



HAL
open science

Contributions à l'optimisation en conception

Dominique Chamoret

► **To cite this version:**

Dominique Chamoret. Contributions à l'optimisation en conception. Sciences de l'ingénieur [physics].
Université de Bourgogne-Franche Comté (UBFC), 2022. tel-04284956

HAL Id: tel-04284956

<https://hal.science/tel-04284956>

Submitted on 15 Nov 2023

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.

Copyright



HABILITATION À DIRIGER DES RECHERCHES
de l'Université de Bourgogne Franche-Comté
préparée au sein de l'Université de Technologie de Belfort-Montbéliard
Spécialité : **Mécanique**

présentée par

DOMINIQUE CHAMORET

Contributions à l'optimisation en conception

Soutenue publiquement le 14 décembre 2022 devant le jury composé de :

JEAN BIGEON	Rapporteur	Directeur de recherche, UMR 6004, Laboratoire des Sciences du Numérique de Nantes
LHASSANE IDOUMGHAR	Rapporteur	Professeur des Universités, Institut IRIMAS, Université de Haute Alsace
PIERRE DUYSINX	Rapporteur	Professeur des Universités, Aerospace and Mechanical Engineering Departement, Université de Liège
SÉBASTIEN ROTH	Examineur	Professeur des Universités, Laboratoire Interdisciplinaire Carnot de Bourgogne, site UTBM, UMR CNRS 6303/Univ. Bourgogne Franche-Comte
SAID ABOUDI	Examineur	Professeur des Universités, Laboratoire Interdisciplinaire Carnot de Bourgogne, site UTBM, UMR CNRS 6303/Univ. Bourgogne Franche-Comte
DOMINIQUE BARCHIESI	Examineur	Professeur des Universités, Gamma3, Université de technologie de Troyes, 12 rue Marie Curie - CS 42060 10004 TROYES CEDEX
GAEL CHEVALLIER	Examineur	Professeur des Universités, FEMTO ST, Département Mécanique Appliquée équipe D.SMART, UFC - UFR

REMERCIEMENTS

- ✕ Aux membres du jury.
- ✕ Aux doctorants qui travaillent ou qui ont travaillé avec moi.

- ✕ Aux amis.
- ✕ A mes enfants, Adrien et Marie.
- ✕ A mon mari, Patrick.

Tu n'es plus là où tu étais, mais tu es partout là où je suis.
Victor Hugo

SOMMAIRE

1	Introduction	xi
I	Notice	1
2	Curriculum Vitæ	3
2.1	Situation actuelle	3
2.2	Parcours et formation	4
II	Synthèse scientifique	5
3	Optimisation	7
3.1	Contexte	7
3.2	Processus d'optimisation	7
3.3	Formulation mathématique d'un problème d'optimisation	8
3.4	Formulation d'un problème d'optimisation en conception	10
3.5	Conclusions et perspectives	15
4	Focus sur PSO	19
4.1	Introduction	19
4.2	Métaheuristiques à base de population	21
4.3	PSO	23
4.4	Prise en compte des contraintes	28
4.5	Un nouveau critère d'arrêt et saut d'essaim	30
4.6	I-PR-PSO	35
4.7	Conclusions et perspectives	42
5	Benchmark	43
5.1	Contexte et objectifs	45
5.2	Structure et contrôle du benchmark	47
5.3	Outils de comparaison et indicateurs de performances	51
5.4	Application et résultats	55

5.5	Conclusion	59
6	Méta-optimisation	61
6.1	Contexte et objectifs	61
6.2	Étude globale	62
6.3	Études locales	66
6.4	Conclusions	72
7	Optimisation topologique et conception	75
7.1	Introduction	75
7.2	Structures innovantes	76
7.3	Optimisation topologique et fabrication additive	82
7.4	Conclusion	84
8	Conclusions et perspectives	85
8.1	Formulation et certification des algorithmes d'optimisation	85
8.2	Structures intelligentes et 4D printing	85
8.3	Simulation et optimisation des systèmes pile à combustible	87

TABLE DES FIGURES

1.1	Démarche suivie pour le projet de recherche et organisation du manuscrit	xii
3.1	Processus d'optimisation : une alternance d'expertise	8
3.2	Formulation d'un problème d'optimisation	9
3.3	Classification d'un problème d'optimisation	10
3.4	Catégorisation	12
3.7	Instaurer un dialogue pour la formulation	13
3.8	Séquencement du dialogue	13
3.9	Exemple de cartes illustrées pour dialogue	14
3.10	État de l'interview à la fin de l'étape d'acquisition des données	14
3.5	Catégorisation des caractéristiques du produit	16
3.6	Catégorisation des caractéristiques du produit	17
4.1	Une classification des méthodes métaheuristiques	21
4.2	Diversification	22
4.3	Intensification	23
4.4	Mode de déplacement des particules dans l'espace des solutions	25
4.5	Deux topologies célèbres	28
4.6	Un nouveau critère d'arrêt - PSO-Radius	31
4.7	Saut d'essaim	31
4.8	Problématique : contrainte résiduelle thermique pour les CMC	32
4.9	Modélisation EF des CMC	33
4.10	Évolution de la CRT : comparaison PSO standard/Radius	34
4.11	Minimum après 600 itérations et $D = 50$ sur l'ensemble des fonctions tests	39
4.12	Graphe de convergence de l'algorithme I-PR-PSO en dimension 50	41
4.13	Validation sur des treillis	42
5.1	Évolution du nombre de publications [31]	45
5.2	Constitution du benchmark	49
5.3	Critères d'arrêts	50
5.4	Traitement du résultat brut d'un run	52

5.5	Résultat d'un cas d'optimisation	53
5.6	Démarche générale pour le calcul du score principal	54
5.7	Score global pour les différents algorithmes	56
5.8	Score par dimension pour les différents algorithmes	57
5.9	Score par fonction pour les différents algorithmes	58
5.10	Score par <i>MaxFes</i> pour les différents algorithmes	58
5.11	Comparatif Benchmark proposé / Littérature	59
6.1	Notre stratégie de méta-optimisation	63
6.2	Plan d'expériences pour l'étude globale	64
6.3	Effets sur le rapport signal/bruit des résultats globaux	65
6.4	Ensemble des études locales	68
6.5	ACP Cas 1 : dimension fixée	69
6.6	ACP Cas 2 : fonction fixée	69
6.7	ACP Cas 3 : <i>MaxFes</i> fixé	70
6.8	Validation par sous-ensembles	72
6.9	Validation par score global	72
7.1	Optimisation topologique	76
7.2	Démarche de la procédure SE - VER - optimisation topologique	77
7.3	Structure cellulaire carrée <i>2D</i> optimisée topologiquement	78
7.4	Modèle numérique d'un métamatériau électromagnétique	79
7.5	Topologie binaire pour les métamatériaux électromagnétiques	80
7.6	Couplage PSO binaire et méthode des éléments finis	81
7.7	Métamatériaux : Topologie obtenue après optimisation	81
7.8	Pièces avec différents angles d'inclinaison et besoins de supports	82
7.9	Code ELiOT : implémentation d'un algorithme d'optimisation topologique	83
7.10	Code ELiOT : implémentation d'un algorithme d'optimisation topologique avec prise en compte de la contrainte d'overhang	84
8.1	Projet 4D printing et optimisation topologique	86

LISTE DES TABLEAUX

4.1	Paramètres algorithmiques du PSO	26
4.2	Paramètres d'adaptation du PSO	27
4.3	Comparaison PSO standard / PSO amélioré	34
4.4	Fonctions tests retenues pour la validation de l'algorithme I-PR-PSO	37
4.5	Paramètres de calculs pour la validation de I-PR-PSO	38
4.6	Résultats de l'optimisation de treillis	42
5.1	Notation du chapitre 5	44
5.2	Ensemble des fonctions objectifs du benchmark	47
5.3	Paramètres de contrôle du benchmark	48
5.4	Phase de calculs : run et séries	49
5.5	Présentation des algorithmes	55
6.1	Facteurs essentiels du plan d'expériences	63
6.2	Facteurs externes	64
6.3	Jeu de paramètres robustes	66
6.4	Paramètres robustes : comparaison test de validation et modèle analytique	66
6.5	Paramètres optimaux par sous-ensembles	71
6.6	Jeux de paramètres du PSO retenus pour validation de notre approche	71

INTRODUCTION

Ce mémoire d’habilitation à diriger des recherches présente une synthèse de mes activités de recherche conduites depuis 2004, date de ma nomination comme maître de conférences à l’université de technologie de Belfort (UTBM). Dans ce document, il s’agit plus de se focaliser sur la partie recherche. Toutefois, pour un enseignant-chercheur, il est difficile de déconnecter complètement son activité de chercheur de celle d’enseignant. Les compétences développées ne sont pas exclusives d’une activité en particulier. Ces dernières décennies les nouvelles technologies et approches numériques ont fortement impacté l’univers de la mécanique numérique et de l’optimisation bougeant les lignes au niveau pédagogique. C’est pourquoi, il semble fondamental de présenter également les activités pédagogiques et collectives. Le bilan de ces années en tant que MCF est révélateur à cet égard de mes motivations personnelles car la durée est suffisamment longue pour faire ressortir les grandes constantes d’un individu.

Organisation du mémoire

C’est pourquoi ce mémoire est divisé en deux parties.

- ❖ L’objectif de la première partie est de faire ressortir l’équilibre que j’ai pu mettre en place entre les quatre missions de l’enseignant-chercheur en termes de recherche, d’enseignement, de responsabilités administratives et d’actions de transfert et de valorisation. Elle est composée de trois chapitres : après une présentation de mon parcours (Chapitre 2). Vient ensuite un chapitre dédié à mes activités scientifiques (Chapitre ??). Un chapitre axé sur mes activités pédagogiques termine cette partie (Chapitre ??).
- ❖ La deuxième partie brosse un tableau plus détaillé de mes activités de recherche. Elle est décomposée en quatre chapitres. Le plan de cette partie suit exactement le projet de recherche décrit sur la figure 1.1 (page xii) et, succinctement dans ce qui suit.

Contexte scientifique et projet de recherche

J’ai été recrutée en tant que maître de conférences en 2004 au laboratoire M3M, équipe MOS (Modélisation et optimisation de structures), département d’enseignement Génie Mécanique et Conception (GMC) de l’Université de Technologie de Belfort-Montbéliard (UTBM). Un des axes principaux de cette équipe était l’optimisation en mécanique

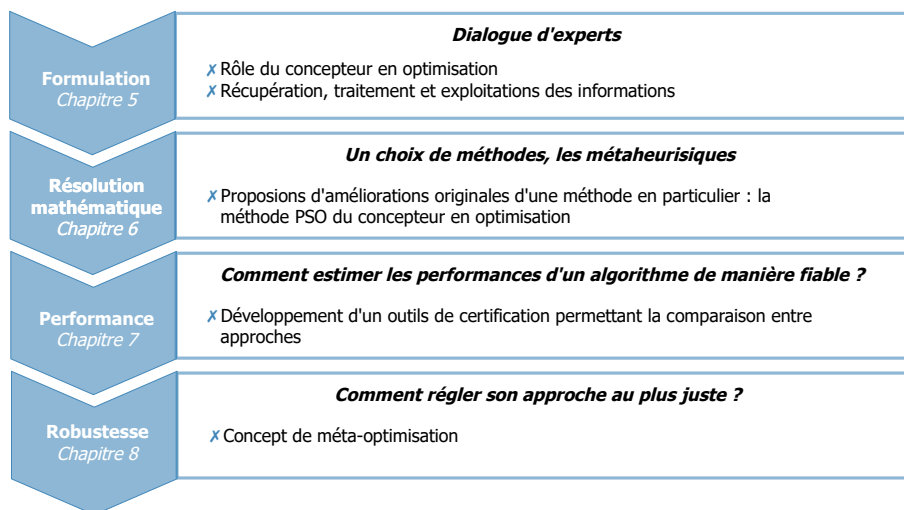


FIGURE 1.1 – Démarche suivie pour le projet de recherche et organisation du manuscrit

des structures. Ayant travaillé dans le domaine de la modélisation du contact (Docteurat et post-doctorat), l'optimisation ne m'était pas un domaine inconnu. Mes connaissances/compétences en mathématiques et mes travaux antérieurs m'ont permis d'appréhender relativement rapidement ce domaine de recherche. Il résulte de mon parcours que l'axe central qui constitue la zone d'ancrage commune à l'ensemble de mes activités traite la modélisation/simulation et l'optimisation de systèmes mécaniques non-linéaires. En effet, la formulation même d'un problème d'optimisation nécessite un modèle permettant une analyse (évaluation) du comportement du système (et/ou structure).

Le contexte UT et l'existence d'une équipe axée conception au sein du laboratoire m'ont conduit à m'intéresser en particulier à l'optimisation en conception. Depuis début 2017, date d'intégration de ma précédente équipe au laboratoire ICB (Laboratoire Interdisciplinaire Carnot de Bourgogne), Unité Mixte de Recherche du CNRS (UMR 6303), département CO2M (Conception, Optimisation et Modélisation en Mécanique), un axe fort de recherche a émergé notamment au sein du département CO2M : la fabrication additive (FA) et le 4D printing. Dans ce contexte, il m'est apparu judicieux de mettre à profit mes compétences en optimisation en particulier ce qui concerne l'optimisation topologique associée à la fabrication additive de manière connexe à ce qui était en cours.

Contexte

Dans un contexte industriel extrêmement concurrentiel, avec des cahiers des charges de plus en plus exigeants, l'approche systémique permet d'envisager des solutions originales et jusqu'alors inexplorées. Cette méthode allie de manière intime simulations, expérimentations, interactions entre sous-systèmes et cycles de conception à tous les niveaux afin d'obtenir un produit plus performant, mieux intégré et de réduire les délais de mise sur marché.

Projet de recherche : l'optimisation en conception

Ce constat a servi de socle au développement de mon projet de recherche dans le cadre de l'optimisation. La direction principale de mes travaux de recherche en optimisation est de développer un/des outils performants pour accompagner au mieux l'ingénieur dans sa démarche d'optimisation en tenant compte des contraintes de l'univers industriel dans un environnement de conception multiphysique et multidisciplinaire tout en gardant la rigueur mathématique liée aux méthodes d'optimisation. La démarche de notre projet de recherche est synthétisée dans la figure 1.1.

A l'heure actuelle, la notion de « meilleur » et donc d'optimisation paraissent évidentes et parfaitement définie. D'un point de vue mathématique un problème d'optimisation consiste à minimiser ou maximiser une fonction objectif dépendant de variables de conception avec ou sans contraintes. Cependant, la traduction dans le langage mathématique d'un problème concret est souvent problématique. En effet, il est nécessaire à l'ingénieur de hiérarchiser et de pondérer des critères pour optimiser sa conception et satisfaire au mieux son cahier des charges. Or,

Or cette démarche d'optimisation, dans le cadre d'une conception de produit complexe et multiphysique, n'est que très peu appliquée à l'heure actuelle en entreprise. L'une des raisons est que l'optimisation est une discipline à part entière qui peut être difficile à appréhender. De plus, l'absence d'outil pour aider à la traduction des préférences exprimées par un concepteur en données d'entrées nécessaires à la manipulation et l'emploi des techniques d'optimisation performantes est un réel frein à leur utilisation qu'elles soient spécifiques ou généralistes. Ce point sera abordé dans le chapitre 3.

La deuxième phase est une étape de résolution faisant appel à un algorithme/une procédure d'optimisation adapté au problème posé. A partir de la formulation mathématique du problème, des algorithmes robustes et efficaces sont mis en œuvre pour la phase de résolution. Les méthodes d'optimisation classiques notamment à base de gradient peuvent s'avérer inefficaces face à des problèmes d'optimisation dits difficiles. En effet, dans de nombreux cas, notamment pour l'optimisation en conception mécanique, les caractéristiques mathématiques des grandeurs telles que la fonction objectif ou le domaine admissible sont complexes à établir : continuité, différentiabilité et convexité. L'obtention d'un minimum global devient alors un vrai challenge dans un temps raisonnable. C'est pourquoi une nouvelle classe d'approches a émergé et fait maintenant partie intégrante du paysage des méthodes d'optimisation : les métaheuristiques. Nous avons choisi de nous intéresser à une méthode en particulier : la méthode par essaim particulière (Particle Swarm Optimization : PSO). Nous avons donc développé des variantes originales de cette approche pour tenter de réduire le temps de calcul, trop important pour permettre l'optimisation dans des délais compatibles avec l'industrie : critère d'arrêt sur le rayon de l'essaim, accélération de convergence par saut de l'essaim. Ce point sera abordé dans le chapitre 4.

Dans un contexte industriel et technologique, le développement de nouvelles approches n'apparaît pas suffisant. Il faut être en mesure de certifier les approches en terme de qualité et de robustesse. En effet, de nombreuses approches d'optimisation ont été mises au point ces dernières années sans véritable certitude sur leur efficacité. Ce sont ces remarques qui nous ont conduit à mettre en place un nouveau benchmark. Le principe des benchmarks est assez simple : exécuter des algorithmes sur série de tests "standardisés" (différents types de fonctions, différentes dimensions ...), récupérer et analyser les résultats via le calcul d'un score. Ce point sera abordé dans le chapitre 5.

Les algorithmes d'optimisation 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 métaheuristique, de connaître parfaitement quels sont les effets de chaque paramètre sur le fonctionnement de la méthode en question. Les paramètres de l'algorithme peuvent être à leur tour optimisés par une méthode d'optimisation : le paramétrage optimal est le résultat d'un problème d'optimisation résolu ou non avec l'algorithme testé. Dans la littérature, les plans expériences ont déjà été utilisés pour régler les paramètres de certains algorithmes d'optimisation. Cependant le constat est le suivant : l'optimisation du paramétrage est réalisée au cas par cas. Ceci rend la réutilisation d'un paramétrage dans un autre contexte inutile. En effet, un paramétrage doit être adapté à chaque situation. L'idée du travail développé est de parvenir à établir des modèles de paramétrisation adaptés à un nombre de situations de références. L'originalité du travail mis en place est de proposer un plan d'expérience sur notre benchmark suivi d'une analyse par composantes principales. La structure même du benchmark rend possible une étude à la fois globale pour établir un jeu de paramètres généralistes robustes via le score global mais aussi une étude plus locale par l'intermédiaire des sous-scores. Ce point sera abordé dans le chapitre 6.

Dans le chapitre 7, nous nous intéresserons à un domaine particulier de l'optimisation, l'optimisation topologique, dans le domaine de la conception et de la fabrication additive. En effet, le concepteur doit être amené à imaginer de nouvelles solutions créatives. L'objectif des travaux en cours consiste donc à intégrer dans la démarche de conception les possibilités offertes par la fabrication additive. Pour ce faire, il semble opportun de mettre en place une approche intégrant simulation, optimisation topologique dans la démarche de conception.

Dans le chapitre 8, nous évoquerons les perspectives de ces travaux et la suite.

I

NOTICE

CURRICULUM VITÆ

2.1/ Situation actuelle

Fonction :	Maître de conférences depuis le 01/02/2004 Responsable de la filière d'enseignement MOST ¹ FISE Mécanique ² , UTBM
Section CNU :	60 (Mécanique, génie mécanique, génie civil)
Établissement :	Université de Technologie de Belfort-Montbéliard (UTBM) 90010 Belfort cedex
Laboratoire de recherche :	Laboratoire Interdisciplinaire Carnot de Bourgogne (ICB) UMR CNRS 6303 Département Conception, Optimisation et Modélisation en Mécanique (CO2M)
Email :	dominique.chamoret@utbm.fr
Tel :	+33 (0) 3 84 58 38 03

1. MOST : Modélisation optimisation systèmes thermomécaniques
2. FISE : Formation Ingénieur Statut Étudiant

2.2/ Parcours et formation

2.2.1/ Titres universitaires

1996 **DEUG Mathématiques, Informatique et Applications aux Sciences (MIAS)**, *Université de Montpellier II*, mention Bien.

1998 **Licence de Mathématiques**, *Université de Montpellier II*.

1998 **Maîtrise d'ingénierie mathématique**, *Université de Montpellier II*.

1999 **DEA de Mathématiques et Mécanique Théorique**, *Université de Montpellier II*, mention Assez Bien.

- ❖ Titre : Étude comparative de deux approches du traitement du contact dans les méthodes de décomposition de domaine

2002 **Doctorat en Mécanique**, *École Centrale de Lyon*, Thèse CIFRE ESI Software préparée au Laboratoire de Tribologie et Dynamique des Systèmes, UM5513, Soutenue le 4 décembre 2002, Mention très honorable avec les félicitations du jury.

- ❖ **Titre** Modélisation du contact - Nouvelles approches numériques
- ❖ **Composition du Jury** : Pr. J.-M. Roelandt, LMR, UMR6066 (Rapporteur, section 60) , Pr. P. Chabrand, LABM, UMSR 2164 (Rapporteur, section 60) , Pr. A. Rassinoux, LMR, UMR6066 (Examinateur) , Pr. H. Zahaouani, LTDS, UMR 5513 /CNRS/ECL/ENISE (Examinateur) , Dr. P. Saillard, Docteur, ESI France (Invité) , Pr. J.-M. Berhgheau LTDS, UMR 5513 /CNRS/ECL/ENISE (Directeur de thèse)

2.2.2/ Parcours professionnel

Depuis **Maître de conférences (section 60)**, *Université de Technologie de Belfort Montbéliard*, Laboratoire Interdisciplinaire Carnot de Bourgogne, Département CO2M, UMR CNRS 6303.

Septembre 2003 **Post-doctorat**, *Delft University of Technology*, Faculty of Aerospace Engineering, Chair of Engineering Mechanics.
à Février 2004

Janvier 2003 **Post-doctorat**, *INSA de Lyon*, LAMCOS.
à Juillet 2003

Novembre 1999 **Préparation d'une thèse de Doctorat**, *Thèse soutenue le 4 décembre 2002 sous la direction du Pr. Jean-Michel Bergheau*.

II

SYNTHÈSE SCIENTIFIQUE

OPTIMISATION

Sommaire

3.1	Contexte	7
3.2	Processus d'optimisation	7
3.3	Formulation mathématique d'un problème d'optimisation	8
3.3.1	Équations de base	8
3.3.2	Classification d'un problème d'optimisation	9
3.4	Formulation d'un problème d'optimisation en conception	10
3.4.1	Contexte - problématique : rôle du concepteur en optimisation	10
3.4.2	Interaction concepteur - Expert en optimisation	11
3.4.3	Catégorisation	11
3.4.4	Déroulement de l'interview concepteur/expert en optimisation	12
3.5	Conclusions et perspectives	15

3.1/ Contexte

Comme évoqué en introduction, d'un point de vue mathématique tout est clair : un problème d'optimisation consiste à minimiser ou maximiser une fonction objectif dépendant de variables de conception avec ou sans contraintes. Cependant dans le processus de conception, tout ceci se complique. Ce constat a servi de socle au développement de mes activités de recherche dans le cadre de l'optimisation. La direction principale de mes travaux de recherche en optimisation est de développer un/des outils performants pour accompagner au mieux l'ingénieur dans sa démarche d'optimisation en tenant compte des contraintes de l'univers industriel dans un environnement de conception multiphysique et multidisciplinaire, mais aussi et surtout en gardant la rigueur mathématique liée aux méthodes d'optimisation.

3.2/ Processus d'optimisation

L'optimisation en conception peut être divisée en plusieurs phases qui alternent entre des expertises différentes : la conception et l'optimisation. C'est bien là, le problème de l'optimisation en conception ! C'est ce point que nous allons évoquer dans la suite de ce chapitre.

D'après [66], les étapes classiques d'un processus de conception sont les suivantes (Figure 3.1) :

- ❖ La première phase s'articule principalement autour d'un ou des outil(s) de simulation. Elle requiert la définition d'un ensemble de variables de conception, la définition d'une ou de plusieurs fonctions objectifs et contraintes issues en général de l'outil de simulation (Éléments finis, modélisation 1D ...). Il s'agit de formuler correctement le problème d'optimisation.
- ❖ La deuxième phase est la phase de résolution faisant appel à un algorithme/une procédure d'optimisation adaptée au problème posé. A partir de la formulation mathématique du problème, des algorithmes robustes et efficaces sont mis en œuvre pour la phase de résolution. Dans ce manuscrit, nous porterons une attention particulière aux méthodes non déterministes telles que les métaheuristiques.
- ❖ La dernière phase est la phase de validation.

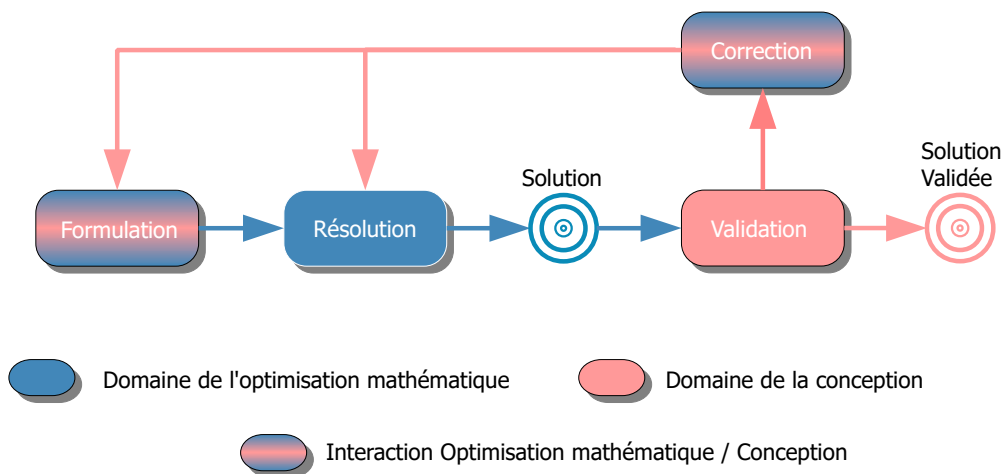


FIGURE 3.1 – Processus d'optimisation : une alternance d'expertise

3.3/ Formulation mathématique d'un problème d'optimisation

Ici les deux expertises prennent tout leur sens. D'un point de vue mathématique, tout est clair mais rien n'est moins sûr du côté du concepteur !

3.3.1/ Équations de base

Les caractéristiques communes à tous les problèmes d'optimisation mettent en évidence un problème qui consiste à sélectionner une décision parmi un ensemble de décisions possibles, de manière à optimiser un certain critère d'un système. En général, l'analyse (ou évaluation) du comportement du système (et/ou structure) passe par l'utilisation d'un/des logiciels de calcul (éléments finis ou autres Modèle Dymola, modelica, ...).

Ainsi, la formalisation mathématique d'un problème d'optimisation requiert trois points essentiels (Figure 3.2) :

- ❖ \mathbf{X} un ensemble de variables d'optimisation permettant de décrire l'état du système. D est la dimension du problème, autrement dit le nombre de variables de conception. \mathcal{S} est l'ensemble admissible de ces variables.
- ❖ F , une fonction coût/objectif dépendant de \mathbf{X} . Cette fonction correspond à un besoin ou à une préférence souvent explicite de la part du concepteur (Son calcul peut résulter d'un modèle).
- ❖ $g_j(\mathbf{X}) \leq 0$ avec $j = 1, \dots, p$, un certain nombre de limitations (contraintes). Il s'agit de restrictions. Une solution qui viole une contrainte est inacceptable, une solution qui respecte toutes les contraintes est acceptable.

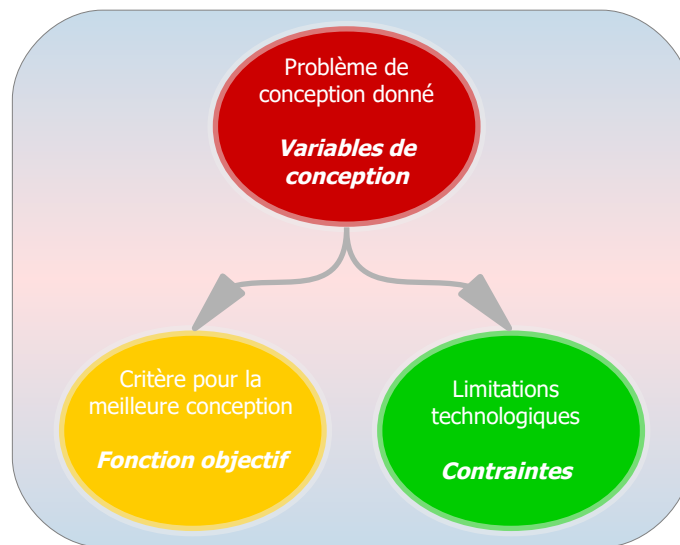


FIGURE 3.2 – Formulation d'un problème d'optimisation

De manière générale, on peut formaliser mathématiquement un problème d'optimisation classique comme suit

$$\left\{ \begin{array}{l} \min F(\mathbf{X}) \quad \text{avec} \quad \mathbf{X} \in \mathcal{S} \subset \mathbb{R}^D \quad \text{et} \quad \mathbf{X} = (X_1, \dots, X_D) \\ \text{soumis à} \quad g_j(\mathbf{X}) \leq 0 \quad \text{avec} \quad j = 1, \dots, p \end{array} \right. \quad (3.1)$$

3.3.2/ Classification d'un problème d'optimisation

Un problème d'optimisation peut être classé en différentes catégories. La nature de ce problème d'optimisation dépend de la formulation et de la nature mathématique des ingrédients de base que sont la fonction objectif, les variables et les contraintes. Une classification complète et détaillée adaptée à l'optimisation en conception est proposée dans [77]. Nous inspirant de celle-ci, nous en proposons une version allégée sur la figure 3.3.

La classification d'un problème d'optimisation est fondamentale. En effet, une démarche de catégorisation des connaissances semble être une étape incontournable pour le processus de

communication entre les concepteurs et les experts en optimisation. De cette classification découlera le choix de l'algorithme le plus performant pour le problème traité!

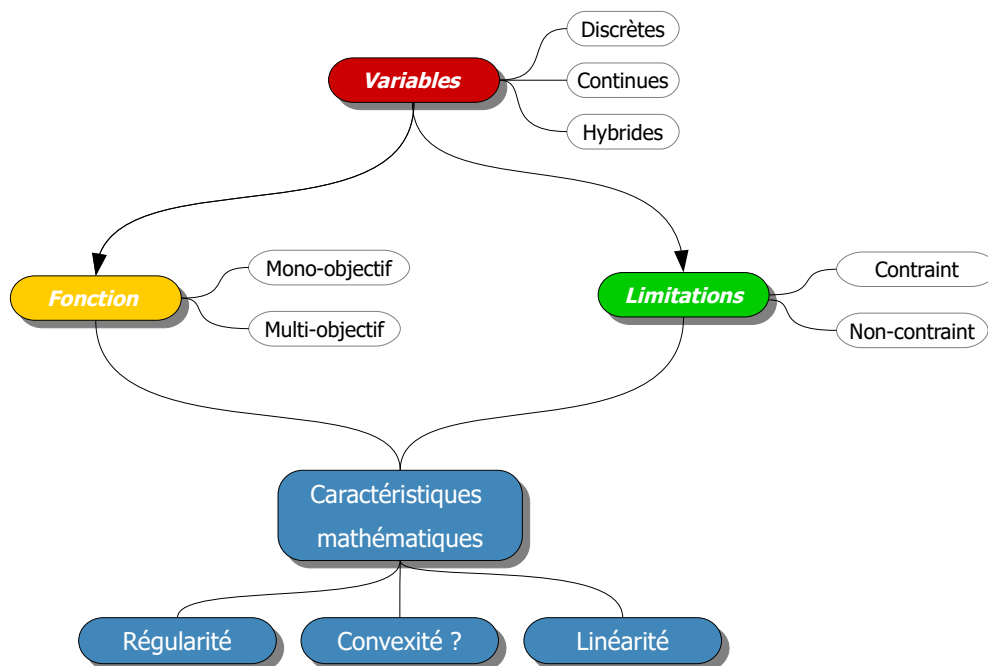


FIGURE 3.3 – Classification d'un problème d'optimisation

3.4/ Formulation d'un problème d'optimisation en conception

3.4.1/ Contexte - problématique : rôle du concepteur en optimisation

Même si de très nombreuses méthodes permettent de résoudre le problème mathématique (Équation 3.1), le concepteur joue toujours un rôle primordial! En effet, formuler un problème d'optimisation exploitable correctement d'un point de vue mathématique est un vrai challenge : un problème peut faire intervenir des critères ou des connaissances subjectives qui peuvent varier d'un concepteur à l'autre et les paramètres sont souvent mal maîtrisés. Le contexte UT et les problématiques au sein du Département CO2M nous conduisent à nous intéresser au développement d'une méthodologie permettant une meilleure interaction entre les concepteurs et les experts métiers en optimisation.

