



HAL
open science

Formalisation et exploration partielle de problèmes de satisfaction de contraintes en conception préliminaire

Thomas Richard de Latour

► **To cite this version:**

Thomas Richard de Latour. Formalisation et exploration partielle de problèmes de satisfaction de contraintes en conception préliminaire. Algorithmes et structure de données [cs.DS]. Nantes Université, 2023. Français. NNT : 2023NANU4047 . tel-04612444

HAL Id: tel-04612444

<https://theses.hal.science/tel-04612444v1>

Submitted on 14 Jun 2024

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.

COLLEGE ED MATHÉMATIQUES

DOCTORAL STIC

PAYS DE LA LOIRE



THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 641

Mathématiques et Sciences et Technologies du numérique,

de l'Information et de la Communication

Spécialité : *Recherche Opérationnelle*

Formalisation et exploration partielle de problèmes de satisfaction de contraintes en conception préliminaire

Par

Thomas RICHARD DE LATOUR

Thèse présentée et soutenue à Nantes, le 11 juillet 2023

Unité de recherche : Laboratoire des Sciences du Numérique de Nantes, UMR6004

Rapporteurs avant soutenance :

Élise VAREILLES Professeur à ISAE-SUPAERO (Toulouse)

Bertrand NEVEU Directeur de Recherche à École des Ponts ParisTech

Composition du Jury :

Président : Pierre-Alain YVARS Professeur des Universités à ISAE-SUPMECA

Examineurs : Élise VAREILLES Professeur à ISAE-SUPAERO (Toulouse)

Bertrand NEVEU Directeur de Recherche à École des Ponts ParisTech

Dir. de thèse : Laurent GRANVILLIERS Professeur des Universités à Université de Nantes

Co-dir. de thèse : Raphaël CHENOUEAU Maître de conférences à École Centrale de Nantes

REMERCIEMENT

De nombreuses personnes ont participé directement et indirectement à ce travail. Je tiens à les remercier.

Merci à mes encadrants, Laurent Granvilliers et Raphaël Chenouard, pour la confiance que vous m'avez accordée dès les premiers jours. Avec vos qualités respectives, vous formez un binôme à la fois exigeant et rassurant. Votre disponibilité et nos discussions scientifiques pendant ces trois ans ont été essentielles.

Merci à Élise Vareilles et Bertrand Neveu d'avoir accepté de rapporter une version très imparfaite de ce manuscrit et dans des délais particuliers. Vos commentaires et nos discussions ont permis d'améliorer grandement le fond et la forme. Merci aussi à Pierre-Alain Yvars, président du Jury, d'avoir pris ce rôle avec intérêt. J'espère que nous aurons l'occasion de poursuivre nos échanges scientifiques.

Merci aux OGREs du LS2N pour votre dynamisme et votre bienveillance. C'était un vrai plaisir de travailler dans votre équipe. Je tiens à remercier particulièrement mes co-doctorants. Rémi qui a endossé parfaitement le rôle de guide au sein du monde universitaire. Adrien, Thomas (bientôt doctorant ?) et Alessandro pour nos échanges scientifiques, mais pas que !

Merci aux membres du projet OSCAR pour ces années de travail en commun. Plus spécialement, merci à Nicolas pour ton implication et nos nombreux échanges enfin fructueux !

Merci à la secte Nantaise Centraliennes des anciens et anciennes du rugby pour votre amitié sans failles.

Merci aux gars du VSN pour toutes ces saisons, dans la victoire comme dans la défaite !

Merci à toutes celles et ceux que j'ai croisés ces dernières années à Nantes, Rennes, Lyon ou ailleurs. Soyez certains de mon amitié.

Merci Tritus d'avoir supporté ces presque quatre ans de collocation, tu es un chic type !

Enfin, merci à mes parents, mes frères et ma sœur pour votre soutien indispensable et votre fierté communicative. Je suis à mon tour fier de chacun de vous !

TABLE DES MATIÈRES

Liste des figures	7
Liste des tables	9
Glossaire	11
I Introduction	13
I.1 Contexte	13
I.2 Le projet OSCAR	15
I.3 Questions de recherche	15
I.4 Structure du document	16
II État de l’art	19
II.1 Le dimensionnement en conception préliminaire	19
II.1.1 L’objet de la conception : le système	19
II.1.2 Le processus de conception	21
II.1.3 Un langage pour la conception : les modèles	23
II.1.4 Le modèle analytique, support de la conception architecturale	27
II.1.5 Les variables en conception	28
II.2 Cadre formel pour le modèle de dimensionnement en conception architecturale	30
II.2.1 Problème de satisfaction de contraintes ou CSP	30
II.2.2 La modélisation des critères de performance	32
II.2.3 Formalisation d’un modèle analytique	35
II.3 Résolution de problème de satisfaction de contraintes	38
II.3.1 Résolution d’un CSP Numérique	38
II.3.2 Exploration de l’espace de recherche	41
II.3.3 Exploration partielle de l’espace de recherche	47
II.3.4 Performance d’un algorithme d’exploration partielle en conception	49
II.4 Bilan	51
III Formalisation de modèles mathématiques en conception préliminaire	53
III.1 Introduction	53
III.1.1 Apports et limites de l’existant	54
III.1.2 Propositions du chapitre	56
III.2 Qualification d’un modèle CSP en conception	57
III.2.1 Caractéristiques décisionnelles d’un modèle CSP	57

III.2.2	Pouvoir décisionnel d'un modèle de dimensionnement CSP	61
III.2.3	Vérification	61
III.3	Approche de formalisation d'un CSP en conception	62
III.3.1	Principe de la méthodologie globale	62
III.3.2	Pré-modélisation	63
III.3.3	Formalisation du modèle initial	64
III.3.4	Agrégation des modèles complémentaires	65
III.3.5	Implémentation et qualification du modèle	66
III.4	Application au problème de la batterie	67
III.4.1	Pré-modélisation	69
III.4.2	Formalisation du problème initial	72
III.4.3	Agrégation des modèles complémentaires	74
III.4.4	Synthèse du modèle	80
III.5	Bilan du chapitre	82
IV	Exploration partielle d'espace de recherche	85
IV.1	Introduction	85
IV.1.1	Motivation	85
IV.1.2	Contexte et objectif	86
IV.2	Indicateurs de qualité et profils de performance	87
IV.2.1	Construire un indicateur de diversité	88
IV.2.2	Uniformité	89
IV.2.3	Couverture	90
IV.2.4	Précisions sur l'utilisation des indicateurs	91
IV.3	Les stratégies de sélection pour des algorithmes Anytime Branch and Prune	91
IV.3.1	Stratégie hybride DFS/BestFS générique	91
IV.3.2	Nouvelles stratégies de sélection	94
IV.3.3	Adimensionnement des critères de tri	95
IV.4	Les autres composants impactants d'un algorithme AB&P pour l'exploration partielle	97
IV.4.1	Choix de la variable à diviser	97
IV.4.2	Distinction des types de variables	98
IV.4.3	Gestion de la précision des variables	99
IV.4.4	Méthode de descente DFS	101
IV.4.5	Gestion des CSP mixtes	101
IV.4.6	Backtrack forcé	103
IV.5	Comparaison des stratégies de sélection AB&P : exemples simples	103
IV.5.1	Problème des <i>m boules</i> : modèle numérique	105
IV.5.2	Problème des <i>m boules</i> : modèle mixte	108
IV.6	Comparaison des stratégies ABPA : benchmark de problèmes de conception	109
IV.6.1	Protocole	109
IV.6.2	Résultats globaux	110

IV.6.3	Profils de performance	113
IV.7	Bilan du chapitre	114
V	Exploration et aide à la décision	115
V.1	Introduction	115
V.1.1	Contexte	115
V.1.2	Approches d'optimisation utilisées	115
V.1.3	Protocole expérimental	116
V.2	La machine à aimants permanents sans encoche	119
V.2.1	Description du modèle	119
V.2.2	Résolution	120
V.2.3	Comparaison des résultats	122
V.3	Le moteur à combustion	123
V.3.1	Description du problème	123
V.3.2	Première exploration de l'espace réalisable	124
V.3.3	Exploration du problème réduit	125
V.3.4	Comparaison avec les résultats de l'optimisation du problème réduit	127
V.4	Le pack Batterie pour véhicule électrique	128
V.4.1	Exploration de l'espace réalisable du problème <i>batterie</i>	129
V.4.2	Sélection de quelques architectures représentatives	130
V.4.3	Analyse de l'espace réalisable	130
V.5	Bilan du chapitre	132
VI	Conclusion	135
VI.1	Contributions et limites	135
VI.1.1	Comment formaliser un modèle mathématique adapté pour la conception préliminaire d'un système complexe ?	135
VI.1.2	Comment explorer efficacement l'espace réalisable d'un CSP numérique ou mixte ?	136
VI.1.3	L'exploration partielle diversifiée est-elle un outil pertinent pour traiter des problèmes de conception architecturale ?	137
VI.2	Perspectives	138
VI.3	Contributions	140
A	Annexe chapitre 1	141
B	Annexe chapitre 2	142
B.1	Agrandissement des BPMN du processus de formalisation	142
B.2	Catalogue de références de cellules li-ion LFP et NMC pour le problème batterie	142
B.3	Modèle minibex du problème de conception de batterie décrit dans [137]	142
C	Annexe chapitre 3	149
C.1	Résultats de l'exploration des problèmes des <i>m boules</i> sous la forme d'un CSP mixte	149
C.2	Problèmes du benchmark	150

D Annexe chapitre 4	151
D.1 Fonctionnement des algorithmes NSGA	151
D.1.1 Les algorithmes évolutionnistes	151
D.1.2 Les algorithmes génétiques	151
D.2 Problème <i>mapse</i>	154
D.3 Problème <i>ice</i>	156
D.4 Problème <i>batterie</i>	157
D.4.1 Calcul de P_{peak}	157
D.4.2 Calcul du vieillissement cyclique	157
D.4.3 Post-traitement de l'exploration (<i>batterie</i>)	158
Bibliographie	161

TABLE DES FIGURES

I.1	Dépassement des limites planétaires	13
I.2	Émissions globales de gaz à effet de serres	14
II.1	Décomposition structurelle d'un système complexe.	20
II.2	Représentation simplifiée du processus de conception.	21
II.3	Les différents diagrammes SysML	24
II.4	Différentes modélisations du concept <i>batterie</i>	25
II.5	Principe général d'un modèle de dimensionnement.	26
II.6	Positionnement des types de modèles sur le processus de conception.	27
II.7	Classification des variables du concept batterie	29
II.8	Espace de conception et espace des objectifs	29
II.9	Phase de formalisation de modèle en conception préliminaire	35
II.10	Vue simplifiée du processus de vérification et validation	36
II.11	Formalisation d'un modèle de dimensionnement selon Vernat	36
II.12	Un boîte $[x]$ de \mathbb{R}^2	40
II.13	Exemple de pavage d'un CSP numérique simple	40
II.14	Construction de l'arbre de recherche avec un algorithme Branch & Prune (B&P)	42
II.15	Propagation et évaluation des contraintes avec <i>HC4</i>	45
II.16	Exploration partielle du problème <i>cerceau</i> avec différentes stratégies de sélection	48
II.17	Comment mesurer la diversité du résultat d'une exploration partielle?	50
III.1	Formalisation en conception préliminaire	53
III.2	Démarche de formalisation de Vernat	55
III.3	Représentation de l'étendue	58
III.4	Principe de la proposition de démarche de formalisation d'un CSP en conception	63
III.5	Phase 2 de formalisation du modèle initial	64
III.6	Phase 3 d'agrégation des modèles complémentaires (MC)	66
III.7	Phase 4 d'implémentation et de qualification du modèle	67
III.8	Modélisations du problème initial de <i>batterie</i> par des diagrammes SysML	70
III.9	Modélisation de la batterie comme une combinaison de sources de tension idéales	72
IV.1	Comparaison des propriétés attendues pour le résultat d'une exploration partielle.	88
IV.2	Sélection de boîtes à explorer après un backtrack	93
IV.3	Illustration du comportement des stratégies de sélection	96
IV.4	Exemple précision relative	100
IV.5	Exemples méthodes de descente sur le problème du <i>cerceau</i>	102

IV.6	Diagramme composants ABPA	104
IV.7	Profils de performance de Q_a et Q_h (300 boules numérique)	107
IV.8	Profils de performance des stratégies d'exploration partielle	113
V.1	Représentation de l'hypervolume dans l'espace des objectifs	117
V.2	Protocole comparaison	117
V.3	Schéma du problème du moteur à aimants permanents (<i>mapse</i>)	119
V.4	Profils de performance de l'exploration du problème <i>mapse</i>	121
V.5	GCP des sous-ensembles de solutions <i>mapse</i>	122
V.6	Comparaison exploration et optimisation <i>mapse</i>	123
V.7	Schéma du moteur à combustion interne à tête plate (<i>ice</i>)	124
V.8	Première exploration de <i>ice</i>	125
V.9	163 solutions représentatives du problème <i>ice</i>	126
V.10	Comparaison de l'exploration et de l'optimisation de <i>ice</i>	127
V.11	Profils de performance de l'exploration du problème <i>batterie</i>	129
V.12	GCP de l'échantillon après l'exploration partielle du CSP de la batterie	131
V.13	Restriction de l'espace réalisable à certaines plages de valeurs du problème <i>batterie</i>	132
A.1	Modélisation de l'état interne de la batterie (<i>batterie</i>) avec le digramme SysML IBD	141
B.1	Phase 3 d'agrégation des modèles complémentaires (MC)	143
B.2	Phase 2 de formalisation du modèle initial	144
C.1	Profils de performance 100 boules numérique	149
D.1	Fonctionnement générique d'un algorithme génétique	152
D.2	Schéma du fonctionnement de NSGA-II [38]	152
D.3	Couverture et hypervolume de la première exploration du problème <i>ice</i>	157
D.4	Mise en évidence des architectures de la référence de cellule $n_{cell} = 3$	158
D.5	Mise en évidence des architectures avec une autonomie supérieure à $REV \geq 400$ km	159

LISTE DES TABLEAUX

II.1	Cahier des charges initial pour la conception d'un pack batterie Li-ion	21
III.1	Définition des propriétés de la complexité d'un modèle	59
III.2	Caractéristiques du pouvoir décisionnel d'un modèle de dimensionnement CSP	62
III.3	Tableau des spécifications d'un concept	64
III.4	Nomenclature du problème de conception de la batterie	67
III.4	Nomenclature du problème de conception de la batterie	68
III.5	Tableau des spécifications de la batterie	71
III.6	Complexité des spécifications du concept du pack batterie	72
III.7	Complexité du modèle initial (MI) du pack batterie	74
III.8	Quelques solutions du modèle initial de la batterie	74
III.9	Coefficients pour le calcul des indicateurs environnementaux (données EcoInvent)	76
III.10	Catalogue de systèmes de refroidissement	79
III.11	Complexité du modèle initial (MI) du pack batterie	80
III.12	Données de référence pour l'exactitude de la batterie I	81
III.13	Comparaison référence/modèle pour évaluer l'exactitude de la batterie	81
IV.1	Résultat de l'exploration partielle du problème numérique des m boules (n=m)	106
IV.2	Nombre de boules et temps de calcul moyen pour le problème des 100 boules	107
IV.3	Ecart type du nombre de solutions par boules, pour 300 ϵ -boîtes réparties dans 100 boules	108
IV.4	Nombre de fois où les stratégies de sélection et de bisection ont donné le meilleur résultat	110
IV.5	Meilleures combinaisons (stratégie) pour chaque problème selon les valeurs de \sum^p	111
IV.6	Moyenne des résultats de l'exploration partielle (benchmark)	111
IV.7	Moyenne des résultats de l'exploration partielle pour les combinaisons (benchmark)	112
V.1	Paramétrage initial des algorithmes NSGA-II et NSGA-III	116
V.2	Nomenclature du problème <i>mapse</i>	119
V.3	Sous-ensembles de solutions du problème <i>mapse</i>	121
V.4	Nomenclature du problème <i>ice</i>	123
V.5	Nouveaux domaines initiaux des V_{co} du problème <i>ice</i>	126
V.6	Comparaison de deux architectures possibles pour le problème <i>ice</i>	127
V.7	Paramètres fixés pour le problème <i>batterie</i> avant la résolution du modèle	128
V.8	Quelques architectures pertinentes issues de l'exploration partielle du problème <i>batterie</i> .	130
B.1	Catalogue fournisseur de cellule Li-ion pour véhicule électrique.	142

LISTE DES TABLEAUX

C.1	Résolution du problème des m boules forme mixte	149
C.2	Écart type problème des m boules mixte	149
C.3	Benchmark de 30 problèmes de conception	150

GLOSSAIRE

AB&P Anytime Branch and Prune.

AC Acidification.

ACV Analyse de Cycle de Vie.

B&P Branch and Prune.

BDD Diagramme de définition de bloc SysML.

BestFS Best First Search, stratégies de sélection pour un algorithme AB&P.

CMOP Problème d'Optimisation Multi-objectif.

COP Problème d'optimisation mono-objectif.

CSP Problème de Satisfaction de Contraintes.

DFS Depth First Search, stratégie de parcours d'un arbre de recherche.

DLFS Depth and Largest First Search.

DMDFS Depth and Most Distant First Search.

DPFS Depth and Perimeter First Search.

DSE Exploration de l'Espace de Recherche (Design Space Exploration).

EU Eutrophisation.

GCP Graphique de Coordonnées Parallèles.

GES Gaz à Effet de Serre.

GWP Global Warming Potential.

IBD Diagramme de blocs internes SysML.

IDFS Inteleaved Depth First Search.

LF Largest First.

LFP Lithium-fer-phosphate (LiFePO₄), chimie de la cathode de batteries Lithium-ion.

MBSE Model Based System Engineering ou ingénierie des systèmes basée sur le modèle.

MC Modèle Complémentaire.

MCP Matériau de Changement de Phase.

MDFS Most Distant First Search.

MI Modèle Initial.

MINLP Mixed Integer Non Linear Programming.

MOP Multi-Objective Mixed-Integer Nonlinear Program.

NMC Lithium-Nickel-Manganese-Cobalt-Oxide (LiNiMnCoO₂), chimie de la cathode de batterie Lithium-ion.

NSGA Non-dominated Sorting Genetic Algorithm.

ParD Diagramme paramétrique SysML.

PEPS Parcimonie Exactitude Précision Spécialisation, Méthode de qualification de modèle.

PPC Programmation Par Contraintes.

RD Ressource Depletion.

RDFS Random Depth First Search.

ReqD Diagramme des exigences SysML.

RR Round Robin.

SSR Smear Sum Relative.

SysML Systems Modeling Language.

V&V Validation et Vérification, procédé de validation de modèle.

Vco Variable de conception.

Vp Variable de performance.

INTRODUCTION

I.1 Contexte

L'ingénierie système (IS)[186] fait face à un nombre croissant d'exigences pour concevoir de nouvelles solutions technologiques. Ce phénomène s'explique en partie par la réduction des ressources disponibles et par les contraintes environnementales. Ces problématiques sont à la fois locales et globales : changement climatique, acidification des océans, pollution de l'air, disparition de la biosphère, etc. Plusieurs travaux représentent les enjeux à l'échelle planétaire par 9 limites au-delà desquelles les conséquences pour l'humanité et son environnement deviennent néfastes (Figure I.1) [152, 166]. Les recherches les plus récentes concluent au franchissement de la cinquième limite depuis le début de l'ère industrielle : l'introduction de nouvelles entités dans l'environnement [139]. Certaines des conséquences du dépassement de ces seuils sont déjà observables, par exemple la pollution plastique [69] ou la fonte des glaciers de moyennes altitudes [65]. Or, les activités humaines sont à la racine de chacune de ces crises. En particulier, l'origine anthropique du réchauffement climatique est aujourd'hui un consensus scientifique, notamment explicité dans les travaux du GIEC (Groupe d'experts intergouvernemental sur l'évolution du climat) [83]. L'état de la planète, et plus précisément du vivant, est donc dépendant de notre capacité à réduire rapidement notre impact sur l'environnement.

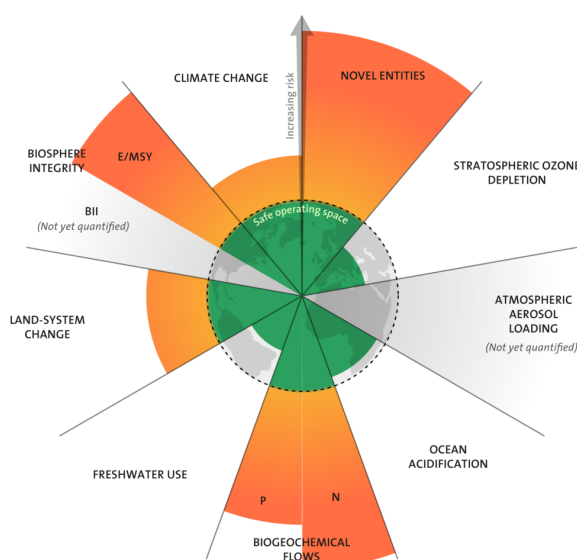


FIGURE I.1 – Le dépassement des limites planétaires en 2022 d'après [152]

Face à ces enjeux, plusieurs gouvernements se sont engagés lors des Accords de Paris de 2015 [135] à « réduire leurs émissions et à coopérer en vue de s'adapter aux effets des changements climatiques ». Ainsi, en plus des limitations matérielles dues à la diminution des ressources, des réglementations viennent contraindre tous les partis concernés. Par exemple, dans le cadre du pacte vert [75], les pays Européens prévoient d'interdire la vente de véhicule thermique après 2035 [148]. Le rôle multisectoriel de l'IS en fait un domaine crucial pour appliquer ces législations. En effet, les décisions prises pendant le processus de conception d'un produit déterminent une grande partie de son impact sur tout son cycle de vie. C'est-à-dire de l'extraction des matériaux à la fin de vie des composants. De plus, les impacts des secteurs liés au cycle de vie d'un produit sont aussi sensibles à sa composition. Les choix des concepteurs sont donc répercutés à la fois dans la production industrielle et dans les secteurs d'activités liés à l'utilisation des produits. C'est par exemple le cas dans le domaine automobile où les véhicules contribuent aux émissions de gaz à effets de serre à la fois dans le secteur des transports et dans le secteur industriel. En France, ces deux secteurs sont à l'origine de 54% des émissions¹ (Figure I.2).

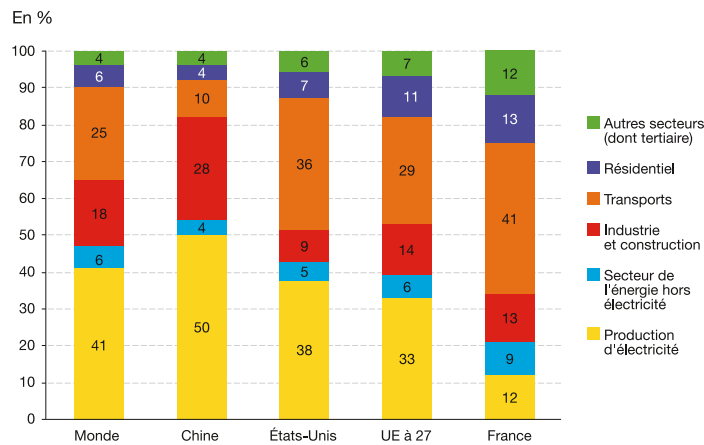


FIGURE I.2 – Émissions globales de gaz à effet de serre (GES) par secteur¹

Pour prendre les bonnes décisions, le cycle de vie du produit doit être considéré le plus tôt possible dans une démarche d'éco-conception. Le principe d'éco-conception est reconnu et défini dans la norme ISO 14006 :2020 [85] comme « l'intégration des aspects environnementaux dans la conception et le développement de produit avec pour objectif la réduction des impacts environnementaux négatifs tout au long du cycle de vie d'un produit ». Il s'agit d'ajouter aux exigences habituelles de performance, de coût et de délai, des contraintes et critères sur l'environnement. L'éco-conception repose principalement sur les résultats de l'Analyse de Cycle de Vie (ACV) [11, 72, 86]. L'ACV permet de faire le bilan environnemental d'un produit sur son cycle de vie et sur un ensemble de critères différents comme la qualité de l'air, l'épuisement des ressources ou l'acidification des océans. Cependant, l'ACV est alimentée par des bases de données sur l'existant, ce qui limite son utilité dans les phases de conception préliminaire où la flexibilité et la créativité sont essentielles. Intégrer un outil d'analyse dans ce processus n'est donc pas immédiat. De plus, les outils d'aide à la décision n'offrent pas la souplesse nécessaire pour prendre en considération des cycles de vie complets. C'est dans ce contexte que le projet ANR OSCAR (Optimisation pour des

1. <https://www.statistiques.developpement-durable.gouv.fr/>

Systemes Complexes Attractifs et duRables) a démarré en 2019.

