$\begin{cases} v_d^i(k+1) = v_d^i(k) + rand_1(p_d^i - x_d^i(k)) + rand_2(\sum_{j=1}^n (\psi_i^j(p_d^j(k) - x_d^i(k))) \\ \psi_i^j(k) = f(\tau_i^j(k), \delta_i^j(k), \xi_i) = \tau_i^j(k) \times \delta_i^j(k) \times \xi_i \end{cases} \quad (2.25)$$

où $\psi_i^j(k)$ modélise l'influence de la particule voisine j sur la particule influencée i pendant la k^{eme} itération. Ce paramètre d'influence est donné en fonction du paramètre de classement τ_i^j qui détermine la *fitness* relative de la particule influente j par rapport aux autres particules du voisinage, et qui est décrit dans l'équation (2.26). $\psi_i^j(k)$ est aussi donné en fonction de l'immédiateté de la particule j par rapport à la particule i . Cette immédiateté correspond à l'inverse de la distance Euclidienne entre les deux particules dans l'espace de dimension D dans lequel elles sont définies. Ainsi, à mesure que les particules s'éloignent, leur immédiateté diminue. Ce paramètre est défini dans l'équation (2.27). L'effet des individus d'un voisinage sur la particule i est modélisé par le paramètre ξ_i qui est défini par l'équation suivante

$$\tau_i^j = \frac{fitness_j(k)}{\sum_{l=0}^{Voisins_i} fitness - l(k)} \quad (2.26)$$

où :

- $fitness_j(k)$ est la fitness de la particule $n^o j$ dans le voisinage de la particule $n^o i$.

- $Voisins_i$ est le nombre de particules dans le voisinage de la particule i .

$$\delta_i^k(k) = \frac{1}{\sqrt{\sum_{d=1}^D (x_d^j(k) - x_d^i(k))^2}} \quad (2.27)$$

où $x_d^j(k)$ et $x_d^i(k)$ qui représentent respectivement les positions des particules j et de la particule candidate i définie en d dimensions dans l'espace des solutions.

$$\xi_i = \alpha N_i^\beta \quad (2.28)$$

avec :

- N_i le nombre d'individus dans le voisinage de la particule $n^o i$.
- $0 < \alpha < 1$ et $0 < \beta < 1$ deux paramètres constants qui gèrent l'influence de la population sur les particules considérées en tant qu'individu.

De plus, comme dit précédemment, seules les γ meilleures particules de l'essaim sont choisies pour influencer la mise à jour des vitesses des particules de l'essaim. Afin de favoriser d'abord l'exploration du domaine, puis l'exploitation des zones prometteuses, γ est adapté de manière dynamique au fur et à mesure des itérations comme spécifié dans l'équation suivante

$$\gamma = n - (n \times \frac{iter}{max_{iter}}) \quad (2.29)$$

Avec n le nombre de particules de l'essaim, $iter$ l'itération en cours, et $iter_{max}$ le nombre maximal d'itérations pouvant être atteint par l'algorithme.

Ainsi, cette variante de PSO propose que les particules ne soient toujours influencées que par un nombre prédéfini de particules, qui sont les meilleures de l'essaim. En en prenant un très grand nombre, les particules sont influencées par la totalité des particules de l'essaim, l'influence de chaque particule étant pondérée pour qu'elle soit proportionnelle au rang de la particule dans l'essaim et à sa *fitness* relative par rapport à l'essaim. La topologie proposée par Akbari et al. est donc dynamique et intelligente dans le sens où la topologie est modifiée en fonction des résultats obtenus par les particules de l'essaim.

Comme nous le verrons dans la partie 2.2.3.3.3, Akbari et al. proposent aussi une version adaptative de PSO, pas seulement lors de la mise à jour de la population, mais aussi en mettant à jour les paramètres de l'algorithme, et c'est justement ce qui rend cet article passionnant, et influent pour la suite de notre travail.

Finalement, cette partie a eu pour but de montrer qu'il est possible d'utiliser des topologies dynamiques pour améliorer les performances de PSO. En effet, la dynamique des particules s'est très rapidement tournée vers des méthodes intelligentes et adaptatives, comme nous le verrons dans la partie 2.2.3.3.3.

2.2.3.3/ PRINCIPALES VARIANTES ET AMÉLIORATIONS

Comme nous l'avons vu précédemment, depuis la première version de l'algorithme PSO proposée par Kennedy et al. en 1995, de nombreuses variantes et améliorations ont été

proposées dans la littérature. Cette partie a pour but de détailler quelques-unes de ces améliorations.

2.2.3.3.1/ PSO binaire Tout d'abord, l'algorithme PSO a été étendu à l'optimisation de problèmes binaires. Les premiers dans ce domaine sont Kennedy et al. qui proposent, en 1997 [73], une version de PSO permettant de faire face aux problèmes mathématiques binaires (Binary Particle Swarm Optimization BPSO). Cette version sera remodifiée à son tour afin d'être adaptée à des cas plus concrets d'optimisation des structures, comme nous le verrons par la suite.

Dans l'article proposé par Kennedy et al. [73], les formules de mise à jour de la vitesse et de l'accélération restent identiques aux formules (2.23), sauf que les vecteurs $\mathbf{P}_{i,best}^k$, $\mathbf{X}_i^k$ et $\mathbf{G}_{i,best}^k$ deviennent des vecteurs binaires composés de 0 et de 1. Ainsi, la vitesse de chaque particule peut être vue comme la probabilité de la particule de changer d'état, c'est à dire de passer de l'état 1 à l'état 0, ou de l'état 0 à l'état 1, pour chacune de ses composantes. Une fonction de transformation sigmoïde $S(V_i^k)$ est utilisée pour transformer la vitesse calculée en probabilité, pour finalement modifier l'état de la particule, comme on le voit dans l'algorithme 8. La fonction sigmoïde utilisée est définie à la manière décrite dans l'équation (2.30). De nombreuses fonctions définies sur des domaines binaires sont testées avec cette version de PSO. Afin de compléter ces résultats, une autre publication [79] compare cet algorithme avec différentes versions des algorithmes génétiques sur de nombreuses fonctions objectif différentes, dépendant notamment du nombre d'extrema locaux et de la dimension des particules (c'est à dire la longueur du vecteur binaire qui constitue une particule). Dans les deux publications, les résultats sont très concluants et montrent que la version binaire de PSO précédemment développée est très efficace et compétitive par rapport à la performance des algorithmes évolutionnaires.

$$S(v_i^k(t+1)) = \frac{1}{1 + e^{-v_i^k(t+1)}} \quad (2.30)$$

Algorithm 8 PSO binaire : Changement d'état des particules binaires

```

if rand() < S(Vik) then
    Xik = 1
else
    Xik = 0
end if

```

Shen a proposé une amélioration de BPSO, afin de modifier le vecteur vitesse de manière plus efficace que dans la version BPSO initiale. Ici, la vitesse V_i de chaque particule i est un scalaire aléatoire compris entre 0 et 1. Le changement de topologie binaire est alors géré par l'algorithme 9 suivant

où a est un nombre aléatoire uniformément distribué dans $[0, 1]$, dont la valeur initiale vaut 0.5, et qui décroît jusqu'à 0.33 à la fin du calcul d'optimisation. Cependant, cette méthode a toujours tendance à rester bloquée dans des optima locaux, en ne favorisant pas suffisamment la phase d'exploration du domaine.

En outre, cette version binaire de PSO ne tient pas compte de la position actuelle d'une particule pour en calculer la prochaine. Ainsi, une version binaire basée sur la représentation génotype/phénotype que l'on retrouve dans le domaine de la génétique est propo-

Algorithm 9 Amélioration de BPSO par Shen et al : Passage du continu au binaire

```

if  $0 < V_i \leq a$  then
   $X_i^{k+1} = X_i^k$ 
else
  if  $a < V_i \leq \frac{1+a}{2}$  then
     $X_i^{k+1} = P_{Best,i}^k$ 
  else
     $X_i^{k+1} = G_{Best}^k$ 
  end if
end if

```

sée par Lee et al. [184] permettant de gérer ce problème. Dans l'étude de la génétique, le génotype correspond au **codage** du phénotype, c'est à dire des caractères visibles ou observables sur l'individu en question. Ainsi, un gène, qui est une séquence de bases azotées ordonnancées dans un ordre précis, est le **codage**, c'est le caractère **génotypique**. Ce gène implique un caractère physique, par exemple la couleur des yeux : c'est le caractère **phénotypique**. Ainsi, basée sur ces observations, cette nouvelle représentation considère que chaque particule possède un caractère génotypique et un caractère phénotypique visible. Ainsi, pour que le caractère phénotypique puisse être modifié, le génotype doit tout d'abord être modifié. On définit ainsi les nouveaux paramètres suivants

- $X_{g,i}^k$: La position génotypique de la particule $n^o i$ à l'itération k .
- $X_{p,i}^k$: La position phénotypique de la particule $n^o i$ à l'itération k .

et on a les nouvelles équations (2.31) de mise à jour des vitesses et positions des particules suivantes

$$\begin{cases} V_i^{k+1} = w \times V_i^k + c_1 \times Rand_1 \times (P_{i,best}^k - X_{p,i}^k) + c_2 \times Rand_2 \times (G_{best,i}^k - X_{p,i}^k) \\ X_{g,i}^{k+1} = X_{g,i}^k + V_i^{k+1} \\ X_{p,i}^{k+1} = 0 \text{ si } rand() \geq S(X_{g,i}^{k+1}) \\ X_{p,i}^{k+1} = 1 \text{ si } rand() < S(X_{g,i}^{k+1}) \end{cases} \quad (2.31)$$

où S est une fonction Sigmoïde appliquée non pas à la vitesse des particules, mais à leurs positions génotypiques dans l'espace des solutions, comme donnée dans l'équation 2.32 suivante

$$S(X_{g,i}^{k+1}) = \frac{1}{1 + e^{-X_{g,i}^{k+1}}} \quad (2.32)$$

Ainsi, les nouvelles positions des particules dépendent aussi de leur position à l'itération d'avant. De plus, il est noté dans cet article que lors de l'utilisation de la première version de BPSO, la vitesse des particules a tendance à converger très rapidement vers les valeurs extrémales V_{min} et V_{max} , il est donc difficile de déplacer les particules dans l'espace des solutions, si les vitesses ne changent que très peu. Ainsi, pour gérer ce problème, il est proposé ici d'utiliser un opérateur de mutation qui changerait les vitesses de certaines particules avec une probabilité donnée. Ce processus de mutation éviterait aussi, à l'inverse, que les vitesses restent bloquées aux valeurs extrémales autorisées par l'algorithme, et ainsi que les particules se déplacent trop vite dans l'espace des solutions,

sans favoriser la phase d'exploitation des zones prometteuses. Cette opération de mutation est identique à celle que l'on observe dans les algorithmes génétiques et que l'on a précédemment développée dans la partie 2.2.2.2.3. La probabilité de mutation reste à choisir par l'utilisateur. Elle est souvent donnée de manière empirique dans la littérature et vaut généralement 0.01.

Enfin, Luh et al. [208] ont proposé d'utiliser une fonction linéaire permettant de transformer les positions génotypiques des particules en positions phénotypiques. Cette fonction linéaire est donnée par

$$L_x(X_{g,i}^0) = \frac{X_{g,i}^0 - X_{g,min}}{X_{g,max} - X_{g,min}} \quad (2.33)$$

Le phénotype est donc donné en fonction de L_x par

$$\begin{cases} X_{p,i}^0 = 1 & \text{si } rand() \leq L_x(X_{g,i}^0) \\ 0 & \text{dans les autres cas} \end{cases} \quad (2.34)$$

Ici, c'est donc bien la position génotypique de la particule qui définit son état phénotypique, et non pas sa vitesse. $rand()$ est un nombre aléatoire uniformément distribué entre 0 et 1.

Enfin, Luh et al. [208] ont aussi proposé de générer les valeurs phénotypiques à partir des valeurs des vitesses des particules, et non plus de leurs positions. Ici, la fonction de transformation des vitesses est donnée par

$$L_v(V_{g,i}^k) = \frac{V_{g,i}^k - V_{g,min}}{V_{g,max} - V_{g,min}} \quad (2.35)$$

La formule permettant de déterminer les bits binaires de la topologie à partir des équations précédentes est donnée par l'algorithme 10.

Algorithme 10 Formules permettant le passage des valeurs génotypiques aux valeurs phénotypiques

```

if  $0 < L_v(V_{g,i}^k) \leq \frac{a}{5}$  then
     $X_{p,i}^{k+1} = X_{p,i}^k$ 
else
    if  $\frac{a}{5} < L_v(V_{g,i}^k) \leq a$  et  $rand() \leq L_v(V_{g,i}^k)$  then
         $X_{p,i}^{k+1} = 1$ 
         $X_{p,i}^{k+1} = 1$ 
    else
        if  $\frac{a}{5} < L_v(V_{g,i}^k) \leq a$  et  $rand() > L_v(V_{g,i}^k)$  then
             $X_{p,i}^{k+1} = 0$ 
        else
            if  $a < L_v(V_{g,i}^k) \leq \frac{1+a}{2}$  then
                 $X_{p,i}^{k+1} = P_{p,Best,i}^k$ 
            else
                 $X_{p,i}^{k+1} = G_{pBest}^K$ 
            end if
        end if
    end if
end if
end if

```

L'utilisation de versions binaires de PSO implique aussi la gestion d'éventuels problèmes de topologie³. Ainsi, des problèmes de damiers peuvent apparaître, rendant la structure proposée par l'algorithme insolvable par le logiciel de calcul par éléments finis et/ou impossible à fabriquer. Pour remédier à cela, deux solutions sont proposées dans la littérature. Soit les éléments créant des damiers dans la topologie sont supprimés, soit d'autres éléments sont ajoutés afin de pallier à cela. Un tel algorithme de gestion des damiers dans le maillage sera développé au chapitre 4.

2.2.3.3.2/ Problèmes dynamiques Dans les versions précédemment décrites de PSO, on considérait que les fonctions objectif étaient certes très fortement non-linéaires (ou tout du moins qu'elles correspondaient à la description des problèmes difficiles faite dans la partie 2.1.1.1), mais qu'elles étaient statiques dans le temps. Cependant, de nombreuses fonctions peuvent être considérées comme dynamiques. Comme décrit dans l'article de Eberhart et al. [100], il existe plusieurs manières dont les fonctions peuvent changer au cours du temps :

- La valeur optimale de la fonction objectif peut se déplacer dans l'espace des solutions au cours du temps,
- La position de la valeur optimale dans l'espace des solutions peut rester constante, mais sa valeur peut changer au cours du temps,
- La position dans l'espace des solutions et la valeur du minimum global de la fonction objectif considérée peuvent changer au cours du temps,
- Sur des domaines à plusieurs dimensions, ces changements peuvent avoir lieu sur une seule ou plusieurs dimensions du domaine, de manière séquentielle ou simultanée.

De fait, deux problèmes doivent être pris en compte et résolus par ces méthodes PSO adaptées [166] :

- La mémoire n'est plus d'actualité du fait du dynamisme de l'environnement (notée OM : *Outdated memory* dans la littérature). En effet, dès qu'un changement a lieu dans l'environnement, la mémoire de la meilleure solution trouvée n'est plus d'actualité, et cette mémoire peut donc guider les particules de l'essaim vers une zone qui n'est plus prometteuse.
- La perte de la diversité de l'essaim (notée DL : *diversity loss* dans la littérature). Ce problème apparaît lorsque l'essaim a convergé et que l'environnement subit un changement. En effet, dès lors que l'algorithme a convergé, les particules ont des vitesses très basses qui leur permettent de chercher plus précisément l'optimum global (phase d'exploitation d'une zone prometteuse). Ainsi, dès que l'environnement subit un changement, les particules ne sont pas réactives et ne se déplacent pas suffisamment rapidement dans le domaine afin de partir à la recherche du nouvel optimum.

Ainsi, pour résoudre ce type de problèmes dynamiques, de nombreuses améliorations de l'algorithme PSO initial ont été proposées pour être en mesure de traiter ce type de problèmes [125]. Eberhart et al. proposent d'utiliser une inertie elle aussi aléatoire. Ainsi, ne sachant jamais si des changements ont eu lieu sur le domaine, on n'est jamais vraiment en mesure de savoir si l'on doit favoriser la phase d'exploration du domaine ou celle d'ex-

3. Le mot *topologie* sera entendu ici au sens de la répartition de matière dans le domaine, dans le cas d'une optimisation topologique appliquée à un domaine physique.

exploitation des zones prometteuses [100]. Ainsi, cette inertie aléatoire permet de résoudre ce problème en ne définissant pas clairement si l'on est dans une phase d'exploration ou d'exploitation. Elle est définie à la manière décrite dans l'équation suivante

$$\omega = 0.5 + \frac{\text{rand}()}{2} \quad (2.36)$$

Le paramètre d'inertie est donc compris entre 0.5 et 1, et la valeur moyenne obtenue est 0.75. Ces résultats sont donc cohérents avec les observations concernant l'inertie optimale qui avaient été faites par Clerc et al. [85].

Bien que cet article propose une solution somme toute cohérente avec le problème de variations aléatoires de la fonction objectif évoqué, il reste assez évasif quant aux résultats obtenus. On peut aisément imaginer que les calculs peuvent être excessivement longs, puisque l'on n'adapte pas l'algorithme aux changements du domaine. L'article de Parsopoulos et al. [104] propose justement de tester cette méthode sur des fonctions Benchmark classiques. En est finalement conclu que cette méthode est très efficace sur les fonctions dynamiques. Il semblerait même que les changements de domaines aideraient l'algorithme PSO à tendre vers l'optimum global de la fonction objectif.

Il a été proposé plus tard de traquer les changements dans le domaine, afin d'appliquer des changements dans la population uniquement lorsque le domaine a évolué. Hu et al. [101] ont proposé la méthode *Changed-Gbest-value* qui réévalue la *fitness* de la meilleure particule à chaque itération. Si cette valeur de *fitness* change, on considère que le domaine a changé et on modifie alors la population. Par suite, Hu et al. [110] proposent une approche diamétralement opposée : *Fixed-Gbest-value* qui considère plutôt que si la position de la particule Gbest ne change pas pendant un certain nombre d'itérations, c'est que le domaine a évolué. En pratique, on étudie plutôt l'évolution des positions des deux meilleures particules, afin d'éviter les fausses alertes, c'est à dire afin de ne pas effectuer de modifications inutiles de la population alors que le domaine n'a pas changé. Ainsi, le nombre d'itérations au bout duquel on considère que le domaine a changé si la particule Gbest n'a pas changé doit être choisi judicieusement. Dans cet article, une première version d'algorithme intelligent, ou tout du moins adaptatif, est aussi proposée, et constitue la partie vraiment passionnante de cet article. Deux méthodes de détection de changement de domaine sont utilisées, et 9 réponses peuvent être appliquées en fonction des détections précédentes, comme on le voit dans le tableau 2.1.

Finalement, l'idée d'un algorithme adaptatif est très intéressante, et permet de faire évoluer la population lorsque le domaine change. Par contre, l'idée de redéfinir aléatoirement une partie de la population choisie de manière arbitraire me paraît difficile à appliquer. La redéfinition des particules devrait, [selon moi](#), n'être faite qu'après un classement de la population, ce qui permettrait de garder les particules les plus performantes, et de redéfinir de manière aléatoire celles qui sont manifestement très loin d'un potentiel minimum global.

Basée sur la méthode de classement pyramidal des particules proposées par Janson et al. [122], et sur la méthode de partition de la population proposée par Rhomdane et al. [233], Janson et al. ont proposé une méthode appelée *Partitionned-Hierarchical Particle Swarm Optimizer* (PH-PSO) permettant de traiter les problèmes d'optimisation définis sur des environnements dynamiques. Pour détecter les changements dans l'environnement,

Méthode de détection	A	"Changed-Gbest-value"
	B	"Fixed-Gbest-value"
Méthode de réponse	0	Ne rien faire
	1	Redéfinir aléatoirement 10 des particules et effacer la mémoire des autres particules
	2	Redéfinir aléatoirement 10 des particules.
	3	Redéfinir aléatoirement la particule Gbest et effacer la mémoire des autres particules.
	4	Redéfinir aléatoirement la particule Gbest.
	5	Effacer la mémoire de toutes les particules.
	6	Redéfinir de manière aléatoire 50 des particules et effacer la mémoire des autres particules.
	7	Redéfinir aléatoirement 50 des particules.
	8	Redéfinir aléatoirement toutes les particules

Tableau 2.1 – Détection de changements dans le domaine et réponses associées

la méthode *Changed-Gbest-value* précédemment décrite est utilisée. La population est organisée en une hiérarchie pyramidale identique à celle décrite dans [122]. Dès qu'un changement est détecté dans l'environnement, la hiérarchie est divisée en plusieurs sous-essaims en séparant la hiérarchie à un level H_{div} précédemment défini, comme on le voit sur la figure 2.21. Les particules situées juste au dessous de la coupure, c'est à dire les particules situées au niveau $H_{div} + 1$ sont complètement réinitialisées (vitesse et position) de manière aléatoire. Les particules situées au niveau $H_{div} + 2$ voient leur mémoire effacée. Ainsi, ces différents sous-essaims évoluent séparément pendant un certain nombre d'itérations (qui peut être variable) avant de se rejoindre en un unique essaim. Cela permet de chercher des potentielles zones prometteuses dans différentes régions de l'espace, lorsque le domaine a été modifié au cours du temps. Dans une deuxième version de cet algorithme, appelée *PH-PSO adaptive* (PH-PSO-a), le nombre d'itérations au bout duquel l'essaim est regroupé est une variable. On considère alors que dès qu'un des sous-essaims contient la meilleure particule Gbest pendant 5 itérations (pas nécessairement consécutives), une zone prometteuse a été trouvée, et les essaims sont regroupés. Après le regroupement, les particules sont influencées par la totalité de l'essaim et convergent donc vers ladite zone prometteuse. Testées sur de nombreuses fonctions dynamiques, ces nouvelles méthodes s'avèrent être assez efficaces sur les domaines dynamiques, sauf lorsque l'on impose un nombre maximum d'itérations assez petit (dans les faits, moins de 30 itérations est un nombre assez petit. Cette valeur a été déterminée de manière empirique dans la littérature). En effet, dans ce cas, et sur certaines fonctions, le domaine ne subit aucune modifications, et la méthode PSO classique est plus efficace que la méthode partitionnée et hiérarchisée, qui est plus compliquée et donc plus lourde en terme de temps de calcul.

Basé sur les observations faites concernant le déplacement des fourmis que l'on a vu dans la partie 2.2.2.1, un mécanisme d'évaporation a été ajouté à PSO afin de traiter les cas où l'environnement est dynamique. En effet, dans les articles [188, 197] de Fernandez-Marquez et al., une critique est faite concernant les précédentes améliorations faites pour adapter PSO aux environnements dynamiques, et notamment concer-

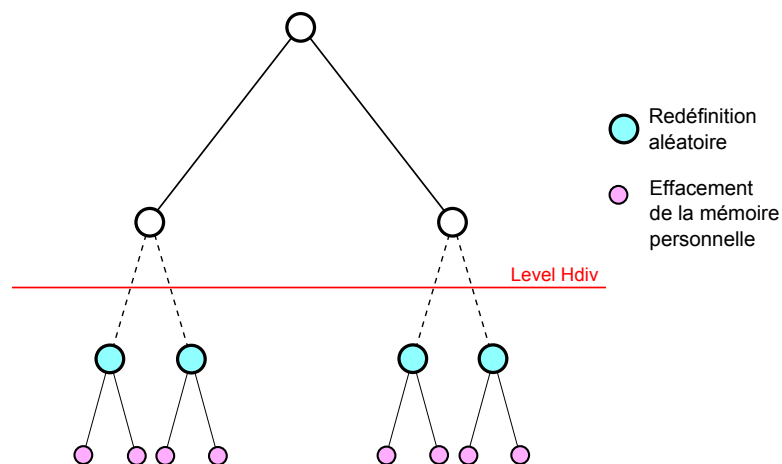


FIGURE 2.21 – Représentation géométrique de la topologie partitionnée et hiérarchisée

nant le fait que l'environnement est constamment réévalué pour vérifier si une modification a eu lieu. De fait, lorsque l'environnement est dynamique, et que la fonction objectif est bruitée, l'évaluation de la fonction objectif donnera nécessairement des valeurs différentes à chacune des évaluations, quand bien même aucune modification n'aurait eu lieu dans le domaine. Ainsi, ces variations de la fonction objectif dues au bruit, et non pas à une véritable modification du domaine, sont mal interprétées par l'algorithme, et les mémoires des particules sont constamment réinitialisées.

Il est donc considéré ici que la confiance allouée à chacune des particules par l'essaim décroît au cours du temps. Un mécanisme d'évaporation pénalisera les points optimaux visités au cours du temps. Ainsi, un nouveau point optimal aura donc plus d'influence qu'un point optimal découvert de nombreuses itérations auparavant.

2.2.3.3.3/ Variantes adaptatives et hybrides de PSO

Généralités préliminaires

Dans cette partie, nous verrons que de nombreuses versions hybrides adaptatives et intelligentes de PSO ont été développées, concernant l'évolution de la topologie ou des paramètres numériques de PSO.

Avant de présenter les différents algorithmes proposés dans le domaine des algorithmes adaptatifs, il est important de noter l'étude empirique réalisée par De Oca et al. [187]. En effet, avant de proposer une nouvelle version de PSO, De Oca et al. ont étudié de manière statistique l'influence de chacun des paramètres de l'algorithme sur sa convergence. Les conclusions suivantes ont été tirées de ce travail :

- L'algorithme PSO peut présenter des phases de stagnation assez importantes au cours du calcul. Ces phases de stagnation sont moins sévères lorsqu'une large population est utilisée et/ou que la topologie n'est pas trop connectée. Ces phases de stagnation sont plus sévères lorsque la taille de la population est petite et/ou que la topologie est très connectée. Notons ici la forte influence de la topologie sur la performance des algorithmes.
- Les algorithmes PSO sont sensibles aux modifications de topologies de la population.
- Des redémarrages aléatoires de l'algorithme peuvent augmenter la performance de l'algorithme. Cependant, les probabilités de redémarrage sont extrêmement dépendantes à la fois du problème étudié et de l'algorithme utilisé. Aucune conclusion ne peut donc être tirée *a priori* concernant ces redémarrages aléatoires.
- Lorsque la qualité de la solution est requise, les algorithmes explorateurs du domaine sont les plus performants.
- Il est possible de contrôler les modifications des paramètres de l'algorithme au cours du temps. Les vitesses de variation de ces paramètres peuvent avoir un effet considérable sur les capacités de convergence de l'algorithme.

Méthodes hybrides

De nombreuses méthodes hybrides de l'algorithme PSO ont d'ores et déjà été développées par le passé. Le premier algorithme hybride a été développé par Angeline [76], et propose d'ajouter les mécanismes de sélection et de remplacement que l'on retrouve dans les algorithmes évolutionnaires (cf. partie 2.2.2.2). Après chaque itération, la *fitness* de chaque particule est comparée à celle des k autres particules de son voisinage (il est important de noter ici que la mémoire des particules n'est pas utilisée pour effectuer ce classement. Seule la valeur de la *fitness* pour la position [actuelle](#) des particules est utilisée). Chaque particule gagne un point pour chaque particule comparée dont la valeur de la *fitness* est plus petite que la sienne. La population est alors classée en fonction de ce score. Une fois les particules classées, les positions et vitesses de la meilleure moitié des particules sont partagées à la moins bonne moitié de l'essaim. La moins bonne moitié de l'essaim est donc téléportée dans l'espace des solutions, et récupère les valeurs des vitesses des meilleures particules. Cependant, ces particules gardent tout de même leur expérience personnelle en mémoire, à savoir le souvenir de leur meilleure expérience personnelle, qui sera utilisée dans le calcul de leur prochaine position dans l'espace des solutions. Ainsi, la dynamique de l'essaim, bien que basée sur le déplacement des particules de l'algorithme PSO, est inspirée des théories évolutionnaires. Les résultats

montrent que ce type d'algorithme hybride est plus performant que le seul algorithme PSO.

De la même manière, Niu et al. [160] utilisent une étape de reproduction basée sur un classement de la population. En effet, basée à la fois sur le mécanisme de reproduction utilisé dans le domaine des algorithmes évolutionnaires, mais aussi sur les théories de sélection naturelle de Darwin, cette méthode est organisée de la manière suivante :

1. Les particules sont classées en fonction de la valeur de la fonction objectif qu'elles obtiennent dans leur position actuelle.
2. L'essaim est divisé en deux groupes en fonction de ce classement : les particules dont la fonction objectif est la plus proche de la valeur à atteindre, et les autres.
3. Les individus les plus faibles sont éliminés, et chaque individu restant est dupliqué, afin de maintenir le nombre d'individus de la population.

Le concept de mutation a lui aussi été adapté à l'algorithme PSO. En effet, Ratnaweera et al. [137] proposent de réinitialiser la vitesse des particules si aucune amélioration n'a lieu pendant un nombre donné d'itérations. Pour aller plus loin, Poli et al. [149, 150] ont même proposé d'utiliser les algorithmes évolutionnaires, non pas pour faire évoluer la population de l'algorithme PSO ou ses paramètres, mais pour générer de nouvelles versions de PSO. D'après les résultats présentés dans ces articles, ces algorithmes évolutionnaires ont généré des nouvelles versions de PSO (c'est à dire des versions dont les paramètres sont eux-mêmes optimisés) plus performantes que celles développées précédemment dans la littérature. Un couplage PSO/ACO (cf partie 2.2.2.1) a notamment été développé par Holden et al. [145], comme nous l'avons vu dans la partie 2.2.2.1.6.

Adaptation de la topologie

De nombreuses modifications et améliorations de l'algorithme PSO initial ont été proposées concernant l'évolution de la population des particules au cours des itérations. Jiang et al. [172] proposent par exemple d'utiliser plusieurs sous-essaims, au sein de la population globale. L'algorithme est défini comme suit :

1. La population est classée en fonction de la performance des particules vis à vis de la fonction objectif à optimiser.
2. La population est partitionnée en sous-essaims contenant chacun m particules.
3. Chacun des ces sous-essaims effectue, **séparément**, des itérations PSO.
4. Au bout d'un certain nombre d'itérations précédemment défini, les essaims sont remélangés. Les particules sont reclassées. L'essaim est à nouveau partitionné.
5. Si les critères de convergence sont atteints, alors on arrête le calcul. Sinon, retour au point n^o3 .

Cette approche est très intéressante non seulement parce qu'elle est très efficace par rapport à l'algorithme PSO classique, mais aussi parce qu'elle commence à faire appel à des notions d'intelligence artificielle qui nous ont intéressés dans la suite de notre travail. En effet, la partition du travail est une notion prépondérante dans de nombreuses méthodes d'optimisation, et méthodes d'intelligence artificielle. On la retrouve notamment dans l'algorithme Ant Colony Optimization (cf. partie 2.2.2.1). De plus, la notion de classement que l'on a déjà vue dans le travail de Akbari et al. [203] et Angeline [76]

est réutilisée, et donne, dans ces trois travaux, des résultats très en deà des capacités de convergence observées chez les autres algorithmes d'optimisation basés sur PSO. Comme on l'a vu dans la partie 2.2.2.1.6, Lovbjerg et al. [103] ont aussi proposé une version de PSO utilisant le concept de sous-populations. Dans ce cas, le seul contact entre les sous-populations est le possible mariage entre deux particules de populations différentes. Le mariage entre deux particules, très fortement inspiré des mécanismes à l'origine des algorithmes évolutionnaires, est défini dans l'équation (2.37) suivante. La position des enfants définis par le mariage de deux particules est finalement défini par un croisement arithmétique entre la position des deux parents.

$$\begin{cases} enfant_1(x_i) = p_i \times parent_1(x_i) + (1.0 - p_i) \times parent_2(x_i) \\ enfant_2(x_i) = p_i \times parent_2(x_i) + (1.0 - p_i) \times parent_1(x_i) \end{cases} \quad (2.37)$$

où p_i est une valeur aléatoire uniformément distribuée dans $[0, 1]$.

Afin de maintenir la taille de la population, les deux parents sont éliminés et remplacés par leurs enfants.

De plus, les vecteurs vitesse des deux enfants sont déterminés comme la somme des vecteurs vitesse des deux parents, normalisés par rapport à la dimension originale de ces vecteurs, comme on le voit dans les équations suivantes

$$\begin{cases} enfant_1(\mathbf{v}) = \frac{parent_1(\mathbf{v}) + parent_2(\mathbf{v})}{|parent_1(\mathbf{v}) + parent_2(\mathbf{v})|} \times |parent_1(\mathbf{v})| \\ enfant_2(\mathbf{v}) = \frac{parent_1(\mathbf{v}) + parent_2(\mathbf{v})}{|parent_1(\mathbf{v}) + parent_2(\mathbf{v})|} \times |parent_2(\mathbf{v})| \end{cases} \quad (2.38)$$

Ainsi, dans cette version partitionnée de PSO, chaque sous-population possède son propre G_{best} , c'est à dire sa propre meilleure particule qui, de fait, n'influence que les individus de sa sous-population. Afin de favoriser le mélange entre populations, et donc l'échange d'informations entre les populations, un paramètre supplémentaire est utilisé appelé PSB (*Probability of same population breeding*) définissant la probabilité d'une particule de se reproduire avec une particule appartenant à la même sous-population qu'elle. Ainsi, la probabilité de se reproduire avec une particule d'une autre sous-population est définie par la probabilité $(1 - PSB)$. La probabilité PSB est fixée de manière empirique à 0.2 dans ce travail. Cependant, puisque les mariages entre populations favorisent le partage des connaissances, et donc l'exploration du domaine des solutions, on pourrait imaginer que la probabilité PSB augmente au cours des itérations. Ainsi, le partage des informations serait favorisé au début du calcul, avantageant ainsi la phase d'exploration du domaine. Au cours des itérations, la probabilité PSB augmentant, la probabilité de reproduction entre les populations diminue, favorisant alors la phase d'exploitation des zones prometteuses.

Modification des paramètres de l'algorithme

Comme nous l'avons vu précédemment dans l'équation (2.24), de nombreux paramètres gèrent le comportement des particules PSO dans l'espace des solutions : les paramètres c_1 et c_2 qui pondèrent respectivement l'influence de l'expérience individuelle et de la communication sociale sur le déplacement des particules [121], et le paramètre d'inertie ω qui contrôle l'impact de l'historique des particules sur ces déplacements futurs, c'est à dire l'influence de la mémoire d'un individu sur son comportement.

Shi et al. [128] ont tout d'abord proposé qu'une grande inertie aurait tendance à favoriser la phase d'exploration du domaine, alors même qu'une petite inertie favoriserait la phase d'exploitation des zones prometteuses. C'est la raison pour laquelle ils proposent de diminuer l'inertie ω au cours des itérations, plutôt que de choisir une valeur fixe précédemment définie, ω étant définie par la formule (2.39) suivante

$$\omega = (iter_{max} - iter_{cur}) \left(\frac{\omega_{initial} - \omega_{final}}{iter_{max}} \right) + \omega_{final} \quad (2.39)$$

Par la suite, la méthode contraire est proposée par Zheng et al. [130]. En effet, il est développé qu'une grande valeur du paramètre d'inertie rendrait l'algorithme plus stable, et qu'une petite valeur impliquerait que les particules fassent des sauts dans le domaine des solutions, rendant leurs déplacements moins stables. C'est la raison pour laquelle Zheng et al. proposent d'augmenter l'inertie ω au cours des itérations. Ainsi, les particules auraient tendance à se déplacer très rapidement dans l'espace des solutions au début de la convergence, favorisant de fait l'exploration du domaine, et à se stabiliser au cours des itérations. ω augmente donc **linéairement** de 0.4 à 0.9 au cours des itérations.

Par la suite, Fan et al. [168] proposent d'adapter l'inertie ω au cours des itérations en fonction de l'évolution de la recherche. Dès que la recherche a tendance à stagner, l'inertie diminue. Dès que la recherche trouve une zone prometteuse, l'inertie augmente, comme on le voit dans les équations (2.40) et (2.41) suivantes

$$\omega = (d)^r \times \omega_{initial} \quad (2.40)$$

$$\begin{cases} Si & f(P_{gd,new}) < f(P_{gd,old}) & alors & r \leftarrow r - 1; \\ Si & f(P_{gd,new}) \geq f(P_{gd,old}) & alors & r \leftarrow r + 1; \end{cases} \quad (2.41)$$

avec :

- d : le taux de réduction. Une constante choisie empiriquement entre 0.4 et 1,
- r : le paramètre d'adaptation dynamique déterminé au cours des itérations par les équations (2.41),
- $P_{gd,old}$: la meilleure position trouvée à l'itération précédente,
- $P_{gd,new}$: la meilleure position trouvée à l'itération en cours.

Dans le premier cas de l'équation (2.41), la nouvelle valeur de la fonction objectif $f(P_{gd,new})$ est inférieure à la valeur de la fonction objectif de l'itération précédente $f(P_{gd,old})$, elle est donc considérée comme étant meilleure dans le contexte d'une optimisation par minimisation de la fonction objectif. C'est la raison pour laquelle la valeur de l'inertie ω diminue, pour essayer d'exploiter la zone prometteuse trouvée.

Dans le second cas de l'équation (2.41), la nouvelle valeur de la fonction objectif $f(P_{gd,new})$ est supérieure à la valeur de l'itération précédente, elle n'est donc pas meilleure (l'algorithme n'a pas trouvé de nouvelle particule G_{best}). C'est la raison pour laquelle la valeur de l'inertie augmente, afin de relancer la phase d'exploration du domaine.

On obtient ainsi l'évolution de ω au cours des itérations de la figure 2.22 suivante.

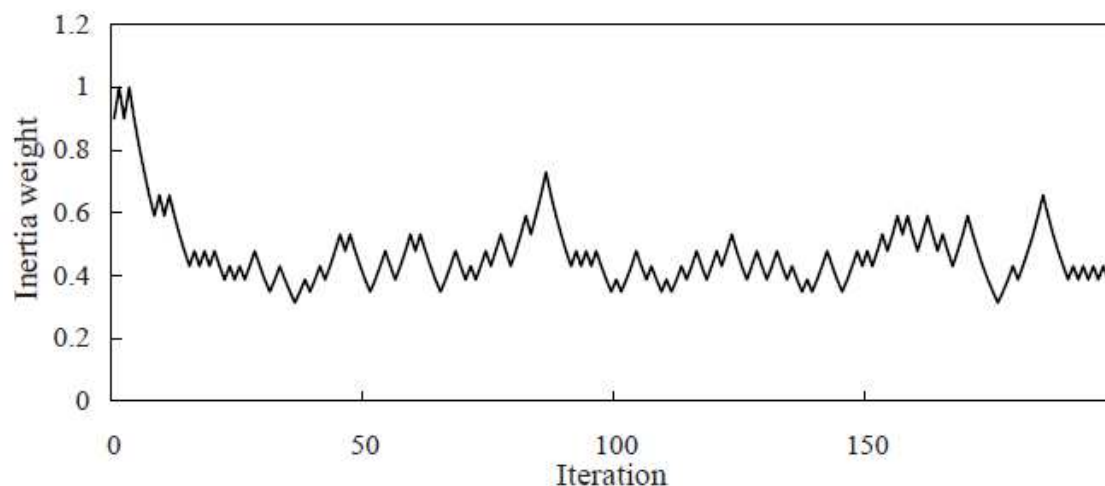


FIGURE 2.22 – Evolution de l'inertie oméga au cours des itérations [168]

On voit bien sur la figure 2.22 que l'inertie peut être amenée à croître ou décroître au cours des itérations, afin de favoriser la convergence de l'algorithme. Les résultats présentés par Fan et al. montrent bien que cet algorithme intelligent est bien plus performant que l'algorithme PSO classique, sur les fonctions objectif sur lesquelles il a été testé.

D'autres méthodes de variations intelligentes de l'inertie ont été proposées notamment par Yang et al. [175] et Tawdross et al. [165]. Finalement, un algorithme PSO complètement adaptatif, basé sur des notions d'intelligence artificielle, a été proposé par Hashemi et al. [206]. Deux algorithmes adaptatifs sont proposés dans ce travail :

- Dans le premier (UPPSO pour *United or Unified Adaptive PSO*), les particules partagent les mêmes paramètres.
- Dans le deuxième (IAPSO pour *Intelligent or Independant Adaptive PSO*), les particules sont indépendantes et ajustent leurs paramètres en utilisant chacun un algorithme d'intelligence artificielle.

Le principe de ces algorithmes d'intelligence artificielle est le choix d'actions, en fonction de résultats obtenus. Ainsi, en fonction des résultats de convergence obtenus, les paramètres ω , c_1 et c_2 sont réadaptés, soit en choisissant une valeur existante dans une table de choix, soit en étant directement modifiés via une formule d'évolution. Finalement, cette méthode rejoint tous les algorithmes intelligents précédemment développés, en adaptant les paramètres de chaque particule, en fonction de leur évolution personnelle dans le domaine des solutions. Bien sûr, cet algorithme est bien plus performant que tous les autres.

2.2.3.3.4/ PSO adapté aux problèmes multiobjectifs Tout comme les autres méthodes metaheuristiques, l'algorithme PSO a été adapté à la résolution des problèmes d'optimisation multiobjectif. En effet, à la différence des problèmes d'optimisation mono-objectif, où la solution optimale est unique, il existe plusieurs solutions optimales à trouver lorsque l'on résout des problèmes d'optimisation multiobjectifs : les solutions optimales au sens de Pareto, comme on l'a vu dans la partie 2.1.2. Ces points représentent les solutions du problème, lorsque tous les objectifs sont considérés simultanément, c'est à dire les points représentant les meilleurs compromis entre les différentes fonctions objec-

tif. Premièrement, Parsopoulos et al. [104] ont été les premiers à proposer une version de PSO adaptée à l'optimisation multiobjectif, et leurs méthodes ont inspiré toutes les recherches dans le domaine, par la suite.

Comme nous l'avons vu dans la partie 2.1.2, la méthode de pondération des objectifs est la plus couramment utilisée pour résoudre des problèmes d'optimisation multiobjectif. Les k objectifs sont sommés et pondérés en une combinaison linéaire, afin de n'obtenir plus qu'une seule fonction objectif à optimiser, comme on le voit dans l'équation suivante

$$F = \sum_{i=1}^k w_i \times f_i(x) \quad (2.42)$$

avec w_i pour $i = 1...k$ qui sont les coefficients de pondération. On a généralement $\sum_{i=1}^k w_i = 1$. La première méthode de gestion de ces coefficients de pondération est la méthode CWA (*Conventional Weighted Aggregation*) dans laquelle les coefficients de pondération sont fixés et statiques au cours des itérations. Cependant, cette méthode ne permet de définir, à chaque fonctionnement de l'algorithme, qu'un seul point Pareto. On doit donc réutiliser l'algorithme autant de fois que l'on souhaite obtenir de points Pareto. Elle n'est pas très efficace pour résoudre des problèmes multiobjectif difficiles, mais très efficace lorsque le front de Pareto est concave (au sens mathématique défini dans la partie 2.1.2). La seconde méthode de gestion des coefficients de pondération est la méthode BWA (*Bang-Bang Weighted Aggregation*). Dans ce cas, et lorsque l'on a deux objectifs à minimiser, les coefficients de pondération évoluent en corrélation avec les formules (2.43)

$$\begin{cases} w_1(t) = \text{sign}(\sin(\frac{2\pi t}{F})) \\ w_2(t) = 1 - w_1(t) \end{cases} \quad (2.43)$$

avec :

- t : l'itération en cours,
- F : La fréquence de changement des coefficients de pondération

En effet, la fonction **sign** implique un changement radical des coefficients w_i à la fréquence F (d'où le nom *Bang-Bang Weighted Aggregation*).

Finalement, la dernière méthode de gestion des coefficients de pondération fait évoluer de manière graduelle les coefficients w_i , grâce aux équations suivantes

$$\begin{cases} w_1(t) = \left| \sin(\frac{2\pi t}{F}) \right| \\ w_2 = 1 - w_1(t) \end{cases} \quad (2.44)$$

Cette méthode est appelée DWA (*Dynamic Weighted Aggregation*). Le changement assez lent des coefficients de pondération force l'algorithme à chercher autour des points du front de Pareto, c'est à dire à favoriser la phase d'**exploitation** des zones prometteuses.

Ces méthodes permettent d'obtenir des résultats cohérents, de manière assez stable et certaine, avec la théorie de Pareto, comme on le voit dans les résultats présentés dans l'article [104].

Deuxièmement, l'approche VEGA, développée par Schaffer et al. [26], adaptée des algorithmes génétiques, que l'on a vu dans la partie 2.2.2.2.3 permet de traiter les

problèmes d'optimisation multiobjectif en utilisant le concept de sous population que l'on a étudié précédemment. Ici, la méthode VEPSO, inspirée de VEGA, est une adaptation de l'algorithme PSO aux problèmes d'optimisation multiobjectif. De la même manière que dans VEGA, les particules de l'essaim sont divisées en deux sous-essaims. Chaque sous-essaim est évalué en fonction d'une seule fonction objectif à traiter. Cependant, les informations de mémoire de chaque sous-essaim sont utilisées pour évaluer les nouvelles vitesses des particules de l'autre sous-essaim. Ainsi, le partage des informations entre les essaims permet une meilleure communication, et donc la détermination concrète et rapide des fronts de Pareto des fonctions évaluées.

Finalement, bien que les méthodes développées par Parsopoulos soient les premières dans le domaine de l'optimisation multiobjectif avec PSO, elles sont, d'après les résultats présentés dans l'article [104] très encourageantes, et donnent des résultats très satisfaisants.

Une autre méthode, en apparence plus simple, a été développée par Hu et al. [111]. En effet, Hu et al. ont déterminé qu'il serait tout aussi simple de fixer, pour chaque particule, une valeur aléatoire de la première fonction objectif. Ainsi, pour chaque particule, il ne resterait plus que la seconde fonction objectif à optimiser. Les particules "tomberaient" sur le front de Pareto, jusqu'à ce que les solutions trouvées soient non dominées par les autres, comme on le voit dans la figure 2.23 suivante

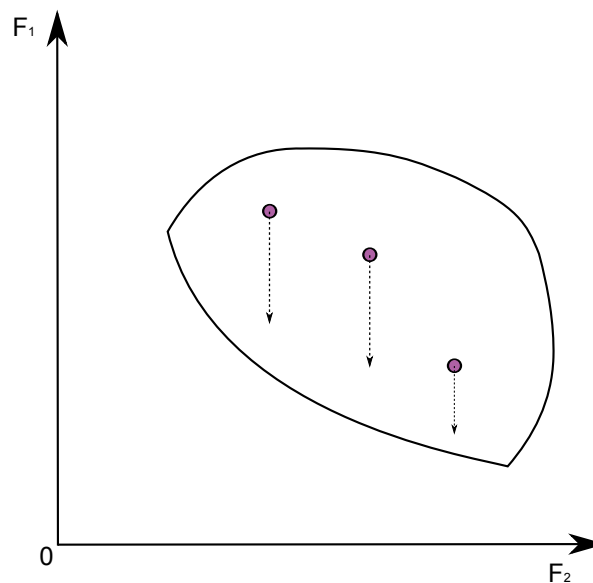


FIGURE 2.23 – Fonctionnement de la méthode d'optimisation multiobjectif développée par Hu et al.

Dans cette méthode, le voisinage est basé sur les méthodes géographiques présentées dans la partie 2.2.3.2.2, et est évolutif au cours des itérations. En effet, à chaque itération, les m particules les plus proches dans l'espace des solutions de chaque particule sont assignées comme faisant partie de son voisinage. D'après les résultats présentés dans l'article [111], cette méthode est tout aussi efficace que celles présentées précédemment.

Finalement, se basant sur l'algorithme PSO classique, et sur la version binaire proposée par Kennedy et al. [73] que l'on a vue dans la partie 2.2.3.3.1, de nombreuses recherches

ont étendu le travail sur les méthodes binaires aux problèmes multi-objectifs [177, 185]. Par la suite, alors que toutes les versions proposées étaient trop lourdes, compliquées à implémenter, et assez inefficaces sur les problèmes mono-objectifs, Shen et al. [138] ont proposé une nouvelle version de PSO binaire : MBPSO (*Modified Binary PSO*). Dans cet algorithme, les formules d'incrémentement des vitesses des particules sont les équations (2.45) suivantes

$$\begin{cases} X_i^{k+1} = X_i^k & \text{si } 0 < V_i^{k+1} \leq \alpha \\ X_i^{k+1} = P_i^k & \text{si } \alpha < V_i^{k+1} \leq \frac{1+\alpha}{2} \\ X_i^{k+1} = P_g & \text{si } \frac{1+\alpha}{2} < V_{ij}^{k+1} \leq 1 \end{cases} \quad (2.45)$$

où :

- α : un réel choisi,
- P_i^k : la position de la meilleure performance personnelle de la particule i à l'itération k ,
- P_g : la position de la meilleure particule de l'essaim.

Dans ces équations (2.45), le paramètre α est appelé **probabilité statique** et doit être choisi de manière judicieuse. En effet, une petite valeur de α aurait tendance à accélérer la convergence de l'algorithme, tout en laissant des particules coincées dans des optima locaux (si α est petit, les particules se déplacent rapidement dans le domaine, mais elles peuvent aussi avoir tendance à récupérer indéfiniment la position de leur meilleure performance personnelle, et donc à rester bloquées dans des optima locaux). A l'inverse, une grande valeur de α rendrait l'algorithme inefficace, puisque les particules ne réutiliseraient pas suffisamment leur mémoire personnelle, et auraient tendance à revenir systématiquement dans la position qu'elles avaient lors de l'itération précédente. Il est noté dans l'article de Wang et al. [209] que ces équations (2.45) ne sont finalement pas si différentes de la version classique de PSO. En effet :

- La première équation (2.45) est identique au terme relatif à l'inertie ω . Elle traduit la confiance que la particule a en sa propre expérience, sa mémoire, et en quelle mesure ces informations l'influencent.
- La deuxième équation (2.45) représente la confiance qu'a la particule en sa meilleure expérience personnelle.
- La troisième équation (2.45) représente en quelle mesure la particule est influencée par l'essaim, et donc par la meilleure particule de l'essaim.

Cependant, bien que cette méthode soit très efficace sur les problèmes multi-objectif classiques, elle ne l'est pas sur les problèmes de type Pareto (cf. partie 2.1.2.2). C'est la raison pour laquelle Wang et al. [209] ont proposé une nouvelle version binaire et multiobjectif de PSO, capable de traiter aussi les problèmes de type Pareto. Dans ce cas, les formules d'incrémentement des positions des particules de l'essaim sont données dans les équations (2.46) suivantes

$$\begin{cases} X_i^{k+1} = X_i^k & \text{pour } 0 < V_i^{k+1} \leq \alpha \\ X_i^{k+1} = P_i^k & \text{pour } \alpha < V_i^{k+1} \leq \beta \\ X_i^{k+1} = P_g & \text{pour } \beta < V_{ij}^{k+1} \leq 1 \end{cases} \quad (2.46)$$

Dans ce cas, le paramètre β ajuste la probabilité qu'une particule suive sa meilleure performance personnelle ou la meilleure performance de l'essaim. Suivant ces formules,

il est aisé de voir que, de la même manière que précédemment, les particules peuvent avoir tendance à rester bloquées dans des optima locaux. C'est la raison pour laquelle trois nouveaux opérateurs ont été ajoutés à ces équations.

- Le premier opérateur est un opérateur de **dissipation** inspiré des algorithmes d'optimisation par colonies de fourmis que l'on a vus dans la partie 2.2.2.1. Cet opérateur agit à la manière de l'équation (2.47) suivante, et réinitialise la position d'une particule avec une probabilité donnée.

$$\mathbf{X}_i^{k+1} = \text{Reini}(\mathbf{X}_i) \quad \text{si } \text{rand} < p_d \quad (2.47)$$

avec p_d la variable aléatoire généralement comprise entre 0.05 et 0.15. p_d a une valeur volontairement assez petite pour éviter la perte notoire d'informations.

- Le deuxième opérateur est un opérateur de **mutation** inspiré des algorithmes évolutionnaires que l'on a vus dans la partie 2.2.2.2. Alors que l'opérateur de dissipation permet de diversifier la population et de conserver, tout au long du fonctionnement de l'algorithme, une certaine exploration du domaine, il peut aussi supprimer des informations importantes de la population, puisqu'il agit directement sur les particules. Afin de pallier à cela, l'opérateur de mutation défini dans l'équation (2.48) agit directement sur les éléments (c'est à dire les bits, puisque l'on travaille sur un domaine binaire) des particules, en les modifiant de manière aléatoire en fonction d'une probabilité p_m . Ainsi, cet opérateur de mutation peut à la fois maintenir une certaine diversité au sein de la population, mais aussi favoriser les phases d'exploitation de zones prometteuses.

$$\text{if } \text{rand} < p_m \text{ alors } \mathbf{X}_i^{k+1} = \begin{cases} 1 & \text{si } \mathbf{X}_i^{k+1} = 0 \\ 0 & \text{si } \mathbf{X}_i^{k+1} = 1 \end{cases} \quad (2.48)$$

- Le troisième opérateur permet la **sélection** et la mise à jour des particules $\mathbf{P}_i$ (c'est à dire la meilleure performance personnelle de la particule i), et $\mathbf{G}_{best}$ (c'est à dire la meilleure performance globale de l'essaim). En effet, comme on l'a vu dans la partie 2.1.2.2, lorsque l'on souhaite effectuer une optimisation multiobjectif au sens de Pareto, on ne cherche pas une seule coordonnée extrême, mais les différents points du front de Pareto qui constituent les meilleurs compromis entre les différentes fonctions objectif considérées. Dans cet article, on utilise donc un paramètre de *niche count*, qui considère le nombre de particules situées dans un rayon autour des particules non dominées. Par exemple, comme on le voit dans la figure 2.24 suivante, le rayon considéré pour chaque particule est calculé grâce à l'équation (2.49) suivante

$$r = \frac{\max_2 + \max_1 - \min_2 - \min_1}{N - 1} \quad (2.49)$$

avec :

- $\max_i$ et $\min_i$: les valeurs maximales et minimales des deux fonctions objectif considérées,
- N : la dimension de la population, c'est à dire le nombre d'individus de l'essaim.

Comme on le voit dans la figure, la particule B possède 4 voisins, alors que la particule A en possède 8. Wang et al. considèrent alors que la particule B est supérieure à la particule A dans le sens où son voisinage est moins encombré. Ainsi, à chaque itération, les particules non dominées sont classées en fonctions de leur paramètre de *niche count*. A chaque génération, la particule $\mathbf{G}_{best}$ est choisie de manière aléatoire

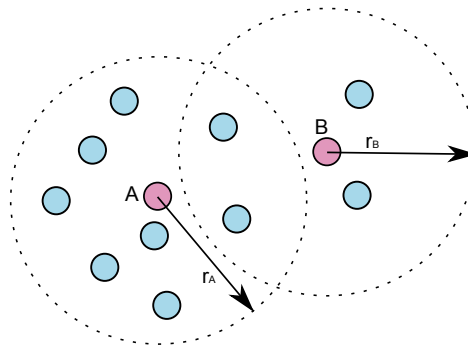


FIGURE 2.24 – Un exemple de calcul de Niche Count pour deux particules

parmi le meilleur décile de la population. Afin d'encourager l'exploration du domaine, les meilleures performances personnelles P_i sont remplacées par les autres particules non-dominées du classement.

2.3/ NOTIONS CONCERNANT LA THÉORIE DES GRAPHS

2.3.1/ INTRODUCTION

D'après [17], l'intelligence artificielle est "la recherche des moyens susceptibles de doter les systèmes informatiques de capacités intellectuelles comparables à celles des êtres humains". En outre, le terme "intelligence artificielle", créé par John McCarthy, est défini par l'un de ses créateurs, Marvin Lee Minsky, comme "la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique" [33].

Comme nous l'avons vu dans le chapitre 2.2 précédent, les méthodes metaheuristiques sont très dépendantes de leurs paramètres intrinsèques. Alors que de nombreux chercheurs ont essayé de trouver les paramètres les plus adaptés, qui proposant de les faire évoluer linéairement ou non linéairement au cours des itérations, qui proposant de les faire évoluer en fonction des réponses données par l'algorithme, il nous paraît évident que seule l'intelligence artificielle pourrait efficacement traiter ce problème de choix des paramètres ou du choix de la topologie de la population des algorithmes metaheuristiques d'optimisation basés sur la gestion de populations.

Ce chapitre a donc pour but de présenter quelques bases et notions élémentaires de la théorie des graphes et de l'intelligence artificielle.

2.3.2/ GÉNÉRALITÉS CONCERNANT LA THÉORIE DES GRAPHS

Née des recherches d'Euler au 18^{ème} siècle, la théorie des graphes s'est imposée comme une branche importante des mathématiques, notamment grâce à l'avènement de l'informatique et de ses théories concernant la notion de réseau. La théorie des graphes permet de résoudre une très grande quantité de problèmes, dans de nombreux domaines différents (réseau social, réseau informatique, télécommunications, et même

en sociologie, avec la compréhension des mouvements de populations), tant le concept de graphe est général [148, 236]. L'étude des graphes et de leurs applications permet d'aborder de nombreuses et très différentes applications. C'est ainsi qu'on développera par exemple des méthodes d'ordonnement de tâches à partir de chemins optimaux dans les graphes, ou encore des propriétés de réseaux de communication, grâce à l'étude de la connectivité des graphes [159].

La théorie des graphes, c'est à dire l'étude de réseaux de communication entre des "nœuds" reliés par des "lignes" (nous verrons une définition mathématique plus adaptée par la suite), nous paraît donc particulièrement adaptée dans ce travail de thèse de doctorat. En effet, alors qu'il a été démontré notamment par Mendes et al. [113, 127, 136, 140] que la topologie de la population peut avoir une réelle influence sur les performances de l'algorithme PSO, la théorie des graphes semble très adaptée à la détermination d'une topologie de la population évolutive, afin d'optimiser les performances de l'algorithme d'optimisation.

A partir de ce modèle simple, des points reliés par des lignes, de nombreuses questions s'imposent déjà [159], qui sont identiques à celles générées par l'étude des topologies de population dans PSO : si l'on considère un réseau de nœuds, les liens permettent-ils à tous les nœuds de communiquer entre eux ? C'est à dire d'un nœud donné, peut-on se rendre à n'importe quel autre, soit directement, soit en passant par d'autres nœuds ? La théorie des graphes est généralement illustrée par le problème dit "des ponts de Königsberg". Résolu par Euler en 1736, il peut s'énoncer de la manière suivante : "En partant d'une zone de la ville, est-il possible de retourner dans la même zone en traversant chacun des sept ponts une fois et une seule ?" et se représente à la manière de la figure 2.25.

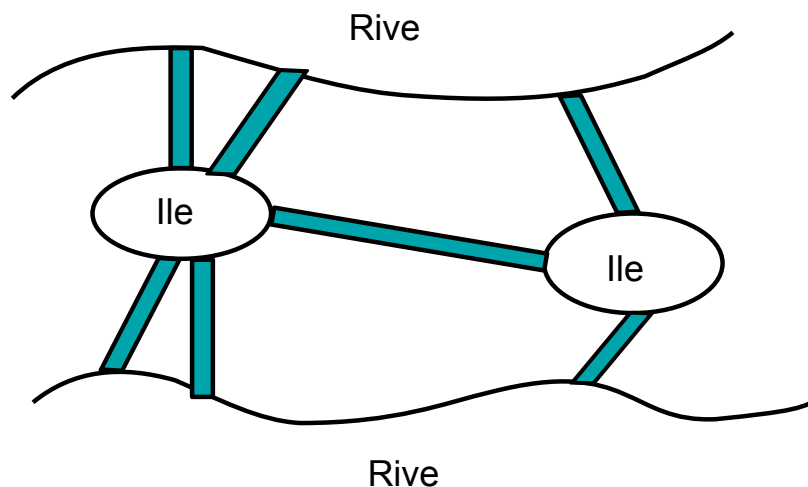


FIGURE 2.25 – Exemple typique de problème à résoudre par la théorie des graphes

Une autre illustration typique est celle du voyageur de commerce, que l'on a vue dans la partie 2.2.2.1.4.

2.3.2.1/ DÉFINITIONS

2.3.2.1.1/ Graphe non-orienté

Principe de base

Mathématiquement, un graphe peut être défini de la manière suivante

Définition 6 : Graphe non-orienté

Un **graphe non-orienté** G est défini par deux ensemble finis [159] : un ensemble X , non vide, d'éléments appelés **sommets**, un ensemble E , qui peut être vide, d'éléments appelés **arêtes**. A chaque arête sont associés deux sommets x et y , distincts ou non, appelés les **extrémités** de E . On note alors $G(X, E)$.

Dans l'exemple représenté sur la figure 2.26, on a donc les définitions suivantes :

- a, b, c et s sont les **sommets** du graphe.
- $1, 2, 3, 4, 5$ et 6 sont les **arêtes** du graphes.
- L'arête 5 est dite **incidente** aux sommets a et c . c et a sont appelés les **extrémités** de l'arête 5.
- Les sommets a et c , étant incidents à une même arête, sont dits **adjacents** ou **voisins**.
- L'arête 4, dont les deux extrémités sont confondues, est appelée **boucle en a** .

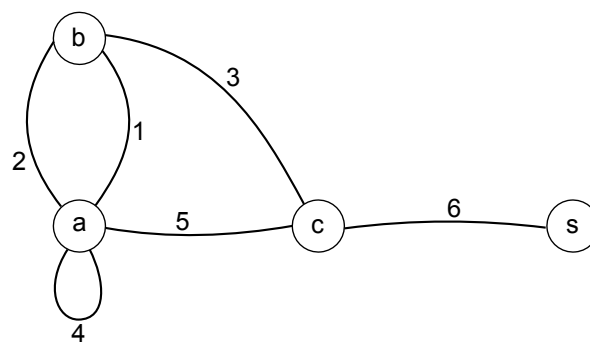


FIGURE 2.26 – Exemple de graphe et définitions

On appelle f la **fonction d'incidence** du graphe. Ainsi, sur l'exemple ci dessus, on a : $f(4) = a$ et $f(2) = f(1) = \{a, b\}$.

L'ensemble des sommets d'un graphe est souvent noté $V(G)$.

Lorsqu'il y a plusieurs arêtes entre deux mêmes sommets, on parle d'**arêtes multiples**.

Un graphe ne présentant ni arêtes multiples, ni boucles et appelé un **graphe simple**. Un graphe non nécessairement simple peut être qualifié de **graphe multiple**.

Degré d'un sommet, ordre d'un graphe

Le **degré** d'un sommet est le nombre d'**arêtes incidentes** à ce sommet [148] (la présence d'une boucle ajoute 1 au degré).

Le **nombre de sommets** d'un graphe est appelé l'**ordre** de ce graphe.

Par exemple, dans le graphe de la figure 2.26, le degré est 6, et l'ordre 4.

2.3.2.1.2/ Graphe orienté

Définition 7 : Graphe orienté

Un **graphe orienté** [159] G est défini par deux ensemble finis : un ensemble X non vide de sommets, un ensemble A d'**arcs**, ou **arêtes orientées**, avec un couple (x, y) de sommets associé à chaque arc qui sont les extrémités de a . Le sommet est, en particulier, l'**origine** de l'arc.

Un exemple de graphe orienté est donné dans la figure 2.27.

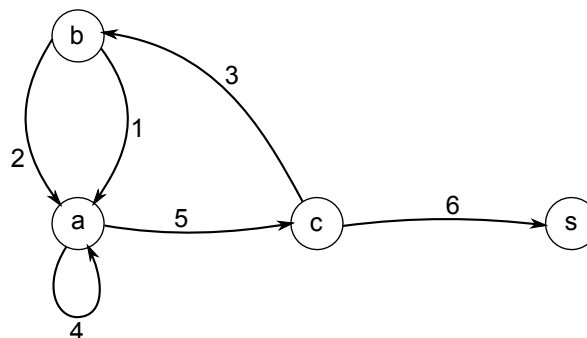


FIGURE 2.27 – Un exemple de graphe orienté

Terminologie

Lorsque (x, y) est le couple associé des extrémités de l'arc a , on dit que

- L'arc a va du sommet x au sommet y .
- L'arc a est incident aux sommets x et y .
- L'arc a sort du sommet x et qu'il entre dans le sommet y .
- Le sommet y est appelé successeur de x , et le sommet x un prédecesseur de y .
- Si $x = y$, alors a est une boucle.
- Si deux arcs ont un même couple associé, on a un arc double. Plus généralement, si n arcs ont un même couple associé, on a un arc n .
- Un graphe orienté est dit strict s'il n'a ni boucles, ni arcs multiples (il peut avoir des arcs opposés).

Pour définir la notion de degré d'un graphe orienté, il faudra tenir compte du sens des flèches. On définit donc le degré extérieur qui compte le nombre de flèches sortant d'un sommet, et le degré intérieur qui compte le nombre de flèches entrant dans un sommet.

Remarque : Si on veut autoriser des flèches multiples entre deux sommets pour définir un **multigraphe orienté**, on pourra définir F non plus comme une **partie** de $S \times S$ mais comme une **famille d'éléments** de $S \times S$.

2.3.2.1.3/ Notions de stockage

Dans un graphe non orienté

La matrice d'adjacence d'un graphe G non-orienté est définie comme suit : Si $X =$

$\{x_1, \dots, x_n\}$ est l'ensemble des sommets d'un graphe G , la matrice M d'adjacence est la matrice carrée d'ordre n telle que $M = (m_{ij})$, où m_{ij} est le nombre d'arêtes d'extrémités x_i et x_j dans G . Une telle matrice est nécessairement **symétrique**.

Remarque concernant le stockage : Une telle représentation d'un graphe prend une place mémoire de l'ordre de n^2 , où n est le nombre de sommets du graphe.

Par exemple, la matrice d'adjacence du graphe non-orienté de la figure 2.26 est donnée dans le tableau 2.2 suivant

	a	b	c	s
a	1	2	1	0
b	2	0	1	1
c	1	1	0	1
s	0	0	1	0

Tableau 2.2 – Matrice d'adjacence du graphe de la figure 2.26

Dans un graphe orienté

La matrice d'adjacence $M = (m_{ij})$ d'un graphe orienté $G = (X, A)$ avec $X = \{x_1, \dots, x_n\}$, est définie en posant m_{ij} égal au nombre d'arc dont le couple associé est (x_i, x_j) . Cette matrice n'est pas nécessairement symétrique.

Par exemple, la matrice d'adjacence du graphe orienté de la figure 2.27 est donnée dans le tableau 2.3 suivant

	a	b	c	s
a	1	0	1	0
b	2	0	0	0
c	0	1	0	1
s	0	0	0	0

Tableau 2.3 – Matrice d'adjacence du graphe de la figure 2.27

Listes et tables

Définition 8 : Matrice et table d'adjacence

Un graphe $G(X, E)$ peut être représenté en associant à chaque sommet $n^o i$ la liste de ses successeurs, correspondant aux 1 de la i^{ieme} ligne de la matrice d'adjacence [108] :

- C'est une liste d'adjacence si on forme une liste de liste. Par exemple, pour le graphe de la figure 2.27, on aurait la liste d'adjacence suivante : $((a \ a \ c)(b \ a \ a)(c \ s)(s))$.
- C'est une table d'adjacence si on utilise une table de la forme du tableau 2.4 suivant.

a		b		c		s
4	5	1	2	3	6	X
a	c	a	a	b	s	X

Tableau 2.4 – Table d’adjacence du graphe de la figure 2.27

La matrice d’incidence d’un graphe $G(X, E)$ est une matrice rectangulaire formée à raison [108] (cf. tableau 2.5) :

- d’une i^{ieme} ligne pour chaque sommet i considéré,
- d’une j^{ieme} colonne pour chaque arc j considéré,
- d’éléments e_{ij} valant :
 - 1 si le sommet i est origine de l’arc j ,
 - -1 si le sommet i est l’extrémité de l’arc j ,
 - 2 si l’arc j est bouclé sur le sommet i ,
 - 0 sinon.

Concernant le stockage informatique de données, il sera parfois préférable d’utiliser une table d’incidence d’encombrement $2 \times n$, avec n le nombre d’arcs du graphe comportant (cf. tableau 2.6) :

- autant de colonnes qu’il y a d’arcs,
- Une ligne présentant l’origine de chaque arc, et une deuxième ligne présentant leurs extrémités.

sommets \ arc	arc					
	1	2	3	4	5	6
a	-1	-1	0	2	1	0
b	1	1	-1	0	0	0
c	0	0	1	0	-1	1
s	0	0	0	0	0	-1

Tableau 2.5 – Matrice d’incidence du graphe de la figure 2.27

arcs	1	2	3	4	5	6
origine	b	b	c	a	a	c
extrémité	a	a	b	a	c	s

Tableau 2.6 – Table d’incidence du graphe de la figure 2.27

2.3.2.2/ INTRODUCTION AUX CHAÎNES DE MARKOV

2.3.2.2.1/ Graphe probabiliste

Définition 9 : Graphe probabiliste

Un graphe probabiliste est un graphe orienté dont on pondère les flèches tel que [148] :

- Il y a au plus une flèche d'un sommet vers un autre.
- Tous les poids sont positifs.
- La somme des poids des flèches partant d'un sommet donné vaut 1.

On peut associer à tout graphe pondéré une matrice stochastique (c'est à dire une matrice dont tous les éléments sont des nombres compris entre 0 et 1, et dont la somme des éléments vaut 1 sur chaque ligne).

2.3.2.2.2/ Chaîne de Markov

Définition

Une **chaîne de Markov** est, de manière générale, un processus de Markov à temps discret. Mathématiquement, un processus de Markov est un processus stochastique possédant la propriété de Markov : la prédiction du futur, sachant le présent, n'est pas rendue plus précise par des éléments d'information supplémentaires concernant le passé. Ainsi, toute l'information utile pour la prédiction du futur est contenue dans l'état présent du processus [234]. On dit aussi que le processus est sans mémoire.

Un processus de Markov à temps discret est une séquence X_n de variables aléatoires à valeurs dans l'espace des états E (qu'on supposera fini). La valeur X_n est l'état du processus à l'instant n .

Ainsi, la chaîne de Markov est définie telle que, pour tout $(i_0, i_1, \dots, i_{n-1}, i, j) \in E^{n+2}$, on a la relation de l'équation (2.50) suivante

$$P(X_{n+1} = j / X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j / X_n = i) \quad (2.50)$$

Le symbole / représentant l'information "sachant que".

Reformulé plus clairement, le futur (temps $n + 1$) ne dépend que du présent (temps n) et non du passé.

Probabilités de transition

La chaîne de Markov est dite homogène si $P(X_{n+1} = j / X_n = i)$ ne dépend pas de n . Le réel $p_{ij} = P(X_{n+1} = j / X_n = i) = P(X_1 = j / X_0 = i)$ est alors appelé **probabilité de transition** de l'état i à l'état j (en une étape).

La famille de nombre $P = P_{i,j(i,j) \in E^2}$ est appelée **matrice de transition**, **noyau de transition**, ou **opérateur de transition** de la chaîne de Markov [234].

La matrice de transition est **stochastique**. On peut donc lui associer un graphe probabiliste dont les sommets sont les états et la flèche de l'état i à l'état j est pondérée par la

probabilité de transition $P_{i,j}$. la somme des termes de n'importe quelle ligne de P donne toujours 1, comme on le voit dans l'équation (2.51) suivante

$$\forall i \in E, \sum_{j \in E} p_{i,j} = 1 \quad (2.51)$$

Loi stationnaire

[234] Il peut exister une ou plusieurs mesures appelées mesures stationnaires notées $\pi = (\pi_i)_{i \in E}$, sur l'espace d'états E , telles que l'on a les équations (2.52) suivantes

$$\begin{cases} \pi = \pi P \\ \forall j \in E, \pi_j = \sum_{i \in E} \pi_i p_{i,j} \end{cases} \quad (2.52)$$

Une mesure stationnaire est une fonction propre de la transposée de la matrice de transition, associée à la valeur propre 1. Elle est appelée probabilité stationnaire, ou loi stationnaire, si elle remplit les conditions suivantes

- $\forall i \in E, \pi_i \geq 0$
- $\sum_{i \in E} \pi_i = 1$.

Si la loi initiale de la chaîne de Markov (c'est à dire la loi X_0) est une probabilité stationnaire π , alors $\forall n \geq 0$ et la loi de X_n est encore π . Plus généralement, on dira que la chaîne de Markov est un processus stationnaire si et seulement si sa loi initiale est une probabilité stationnaire.

Dans le cas des chaînes de Markov à espace d'états discret, on a les propriétés suivantes :

- Une chaîne de Markov est dite irréductible si tout état est accessible à partir de n'importe quel autre état. Dans le cadre d'un graphe plan pondéré, on peut considérer qu'il représente une chaîne de Markov irréductible si tous les sommets sont accessibles depuis n'importe quel autre sommet du graphe (directement ou indirectement).
- Si une chaîne de Markov est irréductible, récurrente positive et apériodique, alors P^k converge vers une matrice dont chaque ligne est l'unique distribution stationnaire π . En particulier, la loi μ_n de X_n converge vers π indépendamment de la loi initiale μ_0 . Dans le cas d'un espace d'état fini, cela se prouve par le théorème de Perron-Frobenius.
- Ainsi, une chaîne de Markov irréductible avec une matrice de transition primitive est appelée "chaîne apériodique". Le test de Frobenius permet de déterminer la primitivité d'une matrice : Une matrice $A \geq 0$ est primitive si et seulement si $A^m > 0 \forall m > 0$.

2.3.3/ L'ALGORITHME PAGERANK

2.3.3.1/ GÉNÉRALITÉS, DÉFINITION

Le modèle PageRank (PR) a été développé par Serguey Brin et Larry Page (fondateurs du moteur de recherche Google) au début des années 2000 [77]. Cet algorithme analyse la structure hyperlien du web pour déterminer un classement des pages web, en fonction

de leur popularité (évaluée de manière probabiliste). Le PageRank est défini comme la mesure *invariante* d'une marche faite par un surfeur aléatoire sur le graphe du web [215].

Le principe de base est d'attribuer à chaque page web un score proportionnel à la probabilité pour un utilisateur parcourant le *graphe* du web de tomber sur cette page en particulier. Ainsi, le PageRank d'une page web i est déterminé en fonction du nombre de pages qui pointent vers elles, mais aussi en fonction du PageRank de ces pages [235]. Ainsi, le score d'une page est défini comme la fréquence de visite du surfeur aléatoire sur cette page, et il est interprété comme la popularité de la page en question [215].

Plus formellement, et pour faire le lien avec la partie 2.3.2 précédente, le déplacement de l'utilisateur sur le graphe du web est considéré comme une marche aléatoire. C'est à dire que l'on considère le graphe orienté dont les sommets représentent les pages web, et les arcs les hyperliens entre ces pages. Ainsi, en supposant que l'utilisateur choisisse chaque lien indépendamment des pages précédemment visitées, on peut considérer qu'un tel graphe du web constitue un processus de Markov, tel qu'on l'a défini précédemment dans la partie 2.3.2.2. Finalement, le PageRank est la *probabilité stationnaire* d'une chaîne de Markov, c'est à dire un vecteur de Perron Frobenius de la matrice d'adjacence du web. L'irréductibilité de la chaîne de Markov considérée garantit que le vecteur stationnaire (c'est à dire le vecteur définissant les PageRanks de toutes les pages du web) existe.

Définition 10 : Matrice irréductible

Une matrice est irréductible si son graphe montre que chaque noeud est accessible depuis tous les autres noeuds. Une chaîne de Markov irréductible avec une matrice de transition *primitive* est appelée chaîne *apériodique*.

2.3.3.2/ CALCUL DU PAGERANK

2.3.3.2.1/ Exemple d'illustration Considérons le graphe orienté de la figure 2.28 suivante.

Considérons que chacune des pages a un PageRank initialisé à 0.25. En effet, dans les nouvelles versions du PageRank, on considère que le web est un graphe probabiliste. L'hypothèse d'une distribution de probabilité comprise entre 0 et 1 est donc considérée. C'est la raison pour laquelle la somme des PageRanks initiaux des pages doit valoir $1/n$ où n est le nombre de pages de la collection.

Chaque page partage son PageRank entre les pages qu'elle pointe grâce aux hyperliens. Ainsi, dans le cas du graphe de la figure 2.28, on a :

- La page B partage son PageRank équitablement entre les pages A et C.
- La page C donne son PageRank à la page A.
- La page D partage son PageRank équitablement entre les pages B, C et A.

Ainsi, si l'on devait calculer le véritable PageRank de la page A (c'est à dire la valeur correspondant *effectivement* aux liens qui pointent vers elle, et non pas sa seule valeur initiale), on aurait le calculs présenté dans l'équation (2.53) suivante

$$PR(A) = \frac{PR(B)}{2} + PR(C) + \frac{PR(D)}{3} \quad (2.53)$$

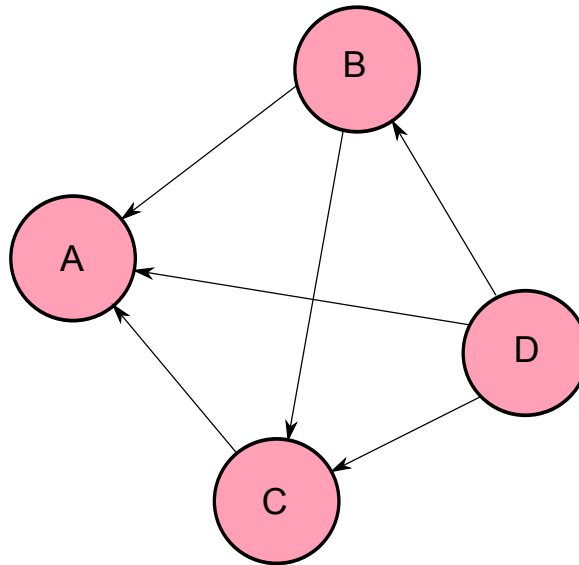


FIGURE 2.28 – Exemple de graphe orienté pour le calcul du PageRank

Et l'application numérique nous donne donc $PR(A) = \frac{11}{24}$.

2.3.3.2.2/ Algorithme simplifié On remarque ainsi, dans l'exemple précédent, que le PageRank d'une page Web est égal à la somme des PageRank des pages pointant vers cette page divisés par le nombre de liens sortant de ces pages. Plus clairement, on a l'équation (2.54) suivante, qui nous donne le calcul du PageRank d'une page P_i , contenue dans un graphe orienté dont un nombre B_{P_i} de pages pointent vers elle.

$$PR(P_i) = \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|} \quad (2.54)$$

avec :

- B_{P_i} : L'ensemble des pages qui pointent vers la page P_i ,
- $|P_j|$: Le nombre de liens sortant de la page P_j .

On comprend donc bien qu'un processus itératif sera nécessaire pour calculer le PageRank des pages considérées. En effet, le problème vient du fait qu'il est nécessaire de connaître les PageRanks des pages de la collection, pour calculer celui d'une page en particulier. Le problème est donc insoluble en l'état.

Ainsi, Brin et Page utilisent un processus itératif défini de la manière suivante :

Le PageRank des pages de la collection est initialisé à $\frac{1}{n}$.

Le processus itératif est donné dans l'équation (2.55) suivante où k est l'itération considérée.

$$PR_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{PR_k(P_j)}{|P_j|} \quad (2.55)$$

Considérons par exemple le graphe de la figure 2.29 suivante.

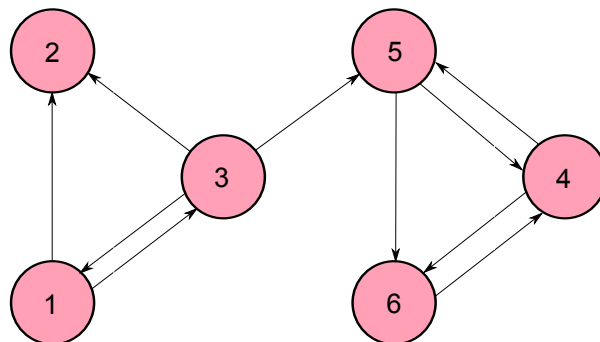


FIGURE 2.29 – Exemple de graphe orienté pour le calcul du PageRank

On initialise le PageRank des 6 sommets du graphe à la valeur $\frac{1}{n} = \frac{1}{6}$.

Si l'on effectue le calcul itératif du PageRank des 6 nœuds du graphe avec l'équation (2.55) précédente, on obtient les résultats de convergence du graphique 2.30 suivant.

2.3.3.2.3/ Prise en compte d'un facteur d'amortissement En réalité, l'algorithme de Google tient compte de la probabilité qu'un surfeur aléatoire (c'est à dire une personne cliquant de manière aléatoire sur des liens de pages web) puisse s'arrêter à tout moment de se déplacer dans le graphe du web. Considéré de manière empirique par Google, ce facteur d'amortissement d vaut 0.85, et correspond, plus précisément, à la probabilité que le surfeur aléatoire ne s'arrête **pas** de cliquer. Le facteur d'amortissement est pris en compte dans l'algorithme à la manière présentée dans l'équation (2.56) suivante

$$PR(A) = \frac{1-d}{N} + d \times \sum_{v \in B_u} \frac{PR(v)}{L(v)} \quad (2.56)$$

avec :

- d : Le facteur d'amortissement fixé empiriquement à 0.85.
- N : Le nombre de pages de la collection considérée. C'est à dire le nombre de sommets du graphe considéré.
- B_u : Les pages pointant vers la page u .
- $L(v)$ le nombre de pages sortant de la page v .

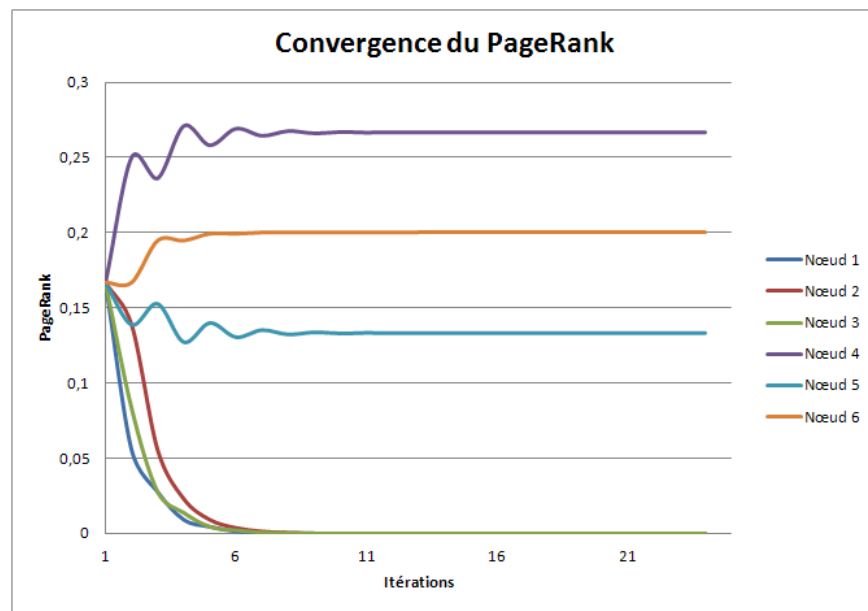


FIGURE 2.30 – Convergence du calcul de PageRank de l'équation (2.55)

2.3.3.3/ CALCUL MATRICIEL

2.3.3.3.1/ Calcul de la matrice de transition du système Les formules (2.55) et (2.54) permettent de calculer, à chaque pas de calcul, le PageRank d'une seule page de la collection. Ceci dit, le web étant constitué de milliards de pages, étant donc un graphe orienté probabiliste constitué de milliards de sommets, le calcul matriciel paraît très adapté pour optimiser le temps de calcul de PageRank de toutes ces pages.