Nous avons donc initié un travail sur des méthodes de recueil des données nécessaires à la mise en place du processus d'optimisation plus exactement à une bonne formulation. Ce travail a débuté pendant la thèse de François Schott [81] et a été mené en collaboration

avec Florence Bazzaro, Maître de conférences Conception centrée utilisateur à l'UTBM. L'objectif est de mettre en évidence la complexité et l'hétérogénéité des activités, des acteurs, des modèles utilisés, afin de positionner ce travail de recherche sur la gestion des informations dans un processus d'optimisation. Les différents domaines d'expertises impliqués sont à l'origine de nombreuses difficultés de partage et de collaboration entre les différents acteurs d'un projet.

3.4.2/ Interaction concepteur - Expert en optimisation

La grande difficulté est d'intégrer les concepteurs dans la démarche d'optimisation sans allonger les délais : le point le plus délicat est la description et la formulation d'un problème. Les besoins des concepteurs doivent être traduits d'un point de vue mathématique dans les trois grandeurs classiques à tout problème d'optimisation : fonction objectif, variables d'optimisation et contraintes comme évoqué dans la section 3.3.

Plusieurs critères compliquent la communication entre les concepteurs et les experts en optimisation. Par exemple, les connaissances implicites, les connaissances subjectives et perceptives des concepteurs sont des éléments sur lesquels nous devons nous pencher. Mais encore, nous retrouvons le manque de connaissances des concepteurs sur les problèmes d'optimisation qui sont eux-mêmes d'une très grande diversité.

De tout ceci, nous pouvons conclure, qu'à afin d'assister la communication entre les concepteurs et experts optimisation, la méthodologie à mettre en place doit :

- ❖ aider à définir tous les paramètres du problème d'optimisation à prendre en compte (i.e. : comment bien faire le tour de la question ?)
- ❖ aider à créer les liens entre ces paramètres.
- ❖ aider à faire ressortir les :
 - ✓ Connaissances implicites.
 - ✓ Les opinions/jugements.
 - ✓ L'expérience.

3.4.3/ Catégorisation

Pour instaurer un dialogue constructif entre concepteurs et experts en optimisation, il nous est apparu fondamental de représenter toutes les caractéristiques, liées aux produits (de type matériel), en une cartographie (semblable à celle réalisée dans l'étude [42]) compréhensible de tous, notamment pour mettre au clair le vocabulaire !

Autrement dit, nous avons recherché un maximum de variables élémentaires mesurables qui caractérisent un produit, pour les catégoriser ultérieurement. Ces caractéristiques pouvaient être ses propriétés mécaniques, ses performances techniques, son aspect, etc. L'objectif était de créer une liste exhaustive. L'idée était d'étendre et d'adapter certains concepts bien connus et issus de l'univers de la conception à la démarche d'optimisation [9]. Les recherches ont été menées en utilisant [4] et les outils TRIZ [18, 15] Cependant il nous est vite apparu que cette catégorisation seulement autour du produit n'était pas complète. Il a fallu intégrer le concept de procédé de fabrication. Ceci nous a amené à établir une catégorisation en deux temps (Figure 3.4).



FIGURE 3.4 – Catégorisation

Catégorisation "produit/matériaux" Dans cette catégorisation, il nous est apparu important de créer trois niveaux, listés ci-dessous avec leur définition associée :

- ❖ **Matériau** : comprend les caractéristiques propres au matériau. Un matériau est défini comme un matériau élémentaire sans la prise en compte de sa forme.
- ❖ **Produit** : comprend les caractéristiques propres au produit. Un produit est défini comme un ou plusieurs matériaux élémentaires étant passés par un procédé.
- ❖ **Couple matériau-produit** : comprend les caractéristiques qui peuvent convenir à la fois au matériau et au produit

Cette catégorisation est proposée dans la figure 3.5, page 16.

Couple "Produit/Procédé" Dans cette catégorisation, il nous est aussi apparu important de créer trois niveaux, listés ci-dessous avec leur définition associée :

- ❖ **Produit** : comprend les caractéristiques à optimiser en considérant uniquement le produit.
- ❖ **Procédé** : comprend les caractéristiques à optimiser en considérant uniquement le procédé.
- ❖ **Couple produit-procédé** : comprend les caractéristiques à optimiser en considérant le produit ainsi que le procédé simultanément.

Cette catégorisation est proposée dans la figure 3.6, page 17.

3.4.4/ Déroulement de l'interview concepteur/expert en optimisation

Avec cette cartographie en main, nous pouvons organiser le dialogue entre les différents protagonistes (Figure 3.7). Un séquençement de ce dialogue est fondamental et se fait en trois temps comme indiqué sur la figure 3.8.

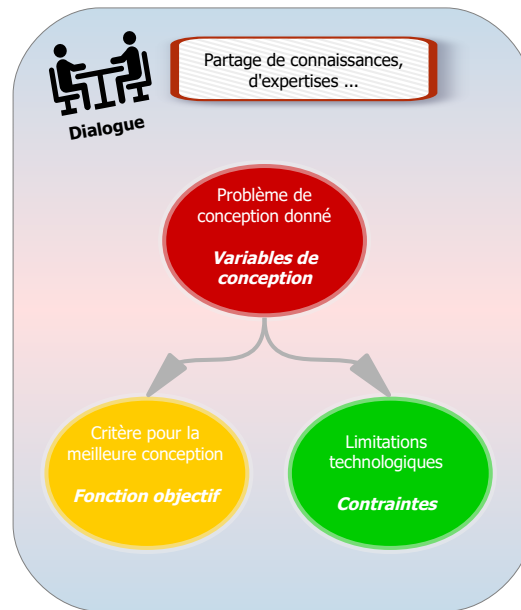


FIGURE 3.7 – Instauration d'un dialogue pour la formulation

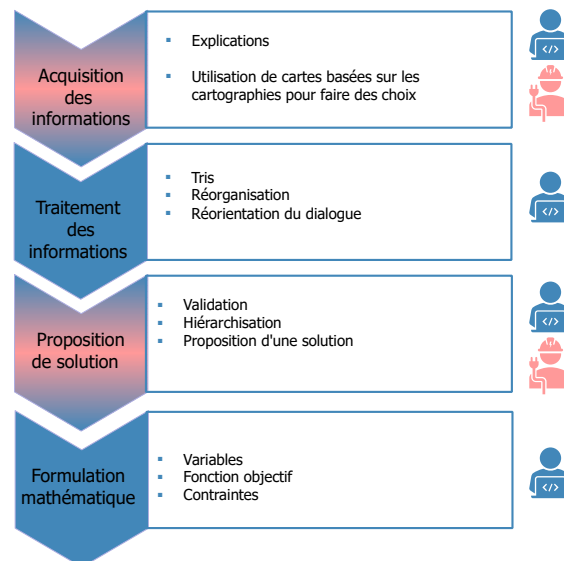


FIGURE 3.8 – Séquencement du dialogue

Acquisition des informations : Cette étape est la plus longue et la plus fastidieuse. L'expert en optimisation utilise des cartes illustrées (Figure 3.9) basées sur les cartographies développées précédemment pour faire émerger les idées du concepteur et le guider dans la direction voulue afin de pouvoir établir les principales données du problème à optimiser.

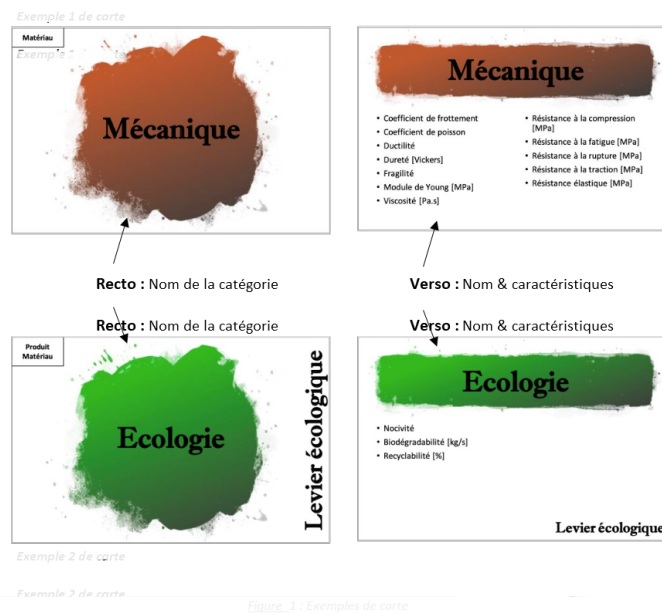


FIGURE 3.9 – Exemple de cartes illustrées pour dialogue

A l'issue de cette étape certaines idées ont émergé et ont commencé à être classées (Figure 3.10).

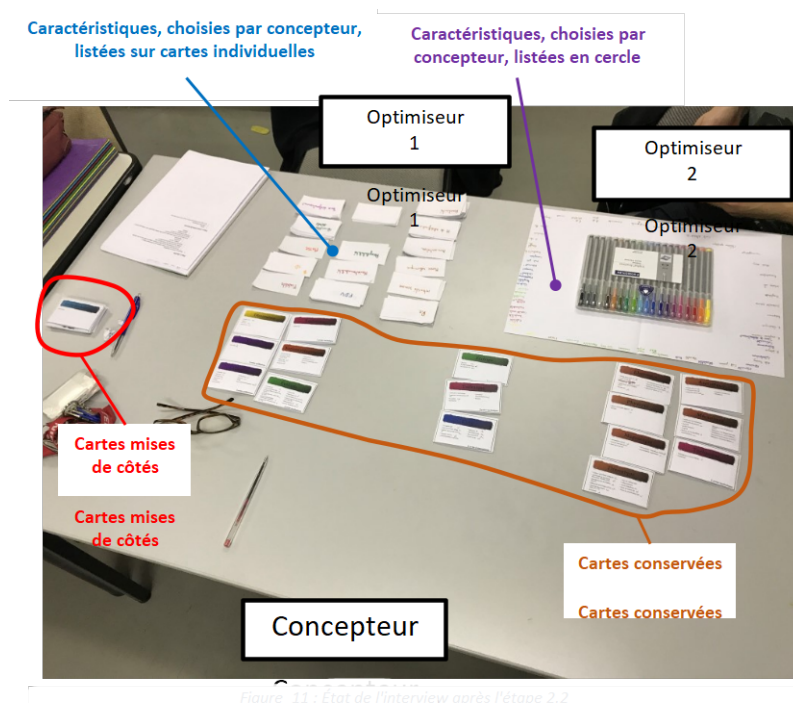


FIGURE 3.10 – État de l'interview à la fin de l'étape d'acquisition des données

Traitement des informations : Une fois cette étape de récolte d'informations terminée, l'expert en optimisation doit reprendre les cartes conservées (Figure 3.10) afin d'établir

un tri concernant les objectifs à atteindre, les limitations et les variables. L'expert en optimisation va pouvoir ainsi réorienter le dialogue si besoin.

Validation : Les premiers choix effectués par l'expert en optimisation sont proposés au concepteur pour validation. Une seconde phase de discussion est ouverte pour hiérarchiser, valider et finalement faire émerger une solution.

Formulation mathématique : Pour terminer, l'expert en optimisation est capable d'établir la formulation complète du problème d'optimisation. Les données nécessaires à l'écriture mathématique sont maîtrisés. Tout est en place pour le traitement algorithmique du problème.

Cette analyse a été menée auprès de concepteurs, calculateurs et experts divers, sous forme d'interviews approfondies dans le cadre de la thèse de François Schott [81] et d'un projet de niveau master [76] , mais ne sera pas détaillée dans ce manuscrit.

3.5/ Conclusions et perspectives

On s'est très vite aperçu de la limite du processus. L'outil informatique et notamment les outils de gestion des données et des connaissances sont une solution pour une efficacité accrue. Dans le cadre de l'encadrement de la thèse de Julien Badin [7], je me suis intéressée aux méthodologies de gestion des connaissances. Dans ce travail, ont été développés une méthodologie et un méta-modèle pour la gestion des données, informations et connaissances pour les activités de conception et de simulation numérique, mis en œuvre plus spécifiquement, lors des phases amont du processus d'ingénierie, afin de favoriser la collaboration entre les métiers "Conception" et "Calcul". Une des perspectives concerne l'extension de ces concepts à l'optimisation.

Une fois le problème d'optimisation bien formulé d'un point de vue mathématique et catégorisé (comme évoqué dans la section 3.3.2), une résolution algorithmique peut être envisagée. C'est ce point que nous allons aborder dans le chapitre suivant.

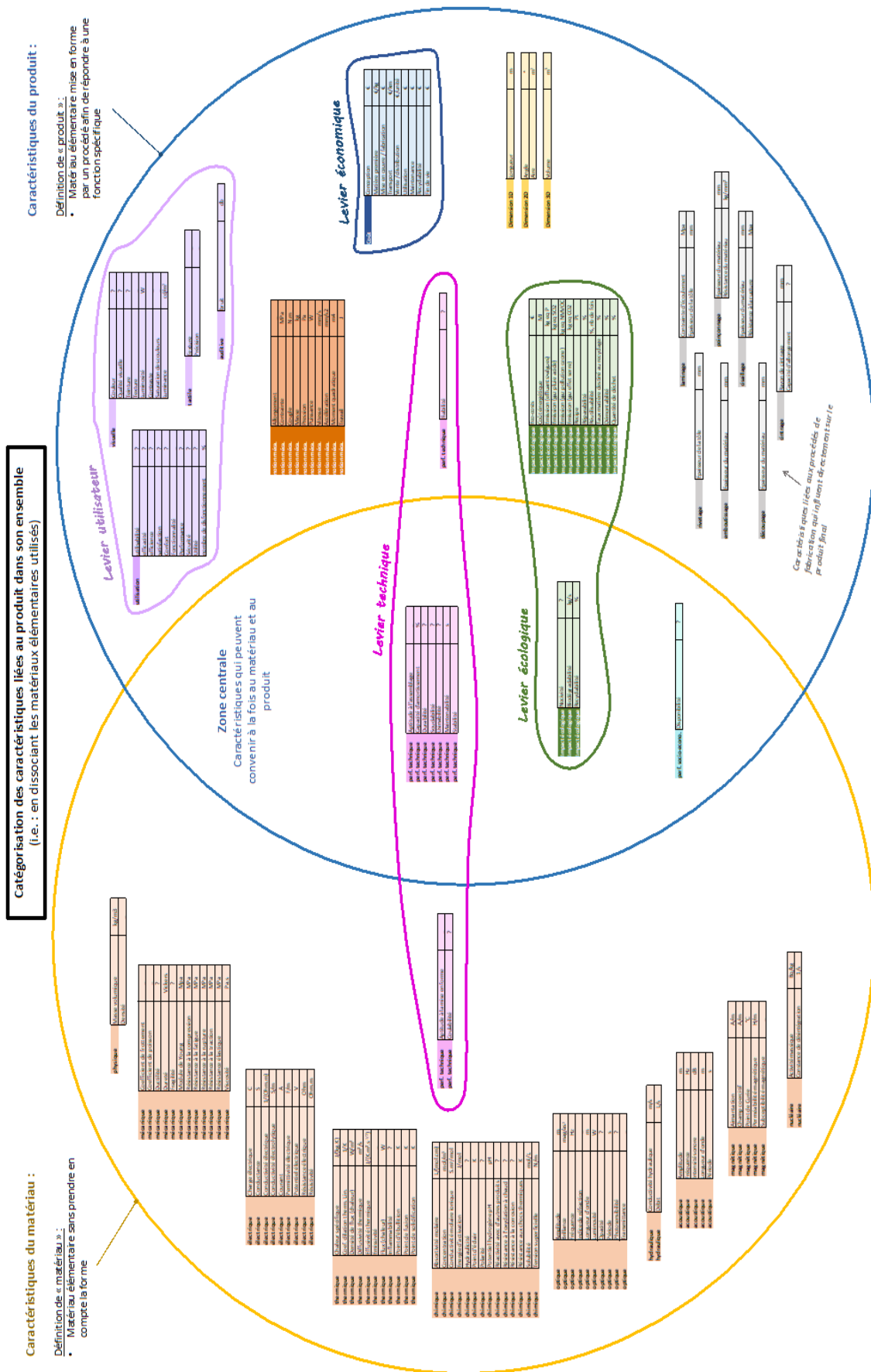


FIGURE 3.5 – Catégorisation des caractéristiques du produit

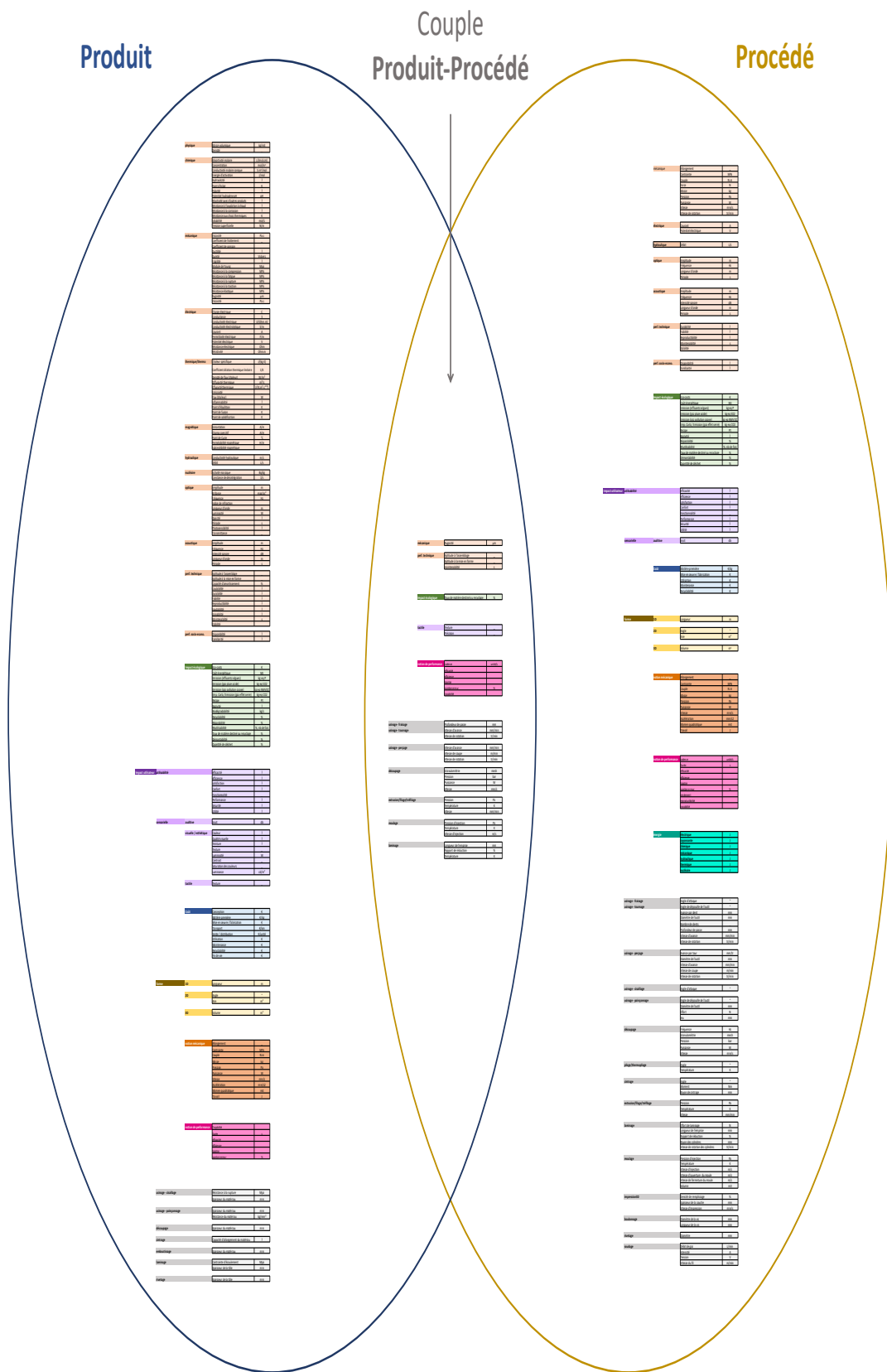


FIGURE 3.6 – Catégorisation des caractéristiques du produit

FOCUS SUR UNE MÉTAHEURISTIQUE ET SES AMÉLIORATIONS

Sommaire

4.1	Introduction	19
4.2	Métaheuristiques à base de population	21
4.2.1	Ingrédients de base	21
4.2.2	Exploration de l'espace	22
4.3	PSO	23
4.3.1	Principe de base	23
4.3.2	Paramètres du PSO	25
4.3.3	Critère d'arrêt et convergence	27
4.3.4	Topologie de voisinage	27
4.4	Prise en compte des contraintes	28
4.4.1	Harmony Search scheme	29
4.4.2	Méthode de pénalisation	29
4.4.3	Fonction en cascades	30
4.5	Un nouveau critère d'arrêt et saut d'essai	30
4.5.1	Un nouveau critère d'arrêt : PSO-Radius	30
4.5.2	Accélération de convergence par saut de l'essai	31
4.5.3	Validation et application aux matériaux composites à matrice céramique	31
4.6	I-PR-PSO	35
4.6.1	Relation topologie - chaînes de Markov à temps discret	35
4.6.2	Principe de base	36
4.6.3	Validation sur un ensemble de fonctions tests	37
4.6.4	Validation en optimisation de structure mécanique	41
4.7	Conclusions et perspectives	42

4.1/ Introduction

Les méthodes d'optimisation classiques notamment à base de gradient peuvent s'avérer inefficaces face à des problèmes d'optimisation dits difficiles. En effet, dans de nombreux

cas notamment pour l'optimisation en conception mécanique, les caractéristiques mathématiques des grandeurs telles que la fonction objectif ou le domaine admissibles sont complexes à établir : continuité, différentiabilité et convexité. L'obtention d'un minimum global devient alors un vrai challenge dans un temps raisonnable.

C'est pourquoi une nouvelle classe d'approches a émergé et fait maintenant partie intégrante du paysage des méthodes d'optimisation : les métaheuristiques. Les métaheuristiques sont des algorithmes - en grande majorité - stochastiques itératifs, qui progressent vers un optimum global.

L'objectif principal des métaheuristiques est d'explorer l'espace de recherche efficacement, en tentant parfois d'apprendre les caractéristiques d'un problème, afin de déterminer une approximation de la meilleure solution. Ces algorithmes peuvent aller de la simple recherche locale à des processus d'apprentissage complexe. Ces approches sont en général non-déterministes et 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. Leur utilisation est possible à la fois dans le domaine de l'optimisation combinatoire comme le domaine de l'optimisation continue. De fait, ces méthodes sont parfaitement adaptées au contexte de l'optimisation dite difficile.

Les qualités premières des métaheuristiques résident dans le fait qu'elles procèdent à une exploration efficace à partir d'un minimum d'informations :

- ❖ Pas de nécessité de connaître la forme du domaine étudié.
- ❖ Pas de nécessité de connaître l'évolution de la fonction objectif dans son domaine.
- ❖ Pas de nécessité de conditions de régularité (Continuité, différentiabilité, convexité ...)
- ❖ Pas de calcul du gradient.

De nombreuses façons de classer les métaheuristiques ont été proposées dans la littérature. Il est quasi impossible de toutes les énumérer ici. Rentrer dans le détail de toutes les métaheuristiques et toutes les citer ici n'est pas le propos. De nombreux auteurs ont abordé ces classifications et référencements des différentes approches [75, 31, 1].

Dans la suite de notre travail nous retiendrons celle qui différencie les méthodes de trajectoires et les méthodes à base de population (Figure 4.1). Une manière de classer les métaheuristiques est de distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire, pour les autres il est question de métaheuristiques à base de population. Il existe un très grand nombre de métaheuristiques différentes tant et si bien qu'il est difficile de toutes les nommer et de donner une définition claire et précise de ces approches. La littérature foisonne de description de ces approches. On peut se reporter par exemple à [24].

Après avoir mené des études sur différentes approches [?, ?], nous avons choisi de nous intéresser à une méthode en particulier : la méthode par essaim particulaire (Particle Swarm Optimization : PSO) pour diverses raisons notamment sa simplicité de mise en œuvre, son faible nombre de paramètres à régler et son efficacité en optimisation dite difficile. Ceci rendait cette approche envisageable dans le cadre de l'optimisation en conception. La suite de ce chapitre est donc dédié à une présentation rapide des métaheuristiques à base de population (Section 4.2). Nous nous focaliserons ensuite sur la méthode PSO (Section 4.3) avec une description détaillée afin d'en saisir toutes les subtilités et comprendre les

limites. Nous verrons ensuite les améliorations que nous avons développées au cours de nos travaux : nouveau critère d'arrêt, tentative d'amélioration de la convergence (Section 4.5) et topologie évolutive (Section 4.6).

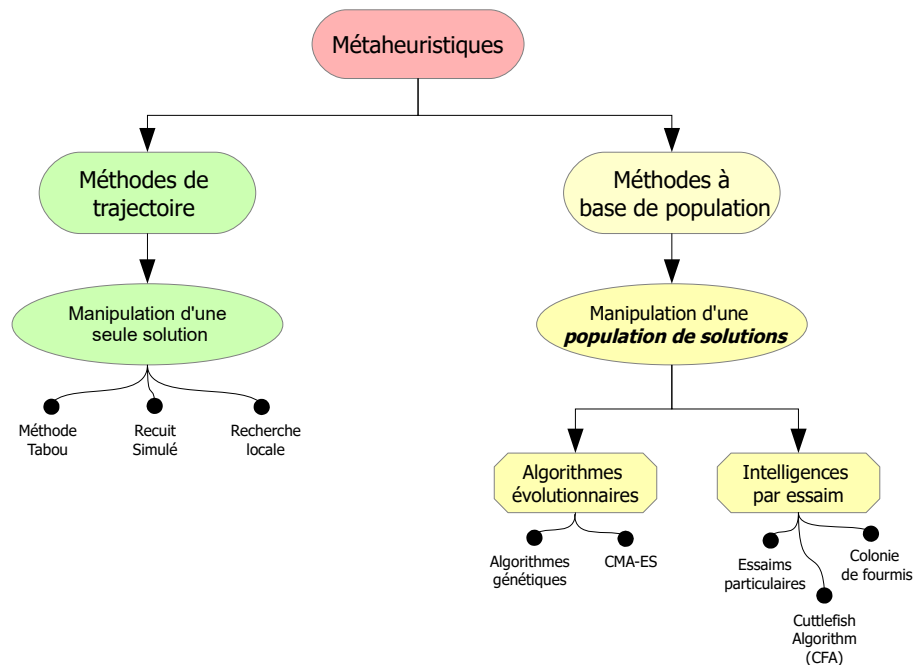


FIGURE 4.1 – Une classification des méthodes métaheuristiques

4.2/ Panorama des métaheuristiques à base de population

Les métaheuristiques à population de solutions améliorent, au fur et à mesure des itérations, une population de solutions. Dans cette catégorie, nous pouvons distinguer les algorithmes évolutionnaires, qui sont une famille d'algorithmes issus de la théorie de l'évolution par la sélection naturelle et les algorithmes d'intelligences par essaim, qui proviennent d'analogies avec des phénomènes biologiques naturels.

Malgré tout, il est possible de mettre en avant certaines propriétés et ingrédients communs à ces différents algorithmes.

4.2.1/ Ingrédients de base

Plusieurs entités communes à toutes les métaheuristiques doivent être définies et sont nécessaires à leur bonne mise en œuvre.

- ❖ Espace de recherche
 - ✓ Il s'agit du domaine dans lequel la recherche va s'effectuer pour trouver la solution optimale. Il doit être à minima borné.

❖ Population

- ✓ Il s'agit ici de l'ensemble de solutions obtenues par échantillonnage de l'espace de recherche. Dans la plupart des cas, cette population est générée au hasard. Nous verrons aussi que la taille de cette population est un élément clé de l'algorithme. Trop petite, elle conduit à une perte de diversité de l'approche et trop grande, elle conduit à des temps de calculs trop importants !

❖ Des règles d'évolution

- ✓ Ces algorithmes modifient la solution courante via un mouvement dicté par des règles.

❖ Un ensemble de paramètres

- ✓ Toutes les décisions liées au fonctionnement de l'algorithme peuvent être paramétrées. Ces approches sont donc régies par des 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 métaheuristique, de connaître parfaitement les effets de chaque paramètre sur le fonctionnement de la méthode en question. Nous détaillerons ce point pour l'algorithme PSO dans la section 4.3.

4.2.2/ Exploration de l'espace

Un dernier point doit être souligné, commun à la plupart des métaheuristiques. Les règles d'évolution de la population permettront d'explorer judicieusement l'espace. Cette exploration repose sur deux concepts fondamentaux : l'intensification et la diversification.

4.2.2.1/ La diversification

Le principe de diversification d'une métaheuristique représente sa capacité à parcourir facilement l'espace de recherche dans le but d'obtenir des solutions les plus différentes les unes des autres (Figure 4.2)

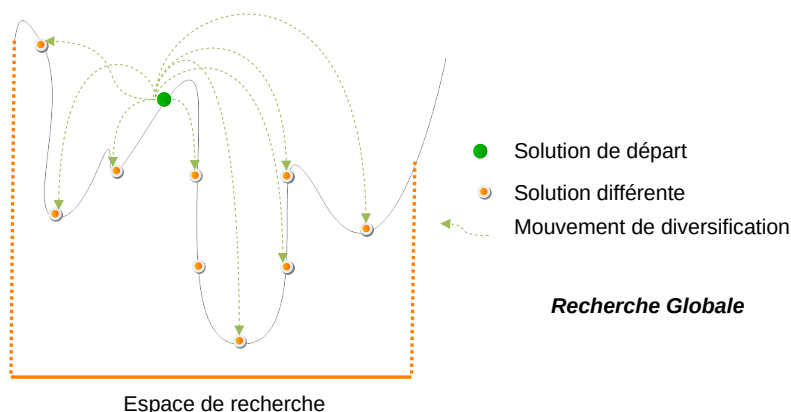


FIGURE 4.2 – Diversification

4.2.2.2/ L'intensification

Le principe d'intensification est la capacité de l'algorithme à amener une solution vers l'optimum local de la zone à laquelle elle est attachée (Figure 4.3). L'avantage mais aussi la faiblesse de l'intensification est de rester bloqué dans la zone dans laquelle la solution se trouve sans possibilité de pouvoir sortir de celle-ci. Donc si les solutions que l'on veut intensifier ne sont pas dans la zone de l'optimum global, il n'y a aucune chance que l'intensification seule permette d'y accéder.

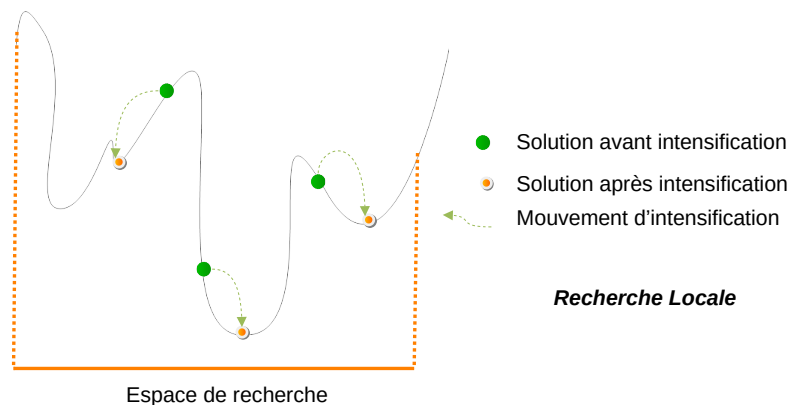


FIGURE 4.3 – Intensification

L'efficacité d'un algorithme repose sur un juste équilibre entre diversification et intensification. Cet équilibre lui-même apparaît donc comme un problème d'optimisation de l'algorithme d'optimisation ! Il peut dépendre de nombreux facteurs, tels que le mécanisme de fonctionnement d'un algorithme, la définition de ses paramètres, le réglage et le contrôle de ces paramètres. De plus, il peut ne pas exister universellement, et il peut varier d'un problème à l'autre. Ces points seront abordés en particulier dans le chapitre 6.