I.2 Le projet OSCAR

Le projet de recherche a été motivé par deux constats. Le premier est celui de la difficulté d'intégrer les outils d'analyse environnementale en amont du processus de conception. L'ACV fonctionne initialement à partir d'architectures figées. L'enjeu est de concevoir le cycle de vie potentiel du produit en parallèle de son architecture système afin de trouver un équilibre entre impact et performance. Le second constat est l'absence d'un cadre souple et interactif pour étudier intelligemment la faisabilité d'un système. En effet, à ce stade du processus de conception, les modèles de dimensionnement sont approximatifs. Les comportements les plus complexes peuvent être omis de la modélisation, notamment les aspects environnementaux. Rechercher des optima à partir de ces modèles n'est donc pas toujours pertinent. Le concepteur risque de négliger des architectures quasi-optimales qui pourrait s'avérer intéressantes au regard de critères qui n'ont pu être modélisés. De plus, les méthodes d'optimisation existantes ne favorisent pas l'interaction recherchée avec le concepteur. Notre position est de laisser au concepteur l'opportunité de faire valoir son expertise dans le choix des architectures à concevoir. En particulier, lorsqu'il doit aussi considérer en parallèle les cycles de vie de ces architectures.

Le rôle de cette thèse est de répondre à la seconde problématique. Pour faciliter la diffusion des résultats au sein de la communauté d'IS, nous avons positionné nos travaux dans le processus de conception classique [132]. Le résultat global du projet OSCAR et plus particulièrement les contributions sur les cycles de vie et leur analyse ne sont pas abordées dans ce manuscrit. Les critères environnementaux sont considérés comme des exigences classiques du cahier des charges. De fait, nos contributions pourront s'appliquer dans des contextes autres que l'éco-conception.

I.3 Questions de recherche

La phase du processus de conception qui nous intéresse ici est la conception architecturale [132]. C'est une étude de faisabilité dont l'objectif est de réduire l'espace réalisable à quelques architectures préliminaires. Elle repose sur la résolution de modèles de dimensionnement représentant à la fois les comportements du système et les exigences du cahier des charges. Le domaine concerné est l'*exploration d'espace de recherche* (ou DSE pour design space exploration) [84, 92]. En conception préliminaire, l'espace de recherche de ces modèles est peu contraint et autorise encore un grand nombre de solutions architecturales. L'ensemble des solutions ne peut généralement pas être généré en un temps raisonnable. De plus, il n'est pas souhaitable de proposer au concepteur un trop grand nombre d'architectures au moment de la prise de décision. Les approches d'optimisation sont couramment utilisées, mais ne sont pas pertinentes dans le contexte du projet OSCAR. Outre les arguments avancés plus haut, le fonctionnement de ces solveurs est opaque et rend difficile la justification et la compréhension des décisions prises à partir de leurs résultats. Finalement, le paradigme de l'*exploration partielle diversifiée* est un bon candidat. L'objectif d'une exploration partielle est d'obtenir un sous-ensemble de solutions bien réparties dans l'espace réalisable [23]. Le résultat doit permettre au concepteur de se rendre compte de l'ensemble des possibles et sélectionner la ou les architectures candidates pour l'étape suivante du processus.

L'exploration partielle diversifiée est un domaine peu étudié en conception architecturale. Il faut donc construire le cadre approprié à son utilisation dans ce contexte. En amont de la résolution, nous avons aussi questionné l'existant sur la formalisation du modèle de dimensionnement [159, 183]. C'est une étape cruciale qui détermine fortement l'apport décisionnel des solutions du modèle. Le paradigme de la programmation par contraintes est le cadre le plus pertinent pour aborder ces sujets [117, 106, 180, 179]. Le modèle et sa résolution sont traités séparément. Le modèle est considéré comme un problème de satisfaction de contraintes (CSP) et la résolution repose sur des méthodes de réduction et de filtrage. De nombreux travaux abordent l'utilisation de CSP en conception [49, 205, 24, 198, 31, 71, 134]. Il reste cependant de nombreux verrous pour appliquer la programmation par contraintes en conception, nous en avons relevé deux en particulier. Le premier est l'adéquation du problème avec le modèle CSP : est-ce qu'il répond bien au besoin formulé dans le cahier des charges ? Le deuxième est lié aux caractéristiques des CSP formulés. Ceux-ci cumulent souvent variables discrètes, variables continues, contraintes non-linéaires, contraintes logiques, etc. Le solveur utilisé doit aussi prendre en compte la réalité des paramètres qu'il manipule (unité, ordre de grandeur). Finalement, notre travail s'est bâti autour de trois questions de recherche :

1. **Comment formaliser un modèle mathématique adapté pour la conception préliminaire d'un système complexe ?**
2. **Comment explorer efficacement l'espace réalisable d'un CSP numérique ou mixte ?**
3. **L'exploration partielle diversifiée est-elle un outil pertinent pour traiter des problèmes de conception architecturale ?**

En répondant à ces questions, nous cherchons à créer une rupture avec les approches courantes d'aide à la décision en conception architecturale. Ce document s'inscrit à la suite d'un ensemble de travaux de thèse notables traitants de programmation par contraintes en conception préliminaire : Fisher [49], Scaravetti [162], Vernat [183], Vareilles [181], Chenouard [23], Collignan [31], Matar [111], Picheral [142], Munzer [121], El Amine [47], Popoff [144], Hartmann [70], Mahiddini [107], Diampovesa [42] et Salehy [157]. Nos contributions se distinguent particulièrement sur deux points. D'une part, en traitant conjointement la modélisation et la résolution des CSP en conception. D'autre part, en proposant un cadre original pour l'exploration partielle de modèles de dimensionnement.

I.4 Structure du document

Le chapitre II fait état de la littérature existante dans les domaines évoqués par les questions de recherche. Il présente les grands principes autour de la conception préliminaire, les enjeux de la formalisation de modèles, le rôle de la programmation par contraintes et le cadre choisi pour la résolution de modèle CSP. L'objectif intrinsèque de ce chapitre est de poser clairement les enjeux de ce travail de recherche. Dans le chapitre III, nous présenterons une nouvelle méthodologie pour construire un modèle de dimensionnement dédié à l'étude de faisabilité en conception. La méthode sera éprouvée à partir du problème de conception d'une batterie pour véhicule électrique. La seconde question de recherche sera discutée dans le chapitre IV. Après avoir défini le cadre algorithmique pour l'exploration partielle, nous étudierons l'ensemble des opérateurs influant sur la recherche de diversité. Plusieurs indicateurs seront

introduits pour comparer les différentes stratégies de recherche mises en place. Dans le chapitre V, nous présenterons notre démarche d'exploration partielle sur trois cas d'étude. Chaque cas sera présenté sous un angle spécifique. Les résultats seront discutés par rapport à des méthodes de résolution plus classiques. Finalement, la conclusion VI permettra de mettre en évidence les apports scientifiques de cette thèse. Nous en profiterons pour discuter des limites rencontrées et des perspectives éventuelles. Les références utilisées sont listées à la fin du manuscrit suivie de quelques annexes pour compléter les propos de chaque chapitre.

ÉTAT DE L'ART

Ce chapitre introduit les éléments essentiels à la compréhension du manuscrit. Nous aborderons notamment pour chaque thématique les travaux de recherche existants et les termes techniques employés. Nous utiliserons le cas d'étude de la conception d'une batterie de véhicule électrique (ou BEV) pour illustrer certains points.

II.1 Le dimensionnement en conception préliminaire

II.1.1 L'objet de la conception : le système

Système complexe

Un système est un ensemble d'éléments en interaction. Ces interactions sont matérialisées par des flux (matière, énergie ou information). Un système est conçu pour réaliser des fonctions dans un environnement donné afin de satisfaire un besoin final. Il peut être qualifié de *complexe* lorsque son comportement global est difficilement prévisible, c'est-à-dire qu'il n'est pas égal à la simple somme des fonctions de ses composants [186, 48].

Cette définition d'un système complexe est large et commune à de nombreux domaines. Elle peut aussi bien désigner un produit industriel, un logiciel, une ville ou encore un organisme vivant. Ce travail est restreint aux *systèmes complexes industriels* ayant une ou plusieurs composantes mécaniques, électroniques ou thermodynamiques. Un système complexe répond à une hiérarchie structurelle interne (Figure II.1). Les éléments de cette structure peuvent former eux-mêmes des sous-systèmes, de la même manière que le système considéré peut être un sous-système dans un système plus important. Le terme *composant* désigne aussi bien un sous-système que les éléments les plus bas dans une décomposition structurelle. Par exemple, une batterie de véhicule électrique (BEV) est un système composé de cellules, d'un carter, d'un système de refroidissement et d'un système de management (BMS). La cathode et l'anode sont des composants unitaires de la cellule. La batterie est, elle-même, un composant du système véhicule électrique.

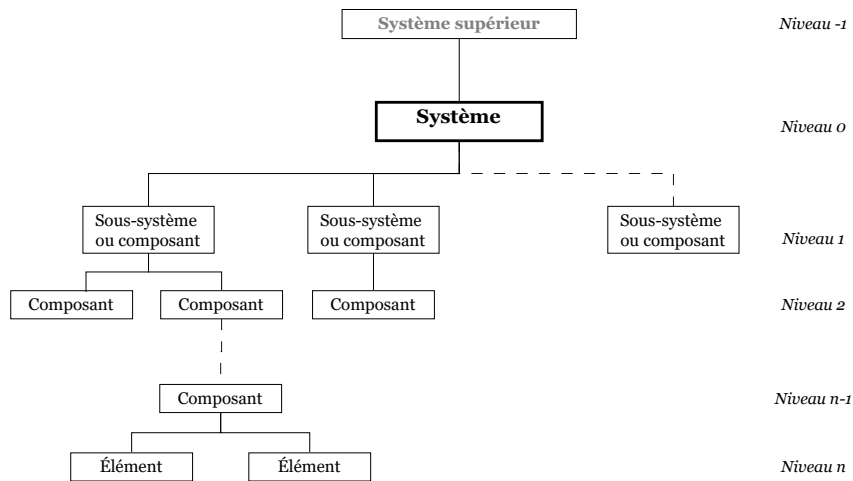


FIGURE II.1 – Décomposition structurelle d'un système complexe.

II.1.1.a L'ingénierie système

L'Association Française d'Ingénierie Système¹ donne la définition suivante (2023) :

Ingénierie Système (IS)

« L'ingénierie système est une démarche méthodologique pour maîtriser la conception des systèmes et produits complexes. Elle englobe l'ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes. »

L'IS est la discipline qui considère un système pendant sa conception, de l'expression du besoin dans un cahier des charges à l'opérabilité du produit fini [12]. Son rôle est de proposer des nouvelles solutions technologiques réalistes et de garantir leur adéquation avec le besoin au fur et à mesure du processus. Elle s'appuie sur différentes représentations du système pour favoriser le dialogue entre les experts et l'évaluation des solutions. L'IS intègre toutes les expertises nécessaires à la réalisation du système, c'est une approche interdisciplinaire qui fait intervenir de nombreuses parties prenantes [186]. Dans la suite de ce manuscrit, nous désignerons par le terme *concepteur* le ou les experts concernés sans distinctions.

II.1.1.b Cas d'une batterie de véhicule électrique (BEV)

Les éléments clés de cet état de l'art seront illustrés à partir d'un problème de conception de pack de batterie Lithium-ion (ou Li-ion) pour véhicule électrique. Cette batterie doit être capable de *stocker et délivrer de l'énergie électrique* pour une voiture de tourisme dont la consommation d'énergie massique k_{elec} est donnée. La masse du véhicule sans la batterie et avec la charge utile d'un passager est fixée à 1200 kg.

1. <https://www.afis.fr>

TABLE II.1 – Cahier des charges initial pour la conception d'un pack batterie Li-ion

Référence	Description
Besoin(s)	Stocker de l'énergie électrique E_{batt} Wh Délivrer de l'énergie électrique pour une consommation moyenne de $110 \text{ Wh.kg}^{-1}.\text{km}^{-1}$
Exigence 1	Limiter la masse de la batterie à 700 kg et son volume à 0.5 m^3
Exigence 2	Respecter un catalogue fournisseur pour les cellules
Exigence 3	Garantir une autonomie de 300 km
Exigence 4	Alimenter un véhicule de 1200 kg sans la batterie
Exigence 5	Assurer une durée de vie kilométrique de 300000 km
Exigence 6	Assurer une durée de vie calendaire de 10 ans
Exigence 7	Limiter le coût à 10000€
Exigence 8	Supporter une charge rapide (150 kW)
Exigences 9-11	Limiter les impacts environnemental à l'usage et à la fabrication (GWP, EU,AC,RD)

La batterie ne doit pas dépasser 700 kg en garantissant une autonomie minimale de 300 km. Les cellules seront sélectionnées dans un catalogue fournisseur (Tableau B.1 en annexe) afin de garantir des durées de vie kilométrique et calendaire minimales pour un scénario d'usage moyen. L'impact environnemental et le coût seront limités par des seuils maximum et la batterie doit supporter une charge rapide. Le cahier des charges est donné par le tableau II.1.

II.1.2 Le processus de conception

Le processus de conception est parfois représenté à l'aide du cycle en V (ou Vee model) [51]. Celui-ci est une compréhension large de la conception qui comprend l'ensemble des phases de la définition du besoin jusqu'à la validation du produit final. Nous considérons plutôt la conception selon le processus de Pahl et al. [132]. Ce processus séquence la démarche de conception de l'expression du besoin à la définition détaillée de l'architecture à concevoir, l'intégration du produit n'est pas considérée. La figure II.2 en donne une représentation simplifiée au format BPMN². Chacune des quatre étapes est caractérisée par des paramètres d'entrée, des données de sortie et des parties prenantes (concepteurs, experts, utilisateurs, etc.). Les paragraphes suivants décrivent leurs rôles.

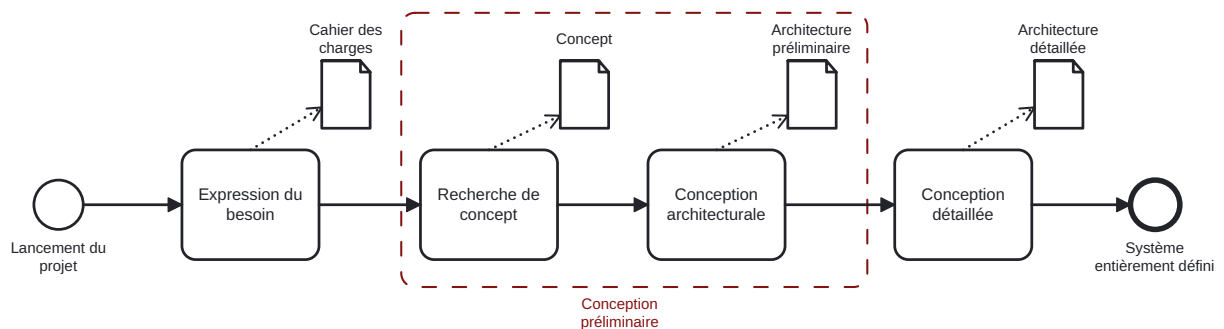


FIGURE II.2 – Représentation simplifiée du processus de conception.

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

Expression du besoin La première étape de la conception est de fixer le cadre du problème en identifiant le besoin et les exigences. Le besoin, la fonction principale du système et les exigences précisent le besoin sur l'ensemble des aspects du système à concevoir : fonctionnel, structurel, utilisation, performance, sécurité et impact environnemental. Le tout est représenté dans un *cahier des charges fonctionnel*.

Recherche de concept La recherche de concept donne les grands principes qui permettent de répondre au problème. C'est une phase de création pour obtenir, à partir de l'existant et d'idées innovantes, un principe de solution (ou concept), c'est-à-dire une abstraction de la solution. Le concept est décrit par différentes représentations (ou modèles) du système, chacune donnant un point de vue spécifique au concepteur (structure, usage, fonctionnement). En parallèle, l'analyse de l'existant permet d'évaluer le champ des alternatives possibles pour chaque composant. Le livrable est un concept ou un *principe de solution* [132] sous la forme d'une collection de modèles et de données préliminaires.

Conception architecturale La conception architecturale (ou pré-dimensionnement) correspond à une étude de faisabilité pour identifier les solutions les plus pertinentes parmi les alternatives réalisables. Les architectures sont générées et évaluées à partir de modèles de comportement simplifiés. Plusieurs itérations sont souvent nécessaires pour réduire l'espace de conception jusqu'à une architecture préliminaire. Les paramètres principaux sont dimensionnés grossièrement à partir de ces modèles. Les matériaux et les composants sont généralement sélectionnés à cette étape, selon les leviers d'actions du concepteur et les contraintes déduites des exigences. La conception architecturale aboutit à une architecture préliminaire ou *architecture produit* [162].

Conception détaillée La dernière étape du processus consiste à fixer précisément l'ensemble des paramètres restants du système. Les incertitudes apparues à l'étape précédente sont éliminées. Les méthodes de calcul et les données utilisées doivent être plus précises (simulation, éléments finis, CAO, etc.). C'est l'étape la plus coûteuse. Le résultat est un document qui détaille tous les aspects du système et de sa production [132].

II.1.2.a La conception préliminaire

La conception préliminaire regroupe la phase de recherche de concepts et la phase de conception architecturale. C'est une partie cruciale qui engage énormément de décisions sur le reste de la conception et sur la forme finale du produit. Il est admis que la conception préliminaire engage jusqu'à 80% des coûts de développement finaux pour un coût et un temps engagé inférieurs à 5% des coûts et temps totaux de conception [10]. De la même manière, les impacts environnementaux sur le cycle de vie du système y sont fortement déterminés [96]. L'objectif est de conceptualiser le système-solution du problème et d'évaluer toutes les alternatives réalisables qui correspondent à ce concept. C'est un travail pluridisciplinaire, soumis à de fortes incertitudes et à de nombreuses hypothèses qui rendent la prise de décision difficile. Pour supporter cette prise de décision, le concepteur utilise des représentations appelées modèles dans une démarche de conception intégrée [49]. L'objectif d'un modèle est de synthétiser la connaissance du système pour la hiérarchiser [162] et de simplifier le problème aux seules données utiles pour obtenir des architectures préliminaires. Ces représentations ne sont pas figées et liées à une étape du processus, elles

sont mises à jour au fur et à mesure de l'état d'avancement de la démarche. L'analyse ou la résolution des modèles doit permettre de répondre à deux questions : Quelles architectures sont envisageables ? Parmi ces architectures laquelle ou lesquelles sont les plus susceptibles de répondre au problème de conception ?

II.1.3 Un langage pour la conception : les modèles

II.1.3.a MBSE

L'*ingénierie basée sur les modèles* (Model-Based Engineering, MBSE) [186] propose de représenter les multiples points de vue du système par des *modèles*.

Modèle

Un modèle est l'abstraction d'un système (réel ou virtuel) à partir d'une symbolique commune [134]. Un modèle repose sur un ensemble fini de données et d'hypothèses pour servir un but [25]. C'est une approximation de la réalité.

Chaque modèle formalise une partie des connaissances sur le système à concevoir. Il participe ensuite à la prise de décision par une synthèse de ces connaissances (modélisation) ou par une évaluation des alternatives (résolution) ou encore par une analyse des architectures solutions (qualification). Les modèles sont répartis en trois catégories selon les caractéristiques du système qu'ils décrivent : fonction, comportement, structure. C'est le cadre Function-Behavior-Structure (FBS) établi par [61] :

- Les modèles fonctionnels explicitent ce que le système doit accomplir ;
- Les modèles comportementaux décrivent comment le système réalise ses fonctions et son état dans un environnement ;
- Les modèles structurels donnent la composition interne du système.

Un concept modélisé sous ces trois aspects permet aux parties prenantes d'avoir une vision globale du problème et à chaque instant de la démarche de conception. L'objectif intrinsèque est de minimiser le nombre d'*essais-erreurs* en réduisant efficacement l'espace réalisable à quelques architectures prometteuses [101]. Les modèles sont soumis à des règles communes définies dans un langage. L'approche MBSE est aujourd'hui bien répandue en ingénierie des systèmes bien qu'aucun langage ne fasse réellement l'unanimité. Chaque langage peut faire l'objet d'une appropriation différente selon l'utilisateur, mais respecte généralement un méta-modèle comme le standard MOF (MetaObject Facility) [186]. Le méta-modèle crée de la redondance d'information entre les modèles pour qu'ils communiquent entre eux et façonnent la vision globale attendue.

II.1.3.b Un exemple de langage de modélisation avec SysML

Le formalisme SysML (Systems Modeling Language) [168] propose neuf diagrammes pour représenter les différents aspects d'un système (figure II.3). C'est un langage suffisamment utilisé et générique pour être adapté à un large panel d'applications comme le montre la littérature [165, 194, 1]. Il est particulièrement adapté à la phase de recherche de concepts pour reformuler et synthétiser les connaissances. Nous

utiliserons quelques diagrammes de ce langage pour décrire le concept du cas d'étude du pack batterie. En particulier, pour la phase de conception architecturale on s'intéressera aux quatre diagrammes suivants : le diagramme de structure (Block Definition Diagram ou BDD), le diagramme paramétrique (Parametric Diagram ou ParD), le diagramme des exigences (Requirement Diagram ou ReqD) et le diagramme de comportement interne (Internal Block Definition ou IBD).

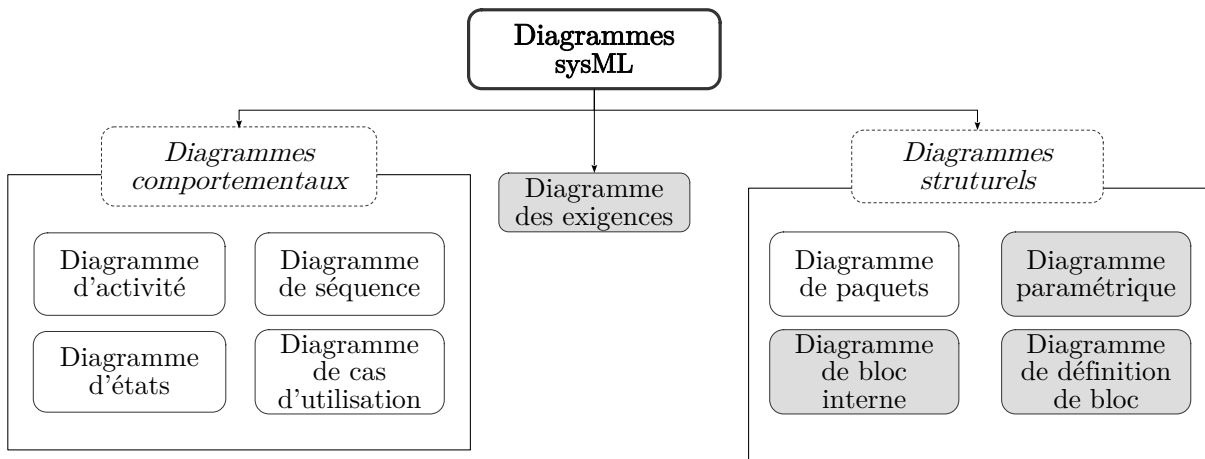


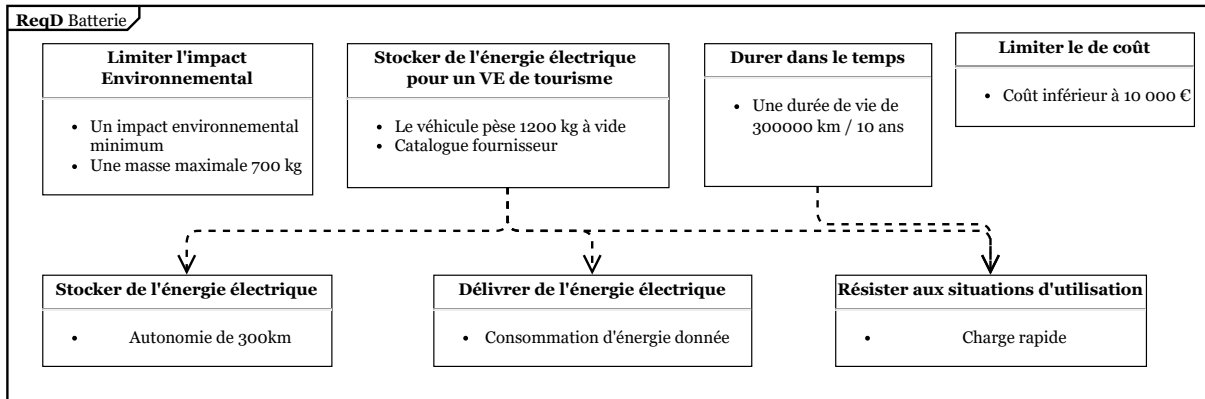
FIGURE II.3 – Les différents diagrammes SysML

Modélisation d'une batterie de véhicule électrique (BEV)

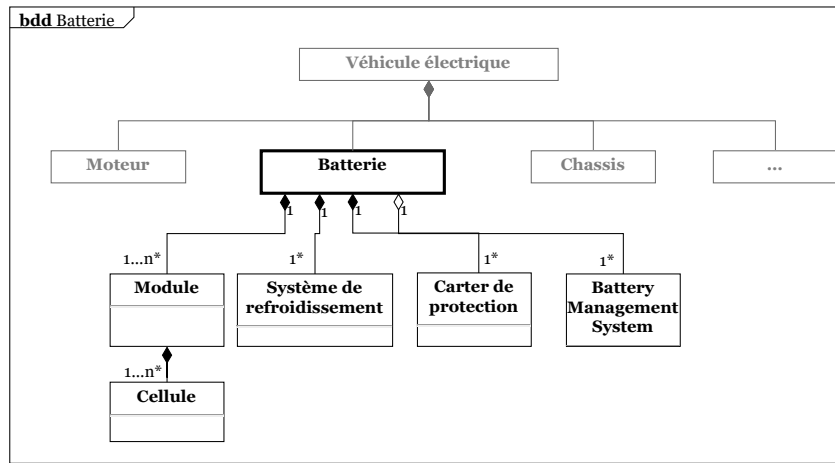
Les figures II.4 montrent quelques représentations du système batterie avec le langage SysML pendant la phase de recherche de concept. Le diagramme des exigences (ReqD) (Figure II.4a) est une représentation du cahier des charges pour supporter l'expression du besoin. Le langage SysML permet de compléter l'information initiale, par exemple en créant des liens entre les blocs d'exigences (hiérarchie, dépendances). Le diagramme de définition de bloc (Figure II.4b) est un modèle structurel. On retrouve les composants les plus importants de la batterie. Un composant peut lui-même être décomposé en sous-composants. À mesure que le processus de conception progresse, chaque bloc peut être enrichi par les caractéristiques du composant, par exemple : sa fonction, les matériaux possibles, la quantité, dimensions, etc. Le diagramme paramétrique II.4c regroupe et relie les grandeurs caractéristiques du système : nombre de cellules, masse, capacité, impact environnemental, etc. Chacun de ses diagrammes donne un point de vue sur le système batterie.