De la même manière que nous l'avons vu dans la partie 2.3.2, la matrice de transition peut être calculée pour le graphe du web, et plus particulièrement pour la chaîne de Markov engendrée par la considération d'un surfeur aléatoire sur ce graphe [134].

On introduit alors une matrice H ($n \times n$) (appelée *The Hyperlink matrix* dans la littérature anglophone) constituée des éléments $H_{ij} = 1/|P_i|$ s'il y a un lien du nœud i au nœud j , et 0 dans le cas contraire. Cette matrice est la matrice d'adjacence probabiliste, c'est à dire la matrice de transition du graphe considéré.

Ainsi, pour le graphe représenté en figure 2.29, la matrice d'adjacence probabiliste serait la suivante

$$H = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.57)$$

Les éléments non nuls de la **ligne** i correspondent aux liens **sortant** de la page i , alors que les éléments non nuls de la **colonne** i correspondent aux liens entrant dans la page i .

Remarques :

- La matrice H est une matrice très clairsemée (appelée *a sparse matrix* dans la littérature anglophone), c'est à dire qu'elle contient une grande quantité de zéros, proportionnellement à ses dimensions.
- On remarque que les noeuds ne présentant pas de liens sortant créent des lignes composées uniquement de zéros. La matrice H n'est donc pas stochastique. La somme des termes de chaque ligne ne vaut pas 1. Cependant, les autres lignes de la matrice satisfont aux conditions nécessaires et suffisantes qui pourraient faire de la matrice H une matrice stochastique. On dit donc qu'elle est **substochastique**.

Ainsi, pour pallier à ce problème, on remplace les lignes 0^T de H par $\frac{1}{n}e^T$ avec n le nombre de pages de la collection, et e^T le vecteur ligne composé uniquement de 1. On obtient ainsi la nouvelle matrice S **stochastique** suivante

$$\bar{S} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.58)$$

Mathématiquement, le passage de la matrice H à la matrice S est donné par la relation (2.59) suivante

$$S = H + a\left(\frac{1}{n} \times e^T\right) \quad (2.59)$$

où $a_i = 1$ si i est un nœud ne présentant aucun lien sortant.

La matrice S est donc stochastique, mais cet ajustement n'est toujours pas suffisant pour assurer l'existence du vecteur stationnaire de la chaîne de Markov considérée. La matrice de transition probabiliste doit être irréductible pour assurer cette propriété. On transforme donc encore une fois la matrice de transition, pour obtenir la matrice **Google** G et on a

$$G = \alpha S + (1 - \alpha) \frac{1}{n} e e^T = \begin{bmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{bmatrix} \quad (2.60)$$

avec :

- $n = 6$
- $0 \leq \alpha \leq 1$ (Dans ce cas, on a choisi $\alpha = 0.9$ [219]).
- $e^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Dans ce modèle, α est un paramètre qui contrôle la proportion du temps pendant laquelle le surfeur aléatoire considéré suit la structure hyperlien du web. Ainsi, pour $\alpha = 0.6$, on considère que le surfeur aléatoire suit la structure hyperlien du web 60% du temps, et utilise des méthodes de téléportation (comme la saisie directe de l'URL de la page, ou la sélection de la page dans une liste de favoris par exemple) [134].

Remarques :

- La matrice $E = \frac{1}{n}ee^T$ est uniforme. Elle assure qu'en cas de téléportation aléatoire sur un des noeuds du graphe considéré, la probabilité de tomber sur chacune des pages de la collection est identique.
- G est stochastique. En effet, toute combinaison convexe de matrices stochastiques (ici, S et E) est stochastique.
- G est irréductible. En effet, chaque page est directement connectée à toutes les autres pages de la collection.
- G est primitive.
- G est une matrice **dense** (elle ne contient aucune valeur nulle). Cela implique un coût très élevé en terme de temps de calcul.

Heureusement, la matrice G peut être directement exprimée en fonction de la matrice H qui est, elle, très clairsemée, à la manière décrite dans l'équation (2.61) suivante

$$G = \alpha S + (1 - \alpha) \frac{1}{n} ee^T + \alpha (H + \frac{1}{n} ae^T) + (1 - \alpha) \frac{1}{n} ee^T \quad (2.61)$$

$$\alpha H + (\alpha a + (1 - \alpha) \frac{1}{n} e) e^T$$

Cette méthode est la fameuse **power method** présentée dans de nombreux articles et permettant le calcul rapide efficace du PageRank d'une très grande collection de pages [134].

Remarque importante : La matrice de transition probabiliste initiale H a dû être modifiée deux fois avant d'obtenir la matrice Google afin d'assurer qu'un **unique** vecteur PageRank π n'existe. Cette nouvelle matrice Google ne représente plus directement la structure hyperlien du web, mais est adaptée afin de pouvoir pallier aux problèmes mathématiques que nous avons présentés ci-avant.

2.3.3.3.2/ Calcul du vecteur PageRank Le vecteur ligne PageRank peut être calculé en résolvant le problèmes aux valeurs propres suivant

$$\pi^T G = \pi^T \quad (2.62)$$

ou en résolvant le système linéaire suivant

$$\pi^T (I - G) = \mathbf{0}^T \quad (2.63)$$

Il existe de nombreuses méthodes pour résoudre ce type de problème. Puisque ce n'est pas notre sujet ici, nous renvoyons les lecteurs aux articles et livres de Langville et al. qui présentent très bien toutes les méthodes utilisées et développées pour calculer le PageRank des pages du web, et stocker efficacement les données [134, 219].

2.3.3.4/ UTILISATION ORIGINALE ET INTÉRESSANTE DE PAGE RANK DANS LA LITTÉRATURE

2.3.3.4.1/ Généralités Des articles particulièrement intéressants concernant l'évaluation probabiliste des métastases dans le cas d'un cancer généralisé ont attiré notre

attention^{4 5}. Ces articles vulgarisent deux articles proposés par P. K. Newton et al. [222, 232], qui utilisent le concept de l'algorithme PageRank pour déterminer les probabilités qu'ont les métastases de se propager d'un organe à l'autre, lors de la généralisation d'un cancer. Comme le dit d'ailleurs très bien P.K. Newton dans son interview au site Slate.fr :

"Basically, we are doing the inverse of what Google does. They know the transition probabilities and compute the steady-state, we know the steady-state and compute the transition probabilities."

Ainsi, Google connaît la matrice de transition G et essaie de calculer le vecteur PageRank π , c'est à dire le vecteur stationnaire de la chaîne de Markov considérée. À l'inverse, Newton et al. connaissent les probabilités stationnaires qu'ont les métastases de se propager dans différents organes, et cherchent de quelle manière ces métastases passent d'un organe à l'autre. Plus spécifiquement, Newton et al. connaissent le vecteur stationnaire de la chaîne de Markov qu'ils considèrent, et cherchent à déterminer [la matrice de transition](#) de la chaîne de Markov considérée. Comme il l'a d'ailleurs été dit sur slate.fr :

"In other words, Google knows where you could go, and uses a mathematical system called Markov chains to determine how likely you are to go there. The researchers knew where cancer did go, and used similar equations to investigate how it goes there".

2.4/ METAMATÉRIAUX ET NOTIONS D'HOMOGENÉISATION

2.4.1/ LES METAMATÉRIAUX

Du grec "meta" signifiant "un niveau supérieur", les metamatériaux désignent une nouvelle classe de matériaux [composites périodiques](#) dont les propriétés physiques homogénéisées peuvent être considérées comme exceptionnelles. De fait, deux étapes sont nécessaires à la conception d'un nouveau metamatériau :

- L'homogénéisation. Afin de pouvoir considérer un multi-matériau, ou composite, comme un metamatériau, il est nécessaire d'homogénéiser ses propriétés physiques. Ainsi, le metamatériau, bien qu'hétérogène, peut être considéré comme un matériau homogène dont les propriétés physiques sont connues à l'échelle microscopique. C'est le cas par exemple des mousses auxétiques, dont les propriétés à l'échelle locale ne sont pas exceptionnelles, alors que la mousse considérée dans sa globalité présente un coefficient de Poisson négatif.
- L'optimisation. Très souvent topologique[192, 217], elle définira la répartition de matière dans le composite afin qu'il présente les propriétés physiques choisies.

Cette partie a pour but de fournir une étube bibliographique de la recherche concernant les metamatériaux dans la littérature.

Dans le domaine de la mécanique, la recherche concernant les metamatériaux s'est

4. "Fighting Cancer With the Google PageRank Algorithm. Sort of." *J. Bittel*. slate.com dernière connexion le 6 juillet 2016.

5. "How researchers are fighting lung cancer using PageRank". D. Harris. gigaom.com dernière connexion le 5 février 2014.

concentrée ces dernières décennies sur l'obtention de metamatériaux - ou de structures - à coefficient de Poisson négatif, appelés aussi **auxétiques**. Historiquement, on situe le début des recherches concernant les structures auxétiques à la fin des années 80, avec notamment les travaux de Lakes [28, 41] et Evans [39, 54]. Leurs travaux montrent notamment de quelle manière une structure en nid d'abeille réentrante présente un coefficient de Poisson négatif, comme on le voit dans la figure 2.31 suivante [39].

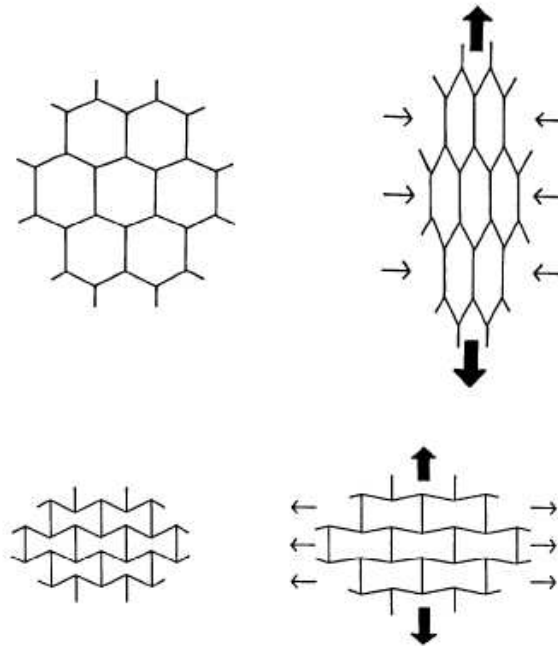


FIGURE 2.31 – Différence de fonctionnement d'une structure réentrante, par rapport à une structure en nid d'abeille classique (haut) Déformation d'une structure à nid d'abeille (bas) Déformation d'une structure réentrante

Notons qu'au tout départ, les travaux concernant un coefficient de Poisson négatif à structure réentrante définissaient en fait des structures, plutôt que des matériaux, puisqu'aucun processus d'homogénéisation n'était utilisé. Les structures réentrantes ont donné lieu à de nombreuses recherches qui ont permis de déterminer leurs propriétés mécaniques qui peuvent être linéaires [57] ou non linéaires [42] ; isotropes ou anisotropes [189]. Leurs formes géométriques ont aussi évolué, passant de la simple forme plane [25, 144], aux modèles en trois dimensions [54]. Des études concernant leurs autres propriétés mécaniques ont aussi été réalisées, montrant notamment que la dureté des matériaux à coefficient de Poisson négatif est augmentée, comparée à celle du même matériau (UHMWPE pour *Ultra High Molecular Weight Polyethylene*) non auxétique [51]. Azoti et al. montrent aussi que le facteur d'amortissement d'un matériau contenant des inclusions de matériau auxétique est amélioré [231]. Notons aussi l'existence de structures chirales, c'est à dire de structures présentant des symétries en miroir, à coefficient de Poisson négatif [74, 180]. Ces structures - tant les structures réentrantes que les structures chirales - ont l'avantage de pouvoir être répétées à l'infini dans au moins deux directions de l'espace. Depuis les années 2000, elles sont souvent étudiées en utilisant des méthodes d'homogénéisation.

Par la suite, dès le début des années 2000, Grima et al. [94] proposent des structures

à formes géométriques tournantes, conférant aussi à la structure globale un coefficient de Poisson négatif. C'est en se déplaçant que les formes géométriques qui constituent la structure lui confèrent cette propriété, comme on le voit dans la figure 2.32 suivante. Les géométries utilisées se diversifient, et on trouve dans la littérature des carrés rotatifs [94, 178], des tétraèdres rotatifs [179], des rectangles rotatifs [199], mais aussi des triangles [198]. La structure triangulaire amenée par Grima et al. présente notamment l'avantage d'être aussi efficace en traction qu'en compression, comme on le voit dans la figure 2.33 suivante

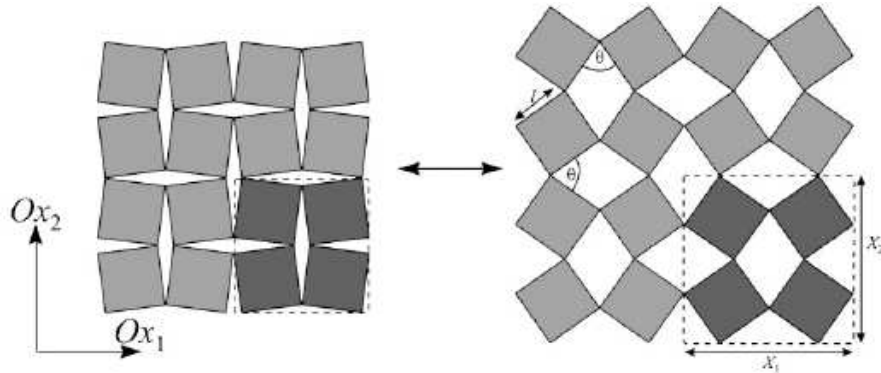


FIGURE 2.32 – Géométrie d'une structure auxétique à carrés tournants.

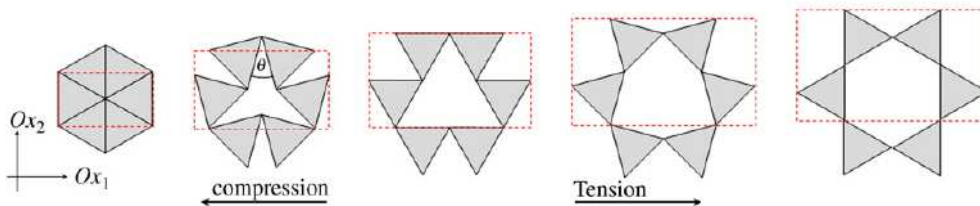


FIGURE 2.33 – Géométrie d'une structure auxétique à triangles rotatifs.

Très vite, ces structures auxétiques ont commencé à prendre la forme de matériaux, notamment avec l'avènement et la fabrication de mousses auxétiques [41, 98, 157, 204]. Bien évidemment, les applications ne sont pas les mêmes [221]. On développe notamment des structures perforées [198] plutôt que des structures réentrantes, qui permettent des processus de fabrication simplifiés. Avec le développement de ces mousses se développent évidemment les méthodes de fabrication. Le processus consiste à utiliser une mousse classique à structure en nid d'abeille, et à la comprimer dans les trois directions de l'espace. En chauffant cette mousse comprimée, la structure initialement en nid d'abeille se déforme pour devenir une structure réentrante, comme on le voit dans la figure 2.34 suivante [32].

Finalement, les méthodes d'homogénéisation s'imposent dans le domaine des metamatériaux, considérant qu'ils sont périodiques et que donc leur étude pourrait se limiter à l'étude d'une seule cellule de périodicité, comme nous le verrons dans la partie 2.4.2. Ici, l'utilisation d'un Volume Elementaire Représentatif simplifie les calculs, et permet pour la première fois de considérer que les structures à coefficient de Poisson négatif deviennent véritablement des metamatériaux [126] puisque l'on extrapole leurs propriétés globales homogénéisées à partir de celles d'une seule cellule de périodicité. La figure

2.35 présente différentes formes de structures auxétiques et leur cellules de périodicité respectives [205].

Comme nous le détaillerons par la suite, la méthode la plus employée pour étudier les metamatériaux n'est pas l'homogénéisation asymptotique, mais la méthode des champs moyens [205, 213]. Moyennant des hypothèses concernant les dimensions du Volume Élémentaire Représentatif (cf. partie 2.4.2), Dos Reis et al. utilisent tout de même une méthode d'homogénéisation asymptotique discrète [195, 214]. Cependant, Czech montre dans sa thèse [211] que le rapport d'échelle entre la cellule de périodicité et le metamatériau n'est pas suffisamment grand pour que les méthodes d'homogénéisation asymptotique puissent être efficaces. Il défend alors l'idée de l'utilisation de la méthode des champs moyens que l'on retrouve grandement dans la littérature, comme nous le verrons dans la partie 2.4.2.2.

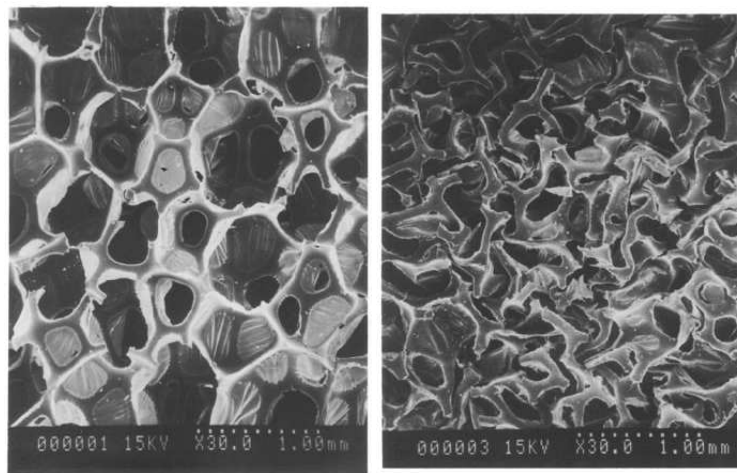


FIGURE 2.34 – Fabrication d'une mousse auxétique à structure réentrante : (Gauche) Microstructure initiale de la mousse en structure nid d'abeille. (Droite) La même mousse après le procédé de compression/chauffage.

2.4.2/ HOMOGÉNÉISATION

2.4.2.1/ PRINCIPES DE CHANGEMENT D'ÉCHELLE : PASSAGE DU MILIEU HÉTÉROGÈNE AU MILIEU HOMOGÈNE

Afin de créer un metamatériau, il est nécessaire de connaître ses propriétés mécaniques globales, c'est à dire homogénéisées. En d'autres termes, bien que sa structure soit hétérogène, composée de plusieurs matériaux homogènes, comme c'est souvent le cas lorsque l'on utilise l'optimisation topologique, ses propriétés - et notamment les termes de son tenseur d'élasticité - doivent pouvoir être considérées comme homogènes. C'est à dire qu'il faut définir un Volume Élémentaire Représentatif (VER) tel qu'il aurait une réponse globale identique à la structure considérée, dès lors que la structure et le VER sont sollicités par un chargement identique. Ainsi, depuis de nombreuses années, de nombreuses recherches ont été entreprises afin de prendre en compte l'hétérogénéité de la matière à l'échelle microscopique dans la modélisation de son comportement macroscopique [194]. Ce type de modélisation par changement d'échelle consiste alors à remplacer un milieu hétérogène par son Milieu Homogène Equivalent (MHE). Les pro-

propriétés mécaniques du MHE - c'est à dire les termes de son tenseur d'élasticité - sont calculées en tenant compte des constituants du milieu hétérogène - on a souvent à faire à un matériau composite - et des paramètres relatifs à sa morphologie - c'est à dire sa topologie. Pour trouver les propriétés homogénéisées, on remplace donc l'étude du corps entier par une portion de celui-ci suffisamment grande pour inclure à la fois les détails de la microstructure, mais aussi pour être considérée comme représentative du matériau à l'échelle globale macroscopique [169]. Cette méthode consiste ainsi à déterminer, grâce à une méthode que nous développerons plus loin, des relations entre les grandeurs physiques des deux échelles microscopique et macroscopique. En pratique, ces relations sont déterminées sur un Volume Élémentaire Représentatif (VER) du milieu hétérogène. Ce VER doit non seulement être suffisamment grand devant la taille des hétérogénéités de la structure, mais aussi suffisamment petit devant la dimension caractéristique de la structure étudiée. Les propriétés homogènes du milieu hétérogène considéré étant souvent définies numériquement par un calcul éléments finis, il est aussi nécessaire que le VER ne soit pas trop grand, pour limiter les temps de calcul. En pratique, lorsque l'on travaille avec un matériau dont les hétérogénéités sont aléatoirement distribuées dans le matériau, on effectue une analyse statistique afin de déterminer la taille minimale du VER [169]. A l'inverse, lorsque l'on travaille avec une structure périodique, le VER n'est réduit qu'à l'étude d'une seule Cellule de Périodicité (CP). La propriété la plus importante du VER, c'est qu'il doit être justement suffisamment représentatif de la structure considérée comme homogène pour que l'augmentation de ce volume ne change pas la valeur des champs mécaniques (contraintes, déformations) calculés [169].

Cette méthode de changement d'échelle est décomposée en trois étapes distinctes (voir figure 2.36 suivante), définies par Zaoui en 2000 [96, 97] :

- La phase de **représentation**. Il s'agit de représenter numériquement le milieu hétérogène. C'est à dire qu'il faut identifier le nombre et les caractéristiques des différentes phases qui composent notre VER. Bien sûr, ces paramètres devront être complétés par des données géométriques, notamment les dimensions des inclusions ou des fibres. Cette étape est fondamentale, elle consiste à choisir le VER, c'est à dire le plus petit volume censé représenter la globalité de la structure ou du matériau considéré.
- La **localisation**. Cette étape consiste à déterminer les champs mécaniques dans la structure induits par un champ moyen [194] dans le VER. Les relations de passage de l'échelle microscopique à l'échelle macroscopique sont ainsi déterminées. Cette étape est fondamentale et bien souvent délicate, car elle consiste à réaliser l'analyse mécanique de la structure, alors même que cette dernière est incomplète, puisque représentée uniquement par son VER. Le VER pouvant être considéré comme un point matériel de la structure, on considèrera ici que les sollicitations auxquelles il est soumis découlent des contraintes et déformations s'établissant au point matériel considéré. Notons qu'entre ces grandeurs et le champs de variables locales intrinsèques au VER à déterminer ne peuvent s'établir que des relations de moyennes [97].

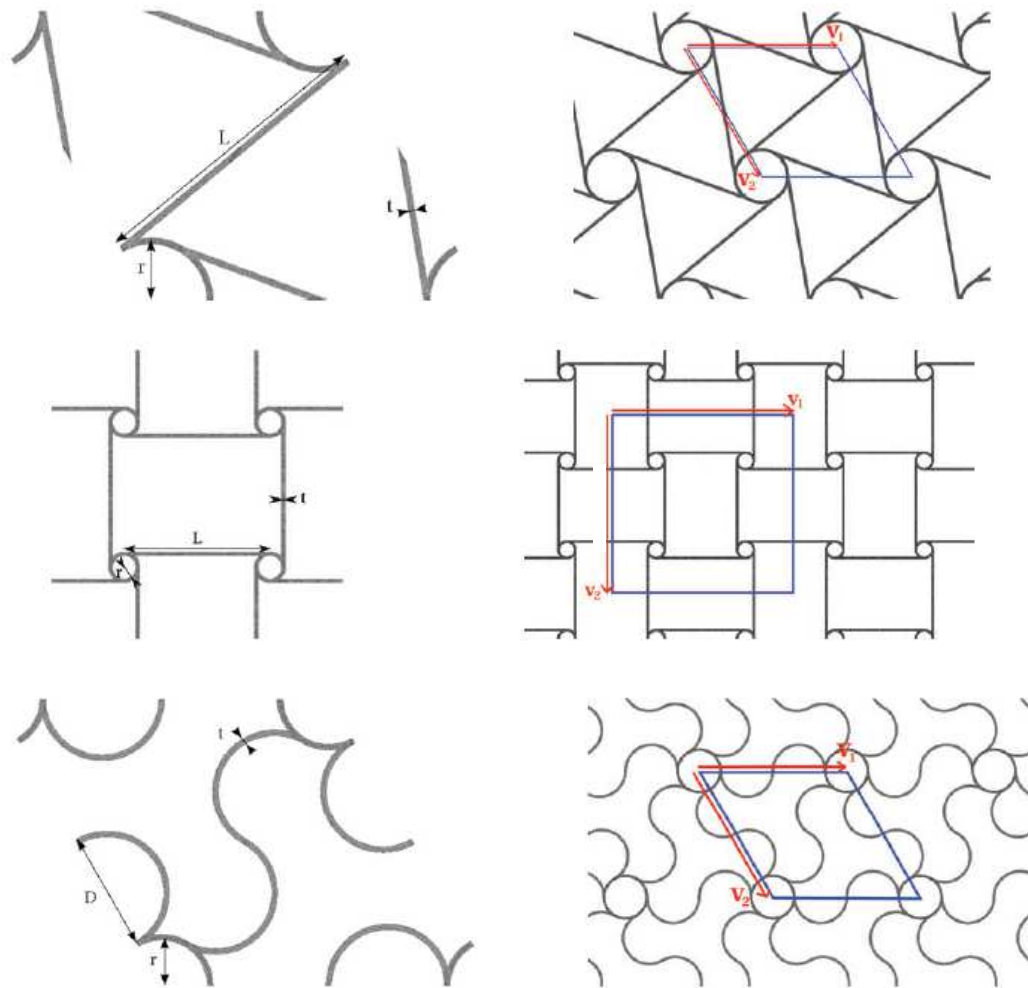


FIGURE 2.35 – Différents types de Volumes Elémentaires Représentatifs, en fonction de la forme du matériau auxétique considéré. (Gauche) Les cellules élémentaires. (Droite) Les treillis assemblés des VER

- L'homogénéisation. Cette étape vise à l'expression des caractéristiques mécaniques du milieu homogène équivalent, en utilisant notamment les relations de moyennes définies à l'étape précédente.

Le VER est défini comme un élément de la structure, ou du matériau, considéré comme représentatif de son comportement mécanique global. On définit ainsi dans la littérature les différentes longueurs caractéristiques du problème [96] :

- d : la dimension caractéristique des hétérogénéités de la structure. Lorsque l'on souhaite uniquement faire un calcul d'homogénéisation, cette dimension est connue. Lorsque l'on souhaite faire un calcul d'optimisation topologique, elle ne l'est pas. On considère alors qu'elle est, au maximum, égale à la dimension l définie ci après.
- l : la dimension du VER.
- L : la dimension caractéristique de la structure considérée. En effet, le VER doit pouvoir être considéré comme un point matériel de la structure. C'est ce que l'on appelle le principe Micro-Meso-Macro [23, 169]. Lorsque l'on souhaite étudier un matériau et non une structure, cette dimension peut être considérée comme infinie.

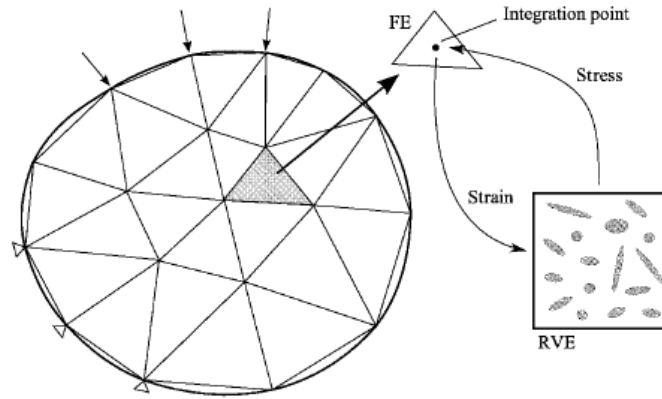


FIGURE 2.36 – Méthode de changement d'échelle : définition du VER [183]

Ainsi, seules les dimensions caractéristiques des hétérogénéités et leur densité dans la matière devront être prises en compte lors de la définition du VER. Ces dimensions caractéristiques doivent, pour valider le VER, vérifier les conditions suivantes [194] :

- $l \ll L$: cette condition est nécessaire afin d'assurer que le VER peut être considéré comme un point infinitésimal de la structure considérée,
- $d \ll l$: cette condition est nécessaire afin d'assurer que le VER est bien représentatif de la structure considérée, c'est à dire qu'il tient bien compte de la taille et de la densité des hétérogénéités au sein de la structure.

2.4.2.2/ MÉTHODE DES CHAMPS MOYENS

Dans le travail de Czech [211], on note que les méthodes d'homogénéisation asymptotique, que nous ne développerons pas ici, peuvent ne pas être très adaptées à l'étude des metamatériaux, lorsque les dimensions du VER sont trop grandes. En effet, les contraintes dimensionnelles nécessaires à leur bon fonctionnement, notamment le rapport qui doit être assez grand entre la dimension caractéristique de la structure et la dimension du VER (le VER est ici la cellule de périodicité, puisque les metamatériaux mécaniques étudiés sont périodiques), ne sont pas respectées. Ainsi, Czech et al. proposent d'utiliser plutôt la méthode des champs moyens. Comme noté dans le travail de Kurukuri [190], tous les matériaux, même homogènes, peuvent être considérés comme hétérogènes à une échelle suffisamment petite. Ainsi, pour pallier à ce problème, la méthode des champs moyens peut être utilisée afin de déterminer les propriétés homogènes du matériau, considérant ses caractéristiques hétérogènes microscopiques. L'hypothèse de la continuité de la matière implique la moyenne statistique des propriétés locales. Pour cela, on considère que les surfaces du VER sont soumises à des champs de déplacements et/ou de force, alors qu'il existe, dans le VER, des champs de contraintes σ_{ij} et de déplacements ϵ_{ij} . Ainsi, si l'on connaît les champs σ_{ij} et ϵ_{ij} , les champs de contraintes et de déplacements moyennés homogénéisés sont donnés respectivement par

$$\bar{\sigma}_{ij} = \frac{1}{V} \int_V \sigma_{ij} dV \quad (2.64)$$

$$\bar{\epsilon}_{ij} = \frac{1}{V} \int_V \epsilon_{ij} dV \quad (2.65)$$

où V représente le volume du VER.

Ces valeurs moyennées sont ensuite considérées comme les champs de contraintes et de déplacements **effectifs** du VER homogénéisé, et les relations entre $\bar{\sigma}_{ij}$ et $\bar{\epsilon}_{ij}$ nous donnent les lois constitutives du matériau homogène. Si l'on considère le matériau comme homogène, on a donc

$$\bar{\sigma}_{ij} = C_{ijkl} \bar{\epsilon}_{ij} \quad (2.66)$$

$$\bar{\epsilon}_{ij} = S_{ijkl} \bar{\sigma}_{ij} \quad (2.67)$$

où C_{ijkl} est le tenseur d'élasticité effectif du matériau considéré, et S_{ijkl} est son tenseur de compliance effectif. Ainsi, dans le cas de matériaux anisotropes, on a 21 coefficients différents à calculer afin de déterminer complètement ses propriétés élastiques.

2.4.2.2.1/ Validation de la méthode : Vérification de l'énergie de déformation

Basée sur la méthode des champs moyens précédentes, la méthode de l'énergie de déformation est efficace et équivalente à une méthode d'homogénéisation asymptotique pour prédire les propriétés effectives d'un matériau périodique et/ou composite [87, 176]. Cette méthode se base sur l'équivalence des énergies de déformation dans les structures hétérogène et homogène. Dans le travail de Kurukuri [190], on montre que ces deux énergies sont effectivement égales.

L'énergie totale de déformation U dans le volume V de la structure homogène est donnée par

$$U = \frac{1}{2} \bar{\sigma}_{ij} \bar{\epsilon}_{ij} V \quad (2.68)$$

alors que l'énergie de déformation U' dans le VER considéré comme hétérogène est donnée par

$$\begin{aligned} U' &= \frac{1}{2} \int_V \sigma_{ij} \epsilon_{ij} dV \\ &= \frac{1}{2} \int_V \sigma_{ij} (\epsilon_{ij} - \bar{\epsilon}_{ij} + \bar{\epsilon}_{ij}) dV \\ &= \frac{1}{2} \int_V \sigma_{ij} (\epsilon_{ij} - \bar{\epsilon}_{ij}) dV + \frac{1}{2} \bar{\epsilon}_{ij} \int_V \sigma_{ij} dV \\ &= \frac{1}{2} \int_V \sigma_{ij} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial \bar{u}_i}{\partial x_j} \right) dV + \frac{1}{2} \bar{\sigma}_{ij} \bar{\epsilon}_{ij} V \end{aligned} \quad (2.69)$$

Si l'on soustrait l'équation (2.68) à la dernière équation de l'équation (2.69), on obtient

$$U - U' = \frac{1}{2} \int_V \sigma_{ij} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial \bar{u}_i}{\partial x_j} \right) dV \quad (2.70)$$

En utilisant l'équation d'équilibre suivante

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0 \quad (2.71)$$

on peut écrire l'équation (2.70) de la forme suivante

$$U' - U = \frac{1}{2} \int_V \frac{\partial}{\partial x_j} (\sigma_{ij}(u_i - \bar{u}_i)) dV \quad (2.72)$$

L'intégrale sur le volume peut alors être transformée en intégrale de surface grâce au théorème de Gauss, et on a

$$U' - U = \frac{1}{2} \int_S \sigma_{ij}(u_i - \bar{u}_i) n_j dS \quad (2.73)$$

où S représente la surface du RVE, et n sa normale extérieure. Sur la surface S , on a

$$u_i = \bar{u}_i \quad (2.74)$$

Et on obtient finalement

$$U' - U = 0 \quad (2.75)$$

On a ainsi montré que les contraintes et déformations moyennes $\bar{\sigma}_{ij}$ et $\bar{\epsilon}_{ij}$ vérifient l'équivalence de l'énergie de déformation entre le matériau considéré comme homogène et le VER de matériaux hétérogènes initial. Les quantités moyennées seront utilisées par la suite pour déterminer les propriétés effectives du matériau considéré, comme nous l'avons vu dans l'équation (2.66).

2.4.2.2.2/ Modélisation numérique et calcul des paramètres effectifs du tenseur d'élasticité

Afin de déterminer les propriétés du tenseur d'élasticité homogène, il est nécessaire d'effectuer des calculs éléments finis définissant les différentes sollicitations (traction/compression, cisaillement) dans les trois directions de l'espace. En deux dimensions, on n'effectuera que trois calculs, deux en traction et un seul en cisaillement [211]. Considérons dans ce cas simple un matériau de type orthotrope en trois dimensions dans un état de contraintes planes. On a donc le système matriciel suivant

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & 0 & 0 & 0 \\ C_{2211} & C_{2222} & C_{2233} & 0 & 0 & 0 \\ C_{3311} & C_{3322} & C_{3333} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{2323} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{1313} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{1212} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{bmatrix} \quad (2.76)$$

Premier état de sollicitation

On suppose ici que l'on impose une déformation unitaire de la forme suivante

$$\bar{\epsilon} = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (2.77)$$

Le vecteur des contraintes peut donc être réécrit de la manière suivante

$$\bar{\sigma} = [C_{1111} \ C_{2211} \ C_{3311} \ 0 \ 0 \ 0]^T \quad (2.78)$$

En remplaçant les deux équations précédentes dans l'équation (2.68), on obtient

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{V}{2} C_{1111} \\ \Rightarrow C_{1111} &= \frac{2U}{V} \end{aligned} \quad (2.79)$$

Second état de sollicitation

On suppose ici que l'on applique un déplacement unitaire de la forme suivante

$$\bar{\epsilon} = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad (2.80)$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [C_{1122} \ C_{2222} \ C_{3322} \ 0 \ 0 \ 0]^T \quad (2.81)$$

En remplaçant ces deux expressions dans l'équation 2.68, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{V}{2} C_{2222} \\ \Rightarrow C_{2222} &= \frac{2U}{V} \end{aligned} \quad (2.82)$$

Les autres états de sollicitations utilisant la même méthode de calcul sont détaillés à l'Annexe A.

Ainsi, ces neuf états de contraintes différents permettent de déterminer complètement la matrice de rigidité du matériau orthotrope 3D considéré.

Hollister et Kikuchi [43], suivis par Drago et Pindera [155] ont été les premiers à proposer une comparaison entre la méthode d'homogénéisation asymptotique et la méthode des champs moyens. Grâce à des analyses par éléments finis, les auteurs ont étudié quantitativement les convergences de ces deux méthodes en comparant l'estimation des densités d'énergie de déformation obtenues avec chacune de ces deux méthodes. Ils concluent que les méthodes d'homogénéisation asymptotique sont plus efficaces que les autres, notamment en raison du manque de connaissances disponibles dans la littérature à ce moment là, qui ne permettent donc pas de vérifier toutes les hypothèses nécessaires à ce type d'analyse. Plus tard, Pecullan et al. [88] étudient les différences entre les conditions aux limites de traction et les conditions aux limites de déplacements. Ils concluent que les conditions aux limites de traction donnent de meilleurs résultats lorsque le ratio entre la quantité de matière dans le VER et le volume total du VER est petit. A contrario, ils concluent que les conditions aux limites de déplacement sont plus efficaces lorsque ce ratio est grand. De nombreuses études numériques ont été proposées dans la littérature afin d'étudier l'effet des types de conditions aux limites sur la prédiction des propriétés

homogènes [55, 59]. Notons que dans le cas d'une optimisation topologique, ces hypothèses ne peuvent pas être vérifiées a priori. Ainsi, on optera plus aisément pour des conditions aux limites de déplacement, aux vues des recherches effectuées dans la littérature [211].

2.4.2.3/ DE L'IMPORTANCE DES CONDITIONS AUX LIMITES

L'application de conditions aux limites particulières est nécessaire afin d'assurer l'équivalence de comportement effectif entre les définitions mécaniques et énergétiques [70, 194]. Il est notamment nécessaire de vérifier la condition de Hill définie par l'équation (2.83) suivante qui définit l'égalité du travail macroscopique et de la moyenne du travail microscopique [97, 194].

$$\frac{1}{2} \langle \bar{\epsilon} : \bar{\sigma} \rangle_V = \frac{1}{2} \overline{E \Sigma} \quad (2.83)$$

Plusieurs types de conditions aux limites peuvent être appliqués pour vérifier cette condition de Hill. On notera notamment les conditions aux limites de déformation [70] ou de contrainte homogène sur le contour, et les conditions aux limites périodiques.

2.4.2.3.1/ Conditions aux limites de déformation homogène sur le contour

Le calcul éléments finis permet de déterminer les champs de contraintes et de déformations dans le VER hétérogène. Grâce aux équations (2.64) et (2.65), on peut obtenir les quantités moyennées sur le volume, qui approximent les contraintes et déformations homogénéisées. Les conditions aux limites de déformation homogène sur le contour consistent à appliquer un vecteur déplacement $\underline{u}$ sur le contour du VER, comme on le voit dans l'équation suivante [194]

$$\underline{u}(\underline{x}) = \underline{\epsilon}^0 \underline{x} \quad \forall \underline{x} \in \partial V \quad (2.84)$$

où $\underline{\epsilon}^0$ est homogène.

La déformation moyenne peut ainsi être calculée, grâce à ce type de condition aux limites, en utilisant le théorème de Gauss. L'équation (2.65) devient alors [30, 38, 70]

$$\bar{\epsilon}_{ij} = \frac{1}{V} \int_V \epsilon_{ij} dV = \frac{1}{2V} \int_S (u_i n_j + u_j n_i) dS \quad (2.85)$$

où V est le volume du VER, S sa surface, u_i est la $i^{\text{ème}}$ composante du vecteur déplacement $\underline{u}$, et n_j la $j^{\text{ème}}$ composante du vecteur normal à la surface S . Notons que l'équation (2.85) est valide uniquement lorsque ϵ_{ij} est continue dans le volume V . Lorsque les déformations ne sont pas continues dans le domaine, il faut soit considérer un autre type de conditions aux limites, soit séparer l'intégrale de volume en plusieurs parties comme on le voit dans l'équation (2.86) suivante [70]

$$\bar{\epsilon}_{ij} = \frac{1}{V} \times \sum_{i=1}^n \left(\int_{V_i} \epsilon_{ij} dV \right) \quad (2.86)$$

Cette équation considère la possibilité où il y a n phases de matériaux différents dans le volume V .

2.4.2.3.2/ Condition aux limites de contrainte homogène sur le contour

Ce type de condition aux limites consiste à appliquer un vecteur contrainte $\underline{t}(\underline{x})$ sur le contour du volume V du VER tel que [194]

$$\underline{t}(\underline{x}) = \underline{\sigma}(\underline{x})\underline{n} = \underline{\Sigma}^0 \underline{n} \quad \forall \underline{x} \in \partial V \quad (2.87)$$

où $\underline{\Sigma}^0$ est une contrainte homogène. Il en résulte, par utilisation du théorème de Gauss, pour le champ de contraintes, la relation de moyenne suivante [97]

$$\overline{\sigma}_{ij} = \frac{1}{V} \int_V \sigma_{ij} dV = \underline{\Sigma}^0 \quad (2.88)$$

2.4.2.3.3/ Conditions aux limites périodiques

Les conditions aux limites périodiques, très largement utilisées dans la littérature [70, 97, 124, 129, 169, 229], consistent à imposer un déplacement tel que [169] les noeuds supérieurs (respectivement à gauche) du maillage ont le même déplacement final que les noeuds inférieurs (respectivement à droite) du maillage, comme on le voit dans la figure 2.37 suivante.

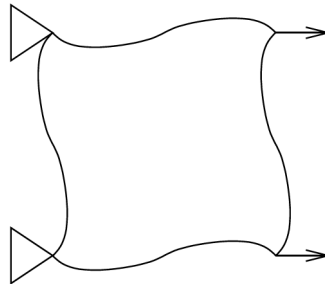


FIGURE 2.37 – Conditions aux limites de périodicité avec application d'un déplacement

Si l'on considère $\underline{\Sigma}^0$ une déformation macroscopique, les conditions aux limites périodiques consistent donc à appliquer un déplacement $\underline{u}$ sur les contours du volume V tel que [30]

$$\underline{u}(\underline{x}) = \underline{\Sigma}^0 \underline{x} + \underline{u}^* \quad (2.89)$$

où $\underline{u}^*$ est un champ de déplacement périodique, dont la période est égale à la longueur du côté de notre VER. Cette partie est inconnue et dépend du chargement [129]. $\underline{\Sigma}$ peut donc ici être considéré comme la moyenne volumique des déformations du VER telle que définie dans l'équation (2.65).

2.4.2.3.4/ Influence des conditions aux limites

Dans la littérature, on remarque des différences de résultats suivant les types de conditions aux limites appliqués au VER, bien que celui-ci soit supposément représentatif

de la structure à l'échelle macroscopique. Czech montre dans son travail que ces différences s'atténuent dès lors que l'on augmente la taille du VER [211], en particulier lorsque celui-ci représente une ou plusieurs cellules de périodicité d'un matériau périodique. Il a aussi été démontré que les conditions aux limites périodiques donnent une meilleure approximation des propriétés homogénéisées par rapport aux conditions homogènes [55, 124, 194, 211].

2.4.2.3.5/ Calcul de changement d'échelles sur Castem

Sur Castem, logiciel français de calcul EF développé par le CEA, les calculs de changement d'échelle sont déjà implémentés. En particulier, la procédure @KEFF permet de déterminer le tenseur d'élasticité apparent à partir d'une microstructure, de ses paramètres matériaux et d'un choix de conditions aux limites⁶. Cette évaluation est réalisée à partir des 6 calculs élémentaires définis dans la partie 2.4.2.2.2. Différents types de conditions aux limites peuvent être choisis :

- @CLPC : utilisation de conditions aux limites périodiques et d'un chargement en contrainte moyenne imposée.
- @CLPD : utilisation de conditions aux limites périodiques et d'un chargement en déformation moyenne imposée.
- @CLDH : utilisation de conditions aux limites en déformation homogène au contour et d'un chargement en déformation moyenne imposée.
- @CLDHC : utilisation de conditions aux limites en déformation homogène au contour et d'un chargement en contrainte moyenne imposée.
- @CLCH : utilisation de conditions aux limites en contrainte homogène au contour et d'un chargement en contrainte moyenne imposée.
- @CLMI1C : utilisation de conditions aux limites mixtes (le déplacement normal et les contraintes tangentielles respectent des conditions uniformes) sur le contour et d'un chargement en contrainte moyenne imposée.
- @CLMI2C : utilisation de conditions aux limites mixtes (le déplacement tangentiel et la contrainte normale respectent des conditions uniformes) sur le contour et d'un chargement en contrainte moyenne imposée.

2.5/ CONCLUSIONS DE L'ÉTUDE BIBLIOGRAPHIQUE, PROBLÉMATIQUES ET PLAN

Le développement des méthodes metaheuristiques s'est intensifié ces dernières années, avec notamment le développement de méthodes hybrides. Ces améliorations ont permis notamment - en tenant compte de l'évolution des capacités de calcul - d'augmenter considérablement les possibilités offertes par les méthodes d'optimisation metaheuristiques, tant en terme de vitesse de convergence qu'en terme de complexité des fonctions traitées.

Cette étude bibliographique a présenté la littérature existante et les bases scientifiques nécessaires à l'élaboration de ce travail de thèse. Le chapitre 3 présentera le développement d'une nouvelle méthode metaheuristique intelligente, Inverse-PageRank-PSO (I-PR-PSO), basée conjointement sur l'algorithme d'optimisation par essaim particulaire présenté en partie 2.2.3, et sur l'algorithme PageRank développé par les concepteurs du

6. www-cast3m.cea.fr. Dernière connexion le 28 avril 2016

moteur de recherche Google, présenté précédemment en partie 2.3.3. Cette méthode sera validée sur des fonctions mathématiques classiques, en optimisation paramétrique et topologique, comme nous l'avons vu dans la partie 2.1.3. Concernant le chapitre 3, les questions scientifiques posées sont les suivantes :

- Comment faire en sorte que la topologie des particules PSO se comporte de manière intelligente ? Cette supposée intelligence peut-elle avoir une influence sur la convergence des calculs ?
- Les capacités d'un tel algorithme sont-elles variables en fonction du type de fonction mathématique sur laquelle il est testé ?
- Comment adapter un tel algorithme à l'optimisation paramétrique contrainte, appliquée à la mécanique ?

Le chapitre 4 montrera comment adapter I-PR-PSO à l'optimisation topologique binaire. Doit-on adapter l'algorithme pour une telle application ? Comment gérer les potentiels damiers dans la topologie ? En mécanique, comment s'assurer que l'on minimise à la fois le volume et la compliance de la structure ?

Le chapitre 5 présentera une première application de PSO à l'optimisation topologique des metamatériaux. Cette application concerne un metamatériau électromagnétique à perméabilité négative, appelé aussi cape d'invisibilité électromagnétique fréquentielle. Concernant cette partie, les questions scientifiques posées sont les suivantes :

- PSO doit-il être adapté à l'optimisation topologique, en particulier sur les problèmes électromagnétiques ?
- Quels paramètres de PSO doivent-ils être choisis ?
- Quelles sont les conditions aux limites spécifiques à appliquer, afin de considérer à la fois l'homogénéisation, mais aussi les possibles répercussions des champs électromagnétiques dans le domaine ?

Le chapitre 6 présentera le travail réalisé dans le domaine des metamatériaux mécaniques. Il proposera notamment des solutions adaptées en terme d'optimisation et d'homogénéisation. On essaiera ici de développer un metamatériau mécanique, constitué d'acier, mais dont les propriétés sont celles d'un élastomère dans une direction de l'espace. Les questions scientifiques posées sont les suivantes :

- Doit-on adapter PSO spécifiquement aux problèmes mécaniques nécessitant une phase d'homogénéisation ?
- Quelles conditions aux limites appliquer afin que le modèle numérique s'adapte à toutes les topologies potentiellement trouvées par le processus d'optimisation ?

Ainsi, nous expliciterons dans chacun des chapitres suivants les efforts réalisés afin de tenter de répondre à ces questionnements scientifiques et ces axes de recherche.

DÉVELOPPEMENT D'UN NOUVEL ALGORITHME D'OPTIMISATION METAHEURISTIQUE INTELLIGENT

Sommaire

3.1	Introduction	100
3.2	Algorithme PageRank inverse	101
3.3	Utilisation de PageRank inverse dans PSO	103
3.3.1	Analogie avec les chaînes de Markov	103
3.3.2	Définition de l'algorithme I-PR-PSO	105
3.3.3	Exemples de calculs réalisés et problématiques rencontrées	108
3.4	Validation sur un benchmark de fonctions mathématiques	109
3.4.1	Paramètres de l'algorithme et fonctions testées	109
3.4.2	Résultats obtenus	111
3.4.3	Pour conclure	115
3.5	Validation en mécanique	115
3.5.1	Présentation du problème	115
3.5.2	Benchmark de treillis et paramètres du calcul	116
3.5.2.1	Treillis à 10 barres	116
3.5.2.2	Treillis à 22 barres	116
3.5.2.3	Paramètres mécaniques et numériques	118
3.5.3	Gestion des contraintes d'optimisation	118
3.5.3.1	Les bornes limites des variables d'optimisations	119
3.5.3.2	Les contraintes intrinsèques au problème	120
3.5.4	Résultats obtenus	120
3.5.4.1	Treillis à 10 barres	120
3.5.4.2	Treillis à 22 barres	122
3.5.5	Pour conclure	122
3.6	Conclusions du chapitre	123

3.1/ INTRODUCTION

Ce chapitre a pour but de présenter les travaux réalisés pour le développement d'un nouvel algorithme d'optimisation intelligent. L'intelligence est ici décrite comme la capacité de l'algorithme à s'adapter au cours des itérations, c'est à dire à modifier son fonctionnement en fonction des résultats qu'il obtient ou de l'évolution du domaine de recherche.

Comme nous l'avons vu dans le chapitre 2, et en particulier dans la partie 2.2.3 concernant l'algorithme PSO, de nombreuses améliorations des capacités initiales de PSO à trouver le minimum de la fonction objectif étudiée ont été proposées dans la littérature. On a vu notamment le travail de Mendes et al. [113, 127, 127, 140] concernant les topologies de la population. Le travail réalisé ici nous a montré que la topologie de la population peut avoir une grande influence sur les résultats obtenus. On en déduit aussi que cette topologie peut être choisie judicieusement en fonction du problème à résoudre. Ainsi, pour un problème que l'on peut qualifier de simple¹ - c'est à dire un problème d'optimisation dont les dimensions et le nombre de variables d'optimisation sont assez réduits - on imagine aisément qu'il ne sera pas nécessaire d'utiliser de topologie géographique, faisant perdre du temps de calcul à chaque itération. Ceci dit, une gestion intelligente et rapide en terme de temps de calcul de la topologie de la population pourrait tout de même améliorer les résultats obtenus et le temps de calcul.

Pourquoi alors ne pas essayer de développer une topologie évolutive, sans considérer la géographie des particules dans le domaine de recherche ?

C'est dans cette problématique que s'inscrit ce travail. L'algorithme Inverse-PageRank-PSO (I-PR-PSO) développé ici propose une topologie adaptative, mais non géographique. Ainsi, elle évolue en fonction des résultats obtenus, et comme nous le verrons par la suite, pondère l'influence des particules les unes sur les autres au cours des itérations. Cet algorithme est ainsi basé conjointement sur l'algorithme PSO et l'algorithme PageRank de Google. En particulier, c'est la version inverse de PageRank présentée par Newton et al. [222] et développée précédemment dans la partie 2.3.3.4 qui a été utilisée ici. C'est dans cette problématique que s'inscrit ce travail. L'algorithme Inverse-PageRank-PSO (I-PR-PSO) développé ici propose une topologie adaptative, mais non géographique. Ainsi, elle évolue en fonction des résultats obtenus, et comme nous le verrons par la suite, pondère l'influence des particules les unes sur les autres au cours des itérations. Cet algorithme est ainsi basé conjointement sur l'algorithme PSO et l'algorithme PageRank de Google. En particulier, c'est la version inverse de PageRank présentée par Newton et al. [222] et développée précédemment dans la partie 2.3.3.4 qui a été utilisée ici.

Ce travail a donné lieu à la publication [242] dans le journal *Advances in Engineering Software* en 2015.

1. On considère ici tout de même l'appartenance d'un problème qualifié de simple à la classe des problèmes NP difficiles définie dans la partie 2.1.

3.2/ ALGORITHME PAGERANK INVERSE : CALCUL DE LA MATRICE STOCHASTIQUE DES TRANSITIONS PROBABILISTES

Le but de ce travail est de déterminer complètement une chaîne de Markov, c'est à dire les transitions probabilistes entre les différents noeuds du graphe, en fonction d'un vecteur stationnaire connu, que l'on notera π_{target} . Le travail de Newton et al. [222] détaille justement une méthode permettant d'ajuster les valeurs de la matrice de transition stochastique, jusqu'à ce que le vecteur stationnaire correspondant à cette matrice corresponde à notre vecteur cible π_{target} . Cette méthode, itérative, consiste simplement à modifier aléatoirement les termes de la matrice des transitions probabilistes, jusqu'à ce que son vecteur stationnaire corresponde à notre vecteur cible.

Le processus itératif est constitué des trois étapes suivantes :

1. Choix d'une matrice initiale C_0 . Cette matrice peut être donnée par des bases de données - comme il l'est fait dans les articles de Newton et al. [222, 232] où les pourcentages stationnaires de propagation des métastases dans les différents organes du corps humain sont connus empiriquement - ou être donnée de manière aléatoire.
2. Utilisation d'un processus itératif pour calculer C_f .
 C_0 est utilisée comme matrice initiale d'un processus itératif. Durant ce calcul, les éléments de la matrice sont ajustés de manière itérative et aléatoire, jusqu'à ce que l'état stationnaire calculé à partir de cette matrice stochastique corresponde au vecteur cible π_{target} , comme nous le verrons dans l'algorithme 11.
3. Parmi les matrices ainsi créées à l'étape 2, choix de la meilleure d'entre elles, de manière qualitative, en fonction de nos connaissances.

Détaillons plus précisément la deuxième étape du pseudo-code précédent [222]. Si l'on appelle C_m la matrice de transition à l'étape m , correspondant au vecteur stationnaire $v_{\infty}^{(m)}$, on a donc le système linéaire suivant

$$v_{\infty}^{(m)}(C_m - I) = 0 \quad (3.1)$$

Notre but est de trouver les éléments de la matrice C_m dont le vecteur stationnaire est en particulier le vecteur cible π_{target} précédemment défini, on a donc

$$\pi_{target}(C_m - I) = 0 \quad (3.2)$$

C'est à dire que l'on a $\|v_{\infty}^{(m)} - \pi_{target}\|^2 = 0$. Du fait que $\pi_{target} \neq v_{\infty}^{(m)}$, on peut calculer un résidu r_m , qui correspond au fait que la relation (3.2) n'est pas vérifiée au cours des itérations, et dont l'expression est la suivante

$$\pi_{target}(C_m - I) = r_m \equiv (\pi_{target} - v_{\infty}^{(m)})(C_m - I) \quad (3.3)$$

avec $\|r_m\|^2 \neq 0$. Notre but est, plus précisément, de trouver les éléments de C_m de telle sorte que $\|r_m\|^2 \leq \epsilon \ll 1$ avec $\epsilon = O(10^{-5})$, où ϵ est notre critère de convergence. En pratique, on calcule $\|\pi_{target} - v_{\infty}^{(m)}\|^2$ directement, et on itère les composantes de C_m

jusqu'à ce que $\|\pi_{target} - v_{\infty}^{(m)}\|^2 < \epsilon$.

Pour réaliser ce calcul, les composantes C_{ij} de la matrice C_m de l'itération m sont modifiées de manière itérative, grâce à l'algorithme 11 suivant.

Algorithm 11 Détail du processus itératif permettant le calcul de la matrice de transition d'une chaîne de Markov, connaissant son vecteur stationnaire

```

[1] Calculer le résidu  $r_m$  à l'étape  $m$ , en commençant par la matrice  $C_0$  ( $m = 0$ ).
[2] Choisir la colonne  $k$  de  $C_m$  correspondant à l'entrée maximale de  $r_m$ .
[3] Choisir la colonne  $c$  de  $C_m$  correspondant à l'entrée minimale de  $r_m$ .
[4] Choisir aléatoirement une ligne  $l$  de  $C_m$ .
[5] Sur cette ligne :
    • Diminuer l'élément  $C_{lk}$  de  $C_m$  de la valeur  $\delta$ .
    • Augmenter l'élément  $C_{lc}$  de  $C_m$  de la valeur  $\delta$ .
[6] Calculer le nouveau  $\|r_{m+1}\|^2$ 
if  $\|r_{m+1}\|^2 < \epsilon$  then
    STOP
else
    Retour à la deuxième étape de l'algorithme.
end if

```

En augmentant et diminuant deux termes d'une même ligne d'un même facteur δ , on voit bien dans cet algorithme que la somme des composantes d'une même ligne de C reste toujours égale à 1, et que donc la matrice reste stochastique et en corrélation avec les hypothèses de Markov réalisées et présentées précédemment.

Grâce à cette méthode, on peut ainsi calculer, moyennant un processus itératif, la matrice de transition d'une chaîne de Markov, connaissant son vecteur stationnaire. D'après les résultats présentés dans [222, 232], ces calculs ne prennent généralement pas plus d'une centaine d'itérations (quelques secondes en terme de CPU), pour un graphe constitué d'une cinquantaine de sommets.

Notons que le paramètre δ doit être calculé comme une fonction du vecteur stationnaire π_{target} cible à atteindre. Après de nombreux essais, nous avons déterminé que δ doit être donné comme l'ordre de grandeur de la plus petite valeur de π_{target} , et est calculé à la manière donnée dans l'algorithme 12 suivant.

Algorithm 12 Algorithme permettant de calculer le facteur δ

```

 $ii = \min(\pi_{target}^T)$ 
magn. = 0
while  $ii \leq 1$  do
     $ii = ii \times 10$ 
    magn. = magn. + 1
end while
 $\delta = 1 \times 10^{-magn.}$ 

```

Dans la matrice finale C_f , les éléments non-nuls de la ligne k correspondent aux liens sortants du noeud k , alors que les éléments non-nuls de la colonne k correspondent aux

liens entrant dans le noeud k .

Du fait des nombreux paramètres aléatoires de l'algorithme 11, la matrice finale C_f peut être légèrement différente d'un calcul à un autre, et ce même si la matrice initiale C_0 est identique. Cette matrice finale dépend finalement des lignes choisies aléatoirement durant l'algorithme 12 précédent. Newton et al. [232] ont réalisé une étude statistique pour montrer les différences entre les matrices finales, conditionnées par la même matrice initiale. Ils montrent que ces différences peuvent être négligées.

En conclusion, ce processus PageRank inverse nous permet de déterminer les termes de la matrice de transition stochastique d'une chaîne de Markov, en choisissant - ou tout du moins en connaissant - le vecteur stationnaire qui définit cette chaîne de Markov.

3.3/ UTILISATION DE L'ALGORITHME PAGERANK INVERSE DANS LA DÉFINITION DE LA TOPOLOGIE DES PARTICULES PSO

3.3.1/ ANALOGIE AVEC LES CHAÎNES DE MARKOV

Dans la version linéaire de PSO que nous avons précédemment vue dans la partie 2.2.3, on considère le voisinage de chacune des particules. C'est à dire que les particules sont liées entre elles, et ce lien indique l'influence qu'elles ont les unes sur les autres. On a donc affaire à un graphe tel que défini précédemment dans la partie 2.3.2 à la page 70. On pourrait ainsi imaginer de pondérer ces liens, c'est à dire de pondérer l'influence que les particules ont les unes sur les autres. On aurait alors affaire à un graphe orienté, dont les liens sont pondérés, tel que défini dans la partie 2.3.2.1.2 à la page 73. Ainsi, considérant que les particules sont des potentielles solutions au problème d'optimisation posé, défini dans le domaine des solutions, elles représentent aussi des potentielles futures positions pour chacune des autres particules. On peut donc considérer que cette pondération représente aussi une probabilité de transition d'un état à un autre. La population PSO pourrait donc être considérée comme un graphe probabiliste tel que défini dans la partie 2.3.2.2.1 à la page 76.

En outre, considérant que l'algorithme PSO est itératif, il utilise à chaque fois les informations connues à l'instant t pour déterminer la position des particules à l'instant $t + 1$. Ainsi, toute l'information utile pour la prédiction du futur est contenue dans l'état présent du processus. Cela fait donc de notre population un graphe orienté particulier : une chaîne de Markov à temps discret.

Compte tenu des considérations précédentes, nous décidons donc de développer une nouvelle gestion de la population des particules PSO. La population que nous développons ici est basée sur le voisinage GBEST présenté dans la partie 2.2.3.2.2 à la page 46, en ajoutant une pondération de l'influence des particules les unes sur les autres. Un graphe très simple pourrait être celui présenté dans la figure 3.1 et dont la matrice stochastique des transitions est donnée dans l'équation (3.4). Ce graphe pourrait représenter un essaim de 4 particules PSO indépendantes.

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \quad (3.4)$$

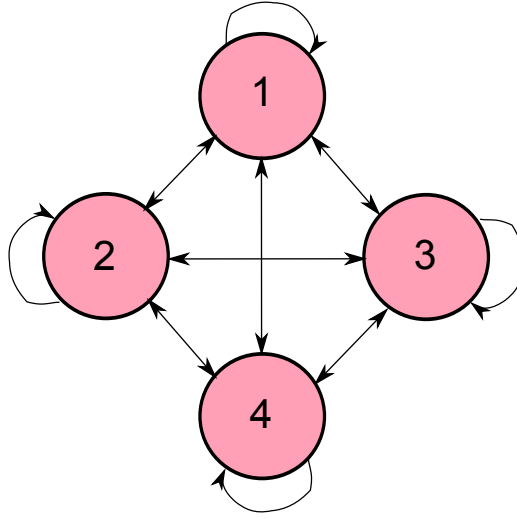


FIGURE 3.1 – Graphe orienté simplifié

Afin que la pondération de l'influence des particules les unes sur les autres soit faite de manière intelligente, il est nécessaire de recalculer cette pondération, c'est à dire les termes de la matrice stochastique des transitions probabilistes, à chaque itération de l'algorithme PSO.

Pour ce faire, il est nécessaire de connaître le vecteur stationnaire de la chaîne de Markov considérée, c'est à dire le vecteur PageRank tel que défini dans la partie 2.3.3. Ainsi, un algorithme d'identification inverse pourra être utilisé afin de déterminer, à partir des valeurs du vecteur PageRank, les valeurs de la matrice de transition.

Dans l'algorithme Inverse-PageRank-PSO (I-PR-PSO) que nous proposons ici, il est donc nécessaire de choisir intelligemment quelles informations pourront être utilisées pour définir le vecteur stationnaire de la chaîne de Markov considérée. Afin que l'influence des particules les unes sur les autres soit définie en fonction de la performance de l'algorithme, nous décidons d'utiliser le succès relatif de chacune des particules pour calculer le vecteur PageRank. Ce vecteur possède donc autant de termes qu'il y a de particules dans l'essaim, et chacun de ces termes, c'est à dire le succès relatif de chacune d'entre elles par rapport à la meilleure particule G_{best} , est défini comme donné dans l'équation (3.5) suivante

$$\pi_{target}^T(1, k) = \left| \frac{fitness(G_{best}) \times 100}{fitness(G_{best}) - fitness(P_k) + \epsilon} \right| \forall k = 1, 2, \dots, n \quad (3.5)$$

où $fitness(X)$ est la valeur de la fonction objectif de la particule X , et n représente le nombre de particules de l'essaim. Le paramètre ϵ (10^{-7} ou 10^{-15} en fonction de la précision de l'ordinateur employé) est utilisé pour éviter une division par 0 lorsque $fitness(G_{best}) = fitness(P_k)$. L'équation (3.5) représente donc une classification des particules non pas basée sur leur rang au sein de la population, mais par rapport à leur distance de la meilleure particule G_{best} . Plus proche est la particule P_k de la meilleure, plus grande sera la valeur lui correspondant dans le vecteur π_{target} . Afin que la somme de chacun des termes du vecteur PageRank soit égale à 1, ce qui est une condition nécessaire puisque π_{target} est un vecteur de probabilités, ce vecteur est ensuite normalisé, comme on le voit dans l'équation (3.6) suivante

$$\pi_{target}^T(1, k) \leftarrow \frac{\pi_{target}^T(1, k)}{\sum_{k=1}^D (\pi_{target}^T(1, k))} \quad \forall k = 1, 2, \dots, n \quad (3.6)$$

où k représente la $k^{\text{ème}}$ composante du vecteur π_{target} .

L'équation (3.5) n'est valable que dans le cas d'une minimisation. En effet, les particules obtenant la valeur la plus petite de la fonction objectif obtiendront le meilleur rang dans le classement, alors que celles obtenant des valeurs plus grandes seront moins bien classées. Notons tout de même que les problèmes d'optimisation par maximisation sont souvent ramenés à des problèmes par minimisation, en considérant l'inverse ou l'opposé de la fonction objectif à maximiser.

Le vecteur défini dans l'équation (3.6) est maintenant considéré comme le vecteur cible à atteindre lors du calcul des termes de la matrice de transitions stochastique C .

Le but de notre algorithme I-PR-PSO est ici de calculer les termes de la matrice C de dimensions $(n \times n)$ connaissant π_{target} . Ainsi, en tenant compte du vecteur stationnaire de la chaîne de Markov, les meilleures particules - c'est à dire celles obtenant les plus petites valeurs de la fonction objectif - auront une plus grande influence sur les autres particules de l'essaim. Notons que les contraintes imposées par la théorie des chaînes de Markov sont $0 \leq C_{kl} \leq 1$ et $\sum_{l=1}^n C_{kl} = 1$. On a donc affaire ici à un processus de PageRank inverse, où l'on connaît le vecteur stationnaire de la chaîne de Markov considérée, et l'on cherche ses probabilités de transitions qui définiront, par la suite, l'influence qu'on les particules les unes sur les autres au cours des itérations.

3.3.2/ DÉFINITION DE L'ALGORITHME I-PR-PSO

Le but de cette partie est de démontrer comment l'utilisation d'un algorithme PageRank inverse pourrait rendre l'algorithme PSO plus intelligent, c'est à dire plus adaptable au cours des itérations en fonction du succès relatif de chacune des particules de l'essaim vis à vis de la meilleure. Comme nous l'avons vu précédemment, et en particulier dans le travail de Mendes et al. [113, 127, 136, 140], la topologie de la population PSO, c'est à dire l'organisation interne des particules et la manière dont elles s'influencent les unes les autres au sein de l'essaim, peut avoir une grande influence sur les résultats obtenus. On a vu notamment que les versions **GBEST** ou *Fully Informed PSO* donnent de très bons résultats, notamment en terme de vitesse de convergence. Ici, nous souhaiterions adapter la topologie GBEST en pondérant l'influence qu'ont les particules les unes sur les autres en fonction de leur réussite relative dans l'essaim.

On rappelle ici les équations (2.23) que l'on a vues précédemment à la page 44 et qui donnent le changement des positions de chacune des particules dans l'espace des solutions de l'algorithme PSO linéaire :

$$\begin{cases} V_i^k = V_i^{k-1} + c_1(P_{i,best}^k - X_i^{k-1}) + c_2(G_{i,best}^k - X_i^{k-1}) \\ X_i^k = X_i^{k-1} + V_i^k \end{cases} \quad (3.7)$$

où l'on a :

- X_i^k : La position de la particule $n^o i$ à l'itération k .
- V_i^k : La vitesse de la particule $n^o i$ à l'itération k .

- $G_{i,best}^k$: La meilleure position atteinte par les particules du voisinage de la particule i , à l'itération k .
- $P_{i,best}^k$: La meilleure position personnelle atteinte par la particule $n^o i$ à l'itération k .

Dans I-PR-PSO, compte tenu des considérations précédentes, les équations de changements des positions des particules dans l'espace des solutions sont les suivantes :

$$\begin{cases} V_i^{t+1} = \omega \times V_i^t + c_1 \times rand_1 \times (P_{i,best}^{t+1} - X_i^t) + c_2 \times rand_2 \times \sum_{j=1}^n C_{ij} \times [P_{j,best}^{t+1} - X_i^t] \\ X_i^{t+1} = X_i^t + V_i^{t+1} \end{cases} \quad (3.8)$$

où C_{ij} est le terme de la $i^{\text{ème}}$ ligne, $j^{\text{ème}}$ colonne de la matrice des transitions stochastiques C précédemment définie.

Comme nous l'avons vu précédemment, chaque particule i de l'essaim est influencée par toutes les particules de l'essaim, et leur influence respective est donnée par les composantes de la ligne i de la matrice de transitions stochastique, c'est à dire $C_{ij} \forall j$. En effet, dans la matrice finale C_f définie par le processus de Markov inverse, les éléments non-nuls de la ligne k correspondent aux liens sortants du nœud k , alors que les éléments non-nuls de la colonne k correspondent aux liens entrant dans le nœud k . C'est à dire que chaque ligne de la matrice des transitions probabilistes définit les probabilités, pour chacune des particules, de passer de leur état actuel à l'état dans lequel se trouvent les autres particules de l'essaim. Dans le cas de PSO, c'est l'influence des particules les unes sur les autres qui déterminera leur mouvement dans l'espace, et non pas leurs probabilités de passer d'un état à un autre dans l'espace des solutions. Il est donc nécessaire ici d'inverser le processus de Markov, et donc de considérer que la probabilité d'une particule de passer d'un état (a) à un autre état (b) devient l'influence de ce deuxième état (b) sur son état actuel (a).

Par exemple, revenons au graphe simplifié que nous avons vu dans la partie 2.3 à la page 76, et que nous rappelons en figure 3.2 suivante. La matrice des transitions probabilistes de ce graphe orienté serait celle donnée en équation (3.9). Cette matrice signifierait, dans le cas d'une topologie de particules PSO, que la particule 1 a 50% de chances de passer dans l'état de la particule 2, et 50% de chances de passer dans l'état de la particule 3. Dans I-PR-PSO, nous souhaitons définir non pas les probabilités de transitions des particules les unes vers les autres, mais l'influence qu'elles ont les unes sur les autres. On considèrera donc ici que la particule 1 est influencée par la particule 2 d'un facteur 1/2 et par la particule 3 d'un facteur 1/2. Ainsi, dans I-PR-PSO, la ligne k de la matrice de transition probabiliste représente en quelle mesure la particule k de l'essaim est influencée par les autres, alors que la colonne k de cette même matrice montre en quelle mesure la particule k influence ses congénères.

$$H = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.9)$$

Evidemment, puisque la somme des termes de chaque ligne de la matrice des transitions probabilistes vaut toujours 1, ce n'est pas le cas de ses colonnes. On considère donc ici

que la somme des influences de chacune des particules sur les autres est supérieure à 1. Dans le cas du calcul des vitesses des particules tel que défini dans les équations (3.8) précédentes, il est absolument nécessaire que l'influence que reçoit chacune des particules ne soit pas supérieure à 1, notamment pour que les vitesses ne deviennent pas trop grandes au bout de quelques itérations seulement. En effet, des vitesses trop

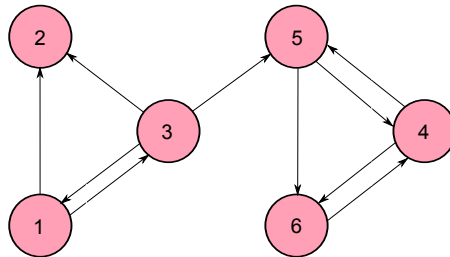


FIGURE 3.2 – Exemple de graphe et définitions

grandes pourraient empêcher l'algorithme de converger, en déplaçant les particules très rapidement aux bords du domaine des solutions. Par contre, l'influence prodiguée par chacune des particules sur les autres peut ne pas être considérée comme une probabilité, ou un pourcentage, fonction de leur état actuel, et peut donc être supérieure à 1.

L'algorithme I-PR-PSO global est décrit dans l'algorithme 13 suivant, où $it_{PSO,max}$ est le nombre maximum d'itérations de l'algorithme I-PR-PSO, et $it_{PR,max}$ est le nombre maximum d'itérations du calcul PageRank inverse.

Algorithm 13 Algorithme I-PR-PSO global

Définition aléatoire de la vitesse des particules

Définition aléatoire de la position des particules

for $iteration_{PSO} = 1$ to $it_{PSO,MAX}$ **do**

 Calcul de la fonction objectif

 Calcul du vecteur stationnaire cible π_{target} avec l'équation (3.5).

 Définition aléatoire de la première matrice de transitions stochastique C_0

 Calcul du premier résidu $r_m = r_0$

while ($\|r_m\| > \epsilon$) ET ($iteration_{PR} \leq it_{PR,MAX}$) **do**

 Recherche de la meilleure matrice C en utilisant l'algorithme 11

end while

 Mise à jour des meilleures performances personnelles $P_{i,best}^{iteration_{PSO}}$

 Mise à jour de la meilleure performance globale de l'essaim G_{Best} et de la valeur de sa fonction objectif dans la mémoire

 Calcul de la nouvelle vitesse des particules avec l'équation (3.8)

 Calcul de la nouvelle position des particules dans l'espace des solutions avec l'équation (3.8)

end for

3.3.3/ EXEMPLES DE CALCULS RÉALISÉS ET PROBLÉMATIQUES RENCONTRÉES

Plusieurs exemples de calculs ont été réalisés afin de calculer une matrice de transition stochastique à partir d'un vecteur PageRank donné. Considérons un essaim constitué de 4 particules PSO, dont les valeurs obtenues de la fonction objectif sont données dans le vecteur stationnaire suivant $\pi_{target}^T = [1 \ 2 \ 3 \ 4]$. Ce vecteur est ensuite normalisé et on obtient finalement le vecteur cible à atteindre suivant : $\pi_{target}^T = [0.0667 \ 0.1333 \ 0.2000 \ 0.6000]$.

La première matrice de transition stochastique C_0 est donnée de manière aléatoire, et chacune de ses lignes est normalisée de telle sorte que la somme des termes de chacune de ses lignes soit égale à 1.

L'utilisation de l'algorithme 11 précédent nous donne le vecteur PageRank final suivant : $\pi_f^T = [0.0737 \ 0.1435 \ 0.2099 \ 0.5729]$ et on a $\|\pi_{target}^T\| - \|\pi_f^T\| = 0.0314$. On voit ici que les deux vecteurs stationnaires, c'est à dire le vecteur initial cible, et le vecteur calculé à partir de la matrice finale C_f sont assez similaires. La matrice finale C_f est donnée dans l'équation (3.10) et la convergence du résidu r_m est donnée en figure 3.3.

$$\begin{bmatrix} 0.0024 & 0.0981 & 0.1648 & 0.7347 \\ 0.0092 & 0.0098 & 0.2633 & 0.7178 \\ 0.0031 & 0.1443 & 0.2773 & 0.8470 \\ 0.1250 & 0.1826 & 0.2773 & 0.4151 \end{bmatrix} \quad (3.10)$$

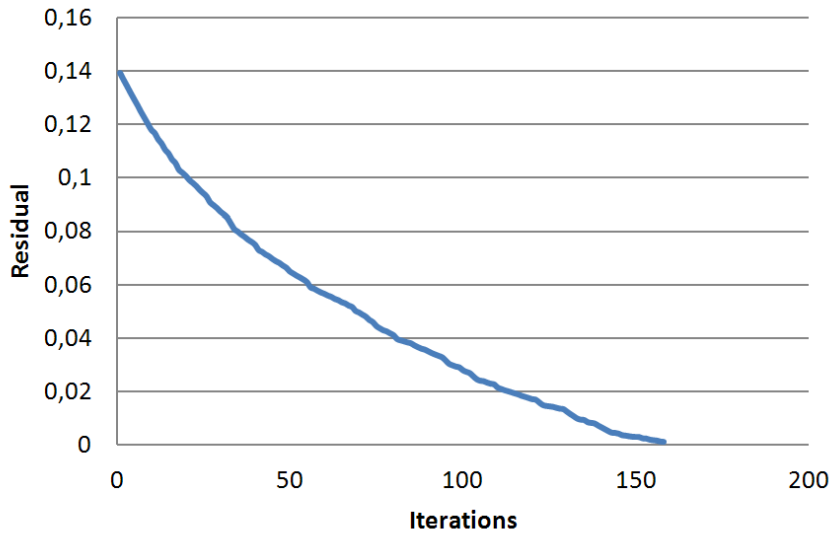


FIGURE 3.3 – Convergence du résidu r_m au cours des itérations

Comme nous le voyons dans la matrice de l'équation (3.10), sur chaque ligne, c'est la quatrième colonne qui contient les valeurs les plus élevées. Ainsi, on voit ici que c'est bel et bien la quatrième particule de l'essaim qui influence le plus les autres particules de l'essaim, dans le sens où les autres particules ont plus de chances de passer de leur état actuel à l'état dans lequel se situe la quatrième particule. En considérant le déplacement des particules dans l'espace des solutions, on dirait que les particules ont plus de chances de se déplacer dans la direction de cette quatrième particule.

Il est important de noter que lorsque les valeurs du vecteur cible initial π_{target} sont trop éloignées les unes des autres (plusieurs puissances de 10), le calcul ne converge pas. En effet, dans ce cas, le facteur δ calculé à partir de l'algorithme 12 précédent est trop petit pour modifier efficacement les valeurs de la matrice de transition stochastique initiale. Par exemple, si le vecteur cible initial est $\pi_{target}^T = \begin{bmatrix} 1 & 1.E^{-10} & 1.E^{-10} & 1.E^{-10} \end{bmatrix}$ avant la normalisation, la matrice finale reste identique à la matrice initiale, car le paramètre δ vaut $1.E^{-10}$.

De la même manière, si les composantes de π_{target} sont quasiment identiques, la topologie finale est une topologie GBEST classique. C'est à dire que la matrice de transitions stochastique est composée de termes égaux ou presque. Dans ce cas, le calcul ne converge pas du tout car on a $\pi_{target}^T = \pi_0^T$ et le premier résidu calculé r_0 est nul et directement inférieur à ϵ_{PR} . L'algorithme 11 n'active donc jamais la boucle de modification de la matrice. Dans ce cas, la matrice obtenue est finalement la matrice aléatoire initialement définie. Cette stratégie correspond à celle proposée par Mohais et al. [140] où il a été suggéré qu'une redéfinition aléatoire de la topologie de la population pouvait être compétitive par rapport aux topologies statiques prédéfinies. De plus, il a été montré dans la littérature que la trop grande proximité des individus pouvait être un facteur déterminant de convergence prématurée de l'algorithme. Cette redéfinition aléatoire des influences des particules les unes sur les autres est une solution à ce problème.

3.4/ VALIDATION SUR UN BENCHMARK DE FONCTIONS MATHÉMATIQUES

3.4.1/ PARAMÈTRES DE L'ALGORITHME ET FONCTIONS TESTÉES

Comme il a été montré dans la littérature que PSO pouvait être plus efficace que les autres méthodes metaheuristiques sur des problèmes en grandes dimensions [207, 239], nous avons décidé de ne comparer notre algorithme I-PR-PSO qu'avec d'autres versions de PSO. Notre algorithme a ainsi été testé et comparé avec trois autres de ses pairs : l'algorithme PSO classique dont les topologies sont définies par les méthodes GBEST, LBEST et 4-clusters précédemment présentées dans la partie 2.2.3.2.2 à la page 46. Les fonctions testées sont résumées dans le tableau 3.2 suivant, où D représente la dimension du problème à traiter. Une étude statistique a été réalisée afin de prouver l'efficacité de notre algorithme. Ainsi, 100 calculs d'optimisation différents ont été réalisés en dimensions 10, 20, 30 et 50 sur chacune des fonctions objectifs du tableau 3.2. Les paramètres de l'algorithme sont donnés dans le tableau 3.1 suivant. Les calculs ont été réalisés au Mésocentre de Calcul de Franche-Comté.

Nombre de particules PSO dans l'essaim	50
Paramètre d'inertie ω	0.8
Constantes d'accélération $c_1 = c_2$	2
Nombre maximal d'itérations PSO	600
Seuil de convergence de l'algorithme PageRank inverse ϵ_{PR}	$1E-03$
Nombre maximal d'itérations du processus PageRank inverse	6000
Dimension du problème à résoudre	10, 20, 30 et 50

Tableau 3.1 – Paramètres du calcul d'optimisation

Fonction	Expression mathématique	Pos. de l'optimum	Val. de l'optimum	Type	domaine	V_{max}
f1 Ackley	$20 + e - 20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}} - e^{\frac{\sum_{i=1}^D \cos(2\pi x_i)}{D}}$	(0, 0, ..., 0)	0	Multimodale	$[-1; 1]$	1
f2 Griewank	$1 + \sum_{i=1}^D \frac{(x_i - 100)^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})$	0	Multimodale	$[-600; 600]$	500	
f3 Rastrigin	$\sum_{i=1}^D x_i^2 - 10\cos(2\pi x_i) + 10$	(0, 0, ..., 0)	0	Multimodale	$[-5.12; 5.12]$	5
f4 Rosenbrock	$\sum_{i=1}^{D-1} 100((x_{i+1} - x_i)^2 + (x_i + 1)^2)$	(1, 1, ..., 1)	0	Unimodale	$[-50; 50]$	50
f5 Sphere	$\sum_i x_i^2$	(0, 0, ..., 0)	0	Unimodale	$[-50; 50]$	50
f6 Rotated hyper ellips.	$\sum_{i=1}^D (\sum_{j=1}^i x_j^2)$	(0, 0, ..., 0)	0	Multimodale	$[-65.536; 65.536]$	65
f7 Shifted Rastrigin	$\sum_{i=1}^D ((x_i - 1)^2 - 10\cos(2\pi(x_i - 1)) + 10) + 390$	(1, 1, ..., 1)	390	Multimodale	$[-5; 5]$	5
f8 Shifted Rosenbrock	$\sum_{i=1}^{D-1} (100((x_{i+1} - 1) - (x_i - 1)^2)^2 + (x_{i+1} - 1)^2) + 390$	(1, 1, ..., 1)	390	Multimodale	$[-100; 100]$	100
f9 Shifted Sphere	$\sum_{i=1}^D ((x_i - 1)^D) + 400$	(1, 1, ..., 1)	450	Multimodale	$[-100; 100]$	100
f10 Shifted Ackley	$-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D ((x_i - 1)^2)}) - \exp(\frac{1}{D}\sum_{i=1}^D \cos(12\pi(x_i - 1)))$	(1, 1, ..., 1)	200	Multimodale	$[-32; 32]$	32
f11 Bohachevsky	$\sum_{i=1}^{D-1} (x_i^{12} + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1})) + 0.7$	(0, 0, ..., 0)	0	Unimodale	$[-15; 15]$	15
f12 Schwefel's problem 1.2	$\sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	(0, 0, ..., 0)	0	Unimodale	$[-65.536; 65.536]$	65.536

Tableau 3.2 – Benchmark de fonctions mathématiques

3.4.2/ RÉSULTATS OBTENUS

On étudie ici les résultats - c'est à dire la valeur de la fonction objectif finale - obtenus avec I-PR-PSO, et on les compare avec ceux obtenus avec les autres versions de PSO testées. Les résultats sont présentés en échelle logarithmique pour une meilleure lisibilité dans les figures 3.4 à 3.7 suivantes.