4.3/ Optimisation par essaim particulaire

Nous avons fait le choix dans nos travaux de nous focaliser sur une métaheuristique en particulier : la méthode par essaim particulaire pour diverses raisons notamment sa simplicité de mise en œuvre.

4.3.1/ Principe de base

L'algorithme d'optimisation par essaim particulaire (appelé Particle Swarm Optimization dans la littérature anglophone) est une méthode d'optimisation fondée sur une population stochastique de particules initialement réparties sur un domaine de recherche. Cette méthode a été publiée la première fois par Kennedy et Eberhart en 1995 [26]. Cette approche s'est très rapidement positionnée 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.

Cette approche génère un essaim de particules dont chaque membre est une **solution éventuelle** du problème d'optimisation traité, à laquelle est associée une valeur de la fonction objectif pour cette position en particulier. Cet essaim vole dans l'espace de recherche (à D dimensions) et chaque membre de celui-ci est attiré par sa meilleure solution et celle de ses voisins. Chaque particule est donc dotée d'une mémoire contenant les données relatives à son vol (position, vitesse et sa meilleure solution du problème) ainsi que de la faculté de communiquer (socialisation) avec son entourage.

Ainsi, chaque particule peut combiner linéairement trois possibilités pour établir sa nouvelle position :

- Suivre sa meilleure performance personnelle,
- Suivre la meilleure performance de son voisinage,
- Suivre sa vitesse actuelle (comportement qualifié d'égoïste dans la littérature).

La position des particules dans l'espace des solutions à l'itération t est fonction de leur position actuelle, 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 :

$$\mathbf{X}_i^{k+1} = f(\mathbf{X}_i^k, \mathbf{V}_i^k, \mathbf{P}_{i,best}^k, \mathbf{G}_{i,best}^k) \quad (4.1)$$

où

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

Dans la version la plus standard, le changement de position s'effectue donc comme exprimé par l'équation (4.2) et illustré par la figure 4.4.

$$\begin{cases} \mathbf{V}_i^{k+1} = \omega_k \mathbf{V}_i^k + c_1 r_1 (\mathbf{P}_{i,best}^k - \mathbf{X}_i^k) + c_2 r_2 (\mathbf{G}_{i,best}^k - \mathbf{X}_i^k) \\ \mathbf{X}_i^{k+1} = \mathbf{V}_i^{k+1} + \mathbf{X}_i^k \end{cases} \quad (4.2)$$

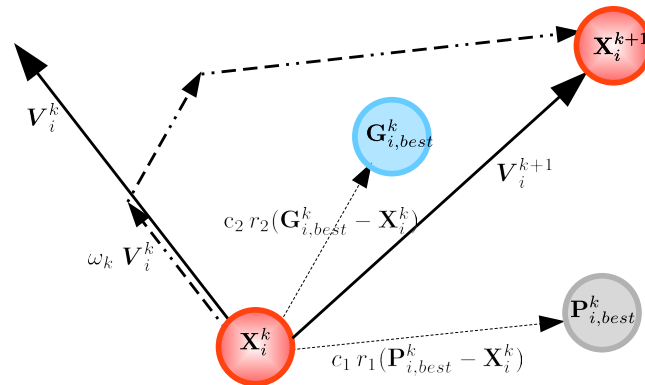


FIGURE 4.4 – Mode de déplacement des particules dans l'espace des solutions

Remarque : Le terme de vitesse est ici abusif. Il serait plus approprié de parler de direction de déplacement.

4.3.2/ Paramètres du PSO

Les paramètres du PSO peuvent être classés en deux catégories bien distinctes :

- ⇒ Les paramètres algorithmiques. Ces paramètres sont intrinsèques à l'algorithme PSO et apparaissent directement dans les équations contrôlant l'essaim (Équation 4.2). Certaines études ont été menées pour tenter de définir correctement ces paramètres mais beaucoup d'utilisateurs se basent encore sur des règles empiriques telles que le critère Poli [71].
- ⇒ Des paramètres dits paramètres d'adaptation.

4.3.2.1/ Paramètres algorithmiques

Ces paramètres sont détaillés dans ce qui suit.

Paramètres de confiance 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. Ces constantes sont multipliées par des nombres aléatoires r_1 et r_2 distribués suivant une loi de probabilité donnée, en générale une loi Normale.

Paramètre d'inertie ω est un paramètre d'inertie. Il permet d'établir un équilibre entre la recherche globale, capacité d'exploration de l'essaim, et la recherche locale, capacité d'exploitation de zones prometteuses locales de l'essaim. Différentes études empiriques ont été menées afin de mettre en évidence des règles pour le choix de ω . Il est préférable d'avoir une inertie décroissante au cours du temps. On considère ainsi 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. Il apparaît préférable de commencer avec une valeur d'inertie élevée, pour promouvoir l'exploration globale, pour ensuite la faire décroître, afin de privilégier l'exploration locale.

Dans les versions du PSO étudiées dans nos travaux, nous avons opté pour une version de ω décroissante dans le temps défini par l'équation suivante [82] :

$$\omega_k = \omega_{max} - (\omega_{max} - \omega_{min}) \frac{it_{max}}{k} \quad (4.3)$$

Tous les paramètres algorithmiques ainsi que leur signification et des valeurs usuelles sont résumés dans le tableau 4.1 .

Nombre de particules Deux aspects sont à prendre en considération pour choisir le nombre de particules de l'essaim : la taille de l'espace de recherche et le ratio entre les capacités de calcul et le temps dédié au calcul. Il n'y a pas vraiment de règles mathématiques pour choisir correctement ce paramètre même si on parvient à trouver quelques pistes empiriques. L'expérience reste la technique la plus efficace pour appréhender le choix de ce paramètre.

Le tableau 4.1 synthétise tous ces paramètres avec des valeurs standards issues de la littérature.

Notation	Signification	Valeur standard [41, 21, 50, 67]
N	Nombre de particules de l'essaim	$10 + 2\sqrt{D}$; [10; 150]
it_{max}	Nombre maximum d'itérations	$10000\frac{D}{N}$; [1000; 100000]
c_1	Comportement cognitif de la particule	[0.5; 3]
c_2	Aptitude sociale de la particule	[0.5; 3]
ω_{min}	Valeur minimale de l'inertie	[0; 0.5]
ω_{max}	Valeur maximale de l'inertie	[0.5; 2]

TABLE 4.1 – Paramètres algorithmiques du PSO

4.3.2.2/ Paramètres d'adaptation

Ces paramètres n'apparaissent pas directement dans l'équation d'actualisation de la position (Équation 4.2). Ils sont liés à la nature du problème : nombre de variables, espaces de recherche. Malgré ceci, il faut les contrôler au mieux et les adapter à chaque situation.

Domaine de recherche Il apparaît important de borner les valeurs des variables. Nous introduisons donc la valeur X_{min} (Resp. X_{max}) la valeur minimale (resp maximale) du domaine de recherche de l'algorithme.

Paramètre de contrôle de la vitesse La vitesse des particules doit être contrôlée afin d'éviter que l'algorithme n'oscille trop. 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. Ce contrôle est géré par le paramètre V_{factor} qui permet de limiter la vitesse sur chaque dimension et ainsi de contrôler l'explosion du système.

Dans le cas du PSO utilisé et développé pour nos travaux, la vitesse V des particules n'est pas limitée. Cependant la vitesse initiale est tirée aléatoirement entre des bornes. Ces bornes V_{min} et V_{max} sont basées sur les limites du problème X_{min} et X_{max} . Ces limites sont calculées comme dans l'équation suivante, où V_{factor} est un coefficient à donner :

$$\begin{cases} V_{max} = V_{factor} \cdot (X_{max} - X_{min}) \\ V_{min} = -V_{max} \\ V_{init} = rand(V_{min}, V_{max}) \end{cases} \quad (4.4)$$

Tous les paramètres d'adaptation ainsi que leur signification et des valeurs usuelles sont résumés dans le tableau 4.2.

Notation	Signification	Remarques
D	Dimension du problème	$D \in [1, 50]$ [?, 77]
X_{min}	Valeur minimale du domaine de recherche	Associée au problème
X_{max}	Valeur maximale du domaine de recherche	Associée au problème
V_{min}	Vitesse minimale atteinte par une particule	Associée au problème
V_{max}	Vitesse maximale atteinte par une particule	Associée au problème
V_{factor}	Coefficient de contrôle de vitesse	[0.01, 0.5]

TABLE 4.2 – Paramètres d'adaptation du PSO

4.3.3/ Critère d'arrêt et convergence

La convergence vers la solution optimale globale n'est pas garantie dans tous les cas de figure même si les expériences dénotent la grande performance de cette approche, ce qui est par ailleurs le cas pour la plupart des métaheuristiques! De ce fait, il est fortement conseillé de doter l'algorithme d'une porte de sortie : l'algorithme PSO s'arrête après avoir atteint un critère d'arrêt.

Le critère d'arrêt peut être différent suivant le problème posé. Si l'optimum global est connu a priori, on peut définir une erreur acceptable comme critère d'arrêt. Sinon, il est habituel de fixer un nombre maximum d'évaluations de la fonction objectif ou un nombre maximum d'itérations comme critère d'arrêt. Cependant, selon le problème posé et les exigences de l'utilisateur, d'autres critères d'arrêt peuvent être utilisés. Différents critères peuvent aussi être combinés.

4.3.4/ Topologie de voisinage

Un dernier point fondamental pour un fonctionnement optimal de l'algorithme PSO concerne la topologie de voisinage [49, 64]. On entend par-là, la manière dont sont choi-

sies les particules qui définissent le voisinage de chacune d'entre elles, et donc dans quelle mesure l'information circule dans l'essaim. On peut aussi parler de graphes d'influences.

Dans la plupart des cas, le voisinage est un voisinage social, sans contrainte de localisation des voisins. Chaque particule communique avec un sous-groupe organisé en topologie particulière. Cependant, certains auteurs font allusion à des voisinages dits géographiques avec une contrainte de localisation sur les voisins. De très nombreux types différents de topologies ont été développés et sont proposés dans la littérature : statique, dynamique. On peut trouver une bonne synthèse de tout ceci dans [28, 60, 62].

Les premières topologies proposées dans la littérature sont statiques. Dans ce cas, chaque particule est toujours influencée par les mêmes autres particules tout au long du processus d'optimisation. Pour illustration, dans l'équation (4.2), la grandeur $\mathbf{G}_{i,best}^k$ est la meilleure valeur donnée par l'ensemble des particules informatrices appartenant au voisinage de la particule considérée i à l'itération k . Dans la version initiale du PSO [26], la topologie de voisinage est dite globale (on dit aussi *GBest*) : une particule est informée par l'ensemble des particules formant l'essaim (Figure 4.5a). Cette topologie a l'avantage de converger rapidement, mais peut défavoriser le processus de diversification au profit de l'intensification avec le risque de converger vers un minimum local. Dans la version *lBest*, chaque particule est liée de manière fixe avec deux autres particules de l'essaim, définies arbitrairement à l'initialisation du processus (Figure 4.5b).

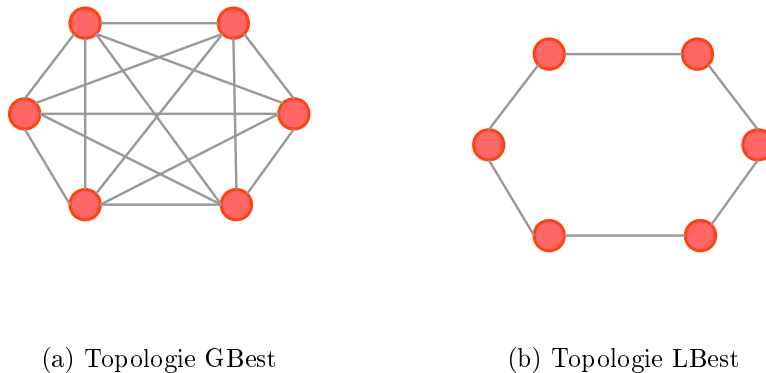


FIGURE 4.5 – Deux topologies célèbres

Après avoir étudié les rouages et les limites de l'approche PSO, nous proposons un résumé de nos recherches autour du PSO en commençant par la prise en compte des contraintes. Ensuite, nous nous sommes focalisés sur la méthode en elle-même via deux voies d'amélioration originales. La première concerne un critère d'arrêt. Nous verrons ensuite comment améliorer l'accélération de la convergence avec un saut d'essaim.

4.4/ Prise en compte des contraintes

Dans tout ce qui était vu et fait précédemment en terme de résolution dans ce chapitre, jamais le problème des contraintes n'a été abordé. En général, la plupart des problèmes d'optimisation en mécanique sont formulés avec des contraintes de deux natures différentes.

- La première catégorie concerne les bornes limites sur les variables d'optimisation, autrement dit les domaines d'admissibilités des variables d'optimisation. Ceci peut être défini comme suit :

$$X_i \in [X_i^{min}, X_i^{max}] \quad \forall i = 1, \dots, D \quad (4.5)$$

Dans cette expression, X_i^{min} (respectivement X_i^{max}) est la limite inférieure (respectivement supérieure) des valeurs de la variable X_i .

- La deuxième catégorie concerne les contraintes spécifiques au problème en lui même (contraintes, déplacements par exemple en mécanique) que nous noterons de manière générale :

$$g_j(\mathbf{X}) \leq 0 \quad \text{avec} \quad j = 1, \dots, p \quad (4.6)$$

Dans le cadre de l'algorithme PSO, il est important de distinguer les deux situations, car elles vont être traitées de manière différente [63]. En effet, si une particule n'est pas dans son domaine admissible suite à un déplacement, à quoi bon traiter le problème mécanique ! Dans ce contexte, nous nous sommes intéressés à l'adaptation d'une approche originale à la prise en compte des limites sur les bornes des variables : la méthode Harmony Search scheme.

4.4.1/ Harmony Search scheme

Pour prendre en compte les variables sur les bornes limites, nous avons utilisé et adapté la méthode Harmony Search scheme à l'algorithme PSO. Cette méthode a été introduite dans [57] et permet de gérer les particules qui, en se déplaçant, sortent du domaine de recherche. Cette technique dérive de l'algorithme Harmony Search (HS) [55]. Dans cette approche, les vecteurs de variables admissibles sont stockés par une mémoire (Harmony Memory : HM). Dès qu'une variable sort du domaine de recherche, elle est remplacée aléatoirement par une particule de la mémoire HS. L'algorithme PSO se prête bien à une adaptation de cette approche. En effet, PSO conserve en mémoire des valeurs a priori acceptables : les meilleures valeurs personnelles atteintes par l'essaim \mathbf{P}_{best} . Dans l'adaptation proposée, les valeurs des variables sorties de leur domaine de recherche sont remplacées aléatoirement par des valeurs choisies dans les mémoires personnelles des particules.

Cette approche a été testée sur des problèmes standards de mécanique de structures, l'optimisation de treillis. Pour plus de détails, nous renvoyons le lecteur à la référence [?].

4.4.2/ Méthode de pénalisation

Pour les contraintes spécifiques au problème, nous avons opté pour la méthode de pénalisation. Le fonctionnement de ce méthode est classique. Elle procède au remplacement du problème avec contraintes par un problème sans contraintes en intégrant les contraintes dans une nouvelle fonction F_ϵ .

$$(P_\epsilon) \left\{ \min F_\epsilon(\mathbf{X}) \right.$$

avec

$$F_\epsilon(\mathbf{X}) = F(\mathbf{X}) + \epsilon p(\mathbf{X})$$

Nous avons utilisé différentes expressions de cette fonction de pénalisation.

4.4.3/ Fonction en cascades

Un nouveau type de méthode de prise en compte des contraintes a été créée : la fonction-objectif en cascade. Cette méthode se rapproche de la relaxation des contraintes [12] et de la propagation des contraintes [78]. La relaxation de contraintes consiste à réduire l'espace de recherche pour éviter les zones ne respectant pas les contraintes. Dans le cas de la propagation de contraintes le problème est résolu par un arbre de décision où les contraintes sont ajoutées les unes après les autres. Dans le cas de la fonction-objectif en cascade, les contraintes sont ordonnées et testées les unes après les autres. En fonction de contraintes respectées ou non, une fonction-objectif en particulier est choisie.

Cette approche a été testée pour un domaine particulier : l'optimisation du chargement d'un container. Ce travail a été effectué en collaboration avec la société my-OCCS [?]

4.5/ Un nouveau critère d'arrêt et saut d'essai

Nous nous sommes donc attachés à développer une approche pour gérer au mieux l'arrêt de l'algorithme PSO dans le contexte de l'optimisation en conception ainsi qu'une approche pour améliorer la convergence. Ce travail a été développé en collaboration avec Sébastien Salmon lors de sa thèse [79]. Ces approches ont été implantées dans le logiciel my-OCCS [?] et mises à disposition sur la plateforme Scilab [80]. Ceci a aussi donné lieu à diverses publications [?, ?, ?, ?].

4.5.1/ Un nouveau critère d'arrêt : PSO-Radius

Pour un PSO classique, le critère d'arrêt le plus commun est le nombre d'itérations maximal autorisé mais peut aussi être, dans le cadre d'une modification, la distance au minimum recherché. Dans notre cadre de travail de l'optimisation en conception, des limitations technologiques peuvent intervenir sur les variables de conception. L'idée principale est d'éviter un nombre important d'itérations inutiles en tenant compte de ces limitations.

En effet, il se peut que la précision de la particule conduisant au meilleur résultat ne puisse pas nécessairement être reproduite ou conçue/fabriquée. Ainsi, est-il possible d'imaginer un critère d'arrêt sur la précision de la mesure du rayon de l'essaim ? Ce rayon de l'essaim est calculé en évaluant la distance de l'ensemble de l'essaim de particules par rapport au leader de la particule à l'aide d'une norme 2 standard. Ensuite, si la distance maximale (Norme infinie) reste inférieure à un critère défini par l'utilisateur pendant un certain nombre d'itérations, l'algorithme est alors arrêté (Figure 4.6). Cette modification porte le nom de "PSO-Radius"

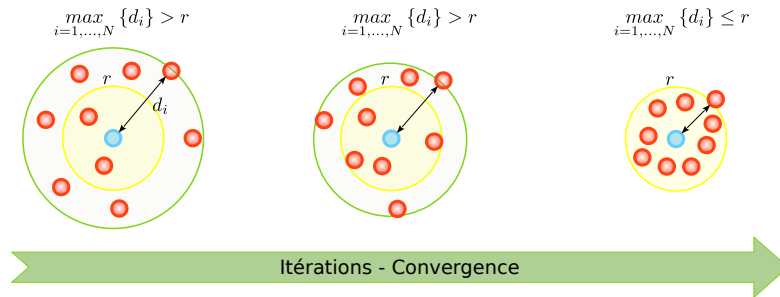


FIGURE 4.6 – Un nouveau critère d'arrêt - PSO-Radius

4.5.2/ Accélération de convergence par saut de l'essaim

Une amélioration a été développée pour accélérer la convergence basée sur un saut d'essaim. Cette approche repose sur deux idées principales :

- ❖ A chaque itération k , la particule leader a la possibilité d'envoyer au hasard de nouvelles particules pour explorer rapidement l'espace, les raptors (Figure 4.7). La vitesse de ces raptors est choisie plus grande que la vitesse des particules classiques dans des directions définies aléatoirement.
- ❖ Si un raptor trouve une meilleure position que la meilleure globale, on procède à un saut de l'essaim, suivant un vecteur saut comme indiqué sur la figure 4.7 : le Carrier saute à l'emplacement du meilleur Raptor, cela permet de définir le vecteur saut. Le reste de l'essaim saute donc dans l'espace suivant ce vecteur. La géométrie de l'essaim est alors conservée permettant de s'échapper d'un minimum local tout en conservant une bonne diversité de celui-ci.

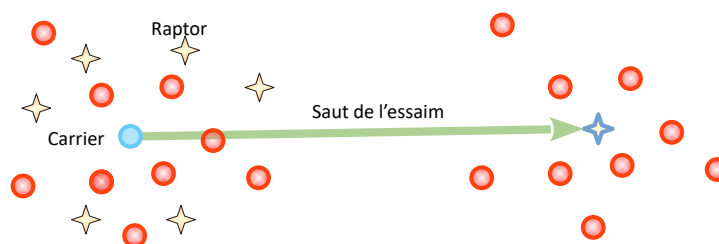


FIGURE 4.7 – Saut d'essaim

4.5.3/ Validation et application aux matériaux composites à matrice céramique

Ces deux approches ont été testées et validées sur des fonctions mathématiques usuelles mais aussi sur des cas d'applications plus complexes. Pour la validation sur des fonctions

mathématiques usuelles issues de benchmarks, nous renvoyons le lecteur à la référence [?].

Dans le contexte de travail de ce mémoire, optimisation pour la conception, il nous est apparu plus pertinent d'illustrer la validation de ces approches par un cas plus concret. L'exemple présenté dans la suite a été abordé pendant l'encadrement des travaux de thèse de Yingjie Xu [90]. Pendant sa thèse, Yingjie Xu [90], s'est intéressé à l'optimisation des matériaux composites à matrice céramique en particulier aux composites à matrice de carbure de silicium renforcée par des fibres de carbone (C/SiC). Des études particulières ont été menées concernant les propriétés de ces matériaux : prédiction des propriétés mécaniques (élastiques), analyses thermiques (optimisation des contraintes thermiques), simulation de l'oxydation à haute température. Dans le cas de cet encadrement, mon apport a concerné en particulier la partie optimisation. Dans ce contexte, les versions améliorées du PSO introduites dans les sections précédentes ont pu être testées pour minimiser les contraintes résiduelles thermiques (TSR).

4.5.3.1/ Présentation du problème

La prise des contraintes résiduelles thermiques (contraintes d'origine thermique dans les fibres et la matrice) joue un rôle majeure dans le comportement des composites à matrices céramiques. Leurs valeurs influencent fortement la contrainte de microfissuration de la matrice. On cherche donc à minimiser cette contrainte résiduelle thermique (CRT). Lors de la fabrication des composites à matrice céramique, les cycles thermiques associés aux propriétés matériaux de chacun des composants engendrent l'apparition de contraintes résiduelles thermiques (Figure 4.8). Ces contraintes sont la source de fissurations matricielles. Notre problématique d'optimisation est la suivante : comment diminuer cette contrainte résiduelle thermique ? Une solution consiste à dimensionner correctement les épaisseurs des différentes couches de l'interface. En effet, le procédé de fabrication permet une gestion de ces dimensions.

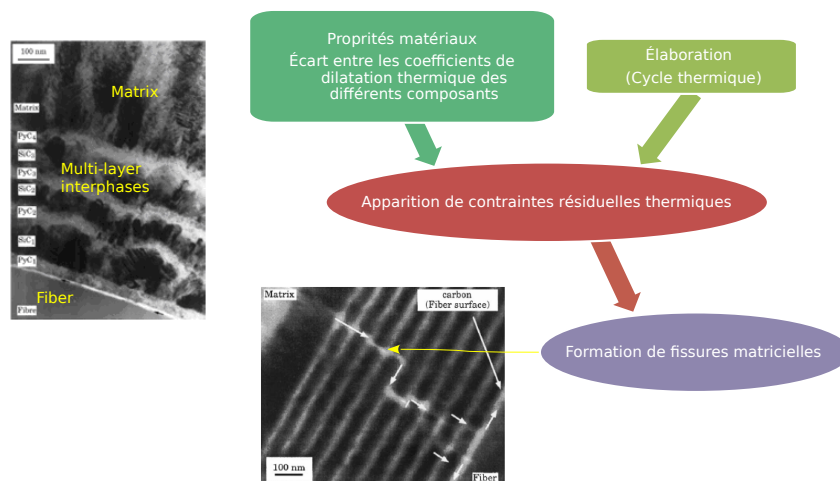


FIGURE 4.8 – Problématique : contrainte résiduelle thermique pour les CMC

L'étude d'optimisation présentée ici est réalisée sur une fibre SiC unidirectionnelle composite à matrice céramique SiC renforcée avec quatre couches alternées d'inter-phase (PyC/SiC/PyC/SiC). D'un point de vue mathématique ce problème s'écrit simplement :

$$\begin{cases} \min_{i=1,\dots,4} CRT(d_i) \\ 0.3 \leq d_i \leq 0.6 \end{cases} \quad (4.7)$$

Dans cette expression d_1, d_2, d_3 et d_4 sont les épaisseurs des couches de l'interphase.

Le diamètre de la fibre de SiC est de $10\mu\text{m}$ et l'épaisseur de la matrice de SiC est $10\mu\text{m}$. La borne supérieure de l'épaisseur de chaque couche d'interphase est $0,6\mu\text{m}$. En pratique, les épaisseurs des couches de l'interface sont généralement limitées à $0,1\mu\text{m}$. Pour éviter tout souci notamment au niveau de l'oxydation, la borne inférieure de l'épaisseur de chacune des couches a été, dans ce problème, limitée à $0,3\mu\text{m}$. La contrainte résiduelle thermique est ici issue d'une modélisation par éléments du composite, associée au concept de volume élémentaire représentatif. Nous ne rentrons pas dans les détails de cette modélisation dans ce rapport. Pour plus d'informations, nous renvoyons le lecteur à la thèse de Yingjie Xu [90]. A titre indicatif, le modèle éléments finis est illustré sur la figure 4.9. Ce modèle élément finis a été connecté aux algorithmes d'optimisation introduits dans les sections précédentes.

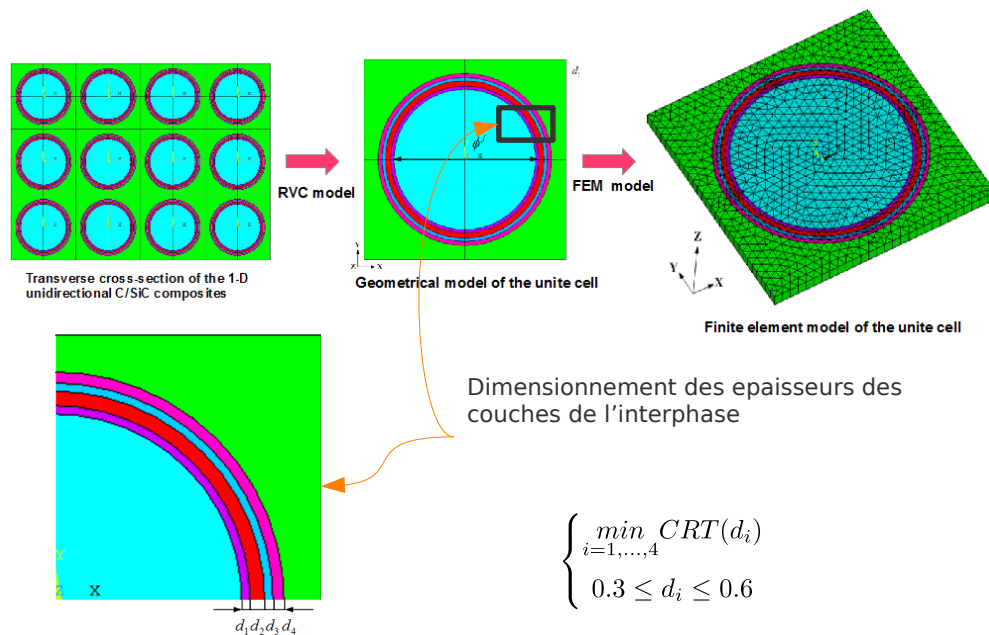


FIGURE 4.9 – Modélisation EF des CMC

4.5.3.2/ Résultats

L'utilisation d'un algorithme permettant de réduire le nombre d'appels à la fonction objectif est parfaitement adaptée au présent problème. En effet, le calcul éléments finis "composite"

est coûteux et réduire le nombre de simulation par éléments finis paraît indispensable.

Les résultats obtenus par nos approches sont comparés à ceux obtenus par une PSO classique afin d'évaluer leurs performances. La figure 4.10 illustre l'évolution de la CRT au cours des itérations de l'algorithme PSO.

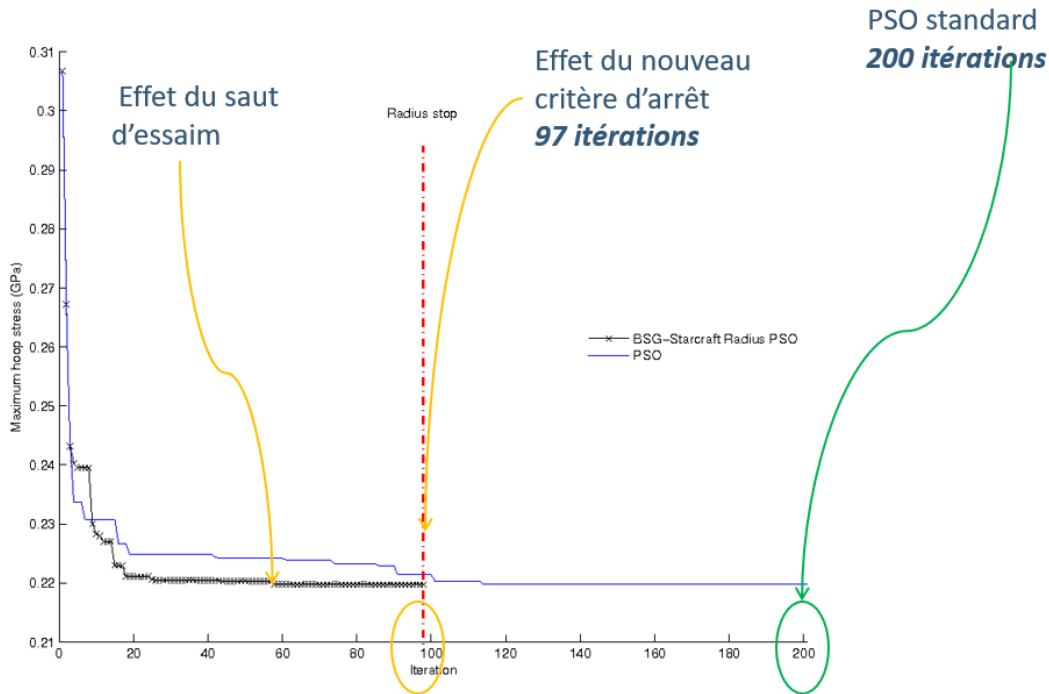


FIGURE 4.10 – Évolution de la CRT : comparaison PSO standard/Radius

Il est à noter que la configuration optimale obtenue dans tous les cas est la suivante :

$$(0,6\mu\text{m}, 0,6\mu\text{m}, 0,3\mu\text{m}, 0,6\mu\text{m})$$

correspondant à une CRT de 22MPa .

Le gain de nos approches se fait sur la rapidité de convergence grâce au critère d'arrêt et la qualité de la solution via le saut de l'essaim. La figure 4.10 illustre parfaitement cet aspect. Avec un PSO standard et un critère d'arrêt fixé à 200 itérations, les 200 itérations sont nécessaires pour atteindre une solution identique à celle obtenue avec notre approche et la modification du critère d'arrêt. Dans notre cas, seulement 97 itérations sont requises ! De plus, nous voyons clairement sur cette figure que la CRT décroît beaucoup plus rapidement qu'avec un algorithme standard.

Méthodes	Itérations	Temps
PSO amélioré	90	8 h
PSO standard	200	12 h

TABLE 4.3 – Comparaison PSO standard / PSO amélioré

Le tableau 4.3 résume les différences majeures et les gains générés par nos développements. A noter, une forte chute du temps de calcul (tests réalisés sur la même machine) associée

à la diminution du nombre d'itérations et donc à la diminution du nombre de simulations éléments finis réalisées.

Ces travaux ont été publiés dans [?].