II.1.3.c Le modèle de dimensionnement

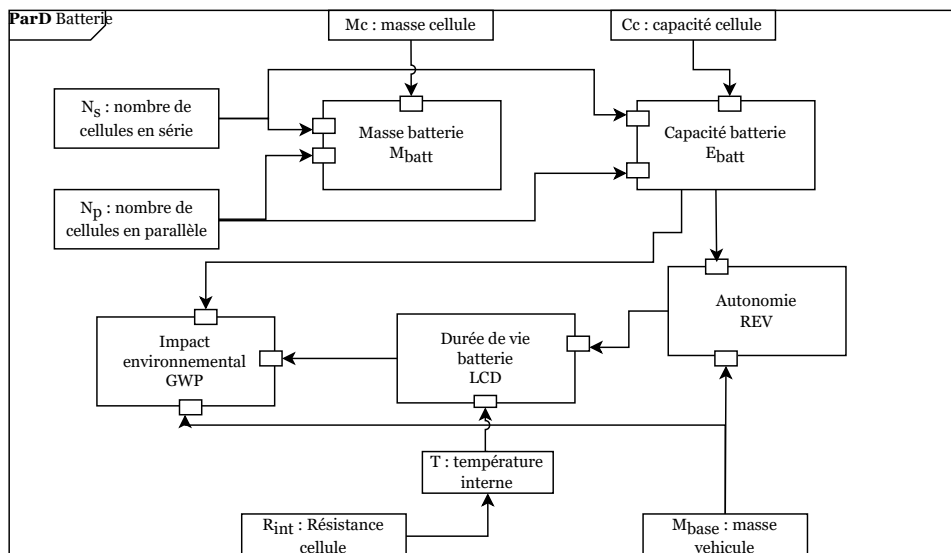
Le modèle de dimensionnement (figure II.5) est l'outil central pour l'aide à la décision pendant la phase de conception architecturale [25]. Son rôle est d'exprimer les degrés de liberté du système (paramètres d'entrée) en tenant compte des exigences fonctionnelles du cahier des charges, du comportement des composants, de l'usage et de l'environnement. La résolution du modèle permet d'obtenir des architectures réalisables et de les évaluer selon un ou plusieurs critères de performance (paramètres de sortie). Les



(a) Diagramme des Exigences (ReqD)



(b) Diagramme de Définition de Bloc (BDD)



(c) Diagramme Paramétrique (ParD)

FIGURE II.4 – Différentes modélisations possibles du concept *batterie* par des diagrammes SysML

valeurs de ces critères servent à discriminer les solutions entre elles. Un modèle de dimensionnement complet représente l'ensemble des aspects d'un système (structure, fonctions, comportement). Il existe plusieurs types de modèles de dimensionnement :

- le modèle de simulation numérique (méthodes par éléments finis 2D ou 3D) [45] ;
- le modèle boîte noire (expressions empiriques, tableau de données) [97] ;
- le modèle boîte blanche analytique [50, 188].

Les modèles de simulation sont très répandus dans l'industrie, ils permettent d'obtenir des résultats précis. En revanche, ils nécessitent un niveau de compréhension du système avancé. En particulier, les composants du système doivent être identifiés pour représenter leurs états transitoires. Ce n'est pas le cas en conception préliminaire où l'on cherche justement à identifier ces composants. Ce support est plutôt adapté à la phase de conception détaillée, notamment pour la représentation des comportements dynamiques des composants (effort, écoulement, etc.). Les modèles boîte noire sont des représentations opaques des comportements physiques. Ils sont souvent issus de relations paramétriques construites de manière empirique pour approcher un phénomène réel. Deux aspects limitent l'usage d'un modèle boîte noire : les incertitudes intrinsèques aux coefficients de ces expressions et la restriction de son domaine de validité au cadre expérimental utilisé. Le modèle boîte blanche analytique est une représentation mathématique explicite de comportements physiques. Les paramètres de ces modèles ont un sens physique bien établi.

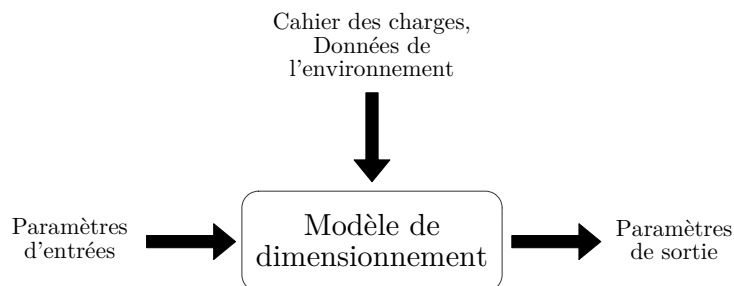


FIGURE II.5 – Principe général d'un modèle de dimensionnement.

Quel modèle pour la conception préliminaire ?

À ce stade, la définition multiphysique du système est soumise à des hypothèses de dimensionnement (régime permanent, usage moyen, comportement idéal, etc.). Celles-ci permettent de simuler grossièrement le comportement du système complexe sans éliminer de solutions potentielles. De plus, les données issues de la recherche de concepts sont souvent partielles, notamment les domaines des paramètres sont peu précis. La modélisation analytique boîte blanche est la représentation la plus pertinente dans ce contexte. En particulier, les modèles analytiques dits « gros grains » (ou *coarse grained model*). Un modèle gros grains est un modèle analytique limité à des comportements statiques pour simplifier la représentation du système [42]. Par exemple, en ingénierie électrique, les modèles de circuits équivalents de Thévenin ou

de Norton caractérisent une source de tension parfaite dans un composant électrique. Ces modèles gros grains permettent de représenter des cellules dans une batterie [77, 33]. Les comportements dynamiques seront pris en compte dans la phase de conception détaillée.

Le comportement multiphysique d'un système complet s'obtient par une composition de différents modèles. Certains phénomènes ne peuvent être exprimés de manière explicite, on peut alors recourir à des expressions empiriques. Par exemple, c'est le cas pour l'estimation de la durée de vie d'une cellule li-ion [35]. On trouve dans la littérature plusieurs exemples de modélisation analytique gros grains de systèmes physiques. Par exemple dans l'aviation [41, 21, 143] ou dans l'automobile [50, 94, 174].

II.1.3.d Positionnement des modèles

Nous pouvons positionner les différents types de modèles cités plus tôt tout au long du processus de conception pour bien identifier leur rôle (figure II.6). Pour ce travail de recherche, certaines modélisations nous intéressent particulièrement. Notamment, le modèle analytique gros grains sera le support privilégié pour l'étude de faisabilité. Dans le chapitre III, nous nous intéresserons à la construction de ce type de modèle sous la forme d'un problème de satisfaction de contraintes. Les autres modèles issus de l'expression du besoin ou de la recherche de concepts constitueront le point de départ de ce chapitre. En particulier les modèles fonctionnels et structurels qui décrivent les caractéristiques statiques du concept.

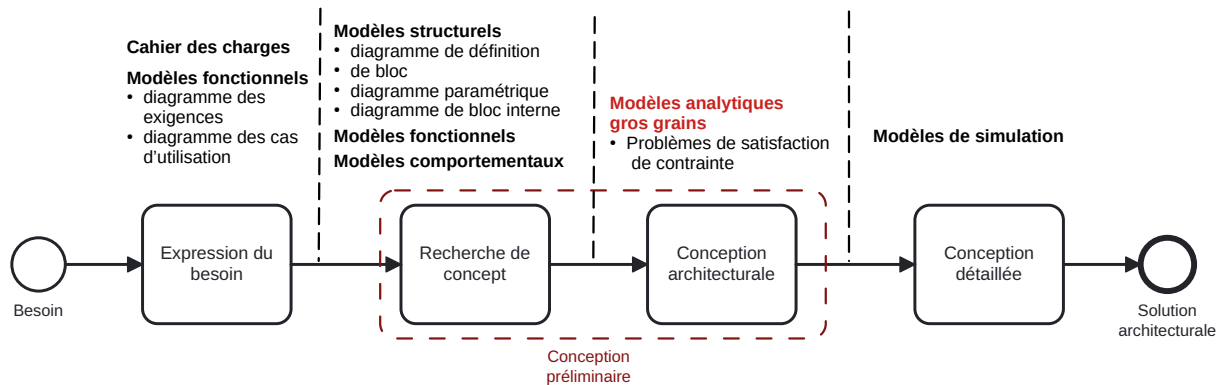


FIGURE II.6 – Positionnement de différents modèles sur le processus de Pahl et al. [132]

II.1.4 Le modèle analytique, support de la conception architecturale

Un modèle analytique est un ensemble de paramètres indéterminés qui décrivent l'état du système. On les appelle *variables de conception* (V_{co}) [162] ou encore degrés de liberté. Les paramètres fixés par le cahier des charges, les hypothèses de conception ou les choix du concepteur donnent le cadre pour le dimensionnement. Enfin, les paramètres de sortie ou *variables de performance* (V_p) permettent au concepteur d'évaluer les architectures générées par la résolution. Les V_{co} et les V_p sont reliées entre elles par l'expression mathématique des comportements du système et des exigences du cahier des charges. Le modèle analytique d'un moteur à combustion tiré de [134] est donné en annexe D.3, il sera étudié dans le chapitre V.

Plusieurs approches récentes dévoilent un nouvel intérêt pour les modèles analytiques gros-grains en conception architecturale. Par exemple, la *conception inversée intégrée* considère une démarche centrée autour des modèles gros grains de satisfaction de contraintes [49, 162]. Ces outils de calcul sont utilisés au plus tôt dans le processus de conception pour réduire rapidement le nombre d'alternatives potentielles. Plus récemment, [200, 42, 201] propose une approche orientée sur la synthèse des contraintes du système appelée Model Based System Synthesis (MBSS). L'approche MBSS développe la conception préliminaire autour du formalisme de la programmation par contraintes (PPC), notamment grâce au langage dédié DEPS [199]. Cela permet de passer plus rapidement à un modèle mathématique bien défini, mais nécessite une connaissance *a priori* des modèles de comportement. De plus, la vue analytique d'un système peut s'avérer moins compréhensible pour certaines parties prenantes au processus de conception.

II.1.5 Les variables en conception

Pour éviter toute ambiguïté entre les notions de variables, paramètres et constantes, nous proposons les définitions suivantes dans le cadre de la conception d'un système :

Une **variable** désigne toute grandeur liée à un état d'un système ou de son environnement et dont la valeur peut changer. Une **constante** est une grandeur du système ou de son environnement dont la valeur est fixée par le concepteur ou les conditions de l'environnement. Un **paramètre** désigne aussi bien une variable ou une constante obtenue à partir d'une variable dont la valeur a été fixée par l'utilisateur.

Dans notre problème de batterie, la masse de la batterie est une variable, la masse du véhicule à vide est un paramètre fixé par le concepteur et la température ambiante est une constante de l'environnement. Les variables du système sont réparties en trois catégories [183, 25] :

- **Les variables de conception** (V_{co}) correspondent aux leviers de conception du concepteur. C'est le nombre minimum de variables qui, lorsqu'elles sont toutes fixées, définissent entièrement une architecture. Deux architectures sont distinctes dès lors qu'elles ont une V_{co} dont la valeur est différente.
- **Les variables de performance** (V_p) parfois appelées variables de décision ou variables critères, ce sont les grandeurs qui permettent d'évaluer une architecture. Elles sont généralement liées aux exigences du cahier des charges.
- **Les variables auxiliaires** sont les variables propres à une modélisation pour faciliter sa compréhension, son implémentation ou sa résolution. Ces variables dépendent directement des V_{co} à travers une ou plusieurs expressions mathématiques.

Par exemple, pour la batterie, le nombre de cellules en série N_s est une variable de conception et l'énergie totale E_{batt} est une variable de performance. Certains modèles peuvent inclure une variable auxiliaire, comme la température interne de la batterie pour exprimer sa durée de vie. Sur la figure II.7, on enrichit le diagramme paramétrique en répartissant les variables entre les trois catégories. Les V_{co} et les V_p sont

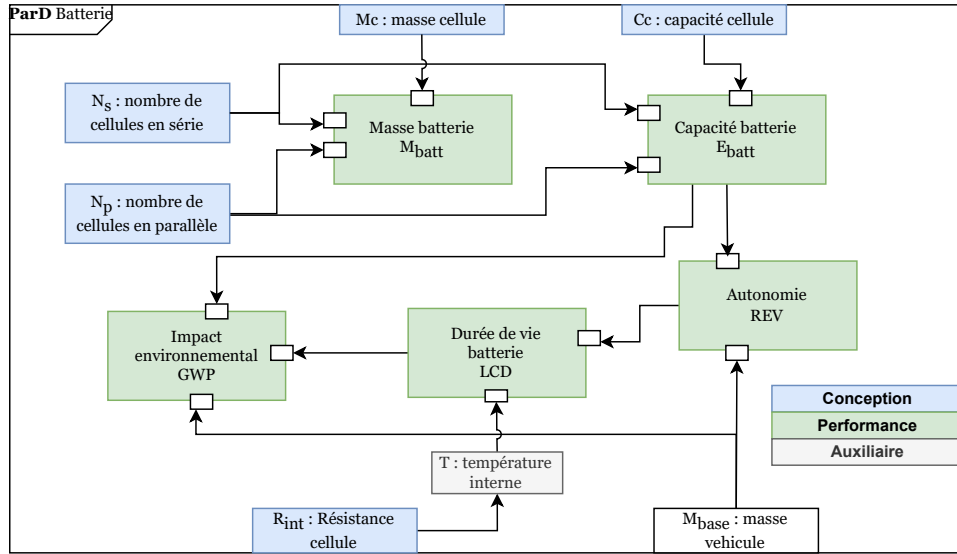


FIGURE II.7 – Exemple de classification des variables du concept de la batterie dans un ParD

des informations communes à l'ensemble des phases de conception. Un *objectif* de conception est une expression liée aux V_p dont on cherche à maximiser ou minimiser la valeur selon l'interprétation du cahier des charges. Pour comprendre entièrement une solution architecturale, le concepteur doit la représenter à la fois dans l'espace des V_{co} (ou *espace de conception*) et dans l'espace des V_p (ou *espace des objectifs*) (figure II.8).

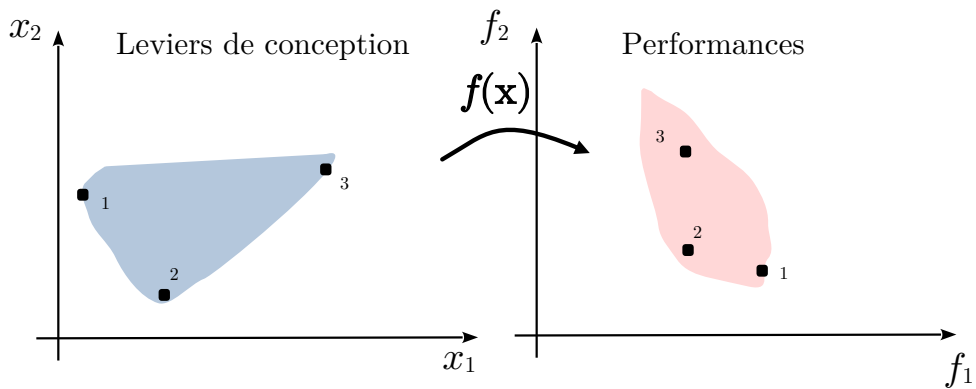


FIGURE II.8 – Les architectures réalisables doivent être représentées et analysées à la fois dans l'espace de conception (x_1, x_2) et dans l'espace des objectifs (f_1, f_2) .

II.2 Cadre formel pour le modèle de dimensionnement en conception architecturale

Le rôle central du modèle de dimensionnement analytique est l'étude de la faisabilité d'un concept pour un problème de conception. L'objectif est d'identifier les alternatives satisfaisantes de conception en amont de toute implémentation ou prototypage pour limiter le nombre d'essai-erreur. L'*Exploration d'Espace de Recherche* (ou DSE pour Design Space Exploration) [84, 92] est le cadre général qui considère les techniques de résolution pour évaluer efficacement l'espace réalisable d'un modèle de dimensionnement. Cela correspond bien aux objectifs de la conception architecturale. On associe généralement la DSE à la recherche d'optima, mais elle englobe tous les outils d'exploration qui rassemblent les trois qualités suivantes [92] :

1. Représenter le problème de manière formelle et suffisamment expressive pour traduire des contraintes complexes ;
2. Résoudre des problèmes de tailles diverses pour un coût raisonnable ;
3. Faciliter l'interaction avec l'utilisateur.

La *Programmation Par Contraintes* (PPC) [153] répond bien aux deux premières exigences. Son principe est de distinguer modélisation et résolution [53]. Généralement, le problème est modélisé sous la forme d'un problème de satisfaction de contraintes (CSP). Un solveur va ensuite résoudre le CSP pour identifier les instanciations possibles des variables [108]. Le dernier point est plus délicat à mettre en place dans un contexte précis, comme celui de la conception préliminaire. Le résultat de la résolution doit être une instanciation précise des variables pour décrire une architecture. La PPC est le cadre choisi pour traiter les trois questions de recherche décrites dans l'introduction. De nombreux travaux ont déjà montré son intérêt en conception architecturale [49, 60, 205, 173, 24, 198, 134]. Il existe plusieurs langages et bibliothèques qui reposent sur le paradigme PPC, par exemple Minibex³, AMPL [52], GAMS [55], MINIZINC [124], DEPS [199]. Nous utiliserons le langage Minibex dans ce document.

II.2.1 Problème de satisfaction de contraintes ou CSP

Problème de satisfaction de contraintes

Un problème de satisfaction de contraintes (CSP) est un triplet $\mathcal{P} = \langle X, D, C \rangle$ défini pour $n \in \mathbb{N}$:

- un ensemble de variables $X = \{x_1, x_2, \dots, x_n\}$;
- une collection de domaines $D = \{d_1, d_2, \dots, d_n\}$ tels que pour $\forall i, d_i$ représente l'ensemble des valeurs possibles pour la variable x_i ;
- un ensemble de contraintes $C = \{c_1, c_2, \dots, c_p\}, p \in \mathbb{N}$, où chaque c_i décrit une restriction sur les variables de X .

3. Minibex : <http://www.ibex-lib.org/doc/minibex.html>

Une solution d'un CSP $\mathcal{P} = \langle X, D, C \rangle$ est une affectation de chaque variable x_i par une valeur de d_i satisfaisant C , on parle d'une instantiation consistante de \mathcal{P} . L'ensemble \mathcal{S} des solutions de \mathcal{P} est défini par $\mathcal{S}_{\mathcal{P}} = \{x \in D \text{ tel que } \forall i, c_i(x)\}$. Un CSP est dit *discret* lorsque les d_i prennent la forme d'ensembles d'entiers, de booléens ou de n-uplets de réelles. En conception architecturale, un domaine discret permet par exemple de représenter un choix de composant ou la multiplicité d'un élément. Un CSP *numérique* admet des variables dont les domaines sont continus sur \mathbb{R} . Ce type de variable modélise notamment le dimensionnement d'une caractéristique physique d'un composant. Un CSP *mixte* présente à la fois des domaines continus et des domaines discrets.

Les *contraintes* sont des expressions qui restreignent les domaines de valeurs des variables et explicitent les combinaisons de valeurs réalisables. En conception, les contraintes permettent de formaliser les exigences du cahier des charges (ex : une limite de consommation), le comportement physique du système (ex : une température d'utilisation) ou des choix de composants, de matériaux (ex : le catalogue d'un fournisseur) [60]. Leurs expressions peuvent prendre différentes formes :

- Des expressions linéaires, par exemple une somme pondérée entre plusieurs variables de performances :

$$24.55x_2 + 26.75x_3 + 39x_4 + 40.5x_5 \leq 30$$

- Des expressions non-linéaires, expressions que l'on rencontre souvent pour exprimer des comportements complexes :

$$x_3 = x_4 \ln(x_1) - \frac{1}{x_2}$$

- Des expressions logiques [59], par exemple pour considérer différentes configurations structurelles :

$$((x_1 > 10) \text{ et } (x_2 < 0.5)) \text{ ou } (x_3 - 1 = 0)$$

- Des contraintes par morceaux [182], par exemple pour décrire le changement de phase d'un matériau :

$$c_1(x) = \begin{cases} c_1^1(x) & \text{si } x_1 \in [-273.15, 0] \\ c_1^2(x) & \text{si } x_1 \in [0, 100] \\ c_1^3(x) & \text{sinon} \end{cases}$$

- Des tables de valeurs, par exemple pour modéliser le choix d'un composant défini par plusieurs variables :

$$\begin{array}{|c|c|c|} \hline x_1 & x_2 & x_3 \\ \hline 10 & 0.1 & 1.237 \\ \hline 15 & 0.1 & 1.53 \\ \hline 20 & 0.2 & 1.79 \\ \hline 25 & 0.2 & 2.03 \\ \hline \end{array}$$

Le modèle analytique gros grains d'un problème de conception préliminaire peut être construit comme un CSP ayant une ou plusieurs de ces caractéristiques. Le type des domaines et la forme des contraintes d'un CSP impactent directement le choix du solveur pour sa résolution [59]. Un problème discret linéaire

n'est pas traité par les mêmes solveurs qu'un problème numérique non-linéaire. Les modèles rencontrés en conception peuvent prendre toutes les formes possibles, le cas le plus générique est celui des problèmes mixtes non-linéaires. Dans ce manuscrit, nous nous intéressons plus particulièrement aux techniques de résolution des CSP numériques non-linéaires [59]. Grâce à des opérateurs spécifiques présentés au chapitre IV, nous étendrons nos contributions aux CSP mixtes non-linéaires avec des contraintes de tables.