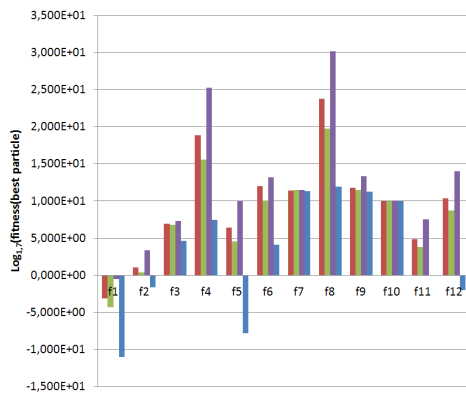


FIGURE 3.4 – Dimension 10

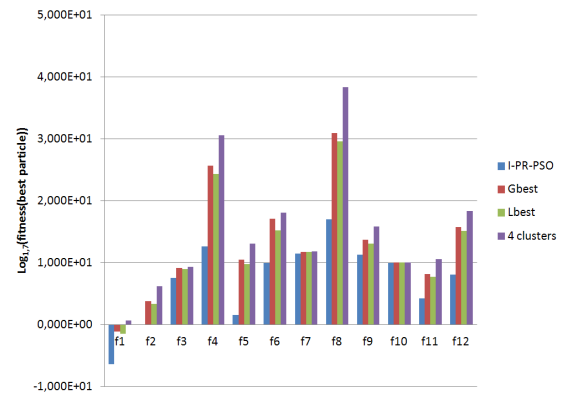


FIGURE 3.5 – Dimension 20

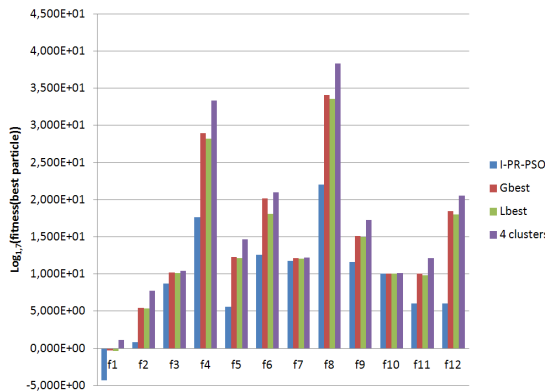


FIGURE 3.6 – Dimension 30

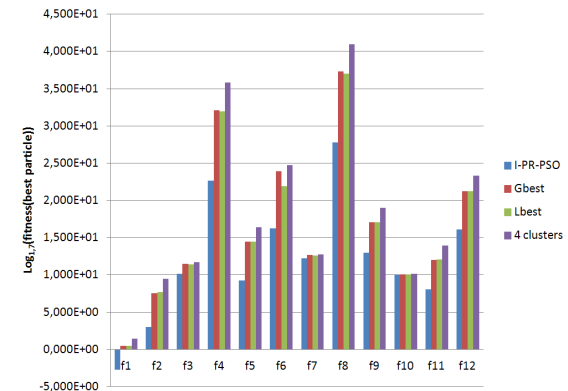


FIGURE 3.7 – Dimension 50

Figures 3.4 - 3.7 : Valeurs obtenues pour les 12 fonctions objectif testées, après 600 itérations. Comparaison de I-PR-PSO avec ses pairs.

Comme on le voit dans les figures 3.4 à 3.7, l'algorithme I-PR-PSO a tendance à se rapprocher toujours plus près du minimum global de la fonction objectif considérée que ses congénères. Les valeurs obtenues sont d'ailleurs données dans le tableau 3.3 suivant.

Basés sur ces résultats, on conclut que notre algorithme I-PR-PSO est plus performant que ses congénères, en tout cas en ce qui concerne ses capacités à se rapprocher du minimum global de la fonction objectif considérée. Bien sûr, le théorème du *no-free lunch* présenté dans la partie 2.2.1.2 à la page 25 nous dit qu'une méthode ne peut être meilleure que toutes les autres *a priori*. Ceci dit, nous pouvons tout de même conclure que I-PR-PSO est plus performant que ses pairs au moins sur les 12 fonctions testées.

Afin d'être en mesure d'évaluer l'algorithme sur ses capacités à converger, notamment en terme de vitesse de convergence, nous présentons dans les figures 3.8 à 3.19 les courbes de convergence de l'algorithme I-PR-PSO en dimension 50, sur les 12 fonctions

	Inverse-PageRank-PSO	PSO GBest	PSO Lbest	PSO 4-clusters
Dimension 10				
f1	2.98E-03 ± 1.263E-03	1.95E-01 ± 5.79E-02	1.01E-01 ± 3.26E-02	7.68E-01 ± 1.24E-01
f2	4.26E-01 ± 1.19E-01	1.77E+00 ± 3.48E-01	1.21E+00 ± 1.68E-01	6.03E+00 ± 1.46E+00
f3	1.17E+01 ± 6.86E+00	4.06E+01 ± 6.04E+00	3.65E+01 ± 6.04E+00	4.91E+01 ± 6.51E+00
f4	5.23E+01 ± 9.39E+01	2.23E+04 ± 1.62E+04	3.96E+03 ± 3.31E+03	6.74E+05 ± 3.62E+05
f5	1.59E-02 ± 1.57E-02	2.97E+01 ± 1.36E+01	1.14E+01 ± 6.34E+00	2.01E+02 ± 5.64E+01
f6	8.89E+00 ± 7.55E+00	5.93E+02 ± 1.98E+02	2.09E+02 ± 8.56E+01	1.11E+03 ± 3.78E+02
f7	4.06E+02 ± 4.93E+00	4.30E+02 ± 5.32E+00	4.37E+02 ± 7.39E+00	4.37E+02 ± 7.39E+00
f8	5.68E+02 ± 4.13E+02	2.98E+05 ± 2.79E+05	3.58E+04 ± 3.88E+04	9.03E+06 ± 5.02E+06
f9	4.00E+02 ± 7.80E-02	5.12E+02 ± 5.82E+01	4.42E+02 ± 2.04E+01	1.19E+03 ± 2.14E+02
f10	2.00E+02 ± 3.80E-01	2.06E+02 ± 8.00E-01	2.04E+02 ± 6.43E-01	2.10E+02 ± 8.87E-01
f11	9.95E-01 ± 6.39E-01	1.32E+01 ± 3.32E+00	7.47E+00 ± 1.90E+00	5.46E+01 ± 1.30E+01
f12	3.48E-01 ± 3.87E-01	2.43E+02 ± 1.06E+02	1.02E+02 ± 6.19E+01	1.73E+03 ± 5.34E+02
Dimension 20				
f1	3.39E-02 ± 1.22E-02	5.51E-01 ± 1.08E-01	4.58E-01 ± 9.59E-02	1.43E+00 ± 1.72E-01
f2	1.06E+00 ± 5.36E-02	7.44E+00 ± 1.74E+00	5.91E+00 ± 1.46E+00	2.69E+01 ± 5.22E+00
f3	5.62E+01 ± 2.55E+01	1.29E+02 ± 1.02E+01	1.17E+02 ± 1.37E+01	1.46E+02 ± 1.09E+01
f4	8.24E+02 ± 8.21E+02	8.22E+05 ± 4.50E+05	4.17E+05 ± 2.51E+05	1.14E+07 ± 4.21E+06
f5	2.29E+00 ± 1.34E+00	2.62E+02 ± 7.46E+01	1.85E+02 ± 4.96E+01	1.03E+03 ± 2.13E+02
f6	1.97E+02 ± 9.46E+01	9.00E+03 ± 2.19E+03	3.32E+03 ± 1.29E+03	1.52E+04 ± 3.86E+03
f7	4.52E+02 ± 2.57E+01	5.19E+02 ± 1.03E+01	5.11E+02 ± 1.35E+01	5.38E+02 ± 1.10E+01
f8	8.35E+03 ± 1.28E+04	1.37E+07 ± 7.24E+06	6.81E+06 ± 3.91E+06	2.92E+08 ± 2.71E+08
f9	4.09E+02 ± 5.00E+00	1.47E+03 ± 2.96E+02	1.07E+03 ± 2.10E+02	4.47E+03 ± 7.34E+02
f10	2.03E+02 ± 4.89E-01	2.09E-02 ± 8.66E-01	2.08E+02 ± 8.03E-01	2.14E+02 ± 6.33E-01
f11	9.76E+00 ± 2.35E+00	7.87E+01 ± 1.72E+01	6.15E+01 ± 1.51E+01	2.75E+02 ± 4.93E+01
f12	7.25E+01 ± 4.12E+01	4.24E+03 ± 1.11E+03	3.15E+03 ± 9.79E+02	1.74E+04 ± 2.88E+03
Dimension 30				
f1	1.01E-01 ± 2.69E-02	8.57E-01 ± 1.46E-01	8.18E-01 ± 1.27E-01	1.78E+00 ± 1.63E-01
f2	1.52E+00 ± 2.33E-01	1.78E+01 ± 3.52E+00	1.74E+01 ± 3.93E+00	6.00E+01 ± 9.33E+00
f3	1.02E+02 ± 4.39E+01	2.27E+02 ± 1.40E+01	2.17E+02 ± 1.93E+01	2.55E+02 ± 1.46E+01
f4	1.14E+04 ± 9.16E+03	4.63E+06 ± 2.08E+06	3.20E+06 ± 1.49E+06	4.77E+07 ± 1.47E+07
f5	1.97E+01 ± 7.62E+00	6.86E+02 ± 1.54E+02	6.14E+02 ± 1.46E+02	2.36E+03 ± 4.15E+02
f6	7.98E+02 ± 4.15E+02	4.41E+04 ± 1.31E+04	1.49E+04 ± 5.18E+03	6.95E+04 ± 2.12E+04
f7	5.12E+02 ± 4.02E+01	6.18E+02 ± 1.40E+01	6.40E+02 ± 1.98E+01	6.46E+02 ± 1.51E+01
f8	1.19E+05 ± 9.57E+04	7.07E+07 ± 3.37E+07	5.34E+07 ± 2.60E+07	6.92E+08 ± 2.71E+08
f9	4.76E+02 ± 2.91E+01	3.05E+03 ± 6.11E+02	2.76E+03 ± 5.17E+02	9.68E+03 ± 1.38E+03
f10	2.04E+02 ± 5.41E-01	2.11E+02 ± 9.36E-01	2.11E+02 ± 8.24E-01	2.15E+02 ± 5.88E-01
f11	2.43E+01 ± 5.45E+00	2.07E+02 ± 4.13E+01	1.83E+02 ± 3.67E+01	6.21E+02 ± 1.00E+02
f12	2.43E+01 ± 4.45E+00	1.77E+04 ± 4.05E+03	1.42E+04 ± 3.24E+03	5.48E+04 ± 8.89E+03
Dimension 50				
f1	2.35E-01 ± 5.10E-02	1.27E+00 ± 1.90E-01	1.26E+00 ± 1.65E-01	2.13E+00 ± 1.25E-01
f2	4.90E+00 ± 1.18E+00	5.41E+01 ± 1.03E+01	5.79E+01 ± 1.13E+01	1.52E+02 ± 1.89E+01
f3	2.20E+02 ± 7.53E+01	4.41E+02 ± 1.82E+01	4.18E+02 ± 2.73E+01	4.90E+02 ± 2.21E+01
f4	1.67E+05 ± 8.13E+04	2.53E+07 ± 9.20E+06	2.29E+07 ± 8.61E+06	1.79E+08 ± 4.93E+07
f5	1.35E+02 ± 3.80E+01	2.13E+03 ± 3.83E+02	2.15E+03 ± 3.59E+02	5.91E+03 ± 8.87E+02
f6	5.59E+03 ± 3.09E+03	3.21E+05 ± 8.21E+04	1.10E+05 ± 4.17E+04	5.10E+05 ± 1.59E+05
f7	6.51E+02 ± 7.02E+01	8.28E+02 ± 1.93E+01	8.08E+02 ± 2.60E+01	8.79E+02 ± 2.29E+01
f8	2.52E+06 ± 1.47E+06	4.01E+08 ± 1.32E+08	3.41E+08 ± 1.23E+08	2.77E+08 ± 7.59E+08
f9	9.66E+02 ± 1.66E+02	8.52E+03 ± 1.45E+03	8.44E+03 ± 1.40E+03	2.38E+04 ± 3.03E+03
f10	2.00E+02 ± 3.80E-01	2.13E+02 ± 5.95E-01	2.13E+02 ± 7.30E-01	2.17E+02 ± 4.22E-01
f11	7.28E+01 ± 1.63E+01	5.87E+02 ± 1.11E+02	6.14E+02 ± 9.87E+01	1.63E+03 ± 2.46E+02
f12	5.18E+03 ± 1.53E+03	7.96E+04 ± 1.49E+04	7.94E+04 ± 1.42E+04	2.34E+05 ± 3.12E+04

Tableau 3.3 – 100 calculs d'optimisation : valeurs obtenues après 600 itérations PSO (moyenne ± écart type)

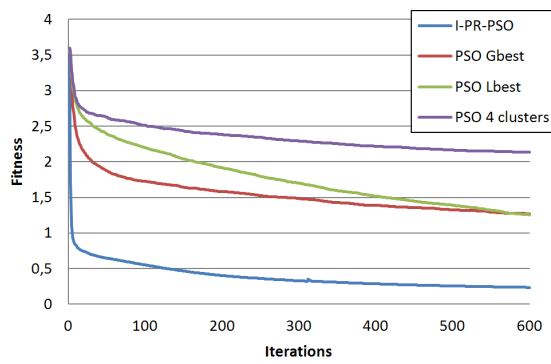


FIGURE 3.8 – Ackley

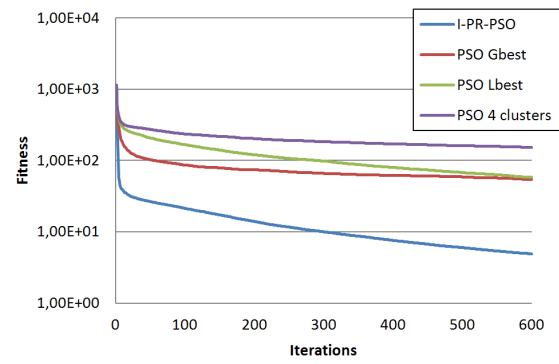


FIGURE 3.9 – Griewank

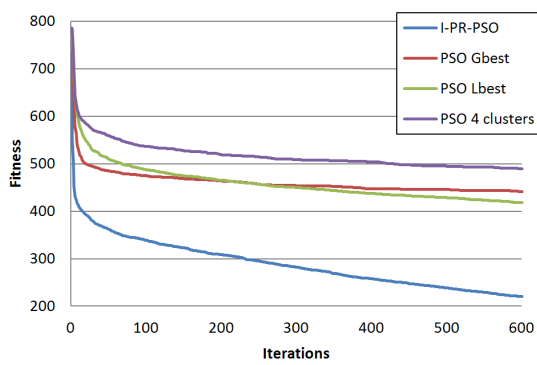


FIGURE 3.10 – Rastrigin

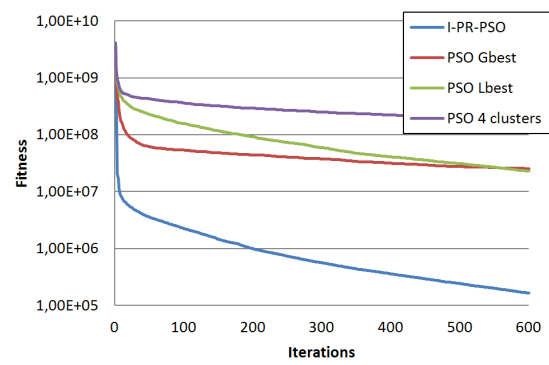


FIGURE 3.11 – Rosenbrock

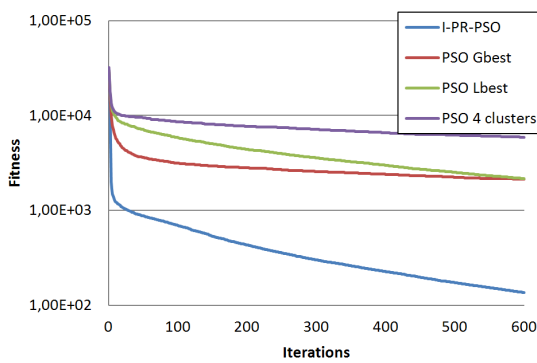


FIGURE 3.12 – Sphere

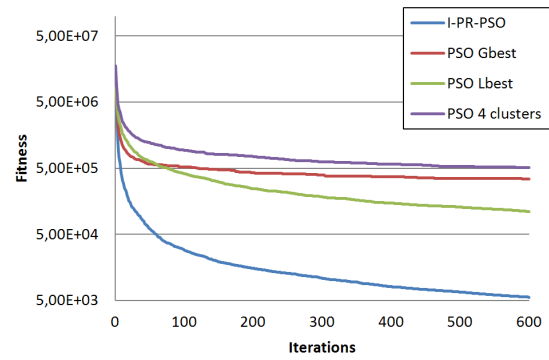


FIGURE 3.13 – Rotate hyper ellipsoid function

Figures 3.8 - 3.13 : courbes de convergence des calculs d'optimisation en dimension 50 (moyenne des 100 calculs réalisés)

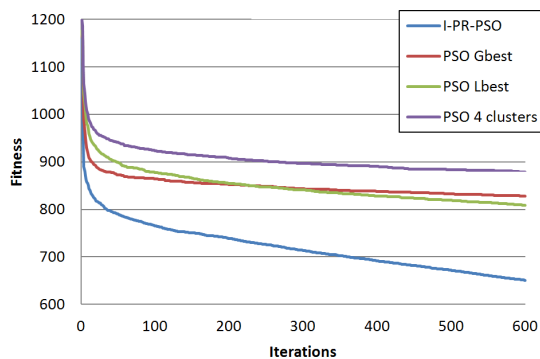


FIGURE 3.14 – Shifted Rastrigin

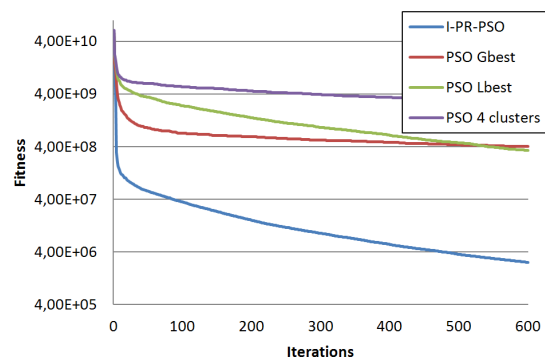


FIGURE 3.15 – Shifted Rosenbrock

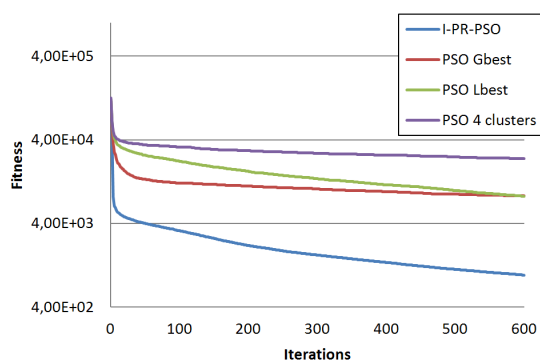


FIGURE 3.16 – Shifted Sphere

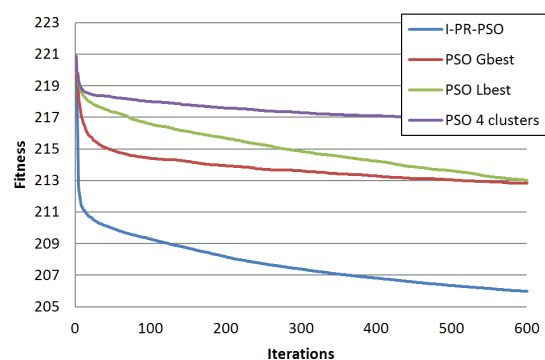


FIGURE 3.17 – Shifted Ackley

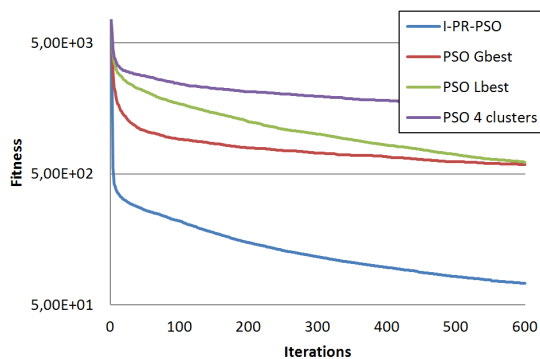


FIGURE 3.18 – Bohachevsky

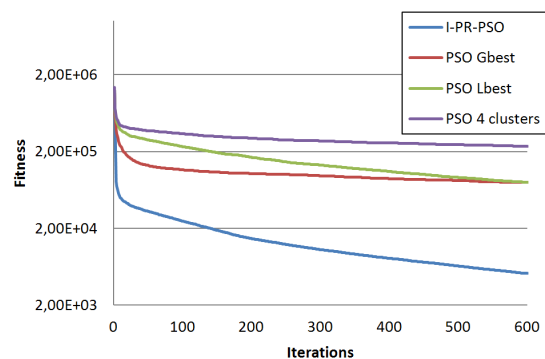


FIGURE 3.19 – Schwefel's problem 1.2

Figures 3.14 - 3.19 : courbes de convergence des calculs d'optimisation en dimension 50 (moyenne des 100 calculs réalisés)

objectif testées. Les courbes de convergence obtenues en dimensions 10, 20 et 30 sont données en Annexe B. Ces graphiques représentent les moyennes des valeurs obtenues pour chacune des 600 itérations des 100 calculs d'optimisation réalisés. Par souci de lisibilité des résultats, certaines de ces courbes sont présentées en échelle logarithmique. Comme on peut le voir sur ces figures, les résultats sont cohérents avec les précédents. C'est à dire que I-PR-PSO est plus performant que ses pairs, notamment en terme de vitesse de convergence. En effet, I-PR-PSO présente une tendance à se rapprocher plus rapidement de la zone prometteuse contenant le minimum global de la fonction objectif,

puis converge plus rapidement vers ce dernier. On remarque que pour certaines fonctions, alors que le minimum de la fonction objectif vaut 0, le résultat du calcul d'optimisation en est encore très éloigné. Ceci s'explique par la très grande dimension du problème à traiter. Rappelons ici que l'objectif d'un processus d'optimisation est bel et bien de trouver une *meilleure* solution que celles trouvées précédemment par d'autres algorithmes, quand bien même elle ne serait pas la meilleure des meilleures.

3.4.3/ POUR CONCLURE

Nous avons vu dans cette section le développement de notre nouvel algorithme I-PR-PSO, basé conjointement sur l'algorithme PSO et l'algorithme PageRank inverse, dont les bases théoriques ont été initialement développées par Brin et Page, les fondateurs de Google. Nous avons montré que cet algorithme est très efficace sur un benchmark de 12 fonctions mathématiques, tant en terme de valeur de la fonction objectif atteinte, qu'en terme de vitesse de convergence.

Cependant, ces essais numériques ne considèrent pas de problèmes d'optimisation avec contraintes. Or, comme il l'a été évoqué dans le Chapitre 2, un algorithme peut n'être efficace que sur un ou plusieurs types de problèmes donnés. La définition des variables (continues ou discrètes), la dimension du problème, et les contraintes d'optimisation peuvent altérer le fonctionnement d'un algorithme, ou en tout cas, le rendre moins efficace.

C'est la raison pour laquelle, afin de valider complètement l'utilisation de I-PR-PSO, il est nécessaire de l'appliquer à des problèmes d'optimisation avec contraintes.

3.5/ VALIDATION EN OPTIMISATION AVEC CONTRAINTES : APPLICATION AU DIMENSIONNEMENT OPTIMAL DES TREILLIS

3.5.1/ PRÉSENTATION DU PROBLÈME

Comme souvent dans le domaine de la mécanique, et comme nous l'avons vu dans la partie 2.1.3 à la page 17, le but de l'optimisation des structures est souvent de minimiser le volume, en considérant une contrainte sur la compliance de la structure, ou l'inverse. Ici, afin d'appliquer I-PR-PSO à un problème contraint en mécanique des structures, nous avons décidé de travailler avec un benchmark de treillis de barres, très utilisé dans la littérature [120, 224].

Le problème d'optimisation choisi dans ce cas est la minimisation de la masse du treillis, en considérant une contrainte d'optimisation sur les contraintes mécaniques maximales admissibles dans les barres et les déplacements des noeuds. Les variables d'optimisation sont les aires des sections des barres. Considérant qu'une particule est une potentielle solution au problème d'optimisation posé, une particule sera ainsi définie comme un vecteur dont le nombre de composantes est égal au nombre de barres dans le treillis. Ainsi, chaque composante de ce vecteur contiendra une potentielle valeur de l'aire de la section de la barre considérée.

3.5.2/ BENCHMARK DE TREILLIS ET PARAMÈTRES DU CALCUL

Les deux treillis testés contenant respectivement 10 et 22 barres sont tirés du travail de Asl et al. [230].

3.5.2.1/ TREILLIS À 10 BARRES

Le treillis considéré est représenté en figure 3.20 suivante. Les sections minimales et maximales des barres sont données dans le tableau 3.4, et les efforts appliqués sont de -100 kips dans la direction y appliqués aux noeuds (2) et (4). Les contraintes mécaniques maximales admissibles dans les barres sont de ± 25 ksi.

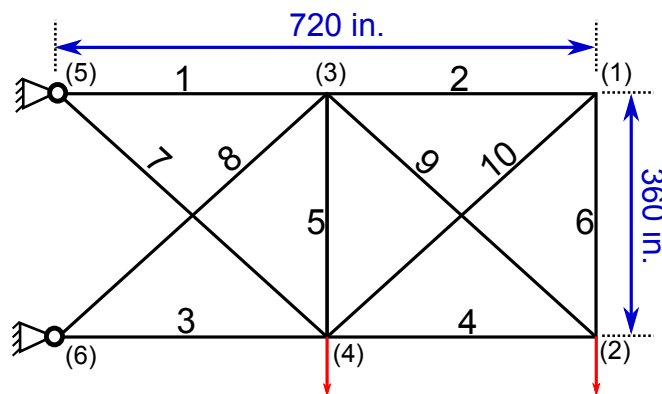


FIGURE 3.20 – Treillis à 10 barres

Variable	Sections	Section min. (in ²)	Section max. (in ²)
1	A_1	0.1	35
2	A_2	0.1	35
3	A_3	0.1	35
4	A_4	0.1	35
5	A_5	0.1	35
6	A_6	0.1	35
7	A_7	0.1	35
8	A_8	0.1	35
9	A_9	0.1	35
10	A_{10}	0.1	35

Tableau 3.4 – Sections minimales et maximales admissibles du treillis à 10 barres

3.5.2.2/ TREILLIS À 22 BARRES

Le treillis considéré est donné en figure 3.21 suivante. Les sections minimales et maximales des barres sont données dans le tableau 3.5. Trois cas de chargement différents ont été traités et sont récapitulés dans le tableau 3.6. Les contraintes mécaniques maximales admissibles dans les barres sont données dans le tableau 3.7 suivant.

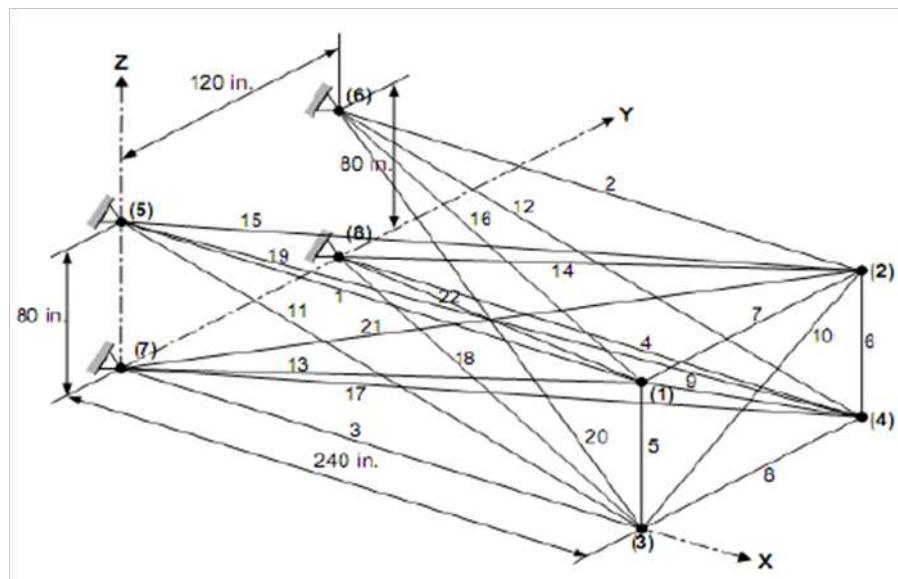


FIGURE 3.21 – Treillis à 22 barres

Variable	Sections	Section min. (in ²)	Section max. (in ²)
1	$A_1 = A_2 = A_3 = A_4$	0.1	3
2	$A_5 = A_6$	0.1	3
3	$A_7 = A_8$	0.1	3
4	$A_9 = A_{10}$	0.1	3
5	$A_{11} = A_{12} = A_{13} = A_{14}$	0.1	3
6	$A_{15} = A_{16} = A_{17} = A_{18}$	0.1	3
7	$A_{19} = A_{20} = A_{21} = A_{22}$	0.1	3

Tableau 3.5 – Sections minimales et maximales admissibles du treillis à 22 barres

Noeud	Chargement 1	Chargement 2	Chargement 3
Direction	P_x, P_y, P_z	P_x, P_y, P_z	P_x, P_y, P_z
1	-20.0 ; 0.0 ; -5.0	-20.0 ; -5.0 ; 0.0	-20.0 ; 0.0 ; 35.0
2	-20.0 ; 0.0 ; -5.0	-20.0 ; -50.0 ; 0.0	-20.0 ; 0.0 ; 0.0
3	-20.0 ; 0.0 ; -30.0	-20.0 ; -50.0 ; 0.0	-20.0 ; 0.0 ; -35.0

Tableau 3.6 – Définition des trois cas de chargement du treillis à 22 barres

Variable	Sections	Compression max (ksi)	Traction max (ksi)
1	$A_1 \rightarrow A_4$	24.0	36.0
2	$A_5 \rightarrow A_6$	30.0	36.0
3	$A_7 \rightarrow A_8$	28.0	36.0
4	$A_9 \rightarrow A_{10}$	26.0	36.0
5	$A_{11} \rightarrow A_{14}$	22.0	36.0
6	$A_{15} \rightarrow A_{18}$	20.0	36.0
7	$A_{19} \rightarrow A_{22}$	18.0	36.0

Tableau 3.7 – Limites des contraintes mécaniques admissibles dans les barres du treillis à 22 barres

3.5.2.3/ PARAMÈTRES MÉCANIQUES ET NUMÉRIQUES

Les paramètres mécaniques et numériques communs aux deux problèmes sont donnés dans le tableau 3.8. Dans les deux cas, les aires des sections initiales sont données de manière aléatoire lors de l'initialisation de l'algorithme d'optimisation.

Treillis	10 barres	22 barres
Module d'Young (ksi)	10^7	10000
Densité de matière (lbm/in ³)	0.1	0.1
Limite élastique (psi)	± 25000	± 25000
Limite des déplacements des noeuds (in.)	± 2	± 2

Tableau 3.8 – Paramètres mécaniques des treillis traités

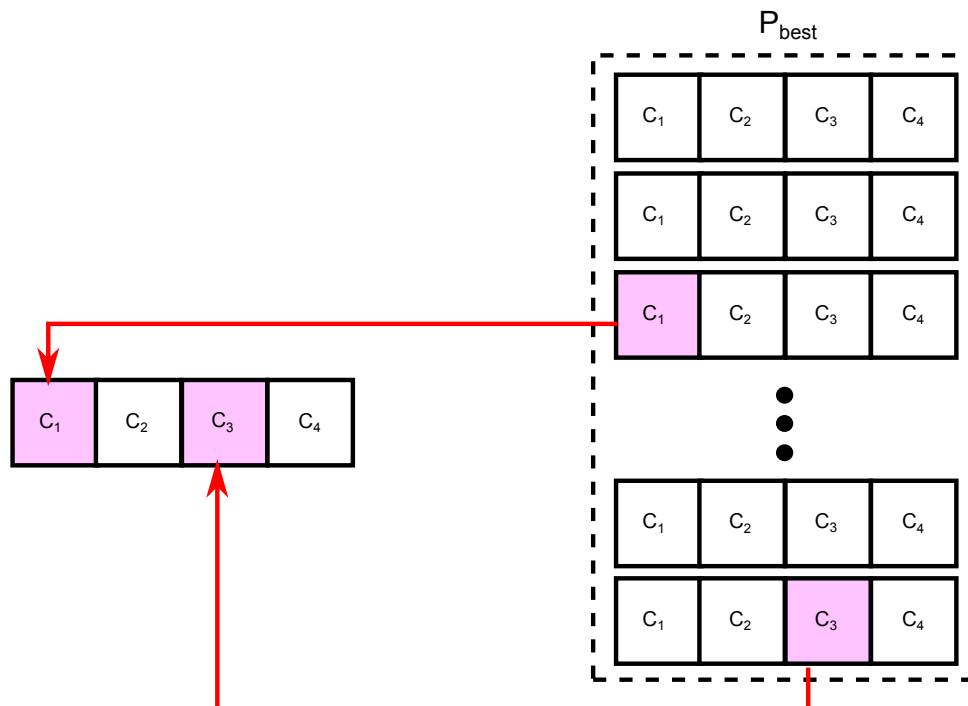
Concernant les paramètres de I-PR-PSO, le nombre maximal d'itérations a été fixé à 1000, le paramètre d'inertie ω a été choisi de manière à décroître linéairement entre 1.2 et 0.4 au cours des itérations, les paramètres c_1 et c_2 valent 2, et le nombre maximal d'itérations du calcul de la matrice stochastique des transitions gérant la topologie de la population PSO est de 6000. 50 particules PSO ont été utilisées. Les calculs d'optimisation ont été réalisés avec le logiciel MATLAB ®, et le calcul de la fonction objectif a été réalisé avec ANSYS Mechanical ®. Une méthode d'interfaçage de ces deux logiciels a donc été développée, afin que tous les calculs soient réalisés de manière entièrement automatisée.

3.5.3/ GESTION DES CONTRAINTES D'OPTIMISATION

Il existe deux types de contraintes d'optimisation à gérer dans ce cas : les contraintes que l'on peut tester avant d'avoir réalisé le calcul de la fonction objectif - comme par exemples les sections minimales et maximales des barres - que l'on appellera les bornes limites des variables d'optimisation, et celles pour lesquelles le calcul éléments finis de la structure est nécessaire - comme par exemple les contraintes mécaniques dans les barres et les déplacements des nœuds - que l'on appellera les contraintes intrinsèques au problème.

3.5.3.1/ LES BORNES LIMITES DES VARIABLES D'OPTIMISATIONS

Les bornes limites des variables d'optimisation ne nécessitent pas le calcul de la fonction objectif, ou un quelconque calcul éléments finis pour être testées. Pour traiter ce type de contraintes d'optimisation, nous avons décidé d'utiliser la méthode *Harmony Search scheme* utilisée notamment par Xu et al. [224]. Cette méthode a initialement été proposée par Li et al. [173] dans le but de gérer les particules qui, en se déplaçant dans le domaine des solutions, sortaient du domaine de recherche. Cette méthode provient de l'algorithme *Harmony search* (HS) [146], dans lequel une mémoire appelée *Harmony Memory* (HM) est utilisée. Cette mémoire conserve les vecteurs de variables admissibles ou acceptables. Ainsi, dès lors qu'une variable sort de son domaine de recherche, elle est remplacée par une valeur choisie aléatoirement dans la HM. De la même manière, PSO conserve en mémoire des valeurs *a priori* acceptables, ou tout du moins meilleures que les particules de l'essaim : ce sont les meilleures valeurs personnelles atteintes par les particules conservées dans les vecteurs P_i^{best} . Ainsi, en appliquant l'algorithme HS à PSO, nous pourrions remplacer les valeurs des variables sorties de leur domaine de recherche par des valeurs choisies de manière aléatoire dans les mémoires personnelles des particules. Toutes les variables n'ont pas le même domaine de recherche, il est donc nécessaire de choisir, pour chaque variable, la composante du vecteur P_i^{best} lui correspondant, comme on le voit dans la figure 3.22. Dans cette figure, on considère que l'on travaille en dimension 4, c'est à dire que chaque particule est un vecteur à 4 composantes. On considère que les composantes C_1 et C_3 ne respectent pas leurs contraintes d'optimisation respectives et que donc elles sont situées en dehors de leur domaine de recherche.

FIGURE 3.22 – Figure récapitulative de la méthode *Harmony Memory*

3.5.3.2/ LES CONTRAINTES INTRINSÈQUES AU PROBLÈME

Comme leur nom l'indique, les contraintes d'optimisation intrinsèques au problème dépendent du problème à résoudre. Dans notre cas, il est donc nécessaire d'effectuer un calcul par éléments finis du modèle de treillis pour connaître leur valeur et pouvoir les tester. Les déplacements des nœuds et les contraintes mécaniques dans les barres sont des exemples de ce type de contraintes d'optimisation.

Pour traiter ce type de contraintes d'optimisation, nous utilisons une méthode de pénalisation de la fonction objectif. On construit ainsi une seule fonction objectif qui tient d'ores et déjà compte des contraintes d'optimisation. Cette méthode de pénalisation à plusieurs étapes a été initialement développée par Parsopoulos et al. [116] pour la gestion des contraintes d'optimisation de PSO. La fonction de pénalisation est définie dans l'équation (3.11) suivante

$$F(X) = f(X) + h(k)H(X) \quad (3.11)$$

où X est une particule de l'essaim, $f(X)$ est la fonction objectif initiale à traiter, $h(k)$ est la valeur de la pénalité qui est modifiée en fonction de l'évolution de l'algorithme, telle que $h(k) = \sqrt{k}$ et $H(X)$ est un facteur de pénalité défini tel que dans l'équation (3.12) suivante

$$H(X) = \sum_{i=1}^m \theta(q_i(X)) q_i(X)^{\gamma(q_i(X))} \quad (3.12)$$

où $q_i(X)$ est une fonction définissant la relative violation de la contrainte d'optimisation $g_i(X)$ définie par $q_i(X) = \max(0, g_i(X))$, $\theta(q_i(X))$ amplifie la valeur de la pénalité en fonction de la relative violation $q_i(X)$, et $\gamma(q_i(X))$ est la puissance d'amplification de la fonction de pénalité.

Nous proposons ici de modifier les valeurs du facteur $\theta(q_i(X))$. En effet, les valeurs initialement données dans [116] nous ont donné des résultats encourageants, mais pour lesquels les contraintes d'optimisation étaient systématiquement violées. Ainsi, nous choisissons de pénaliser plus fortement notre fonction objectif afin d'éviter tout dépassement des contraintes d'optimisation. L'algorithme 14 suivant nous donne les méthodes employées afin de calculer les paramètres γ et θ .

3.5.4/ RÉSULTATS OBTENUS

3.5.4.1/ TREILLIS À 10 BARRES

Les résultats, c'est à dire les valeurs des variables d'optimisation et la valeur de la masse du treillis, concernant le treillis à 10 barres sont donnés dans le tableau 3.9 suivant. En comparant ce tableau au tableau 3.4 précédent, on voit que toutes les contraintes des variables limites sont respectées. Les contraintes mécaniques dans les barres et les déplacements des nœuds sont récapitulés respectivement dans les tableaux 3.10 et 3.11 suivants. Comme on le voit dans ces tableaux, les contraintes d'optimisation concernant les contraintes mécaniques dans les barres sont vérifiées, alors que deux des 10 contraintes d'optimisation concernant les déplacements des nœuds sont très légèrement violées.

Algorithm 14 Algorithme permettant de calculer les paramètres γ et θ

```

if  $q_i(X) < 1$  then
   $\gamma(q_i(X)) = 1$ 
else
   $\gamma(q_i(X)) = 2$ 
end if
if  $q_i(X) < 0.001$  then
   $\theta(q_i(X)) = 10000$ 
else if  $q_i(X) \leq 0.1$  then
   $\theta(q_i(X)) = 20000$ 
else if  $q_i(X) \leq 1$  then
   $\theta(q_i(X)) = 100000$ 
else
   $\theta(q_i(X)) = 300000$ 
end if

```

Ces résultats sont tout de même considérés comme acceptables dans la littérature, où certaines contraintes sont légèrement violées pour accepter une meilleure solution [230].

Variables (in2)	Schmit et al. [10]	Schmit et Miura [14]	Venkayya [8]	Notre étude	Sedaghati [152]
A_1	33.43	30.67	30.42	31.13	30.52
A_2	0.1	0.1	0.128	0.10	0.1
A_3	24.26	23.76	23.41	23.39	23.19
A_4	14.26	14.59	14.91	15.68	15.22
A_5	0.1	0.1	0.101	0.1	0.1
A_6	0.1	0.1	0.101	0.45	0.55
A_7	8.388	8.578	8.696	7.52	7.45
A_8	20.74	21.07	21.08	20.84	21.03
A_{10}	0.1	0.1	0.186	0.10	0.1
Masse (lb)	5089.0	5076.85	5084.9	5065.84	5060.58

Tableau 3.9 – Résultats obtenus sur le treillis à 10 barres

Barre	Contrainte mécanique (ksi)
1	6.7702
2	-1.1471
3	-8.4049
4	-6.4388
5	24.9244
6	-1.1423
7	18.8227
8	-7.1778
9	6.4278
10	1.6702

Tableau 3.10 – Contraintes mécaniques dans les barres du treillis à 10 barres

Nœud	Déplacement selon l'axe y (mm)
1	-2.0426
2	-2.0014
3	-0.76054
4	-1.6578

Tableau 3.11 – Déplacements des noeuds du treillis à 10 barres

3.5.4.2/ TREILLIS À 22 BARRES

Les résultats concernant les trois cas de chargement du treillis à 22 barres sont donnés dans le tableau 3.12 suivant. Les contraintes mécaniques dans les barres et les déplacements des noeuds sont donnés respectivement dans les tableaux 3.13 et 3.14. On ne redonne pas ici les déplacements des noeuds 5, 6, 7 et 8 qui sont encastrés. Comme on le voit dans les tableaux 3.13 et 3.14, aucune des contraintes d'optimisation n'est violée. En outre, les résultats sont bien meilleurs que ceux de la littérature, dans les trois cas. Le tableau 3.15 compare nos résultats avec ceux de la littérature, dans le 1er cas de chargement.

Variable (in2)	Cas 1	Cas 2	Cas 3
A_1	2.9989	2.1465	0.60027
A_2	1.4992	1.1947	2.6978
A_3	0.10031	2.0717	0.5969
A_4	0.10044	1.0714	1.4318
A_5	2.9989	0.90091	2.1599
A_6	1.5881	2.2245	2.3347
A_7	1.6987	1.8172	0.14755
Masse (lb)	978.1	849.25	642.0941

Tableau 3.12 – Résultats obtenus sur le treillis à 22 barres

3.5.5/ POUR CONCLURE

En conclusion, l'algorithme I-PR-PSO a fait ses preuves dans le domaine de l'optimisation avec contraintes, appliquée en particulier à la mécanique. En effet, I-PR-PSO a donné des résultats aussi bons, sinon meilleurs que ceux proposés dans la littérature, en terme de fonction objectif et de respect des contraintes d'optimisation. Le couplage de I-PR-PSO avec d'autres méthodes de gestion des contraintes d'optimisation est nécessaire, et permet de gérer correctement ces contraintes. Ceci dit, les contraintes d'optimisation sont souvent gérées de manière plus simple, en utilisant une méthode classique de pondération des objectifs et des contraintes. Ce couplage entre I-PR-PSO, la méthode *Harmony Memory* de gestion des variables limites et d'une méthode de pénalisation donnent des résultats encourageants.

3.6/ CONCLUSIONS DU CHAPITRE

Le nouvel algorithme d'optimisation Inverse-PageRank-PSO que nous avons proposé est basé conjointement sur l'algorithme d'optimisation par essaim particulaire PSO, et sur l'algorithme PageRank - utilisé notamment par le moteur de recherche Google - dans sa version inverse. Cette méthode permet notamment de pondérer l'influence des particules les unes sur les autres de manière intelligente, car elle s'adapte au succès relatif de l'algorithme au cours des itérations. Le développement de I-PR-PSO ainsi que l'étude de ses capacités de convergence sur un benchmark de fonctions mathématiques représentent notre contribution originale concernant les aspects numériques de la thèse. En outre, afin de valider cet algorithme dans le domaine de l'optimisation avec contraintes, il a été appliqué au dimensionnement optimal des treillis. Dans ce cas, différents types de contraintes d'optimisation ont été considérés : les contraintes des variables limites, et les contraintes d'optimisation intrinsèques au problème pour lesquelles le calcul numérique du problème mécanique considéré est nécessaire. L'application de I-PR-PSO à un benchmark de fonctions mathématiques et à des problèmes mécaniques montrent sa supériorité par rapport à ses congénères, notamment en terme de vitesse de convergence et de capacité à se rapprocher au plus près du minimum global de la fonction objectif considérée.

Déplacements des noeuds (in.)									
Noeud	Cas 1			Cas 2			Cas 3		
	u_x	u_y	u_z	u_x	u_y	u_z	u_x	u_y	u_z
1	0.2271	-7.87E-2	-1.9216	-0.5287	-1.8309	5.22E-3	-0.1552	0.6637	1.1745
2	0.2271	7.87E-2	-1.9216	0.3751	-1.9116	2.38E-2	-8.43E-2	0.6787	-1.1207
3	-0.3803	-4.11E-2	-2	-0.5287	-1.8309	-5.22E-3	-8.43E-2	-0.6787	1.1207
4	-0.3803	4.11E-2	-2	0.3751	-1.9116	-2.38E-2	-0.1552	-0.6637	-1.1745

Tableau 3.14 – Déplacements des noeuds du treillis à 22 barres

Variables (in3)	Scheu et Schmit [9]	Khan et Willmert [18]	Farshi et Ziazi [196]	Notre étude
A_1	2.629	2.563	2.6250	2.9989
A_2	1.162	1.553	1.2164	1.4992
A_3	0.343	0.281	0.3466	0.10031
A_4	0.423	0.512	0.4161	0.10044
A_5	2.782	2.626	2.7732	2.9989
A_6	2.173	2.131	2.0870	1.5881
A_7	1.952	2.213	2.0314	1.6987
Masse (lb)	1024.8	1034.74	1023.98	978.1

Tableau 3.15 – Comparaison des résultats avec ceux de la littérature. Treillis à 22 barres, cas de chargement $n^o 1$

Contraintes mécaniques dans les barres (ksi)			
Barre	Cas 1	Cas 2	Cas 3
1	9.4659	-22.0292	-6.4668
2	9.4659	15.6292	-3.5141
3	-15.8457	-22.0292	-3.5141
4	-15.8457	15.6292	-6.4668
5	9.8072	1.3043	6.7256
6	9.8072	5.9551	6.7256
7	13.1114	-6.7252	1.2518
8	6.8459	-6.7252	1.2518
9	9.9259	-3.5390	13.7578
10	9.9259	-3.5390	-7.8857
11	10.7389	-19.7611	-17.1713
12	10.7389	14.364	8.8609
13	-15.5000	-19.7611	8.8609
14	-15.5000	14.3640	-17.1713
15	8.8838	-19.3571	8.5017
16	8.8838	12.892	-16.2361
17	-11.9920	-19.3571	-16.2361
18	-11.9920	12.8920	8.5017
19	9.3951	-17.5339	-2.9261
20	9.3951	11.8929	-3.6279
21	-11.4490	-17.5339	-3.6279
22	-11.4490	11.8929	-2.9261

Tableau 3.13 – Contraintes mécaniques dans les barres pour les treillis à 22 barres

DÉVELOPPEMENT D'UNE MÉTHODE D'OPTIMISATION METAHEURISTIQUE HYBRIDE : APPLICATION À L'OPTIMISATION TOPOLOGIQUE EN MÉCANIQUE

Sommaire

4.1	Introduction	126
4.2	Définition du nouvel algorithme ESO-I-PR-PSO	126
4.2.1	Binarisation de l'algorithme I-PR-PSO	126
4.2.2	Analyse des damiers dans la topologie	128
4.2.3	Algorithme ESO-I-PR-PSO global	131
4.2.4	Définition des problèmes d'optimisation à résoudre	132
4.3	Choix des paramètres de I-PR-PSO	134
4.3.1	Analyse des paramètres de confiance c_1 et c_2	135
4.3.2	Analyse du paramètre d'inertie ω	137
4.3.3	Analyse du nombre de particules de l'essaim	140
4.3.4	Conclusions de l'étude paramétrique	142
4.4	Problèmes traités, résultats et discussion	142
4.5	Conclusions du chapitre	154

4.1/ INTRODUCTION

Alors que l'algorithme d'optimisation par essaim particulaire PSO a été développé à la fin des années 80, ses adaptations à l'optimisation topologique - et appliquée en particulier à la mécanique - restent très limitées. Bien que de nombreux algorithmes aient été proposés dans la littérature, rares sont ceux qui proposent des solutions dont les topologies sont régulières et symétriques lorsque le problème mécanique s'y prête [208].

Ce chapitre a pour but de proposer un nouvel algorithme hybride, basé conjointement sur la méthode I-PR-PSO précédemment développée dans le chapitre 3 et l'algorithme Evolutionary Structural Optimization (ESO) [1, 106, 200, 241] que nous avons présenté dans le chapitre 2 à la page 21. Nous verrons notamment comment ces deux algorithmes peuvent se guider l'un l'autre, afin de converger ensemble vers une solution régulière et, si le problème l'est, symétrique. En effet, alors que l'algorithme PSO a tendance à trouver des solutions optimales, elles ne sont pas toujours régulières, alors que ESO converge systématiquement vers des solutions régulières, qui peuvent ne pas être les meilleures [106, 167, 200] si les paramètres de l'algorithme sont mal choisis au départ. Finalement, c'est en combinant les avantages des deux algorithmes que nous montrerons que l'on peut trouver des solutions optimales, en minimisant en plus le temps de calcul nécessaire. En outre, ces deux algorithmes considéreront des objectifs différents. Alors que l'un minimisera la compliance, l'autre minimisera le volume. Bien que ces objectifs soient apparemment contradictoires, notre algorithme les optimisera de manière séquentielle jusqu'à convergence.

Après une étude paramétrique permettant de déterminer les paramètres de notre nouvel algorithme ESO-I-PR-PSO les plus adaptés à la résolution de nos problèmes d'optimisation topologique, notre algorithme sera testé et validé sur un benchmark de 7 problèmes mécaniques classiquement proposés dans la littérature.

4.2/ DÉFINITION DU NOUVEL ALGORITHME ESO-I-PR-PSO

4.2.1/ BINARISATION DE L'ALGORITHME I-PR-PSO

Comme nous l'avons vu précédemment dans le chapitre 2.2, à la page 54, la méthode principale permettant de rendre l'algorithme PSO binaire consistait à utiliser une fonction Sigmoidale transformant les valeurs des vitesses ou des positions des particules en valeurs binaires. Développé par Kennedy et al. [73], la version binaire de PSO - appelée BPSO dans la littérature - considère que la vitesse des particules peut être vue comme la probabilité pour chaque bit du vecteur position de changer d'état, c'est à dire de passer de l'état 0 à l'état 1 et inversement. En effet, dans un espace binaire, la vitesse peut être vue comme le nombre de bits changés par itération, c'est à dire la distance de Hamming entre l'état d'une particule à l'itération n et l'état de la même particule à l'itération $n + 1$. La méthode proposée par Kennedy et al. [73] est donnée dans l'équation suivante

$$\text{if } (rand() \leq S(v)) \text{ alors } x = 1 \text{ sinon } x = 0 \quad (4.1)$$

où $S(v)$ représente la fonction Sigmoidale employée. Plus tard, Lee et al. [184] ont proposé d'utiliser une représentation génotype/phénotype, empruntée au fonctionnement des algorithmes génétiques. Ainsi, ce n'est plus la vitesse des particules mais leur position qui

est ainsi transformée en valeur binaire par la fonction Sigmoidé. On considère donc ici deux types de paramètres, les paramètres génotypiques, comme la vitesse V_g et la position X_g , contenus dans un espace continu, et la position phénotypique X_p qui contient les valeurs binaires définissant la topologie de la particule concernée. C'est donc la position génotypique X_g qui est décodée pour donner la position phénotypique comme on le voit dans l'équation suivante

$$X_{p,i,j}(k+1) = \begin{cases} 0 & \text{if } rand() \geq S(X_{g,i,j}(k+1)) \\ 1 & \text{if } rand() \leq S(X_{g,i,j}(k+1)) \end{cases} \quad (4.2)$$

$$S(X_{g,i,j}(k+1)) = \frac{1}{1 + e^{-X_{g,i,j}(k+1)}} \quad (4.3)$$

où i représente l'indice de la particule dans l'essaim, j est l'indice de la composante de cette particule à modifier, et k est l'itération en cours.

Nous proposons ici de simplifier ce mécanisme de passage du domaine continu au binaire. Dans toutes les versions de PSO, I-PR-PSO comprise, chaque particule - et sa position en particulier - est une potentielle solution au problème d'optimisation posé. Une version binaire de PSO appliquée à l'optimisation topologique devrait donc simplement décrire la répartition de matière dans la structure considérée, comme on le voit dans la figure 4.1 suivante. Dans notre nouvel algorithme, nous choisissons de nous affranchir de la phase de décodage de la position ou de la vitesse des particules avec une fonction Sigmoidé. Pour ce faire, la position de chaque particule est continue, et chacune de ses composantes est contenue dans le domaine $[-1, 1]$. Le vecteur position X_i^k de la particule i à l'itération k est alors décodé en une matrice de topologie notée T_i^k de la manière suivante, où n représente le nombre d'éléments dans le domaine de design, c'est à dire le nombre de composantes de la matrice T .

$$\begin{aligned} \forall i \in [0, n] \\ \text{si } X_i^k \geq 0 \text{ alors } T_i^k &= 1 \\ \text{sinon } T_i^k &= 0 \end{aligned} \quad (4.4)$$

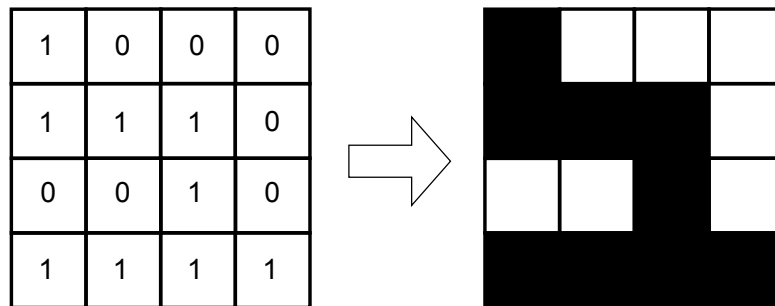


FIGURE 4.1 – Décodage de la matrice de topologie

Ainsi, la topologie sera définie dans le logiciel de calcul par éléments finis utilisé comme dans la figure 4.2 suivante. Les éléments correspondant aux bits valant 1 dans la matrice T_i^k contiendront du matériau alors que ceux dont correspondent aux bits nuls seront vides. Notons qu'il n'est pas nécessaire de supprimer complètement les éléments finis considérés comme vides du maillage de la structure. En effet, il a été montré dans la littérature [31, 45, 53, 69] que le "vide" peut être remplacé par un matériau fictif ([ghost](#)

material dans la littérature anglophone) dont le module d'Young est bien plus petit (10^{-2} ou 10^{-3}) que celui du matériau employé. Comme nous l'avons vu dans la partie 2.2.3, les vitesses des particules doivent être contraintes afin que le calcul puisse converger. Ces contraintes assurent notamment que les vitesses des particules ne sont pas trop grandes, ce qui pousserait les particules aux bords du domaine en quelques itérations, et engendrerait des problèmes de convergence prématurée [137, 172, 203]. Ainsi, ces contraintes concernant les vitesses des particules doivent être données en fonction des positions extrêmes des particules. Les vitesses continues seront ainsi contenues dans $[-V_{max}, V_{max}]$ avec $V_{max} = 10\% \times X_{max} = 0.1$.

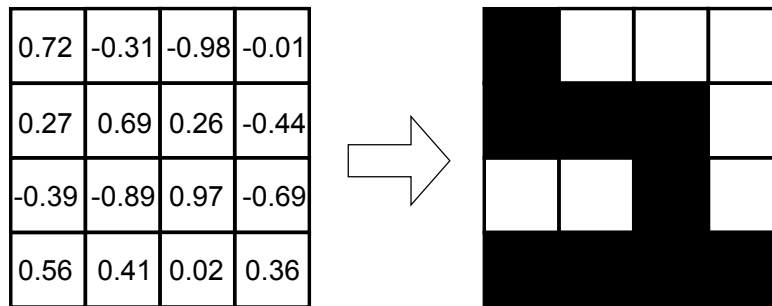


FIGURE 4.2 – Modification de la procédure de binarisation de I-PR-PSO

Ainsi, grâce à cette représentation simplifiée, les positions des particules n'ont pas besoin de décodage particulier : le signe de chacune de leurs composantes donne directement la répartition de matière dans la structure étudiée. Si la composante de la matrice T est négative, l'élément lui correspondant contiendra du matériau fictif, dans le cas contraire, il contiendra du matériau classique.

4.2.2/ ANALYSE DES DAMIERS DANS LA TOPOLOGIE

Une fois que les particules sont définies par la méthode précédemment décrite, chacune des topologies est analysée afin de vérifier qu'elle ne contient pas de damiers, et que tous les éléments finis contenant du matériau sont connectés entre eux. Deux éléments sont considérés comme connectés s'ils ont un côté en commun, alors qu'ils sont considérés comme déconnectés s'il ne partagent qu'un nœud. Du fait que nous utilisons un matériau fictif, ou *ghost material*, le calcul éléments finis aboutira certainement vers une solution, mais cet artifice numérique ne dispense pas la structure d'être instable mécaniquement parlant [208]. Ainsi, nous avons développé un algorithme qui modifiera la topologie jusqu'à ce qu'elle soit continue, et qu'elle ne contienne plus de damiers, comme on le voit dans la figure 4.3 suivante. Seuls des éléments sont ajoutés à la topologie, aucun ne peut être supprimé. En effet, puisqu'en mécanique nous considérons souvent soit la minimisation du volume de la structure, soit des contraintes d'optimisation sur le volume, nous préférons nous placer dans le cas le plus défavorable. L'ajout d'éléments diminuera certainement un peu les performances de l'algorithme, mais permettra de nous assurer que les résultats ne sont pas biaisés par la suppression d'éléments pouvant être utiles au fonctionnement mécanique de la structure.

Notons ici qu'il n'est jamais précisé dans la littérature de quelle manière PSO garde en mémoire les modifications qui sont faites par ce biais dans la topologie. Souvent la mémoire n'est pas réactualisée, comme c'est le cas dans le travail de Jiakela et al. [95], où il est dit "*connectivity analysis does not modify corresponding chromosome allele values*".

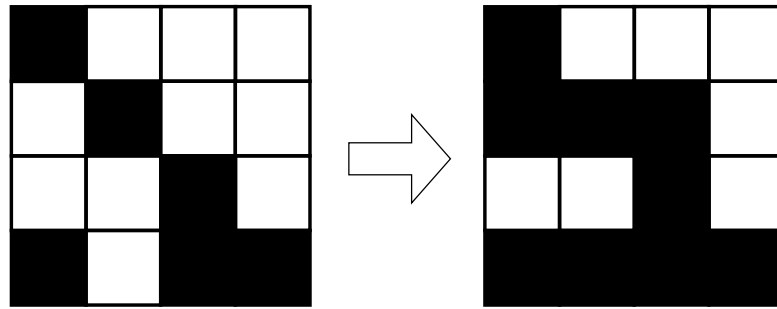


FIGURE 4.3 – Procédure de modification des damiers dans la topologie

C'est à dire que la position des particules n'est pas modifiée, et l'algorithme ne garde pas en mémoire les modifications effectuées par l'analyse des damiers de la topologie. Dans ESO-I-PR-PSO, pour chaque bit modifié, nous modifions aussi sa valeur correspondante dans le vecteur position de la particule considérée. Ainsi, pour chaque bit passant de 0 à 1, sa valeur correspondante dans X_i^k sera égale à X_{max} .

Notre algorithme "scanne" la matrice de topologie T_i^k en analysant les bits 4 à 4. Sont considérés comme des damiers les deux groupements de 4 bits présentés dans la figure 4.4. Afin que tous les damiers puissent être éliminés de la topologie, la structure doit être scannée méthodiquement, comme on le voit dans l'algorithme 15 que nous avons développé, où $size_x$ représente le nombre d'éléments finis de la structure dans la direction x et $size_y$ représente le nombre d'éléments finis de la structure dans la direction y .

0	1	1	0
1	0	0	1

FIGURE 4.4 – Groupements de bits de la matrice de topologie donnant lieu à des damiers dans la structure

Pour plus de clarté, cet algorithme est représenté de manière graphique dans la figure 4.5 suivante. L'algorithme 15 correspond à vérifier d'abord le groupement de bits rouge, ainsi que toutes ses translations dans la matrice T par pas de deux bits, puis le groupement de bits vert ainsi que toutes ses translations, puis le groupement de bits bleu et toutes ses translations, puis le groupement violet et toutes ses translations. Si un damier tel que décrit dans la figure 4.4 précédente est trouvé par l'algorithme, l'une des deux composantes nulles du groupement est choisie aléatoirement et mutée en un 1.

Algorithm 15 Algorithme de vérification des damiers dans la topologie

```

for chaque particule  $i$  de l'essaim do
  for  $m=1 : 2 : size_y-1$  do
    for  $n=1 : 2 : size_x-1$  do
      Vérification des bits  $T_{m,n}, T_{m+1,n}, T_{m,n+1}, T_{m+1,n+1}$  de la matrice  $T_i$ .
      if Le groupement de bits présente un damiers tel que présenté dans la figure
      4.4 then
        Modification aléatoire d'un des bits nuls du groupement d'élément  $0 \Rightarrow 1$ 
        Mise à jour du bit correspondant dans le vecteur vitesse  $\Rightarrow X_{max}$ 
      end if
    end for
  end for
  for  $m=1 : 2 : size_y-1$  do
    for  $n=2 : 2 : size_x-2$  do
      Vérification des bits  $T_{m,n}, T_{m+1,n}, T_{m,n+1}, T_{m+1,n+1}$  de la matrice  $T_i$ .
      if Le groupement de bits présente un damiers tel que présenté dans la figure
      4.4 then
        Modification aléatoire d'un des bits nuls du groupement d'élément  $0 \Rightarrow 1$ 
        Mise à jour du bit correspondant dans le vecteur vitesse  $\Rightarrow X_{max}$ 
      end if
    end for
  end for
  for  $m=2 : 2 : size_y-2$  do
    for  $n=2 : 2 : size_x-2$  do
      Vérification des bits  $T_{m,n}, T_{m+1,n}, T_{m,n+1}, T_{m+1,n+1}$  de la matrice  $T_i$ .
      if Le groupement de bits présente un damiers tel que présenté dans la figure
      4.4 then
        Modification aléatoire d'un des bits nuls du groupement d'élément :  $0 \Rightarrow 1$ 
        Mise à jour du bit correspondant dans le vecteur vitesse :  $\Rightarrow X_{max}$ 
      end if
    end for
  end for
  for  $m=2 : 2 : size_y-2$  do
    for  $n=1 : 2 : size_x-1$  do
      Vérification des bits  $T_{m,n}, T_{m+1,n}, T_{m,n+1}, T_{m+1,n+1}$  de la matrice  $T_i$ .
      if Le groupement de bits présente un damiers tel que présenté dans la figure
      4.4 then
        Modification aléatoire d'un des bits nuls du groupement d'élément  $0 \Rightarrow 1$ 
        Mise à jour du bit correspondant dans le vecteur vitesse  $\Rightarrow X_{max}$ 
      end if
    end for
  end for
end for

```

4.2.3/ ALGORITHME ESO-I-PR-PSO GLOBAL

Cette partie présente le fonctionnement de l'algorithme ESO-I-PR-PSO global. Compte tenu des considérations précédentes concernant la gestion des damiers dans la topologie, nous sommes maintenant en mesure de décrire le fonctionnement global de l'algorithme ESO-I-PR-PSO, et en particulier de quelle manière les deux algorithmes ESO et I-PR-PSO communiquent et se guident l'un l'autre vers l'obtention d'une structure optimale, régulière et continue. Ici, nous choisissons de minimiser la compliance de la structure, approximée par le déplacement maximal du point d'application de la force, en considérant une contrainte d'optimisation sur le volume de la structure.

Comme il l'a été fait dans la littérature, et en particulier dans le travail de Luh et al. [208], l'initialisation des particules est réalisée avec une première itération ESO. En effet, il a été noté que, de par l'augmentation relativement rapide des vitesses des particules de PSO, cet algorithme a plus de facilités à rajouter des éléments à la topologie qu'à enlever. Pour ce faire, les topologies initiales sont considérées comme pleines, c'est à dire que tous les éléments contiennent du matériau. Chaque topologie est analysée par éléments finis, et l'on supprime un certain pourcentage des éléments les moins contraints de la structure. Le pourcentage d'éléments supprimés lors de cette première itération ESO pourra varier en fonction du problème à traiter.

Après cette première phase d'initialisation, l'algorithme I-PR-PSO seul cherche une solution optimale au problème d'optimisation posé, pour laquelle le volume final est supérieur au volume imposé par la contrainte d'optimisation. On choisira, pour cette première étape de calcul, $(V/V_0)_{ini} = 80\%$. Ce calcul d'optimisation par I-PR-PSO est considéré comme convergé lorsque l'algorithme ne trouve plus de meilleure solution pendant 5 itérations consécutives, c'est à dire lorsque $G_{best}^{k+1} - G_{best}^k \leq 1E - 10$. On appelle alors l'algorithme ESO, qui va travailler de paire en séquentiel avec I-PR-PSO. Les $rr\%$ éléments les moins contraints de la structure sont supprimés (c'est à dire que le matériau les composant est remplacé par le *ghost material*), où rr est un pourcentage qui évolue au cours des itérations. Une fois cette itération ESO réalisée, on laisse à nouveau I-PR-PSO modifier la structure à la recherche d'une topologie optimale. L'algorithme ESO est rappelé à chaque fois que I-PR-PSO stagne pendant 5 itérations consécutives. Ce fonctionnement séquentiel est réalisé jusqu'à ce que les contraintes d'optimisation soient respectées, c'est à dire jusqu'à ce que le volume de la structure soit inférieur à un certain pourcentage du volume initial choisi par l'utilisateur.

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

FIGURE 4.5 – Représentation graphique de l'algorithme de vérification des damiers dans la topologie

A chaque fois que l'algorithme ESO est appelé, le pourcentage d'éléments à supprimer évolue de la manière suivante

$$rr = rr + evol_{rr} \quad (4.5)$$

où les valeurs initiales de rr et $evol_{rr}$ sont choisies par l'utilisateur en fonction du problème à traiter. Lorsque l'on s'approche de la topologie finale, c'est à dire lorsque le volume de la structure s'approche du volume imposé par la contrainte d'optimisation, on a $V/V_0 - V_{target} \leq 0.1$, et l'on décroît le pourcentage d'évolution du paramètre rr , en remplaçant $evol_{rr}$ par $evol_{rr} \leftarrow evol_{rr}/2$. Ainsi, l'algorithme peut raffiner la structure obtenue d'une manière plus lente, et converger au mieux vers le volume V_{target} demandé.

Un problème rencontré fréquemment lors de l'utilisation des méthodes metaheuristiques est que la vitesse de convergence dépend grandement de l'initialisation des particules. Il n'est donc pas possible de connaître à l'avance le nombre d'itérations nécessaire à la convergence. Il est souvent choisi un nombre d'itérations assez grand, afin d'être en mesure de s'assurer que le calcul aura convergé lorsque le nombre maximal d'itérations aura été atteint. Ici, nous proposons de choisir un nombre maximal d'itérations it_{max} assez petit (20 ou 30 itérations), mais de conserver la possibilité d'ajouter des itérations si le calcul n'est pas considéré comme convergé. Ainsi, si $V/V_{ini} - V_{target} \geq 0.1$, alors $it_{max} = it_{max} + 1$. L'algorithme ESO-I-PR-PSO global est récapitulé dans l'algorithme 16.

4.2.4/ DÉFINITION DES PROBLÈMES D'OPTIMISATION À RÉSOUDRE

Nous avons donc affaire à deux problèmes d'optimisation couplés, qui seront résolus de manière séquentielle. Le premier problème d'optimisation, résolu en premier lieu par l'algorithme I-PR-PSO, est défini de la manière suivante

$$\begin{aligned} \min f^T \mathbf{u} \quad (&\equiv \min d) \\ V &\leq \alpha V_0, \text{ avec } \alpha < 1 \end{aligned}$$

où d représente le déplacement maximal du point d'application de la force.

Alors que le deuxième problème d'optimisation résolu par ESO est le suivant

$$\begin{aligned} \min V \\ \sigma_{VM} &\leq \sigma_{VM}^* \end{aligned}$$

Ces deux problèmes d'optimisation sont en réalité contradictoires, puisque justement c'est généralement en augmentant le volume d'une structure qu'on minimise sa compliance. Et en effet, comme nous le verrons dans les résultats qui suivront, lorsque ESO résout le second problème d'optimisation, c'est à dire lorsque l'on minimise le volume, la compliance augmente légèrement. C'est la raison pour laquelle nous avons décidé d'imposer, pour le premier problème d'optimisation, $\alpha = 0.8$. Ainsi, la compliance minimisée trouvée par I-PR-PSO reste tout de même acceptable, même après la minimisation du volume imposée par le second problème d'optimisation.

Ces deux problèmes d'optimisation couplés pourraient ainsi être considérés comme un problème d'optimisation multiobjectif, et notre méthode comme une méthode qui minimise les différents objectifs de manière successive, comme c'est le cas dans [la méthode du compromis](#), [la méthode du but programmé](#) ou encore [la méthode d'ordonnancement](#)

lexicographique que nous avons vues dans la partie 2.1 à la page 13.

Algorithm 16 Algorithme ESO-I-PR-PSO global

Initialisation of ω , rr , $evol_{rr}$.

Initialisation des particules X_i^1 .

Initialisation aléatoire des vitesses des particules V_i^1 .

Transformation des vecteurs position X_i^1 en matrices de topologie T_i^1 .

Calcul par éléments finis. Itération ESO : Suppression des $rr_{ini}\%$ éléments les moins contraints de la topologie.

while $it_{PSO} \leq it_{max}$ **do**

 Actualisation de ω

 Calcul par éléments finis de toutes les particules définies par leurs matrices de topologie respectives.

if $G_{best}^{k+1} - G_{best}^k \leq 1E - 10$ pendant 5 itérations consécutives **then**

for chaque particule **do**

 Calcul de rr : $rr = rr + evol_{rr}$.

 Itération ESO :

 Suppression des $rr\%$ éléments les moins contraints.

 Mise à jour des positions correspondantes $X_{i,j}^k \forall i \forall j = X_{max}$.

 Analyse des damiers et de la connectivité.

 Calcul éléments finis de la nouvelle topologie.

end for

 Mise à jour de la mémoire de la meilleure particule de l'essaim G_{best} .

end if

 algorithme I-PR-PSO : Mise à jour de toutes les positions X_i^k et vitesses V_i^k via l'eq. (3.8).

 Mise à jour de la mémoire de la meilleure particule de l'essaim G_{best} et des mémoires personnelles des particules $P_{best,i}^k$.

 Analyse des damiers et de la connectivité.

if $it_{PSO} = it_{max}$ **then**

if $volume_{G_{best}} / V_{ini} \geq V_{target}$ **then**

$it_{max} = it_{max} + 1$

end if

end if

$it_{PSO} = it_{PSO} + 1$

end while

4.3/ CHOIX DES PARAMÈTRES DE I-PR-PSO

Il a été montré dans la littérature, et en particulier par De Oca et al. [187] que les paramètres d'un algorithme par essaim particulaire doivent être choisis judicieusement, et peuvent dépendre fortement du problème à traiter. Dans leur travail, De Oca et al. proposent de réaliser une étude paramétrique empirique de l'influence des paramètres sur les capacités de l'algorithme à converger. Bien sûr, une solution radicale consisterait à utiliser un meta-algorithme par essaim particulaire [201] qui déterminerait au cours de son fonctionnement les meilleurs paramètres à employer en fonction de ses capacités. Ici, nous avons décidé d'employer la méthode de De Oca et al. [187] et avons réalisé une étude paramétrique empirique du fonctionnement de ESO-I-PR-PSO. Les paramètres testés sont c_1 , c_2 et ω . Le problème traité ici est le problème de *cantilever beam* classiquement utilisé dans la littérature et illustré en figure 4.6, et les paramètres mécaniques de la structure sont donnés dans le tableau 4.1. On choisit d'imposer une valeur maximale de 50 itérations.

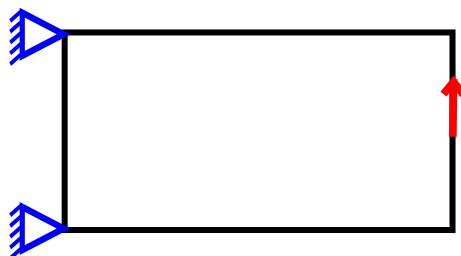


FIGURE 4.6 – Le problème classique de la cantilever beam

Dimensions du domaine de design	0.16 X 0.1m
Force appliquée	3000N
Module d'Young du matériau	210 000 MPa
Module d'Young du <i>ghost material</i>	210 000 Pa
Coefficient de Poisson des deux matériaux	0.3
Discretisation du domaine de design	32X20 éléments finis
Type d'élément fini	Q4
Nombre maximal d'itérations	50

Tableau 4.1 – Paramètres mécaniques du problème à traiter

L'objectif de ce problème d'optimisation est de minimiser la compliance de la structure, ce qui est équivalent à minimiser le déplacement maximal du point d'application de la force, dans le cas de l'application d'une seule force concentrée. Une contrainte d'inégalité est considérée sur le volume de la structure : le volume final de la structure doit être inférieur ou égal à 70% du volume initial de la structure. Le volume est approximé en comptant le nombre d'éléments finis contenant du matériau dans la topologie. Une méthode de pénalisation de la fonction objectif est utilisée, afin d'envoyer sa valeur à une valeur considérée comme très grande si la contrainte d'optimisation est violée, telle que

$$fitness_i = e^{fitness_i} + 1000 \quad (4.6)$$

où $fitness_i$ représente la valeur de la fonction objectif de la particule i de l'essaim.

Le but de ce travail n'est pas de déterminer la meilleure topologie possible, mais d'analyser les capacités de convergence de notre algorithme, en fonction du choix de ses paramètres.

4.3.1/ ANALYSE DES PARAMÈTRES DE CONFIANCE c_1 ET c_2

Nous proposons d'analyser premièrement l'influence des paramètres de confiance c_1 et c_2 . Pour rappel, ces paramètres jaugent la manière dont l'essaim accorde sa confiance plutôt à la meilleure particule globale de l'essaim ou aux mémoires personnelles des particules. Les huit paires de paramètres testées sont récapitulées dans le tableau 4.2 suivant

c_1	0	1	2	3	4	1
c_2	4	3	2	1	0	1

Tableau 4.2 – Paires des paramètres de confiance testées

Pour chaque paire de paramètres, nous analyserons les courbes de convergence des calculs réalisés, et en particulier la valeur finale de la fonction objectif obtenue, ainsi que la vitesse de convergence - ou en tout cas la capacité de l'algorithme à se rapprocher d'une zone prometteuse dans les 10 premières itérations de l'algorithme.

Les courbes de convergence des 6 paires différentes de paramètres sont données dans les figures 4.7-4.9 suivantes

Ces courbes de convergence représentent l'évolution de la fonction objectif, c'est à dire la valeur de la compliance de la structure, au cours des itérations. Comme on le voit sur la figure 4.8 sur laquelle on zoom sur le début de la convergence, il semblerait que les couples $c_1 = 1; c_2 = 1$, $c_1 = 4; c_2 = 0$ et $c_1 = 3, c_2 = 1$ donnent à l'algorithme la capacité de se diriger plus rapidement vers les zones prometteuses du domaine de recherche. On en déduit que c_1 doit être supérieur ou égal à c_2 afin d'améliorer les capacités de l'algorithme dans sa phase d'exploration du domaine de recherche. En outre, la figure 4.9 montre que le couple $c_1 = 1; c_2 = 1$ est le plus à même de pousser les particules à converger vers une solution la plus petite possible. Pour la suite des calculs, nous choisirons donc le couple de paramètres de confiance $c_1 = 1, c_2 = 1$.

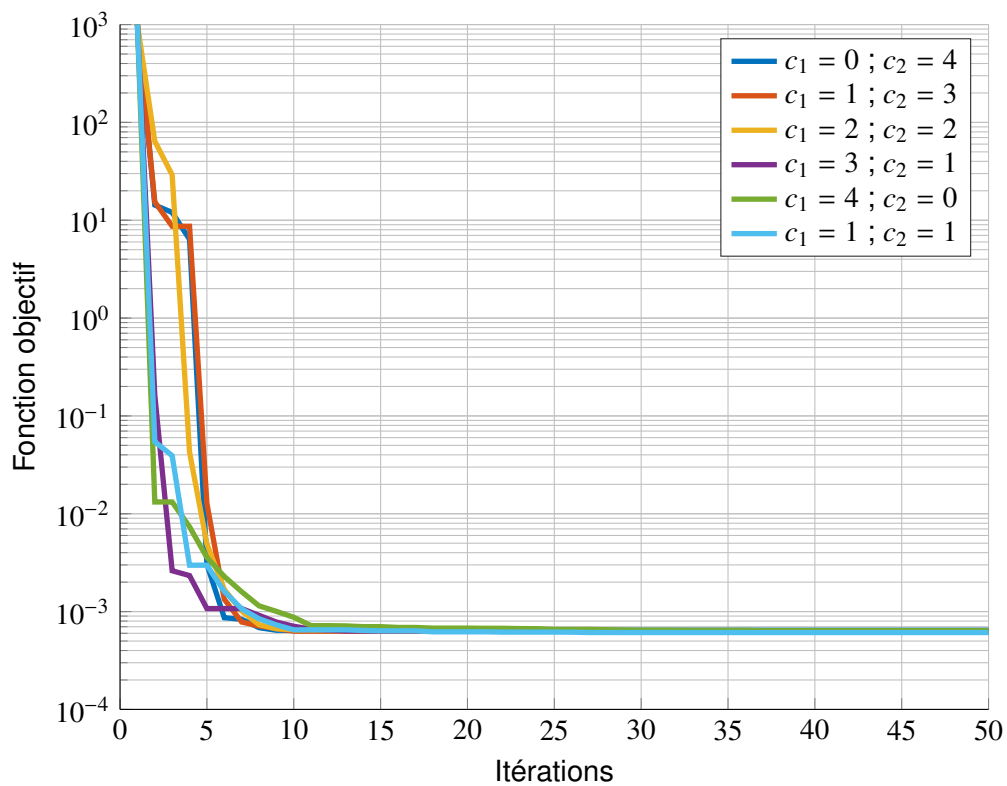


FIGURE 4.7 – Analyse des paramètres de confiance : courbe de convergence complète

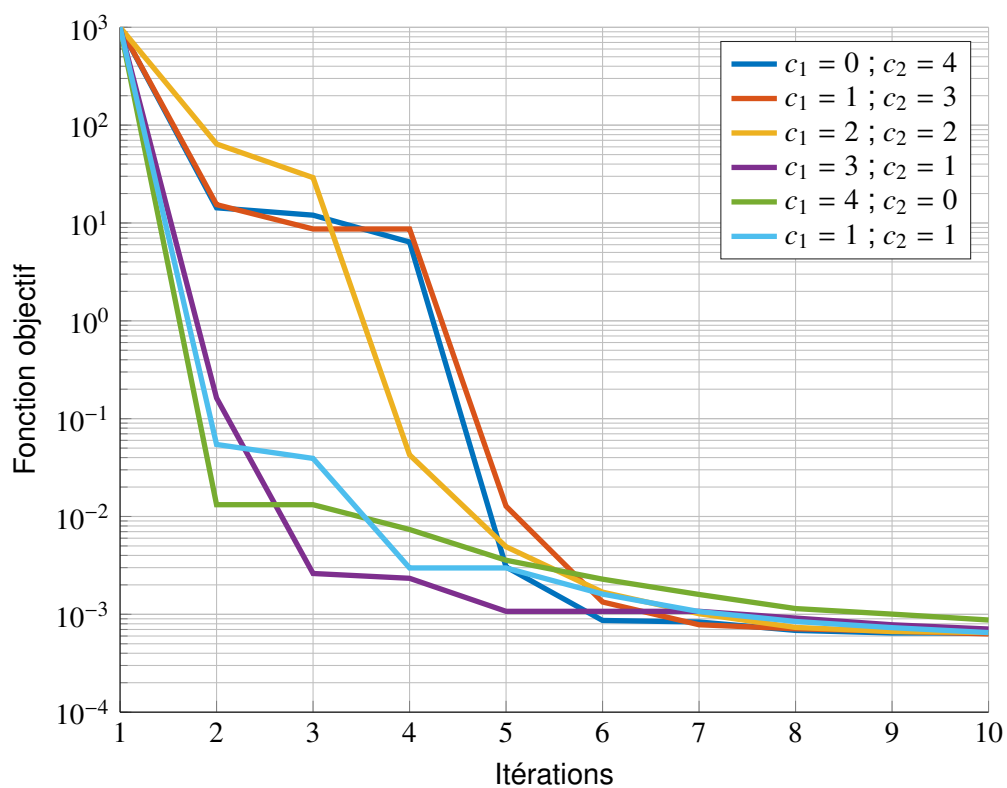


FIGURE 4.8 – Analyse des paramètres de confiance : début de la courbe de convergence

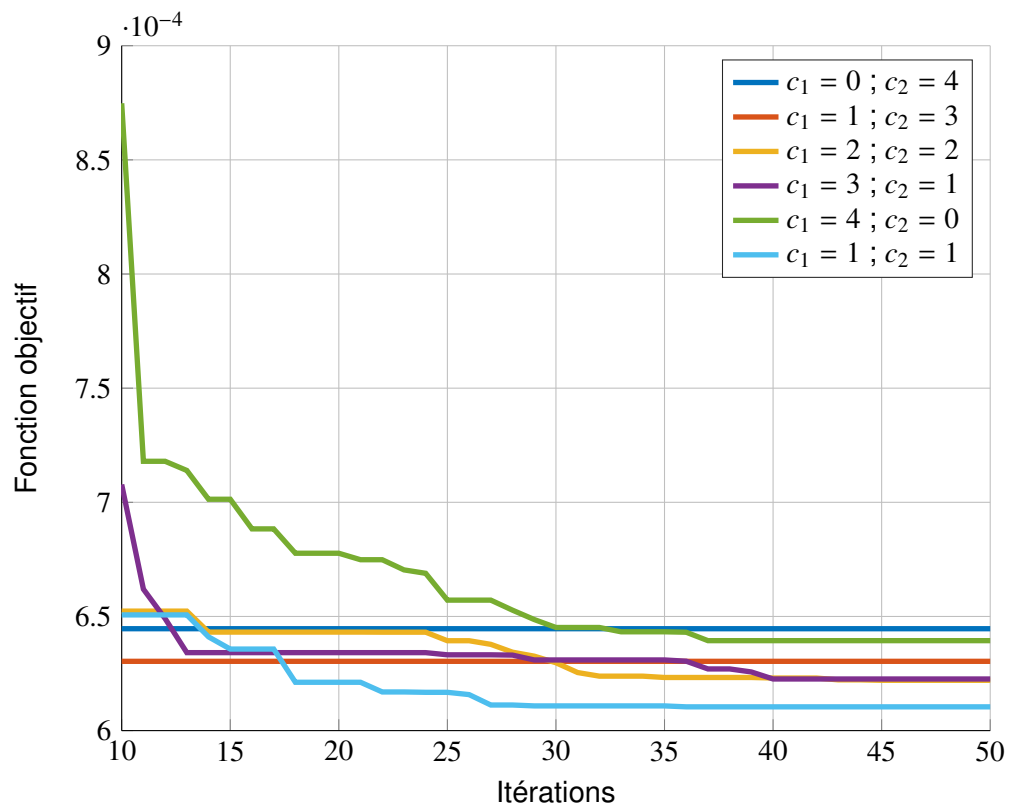


FIGURE 4.9 – Analyse des paramètres de confiance : fin de la courbe de convergence

4.3.2/ ANALYSE DU PARAMÈTRE D'INERTIE ω

Le but de cette partie est de déterminer quelle valeur du paramètre ω favorise à la fois la phase d'exploration du domaine, mais aussi la phase d'exploitation des zones prometteuses. Huit valeurs différentes de ω seront testées et sont les suivantes

- 0.4,
- 0.6,
- 0.8,
- 1.2,
- Décroissante linéairement de 1.2 à 0.4 par pas de $0.8/50 = 0.016$,
- Décroissante linéairement de 0.8 à 0.1 par pas de $0.7/50 = 0.014$,
- Il a été montré dans la littérature [81, 82, 90, 130, 168, 184] qu'une grande valeur de ω favoriserait la phase d'exploration du domaine, alors qu'une petite valeur aurait tendance à favoriser la phase d'exploitation des zones prometteuses. En effet, plus les vitesses des particules sont grandes, plus elles présentent de grands déplacements dans le domaine des solutions. Elles ont donc plus de chances de découvrir des zones encore inexploitées. *A contrario*, de petites vitesses auraient tendance à limiter grandement les déplacements des particules dans le domaine, favorisant ainsi la recherche locale des particules autour d'un point particulier du domaine. Ici, nous proposons donc de conserver une valeur assez grande de ω durant la première moitié des itérations du calcul - on choisira $\omega = 0.8$ - puis de décroître sa valeur linéairement de 0.8 à 0.1 pendant la seconde moitié des itérations du calcul.