4.6/ Algorithme Inverse-PageRank-PSO : PSO à topologie évolutive

Cette partie a pour but de présenter une nouvelle évolution du PSO basée sur l'intelligence de l'algorithme. L'intelligence est ici décrite comme la capacité de l'algorithme à s'adapter au cours des itérations, c'est à dire à adapter son action en fonction des résultats qu'il obtient ou de l'évolution du domaine de recherche. Cette nouvelle méthode métaheuristique intelligente, Inverse-PageRank-PSO (I-PR-PSO) est basée conjointement sur l'algorithme d'optimisation par essaim particulière présenté dans la section 4.3, et sur l'algorithme PageRank® développé par les concepteurs du moteur de recherche Google® [13, 65].

Après s'être intéressé à des améliorations du PSO concernant les critères d'arrêt et les sauts d'essaims, il nous est apparu original d'explorer une nouvelle piste d'amélioration : travailler sur la topologie de l'essaim. L'idée principale de cette amélioration est la suivante : mettre en place une topologie de la population évolutive à chaque itération afin de redéfinir les influences de chaque particule sur le reste de l'essaim.

Ce travail a été initié pendant l'encadrement de la thèse de Noëlie Di Cesare [23] et a donné lieu à différentes publications [?, ?]

4.6.1/ Relation topologie - chaines de Markov à temps discret

Avant de décrire le fonctionnement de ce nouvel algorithme, il est important de souligner le lien fort entre topologie de voisinage et chaîne de Markov.

En effet, comme évoqué dans la section 4.3.4, les particules sont liées entre elles, et ce lien indique l'influence qu'elles ont les unes sur les autres. Les particules liées peuvent être considérées comme une chaîne de Markov à temps discret. La qualité des solutions définit la transition de probabilité de la chaîne, qui détermine l'influence des particules sur les autres. On pourrait envisager de pondérer ces liens, c'est à dire de pondérer l'influence que les particules ont les unes sur les autres. On définirait ainsi un graphe orienté, avec des liens pondérés. On peut donc considérer que cette pondération représente aussi une probabilité de transition d'un état à un autre.

C'est à partir de ce constat que l'algorithme (I-PR-PSO) a été développé. Nous proposons ici une description de la philosophie générale de l'approche. Pour plus de détails, nous renvoyons le lecteur à notre article [?].

Une chaîne de Markov à temps discret est un système mathématique qui décrit les transitions d'un état à un autre, donnés dans un espace d'états discrets. Cette méthode mathématique est caractérisée par la propriété de Markov : la prédiction du futur n'est déterminée ni par l'état présent, ni par les précédents états passés. Ainsi, une chaîne de Markov peut aussi être décrite par une séquence de graphes orientés, dans lesquels les arêtes sont pondérées par les probabilités de se déplacer de l'état au temps n , à l'état suivant au temps $n + 1$. On peut ainsi décrire ce processus plus simplement, grâce à une matrice de transition de probabilités. Si la probabilité de passer de l'état i à l'état j en un pas de temps est donnée par $P(i|j) = C_{ij}$, alors la matrice de connectivité stochastique est donnée par $C_{i,j}$ où i est

l'indice des lignes, et j l'indice des colonnes de la matrice. La somme des probabilités de passer de l'état i à tous les autres est toujours 1, on a $\sum_j C_{i,j} = 1$.

On note aussi que la probabilité de passer d'un état à un autre en k pas de temps discrets est donnée par \mathbf{C}^k . Ainsi, on peut définir un vecteur de probabilité stationnaire $\boldsymbol{\pi}$ que l'on peut définir comme le régime stationnaire du processus de Markov à temps discret considéré. Pour une matrice de connectivité stochastique dont tous les termes sont strictement positifs, le vecteur de probabilité stationnaire $\boldsymbol{\pi}$ est unique et peut être calculé grâce à l'équation suivante :

$$\forall i \quad \lim_{k \rightarrow \infty} (\mathbf{C}^k)_{i,j} = \pi_j \quad (4.8)$$

Dans l'algorithme PageRankGoogle[®], de Google[®], la chaîne de Markov considérée est connue et représente l'ensemble des pages web et des liens entre elles. Les probabilités de transitions entre chacune d'entre elles peuvent être calculées, et le vecteur PageRank, c'est à dire l'état stationnaire du processus de Markov, est ainsi déduit [54]. Dans l'approche développée ici, on considère le graphe orienté défini par les nœuds que sont les particules et les arêtes que sont les liens d'influence entre elles. Ces liens sont orientés de telle sorte que l'arête allant de la particule i à la particule j définit l'influence de la particule i sur la particule j .

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 \mathbf{G}_{best} , est défini comme suit :

$$\pi_{target}^T(1, i) = \left| \frac{F(\mathbf{G}_{best}) \times 100}{F(\mathbf{G}_{best}) - F(\mathbf{X}_k) + \varepsilon} \right| \forall k = 1, \dots, n \quad (4.9)$$

où $F(\mathbf{X})$ est la valeur de la fonction objectif de la particule \mathbf{X} , et n représente le nombre de particules de l'essaim. Le paramètre ε est utilisé pour éviter d'annuler le dénominateur.

4.6.2/ Principe de base

De cette manière, la topologie de la population est évolutive et change à chaque itération, afin de redéfinir les influences de chaque particule sur le reste de l'essaim. Les équations de changements des positions des particules dans l'espace des solutions sont donc les suivantes :

$$\begin{cases} \mathbf{V}_i^{k+1} = \omega_k \mathbf{V}_i^k + c_1 r_1 (\mathbf{P}_{i,best}^k - \mathbf{X}_i^k) + c_2 r_2 \sum_{j=1}^{j=n} C_{ji} (\mathbf{P}_{j,best}^k - \mathbf{X}_i^k) \\ \mathbf{X}_i^{k+1} = \mathbf{V}_i^{k+1} + \mathbf{X}_i^k \end{cases} \quad (4.10)$$

où C_{ij} est le terme de la $i^{\text{ème}}$ ligne, $j^{\text{ème}}$ colonne de la matrice des transitions stochastiques précédemment définies. La particule i est influencée par toutes les particules de l'essaim, et leur influence respective est donnée par les composantes de la colonne i de \mathbf{C} .

4.6.3/ Validation sur un ensemble de fonctions tests

L'algorithme proposé a été testé sur différentes fonctions de référence issues de la littérature et des différents benchmarks existants. Nous reparlerons de ces fonctions dans les chapitres à venir. Dans le cas présent, 12 fonctions ont été retenues pour la validation de l'algorithme (Tableau 4.4). Le choix a été fait en sorte de balayer le plus de difficultés mathématiques possibles : fonction unimodale vs multimodale, minimum global vs minimum local, paysage de fonctions différents. Le détail des informations concernant ces fonctions peut être trouvé dans notre article [?].

Fonction	Notation
f_1	Ackley
f_2	Griewank
f_3	Rastrigin
f_4	Rosenbrock
f_5	Sphere
f_6	Rotated hyper ellips
f_7	Shifted Rastrigin
f_8	Shifted Rosenbrock
f_9	Shifted Sphere
f_{10}	Shifted Ackley
f_{11}	Bohachevsky
f_{12}	Schwefel's problem 1.2

TABLE 4.4 – Fonctions tests retenues pour la validation de l'algorithme I-PR-PSO

Une étude statistique a été réalisée afin d'obtenir des résultats suffisants pour prouver l'efficacité de I-PR-PSO. Sur chaque fonction, 100 exécutions ont été effectuées en dimensions 10, 20, 30 et 50, avec 50 particules. Comme il a été montré dans la littérature que PSO peut être plus efficace que d'autres méthodes métaheuristiques en grandes dimensions [48] [85], cette recherche se concentre particulièrement sur la comparaison entre différentes variantes de PSO. Ainsi, I-PR-PSO a été comparé à trois versions différentes de PSO classique, c'est-à-dire avec la topologie *GBest* présentée précédemment, la topologie *LBest*, et la topologie 4-clusters, avec les mêmes paramètres de calcul. Les paramètres de calcul sont donnés dans le tableau 4.5.

Notation	Valeur
N	50
it_{max}	600
c_1	2
c_2	2
ω	0.4
V_{min}	-10
V_{max}	10
Seuil de convergence de l'algorithme PageRank inverse	$\epsilon_{PR} = 1E - 03$
Nombre maximal d'itérations du processus PageRank inverse	6000

TABLE 4.5 – Paramètres de calculs pour la validation de I-PR-PSO

Il ressort de ces comparaisons deux points importants. Le premier concerne la robustesse de l'algorithme I-PR-PSO. Cet algorithme a une forte capacité à trouver l'optimum global de la fonction objectif considérée plus que ses pairs, et que l'I-PR-PSO atteint un équilibre entre les phases d'exploration et d'exploitation nécessaires pour trouver l'optimum global en particulier en grandes dimensions. Ceci est notamment illustré sur la Figure 4.11 en dimension 50. Le deuxième point concerne son efficacité. Un critère indépendant et universel permettant d'évaluer ce point est le nombre d'évaluations de la fonction objectif. On trace l'historique d'une exécution sur un graphe qualité vs convergence. L'algorithme I-PR-PSO est plus rapide que les autres algorithmes testés en termes du nombre d'évaluations de la fonction objectif, nécessaires à la convergence. Ainsi, I-PR-PSO pourrait s'avérer très efficace dans les applications mécaniques lorsque la méthode des éléments finis est utilisée car, dans ce contexte, les évaluations de la fonction coût sont très coûteuses. Ce point est très net sur la figure en dimension 50 pour l'ensemble des fonctions testées (Figure 4.12).

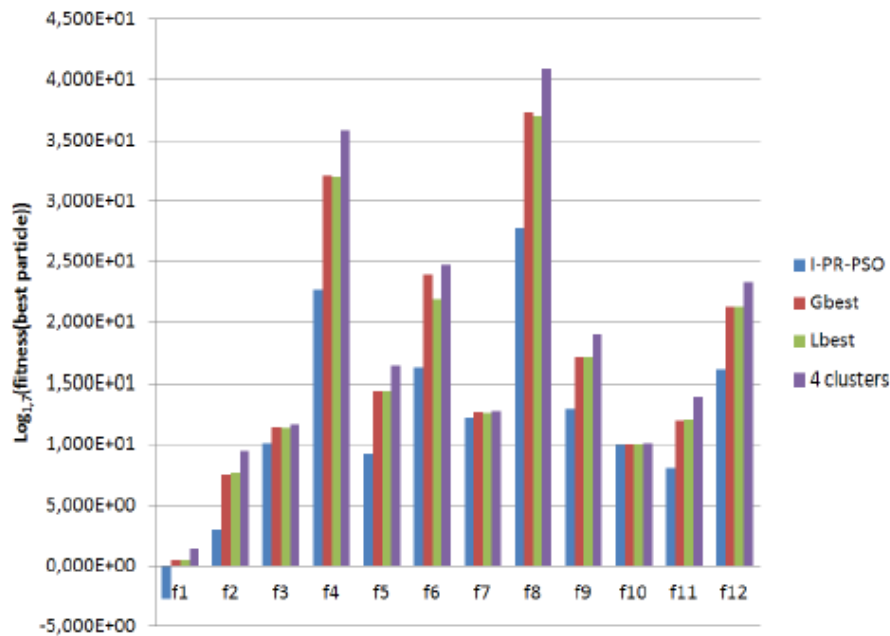
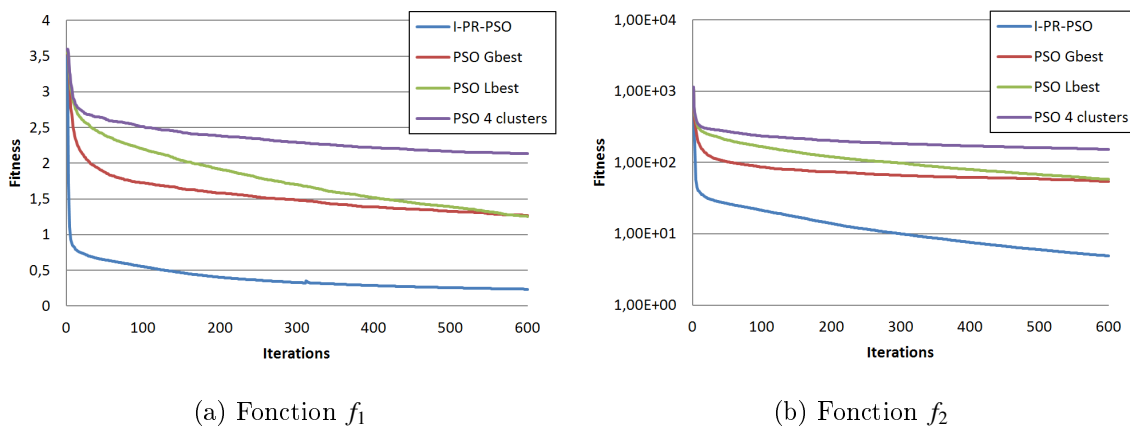
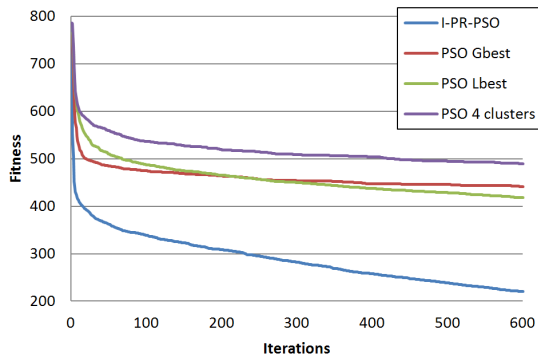
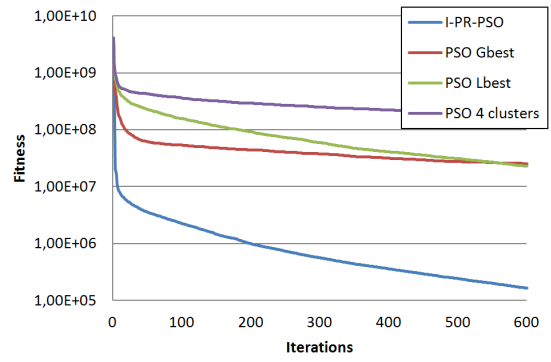


FIGURE 4.11 – Minimum après 600 itérations et $D = 50$ sur l'ensemble des fonctions tests

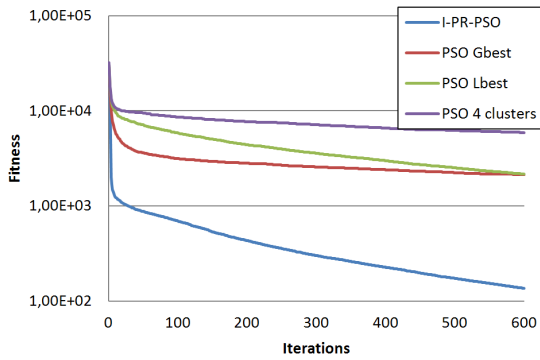




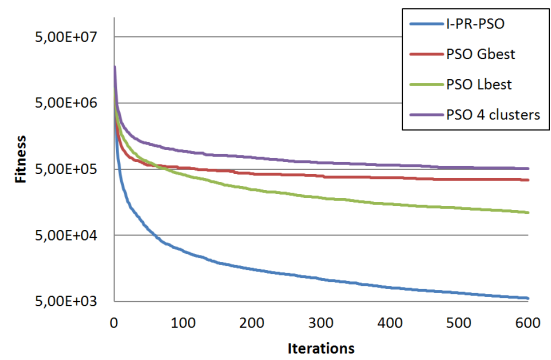
(c) Fonction f_3



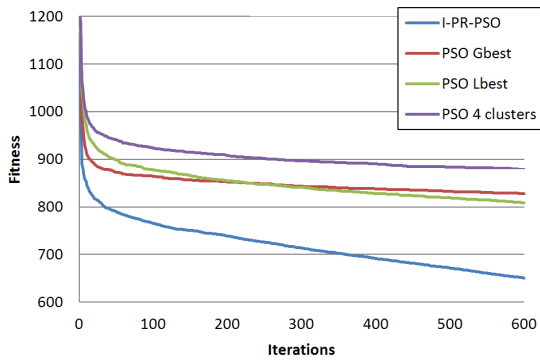
(d) Fonction f_4



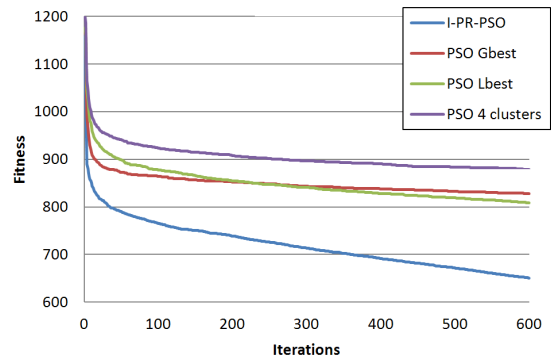
(e) Fonction f_5



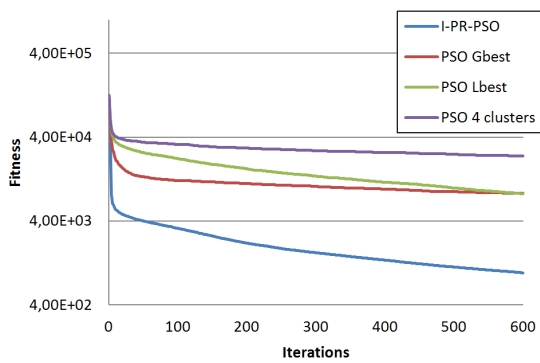
(f) Fonction f_6



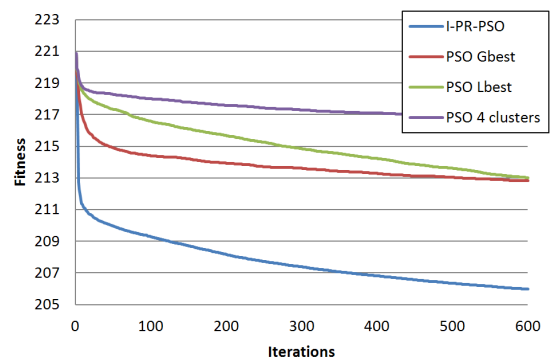
(g) Fonction f_7



(h) Fonction f_8



(i) Fonction f_9



(j) Fonction f_{10}

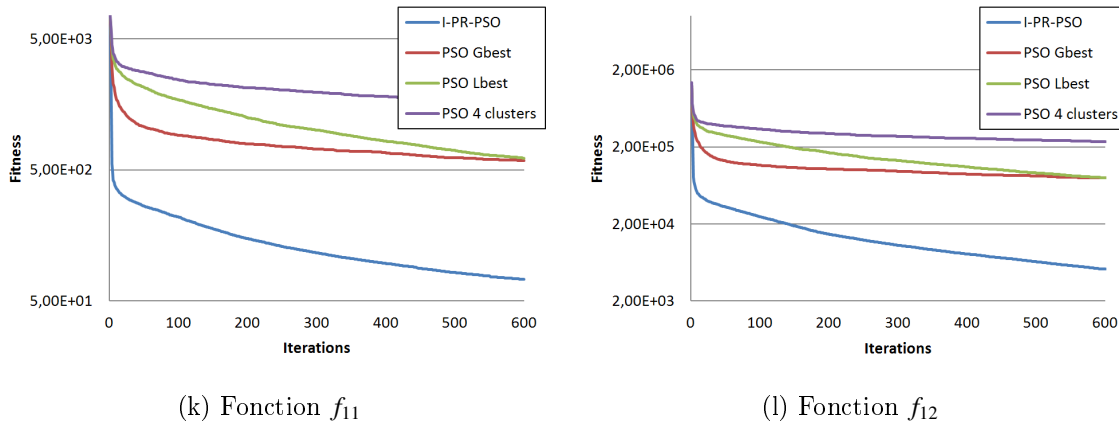


FIGURE 4.12 – Graphe de convergence de l’algorithme I-PR-PSO en dimension 50

Remarques Pour ne pas alourdir le document, nous ne présentons ici qu’une partie des résultats. Ces résultats peuvent être trouvés en intégralité dans [?].

4.6.4/ Validation en optimisation de structure mécanique

Valider sur des fonctions tests connues et éprouvées est fondamental mais dans notre souci d’optimisation en conception, une validation sur des cas mécaniques, en optimisation dite de structure, avec contraintes, est fondamental. Nous avons donc appliqué cette méthode en optimisation de structures en particulier à des treillis mécaniques. Ce sont des problèmes classiques et bien connus en optimisation de structures (optimisation de forme) permettant ainsi une bonne validation de l’algorithme. Les difficultés associées à ces problèmes sont de nature différentes. On peut citer la non-convexité des fonctions objectifs, la prise en compte des contraintes mécaniques.

Dans ce cas, la fonction objectif est la masse de la structure, soumise à des limitations sur les contraintes mécaniques axiales dans les barres et les déplacements des nœuds, et les variables d’optimisation sont les sections des barres. Le problème d’optimisation considéré peut donc être formulé de la manière suivante :

$$\left\{ \begin{array}{ll} \min V(\mathbf{X}) & \text{et} \quad \mathbf{X} = (X_1, \dots, X_D) \\ \text{soumis à} \quad g_j(\mathbf{X}) \leq 0 & \text{avec} \quad j = 1, \dots, p \\ X_i \in [X_i^{\min}, X_i^{\max}] & \forall i = 1, \dots, D \end{array} \right. \quad (4.11)$$

où

- ❖ V est le volume de la structure,
- ❖ X_i est la section de la barre i ,
- ❖ $g_j(\mathbf{X})$ sont des contraintes de flambage et de déplacement,
- ❖ X_i^{\min} (resp X_i^{\max}) est la valeur minimale (maximale) tolérable pour la section i .

4.6.4.1/ Résultats

Nous présentons ici les résultats obtenus dans le cas d’une étude de deux systèmes de treillis [5] : treillis à 10 barres et treillis à 22 barres (Figure 4.13).

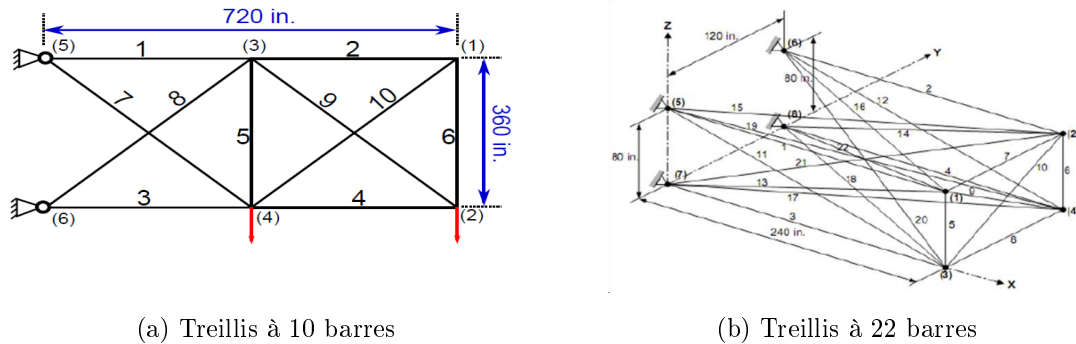


FIGURE 4.13 – Validation sur des treillis

Les résultats obtenus sont résumés dans le tableau 4.6. Comme on le voit ici, les résultats obtenus sont convaincants puisque meilleurs que ceux donnés dans la littérature, non seulement parce que la valeur de la fonction objectif est plus faible, mais aussi et surtout parce que toutes les contraintes d'optimisation sont vérifiées.

Treillis	22 barres			Treillis	10 barres
	Cas de chargement				
Variables (in2)	(1)	(2)	(3)	Variables (in2)	Chargement (1)
A1	2,9073	2,1465	0,60027	A1	31,1378
A2	1,2738	1,1947	2,6978	A2	0,10012
A3	0,11017	2,0717	0,5969	A3	23,3941
A4	0,11644	1,0714	1,4318	A4	15,6869
A5	2,8741	0,90091	2,1599	A5	0,10049
A6	1,5886	2,2245	2,3347	A6	0,45539
A7	2,0237	1,8172	0,14755	A7	7,5238
Masse (lb)	993,48	849,25	642,0941	A8	20,8423
				A9	20,9179
				A10	0,10249
				Masse (lb)	5065,84

TABLE 4.6 – Résultats de l'optimisation de treillis

4.7/ Conclusions et perspectives

Ce chapitre a été consacré à l'étude détaillée d'une méthode d'optimisation métaheuristique particulière : l'algorithme PSO. Mes années de recherche ont conduit à différentes voies d'amélioration que j'ai pu développées notamment par le biais d'encadrement de thèses et stages :

- ❖ Un nouveau critère d'arrêt,
- ❖ Une stratégie pour améliorer la convergence,
- ❖ Une nouvelle version du PSO à topologie évolutive.

Les différentes approches ont été validées via des fonctions mathématiques tests de référence issues de la littérature et sur des exemples issus de la mécanique.

C'est en effectuant ces différents tests de validation et en cherchant les fonctions mathématiques de références qu'une interrogation fondamentale est apparue : comment certifier avec certitude nos approches ? Existe-t-il des outils permettant de comparer de manière universelle tous les algorithmes ? C'est ce problème que nous allons aborder dans le chapitre suivant.

COMPARAISON ET PERFORMANCES D'ALGORITHMES D'OPTIMISATION

Sommaire

5.1	Contexte et objectifs	45
5.2	Structure et contrôle du benchmark	47
5.2.1	Paramètres de contrôle	47
5.2.2	Fonction objectif	48
5.2.3	Dimension	48
5.2.4	Nombre maximum d'évaluations de la fonction objectif	48
5.2.5	Run et séries	48
5.2.6	Critère d'arrêt d'un run d'un cas d'optimisation	49
5.3	Outils de comparaison et indicateurs de performances	51
5.3.1	Résultat d'un run	51
5.3.2	Résultat d'un cas d'optimisation	52
5.3.3	Score global	53
5.3.4	Sous-scores	55
5.4	Application et résultats	55
5.4.1	Score global	56
5.4.2	Sous-scores	57
5.5	Conclusion	59

Notation

Notations générales	
D	Dimension/Nombre de variables du problème d'optimisation
F	Fonction-objectif
X	Variable d'optimisation
Paramètres de contrôle du benchmark	
FES	Nombre d'évaluations autorisées pour résoudre le problème d'optimisation (critère d'arrêt)
$MaxFES$	Coefficient utilisé pour calculer FES en fonction de D
N_D	Nombre de dimensions utilisées par le benchmark
N_F	Nombre de fonctions objectif
N_M	Nombre de $MaxFes$
N_C	Nombre total de cas d'optimisation du benchmark
N_T	Nombre de runs pour un cas d'optimisation (étude statistique)
N_R	Nombre total de runs du benchmark
Cas d'optimisation	
V_G	Résultat brut obtenu après un run pour un cas d'optimisation
E_G	Nombre brut d'évaluations nécessaires pour un run d'un cas d'optimisation
V_n	Résultat normalisé du run d'un cas d'optimisation
E_n	Nombre normalisé d'évaluation du run d'un cas d'optimisation
R_R	Résultat normalisé d'un run d'un cas d'optimisation
\mathfrak{R}_R	Ensemble des runs résultat d'un cas d'optimisation
Opérateurs mathématiques	
Υ_E	Opérateur de normalisation du nombre brut d'évaluations
Υ_R	Opérateur de normalisation du résultat brut
\mathcal{A}_R	Opérateur d'agrégation utilisé pour calculer le résultat d'un run
\mathcal{A}_α	Opérateur d'agrégation utilisé pour calculer les résultats temps court
\mathcal{A}_Ω	Opérateur d'agrégation utilisé pour calculer les résultats temps long
\mathcal{A}_C	Opérateur d'agrégation utilisé pour calculer les résultats de cas
\mathcal{A}_M	Opérateur d'agrégation des $MaxFes$
\mathcal{A}_F	Opérateur d'agrégation des fonctions
\mathcal{A}_D	Opérateur d'agrégation des dimensions
Indicateurs de performances	
S	Score global
R_R	Résultat d'un run
R_C	Résultat d'un cas d'optimisation
R_α	Résultat α
R_Ω	Résultat Ω

TABLE 5.1 – Notation du chapitre 5

5.1/ Contexte et objectifs

Dans un contexte de développement de nouvelles méthodes d'optimisation à outrance, pour preuve le graphique 5.1, il nous est apparu fondamental de proposer un outil fiable pour "certifier" toutes ces approches. En effet, ces dernières années de nombreuses métaheuristiques ont été développées. Il paraît compliqué voire impossible de les citer toutes tant leur nombre est conséquent, sans compter que certaines approches peuvent être hybrides. Nous nous retrouvons donc face à une multitude d'approches et on peut se demander si ces méthodes délibérément améliorées ou hybridées ne deviennent pas inutilement compliquées comme très justement évoqué dans la référence [69] !

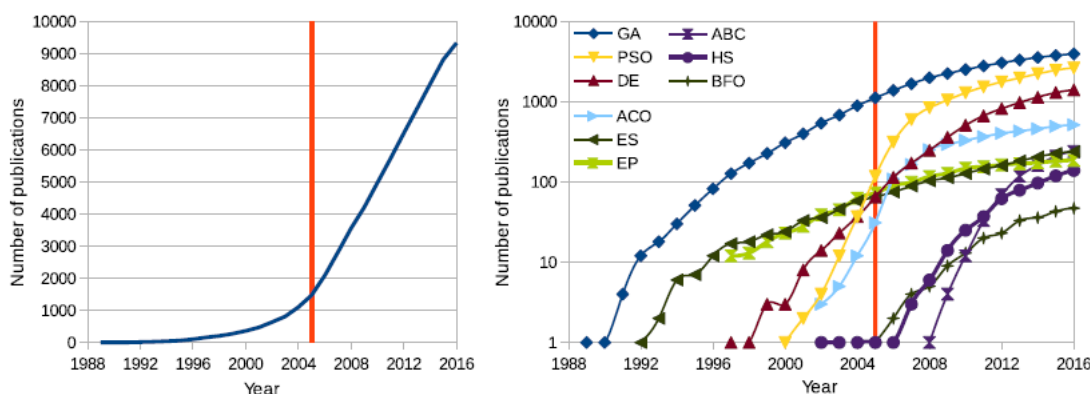


FIGURE 5.1 – Évolution du nombre de publications [31]

La facilité d'utilisation apparente d'algorithmes métaheuristiques pour l'optimisation ne doit pas cacher le fait qu'ils doivent être conçus et impérativement validés rigoureusement.

Deux principales questions peuvent alors être posées :

1. Comment estimer les performances d'un algorithme de manière fiable ?
 À cette question est forcément associé le concept de comparaison entre les algorithmes. On souhaiterait pouvoir s'assurer de l'efficacité d'un algorithme mais aussi le comparer de manière certifiée aux autres.
2. Comment régler les paramètres pour obtenir la meilleure performance possible ? Ce point sera abordé dans le chapitre 6.

Lorsqu'un nouvel algorithme est développé, la plupart du temps, il est comparé à quelques autres sur la base des moyennes et des écarts types de résultats finaux de quelques fonctions [31]. En outre, les fonctions tests sélectionnées sont bien connues, comme celle proposée par De Jong [22]. Cette connaissance a priori peut conduire à une évaluation erronée de l'efficacité de l'algorithme. Pour éviter des conclusions incorrectes, les algorithmes doivent être testés sur des benchmarks [31, 59].

L'année 2005 marque un tournant dans l'évaluation des algorithmes d'optimisation avec l'apparition des premiers benchmarks. Les deux plus populaires ont été présentés lors de sessions spéciales dans des congrès internationaux dédiés à l'optimisation :

- ❖ CEC associé au congrès "IEEE Congress on Evolutionary Computation" [86, 17].
- ❖ BBOB (Black-Box Optimization Benchmarking) associé au congrès "Genetic and Evolutionary Computation Conference" (GECCO) [39, 36]. Plus récemment, une

plateforme conçue pour comparer les optimiseurs continus en boîte noire a été développée : Comparing Continuous Optimizers (COCO) [38]. Vous trouverez plus de détails sur cette approche dans [37, 40, 2].

Ces benchmarks jouent un rôle fondamental à plusieurs titres. Ils permettent de tester les nombreux algorithmes développés ces dernières années, de les certifier et attester de leur efficacité. En effet, très peu d'approches ont de véritables résultats mathématiques attestant de leur convergence. Ils permettent aussi les comparaisons entre tous ces algorithmes afin d'établir des classements en fonction de la situation.