Exemple d'un modèle de batterie

Le problème II.1 est un CSP mixte non-linéaire issu d'un problème de conception d'une batterie pour moto électrique décrit dans [137]. Parmi les 8 variables de conception, les domaines des variables $(C_c, M_c, J_{cell}, R_{int}, I_{max})$ sont des n-uplets de réels, Les variables (N_s, N_p) sont des entiers et la variable τ_{mcp} évolue dans un domaine continu. Il y a des contraintes non-linéaires (comme c_4), linéaires (c_3) et une contrainte de table (c_8). Ce modèle gros grain est simple mais il permet au concepteur d'étudier rapidement la faisabilité d'architectures de batterie pour l'application prévue. Il sera adapté dans le chapitre III pour modéliser une partie du problème de conception du pack batterie pour voiture électrique.

$$X = (C_c, M_c, J_{cell}, R_{int}, I_{max}, N_s, N_p, \tau_{mcp}) \quad (\text{II.1})$$

$$D = ([2, 4], [0.01, 1], [1, 10], [10, 200], [10, 20], [1, 200], [1, 200], [0.05, 0.16])$$

sujet à.

$$c_1 : 3.66N_sN_pC_c \geq 9000$$

$$c_2 : 3.66I_{max}N_sN_p \geq 42000$$

$$c_3 : 3.66N_s \leq 110$$

$$c_4 : N_s(2.5 - R_{int}(42000/(3.66N_sN_p)))/1000/N_p \geq 34$$

$$c_5 : N_sN_pM_c(1 + \tau_{mcp}) \leq 60;$$

$$c_6 : J_{cell}N_sN_p \leq 5000$$

$$c_7 : (23000/(3.66N_s)) \leq 600$$

	C_c	M_c	J_{cell}	R_{int}	I_{max}
c_8	2.35	0.044	4.4	80	20
	2.95	0.047	5	80	20
	2.75	0.044	4.8	120	10
	3.15	0.044	5.2	120	10
	2.9	0.045	4.9	85	15

II.2.2 La modélisation des critères de performance

Considérons la modélisation d'un problème de conception sous la forme d'un CSP, $\mathcal{P} = \langle X, D, C \rangle$. Dans le cahier des charges, certaines exigences expriment des cibles ou des seuils sur les performances du système. Par exemple, l'exigence 7 du pack batterie (Tableau II.1) catégorise l'impact environnemental comme un critère de performance. De cette manière, le concepteur identifie un ensemble fini de variables de performances $V_p = \{y_1, \dots, y_k\}$, $k \in \mathbb{N}$. Chaque y_i peut s'exprimer en fonction des variables de conception V_{co} de X . Dans le cadre de la satisfaction de contraintes, les exigences sont représentées par des inégalités sur les V_p appelées ϵ -contraintes : $y(x) - y^{cible} \leq \epsilon$. L'objectif est de restreindre la recherche de solutions autour des valeurs désirées y^{cible} . Dans l'exemple du CSP II.1, l'expression c_6 est équivalente à une ϵ -contrainte sur le prix la batterie du véhicule.

Lorsque toutes les V_p sont considérées de cette manière, aucune préférence n'est exprimée entre les V_p . Le CSP résultant peut être résolu entièrement ou partiellement. On parle respectivement d'exploration complète ou d'exploration partielle de l'espace réalisable. L'exploration est peu utilisée dans un contexte

de conception préliminaire dans la littérature. Notamment lorsque le CSP traité est numérique ou mixte. L'objet des chapitres IV et V de ce manuscrit est de proposer un cadre pour l'exploration partielle dans ce domaine.

En général le concepteur préfère convertir le problème de satisfaction de contrainte en un problème de recherche d'optima. Les solutions sont alors discriminées pendant la résolution à partir des préférences du concepteur sur les valeurs des V_p . Le problème de satisfaction de contraintes devient un *problème d'optimisation* dont les V_p forment le ou les objectifs à minimiser ou maximiser. De la même manière que la PPC distingue modélisation et résolution, on différencie ici les notions de critère de conception et d'objectif de résolution.

Un **critère de conception** est une variable de performance pour laquelle il existe une valeur idéale aux yeux du concepteur. La valeur d'une V_p est idéale lorsqu'elle ne peut être améliorée sans sortir de l'espace réalisable. Cette notion est liée au modèle. Un **objectif** est une fonction f à minimiser construite à partir d'un ou plusieurs critères de conception. Cette notion est liée à la résolution du modèle.

Les critères de conception peuvent être considérés de différentes façons pour former les objectifs de la résolution du problème [30] :

- Méthode *a priori* : les critères sont ordonnés en amont de la résolution selon les préférences du concepteur ;
- Méthode *progressive* : les préférences du concepteur sont appliquées au fur et à mesure de la résolution (approches interactives) ;
- Méthode *a posteriori* : Les préférences sont formulées lors de l'analyse des solutions, une fois le problème résolu.

Du point de vue de la modélisation, on distingue les représentations « mono-objectif » et « multi-objectifs » [27, 167].

II.2.2.a L'approche mono-objectif

Les méthodes mono-objectifs traitent les critères V_p dans un objectif de résolution unique. Le problème résultant est composé d'un CSP et d'une fonction objectif, c'est un *Problème d'optimisation mono-objectif sous-contraintes*.

Problème d'optimisation mono-objectif

Soit un CSP $\mathcal{P} = \langle X, D, C \rangle$, \mathcal{S} l'ensemble des solutions de \mathcal{P} et $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction objectif. Le COP \mathcal{P}_f associé à \mathcal{P} et f est défini par :

$$\min_{x \in \mathcal{S}} f(x)$$

Un élément \mathbf{s}^* de \mathcal{S} est dit *optimal* si $\forall \mathbf{s} \in \mathcal{S}, f(\mathbf{s}^*) \leq f(\mathbf{s})$. $f(\mathbf{s}^*)$ est appelé l'*optimum global*. On note \mathcal{S}^* l'ensemble des solutions optimales globales de \mathcal{P}_f . La classe des COP ayant des variables mixtes et des contraintes non-linéaires est souvent désigné par *Mixed-Integer Nonlinear Program* (MINLP). La fonction f peut être construite de plusieurs manières en fonction des critères de conception y_i [27] :

- avec une fonction de distance globale aux valeurs de référence y_i^* des y_i :

$$f(x) = \sum_i \frac{y_i(x) - y_i^*}{y_i^*}$$

- avec une fonction d'agrégation pondérée linéaire sur les critères :

$$f(x) = \sum_i \alpha_i y_i(x), \forall i, \alpha_i \in \mathbb{R}$$

- avec un seul critère objectif f^* , les autres critères sont traités à l'aide d' ϵ -contraintes :

$$y_i(x) - y_i^{min} \leq \epsilon_i$$

avec y_i^{min} la valeur idéale de y_i et ϵ_i la tolérance sur les valeurs de y_i

- avec un ordre lexicographique où les critères sont minimisés séquentiellement dans un ordre défini par l'utilisateur.

II.2.2.b L'approche multi-objectifs

Les V_p peuvent être aussi considérées séparément dans un problème multi-objectif CMOP, chaque critère y_i devient un objectif f_i unique.

Problème multi-objectifs

Soit un CSP $\mathcal{P} = \langle X, D, C \rangle$, \mathcal{S} l'ensemble des solutions de \mathcal{P} et un ensemble de $k \geq 2$ fonctions objectif $f = (y_1, \dots, y_k) = (f_1, \dots, f_k) : \mathbb{R}^n \rightarrow \mathbb{R}^k$. Le CMOP $\mathcal{P}_{f_1, \dots, f_k}$ associé à \mathcal{P} est défini par :

$$\min_{x \in \mathcal{S}} (f_1(x), \dots, f_k(x))$$

Pour deux solutions \mathbf{s}_1 et \mathbf{s}_2 de \mathcal{S} , on dit que \mathbf{s}_1 *domine* \mathbf{s}_2 si $\forall i \in \{1, \dots, k\}$ nous avons $f_i(\mathbf{s}_1) \leq f_i(\mathbf{s}_2)$ et $\exists j \in \{1, \dots, k\}$ tel que $f_j(\mathbf{s}_1) < f_j(\mathbf{s}_2)$. Toute solution \mathbf{s}^* de $\mathcal{P}_{f_1, \dots, f_k}$ est dite *Pareto optimale* si elle n'est dominée par aucune autre solution de \mathcal{S} , elle est alors désignée comme *non-dominée*. Résoudre $\mathcal{P}_{f_1, \dots, f_k}$ revient à calculer l'ensemble des points non-dominés de \mathcal{S} , c'est l'*ensemble de Pareto*. Le *front de Pareto* désigne l'ensemble de ces points représentés dans l'espace des objectifs $f(D) \subset \mathbb{R}^k$. Les CMOP dont les variables évoluent dans des domaines mixtes avec des contraintes non-linéaires sont qualifiés de *Multi-Objective Mixed-Integer Nonlinear Program* (MOMINLP ou plus simplement MOP).

II.2.3 Formalisation d'un modèle analytique

La formalisation d'un modèle est le processus qui séquence sa construction en amont de son utilisation. Lorsqu'elle concerne un modèle de dimensionnement gros-grain, elle est le lien entre la recherche de concept et la conception architecturale (Figure II.9). Dans son appréhension la plus complète, la formalisation comprend le développement du modèle (ou modélisation), sa vérification mathématique et informatique et sa validation à partir de données de référence. Les deux derniers points forment le paradigme *Vérification et Validation* (V&V) très documenté dans la littérature [159, 128, 172, 131]. Le plus souvent la modélisation en conception préliminaire est réalisée à partir de modèles existants ayant déjà été soumis à ces étapes dans des contextes différents. La formalisation d'un modèle revient davantage à un travail d'adaptation pour assurer l'adéquation du modèle avec le concept, les exigences du cahier des charges, le support de représentation (langage) et l'outil de résolution.

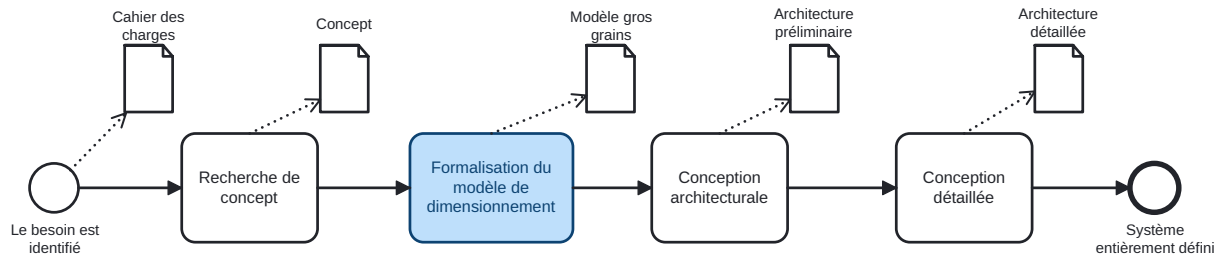


FIGURE II.9 – Positionnement de la formalisation du modèle de dimensionnement en conception préliminaire

II.2.3.a Confirmation, Vérification et Validation

Le processus de V&V est représenté dans la figure II.10. Cette approche s'appuie sur l'évaluation de la modélisation en trois points. La *confirmation* assure l'adéquation du concept et de ses représentations avec l'objet d'étude, on parle aussi de validation conceptuelle. Ensuite, la *vérification* permet de corriger les erreurs d'implémentation et de quantifier les erreurs de calcul issues de la résolution. Finalement, les résultats de la simulation du modèle sont comparés avec des données expérimentales lors de l'étape de *validation*. Cette approche est générique et s'applique aussi bien aux modèles de la recherche de concept qu'aux modèles analytiques de dimensionnement.

II.2.3.b Hypothèses, réalisation et qualification

Dans son travail de thèse, Vernat [183] propose une approche de modélisation et de validation pour des modèles analytiques en conception préliminaire. Le processus de formalisation est découpé en trois phases présentées sur la figure II.11 : *hypothèses, réalisation et qualification*. L'approche dans son ensemble est une méthode *bottom-up*, c'est-à-dire de la modélisation des éléments à la modélisation du système complet (de bas en haut sur une représentation structurelle). Dans un premier temps, la méthode de formalisation est appliquée aux composants et à leurs interactions physiques pour créer un ensemble de modèles locaux. Le modèle global est ensuite construit comme une agrégation de ces modèles locaux. Les trois étapes,

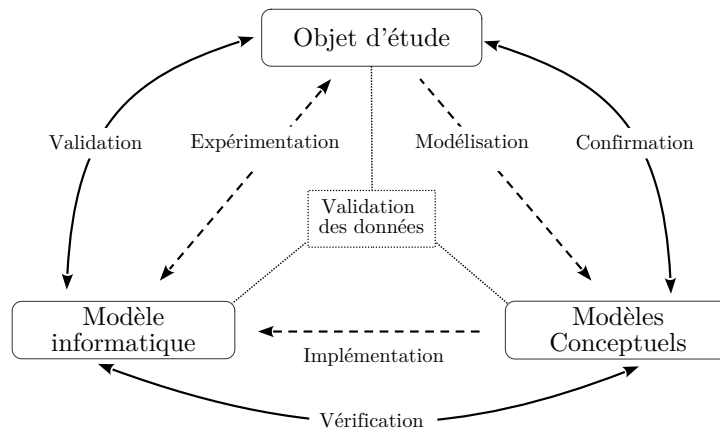


FIGURE II.10 – Représentation simplifiée du processus de Vérification et Validation [159, 172]

hypothèses, réalisation et qualification sont systématiquement appliquées. L'intérêt de cette approche est qu'elle peut s'appuyer sur le paradigme MBSE et qu'elle est associée à un ensemble de métriques définies par l'auteur. D'autres travaux ont repris ces principes en les adaptant [25, 107]. Les paragraphes suivants explicitent chaque étape.

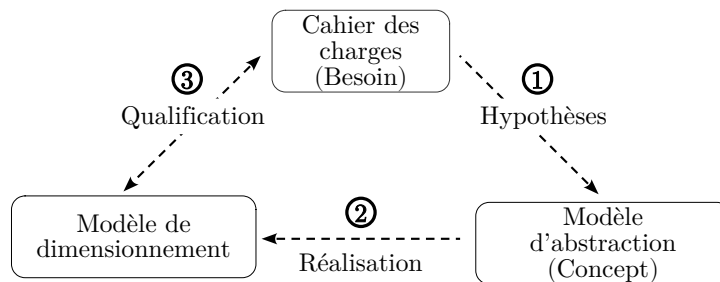


FIGURE II.11 – Formalisation d'un modèle de dimensionnement selon Vernat [183]

Hypothèses

La première étape est d'identifier les hypothèses qui fixent le cadre de l'abstraction des comportements du système. C'est un des rôles de la recherche de concept. Ces hypothèses limitent la capacité du modèle à reproduire fidèlement la réalité, mais elles facilitent l'étude de faisabilité de la conception architecturale. La formalisation du modèle de dimensionnement repose souvent sur l'adaptation et l'agrégation de modèles de comportement existants. Ces modèles sont construits à partir d'hypothèses que le concepteur doit aussi identifier. Dans l'ensemble, les hypothèses de conception peuvent porter sur l'environnement (ex : une température ambiante constante), l'utilisation (ex : un scénario d'usage moyen), le comportement physique (ex : cas du régime permanent), les expressions (ex : modèle analytique empirique) ou les données (ex : prédictions sur l'évolution du coût de matières premières). Connaître ces hypothèses est

indispensable pour maîtriser l'apport des modèles dans la prise de décision.

Réalisation

La phase de réalisation est l'écriture mathématique du modèle et son implémentation dans le langage adapté à sa résolution. Pendant cette étape, le concepteur doit faire attention à la cohérence conceptuelle et mathématique du modèle. La cohérence conceptuelle est l'adéquation du modèle vis-à-vis du contexte donné par le concept et des besoins du cahier des charges. Le modèle doit aussi être cohérent sur le plan mathématique, c'est-à-dire que les variables et expressions du modèle ne s'excluent pas entre eux (unité, domaines de validité, hypothèses, précision). L'approche choisie pour la résolution doit être adaptée aux constituants du modèle. Par exemple, le solveur CPLEX [175] n'est pas pertinent pour la résolution de modèles dont les expressions sont fortement non-linéaires. L'étape de réalisation nécessite parfois des itérations pour rendre le modèle suffisamment cohérent.

Qualification du modèle

Enfin, le concepteur doit s'assurer que le modèle soit suffisamment précis et exact sans être trop complexe ou trop spécialisé. C'est la qualification du modèle. La méthode de qualification de Vernat repose sur l'évaluation des critères PEPS [184] : Parcimonie, Exactitude, Précision et Spécialisation.

- **Parcimonie.** Un modèle est parcimonieux lorsqu'il est composée du plus faible nombre de variables et d'expressions possibles. La parcimonie est interprétée ici comme l'inverse de la complexité. Un modèle parcimonieux est, en théorie, plus simple à résoudre. Soit n le nombre de variables et p le nombre de contraintes, la parcimonie est initialement calculée par la relation : $\frac{1}{n+p}$.
- **Spécialisation.** Plus un modèle repose sur un grand nombre d'hypothèses de modélisation, plus il est spécialisé. L'auteur propose une estimation de la spécialisation à partir du modèle structurel du système. La mesure correspond au niveau le plus bas modélisé dans la décomposition hiérarchique des composants du système.
- **Exactitude.** L'exactitude évalue la capacité d'un modèle à reproduire le réel. Ce critère dépend donc de l'existence de valeurs de référence pour le système à concevoir. L'exactitude est évaluée par l'erreur entre les solutions issues de la résolution du modèle et des données de références. Cette mesure est réalisée dans l'espace des V_p et idéalement sur un grand nombre de références pour s'assurer d'une bonne exactitude dans tout l'espace des solutions.
- **Précision.** Selon Vernat, la précision d'un modèle comprend à la fois l'incertitude liée aux paramètres d'entrée et l'imprécision des relations qui composent le modèle. L'auteur propose d'évaluer la précision en faisant varier les paramètres incontrôlables sur leur intervalle de valeurs puis en évaluant les conséquences sur les valeurs des V_p .

Selon cette approche de multi-critères, un bon modèle maximise la parcimonie, maximise l'exactitude, maximise la précision et minimise la spécialisation. Elle peut s'appliquer à de nombreux modèles analytiques, aussi bien à l'échelle des modèles composants qu'à l'échelle du système global. L'objectif de l'auteur est surtout de pouvoir comparer les modèles à différents niveaux de modélisation.

II.3 Résolution de problème de satisfaction de contraintes

Selon la manière dont le concepteur souhaite considérer les V_p , un modèle de dimensionnement peut être construit comme un COP, comme un CMOP ou comme un CSP. L'optimisation est l'approche la plus présente dans la littérature pour résoudre des problèmes de conception, les formes COP et CMOP sont donc les plus répandues [203, 102, 16, 80]. Cependant, on constate quelques limites à l'utilisation de l'optimisation comme méthode de résolution :

- le nombre croissant de critères à optimiser dans les modèles qui complexifient la résolution et l'analyse ;
- les incertitudes intrinsèques aux modèles gros grains remettent en cause la notion d'optimalité des architectures solutions ;
- le fonctionnement en boîte noire des solveurs empêche la création d'outils interactifs pour l'utilisateur.

L'optimisation n'est donc pas toujours la meilleure approche pour résoudre un problème de dimensionnement. L'exploration de l'espace de recherche sans optimisation est une alternative possible, mais encore peu étudiée. Nous nous intéressons à cette approche comme support pour l'étude de faisabilité en conception architecturale. Explorer signifie générer des solutions de ce CSP. Sans optimisation, le but est alors de satisfaire les contraintes du CSP en considérant les V_p par des inégalités sur des valeurs seuils (ϵ -contraintes). Pour explorer des CSP mixtes non-linéaires, nous nous plaçons d'abord dans le cadre des CSP numériques qui sera élargi aux CSP mixtes au moyen d'opérateurs spécifiques. Le cadre reste limité aux CSP *statiques* (l'ensemble des variables est connu au départ) par opposition aux CSP dynamiques [116, 76]. Dans cette section, nous introduisons les méthodes existantes pour la résolution de CSP numériques non-linéaires.

II.3.1 Résolution d'un CSP Numérique

Un CSP numérique peut être résolu à partir de techniques basées sur des solveurs locaux couplés à une discrétisation des domaines continus ou basées sur l'arithmétique par intervalles [8, 118]. Nous utiliserons la seconde approche. L'arithmétique par intervalles permet de manipuler des ensembles continus de réels sous la forme d'intervalles représentés par leurs bornes. Nous considérons ici exclusivement les intervalles fermés :

Intervalle

Un intervalle fermé $[x]$ est un ensemble de réels défini par une borne inférieure \underline{x} et une borne supérieure \bar{x} telles que

$$[x] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\}.$$

L'ensemble des intervalles de \mathbb{R} est noté \mathbb{I} . Deux intervalles $[x]$ et $[y]$ sont dits égaux si $\underline{x} = \underline{y}$ et $\bar{x} = \bar{y}$. Un intervalle $[x]$ est dit dégénéré si $\underline{x} = \bar{x}$, il est alors identifié au réel x . On étend l'ensemble des opérations

binaires aux intervalles. Pour les opérations les plus courantes donnent :

$$\begin{aligned} [\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \\ [\underline{x}, \bar{x}] - [\underline{y}, \bar{y}] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \\ [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}] &= [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})] \\ 1/[\underline{y}, \bar{y}] &= [\min(1/\underline{y}, 1/\bar{y}), \max(1/\underline{y}, 1/\bar{y})] \text{ si } 0 \notin [\underline{y}, \bar{y}] \end{aligned}$$

Les fonctions réelles sont étendues à \mathbb{I} et pour une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, on appelle $[f] : \mathbb{I} \rightarrow \mathbb{I}$ une extension de f si $\forall x [x] \subset \mathbb{R}, f(x) \in [f]([x])$. En particulier, les fonctions monotones peuvent être exprimées sur les intervalles immédiatement, par exemple $exp([x]) = [exp(\underline{x}), exp(\bar{x})]$ [88]. Pour les fonctions non-monotones les expressions sont plus complexes, par exemple pour la valeur absolue :

$$|[x, \bar{x}]| = \begin{cases} [\underline{x}, \bar{x}] & \text{si } \underline{x} \geq 0 \\ [-\bar{x}, -\underline{x}] & \text{si } \bar{x} \leq 0 \\ [0, \max(\underline{x}, \bar{x})] & \text{si } 0 \in [\underline{x}, \bar{x}] \end{cases}$$

On définit la largeur d'un intervalle $[x]$ par $w([x]) = \bar{x} - \underline{x}$. $|[x]| = \max(|\underline{x}|, |\bar{x}|)$ est la valeur absolue et $m([x]) = (\bar{x} + \underline{x})/2$ le centre d'un intervalle. L'union et l'intersection de deux intervalles sont définies comme suit :

$$\begin{aligned} [x] \cup [y] &= \{z \in [x] \text{ ou } z \in [y]\} \\ [x] \cap [y] &= \{z \in [x] \text{ et } z \in [y]\} \end{aligned}$$

La plupart du temps l'union de deux intervalles n'est pas intervalle. On peut privilégier l'utilisation de l'enveloppe (aussi appelée *hull*). L'enveloppe d'un ensemble d'intervalles I de \mathbb{I} est le plus petit intervalle contenant I , on note cet intervalle $\square I$. Un intervalle $[x]$ permet de représenter un réel a par un encadrement $\underline{x} \leq a \leq \bar{x}$ à une précision ϵ près, soit $w([x]) \leq \epsilon$. En pratique, les réels formant les bornes des intervalles sont représentés par des nombres flottants selon la norme IEEE 754 [81] sous la forme :

$$f = (-1)^s 2^e m$$

où s est le bit de signe, e l'exposant et m la mantisse.

Boîtes

Une *boîte* $[\mathbf{x}]$ (ou vecteur d'intervalles) de dimension $n \in \mathbb{N}$ est un produit cartésien d'intervalles $[\mathbf{x}] = [x_1] \times [x_2] \times \dots \times [x_n] \subset \mathbb{I}^n$.