- La dernière solution est la même que la précédente, si ce n'est que la valeur initiale de ω vaut 1.2 pendant la première moitié des itérations du calcul. La valeur de ω décroît donc linéairement de 1.2 à 0.1 pendant la seconde moitié des itérations du calcul.

Les courbes de convergence représentant l'évolution de la valeur de la fonction objectif au cours des itérations sont présentées en figures 4.10-4.12 suivantes.

On voit sur la figure 4.10 que les différentes valeurs de ω ont amené l'algorithme dans la même plage des valeurs finales, entre $1E-3$ et $1E-2$ à partir de la 5ème itération. On voit aussi sur la figure 4.11 que la valeur de ω décroissante entre 1.2 et 0.4 prodigue plus de facilités pour pousser l'algorithme aux abords de la zone prometteuse dans lequel se situe le minimum global de la fonction objectif considérée. Ceci dit, la figure 4.12 nous montre que c'est finalement la valeur $\omega = 1.2$ qui donne la valeur la plus faible de la fonction objectif. Il faut donc faire un choix entre ces deux possibilités. Basés sur les résultats de la littérature [81, 82, 90, 130, 168, 184], nous choisissons tout de même de travailler avec une valeur décroissante de ω . Les capacités de l'algorithme à converger, durant les dernières itérations, vers le minimum global de la fonction objectif doivent absolument être améliorées. Et si les vitesses des particules sont trop grandes, elles ne pourront pas converger efficacement vers ce minimum, en passant indéfiniment au-dessus¹ du point du domaine de recherche minimisant la valeur de la fonction objectif, sans jamais l'atteindre. Nous choisissons donc de travailler avec une valeur de ω décroissante linéairement entre 1.2 et 0.1 au cours des itérations.

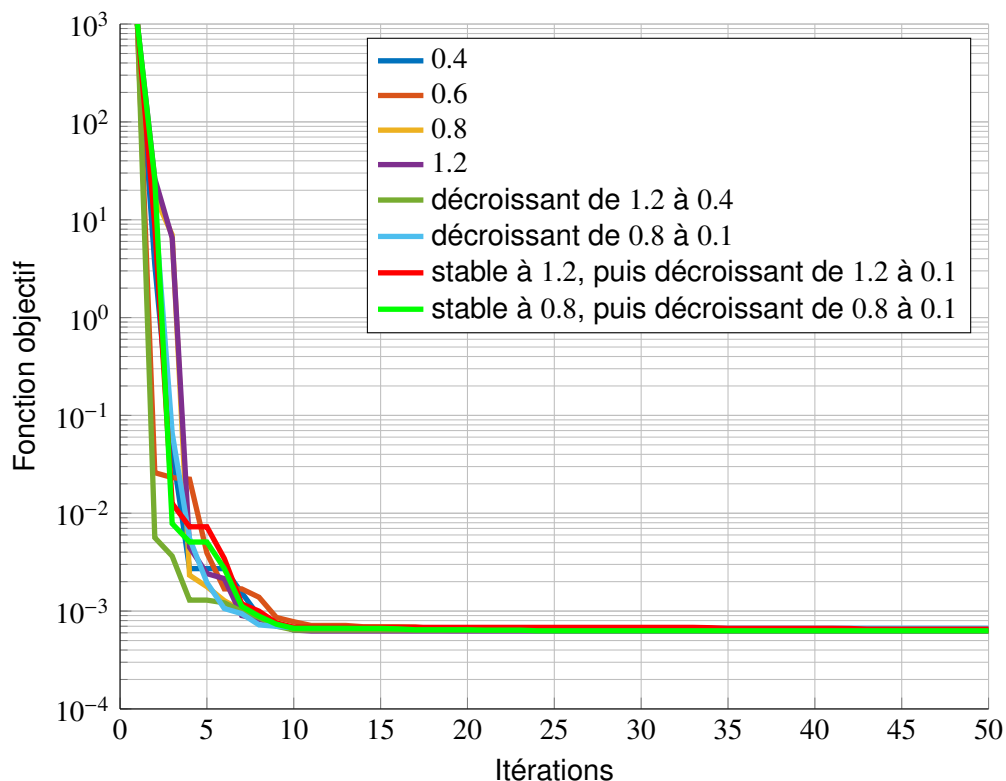
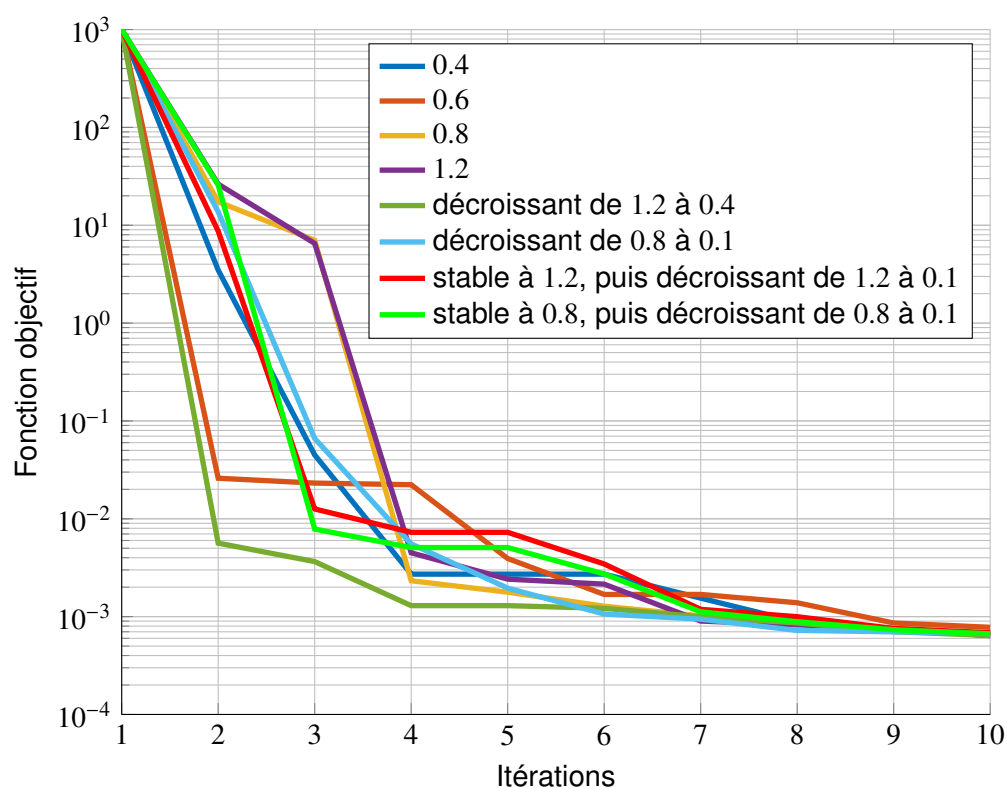
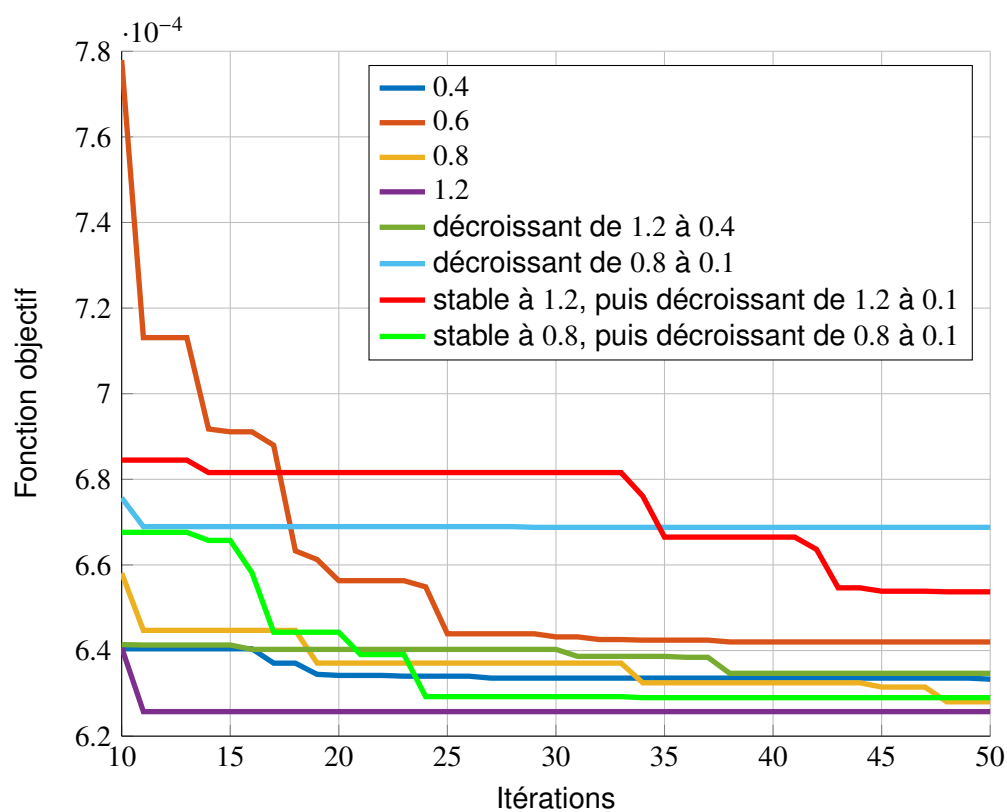


FIGURE 4.10 – Etude de l'influence du paramètre ω : Courbe de convergence complète

1. le mot "through" utilisé dans ce contexte dans la littérature anglophone semble décrire plus précisément ce processus.

FIGURE 4.11 – Etude de l'influence du paramètre ω : Zoom sur le début de la convergenceFIGURE 4.12 – Etude de l'influence du paramètre ω : Zoom sur la fin de la convergence

4.3.3/ ANALYSE DU NOMBRE DE PARTICULES DE L'ESSAIM

Le but de cette partie est de choisir le bon nombre de particules de notre essaim. Dans toute optimisation avec PSO - et en particulier lorsqu'un calcul par éléments finis est nécessaire pour calculer la valeur de la fonction objectif - il est nécessaire de faire le bon compromis entre un nombre très grand de particules qui nécessairement amènera à de meilleurs résultats, et un nombre plus faible de particules qui réduira considérablement le temps de calcul. Le but ici est donc d'essayer de déterminer à partir de quelle quantité de particules les résultats sont acceptables. Ou de déterminer à partir de quelle quantité de particules l'amélioration des résultats ne compense pas le temps de calcul perdu. Huit valeurs différentes seront testées : 5, 10, 15, 20, 25, 30, 35 et 40 particules. Les courbes de convergence sont données dans les figures 4.13-4.15 suivantes.

Comme nous l'avions pressenti, plus le nombre de particules dans l'essaim est grand, plus grandes sont les capacités de l'algorithme à converger d'abord dans les zones prometteuses du domaine, puis à converger vers le minimum global de la fonction objectif considérée. Cependant, on note sur les figures 4.14 et 4.15 qu'à partir de 30 particules, les capacités de l'algorithme ne sont pas tellement améliorées, même si l'on ajoute des particules à l'essaim. La valeur de la fonction objectif est diminuée, mais cette diminution est de l'ordre de $10E - 4$. Nous décidons donc de travailler avec 30 particules, afin de diminuer le temps de calcul de la fonction objectif par éléments finis, tout en conservant les capacités de notre algorithme à converger vers le minimum global de la fonction objectif considérée.

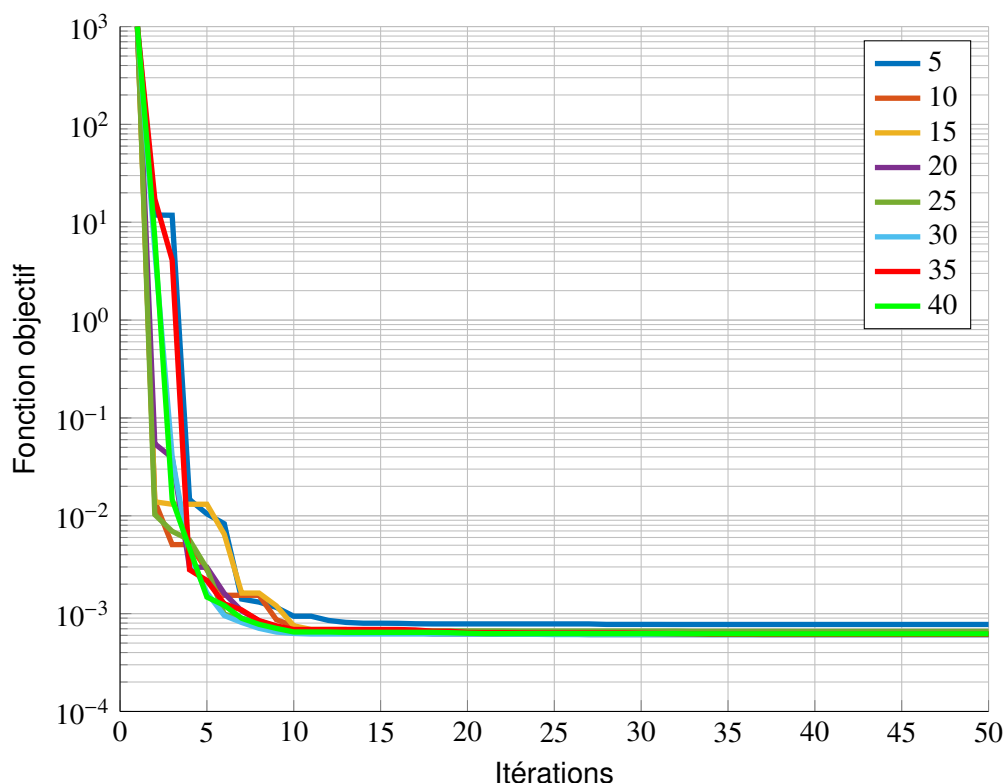


FIGURE 4.13 – Etude de l'influence du nombre de particules : courbe complète

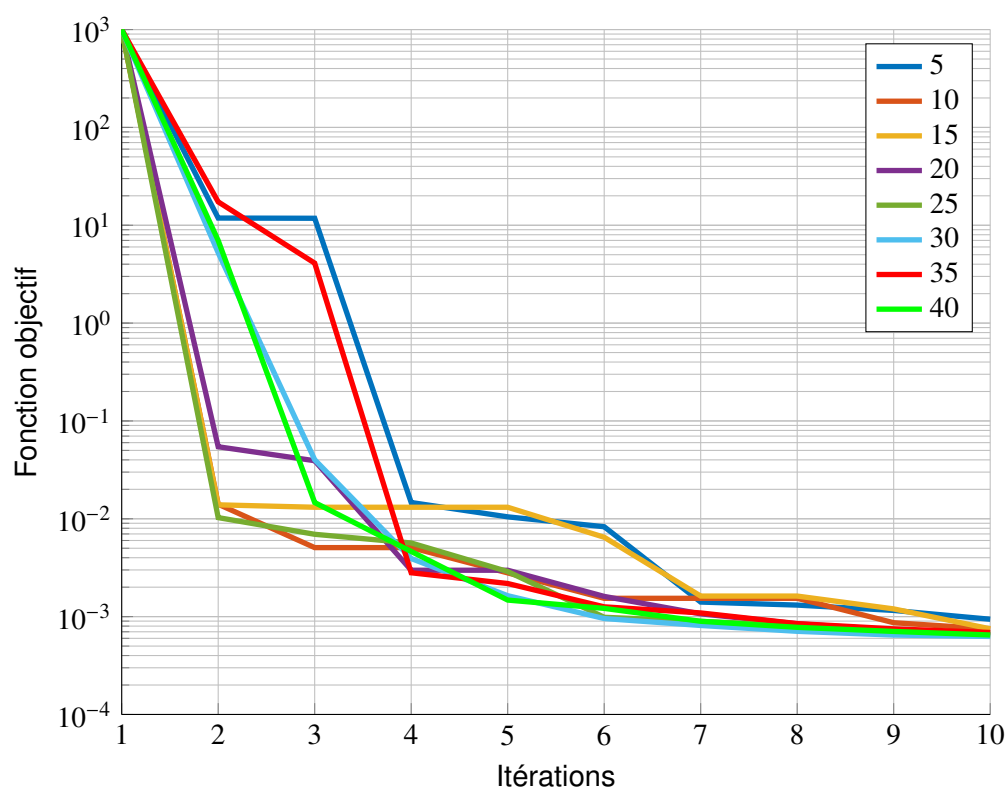


FIGURE 4.14 – Etude de l'influence du nombre de particules : début de la convergence

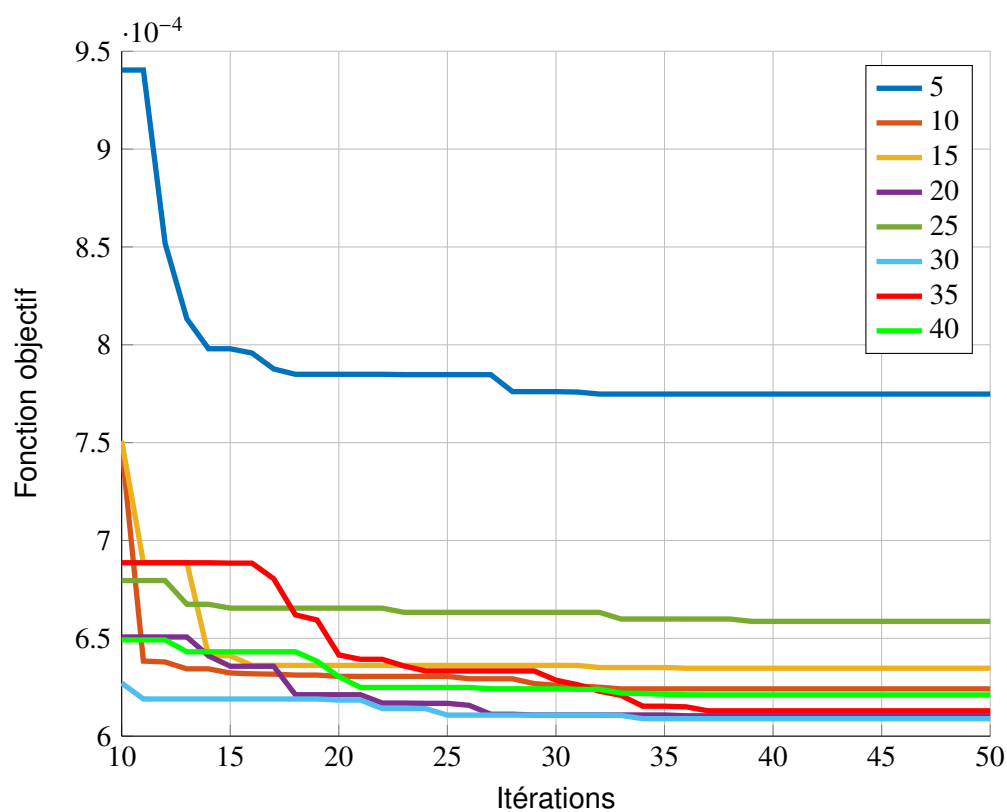


FIGURE 4.15 – Etude de l'influence du nombre de particules : fin de la convergence

4.3.4/ CONCLUSIONS DE L'ÉTUDE PARAMÉTRIQUE

Le but de cette étude paramétrique est d'améliorer les capacités de l'algorithme ESO-I-PR-PSO en choisissant les meilleurs paramètres c_1 , c_2 , ω , ainsi que le nombre de particules de l'essaim. En terme de capacités de convergence, deux aspects doivent être considérés : la capacité de l'algorithme à explorer le domaine, c'est à dire à s'approcher rapidement (en terme de nombre d'itérations) d'une zone prometteuse du domaine, mais aussi sa capacité à converger vers le minimum global de la fonction objectif dans la zone prometteuse considérée.

Compte tenu des ces considérations, les paramètres suivants ont été choisis, et seront réutilisés durant toute la suite des calculs :

- $c_1 = 1$ et $c_2 = 1$,
- ω est décroissant linéairement entre 1.2 et 0.1 au cours des itérations,
- 30 particules constituent l'essaim.

4.4/ PROBLÈMES TRAITÉS, RÉSULTATS ET DISCUSSION

Sept problèmes d'optimisation topologique ont été traités, pour tester et valider la performance de notre algorithme ESO-I-PR-PSO, et sont récapitulés dans le tableau 4.4 suivant. Les paramètres mécaniques communs à tous les problèmes traités sont récapitulés dans le tableau 4.3 suivant.

Dimensions du domaine de design	0.16 X 0.1m
Epaisseur des éléments finis	0.001 m
Force appliquée	3000N
Module d'Young du matériau	210 000 MPa
Module d'Young du <i>ghost material</i>	210 000 Pa
Coefficient de Poisson des deux matériaux	0.3
Discretisation du domaine de design	32X20 éléments finis
Type d'éléments finis	Q4
Nombre maximal d'itérations	50

Tableau 4.3 – Paramètres mécaniques du problème à traiter

Le but de ce travail est de minimiser la compliance de la structure, approximée par la valeur du déplacement du point d'application de la force, en considérant une contrainte d'optimisation sur le volume de la structure. Cinq contraintes d'optimisation V/V_0 ont été traitées, pour chaque problème mécanique, et sont 0.3, 0.4, 0.5, 0.6 et 0.7. Les pourcentages d'éléments supprimés lors de la première itération ESO ont été déterminés de manière empirique et sont récapitulés dans le tableau 4.4. Lorsque les problèmes mécaniques sont symétriques, il semble judicieux de le considérer lors de la résolution de notre problème d'optimisation. Ainsi, sans considérer la symétrie d'un point de vue mécanique - c'est à dire en n'effectuant que la moitié du calcul par éléments finis, sous réserve d'application de conditions aux limites de symétries - nous la considérons d'un point de vue optimisation, c'est à dire en faisant une symétrie axiale de la topologie dans le domaine de design. Ainsi, on ne considère dans ces cas que la moitié des variables d'optimisation, ce qui accélère le temps de calcul et facilite le travail de ESO-I-PR-PSO, et la matrice de topologie est remplie en effectuant une symétrie miroir des variables d'optimisation. Les

topologies obtenues dans chacun des cas ainsi que les contraintes de Von Mises et les valeurs extrêmes des contraintes de Von Mises sont données dans les figures 4.5-4.11 suivantes, sur lesquelles les éléments bleus correspondent au *ghost material*, alors que les éléments rouges représentent les éléments constitués de matériau. Le calcul de la fonction objectif a été réalisé par la méthode des éléments finis, avec le logiciel Cast3m[®], alors que les calculs d'optimisation ont été implémentés sous Matlab. Une méthode d'interfaçage entre ces deux logiciels a donc été développée et utilisée ici.

Comme on le voit dans ces figures, les topologies obtenues sont régulières et symétriques lorsque le problème s'y prête. Notons que les topologies sont un peu moins régulières dans les cas 2 et 3 qui ne sont pas des problèmes mécaniques symétriques. Selon nous, ce manque de régularité vient du très grand nombre de variables d'optimisation, qui correspond au nombre total d'éléments dans le domaine de design, que l'algorithme I-PR-PSO doit gérer.

Concernant la convergence de l'algorithme, nous n'analyserons ici que 3 cas. Les autres courbes de convergence sont présentées en annexe C. Les figures 4.16-4.17 suivantes montrent l'évolution de la valeur de la fonction objectif et du rapport de volume V/V_0 au cours des itérations, dans le cas 7 lorsque l'on impose $V/V_0 \leq 0.5$. Ces courbes sont mises en regard avec l'évolution du paramètre rr .

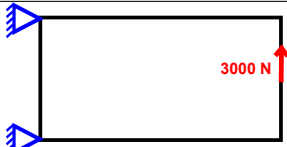
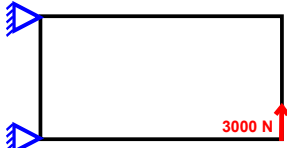
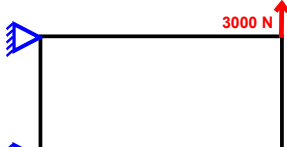
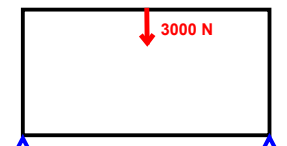
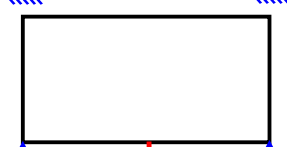
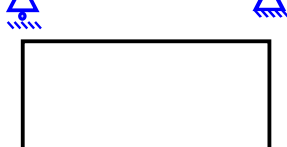
	Domaine de design	Chargement	Conditions aux limites	rr_{ini}
Cas 1		noeud au milieu du côté droit	Nœuds fixés : Haut-gauche et bas-gauche	5%
Cas 2		noeud en bas du côté droit	Nœuds fixés : Haut-gauche et bas-gauche	5%
Cas 3		noeud en haut du côté droit	Nœuds fixés : Haut-gauche et bas-gauche	5%
Cas 4		noeud au milieu du côté haut	Nœuds fixés : Bas-gauche et bas-droit	7%
Cas 5		noeud au milieu du côté bas	Nœuds fixés : Bas-gauche et bas-droit	10%
Cas 6		noeud au milieu du côté haut	Nœud fixé en bas-droit, u_y bloqué en bas-gauche	10%
Cas 7		noeud au milieu du côté bas	Nœud fixé en bas-droit, u_y bloqué en bas-gauche	8%

Tableau 4.4 – Benchmark de problèmes mécaniques

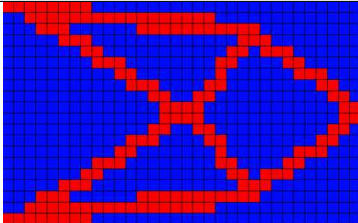
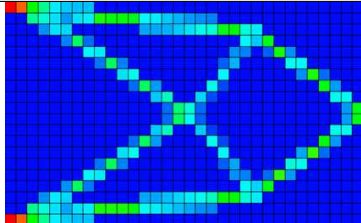
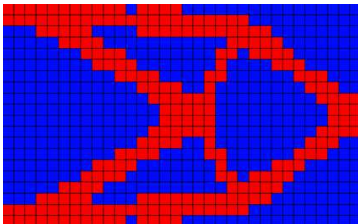
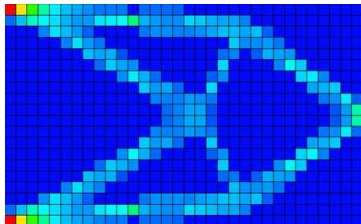
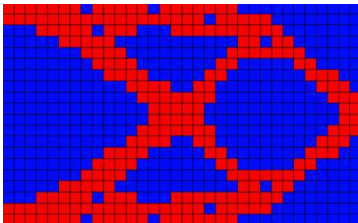
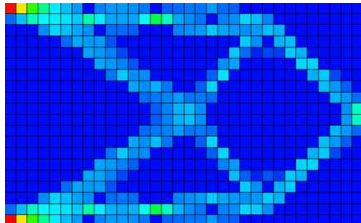
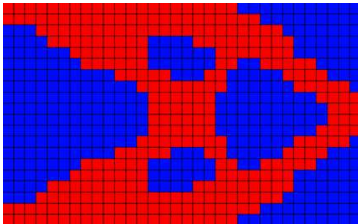
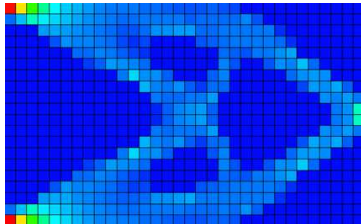
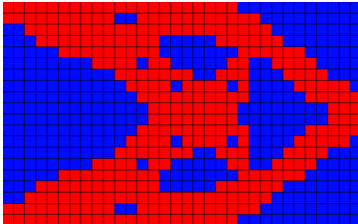
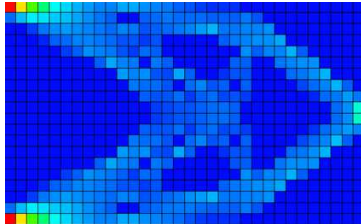
V/V_0	Topologie	Contraintes de Von Mises	$\sigma_{min}^{VM}-\sigma_{max}^{VM}$
0.3			2.18E-4 3.08E+4
0.4			1.30E-4 3.10E+4
0.5			2.11E-4 3.11E+4
0.6			1.38E-4 3.11E+4
0.7			9.86E-5 3.11E+4

Tableau 4.5 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 1

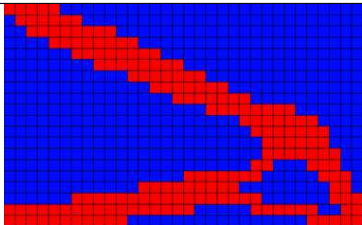
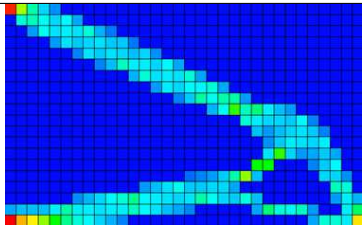
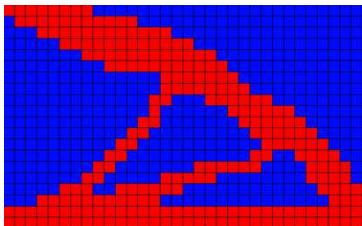
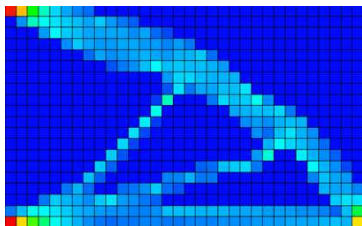
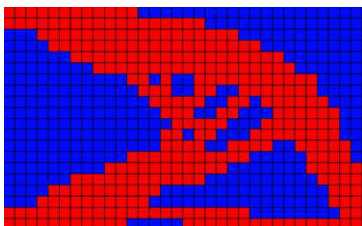
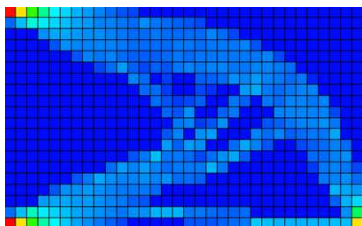
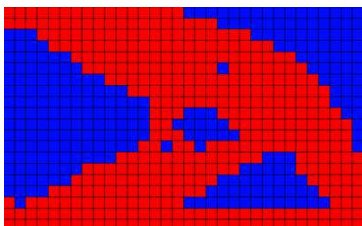
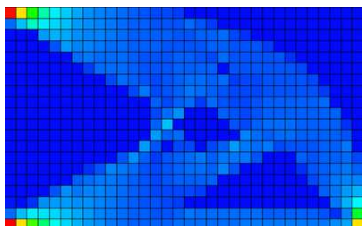
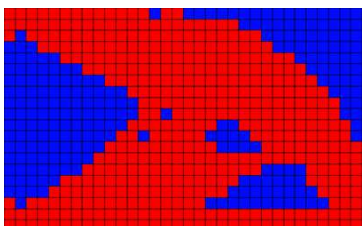
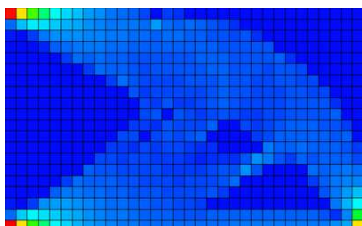
V/V_0	Topologie	Contraintes de Von Mises	$\sigma_{min}^{VM} - \sigma_{max}^{VM}$
0.3			5.12E-5 3.18E+4
0.4			1.35E-4 3.11E+4
0.5			9.82E-5 3.11E+4
0.6			6.83E-5 3.11E+4
0.7			8.00E-5 3.11E+4

Tableau 4.6 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 2

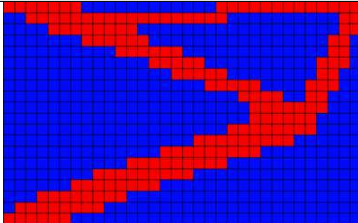
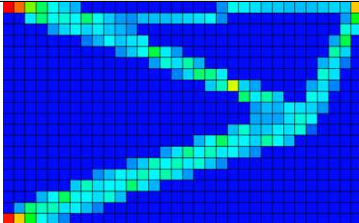
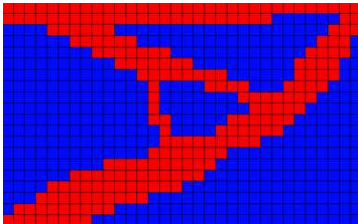
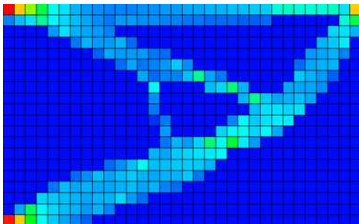
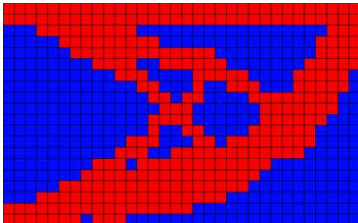
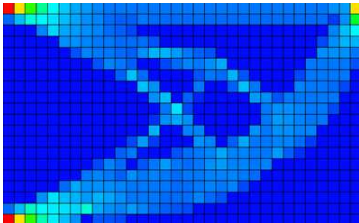
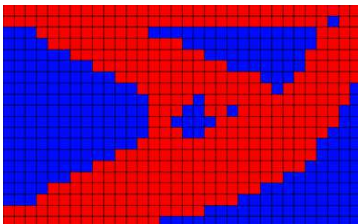
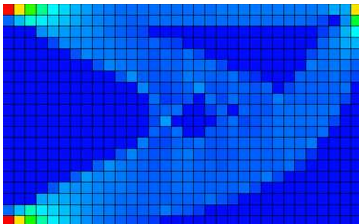
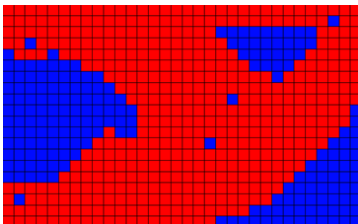
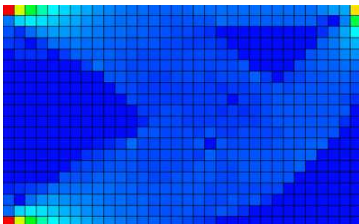
V/V_0	Topologie	Contraintes de Von Mises	$\sigma_{min}^{VM}-\sigma_{max}^{VM}$
0.3			1.70E-4 3.08E+4
0.4			1.27E-4 3.09E+4
0.5			8.28E-5 3.11E+4
0.6			7.96E-5 3.11E+4
0.7			7.37E-5 3.15E+4

Tableau 4.7 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 3

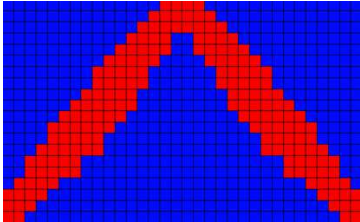
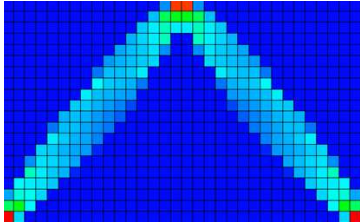
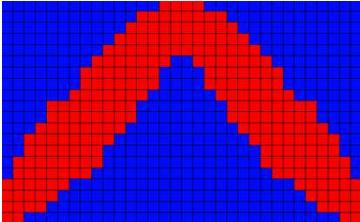
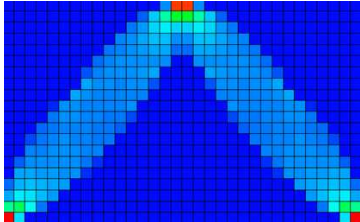
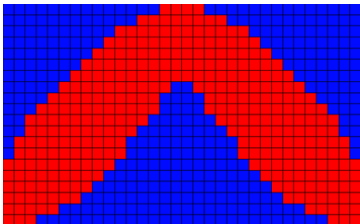
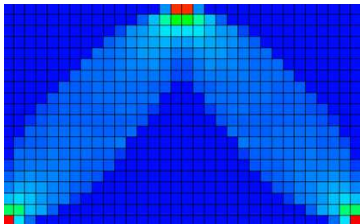
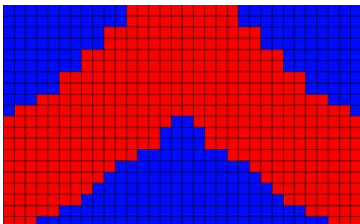
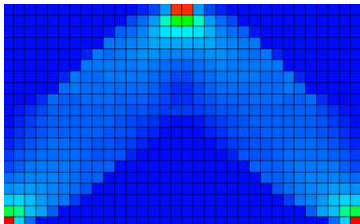
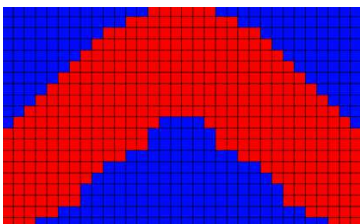
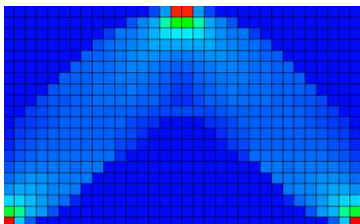
V/V_0	Topologie	Contraintes de Von Mises	$\sigma_{min}^{VM} - \sigma_{max}^{VM}$
0.3			3.17E-5 1.08E+4
0.4			2.41E-5 1.08E+4
0.5			1.70E-5 1.06E+4
0.6			2.20E-5 1.06E+4
0.7			1.66E-5 1.04E+4

Tableau 4.8 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 4

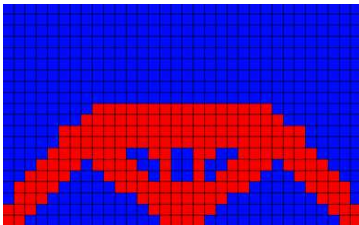
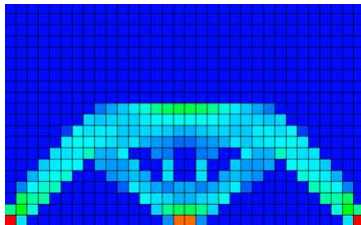
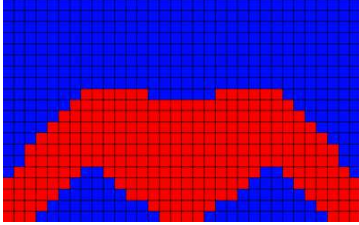
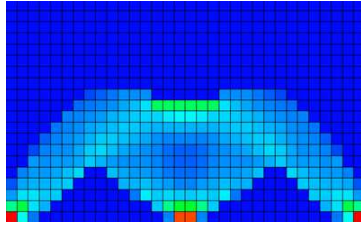
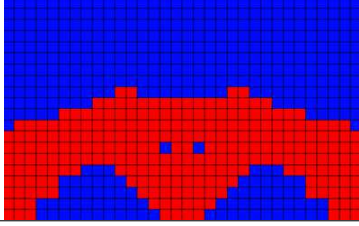
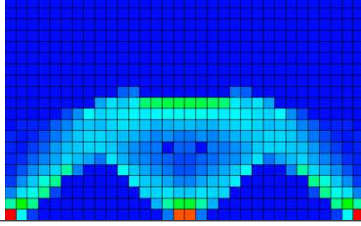
V/V_0	Topologie	Contraintes de Von Mises	$\sigma_{min}^{VM} - \sigma_{max}^{VM}$
0.3			3.51E-5 1.17E+4
0.4			2.72E-5 1.11E+4
0.5			4.50E-5 1.12E+4

Tableau 4.9 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 5

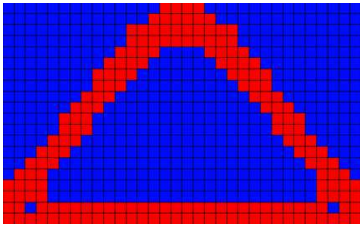
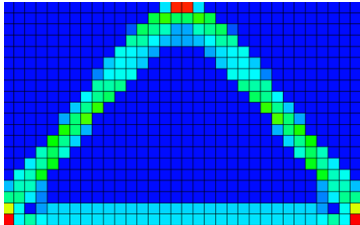
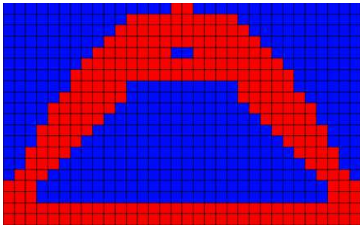
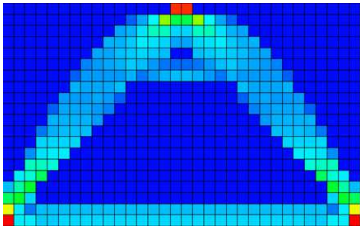
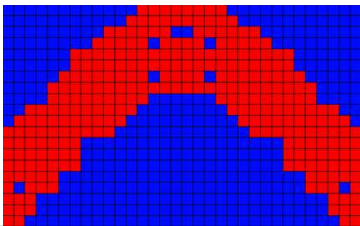
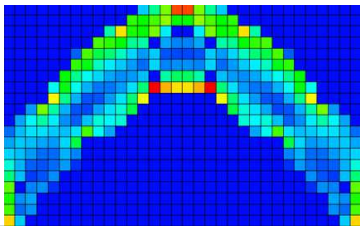
V/V_0	Topology	Contraintes de Von Mises	$\sigma_{min}^{VM} - \sigma_{max}^{VM}$
0.3			4.95E-5 1.09E+4
0.4			2.70E-5 1.09E+4
0.5			1.33E-4 1.48E+4

Tableau 4.10 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 6

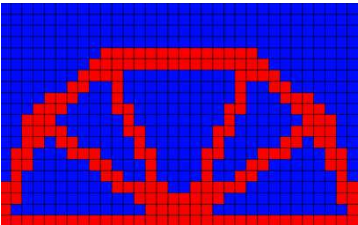
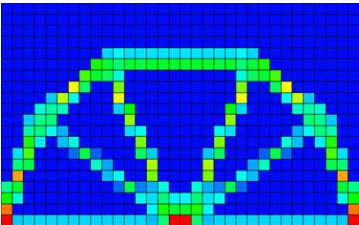
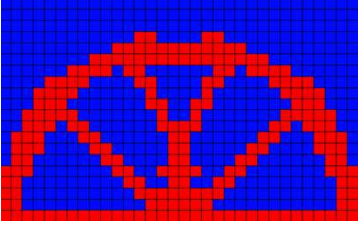
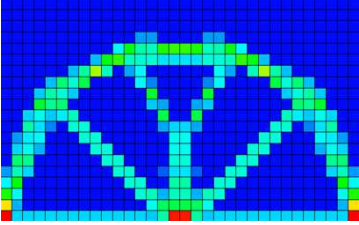
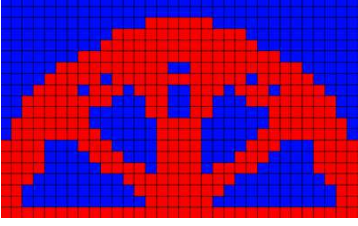
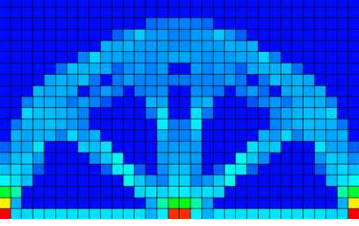
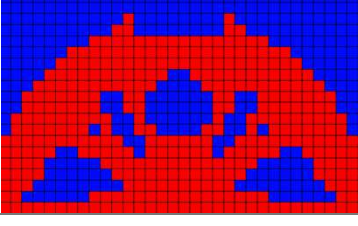
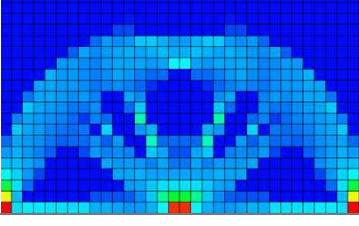
V/V_0	Topology	Contraintes de Von Mises	$\sigma_{min}^{VM} - \sigma_{max}^{VM}$
0.3			7.34E-5 1.08E+4
0.4			5.76E-5 1.08E+4
0.5			2.89E-5 1.08E+4
0.6			2.18E-5 1.08E+4

Tableau 4.11 – Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 7

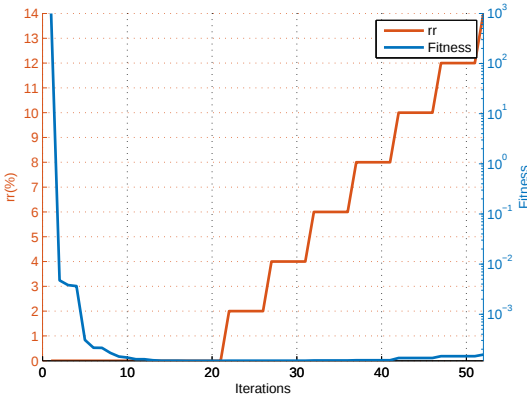


FIGURE 4.16 – Fonction objectif

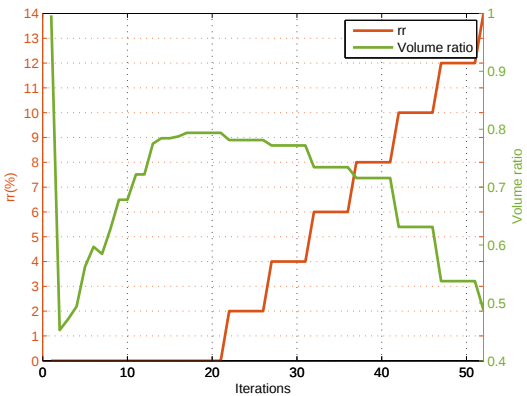


FIGURE 4.17 – Rapport V/V_0

Cas 7 : courbes de convergence de la fonction objectif et de rapport de volume, considérant la contrainte $V/V_0 \leq 0.5$

Comme on le voit sur les figures 4.16-4.17, l'algorithme I-PR-PSO travaille seul jusqu'à l'itération 22, ou l'algorithme ESO prend le relais. Sur la figure 4.17 en particulier, on voit bien que le volume de la structure diminue brusquement lors de la première itération, il s'agit là de l'initialisation des particules avec une première itération ESO. I-PR-PSO ajoute alors des éléments à la structure jusqu'à atteindre le rapport de volume intermédiaire choisi $V/V_0 = 0.8$, comme nous l'avons présenté précédemment à la page 131. Dès lors que le calcul peut être considéré comme convergé, c'est à dire que l'on a $G_{best}^{k+1} - G_{best}^k \leq 1E - 10$, l'algorithme ESO prend le relais, et optimise la structure ainsi trouvée, jusqu'à ce que l'on atteigne le rapport de volume final demandé de 0.5. I-PR-PSO et ESO travaillent alors en binôme de manière séquentielle. Cette communication entre les deux algorithmes permet plusieurs avantages :

- D'une part, on diminue grandement le nombre d'itérations et donc d'appels au calcul de la fonction objectif par rapport à un algorithme d'optimisation par essaim particulaire seul [208], ou un algorithme ESO seul [167, 200].
- D'autre part, les topologies sont bien plus régulières comparées à l'utilisation de ces deux algorithmes seuls, d'autant plus lorsque les symétries de la structure sont considérées afin de diminuer le nombre de variables d'optimisation. L'annexe C présente aussi en seconde partie des résultats d'optimisation topologique de la littérature réalisés avec un algorithme d'optimisation par essaim particulaire seul.

Comme on le voit dans la figure 4.16 en particulier, la valeur de la fonction objectif augmente quelque peu lorsque l'algorithme ESO prend le relais. Ceci est dû à la contradiction des objectifs, tel que nous l'avons vu précédemment p. 132. Ici, en considérant une contrainte d'optimisation $V/V_0 \leq 0.8$ durant la première étape du processus d'optimisation, notre algorithme s'assure que la valeur de la compliance est faible ($1,1797E-4$ m) lorsque ESO prend le relais. Ainsi, même si la suppression d'éléments augmente quelque peu la compliance de la structure, l'algorithme global converge vers une solution acceptable, dont les contraintes d'optimisation concernant le volume sont respectées.

L'annexe D récapitule les valeurs numériques atteintes par notre algorithme ESO-I-PR-PSO. Sont comparées en particulier les valeurs de la fonction objectif et du rapport de volume V/V_0 trouvées à la fin de la première étape de calcul, c'est à dire lorsque ESO prend le relais, et à la fin du calcul, lorsque tous les objectifs (compliance minimisée et volume atteint) sont atteints. On voit bien dans ce tableau que la valeur de la fonction objectif augmente légèrement lorsque l'algorithme ESO commence son travail de minimisation du volume, mais cette perte de la compliance est de l'ordre - en général - du dixième de millimètre, ce que l'on peut considérer comme négligeable.

Analysons maintenant les résultats du cas 5, lorsque l'on impose $V/V_0 \leq 0.4$. Les courbes de convergence sont présentées en figures 4.18-4.19 suivantes

Comme on le voit sur la figure 4.19, l'algorithme I-PR-PSO n'atteint jamais le rapport de volume initial $V/V_0 = 0.8$ demandé. Ainsi, l'algorithme ESO qui prend le relais à l'itération 24 n'a plus beaucoup d'éléments à éliminer afin d'atteindre l'objectif final de rapport de volume $V/V_0 \leq 0.4$. C'est la raison pour laquelle l'algorithme ESO ne travaille pas du tout lorsque l'on souhaite vérifier la condition $V/V_0 \leq 0.5$, comme on le voit dans les figures 4.20-4.21 suivantes, le paramètre rr reste nul pendant toute la durée du calcul. Lorsque l'on atteint le nombre maximal d'itérations, fixé à 20, le rapport de volume est déjà inférieur aux 50% demandés, et l'algorithme s'arrête. C'est la raison pour laquelle nous n'avons pas présenté tous les résultats dans les cas 5, 6 et 7. Dans certains cas donc, l'algorithme I-PR-PSO est capable de travailler seul, sans l'aide de l'algorithme ESO, dans le sens où

le volume atteint est suffisamment bas lorsque l'on minimise la compliance.

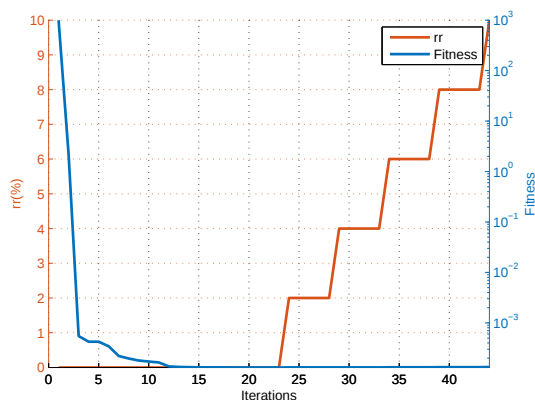
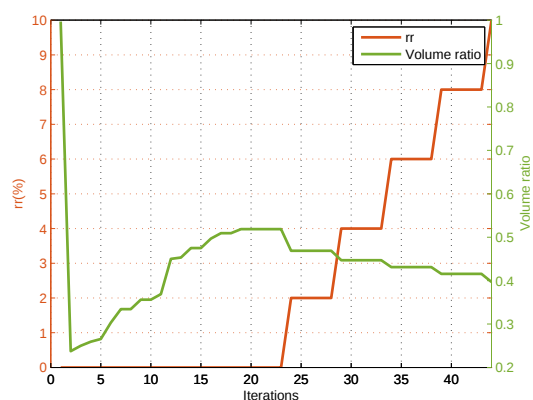


FIGURE 4.18 – Fonction objectif

FIGURE 4.19 – Rapport V/V_0

Cas 5 : courbes de convergence de la fonction objectif et du rapport de volume, considérant $V/V_0 \leq 0.4$

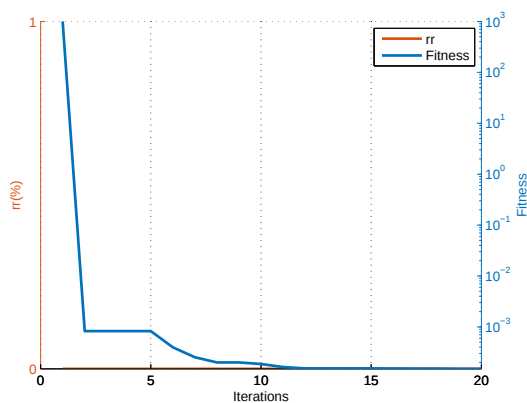
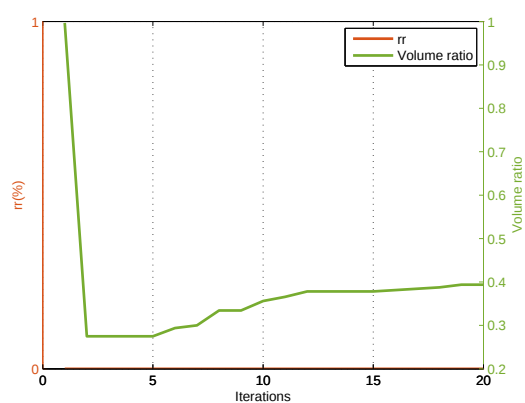


FIGURE 4.20 – Fonction objectif

FIGURE 4.21 – Rapport V/V_0

Cas 5 : courbes de convergence de la fonction objectif et du rapport de volume, considérant $V/V_0 \leq 0.4$

4.5/ CONCLUSIONS DU CHAPITRE

Ce chapitre propose une nouvelle méthode d'optimisation topologique appliquée à la mécanique, basée conjointement sur notre algorithme Inverse-PageRank-PSO et sur la méthode Evolutionary Structural Optimization. Dans notre nouvel algorithme ESO-I-PR-PSO, les deux algorithmes travaillent de paire, de manière séquentielle. I-PR-PSO guide tout d'abord l'algorithme vers une solution potentiellement acceptable, c'est à dire une solution qui minimise la compliance de la structure considérée. Puis ESO continue à optimiser la structure, pour minimiser une fonction objectif concernant le volume de la structure, en considérant des contraintes d'optimisation sur les contraintes de Von Mises dans les éléments, tout en conservant les propriétés acquises par I-PR-PSO.

Après une étude paramétrique, nous avons déterminé quelles étaient les valeurs de ω , c_1 , c_2 et du nombre de particules de l'essaim les plus adaptées à notre problème, comme il l'avait été fait dans le travail de De Oca et al. [187]. Cette étude paramétrique nous a semblée nécessaire, puisqu'il a été montré dans la littérature que ces paramètres sont déterminant quant à la capacité de convergence de tout algorithme par essaim particulaire. Ceci dit, il semblerait qu'un meta-algorithme aurait pû être une meilleure solution, en ajustant, au cours des itérations, ces paramètres. Ces hypothèses feront certainement l'objet d'un travail futur.

Les deux problèmes d'optimisation classiques en mécanique des structures (minimisation de la compliance avec une contrainte d'optimisation sur le volume et minimisation du volume de la structure avec des contraintes d'optimisation sur les contraintes mécaniques de Von Mises) sont tous les deux traités par notre algorithme ESO-I-PR-PSO. Bien que ces objectifs soient, en apparence, contradictoires, ils sont traités de manière séquentielle, d'abord par l'algorithme I-PR-PSO, puis par l'algorithme ESO, de telle sorte que tous les objectifs sont atteints, et les contraintes d'optimisation vérifiées à la fin du calcul. Cette méthode de séparation des objectifs avait déjà été proposée dans la littérature, appelée [la méthode du compromis](#), [la méthode du but programmé](#) ou encore [la méthode d'ordonnancement lexicographique](#). Le traitement de ces deux objectifs de manière séquentielle nous aura permis d'obtenir des résultats qui minimisent le volume tout en maximisant la compliance.

Les résultats montrent que les topologies obtenues sont plus régulières que celles obtenues par un algorithme d'optimisation par essaim particulaire seul. En outre, ils montrent aussi que notre algorithme diminue le temps de calcul, en minimisant le nombre d'itérations nécessaire à la convergence, et donc le nombre d'appels au calcul de la fonction objectif par éléments finis.

OPTIMISATION DES METAMATÉRIAUX ÉLECTROMAGNÉTIQUES : DÉVELOPPEMENT D'UNE CAPE D'INVISIBILITÉ ÉLECTROMAGNÉTIQUE FRÉQUENTIELLE

Sommaire

5.1	Introduction	156
5.2	Définition du problème d'optimisation à résoudre	158
5.2.1	Définition du modèle numérique	158
5.2.2	Equations gouvernant le problème	160
5.2.3	Calcul de la perméabilité effective	161
5.3	Formulation du problème d'optimisation	163
5.4	Algorithme de résolution : Utilisation d'une version binaire de PSO	164
5.5	Implémentations numériques	166
5.5.1	Généralités concernant le modèle numérique et les unités employées	166
5.5.2	Exemples numériques	167
5.5.2.1	Minimisation de la perméabilité effective à 10 THz	167
5.5.2.2	Minimisation de la perméabilité effective à 15 THz	170
5.6	Conclusions du chapitre	172

5.1/ INTRODUCTION

De manière générale, et comme nous l'avons vu dans la partie 2.4, les metamatériaux sont une nouvelle classe de matériaux dont les propriétés physiques exceptionnelles n'existent pas dans la nature et ce, bien que les matériaux qui les composent aient des propriétés physiques connues et tout à fait classiques. De fait, de nombreuses recherches ont été réalisées ces dernières années afin à la fois d'exprimer leurs caractéristiques physiques, mais aussi de les optimiser. Ces metamatériaux sont très souvent composites, il est alors nécessaire d'employer un processus d'optimisation topologique afin de déterminer la répartition de matière optimale, c'est à dire celle qui leur confèrera ces propriétés physiques exceptionnelles.

De la même manière que les metamatériaux mécaniques, les metamatériaux électromagnétiques présentent une structure périodique. Les travaux les concernant ont montré notamment qu'il était possible d'obtenir des matériaux dont l'indice de réfraction et/ou la permittivité et la perméabilité sont négatives. Chaque cellule de périodicité est ainsi composée d'une matrice d'un premier matériau - appelé le *background material* dans la littérature - et d'inclusions d'un second matériau diélectrique dans le premier. Même si les deux matériaux ne présentent pas les propriétés exceptionnelles énoncées ci-dessus, c'est finalement la combinaison des deux qui peut présenter ces caractéristiques électromagnétiques, lorsque la structure est illuminée par un champ électromagnétique [220].

La possibilité d'obtenir ces propriétés électromagnétiques exceptionnelles a été présentée pour la première fois par Veselago en 1968 [7]. Il a en effet montré qu'un arrangement original et périodique de matériaux classiques pouvait engendrer une perméabilité et/ou une permittivité négatives. Par suite, Pendry et al. [89] et Smith et al. [118, 153] ont montré que des arrangements d'anneaux métalliques (*split-ring resonators*) et de ressorts métalliques pouvaient présenter un indice de réfraction négatif à une fréquence d'illumination donnée. Ils ont aussi montré que la perméabilité et la permittivité effectives d'un metamatériau peuvent être calculées à partir des paramètres de dispersion (*the scattering parameters*), aussi appelés S-parameters.

Ainsi, Chen et al. [132], Cotuk [142], Smith et al. [118], Feng [156], et Cem Hasar et al. [216] ont respectivement travaillé sur le développement de méthodes efficaces pour calculer les paramètres effectifs du metamatériau considéré à partir des S-parameters, comme des méthodes matricielles, que l'on retrouve d'ailleurs aujourd'hui implémentées dans certains logiciels de calcul par éléments finis. D'ailleurs, Kannellopoulos et al. [40] et N'Guessan [61] ont montré que la méthode des éléments finis peut être utilisée et adoptée pour la résolution de problèmes électromagnétiques hautes fréquences. Le but de leur travail a été de déterminer et d'adapter un modèle numérique pour discrétiser les lois de Maxwell en trois dimensions et de définir précisément les conditions aux limites à implémenter pour pouvoir résoudre ce type de problèmes d'impacts de champs électromagnétiques. Finalement, trois méthodes différentes ont été développées et implémentées pour discrétiser le domaine de calcul en éléments finis, compte tenu des lois de Maxwell à résoudre et des conditions aux limites à prendre en compte [182] :

- Premièrement, des conditions aux limites absorbantes (appelées *Absorbing Boundary Conditions* ABC) [40] ont été proposées initialement pour les domaines non-périodiques [162]. Cette formulation est très efficace et peut être adaptée à la fois pour les problèmes de dispersion et les problèmes de radiation électromagnétiques. Par la suite, elles ont été utilisées pour les implémentations numériques de

structures infiniment périodiques. Elles permettent de limiter les réflexions retour des champs électromagnétiques dans le domaine de calcul.

- Deuxièmement, des couches parfaitement adaptées (*Perfectly Matched Layers* notées PML) ont été développées afin d'absorber les réflexions retour des champs électromagnétiques dans le domaine. Ces PML atténuent ainsi les phénomènes de réflexions internes dans les structures en trois dimensions modélisées par différences finies [65, 112, 123]. Les PML ont ainsi été utilisées comme des conditions aux limites absorbantes dans la méthode des éléments finis [52], pour résoudre des problèmes de dispersion électromagnétique en trois dimensions [62].
- Troisièmement, moyennant un coup de calcul très élevé [182], les équations des intégrales limites permettant l'atténuation complète des phénomènes de réflexions dans le domaine ont été proposées. Cette méthode a été développée à l'origine pour discrétiser le domaine de calcul en éléments finis [24, 37, 80, 131], et a montré des résultats exacts comparés aux études expérimentales. Afin d'accélérer la convergence des calculs notamment en 3 dimensions [115], une méthode de préconditionnement a été proposée.

A la suite de ces recherches, de nombreuses applications ont été proposées, afin de tirer profit des extraordinaires propriétés que présentaient ces metamatériaux, comme des capes d'invisibilité [193], des guides d'ondes [218], des antennes [186], et des procédés d'invisibilité acoustique [135]. Enfin, connaissant de nombreuses propriétés physiques des metamatériaux électromagnétiques, les recherches ont commencé à s'orienter dans le domaine de l'optimisation afin d'obtenir les meilleures propriétés possibles. Zhou et al. [202] proposent d'utiliser la méthode des courbes de niveaux précédemment présentée dans la partie 2.1 afin d'obtenir des metamatériaux doublement négatifs, c'est à dire dont la perméabilité et la permittivité sont simultanément négatives. Dans ce cas, la fonction objectif a été formulée en fonction du flux de courant électrique. Par suite, Zhou et al. [210] ont proposé une deuxième méthode *level-set* pour optimiser les propriétés électromagnétiques des metamatériaux. Cette fois-ci, les *S-parameters* ont été utilisés afin d'exprimer la perméabilité effective du metamatériau considéré, et d'en faire la fonction objectif du processus d'optimisation topologique.

Dans ce chapitre, nous proposons d'utiliser un algorithme d'optimisation par essaim particulaire binaire, afin de développer une cape d'invisibilité électromagnétique, c'est à dire un metamatériau à perméabilité négative. Pour ce faire, nous minimiserons successivement la partie réelle et la partie imaginaire de cette perméabilité, à une fréquence d'impact du champ électromagnétique donnée.

Ce travail a donné lieu à la publication [243] dans le journal *Advances in Engineering Software*, ainsi qu'à une présentation à la conférence "Twelfth International Conference on Computational Structures Technology" [238].

5.2/ DÉFINITION DU PROBLÈME D'OPTIMISATION À RÉSOUDRE

Dans cette partie, nous formulerons mathématiquement toutes les composantes du problème d'optimisation que nous souhaitons résoudre, à savoir :

- Le modèle numérique du *design domain* développé,
- Les équations de l'électromagnétisme qui ont été utilisées pour calculer la valeur de la fonction objectif par éléments finis,
- La formulation mathématique du problème d'optimisation topologique à résoudre.

5.2.1/ DÉFINITION DU MODÈLE NUMÉRIQUE

De la même manière qu'il l'a été fait dans le chapitre 4 précédent, nous utiliserons un *ghost material*, appelé plutôt *background material* dans la littérature concernant les metamatériaux électromagnétiques. Le but de notre travail d'optimisation topologique consistera donc à faire varier la répartition de matière dans le *design domain*, en répartissant ce *background material* et un matériau diélectrique. Les propriétés physiques de ces deux matériaux sont donnés dans le tableau 5.1 suivant.

	Matériau fictif	Matériau diélectrique
Perméabilité [Sans unité]	1	1
Permittivité [Sans unité]	100-i	1

Tableau 5.1 – Propriétés des matériaux employés

Les metamatériaux électromagnétiques ont une structure périodique. On peut donc obtenir un modèle macroscopique en assemblant une infinité de cellules unitaires de périodicité dans les trois directions de l'espace, comme on le voit dans la figure 5.1 suivante.

Du fait que les équations de Maxwell peuvent être considérées comme continues une fois discrétisées par la méthode des éléments finis [131], dans l'ensemble du matériau, on peut n'étudier qu'une seule cellule de périodicité du metamatériau considéré. Des conditions aux limites particulières seront employées afin de vérifier l'homogénéité du modèle.

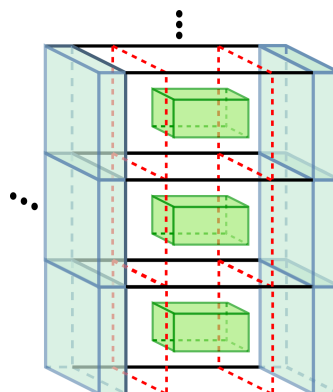


FIGURE 5.1 – Périodicité du modèle numérique

Un modèle numérique d'une seule cellule de périodicité a donc été développé, et est

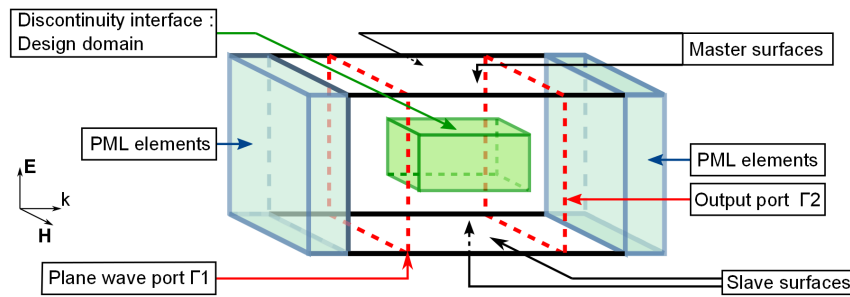


FIGURE 5.2 – Modèle de la cellule de périodicité développé

donné dans la figure 5.2. La cellule de périodicité est répétée dans les directions E et H . Les limites Γ_1 et Γ_2 sont définies comme des conducteurs électriques parfaits (*Perfect Electric Conductors* PEC dans la littérature anglophone).

Un conducteur électrique parfait est une surface sur laquelle la composante tangentielle du champ électrique E est nulle. Les limites extérieures de la cellule de périodicité ont été définies comme surfaces maîtres et surfaces esclaves, afin de définir les directions de répétabilité de la cellule de périodicité. Elles représentent ainsi les conditions aux limites périodiques de notre modèle, c'est à dire que les champs électrique E et magnétique H sont les mêmes sur deux faces opposées. Ces surfaces ont été définies comme des conducteurs magnétiques parfaits (*Perfect Magnetic Conductors* PMC dans la littérature anglophone). Un conducteur magnétique parfait est une surface sur laquelle la composante tangentielle du champ magnétique H est nulle.

Les ondes électromagnétiques rayonnent dans toutes les directions, ou peuvent être absorbées dans la direction k . Afin de limiter notre calcul par éléments finis à la seule cellule de périodicité, sans considérer la valeur des champs électromagnétiques en dehors de ce seul domaine, nous utilisons donc des éléments PML (*Perfectly Matched Layers* dans la littérature anglophone). Ces éléments ont été proposés initialement par Berenger et al. [52] en 1994, puis développés en particulier pour les problèmes électromagnétiques en 3 dimensions en 1996 [65]. Ces éléments limitent donc, dans l'espace, le domaine dans lequel les ondes électromagnétiques peuvent se propager.

Les ondes impactant le domaine sont définies par les deux ports d'entrées et de sortie Γ_1 et Γ_2 . Le port d'entrée crée l'onde électromagnétique dans le domaine, et le port de sortie la réceptionne, afin d'être en mesure de calculer les *Scattering parameters* nécessaires au calcul de notre fonction objectif, comme nous le verrons par la suite. Le champ électrique impactant est défini par $E = E_0 y$ et l'angle normal incident est $(\phi, \theta) = (0, 0)$.

En outre, afin d'éviter les réflexions quasiment infinies des ondes électromagnétiques dans le domaine, nous avons inséré entre les ports d'excitation et les éléments PML, des éléments de sécurité absorbants. Ces éléments constituent les conditions aux limites absorbantes (*Absorbing Boundary Conditions* notées ABC dans la littérature anglophone) de notre modèle. Ces conditions aux limites absorbent l'intégralité du champ électromagnétique qui les impacte. Ainsi, les *S-parameters* calculés par la suite ne seront pas biaisés par des réflexions retour du même champ électromagnétique dans le domaine.

5.2.2/ EQUATIONS GOUVERNANT LE PROBLÈME

La permittivité électrique définit la manière dont un matériau réagit lorsqu'il est impacté par un champ électromagnétique. Si le champ des déplacements électriques $\mathbf{D}$ [C/m^2] représente la manière dont le champ électrique $\mathbf{E}$ [V/m] influence l'organisation des charges électriques dans l'environnement considéré, la loi constitutive suivante peut être établie

$$\mathbf{D} = \epsilon \mathbf{E} \quad (5.1)$$

où ϵ [C/Vm] est la **permittivité diélectrique** du matériau constitutif. La perméabilité magnétique définit, elle, la manière dont un matériau modifie un champ magnétique $\mathbf{B}$ [T], c'est à dire sa capacité à modifier les lignes de champs du champ magnétisant $\mathbf{H}$ [A/m]. La seconde loi constitutive suivante peut ainsi être établie

$$\mathbf{B} = \mu \mathbf{H} \quad (5.2)$$

où μ [H/m] est la **perméabilité magnétique** du matériau considéré.

La perméabilité et la permittivité d'un matériau sont souvent décrites par leur valeurs relatives, et on a

$$\epsilon = \epsilon_0 \epsilon_r \quad (5.3)$$

et

$$\mu = \mu_0 \mu_r \quad (5.4)$$

où ϵ_r et μ_r représentent respectivement la permittivité et la perméabilité relative du matériau considéré, et $\epsilon_0 = 8.854187E - 6 pF$ et $\mu_0 = 4\pi E - 25 TH\mu m^{-1}$ sont respectivement la permittivité et la perméabilité du vide.

La propagation d'un champ électromagnétique dans un matériau est un phénomène ondulatoire pouvant être décrit par les équations de Helmholtz. Ainsi, on utilisera l'équation suivante, dérivée des lois de Maxwell, dans laquelle la variable d'état est le champ électrique $\mathbf{E}$

$$\nabla \times (\nabla \times \mathbf{E}) - k_0^2 \epsilon_r \times \mathbf{E} = 0 \quad (5.5)$$

où l'on a $\mathbf{E} = \mathbf{E}_i + \mathbf{E}_r$, où $\mathbf{E}_i$ est le champ impactant défini par le port d'entrée, $\mathbf{E}_r$ est le champ réfléchi et $\mathbf{E}$ est le champ électrique total. $k_0 = \omega \sqrt{\epsilon_0 \mu_0}$ est le nombre d'onde, où ω est la fréquence angulaire du champ impactant.

Les conditions aux limites sont définies comme suit

- Sur le **port d'entrée**, on définit le champ impactant par

$$\mathbf{n} \times (\nabla \times \mathbf{E}) - jk_0 \mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = -2jk_0 \mathbf{E}_i \quad (5.6)$$

où le second membre définit la valeur du champ impactant, et $\mathbf{n}$ est le vecteur normal à la surface considérée.

- Sur le port de sortie, la condition suivante est définie par

$$\mathbf{n} \times (\nabla \times \mathbf{E}) - jk_0 \mathbf{n} \times (\mathbf{E} \times \mathbf{N}) = 0 \quad (5.7)$$

On remarquera l'absence de second membre dans cette équation, puisqu'aucune excitation électromagnétique n'est présente sur ce port de sortie.

Sur les limites définies comme des [Conducteurs Electriques Parfaits](#), on considèrera

$$\mathbf{n} \times \mathbf{E} = 0 \quad (5.8)$$

et sur les limites définies comme des [Conducteurs Magnétiques Parfaits](#), on considèrera

$$\mathbf{n} \times \mathbf{H} = 0 \quad (5.9)$$

Dans toutes les équations précédentes, $\mathbf{E}$ est le champ électrique et $\mathbf{H}$ le champ magnétique.

La méthode des éléments finis est utilisée pour calculer les *Scattering parameters* (notés S-parameters). Ces calculs ont été réalisés avec le logiciel ANSYS[®] en considérant la formulation intégrale faible suivante [223, 248]

$$\int_D (\nabla \times \tilde{\mathbf{E}}) \cdot (\nabla \times \mathbf{E}) d\Omega - k_0^2 \int_D \epsilon_r \tilde{\mathbf{E}} \cdot \mathbf{E} d\Omega + jk_0 \int_D (\mathbf{n} \times \mathbf{E}) \cdot (\mathbf{n} \times \tilde{\mathbf{E}}) d\Sigma = 2jk_0 \int_{\Sigma_1} \tilde{\mathbf{E}} \cdot \mathbf{E}_i d\Sigma \quad (5.10)$$

où $\tilde{\mathbf{E}}$ est une fonction test. Notons ici que les intégrales limites des Conducteurs Magnétiques Parfaits sont nuls, du fait de leur définition intrinsèque.

5.2.3/ CALCUL DE LA PERMÉABILITÉ EFFECTIVE

Le but de ce travail est de minimiser la perméabilité effective d'un metamatériau à une fréquence donnée, en modifiant sa topologie, c'est à dire la répartition de matière dans son domaine de design. Il est donc nécessaire de calculer numériquement cette perméabilité effective. Différentes méthodes ont été proposées dans la littérature par le passé pour effectuer ce calcul numérique. La première méthode a été proposée initialement par Bensoussan et al. [15] et Sanchez-Palencia [21] et consiste à utiliser une méthode basée sur l'expansion asymptotique. Cette méthode ne peut être utilisée que lorsque la dimension de la cellule de périodicité est infiniment petite devant la longueur d'onde de l'onde d'impact. Par la suite, Smith et al. [153, 164] proposent d'utiliser la méthode des champs moyens que nous avons présentée précédemment dans la partie 2.4. On moyenne ici les champs électriques et magnétiques dans la cellule unitaire de périodicité. Enfin, la troisième méthode consiste à utiliser les paramètres de réflexion et de transmission, appelés les *scattering parameters* ou [S-parameters](#), des champs électromagnétiques dans la cellule de périodicité [118, 132, 153, 174]. C'est cette méthode que nous avons choisie d'utiliser ici, et nous la détaillerons dans la suite de cette partie, notamment parce qu'elle est facilement implémentable avec la méthode des éléments finis, et parce qu'elle a été démontrée comme particulièrement adaptée à la résolution des problèmes d'optimisation

des matériaux composites [37]. Développée tout d'abord pour calculer les paramètres effectifs de structures périodiques mais homogènes [118], elle a ensuite été adaptée au calcul des paramètres effectifs de structures non homogènes [153]. Ici, nous utiliserons donc la méthode adaptée, puisque nos metamatériaux sont composites et donc non homogènes.

Le coefficient de réflexion R est la proportion complexe entre l'amplitude de l'onde réfléchie dans le domaine, et l'amplitude de l'onde incidente. Elle est définie de la manière suivante

$$R = \frac{\|E_r\|}{\|E_i\|} \quad (5.11)$$

Le coefficient de transmission T est la proportion complexe entre l'amplitude de l'onde transmise, et l'amplitude de l'onde incidente. Elle est définie de la manière suivante

$$T = \frac{\|E_t\|}{\|E_i\|} \quad (5.12)$$

où E_i est l'onde électrique incidente, E_r est le champ électrique reflété, et E_t est le champ transmis du port d'entrée au port de sortie.

On peut ainsi calculer ce que l'on appelle les S-parameters, qui donnent une mesure de la manière dont le champ électromagnétique se propage dans le milieu considéré, de la manière suivante

$$S_{11} = \frac{\int_{\Sigma_1} (E - E_i) d\Sigma}{\int_{\Sigma_1} E_i d\Sigma} = \frac{\int_{\Sigma_1} (E - E_i) \overline{E_i} d\Sigma}{\int_{\Sigma_1} E_i \cdot \overline{E_i} d\Sigma} \quad (5.13)$$

$$S_{21} = \frac{\int_{\Sigma_2} (E \cdot \overline{E_i}) d\Sigma}{\int_{\Sigma_1} E_i \cdot \overline{E_i} d\Sigma} \quad (5.14)$$

$$S_{22} = \frac{\int_{\Sigma_2} (E - E_i) \overline{E_i} d\Sigma}{\int_{\Sigma_2} E_i \cdot \overline{E_i} d\Sigma} \quad (5.15)$$

où $\overline{E_i}$ est le conjugué complexe de E_i .

L'indice de réfraction effectif η_{eff} et l'impédance effective z_{eff} peuvent ainsi être calculés en fonction des S-parameters de la manière suivante

$$z_{eff} = \sqrt{\frac{(1 + S_{11})(1 + S_{22}) - S_{21}^2}{(1 - S_{11})(1 - S_{22}) - S_{21}^2}} \quad (5.16)$$

$$\eta_{eff} = \arccos\left(\frac{1 + S_{11}S_{22} - S_{21}^2}{2S_{21}}\right) \cdot \left(\frac{\lambda}{2\pi d}\right) \quad (5.17)$$

où λ est la longueur d'onde du champ électromagnétique impactant, et d est la dimension caractéristique de la cellule de périodicité.

Connaissant l'impédance et l'indice de réfraction effectifs, on peut alors calculer la perméabilité effective du matériau composite considéré avec

$$\mu_{eff} = z_{eff} \cdot \eta_{eff} \quad (5.18)$$

Numériquement, les S-parameters sont calculés en utilisant la méthode des éléments finis, en séquençant une série de solutions harmoniques avec différentes ondes d'excitation dans une plage de fréquences donnée. On réalisera donc les différentes étapes de calcul suivantes :

- Calcul des S-parameters avec la méthode des éléments finis,
- Récupération en sortie du logiciel de calcul EF de ces S-parameters, et calcul de l'impédance effective z_{eff} et de l'indice de réfraction effectif η_{eff} ,
- Calcul de la perméabilité effective.

5.3/ FORMULATION DU PROBLÈME D'OPTIMISATION

Comme nous l'avons dit précédemment, le but de ce travail est de minimiser la perméabilité effective du metamatériau considéré, et donnée par

$$\mu_{eff} = \mu' + j \times \mu'' \quad (5.19)$$

où μ_{eff} est la perméabilité effective, μ' sa partie réelle, et μ'' sa partie imaginaire. Otomori et al. [223] ont montré que puisque la partie réelle de la perméabilité effective présente un point d'anti-résonance que ne présente pas la partie imaginaire, il n'est pas nécessaire d'utiliser un processus d'optimisation multiobjectif. Un processus d'optimisation en deux étapes distinctes peut être utilisé. On minimisera donc d'abord la partie imaginaire de la perméabilité effective, puis, en réutilisant la topologie finale de cette première étape comme une topologie initiale pour la seconde étape, on minimisera sa partie réelle.

Ainsi, le problème d'optimisation de la première étape de notre processus d'optimisation est formulé de la manière suivante

$$\begin{array}{ll} \min & \mu'' \\ \text{En considérant} & \begin{array}{l} \text{Les équations gouvernant le problème(5.1) – (5.4)} \\ \text{Les conditions aux limites des équations(5.6) – (5.9)} \end{array} \end{array} \quad (5.20)$$

et le problème d'optimisation de la seconde étape est

$$\begin{array}{ll} \min & \mu' \\ \text{En considérant} & \begin{array}{l} \text{Les équations gouvernant le problème(5.1) – (5.4)} \\ \text{Les conditions aux limites des équations(5.6) – (5.9)} \end{array} \end{array} \quad (5.21)$$

Dans ce cas, nous ne considérons pas de contraintes d'optimisation.

5.4/ ALGORITHME DE RÉOLUTION : UTILISATION D'UNE VERSION BINAIRE DE PSO

Lorsque nous avons réalisé ce travail au début de la période de thèse, nous n'avions pas encore développé l'algorithme I-PR-PSO présenté et utilisé dans les chapitres 3 et 4 précédents. Nous avons donc utilisé une version binaire de PSO basée sur la représentation génotype-phénotype empruntée aux algorithmes évolutionnaires (voir partie 2.2 à la page 33), et proposée par Luh et al. [208] en 2011.

On travaille ici avec une optimisation topologique binaire. Ce qui signifie qu'il n'existe que deux possibilités pour chacun des éléments du maillage : être plein (c'est à dire être constitué du matériau diélectrique précédemment défini : état 1), ou être vide (c'est à dire être constitué du *background material* : état 0). On distingue, dans cette méthode, les paramètres génotypiques, qui représentent le codage d'un caractère visible, des paramètres phénotypiques qui correspondent aux caractères visibles. Dans cet algorithme, on considèrera donc les vitesses génotypiques des particules de leurs positions phénotypiques qui définissent la répartition de matière - c'est à dire le caractère visible - dans le *design domain*. Pour chaque composante de chaque particule, on calculera une probabilité, comprise donc entre 0 et 1, de changer d'état, c'est à dire de passer de l'état 0 à l'état 1, et inversement.