Le principe d'un benchmark est assez simple : exécuter des algorithmes sur des exemples de référence, récupérer et analyser les résultats via le calcul d'un score. Deux entités sont donc nécessaires à la constitution d'un benchmark :

- ❖ Un ensemble de cas test. Dans la majorité des cas, les algorithmes à étudier sont testés sur un ensemble de fonctions mathématiques bien connues dites "fonctions tests". Il est à noter que pour la plupart de ces fonctions, l'optimum global est connu. Ceci rend possible un calcul simple du nombre d'évaluations (ie le nombre d'appels à la fonction objectif). Ceci permet aussi de discuter la meilleure valeur obtenue pour la fonction objectif, autorisant l'introduction d'un critère d'arrêt.
- ❖ Un score. Après l'exécution des différents tests sur les fonctions de référence un score est calculé. C'est ce score qui va permettre de juger de l'efficacité ou non d'un algorithme par rapport à l'ensemble des fonctions testées.

Une rapide comparaison de ces deux principaux benchmarks conduit aux remarques suivantes. Le CEC repose sur une stratégie "Fixed budget" : le temps de résolution du problème est fixe. Il consiste à arrêter l'algorithme après un certain temps en général exprimé en nombre d'évaluations de la fonction objectif (D'autres options sont possibles [89]). Dans un contexte d'optimisation en conception avec des problèmes réels, cette approche perd en pertinence. En effet, un appel à la fonction objectif (calcul via un modèle complexe par exemple) peut être très coûteux en temps CPU et donc le nombre d'appels devrait être réduit au minimum. Ce point a déjà été évoqué dans le chapitre avec nos améliorations. Le BBOB repose sur une stratégie "Fixed target" qui consiste à arrêter l'algorithme quand une valeur de la fonction objectif est atteinte. Cette approche soulève un autre problème important : comment gérer un algorithme qui n'atteint pas la cible ou dans un temps trop long ?

Ce sont ces remarques qui nous ont conduit à mettre en place un nouveau benchmark. Le benchmark développé ici fonctionne sur le schéma standard. La vraie originalité réside dans le calcul des scores et donc dans l'évaluation de la performance de l'algorithme testé. L'idée ici est de fournir une vraie mesure de référence exploitable pour ceux qui souhaitent maîtriser, évaluer et améliorer leur algorithme. Ce nouveau calcul de scores s'attache à évaluer équitablement les algorithmes en mettant en avant des critères relatifs aux problématiques rencontrées dans un contexte industriel. Les résultats bruts des évaluations des fonctions objectifs sont traités à différents niveaux. Pour synthétiser les résultats bruts en un indicateur simple, des techniques d'agrégation [19] sont utilisées. L'originalité du travail proposé se trouve dans l'utilisation d'un niveau de satisfaction pour calculer le score. En effet, pour que ces données brutes soient comparables, elles sont normalisées grâce à une métrique particulière. La fonction de désidérabilité de Harrington [74] est utilisée ici pour calculer cette satisfaction [34, 45].

5.2/ Architecture du benchmark proposé et paramètres de contrôle

Avant de commencer une expérience, un ensemble de données de référence doit être choisi. L'idée principale est d'exécuter un algorithme avec un ensemble de paramètres fixés sur des cas d'optimisation connus afin de mesurer ses performances.

5.2.1/ Paramètres de contrôle

Un cas d'optimisation sera entièrement défini par la donnée de trois paramètres :

1. La fonction objectif notée F .
2. La dimension du problème à traiter notée D .
3. Le nombre maximum d'évaluations possibles de la fonction objectif noté FES . Ce nombre FES est calculé en prenant ce qui existe dans la littérature [6] à partir d'un coefficient $MaxFES$ comme suit :

$$FES = 10000 \times D \times MaxFES \quad (5.1)$$

Catégories	Numéro	Fonctions
Unimodal	1	Bent Cigar
	2	Discus
Multimodale	3	Weierstrass
	4	Schwefel
	5	Katsuura
	6	HappyCat
	7	HGBat
	8	Griewank plus Rosenbrock
	9	Scaffer
Hybride	10	Hybrid 1 (Schwefel, Rastrigin, High Conditioned Elliptic)
	11	Hybrid 2 (Griewank, Weierstrass, Rosenbrock, Scaffer)
	12	Hybrid 3 (Katsuura, HappyCat, Griewank, Rosenbrock, Schwefel, Ackley)
Composée	13	Composition 1 (Rosenbrock, High Conditioned Elliptic, Bent Cigar)
	14	Composition 2 (Schwefel, Rastrigin, High Conditioned Elliptic)
	15	Composition 3 (HGBat, Rastrigin, Schwefel, Weierstrass, High Conditioned Elliptic)

TABLE 5.2 – Ensemble des fonctions objectifs du benchmark

5.2.2/ Fonction objectif

Le choix des fonctions est fondamental comme évoqué dans [8]. Nous avons opté pour des fonctions mathématiques usuelles, mais comme évoqué dans [32], ce choix ne sera jamais parfait ! Les fonctions objectifs sont au nombre de 15. Elles sont issues du CEC 2015 [73]. Nous les avons choisies en fonction de leurs caractéristiques mathématiques pour tenter de couvrir un maximum de situations possibles (Paysage mathématique des fonctions). Le détail de ces fonctions est discuté dans le tableau 5.2 et leur expression mathématique complète peut être trouvée dans la thèse de François Schott [81].

5.2.3/ Dimension

Quatre valeurs de dimensions sont retenues pour toutes les fonctions. Nous nous situons dans le contexte industriel de l'optimisation pour l'aide à la conception. Dans ce cadre-là, la plupart des problèmes ont entre 8 et 24 variables avec une moyenne à 15 [66]. Bien entendu, des problèmes de dimensions bien plus élevées existent ! Néanmoins, compte tenu de notre contexte, il semble judicieux de choisir des dimensions inférieures à 50 (Tableau 5.3.)

Grandeur considérée	Nombre	Valeurs étudiées
Dimension	$N_D = 4$	$D \in \{10, 20, 30, 50\}$
Fonctions objectifs	$N_F = 15$	$\begin{cases} N_{uni} = 2 \\ N_{multi} = 7 \\ N_{hybrid} = 3 \\ N_{comp} = 3 \end{cases}$
<i>MaxFEs</i>	$N_M = 7$	$MaxFEs \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$
Nombre de cas d'optimisation	$N_C = 420$	$N_C = N_D \times N_M \times N_F$

TABLE 5.3 – Paramètres de contrôle du benchmark

5.2.4/ Nombre maximum d'évaluations de la fonction objectif

Sept valeurs de *MaxFEs* sont utilisées par le benchmark (Tableau 5.3).

5.2.5/ Run et séries

La nature stochastique de certains algorithmes à tester nécessite la répétition du même cas d'optimisation un grand nombre de fois. Ceci permettra de mettre en place une étude statique indispensable dans le cas des métaheuristiques. Il a été prouvé que $N_T = 51$ évaluations (Run) sont suffisantes pour faire des différences pertinentes de performance avec une signification statistique [58]. Dans la suite, l'ensemble des exécutions du même cas d'optimisation sera appelé une série.

Éléments	Nombre
Séries	$N_T = 51$
Runs	$N_R = N_T \times N_C = 21420$

TABLE 5.4 – Phase de calculs : run et séries

Toutes les informations relatives à la constitution de ce benchmark sont synthétisées dans la Figure 5.2 et le tableau 5.3.

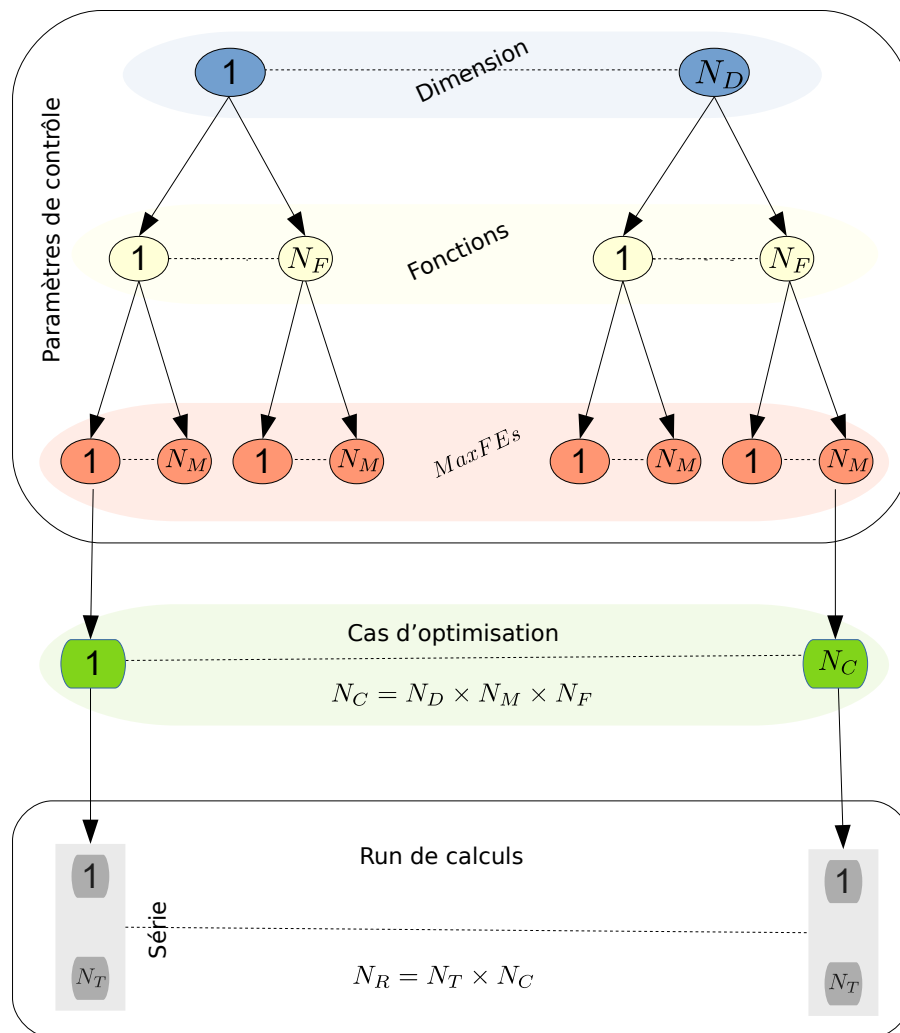


FIGURE 5.2 – Constitution du benchmark

5.2.6/ Critère d'arrêt d'un run d'un cas d'optimisation

Le processus d'optimisation sur un cas donné i prend fin quand un critère d'arrêt est atteint. Avant de parler de résultat, la problématique suivante doit être évoquée : quelles sont les

conditions qui font que la résolution de ce cas i s'arrête et vont permettre de récupérer les données indispensables à la mise en place de nos indicateurs? Deux stratégies sont envisageables [37] :

- ❖ La stratégie "Fixed budget" consiste à arrêter l'algorithme après un certain nombre d'évaluations de la fonction objectif.
- ❖ La stratégie "Fixed target" consiste à arrêter l'algorithme quand une valeur de la fonction objectif est atteinte.

Comme l'explique [70], ces deux approches, bien que valables et justifiées par des besoins pratiques, ont des limites qui les rendent inappropriées pour les cas réels si elles sont utilisées individuellement. Une approche intéressante consiste à combiner les deux scénaris [89]. C'est ce que nous proposons de faire dans le travail proposé en allant plus loin : trois stratégies d'arrêt (choisis dans [66]) sont associées pour construire le critère d'arrêt de notre benchmark (Figure 5.3). Le run de calcul d'un cas d'optimisation sera arrêté si au moins l'une des trois conditions suivantes est atteinte :

- ❖ Le nombre maximum d'évaluations de la fonction objectif est atteint.
- ❖ Une valeur de la fonction objectif est atteinte.
- ❖ L'un des critères d'arrêt potentiel de l'algorithme testé est atteint.

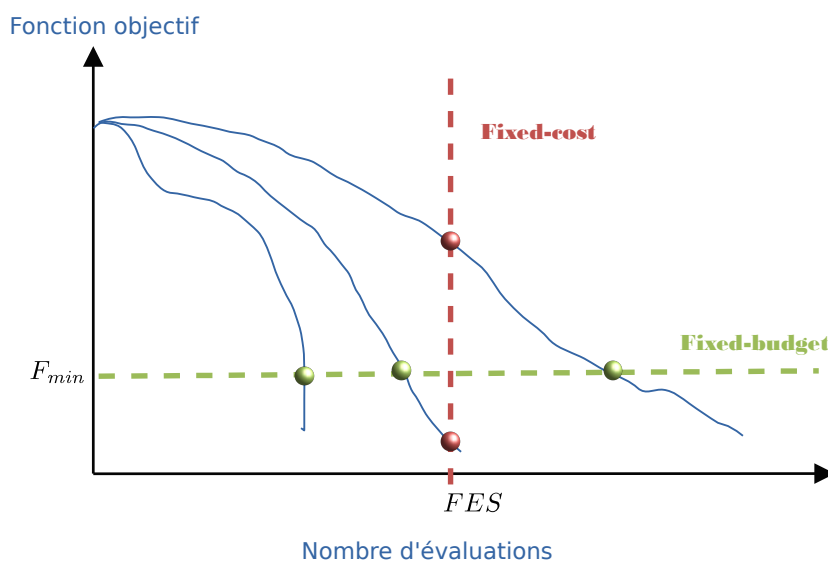


FIGURE 5.3 – Critères d'arrêts

5.3/ Outils de comparaison et indicateurs de performances

Maintenant que la constitution et la structuration du benchmark sont bien établies, se pose la question du calcul d'un/des scores. L'approche développée doit être capable de répondre à deux attentes fondamentales et extrêmement différentes.

Le première attente concerne la mise en place d'un outil standard de certification et comparaison des algorithmes. A cet effet, un score global est défini de manière à évaluer au mieux la performance d'un algorithme, dans un contexte général. Ce score tient compte de toutes les dimensions, de toutes les fonctions et de tous les *MaxFEs*.

La deuxième attente concerne la mise en place d'un outil de compréhension du comportement d'un algorithme face à des situations données. Pour ce faire, le concept de sous-scores est introduit. Les sous-scores fournissent des informations spécifiques comme l'influence de la dimension ou de la fonction-objectif sur les résultats. En effet, il est difficile de trouver une métaheuristique qui obtienne de bons résultats sur tous les problèmes. Les sous-scores sont essentiels pour déterminer sur quels problèmes un algorithme fonctionnera correctement. Ils montrent les qualités et défauts d'un algorithme. Ces informations peuvent être utilisées pour modifier ou régler un algorithme de manière soit à le rendre polyvalent soit à le spécialiser sur un type de problème. De plus, en connaissant les sous-scores de plusieurs algorithmes, il est possible de choisir le plus adapté à un problème donné.

Dans un contexte industriel, les cas où l'expression analytique de la fonction objectif est connue sont extrêmement rares voire inexistants. En général, la fonction objectif est le résultat d'une simulation exécutée sur un modèle complexe : concept de cosimulation. Les nombreuses simulations numériques nécessaires à l'évaluation de la fonction objectif augmentent nettement le coût numérique de la résolution du problème d'optimisation.

Toutes ces remarques conduisent à la conclusion suivante : il semble primordial de faire des efforts pour limiter le nombre d'évaluations de la fonction objectif lors de la résolution du problème d'optimisation. Cette grandeur est étroitement liée à l'algorithme mis en œuvre mais tout doit être fait pour tenter de la réduire. C'est pourquoi l'expression du score global proposé dans ce travail prend en compte ce nombre d'évaluations dans son calcul.

5.3.1/ Résultat d'un run

Dans le cas général de beaucoup de benchmarks, le résultat d'un run est basé sur la valeur de la fonction objectif et/ou l'atteinte du critère d'arrêt. Dans le présent travail, l'approche est totalement différente et originale. Le calcul du résultat d'un run repose sur différents opérateurs mathématiques d'agrégation et de normalisation. Le résultat d'un run, noté R_R , est issu de l'agrégation de deux grandeurs normalisées afin de quantifier la qualité du résultat obtenu en fonction du temps investi.

- ❖ La valeur brute de la fonction objectif sur un cas d'optimisation ne peut pas être utilisée telle quelle dans le calcul d'un indicateur de performance. Pour pouvoir mettre en place une grandeur la plus universelle possible, il est fondamental de pouvoir comparer ce qui est comparable. La valeur brute issue du run de calcul V_G est donc normalisée pour obtenir V_N .
- ❖ Le nombre d'évaluations de la fonction objectif E_G réalisées par l'algorithme est normalisé pour obtenir E_N .

Ces deux grandeurs normalisées sont ensuite agrégées pour le résultat R_R comme donné

dans l'équation 5.2. La figure 5.4 résume la démarche et les différentes opérations mises en place.

$$R_R = \mathcal{A}_R(\Upsilon_E(E_G); \Upsilon_V(V_G)) \quad (5.2)$$

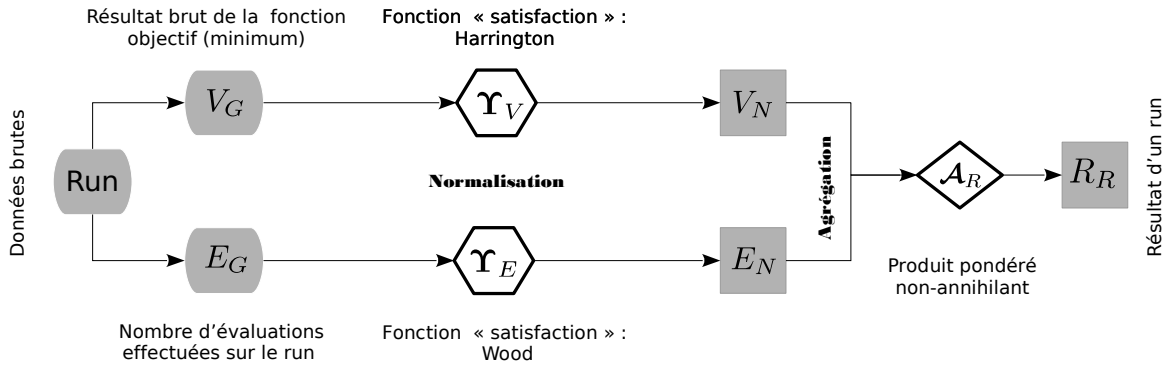


FIGURE 5.4 – Traitement du résultat brut d'un run

Normalisation : Pour régler ce problème de mise à l'échelle du résultat brut de la fonction objectif V_G , nous utilisons une fonction de désirabilité qui permet de transformer le résultat brut d'un run en une grandeur adimensionnelle : la fonction de désirabilité de Harrington [74]. Point extrêmement intéressant d'un point de vue pratique, cette fonction permet aussi d'exprimer un niveau de satisfaction. Un niveau de désirabilité proche de 0 signifie que le résultat est loin du minimum de la fonction objectif et donc que la solution n'est pas parfaitement adaptée au problème. A l'inverse un niveau proche de 1 traduit le fait que le minimum obtenu est de bonne qualité. Cette fonction a été retenue dans le cas "The lower, the better" et une échelle logarithmique a été choisie. En ce qui concerne le traitement du nombre brut d'évaluations E_G , la fonction de Wood a été mise en place.

Agrégation : Les deux valeurs normalisées obtenues E_n et V_n sont agrégées afin d'obtenir une seule grandeur représentative du résultat. Pour ce faire, le problème d'optimisation est vu par le prisme de l'ingénieur concepteur : quand on optimise, on cherche à obtenir le meilleur résultat dans le plus court laps de temps. De fait, la formule d'agrégation utilisée ici est de type produit pondéré ne respectant pas l'axiome d'annihilation.

Les expressions mathématiques détaillées des différents opérateurs sont accessibles dans [?].

5.3.2/ Résultat d'un cas d'optimisation

Les résultats des N_T runs d'une série (soit un cas d'optimisation) doivent maintenant être traités. En général, des paramètres statistiques comme la moyenne ou l'écart-type sont calculés. Ce n'est pas du tout le cas ici. Si le temps de calcul de la fonction objectif est jugé acceptable, le nombre d'appels à la fonction objectif n'impacte alors que faiblement le temps de résolution du problème d'optimisation. Il est alors possible d'imaginer une répétition des problèmes d'optimisation à moindre coût. L'optimisation devient alors un

outil d'aide à la décision rentable et envisageable. Inversement si le temps de calcul de la fonction objectif est élevé, le temps de traitement du problème d'optimisation devient inacceptable. Pour distinguer ces deux situations différentes, deux types de résultats sont évalués : le résultat α et le résultat Ω .

5.3.2.1/ Résultat α

Ce résultat correspond au résultat que l'on peut obtenir si le cas d'optimisation peut être résolu un grand nombre de fois. C'est pourquoi on prend le meilleur résultat du tirage. Le résultat α est la valeur maximale de tous les runs.

5.3.2.2/ Résultat Ω

Dans un contexte de cosimulation, un run peut s'avérer très chronophage. Il est alors difficilement envisageable de réaliser plusieurs runs du même problème pour améliorer la solution. Le résultat Ω va donc reposer sur le concept d'intervalle de confiance à 95% sur l'ensemble des runs.

5.3.2.3/ Résultat d'un cas d'optimisation

Pour déterminer le résultat d'une série (ie l'ensemble des runs sur un cas d'optimisation), nous définissons la moyenne des résultats α et Ω . Ce résultat est noté R_C .

La démarche est illustrée sur la figure 5.5.

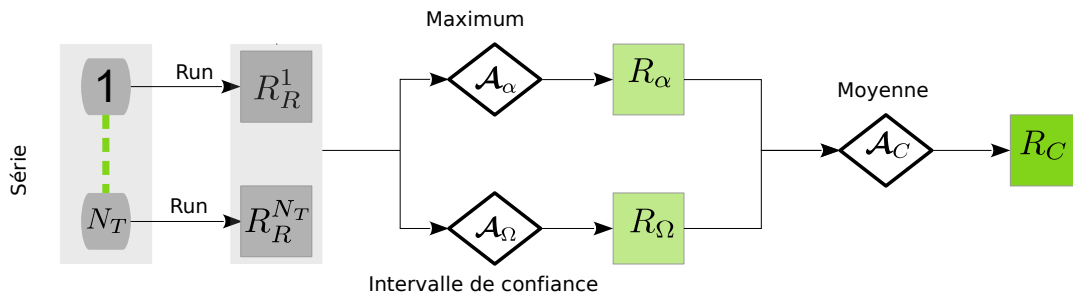


FIGURE 5.5 – Résultat d'un cas d'optimisation

5.3.3/ Score global

Afin d'établir le score global, tous les résultats de tous les cas d'optimisation doivent être regroupés pour définir un seul indicateur. L'agrégation résume un ensemble des données en un seul indicateur. Le processus de calcul du score global est composé de trois étapes d'agrégation :

Étape 1 : Les résultats de tous les cas d'optimisation sont regroupés et agrégés par *MaxFEs*.

Étape 2 : Tous les résultats de *MaxFEs* sont agrégés par les fonctions.

Étape 3 : Pour terminer, tous les résultats des fonctions sont agrégés par rapport aux dimensions, ce qui donne le score global (S).

Ceci peut être écrit d'une succession d'agrégations donné par l'équation 5.3. La démarche est synthétisée dans la figure 5.6

$$S = \mathcal{A}_D(\mathcal{A}_F(\mathcal{A}_M(R_C))) \quad (5.3)$$

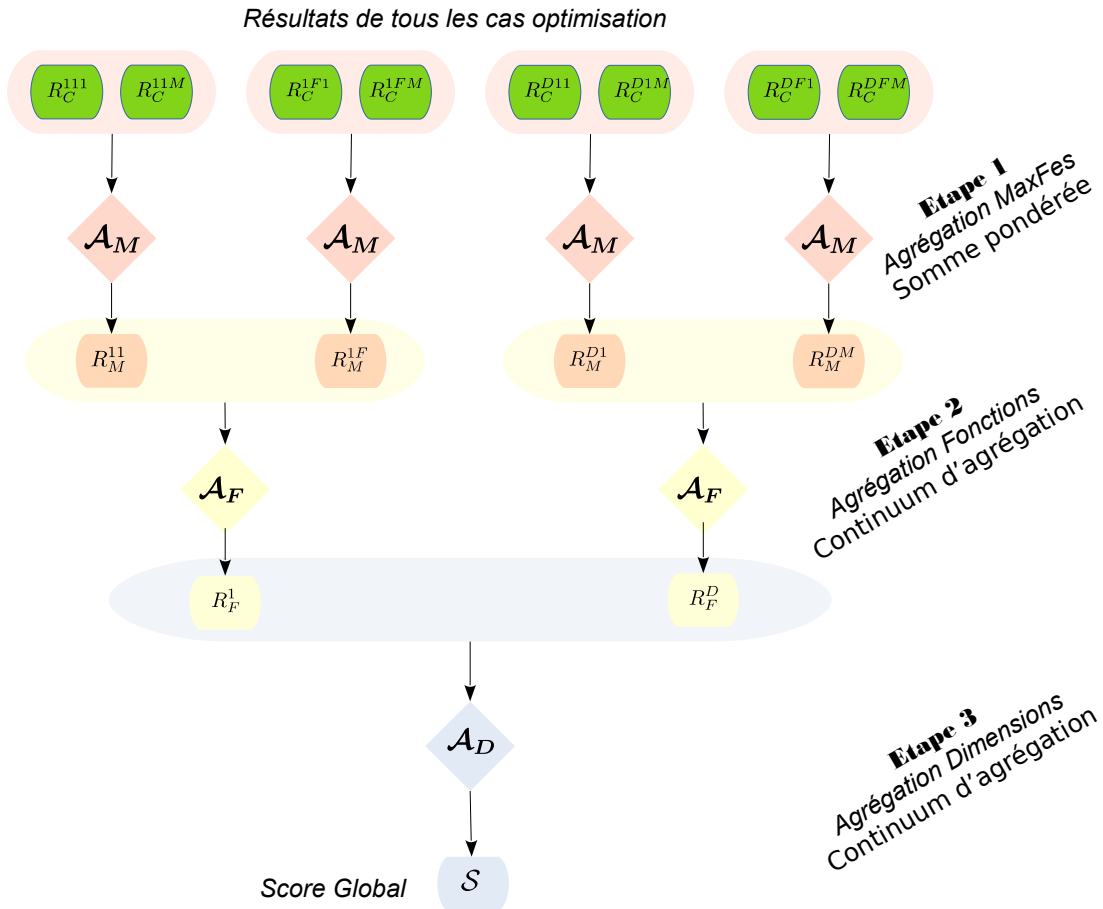


FIGURE 5.6 – Démarche générale pour le calcul du score principal

Nous ne donnons pas ici les expressions mathématiques de tous les opérateurs d'agrégation. Tous ces opérateurs sont développés dans notre article [?].

5.3.4/ Sous-scores

Une des originalités majeures de ce benchmark réside dans la possibilité de calcul des sous-scores. Le score global est un indicateur généraliste du comportement d'un algorithme. Ce score global est un point de départ extrêmement intéressant mais pas suffisant. En effet, ce score ne permet pas de caractériser pleinement un algorithme. Il ne permet pas d'identifier ses spécificités et notamment ses réactions face à une dimension donnée ou un paysage de fonction objectif particulier. Pour obtenir ce type d'informations, les sous-scores sont introduits. De part la structure du benchmark, trois sous-scores sont disponibles :

- ❖ Les sous-scores par dimensions,
- ❖ Les sous-scores par fonctions,
- ❖ Les sous-scores par *MaxFEs*.

Ces grandeurs sont basées sur la même démarche que le score global mais calculées sur des sous-ensembles de cas d'optimisation restreint. A titre d'exemple, si nous nous intéressons au sous-score dimension S_{D20} , ce sous-score sera élaboré en prenant en compte seulement tous les cas d'optimisation de dimension 20.

5.4/ Application et résultats

Tous les développements associés à ce benchmark ont été réalisés dans le cadre de la thèse CIFRE de François Schott avec la société My-OCCS [81] dans le logiciel OROS. Le benchmark a été mis en œuvre sur les algorithmes présentés dans le tableau 5.5. La plupart des algorithmes retenus pour cette validation sont des algorithmes pris dans leur version d'origine sans aucune modification et amélioration : l'idée est de vraiment établir une comparaison fiable entre les capacités des différents algorithmes. Les paramètres utilisés par les algorithmes testés sont basés sur les publications initiales et ont été ajustés à partir de l'expérience des utilisateurs et des paramètres obtenus par méta-optimisation. Les informations détaillées sur les algorithmes, telles que leur pseudo-code et leurs paramètres, sont disponibles dans [81].

En terme de temps de calcul, le benchmark proposé est équivalent au CEC [16, 17]. Avec le serveur de calcul utilisé¹, il faut quelques heures pour évaluer un algorithme avec le benchmark proposé.

Algorithmes	Familles	Version/Origine
PSO	Population	Version inertiel [?]
CMAES	Evolutionnaire	Version d'origine [35]
Algorithme Génétique	Génétiques	Version d'origine [43]
Cuttlefish	Population	Version d'origine [27]
Recuit simulé	Evolutionnaire	Version d'origine [51]

TABLE 5.5 – Présentation des algorithmes

Il est clair que notre étude est limitée en terme de nombre d'algorithmes. Nous aurions pu tester d'autres algorithmes, mais il fallait faire des choix ! Quelques grandes idées issues

1. Asus Z10PE-D16WS, 2 x (Intel(R) Xeon(R) CPU E5-2620 v3 @2,40GHz, 2401 MHz, 6 cœurs, 12 processeurs, 64 Go - 8 x (Micron crucial Mémoire Premium 8GB DDR4-2133)

de la communauté sur les différentes stratégies et leur efficacité sont développées dans [66, 69, 31, 91] et nous ont permis d'établir ce choix : un algorithme parmi les familles les plus efficaces.

5.4.1/ Score global

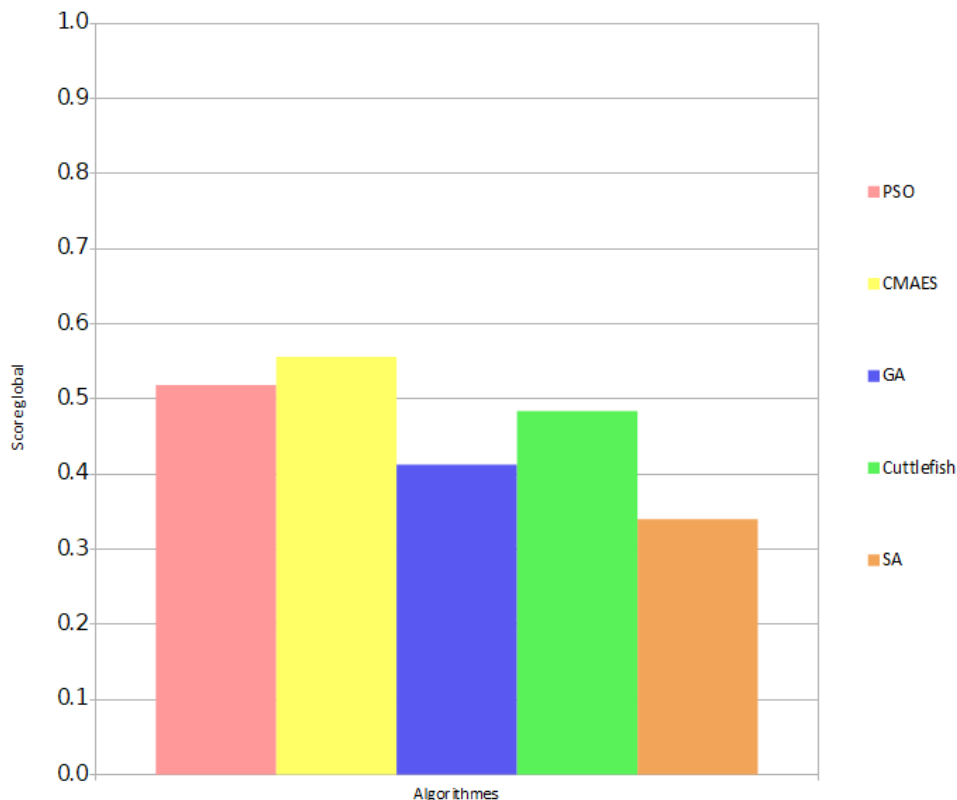


FIGURE 5.7 – Score global pour les différents algorithmes

Le benchmark proposé présente des scores homogènes avec des valeurs maximales limitées. D'après les résultats obtenus, ce benchmark est en bon accord avec la littérature. Les algorithmes CMAES montrent un fort potentiel [35]. Pour les deux benchmarks historiques (CEC et BBOB), cette affirmation est confirmée et le CMAES apparaît comme l'algorithme de référence à l'heure actuelle.