La figure II.12 montre une boîte en dimension 2. On peut définir la largeur d'une boîte $w([\mathbf{x}])$ comme le maximum des largeurs de ses composants et le volume $v([\mathbf{x}])$ comme le produit des largeurs de ses composants. L'enveloppe d'un ensemble de boîtes B est la boîte de plus faible volume contenant B et

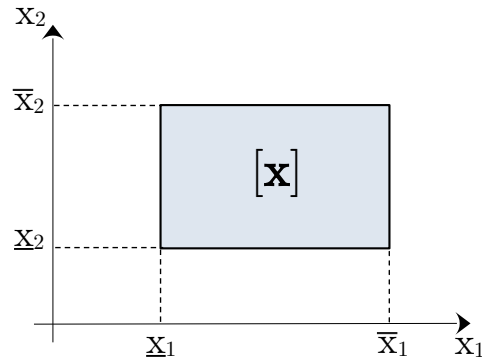


FIGURE II.12 – Une boîte $[x]$ de \mathbb{R}^2

noté $\square B$. Un *pavage* d'un ensemble $\Sigma \subset \mathbb{R}^n$ est un ensemble de boîtes de \mathbb{I}^n dont l'union contient Σ . Une fonction intervalle $[f] : \mathbb{I}^n \rightarrow \mathbb{I}^m$ est l'extension de $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ si $\forall x \in [x] \subset \mathbb{I}^n, f(x) \in [f]([x])$. Cette extension des fonctions réelles sur les boîtes permet d'exprimer les contraintes pour des variables continues [88]. Un CSP numérique peut alors s'écrire $\mathcal{P} = \langle X, [x]^0, C \rangle$, avec $[x]^0$ la boîte des domaines initiaux pour toutes les variables de X .

Soit $\epsilon = \epsilon_1, \dots, \epsilon_n$ un vecteur regroupant les précisions pour chaque variable de X . Une ϵ -boîte est une boîte $[x] \subset \mathbb{I}^n$ telle que $\forall i, w([x]) \leq \epsilon_i$. Résoudre un CSP numérique \mathcal{P} pour une précision donnée ϵ consiste à calculer un *pavage* de $\mathcal{S}_{\mathcal{P}}$ composé de ϵ -boîtes. Prenons l'exemple simple du CSP *cerceau* dont l'espace des solutions est compris entre deux cercles concentriques. La figure II.13 donne une ϵ -solution de ce problème pour $\epsilon = 0.2$ (à gauche) et le pavage complet de l'espace des solutions (à droite).

cerceau : $X = (x, y)$
 $x^2 + y^2 \leq 4$
 $x^2 + y^2 \geq 2$

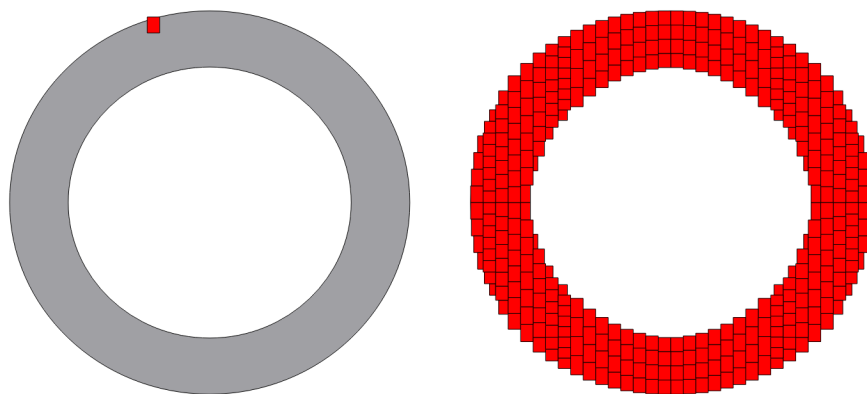


FIGURE II.13 – Une ϵ -boîtes (gauche) et le pavage (droite) du CSP numérique *cerceau*, pour la précision absolue $\epsilon = 0.2$

II.3.2 Exploration de l'espace de recherche

On oppose l'*exploration exhaustive ou complète* (en anglais : complete exploration) à l'*exploration partielle* (partial exploration). La première cherche à obtenir l'ensemble du réalisable (*i.e.* un pavage pour les domaines continus). La seconde calcule quelques solutions (ϵ -boîtes pour les domaines continus), en respectant un critère d'arrêt. L'exploration exhaustive est privilégiée pour les CSP discrets dont la taille de l'espace des solutions est raisonnable. Pour des CSP numériques ou mixtes, il est fréquent que le nombre de ϵ -solutions soit très grand, en particulier lorsque la précision diminue sur les domaines continus. Même face à des problèmes avec peu de variables : pour 3 V_{co} et une précision relative de 10^{-3} le nombre hypothétique de ϵ -solutions est déjà de 10^9 . Cela est particulièrement vrai pour les CSP en conception préliminaire qui sont souvent sous-contraints. L'exploration partielle *diversifiée* vise à calculer quelques solutions bien réparties dans l'espace afin que l'utilisateur ait une bonne représentation de l'ensemble du réalisable. Le résultat doit être un compromis entre le nombre de solutions et la représentativité. Un tel sous-ensemble de solutions est alors qualifié de *représentatif* [161]. L'intérêt en conception est de proposer au concepteur un nombre raisonnable de solutions à analyser.

II.3.2.a Exploration complète : Algorithme Branch and Prune

Les approches de type *generate-and-test* (évaluer chacune de sesinstanciations possibles) ne sont pas efficaces pour traiter des domaines continus [91]. Une bonne méthode pour explorer un CSP numérique est d'essayer d'éliminer rapidement les zones infaisables de l'espace de recherche. Ce principe peut être mis en œuvre à l'aide d'un algorithme Branch & Prune (B&P) [179]. Un B&P alterne des étapes de réduction (ou filtrage) et de division pour former un arbre de recherche. Les étapes de réduction permettent d'éliminer efficacement les parties non faisables de l'espace et la division sépare le problème initial en sous-problèmes plus petits. Lorsque la réduction élimine une boîte ou qu'une solution est obtenue, un retour en arrière (*oubacktrak*) est effectué pour sélectionner une nouvelle boîte à explorer. Une boîte est considérée comme solution lorsque qu'elle atteint la précision demandée et qu'elle n'a pas été éliminée par filtrage. L'algorithme 1 présente un B&P pour traiter un CSP numérique \mathcal{P} avec une précision ϵ donnée. Cet algorithme permet une exploration exhaustive de l'espace de recherche.

L'opération de division (Diviser dans l'algorithme 1) est généralement une *bissection*, *i.e.* une séparation en deux sous-problèmes distincts, pour former un arbre binaire de recherche. La profondeur de l'arbre de recherche est déterminée par le nombre de bisections nécessaires pour atteindre une ϵ -boîte. La méthode Contracter_C réduit l'espace de recherche sans perdre de solutions à partir de techniques de filtrage des domaines parfois appelés *contracteur* [106, 88]. À chaque *backtrack*, la structure de donnée L détermine la prochaine boîte à explorer, c'est la *sélection* de boîte (*node selection* en anglais). On utilise généralement une approche en profondeur (*depth-first search* (DFS) [170]). Le principe de DFS est de traiter la liste des boîtes non-explorées L à la manière d'une pile, cela permet de converger rapidement vers des solutions rapprochées. La manière globale dont l'arbre de recherche est construit pendant l'exploration est définie par la technique de bissection et la stratégie de sélection. Ces deux opérateurs algorithmiques forment la *stratégie de recherche* de l'algorithme B&P. La figure II.14 montre le rôle de chaque opérateur lors de la construction de l'arbre de recherche pour un CSP numérique.

Dans le cas d'une exploration complète, l'efficacité de l'algorithme B&P dépend davantage des opéra-

Algorithme 1 Algorithme Branch-and-Prune

Entrées : CSP $\mathcal{P} = \langle X, [\mathbf{x}]^0, C \rangle$, précision $\epsilon > 0$
Sortie : pavage \mathcal{S} de l'ensemble $\mathcal{S}_{\mathcal{P}}$

```

1:  $L \leftarrow \{[\mathbf{x}]^0\}$ 
2:  $\mathcal{S} \leftarrow \emptyset$ ;
3: tant que ( $L \neq \emptyset$ ) faire
4:    $[\mathbf{x}] \leftarrow \text{Extraire}(L)$ ;
5:    $\text{Contracter}_C([\mathbf{x}])$ ;
6:   si  $[\mathbf{x}]$  n'est pas vide alors
7:     si  $w([\mathbf{x}]) \leq \epsilon$  alors
8:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{[\mathbf{x}]\}$ ;
9:     sinon
10:       $\{[\mathbf{x}_1], [\mathbf{x}_2]\} \leftarrow \text{Diviser}([\mathbf{x}])$ ;
11:       $L \leftarrow \text{Insérer}([\mathbf{x}_1], [\mathbf{x}_2])$ ;
12:    fin si
13:  fin si
14: fin tant que
  
```

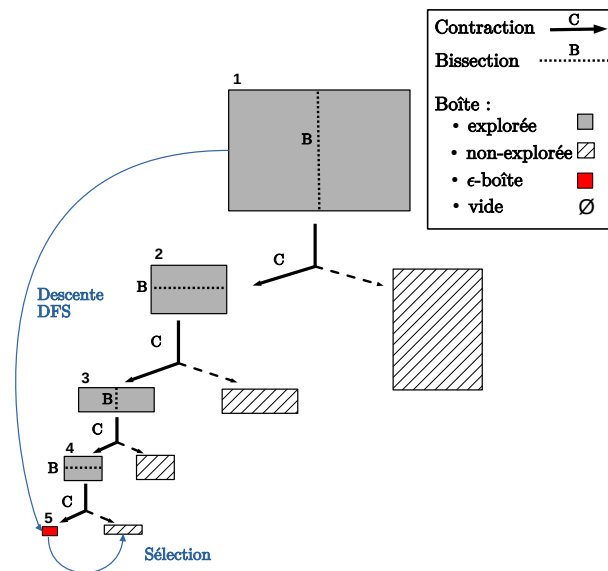


FIGURE II.14 – Illustration du rôle des opérateurs de l'algorithme Branch & Prune (B&P)

teurs de réduction et de division que de la stratégie de sélection et DFS est alors une méthode pertinente. Pour une exploration partielle, chaque opérateur affecte l'efficacité de la résolution. La complexité de l'algorithme augmente avec le nombre de variables et la précision ϵ . Pour réduire le nombre de division, on peut chercher à identifier des continuums de solutions en vérifiant la satisfaction globale des contraintes [185, 20]. Les identifier n'est pas toujours possible, cela dépend beaucoup des caractéristiques du modèle. De plus, les CSP des modèles de dimensionnement en conception peuvent être mixtes, réduisant l'occurrence de ces zones réalisables continues. Enfin, un concepteur privilégie souvent des solutions clairement instanciées plutôt qu'un ensemble de domaines grossiers au sein de continuums de solution. Dans ce manuscrit, l'exploration d'un CSP n'est considérée qu'à travers la génération de ϵ boîtes dans l'espace réalisable.

II.3.2.b Filtrage des domaines

Cette section présente les principes utilisés par l'opérateur de réduction Contracter_c dans l'algorithme 1.

Consistance locale

L'objectif des techniques de consistance est de réduire efficacement les domaines des variables à chaque nœud de l'arbre de recherche. Cette notion a d'abord été développée pour les CSP discrets [117, 106]. La technique la plus utilisée est celle de la consistance d'arc [106], pour une contrainte binaire cela donne la définition suivante :

Consistance d'arc

Une contrainte c sur les variables $X = \{x_1, x_2\}$ est *arc-consistante* sur les domaines discrets (d_1, d_2) si :

$$\forall a \in d_1, \exists b \in d_2 : c(a, b)$$

$$\forall b \in d_2, \exists a \in d_1 : c(a, b)$$

Cette définition de consistance locale peut être étendue pour les contraintes k-aires [54], c'est-à-dire les contraintes contenant k variables distinctes de X . Un CSP est dit arc-consistant si toutes ses contraintes sont arc-consistantes sur l'ensemble de ses domaines. Pour traiter la consistance des CSP numériques il faut adapter les techniques aux domaines continus [103, 9]. Par exemple, la *hull-consistance* permet de réduire chaque domaine des variables impliquées dans une contrainte au plus petit intervalle contenant la projection de la contrainte sur ce domaine.

Hull-consistance

Soit c une contrainte réelle sur $X = (x_1, \dots, x_n)$ et $D = ([x_1], \dots, [x_n])$ une boîte, c est dite hull-consistante par rapport à D si et seulement si $D = \square_{\rho_c} \cap D$.

La *3B-consistance* [103] vérifie la hull-consistance en considérant les intervalles dégénérés pour les bornes des domaines des variables. La *CID-consistance* introduite dans [177] étend la disjonction constructive aux intervalles en appliquant la hull-consistance à un découpage de la boîte initiale en sous-boîtes. Ces techniques sont détaillées dans [88].

Les contracteurs

Contracteur

Un contracteur désigne un algorithme de filtrage capable de maintenir la cohérence entre les domaines des variables et la satisfaction des contraintes du problème. Pour un ensemble de contraintes C sur un ensemble de variable X , un opérateur \mathcal{C} est un contracteur si pour une boîte $[\mathbf{x}]$:

$$\forall [\mathbf{x}] \in \mathbb{I}^n, \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] \text{ et } \forall x \in [\mathbf{x}], C(x) \Rightarrow x \in \mathcal{C}([\mathbf{x}])$$

La notion de contracteur regroupe les méthodes de réduction des domaines intervalles [88]. Il existe de nombreux opérateurs pour les CSP linéaires comme les contracteurs de Newton, Krawczyk ou Gauss-Seidel [125, 88]. Les contracteurs de Newton et Krawczyk peuvent être étendus aux CSP non-linéaires carrés et différentiables [5]. En particulier l’opérateur *XNewton* [3] implémenté dans la bibliothèque IBEX⁴. En plus de garantir que les zones éliminées sont infaisables, ces méthodes certifient l’existence de solutions. Prouver l’existence et l’unicité d’une boîte-solution permet notamment de réduire significativement le nombre de bisections et de converger plus rapidement. Ces opérateurs sont efficaces pour traiter des systèmes bien contraints (nombre de variables = nombre de contraintes), linéaires ou différentiables. Plusieurs travaux ont permis d’étendre ces propriétés aux systèmes sous-contraints au moyen d’opérateurs sur des parallélotopes [63, 110, 109]. Cependant, ces mécanismes montrent leurs limites lorsque le nombre de dimensions augmente et que les expressions du CSP sont complexes (non-linéaires entre autres). Aussi, ils n’empêchent pas l’accumulation de ϵ -boîtes sur les frontières de l’espace des solutions. Les CSP rencontrés en conception préliminaire sont souvent mixtes, non-linéaires et sous-contraints. Par soucis d’efficacité, dans nos algorithmes, on utilisera la propagation des contraintes sans certification. La propagation de contraintes sans preuve numérique donne uniquement la garantie de l’infaisabilité des zones éliminées à chaque réduction. La précision utilisée doit être suffisamment faible pour filtrer au mieux les domaines lors de l’exploration. De plus, du point de vue du concepteur, il est préférable de manipuler des architectures bien instanciées plutôt que des continuums de solutions.

Les contracteurs que nous utiliserons dans `ContracterC` (algorithme 1) reposent sur les mécanismes de consistance locale dédiés aux domaines continus. Le premier contracteur est *HC4* basé sur la *hull-consistance* et introduit dans [7]. Les figures II.15 illustrent le comportement de *HC4* à partir de la contrainte $3x^2 = z - xy$ avec les domaines $(d_x, d_y, d_z) = ([-10, 10], [1, 16], [0, 80])$. Le résultat de la

4. <http://www.ibex-lib.org/>

contraction est $([0, \sqrt{80}], [1, 16], [0, 80])$. Pour certains problèmes, nous utiliserons le contracteur *ACID* [126] qui repose sur la *3b-consistance* (contracteur 3BCID) et la *cid-consistance* [103, 177]. En pratique ces contracteurs seront utilisés dans une méthode de point fixe : la boîte est contractée successivement jusqu'à atteindre $\mathcal{C}([\mathbf{x}]) = [\mathbf{x}]$. Il est aussi possible de composer deux contracteurs \mathcal{C}_1 et \mathcal{C}_2 entre eux (équation II.2). Le résultat de la composition de *HC4* et *ACID* sera noté *HC4/ACID*.

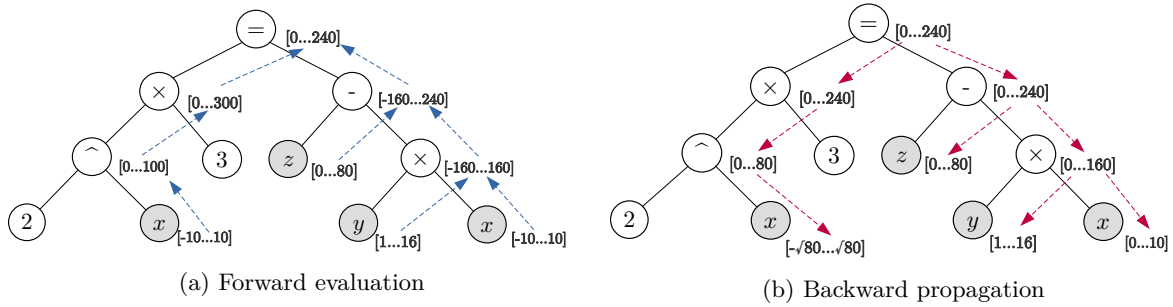


FIGURE II.15 – Propagation et évaluation des contraintes avec *HC4* sur l'égalité $3x^2 = z - xy$ et les domaines $(d_x, d_y, d_z) = ([-10, 10], [1, 16], [0, 80])$.

$$\mathcal{C}_1 \circ \mathcal{C}_2([\mathbf{x}]) = \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}])) \quad (\text{II.2})$$

II.3.2.c Stratégie de bisection

Un *bisecteur* divise un problème en le découpant en deux selon une de ses dimensions. Dans certains cas particuliers, on peut vouloir créer un déséquilibre entre les tailles des deux sous-problèmes obtenus, mais l'approche par défaut génère deux parties de taille équivalente.

Bisecteur

Soit $[\mathbf{x}] = [x_1] \times \dots \times [x_n]$ une boîte de \mathbb{I}^n . Pour une dimension donnée $1 \leq i \leq n$, un opérateur de bisection β divise $[\mathbf{x}]$ en deux boîtes, que l'on peut désigner par boîte gauche : $[\mathbf{x}]_g = [x_1] \times \dots \times [x_i, \frac{\bar{x}_i + x_i}{2}] \times \dots \times [x_n]$, et boîte droite $[\mathbf{x}]_d = [x_1] \times \dots \times [\frac{\bar{x}_i + x_i}{2}, \bar{x}_i] \times \dots \times [x_n]$.

Un bisecteur est caractérisé par le choix de la dimension i à découper dans la boîte $[\mathbf{x}]$, c'est la *stratégie de bisection*. La stratégie de bisection est une des deux composantes de la stratégie de recherche. Les techniques les plus courantes sont les suivantes :

- *Round-Robin* (RR) : sélectionne les dimensions les unes après les autres dans un ordre fixe (file circulaire) ;
- *Largest-First* (LF) : traite en priorité le domaine de plus grande largeur [25] ;
- *Smear Sum Relative* (SSR) : sélectionne la variable maximisant la fonction *smearRel* [95, 178].

RR et LF sont efficaces pour beaucoup de problèmes et leur implémentation est immédiate. La stratégie SSR revient à prioriser les variables les plus sensibles numériquement aux expressions composant le CSP. Elle est introduite dans [178] et s'appuie sur une variante de la fonction *smear* [95]. Pour une variable x_i d'un CSP numérique, $smear(x_i, c_j)$ est donné par :

$$smear(x_i, c_j) = \sum_{c_j} \left| \left[\frac{\delta c_j}{\delta x_i} \right] ([\mathbf{x}]) \right| w([x_i]) \quad (\text{II.3})$$

Où les c_i sont les expressions des contraintes du système d'équations (égalités et inégalités). *Smear Sum Relative* considère l'impact relatif de chaque variable $smearRel(x_i)$:

$$smearRel(x_i) = \sum_{c_j} \frac{smear(x_i, c_j)}{\sum_{x_k} smear(x_k, c_j)} \quad (\text{II.4})$$

Pour explorer des problèmes de conception, il est souvent intéressant de privilégier la bisection sur les variables de conception. Cette technique est évoquée dans [25]. Nous discuterons plus largement de ses implications dans le chapitre III. Nous verrons qu'explorer partiellement un espace de recherche demande d'adapter la stratégie de découpage des boîtes, notamment pour traiter des problèmes de dimensionnement.

II.3.2.d Précision

Lors de la résolution d'un CSP numérique, la précision sur chaque dimension influence à la fois la complexité de l'exploration et la forme de la solution finale. La précision peut être définie de manière absolue ou de manière relative.

Soit $\epsilon = (\epsilon_1, \dots, \epsilon_n)$ un vecteur de précisions de \mathbb{R}^n et $[\mathbf{x}]$ une ϵ -boîtes de \mathbb{I}^n on définit classiquement :

Précision absolue :

$$\forall i, w([x_i]) \leq \epsilon_i$$

Précision relative :

$$\forall i, w([x_i]) \leq \epsilon_i \left(\frac{|\bar{x}_i| + |x_i|}{2} \right)$$

Dans les deux cas, la taille de l'espace réalisable (*i.e* le nombre de ϵ -boîtes) est directement liée aux valeurs des précisions ϵ_i choisies pour chaque variable. Par exemple, dans le cas d'un CSP $\mathcal{P} = \langle X, [\mathbf{x}]^0, C \rangle$ numérique, on peut ainsi estimer le nombre maximum de solutions $T(\mathcal{P})$ contenues dans l'espace de recherche avec la formule II.5⁵, où $[x]_i^0$ est le i -ème membre de la boîte initiale $[\mathbf{x}]^0$. De la même façon, on pourrait estimer la profondeur de l'arbre de recherche ou le nombre de bisections maximum pour atteindre une ϵ -solution.

5. Uniquement si les bornes des domaines initiaux sont finies.

$$T(\mathcal{P}) = \prod_i^n \frac{\text{Largeur}(\text{Domaine Initial})}{\text{Précision}} = \prod_i^n \frac{w([x]_i^0)}{\epsilon_i} \quad (\text{II.5})$$

II.3.3 Exploration partielle de l'espace de recherche

Les algorithmes d'exploration partielle appartiennent à la famille des algorithmes de résolution partielle (ou *anytime algorithms*) [36]. Nous nous intéressons encore plus particulièrement aux *algorithmes d'exploration partielle diversifiée*, dont l'objectif est de calculer un échantillon de solutions représentatif de l'espace complet des solutions. Pour représenter au mieux l'espace réalisable, un sous-ensemble de solutions doit couvrir l'espace de manière uniforme. Cela revient à maximiser la *diversité* au sein de ses éléments. Cette notion a largement été étudiée en optimisation multi-objectifs [161, 73, 140, 104, 89, 82], la diversité y est surtout définie dans l'espace des objectifs sur les V_p . La diversité du résultat d'une exploration partielle peut concerner à la fois l'espace de conception et l'espace des objectifs. Ce point de vue est à notre connaissance peu traité dans la littérature. Les stratégies classiques que l'on retrouve dans les algorithmes d'exploration complète des CSP numériques, comme pour la sélection ou RR pour la bisection, ne sont plus efficaces pour une exploration partielle. Les approches de recherche *stochastiques* ne sont pas considérées ici. Une méthode déterministe est préférable pour la recherche d'architectures en conception, notamment pour la reproductibilité des résultats. Les approches de recherche locale n'ont pas été étudiés, mais font partie des perspectives notables pour améliorer les algorithmes qui seront proposés dans les chapitres IV et V.