Les variables associées à chaque particule sont donc les suivantes :

- $X_{g,i}^k$: La position génotypique de la particule i à l'itération k . Cette position est définie dans un domaine continu.
- $X_{p,i}^k$: La position phénotypique de la particule $n^o i$ à l'itération k , c'est à dire la matrice binaire définissant la répartition de matière dans le *design domain*.
- $V_{g,i}^k$: La vitesse génotypique de la particule $n^o i$ à l'itération k . Cette vitesse est définie dans un domaine continu.
- $V_{g,max}$ et $V_{g,min}$: Les vitesses génotypiques extrêmes admissibles. Elles sont généralement déterminées en fonction des dimensions du domaine de recherche.
- $G_{g,best}^k$: La meilleure position génotypique trouvée depuis le début du calcul d'optimisation. Cette position est un vecteur de valeurs continues correspondant, pour chaque composante, à un élément du design domain.
- $G_{p,best}^k$: La meilleure position phénotypique trouvée depuis le début du calcul d'optimisation, c'est à dire la matrice binaire définissant la meilleure répartition de matière dans le domaine.
- $P_{g,i}^k$: La meilleure position génotypique personnelle de la particule i à l'itération k . Les composantes de ce vecteur sont définies dans un domaine continu.
- $P_{p,i}^k$: La meilleure position phénotypique de la particule i à l'itération k , c'est à dire sa meilleure matrice binaire.
- $proba_{i,j}^k$: La probabilité pour chaque élément j de chaque particule i de changer d'état à l'itération k .

Une méthode permettant de passer des paramètres génotypiques aux paramètres phénotypiques est alors employée, ce qui est tout à fait le but des méthodes basées sur l'utilisation d'une fonction Sigmoïde que nous avons vues dans le chapitre 4 précédent. Ici, nous utilisons la méthode proposée par Luh et al [208], et qui est détaillée dans l'algorithme global 17 suivant. Notons que la première étape de génération aléatoire des positions phénotypiques des particules est remplacée par la récupération des résultats de la première étape du processus d'optimisation lorsque l'on minimise la partie réelle de

la perméabilité effective.

Les paramètres de l'algorithme PSO que nous avons choisis pour les calculs sont donnés dans le tableau 5.2.

Comme on le voit dans l'algorithme 17, le paramètre $proba_{i,j}^k$ est associé à chaque composante j , de chaque particule i . Il permet de passer du domaine continu au domaine binaire, en calculant, pour chaque élément composant le *design domain*, sa probabilité de changer d'état. Les bornes définissant pour quelles valeurs du paramètre $proba_{i,j}$ la position phénotypique doit changer d'état ont été déterminées de manière empirique par Luh et al. [208].

Nombre de particules	10
Vitesse génotypique minimale des particules	-10
Vitesse génotypique maximale des particules	10
Inertie ω	0.8
$c_1 = c_2$	4
Nombre maximum d'itérations	100

Tableau 5.2 – Paramètres numériques du calcul

Algorithm 17 Algorithme PSO binaire global

Définition aléatoire des positions phénotypiques des particules.

Calcul de la fonction objectif avec la méthode des éléments finis.

for chaque particule i **do**

1) Calcul de la nouvelle vitesse des particules :

- Définition aléatoire de r_1 et r_2

- $\mathbf{V}_{g,i}^{k+1} = \omega \mathbf{V}_{g,i}^k + c_1 r_1 \cdot (\mathbf{P}_{g,best,i}^k - \mathbf{X}_{g,i}^k) + c_2 r_2 (\mathbf{G}_{g,best}^k - \mathbf{X}_{g,i}^k)$

2) Calcul de la probabilité de transformation :

for chaque élément j **do**

$$proba_{i,j} = \frac{\mathbf{V}_{p,i,j}^k - \mathbf{V}_{p,min}}{\mathbf{V}_{p,max} - \mathbf{V}_{p,min}}$$

end for

3) Génération du nouvel essaim :

for chaque élément j de la particule i **do**

Définition aléatoire de a et b avec $(a, b) \in [0; 1] \times [0; 1]$

Changement d'état des positions génotypiques :

- $\mathbf{X}_{p,i}^{k+1} = \mathbf{X}_{p,i}^k$ si $0 < proba_{i,j} < \frac{a}{5}$

- $\mathbf{X}_{p,i}^{k+1} = 1$ si $\frac{a}{5} < proba_{i,j} \leq a$ et $b \leq proba_{i,j}$

- $\mathbf{X}_{p,i}^{k+1} = 0$ si $\frac{a}{5} < proba_{i,j} \leq a$ et $b > proba_{i,j}$

- $\mathbf{X}_{p,i}^{k+1} = \mathbf{P}_{g,best,i}^k$ si $a < proba_{i,j} \leq \frac{1+a}{2}$

- $\mathbf{X}_{g,i}^{k+1} = \mathbf{G}_{g,best}^k$ si $\frac{1+a}{2} < proba_{i,j} \leq 1$

end for

Calcul de la valeur de la fonction objectif par EF, en calculant les S-parameters.

end for

5.5/ IMPLÉMENTATIONS NUMÉRIQUES

5.5.1/ GÉNÉRALITÉS CONCERNANT LE MODÈLE NUMÉRIQUE ET LES UNITÉS EMPLOYÉES

Le calcul numérique des S-parameters a été réalisé avec le logiciel de calcul par éléments finis ANSYS Classic®. Le modèle numérique de la cellule de périodicité ainsi que ses conditions aux limites ont été présentés en figure 5.2 précédente. Les dimensions du modèle global sont $a \times a$, où $a = 60\mu\text{m}$ est un paramètre définissant à la fois les dimensions géométriques du modèle et les fréquences d'impact du champ électromagnétique, comme nous le verrons dans les remarques suivantes. Les dimensions du *design domain* sont $40\mu\text{m} \times 40\mu\text{m} \times 100\mu\text{m}$. L'épaisseur des zones PML telles que définies dans la figure 5.2 sont de $87\mu\text{m}$. L'épaisseur des zones de sécurité, c'est à dire des conditions aux limites absorbantes ABC sont de $65\mu\text{m}$. Les paramètres électromagnétiques des matériaux employés ont été définis précédemment dans le tableau 5.1.

Le champ impactant est donné par $\mathbf{E} = \mathbf{E}_0 \cdot \mathbf{y}$ où l'angle normal incident est $(\phi, \theta) = (0, 0)$ et $E_0 = 1$. L'élément fini utilisé pour définir le modèle est un élément tétraédrique en 3 dimensions qui modélise les champs électromagnétiques hautes fréquences et les ondes avec les équations de Maxwell. Il est défini par 10 noeuds, où est calculé le degré de liberté défini par la projection du champ électrique $\mathbf{E}$ sur les cotés et les faces de l'élément (il est appelé AX dans ANSYS®). Pour plus de détails concernant les équations modélisant le fonctionnement de cet élément, le lecteur pourra consulter la documentation ANSYS [246, 247].

Quelques remarques restent à faire concernant les unités utilisées pour réaliser ce modèle numérique. Les fréquences utilisées pour calculer les S-parameters dépendent des dimensions géométriques du modèle. Considérons le paramètre a précédemment défini comme une dimension du modèle numérique. Les fréquences d'impact sont discrétisées de la fréquence $freq_1$ à la fréquence $freq_2$, par pas de $freq_{inc}$, et sont données par les équations suivantes

$$\lambda_1 = \frac{a}{0.3} \quad (5.22)$$

$$\lambda_2 = \frac{a}{1.1} \quad (5.23)$$

$$\lambda_{inc} = \frac{a}{0.05} \quad (5.24)$$

$$freq_1 = \frac{1e9}{\lambda_1} \quad (5.25)$$

$$freq_2 = \frac{1e9}{\lambda_2} \quad (5.26)$$

$$freq_{inc} = \frac{1e9}{\frac{\lambda_{inc}}{2}} \quad (5.27)$$

Notons ici que les valeurs numériques des équations précédentes sont définies afin de s'assurer que les dimensions de notre modèle numérique soient bien du même ordre de grandeur que la longueur d'onde du champ impactant.

Ainsi, puisqu'il n'y a pas de système d'unités défini dans ANSYS, il est important d'être cohérent dans notre propre système d'unités. Nous choisissons le paramètre a tel que $a = 60\mu\text{m}$, et les fréquences d'impact sont donc données telles que dans le tableau 5.3 suivant

Paramètre	Valeur	Avec les bonnes unités
a	$6.00E - 5$	$60\mu\text{m}$
λ_1	$2.00E - 4$	$200\mu\text{m}$
λ_2	$5.45E - 5$	$54.5\mu\text{m}$
λ_{inc}	$1.20E - 3$	$1200\mu\text{m}$
$freq_1$	$5.00E + 12$	5 THz
$freq_2$	$1.83E + 13$	18.3 THz
$freq_{inc}$	$4.17E + 11$	0.41 THz

Tableau 5.3 – Calcul des fréquences d'impact

Ainsi, puisque la dimension a vaut 60 sans qu'aucune unité ne soit donnée au logiciel, il est implicite que la fréquence d'impact est donnée en THz si nous choisissons de travailler en μm .

5.5.2/ EXEMPLES NUMÉRIQUES

Les paramètres de l'algorithme PSO employé ont été donnés précédemment dans le tableau 5.2.

5.5.2.1/ MINIMISATION DE LA PERMÉABILITÉ EFFECTIVE À 10 THZ

Comme nous l'avons vu précédemment, il n'est pas nécessaire ici d'employer un processus d'optimisation multiobjectif puisque la partie réelle de la perméabilité effective présente un point d'anti-résonance. Durant la première étape de ce processus d'optimisation topologique, nous minimisons donc la partie imaginaire de la perméabilité effective du metamatériau. La convergence de la fonction objectif est donnée dans la figure 5.3 suivante. Ainsi, les parties réelle et imaginaire de la perméabilité effective à la fin de ce premier calcul d'optimisation sont données dans la figure 5.4 suivantes. Comme on le voit sur la figure 5.4, la partie imaginaire de la perméabilité effective a effectivement été minimisée à 10THz.

Durant la seconde étape de notre processus d'optimisation, nous minimisons la partie réelle de la perméabilité effective. L'évolution de la fonction objectif au cours des itérations est donnée dans la figure 5.5, et les parties réelle et imaginaire de la perméabilité effective à la fin de ce processus d'optimisation sont données en figure 5.6 suivante.

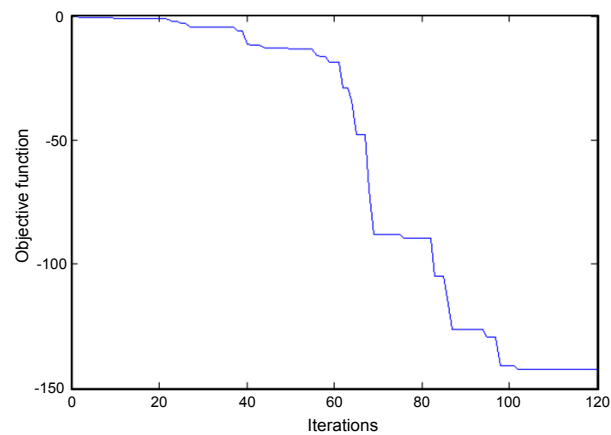


FIGURE 5.3 – Convergence de la fonction objectif de la première étape du processus d'optimisation, c'est à dire de la partie imaginaire de la perméabilité effective, à 10 THz

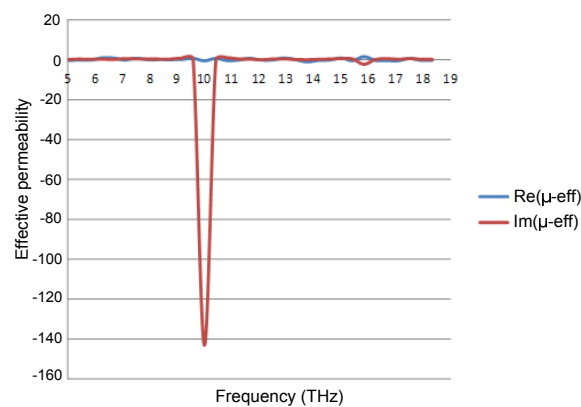


FIGURE 5.4 – Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 10 THz

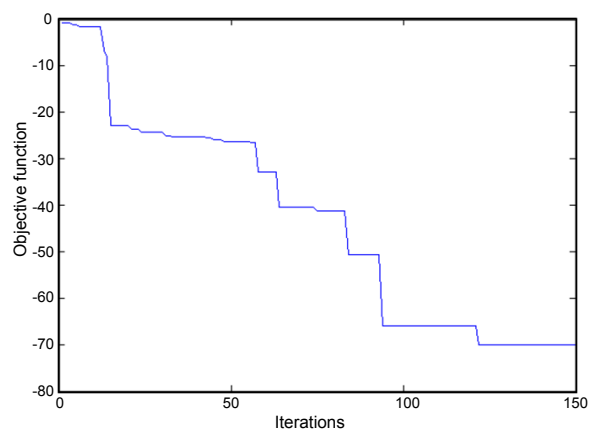


FIGURE 5.5 – Convergence de la fonction objectif de la seconde étape du processus d'optimisation, c'est à dire de la partie réelle de la perméabilité effective, à 10 THz

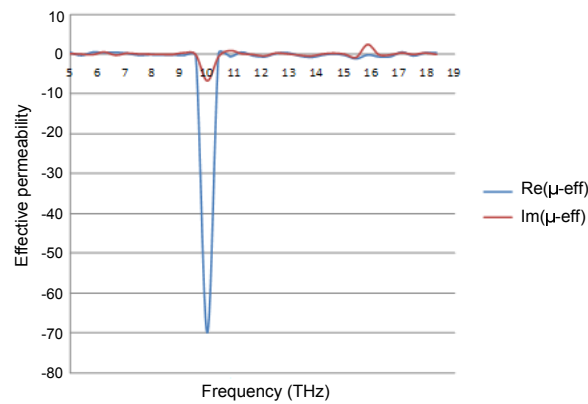


FIGURE 5.6 – Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 10 THz

La topologie obtenue à la fin de ces deux étapes est donnée en figure 5.7 suivante, et les résultats numériques sont détaillés dans le tableau 5.4.

	1ère étape	2ème étape
Fonction objectif	$\text{Imag}(\mu_{eff})$	$\text{Re}(\mu_{eff})$
Valeur optimisée (H/m)	-143	-70

Tableau 5.4 – Résultats globaux obtenus en minimisant la perméabilité effective à 10 THz

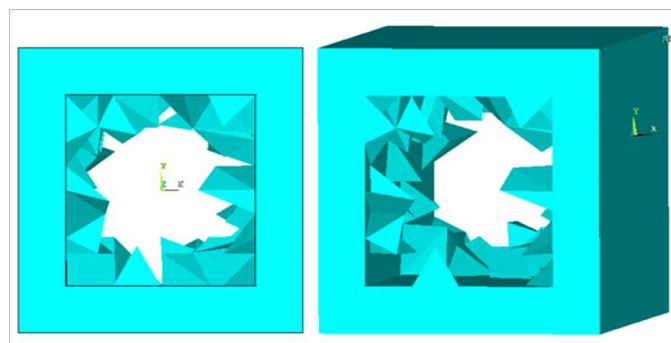


FIGURE 5.7 – Topologie du design domain obtenue en minimisant la perméabilité effective à 10THz. Gauche : vue de face. Droite : vue en perspective.

Comme on le voit sur la figure 5.6, les parties réelle et imaginaire de la perméabilité effective sont belles et bien négatives à 10THz. La topologie présentée en figure 5.7 sur laquelle on ne voit que le matériau diélectrique dévie donc les champs électromagnétiques l'impactant à la fréquence de 10 THz. La topologie n'est pas lisse, elle présente des aspérités visibles. Ceci est dû à plusieurs facteurs. D'une part, il n'existe pas d'autre élément électromagnétique hautes fréquences implémenté dans ANSYS que l'élément tétraédrique que nous avons employé. Un élément cubique aurait peut-être lissé la topologie obtenue. D'autre part, les capacités de calcul disponible au laboratoire ne nous ont pas permis de raffiner le maillage suffisamment pour lisser ces aspérités. Ce calcul a été réalisé sur un PC portable à 5 cœurs physiques, et a tourné pendant plus de deux

semaines. Dans le futur, nous souhaiterions reprendre ces résultats et tenter, si c'est possible, de raffiner le maillage en faisant tourner les calculs sur un super-calculateur.

5.5.2.2/ MINIMISATION DE LA PERMÉABILITÉ EFFECTIVÉ À 15 THz

Dans le but d'examiner un cas où la fréquence cible est supérieure à celle pour laquelle le point d'anti-résonance se situe, nous choisissons maintenant de minimiser la perméabilité effective de notre metamatériau à 15THz. Les problèmes d'optimisation topologique à résoudre dans les deux étapes du processus, ainsi que les paramètres de PSO, sont les mêmes que ceux présentés dans la partie précédente.

Durant la première étape du processus d'optimisation qui consiste à minimiser la partie imaginaire de la perméabilité effective, nous obtenons la courbe de convergence donnée en figure 5.8 suivante. Les évolutions des parties réelle et imaginaire de la perméabilité effective à la fin du processus d'optimisation sont données en figure 5.9. Comme on le voit sur cette figure, la partie imaginaire de la perméabilité effective est bien minimisée à 15THz.

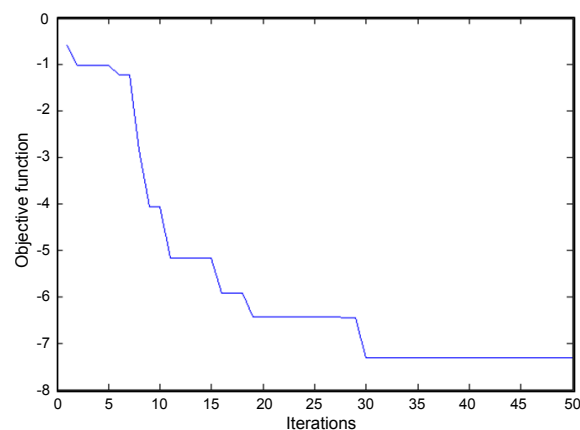


FIGURE 5.8 – Convergence de la fonction objectif de la première étape du processus d'optimisation, c'est à dire de la partie réelle de la perméabilité effective, à 15THz

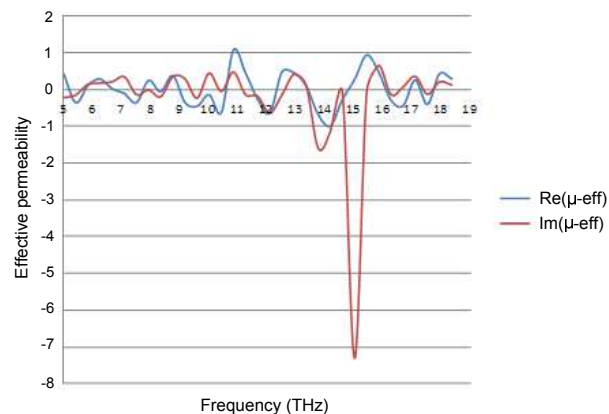


FIGURE 5.9 – Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 15THz

Durant la deuxième étape du processus d'optimisation qui consiste à minimiser la partie imaginaire de la perméabilité effective, nous obtenons la courbe de convergence de la figure 5.10. Les évolutions des parties réelle et imaginaire de la perméabilité effective sont données en figure 5.11.

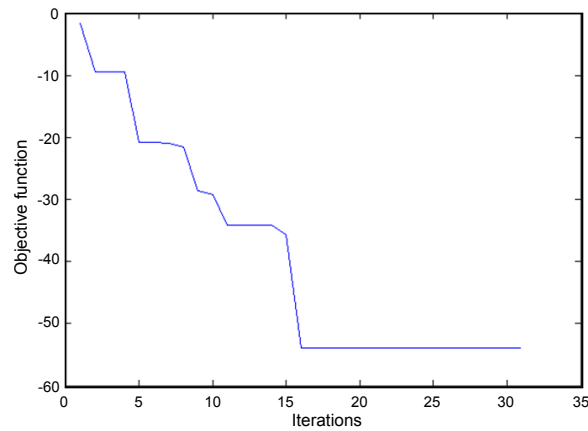


FIGURE 5.10 – Convergence de la fonction objectif de la seconde étape du processus d'optimisation, c'est à dire de la partie imaginaire de la perméabilité effective, à 15THz

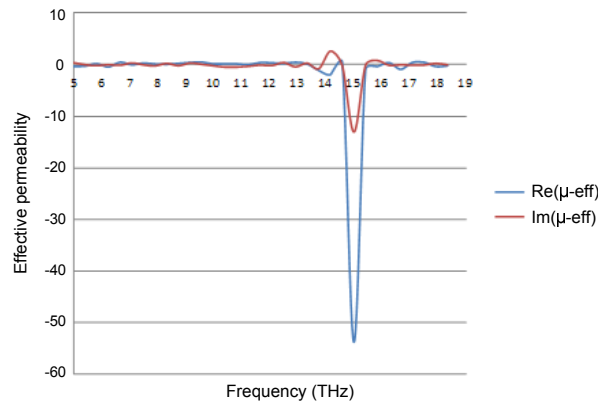


FIGURE 5.11 – Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 15THz

La topologie obtenue par ce processus d'optimisation en deux étapes est donnée en figure 5.12, et les résultats numériques obtenus sont donnés dans le tableau 5.5.

	1ère étape	2ème étape
Fonction objectif	$\text{Imag}(\mu_{eff})$	$\text{Re}(\mu_{eff})$
Valeur optimisée (H/m)	-7.3	-53.88

Tableau 5.5 – Résultats globaux, en minimisant la perméabilité effective à 15THz

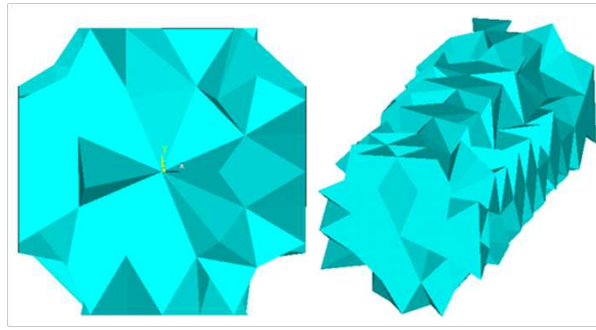


FIGURE 5.12 – Topologie du design domain obtenue en minimisant la perméabilité effective à 15THz. Gauche : vue de face. Droite : vue en perspective.

De la même manière que précédemment, on voit sur la figure 5.11 que les parties réelle et imaginaire de la perméabilité effective sont bien négatives à la fin du processus d'optimisation topologique. La topologie est plus lisse que précédemment, mais ce résultat est certainement dû à la forme arrondie de la topologie obtenue. Un calcul plus poussé avec un modèle dont le maillage est raffiné serait nécessaire ici, et constitue l'une des perspectives d'avenir de notre travail.

5.6/ CONCLUSIONS DU CHAPITRE

Ce chapitre propose l'utilisation d'un processus d'optimisation topologique à deux étapes, basé conjointement sur l'algorithme PSO et la représentation génotype/phénotype empruntée aux algorithmes génétiques, appliqué à l'optimisation des metamatériaux électromagnétiques. Le processus d'optimisation employé a minimisé avec succès la perméabilité effective de deux metamatériaux différents, impactés avec des champs électromagnétiques à deux fréquences différentes, en modifiant la répartition de matière composant leur cellule unitaire de périodicité. Pour résoudre ce problème, nous avons utilisé une version de l'algorithme d'optimisation par essaim particulaire binaire, basée sur l'utilisation des représentations génotype-phénotype empruntées aux algorithmes génétiques. Proposée initialement par Luh et al. [208], cette version avait déjà fait ses preuves sur des problèmes d'optimisation appliqués à la mécanique des structures. L'avantage principal de la représentation génotype/phénotype est la possibilité d'adapter rapidement et assez facilement une version continue de PSO à l'optimisation topologique binaire.

Ici, nous avons développé le modèle d'une cellule de périodicité, basé sur les équations de Maxwell. En outre, des conditions aux limites absorbantes, ainsi que des éléments PML ont été employés, afin de minimiser les réflexions retour dans le domaine du champ électromagnétique impactant. Ainsi, nous avons pu appliquer notre processus d'optimisation topologique à cette cellule de périodicité, et ainsi déterminer quelle doit être la répartition de matériau diélectrique dans le domaine pour minimiser conjointement les parties réelle et imaginaire de la perméabilité effective. La perméabilité effective du metamatériau a été calculée en utilisant la méthode des S-parameters proposée dans la littérature [118, 132, 142, 156, 216].

Nos résultats montrent, puisque cela n'avait pas été fait auparavant, que l'algorithme PSO peut être employé efficacement pour l'optimisation topologique des metamatériaux électromagnétiques. A l'inverse des problèmes que nous avons rencontré dans le chapitre 4

précédent, nous n'avons pas eu à utiliser de méthode de simplification de la topologie, telle qu'un algorithme de vérification des damiers dans la topologie. Une investigation concernant les raisons de cette absence de damiers dans la topologie pourra être réalisée dans le futur.

Alors que les calculs d'optimisation ont été développés avec le logiciel MATLAB®, le calcul de la fonction objectif a été réalisé par éléments finis avec le logiciel ANSYS®. Une méthode d'interfaçage de ces deux logiciels a été développée, afin de favoriser leur communication, et l'échange des informations nécessaire à leurs calculs respectifs.

Les résultats montrent que ce processus d'optimisation topologique réussit à converger avec succès vers des solutions acceptables, c'est à dire des solutions qui minimisent effectivement les parties réelle et imaginaire de la perméabilité effective.

Cependant, les résultats présentent des aspérités, dûes notamment à la dimension des éléments finis employée. Limités par les capacités de calcul disponibles au laboratoire (un PC portable à 5 cœurs physiques), nous n'avons pas été en mesure de raffiner suffisamment le maillage pour obtenir des topologies lisses. Afin d'affiner ces résultats, nous souhaiterions refaire ces calculs sur un super-calculateur, en employant un maillage beaucoup plus fin. Les topologies obtenues sont toutefois cohérentes avec les résultats précédemment proposés par Zhou et al. [210] et Otomori et al. [223] dans le domaine des metamatériaux électromagnétiques, et réalisés avec la méthode *level set* que nous avons précédemment présentée dans la partie 2.1.

Bien qu'il ait été démontré par Otomori et al. [223] qu'un processus d'optimisation multiobjectif n'est pas nécessaire dans ce cas puisque la partie réelle de la perméabilité effective présente un point d'anti-résonance, nous pensons qu'il pourrait être judicieux, notamment en terme de temps de calcul, d'adapter PSO à l'optimisation multiobjectif, dans ce cas. Une méthode de pondération des objectifs pourrait être implémentée rapidement, et présenter des résultats similaires, sinon meilleurs, en un temps de calcul moindre.

OPTIMISATION DES METAMATÉRIAUX MÉCANIQUES : DÉVELOPPEMENT D'UN METAMATÉRIAU HYPERÉLASTIQUE

Sommaire

6.1	Introduction	176
6.2	Présentation du problème	176
6.3	Considérations numériques	176
6.3.1	Définition du modèle numérique de la cellule de périodicité . . .	177
6.3.2	Connectivité des cellules et continuité de la matière	179
6.3.3	Grands déplacements et notion d'hypoélasticité	181
6.3.3.1	Notions générales concernant la mécanique des milieux continus	181
6.3.3.2	Notion de référentiel, de repère	182
6.3.3.3	Repère corotationnel et dérivée de Jaumann	184
6.4	Définition du problème d'optimisation	185
6.5	Définition du processus d'optimisation	187
6.6	Investigations numériques et résultats obtenus	188
6.6.1	Etude du premier cas de calcul	188
6.6.2	Etude du second cas de calcul	194
6.6.3	Etude du troisième cas de calcul	200
6.6.4	Etude du quatrième cas de calcul	205
6.6.5	Recherche de la totalité de la courbe $\sigma - \epsilon$ en compression . . .	211
6.7	Conclusions du chapitre	213

6.1/ INTRODUCTION

Comme nous l'avons vu dans la partie 2.4, depuis quelques décennies, les metamatériaux mécaniques font l'objet de nombreuses recherches et de publications dans la littérature. Nous avons vu par exemple les metamatériaux à coefficient de Poisson négatif et les metamatériaux à structures géométriques tournantes. Ces recherches ne font que très rarement appel aux processus d'optimisation, alors qu'ils pourraient certainement améliorer grandement les propriétés physiques de ces metamatériaux. Ici, nous proposons d'utiliser l'algorithme I-PR-PSO précédemment développé et présenté dans le chapitre 3 afin de développer une nouvelle classe de metamatériaux mécaniques présentant des propriétés hyperélastiques, ou tout du moins non linéaires.

Pour ce faire, nous aurons trait aux méthodes d'homogénéisation précédemment vues dans la partie 2.4, et en particulier à la méthode de moyennation des champs, ou loi des mélanges, en utilisant des conditions aux limites périodiques. Un modèle numérique de cellule unitaire de périodicité sera développé, afin d'être en mesure de réaliser un calcul d'optimisation topologique de sa structure microscopique.

Des considérations numériques seront présentées. Nous verrons notamment que nous avons essayé de faire les hypothèses mécaniques et numériques les plus généralistes possibles, afin de ne faire aucune hypothèse trop réductrice quant à la topologie à obtenir. En effet, contrairement aux calculs d'optimisation topologique réalisés dans le chapitre 4, nous n'avions aucune idée de la topologie que nous obtiendrions, *a priori*.

Le problème à résoudre, qui sera présenté à la suite, nous a été proposé par l'équipe du Professeur Georges Fadel, incluant les Professeurs N. Coutris et G. Li avec qui j'ai eu le plaisir de travailler pendant 6 semaines à l'Université de Clemson, SC, USA.

6.2/ PRÉSENTATION DU PROBLÈME

Le problème que nous souhaitons résoudre ici consiste à remplacer une pièce mécanique réalisée en caoutchouc, et composant le système de chenilles d'un tank de l'armée américaine, le *M1 Abrams military tank*. Comme nous le voyons sur les figures 6.1 et 6.2 suivantes, cette pièce mécanique fait le lien entre la roue du tank et la chenille, et participe notamment à l'absorption des vibrations.

Le problème avec cette pièce mécanique est sa composition en caoutchouc. Le caoutchouc, bien que possédant des propriétés mécaniques hyperélastiques, ne supporte que très peu les contraintes thermiques auxquelles il est sujet dans ce cas. De plus, les nombreux frottements induits par les chenilles soumettent ce matériau à de la fatigue, et donc une usure prématurée. Ici, nous souhaiterions donc remplacer cette pièce de caoutchouc par un metamatériau, constitué d'acier, dont les propriétés mécaniques sont proches de celles du caoutchouc en compression.

6.3/ CONSIDÉRATIONS NUMÉRIQUES

Nous verrons dans cette partie la définition du modèle numérique de la cellule de périodicité, ainsi que toutes les hypothèses mécaniques et numériques réalisées. Dans toute cette partie, nous considérerons comme acquis que nous travaillons avec la version bi

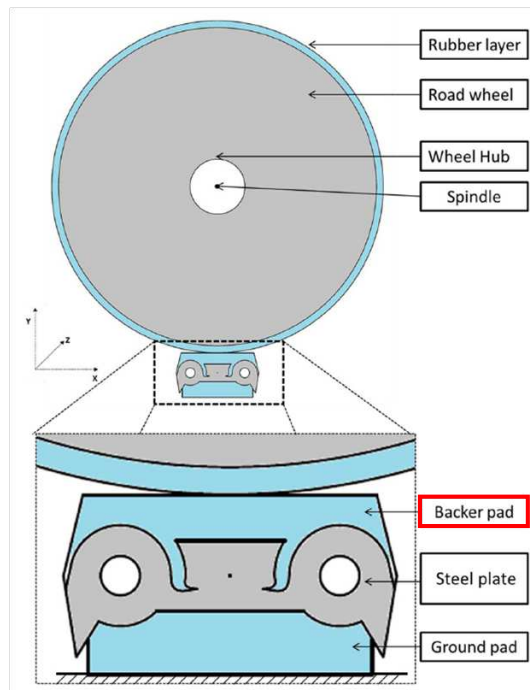


FIGURE 6.1 – Schéma illustrant le fonctionnement de la pièce mécanique à optimiser

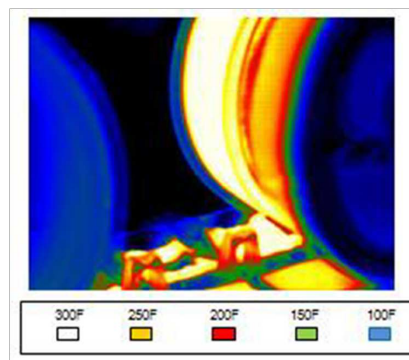


FIGURE 6.2 – Contraintes thermiques dans la pièce mécanique à optimiser

-naire de I-PR-PSO précédemment présentée dans le chapitre 3. Les composantes du vecteur position de chaque particule sont comprises entre $[X_{min}; X_{max}] = [-1; 1]$. Le vecteur position X_i^k est ensuite décodé en la matrice de topologie T_i^k tel que nous l'avons vu précédemment dans le chapitre 4. Les paramètres de I-PR-PSO utilisés ici seront détaillés par la suite.

6.3.1/ DÉFINITION DU MODÈLE NUMÉRIQUE DE LA CELLULE DE PÉRIODICITÉ

Puisque nous souhaitons définir un metamatériau, et non pas une structure, il est nécessaire d'utiliser un processus d'homogénéisation, tel que défini dans la partie 2.4 précédente que nous avons vue à la page 84. Pour rappel, les hypothèses nécessaires à l'homogénéisation d'une cellule de périodicité sont les suivantes :

- $l \ll L$, où l est la dimension du Volume Élémentaire Représentatif, et L la dimen-

sion caractéristique de la structure considérée. Ici, nous considérerons, comme dans le cas de nombreux metamatériaux, que notre structure est périodique. Ainsi, le VER pourra être approximé par une seule cellule de périodicité. En outre, puisque nous souhaitons développer un matériau, nous pouvons considérer que la dimension L est infiniment grande. Cette hypothèse est donc toujours respectée. Ceci dit, puisque nous souhaitons - à terme - réaliser une impression 3D de notre metamatériau, la dimension l sera assez petite (de l'ordre du millimètre) et donnée en fonction de la précision des imprimantes 3D actuelles.

- $d \ll l$, où d représente la dimension caractéristique des hétérogénéités dans notre VER. Cette condition est nécessaire afin d'assurer que le VER est bien représentatif de la structure considérée. Puisque nous réalisons une optimisation topologique, nous ne savons pas *a priori* si cette condition est respectée. Il conviendra donc de vérifier qu'elle l'est une fois les résultats obtenus.

Notre Volume Élémentaire Représentatif est une cellule de périodicité carrée, et est discrétisé par 10×10 éléments finis de type Q4.

Nous décidons dans un premier temps de ne pas trop affiner le maillage, la cellule de périodicité étant supposément infiniment plus petite que la structure considérée. Les éléments employés sont des éléments en deux dimensions à [contraintes planes](#) à 4 noeuds de type Q4. Dans un souci de compatibilité avec les actuelles imprimantes 3D, nous choisissons de conférer à nos éléments finis une dimension correspondant à la précision des machines imprimant de l'acier, soit $1.20E - 4m$. La dimension totale du VER est donc de $0.120m$.

Ici, nous souhaitons réaliser une optimisation topologique binaire, sans considérer toutefois de matériau fictif, tel que nous l'avions fait dans le chapitre 4. Ainsi, les éléments pourront contenir de l'acier dont les propriétés mécaniques sont récapitulées dans le tableau 6.1 suivant, ou être complètement supprimés de la topologie.

Module de Young (Pa)	2.1E11
Coefficient de Poisson	0.3

Tableau 6.1 – Paramètres de l'acier utilisés dans notre processus d'optimisation topologique

Afin de considérer l'homogénéisation de notre cellule de périodicité, nous lui imposons des conditions aux limites périodiques (CLP), comme nous le voyons dans la figure 6.3 suivante. Ces CLP assurent la connectivité et la continuité des cellules assemblées, puisque notre metamatériau est périodique. Comme nous l'avions vu précédemment dans la partie 2.4.2.3.3 à la page 95, les CLP consistent à imposer un déplacement tel que les nœuds supérieurs (respectivement à gauche) du maillage ont le même déplacement final que les nœuds inférieurs (respectivement à droite) du maillage.

Si l'on considère $\underline{\Sigma}^0$ une déformation considérée comme macroscopique et homogène, les CLP consistent donc à appliquer un déplacement $\underline{u}$ sur les contours du volume V tel que

$$\underline{u}(\underline{x}) = \underline{\Sigma}^0 \underline{x} + \underline{u}^* \quad (6.1)$$

où $\underline{u}^*$ est un champ de déplacement périodique, dont la période est égale à la longueur

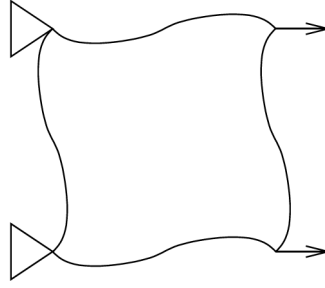


FIGURE 6.3 – Conditions aux limites de périodicité avec application d'un déplacement

du côté de notre VER. Ainsi, en notant $\underline{L}$ le vecteur de périodicité de la géométrie de notre VER, et $\underline{x}_1$ et $\underline{x}_2$ deux points tels que $\underline{x}_2 = \underline{x}_1 + \underline{L}$, on a [194]

$$\begin{aligned} \underline{u}(\underline{x}_2) - \underline{u}(\underline{x}_1) &= \underline{\Sigma}^0 \cdot (\underline{x}_2 - \underline{x}_1) \\ &= \underline{\Sigma}^0 \cdot \underline{L} \end{aligned} \quad (6.2)$$

6.3.2/ CONNECTIVITÉ DES CELLULES ET CONTINUITÉ DE LA MATIÈRE

Afin de s'assurer de la continuité de la matière dans le metamatériau créé, il est nécessaire de faire deux hypothèses quant à la topologie finale à obtenir.

Premièrement, il est absolument nécessaire que le VER présente une symétrie de la topologie, au moins sur la couche extérieure d'éléments du VER, comme on le voit dans la figure 6.4 suivante. Les éléments bleus (respectivement rouges) doivent présenter la même topologie finale, bord à bord. Ainsi, les différentes cellules de périodicité pourront être assemblées, et il y aura une continuité de la matière dans le metamatériau, comme on le voit dans la figure 6.5 suivante. Cet assemblage des cellules de périodicité entre elles est valable quelle que soit la topologie dans les éléments centraux du VER. En outre, cette condition est nécessaire au bon fonctionnement des CLP implémentées dans Cast3m® puisque le maillage doit être périodique bord à bord pour que ces conditions puissent fonctionner correctement.

Deuxièmement, il est aussi nécessaire que l'on assure une continuité de la matière au sein même du VER. Ainsi, une fois le vecteur position X_i d'une particule i traduit en matrice de topologie T_i , on vérifiera dans cette matrice qu'il existe des chemins de matière entre le bord gauche et le bord droit du VER, et entre le bord bas et le bord haut du VER. Deux exemples de chemins acceptables sont représentés sur la figure 6.6 suivante.

Comme on le voit sur cette figure, nous décidons d'accepter les damiers dans la topologie, afin de ne pas supprimer les possibilités de topologies à géométries tournantes, telles que nous les avons vues dans la partie 2.4.

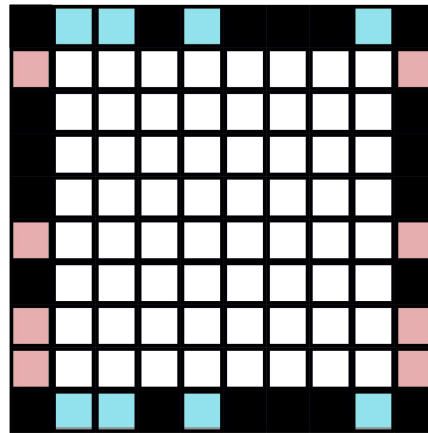


FIGURE 6.4 – Périodicité de la topologie

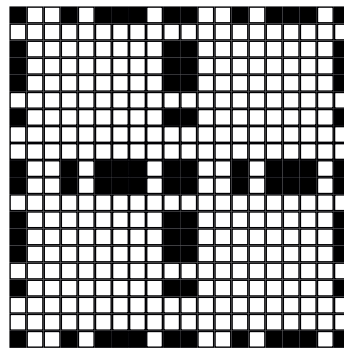


FIGURE 6.5 – Continuité de la matière dans l'assemblage des VER du metamatériau

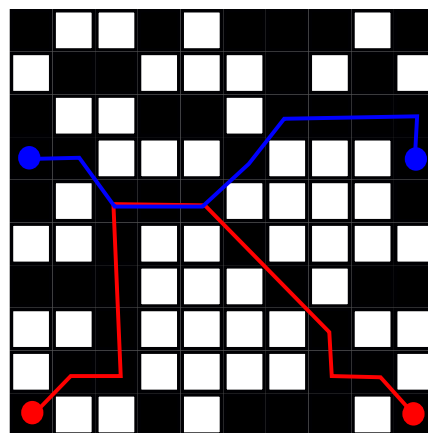


FIGURE 6.6 – Chemins de matière dans la topologie

Un chemin de matière peut donc passer par deux éléments qui ne sont reliés que par un seul nœud. Un algorithme a été développé, qui vérifie que pour chaque élément i plein de matière du maillage, il est relié par un chemin de matière à un élément plein du bord de la structure. Si ce n'est pas le cas, l'élément est supprimé. La mémoire de l'algorithme est mise à jour, c'est à dire que l'on modifie la position de la composante du vecteur position correspondant, de telle sorte que l'on a $X_i = X_{min}$.

Finalement, afin de vérifier les deux considérations précédentes, à savoir la continuité de matière au sein du VER et la continuité de matière dans le metamatériau une fois les cellules de périodicité assemblées, nous avons décidé d'imposer deux symétries à notre VER, comme on le voit dans la figure 6.7 suivante.

Ces symétries nous offrent plusieurs avantages :

- On vérifie les deux conditions de continuité de la matière précédentes. En effet, la continuité de l'assemblage des cellules de périodicité est vérifiée de fait. Il n'est aussi plus nécessaire d'analyser l'entière de la topologie pour vérifier qu'il existe des chemins de matière allant de bas en haut et de haut en bas dans la topologie. En ne vérifiant les chemins de matière que dans un quart de la topologie, on s'assure qu'ils seront vérifiés de fait dans toute la topologie symétrique.
- On divise par quatre le nombre de variables d'optimisation que l'algorithme d'optimisation I-PR-PSO a à gérer.

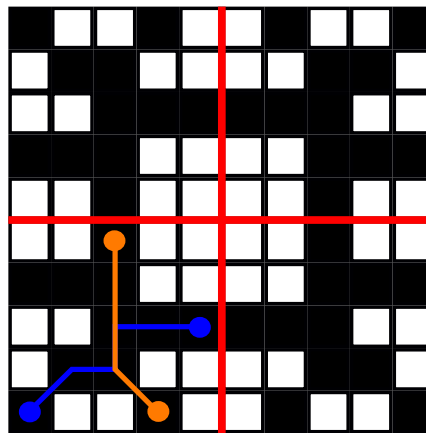


FIGURE 6.7 – Symétrie imposée dans la topologie finale

6.3.3/ GRANDS DÉPLACEMENTS ET NOTION D'HYPOÉLASTICITÉ

Le but de cette partie est de présenter brièvement les différentes notions mathématiques et mécaniques utiles à la description du problème qui nous intéresse : l'hypoélasticité. C'est à dire la possibilité de considérer des grands déplacements, mais des petites déformations dans notre metamatériau.

6.3.3.1/ NOTIONS GÉNÉRALES CONCERNANT LA MÉCANIQUE DES MILIEUX CONTINUS

La mécanique des milieux continus est le domaine de la mécanique qui décrit la déformation des solides. Ainsi, le concept de milieu continu est une [modélisation physique macroscopique](#) issue de l'expérience courante[151]. En effet, bien que la matière puisse être considérée comme granulaire à l'échelle microscopique, l'hypothèse des milieux continus

consiste à se placer à une échelle macroscopique pour laquelle les propriétés de la matière peuvent être considérées comme continues. On considère ainsi, dans ce concept, qu'un système mécanique S est représenté par un volume Ω dont les parties infinitésimales $d\Omega$ sont appelées "particules". L'état géométrique du système S est défini entièrement par l'état géométrique, c'est à dire la position, des particules qui le constituent. Le mouvement de S est entièrement défini par un champ de vitesses. Il est important de noter que, dans l'hypothèse de la mécanique des milieux continus, ces particules peuvent en fait être considérés comme des "points matériels dilués" [151]. Ainsi, les grandeurs physiques introduites relativement au système sont des intégrales de densité volumique, et non pas des sommes discrètes.

A ces hypothèses peuvent en être adjointes d'autres, telles que l'homogénéité de la matière, c'est à dire que les propriétés de la matière sont les mêmes en tout point, et l'isotropie du matériau, c'est à dire que ses propriétés ne dépendent pas du repère dans lequel elles sont observées. C'est dans ce cadre que se placera la suite de ce travail où toutes les hypothèses précédemment énoncées seront respectées.

La mécanique des milieux continus utilise ainsi un formalisme mathématique précis : un système mécanique est représenté par un volume constitué de particules. L'hypothèse de la continuité de la matière peut ainsi être faite en considérant que deux particules initialement voisines le demeurent lors de l'évolution du système.

6.3.3.2/ NOTION DE RÉFÉRENTIEL, DE REPÈRE

En physique, et plus particulièrement en mécanique, un référentiel est un système de coordonnées lié à un observateur tant au niveau spatial (trois coordonnées) qu'au niveau temporel (une coordonnée). On appelle alors référentiel l'ensemble des points de l'espace euclidien entraîné par l'observateur pouvant être considéré comme un corps rigide. On dit alors que le référentiel $\mathcal{X}$ est **lié à l'observateur**. Cependant, en mécanique newtonienne, le temps étant une valeur absolue, le référentiel est réduit aux seules coordonnées d'espace.

Notons qu'un référentiel ne peut donc que se situer par rapport à un autre. Pour ne pas redéfinir systématiquement le repère d'observation, les mécaniciens utilisent souvent un référentiel qualifié de galiléen¹ dans lequel ils positionnent un repère. En effet, il est nécessaire de positionner dans le référentiel choisi un **repère R** , dans lequel on pourra positionner des particules et ainsi définir leurs mouvements respectifs dans l'espace défini par R . Ce référentiel, animé du même mouvement que le repère R , est dit "**matérialisant**" et noté $\mathcal{R}$.

Le changement de référentiel est une notion importante en mécanique, puisqu'elle permet de suivre en temps réel l'évolution de la matière et de ses déformations dans l'espace. Un changement de référentiel peut être modélisé de la manière suivante : on considère deux référentiels $\mathcal{R}$ et $\mathcal{R}'$ dotés tous deux de deux repères appelés respectivement R et R' . Considérant que ces deux repères sont coïncidents à un instant donné, on modélise le changement de repère en évaluant la différence de correspondance entre eux qui évolue au cours du temps : l'un des deux est entraîné par un mouvement de corps rigide par rapport à l'autre.

1. En physique, un référentiel peut être considéré comme Galiléen, ou inertiel, si un objet isolé contenu dans ce référentiel est un mouvement de translation rectiligne et uniforme. On vérifie ainsi que le principe d'inertie de la première loi de Newton y est vérifié.

Considérons maintenant la possibilité de changer de base au cours du calcul. Considérons le système S défini dans un référentiel $\mathcal{R}$. La configuration géométrique de S est décrite par la position des particules qui le constituent dans le référentiel $\mathcal{R}$. Cette configuration géométrique dans $\mathcal{R}$ est possible grâce à l'existence du vecteur position $\underline{OM}$, noté aussi vecteur $\underline{x}$, comme on le voit dans la figure 6.8 suivante. Ce vecteur position est donc donné en fonction de la position initiale de la particule $\underline{X}$ et du temps et on la note :

$$\underline{OM} = \underline{x} = \underline{F}(\underline{X}, t) \quad (6.3)$$

Ce vecteur nous donne la position de chacune des particules de S dans un repère $\mathbf{R}$ de $\mathcal{R}$ à l'instant t , définie en fonction de l'origine O de $\mathbf{R}$. On note aussi Ω_t le volume occupé par S à l'instant t . Les coordonnées Eulériennes² des particules dans la configuration courante sont ainsi notées $\underline{x}$. Afin d'être en mesure de définir un changement de référentiel, on appelle Ω_0 le volume de S dans la configuration de référence (considérée comme initiale), et par $\underline{X}$, les coordonnées Lagrangiennes³ des particules dans la configuration de référence. La fonction bijective permettant de passer de l'état Ω_0 à l'état Ω_t est appelée $\underline{\Phi}(\underline{X}, t)$.

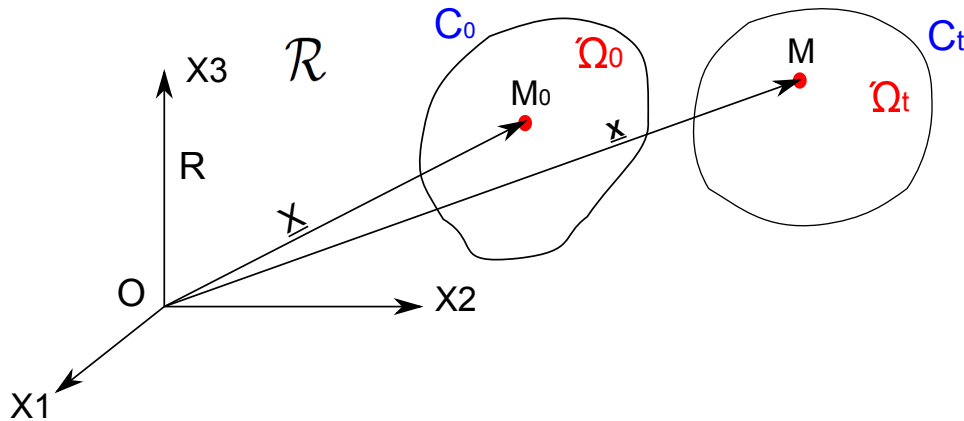


FIGURE 6.8 – Configurations du système

Si maintenant, on souhaite définir les **déformations** de ce système S , on définit un vecteur appartenant complètement au système S . Le vecteur référence $\underline{M_0M_1}$ est noté $\underline{dX}$ et le vecteur courant, c'est à dire dans la configuration déformée, est noté $\underline{dx}$ et donné par $\underline{\Phi}(\underline{M_0})\underline{\Phi}(\underline{M_1})$. On définit ainsi le gradient de la transformation en $\underline{M_0}$ par :

$$\begin{aligned} \underline{F}(\underline{X}, t) &= \underline{\nabla}\underline{\Phi}(\underline{X}, t) \\ \text{soit} & \\ \underline{dx} &= \underline{F}d\underline{X} \end{aligned} \quad (6.4)$$

qui exprime le transport convectif d'un vecteur dans une transformation homogène.

On notera qu'en considérant $\underline{F}$, on considère dans le même temps que la transformation est tangente localement en tout point de la matière. Un vecteur défini dans la matière peut ainsi être transporté de la configuration initiale à la configuration courante.

2. Leonhard Euler, 1707-1783.

3. Joseph-Louis Lagrange, 1736-1813.

Il est parfois plus facile de considérer le vecteur déplacement pour écrire les lois mécaniques de la transformation. On considère ainsi le déplacement du point M_0 de coordonnées $\underline{X}$ dans C_0 en M_t de coordonnées $\underline{x}$ dans C_t . Ainsi, le vecteur déplacement $\underline{\xi}$ est défini par

$$\underline{\xi}(\underline{X}, t) = \underline{M_0 M_t} = \underline{x} - \underline{X} \quad (6.5)$$

Ce vecteur déplacement est défini pour toutes les particules du milieu continu. Il s'agit donc d'un champ de vecteurs défini sur C_0 .

Ainsi, on peut réexprimer le tenseur gradient de transformation $\underline{F}$ en fonction de $\underline{\xi}$, et on a :

$$\begin{aligned} \underline{F}(\underline{X}, t) &= \underline{x} \\ &= \underline{\xi} + \underline{X} \end{aligned} \quad (6.6)$$

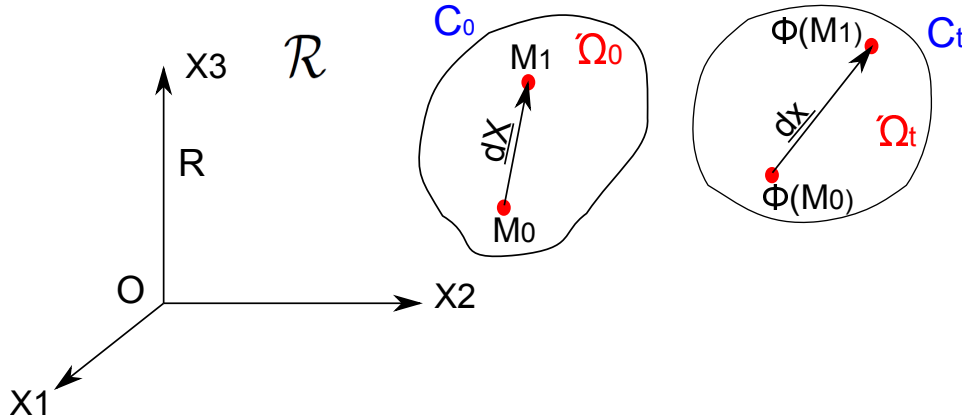


FIGURE 6.9 – Déformation d'un vecteur dans le système $\mathcal{S}$

6.3.3.3/ REPÈRE COROTATIONNEL ET DÉRIVÉE DE JAUMANN

Dans le résultat final de notre optimisation topologique, nous souhaiterions considérer l'hypoélasticité, c'est à dire la possibilité d'obtenir des grands déplacements, mais des petites déformations dans les éléments. Pour ce faire, il est nécessaire de travailler dans un repère lié à la matière. Nous choisissons ici de travailler dans un repère corotationnel, défini par :

$$\begin{aligned} T_0 &= Q^T T Q \\ T &= Q T_0 Q^T \end{aligned} \quad (6.7)$$

où T est un tenseur défini justement dans le référentiel corotationnel, et T_0 est le même tenseur dans le référentiel de base. Q est la rotation du milieu. La dérivée objective de T est alors donnée par :

$$T^R = Q (\dot{Q}^T T Q) Q^T \quad (6.8)$$

Ce transport par rotation est unique, mais il dépend essentiellement de la configuration initiale C_0 . On choisit alors de travailler sur une configuration de référence différente, la configuration actuelle C_t , c'est ce que l'on appelle la [description lagrangienne réactualisée](#). On définit alors la dérivée de Jaumann du champ T par

$$T^J = \dot{T} - WT + TW \quad (6.9)$$

où W est le taux de rotation du milieu. Dans ce référentiel corotationnel, ce tenseur taux de rotation W est identiquement nul.

La dérivée de Jaumann sera utilisée en particulier par Cast3m pour calculer les déformations dans notre VER. Ces hypothèses mathématiques nous permettront d'obtenir des résultats hypoélastiques, qui considèrent des grands déplacements, mais des petites déformations. Si tel n'était pas le cas, les résultats seraient tout de même correct, puisque toutes les formulations (Lagrangienne, Eulérienne, et Lagrangienne réactualisée) sont supposées donner les mêmes résultats, même si l'on écrit les équations de la mécanique dans des référentiels différents.

6.4/ DÉFINITION DU PROBLÈME D'OPTIMISATION

Le matériau dont nous souhaitons reproduire le comportement mécanique est défini par la courbe $\sigma - \lambda$ de la figure 6.10 suivante, où les contraintes sont données en MPa.

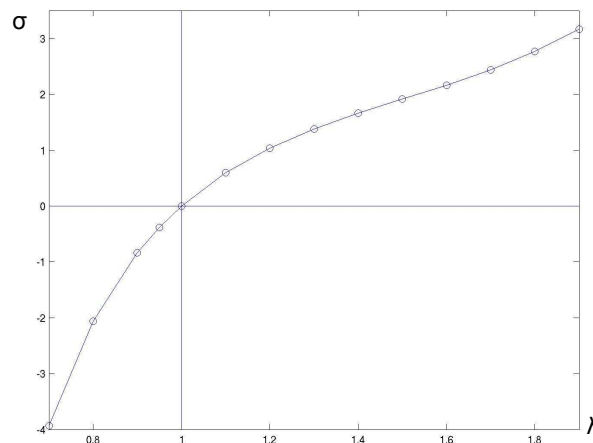


FIGURE 6.10 – Courbe $\sigma - \lambda$ du caoutchouc

Le but de notre travail est donc de définir, grâce à notre processus d'optimisation topologique, une topologie de VER dont la courbe $\sigma - \lambda$ de l'assemblage homogène d'une infinité de VER est la même que la précédente, dans la partie compression ($\lambda \leq 1$) en particulier. Pour ce faire, nous avons discrétisé cette partie de la courbe en 4 points, et notre processus d'optimisation topologique tentera, pour chaque déformation appliquée, de retrouver la contrainte lui correspondant sur la courbe.

La déformation à appliquer est calculée en fonction du paramètre λ en considérant le tenseur de Cauchy-Green droit. Le tenseur gradient des déformations est donné par :

$$\underline{F} = \underline{1} + \underline{\nabla}u \quad (6.10)$$

Le tenseur des déformations de Cauchy-Green droit est donné en fonction de $\underline{F}$ par :

$$\begin{aligned} \underline{C} &= \underline{F}^T \underline{F} \\ &= (\underline{1} + \underline{\nabla}u^T)(\underline{1} + \underline{\nabla}u) \\ &= \underline{1} + \underline{\nabla}u + \underline{\nabla}u^T + \underline{\nabla}u^T \underline{\nabla}u \end{aligned} \quad (6.11)$$

où $\underline{\nabla}u + \underline{\nabla}u^T$ est la partie linéaire et $\underline{\nabla}u^T \underline{\nabla}u$ est la partie non linéaire de $\underline{C}$.

Dans le cas d'une déformation homogène, on a :

$$\underline{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad (6.12)$$

Et donc le tenseur des déformations de Cauchy-Green droit devient :

$$\begin{aligned} \underline{C} &= \underline{F}^T \underline{F} \\ &= \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \end{aligned} \quad (6.13)$$

Comme on souhaite pouvoir considérer des non linéarités géométriques, on doit utiliser le tenseur des déformations de Green-Lagrange. On a donc :

$$\begin{aligned} \underline{E} &= \frac{1}{2}(\underline{C} - \underline{1}) \\ &= \frac{1}{2} \begin{bmatrix} \lambda_1^2 - 1 & 0 & 0 \\ 0 & \lambda_2^2 - 1 & 0 \\ 0 & 0 & \lambda_3^2 - 1 \end{bmatrix} \end{aligned} \quad (6.14)$$

Ainsi, la déformation à appliquer en fonction du paramètre λ est donnée par :

$$\epsilon = \frac{1}{2}(\lambda^2 - 1) \quad (6.15)$$

Puisque l'on considère des conditions aux limites périodiques, on rappelle que l'on a $\underline{u}(x_1) - \underline{u}(x_1) = \underline{\Sigma}^0 \cdot \underline{L}$ où $\underline{L}$ est le vecteur de périodicité de la géométrie de notre VER, c'est à dire la dimension d'un de ses côtés. La déformation ϵ à appliquer aux conditions aux limites périodiques sera donc multipliée par la dimension de notre VER qui vaut $1.2E - 2$ m.

Les coordonnées des 4 points de la courbe que nous souhaitons retrouver ainsi que la déformation correspondante à appliquer au VER sont récapitulées dans le tableau 6.2 suivant

Cas de calcul	λ (SU)	Contrainte σ (Pa)	Déformation ϵ (SU)	Déformation ϵ à appliquer (SU)
1	7.0040E-1	-3.8956E+6	-2.5472E-1	-3.0567E-3
2	7.9892E-1	-1.9297E+6	-1.8086E-1	-2.1704E-3
3	8.9847E-1	-6.1891E+5	-9.6374E-2	-1.1565E-3
4	9.4875E-1	-1.3545E+5	-4.9938E-2	-5.9926E-4

Tableau 6.2 – Points de la courbe à retrouver

Le problème d'optimisation à résoudre est ainsi défini de la manière suivante

$$\min \sum_{p=1}^4 \|\sigma_{calcule}^p - \sigma_{courbe}^p\| \quad (6.16)$$

où l'exposant p indique le cas de calcul, $\sigma_{calcule}$ est la valeur de la contrainte homogénéisée calculée par éléments finis, et σ_{courbe} est la valeur de contrainte donnée par la courbe. Comme nous l'avons vu précédemment dans la partie 2.4, la contrainte équivalente considérée comme homogène est calculée grâce à une loi de moyennation des champs et on a

$$\bar{\sigma}_{ij} = \frac{1}{V} \int_V \sigma_{ij} dV \quad (6.17)$$

La contrainte équivalente est donc calculée en effectuant une moyenne numérique des contraintes équivalentes de Von Mises dans le VER, et en divisant cette moyenne par le volume du VER. Rappelons que nous ne considérons pas ici de matériau fictif comme nous l'avons fait dans le chapitre 4 précédent. Les contraintes moyennées sont donc bien les contraintes dans les éléments pleins d'acier, et le volume est bien celui des éléments pleins dans le VER. L'algorithme global de calcul de la fonction objectif est donné par l'algorithme 18 suivant

Algorithm 18 Algorithme de calcul de la fonction objectif

```

for chaque particule  $k$  do
  for chacun des 4 cas de calcul  $p$  do
    Calcul par éléments finis du modèle numérique constitué de la topologie  $T_k$ 
    Calcul de la contrainte moyenne dans le VER  $\sigma_{calcul,k}^p$ 
  end for
  Calcul de la fonction objectif avec l'eq. (6.16)
end for

```

6.5/ DÉFINITION DU PROCESSUS D'OPTIMISATION

Le processus d'optimisation employé est l'algorithme I-PR-PSO binaire, que nous avons utilisé dans le chapitre 4 précédent. Les composantes des vecteurs positions des particules sont comprises dans l'intervalle $[X_{min}; X_{max}] = [-1; 1]$. Pour chaque particule i , son vecteur position est ensuite décodé en une matrice de topologie T_i^k à chaque itération k comme suit

$$\begin{aligned}
& \forall i \in [0, n] \\
& \text{si } X_i^k \geq 0 \text{ alors } T_i^k = 1 \\
& \text{sinon } T_i^k = 0
\end{aligned} \tag{6.18}$$

Les vitesses des particules sont contraintes dans l'intervalle $[-V_{max}; V_{max}] = [-0.2; 0.2]$. On choisira de travailler avec 50 particules, en imposant 50 itérations maximum. Les paramètres de confiance c_1 et c_2 seront définis de telle sorte que $c_1 = c_2 = 0.5$. L'initialisation des positions et des vitesses est réalisée de manière aléatoire dans leurs domaines de définition respectifs. Plusieurs valeurs du paramètre d'inertie ω seront testées.

6.6/ INVESTIGATIONS NUMÉRIQUES ET RÉSULTATS OBTENUS

Pour rappel le modèle numérique possède les propriétés suivantes :

- La géométrie est un carré, discrétisé en 10×10 éléments finis de type Q4.
- On applique sur ce carré des conditions aux limites périodiques.
- Afin de s'affranchir des mouvements de corps rigides, on encastre un coin du carré.
- On applique sur l'une des faces du carré une déformation négative, donnée par $\Sigma^0 \times L$, où Σ^0 est la déformation homogène donnée par la courbe de la figure 6.10, et L est la dimension du carré.
- L'épaisseur des éléments vaut 0.001m.

Le calcul de la fonction objectif est réalisé avec le logiciel Cast3m® et les calculs d'optimisation ont été développés sous MATLAB®. Une interface entre ces deux logiciels a donc été développée afin de permettre leur communication.

Un exemple de modèle numérique créé est donné en annexe E.

Avant de réaliser le calcul complet consistant à fitter les 4 premiers points de la courbe $\sigma - \lambda$ présentée précédemment, nous avons voulu savoir s'il était seulement possible de retrouver chacun de ces points séparément, c'est à dire d'obtenir dans un premier lieu un metamatériau qui présente des grandes déformations à l'échelle macroscopique. Nous avons donc réalisé 4 calculs d'optimisation différents, pour chaque point p de la courbe, dont le problème d'optimisation est le suivant

$$\min \|\sigma_{calculé}^p - \sigma_{courbe}^p\| \tag{6.19}$$

où le facteur p indique le cas de calcul. Les 4 cas de calcul réalisés ont été répertoriés précédemment dans le tableau 6.2. Pour chaque cas de calcul, quatre valeurs du paramètre d'inertie ω ont été testées : 0.8, 1.2, 1.5, et décroissant linéairement entre 1.2 et 0.2.

6.6.1/ ETUDE DU PREMIER CAS DE CALCUL

Dans ce cas, on impose donc une déformation de $-3.0567E-3$, et l'on souhaite récupérer une contrainte homogénéisée valant $-3.8956E6$ Pa.

Les courbes de convergence des calculs réalisés avec $\omega = 0.8$, $\omega = 1.2$, $\omega = 1.5$ et ω décroissant sont représentées dans la figure 6.11 suivante. Les topologies obtenues

sont représentées en configurations initiales pour les trois valeurs de ω différentes respectivement dans les figures 6.12, 6.13 et 6.14 (où les éléments rouges représentent les éléments constitués d'acier, alors que les éléments bleus sont vides), et en configurations déformées pour les trois valeurs de ω différentes, respectivement dans les figures 6.15, 6.16 et 6.17. Les résultats obtenus, c'est à dire les valeurs des contraintes homogénéisées, ainsi que les pourcentages d'erreur sont donnés dans le tableau 6.3.

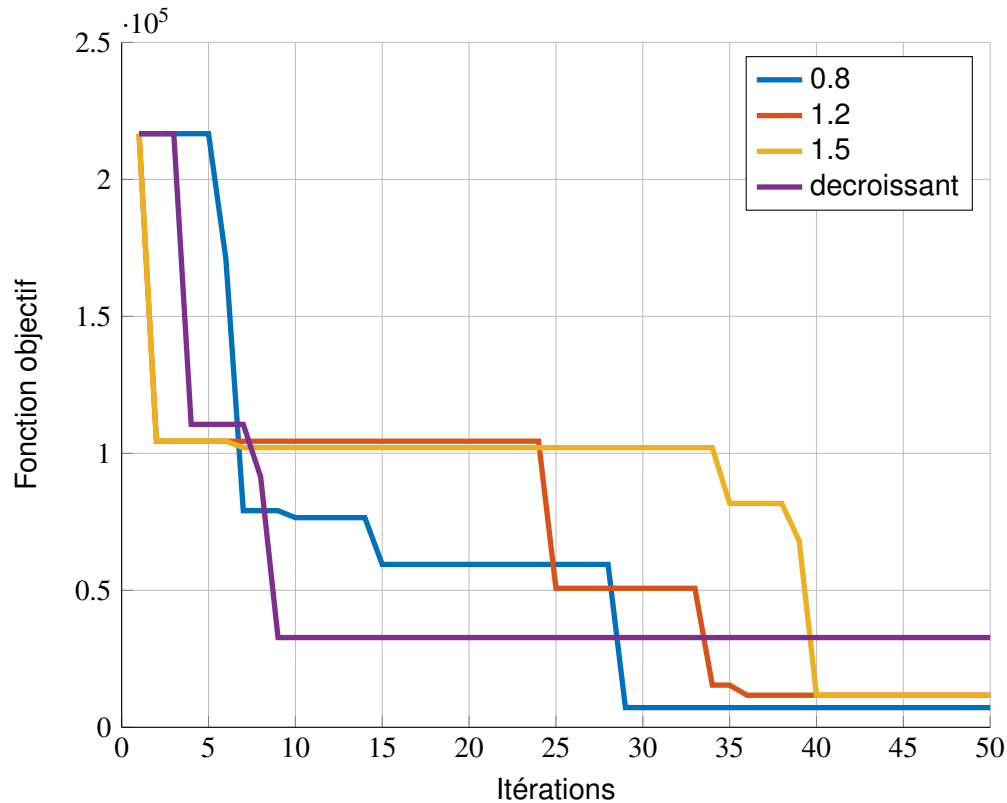


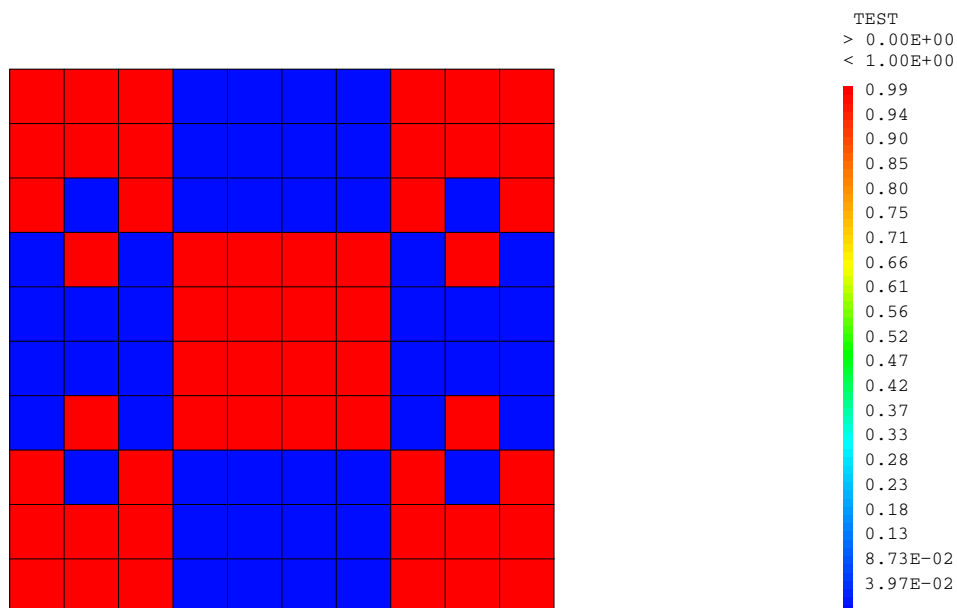
FIGURE 6.11 – Cas 1 : Courbes de convergence des 4 calculs d'optimisation topologique

ω	0.8	1.2	1.5	décroissant
Contrainte homogénéisée (Pa)	$-3.9027E6$	$-3.9073E6$	$-3.9073E6$	$-3.8628E6$
Pourcentage d'erreur (%)	0.1841	0.2997	0.2997	0.8480
Fonction objectif	$7.1856E3$	$1.1713E4$	$1.1713E4$	$3.2759E4$

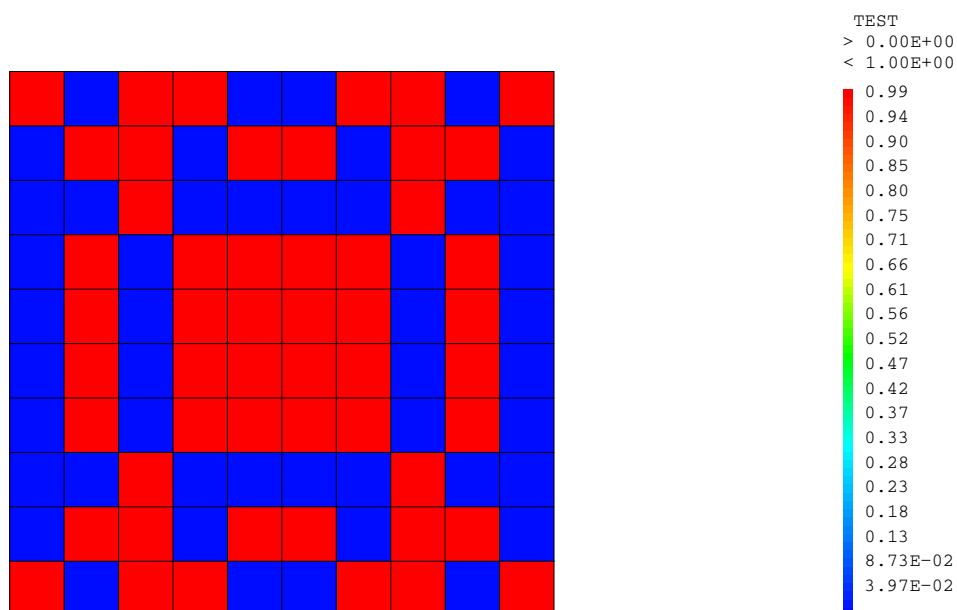
Tableau 6.3 – Cas 1 : Valeurs numériques obtenues

Comme on le voit dans ces figures, les calculs pour $\omega = 1.2$ et $\omega = 1.5$ convergent vers des solutions identiques. Pour les résultats obtenus avec des valeurs fixes de ω , on obtient des topologies à géométries tournantes, dont les contraintes dans les éléments sont assez faibles, alors que la structure présente des grands déplacements. On remarquera ici l'influence des conditions aux limites périodiques, qui induisent bien des déplacements identiques des côtés de la cellule de périodicité. On notera aussi que l'encastrement du nœud inférieur gauche impose, de fait, des déplacements nuls aux quatre coins de la géométrie. Assemblées les unes aux autres, ces cellules de périodicité créeraient un metamatériau qui présente des grandes déformations en compression à l'échelle macroscopique, lorsqu'on leur impose une déformation de $-3.0567E - 3$. On montre ici qu'il

est possible de créer un metamatériau constitué uniquement d'acier qui présente des grandes déformations à l'échelle macroscopique induites par des grands déplacements à l'échelle microscopique, tout en ne subissant pas de contraintes élevées, lorsqu'on lui impose une grande déformation homogène. Et en particulier lorsque l'on considère des géométries tournantes telles que nous les avons vues dans la partie 2.4 à la page 86.



GIBI FECIT

FIGURE 6.12 – Cas 1 : topologie obtenue pour $\omega = 0.8$ 

GIBI FECIT

FIGURE 6.13 – Cas 1 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$

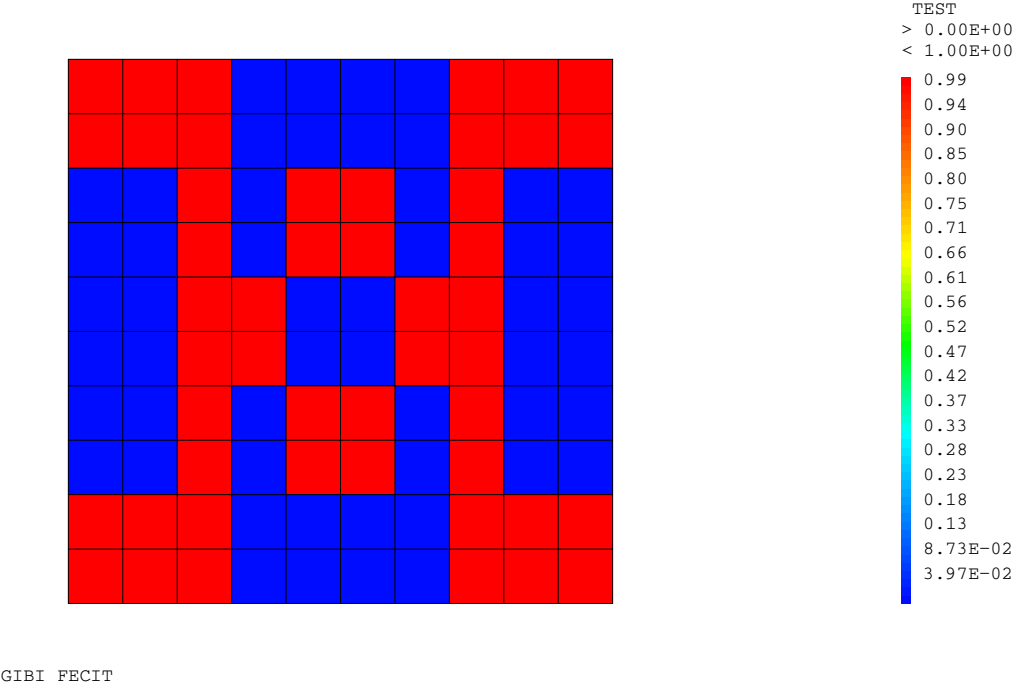


FIGURE 6.14 – Cas 1 : topologie obtenue pour ω décroissant

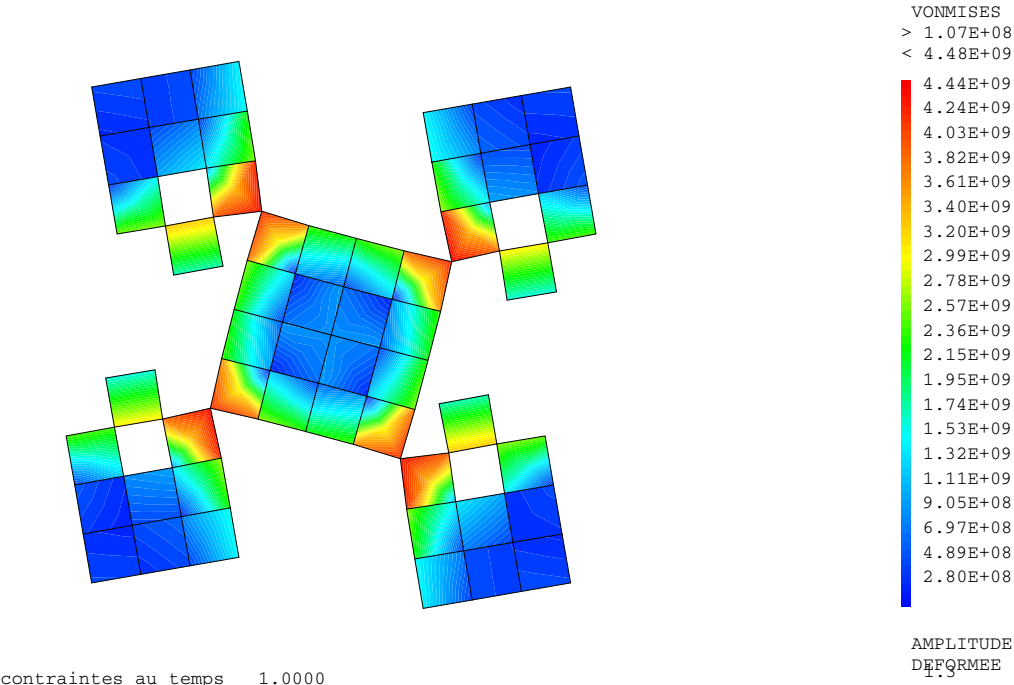


FIGURE 6.15 – Cas 1 : topologie obtenue pour $\omega = 0.8$ en configuration déformée

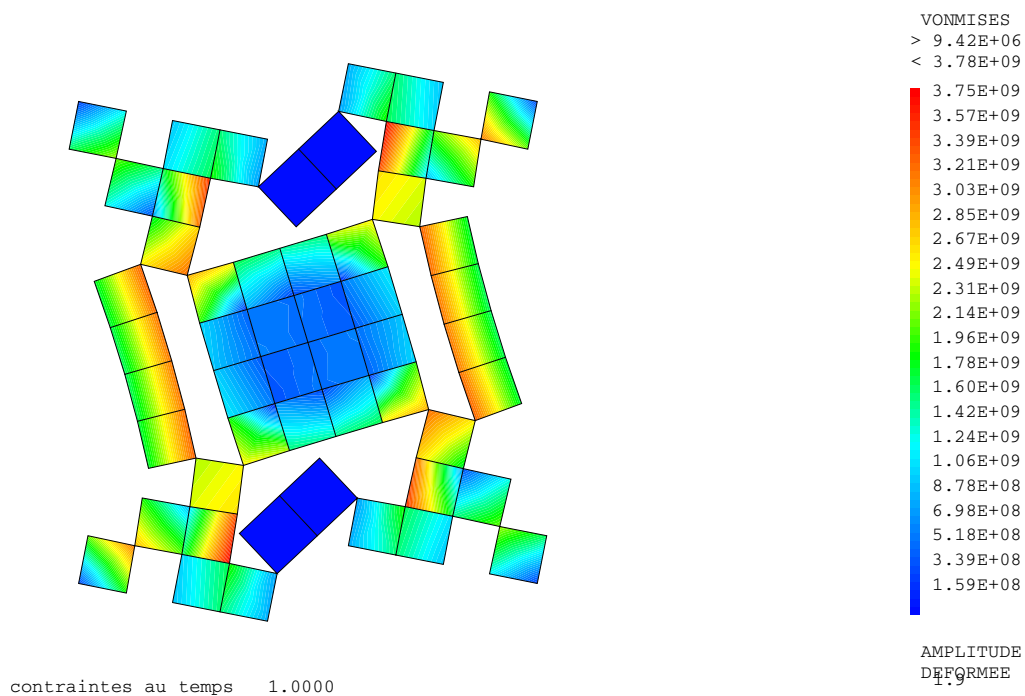


FIGURE 6.16 – Cas 1 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$ en configuration déformée

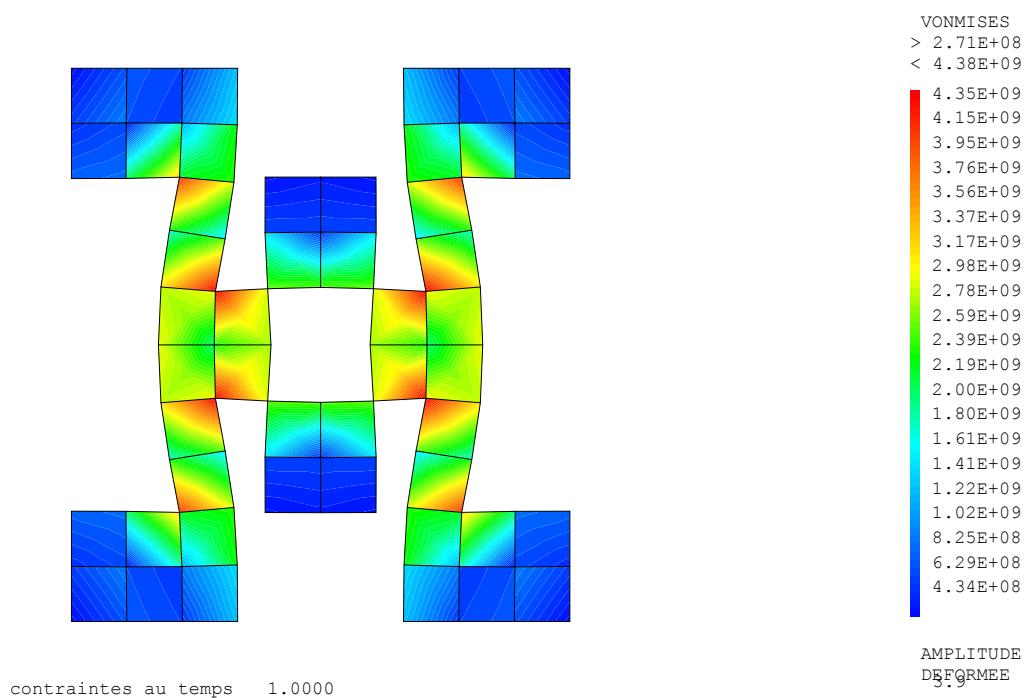


FIGURE 6.17 – Cas 1 : topologie obtenue pour ω décroissant en configuration déformée

En revanche, le dernier résultat, obtenu pour ω décroissant, ne présente pas de géométrie tournante, même si ces résultats sont acceptables (le pourcentage d'erreur reste inférieur à 1%). Dans ce cas, on remarque que l'assemblage linéaire d'éléments Q4 forment des zones dont les déformations ressemblent à celles de poutres épaisses. La déforma-

tion de ces zones impliquent de grands déplacements des extrémités du VER à l'échelle microscopique, qui eux-mêmes impliquent des grandes déformations à l'échelle macroscopique, une fois les cellules de périodicité assemblées. Une étude pourrait être réalisée en utilisant des éléments de poutres travaillant en flexion afin de comparer les résultats à ceux-ci.