En raison de leur architecture et de leurs paramètres, les autres algorithmes testés font une part plus importante à l'aléatoire. Ceci explique pourquoi ils ont les résultats les plus faibles. Le Cuttlefish et le PSO ont des résultats proches car ils utilisent tous deux le déplacement vers le meilleur résultat local et global. En fait, ils sont basés sur un bon équilibre entre aléatoire et logique. Enfin, CMAES est le meilleur algorithme d'optimisation car il utilise la technique la plus sophistiquée pour collecter des informations sur les valeurs itérées de la fonction objectif.

5.4.2/ Sous-scores

5.4.2.1/ Score par dimensions

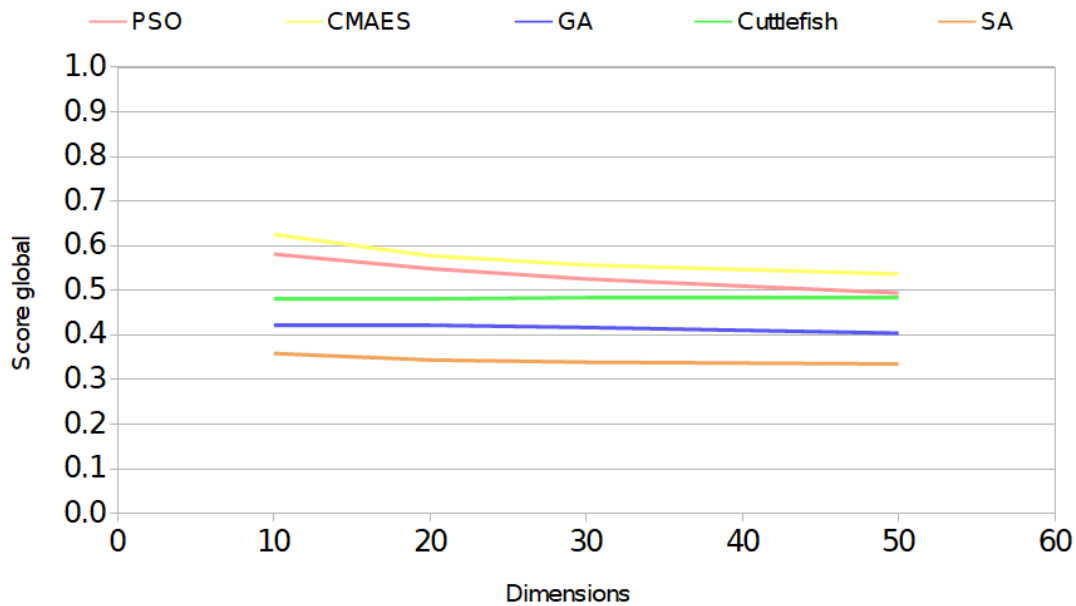


FIGURE 5.8 – Score par dimension pour les différents algorithmes

La figure 5.8 présente les scores des algorithmes pour les différentes dimensions du benchmark. Il ressort les conclusions suivantes de l'analyse de ce graphique.

- ❖ Ces scores diminuent lorsque la dimension augmente. Ceci paraît complètement naturel, plus la dimension est élevée plus le problème est complexe et donc la convergence difficile à atteindre.
- ❖ Les algorithmes Génétiques et l'algorithme CuttleFish ont un score relativement stable. Ceci traduit une forte capacité à explorer les espaces de recherche de grandes dimensions. Il est probable que le caractère aléatoire des algorithmes (PSO, CMAES et SA) dégrade leur capacité en grandes dimensions.

5.4.2.2/ Score par fonctions

La figure 5.9 présente les scores des algorithmes pour les différentes fonctions du benchmark. La première conclusion que l'on peut tirer de ces résultats est la mise en évidence du no free lunch theorem. En effet aucun algorithme n'est capable d'obtenir de bons résultats sur toutes les fonctions. On note que les fonctions les plus simples sont les fonctions uni-modales et de compositions. Les fonctions les plus difficiles sont les fonctions multi-modales suivies par les fonctions hybrides. L'intérêt de ce score par fonctions est de parvenir à mettre en avant les spécificités et l'efficacité de chaque algorithme par type catégorie de fonctions. Il ressort de la figure 5.9 que notre version de GA est efficace sur les fonctions hybrides tandis que les algorithmes CMAES et Cuttlefish sont performants sur les fonctions uni-modales.

Finalement, aucune règle absolue concernant l'efficacité ne peut être établie. Cela pourrait signifier que le choix et la classification des fonctions de notre benchmark (basé sur les

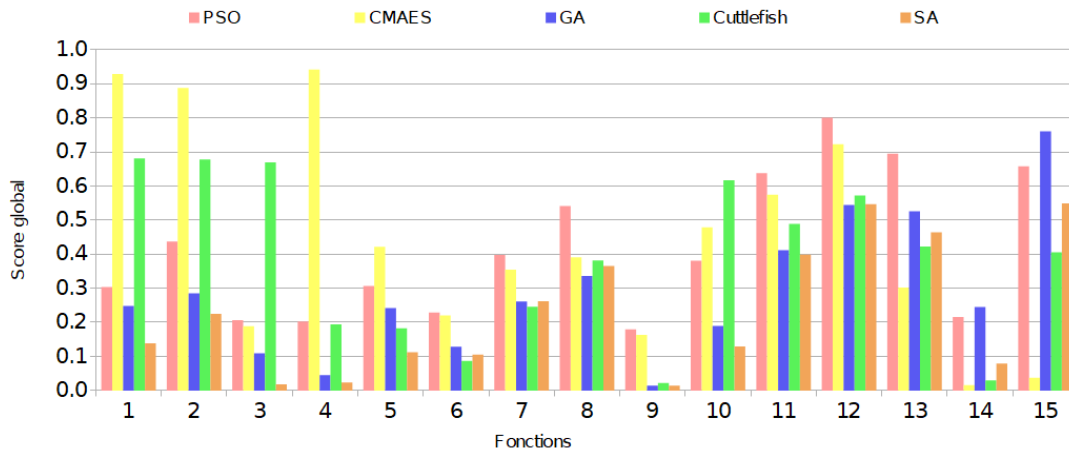
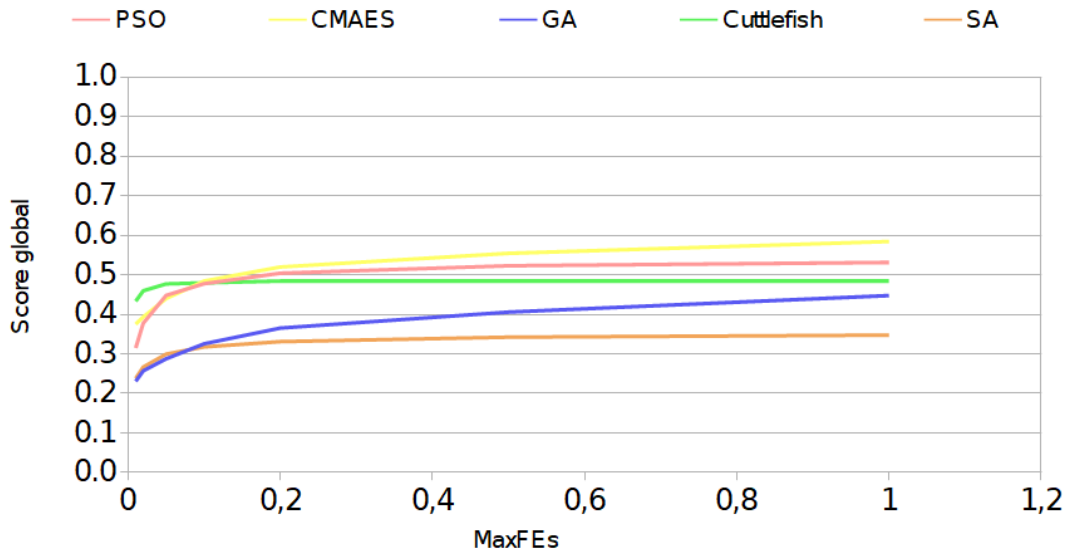


FIGURE 5.9 – Score par fonction pour les différents algorithmes

fonctions CEC) ne sont pas parfaitement cohérents. Ce constat est expliqué en partie dans [32]. Les raisons pour lesquelles certaines fonctions, comme la fonction 9 ou la fonction 14, sont plus difficiles à dépasser que d'autres, comme la fonction 12, sont assez difficiles à expliquer. Pour ce faire, il conviendrait de mener une étude appropriée des mécanismes des algorithmes [84] et du paysage des fonctions [32]. Ce point fait partie des évolutions du benchmark prévues dans mes activités.

5.4.2.3/ Score par *MaxFes*

FIGURE 5.10 – Score par *MaxFes* pour les différents algorithmes

Pour conclure l'étude des sous-scores, la figure 5.10 présente les scores des algorithmes pour les différentes valeurs de *MaxFes* du benchmark. Nous constatons que l'efficacité des algorithmes différents en fonction du nombre d'évaluations : un algorithme performant avec

un *MaxFes* faible ne le restera pas avec un *MaxFex* grand. Ce point a été déjà mentionné dans [69]. Selon cette même source, le PSO fonctionne bien lorsque les évaluations sont limitées, l'AG lui est plus efficace lorsque les évaluations sont autorisées en grande quantité.

5.5/ Conclusion

Ces dernières années, un grand nombre de nouveaux algorithmes d'optimisation, dont la plupart sont des métaheuristiques, ont été développés. Mesurer et certifier l'efficacité de tous ces algorithmes afin de les classer suivant les problèmes à traiter est un problème plus qu'ouvert !

Dans le travail présenté dans ce chapitre, nous avons tenté de proposer un moyen original de mesurer l'efficacité des algorithmes. La figure 5.11 propose une synthèse des différences majeures entre notre approche et les deux leaders actuels.

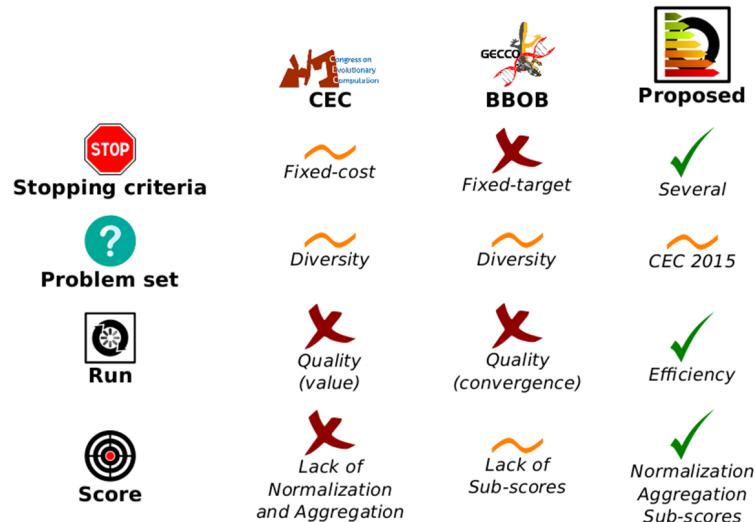


FIGURE 5.11 – Comparatif Benchmark proposé / Littérature

La nouveauté réside dans le calcul du score issu des différents cas tests. Les résultats d'une exécution prennent en compte une valeur normalisée de la fonction objectif à la fin de l'optimisation et une vitesse de convergence normalisée. Les résultats des cas d'optimisation sont calculés à l'aide de statistiques sur les résultats d'une exécution. Le résultat des cas passe par trois étapes d'agrégation pour produire un score. Pour aller plus loin dans l'analyse que le simple classement, le concept de sous-score a été introduit. Ces sous-scores peuvent être utilisés pour analyser un algorithme de manière fine et sélectionner un algorithme parmi un ensemble d'algorithmes testés pour un problème donné.

Nous sommes cependant bien conscients des limites du benchmark proposé. Il a été conçu pour les algorithmes métaheuristiques de recherche globale. Le score principal est une mesure d'efficacité et non une mesure de qualité [8]. Plusieurs voies d'amélioration sont en cours d'étude. Un point particulier concerne notamment le passage à des fonctions objectives issues de la mécanique bien connues : les structures treillis. Ceci permettrait de tester des algorithmes sur des cas réels.

ROBUSTESSE DES ALGORITHMES D'OPTIMISATION

Sommaire

6.1	Contexte et objectifs	61
6.2	Étude globale	62
6.2.1	Plan d'expériences : plan produit de Taguchi	62
6.2.2	Réponses du plan	65
6.2.3	Résultats	65
6.3	Études locales	66
6.3.1	Méthode	67
6.3.2	Résultats	68
6.3.3	Validation	70
6.4	Conclusions	72

6.1/ Contexte et objectifs

Il existe un nombre impressionnant de métaheuristiques pour l'optimisation globale. Les performances de ces algorithmes varient énormément, et comme le stipule le No Free Lunch Theorem, elles sont intimement liées à la structure du problème à optimiser et au paramétrage de l'algorithme. 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 métaheuristique, de connaître parfaitement quels sont les effets de chaque paramètre sur le fonctionnement de la méthode en question.

Le choix de paramètres notamment pour les métaheuristiques peut être issu de l'expertise autrement dit une gestion des paramètres "à ma main" [84]. Mais une autre solution existe pour tenter de régler les paramètres d'une approche d'optimisation au plus juste : la méta-optimisation. Il s'agit tout simplement d'optimiser l'algorithme d'optimisation. Ces dernières années de nombreuses approches d'ajustement automatique des paramètres ont été proposées [56, 14, 46, 20].

Globalement il ressort que les approches de paramétrage en optimisation peuvent être classifiées selon que la méthode soit "en ligne" ou "hors ligne". Dans l'approche "hors ligne", les paramètres apparaissent alors comme une donnée utilisateur fondamentale et

sont fixés avant l'exécution d'un problème d'optimisation. On peut parler d'un réglage, paramétrage des paramètres en amont.

Dans l'approche "en ligne", les paramètres sont contrôlés et actualisés pendant le problème d'optimisation. On parle dans ce cas plutôt d'un contrôle des paramètres.

Nous nous sommes particulièrement intéressés aux méthodes "hors ligne". En effet, dans un contexte industriel, l'idée était de trouver des jeux de paramètres robustes et adaptables à de nombreuses situations. Ceci n'est absolument pas envisageable avec les stratégies "en ligne". Dans une stratégie hors ligne, deux stratégies prédominent. Les paramètres de l'algorithme peuvent être à leur tour optimisés par une méthode d'optimisation : le paramétrage optimal est le résultat d'un problème d'optimisation résolu avec ou sans l'algorithme testé. Dans ce cas, aucune interaction entre les paramètres n'est étudiée, ce qui est un défaut majeur de cette approche. Les paramètres peuvent être ajustés via un plan d'expériences. C'est cette méthode que nous avons choisi de mettre en place dans notre travail. En effet, il nous est apparu fondamental de déterminer les meilleurs paramètres pour un algorithme donné mais aussi est surtout, il apparaît essentiel de maîtriser, voire anticiper les interactions entre les paramètres d'un algorithme. Ceci sera rendu possible par l'utilisation d'un plan d'expériences. De plus, la mise en place d'un plan d'expériences doit nous permettre d'établir un modèle de comportement pour nos paramètres à moindre coût.

Dans la littérature, les plans d'expériences ont déjà été utilisés pour régler les paramètres de certains algorithmes d'optimisation. Cependant le constat est le suivant : l'optimisation du paramétrage est réalisé au cas par cas. Ceci rend la réutilisation d'un paramétrage dans un autre contexte inutile. En effet, un paramétrage doit être adapté à chaque situation. L'idée du travail développé et initié dans la thèse de François Schott [81] est de parvenir à établir des modèles de paramétrisation adaptés à un certain nombre de situations de référence : l'originalité de travail ici est de proposer un plan d'expériences sur le benchmark introduit dans le chapitre 5 suivie d'une analyse par composantes principales. La structure même du benchmark rend possible une étude à la fois globale pour établir un jeux de paramètres généralistes robustes via le score global mais aussi une étude plus locale par l'intermédiaire des sous-scores.

C'est cette démarche que nous nous proposons de présenter dans ce chapitre (illustrée sur la figure 6.1), en particulier dans le cas de l'algorithme PSO (Chapitre 3, page 7)

6.2/ Étude globale

6.2.1/ Plan d'expériences : plan produit de Taguchi

La méthode de Taguchi est connue pour être la plus efficace lorsqu'on traite plusieurs facteurs. Outre la détermination des niveaux optimaux, cette méthode établit l'importance relative des facteurs individuels en fonction de leurs principaux effets. Un plan produit correspond à une structure de plan d'essai permettant d'étudier séparément :

- ❖ l'effet des facteurs essentiels ;
- ❖ l'effet des facteurs externes.

Certains paramètres sont plus compliqués que d'autres à contrôler et peuvent intervenir sur le benchmark. Ces paramètres vont être considérés comme des facteurs bruits.

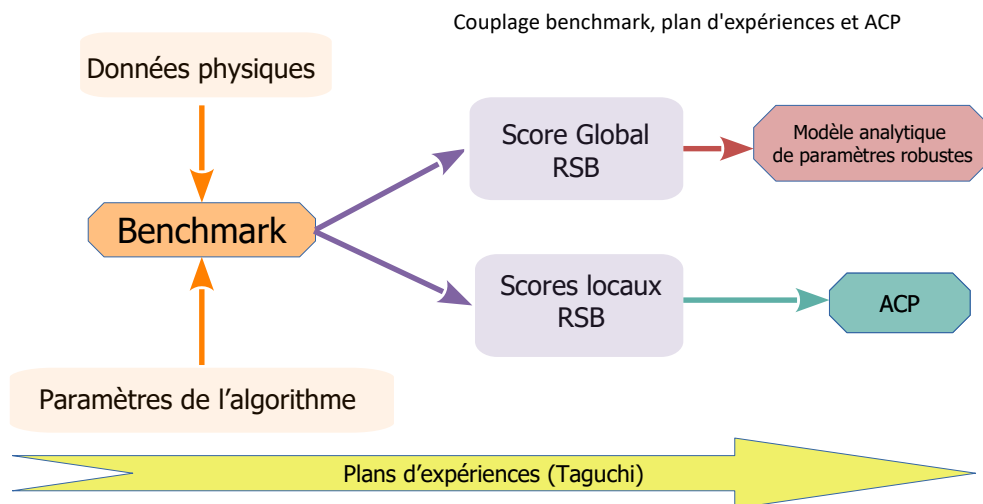


FIGURE 6.1 – Notre stratégie de méta-optimisation

6.2.1.1/ Facteurs essentiels

Dans la section 4.3.2 (page 25), nous avons présenté les paramètres qui influencent le comportement des particules dans leur déplacement à la recherche de l'optimum, notamment les paramètres algorithmiques (Tableau 4.1, page 26). Les paramètres algorithmiques du PSO sont donc choisis comme facteurs essentiels du plan d'expériences. Le choix des différentes valeurs des modalités est basé sur ce que nous avons pu trouver dans la littérature [41, 21, 50, 67]. Le tout est synthétisé dans le tableau 6.1.

Facteurs	Label	Nombre de modalités	Modalité 1	Modalité 2	Modalité 3
N	A	3	20	50	100
c_1	B	3	0.5	1	1.5
c_2	C	3	0.5	1	1.5
ω_{min}	D	3	0.2	0.3	0.4
ω_{max}	E	3	0.7	0.8	0.9
V_{factor}	F	3	0.05	0.1	0.3

TABLE 6.1 – Facteurs essentiels du plan d'expériences

Nous avons donc un plan à 6 facteurs et 3 modalités par facteurs. Dans le cas général, ie un plan complet, il serait nécessaire d'effectuer $3^6 = 729$ expériences. Une expérience est dans le cas présent une exécution du benchmark, exécution relativement couteuse. En effet, un grand nombre de cas d'optimisation doivent être résolus : $729 \times 420 = 306180$. Nous aimerions obtenir un maximum d'informations en faisant un minimum de tests. Dans ce contexte, un plan fractionnaire s'est imposé [3]. Nous nous sommes limités à l'étude de 27 cas d'optimisation, avec plan orthogonal L_{27} .

6.2.1.2/ Facteurs externes

Nous nous intéressons maintenant aux facteurs externes. Certains paramètres d'un problème d'optimisation peuvent influencer fortement le comportement de l'algorithme étudié et donc le score du benchmark indépendamment des données de l'algorithme, c'est ce que nous avons nommé les paramètres physiques (Section 4.3.2.2, Tableau 4.2, page 27). Ces paramètres sont indispensables au bon fonctionnement de l'algorithme mais sont liés au problème d'optimisation à traiter : la dimension du problème, les bornes des variables appartiennent naturellement à cette catégorie. Un autre paramètre est considéré ici : le nombre maximum d'évaluations autorisées de la fonction objectif *MaxFEs*. Le détail des modalités est décrit dans le tableau 6.2.

Facteurs	Label	Nombre de modalités	Modalité
<i>D</i>	N_1	4	{10, 20, 30, 50}
<i>Function</i>	N_2	15	$f_1 \cdots f_{15}$
<i>MaxFEs</i>	N_3	7	{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1}

TABLE 6.2 – Facteurs externes

Dans la méthode de Taguchi, au lieu de chercher à éliminer les facteurs externes (on parle aussi de facteurs bruits), on cherche à diminuer leur impact. Pour ce faire, pour chaque expérience du plan interne, un plan complet sur les facteurs externes est réalisé comme décrit sur la figure 6.2.

Étude globale		N_1	Plan complet	
Plan interne		N_2	• 3 Facteurs <i>externes</i> (Paramètres d'adaptation du PSO)	
Plan externe		N_3	→ 420 expériences	
	A B C D E F		1 2 3 420	
1	Plan fractionnaire → 6 facteurs <i>internes</i> (Paramètres algorithmiques du PSO) → 27 expériences (L_{27})		Réponses • Score global du benchmark • Moyennes • RSB	
2				
3				
...				
...				
27				

FIGURE 6.2 – Plan d'expériences pour l'étude globale

6.2.2/ Réponses du plan

Pour chaque expérience du plan interne (ie chaque ligne), les grandeurs suivantes sont calculées :

- ❖ le score global du benchmark,
- ❖ m (respectivement σ), la moyenne arithmétique (écart-type) des 420 valeurs mesurées (résultats du plan externe).
- ❖ RSB , le rapport signal bruit.

6.2.3/ Résultats

6.2.3.1/ Jeu de paramètres robustes

A partir du plan d'expériences interne, nous sommes en mesure d'établir un jeu de paramètres (des facteurs internes) robustes. Nous pouvons déterminer le niveau de chaque facteur permettant de maximiser le RSB ainsi que le score global. L'idée ici est de trouver un ensemble de paramètres que nous pourrions appeler "généralistes robustes" capables de réagir correctement dans la plupart des situations pour l'algorithme testé. C'est ici qu'apparaît tout l'intérêt du benchmark et de sa structure. C'est pour cette raison que nous parlons d'étude globale. Ainsi, le réglage optimal pour maximiser le RSB est présenté sur la figure 6.3. Le tout est résumé dans le tableau 6.3.

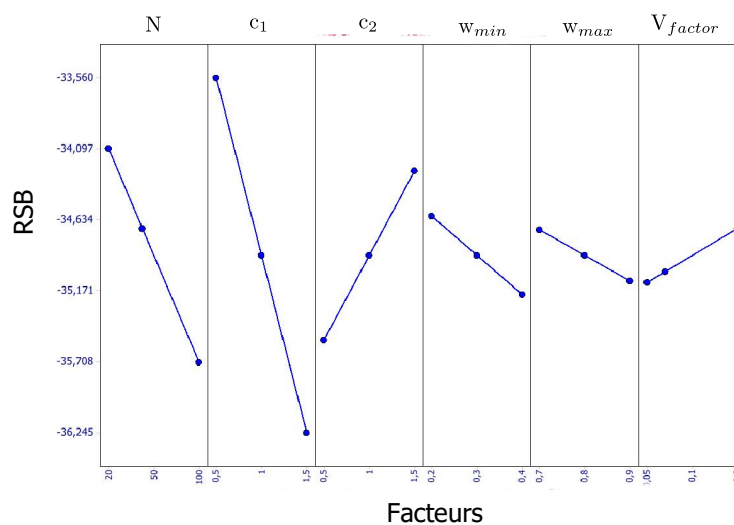


FIGURE 6.3 – Effets sur le rapport signal/bruit des résultats globaux

N	c ₁	c ₂	w _{min}	w _{max}	V _{factor}
A	B	C	D	E	F
20	0.5	1.5	0.2	0.7	0.3

TABLE 6.3 – Jeu de paramètres robustes

6.2.3.2/ Modèle analytique prédictif

Un point fort de l'utilisation des plans d'expériences dans le cadre de la méta-optimisation est le suivant : nous sommes en mesure de construire **un modèle analytique des réponses** en fonction des entrées et de leurs interactions. Pour une réponse r du plan et en l'absence de toutes informations, on se donne à priori, une loi d'évolution dont la forme la plus générale est la suivante :

$$r_{th} = H(p_1, p_2, \dots, p_r) \quad (6.1)$$

Cette fonction est trop générale et il est d'usage de l'écrire sous forme polynomiale comme suit :

$$r_{th} = \alpha_0 + \alpha_1 A + \alpha_2 B + \alpha_3 C + \alpha_4 D + \alpha_5 E + \alpha_6 F + \epsilon \quad (6.2)$$

α_i sont les coefficients du modèles mathématiques adopté a priori. Ils ne sont pas connus et vont être déterminés à partir de nos expériences numériques, donc à partir de l'exécution du benchmark.

La question qui se pose à pose concernant ce modèle est celui de sa validation. Pour ce faire, dans le cas du PSO, nous avons réalisé un test de validation. En effet, la combinaison de paramètres proposés (Tableau 6.3) ne fait pas partie du cadre testé par le plan. Il est donc nécessaire d'effectuer un test de validation. Le benchmark a donc été relancé avec ces paramètres spécifiques. Ensuite ces valeurs "expérimentales" (notées r_{val}) ont été comparées à celles obtenues via le modèle analytique prédictif notées r_{th} (Équation 6.2). La comparaison entre les deux approches est présentée dans le tableau 6.4. Les résultats sont plutôt probants et valident pleinement le modèle prédictif.

Réponses	r_{th}	r_{val}	$\frac{r_{th} - r_{val}}{r_{val}}$
Score global	0,539	0,502	7,3%
RSB	-28,390	-28,904	1,8%
Moyenne arithmétique	0,353	0,346	2,1%

TABLE 6.4 – Paramètres robustes : comparaison test de validation et modèle analytique

6.3/ Études locales

Dans la partie précédente, un jeu de paramètres robustes de l'algorithme PSO a été obtenu à partir du score global. Ce paramétrage est "passe partout" et extrêmement généraliste. Ceci est un véritable atout dans un contexte industriel quand l'ingénieur ne sait pas quel jeu de paramètres choisir mais cela peut être aussi son principal inconvénient face à des

problèmes très spécifiques ! Là est tout le paradoxe. La dimension du problème étudié, la nature de la fonction (son paysage), le nombre d'évaluations autorisées de la fonction objectif sont des aspects sur lesquels il faut aussi impérativement se concentrer.

Une étude plus approfondie est donc nécessaire pour mieux comprendre l'impact ces paramètres sur le comportement des algorithmes étudiés, en particulier du PSO dans le cas présent. Il nous est donc apparu fondamental d'aller plus loin et de raffiner notre recherche de paramètres robustes en menant des études que nous nommerons études locales. L'idée principale est d'analyser les liens entre les effets des paramètres de l'algorithme étudié. Pour ce faire, le plan d'expériences est combiné à une Analyse en Composantes Principales (ACP) pour les sous-scores : en dimension, fonctions et *MaxFes*.

6.3.1/ Méthode

6.3.1.1/ Plan d'expériences par sous-ensembles

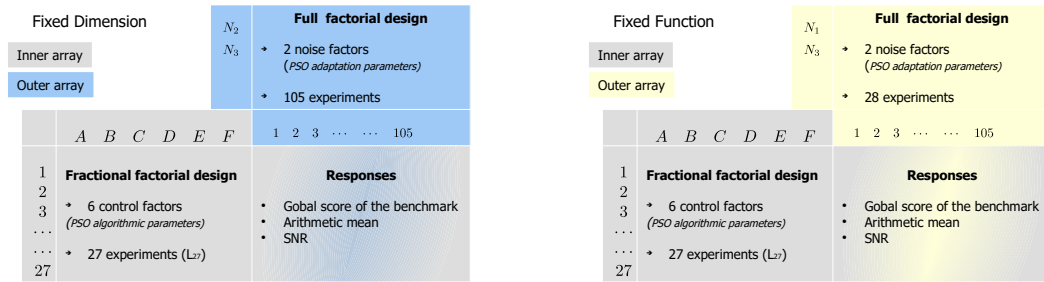
Un plan similaire à celui utilisé pour l'étude globale a été réalisé pour chaque paramètre physique suivant : dimensions, fonctions et *MaxFes*. Ceci a conduit à 26 plans supplémentaires comme décrit sur la figure 6.4. Afin de faciliter l'interprétation et la visualisation des résultats des nouveaux plans, nous avons réalisé une analyse par composantes principales (ACP). Les variables sont les paramètres physiques regroupés en trois sous-ensembles : dimension, fonction et *MaxFes* (Tableau 6.2). Ils sont fixés individuellement pour l'exécution des 26 plans alors que les autres paramètres physiques ne sont pas fixés. L'ACP est ensuite réalisée sur les valeurs du SNR pour tenter de déterminer les paramètres les plus influents.

Intéressons-nous, à présent, à la manière dont la variation des paramètres physiques influence les effets et interactions des facteurs du plan interne (paramètres algorithmiques). Ces 28 paramètres physiques sont les variables de l'analyse. Les individus sont les effets et interactions des facteurs (*A*, *B*, *C*, *D*, *E* et *F*) du plan interne, ainsi que la partie résiduelle. L'ACP est une méthode descriptive d'analyse multivariée : elle vise à décrire la manière dont un grand nombre de variables sont liées entre elles.

6.3.1.2/ Analyse par composantes principales

Le principe général de l'ACP est de réduire la dimension d'un ensemble de données dépendant d'un grand nombre de variables interdépendantes, tout en conservant l'ensemble de l'information décrivant la manière dont les données varient entre elles. Ce résultat est obtenu par la définition d'un nouvel ensemble de variables, appelées composantes principales (CP), décorréélées les unes des autres, représentant l'ensemble des variables d'origine.

Les CP sont ordonnées de telle sorte que la majeure partie des informations de variations entre les variables d'origines sont présentes dans les premières CP. De plus amples informations sur l'outil ACP sont détaillées dans [47]. Par ce changement de variables et cette réduction de variables, l'ACP permet une représentation graphique de la variation des données d'un plan d'expériences par exemple. Au-delà de la représentation graphique, par la réduction des variables, l'ACP permet de compresser les données, en réduisant le nombre de dimensions sans perdre beaucoup d'informations.



(a) Cas 1 : dimension fixée

(b) Cas 2 : fonction fixée

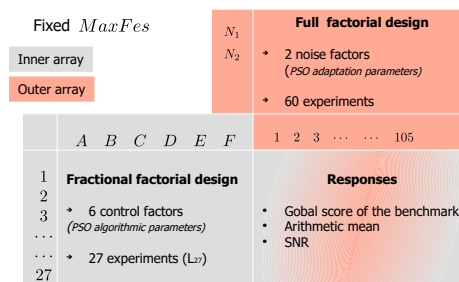
(c) Cas 3 : *MaxFes* fixée

FIGURE 6.4 – Ensemble des études locales

6.3.2/ Résultats

6.3.2.1/ Cas 1 : influence de la dimension

D'après la figure 6.5, les variables sont particulièrement bien corrélées car elles sont presque toutes concentrées au même endroit dans le cas où la dimension est égale à 20, 30 ou 50. Cela signifie que la variation de la dimension (entre 20 et 50) a peu d'effet sur le choix des niveaux des paramètres. Lorsque la dimension est égale à 10, l'effet du nombre de particules (paramètre *A*) est nettement plus important que dans les trois autres cas. Des résultats similaires ont été publiés récemment dans [52]. Ceci nous permet de valider notre approche.