II.3.3.a Algorithme Anytime Branch and Prune

Algorithme 2 Algorithme Anytime Branch-and-Prune proposé dans [26]

Entrées : CSP $\mathcal{P} = \langle X, [\mathbf{x}]^0, \mathcal{C} \rangle$, précision $\epsilon > 0$, fonction ρ , critère d'arrêt ϕ

Sortie : Sous-ensemble \mathcal{S} du pavage $\mathcal{S}_{\mathcal{P}}$

```

1:  $L \leftarrow \{[\mathbf{x}]^0\}$ ;
2:  $\mathcal{S} \leftarrow \emptyset$ ;
3: tant que ( $L \neq \emptyset$ ) et  $\phi$  est faux faire
4:    $[\mathbf{x}] \leftarrow \text{Extraire}(L)$ ;
5:    $\text{Contracter}_{\mathcal{C}}([\mathbf{x}])$ ;
6:   si  $[\mathbf{x}]$  n'est pas vide alors
7:     si  $w([\mathbf{x}]) \leq \epsilon$  alors
8:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{[\mathbf{x}]\}$ ;
9:        $L \leftarrow \text{Trier}(L, \mathcal{S}, \rho)$ ;
10:    sinon
11:       $\{[\mathbf{x}_1], [\mathbf{x}_2]\} \leftarrow \text{Diviser}([\mathbf{x}])$ ;
12:       $\text{Evaluer}_{\rho}([\mathbf{x}_1])$ ;
13:       $\text{Evaluer}_{\rho}([\mathbf{x}_2])$ ;
14:       $L \leftarrow \text{Insérer}([\mathbf{x}_1], [\mathbf{x}_2])$ ;
15:    fin si
16:  fin si
17: fin tant que
    
```

L'algorithme Anytime Branch and Prune (AB&P) pour l'exploration partielle diversifiée de CSP numérique a été introduit pour la première fois dans [23]. Les auteurs y décrivent la stratégie de sélection *Most-Distant First Search* (MDFS). L'algorithme 2 décrit le fonctionnement de MDFS. Cet algorithme ajoute deux composantes à l'algorithme B&P classique (algorithme 1) : un critère d'arrêt ϕ et une fonction de tri ρ . Le critère d'arrêt indique à quel moment l'exploration doit être prématurément arrêtée. Par exemple lorsqu'un nombre maximal de solutions est atteint. Après chaque *backtrack*, la fonction de tri permet d'évaluer les nœuds non explorés de l'arbre pour ordonner L . La fonction de tri proposée est construite pour maximiser la distance entre les solutions déjà calculées $\mathcal{S} = (s_1, \dots, s_m)$ et la prochaine boîte à explorer de L . Soit $[\mathbf{x}]$ une boîte de L :

$$\rho_{md}([\mathbf{x}] = \min(d_h([\mathbf{x}], s_1), \dots, d_h([\mathbf{x}], s_m))) \quad (\text{II.6})$$

Où d_h est la distance de Hausdorff [74]. Celle-ci est définie entre deux intervalles $[x]$ et $[y]$ par $d_h([x], [y]) = \max(|\underline{x} - \underline{y}|, |\bar{x} - \bar{y}|)$ et entre deux boîtes $[\mathbf{x}]$ et $[\mathbf{y}]$ de \mathbb{I}^n par $d_h([\mathbf{x}], [\mathbf{y}]) = \max\{d_h([x_i], [y_i]) : 1 \leq i \leq n\}$. MDFS a été construit pour maximiser la distance entre les solutions. Dans les faits, MDFS converge difficilement car trouver chaque ϵ -boîte solution revient à résoudre un problème d'optimisation globale [23]. Pour augmenter l'efficacité de cette exploration, les auteurs proposent une solution hybride entre MDFS et DFS [93, 119] appelée *Depth and Most-Distant Depth First Search* (DMDFS). La structure L n'est plus maintenue ordonnée après chaque division mais uniquement après chaque nouvelle solution. DMDFS donne une bonne répartition des solutions dans l'espace en temps beaucoup plus raisonnable que MDFS. Sur l'exemple du cercle (Figure II.13) MDFS génère 1000 ϵ -boîtes en 19,8 secondes ($\epsilon = 10^{-3}$) quand DMDFS met 1,9 secondes.

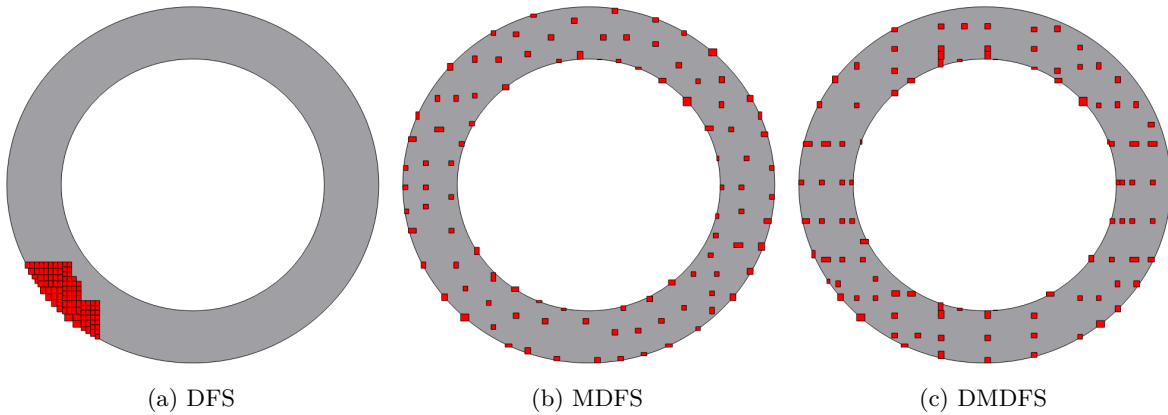


FIGURE II.16 – Exploration partielle du problème *cerceau* avec différentes stratégies de sélection présentées au chapitre IV (100 solutions).

II.3.3.b Stratégie de recherche Best-First Search

La stratégie de sélection MDFS peut-être aussi interprétée avec le paradigme *Best-First Search* (BestFS) [40]. La fonction de tri ρ est désignée par *measure-of-best function*. BestFS est initialement définie en optimisation globale [14, 119]. L'objectif d'une stratégie BestFS est converger efficacement vers

un optimum global en sélectionnant prioritairement les nœuds de l'arbre de recherche qui améliorent la borne supérieure ou inférieure de l'optimum recherché. La notion de stratégie BestFS est simplement étendue aux algorithmes d'exploration partielle AB&P, ρ désigne alors la fonction de tri. De la même façon, DMDFS correspond à une stratégie de sélection hybride BestFS/DFS. Ces méthodes hybrides sont aussi évoquées dans plusieurs travaux sur les algorithmes Branch & Bound pour accélérer la convergence [119] ou encore dans [127] avec la procédure *FeasibleDiving*. Nous reviendrons sur la description des approches hybrides pour l'exploration partielle dans le chapitre IV. La fonction de tri ρ utilisée pour MDFS et DMDFS sera désormais notée ρ_{md} .

II.3.4 Performance d'un algorithme d'exploration partielle en conception

Dans [204], l'auteur désigne un outil de résolution partielle comme un algorithme capable de donner une approximation correcte de la solution globale quels que soient le temps de calcul alloué. La notion de performance classique (résolue/non-résolue) n'est plus suffisante. Le résultat d'une résolution partielle est évalué par une mesure, souvent pluri-factorielle, de sa qualité. Certaines des qualités attendues pour un algorithme *anytime* sont présentées dans ce papier [204].

1. La qualité du résultat doit être mesurable ;
2. La mesure de la qualité peut être réalisée en temps constant ;
3. La qualité du résultat ne doit pas décroître au fur et à mesure de la résolution ;
4. La qualité du résultat doit être la plus déterministe possible, en corrélation avec le temps de calcul et les intrants ;
5. La performance doit être maximale dans les premiers temps de la résolution ;
6. La résolution doit pouvoir être interrompue à n'importe quel instant ;
7. La résolution doit pouvoir reprendre après avoir été arrêtée.

Ces qualités peuvent être transposées au cas particulier des algorithmes anytime pour l'exploration partielle diversifiée. Les points 3 et 4 de cette liste sont bien vérifiés par les algorithmes AB&P car ce sont des algorithmes déterministes. Des conditions de résolutions identiques entraînent un résultat identique et la qualité du résultat n'est pas dégradée si une nouvelle solution est trouvée. Le critère 5 est analysé à partir de l'étude menée au chapitre IV. Les points 6 et 7 représentent surtout un travail d'implémentation et une gestion de la conservation de l'état de l'exploration (notamment les nombre flottants sur les bornes des intervalles). Finalement, les deux premiers points seront abordés dans le chapitre IV en introduisant des indicateurs de qualités spécifiques.

II.3.4.a Recherche de la diversité

Un algorithme d'exploration partielle vise à maximiser la diversité d'un sous-ensemble de solutions de manière efficace. Estimer la performance de ces algorithmes avec un temps de calcul n'est pas suffisant. Il faut être capable de mesurer la diversité obtenue. L'évaluation de la répartition des solutions dans un espace de recherche est surtout étudiée dans le domaine de l'optimisation. Dans le cadre des problèmes mono-objectif, les travaux [73, 140, 82] intègrent les notions de diversité et de similarité sous la forme de

classes spécifiques de problèmes d'optimisation. La diversité pour les COP mixtes linéaires est étudié dans [62]. La répartition des solutions est aussi très largement traitée dans la littérature sur les performances des heuristiques pour l'optimisation multi-objectifs [161, 195, 206, 151, 4]. Il s'agit alors d'évaluer à la fois la répartition des solutions et la convergence vers le front de Pareto, donc seulement dans l'espace des objectifs. La qualité d'un sous-ensemble de solutions *représentatif* est évaluée selon quatre aspects : la *convergence*, l'*uniformité*, la *couverture* et la *cardinalité* [164, 104, 4]. La convergence exprime la proximité entre le résultat obtenu et la solution idéale. L'uniformité traduit une répartition égale des solutions dans l'espace. La capacité d'un échantillon de solutions à couvrir la frontière de l'espace des solutions est caractérisé par la couverture [161]. La cardinalité correspond simplement au nombre de solutions dans l'échantillon. La population de solutions idéale est alors celle qui maximise l'uniformité, la couverture et la convergence pour une cardinalité minimale.

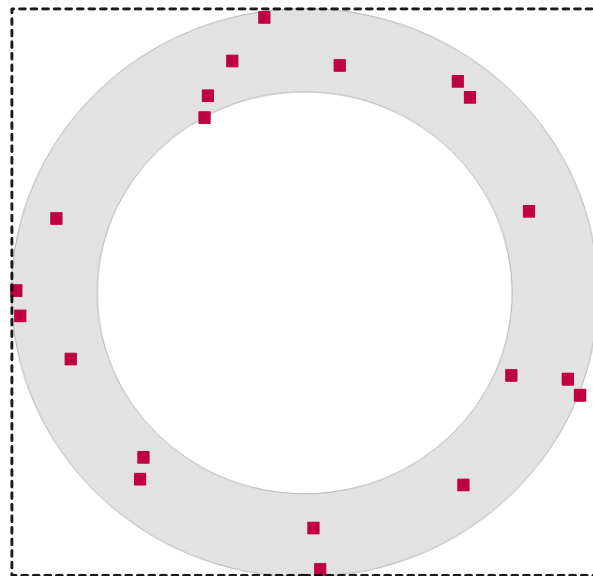


FIGURE II.17 – Exploration partielle du modèle *cerceau* (20 ϵ -boîtes). Comment mesurer la diversité du résultat d'une exploration partielle sans connaissances a priori du réalisable ?

La notion de convergence n'est pas pertinente pour le cadre de l'exploration. De plus, les nombreux indicateurs existants sur l'uniformité et la couverture dépendent souvent de l'existence de points de référence [207, 104, 89]. Le concepteur n'a pas nécessairement accès à des valeurs de référence autres que les limites du problème avant la première résolution. De plus, la diversité doit être évaluée à la fois dans l'espace de conception et dans l'espace des objectifs. La figure II.17 donne le résultat de l'exploration du modèle *cerceau* arrêtée prématurément après 20 ϵ -boîtes. Comment évaluer la diversité de ce résultat avec la seule connaissance du domaine initial (ligne pointillée) ? Dans le chapitre IV nous reprendrons ces définitions pour construire des indicateurs adaptés à l'exploration partielle diversifiée.

II.4 Bilan

Un existant limité pour la formalisation de modèles mathématiques ...

La formalisation de modèles analytiques gros grains pour la conception reste un sujet peu abordé dans la littérature. Les deux paradigmes évoqués présentent quelques limites dans leur utilisation. D'un côté, le cadre V&V est très générique et ne considère que le problème de l'évaluation de modèle. D'un autre côté, le modèle PEPS ne considère pas tous les aspects décisionnels d'un modèle, notamment l'adéquation au besoin et au contexte. Dans le chapitre III nous revenons sur ces limites plus en détail. À partir des contributions et des limites de l'existant, nous développerons une nouvelle approche globale de formalisation et les outils pour qualifier un modèle de dimensionnement.

Un paradigme de résolution prometteur ...

L'exploration partielle diversifiée est un paradigme intéressant pour traiter les problèmes de dimensionnement en conception préliminaire. La littérature est limitée sur ce sujet, mais les algorithmes existants comme DMDFS [26] posent les bases pour le développement d'un cadre approprié. Trois chantiers sont abordés dans ce manuscrit. Le premier est l'introduction de stratégies de sélection plus efficaces pour explorer partiellement des CSP issues de problèmes de conception. Le second est la construction et l'application d'indicateurs pour évaluer la diversité au sein d'un sous-ensemble de solutions. Le chapitre IV présente le travail réalisé sur ces deux points autour d'un benchmark de problèmes de conception. Enfin, le troisième chantier concerne l'utilisation et le positionnement de l'exploration partielle comme outil de résolution en conception architecturale. Le chapitre V aborde cette question en s'appuyant sur 3 problèmes de conception préliminaire.

FORMALISATION DE MODÈLES MATHÉMATIQUES EN CONCEPTION PRÉLIMINAIRE

III.1 Introduction

Le processus de conception préliminaire est divisé en deux phases : la recherche de concept et la conception architecturale. Nous nous intéressons ici à l'articulation entre ces deux étapes. D'un côté, le concepteur tire de la recherche de concepts un ensemble de modèles d'abstraction qui décrivent le système à concevoir sous plusieurs aspects : fonctionnel, comportemental et structurel. De l'autre, la conception architecturale s'appuie sur un modèle de dimensionnement analytique pour étudier la faisabilité et la pertinence de solutions préliminaires au problème de conception. Le lien entre les deux phases est la *formalisation* du modèle analytique gros grain (Figure III.1). C'est une étape primordiale qui détermine la pertinence des solutions issues de la résolution du modèle, et donc détermine aussi la qualité des décisions prises lors de la conception architecturale.

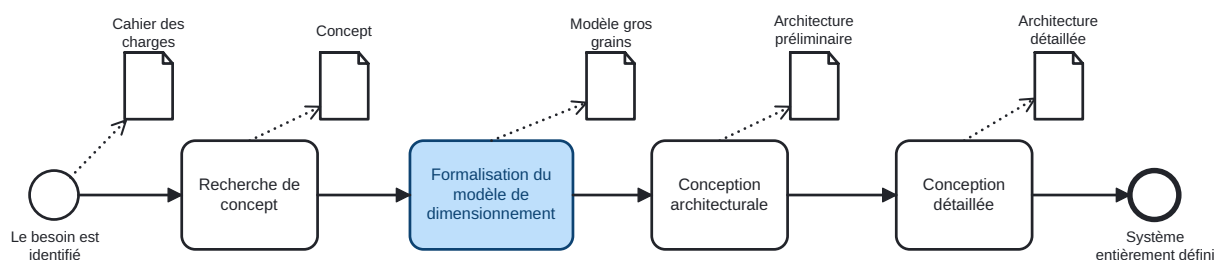


FIGURE III.1 – Positionnement de la formalisation du modèle de dimensionnement en conception préliminaire

De l'état de l'art vu au chapitre précédent, il ressort deux cadres existants pour la formalisation : le paradigme vérification et validation (V&V) [159, 172], et l'approche de Vernat [184]. Ces travaux mettent en avant les enjeux de la modélisation et introduisent des notions importantes pour sa compréhension. En revanche, ils ne répondent pas à l'ensemble des attentes pour la construction de modèles de dimensionnement en conception préliminaire. Dans les paragraphes suivants, nous détaillons les intérêts et les limites de chaque processus.

III.1.1 Apports et limites de l'existant

III.1.1.a Validation et vérification

Le paradigme V&V est restreint à l'évaluation du modèle, les autres aspects de la formalisation ne sont pas considérés. C'est une approche qui raisonne essentiellement sur des modèles déjà construits dans l'objectif de les accréditer pour un contexte donné. Aussi, les étapes de vérification et de validation dépendent d'un schéma expérimental réalisé en parallèle de la modélisation. Ces expérimentations servent à donner des valeurs de référence pour comparer les solutions du modèle au réel. Lorsque le modèle considéré doit être corrigé, les adaptations sont aussi faites à partir des données expérimentales [172]. Or, au stade de la conception préliminaire, il est difficile à mettre en place des expérimentations. D'une part car les connaissances sur les constituants ne sont pas assez avancées. D'autre part, car les solutions techniques envisagées n'ont pas toujours d'équivalents dans le réel. Enfin, le domaine d'application de la V&V n'est pas suffisamment précis pour le cas particulier des modèles de dimensionnement type CSP. Plusieurs aspects essentiels n'apparaissent pas, comme les notions de contraintes ou de complexité. Malgré ces trois points, le paradigme V&V permet de positionner nos travaux sur les différentes étapes pour l'évaluation d'un modèle. Un modèle doit être validé sur quatre niveaux [128, 160] :

1. Niveau conceptuel : déterminer si les modélisations du concept correspondent au cahier des charges sur les aspects structure, logique, mathématique, et fonctionnel. Estimer si les hypothèses sous-jacentes sont raisonnables ;
2. Niveau implémentation : vérifier la validité du code suite à l'implémentation du modèle et quantifier la précision numérique du résultat ;
3. Niveau opérationnel : déterminer avec quel degré le modèle retourne des résultats proche du réel ;
4. Niveau des données : déterminer la disponibilité des données, le niveau de confiance que l'on peut leur attribuer et leur précision.

III.1.1.b Formalisation de Vernat

La démarche proposée par Vernat repose sur trois propositions. La première est le processus : hypothèses, réalisation, qualification. La seconde est une méthode de qualification autour des critères PEPS. La troisième est le cheminement du local vers le global à partir de la décomposition structurelle du système établie pendant la recherche de concept. Les deux premières propositions ont été introduites dans le chapitre précédent II.2.3.b.

Approche bottom-up

Un concept répondant à une problématique de conception peut être découpé en trois types d'éléments : les fonctions pour satisfaire les exigences, les composants pour réaliser les fonctions et les flux (ou interactions) pour relier ces composants. L'échelle locale d'un système est celle des composants et des flux, l'échelle globale est celle des fonctions et du besoin. L'approche de Vernat s'appuie sur la modélisation a priori des composants et des interactions pour agréger ces modèles locaux dans un modèle global. Le

processus hypothèse, réalisation, qualification et la méthode PEPS sont appliqués pour les deux échelles. La figure III.2 donne une représentation simple de la démarche. Le PEPS permet à la fois d'identifier le niveau de modélisation le plus pertinent pour chaque flux/composant et de qualifier et valider le modèle global. La cohérence et la complétude du modèle sont vérifiées lors du regroupement des modèles locaux.

L'avantage de la proposition de Vernat est qu'elle autorise la formalisation de représentations complètes et adaptées à la décomposition structurelle prévue lors de la recherche des concepts. C'est une approche pertinente pour les experts qui ont une bonne connaissance de l'existant pour modéliser des comportements, à tous les niveaux du système. Plusieurs des notions introduites dans ces travaux sont essentielles au développement d'une approche de formalisation.

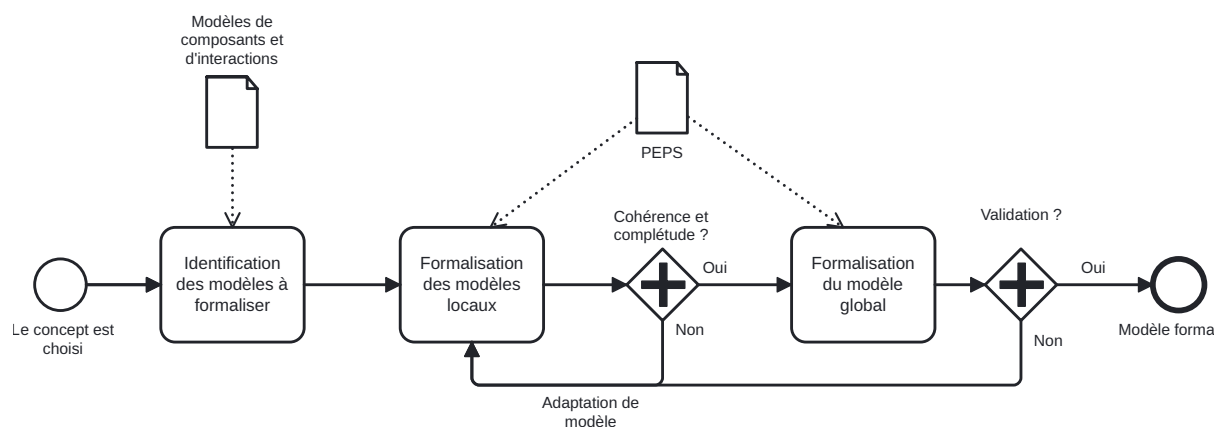


FIGURE III.2 – Démarche de formalisation de Vernat

Les limites rencontrées

Appliquer le processus complet peut être laborieux car chaque composant doit être modélisé, puis chaque modèle évalué. Sur l'exemple de la batterie, on trouve 8 modèles à considérer uniquement dans les premiers niveaux de la décomposition : 4 modèles de composants (cellules, refroidissement, carter et BMS) et 4 modèles de flux (électrique, thermique, mécanique, physique). Une fois que ces modèles sont agrégés et qu'ils représentent complètement le système batterie, il faut s'assurer de la cohérence de l'ensemble. Ce travail peut s'avérer délicat puisque les modèles locaux ne proviennent pas toujours des mêmes sources. Si une incohérence est détectée ou si le modèle est incomplet, la méthodologie prévoit de reprendre le travail à l'échelle locale.

Ensuite, cette approche du local vers le global crée le risque de perdre de vue les objectifs de conception au cours du processus. Au départ, les modèles de composants et d'interactions ne sont pas exactement adaptés au contexte de conception. Il faut souvent les reformuler avant de les agréger au modèle complet. Or, la méthode PEPS ne propose pas de repères pour s'assurer que l'agrégation d'un modèle local respecte le but initial du problème. On peut aussi questionner la capacité des modèles globaux issues de ce processus à être réutilisés dans un contexte différent. En effet, par construction, le résultat du regroupement de

modèles est très spécifique à la décomposition structurelle prévue dans le concept. Un concept proche, mais avec une structure différente, ne pourra que difficilement bénéficier du résultat d'une modélisation précédente.

Finalement, l'étape de qualification à partir de la méthode PEPS s'appuie sur une évaluation très subjective des caractéristiques du modèle. Notamment pour la notion de spécialisation. L'auteur relève d'ailleurs les limites de chaque critère dans [184]. Par exemple, le calcul de la parcimonie à partir du nombre de variables et d'équations ne donne qu'une vision étroite de la complexité. De plus, ces critères ne sont pas pertinents à évaluer à chaque étape. L'exactitude du modèle, qui correspond à la confrontation des résultats avec une ou plusieurs références, ne doit être évaluée qu'à la toute fin du processus.

III.1.2 Propositions du chapitre

III.1.2.a Une démarche de formalisation de modèles pour la conception architecturale

Le but de ce chapitre est de proposer une nouvelle méthode de formalisation pour un modèle de dimensionnement gros grains. Le point de départ de la méthode est un ensemble de connaissances sur le système issu de l'analyse du besoin et de la recherche de concept. Cet ensemble prend généralement la forme d'un cahier des charges, d'une collection de modèles conceptuels et d'une base de données sur l'existant. Pour ce travail, la modélisation est restreinte au paradigme des problèmes de satisfaction de contraintes (CSP) dont le formalisme permet de représenter la plupart des problèmes en conception. Ce cadre nous permettra d'introduire des méthodes d'évaluation plus précises que dans le cas général. La gestion des critères de performance est ainsi limitée à l'introduction de d'inégalités autour des valeurs attendues.

Le livrable attendu à l'issue de notre processus de formalisation est donc un modèle mathématique (ou un ensemble de modèles mathématique). Dans la mesure du possible, ce modèle doit être implémenté dans le langage de programmation approprié. Le langage doit permettre la modélisation de toutes les contraintes et variables, et la résolution avec la méthode choisie. Le concepteur doit aussi maîtriser la capacité du modèle à supporter la prise de décision.

III.1.2.b Le pouvoir décisionnel du modèle de dimensionnement

Idéalement, le modèle doit inclure toutes les alternatives prévues par le concept, exprimer toutes les exigences du cahier des charges et représenter chaque constituant du concept. Cependant, la modélisation est toujours limitée à plusieurs niveaux. Premièrement, après la recherche de concept, beaucoup d'incertitudes persistent sur le système et ses composants. Le concept lui-même repose sur un ensemble d'hypothèses. Deuxièmement, il n'est pas toujours possible d'exprimer un comportement, notamment les phénomènes dynamiques locaux qui nécessitent des expressions temporelles comme des équations aux dérivées partielles. Ces aspects plus complexes du système seront pris en compte uniquement lors de la phase de conception détaillée. Ensuite, certains composants, flux ou fonctions du concept ne sont pas pertinents à considérer à ce stade du processus de conception. Par exemple, dans le problème batterie le dimensionnement du BMS est rarement considéré avant la conception détaillée. Enfin, le concepteur doit aussi prendre en compte les limites des solveurs.