On remarque sur les courbes de convergence que les différentes valeurs de ω ont trouvé, à différentes étapes du calcul, des valeurs identiques. En effet, en ayant appliqué deux symétries dans la topologie, nous avons réduit le domaine de recherche à un hypercube en dimension 25, ce qui est assez faible comparé aux autres calculs d'optimisation topologique réalisés dans les chapitres 4 et 5, où les particules présentaient plusieurs centaines de composantes. Dans ce cas, l'algorithme présente donc une probabilité assez grande de trouver deux fois la même solution.

Par curiosité, analysons maintenant l'entièreté de la courbe $\sigma-\epsilon$ en compression des trois topologies différentes que nous avons obtenues. Nous appliquons donc successivement aux différentes topologies les 4 cas de calculs précédemment présentés dans le tableau 6.2, et récupérons la contrainte homogène dans chaque VER. On obtient les courbes de déformation de la figure 6.18 suivante.

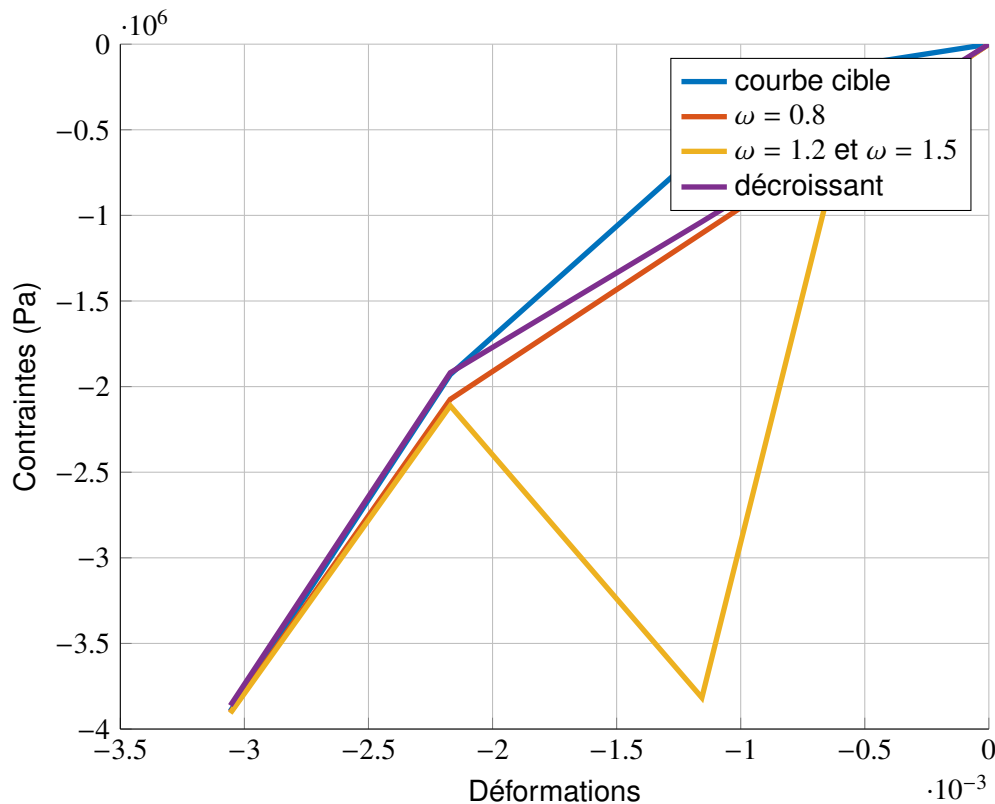


FIGURE 6.18 – Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 1

Comme on le voit sur ces courbes, les topologies obtenues avec $\omega = 0.8$ et ω décroissant présentent un comportement non-linéaire assez proche de celui du caoutchouc, dans une direction de l'espace. Et ce, même si les topologies obtenues sont très différentes, l'une présentant des géométries tournantes, et l'autre des zones dont les déformations

ressemblent à celles de poutres épaisses en flexion, bien qu'elles soient constituées uniquement d'éléments Q4. En revanche, la topologie obtenue avec $\omega = 1.2$ et $\omega = 1.5$ présente une instabilité géométrique somme toute intéressante, même si elle ne présente pas le comportement mécanique ciblé. Cette instabilité géométrique est due au flambage soudain de la structure, lorsque la déformation imposée dépasse un certain seuil. La rotation soudaine des éléments implique des grandes déformations soudaines dans le VER, et une diminution soudaine des contraintes dans les éléments.

6.6.2/ ETUDE DU SECOND CAS DE CALCUL

Dans ce cas, on impose une déformation de $-2.1703E-3$, et l'on souhaite récupérer une contrainte homogénéisée valant $-1.9297E6Pa$.

Les courbes de convergence des calculs réalisés avec $\omega = 0.8$, $\omega = 1.2$, $\omega = 1.5$ et ω décroissant sont représentées dans la figure 6.19. Les topologies obtenues sont représentées en configurations initiales non déformées pour les 4 valeurs de ω différentes respectivement dans les figures 6.20, 6.21, 6.22 et 6.23 où les éléments rouges représentent les éléments constitués d'acier alors que les éléments bleus sont vides, et en configurations déformées pour les quatre valeurs de ω différentes respectivement dans les figures 6.24, 6.25, 6.26 et 6.27. Les résultats obtenus, c'est à dire les valeurs des contraintes homogénéisées et les pourcentages d'erreur, sont donnés dans le tableau 6.4.

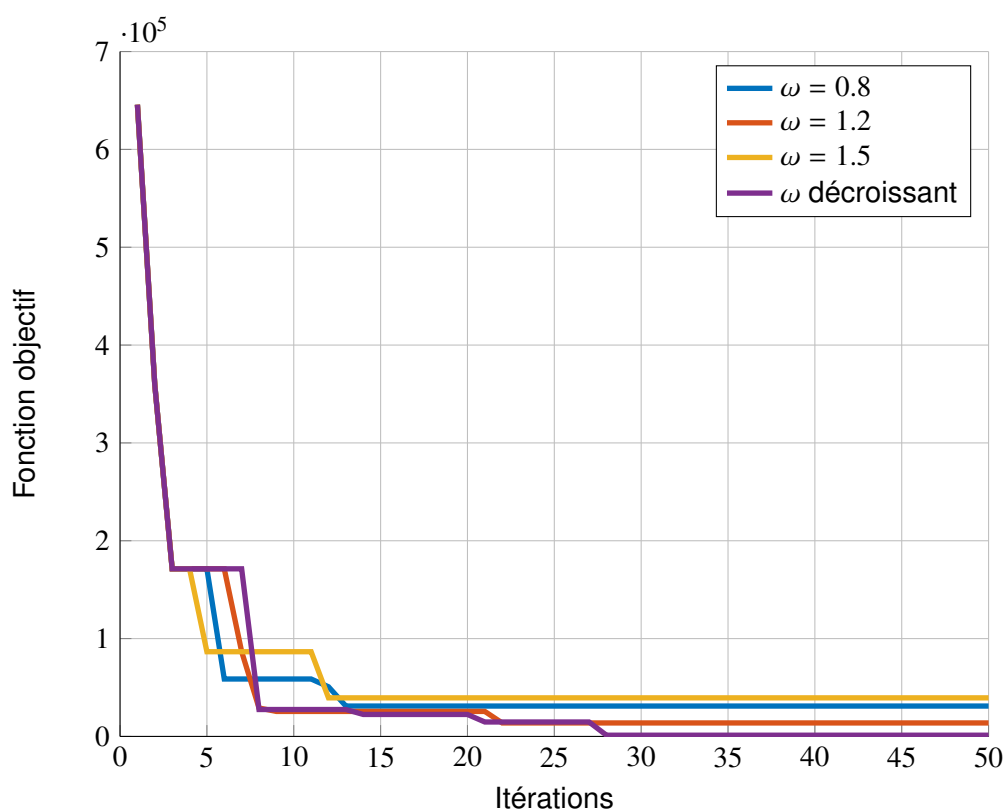


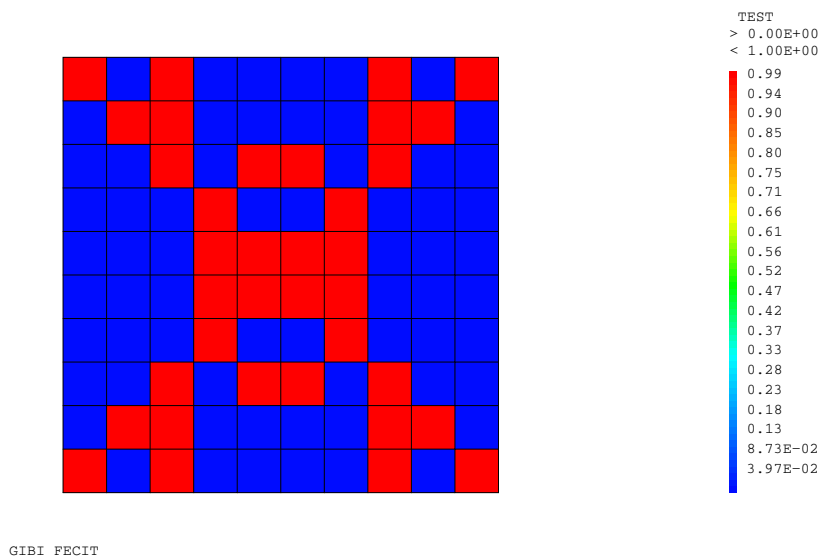
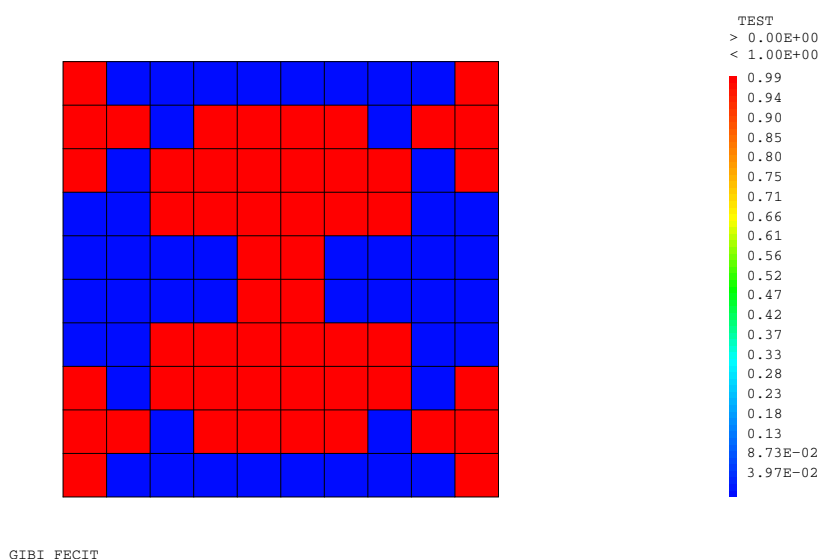
FIGURE 6.19 – Cas 2 : Courbes de convergence des 4 calculs d'optimisation topologique

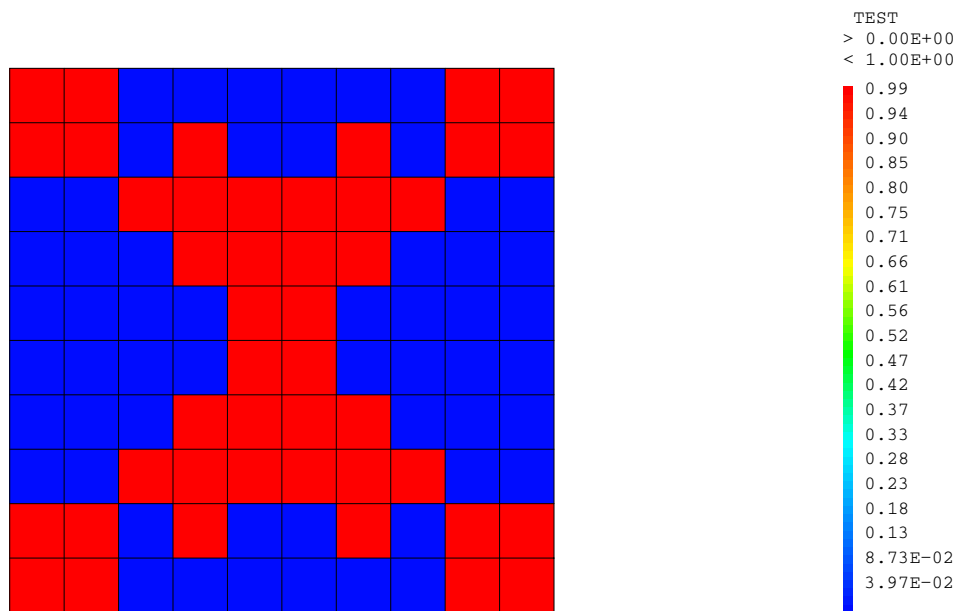
Comme on le voit dans ces figures, tous les résultats obtenus présentent des topologies

ω	0.8	1.2	1.5	décroissant
Contrainte homogénéisée (Pa)	$-1.9608E6$	$-1.9435E6$	$-1.8903E6$	$-1.9285E6$
Pourcentage d'erreur (%)	1.5833	0.7088	2.0856	0.0646
Fonction objectif	$3.1045E4$	$1.3777E4$	$3.9425E4$	$1.2468E3$

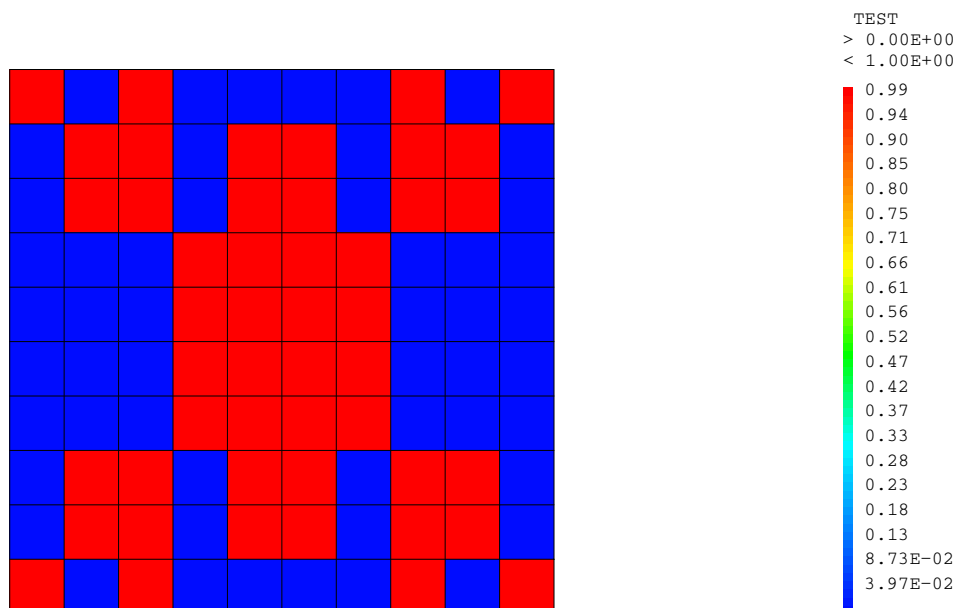
Tableau 6.4 – Cas 2 : Valeurs numériques obtenues

à géométries tournantes, qui tournent par 4 points de rotation en particulier. Bien que les pourcentages d'erreur soient supérieurs à ceux obtenus lors de l'étude du premier cas de calcul, ces résultats nous semblent admissibles et cohérents : les pourcentages d'erreur sont inférieurs à 2.5% dans tous les cas.

FIGURE 6.20 – Cas 2 : topologie obtenue pour $\omega = 0.8$ FIGURE 6.21 – Cas 2 : topologie obtenue pour $\omega = 1.2$

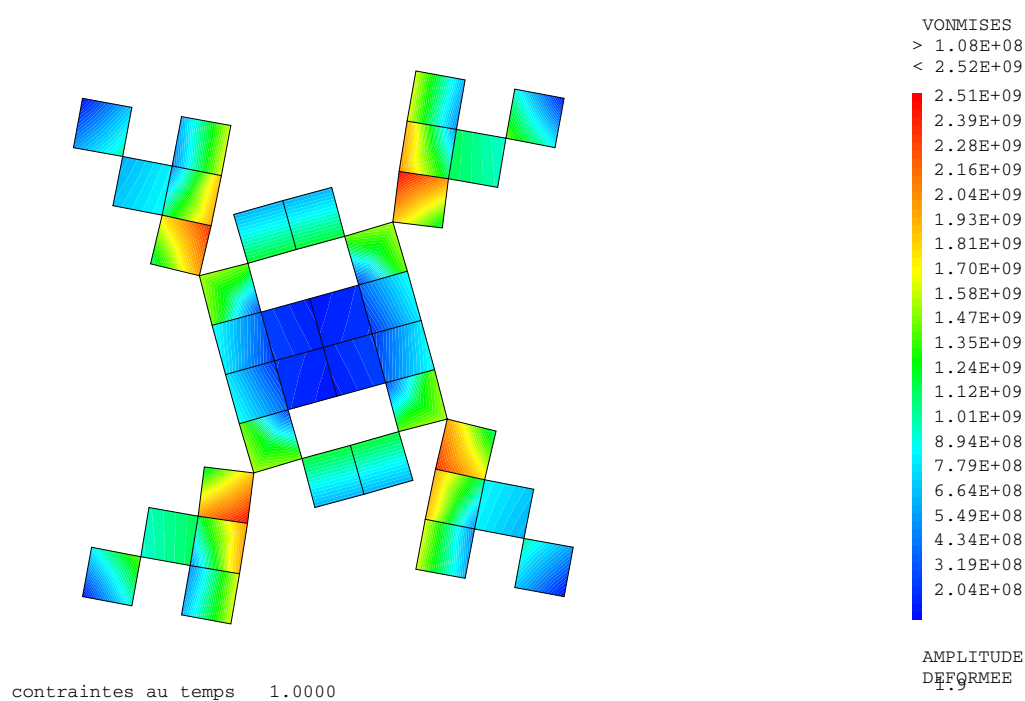
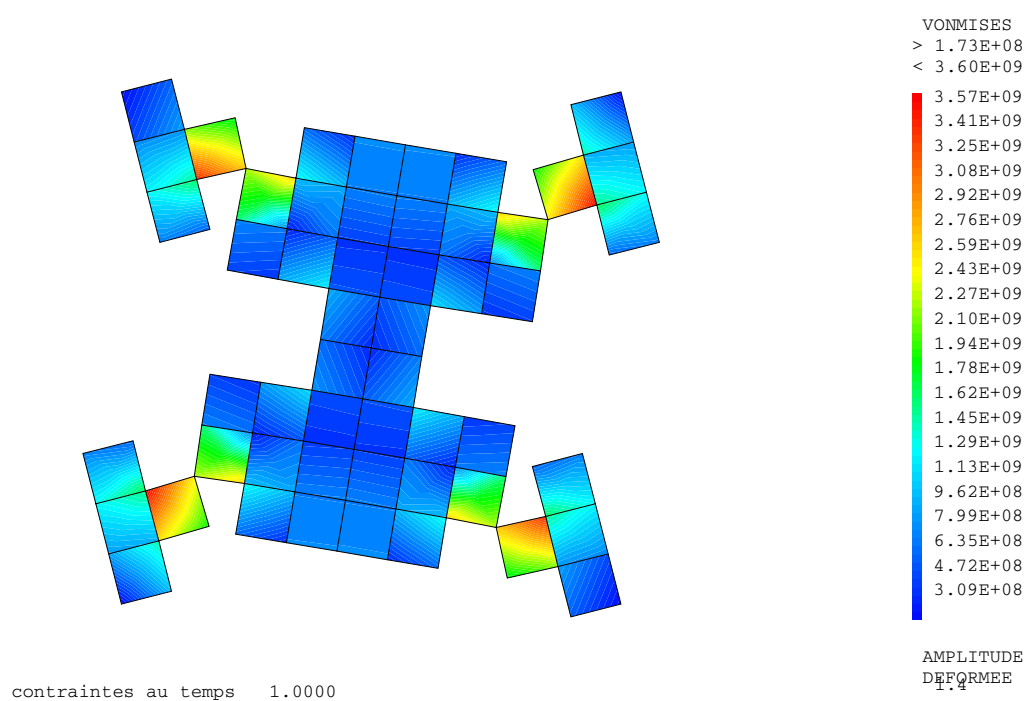


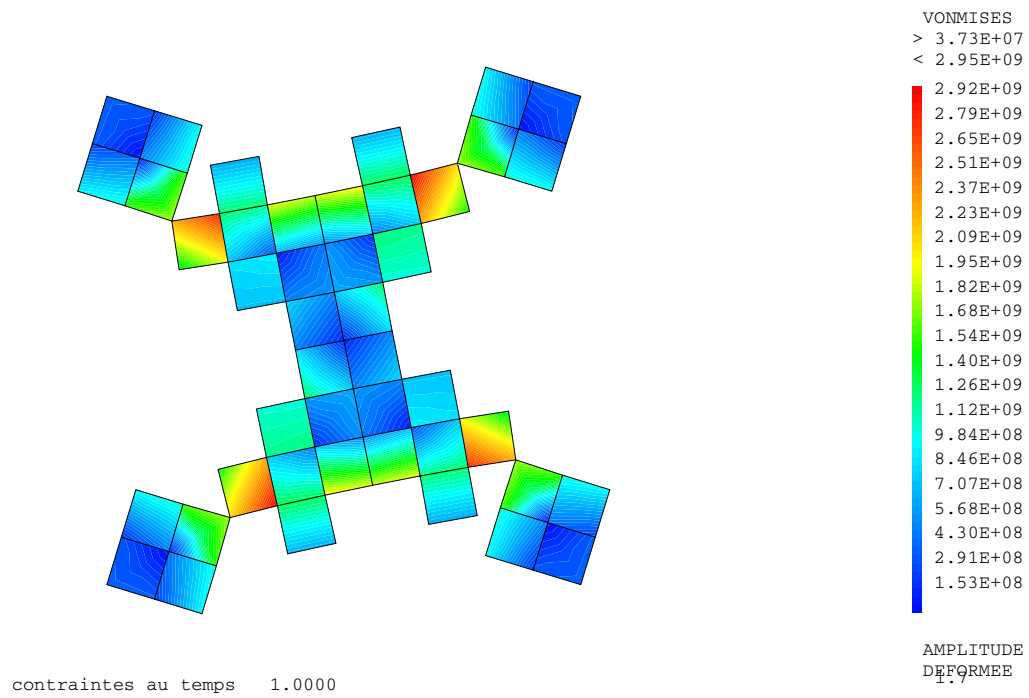
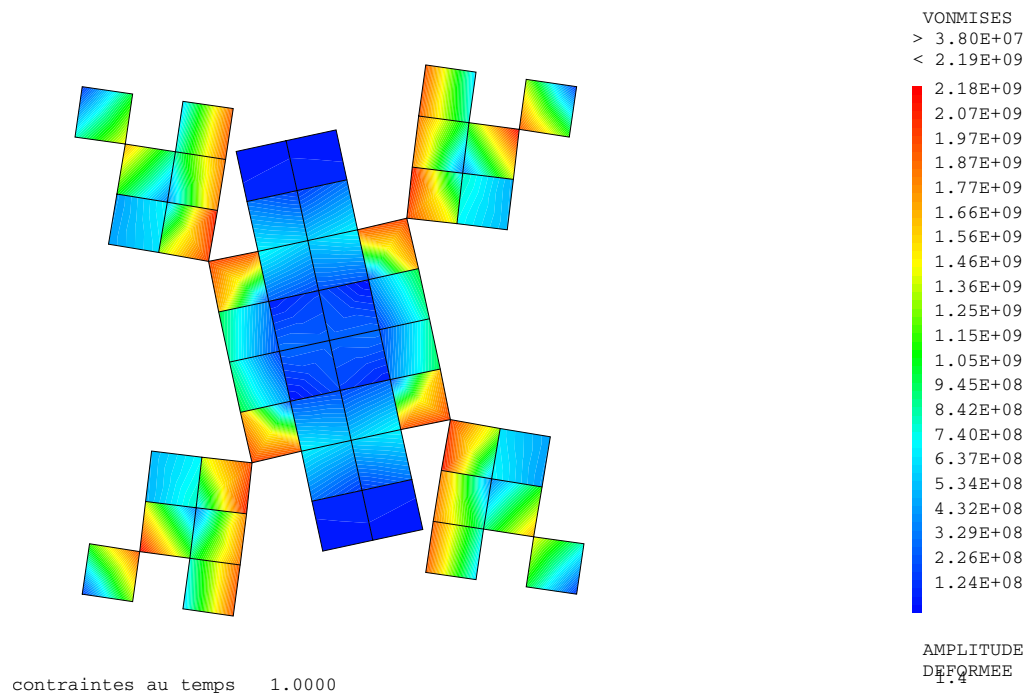
GIBI FECIT

FIGURE 6.22 – Cas 2 : topologie obtenue pour $\omega = 1.5$ 

GIBI FECIT

FIGURE 6.23 – Cas 2 : topologie obtenue pour ω décroissant

FIGURE 6.24 – Cas 2 : topologie obtenue pour $\omega = 0.8$ en configuration déforméeFIGURE 6.25 – Cas 2 : topologie obtenue pour $\omega = 1.2$ en configuration déformée

FIGURE 6.26 – Cas 2 : topologie obtenue pour $\omega = 1.5$ en configuration déforméeFIGURE 6.27 – Cas 2 : topologie obtenue pour ω décroissant en configuration déformée

Par curiosité, analysons maintenant l'entièreté de la courbe $\sigma - \epsilon$ de ces géométries. Pour ce faire, on applique successivement les 4 cas de calcul à ces 4 géométries, et on récupère à chaque fois la valeur de la contrainte homogénéisée. Les courbes $\sigma - \epsilon$ des 4 topologies trouvées sont représentées dans la figure 6.28.

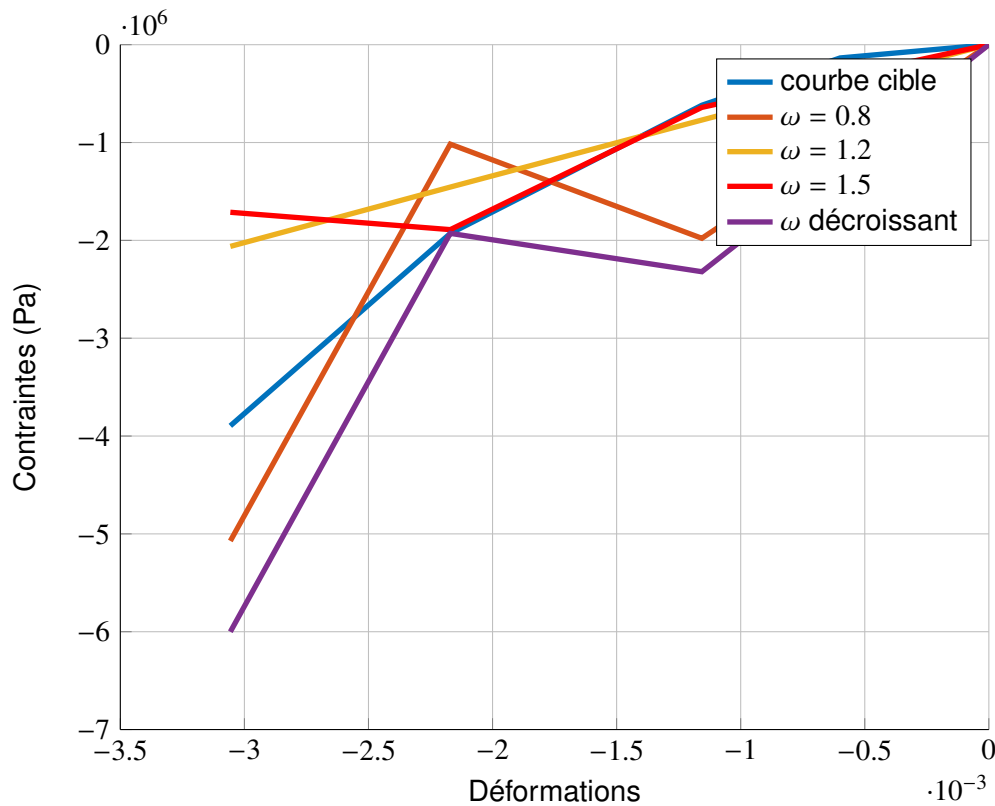


FIGURE 6.28 – Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 2

Comme on le voit sur ces courbes, la seule géométrie à ne pas présenter d'instabilité est celle obtenue avec $\omega = 1.2$. On remarque aussi sur les figures 6.24-6.27 sur c'est la seule géométrie à ne pas tourner dans le même sens que les autres. Sa topologie lui permet des déplacements moins brusques, sans que la structure ne flambe. Elle présente donc un comportement non-linéaire, bien qu'il soit assez éloigné de celui du caoutchouc que nous souhaitons retrouver.

6.6.3/ ÉTUDE DU TROISIÈME CAS DE CALCUL

Dans ce cas, on impose une déformation de $-1.1564E - 3$ et on souhaite récupérer une contrainte homogénéisée de $-6.1891E + 5Pa$.

Les courbes de convergence des calculs réalisés avec $\omega = 0.8$, $\omega = 1.2$, $\omega = 1.5$ et ω décroissant sont représentées dans la figure 6.29. Les topologies obtenues sont représentées en configurations initiales non déformées pour les 4 valeurs de ω différentes testées respectivement dans les figures 6.30-6.32 où les éléments rouges sont constitués d'acier et les éléments bleus sont vides, et en configurations déformées pour les 4 valeurs de ω différentes respectivement dans les figures 6.33-6.35. Les résultats obtenus, c'est à dire les valeurs des contraintes homogénéisées sont données dans le tableau 6.5.

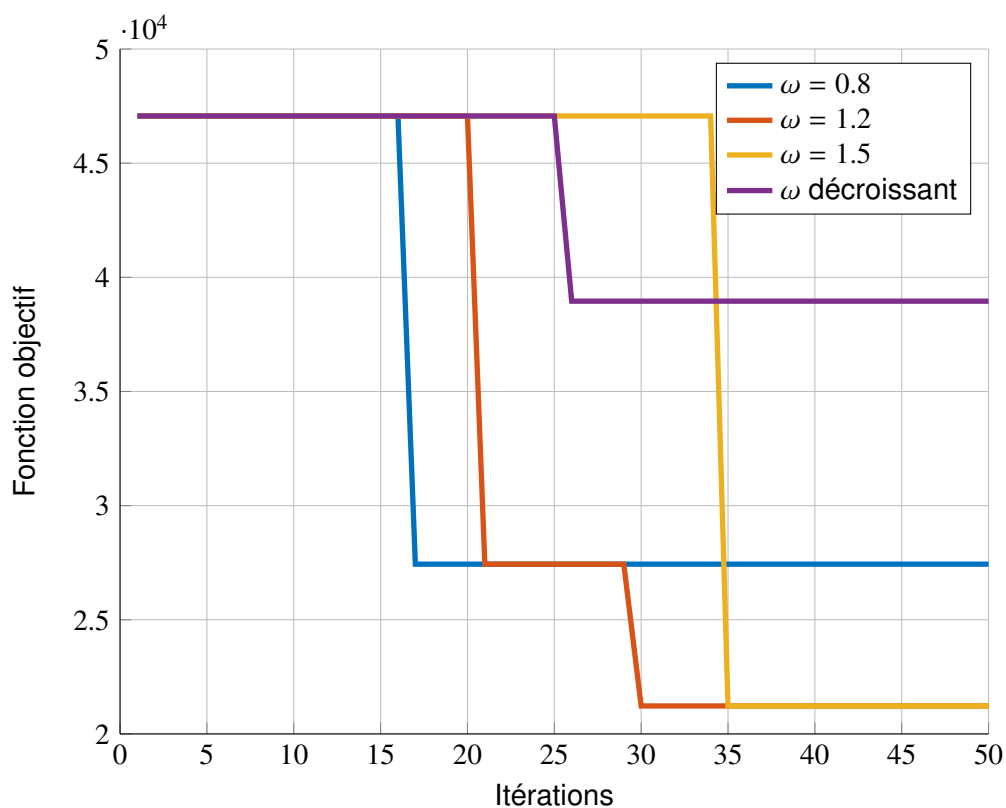


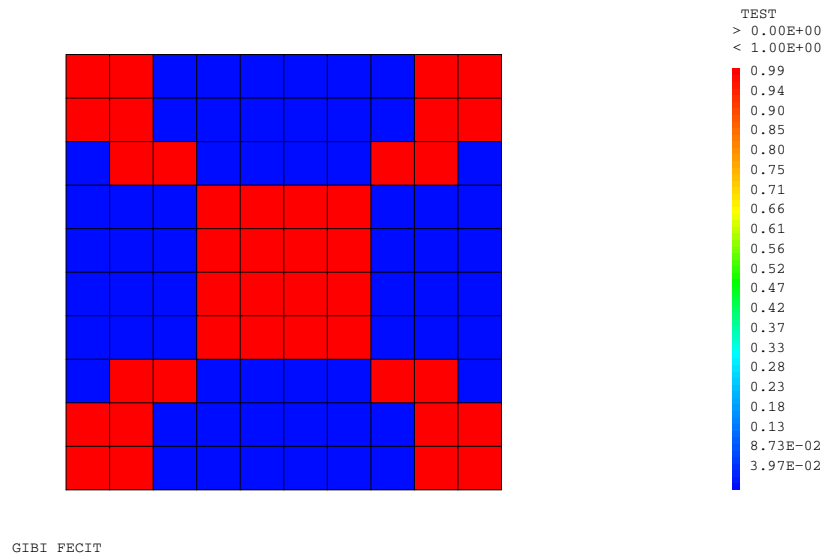
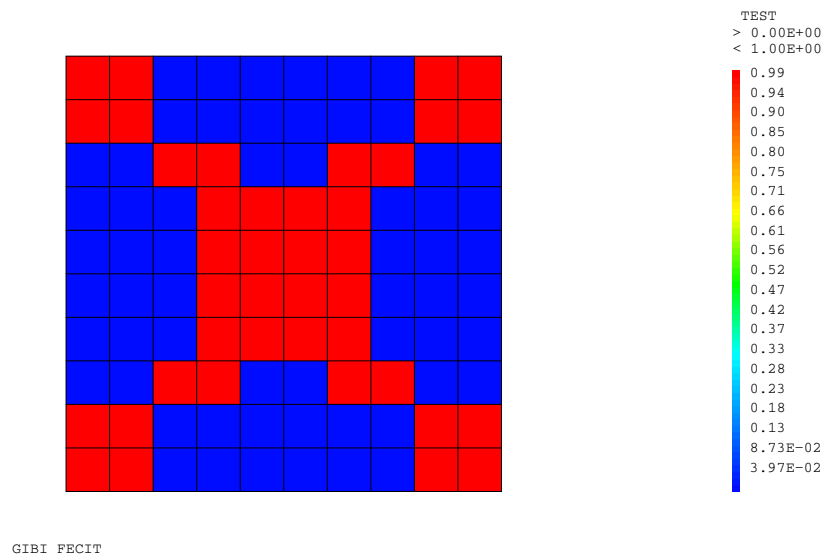
FIGURE 6.29 – Cas 3 : Courbes de convergence des 4 calculs d'optimisation topologique

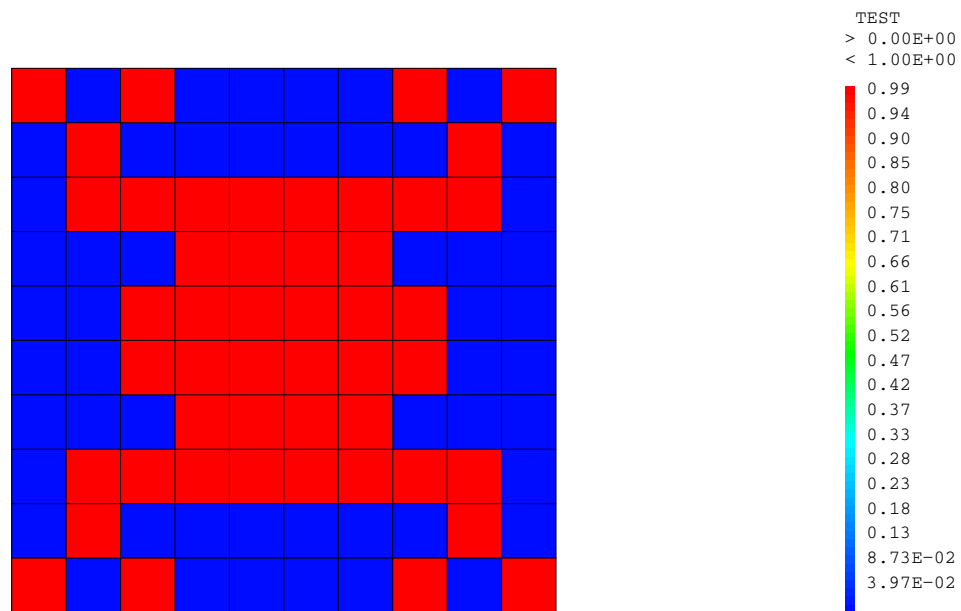
ω	0.8	1.2	1.5	décroissant
Contrainte homogénéisée (Pa)	$-5.9147E5$	$-5.9768E5$	$-5.9768E5$	$-6.5786E5$
Pourcentage d'erreur (%)	4.6384	3.5512	3.5512	5.9215
Fonction objectif	$2.7435E4$	$2.1225E4$	$2.1225E4$	$3.8956E4$

Tableau 6.5 – Cas 3 : Valeurs numériques obtenues

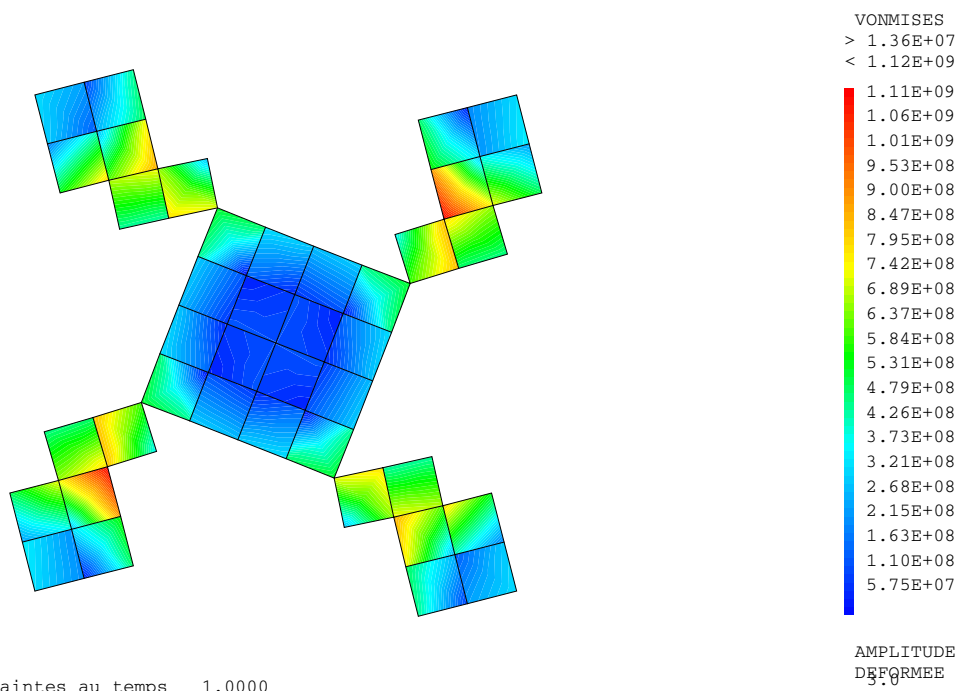
Dans ce cas là, la convergence est moins marquée, notre algorithme d'optimisation I-PR-PSO a convergé très rapidement vers des solutions optimales. Les pourcentages d'erreur sont donnés de l'ordre de 5%. Les résultats trouvés pour $\omega = 1.2$ et $\omega = 1.5$ sont identiques. Les deux premières géométries des figures 6.33 et 6.34 présentent des

géométries tournantes, alors que la dernière géométrie trouvée avec ω décroissant n'en présente pas. On remarque cependant sur ce VER en particulier les zones dont les déformations ressemblent à celles de poutres en flexion que nous avons déjà observées précédemment. Ici encore, on montre qu'il est possible avec I-PR-PSO de trouver des géométries de VER qui subissent de grands déplacements et de grandes rotations à l'échelle microscopique, qui induisent de grandes déformations à l'échelle macroscopique.

FIGURE 6.30 – Cas 3 : topologie obtenue pour $\omega = 0.8$ FIGURE 6.31 – Cas 3 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$



GIBI FECIT

FIGURE 6.32 – Cas 3 : topologie obtenue pour ω décroissantFIGURE 6.33 – Cas 3 : topologie obtenue pour $\omega = 0.8$ en configuration déformée

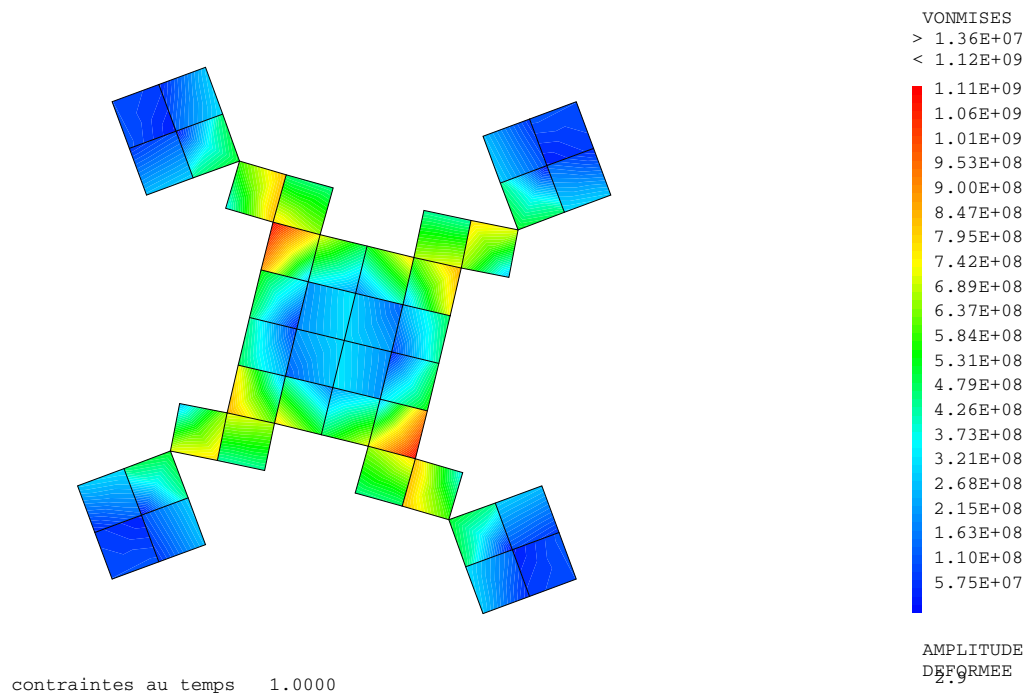


FIGURE 6.34 – Cas 3 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$ en configuration déformée

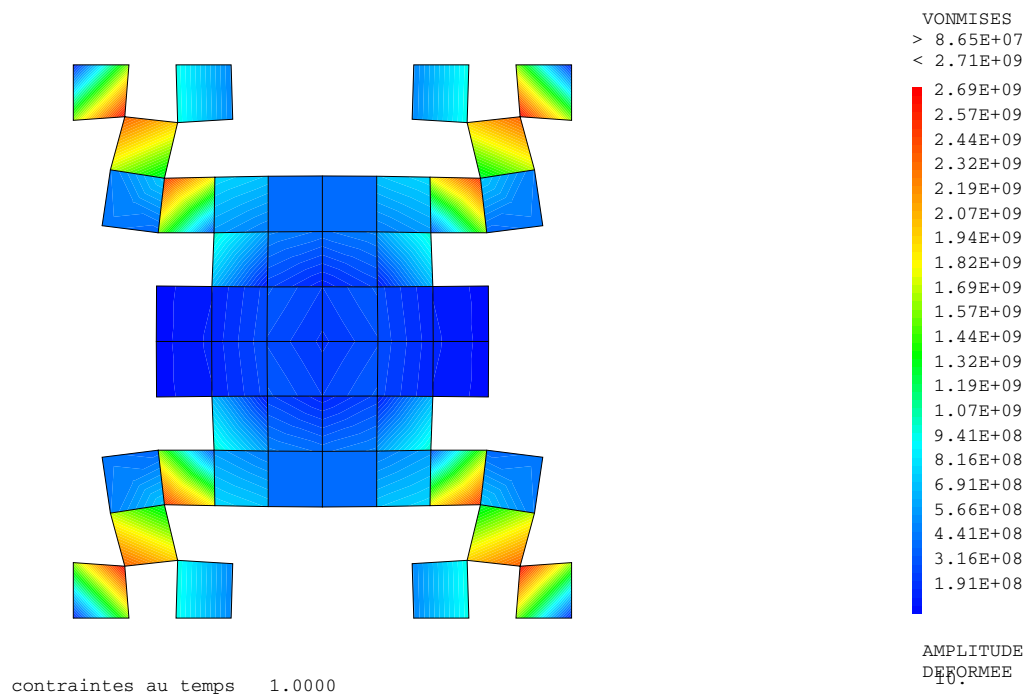


FIGURE 6.35 – Cas 3 : topologie obtenue pour ω décroissant en configuration déformée

Analysons maintenant l'entièreté des courbes $\sigma - \epsilon$ en compression des topologies obtenues, données dans la figure 6.36. Toutes les topologies présentent ici un point d'instabilité, dont la déformation imposée est soit inférieure ($\omega = 0.8$) soit supérieure ($\omega = 1.2$ et 1.5 et ω décroissant) à la déformation imposée initialement pour étudier le troisième cas

de calcul. Bien que les résultats obtenus pour ce troisième cas de calcul soient assez proches de ceux recherchés, les courbes complètes obtenues sont assez éloignées de celle que nous souhaitons obtenir initialement.

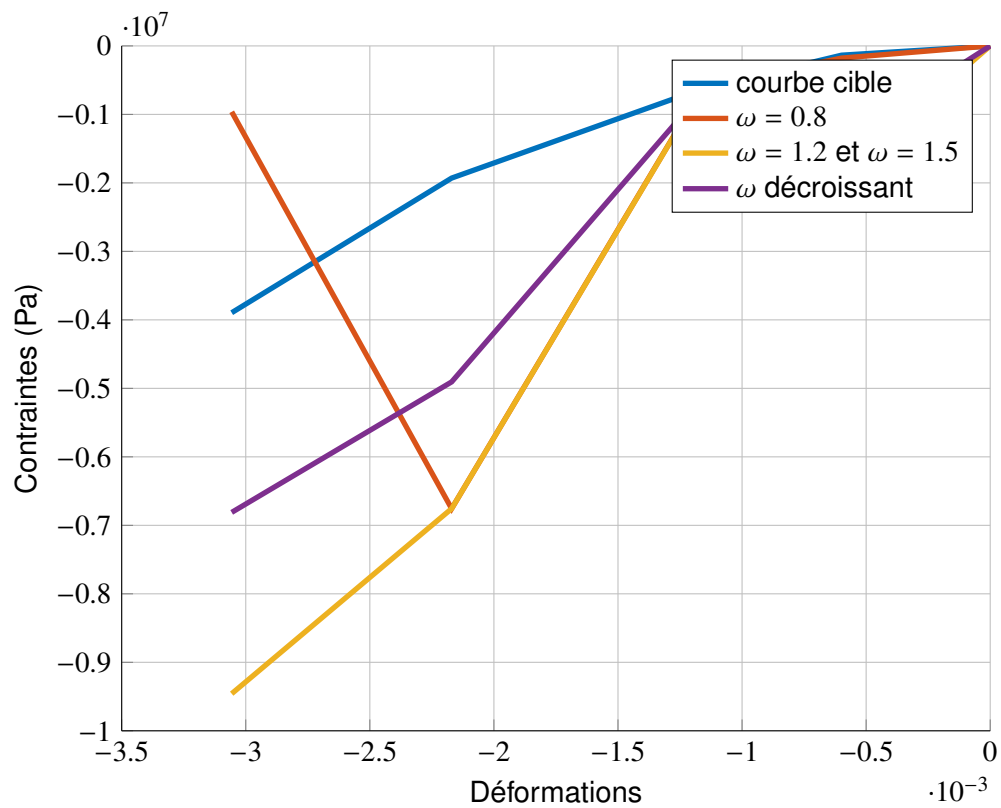


FIGURE 6.36 – Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 3

6.6.4/ ÉTUDE DU QUATRIÈME CAS DE CALCUL

Dans ce cas, on impose une déformation de $-5.9925E - 4$ et l'on souhaite obtenir une contrainte homogénéisée de $-1.3545E + 5Pa$.

Les courbes de convergence des calculs réalisés avec $\omega = 0.8$, $\omega = 1.2$, $\omega = 1.5$ et ω décroissant sont représentées dans la figure 6.37. Les topologies obtenues sont représentées en configurations initiales non déformées pour les 4 valeurs de ω différentes respectivement dans les figures 6.38-6.41 où les éléments rouges sont constitués d'acier et les éléments bleus sont vides, et en configurations déformées pour les 4 valeurs de ω différentes dans les figures 6.42-6.45. Les résultats obtenus, c'est à dire les valeurs des contraintes homogénéisées et les pourcentages d'erreur, sont donnés dans le tableau 6.6.

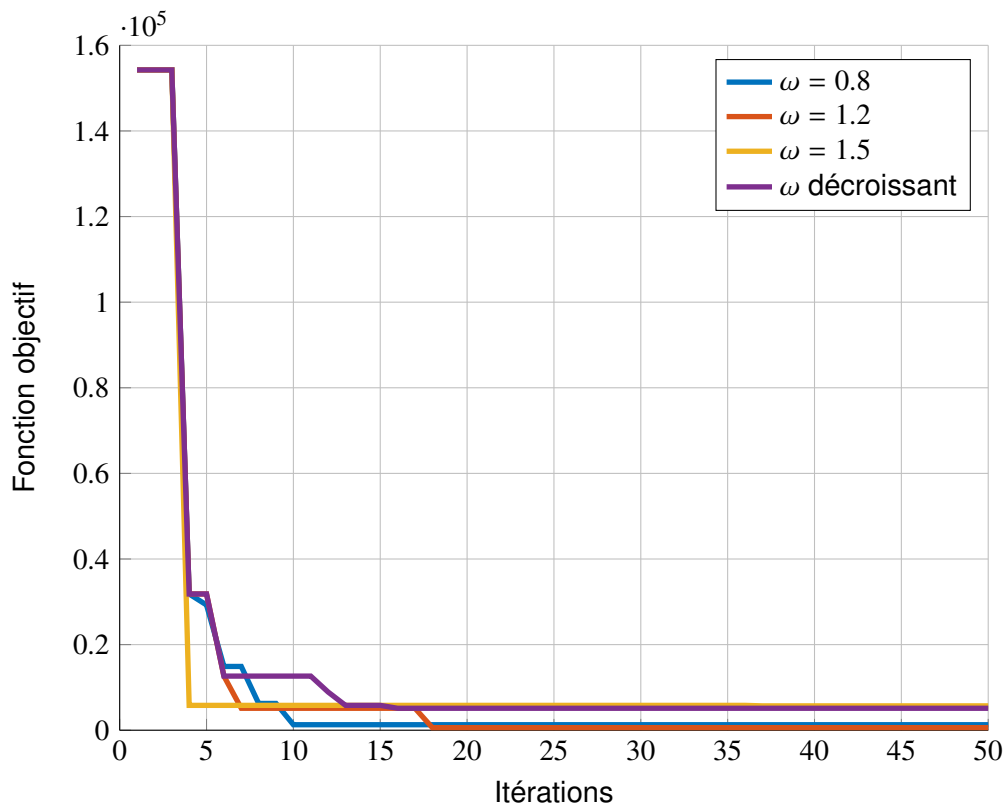
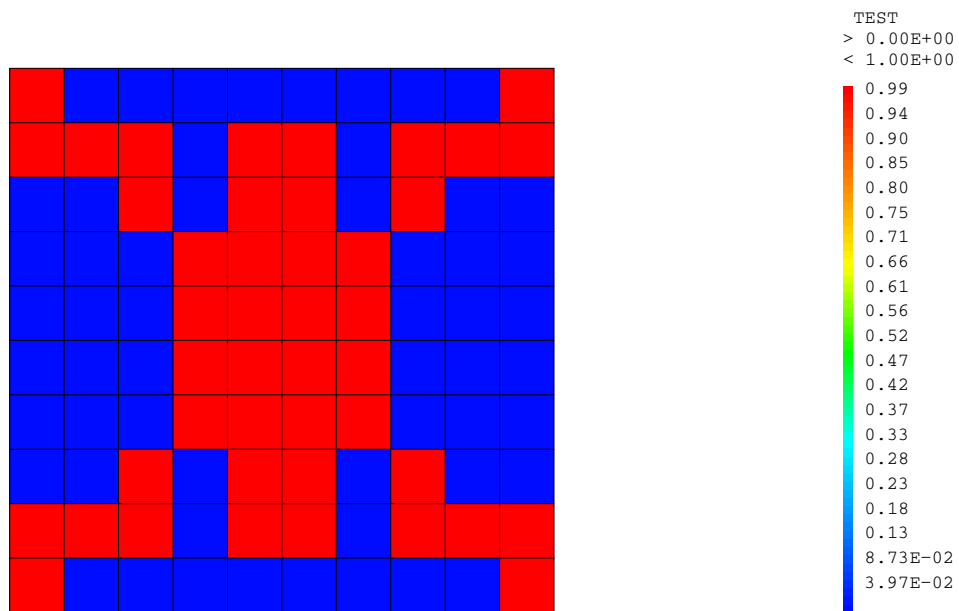


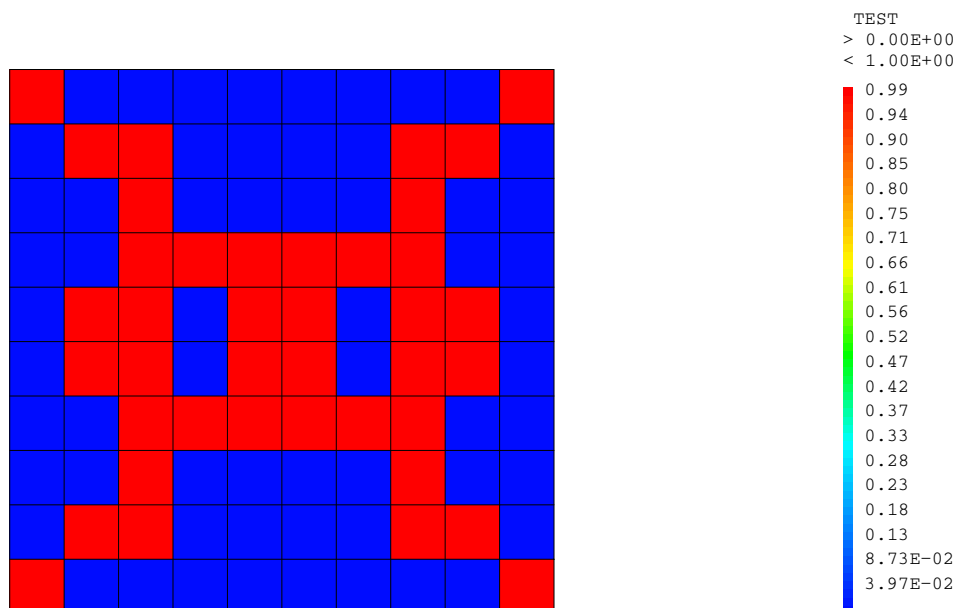
FIGURE 6.37 – Cas 4 : Courbes de convergence des 4 calculs d'optimisation topologique

ω	0.8	1.2	1.5	décroissant
Contrainte homogénéisée (Pa)	$-1.3674E5$	$-1.3605E5$	$-1.2976E5$	$-1.3032E5$
Pourcentage d'erreur (%)	0.9398	0.4439	4.3860	3.9343
Fonction objectif	$1.2851E3$	$6.0401E2$	$5.6913E3$	$5.1274E3$

Tableau 6.6 – Cas 4 : Valeurs numériques obtenues

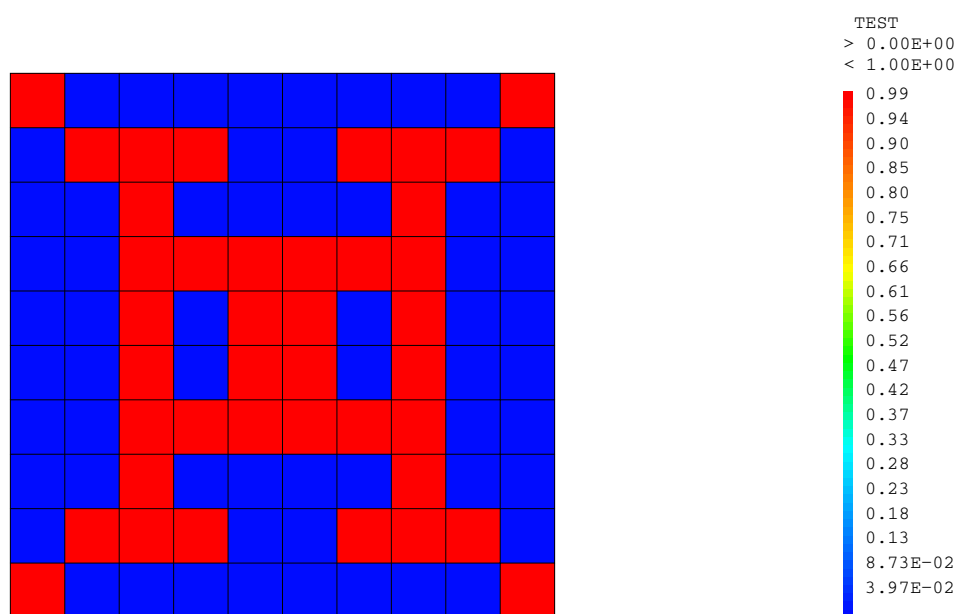


GIBI FECIT

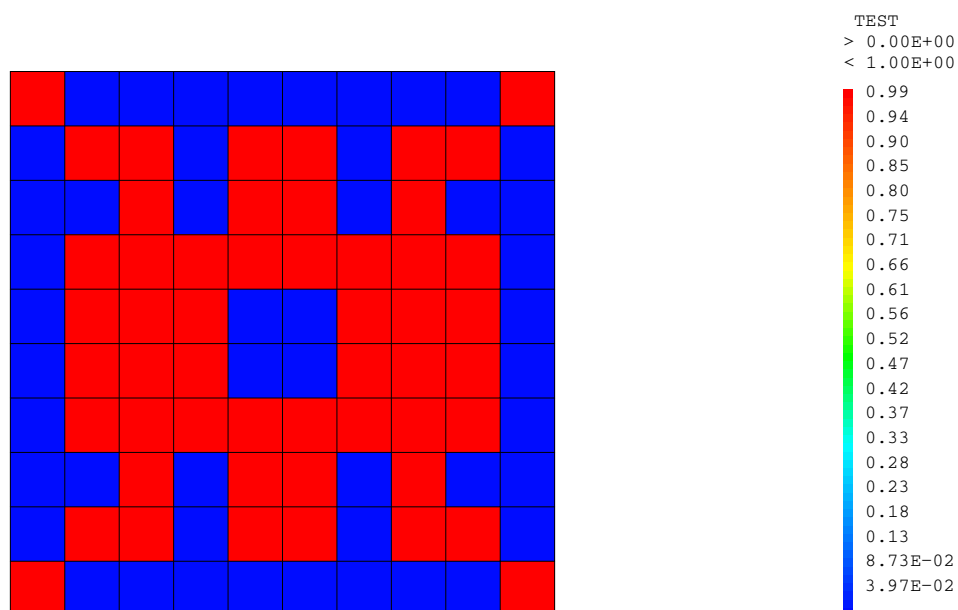
FIGURE 6.38 – Cas 4 : topologie obtenue pour $\omega = 0.8$ 

GIBI FECIT

FIGURE 6.39 – Cas 4 : topologie obtenue pour $\omega = 1.2$

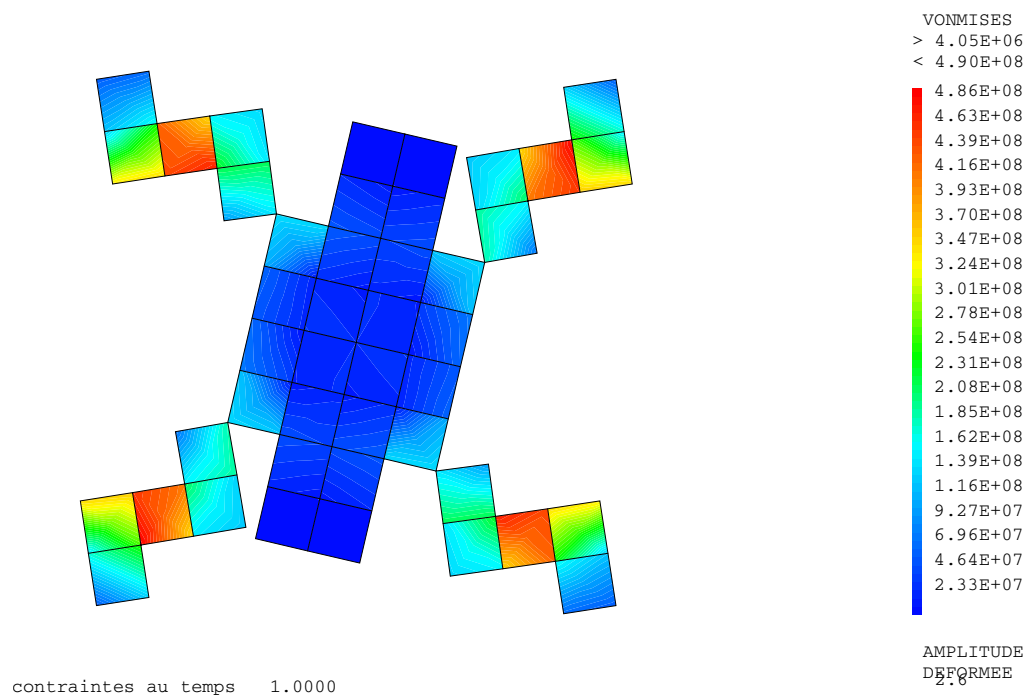
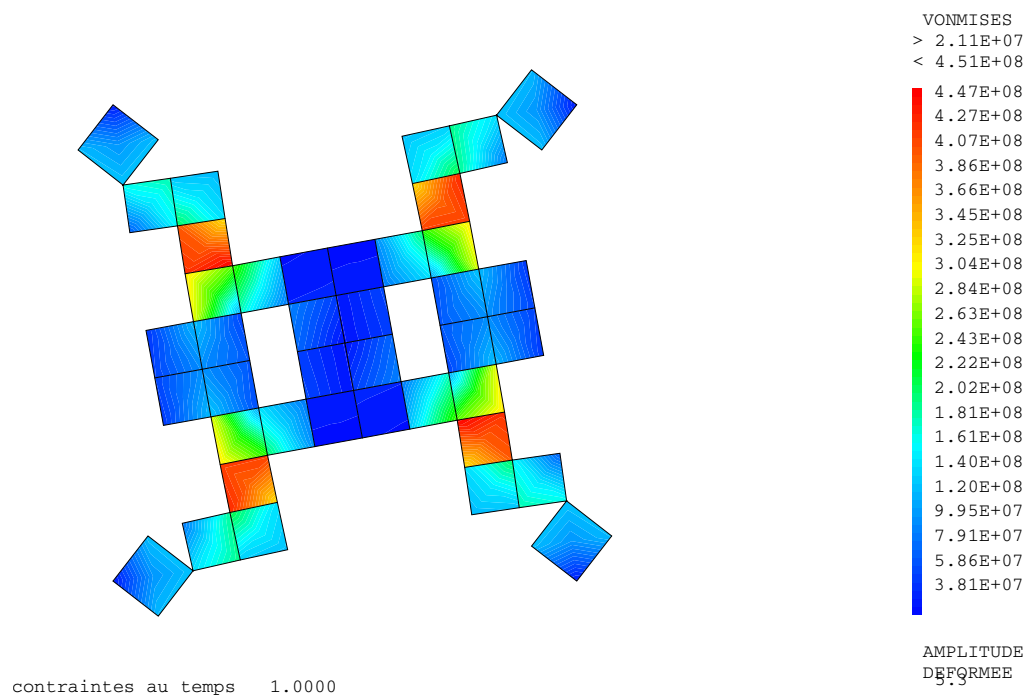


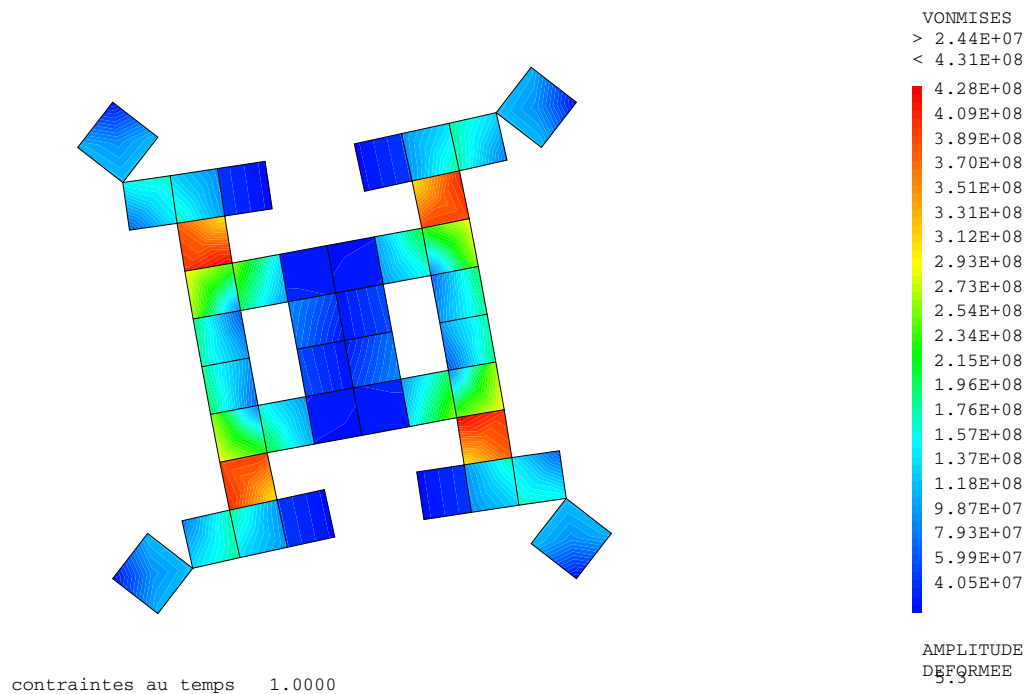
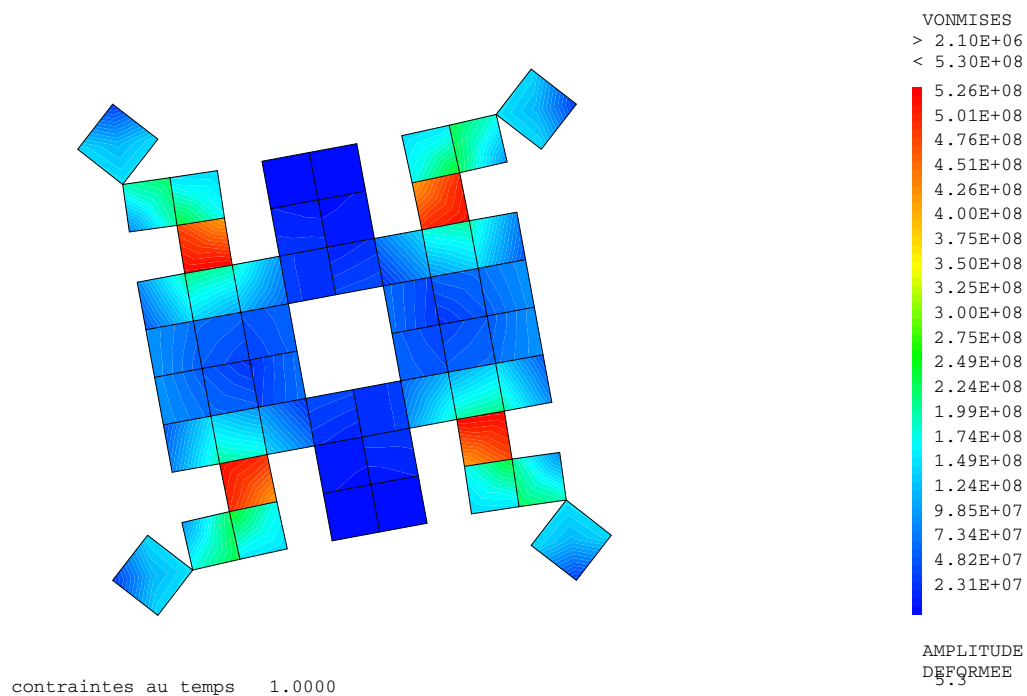
GIBI FECIT

FIGURE 6.40 – Cas 4 : topologie obtenue pour $\omega = 1.5$ 

GIBI FECIT

FIGURE 6.41 – Cas 4 : topologie obtenue pour ω décroissant

FIGURE 6.42 – Cas 4 : topologie obtenue pour $\omega = 0.8$ FIGURE 6.43 – Cas 4 : topologie obtenue pour $\omega = 1.2$

FIGURE 6.44 – Cas 4 : topologie obtenue pour $\omega = 1.5$ FIGURE 6.45 – Cas 4 : topologie obtenue pour ω décroissant

Toutes les géométries présentent des géométries tournantes. On remarque que trois des géométries ($\omega = 1.2$, $\omega = 1.5$, ω décroissant) sont quasiment similaires, dans le sens où elles présentent toutes une rotation sur les éléments des coins du notre carré.

Si l'on analyse l'entièreté des courbes $\sigma - \epsilon$ en compression de ces topologies, on obtient les courbes présentées en figure 6.46 suivante. Comme on le voit sur ces courbes,

seule la topologie obtenue par l'algorithme considérant ω décroissant présente un point d'instabilité. Les autres topologies présentent une non-linéarité sur leurs courbes $\sigma - \epsilon$ respectives. Bien que ne fittant pas la courbe du caoutchouc que nous souhaitons retrouver, ces résultats sont tout de même intéressants. Ils montrent encore une fois qu'un matériau constitué d'acier peut avoir un comportement non-linéaire si sa topologie s'y prête à l'échelle microscopique, mais aussi présenter des grandes déformations à l'échelle macroscopique.

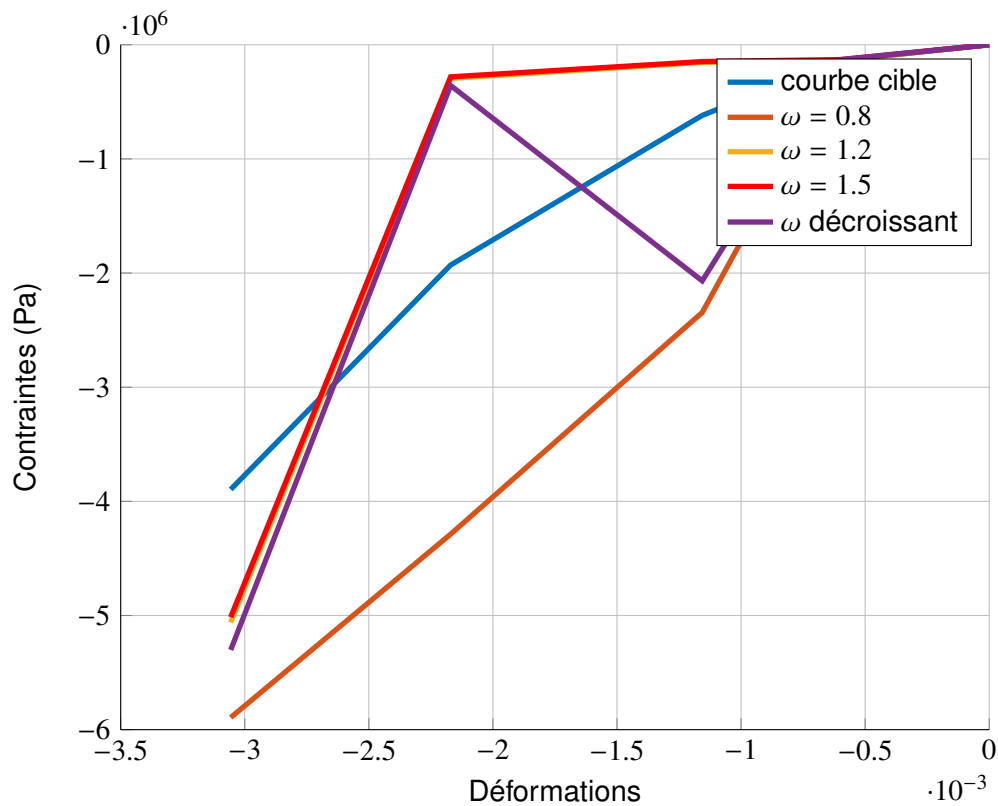


FIGURE 6.46 – Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 4

6.6.5/ RECHERCHE DE LA TOTALITÉ DE LA COURBE $\sigma - \epsilon$ EN COMPRESSION

Dans ce cas, on applique pour chaque topologie - c'est à dire pour chaque particule de notre essaim I-PR-PSO - les 4 déformations des cas de calcul de la courbe, et l'on récupère pour chacun la valeur de la contrainte homogénéisée. Le problème à résoudre est ainsi défini de la manière suivante

$$\min \sum_{p=1}^4 \|\sigma_{calcul}^p - \sigma_{courbe}^p\| \quad (6.20)$$

où l'exposant p indique le cas de calcul, σ_{calcul} est la valeur de la contrainte calculée dans le VER, et σ_{courbe} est la valeur de contrainte de la courbe à retrouver.

Les courbes de convergence des 4 calculs d'optimisation, réalisés avec les 4 valeurs de ω , sont données en figure 6.47 suivante.

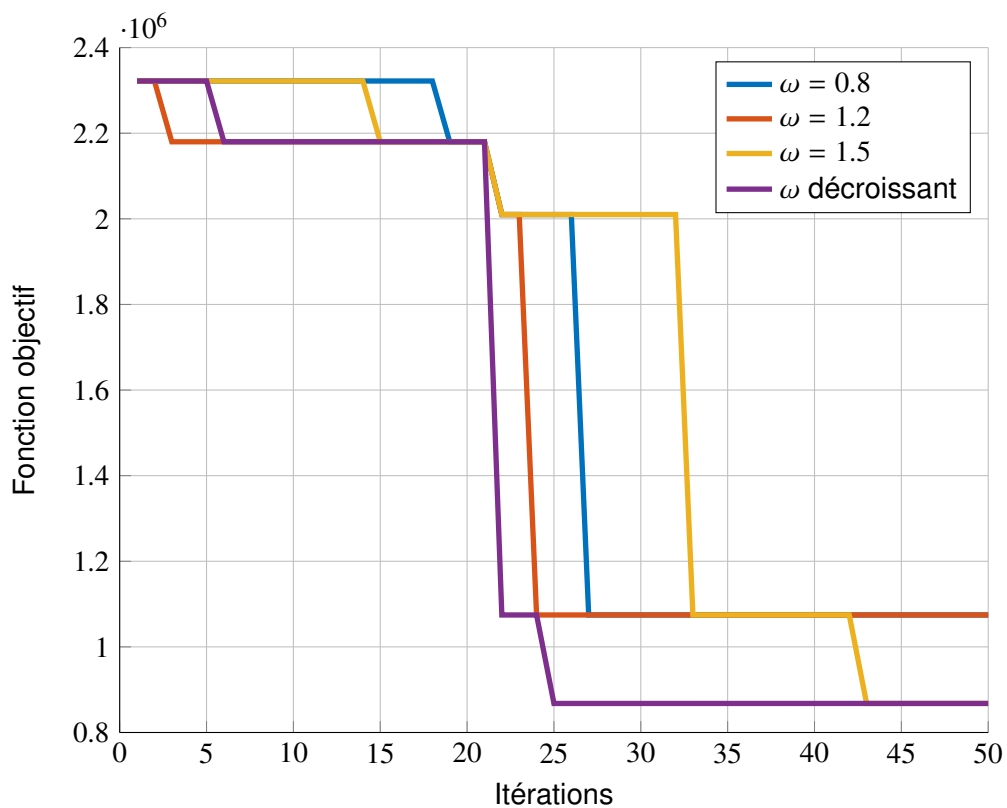
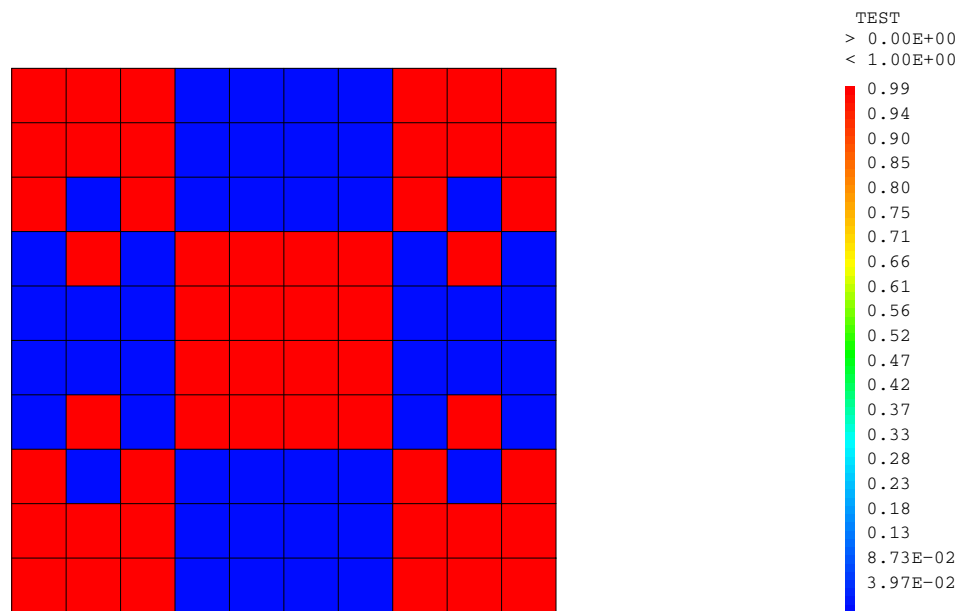


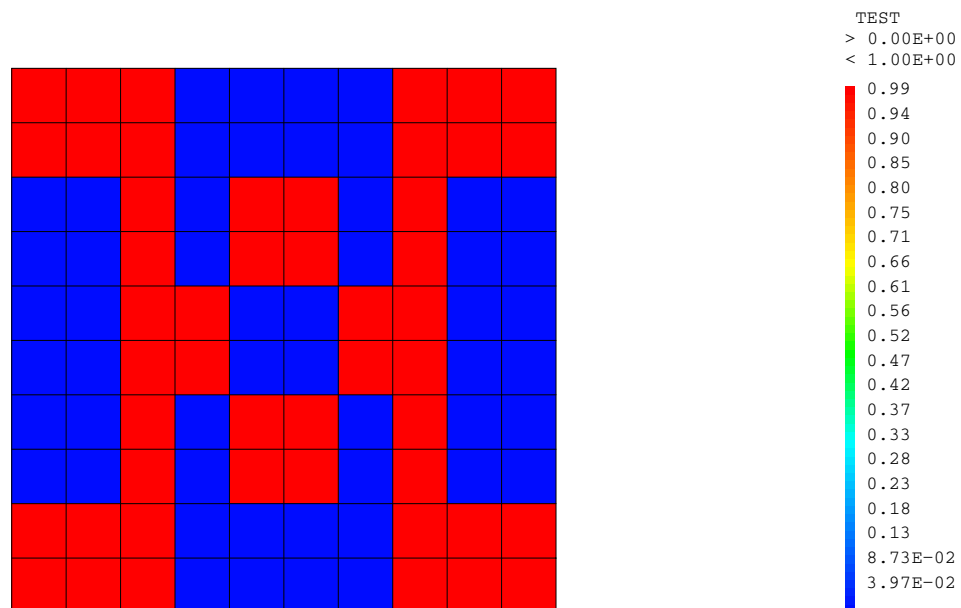
FIGURE 6.47 – Courbe de compression complète : Courbes de convergence des 4 calculs d'optimisation topologique

Comme on le voit sur cette figure, les 4 calculs d'optimisation ont convergé deux à deux vers les mêmes solutions. Les topologies obtenues sont représentées en configurations initiales non déformées dans les figures 6.48-6.49 suivantes.



GIBI FECIT

FIGURE 6.48 – Courbe de compression complète : topologie obtenue pour $\omega = 0.8$ et $\omega = 1.2$



GIBI FECIT

FIGURE 6.49 – Courbe de compression complète : topologie obtenue pour $\omega = 1.5$ et ω décroissant

Les résultats obtenus pour chaque cas de calcul, ainsi que la valeur de la fonction objectif sont récapitulés dans le tableau 6.7.

Contraintes homogénéisées (Pa)					
Cas de calcul					Fonction objectif
ω	1	2	3	4	
0.8	-3.9028E6	-2.0744E6	-1.1048E6	-5.7228E5	1.0746E6
1.2	-3.9028E6	-2.0744E6	-1.1048E6	-5.7228E5	1.0746E6
1.5	-3.8628E6	-1.9190E6	-1.0371E6	-5.4163E5	8.6786E5
décroissant	-3.8628E6	-1.9190E6	-1.0371E6	-5.4163E5	8.6786E5

Tableau 6.7 – Valeurs numériques obtenues lors de l'étude de la courbe de compression complète

Les résultats étant les mêmes que précédemment, les courbes $\sigma - \epsilon$ de ces topologies sont données dans la figure 6.18 que nous avons vue précédemment à la page 193. Ces résultats présentent une non-linéarité sur la courbe $\sigma - \epsilon$ qui se rapproche de celle du caoutchouc dont nous souhaitons mimer le comportement mécanique.

6.7/ CONCLUSIONS DU CHAPITRE

Ce chapitre propose d'appliquer la version binaire de l'algorithme I-PR-PSO que nous avons développé dans les chapitres 3 et 4 à l'optimisation topologique des metamatériaux mécaniques. Dans ce cas, nous avons tenté de développer un metamatériau constitué d'acier dont les propriétés mécaniques sont celles du caoutchouc en compression. Pour ce faire, nous avons discrétisé cette courbe de compression en 4 points pour lesquels nous avons tenté, en appliquant la déformation homogène correspondante, de retrouver la contrainte considérée comme homogène. Contrairement à ce qui avait été fait dans le chapitre 4, nous n'avons pas appliqué d'algorithme de vérification des damiers dans la topologie. De précédents résultats de la littérature montrent que les topologies à géométries tournantes présentent des propriétés mécaniques intéressantes lorsqu'elles sont homogénéisées. C'est la raison pour laquelle nous avons décidé de conserver la possibilité d'obtenir de telles topologies. En outre, nous n'avons pas utilisé de matériau fictif pour remplacer le vide. Les éléments vides ont été complètement supprimés du maillage. Un algorithme de vérification des chemins de matière a donc été développé, afin de s'assurer de la double continuité de matière : entre les différents VER assemblés, et au sein même du VER.