6.3.2.2/ Cas 2 : influence de la fonction

D'après la figure 6.6, les variables sont plus ou moins bien corrélées. Cela signifie que le choix de la fonction influence clairement les effets des facteurs. Ce n'est pas vraiment une surprise. Dans notre cas avec l'algorithme PSO, nous remarquons que les fonctions peuvent être assignées à des groupes. Le plus évident est un groupe composé des fonctions 13, 14 et 15. Ces fonctions partageront le même paramètre optimal. Il y a 3 autres groupes forts de 2 fonctions 3–12, 6–7 et 5–8. Nos résultats illustrent la difficulté de déterminer comment évaluer la qualité de l'ensemble des problèmes choisis. Les fonctions du benchmark (Tableau 5.2, page 47) utilisées dans notre étude sont regroupées en catégories qui ont été établies

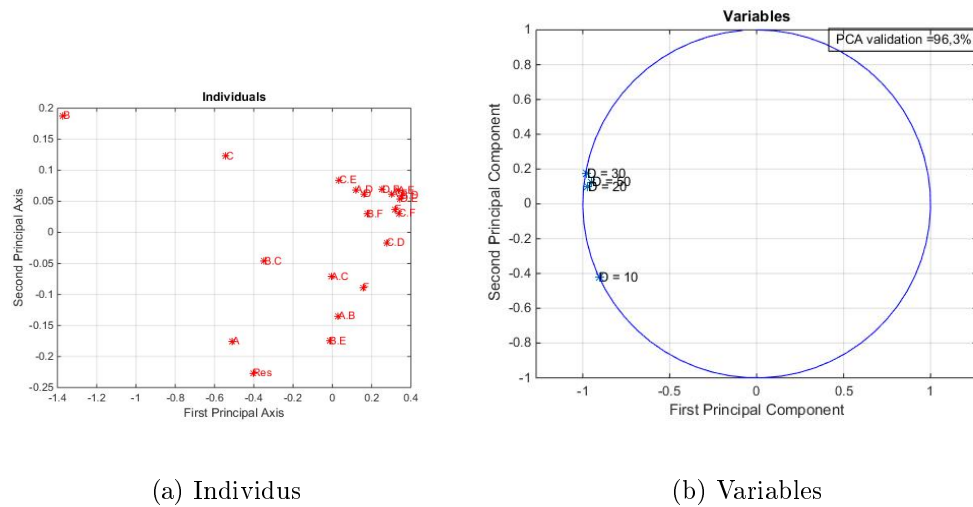


FIGURE 6.5 – ACP Cas 1 : dimension fixée

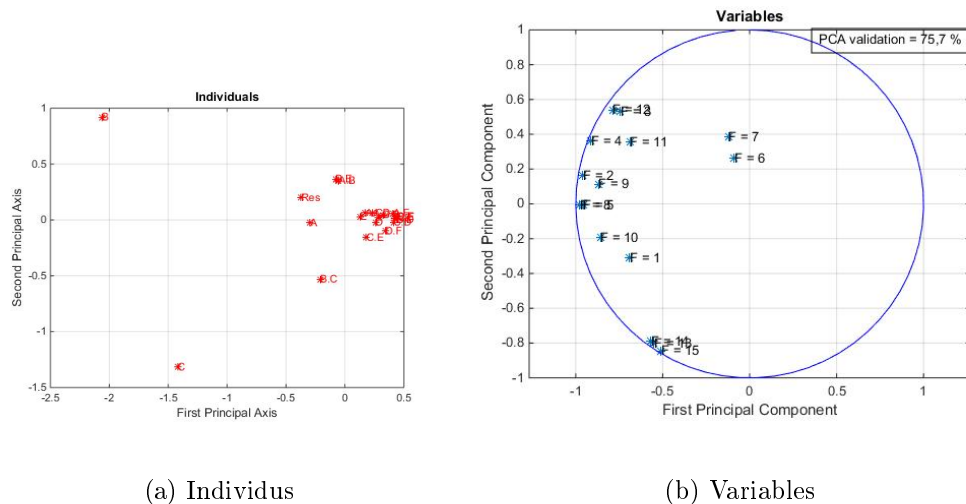
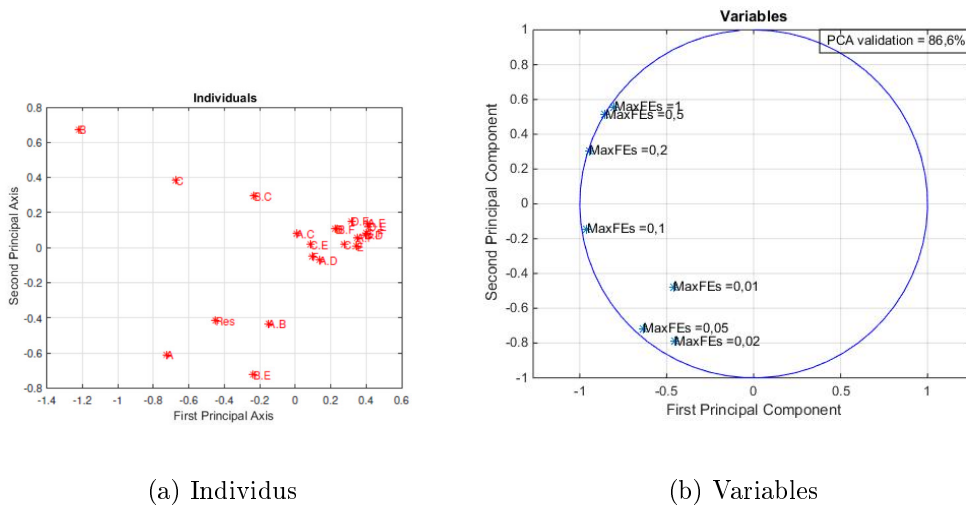


FIGURE 6.6 – ACP Cas 2 : fonction fixée

par la CEC [16]. Mais nos résultats ne reflètent pas ce classement. Cela nous autorise à penser que cette catégorisation du benchmark CEC et le choix des fonctions ne sont pas idéaux. L'absence de métrique pour caractériser les paysages de fonctions est probablement la raison principale de ceci. Cet aspect est confirmé par [32].

6.3.2.3/ Cas 3 : influence du nombre d'évaluations $MaxFes$

D'après la figure 6.7, les variables sont moins bien corrélées que dans le cas de la dimension. Cela signifie que la variation de $MaxFes$ (entre 0.01 et 1) a plus d'influence sur l'effet des paramètres algorithmiques que la variation de la dimension. Si l'on regarde plus en détail, si l'on exclut le cas où $MaxFes$ est au plus bas (0,01), on constate que ces variables sont ordonnées. En ce qui concerne les individus, les facteurs B et C ont de plus en plus d'effet lorsque $MaxFes$ augmente. Cette tendance s'inverse cependant en ce qui concerne l'effet du facteur A , de l'interaction BE , de l'interaction AB ainsi que de la partie résiduelle Res .

FIGURE 6.7 – ACP Cas 3 : *MaxFes* fixé

6.3.2.4/ Exploitation des différents plans d'expériences

L'exploitation des autres plans permet de déduire des paramètres optimaux pour chaque sous-score du benchmark. Ces paramètres optimaux sont présentés dans le Tableau 6.5. Pour la plupart des sous-scores, le paramètre optimal est le même que celui du score global. Cependant, ce n'est pas le cas pour certains sous-scores de fonctions. Pour celles-ci, les changements de valeur des paramètres semblent encourager la recherche globale. Ceci est logique car la plupart d'entre elles sont des fonctions multimodales.

	N	c_1	c_2	w_{\min}	w_{\max}	V_{factor}
Score Global	20	0.5	1.5	0.2	0.7	0.3
Sous-scores par dimensions	20	0.5	1.5	0.2	0.7	0.3
Sous-scores par <i>MaxFes</i>	20	0.5	1.5	0.2	0.7	0.3
Sous-scores par fonctions 3, 4, 8, 10, 11, 12, 13, 14, 15	20	0.5	1.5	0.2	0.7	0.3
Sous-scores par fonctions 5, 6, 9	-	-	-	0.4	-	-
Sous-scores par fonction 1	-	1.5	-	0.4	-	0.05
Sous-scores par fonction 2	-	-	0.5	-	-	-
Sous-scores par fonction 7	-	-	0.5	0.4	-	-

TABLE 6.5 – Paramètres optimaux par sous-ensembles

6.3.3/ Validation

Dans cette partie, nous souhaitons valider les jeux de paramètres déterminés précédemment. La validation est un véritable défi en soi ! La littérature scientifique regorge de paramètres optimisés pour telle ou telle situation, telle ou telle fonction. Il est impossible de les passer tous en revue. Il faut donc faire des choix. Nous allons donc comparer les scores obtenus avec nos paramètres et des jeux de paramètres issus de la littérature. Les jeux de paramètres retenus pour notre comparaison sont synthétisés dans le tableau 6.6.

Paramètres	N	c_1	c_2	w_{\min}	w_{\max}	V_{factor}
Paramètres de Khosla [50]	100	2	2	0.2	0.9	0.1
Paramètres de Wang[88]	25	2	1	0.1	0.3	0.1
Paramètres de Das [21]	50	2	2	0.9	1.2	0.1

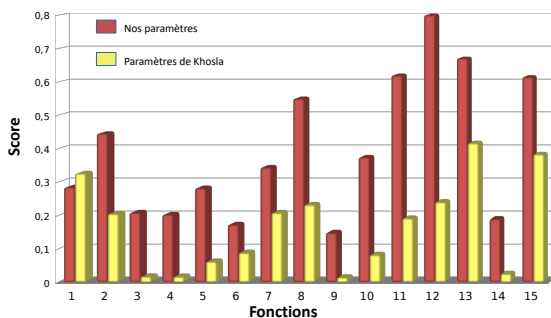
TABLE 6.6 – Jeux de paramètres du PSO retenus pour validation de notre approche

Les résultats obtenus (Tableau 6.5) nous permettent d’affirmer que nous disposons d’un jeu de paramètres de nature différente : un jeu de paramétrage généraliste, robuste dans la plupart des situations et des paramétrages plus ciblés dits locaux. La structure du benchmark combinant l’étude locale et globale conduit à cette situation. Ainsi, notre stratégie de validation va se dérouler en temps comme décrit dans ce qui suit.

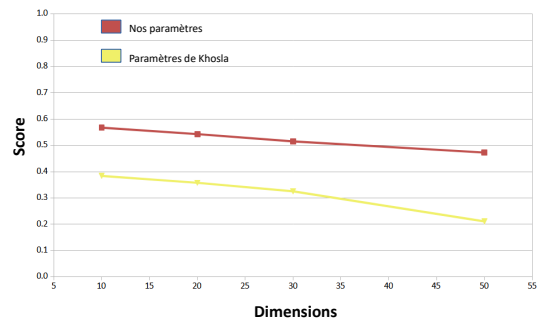
6.3.3.1/ Validation par sous-ensembles fonctions et dimensions

Nous procédons ici à une validation du jeu de paramètres pour des études ciblées. Nous nous focalisons donc sur les fonctions et les dimensions. Pour ce faire, nous calculons les sous-scores par fonctions et par dimensions évalués avec notre jeu de paramètres (Tableau 6.5) avec le même sous-score calculé cette fois avec un jeu de paramètres issus de la littérature pour une même version de PSO et obtenus par un plan d’expériences [50], nommés ici paramètres de Khosla (Tableau 6.6).

La figure 6.8a présente l’efficacité des approches pour les fonctions du benchmark et la figure 6.8b représente l’efficacité par rapport aux dimensions. Dans [50], seules deux fonctions sont testées (fonction de Rosenbrock et de Griewank). Nous avons décidé de comparer les jeux de paramètres sur l’ensemble des 15 fonctions de notre benchmark. La première conclusion est claire : notre approche est bien meilleure sur les deux fonctions initiales mais aussi sur l’ensemble du panel. L’approche proposée a de meilleurs résultats que toutes les autres approches testées pour chaque dimension étudiée.



(a) Scores par fonctions



(b) Scores par dimensions

FIGURE 6.8 – Validation par sous-ensembles

6.3.3.2/ Validation score global

L'idée ici est de vérifier que le jeu de paramètres obtenu à partir du score global est réellement robuste. Pour ceci, nous allons comparer le score global calculé avec notre jeu de paramètres et le score global obtenu avec des jeux de paramètres obtenus pour des cas dits réels : paramètres de Wang [88] et paramètres de Daas [21]. La figure 6.9 illustre le score global obtenu pour chacun des jeux de paramètres. Les résultats de Wang et de Das sont surpassés de 10%. On peut en conclure que notre approche semble pertinente.

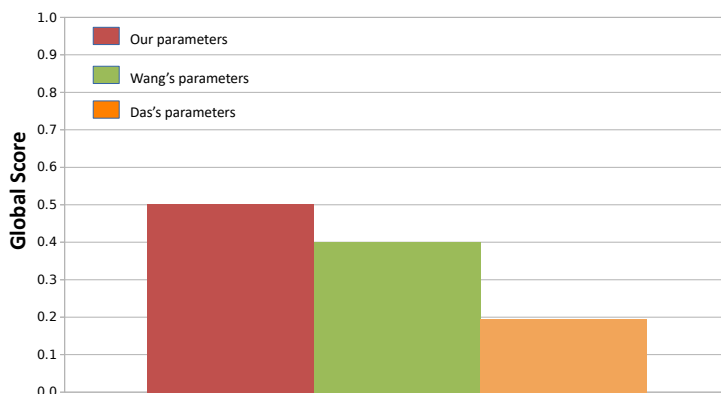


FIGURE 6.9 – Validation par score global

6.4/ Conclusions

L'amélioration de l'efficacité d'un algorithme par le réglage de ses paramètres est toujours d'actualité pour preuve les publications récentes [87]. Le réglage des paramètres a toujours été un sujet complexe, d'autant plus que l'efficacité d'un réglage dépend fortement du problème. La méta-optimisation est une méthode intéressante pour ajuster les paramètres. Dans notre travail, deux objectifs sont visés : obtenir un paramétrage générique efficace et obtenir une collection de paramétrages alternatifs pour pouvoir définir les paramètres d'un problème spécifique. L'algorithme PSO a été choisi pour tester cette approche car il est efficace et couramment utilisé. L'originalité des développements proposés repose sur les outils mis en place : un benchmark original associé à des méthodes statistiques d'apparence simple mais au combien efficaces.

OPTIMISATION TOPOLOGIQUE EN CONCEPTION

Sommaire

7.1	Introduction	75
7.2	Structures innovantes	76
7.2.1	Structure nid-d'abeilles	76
7.2.2	Métamatériaux électromagnétiques	78
7.3	Optimisation topologique et fabrication additive	82
7.4	Conclusion	84

7.1/ Introduction

Les parties précédentes ont mis en évidence mon travail autour de l'optimisation en conception mécanique. Dans ce contexte, il paraissait difficile de ne pas s'intéresser au domaine de l'optimisation de structures en particulier l'optimisation topologique, même si nous pouvons dire que ce type d'optimisation est à part. L'optimisation topologique consiste à mettre la bonne quantité de matière au bon endroit et pour la bonne raison. En terme de formulation mathématique, pour revenir aux concepts introduits dans le chapitre 3 (page 7), nous avons la formulation du problème comme suit :

- ❖ Fonction objectif
 - ✗ Masse, Volume, compliance ...
- ❖ Variables
 - ✗ La topologie!
- ❖ Contraintes
 - ✗ Issues des exigences du cahier des charges.
 - ✗ Issues des procédés de fabrication.

D'un point de vue mathématique, la principale difficulté inhérente à la résolution de ce problème est de parvenir à définir les zones composées de matière et celles qui doivent être vides. Dans un contexte éléments finis (2D ou 3D), un problème d'optimisation topologique de structure continue consiste à identifier les éléments qui doivent être composés de matière

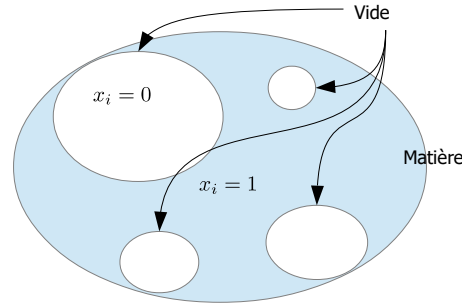


FIGURE 7.1 – Optimisation topologique

et ceux qui doivent être vides afin de minimiser la fonction objectif tout en tenant compte des contraintes comme présenté dans l'équation suivante :

$$(P) \left\{ \begin{array}{ll} \min F(\mathbf{X}) & \text{avec } \mathbf{X} = (x_1, x_2, \dots, x_n) \\ \text{soumis à} & g_j(\mathbf{X}) \leq 0 \quad \text{avec } j = 1, \dots, p \\ & h_i(\mathbf{X}) = 0 \quad \text{avec } i = 1, \dots, q \\ & x_i \in \{0, 1\} \quad \forall i = 1, \dots, n \end{array} \right. \quad (7.1)$$

Dans cette expression n est le nombre d'éléments de la structure et $\mathbf{X} \in \mathbb{R}^n$ est le vecteur des variables binaires qui représente la présence ($x_i = 1$) ou l'absence ($x_i = 0$) de matière dans chaque élément i (Figure 7.1).

Nous ne présentons pas ici, les différentes approches développées pour résoudre ce type de problème. Nous renvoyons le lecteur aux références [25, 10, 44].

7.2/ Optimisation topologique au service de la conception de structures innovantes

Les avancées scientifiques majeures de ces dernières années dans les domaines de la mécanique et des matériaux s'effectuent à la frontière de ces deux champs disciplinaires, en mettant notamment en place des approches multi-échelles permettant d'établir des connexions entre l'analyse microstructurale d'une part et la conception et à la fabrication de pièces mécaniques à l'échelle macroscopique de l'ingénieur d'autre part. Dans ce contexte, le contrôle, la caractérisation, la modélisation numérique multi-échelle des (micro)structures pouvant être évolutives jouent un rôle primordial. Le développement des outils de simulation performants intégrant des techniques avancées d'optimisation topologique apparaît pertinent. L'optimisation topologique peut conduire à un choix judicieux des géométries, obtenant ainsi des structures/matériaux innovants.

7.2.1/ Structure nid-d'abeilles

Dans un travail mené en collaboration avec le Pr. Zhang au travers de l'encadrement de la thèse de Kepeng Qiu [?], l'optimisation topologique a été mise en œuvre pour concevoir

des solides cellulaires à périodicité (structures nid d'abeille).

Il est connu que les performances structurelles des solides cellulaires à microstructures périodiques dépendent de la géométrie de l'élément de volume représentatif (VER) concerné. Dans le travail proposé, l'optimisation topologique a été utilisée à l'échelle de la cellule pour maximiser la rigidité des solides cellulaires sous une contrainte de volume. La méthode SIMP (Solid Isotropic Material with Penalization) a été utilisée pour résoudre le problème d'optimisation topologique. Pour conserver la périodicité structurelle et réduire le temps de calcul, le concept de super élément est introduit pour caractériser le VER et, dans le même temps, le problème d'optimisation topologique est résolu.

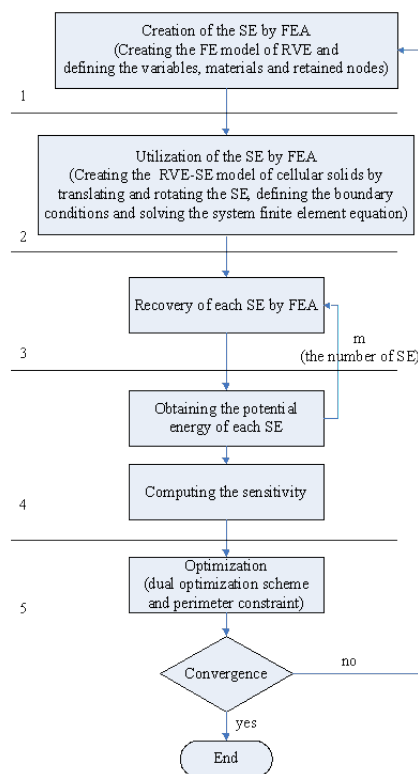


FIGURE 7.2 – Démarche de la procédure SE - VER - optimisation topologique

Une vue d'ensemble de la procédure de conception complète comprenant cinq phases (création du SE, utilisation du SE, récupération de chaque SE, analyse de sensibilité et optimisation) est 7.2.

Un exemple de résultat obtenu pour des structures carrées est illustré par la figure 7.3.

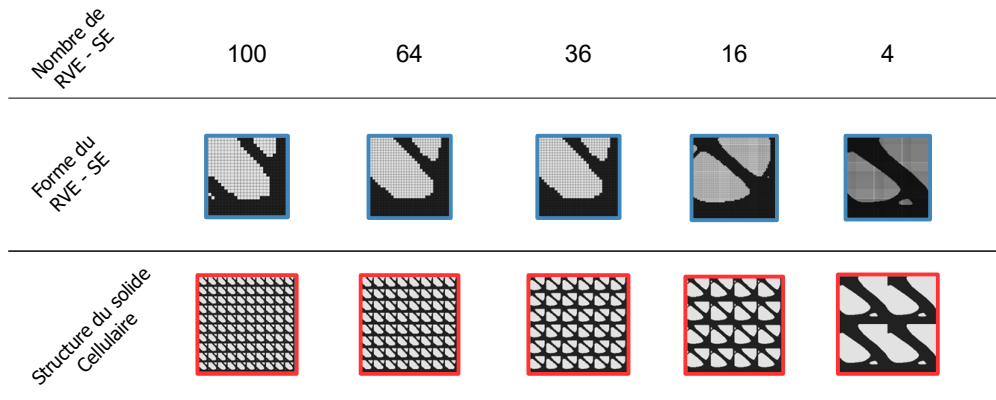


FIGURE 7.3 – Structure cellulaire carrée 2D optimisée topologiquement

Tous les détails de l'approche proposée ont été publiés dans [?] et dans la thèse de Kepeng Qiu [72].

7.2.2/ Métamatériaux électromagnétiques

Au fur et à mesure que mes connaissances/compétences sur les métaheuristiques progressaient, une suite logique a été de tenter de mettre en place les métaheuristiques étudiées dans le cadre binaire de l'optimisation topologie. Pour aller plus loin dans notre réflexion sur l'optimisation topologique nous avons voulu l'appliquer à un domaine particulier et original : les métamatériaux électromagnétiques.

De la même manière que les métamatériaux mécaniques, les métamatériaux 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 répartition intelligente de ces deux matériaux qui peut présenter ces caractéristiques électromagnétiques, lorsque la structure est illuminée par un champ électromagnétique.

Les difficultés relatives à ce problème sont de trois natures :

- ❖ Établir un modèle numérique.
- ❖ Établir la formulation du problème d'optimisation topologique.
- ❖ Choisir une méthode de résolution : quelle méthode pour résoudre ce problème d'optimisation topologique.

Dans ce contexte, nous avons mis en place un algorithme d'optimisation par essaim particulaire binaire, afin de développer une cape d'invisibilité électromagnétique, c'est à dire un métamatériau à perméabilité négative. Pour ce faire, la partie réelle et la partie imaginaire

de cette perméabilité sont minimisées, à une fréquence d'impact du champ électromagnétique donnée.

7.2.2.1/ Modélisation numérique

Notre modélisation repose sur une modélisation éléments finis. Les métamatériaux 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 7.4a. Du fait que les équations de Maxwell peuvent être considérées comme continues une fois discrétisées par la méthode des éléments finis, dans l'ensemble du matériau, on peut n'étudier qu'une seule cellule de périodicité du métamatériau considéré. Des conditions aux limites particulières seront employées afin de vérifier l'homogénéité du modèle, comme décrit sur la figure 7.4b. Le modèle d'une cellule de périodicité, basé sur les équations de Maxwell, a été développé. 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. Le détail des équations et de la formulation du problème sont détaillés dans notre article [?].

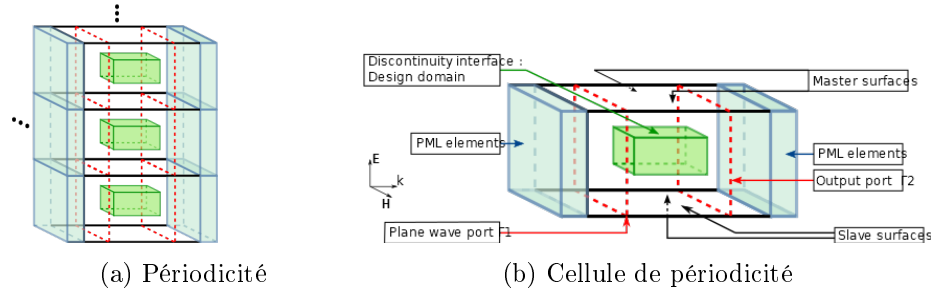


FIGURE 7.4 – Modèle numérique d'un métamatériau électromagnétique

7.2.2.2/ Formulation du problème d'optimisation

Le processus d'optimisation topologique a pu être appliqué à une cellule de périodicité afin de déterminer quelle doit être la répartition de matériau diélectrique dans le domaine. Pour ce faire, nous minimisons conjointement les parties réelle et imaginaire de la perméabilité effective :

$$\mu_{eff} = \mu' + j \mu'' \quad (7.2)$$

où μ_{eff} est la perméabilité effective, μ' sa partie réelle, et μ'' sa partie imaginaire. Le problème d'optimisation se traite en deux temps. On minimise 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 minimise sa partie réelle comme évoqué par les équations suivantes :

$$\begin{aligned} & \min \mu'' \\ & \text{Respect des équations gouvernant le problème} \\ & \text{Respect conditions aux limites} \end{aligned} \quad (7.3)$$

et

$$\begin{aligned} & \min \mu' \\ & \text{Respect des équations gouvernant le problème} \\ & \text{Respect conditions aux limites} \end{aligned} \quad (7.4)$$

La perméabilité du métamatériau est calculée à partir des paramètres de dispersion (the scattering parameters), aussi appelés S-parameters.

7.2.2.3/ Résolution

Dans ce travail, la topologie est considérée ici comme binaire. Une version binaire de l'algorithme PSO a été utilisée [61] pour résoudre le problème. Dans ce cadre précis, chaque particule est une potentielle solution du problème d'optimisation, à savoir une répartition de la matière, donc une topologie. La gestion binaire de la matière s'appuie sur le maillage éléments finis. Il n'existe que deux possibilités pour chacun des éléments du maillage (Figure 7.5) :

- ❖ être plein, c'est à dire être constitué du matériau diélectrique précédemment défini (état 1),
- ❖ être vide, c'est à dire être constitué du background material (état 0).

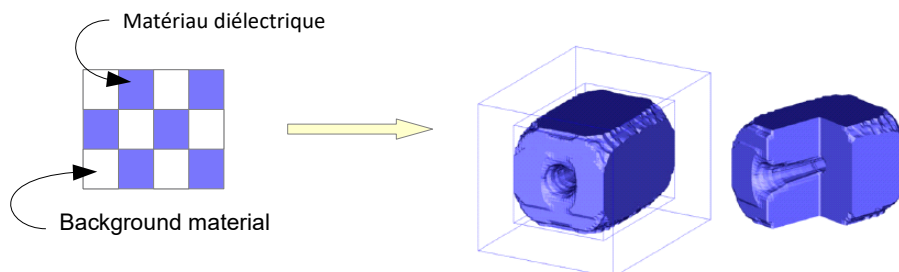


FIGURE 7.5 – Topologie binaire pour les métamatériaux électromagnétiques

La différence majeure entre la version standard continue du PSO, page 24, équation (4.2), et la version binaire, se situe dans la mise à jour de la position qui prend la forme suivante :

$$\mathbf{X}_i^{k+1} = \begin{cases} 0 & \text{si } rand() \geq S(\mathbf{V}_i^{k+1}) \\ 1 & \text{si } rand() < S(\mathbf{V}_i^{k+1}) \end{cases} \quad (7.5)$$

Dans cette expression,

- ❖ S est la fonction Sigmoïde appliquée à la vitesse des particules comme suit :

$$S(\mathbf{V}_i^{k+1}) = \frac{1}{1 + e^{\mathbf{V}_i^{k+1}}} \quad (7.6)$$

- ❖ $rand()$ est un nombre aléatoire distribué suivant une loi Normale.

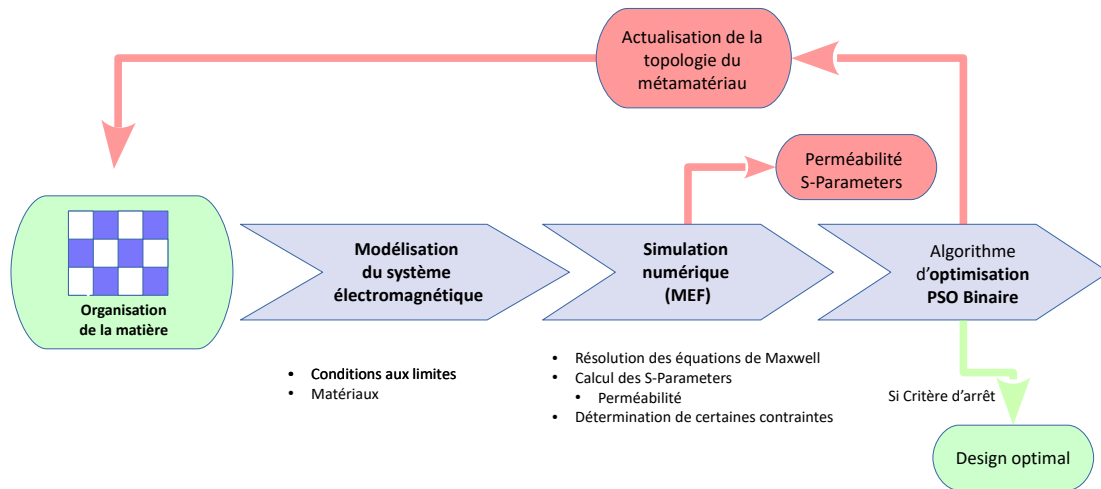


FIGURE 7.6 – Couplage PSO binaire et méthode des éléments finis

Une des difficultés dans la mise en place de cette stratégie est le couplage avec la méthode des éléments finis. En effet, 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. L'organigramme du développement est détaillé dans la figure pour une particule. Ceci est à réitérer pour chaque particule et pour chaque itération (Figure 7.6). Le détail de ces travaux peut être trouvé dans [?, ?].

7.2.2.4/ Résultats

Un exemple de topologie obtenue après minimisation de la perméabilité effective à 10MHz est illustrée sur la figure

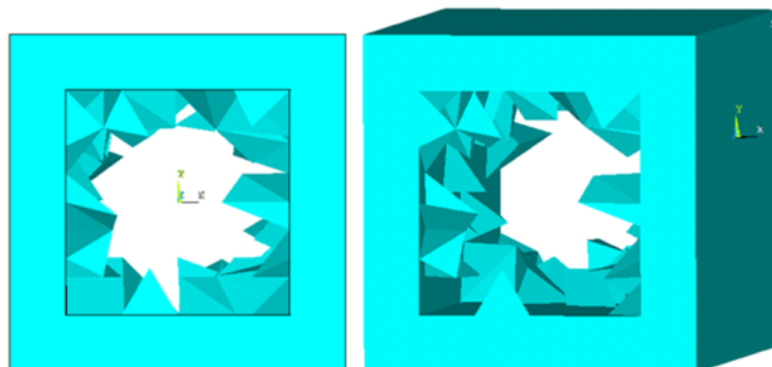


FIGURE 7.7 – Métamatériaux : Topologie obtenue après optimisation

7.3/ Optimisation topologique et fabrication additive

Depuis début 2017 (date d'intégration de ma précédente équipe au laboratoire ICB département COMM), un axe fort de recherche a émergé au sein du laboratoire ICB et notamment au sein du département COMM : la fabrication additive (FA) et le **4D** printing. Dans ce contexte, il m'est apparu judicieux de mettre à profit mes compétences en simulation numérique en mécanique non-linéaire et en optimisation en particulier ce qui concerne l'optimisation topologique associée à la fabrication additive de manière connexe à ce que qui était en cours.