La construction d'un modèle idéal est donc trop complexe, souvent non souhaitable voir irréalisable [159]. En pratique, la modélisation s'appuie sur des hypothèses qui s'ajoutent à celles du concept ainsi que sur des techniques d'implémentation. Le modèle résultant est alors imparfait, le concepteur doit maîtriser son pouvoir décisionnel pour utiliser les architectures issues de sa résolution. Le pouvoir décisionnel d'un modèle est sa capacité à aider la prise de décision. L'objectif est de permettre au concepteur d'apprécier les avantages et les limites du modèles pour faire des choix en conséquence. Dans ce chapitre, nous proposons la sélection d'un ensemble de caractéristiques qui définissent ce pouvoir décisionnel. Pour fixer les idées, les méthodes d'évaluation de ces caractéristiques seront adaptées à la formalisation de modèles type CSP mixtes. Plusieurs objectifs sous-jacents se dégagent de notre analyse de l'existant :

1. Centrer le processus autour du besoin exprimé dans le cahier des charges ;
2. Maximiser les chances d'obtenir un modèle pertinent, même incomplet ;
3. Maîtriser le pouvoir décisionnel du modèle ;
4. Faciliter la réutilisation du modèle.

Notre démarche est construite autour de ces objectifs. Le problème de la batterie nous permettra d'illustrer notre approche et l'évaluation du pouvoir décisionnel.

III.2 Qualification d'un modèle mathématique en conception

III.2.1 Caractéristiques décisionnelles d'un modèle mathématique

Un modèle de dimensionnement ne peut pas être qualifié de manière binaire comme bon ou mauvais [159]. Un modèle bien construit est un modèle dans lequel le concepteur a suffisamment confiance pour utiliser ses solutions pour prendre des décisions. Évaluer un modèle signifie estimer cette confiance. Contrairement à la méthode de Vernat, notre objectif n'est pas de comparer des modèles mais davantage d'estimer de manière absolue un pouvoir décisionnel. Celui-ci est caractérisé par différentes grandeurs. Chacune de ces grandeurs répond à une question sur le modèle :

- L'étendue : quelle proportion de l'espace de recherche initial est représentée par le modèle ?
- La cohérence : les paramètres et expressions du modèle sont-ils cohérents (unités, ordre de grandeurs, hypothèses, etc.) ?
- La complexité : dans quelle mesure le modèle est difficile à résoudre ?
- La précision : quelle est la précision des paramètres d'entrée et la précision attendue des paramètres de sortie ?
- La complétude : combien d'éléments (composants, flux, fonctions) du système ont pu être modélisés ?
- L'exactitude : quelle est la capacité du modèle à reproduire des comportements de référence ?

III.2.1.a Étendue

Nous définissons la notion d'étendue d'un modèle comme la capacité du modèle à couvrir l'espace de recherche du concept. Les domaines des variables du modèle peuvent ne pas couvrir entièrement

l'espace initial de conception (D_I) exprimé pendant la recherche de concept. Par exemple, lorsqu'un type de composants prévu dans le concept est omis par le modèle ou lorsque le domaine de validité d'une expression est limité à certaines portions de l'espace de recherche. Deux scénarios peuvent être considérés pour prendre en compte cette zone manquante dans l'espace initial de conception du modèle D_M : réduire le concept initial ou formaliser un modèle spécifique pour cette zone. On propose d'exprimer mathématiquement l'étendue E_T comme le pourcentage de couverture de l'espace de conception initial couvert par le domaine de validité du modèle :

$$E_T = \frac{1}{n_{vco}} \sum_i \frac{w(d_i^M)}{w(d_i^C)} \quad (III.1)$$

Avec n_{vco} le nombre de variables de conception et pour chaque variable de conception x_i , d_i^C son domaine prévu par le concept et d_i^M son domaine de validité dans le modèle. Si x_i est une variable numérique, $w(d_i)$ exprime la largeur du domaine d_i . Si x_i est une variable discrète, $w(d_i)$ correspond au nombre de valeurs dans le domaine d_i . Cette formulation n'est valable que si $D_M = D_M \cap D_C$ (Figure III.3a). Dans le cas contraire, l'espace D_M peut être d'abord ramené à $D_M \cap D_C$ comme dans la figure III.3b. Si l'intersection entre les deux espaces est vide, le modèle n'est pas valable pour le problème initial. Si le modèle forme exactement l'espace de conception du concept on obtient $\sum_i \frac{w(d_i^M)}{w(d_i^C)} = n_{vco}$ puis $E_T = 1$.

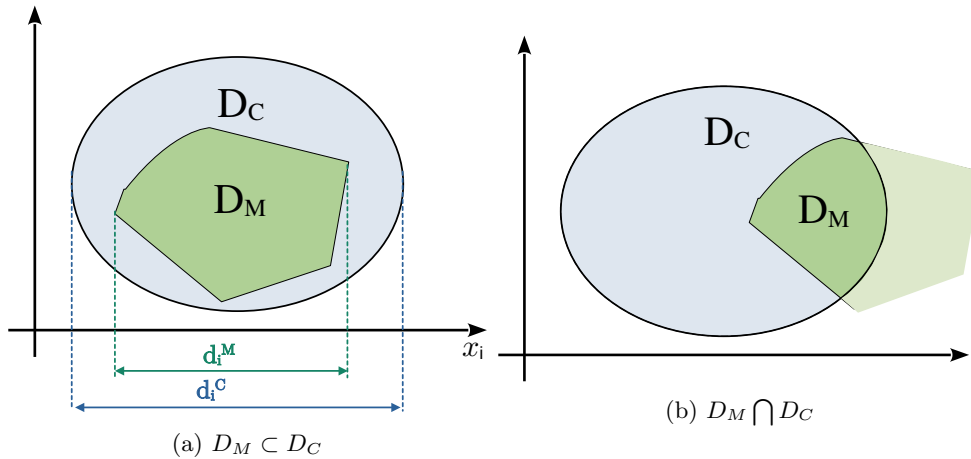


FIGURE III.3 – Représentations de l'étendue du modèle D_M et du concept D_C dans l'espace de conception

III.2.1.b Cohérence

Un modèle est cohérent lorsque les unités et les ordres de grandeur sont respectés pour l'ensemble de ses paramètres. La cohérence est une propriété binaire du modèle, soit le modèle est cohérent, soit il ne l'est pas. Le concepteur doit s'assurer de la cohérence du modèle au fur et à mesure du processus de formalisation. La principale source d'incohérence est l'agrégation de modèles de sources différentes. Il faut s'assurer de la cohérence du résultat de l'agrégation de deux modèles avant d'en considérer un troisième à agréger. Dans notre processus, deux pratiques permettent d'assurer la cohérence globale. La première est la construction d'un tableau des spécifications du concept dont l'objectif est de prévoir la composition

du modèle au niveau de ses variables, des unités, des ordres de grandeur et des leviers de conception. La seconde est l'agrégation progressive des modèles entre eux qui permet le suivi de la cohérence globale du modèle.

III.2.1.c Complexité

La complexité d'un modèle exprime la difficulté apparente pour sa résolution, indépendamment des performances du solveur. On distingue deux cas de figures. Soit le concepteur est limité à un seul solveur disponible. Dans ce cas, la complexité permet de s'assurer que le modèle peut être traité par ce solveur. Soit le concepteur évalue la complexité pour identifier la méthode de résolution et le langage d'implémentation les plus adéquats.

Plusieurs attributs peuvent être considérés pour apprécier la complexité avant toute résolution : le nombre de degrés de liberté réel (V_{co}) n_{vco} , le nombre de contraintes (n_{ineq}), le nombre d'égalités (n_{eq}), le type des domaines des variables ($[d]$), le type des expressions (c_{tr} pour contraintes), le nombre de critères de performance (n_{vp}) et la taille du problème ($T(\mathcal{P})$). La taille du problème est ici une estimation du nombre de solutions potentielles au problème ; grâce à la formule II.5 pour les domaines continus et par énumération pour les domaines discrets. Au fur et à mesure de la modélisation, les données relatives à la complexité sont mises à jour (par exemple : augmentation du nombre de variables, changement du type de problème). La complexité peut être représentée par le tableau III.1a.

Propriété	Valeurs possibles	Caractéristique	Symbole
n_{vco}	Entier positif	Linéaire	L
n_{eq}	Entier positif	Quadratique	Q
n_{ineq}	Entier positif	Non-linéaire	NL
n_{vp}	Entier positif	Table	T
$[d]$	{Continu, Discret, Mixte}	Par Morceaux	P
c_{tr}	{L,Q,NL,T,P,G}	Logique	G
$T(\mathcal{P})$	$\approx 10^x$ pour une précision donnée		

(a) Tableau de la complexité

(b) Classification des expressions

TABLE III.1 – Données attendues pour évaluer la complexité d'un modèle et classification possible des expressions des contraintes

Le *principe de parcimonie* est une règle pour limiter le nombre de variables et d'expressions dans une modélisation. Ce principe est utilisé dans [184] pour comparer la complexité de deux modèles. En pratique, il n'est pas suffisant pour traduire la complexité réelle du modèle. D'une part, le type des contraintes a souvent davantage d'importance que leur nombre, d'autre part il arrive que l'introduction de variables auxiliaires facilite la résolution du modèle. C'est notamment le cas lorsqu'on linéarise un système d'équations non-linéaires.

La connaissance de l'ensemble des propriétés du tableau III.1a donne une idée plus précise de la complexité du modèle. Le type des variables et des contraintes donne la classe de problème correspondante. Le tableau III.1b donne la classification (non exhaustive) utilisée pour ce travail. On peut aussi se référer aux documentations plus précises des benchmarks de problèmes d'optimisation comme GAMS¹. Classer

1. https://www.gams.com/latest/gamslib_ml/libhtml/index.html#gamslib

le modèle permet d'orienter fortement le choix du solveur et du langage. Ce choix est aussi déterminé par la gestion des critères de performance décidés par le concepteur : solveur pour l'optimiser ou solveur pour explorer. Par exemple, si le modèle prend la forme d'un problème d'optimisation mono-objectif non-linéaire mixte, on pourra utiliser le solveur BARON [155]. Finalement, le choix final du solveur dépend de la complexité globale du modèle, de l'expertise des utilisateurs et de la disponibilité du solveur.

III.2.1.d Précision

La précision d'un modèle est à distinguer de la précision des résultats. Cette dernière correspond à l'intervalle de confiance que l'on peut accorder aux valeurs de ses variables. La précision d'un modèle représente l'imprécision sur les paramètres d'entrée et les relations qui le composent. En effet, les constantes et les bornes des domaines des variables d'un modèle peuvent être soumis à une tolérance, par exemples si leurs valeurs sont obtenues empiriquement. De la même manière, si le modèle est construit avec des expressions semi-empiriques, le concepteur peut prendre en compte l'imprécision due à ces expressions. La précision P_R d'un modèle s'évalue à partir de l'expression ou du paramètre ayant la moins bonne précision relative. Il est possible qu'un modèle ait une précision considérée comme infinie s'il n'y a aucune incertitude sur les valeurs des paramètres d'entrée. L'objectif est d'indiquer au concepteur la présence de sources d'imprécision au sein du modèle et de donner une borne inférieure pour la précision ϵ au moment de la résolution. Fixer la précision du solveur avec une valeur inférieure à cette borne est contre productif.

III.2.1.e Exactitude

L'exactitude d'un modèle est associée à la notion de validation dans le cadre V&V, c'est-à-dire évaluer la capacité d'un modèle à reproduire des résultats de référence. Le calcul de l'exactitude donne l'erreur (ou la distance) entre les solutions et un ou plusieurs comportements de référence. Cette erreur est généralement évaluée dans l'espace des objectifs sur les critères de performance du modèle [183, 138, 15]. L'enjeu est d'identifier les bonnes valeurs de références pour réaliser cette mesure. Le paradigme V&V se base sur un plan d'expérience pour établir les valeurs de référence en parallèle du processus de modélisation. Cependant, le concept n'est pas assez précis à ce stade du processus de concept pour envisager un prototype. Une autre solution est d'identifier des modèles existants reconnus comme valides dans la littérature ou dans des abaques. Plus le nombre de références est important, plus la mesure sera précise. Plusieurs méthodes existent pour mesurer la distance avec les points de référence, [113, 184, 47]. Les méthodes de mesure locales renseignent sur les valeurs extrêmes de l'erreur, par exemple avec l'erreur maximale E_M III.2 :

$$E_M = \max_i (|\mathbf{s}_i - \tilde{\mathbf{s}}_i|) \quad (\text{III.2})$$

Avec \mathbf{s}_i les solutions issues du modèle et $\tilde{\mathbf{s}}_i$ les solutions de référence. Cette expression est privilégiée lorsque que peu de valeurs de référence sont disponibles. Il existe aussi des méthodes globales comme la mesure de l'erreur moyenne \bar{E} . Les mesures globales sont pertinentes lorsqu'il y a un grand nombre de références à conditions qu'elles soient réparties de manière homogène dans l'espace réalisable.

Récemment, de nouvelles méthodes d'évaluation globale de l'exactitude ont été introduites. L'approche proposée par [138] cherche à intégrer les préférences du concepteur dans la mesure. Celle-ci est alors divisée

en deux indicateurs pour maîtriser à la fois le risque lié à l'exactitude du modèle et la satisfaction des critères de performance. Bouyarmane [15] introduit, en plus de la mesure globale de l'erreur, un indicateur subjectif sur le degré des hypothèses considérées pour la modélisation.

III.2.1.f Complétude

La mesure de l'étendue seule n'est pas suffisante pour s'assurer de l'adéquation du modèle avec le concept. La complétude désigne la capacité d'un modèle à représenter le fonctionnement de chaque élément du concept (composants, flux, fonctions). Cette capacité se manifeste en explicitant le rôle de chaque expression du modèle dans les décompositions structurelle et fonctionnelle du concept. Ainsi, l'étendue est une analyse sur les variables et leurs domaines quand la complétude est une analyse sur les expressions (contraintes, égalités). À ce stade du processus, il n'est pas toujours possible ou même nécessaire de construire une représentation exhaustive. Certains composants ou certaines fonctionnalités ne sont pas pertinents à considérer lors de la phase de conception architecturale. Par exemple, la modélisation du carter de la batterie n'est pas nécessaire au stade de la conception préliminaire. Le dimensionnement du carter interviendra dans un second temps, lorsque les paramètres principaux du système seront largement fixés. Nous proposons de définir la complétude comme la différence de composition entre le modèle réel et le modèle prévu par le concept. Un élément désigne ici un composant (moteur, batterie,...), un flux (électrique, thermique,...) ou une exigence (capacité, masse,...). On note k_{max} le nombre d'éléments du concept et k_{model} le nombre d'éléments réellement représentés par le modèle. La complétude C_P est évaluée par la formule suivante :

$$C_P = \frac{k_{model}}{k_{max}} \quad (III.3)$$

III.2.2 Pouvoir décisionnel d'un modèle de dimensionnement

Parmi les six propriétés introduites, trois renseignent sur la capacité du modèle à répondre dans l'espace de recherche prévu par le concept : la complétude, la cohérence et l'étendue. Les trois autres représentent la pertinence analytique des constituants du modèle et des résultats. En dehors de l'exactitude, les caractéristiques peuvent être évaluées en amont de la phase de résolution, au fur et à mesure de la modélisation. La définition et le rôle de chaque grandeur est résumé dans le tableau récapitulatif III.2. On considère que le pouvoir décisionnel d'un modèle est maîtrisé par l'utilisateur lorsque les 6 indicateurs sont évalués. Sur le plan décisionnel, elles se répartissent en trois types selon l'aspect du modèle concerné : l'adéquation au concept, la résolution et la validité.

La complétude et l'étendue mesurent l'adéquation du modèle au concept. La précision et la complexité renseignent sur les caractéristiques du modèles qui vont déterminer la stratégie de résolution et les paramètres éventuels du solveur. L'exactitude et la cohérence indiquent si le modèle est valide et quel est le degré de validité vis-à-vis du réel.

III.2.3 Vérification

Dans [172], l'auteur distingue deux formes de vérification : la vérification du code et la vérification du calcul. La vérification du code implique de s'assurer de la fiabilité et de la robustesse du solveur pour

TABLE III.2 – Caractéristiques du pouvoir décisionnel d’un modèle de dimensionnement CSP

Caractéristiques	Label	Définition	Rôle
<i>Adéquation au concept</i>			
Étendue	E_T	Capacité du modèle à répondre au besoin	Estimer l’adéquation avec les variables du concept
Complétude	C_P	Capacité du modèle à représenter la structure	Estimer l’adéquation avec les éléments du concept
<i>Résolution</i>			
Complexité	C_X	Difficulté apparente d’un modèle à être résolu	Déterminer le langage et le solveur
Précision	P_R	Précision intrinsèque du modèle	Évaluer la confiance dans le modèle
<i>Validité</i>			
Cohérence	C_H	Respect des ordres de grandeur et des unités	Assurer la justesse du modèle
Exactitude	E_X	Distance entre le modèle et une référence	Valider les résultats du modèle

la résolution, et de l’absence d’erreurs lors de l’implémentation dans le langage de programmation. La vérification du calcul analyse la convergence lors de la résolution et estime l’erreur numérique induite par la résolution sur les solutions. Les méthodes de vérification dépendent fortement du langage et du solveur choisi. Pour cette raison, la vérification du code est séparée du pouvoir décisionnel dans notre approche. Comme pour la validation, la vérification peut être supportée par des valeurs de références [128] reconnues comme parfaitement exactes.

Dans notre cas, la vérification est ramenée à une mesure binaire : le code est correct ou non. Trois aspects du code sont vérifiés. Premièrement, le code ne doit pas présenter d’erreur d’implémentation. Deuxièmement, la résolution doit converger vers une ou plusieurs solutions. Enfin, les premières solutions identifiées doivent être acceptables. Ce dernier point est subjectif et indépendant de l’exactitude, c’est l’expertise du concepteur qui lui permet de déterminer si les valeurs sont réalistes ou non. Si ces trois aspects sont vérifiés, l’implémentation est juste, sinon elle est fausse. Si l’implémentation est fausse, il faut revoir la construction du modèle. De fait il est souhaitable d’implémenter le modèle au fur et à mesure de la formalisation pour limiter les itérations.

III.3 Approche de formalisation d’un modèle mathématique en conception

III.3.1 Principe de la méthodologie globale

La figure III.4 représente dans un diagramme BPMN les grandes étapes de la méthodologie globale. Le point de départ est le cahier des charges issu de l’analyse du besoin et le concept issu de la recherche de concept. Tout ceci est synthétisé dans un ensemble de modèles fonctionnels, structurels et comportementaux. La démarche est divisée en quatre phases successives : la pré-modélisation, la formalisation du modèle initial, l’agrégation des modèles complémentaires et la synthèse du modèle. Le rôle de chaque étape est défini dans les paragraphes suivants.

Description succincte des rôles de chaque étape

1. Le rôle de la pré-modélisation est de fixer le cadre de la démarche pour, d’une part, construire

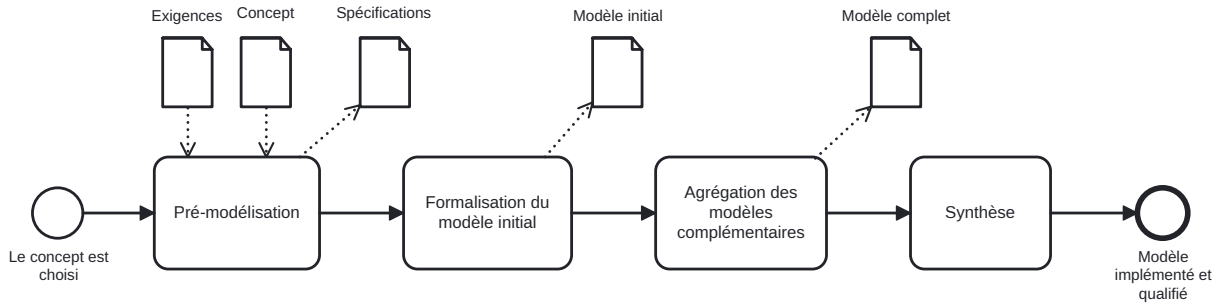


FIGURE III.4 – Principe de la proposition de démarche de formalisation d'un modèle mathématique en conception

le modèle autour du besoin initial, et d'autre part, représenter au mieux les éléments du concept. Pour parvenir au premier point, la première partie de la modélisation est consacrée uniquement au *problème principal*. Le problème principal est une réduction du problème à l'exigence principale du système. Le modèle associé ne doit représenter que les éléments essentiels du concept (composants, flux, fonctions), c'est la représentation la plus générique possible du problème que l'on désigne par *modèle initial* (MI). Pour que le modèle représente correctement le concept, nous proposons d'anticiper une partie de ses constituants. Concrètement, il s'agit de caractériser les variables de conception, les critères de performance, leur domaine respectif, leur unité et les leviers d'action sur les V_{co} . Le résultat de cette anticipation est synthétisé dans le *tableau des spécifications*.

2. La seconde phase est la représentation du problème principal dans un MI. Le tableau des spécifications permet de guider le choix du modèle. Le modèle est alors adapté au contexte, qualifié partiellement et implémenté pour valider cette phase. Le MI résultant est le socle de la modélisation complète.
3. Dans la troisième phase, le MI est complété par la modélisation des composants, flux et fonctions omis lors de la phase précédente. Ces *modèles complémentaires* (MC) sont agrégés au fur et à mesure. L'évaluation du pouvoir décisionnel du modèle est mis à jour à chaque agrégation. Cette phase s'arrête lorsque le modèle est complet ou lorsque plus aucun modèle complémentaire pertinent ne peut être agrégé. Lors de cette phase, le concepteur doit être particulièrement attentif à la cohérence du modèle.
4. Finalement, le modèle résultant est implémenté et qualifié dans la phase de synthèse. Le dernier aspect du pouvoir décisionnel est évalué en mesurant l'exactitude si des comportements de référence peuvent être identifiés. Les paragraphes suivants reviennent en détail sur chaque phase.

III.3.2 Pré-modélisation

III.3.2.a Identifier le problème principal et ses éléments conceptuels

Le problème principal est le problème de conception réduit aux besoins principaux dans le cahier des charges et aux composants associés. Les fonctions principales sont les fonctions qui permettent de satisfaire les besoins principaux. En d'autres termes, ce sont les fonctions suffisantes qui confèrent au système son

TABLE III.3 – Tableau des spécifications d’un concept

Composants	Variable	Unités	Catégorie	Levier/Direction	Ordre de grandeur
Système	masse	kg	V_p	Min	10
Composant 1	longueur	m	V_{co}	Dimensionnement	$[0, 1]$
	hauteur	m	V_{co}	Dimensionnement	$[0.001, 0.002]$
Composant 2	matériau	-	V_{co}	Choix	$\{1, 2, 3\}$
...

essence. Ces fonctions principales peuvent être identifiées à partir d’une représentation hiérarchisée des exigences du cahier des charges. C’est notamment le rôle du diagramme des exigences dans le langage SysML. Pour le système batterie, le problème principal est défini par les fonctions « stocker de l’énergie » et « délivrer l’énergie stockée » dans le tableau II.1. Une fois le problème principal identifié, il faut sélectionner les éléments du concept strictement nécessaires à la résolution des fonctions principales (composants, flux physiques, critères de performances).

III.3.2.b Spécifications d’un problème de conception

Construire les spécifications du problème de conception consiste à matérialiser l’espace de conception et l’espace des objectifs tels que prévu par la recherche de concept. L’espace de conception du concept est tiré des représentations comportementales et structurelles. Pour chaque composant du système, on identifie la ou les V_{co} parmi les paramètres de conception (par exemple en s’aidant du diagramme paramétrique SysML). L’expertise du concepteur et les données sur l’existant permettent de fixer les bornes des domaines de chaque variable et le type de levier de conception (dimensionnement ou choix). Les critères de performance découlent des exigences du cahier des charges fonctionnelles. Pour ces variables, l’information est généralement moins complète que pour les V_{co} . On donnera plutôt leur unité, l’ordre de grandeur et la direction de performance. Le résultat de cette étape est le tableau des spécifications III.3.