La construction du modèle numérique de Volume Élémentaire Représentatif a été développée sous Cast3m® et présente les propriétés suivantes :

- La géométrie est un carré en deux dimensions discrétisé en 10×10 éléments finis de type Q4.
- En considérant les hypothèses des méthodes d'homogénéisation, et en particulier de la méthode de moyennation des champs, nous avons appliqué des conditions aux limites périodiques aux 4 côtés de notre VER. Afin d'éliminer les potentiels mouvements de corps rigides, nous avons encastré un coin de notre VER.
- Une déformation considérée comme homogène et calculée en conséquence est appliquée sur l'un des côtés du carré.

Afin de considérer les potentiels grands déplacements dans notre VER déformé, nous avons choisi d'utiliser la dérivée de Jaumann pour calculer les déformations, la formu-

lation Lagrangienne réactualisée, et un repère corotationnel, permettant de travailler à chaque instant de calcul dans la configuration déformée actuelle.

L'optimisation a été réalisée avec notre algorithme I-PR-PSO développé sous MATLAB®, en considérant $c_1 = c_2 = 0.5$. Différentes valeurs de l'inertie ω ont été testées pour chaque cas de calcul : 0.8, 1.2, 1.5 et décroissante linéairement entre 1.2 et 0.2.

Les résultats montrent d'une part qu'il est possible de créer un metamatériau constitué d'acier qui présente des grandes déformations à l'échelle macroscopique, dûes notamment aux grands déplacements présents à l'échelle microscopique dans le VER. Les VER ainsi obtenus présentent des géométries tournantes, telles que celles proposées dans la littérature lors du développement de metamatériaux à coefficient de Poisson négatif. On montre aussi que certaines topologies en particulier présentent une non-linéarité de la courbe $\sigma - \epsilon$ qui est assez proche de celle du caoutchouc. Certains résultats présentent des non-linéarités géométriques dûes au flambage soudain de la structure.

En utilisant un nombre assez grand de particules afin de maximiser les chances de converger vers une solution optimale, nous avons augmenté grandement le temps de calcul. Pour pallier à ce problème, les calculs ont été réalisés de manière parallèle au Mésocentre de Calcul de Franche-Comté.

Dans le futur, nous souhaiterions raffiner le maillage dans le VER, afin de tenter d'affiner les résultats, et d'obtenir une courbe $\sigma - \epsilon$ plus proche de celle du caoutchouc qu'elle ne l'est déjà. Avec l'avancée des technologies concernant le développement des imprimantes 3D, nous pourrions essayer d'imprimer les géométries que nous avons développées ici (en réalisant un carré contenant 8×8 cellules de périodicité [212]), afin de réaliser des essais en traction/compression sur ces nouveaux metamatériaux.

CONCLUSIONS DE LA THÈSE

Ce travail de thèse avait pour objet le développement d'une méthode d'optimisation métaheuristique qualifiée d'intelligente, ainsi que ses différentes applications à la mécanique classique et au développement des metamatériaux.

7.1/ BILAN DES TRAVAUX RÉALISÉS

Le chapitre 3 présente le développement d'un algorithme d'optimisation qualifié d'intelligent, basé conjointement sur l'algorithme d'optimisation par essaim particulaire PSO, et l'algorithme PageRank utilisé par le moteur de recherche Google. Cet algorithme utilise les propriétés des chaînes de Markov afin de redéfinir en temps réel la topologie de la population des particules PSO. En effet, les particules - qui se déplacent à chaque itération dans l'espace des solutions - représentent des potentielles solutions au problème d'optimisation traité. Ainsi, on peut considérer qu'elles représentent des états dans lesquels les particules peuvent être au cours du calcul. On considère donc que notre essaim de particules PSO est une chaîne de Markov, représenté par un graphe probabiliste, dont les nœuds sont les particules, et les arêtes sont les liens qui définissent l'influence qu'elles ont les unes sur les autres. Notre algorithme Inverse-PageRank-PSO utilise le succès relatif des particules au cours des itérations pour définir le vecteur stationnaire π de la chaîne de Markov considérée. Un algorithme PageRank inverse détermine ainsi, à partir de ce vecteur stationnaire, la matrice des connectivités des particules, c'est à dire les facteurs définissant en quelle mesure elles s'influencent les unes les autres au cours des itérations. En modifiant ainsi la pondération des influences des particules les unes sur les autres, notre algorithme a montré qu'il était plus efficace que ses pairs sur les 12 fonctions objectifs mathématiques testées. Il est notamment plus rapide que ses congénères à se rapprocher d'une zone prometteuse du domaine de recherche, mais il se rapproche aussi plus près du minimum global de la fonction objectif considérée. En optimisation avec contraintes, l'algorithme a été validé sur des problèmes mécaniques de treillis, dont on souhaite minimiser la masse, en considérant une contrainte d'optimisation sur les contraintes mécaniques maximales admissibles dans les barres et les déplacements des nœuds. Afin d'adapter I-PR-PSO à l'optimisation avec contraintes, nous l'avons couplé avec une méthode de type *Harmony memory* qui gère les bornes limites des variables d'optimisation, et une méthode de pondération des objectifs qui gère les contraintes intrinsèques au problème pour lesquelles le calcul par éléments finis de la structure est indispensable. Les résultats montrent que notre algorithme est bien adapté à l'optimisation avec contraintes, en trouvant des meilleures solutions que celles données dans la

littérature sur les mêmes problèmes.

Le chapitre 4 propose une nouvelle méthode d'optimisation topologique développée en particulier pour les problèmes mécaniques, basée conjointement sur notre algorithme I-PR-PSO et l'algorithme *Evolutionary Structural Optimization* (ESO). Ici, les deux algorithmes travaillent de manière séquentielle, en considérant chacun un problème d'optimisation différent. Alors que I-PR-PSO tente de minimiser la compliance de la structure en considérant des contraintes d'optimisation sur le volume de la structure, ESO minimise le volume de la structure en considérant des contraintes d'optimisation sur les contraintes équivalentes de Von Mises dans les éléments du maillage. Bien que ces deux problèmes d'optimisation soient contradictoires, ils sont traités de manière séquentielle, de telle sorte que tous les objectifs sont atteints à la fin du calcul, même si l'on doit augmenter légèrement la fonction objectif de l'un pour améliorer l'autre. Cette méthode de séparation des objectifs avait déjà été proposée dans la littérature, appelée [la méthode du compromis](#), [la méthode du but programmé](#) ou encore [la méthode d'ordonnancement lexicographique](#). En outre, afin de gérer les problèmes de damiers que l'on retrouve classiquement en optimisation topologique, un algorithme de vérification des damiers a été développé. Contrairement à ce qui a été fait dans la littérature, cet algorithme modifie la position des particules si elles présentent un ou plusieurs damiers, de telle sorte que l'algorithme garde en mémoire la modification effectuée. Les résultats montrent que les topologies obtenues sont plus régulières que celles obtenues par un algorithme d'optimisation par essaim particulaire seul. En outre, ils montrent aussi que notre algorithme diminue le temps de calcul, en minimisant le nombre d'itérations nécessaire à la convergence, et donc le nombre d'appels au calcul de la fonction objectif par éléments finis.

Le chapitre 5 propose d'adapter l'algorithme PSO à l'optimisation topologique, en utilisant une représentation génotype-phénotype empruntée aux algorithmes génétiques, et de l'appliquer au développement d'une cape d'invisibilité électromagnétique fréquentielle. Ici, un modèle de cellule de périodicité basé sur les équations de Maxwell a été développé, en considérant des conditions aux limites particulières, permettant notamment de minimiser les réflexions retour du champ électromagnétique impactant le domaine. En appliquant notre algorithme à cette cellule de périodicité, nous avons pu déterminer quelle répartition de matériau diélectrique dans le domaine de design minimise effectivement les parties réelle et imaginaire de la perméabilité effective. Du fait que la partie réelle de la perméabilité présente un point d'anti-résonance, mais pas la partie imaginaire, il n'a pas été nécessaire d'utiliser un processus d'optimisation multiobjectif. Nous avons donc employé un processus d'optimisation à deux étapes, en utilisant la topologie finale de la première étape comme une topologie initiale pour la seconde étape. Les résultats montrent que ce processus d'optimisation topologique réussit à converger avec succès vers des solutions acceptables, c'est à dire des solutions qui minimisent effectivement les parties réelle et imaginaire de la perméabilité effective.

Le chapitre 6 propose d'utiliser la version binaire de notre algorithme I-PR-PSO précédemment développée dans le chapitre 4 afin de développer un métamatériau mécanique. Le but de ce travail était de développer un métamatériau constitué d'acier, dont la courbe $\sigma - \epsilon$ se rapproche de celle d'un caoutchouc dans une seule direction de l'espace. Pour ce faire, nous avons développé une cellule de périodicité discrétisée en 10×10 éléments finis de type Q4, dans laquelle nous avons tenté de déterminer la répartition de matière pour qu'elle présente, à l'échelle macroscopique, ces propriétés non-linéaires. L'homogénéisation de ces propriétés mécaniques a été réalisée notamment en lui appliquant des conditions aux limites périodiques, et en considérant une loi de moyennation des champs - ou

loi des mélanges - pour retrouver les contraintes homogénéisées à partir des contraintes dans la cellule. Ne sachant pas, *a priori*, vers quelles topologies notre algorithme d'optimisation convergerait, nous avons tenté de développer un modèle numérique le plus général possible. Pour ce faire, et afin d'être en mesure de considérer des potentielles grandes transformations dans le Volume Élémentaire Représentatif, nous avons utilisé la dérivée de Jaumann pour calculer les déformations, et avons travaillé dans un référentiel corotationnel en utilisant la formulation Lagrangienne réactualisée. Nous n'avons pas employé ici d'algorithme de vérification des damiers dans la topologie. En effet, des topologies à géométries tournantes avaient précédemment été proposées dans la littérature, et présentaient des propriétés mécaniques intéressantes, comme un coefficient de Poisson négatif. Nous souhaitions donc considérer la possibilité de converger vers de telles géométries. Les résultats montrent qu'il est possible de développer un metamatériau constitué d'acier dont les propriétés sont hyperélastiques - ou tout du moins non linéaires - dans une direction de l'espace. En effet, de grands déplacements à l'échelle microscopique induisent de grandes déformations à l'échelle macroscopique.

7.2/ PERSPECTIVES ET AMÉLIORATIONS

La majeure partie des problèmes que nous avons rencontrés concerne le raffinement des maillages que nous avons réalisés lors du développement de modèles numériques de structures, mécaniques ou électromagnétiques. En effet, les capacités de calcul à notre disposition n'ont pas permis d'appliquer nos processus d'optimisation topologique à des maillages suffisamment fins pour obtenir des résultats qui satisfont complètement aux objectifs initiaux à atteindre. Dans le domaine des metamatériaux mécaniques, la dimension des éléments était toutefois limitée par la précision des potentielles imprimantes 3D utilisables pour fabriquer notre metamatériau. En effet, la précision d'impression nous a donné la dimension des éléments finis, et les différentes hypothèses à respecter pour que l'homogénéisation soit correcte nous a imposé une dimension maximale du VER à respecter.

Ainsi, dans le futur, avec l'amélioration des capacités de calcul à notre disposition et de la précision des imprimantes 3D, nous souhaiterions affiner certains résultats en affinant le maillage des modèles numériques réalisés.

En outre, certaines de nos topologies mériteraient d'être fabriquées afin de faire des essais pour vérifier leurs propriétés mécaniques. Nous envisageons notamment d'imprimer en 3D un Volume Représentatif constitué de plusieurs cellules de périodicités trouvées dans le chapitre 6 afin de confirmer leur comportement hyperélastique dans une direction de l'espace. Une phase de reconception devra être réalisée, puisque les géométries tournantes que nous avons considérées sont réalisées par la possible rotation des éléments Q4 autour de l'un des noeuds qui les constituent. On devra donc concevoir des micro-liaisons pivot qui permettent de reproduire ce fonctionnement mécanique.

BIBLIOGRAPHIE

- [1] CUI, C., OHMORI, H., AND SASAKI, M. **Strcutural design by extended ESO method**. In *Proceedings of frontiers of computational sciences symposium, Nagoya, Japan* (11-13 October 2005), pp. 149–156.
- [2] OHMORI, H., FUTAI, H., IJIMA, T., MUTO, A., AND HASEGAWA, Y. **Application of computational morphogenesis to structural design**. In *Proceedings of Frontiers of Computational Sciences Symposium* (11-13 october 2005), pp. 45–52.
- [3] FRASER, A. **Simulation of genetic systems by automatic digital computers**. *Australian Journal of Biological Sciences* 10 (1957), 484–491.
- [4] ZOUTENDIJK, G. **Methods of feasible directions**. Elsevier, 1960.
- [5] DANTZIG, G. **Linear Programming and Extensions**. CA : RAND Corporation, 1963.
- [6] NELDER, J., AND MEAD, R. **A Simplex Method for Function Minimization**. *The Computer Journal* 7, 4 (1965), 308–313.
- [7] VESELAGO, V. **The electrodynamics of substances with simultaneously negative values of permittivity and permeability**. *Soviet Physics Uspekhi* 10, 4 (1968), 509–514.
- [8] VENKAYYA, V. **Design of optimum structures**. *Computers and Structures* 1 (1971), 265–309.
- [9] SHEU, C., AND SCHMIT, L. **Minimum weight design of elastic redundant trusses under multiple static load conditions**. *AIAA Journal* 10 (1972), 155–162.
- [10] SCHMIT, L., AND FARSHI, B. **Some approximation concepts for structural synthesis**. *AIAA Journal* 12 (1974), 692–699.
- [11] HAIMES, Y. Y., HALL, W. A., AND FREEDMAN, H. T. **Multi-objective optimization in wather resources systems : The surrogate worth trade-off method**, vol. 3. 1975.
- [12] HOLLAND, J. **Adaptation in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence**, vol. 8. Oxford, England : U Michigan Press, 1975.
- [13] LIN, J. **Multiple-objective problems : Pareto-optimal solutions by method of proper equality constraints**. *IEEE Transactions on Automatic Control* 21, 5 (1976), 641–650.
- [14] SCHMIT, L., AND MIURA, H. **Approximation concepts for efficient structural synthesis**. Tech. Rep. NASA-CR-2552, NASA, 1976.

- [15] BENSOUSSAN, A., LIONS, J., AND PAPANICOLAOU, G. **Asymptotic analysis for periodic structures**, vol. 5. American Mathematical Soc., 1978.
- [16] GIESY, D. **Calculation of Pareto-Optimal solutions to multiple-objective problems using threshold-of-acceptability constraints**. *IEEE Transactions on Automatic Control* 23, 6 (1978), 1114–1115.
- [17] *La Recherche* 22, 96 (1979), 70p.
- [18] KHAN, M., WILLMERT, K., AND THORNTON, W. **An optimality criterion method for large-scale structures**. *AIAA Journal* 17 (1979), 753–761.
- [19] TABAK, D. **Application of multiobjective optimization in aircraft control systems design**. *Automatica* 15, 5 (1979), 595 – 600.
- [20] BEN-TAL, A. **Characterization of Pareto and Lexicographic Optimal Solutions**. In *Multiple Criteria Decision Making Theory and Application*, G. Fandel and T. Gal, Eds., vol. 177 of *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin Heidelberg, 1980, pp. 1–11.
- [21] SANCHEZ-PALENCIA, A. **Non-homogeneous media and vibration theory**. Springer-Verlag, 1980.
- [22] LATANE, B. **The psychology of social impact**. *American Psychologist* 36 (1981), 343–356.
- [23] HASHIN, Z. **Analysis of composite materials - a survey**. *J. Appl. Mech.* 50 (1983), 481–505.
- [24] MORGAN, M., CHAN, C., HILL, S., AND BARBER, P. **Finite-element-boundary integral formulation for electromagnetic scattering**. *Wave Motion* 6 (1984), 91–103.
- [25] ALMGREN, R. **An isotropic three-dimensional structure with Poisson ratio = -1**. *J. Elasticity* 15 (1985), 427–430.
- [26] SCHAFFER, J. **Multiple objective optimization with vector evaluated genetic algorithms**. In *Proceedings of the 1st International Conference on Genetic Algorithms* (Hillsdale, NJ, USA, 1985), L. Erlbaum Associates Inc., pp. 93–100.
- [27] ACKLEY, D. H. **A Connectionist Machine for Genetic Hillclimbing**. Kluwer Academic Publishers, Norwell, MA, USA, 1987.
- [28] LAKES, R. **Foam structures with a negative Poisson's ratio**. *Science* 235, 4792 (1987), 1038–1040.
- [29] REYNOLDS, C. **Flocks, herds and schools : A distributed behaviour model**. *SIGGRAPH '87 : Proceedings of the 14th annual Conference on Computer Graphics and Interactive Techniques* (1987).
- [30] SUQUET, P. **Elements of homogenization theory for inelastic solid mechanics**. In *Homogenization techniques for composite media*, E. Sanchez-Palencia and A. Zaoui, Eds. Springer-Verlag, Berlin, 1987, pp. 194–275.

- [31] BENDSOE, M., AND KIKUCHI, N. **Generating optimal topologies in structural design using a homogenization method.** *Comp. Meth. Appl. Mech. Engng* 71 (1988), 197–224.
- [32] FRIIS, E., LAKES, R., AND PARK, J. **Negative Poisson's ratio polymeric and metallic foams.** *Journal of Materials Science* 23, 12 (1988), 4406–4414.
- [33] MINSKY, M. **La société de l'esprit.** InterEditions, 1988.
- [34] SCHAFFER, J., Ed. *Proceedings of the Third International Conference on Genetic Algorithms* (San Francisco, CA, USA, 1989), Morgan Kaufmann Publishers Inc.
- [35] GOLDBERG, D. **Genetic Algorithms in search, Optimization and Machine Learning.** Addison-Wesley, 1989.
- [36] ESCHENAUER, H., KOSKI, J., AND OSYCZKA, A. **Multicriteria design optimization : procedures and application.** Springer-Verlag, 1990.
- [37] YUAN, X. **Three-dimensional electromagnetic scattering from inhomogeneous objects by the hybrid moment and finite element method.** *IEEE Transactions on Microwave Theory and Techniques* 38, 8 (1990), 1053–1058.
- [38] ABOUDI, J. **Mechanics of composite materials, a unified micromechanical approach.** Elsevier Science Publishers, Amsterdam, 1991.
- [39] EVANS, K. E. **Auxetic polymers : a new range of materials.** *Endeavour* 15, 4 (1991), 170 – 174.
- [40] KANELLOPOULOS, V., AND WEBB, J. **A numerical study of vector absorbing boundary conditions for the finite-element solution of Maxwell's equations.** *IEEE Microwave and Guided Wave Letters* 1, 11 (1991), 325–327.
- [41] LAKES, R. **Deformation mechanisms in negative Poisson's ratio materials : structural aspects.** *Journal of Materials Science* 26, 9 (1991), 2287–2292.
- [42] CHOI, J., AND LAKES, R. **Non-linear properties of metallic cellular materials with a negative Poisson's ratio.** *Journal of Materials Science* 27, 19 (1992), 5375–5381.
- [43] HOLLISTER, S., AND KIKUCHI, N. **A comparison of homogenization and standard mechanics analyses for periodic porous composites.** *Computational Mechanics* 10 (1992), 73–95.
- [44] JAKOB, W., GORGES-SCHLEUTER, M., AND BLUME, C. In *Application of Genetic Algorithms to Task Planning and Learning.* (1992), R. Männer and B. Manderick, Eds., Elsevier, pp. 293–302.
- [45] JENSEN, E. **Topological structural design using genetic algorithm.** PhD thesis, Purdue University, 1992.
- [46] MILLONAS, M. **Swarms, phase transitions, and collective intelligence.** In *Third artificial life conference, Santa Fe, NM (United States).* 1992.
- [47] ROZVANY, G., ZHOU, M., AND BIRKER, T. **Generalized shape optimization without homogenization.** *Structural Optimization* 4 (1992), 250–252.

- [48] ALLAIRE, G., AND KOHN, R. **Optimal design for minimum weight and compliance in plane stress using extremal microstructures.** *Europ. J. Mech. A/Solids* 12, 6 (1993), 839–878.
- [49] JONES, G., BROWN, R., CLARK, D., WILLETT, P., AND GLEN, R. **Searching databases of two-dimensional and three-dimensional chemical structures using genetic algorithms.** In *Proceedings of the 5th International Conference on Genetic Algorithms* (1993), Morgan Kaufmann Publishers Inc., pp. 597–602.
- [50] KIRSCH, U. **Structural optimization.** Springer-Verlag Berlin Heidelberg, 1993.
- [51] ALDERSON, K., PICKLES, A., NEALE, P., AND EVANS, K. **Auxetic polyethylene : The effect of a negative Poisson's ratio on hardness.** *Acta Metallurgica and Materialia* 42, 7 (1994), 2261 – 2266.
- [52] BERENGER, J. **A perfectly matched layer for the absorption of electromagnetic waves.** *J. Comput. Phys.* 114, 2 (1994), 185–200.
- [53] CHAPMAN, C., SAITOU, K., AND JAKIELA, M. **Genetic algorithms as an approach to configuration and topology design.** *Journal of Mechanical Design* 116 (1994), 1005–1012.
- [54] EVANS, K., NKANSAH, M., AND HUTCHINSON, I. **Auxetic foams : Modelling negative Poisson's ratios.** *Acta Metallurgica and Materialia* 42, 4 (1994), 1289 – 1294.
- [55] HAZANOV, S., AND HUET, C. **Order relationships for boundary conditions effect in heterogeneous bodies smaller than the representative volume.** *Journal of the Mechanics and Physics of Solides* 42 (1994), 1995–2011.
- [56] BENDSOE, M. **Optimization of Structural Topology, Shape and Material.** Springer, Berlin, 1995.
- [57] CHOI, J., AND LAKES, R. **Analysis of elastic modulus of conventional foams and of re-entrant foam materials with a negative Poisson's ratio.** *International Journal of Mechanical Sciences* 37, 1 (1995), 51 – 59.
- [58] EBERHART, R., AND KENNEDY, J. **A new optimizer using particle swarm theory.** In *MHS '95., Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (1995), pp. 39–43.
- [59] HAZANOV, S., AND AMIEUR, M. **On overall properties of elastic heterogeneous bodies smaller than the representative volume.** *International Journal of Engineering Science* 33 (1995), 1289–1301.
- [60] KENNEDY, J., AND EBERHART, R. **Particle swarm optimization.** In *Proceedings., IEEE International Conference on Neural Networks* (1995), vol. 4, pp. 1942–1948.
- [61] N'GUESSAN, J. **Méthode des éléments finis mixtes et conditions aux limites absorbantes pour la modélisation des phénomènes électromagnétiques hyperfréquences.** PhD thesis, Ecole Centrale de Lyon, 1995.
- [62] SACHS, Z. **A perfectly matched anisotropic absorber for use as an absorbing boundary condition.** *IEEE Transactions on Antennas and Propagation* 43, 12 (1995), 1460–1463.

- [63] ALLAIRE, G., BELHACHMI, Z., AND JOUVE, F. **The homogenization method for topology and shape optimization, single and multiple loads case.** *Revue Européenne des Eléments Finis* 5 (1996), 649–672.
- [64] ALLAIRE, G., BONNETIER, E., FRANCFORT, G., AND JOUVE, F. **A topological approach to shape optimization.** *Applied Stochastics and Optimization, O. Mahrenholtz et al. ICIAM 95 Special issue of ZAMM* 76 (1996), 255–258.
- [65] BERRENGER, J. **Three dimensional perfectly matched layer for the absorption of electromagnetic waves.** *J. Comput. Phys.* 127 (1996), 163–179.
- [66] CHU, D. N., XIE, Y., HIRA, A., AND STEVEN, G. **Evolutionary structural optimization for problems with stiffness constraints.** *Finite Elements in Analysis and Design* 21 (1996), 239–251.
- [67] DORIGO, M., MANIEZZO, V., AND COLORNI, A. **The ant system : Optimization by a colony of cooperating agents.** *IEEE Transactions on Systems, Man, and Cybernetics - Part B* 26, 1 (1996), 29–41.
- [68] DUYSINX, P. **Optimisation Topologique : du milieu continu à la structure élastique.** PhD thesis, University of Liege, Belgium, 1996.
- [69] KANE, C., AND SCHOENAUER, M. **Topological optimum design using genetic algorithms.** *Control and Cybernetics* 25, 15 (1996).
- [70] SUN, C., AND VAIDYA, R. **Prediction of composite properties from a representative volume element.** *Composites Science and Technology* 56 (1996), 171–179.
- [71] ALLAIRE, G., BONNETIER, E., FRANCFORT, G., AND JOUVE, F. **Shape optimization by the homogenization method.** *Numerische Mathematik* 76 (1997), 27–68.
- [72] ALLAIRE, G., AND JOUVE, F. **Optimisation topologique de structures élastiques.** *Actes du 3ème Colloque National en Calcul des Structures, B. Peseux et al. eds.* (1997), 49–58.
- [73] KENNEDY, J., AND EBERHART, R. **A discrete binary version of the particle swarm algorithm.** In *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation.* (1997), vol. 5, pp. 4104–4108.
- [74] PRALL, D., AND LAKES, R. **Properties of a chiral honeycomb with a Poisson's ratio of -1.** *International Journal of Mechanical Sciences* 39, 3 (1997), 305–314.
- [75] WOLPERT, D. H., AND MACREADY, W. G. **No free lunch theorems for optimization.** *IEEE Transactions on Evolutionary Computation* 1 (1997), 67–82.
- [76] ANGELINE, P. **Using selection to improve particle swarm optimization.** In *Proceedings IEEE World Congress on Computational Intelligence, Evolutionary Computation* (1998), pp. 84–89.
- [77] BRIN, S., AND PAGE, L. **The Anatomy of a Large-scale Hypertextual Web Search Engine.** *Comput. Netw. ISDN Syst.* 30 (1998), 107–117.
- [78] FONSECA, C. M., AND FLEMING, P. J. **Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part I : A unified formulation.** *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans* 28 (1998), 26–37.

- [79] KENNEDY, J., AND SPEARS, W. **Matching algorithms to problems : An experimental test of the particle swarm and some genetic algorithms on the multi-modal problem generator**. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (1998), pp. 78–83.
- [80] SHENG, X., AND JIN, J. **On the formulation of hybrid finite-element and boundary integral methods for 3-D scattering**. *IEEE Transactions on Antennas and Arrays* 46, 3 (1998), 303–311.
- [81] SHI, Y., AND EBERHART, R. **A modified Particle Swarm Optimizer**. In *Proceedings. IEEE World Congress on Computational Intelligence* (1998), pp. 69–73.
- [82] SHI, Y., AND EBERHART, R. **Parameter selection in Particle Swarm Optimization**. In *Proceedings of the 7th International Conference on Evolutionary Programming, EP' 98, Springer-Verlag, London, UK* (1998).
- [83] SIGMUND, O., AND PETERSSON, J. **Numerical instability in topology optimization : a survey on procedures dealing with checkerboards, mesh-dependencies and local minima**. *Struct. Optim.* 16 (1998), 68–75.
- [84] BENDSOE, M., AND SIGMUND, O. **Material interpolations in topology optimization**. *Arch. Appl. Mech.* 69 (1999), 635–654.
- [85] CLERC, M. **The swarm and the queen : towards a deterministic and adaptive Particle Swarm Optimization**. In *CEC 99. Proceedings of the 1999 Congress on Evolutionary Computation* (1999), vol. 3, p. 1957.
- [86] COELLO, C. **An updated survey of evolutionary multiobjective optimization techniques : state of the art and future trends**. In *CEC 99. Proceedings of the 1999 Congress on Evolutionary Computation* (1999), vol. 1, p. 13.
- [87] HORI, M., AND NEMAT-NASSER, S. **On two micromechanics theories for determining micro-macro relations in heterogeneous solids**. *Mechanics of Materials* 31, 10 (1999), 667–682.
- [88] PECULLAN, S., GIBIANSKY, L., AND TORQUATO, S. **Scale effect on the elastic behaviour of periodic and hierarchical two-dimensional composites**. *Journal of the Mechanics and Physics of Solids* 47 (1999), 1509–1542.
- [89] PENDRY, J., HOLDEN, A., STEWART, W., AND YOUNGS, I. **Magnetism from conductors, and enhanced non-linear phenomena**. *IEEE Trans. Microwave Theory Tech.* 47 (1999), 2075–2084.
- [90] SHI, Y., AND EBERHART, R. **Empirical study of Particle Swarm Optimization**. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99* (1999), vol. 3.
- [91] SUGANTHAN, P. **Particle swarm optimiser with neighbourhood operator**. In *CEC 99. Proceedings of the 1999 Congress on Evolutionary Computation* (1999), vol. 3, p. 1962.
- [92] YANG, X., XIE, Y., AND STEVEN, G. **Bidirectional evolutionary method for stiffness optimization**. *AIAA Journal* 37, 11 (1999), 1483–1488.

- [93] CAMAZINE, S., DENEUBOURG, J., FRANCK, N., SNEYD, J., THERAULAZ, G., AND BONABEAU, E. **Self-organization in Biological Systems**. Princeton University Press, 2000.
- [94] GRIMA, J., AND EVANS, K. **Auxetic behavior from rotating squares**. *Journal of Materials Science Letters* 19, 17 (2000), 1563–1565.
- [95] JAKIELA, M., CHAPMAN, C., DUDA, J., ADEWUYA, A., AND SAITOU, K. **Continuum structural topology design with genetic algorithms**. *Comput. Methods Appl. Mech. Engrg.* 186 (2000), 339–356.
- [96] ZAOU, A., AND MASSON, R. **Micromechanics-based modeling of plastic polychrystals : an affine formulation**. *Material Science and Engineering A* 285 (2000), 418–424.
- [97] BORNERT, M., BRETHER, T., AND GILORMINI, P. **Homogénéisation en mécanique des matériaux 1 - Matériaux aléatoires élastiques et milieux périodiques**. Hermès, Paris, 2001.
- [98] BRANDEL, B., AND LAKES, R. **Negative poisson's ratio polyethylene foams**. *Journal of Materials Science* 36, 24 (2001), 5885–5893.
- [99] DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [100] EBERHART, R., AND SHI, Y. **Tracking and optimizing dynamic systems with particle swarms**. In *Proceedings of the 2001 Congress on Evolutionary Computation* (2001), vol. 1, pp. 94–100.
- [101] HU, X., AND EBERHART, R. **Tracking dynamic systems with PSO : where's the cheese**. *Proceedings of the workshop on particle swarm optimization* (2001), 80–83.
- [102] JIN, Y., OLHOFF, M., AND SENDHOFF, B. **Dynamic weighted aggregation for evolutionary multi-objective optimization : Why does it work and how ?** *Proc. GECCO 2001 Conf.* (2001).
- [103] LOVBJERG, M., RASMUSSEN, T. K., AND KRINK, T. **Hybrid particle swarm optimiser with breeding and subpopulations**. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (2001), Morgan Kaufmann, pp. 469–476.
- [104] PARSOPOULOS, K. E., AND VRAHATIS, M. N. **Particle swarm optimizer in noisy and continuously changing environments**. In *Artificial Intelligence and Soft Computing* (2001), IASTED/ACTA Press, pp. 289–294.
- [105] SIGMUND, O. **A 99 line topology optimization code written in matlab**. *Struct. Multidisc. Optim.* 21 (2001), 120–127.
- [106] ZHOU, M., AND ROZVANY, G. **On the validity of ESO type methods in topology optimization**. *Structural Multidisciplinary optimization* 21 (2001), 80–83.
- [107] COLLETTE, Y., AND SIARRY, P. **Optimisation multiobjectif**. Ed. Techniques Ingénieur, 2002.

- [108] FRÉCON, L. **Éléments de mathématiques discrètes**. Presses Polytechniques et Universitaires Romandes, 2002.
- [109] G. ALLAIRE, F. JOUVE, AND TOADER, A. **A level-set method for shape optimization**. *Comptes Rendus Mathématiques* 334, 12 (2002), 1125 – 1130.
- [110] HU, X., AND EBERHART, R. **Adaptive particle swarm optimization : detection and response to dynamic systems**. In *CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation* (2002), vol. 2, pp. 1666–1670.
- [111] HU, X., AND EBERHART, R. **Multiobjective optimization using dynamic neighborhood particle swarm optimization**. In *CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation* (2002), vol. 2, pp. 1677–1681.
- [112] JIAO. **An effective algorithm for implementing perfectly matched layers in time-domain finite-element simulation of open regions EM problems**. *IEEE Transactions on Antennas and Propagation*, 11 (2002), 1615–1623.
- [113] KENNEDY, J., AND MENDES, R. **Population structure and particle swarm performance**. In *CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation* (2002), vol. 2, pp. 1671–1676.
- [114] KRINK, T., AND LOVBJERG, M. **The lifecycle model : Combining particle swarm optimisation, genetic algorithms and hillclimbers**. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature* (2002), PPSN VII, pp. 621–630.
- [115] LIU, J., AND JIN, J. **A highly effective preconditionner for solving the finite element-boundary integral matrix equation of 3D scattering**. *IEEE Transactions on Antennas and Propagation* 50, 9 (2002), 1212–1221.
- [116] PARSOPOULOS, K., AND VRAHATIS, M. **Particle swarm optimization method for constrained optimization problems**. In *Proceedings of the 2nd Euro-International Symposium on Computational Intelligence, Kosice, Slovakia* (2002).
- [117] PARSOPOULOS, K. E., AND VRAHATIS, M. N. **Particle swarm optimization method in multiobjective problems**. In *In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)* (2002), ACM Press, pp. 603–607.
- [118] SMITH, D., SCHULTZ, S., MARKOS, P., AND SOUKOULIS, C. **Determination of effective permittivity and permeability of metamaterials from reflexion and transmission coefficient**. *Physical Review* 65 (2002), 1–7.
- [119] TCHERNIAK, D. **Topology optimization of resonating structures using SIMP method**. *International Journal for Numerical Methods in Engineering* 54 (2002), 1605–1622.
- [120] BENDSOE, M., AND SIGMUND, O. **Topology optimization, Theory, Methods and Applications**. Springer, 2003.
- [121] DRÉO, J., PÉROWSKI, A., SIARRY, P., TAILLARD, E., DURAND, N., AND GOTTELAND, J. **Métaheuristiques pour l'Optimisation Difficile**. Eyrolles, 2003.

- [122] JANSON, S., AND MIDDENDORF, M. **A hierarchical Particle Swarm Optimizer**. In *Proceedings of The 2003 Congress on Evolutionary Computation, CEC '03*. (2003), vol. 2, pp. 770–776.
- [123] JIAO. **Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers**. *IEEE Transactions on Antennas and Propagation* 51, 2 (2003), 296–305.
- [124] KANIT, T., FOREST, S., GALLIET, I., MOUNOURY, V., AND JEULIN, D. **Determination of the size of the representative volume element for random composites : statistical and numerical approach**. *International Journal of Solids and Structures* 40 (2003), 3647–3679.
- [125] LI, X., AND DAM, K. **Comparing particle swarms for tracking extrema in dynamic environments**. In *CEC '03. The 2003 Congress on Evolutionary Computation* (2003), vol. 3, pp. 1772–1779.
- [126] LIM, T. **Constitutive relationship of a material with unconventional Poisson's ratio**. *Journal of Materials Science Letters* 22, 24 (2003), 1783–1786.
- [127] MENDES, R., KENNEDY, J., AND NEVES, J. **Watch the neighbor or how the swarm can learn from its environment**. In *SIS '03. Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (2003), pp. 88–94.
- [128] SHI, X., LU, Y., ZHOU, C., LEE, H., LIN, W., AND LIANG, Y. **Hybrid evolutionary algorithms based on PSO and GA**. In *Proceedings of The 2003 Congress on Evolutionary Computation, CEC'03* (2003), vol. 4, pp. 2393–2399.
- [129] XIA, Z., ZHANG, Y., AND ELLYIN, F. **A unified periodical boundary conditions for representative volume element of composites and applications**. *International Journal of Solids and Structures* 40 (2003), 1907–1921.
- [130] ZHENG, Y., MA, L., ZHANG, L., AND QIAN, J. **On the convergence analysis and parameter selection in particle swarm optimization**. In *Proceedings of the International Conference on Machine Learning and Cybernetics* (2003), vol. 3, pp. 1802–1807.
- [131] BOTHA, M. **On the variational formulation of hybrid finite-element-boundary integral techniques for electromagnetic analysis**. *IEEE transactions on antennas and propagation* 52, 11 (2004), 3037–3047.
- [132] CHEN, X., GRZEGORCZYK, T., WU, B., PACHECO, J., AND KONG, J. A. **Robust method to retrieve constitutive effective parameters of metamaterials**. *Physical Review E* 70 (2004), 1–7.
- [133] JANSON, S., AND MIDDENDORF, M. **A hierarchical particle swarm optimizer for dynamic optimization problems**. In *Applications of Evolutionary Computing*, vol. 3005 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 513–524.
- [134] LANGVILLE, A. N., AND MEYER, C. D. **Deeper inside PageRank**. *Internet Mathematics* 1 (2004).

- [135] LI, J., AND CHAN, C. **Double-negative acoustic metamaterial**. *Phys. Rev. E* 70, 5 (2004).
- [136] MENDES, R., KENNEDY, J., AND NEVES, J. **The fully informed particle swarm : Simpler, maybe better**. *IEEE Transactions on Evolutionary Computation* 8 (2004), 204–210.
- [137] RATNAWEERA, A., HALGAMUGE, S., AND WATSON, H. **Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients**. *IEEE Transactions on Evolutionary Computation* 8, 3 (2004), 240–255.
- [138] SHEN, Q., JIANG, J., JIAO, C., SHEN, G., AND YU, R. **Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling : QSAR studies of antagonism of angiotensin II antagonists**. *Eur J Pharm Sci* 22 (2004), 145–52.
- [139] ALLAIRE, G., GOURNAY, F. D., AND JOUVE, F. **Optimisation de structures par la méthode des lignes de niveau**. *Proceedings de la conférence CSMA 2005, Giens* (2005).
- [140] ARVIND, S., MENDES, R., WARD, C., AND POSTHOFF, C. **Neighborhood restructuring in particle swarm optimization**. In *LNCS 3809. Proceedings of the 18th Australian Joint Conference on Artificial Intelligence* (2005), Springer, pp. 776–785.
- [141] BRUNS, T. **A reevalutaion of the simp method with filtering and an alternative formulation for solid-void topology optimization**. *Structural Multidisciplinary Optimization* 30 (2005), 428,436.
- [142] COTUK, U. **Scattering from multi-layered metamaterials using wave matrices**. PhD thesis, Naval postgraduate school, Monterey, Canada, 2005.
- [143] FUCHS, M., JINY, S., AND PELEG, N. **The SRV constraint for 0/1 topological design**. *Structural and Multidisciplinary Optimization* 30 (2005), 320–326.
- [144] GASPAR, N., REN, X., SMITH, C., GRIMA, J., AND EVANS, K. **Novel honeycombs with auxetic behaviour**. *Acta Materialia* 53, 8 (2005), 2439 – 2445.
- [145] HOLDEN, N., AND FREITAS, A. **A hybrid Particle Swarm/Ant Colony Algorithm for the classification of hierarchical biological data**. In *Proceedings of the IEEE Swarm Intelligence Symposium, SIS'05* (2005), pp. 100–107.
- [146] LEE, K., AND GEEM, Z. **A new meta-heuristic algorithm for continuous engineering optimization : Harmony Search theory and practice**. *Computer Methods in Applied Mechanics and Engineering* 194 (2005), 3902–3933.
- [147] MARTINEZ, J. **A note on the theoritical convergence properties of the SIMP method**. *Structural Multidisciplinary Optimization* 29 (2005), 319–323.
- [148] MÉNY, J., ALDON, G., AND XAVIER, L. **Introduction à la théorie des graphes : Butinage graphique**. Canopé - CRDP de Lyon, 2005.

- [149] POLI, R., DI, C., CHIO, W., AND LANGDON, B. **Exploring extended particle swarms : A genetic programming approach**. In *In Proceedings of the 2005 conference on Genetic and evolutionary computation* (2005), ACM Press, pp. 169–176.
- [150] POLI, R., LANGDON, W., AND HOLLAND, O. **Extending particle swarm optimisation via genetic programming**. In *Genetic Programming*, M. Keijzer, A. Tettamanzi, P. Collet, J. Hemert, and M. Tomassini, Eds., vol. 3447 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 291–300.
- [151] SALENCON, J. **Mécanique des milieux continus, Tome I, Concepts généraux**. Les éditions de l'Ecole Polytechnique, 2005.
- [152] SEDAGHATI, R. **Benchmark case studies in structural design optimization using the force method**. *International Journal of Solids and Structures* 42 (2005), 5848–5871.
- [153] SMITH, D., VIER, D., KOSCHNY, T., AND SOUKOULIS, C. **Electromagnetic parameter retrieval from inhomogeneous metamaterials**. *Physical Review E* 71 (2005), 1–11.
- [154] COELLO, C., LAMONT, G., AND VELDHIJZEN, D. **Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)**. Springer-Verlag New York, Inc., 2006.
- [155] DRAGO, A., AND PINDER, M. **Micro-macromechanical analysis of heterogeneous materials : Macroscopically homogeneous vs periodic microstructures**. *Composites Sciences and Technology* 67, 6 (2006), 1243–1263.
- [156] FENG, B. **Extracting material constitutive parameters from scattering parameters**. PhD thesis, Naval postgraduate school. Monterey, Canada, 2006.
- [157] GRIMA, J., GATT, R., RAVIRALA, N., ALDERSON, A., AND EVANS, K. **Negative poisson's ratio in cellular foam materials**. *Materials Science and Engineering : A* 423, 1-2 (2006), 214–218.
- [158] IQBAL, M., DE OCA, M., AND MONTES, A. **An estimation of distribution particle swarm optimization algorithm**. In *Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence* (Berlin, Heidelberg, 2006), ANTS'06, Springer-Verlag, pp. 72–83.
- [159] MÉNY, J., ALDON, G., AND XAVIER, L. **Théorie des graphes et applications**. Hermès Lavoisier, 2006.
- [160] NIU, B., ZHU, Y., HU, K., LI, S., AND HE, X. **A novel particle swarm optimizer using optimal foraging theory**. In *Computational Intelligence and Bioinformatics*, D. Huang, K. Li, and G. Irwin, Eds., vol. 4115 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 61–71.
- [161] PASUPULETI, S., AND BATTITI, R. **The gregarious particle swarm optimizer (G-PSO)**. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO '06* (2006), ACM, pp. 67–74.

- [162] PETERSSON, L. **Analysis of periodic structures via a time-domain finite-element formulation with a floquet ABC.** *IEEE Transactions on Antennas and Propagation* 54, 3 (2006), 933–944.
- [163] SID, B. **Optimisation topologique par algorithmes génétiques.** PhD thesis, Université de Technologie de Belfort-Montbéliard, décembre 2006.
- [164] SMITH, D., AND PENDRY, J. **Homogenization of metamaterials by field averaging.** *J. Opt. Soc. Am. B* 23, 3 (2006), 391–403.
- [165] TAWDROSS, P., AND KONIG, A. **Local parameters particle swarm optimization.** *International Conference on Hybrid Intelligent Systems* (2006), 52.
- [166] BLACKWELL, T. **Particle swarm optimization in dynamic environments.** In *Evolutionary Computation in Dynamic and Uncertain Environments*, S. Yang, Y. Ong, and Y. Jin, Eds., vol. 51 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, 2007, pp. 29–49.
- [167] EDWARDS, C., KIM, H., AND BUDD, C. **An evaluative study on ESO and SIMP for optimization of a cantilever tie-beam.** *Structural Multidisciplinary Optimization* 34 (2007), 403–414.
- [168] FAN, S., AND CHANG, J. **A modified particle swarm optimizer using an adaptive dynamic weight scheme.** In *Digital Human Modeling*, V. Duffy, Ed., vol. 4561 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 56–65.
- [169] GITMAN, I., ASKES, H., AND SLUYS, L. **Representative volume element : Existence and size determination.** *Engineering Fracture Mechanics* 74 (2007), 2518–2534.
- [170] HOLDEN, N., AND FRIETAS, A. **A hybrid PSO/ACO algorithm for classification.** In *Proceedings of the 9th Genetic and Evolutionary Computation Conference Workshop on Particle Swarms : The Second Decade, GECCO 07* (2007), ACM Press, pp. 2745–2750.
- [171] HSIEH, C., CHEN, C., AND CHEN, Y. **Particle swarm guided evolution strategy.** In *Proceedings of the 9th annual Conference on Genetic and Evolutionary Computation* (2007), GECCO '07, pp. 650–657.
- [172] JIANG, Y., HU, T., HUANG, C., AND WU, X. **An improved particle swarm optimization algorithm.** *Applied Mathematics and Computation* 193, 1 (2007), 231 – 239.
- [173] LI, L., HUANG, Z., LIU, F., AND WU, Q. **A heuristic particle swarm optimizer for optimization of pin connected structures.** *Computers and Structures* 85 (2007), 340–349.
- [174] LUBKOWSKI, G., SCHUMANN, R., AND WEILAND, T. **Extraction of effective metamaterial parameters by parameter fitting of dispersive model.** *Microw. Opt. Technol. Lett.* 49 (2007), 391–403.
- [175] YANG, X., YUAN, J., YUAN, J., AND MAO, H. **A modified particle swarm optimizer with dynamic adaptation.** *Applied Mathematics and Computation* 189 (2007), 1205–1213.

- [176] ZHANG, W., WANG, F., DAI, G., AND SUN, S. **Topology optimal design of material microstructures using strain energy-based method.** *Chinese Journal of Aeronautics* 20, 4 (2007), 320 – 326.
- [177] ABDULLATIFF, N., TSIMENIDIS, C., SHARIF, B., AND LADHA, C. **Dynamic clustering using binary multi-objective Particle Swarm Optimization for wireless sensor networks.** In *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2008.* (2008), pp. 1–5.
- [178] GRIMA, J., FARRUGIA, P., CARUANA, C., GATT, R., AND ATTARD, D. **Auxetic behaviour from stretching connected squares.** *Journal of Materials Science* 43, 17 (2008), 5962–5971.
- [179] GRIMA, J., ZAMMIT, V., GATT, R., ATTARD, D., CARUANA, C., AND BRAY, T. C. **On the role of rotating tatrahedra for generating auxetic behavior in NAT and related systems.** *Journal of Non Crystalline Solids* 354, 35-39 (2008), 4214–4220.
- [180] HASSAN, M., SCARPA, F., RUZZENE, M., AND MOHAMMED, N. **Smart shape memory alloy chiral honeycomb.** *Materials Science and Engineering : A* 481-482 (2008), 654–657.
- [181] HOLDEN, N., AND FREITAS, A. **A hybrid PSO/ACO algorithm for discovering classification rules in data mining.** *J. Artif. Evol. App.* 2008 (2008), 2–11.
- [182] JIN, J., LOU, Z., LI, Y., RILEY, N., AND RILEY, D. **Finite element analysis of complex antennas and arrays.** *IEEE Transactions on Antennas and Propagation* 58, 8 (2008), 2222–2240.
- [183] KACZMARCZYK, L., PEARCE, C., AND BICANIC, N. **Scale transition and enforcement of rve boundary conditions in second-order computational homogenization.** *International Journal for Numerical Methods in Engineering* 74 (2008), 506–522.
- [184] LEE, S., SOAK, S., OH, S., PEDRYCZ, W., AND JEON, M. **Modified Binary Particle Swarm Optimization.** *Progress in Natural Science* 18, 9 (2008), 1161 – 1166.
- [185] PENG, C., AND XU, X. **A hybrid algorithm based on immune BPSO and N-1 principle for PMU multi-objective optimization placement.** In *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. DRPT 2008.* (2008), pp. 610–614.
- [186] UEDA, T. **Dielectric-resonator-based composite right/left handed transmission lines and their application to leaky wave antenna.** *IEEE Transactions on Microwave Theory and Techniques* 56, 10 (2008), 2259–2269.
- [187] DE OCA, M., STUTZLE, T., BIRATTARI, M., AND DORIGO, M. **Frankenstein’s PSO : A composite Particle Swarm Optimization Algorithm.** *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 1120–1132.
- [188] FERNANDEZ-MARQUEZ, J., AND ARCOS, J. **An evaporation mechanism for dynamic and noisy multimodal optimization.** In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation* (2009), GECCO '09, ACM, pp. 17–24.

- [189] GASPAR, N., SMITH, C., AND EVANS, K. **Auxetic behaviour and anisotropic heterogeneity**. *Acta Materialia* 57, 3 (2009), 875 – 880.
- [190] KURUKURI, S. **Homogenization theory for heterogeneous materials**. In *Macro-Micro Theory on Multifield Coupling Behavior of Heterogeneous Materials*. Springer Berlin Heidelberg, 2009, pp. 7–58.
- [191] ROZVANY, G. **A critical review of established methods of structural topology optimization**. *Structural Multidisciplinary Optimization* 37 (2009), 217–237.
- [192] SIGMUND, O. **Systematic Design of Metamaterials by Topology Optimization**. In *Proceedings of IUTAM Symposium on Modelling Nanomaterials and Nanosystems, Aalborg, Denmark, 19–22 May 2008*. Springer Netherlands, 2009, pp. 151–159.
- [193] WOOD, B. **Metamaterials and invisibility**. *C.R. Physics* (2009), 379–390.
- [194] DIPOLA, F. **Modélisation multi-échelles du comportement thermo-élastique de composites à particules sphériques**. PhD thesis, Ecole Centrale des Arts et des Manufactures - "Ecole Centrale Paris", 2010.
- [195] DOSREIS, F. **Homogénéisation automatique des milieux discrets périodiques. Applications aux mousses polymères et aux milieux auxétiques**. PhD thesis, Institut National Polytechnique de Lorraine, 2010.
- [196] FARSHI, B., AND ZIAZI, A. **Sizing optimization of truss structures by method of centers and force formulation**. *International journal of Solids and Structures* 47 (2010), 2508–2524.
- [197] FERNANDEZ-MARQUEZ, J., AND ARCOS, J. **Adapting particle swarm optimization in dynamic and noisy environments**. In *IEEE Congress on Evolutionary Computation (CEC)* (2010), pp. 1–8.
- [198] GRIMA, J., GATT, R., ELLUL, B., AND CHETCUTI, E. **Auxetic behaviour in non-crystalline materials having star or triangular shaped perforations**. *Journal of Non-Crystalline Solids* 356, 37-40 (2010), 1980–1987.
- [199] GRIMA, J., MANICARO, E., AND ATTARD, D. **Auxetic behaviour from connected different-sized squares and rectangles**. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science* (2010).
- [200] HUANG, X., AND XIE, Y. **A further review of ESO type methods for topology optimization**. *Struct. Multidisc. Optim.* 41 (2010), 671–683.
- [201] PEDERSEN, M. H. **Good parameters for Particle Swarm Optimization**. Tech. rep., Hvass Laboratories, 2010.
- [202] ZHOU, S., LI, W., SUN, G., AND LI, Q. **A level-set procedure for the design of electromagnetic metamaterials**. *Opt. Exp.*, 7 (2010), 6693–6702.
- [203] AKBARI, R., AND ZIARATI, K. **A rank based particle swarm optimization algorithm with dynamic adaptation**. *Journal of Computational and Applied Mathematics* 235, 8 (2011), 2694 – 2714.

- [204] BIANCHI, M., SCARPA, F., BANSE, M., AND SMITH, C. **Novel generation of auxetic open cell foams for curved and arbitrary shapes.** *Acta Materialia* 59, 2 (2011), 686–691.
- [205] DIRRENBERGER, J., FOREST, S., JEULIN, D., AND COLIN, C. **Homogenization of periodic auxetic materials.** *Procedia Engineering* 10 (2011), 1847 – 1852.
- [206] HASHEMI, A., AND MEYBODI, M. **A note on the learning automata based algorithms for adaptive parameter selection in PSO.** *Applied Soft Computing* 11, 1 (2011), 689–705.
- [207] LIMA, C., LAPA, C., PEREIRA, N., CUNHA, J. D., AND ALVIM, A. **Comparison of computational performance of GA and PSO optimization techniques when designing similar systems - typical PWR core case.** *Annals of Nuclear Energy* 38 (2011), 1339–1346.
- [208] LUH, G., LIN, C., AND LIN, Y. **A binary particle swarm optimization for continuum structural topology optimization.** *Appl. Soft Comput.* 11, 2 (Mar. 2011), 2833–2844.
- [209] WANG, L., YE, W., FU, X., AND MENHAS, M. **A modified multi-objective binary particle swarm optimization algorithm.** In *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds., vol. 6729 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 41–48.
- [210] ZHOU, S., LI, W., CHEN, Y., SUN, G., AND LI, Q. **Topology optimization for negative permeability metamaterials using level-set algorithm.** *Acta Materialia* 59 (2011), 2624–2636.
- [211] CZECH, C. **Design of Meta-Materials Outside the Homogenization Limit Using Multiscale Analysis and Topology Optimization.** PhD thesis, Clemson University, SC, USA, 2012.
- [212] CZECH, C., GUARNERI, P., GIBERT, J., AND FADEL, G. **On the accurate analysis of linear elastic meta-material properties for use in design optimization problems.** *Composites Science and Technology* 72 (2012), 580–586.
- [213] DIRRENBERGER, J., FOREST, S., AND JEULIN, D. **Elastoplasticity of auxetic materials.** *Computational Materials Science* 64 (2012), 57 – 61. Proceedings of the 21st International Workshop on Computational Mechanics of Materials (IWCMM 21).
- [214] DOSREIS, F., AND GANGHOFFER, J. **Equivalent mechanical properties of auxetic lattices from discrete homogenization.** *Computational Materials Science* 51, 1 (2012), 314 – 321.
- [215] FERCOQ, O. **Optimization of Perron eigenvectors and applications : From web ranking to chronotherapeutics.** PhD thesis, Ecole Polytechnique, 2012.
- [216] HASAR, U. C., BARROSO, J., SABAH, C., KAYA, Y., AND ERTRUGUL, M. **Differential uncertainly analysis for evaluating the accuracy of S-parameters retrieval methods for electromagnetic properties of metamaterial slabs.** *Optics Express* 20, 27 (2012), 29002–29022.

- [217] KAMINAKIS, N., AND STRAVROULAKIS, G. **Topology optimization for compliant mechanisms, using evolutionary-hybrid algorithms and application to the design of auxetic materials.** *Composites : Part B* 43 (2012), 2655–2668.
- [218] KIM, S., KUSTER, E., HOLLOWAY, C., SCHER, A., AND BAKER-DAVIS, J. **Effective material property extraction of a metamaterial by taking effects into account at TE/TM polarized incident.** *Progress in Electromagnetics Research B* 36 (2012), 1–33.
- [219] LANGVILLE, A., AND MEYER, C. **Google's PageRank and Beyond. The science of search engine ranking.** Princeton University Press, 2012.
- [220] LIU, W., XIAO, J., YAN, J., AND TIAN, B. **Periodic waves in non-linear metamaterials.** *Physics Letters A* 376, 16 (2012), 1344–1346.
- [221] MARS, W., AND OSTBERG, D. **Fatigue damage of an elastometric tank track component.** In *Proceedings of the 12' Simulia Community Conference* (2012).
- [222] NEWTON, P., MASON, J., BETHEL, K., BAZHENOVA, L., NIEVA, J., AND KUHN, P. **A Stochastic Markov Chain Model to Describe Lung Cancer Growth and Metastasis.** *PLoS ONE* 7, 4 (2012).
- [223] OTOMORI, M., YAMADA, T., IZUI, K., NISHIWAKI, S., AND ANDKJAER, J. **A topology optimization method based on the level set method for the design of negative permeability dielectric metamaterials.** *Comput. Methods Appl Mech Eng* 237-240 (2012), 192–211.
- [224] XU, Y., DOMASZEWSKI, M., CHAMORET, D., ZHANG, W., AND KORVINK, J. **Particle Swarm Optimization for non-convex problems of size and shape optimization of trusses.** In *Proceedings of the Eleventh International Conference on Computational Structures Technology, paper 67* (2012).
- [225] **Page wikipedia : Algorithme évolutionniste**, nov 2013.
- [226] **Page wikipedia : Heuristique**, nov 2013.
- [227] **Page wikipedia : Probleme NP-complet**, nov 2013.
- [228] **Page wikipedia : Théorie de la complexité**, nov 2013.
- [229] ALSAYEDNOOR, J., HARRISON, P., AND GUO, Z. **Large strain compressive response of 2-d periodic representative volume element for random foam microstructures.** *Mechanics of Materials* 66 (2013), 7–20.
- [230] ASL, R., ASLANI, M., AND PANAHI, M. **Sizing optimization of trusses structures using a hybridized genetic algorithm.** *Cornell University Library* (2013).
- [231] AZOTI, W., BONFOH, N., KOUTSAWA, Y., BELOUETTAR, S., AND LIPINSKI, P. **Influence of auxeticity of reinforcements on the overall properties of viscoelastic composite materials.** *Mechanics of Materials* 61, 0 (2013), 28 – 38.
- [232] NEWTON, P. K., MASON, J., BETHEL, K., BAZHENOVA, L., NIEVA, J., NORTON, L., AND KUHN, P. **Spreaders and Sponges Define Metastasis in Lung Cancer : A Markov Chain Monte Carlo Mathematical Model.** *Cancer Research* 73, 9 (2013), 2760–2769.

- [233] ROMDHANE, L., CHAABANI, Y., AND ZARDI, H. **A Robust Ant Colony Optimization-based Algorithm for Community Mining in Large Scale Oriented Social Graphs**. *Expert Systems with Applications* (2013).
- [234] **Page wikipedia : Chaîne de Markov**, Janv 2014.
- [235] **Page wikipedia : Pagerank**, Fév 2014.
- [236] **Page wikipedia : Théorie des graphes**, janv 2014.
- [237] CAZACU, R., AND GRAMA, L. **Overview of structural topology optimization methods for plane and solid structures**. *Fascicle of Management and Technological Engineering* 3 (2014).
- [238] DICESARE, N., DOMASZEWSKI, M., AND CHAMORET, D. **A Binary Particle Swarm based Topology Optimization for the Design of Negative Permeability Dielectric Metamaterials**. In *Proceedings of The Twelfth International Conference on Computational Structures Technology, Naples, Italy, 2-5 september 2014* (2014), B. Topping and P. Ivanyi, Eds.
- [239] KAVEH, A., AND ZOLGHADR, A. **Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints**. *Advances in Engineering Software* 76 (2014), 9–30.
- [240] ROZVANY, G., AND LEWINSKI, T. **Topology Optimization in Structural and Continuum Mechanics**. Springer, 2014.
- [241] ZHAO, F. **A nodal variable ESO (BESO) method for structural topology optimization**. *Finite elements in Analysis and Design* 86 (2014), 34–40.
- [242] DICESARE, N., CHAMORET, D., AND DOMASZEWSKI, M. **A new hybrid PSO algorithm based on a stochastic Markov chain model**. *Advances in Engineering Software* 90 (2015), 127–137.
- [243] DICESARE, N., CHAMORET, D., AND DOMASZEWSKI, M. **Optimum topological design of negative permeability dielectric metamaterial using a new binary particle swarm algorithm**. *Advances in Engineering Software In Press* (2015).
- [244] HARDJASAPUTRA, H. **Evolutionary structural optimization as tool in finding strut-and-tie-models for designing reinforced concrete deep beam**. *Procedia Engineering* 125 (2015), 995–1000.
- [245] NI, X., WONG, Z. J., MREJEN, M., WANG, Y., AND ZHANG, X. **An ultrathin invisibility skin cloak for visible light**. *Science* 349, 6254 (2015), 1310–1314.
- [246] ANSYS, INC. **Element Reference**, release 12.0, April 2009.
- [247] ANSYS, INC. **High-Frequency Electromagnetic Analysis Guide**, release 12.0, April 2009.
- [248] ANSYS, INC. **Theory Reference for the Mechanical APDL and Mechanical Applications**, release 12.1, November 2009.

TABLE DES FIGURES

2.1	Blocage d'un algorithme de résolution dans un optimum local	8
2.2	Classification générale des méthodes d'optimisation mono-objectif	10
2.3	Les deux types de fronts de Pareto : Convexe et concave	12
2.4	Principe de la méthode de la distance à un objectif de référence	14
2.5	Les trois types d'optimisation structurelle	18
2.6	(haut) Conditions aux limites et deux initialisations différentes de la topologie. (milieu) Solutions à la 50ème et la 100ème itération pour la première initialisation. (bas) Solutions à la 50ème et la 100ème itération pour la seconde initialisation.	23
2.7	Classification des méthodes metaheuristiques en fonction de la classe du problème à résoudre	27
2.8	(1) La première fourmi trouve la source de nourriture F, via un chemin quelconque (a), puis dépose une piste de phéromones en revenant au nid par le chemin le plus court (b). (2) Les fourmis empruntent indifféremment l'un des quatre chemins, mais renforcent tout de même la piste du chemin le plus court lors de leur retour au nid. (3) Les fourmis empruntent le chemin le plus court. Les chemins les plus longs perdent leur pistes de phéromones qui s'évaporent au court du temps.	29
2.9	(1) Une fourmi choisit un trajet au hasard et trace une piste de phéromones. (2) L'ensemble des fourmis parcourt un certain nombre de trajets, chaque fourmi déposant une certaine quantité de phéromones sur son trajet dépendant de la longueur du trajet qu'elle a emprunté. (3) Alors que les pistes les moins empruntées s'évaporent, les pistes les plus empruntées, et donc les plus courtes, s'intensifient. (4) Le meilleur chemin est finalement le plus court qui a été choisi de manière non intentionnelle par la colonie.	31
2.10	Pseudo code d'un algorithme évolutionnaire	36
2.11	Croisement "1 point" de deux génotypes composés de 5 bits chacun : (a) Un point de coupure est choisi. (b) On réalise l'échange des parties coupées. (c) Résultat du croisement.	36
2.12	Croisement uniforme. (a) Choix des symboles à échanger en fonction du masque. (b) Résultat du croisement uniforme.	37
2.13	Evolution de la population dans l'algorithme LifeCycle	39
2.14	Mode de déplacement des particules dans l'espace des solutions	43
2.15	Représentation géométrique de la topologie LBEST	47

2.16 Représentation géométrique de la topologie GBEST	47
2.17 Représentation géométrique de la topologie four clusters	48
2.18 Représentation géométrique de la topologie pyramidale	48
2.19 Représentation géométrique de la topologie grégaire	49
2.20 Représentation géométrique de la topologie hiérarchique	51
2.21 Représentation géométrique de la topologie partitionnée et hiérarchisée . .	60
2.22 Evolution de l'inertie oméga au cours des itérations [168]	65
2.23 Fonctionnement de la méthode d'optimisation multiobjectif développée par Hu et al.	67
2.24 Un exemple de calcul de Niche Count pour deux particules	70
2.25 Exemple typique de problème à résoudre par la théorie des graphes	71
2.26 Exemple de graphe et définitions	72
2.27 Un exemple de graphe orienté	73
2.28 Exemple de graphe orienté pour le calcul du PageRank	79
2.29 Exemple de graphe orienté pour le calcul du PageRank	80
2.30 Convergence du calcul de PageRank de l'équation (2.55)	81
2.31 Différence de fonctionnement d'une structure réentrante, par rapport à une structure en nid d'abeille classique (haut) Déformation d'une structure à nid d'abeille (bas) Déformation d'une structure réentrante	85
2.32 Géométrie d'une structure auxétique à carrés tournants.	86
2.33 Géométrie d'une structure auxétique à triangles rotatifs.	86
2.34 Fabrication d'une mousse auxétique à structure réentrante : (Gauche) Mi- crostructure initiale de la mousse en structure nid d'abeille. (Droite) La même mousse après le procédé de compression/chauffage.	87
2.35 Différents types de Volumes Elémentaires Représentatifs, en fonction de la forme du matériau auxétique considéré. (Gauche) Les cellules élémen- taires. (Droite) Les treillis assemblés des VER	89
2.36 Méthode de changement d'échelle : définition du VER [183]	90
2.37 Conditions aux limites de périodicité avec application d'un déplacement . .	95
3.1 Graphe orienté simplifié	104
3.2 Exemple de graphe et définitions	107
3.3 Convergence du résidu r_m au cours des itérations	108
3.4 Dimension 10	111
3.5 Dimension 20	111
3.6 Dimension 30	111
3.7 Dimension 50	111

3.8 Ackley	113
3.9 Griewank	113
3.10 Rastrigin	113
3.11 Rosenbrock	113
3.12 Sphere	113
3.13 Rotate hyper ellipsoid function	113
3.14 Shifted Rastrigin	114
3.15 Shifted Rosenbrock	114
3.16 Shifted Sphere	114
3.17 Shifted Ackley	114
3.18 Bohachevsky	114
3.19 Schwefel's problem 1.2	114
3.20 Treillis à 10 barres	116
3.21 Treillis à 22 barres	117
3.22 Figure récapitulative de la méthode <i>Harmony Memory</i>	119
4.1 Décodage de la matrice de topologie	127
4.2 Modification de la procédure de binarisation de I-PR-PSO	128
4.3 Procédure de modification des damiers dans la topologie	129
4.4 Groupements de bits de la matrice de topologie donnant lieu à des damiers dans la structure	129
4.5 Représentation graphique de l'algorithme de vérification des damiers dans la topologie	131
4.6 Le problème classique de la cantilever beam	134
4.7 Analyse des paramètres de confiance : courbe de convergence complète .	136
4.8 Analyse des paramètres de confiance : début de la courbe de convergence	136
4.9 Analyse des paramètres de confiance : fin de la courbe de convergence .	137
4.10 Etude de l'influence du paramètre ω : Courbe de convergence complète .	138
4.11 Etude de l'influence du paramètre ω : Zoom sur le début de la convergence	139
4.12 Etude de l'influence du paramètre ω : Zoom sur la fin de la convergence .	139
4.13 Etude de l'influence du nombre de particules : courbe complète	140
4.14 Etude de l'influence du nombre de particules : début de la convergence .	141
4.15 Etude de l'influence du nombre de particules : fin de la convergence . . .	141
4.16 Fonction objectif	151
4.17 Rapport V/V_0	151
4.18 Fonction objectif	153

4.19 Rapport V/V_0	153
4.20 Fonction objectif	153
4.21 Rapport V/V_0	153
5.1 Périodicité du modèle numérique	158
5.2 Modèle de la cellule de périodicité développé	159
5.3 Convergence de la fonction objectif de la première étape du processus d'optimisation, c'est à dire de la partie imaginaire de la perméabilité effective, à 10 THz	168
5.4 Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 10 THz	168
5.5 Convergence de la fonction objectif de la seconde étape du processus d'optimisation, c'est à dire de la partie réelle de la perméabilité effective, à 10 THz	168
5.6 Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 10 THz	169
5.7 Topologie du design domain obtenue en minimisant la perméabilité effective à 10THz. Gauche : vue de face. Droite : vue en perspective.	169
5.8 Convergence de la fonction objectif de la première étape du processus d'optimisation, c'est à dire de la partie réelle de la perméabilité effective, à 15THz	170
5.9 Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 15THz	170
5.10 Convergence de la fonction objectif de la seconde étape du processus d'optimisation, c'est à dire de la partie imaginaire de la perméabilité effective, à 15THz	171
5.11 Evolution de la perméabilité effective, calculée à partir des S-parameters. Minimisation à 15THz	171
5.12 Topologie du design domain obtenue en minimisant la perméabilité effective à 15THz. Gauche : vue de face. Droite : vue en perspective.	172
6.1 Schéma illustrant le fonctionnement de la pièce mécanique à optimiser	177
6.2 Contraintes thermiques dans la pièce mécanique à optimiser	177
6.3 Conditions aux limites de périodicité avec application d'un déplacement	179
6.4 Périodicité de la topologie	180
6.5 Continuité de la matière dans l'assemblage des VER du metamatériau	180
6.6 Chemins de matière dans la topologie	180
6.7 Symétrie imposée dans la topologie finale	181
6.8 Configurations du système	183
6.9 Déformation d'un vecteur dans le système $\mathcal{S}$	184

6.10 Courbe $\sigma - \lambda$ du caoutchouc	185
6.11 Cas 1 : Courbes de convergence des 4 calculs d'optimisation topologique .	189
6.12 Cas 1 : topologie obtenue pour $\omega = 0.8$	190
6.13 Cas 1 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$	190
6.14 Cas 1 : topologie obtenue pour ω décroissant	191
6.15 Cas 1 : topologie obtenue pour $\omega = 0.8$ en configuration déformée	191
6.16 Cas 1 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$ en configuration déformée	192
6.17 Cas 1 : topologie obtenue pour ω décroissant en configuration déformée . .	192
6.18 Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 1	193
6.19 Cas 2 : Courbes de convergence des 4 calculs d'optimisation topologique .	194
6.20 Cas 2 : topologie obtenue pour $\omega = 0.8$	195
6.21 Cas 2 : topologie obtenue pour $\omega = 1.2$	195
6.22 Cas 2 : topologie obtenue pour $\omega = 1.5$	196
6.23 Cas 2 : topologie obtenue pour ω décroissant	196
6.24 Cas 2 : topologie obtenue pour $\omega = 0.8$ en configuration déformée	197
6.25 Cas 2 : topologie obtenue pour $\omega = 1.2$ en configuration déformée	197
6.26 Cas 2 : topologie obtenue pour $\omega = 1.5$ en configuration déformée	198
6.27 Cas 2 : topologie obtenue pour ω décroissant en configuration déformée . .	198
6.28 Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 2	199
6.29 Cas 3 : Courbes de convergence des 4 calculs d'optimisation topologique .	200
6.30 Cas 3 : topologie obtenue pour $\omega = 0.8$	201
6.31 Cas 3 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$	201
6.32 Cas 3 : topologie obtenue pour ω décroissant	202
6.33 Cas 3 : topologie obtenue pour $\omega = 0.8$ en configuration déformée	202
6.34 Cas 3 : topologie obtenue pour $\omega = 1.2$ et $\omega = 1.5$ en configuration déformée	203
6.35 Cas 3 : topologie obtenue pour ω décroissant en configuration déformée . .	203
6.36 Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 3	204
6.37 Cas 4 : Courbes de convergence des 4 calculs d'optimisation topologique .	205
6.38 Cas 4 : topologie obtenue pour $\omega = 0.8$	206
6.39 Cas 4 : topologie obtenue pour $\omega = 1.2$	206
6.40 Cas 4 : topologie obtenue pour $\omega = 1.5$	207
6.41 Cas 4 : topologie obtenue pour ω décroissant	207
6.42 Cas 4 : topologie obtenue pour $\omega = 0.8$	208

6.43 Cas 4 : topologie obtenue pour $\omega = 1.2$	208
6.44 Cas 4 : topologie obtenue pour $\omega = 1.5$	209
6.45 Cas 4 : topologie obtenue pour ω décroissant	209
6.46 Courbes contraintes-déformations complètes obtenues lors de l'analyse de cas de charge 4	210
6.47 Courbe de compression complète : Courbes de convergence des 4 calculs d'optimisation topologique	211
6.48 Courbe de compression complète : topologie obtenue pour $\omega = 0.8$ et $\omega = 1.2$	212
6.49 Courbe de compression complète : topologie obtenue pour $\omega = 1.5$ et ω décroissant	212
 B.1 Ackley	 258
B.2 Griewank	258
B.3 Rastrigin	258
B.4 Rosenbrock	258
B.5 Sphere	258
B.6 Rotate hyper ellipsoid function	258
B.7 Shifted Rastrigin	259
B.8 Shifted Rosenbrock	259
B.9 Shifted Sphere	259
B.10 Shifted Ackley	259
B.11 Bohachevsky	259
B.12 Schwefel's problem 1.2	259
B.13 Ackley	260
B.14 Griewank	260
B.15 Rastrigin	260
B.16 Rosenbrock	260
B.17 Sphere	260
B.18 Rotate hyper ellipsoid function	260
B.19 Shifted Rastrigin	261
B.20 Shifted Rosenbrock	261
B.21 Shifted Sphere	261
B.22 Shifted Ackley	261
B.23 Bohachevsky	261
B.24 Schwefel's problem 1.2	261
B.25 Ackley	262

B.26 Griewank	262
B.27 Rastrigin	262
B.28 Rosenbrock	262
B.29 Sphere	262
B.30 Rotate hyper ellipsoid function	262
B.31 Shifted Rastrigin	263
B.32 Shifted Rosenbrock	263
B.33 Shifted Sphere	263
B.34 Shifted Ackley	263
B.35 Bohachevsky	263
B.36 Schwefel's problem 1.2	263
C.1 Résultats de la littérature [208]. Optimisation topologique avec un algorithme d'optimisation par essaim particulaire seul	279
C.2 Résultats de la littérature [208]. Optimisation topologique avec un algorithme d'optimisation par essaim particulaire seul	280

LISTE DES TABLES

2.1	Détection de changements dans le domaine et réponses associées	59
2.2	Matrice d'adjacence du graphe de la figure 2.26	74
2.3	Matrice d'adjacence du graphe de la figure 2.27	74
2.4	Table d'adjacence du graphe de la figure 2.27	75
2.5	Matrice d'incidence du graphe de la figure 2.27	75
2.6	Table d'incidence du graphe de la figure 2.27	75
3.1	Paramètres du calcul d'optimisation	109
3.2	Benchmark de fonctions mathématiques	110
3.3	100 calculs d'optimisation : valeurs obtenues après 600 itérations PSO (moyenne $\pm$ écart type)	112
3.4	Sections minimales et maximales admissibles du treillis à 10 barres	116
3.5	Sections minimales et maximales admissibles du treillis à 22 barres	117
3.6	Définition des trois cas de chargement du treillis à 22 barres	117
3.7	Limites des contraintes mécaniques admissibles dans les barres du treillis à 22 barres	118
3.8	Paramètres mécaniques des treillis traités	118
3.9	Résultats obtenus sur le treillis à 10 barres	121
3.10	Contraintes mécaniques dans les barres du treillis à 10 barres	121
3.11	Déplacements des noeuds du treillis à 10 barres	122
3.12	Résultats obtenus sur le treillis à 22 barres	122
3.14	Déplacements des noeuds du treillis à 22 barres	123
3.15	Comparaison des résultats avec ceux de la littérature. Treillis à 22 barres, cas de chargement $n^o 1$	123
3.13	Contraintes mécaniques dans les barres pour les treillis à 22 barres	124
4.1	Paramètres mécaniques du problème à traiter	134
4.2	Paires des paramètres de confiance testées	135
4.3	Paramètres mécaniques du problème à traiter	142
4.4	Benchmark de problèmes mécaniques	144

4.5 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 1	145
4.6 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 2	146
4.7 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 3	147
4.8 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 4	148
4.9 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 5	149
4.10 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 6	150
4.11 Résultats (topologie obtenue et contraintes de Von Mises dans la structure) : cas 7	151
5.1 Propriétés des matériaux employés	158
5.2 Paramètres numériques du calcul	165
5.3 Calcul des fréquences d'impact	167
5.4 Résultats globaux obtenus en minimisant la perméabilité effective à 10 THz	169
5.5 Résultats globaux, en minimisant la perméabilité effective à 15THz	171
6.1 Paramètres de l'acier utilisés dans notre processus d'optimisation topologique	178
6.2 Points de la courbe à retrouver	187
6.3 Cas 1 : Valeurs numériques obtenues	189
6.4 Cas 2 : Valeurs numériques obtenues	195
6.5 Cas 3 : Valeurs numériques obtenues	200
6.6 Cas 4 : Valeurs numériques obtenues	205
6.7 Valeurs numériques obtenues lors de l'étude de la courbe de compression complète	213
C.1 Courbes de convergence de ESO-I-PR-PSO dans le cas 1	266
C.2 Courbes de convergence de ESO-I-PR-PSO dans le cas 1	267
C.3 Courbes de convergence de ESO-I-PR-PSO dans le cas 2	268
C.4 Courbes de convergence de ESO-I-PR-PSO dans le cas 2	269
C.5 Courbes de convergence de ESO-I-PR-PSO dans le cas 3	270
C.6 Courbes de convergence de ESO-I-PR-PSO dans le cas 3	271
C.7 Courbes de convergence de ESO-I-PR-PSO dans le cas 4	272
C.8 Courbes de convergence de ESO-I-PR-PSO dans le cas 4	273

C.9	Courbes de convergence de ESO-I-PR-PSO dans le cas 5	274
C.10	Courbes de convergence de ESO-I-PR-PSO dans le cas 6	275
C.11	Courbes de convergence de ESO-I-PR-PSO dans le cas 7	276
C.12	Courbes de convergence de ESO-I-PR-PSO dans le cas 7	277
D.1	Valeurs atteintes par les deux étapes de notre algorithme d'optimisation ESO-I-PR-PSO	282

LISTE DES ALGORITHMES

1	L'algorithme ESO global	22
2	L'algorithme "Ant System"	32
3	L'algorithme évolutionnaire	35
4	L'algorithme "life cycle"	38
5	L'algorithme PSO-GA parallèle	40
6	L'algorithme PSO-GA en série	41
7	L'algorithme PSO en variables continues	44
8	PSO binaire : Changement d'état des particules binaires	54
9	Amélioration de BPSO par Shen et al : Passage du continu au binaire	55
10	Formules permettant le passage des valeurs génotypiques aux valeurs phénotypiques	56
11	Détail du processus itératif permettant le calcul de la matrice de transition d'une chaîne de Markov, connaissant son vecteur stationnaire	102
12	Algorithme permettant de calculer le facteur δ	102
13	Algorithme I-PR-PSO global	107
14	Algorithme permettant de calculer les paramètres γ et θ	121
15	Algorithme de vérification des damiers dans la topologie	130
16	Algorithme ESO-I-PR-PSO global	133
17	Algorithme PSO binaire global	165
18	Algorithme de calcul de la fonction objectif	187

TABLE DES ANNEXES

Numérotation	Titre	Pages de citation
A	Etats de sollicitations	93
B	Courbes de convergence obtenues avec l'algorithme I-PR-PSO	114
C	Résultats complémentaires : courbes de convergence des calculs d'optimisation topologique réalisés avec ESO-I-PR-PSO	143 152
D	Résultats complémentaires : Valeurs atteintes par l'algorithme ESO-I-PR-PSO	152
E	Exemple de modèle numérique de VER défini avec Cast3m	188

ETATS DE SOLLICITATIONS

Ce chapitre donne les différents états de sollicitation nécessaires à la détermination des composantes du tenseur d'élasticité d'un matériau que l'on souhaite homogénéisé.