La fabrication additive offre aux concepteurs un espace de conception sans précédent. Cet espace de conception peut être pleinement exploité par les potentialités offertes par l'optimisation topologique. Les méthodes d'optimisation topologique constituent un outil numérique idéal pour déterminer automatiquement des formes optimales sans avoir de pré-requis sur la structure initiale. Les géométries obtenues par optimisation topologique, et sa capacité à découvrir des solutions nouvelles et performantes, en font un outil de conception intimement lié à la fabrication additive.

Même si la fabrication additive ouvre considérablement l'espace de conception et offre aux concepteurs des avantages tels que la réduction du temps de production et des économies considérables de matériaux, il reste des contraintes et des limitations de fabrication qui doivent finalement être intégrées dans la méthodologie d'optimisation topologique pour tirer pleinement parti des capacités et des libertés offertes par ce processus de fabrication innovant [30, 53, 33]. En effet, le concepteur doit être amené à imaginer de nouvelles solutions créatives mais il faut surtout s'assurer que celles-ci soient réalistes et réalisables.

Nous sommes focalisés sur l'une de ces contraintes de fabrication : la contrainte de surplomb (overhang). La Figure 7.8 montre un exemple de deux géométries de pièces caractérisées par un angle critique de porte à faux différent. Un angle supérieur à 45° implique le besoin de supports supplémentaires. De ce fait, pour la fabrication additive, il est plus intéressant de concevoir des géométries qui peuvent s'auto-supporter et donc qui ne nécessitent pas de supports additionnels.

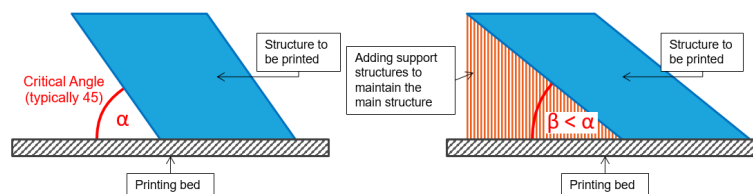


FIGURE 7.8 – Pièces avec différents angles d'inclinaison et besoins de supports

Ce travail s'inscrit dans le cadre d'un mécénat de compétences avec la société Altran. Donc, tous les développements ont été faits dans le code interne de la société Altran appelé ELiOT. Les algorithmes implémentés dans ce code reposent sur les travaux de M. Bendsoe, and O. Sigmund. [11, 83]. Son fonctionnement est décrit rapidement sur la figure 7.9.

L'idée développée est d'introduire un filtre de calcul de géométrie imprimable dans le processus itératif d'optimisation. Nous proposons le concept d'éléments de support ainsi qu'un paramètre appelé centre de gravité (COG) qui permet à l'utilisateur d'introduire un maillage non structuré dans le processus de conception. Cette fonction est une fonction

manquante dans [53, 29] qui limite l'utilisateur à un maillage structuré. La nouveauté de notre travail est de calculer le centre de gravité de chaque élément, une fois le centre de gravité de chaque élément défini, les couches d'impression sont définies en fonction de la direction d'impression. En plus des critères mentionnés, nous définissons un angle critique pour chaque élément qui balaie les éléments de la couche inférieure en une forme conique. Tous les éléments qui sont simultanément inclus dans la couche inférieure immédiate et dont le centre de gravité est balayé par le cône de l'angle critique sont considérés par le code comme des éléments de support. L'algorithme proposé est synthétisé par la Figure 7.10.

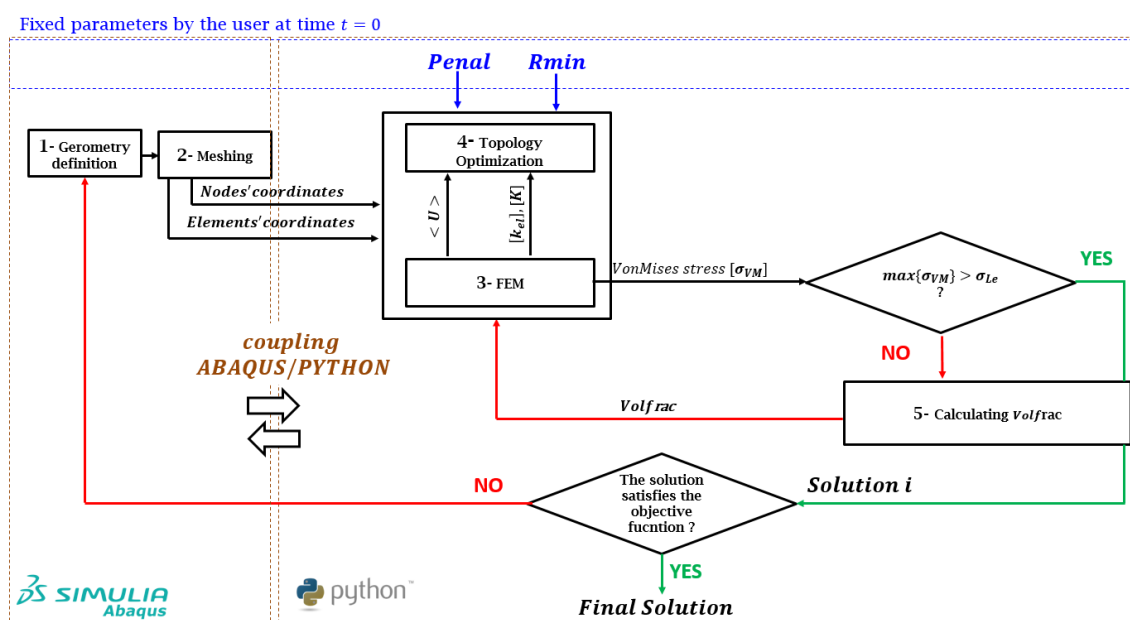


FIGURE 7.9 – Code ELiOT : implémentation d'un algorithme d'optimisation topologique

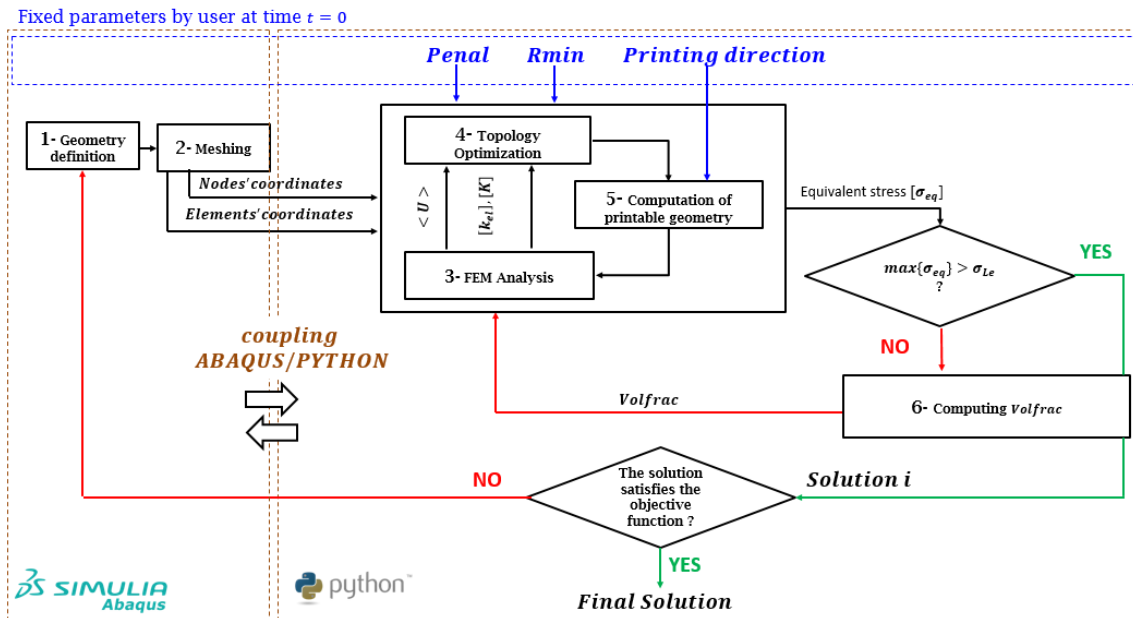


FIGURE 7.10 – Code ELiOT : implémentation d’un algorithme d’optimisation topologique avec prise en compte de la contrainte d’overhang

Ce travail a donné lieu à une publication [?], tous les détails y sont mentionnés.

7.4/ Conclusion

Dans ce chapitre, nous avons évoqué l’optimisation topologique et son application aux structures innovantes ainsi qu’à la fabrication additive. Les deux concepts sont forcément liés : l’optimisation topologique conduit à des formes associées à des fonctionnalités nouvelles, on peut parler de structures intelligentes.

CONCLUSIONS ET PERSPECTIVES

Dans ce mémoire, ont été présentés les travaux que j'ai pu mener ces dernières années autour de l'optimisation en conception. Je terminerai ce document par les axes que je souhaite développer dans les années à venir autour de l'optimisation mais aussi autour de la simulation numérique de systèmes complexes.

8.1/ Formulation et certification des algorithmes d'optimisation

Après m'être intéressée au développement de nouvelles approches notamment métaheuristiques (Chapitre 4), il ressort deux points fondamentaux pour vraiment amener l'optimisation au cœur de la conception :

- ❖ Être capable de formuler un problème d'optimisation est encore un problème ouvert. Pour parvenir à traduire mathématiquement un problème à partir des besoins des concepteurs, nous nous sommes intéressés à la manière de recueillir les données mais ceci n'est clairement pas suffisant. Il faut maintenant passer à la gestion des données et des connaissances, autrement dit à la mise en place de stratégies pour aller plus loin en utilisant notamment ce que nous avons fait dans le cadre de la thèse de Julien Badin [7]. L'idée est de parvenir à mettre en œuvre une approche permettant le partage des données, des informations et des connaissances en temps réel, en tenant compte de leur niveau de maturité (KCMModel [?]).
- ❖ La certification et la robustesse des algorithmes sont fondamentales dans un contexte d'optimisation dit difficile. C'est, dans cet objectif, que le benchmark a été développé (Chapitre 5). Le calcul du score global et la possibilité d'obtenir des sous-scores sont des points originaux. Cependant ce benchmark peut et doit être clairement enrichi, par l'ajout de cas plus concrets, en particulier issus de la mécanique. Il peut même servir de base à de nombreux domaines d'applications.

8.2/ Structures intelligentes et 4D printing

Une des applications de l'optimisation que je souhaite développer est la mise en œuvre de l'optimisation pour la conception de structures dites intelligentes. Ce travail a déjà débuté sous différentes formes mais devrait s'intensifier dans un futur proche.

Optimisation de smart composites

Au travers mon implication dans le projet ANR ASTRIA (??, page ??), je m'intéresse à l'optimisation de Smart composites. L'objectif de ce travail est le développement des structures composites adaptables à leur environnement.

Optimisation et 4D printing

Un axe fort a émergé ces dernières années dans mes travaux : l'optimisation topologique et notamment son application, en particulier à l'impression 3D. Mais, il nous semble maintenant pertinent d'aller plus loin et de se diriger vers l'impression 4D, thématique forte du Laboratoire ICB/CO2M. Ce travail est déjà initié en collaboration avec le Pr. Demoly et s'est concrétisé par l'encadrement de deux stages de master (Section ??, page ??). Ceci a donné lieu à la participation à un workshop [?], un article soumis et au dépôt d'un projet EIPHI (Projet BOOST-4D¹), Bourgogne Franche-Comté² en juin 2022. Un financement d'encadrement de thèse a été demandé. L'impression 4D, apparue en 2013 (> 40 de croissance/an), associe la fabrication additive avec des matériaux actifs via une stimulation énergétique (e.g. température, lumière, solvant, pH, humidité, champ électrique/magnétique, etc.). Les objets (ou structures) ainsi créés disposent de capacités de changement de propriétés et/ou de formes exploitant l'énergie dans leur environnement d'usage. Au-delà d'effets d'annonce illustrés dans la littérature scientifique, des faiblesses de performances des actionneurs 4D existent : temps de stimulation, réversibilité, répétabilité (fatigue fonctionnelle), encombrement du stimulateur artificiel, résistance mécanique, etc. Les efficacités modestes – qui limitent l'atteinte du marché – sont améliorables par une approche multi-matériaux. L'idée est de travailler sur l'agencement numérique des matériaux et des stimuli en conception pour atteindre un changement de forme souhaité et introduit la question de la maîtrise de l'optimisation topologique multi-échelle et des métamatériaux afin d'atteindre des performances maîtrisées en problèmes inverses. Un couplage avec des techniques de machine Learning est également proposé afin d'améliorer la performance des procédures de calcul (Figure 8.1).

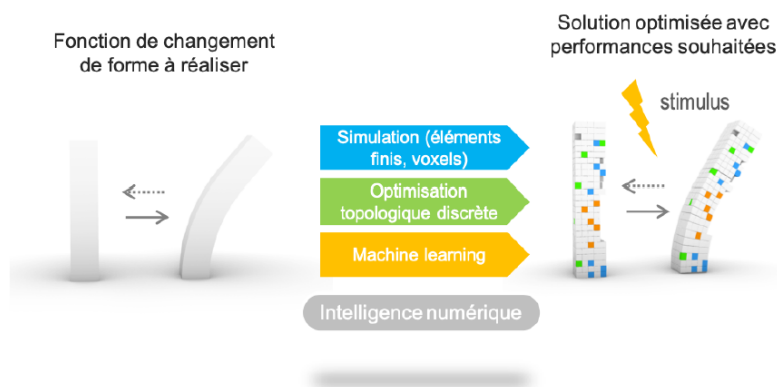


FIGURE 8.1 – Projet 4D printing et optimisation topologique

1. Amélioration de l'efficacité des actionneurs par impression 4D pour l'industrie du futur
 2. Programme d'Investissements d'Avenir " École Universitaire de Recherche" EIPHI

8.3/ Simulation et optimisation des systèmes pile à combustible

Un deuxième axe fort que je souhaite continuer à développer voire intensifier est l'optimisation d'un système complexe particulier : le système pile à combustible (PAC). La plupart des études antérieures concernant les piles à combustible se sont focalisées sur les aspects physico-chimiques des piles. Les aspects mécaniques ont longtemps été placés au second plan malgré leurs impacts forts sur les performances. Depuis une dizaine d'années, les couplages mécaniques sont au cœur des études. L'objectif majeur de mon travail est de mettre en place des outils numériques afin de comprendre l'influence des incertitudes des paramètres mécaniques (épaisseur de la GDL, rayon de courbure de la plaque bipolaire et porosité de la GDL) pour optimiser la performance et la durabilité de la pile. Une grandeur joue un rôle fondamental dans la performance de la pile : la pression de contact entre les composants de la pile. Il est donc nécessaire de porter une attention particulière à la répartition des pressions de contact entre GDL/BPP et GDL/MEA. Ce travail a débuté lors de la thèse de Ghinwa Oudat [68] et s'est poursuivi par le stage de master de Luca Marcelli. Ce dernier devrait continuer en thèse en octobre 2022 suite à l'obtention d'un contrat doctoral avec l'Université Paris-Saclay (Co-encadrement avec Denis Candusso, Yann Meyer).

L'objectif général de mes travaux est d'aider à l'amélioration de systèmes mécaniques. L'application de l'optimisation est infinie, pour preuve mes débuts en biomécanique. A chaque verrou technologique levé, de nouvelles perspectives apparaissent et mes activités actuelles pourraient m'amener vers d'autres domaines d'applications.

RÉFÉRENCES

- [1] Chapter 10 - metaheuristic algorithms : A comprehensive review. In A. K. Sangaiah, M. Sheng, and Z. Zhang, editors, *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 185–231. Academic Press, 2018.
- [2] O. Ait Elhara. *Stochastic Black-Box Optimization and Benchmarking in Large Dimensions*. PhD thesis, Université Paris Saclay (COMUE), 2017.
- [3] J. Alexis. *Pratique industrielle de la méthode Taguchi*. Association Française de Normalisation (AFNOR), 1995.
- [4] M. F. Ashby. *Materials Selection in Mechanical Design*. Butterworth-Heinemann, Oxford, fourth edition, 2011.
- [5] R. N. Asl, M. Aslani, and M. S. Panahi. Sizing optimization of truss structures using a hybridized genetic algorithm, 2013.
- [6] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Y. Qu. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. Technical report, 2016.
- [7] J. Badin. *Ingénierie hautement productive et collaborative à base de connaissances métier : vers une méthodologie et un méta-modèle de gestion des connaissances en configurations*. PhD thesis, Université de Technologie de Belfort-Montbéliard, 2011.
- [8] V. Beiranvand, W. Hare, and Y. Lucet. Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18(4) :815–848, 2017.
- [9] J. Bénabès. *Optimisation intégrée et interactive pour l’agencement d’espace*. PhD thesis, Ecole Centrale de Nantes (ECN), 2011.
- [10] M. P. Bendsoe and O. Sigmund. *Topology Optimization : Theory, Methods and Applications*. Springer, Feb. 2004.
- [11] M. P. Bendsoe and O. Sigmund. *Topology Optimization : Theory, Methods and Applications*. Springer, Feb. 2004.
- [12] C. Bessiere. Chapter 3 - constraint propagation. In P. v. B. Francesca Rossi and T. Walsh, editors, *Handbook of Constraint Programming*, volume Volume 2, pages 29–83. Elsevier, 2006.
- [13] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1) :107–117, 1998. Proceedings of the Seventh International World Wide Web Conference.
- [14] H. Chen, Q. Feng, X. Zhang, S. Wang, Z. Ma, W. Zhou, and C. Liu. A meta-optimized hybrid global and local algorithm for well placement optimization. *Computers & Chemical Engineering*, 117 :209 – 220, 2018.
- [15] J. L. Chen and C.-C. Liu. An eco-innovative design method by green qfd and triz tools. In *DS 31 : Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm*, 2003.

- [16] Q. Chen, B. Liu, Q. Zhang, J. Liang, P. N. Suganthan, and B. Qu. Problem definition and evaluation criteria for cec 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization. Technical report, Oct. 2013.
- [17] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao. Benchmark functions for the cec 2017 competition on many-objective optimization. Technical report, 2017.
- [18] D. Choulier and P.-A. Weite. *Découvrir et appliquer les outils de TRIZ*. UTBM Editions, 2011.
- [19] A. Collignan. *Méthode d'optimisation et d'aide à la décision en conception mécanique : application à une structure aéronautique*. PhD thesis, 2011.
- [20] B. Crawford, R. Soto, E. Monfroy, G. Astorga, J. García, and E. Cortes. A meta-optimization approach for covering problems in facility location. In J. C. Figueroa-García, E. R. López-Santana, J. L. Villa-Ramírez, and R. Ferro-Escobar, editors, *Applied Computer Sciences in Engineering*, pages 565–578, Cham, 2017. Springer International Publishing.
- [21] A. Das, A. Majumder, and P. K. Das. Detection of apposite PSO parameters using taguchi based grey relational analysis : Optimization and implementation aspects on manufacturing related problem. *Procedia Materials Science*, 6(Complete) :597–604, 2014.
- [22] K. A. De Jong. *Analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, USA, 1975.
- [23] N. Di Césaire. *Développement d'une nouvelle méthode metaheuristique pour l'optimisation topologique des structures et des metamatériaux*. PhD thesis, Université de Technologie de Belfort-Montbéliard, 2016.
- [24] J. Dréo, A. Chatterjee, A. Pétrowski, P. Siarry, and E. Taillard. *Metaheuristics for Hard Optimization : Methods and Case Studies*. Springer Berlin Heidelberg, 2006.
- [25] P. Duysinx. *Optimisation topologique : du milieu continu a la structure elastique*. PhD thesis, ULiège - Université de Liège, 1996.
- [26] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS '95.*, pages 39–43, 1995.
- [27] A. Eesa, A. Mohsin Abdulazeez Brifcani, and Z. Orman. A novel bio-inspired optimization algorithm. *International Journal of Scientific & Engineering Research*, 4 :1978–1986, 2013.
- [28] A. El Dor. *Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en électronique*. PhD thesis, Université Paris-Est, Dec. 2012.
- [29] E. Fernández, C. Ayas, M. Langelaar, and P. Duysinx. Topology optimisation for large-scale additive manufacturing : generating designs tailored to the deposition nozzle size. *Virtual and Physical Prototyping*, 16(2) :196–220, 2021.
- [30] A. Garaigordobil, R. Ansola, E. Veguería, and I. Fernandez. Overhang constraint for topology optimization of self-supported compliant mechanisms considering additive manufacturing. *Computer-Aided Design*, 109 :33–48, 2019.
- [31] C. García-Martínez, P. D. Gutiérrez, D. Molina, M. Lozano, and F. Herrera. Since cec 2005 competition on real-parameter optimisation : a decade of research, progress and comparative analysis's weakness. *Soft Computing*, 21(19) :5573–5583, Oct. 2017.

- [32] R. W. Garden and A. P. Engelbrecht. Analysis and classification of optimisation benchmark functions and benchmark suites. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1641–1649. IEEE, 2014.
- [33] A. T. Gaynor and J. K. Guest. Topology optimization considering overhang constraints : Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5) :1157–1172, 9 2016. Preliminary results of this study were presented at the 15th AIAA/ISSMO MAO Conference at Aviation 2014, June 16-20, 2014, Atlanta, Georgia, USA ; and at WCSMO-11, June 7-11, 2015, Sydney, Australia.
- [34] W. J. Gomes, A. T. Beck, R. H. Lopez, and L. F. Miguel. A probabilistic metric for comparing metaheuristic optimization algorithms. *Structural Safety*, 70 :59–70, 2018.
- [35] N. Hansen. The cma evolution strategy : a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.
- [36] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010 : Experimental setup. Research Report RR-7215, 2010.
- [37] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockhoff. Coco : A platform for comparing continuous optimizers in a black-box setting. ArXiv e-prints, arXiv :1603.08785, July 2016.
- [38] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. Coco : a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1) :114–144, 2021.
- [39] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-Parameter Black-Box Optimization Benchmarking 2009 : Noiseless Functions Definitions. Research Report RR-6829, 2009.
- [40] N. Hansen, T. Tusar, O. Mersmann, A. Auger, and D. Brockhoff. Coco : The experimental procedure. ArXiv e-prints, arXiv :1603.08776, 2016.
- [41] K. R. Harrison, B. M. Ombuki-Berman, and A. P. Engelbrecht. Optimal parameter regions for particle swarm optimization algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017.
- [42] M. H. Hobballah, A. Ndiaye, F. Michaud, and M. Irle. Formulating preliminary design optimization problems using expert knowledge : Application to wood-based insulating materials. *Expert Systems with Applications*, 92 :95–105, 2018.
- [43] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992.
- [44] X. Huang and Y. M. Xie. *Evolutionary Topology Optimization of Continuum Structures*. John Wiley & Sons, Ltd, 2010.
- [45] Y. Huang, Y. Jin, and Y. Ding. New performance indicators for robust optimization over time. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1380–1387, May 2015.
- [46] A. Jaafari, M. Panahi, B. T. Pham, H. Shahabi, D. T. Bui, F. Rezaie, and S. Lee. Meta optimization of an adaptive neuro-fuzzy inference system with grey wolf optimizer and biogeography-based optimization algorithms for spatial prediction of landslide susceptibility. *CATENA*, 175 :430 – 445, 2019.
- [47] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.
- [48] A. Kaveh and A. Zolghadr. Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints. *Advances in Engineering Software*, 76 :9–30, 2014.

- [49] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1671–1676 vol.2, 2002.
- [50] A. Khosla, S. Kumar, and K. Aggarwal. Identification of strategy parameters for particle swarm optimizer through taguchi method. *Journal of Zhejiang University-SCIENCE A*, 7(12) :1989–1994, 2006.
- [51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, 1983.
- [52] U. Škvorc, T. Eftimov, and P. Korošec. Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. *Applied Soft Computing*, 90 :106138, 2020.
- [53] M. Langelaar. Topology optimization of 3d self-supporting structures for additive manufacturing. *Additive Manufacturing*, 12 :60–70, 2016.
- [54] A. N. Langville and C. D. Meyer. Langville and meyer : Deeper inside pagerank. 2004.
- [55] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization : harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194(36-38) :3902–3933, 2005.
- [56] H. Li, Z. Huang, X. Liu, C. Zeng, and P. Zou. Multi-fidelity meta-optimization for nature inspired optimization algorithms. *Applied Soft Computing*, 96 :106619, 2020.
- [57] L. Li, Z. B. Huang, F. Liu, and Q. H. Wu. A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, 85 :340–349, 2007.
- [58] J. Liang, B. Qu, and P. Suganthan. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635, 2013.
- [59] Q. Liu, W. V. Gehrlein, L. Wang, Y. Yan, Y. Cao, W. Chen, and Y. Li. Paradoxes in numerical comparison of optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 24(4) :777–791, 2020.
- [60] P. Loubiere. *Amélioration des métaheuristiques d’optimisation à l’aide de l’analyse de sensibilité*. PhD thesis, 2016. Thèse de doctorat dirigée par Siarry, Patrick et Chelouah, Rachid Informatique Paris Est 2016.
- [61] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin. A binary particle swarm optimization for continuum structural topology optimization. *Appl. Soft Comput.*, 11 :2833–2844, 2011.
- [62] N. Lynn, M. Z. Ali, and P. N. Suganthan. Population topologies for particle swarm optimization and differential evolution. *Swarm and Evolutionary Computation*, 39 :24–35, 2018.
- [63] I. Mazhoud, K. Hadj-Hamou, J. Bignon, and P. Joyeux. Particle swarm optimization for solving engineering problems : A new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, 26(4) :1263–1273, 2013.
- [64] R. Mendes, J. Kennedy, and J. Neves. Watch thy neighbor or how the swarm can learn from its environment. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, pages 88–94, 2003.
- [65] P. K. Newton, J. Mason, K. Bethel, a. J. N. Lyudmila A Bazhenova, and P. Kuhn. A stochastic markov chain model to describe lung cancer growth and metastasis. *PLoS One*, 4, 2012.

- [66] A.-T. Nguyen, S. Reiter, and P. Rigo. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113(Supplement C) :1043 – 1058, 2014.
- [67] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi. Fuzzy self-tuning pso : A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*, 2017.
- [68] G. Ouaidat. *Intégration des incertitudes des paramètres de conception dans les modèles mécaniques d'une pile à combustible*. PhD thesis, Université de Technologie de Troyes, 2020.
- [69] A. P. Piotrowski and J. J. Napiorkowski. Some metaheuristics should be simplified. *Information Sciences*, 427 :32 – 62, 2018.
- [70] A. P. Piotrowski, M. J. Napiorkowski, J. J. Napiorkowski, and P. M. Rowinski. Swarm intelligence and evolutionary algorithms : Performance versus speed. *Information Sciences*, 384 :34–85, 2017.
- [71] R. Poli and D. Broomhead. Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, pages 134–141, New York, NY, USA, 2007. ACM.
- [72] K. Qiu. *Analysis and optimal design of lightweight sandwich structures and materials*. PhD thesis, Université de Technologie de Belfort-Montbéliard ; Northwestern Polytechnical University (Chine), 2008.
- [73] B. Qu, J. Liang, Z. Wang, Q. Chen, and P. Suganthan. Novel benchmark functions for continuous multimodal optimization with comparative results. *Swarm and Evolutionary Computation*, 26 :23–34, 2016.
- [74] T. Quirante, P. Sebastian, and Y. Ledoux. A trade-off function to tackle robust design problems in engineering. *Journal of Engineering Design*, 24(1) :64–81, 2013.
- [75] H. Rajabi Moshtaghi, A. Toloie Eshlaghy, and M. R. Motadel. A comprehensive review on meta-heuristic algorithms and their classification with novel approach. *Journal of Applied Research on Industrial Engineering*, 8(1) :63–89, 2021.
- [76] T. Richard and N. Muszynski. Outil d'aide à la traduction des besoins concepteurs pour la formulation des problèmes d'optimisation. Projet de recherche – TX55, Université de Technologie de Belfort-Montbéliard, 2019.
- [77] R. Roy, S. Hinduja, and R. Teti. Recent advances in engineering design optimisation : Challenges and future trends. *CIRP Annals*, 57(2) :697–715, 2008.
- [78] M. A. Salido. A non-binary constraint ordering heuristic for constraint satisfaction problems. *Applied Mathematics and Computation*, 198(1) :280–295, Apr. 2008.
- [79] S. Salmon. *Caractérisation, identification et optimisation des systèmes mécaniques complexes par mise en œuvre de simulateurs hybrides matériels-logiciels*. PhD thesis, Université de Technologie de Belfort Montbéliard, 2012.
- [80] S. Salmon. Particle swarm optimization toolbox, 2021.
- [81] F. Schott. *Contribution to robust and fiabilist optimization : Application to the design of acoustic radio-frequency filters*. PhD thesis, Université Bourgogne Franche-Comté, 2019.
- [82] X. H. Shi, Y. H. Lu, C. G. Zhou, H. P. Lee, W. Z. Lin, and Y. C. Liang. Hybrid evolutionary algorithms based on pso and ga. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2393–2399 Vol.4, Dec 2003.

- [83] O. Sigmund. A 99 line topology optimization code written in matlab. 21(2) :120–127.
- [84] K. Sorensen. Metaheuristics - the metaphor exposed. *International Transactions in Operational Research*, 22(1) :3–18, 2012.
- [85] C. A. Souza Lima, C. M. F. Lapa, C. M. d. N. Pereira, J. J. da Cunha, and A. C. M. Alvim. Comparison of computational performance of ga and pso optimization techniques when designing similar systems – typical pwr core case. *Annals of Nuclear Energy*, 38(6) :1339–1346, 2011.
- [86] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical report, 2005.
- [87] J. Tondut, C. Ollier, N. Di Cesare, J. Roux, and S. Ronel. An automatic kriging machine learning method to calibrate meta-heuristic algorithms for solving optimization problems. *Engineering Applications of Artificial Intelligence*, 113 :104940, 2022.
- [88] H. Wang, Q. Geng, and Z. Qiao. Parameter tuning of particle swarm optimization by using taguchi method and its application to motor design. In *2014 4th IEEE International Conference on Information Science and Technology*, pages 722–726, Apr. 2014.
- [89] T. Weise, R. Chiong, J. Lassig, K. Tang, S. Tsutsui, W. Chen, Z. Michalewicz, and X. Yao. Benchmarking optimization algorithms : An open source framework for the traveling salesman problem. *IEEE Computational Intelligence Magazine*, 9(3) :40–52, Aug. 2014.
- [90] Y. Xu. *Prediction of properties and optimal design of microstructure of multi-phase and multi-layer C/SiC composites*. PhD thesis, Université de Technologie de Belfort Montbéliard et Northwestern Polytechnical University (Chine), 2011.
- [91] X.-S. Yang, S. Koziel, and L. Leifsson. Computational Optimization, Modelling and Simulation : Past, Present and Future. *Procedia Computer Science*, 29(Supplement C) :754 – 758, 2014.

Résumé :

Dans un contexte industriel extrêmement concurrentiel, avec des cahiers des charges de plus en plus exigeants, l'approche systémique permet d'envisager des solutions originales et jusqu'à alors inexplorées. Cette méthode allie de manière intime simulations, expérimentations, interactions entre sous-systèmes et cycles de conception à tous les niveaux afin d'obtenir un produit plus performant, mieux intégré et de réduire les délais de mise sur marché. Ce constat a servi de socle au développement de mon projet de recherche dans le cadre de l'optimisation. La direction principale des travaux de recherche présentés dans ce manuscrit est de développer un/des outils performants pour accompagner au mieux l'ingénieur dans sa démarche d'optimisation en tenant compte des contraintes de l'univers industriel dans un environnement de conception multiphysique et multidisciplinaire tout en gardant la rigueur mathématique liée aux méthodes d'optimisation.

Mots-clés : Optimisation, Conception, métaheuristique, Performances, Robustesse