On rappelle ici que l'énergie totale de déformation U dans le volume V de la structure homogène est donnée par

$$U = \frac{1}{2} \bar{\sigma}_{ij} \bar{\epsilon}_{ij} V \quad (\text{A.1})$$

Troisième état de sollicitation

On suppose ici que l'on impose un déplacement unitaire de la forme suivante

$$\bar{\epsilon} = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T \quad (\text{A.2})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [C_{1133} \ C_{1133} \ C_{3333} \ 0 \ 0 \ 0]^T \quad (\text{A.3})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} C_{3333} V \\ \Rightarrow C_{3333} &= \frac{2U}{V} \end{aligned} \quad (\text{A.4})$$

Quatrième état de sollicitation

On suppose ici que l'on impose un déplacement unitaire de la forme suivante

$$\bar{\epsilon} = [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T \quad (\text{A.5})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [0 \ 0 \ 0 \ 0 \ 0 \ C_{1212}] \quad (\text{A.6})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} C_{1212} V \\ \Rightarrow C_{1212} &= \frac{2U}{V} \end{aligned} \quad (\text{A.7})$$

Cinquième état de sollicitation

On suppose ici que l'on impose un déplacement unitaire de la forme suivante

$$\bar{\epsilon} = [0 \ 0 \ 0 \ 1 \ 0 \ 0]^T \quad (\text{A.8})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [0 \ 0 \ 0 \ C_{2323} \ 0 \ 0]^T \quad (\text{A.9})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} C_{2323} V \\ \Rightarrow C_{2323} &= \frac{2U}{V} \end{aligned} \quad (\text{A.10})$$

Sixième état de sollicitation

On suppose ici que l'on impose un déplacement unitaire de la forme suivante

$$\bar{\epsilon} = [0 \ 0 \ 0 \ 0 \ 1 \ 0]^T \quad (\text{A.11})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [0 \ 0 \ 0 \ C_{1313} \ 0]^T \quad (\text{A.12})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} C \\ &= \frac{1}{2} C_{1313} V \\ \Rightarrow C_{1313} &= \frac{2U}{V} \end{aligned} \quad (\text{A.13})$$

Septième état de sollicitation

On suppose ici que l'on impose un déplacement de la forme suivante

$$\bar{\epsilon} = [1 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad (\text{A.14})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = [C_{1111} + C_{1122} \ C_{2211} + C_{2222} \ C_{3311} + C_{3322} \ 0 \ 0 \ 0]^T \quad (\text{A.15})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} (C_{1111} + C_{1122} + C_{2211} + C_{2222}) V \\ &= \frac{1}{2} (C_{1111} + 2C_{1122} + C_{2222}) \\ \Rightarrow C_{1122} &= \frac{U}{V} - \frac{C_{1111}}{2} - \frac{C_{2222}}{2} \end{aligned} \quad (\text{A.16})$$

Huitième état de sollicitation

On suppose ici que l'on impose un déplacement de la forme suivante

$$\bar{\epsilon} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.17})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = \begin{bmatrix} C_{1122} + C_{1133} & C_{2222} + C_{2233} & C_{3322} + C_{3333} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.18})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} (C_{2222} + C_{2233} + C_{3322} + C_{3333}) V \\ &= \frac{1}{2} (C_{2222} + 2C_{2233} + C_{3333}) \\ \Rightarrow C_{2233} &= \frac{U}{V} - \frac{C_{2222}}{2} - \frac{C_{3333}}{2} \end{aligned} \quad (\text{A.19})$$

Neuvième état de sollicitation

On suppose ici que l'on impose un déplacement de la forme suivante

$$\bar{\epsilon} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.20})$$

Le vecteur des contraintes peut donc être écrit de la manière suivante

$$\bar{\sigma} = \begin{bmatrix} C_{1111} + C_{1133} & C_{2211} + C_{2233} & C_{3311} + C_{3333} & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.21})$$

En remplaçant ces deux expressions dans l'équation A.1, on a

$$\begin{aligned} U &= \frac{1}{2} \bar{\sigma} \bar{\epsilon} V \\ &= \frac{1}{2} (C_{1111} + C_{1133} + C_{3311} + C_{3333}) V \\ \Rightarrow C_{1133} &= \frac{U}{V} - \frac{C_{1111}}{2} - \frac{C_{3333}}{2} \end{aligned} \quad (\text{A.22})$$

B

COURBES DE CONVERGENCE OBTENUES AVEC L'ALGORITHME I-PR-PSO

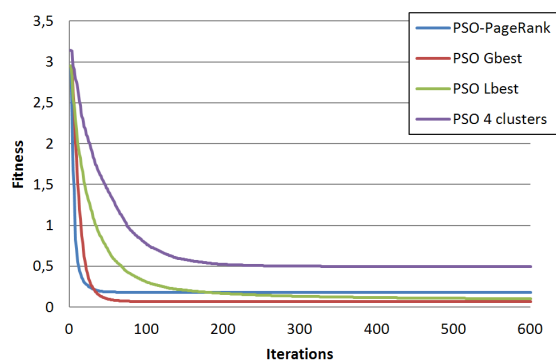


FIGURE B.1 – Ackley

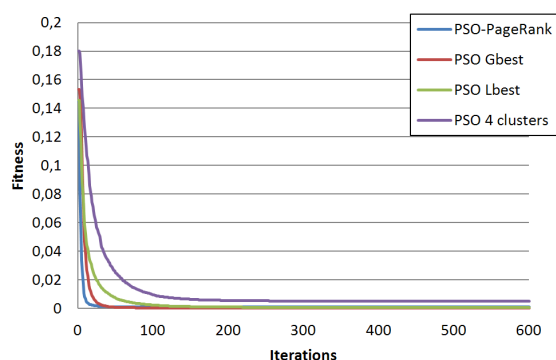


FIGURE B.2 – Griewank

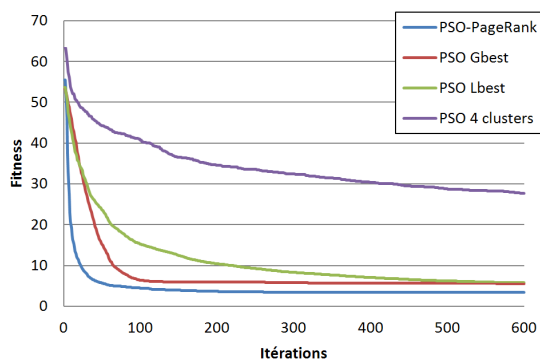


FIGURE B.3 – Rastrigin

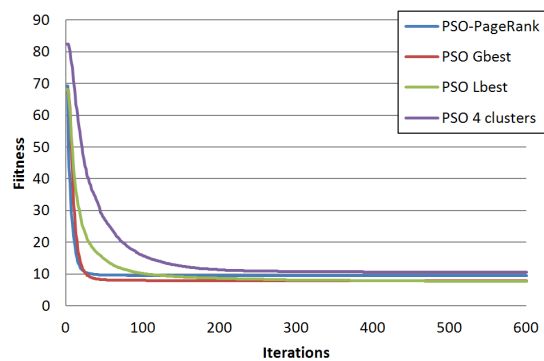


FIGURE B.4 – Rosenbrock

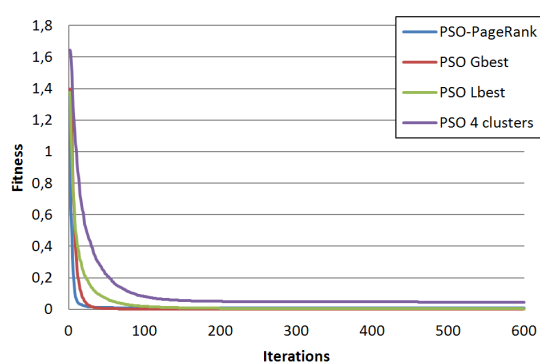


FIGURE B.5 – Sphere

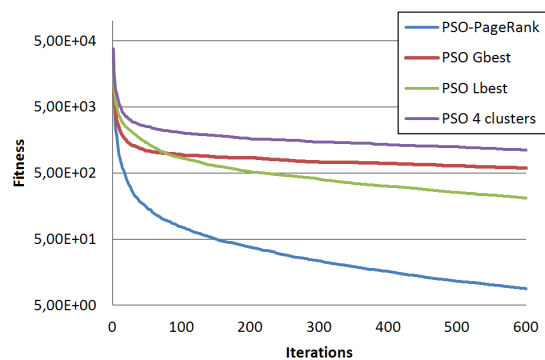


FIGURE B.6 – Rotate hyper ellipsoid function

Figures B.1 - B.6 : Courbes de convergence de I-PR-PSO en dimension 10 (moyenne des 100 calculs réalisés, pour chaque itération)

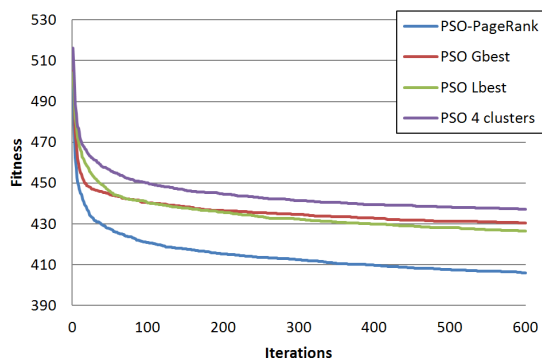


FIGURE B.7 – Shifted Rastrigin

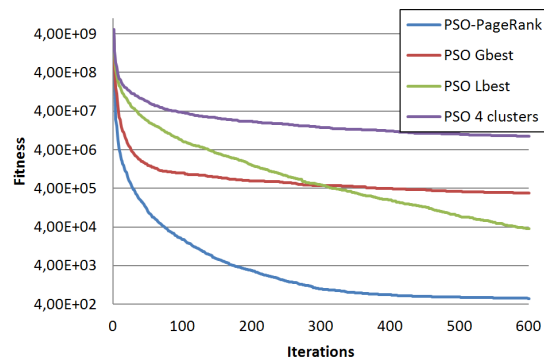


FIGURE B.8 – Shifted Rosenbrock

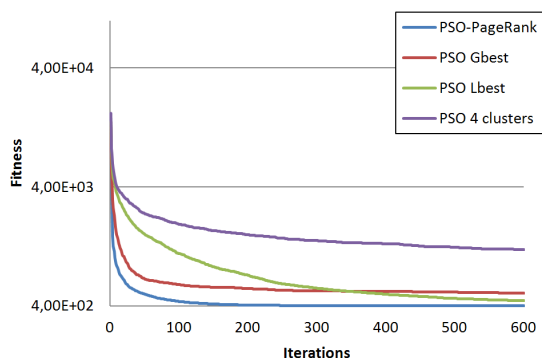


FIGURE B.9 – Shifted Sphere

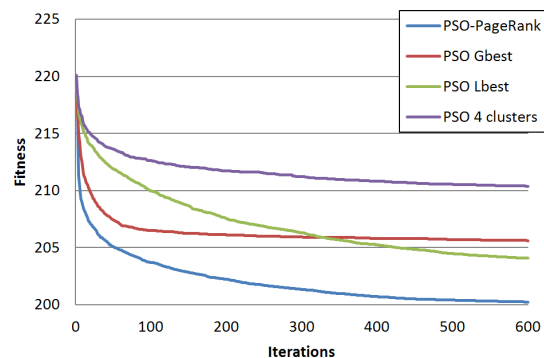


FIGURE B.10 – Shifted Ackley

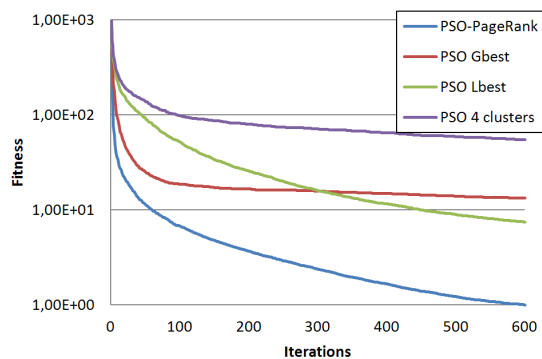


FIGURE B.11 – Bohachevsky

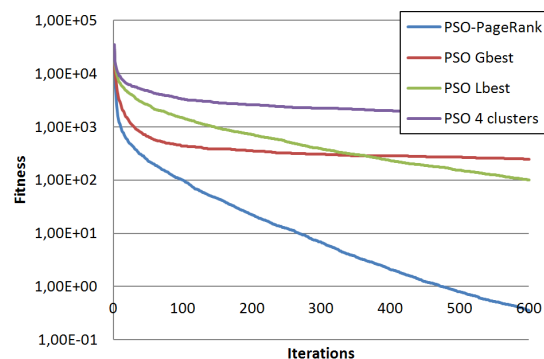


FIGURE B.12 – Schwefel's problem 1.2

Figures B.7 - B.12 : Courbes de convergence de I-PR-PSO en dimension 10 (moyenne des 100 calculs réalisés, pour chaque itération)

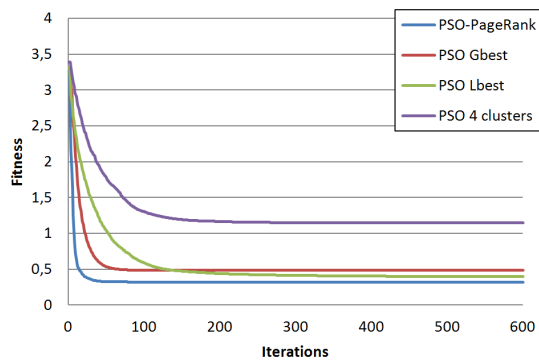


FIGURE B.13 – Ackley

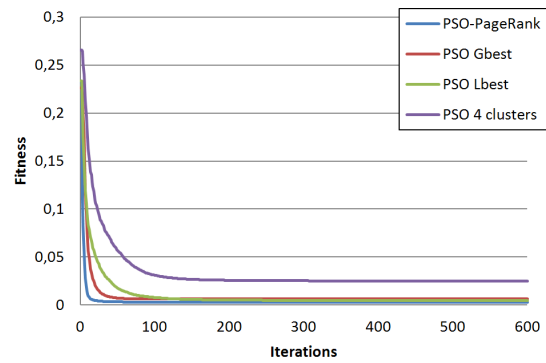


FIGURE B.14 – Griewank

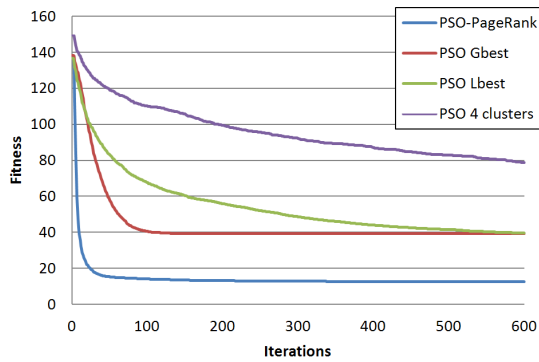


FIGURE B.15 – Rastrigin

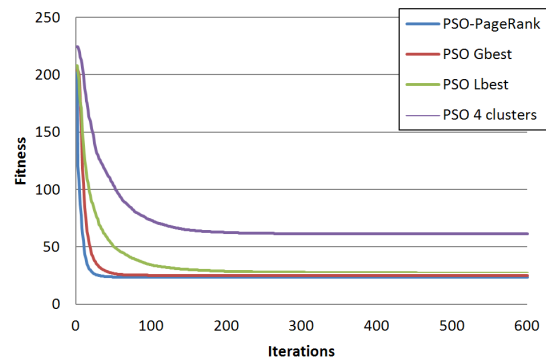


FIGURE B.16 – Rosenbrock

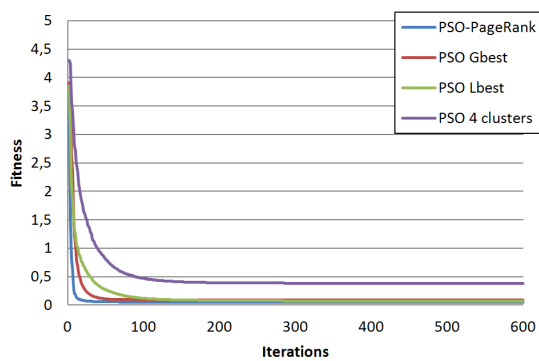


FIGURE B.17 – Sphere

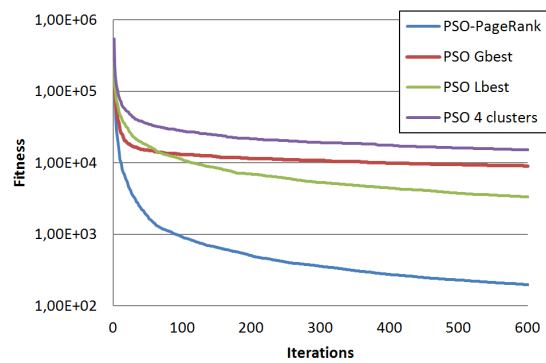


FIGURE B.18 – Rotate hyper ellipsoid fonction

Figures B.13 - B.18 : Courbes de convergence de I-PR-PSO en dimension 20 (moyenne des 100 calculs réalisés, pour chaque itération)

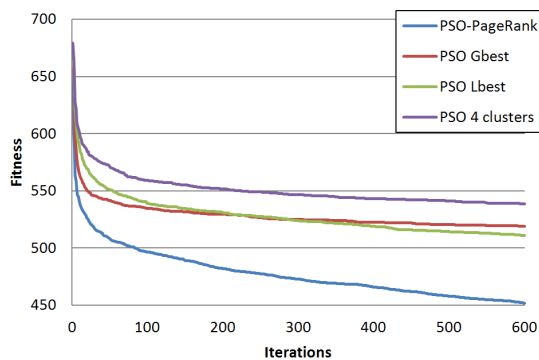


FIGURE B.19 – Shifted Rastrigin

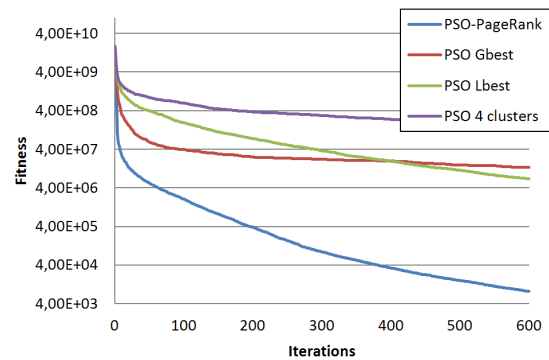


FIGURE B.20 – Shifted Rosenbrock

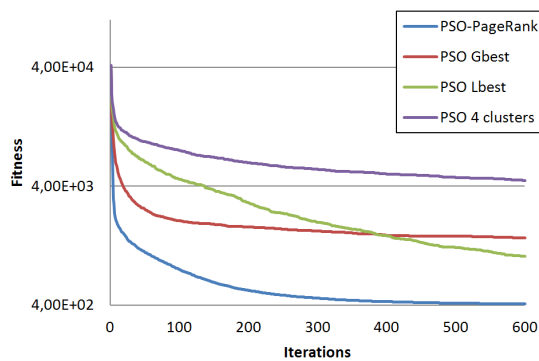


FIGURE B.21 – Shifted Sphere

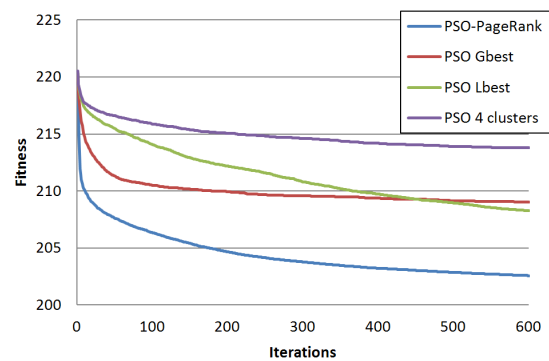


FIGURE B.22 – Shifted Ackley

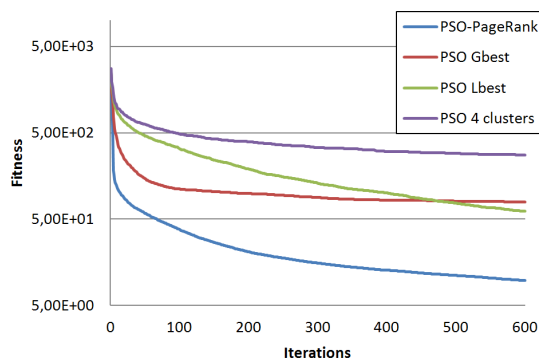


FIGURE B.23 – Bohachevsky

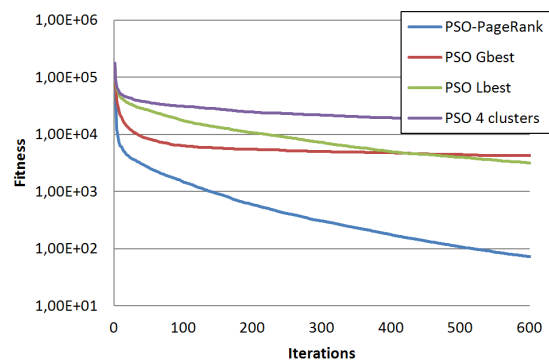


FIGURE B.24 – Schwefel's problem 1.2

Figures B.19 - B.24 : Courbes de convergence de I-PR-PSO en dimension 20 (moyenne des 100 calculs réalisés, pour chaque itération)

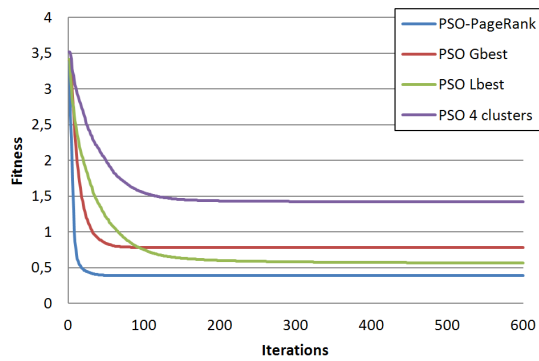


FIGURE B.25 – Ackley

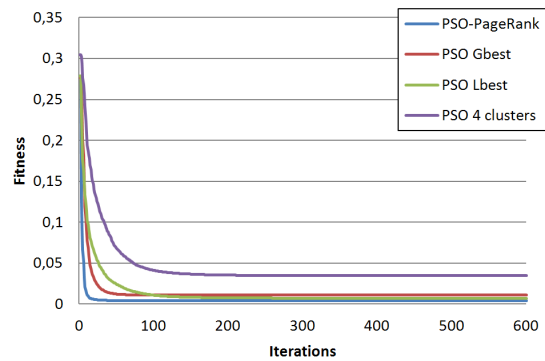


FIGURE B.26 – Griewank

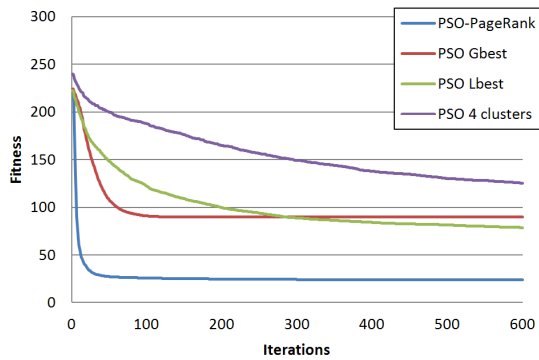


FIGURE B.27 – Rastrigin

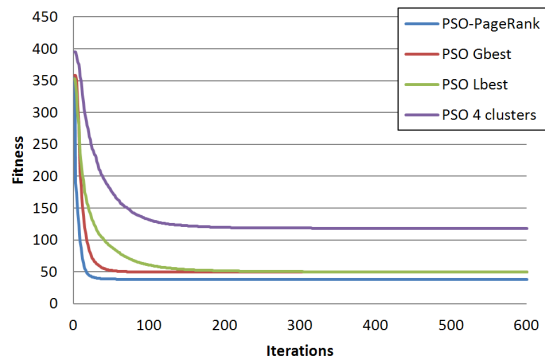


FIGURE B.28 – Rosenbrock

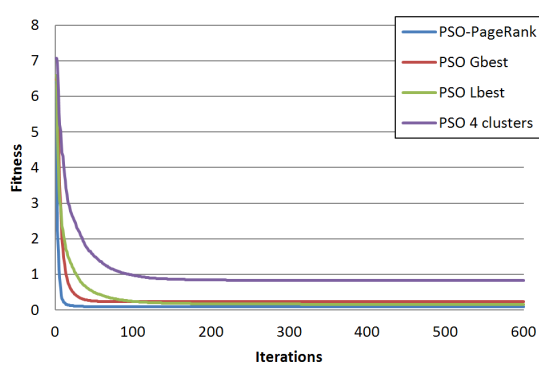


FIGURE B.29 – Sphere

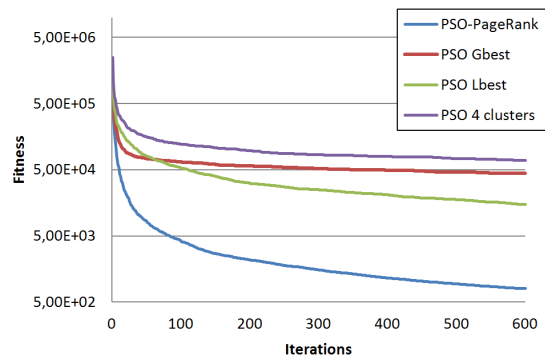


FIGURE B.30 – Rotate hyper ellipsoid function

Figures B.25 - B.30 : Courbes de convergence de I-PR-PSO en dimension 30 (moyenne des 100 calculs réalisés, pour chaque itération)

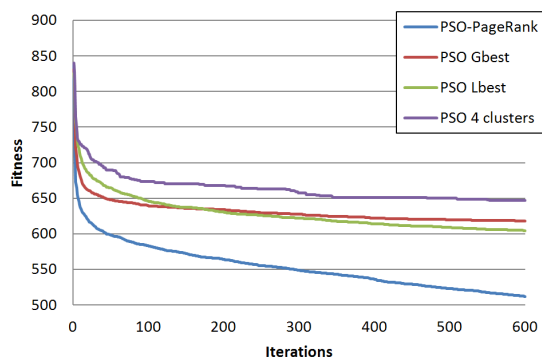


FIGURE B.31 – Shifted Rastrigin

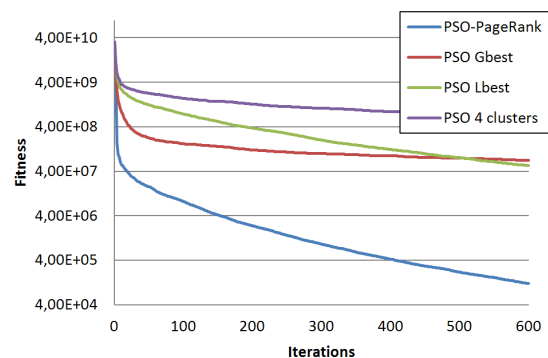


FIGURE B.32 – Shifted Rosenbrock

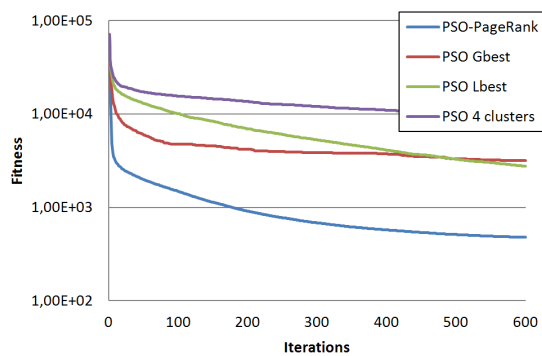


FIGURE B.33 – Shifted Sphere

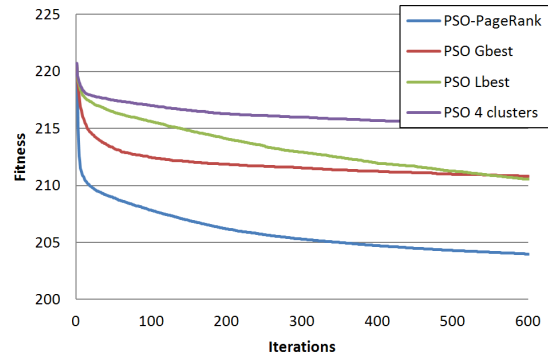


FIGURE B.34 – Shifted Ackley

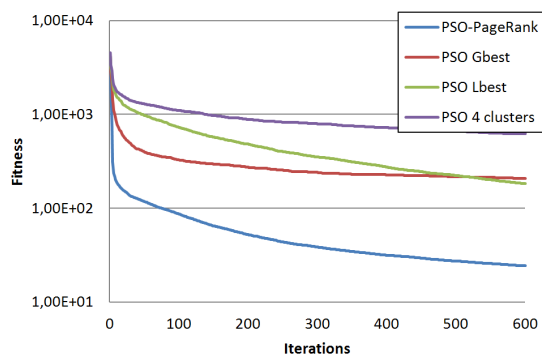


FIGURE B.35 – Bohachevsky

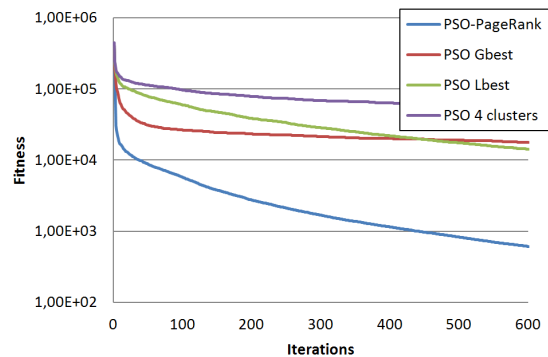


FIGURE B.36 – Schwefel's problem 1.2

Figures B.31 - B.36 : Courbes de convergence de I-PR-PSO en dimension 30 (moyenne des 100 calculs réalisés, pour chaque itération)

RÉSULTATS COMPLÉMENTAIRES : COURBES DE CONVERGENCE DES CALCULS D'OPTIMISATION TOPOLOGIQUE RÉALISÉS AVEC ESO-I-PR-PSO

C.1/ COURBES DE CONVERGENCE DE TOUS LES PROBLÈMES D'OPTIMISATION TRAITÉS

Cette annexe récapitule les courbes de convergence des calculs d'optimisation topologique réalisés avec ESO-I-PR-PSO, sur les 7 cas de structures mécaniques traitées. Ces courbes représentent les évolutions de la fonction objectif et du rapport de volume V/V_0 en fonction des itérations, mises en regard de l'évolution du paramètre rr .

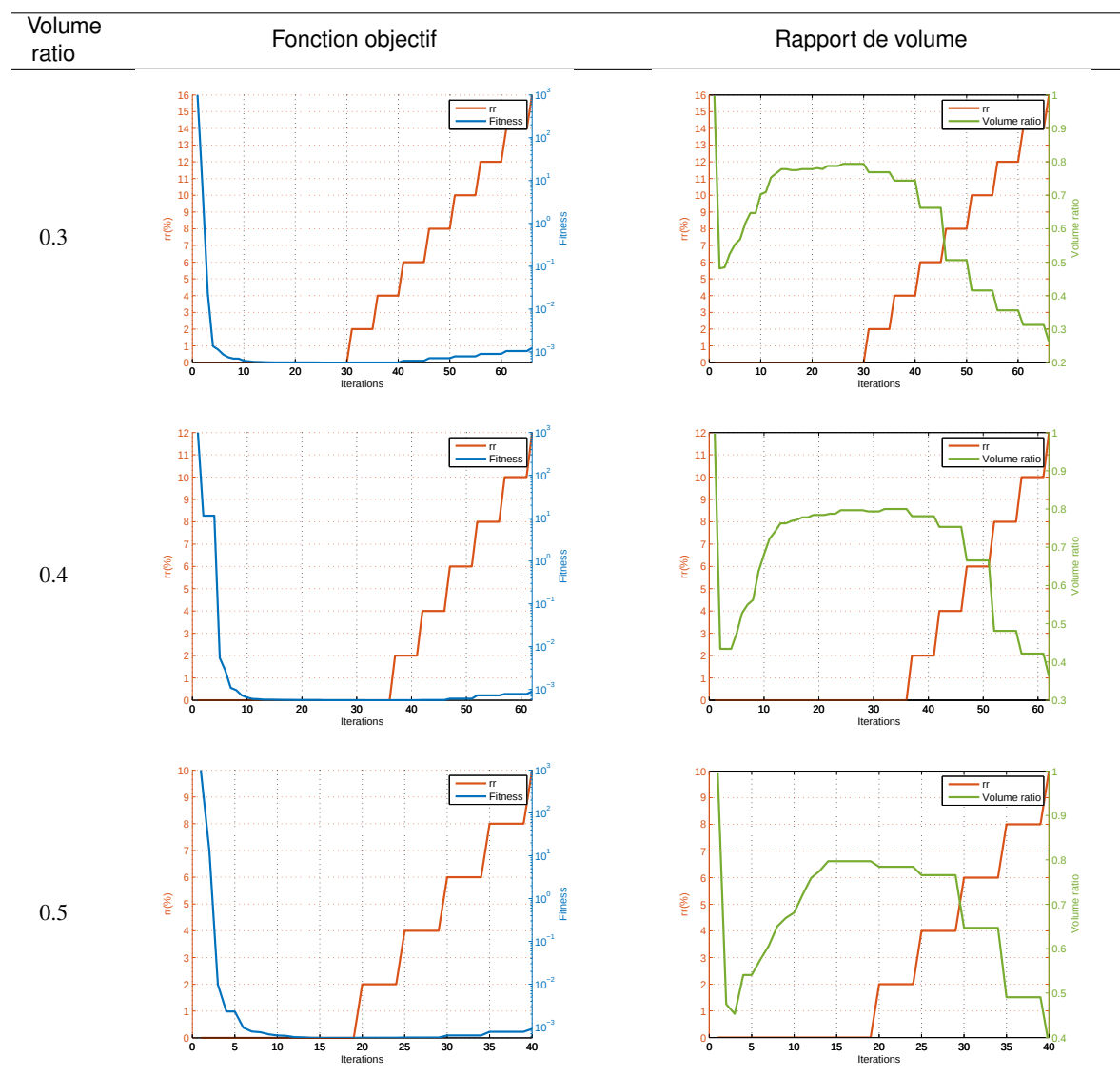


Tableau C.1 – Courbes de convergence de ESO-I-PR-PSO dans le cas 1

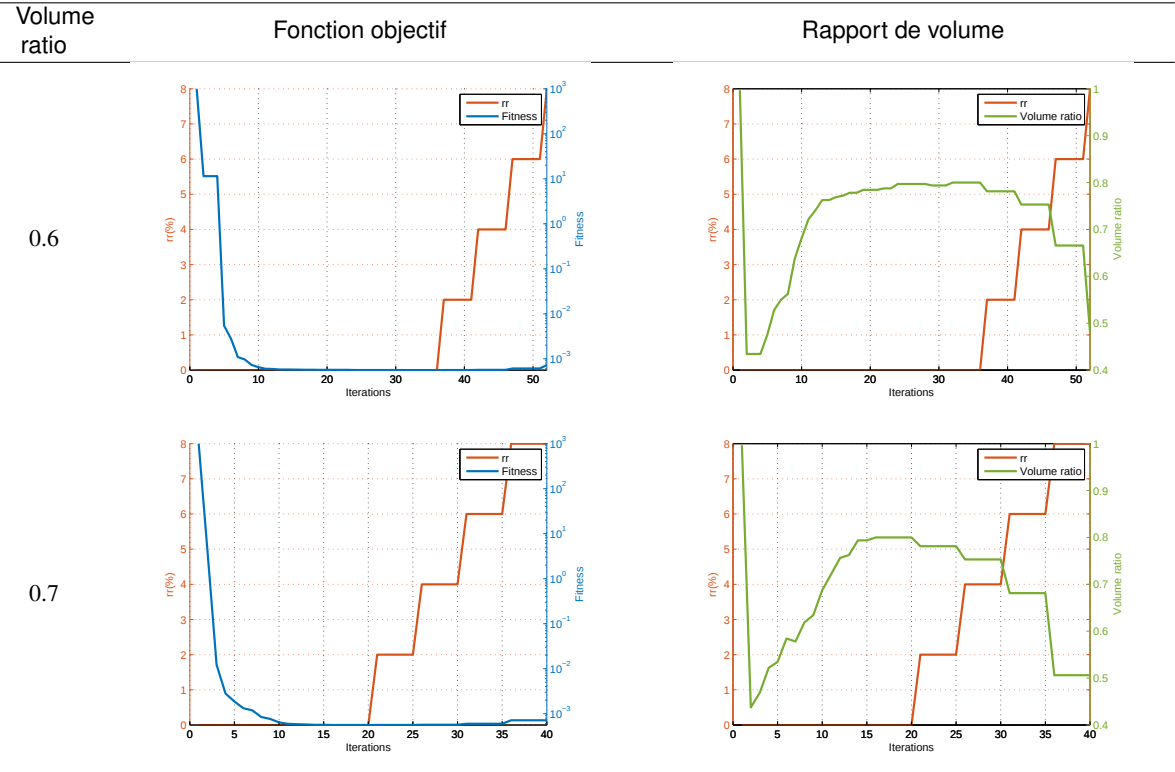


Tableau C.2 – Courbes de convergence de ESO-I-PR-PSO dans le cas 1

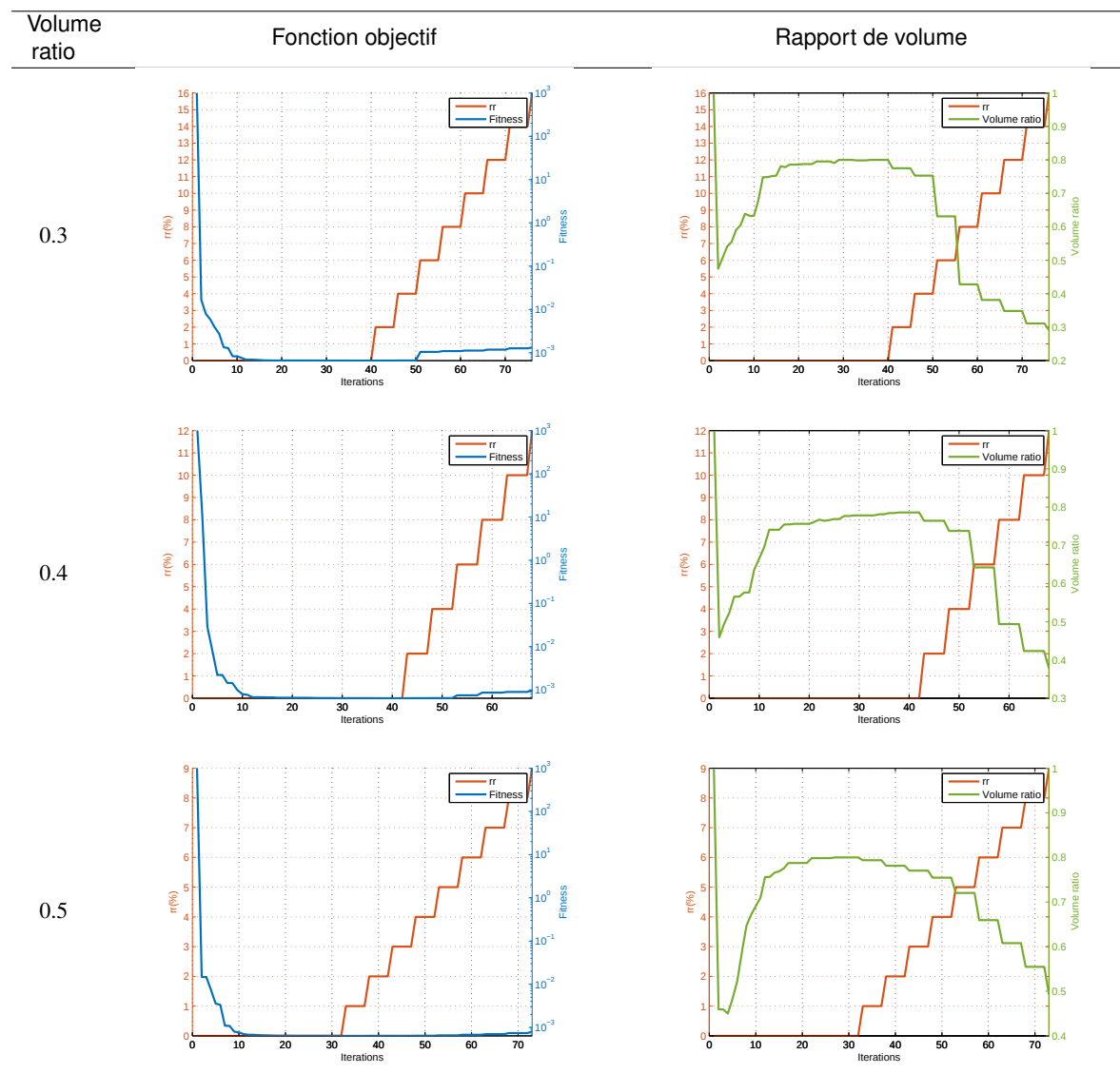


Tableau C.3 – Courbes de convergence de ESO-I-PR-PSO dans le cas 2

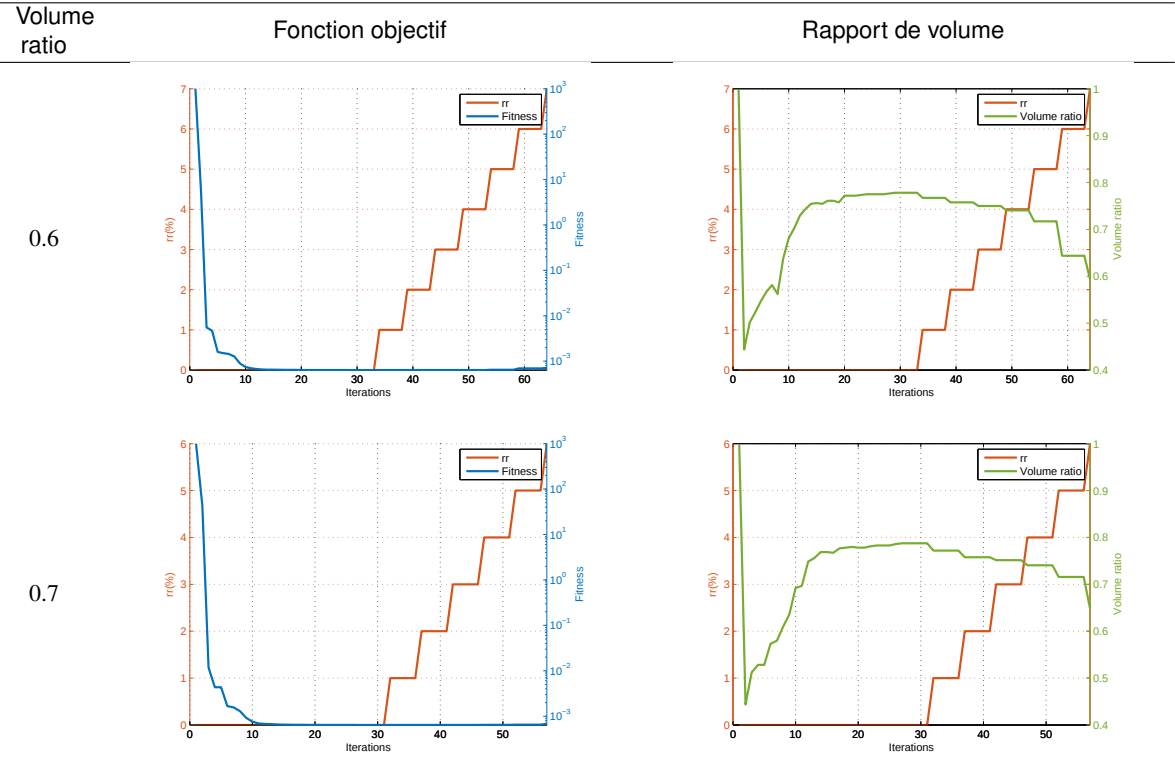


Tableau C.4 – Courbes de convergence de ESO-I-PR-PSO dans le cas 2

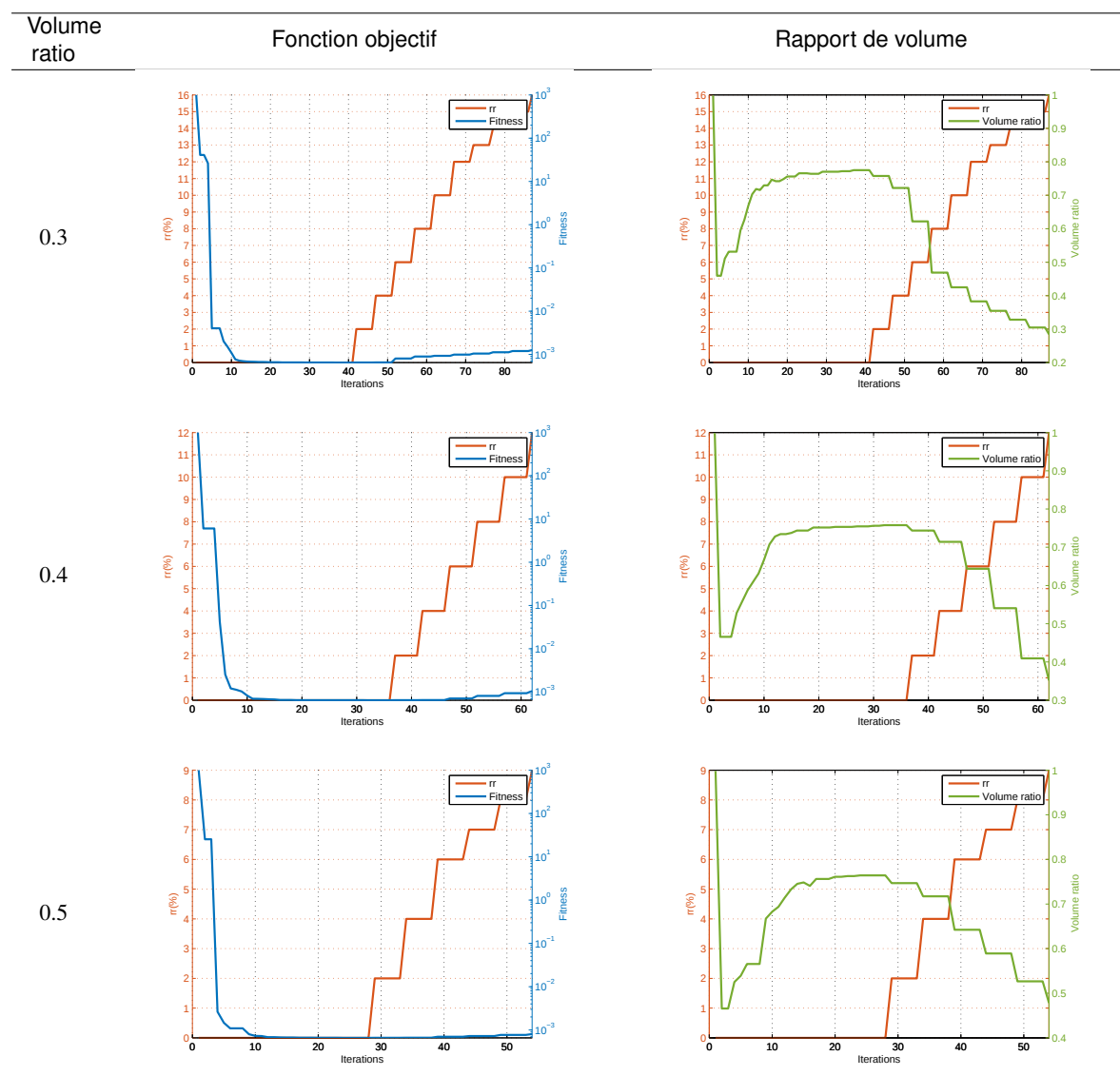


Tableau C.5 – Courbes de convergence de ESO-I-PR-PSO dans le cas 3

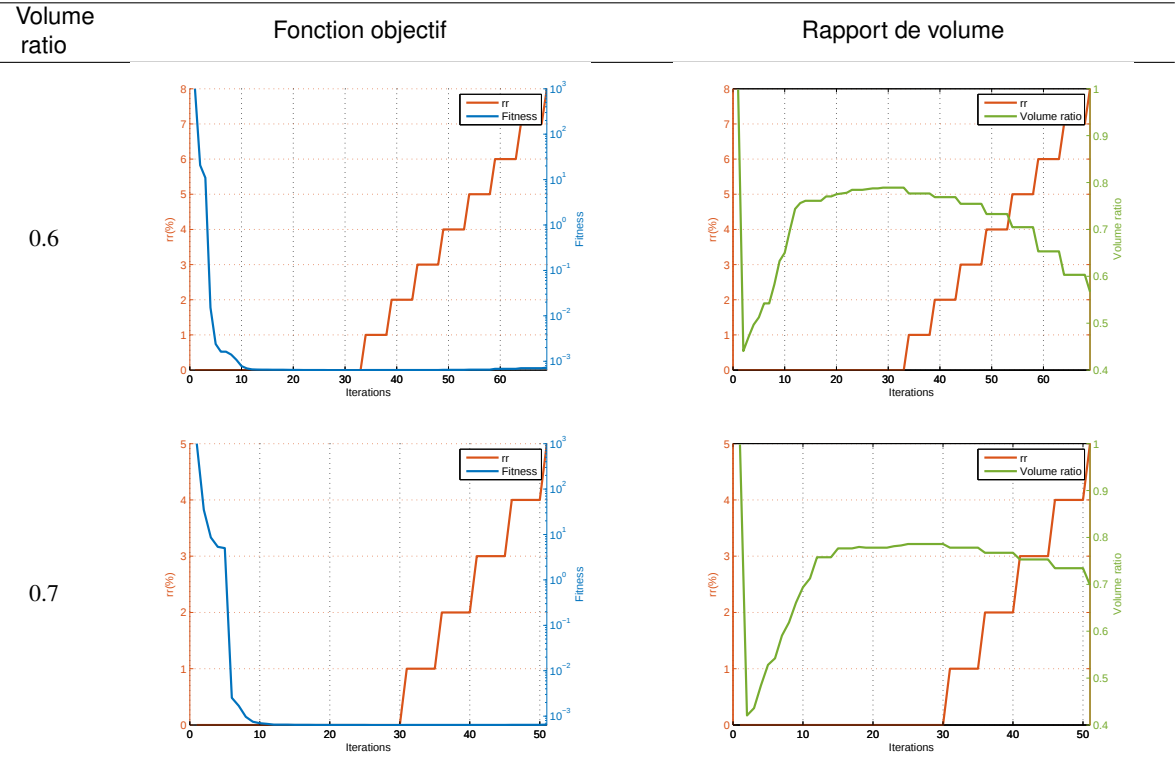


Tableau C.6 – Courbes de convergence de ESO-I-PR-PSO dans le cas 3

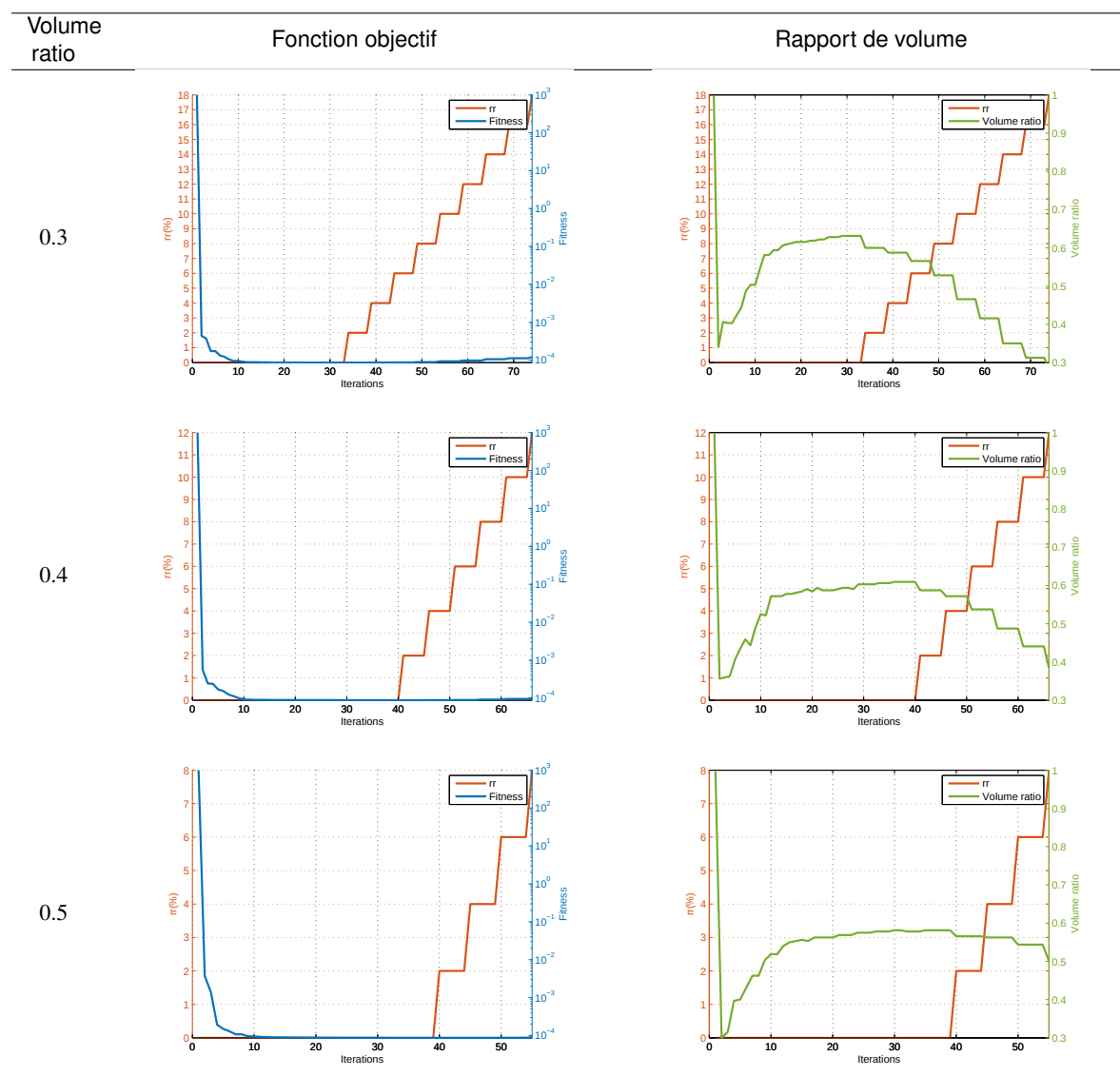


Tableau C.7 – Courbes de convergence de ESO-I-PR-PSO dans le cas 4

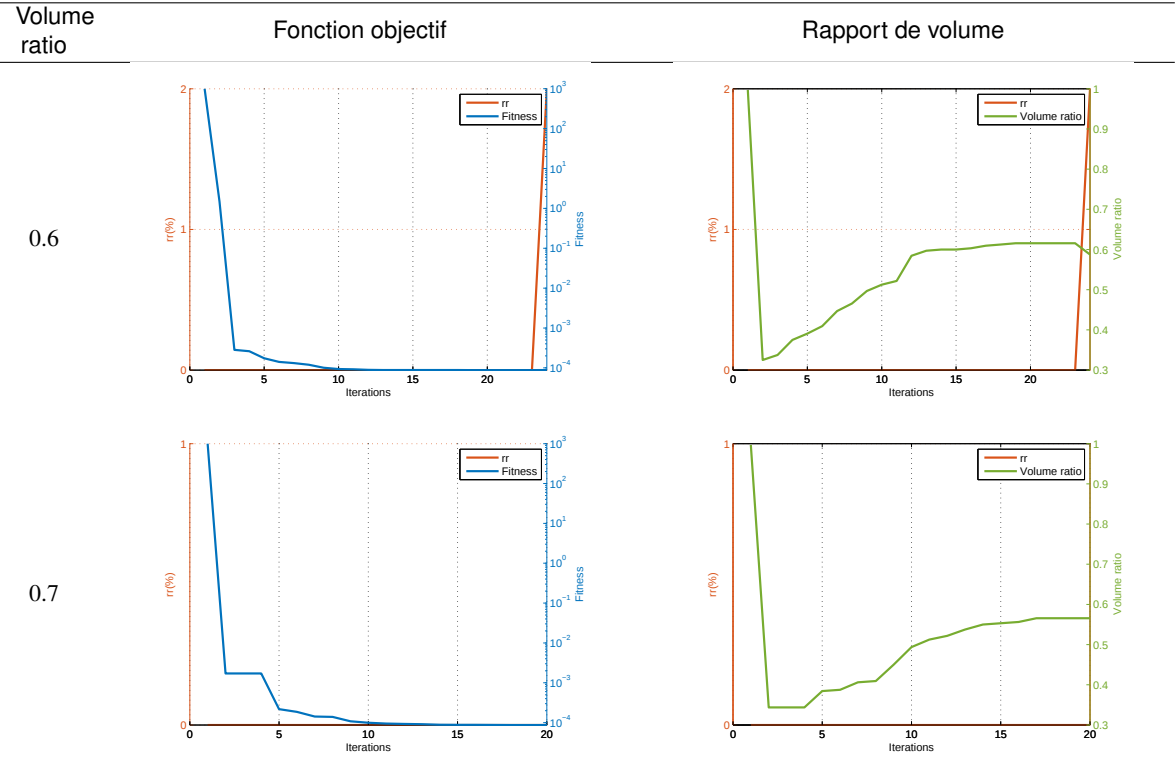


Tableau C.8 – Courbes de convergence de ESO-I-PR-PSO dans le cas 4

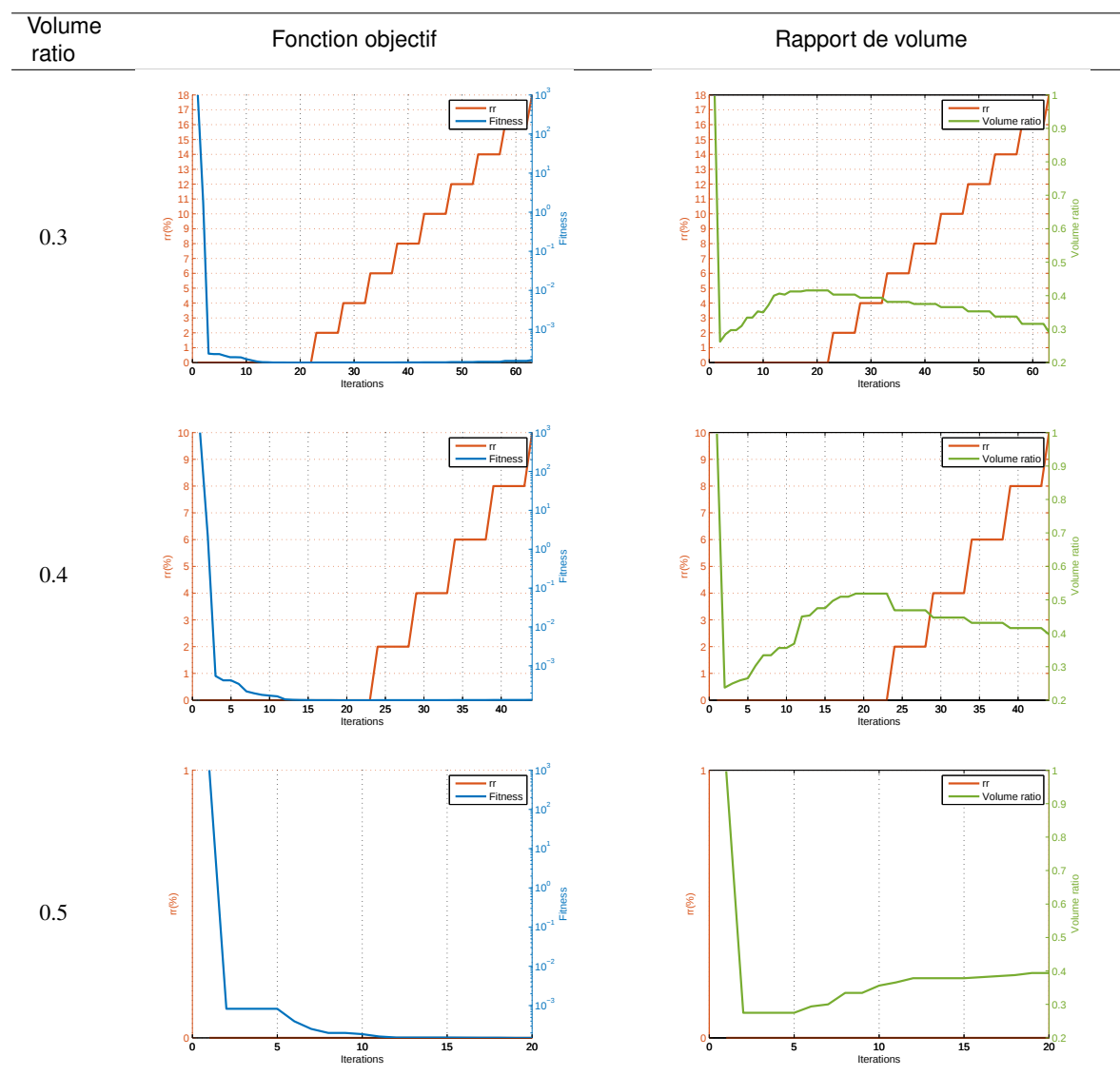


Tableau C.9 – Courbes de convergence de ESO-I-PR-PSO dans le cas 5

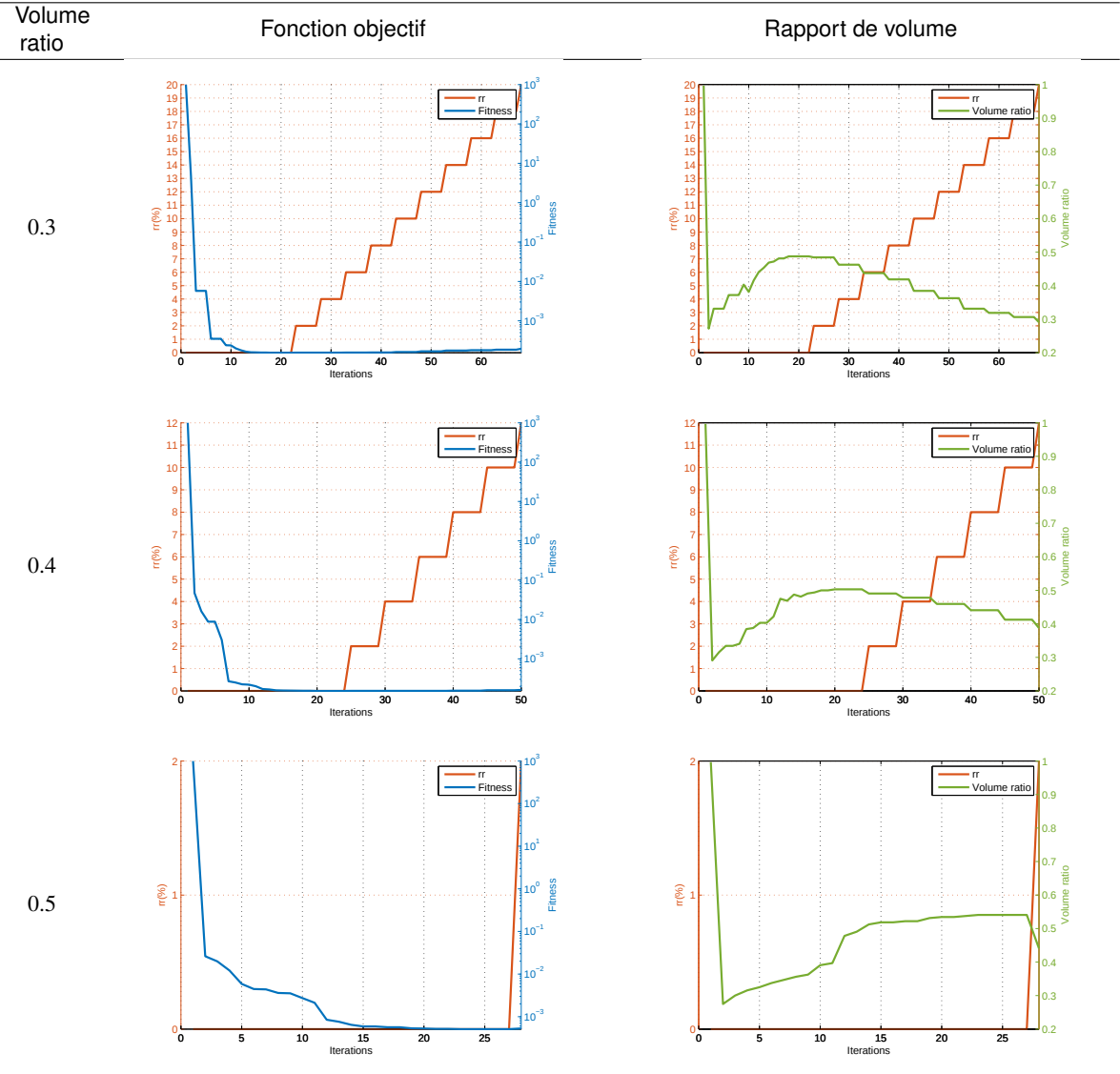


Tableau C.10 – Courbes de convergence de ESO-I-PR-PSO dans le cas 6

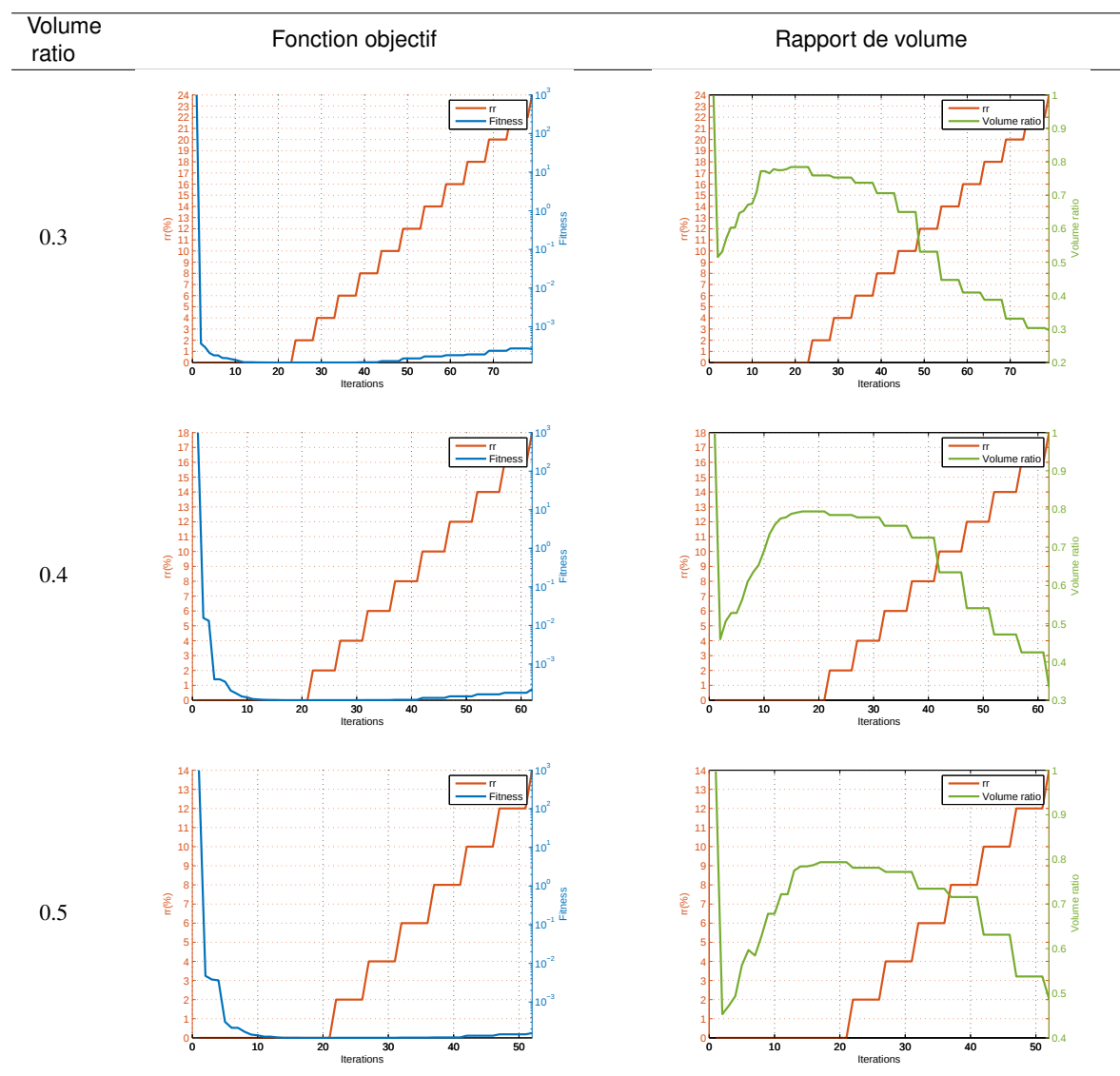


Tableau C.11 – Courbes de convergence de ESO-I-PR-PSO dans le cas 7

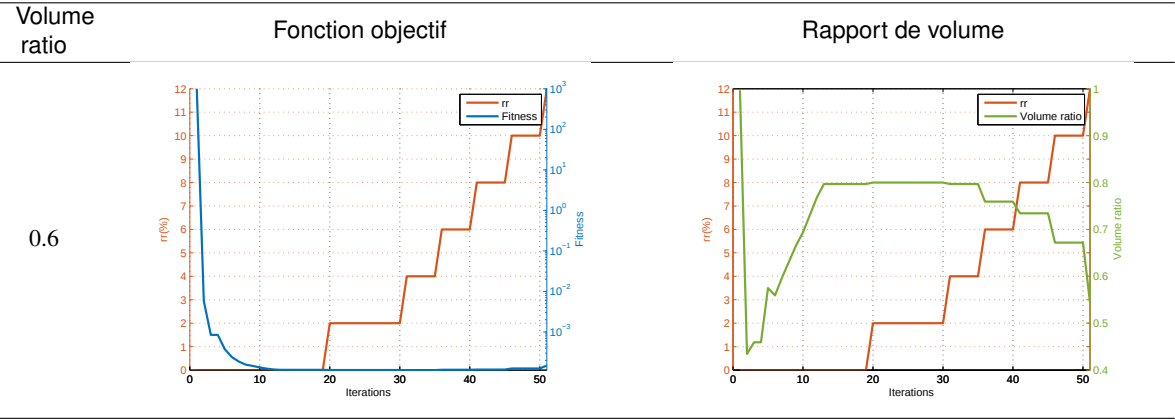


Tableau C.12 – Courbes de convergence de ESO-I-PR-PSO dans le cas 7

C.2/ RÉSULTATS D'OPTIMISATION TOPOLOGIQUE RÉALISÉS AVEC PSO SEUL

Cette section présente quelques résultats de la littérature concernant l'optimisation topologique avec un algorithme par essaim particulaire [208]. En comparaison avec nos résultats, on voit ici que les topologies ne sont pas très régulières, ni symétriques. Ces résultats montrent l'immense difficulté pour obtenir des résultats réguliers avec le seul algorithme PSO, d'autant plus lorsque le nombre de variables d'optimisation est très grand.

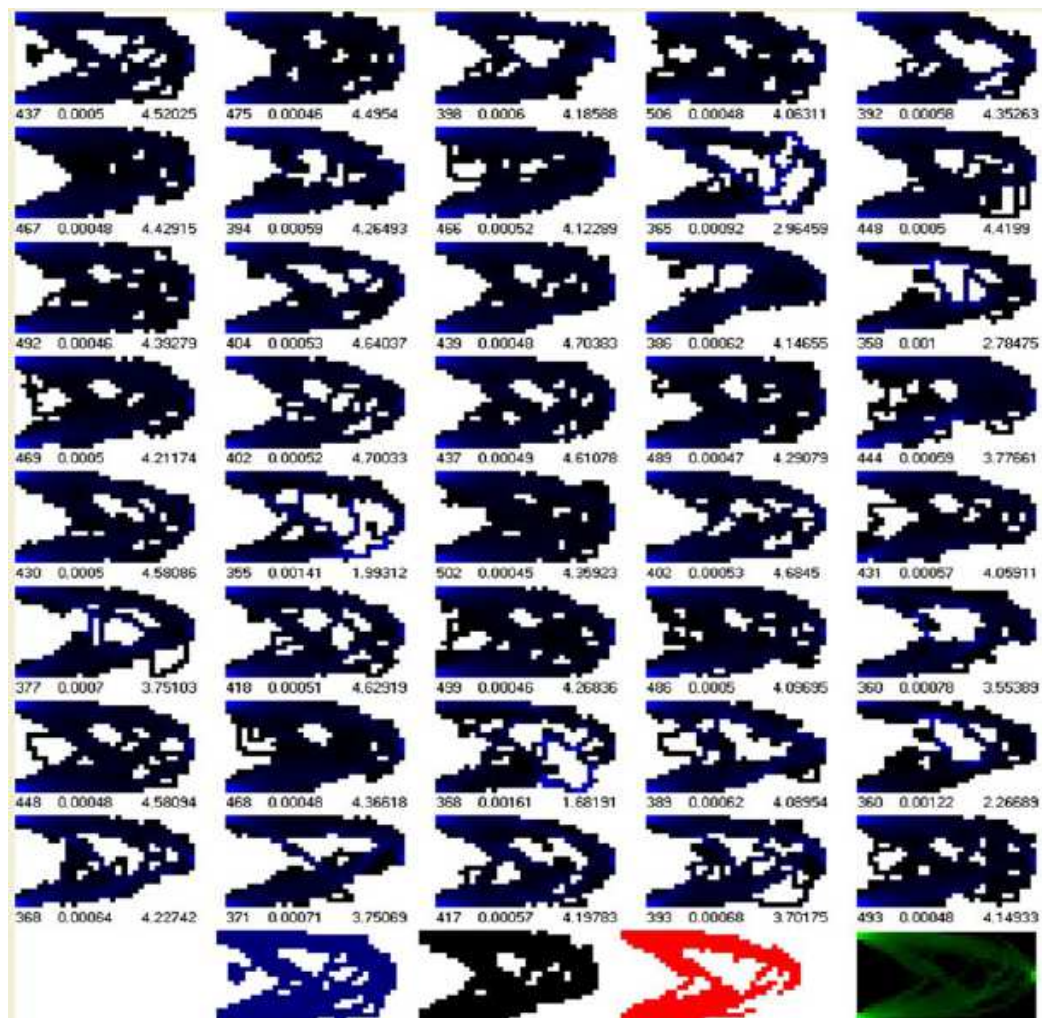


FIGURE C.1 – Résultats de la littérature [208]. Optimisation topologique avec un algorithme d'optimisation par essaim particulaire seul

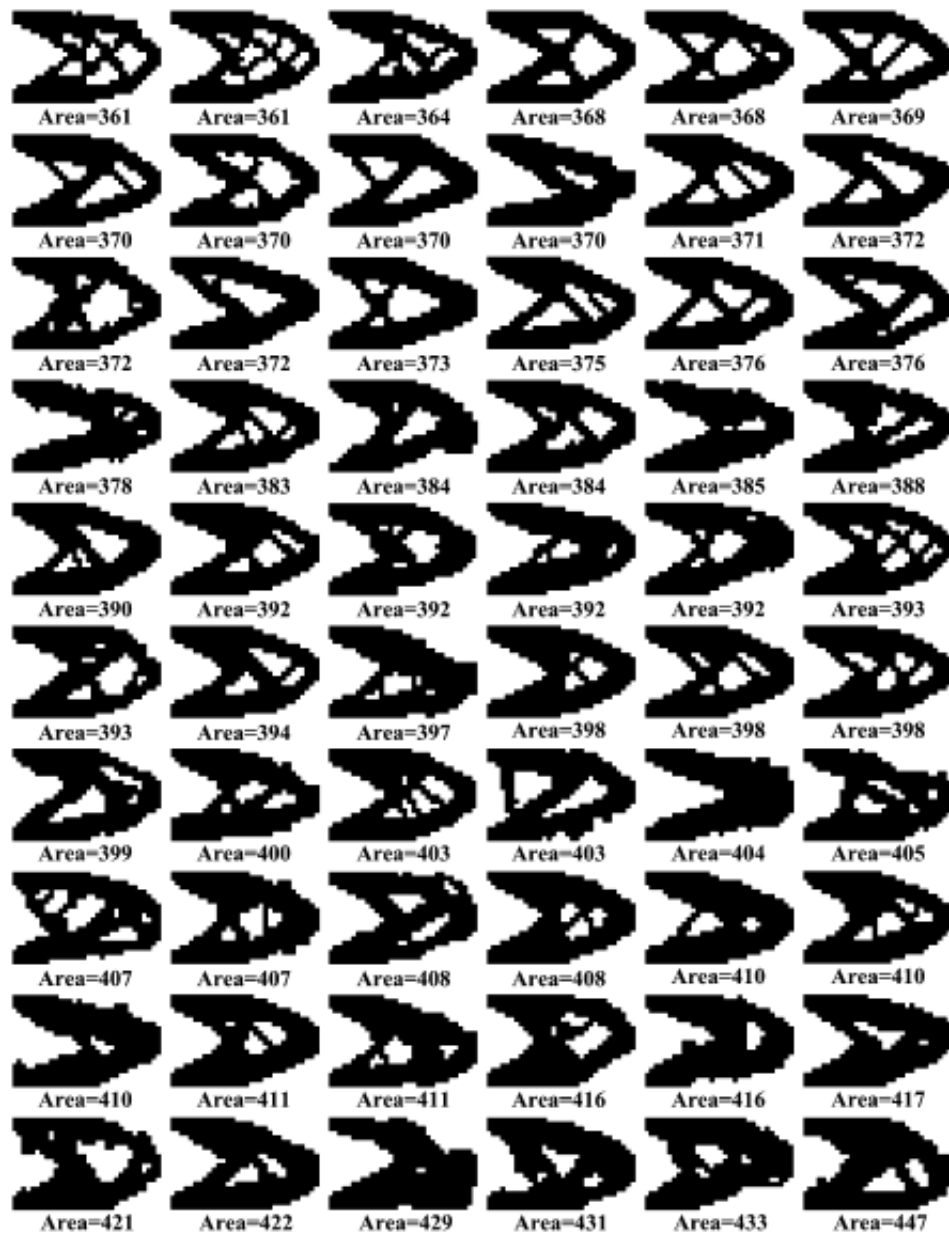


FIGURE C.2 – Résultats de la littérature [208]. Optimisation topologique avec un algorithme d'optimisation par essaim particulaire seul

D

RÉSULTATS COMPLÉMENTAIRES : VALEURS ATTEINTES PAR L'ALGORITHME ESO-I-PR-PSO

Lorsque ESO prend le relais			A la fin du calcul			rr_{max}
V/V_0 demandé	V/V_0	Fonction obj.	V/V_0	Fonction obj.	Itérations	
CAS 1						
0.3	0.76875	5.6673E-4	0.2625	1.2E-3	66	16
0.4	0.78125	5.6179E-4	0.3625	8.9042E-4	62	12
0.5	0.784375	5.6339E-4	0.384375	9.0805E-4	40	10
0.6	0.78125	5.6179E-4	0.48125	7.2853E-4	52	8
0.7	0.78125	5.6237E-4	0.50625	7.1451E-4	40	8
CAS 2						
0.3	0.775	6.5754E-4	0.290625	1.3217E-3	76	16
0.4	0.7640625	6.4217E-4	0.378125	9.6724E-4	68	12
0.5	0.79375	6.3869E-4	0.4921875	8.0132E-4	73	9
0.6	0.7671875	6.3778E-4	0.59375	7.1375E-4	64	7
0.7	0.771875	6.4202E-4	0.6484375	3.8021E-4	57	6
CAS 3						
0.3	0.7578125	6.4977E-4	0.284375	1.2607E-3	87	16
0.4	0.74375	6.4595E-4	0.3515625	1.0473E-3	62	12
0.5	0.746875	6.5924E-4	0.478125	8.0613E-4	54	9
0.6	0.7765625	6.3844E-4	0.5671875	7.2207E-4	69	8
0.7	0.778125	6.3982E-4	0.6984375	6.543E-4	51	5
CAS 4						
0.3	0.6	8.59E-5	0.28125	1.2002E-4	74	18
0.4	0.5875	8.81E-5	0.384375	1.0032E-45	66	12
0.5	0.565625	8.70E-5	0.5	8.99E-5	55	8
0.6	XXX	XXX	0.590625	8.76E-5	20	0
0.7	XXX	XXX	0.565625	8.74E-5	20	0
CAS 5						
0.3	0.403125	1.4202E-4	0.290625	1.6544E-4	63	18
0.4	0.46875	1.3089E-4	0.396875	1.3354E-4	44	10
0.5	XXX	XXX	0.39375	1.5027E-4	20	0
CAS 6						
0.3	0.484375	1.5515E-4	0.290625	1.9547E-4	68	20
0.4	0.490625	1.5315E-4	0.3875	1.6242E-4	50	12
0.5	0.440625	5.2919E-4	0.440625	5.2919E-4	28	2
CAS 7						
0.3	0.759375	1.1869E-4	0.296875	2.6534E-4	79	24
0.4	0.784375	1.1578E-4	0.334375	2.2363E-4	62	18
0.5	0.78125	1.1797E-4	0.4875	1.5627E-4	52	14
0.6	0.8	1.1537E-4	0.54375	1.488E-4	51	12

Tableau D.1 – Valeurs atteintes par les deux étapes de notre algorithme d'optimisation ESO-I-PR-PSO

EXEMPLE DE MODÈLE NUMÉRIQUE DE VER DÉFINI AVEC CAST3M

*-----OPTIONS GENERALES DU MODELE ;

OPTI ECHO 0 ;
OPTI DIME 2 ELEM QUA4 MODE PLAN CONT EPSI JAUMANN ;
OPTI ERRE FATAL ;

*CREATION DE LA GEOMETRIE ;

P1 = 0. 0. ;
P2 = 0.120 0. ;
P3 = 0.120 0.120 ;
P4 = 0. 0.120 ;
L1 = DROI 10 P1 P2 ;
L2 = DROI 10 P2 P3 ;
L3 = DROI 10 P3 P4 ;
L4 = DROI 10 P4 P1 ;
GEO1 = DALL L1 L2 L3 L4 ;

*CHOIX DU COMPORTEMENT ET DES MATERIAUX ;

MOD1 = MODE GEO1 MECANIQUE ELASTIQUE ISOTROPE QUA4 ;
MAT1 = MATE MOD1 YOUN 2100000000000 NU 0.3 DIM3 0.001 ;

*Particule 11 ;

*CREATION DU MCHAML ;

OPTI ACQU 'topologie11.txt' ; *(où topologie11.txt est un fichier texte contenant la matrice binaire définissant la répartition de matière dans le domaine)
ACQU TOPO*LISTREEL 100 ;

*CREATION DU MCHAML ;

CH1 = MANU 'CHML' MOD1 'REPA' 'TEST' TOPO ;

*RESTRICTION DU MODELE ;

MODI1 = CH1 ELEM 'SUPERIEUR' 0.5 ;

MODA1 = REDU MOD1 MODI1 ;
 MATA1 = REDU MAT1 MODI1 ;
 MOD = MODA1 ;
 MAT = MATA1 ;

***CONDITIONS AUX LIMITES ;**

CL1 = BLOQUER DEPLA ROTA P1 ;
 CL = CL1 ;

***CONDITIONS AUX LIMITES PERIODIQUES ;**

RIG1 = 'RELA' 'UX' L2 - 'UX' (INVE L4) ;
 RIG2 = 'RELA' 'UY' L2 - 'UY' (INVE L4) ;
 RIG3 = 'RELA' 'UX' L3 - 'UX' (INVE L1) ;
 RIG4 = 'RELA' 'UY' L3 - 'UY' (INVE L1) ;

***Cas calcul 1 ;**

*** DEPLACEMENT IMPOSE _____ ;**

CHAR1 = DEPI RIG1 -0.0030567 ; **(On impose ici la valeur de la déformation voulue)*
 CHAR2 = DEPI RIG2 0. ;
 CHAR3 = DEPI RIG3 0. ;
 CHAR4 = DEPI RIG4 0. ;
 CHARTOT = CHAR1 ET CHAR2 ET CHAR3 ET CHAR4 ;
 LIS1 = PROG 0. 1. ;
 LIS2 = PROG 0. 1. ;
 EVT1 = EVOL MANU 'T' LIS1 'D(T)' LIS2 ;
 CHA1 = CHAR 'DIMP' CHARTOT EVT1 ;

*** MODELE NON LINEAIRE _____ ;**

LIS_TPS = PROG 0. PAS 0.01 1. ;
 TPS_SAUV = PROG 1. ;
 TAB1 = TABLE ;
 TAB1.'PROCESSEURS' = MONO_PROCESSEUR ;
 TAB1.'CONVERGENCE_FORCEE' = FAUX ;
 TAB1.'GRANDS_DEPLACEMENTS' = VRAI ;
 TAB1.'MODELE' = MOD ;
 TAB1.'CARACTERISTIQUES' = MAT ;
 TAB1.'CHARGEMENT'=CHA1 ;
 TAB1.'PREDICTEUR'=HPP ;
 TAB1.'BLOCAGES_MECAIQUES'=CL ET RIG1 ET RIG2 ET RIG3 ET RIG4 ;
 TAB1.'TEMPS_CALCULES'=LIS_TPS ;
 TAB1.'TEMPS_SAUVES'=TPS_SAUV ;
 TAB1.'PAS_AJUSTE' = VRAI ;
 PASAPAS TAB1 ;

***RECUPERATION DES RESULTATS ;**

TAB2 = TAB1.CONTRAINTES ;

```
MCHAML1 = TAB2.100;  
INT_CONT = INTG MOD MCHAML1 'SMXX'; * Intégration numérique des contraintes
```


Résumé :

L'optimisation offre la possibilité, dans de nombreux domaines, d'améliorer les performances d'un système donné, qu'il soit physique ou mathématique. Depuis quelques décennies, les méthodes d'optimisation metaheuristiques ont fait leurs preuves, notamment dans le domaine de la mécanique. Du grec *meta* signifiant "un niveau au dessus", les metaheuristiques permettent de s'affranchir du calcul des sensibilités souvent problématique quant à la résolution de problèmes d'optimisation complexes et/ou NP difficiles. En outre, elles ont la capacité à analyser simultanément l'ensemble du domaine des solutions, ce qui leur permet converger efficacement vers l'optimum global de la fonction objectif considérée. Notre travail propose le développement d'une nouvelle méthode metaheuristique intelligente, basée conjointement sur l'algorithme d'optimisation par essaim particulaire PSO, et l'algorithme PageRank développé par MM. Brin et Page, et utilisé par le moteur de recherche Google. Cet algorithme, appelé Inverse-PageRank-PSO (I-PR-PSO), a été validé sur un benchmark de fonctions mathématiques, puis en optimisation contrainte sur des treillis mécaniques. Interfacée avec l'algorithme Evolutionary Structural Optimization (ESO), elle a été adaptée à l'optimisation topologique et a permis de trouver des résultats dont les topologies sont régulières et les temps de calcul minimisés. Dans le domaine des metamatériaux, nous avons développé une cape d'invisibilité électromagnétique fréquentielle, c'est à dire un metamatériau dont les parties réelle et imaginaire de la perméabilité effective sont négatives. En appliquant notre algorithme I-PR-PSO aux metamatériaux mécaniques, nous avons montré qu'il est possible de développer un metamatériau constitué d'acier qui présente des grandes déformations à l'échelle macroscopique, dûes notamment aux grands déplacements présents dans le Volume Élémentaire Représentatif à l'échelle microscopique.

Mots-clés : Optimisation, Metaheuristiques, Mécanique, Metamatériaux

Abstract:

Based on a recent research concerning the PageRank algorithm used by the famous search engine Google, a new Inverse-PageRank-Particle Swarm Optimizer (I-PR-PSO) is developed, in order to improve the performances of classic PSO. After having been tested and validated on a benchmark of classical mathematical functions, this algorithm has been validated on constrained optimization, applied on classical trusses of the literature. Interfaced with the Evolutionary Structural Optimization algorithm, this algorithm has shown its performances on topology optimization, applied to structural mechanics. Finally, using the performances of our newly developed algorithm, we have developed metamaterials. In electromagnetics, a frequential cloaking device has been developed, minimizing the effective permeability of the considered Representative Volume Element. In mechanics, we have developed a metamaterial made of steel which exhibits hyper-elastic - or, at least, non linear - mechanical behaviour. Combining great displacements and rotations at microscale, the developed metamaterial exhibits great deformations at the macroscale as well.

Keywords: Optimization, Metaheuristics, Mechanics, Metamaterials



■ École doctorale SPIM - Université de Technologie Belfort-Montbéliard

F - 90010 Belfort Cedex ■ tél. +33 [0]3 84 58 31 39

■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