III.3.3 Formalisation du modèle initial

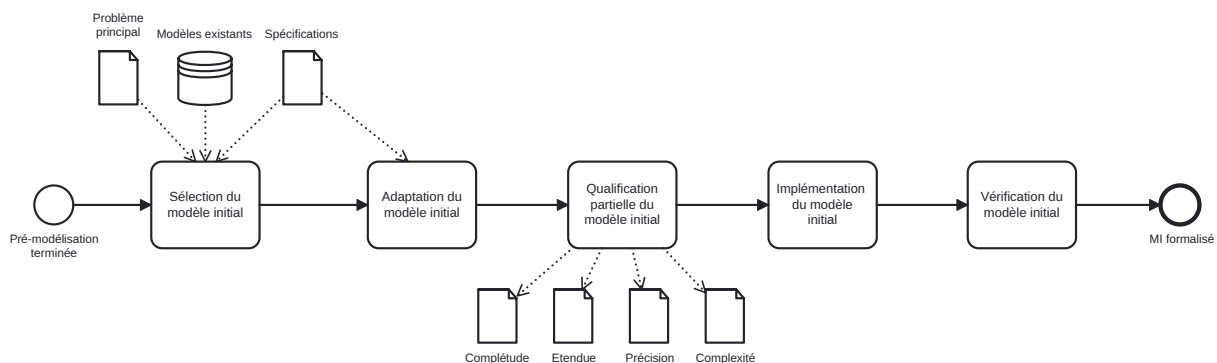


FIGURE III.5 – Phase 2 de formalisation du modèle initial

La figure² III.5 présente le sous-processus de la deuxième phase. La formalisation du MI repose sur le problème principal, le tableau des spécifications et les modèles existants du système. Les modèles existants proviennent de l'expertise du concepteur, de la littérature ou de banques de modèles. Si plusieurs modèles semblent pertinents pour répondre au problème de conception, les principes suivants permettent de guider le choix du concepteur :

- L'information doit être complète. Il faut connaître en particulier les valeurs des constantes, les bornes des domaines, le cadre des formules empiriques, les hypothèses de modélisation, etc.
- Les hypothèses du modèle existant ne doivent pas être en contradiction avec le cadre du concept.
- Le nombre d'expressions et de paramètres issus de simulations empiriques doit être limité. Les modèles semi-empiriques sont difficilement réutilisables. Chaque expression empirique risque de réduire la précision du modèle et d'impacter l'exactitude.
- Le modèle final doit respecter les leviers de conception prévus par le concept. Un choix de composant ne peut pas être représenté par des variables numériques.
- Il est souvent pertinent de privilégier un modèle dont on maîtrise la composition et l'origine des données. Comprendre une nouvelle modélisation requiert un effort particulier et augmente le risque d'incohérence.

Dans la majorité des cas, un travail d'adaptation des modèles utilisés est nécessaire. Le tableau des spécifications permet de reconnaître les éléments à modifier. Le degré d'adaptation de modèles mathématiques en conception est dépendant du contexte, cependant plusieurs procédés fréquemment rencontrés peuvent être listés :

- Élargir ou réduire les domaines des variables pour coller à l'espace de conception du concept ;
- Éliminer les parties non associées au problème principal ;
- Modifier les types des variables selon les leviers de conception, par exemple convertir une variable continue en variable discrète ;
- Fixer des paramètres ou convertir une constante en variable ;
- Actualiser les valeurs des constantes et tableaux de valeurs.

Cette phase est source d'erreurs de modélisation, le concepteur doit être particulièrement vigilant à la cohérence du résultat pour chaque remaniement. Le MI est ensuite qualifié partiellement en évaluant les caractéristiques du pouvoir décisionnel qui peuvent être estimées : la complexité, la précision et l'étendue. Les valeurs de ces grandeurs sont mises à jour au fur et à mesure de l'avancée de la démarche. Le concepteur est guidé par l'état du pouvoir décisionnel du modèle. Il faut chercher à augmenter la complétude et l'étendue du modèle sans trop dégrader la précision et la complexité.

La dernière étape de la phase 2 est une implémentation préliminaire du MI. Le but est de s'assurer de la cohérence du MI avant d'entamer l'agrégation des modèles complémentaires. L'évaluation de la complexité à l'étape précédente et les spécifications du concept permettent d'identifier le solveur et le langage pour le MI. En fonction du système, le concepteur réalise les tests unitaires pertinents pour vérifier l'existence de solutions et leur réalisme. Il n'est pas pertinent d'évaluer l'exactitude à ce stade du processus. Le livrable de cette phase est le modèle initial qualifié, implémenté et vérifié partiellement.

2. Un agrandissement de la figure est proposé en annexe B.2

III.3.4 Agrégation des modèles complémentaires

L'objectif est de compléter le modèle pour représenter au mieux les choix de conception identifiés pendant la recherche de concept. La première difficulté réside dans l'équilibre à trouver entre toutes les caractéristiques du modèle. En effet, certaines sont antagonistes comme la complétude et la complexité ou l'étendue et l'exactitude. La seconde difficulté est la disponibilité des modélisations ou leurs pertinences. Il n'est pas toujours possible de trouver un modèle analytique pour chaque élément du système (composant, flux, critère de performance). La figure³ III.6 présente les étapes de la phase 3.

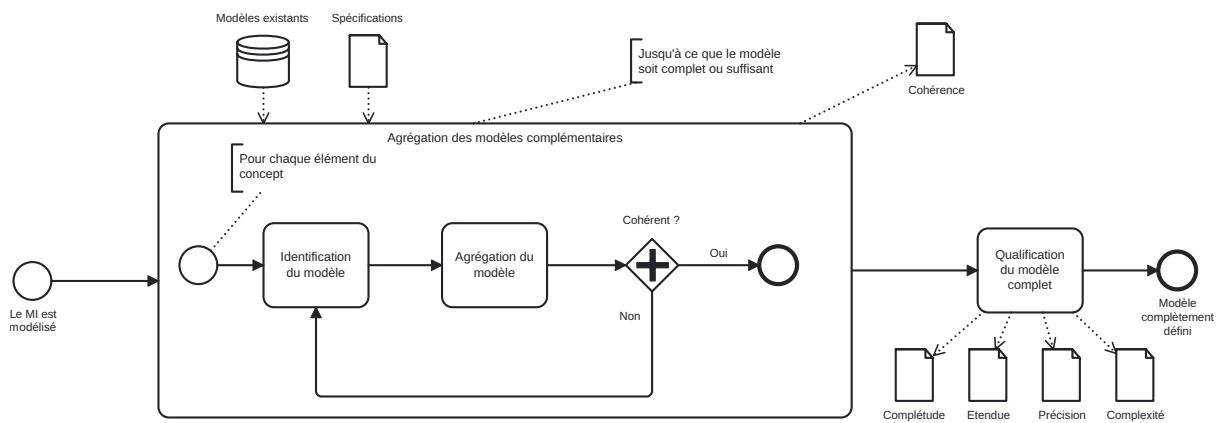


FIGURE III.6 – Phase 3 d'agrégation des modèles complémentaires (MC)

Le MI et les spécifications, développés lors des phases précédentes, permettent de guider la sélection des modèles complémentaires. Si plusieurs alternatives de modélisation existent pour le même élément, les principes utilisés pour le MI pourront de nouveau être appliqués. L'agrégation d'un MC nécessite en plus d'identifier les variables et paramètres qui font le lien avec le MI. La construction de ce lien peut demander l'utilisation de variables auxiliaires et d'expressions intermédiaires. La cohérence du résultat de l'agrégation d'un MC doivent être vérifiés systématiquement. L'implémentation des MC au fur et à mesure de leur agrégation et la vérification du code peut aider à maîtriser cette cohérence.

III.3.5 Implémentation et qualification du modèle

Pour finaliser l'estimation du pouvoir décisionnel du modèle, il reste à mesurer son exactitude. Au préalable, il faut implémenter le modèle complet, vérifier le code et déterminer la ou les références qui serviront à l'évaluation de l'exactitude. Deux cas de figure existent pour l'implémentation. Dans le premier cas, l'agrégation des MC n'a pas sensiblement modifié la complexité globale du modèle, on conserve alors ce qui a été réalisé. Dans le second cas, la complexité du modèle et l'implémentation doivent être revues. A priori, si le tableau de la spécification et le MI ont été réalisés correctement, la complexité finale peut être correctement anticipée pour éviter le second scénario.

3. Un agrandissement de la figure est proposé en annexe B.1

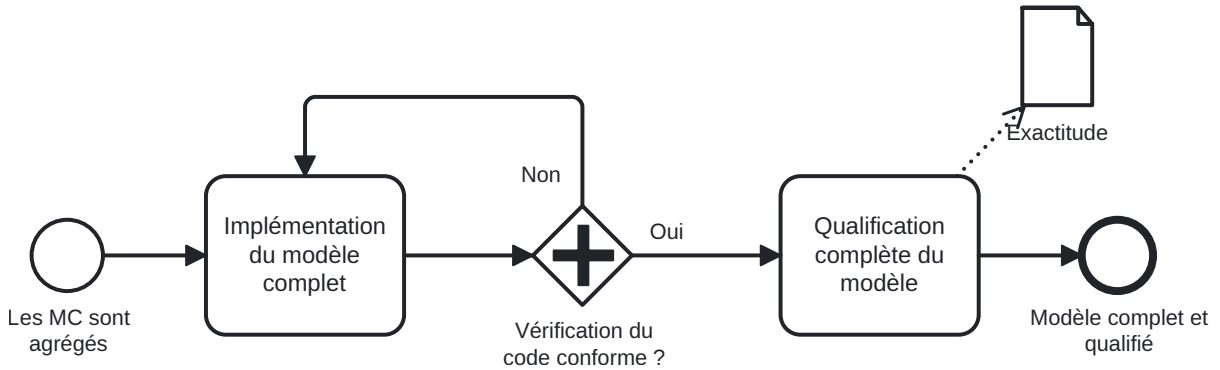


FIGURE III.7 – Phase 4 d’implémentation et de qualification du modèle

III.4 Application au problème de la batterie

Dans cette section, nous illustrons la démarche de formalisation en utilisant le problème de conception d’un pack batterie Li-ion (ou Lithium-ion) pour véhicule électrique. Pour ce cas d’étude, nous nous appuyons sur le cahier des charges II.1 ainsi que sur le concept développé tout au long du chapitre II. Notre expertise se base principalement sur une recherche bibliographique approfondie, et notre objectif principal est de tester la méthodologie de modélisation. Nous utiliserons des hypothèses pour simplifier le problème et clarifier les explications au fur et à mesure de la modélisation. La conception d’un pack batterie pour véhicule électrique est un sujet largement étudié, avec de nombreux niveaux de modélisation abordant différents aspects tels que la structure, la thermodynamique [78], le coût [123], l’écoconception [58], la durée de vie [6], et autres. L’information disponible est abondante, mais construire un modèle gros grains pertinent pour un contexte précis reste un vrai défi. La nomenclature utilisée est présentée dans le tableau III.4.

TABLE III.4 – Nomenclature du problème de conception de la batterie

Nom	Description	Unité
N_s	Nombre de cellules en série	-
N_p	Nombre de cellules en parallèle	-
N_{ms}	Nombre de modules en série	-
N_{mp}	Nombre de modules en parallèle	-
$Type$	Chimie de la cellule	-
$Form$	Géométrie de la cellule (cylindrique, pouch, prismatique)	-
J_c	Coût de la batterie	€
C_c	Capacité de la cellule	Ah
M_c	Masse de la cellule	kg
U_c	Voltage nominal de la cellule	V
U_{cmin}	Voltage minimal de la cellule	V
U_{cmax}	Voltage maximal de la cellule	V
V_c	Volume de la cellule	m ³
I_{max}	Courant maximal admissible par la cellule	A

TABLE III.4 – Nomenclature du problème de conception de la batterie

Nom	Description	Unité
R_{int}	Résistance interne de la cellule	mOhm
U_{bp}	Voltage nominal du pack	V
E_{batt}	Capacité de la batterie en W	W
M_{max}	Masse maximale du pack	kg
M_{base}	Masse du véhicule sans la batterie	kg
M_{batt}	Masse de la batterie	kg
REV	Autonomie du véhicule	km
J	Coût de la batterie	€
Q_{cy}	Dégradation cyclique de la capacité	%
LCD	Durée de vie kilométrique	km
N_{cy}	Durée de vie en nombre de cycles	-
N_{ans}	Durée de vie en années	ans
P_{peak}	Puissance maximale demandée	W
$C - rate$	Courant de charge ou de décharge	h-1
T	Température d'usage moyen	K
$T_{C/3}$	Température à $C - rate = C/3$	K
T_{charge}	Température interne en charge rapide (150kW)	K
ΔT	Augmentation de température dans la batterie	%
λ	Capacité thermique spécifique	J.kg-1.K-1
Q	Chaleur générée par la batterie	%
t	Temps d'un cycle de charge ou de décharge	s
D_{od}	Profondeur de décharge moyenne	%
MCP	Matériau de changement de phase	-
M_{mcp}	Masse du MCP	kg
τ_{mcp}	Quantité de MCP	%
λ_{mcp}	Capacité thermique spécifique du MCP	J.kg-1.K-1
L	Chaleur latente du MCP	J.kg-1
q_{cool}	Puissance du système de refroidissement	W
ref_{carter}	Référence matériau Carter	-
ref_{carter}	Référence système de gestion BMS	-
GWP	Émissions de GES à l'utilisation et fabrication	kg.co2-eq
AC	Acidification	kg.So2-eq
EU	Eutrophisation	kg.PO4—eq
RD	Déplétion des ressources	kg.Sb-eq
k_{avg}	Consommation d'énergie moyenne massique	Wh.km-1.kg-1
E_{ut}	Énergie utilisée pour déplacer la batterie	W
E_{ri}	Perte d'énergie à l'utilisation	W
I_{elec}	Impact moyen pour la production d'un kilowatt-heure	-
I_{manuf}	Impact pour la fabrication d'un kilo de cellule	-

Dans notre cahier des charges, un catalogue de cellules est évoqué. Pour fixer les idées, ce catalogue est

matérialisé par le tableau B.1 en annexe. Ces données sont issues de différents fournisseurs de cellules de batterie Li-ion pour voiture électrique. Li-ion est la technologie la plus utilisée dans les véhicules actuels. Les géométries de cellule disponibles sont limitées à cylindrique et prismatique et les chimies à NMC (Nickel Manganèse Cobalt) et LFP (Lithium Fer Phosphate) [208]. Enfin, l'impact environnemental est calculé sur quatre indicateurs : les émissions de gaz à effet de serre (GWP), l'acidification (AC), l'eutrophisation (EU) et la consommation des ressources (RD). La base de données utilisée est EcoInvent⁴ et la méthode CML [66, 44].

III.4.1 Pré-modélisation

III.4.1.a Problème principal

Le cahier des charges de la batterie est simple à interpréter, les fonctions principales de la batterie sont *stocker de l'énergie* et *délivrer l'énergie stockée au véhicule*. Les composants liés à leur réalisation sont les cellules, les autres sous-systèmes ne servent pas directement cet objectif. Le contexte actuel de dimensionnement des constructeurs européens de voitures électriques impose de choisir des cellules dans un catalogue fournisseur. La cellule est donc le plus bas niveau structurel de modélisation. Le concept réduit à son besoin principal est résumé dans les diagrammes SysML de définition de blocs III.8b et des exigences III.8a.

III.4.1.b Spécifications du concept

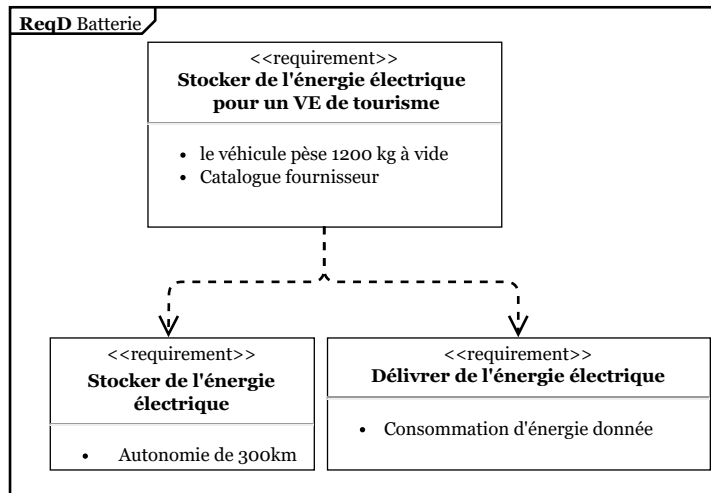
Le contenu attendu dans le tableau des spécifications est proche de l'information synthétisée dans un diagramme paramétrique en langage SysML. Dans un premier temps, il faut identifier les paramètres du diagramme qui sont des leviers de conception et les composants associés à ces paramètres. Chaque composant du système doit être représenté par une variable de conception à minima. En respectant le contexte du cahier des charges, on estime dans un deuxième temps le domaine de validité des V_{co} listés. Les domaines réels ou discrets ainsi créés ne doivent éliminer aucune alternative de conception. Les bornes ne peuvent être infinies. Les leviers de conception sont ensuite différenciés entre un choix (discret) ou un dimensionnement (continu), par défaut la valeur du levier est le dimensionnement.

Les V_p sont obtenues à partir du cahier des charges et des modélisations fonctionnelles du concept. Pour la batterie, 10 critères sont identifiés : volume, masse, coût, 4 impacts environnementaux, durée de vie et la température d'utilisation pour un usage moyen. On associe à chacun de ces critères une direction de performance et l'ordre de grandeur de la valeur attendue à partir des connaissances sur l'existant. Le tableau III.5 donne le résultat des spécifications du concept de la batterie. Le modèle que l'on cherche à construire devrait contenir l'ensemble de ces paramètres.

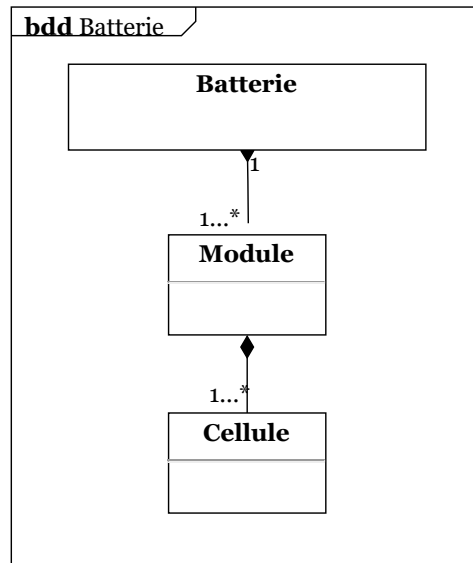
III.4.1.c Conjectures sur les caractéristiques décisionnelles du modèle

Avant de commencer la modélisation, il est possible d'anticiper certaines des caractéristiques décisionnelles du modèle à partir des spécifications du concept. Il s'agit aussi de donner les valeurs de référence pour le calcul de la complétude et de l'étendue. La précision est fonction de la tolérance sur les données

4. <https://ecoinvent.org/>



(a) Diagramme des exigences (ReqD) du MI de la batterie



(b) Diagramme de définition de blocs (BDD) du MI de la batterie

FIGURE III.8 – Modélisations du problème initial de *batterie* par des diagrammes SysML

TABLE III.5 – Tableau des spécifications de la batterie (Dim = Dimensionnement)

Composants	Variables	Catégorie	Levier	Domaine
Batterie	N_{ms}	V_{co}	Dim	[1, 100]
	N_{mp}	V_{co}	Dim	[1, 100]
	E_{batt}	V_p	Max	[30000, 200000]
	V_{batt}	V_p	Min	0.4
	M_{batt}	V_p	Min	600
	J_{batt}	V_p	Min	10000
	LCD	V_p	Max	300000
	N_{ans}	V_p	Max	10
	GWP	V_p	Min	10000
	AC	V_p	Min	100
	EU	V_p	Min	30
	RD	V_p	Min	2
	T	V_{aux}	-	25
Module	N_s	V_{co}	Dim	[1, 2000]
	N_p	V_{co}	Dim	[1, 2000]
Cellule	C_c	V_{co}	Choix	{3.2, 300}
	$Type$	V_{co}	Choix	{ NMC , LFP }
	$Form$	V_{co}	Choix	{ Cyl , $Prism$ }
Refroidissement actif	q_{cool}	V_{co}	Dim	[0, 10000]
Refroidissement passif	τ_{mcp}	V_{co}	Dim	[0, 0.15]
	MCP	V_{co}	Choix	{1, 2}
Carter	ref_{carter}	V_{co}	Choix	{1}
BMS	ref_{BMS}	V_{co}	Choix	{1}

des catalogues fournisseurs. La valeur de référence k_{max} pour la complétude est la somme des éléments du concept (flux, fonctions et composants) identifiés par la recherche de concept sans leurs multiplicités. Soit, en considérant une fonction pour chaque indicateur d’impact, 2 flux (thermique et électrique), 13 fonctions (11 exigences et 2 besoins dans II.1) et 6 composants (modules, cellules, refroidissement actif, refroidissement passif, Carter et BMS) finalement $k_{max} = 21$. La valeur de référence pour l’étendue est le nombre de V_{co} (équation III.1). À partir du tableau III.5 on obtient $n_{vco} = 12$. Enfin, les valeurs que l’on peut prévoir pour la complexité sont données dans le tableau III.6.

TABLE III.6 – Complexité des spécifications du concept du pack batterie

n_{vco}	n_{eq}	n_{ineq}	n_{vp}	$[d]$	c_{tr}	$T(\mathcal{P})$
12	?	?	10	Mixte	T	$\approx 1e^{14}$ ($\epsilon = 10^{-3}$)

III.4.2 Formalisation du problème initial

De nombreux modèles de batterie sont disponibles dans la littérature [149, 196, 156, 58, 13, 137, 33, 98]. Le premier travail est d’identifier le niveau de modélisation adapté. Les modélisations avec des informations incomplètes ne sont pas retenues. On regarde ensuite les hypothèses de conception. La contrainte du choix des cellules dans un catalogue permet d’éliminer les modélisations de l’état interne d’une cellule. Plusieurs modélisations sont trop spécialisées pour être considérées à ce stade du processus. Par exemple le modèle de [156] est un modèle de coût dédié aux cellules cylindriques et le modèle proposé par [149] comprend des dérivées temporelles peu pertinentes dans notre contexte et complexes à modéliser dans un CSP.

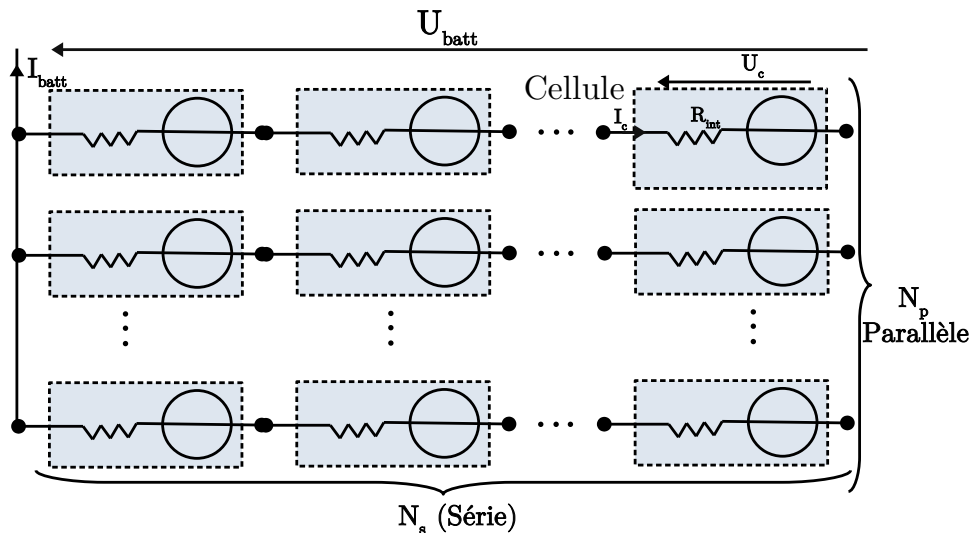


FIGURE III.9 – Modélisation de la batterie comme une combinaison de sources de tension idéales en série et en parallèle

Le modèle qui nous semble le plus adapté est celui de [137] qui se positionne au bon niveau d’approxi-

mation pour la conception architecturale. Ce travail a été choisit car il repose sur un choix de cellule sur catalogue comme prévu par notre concept. Il repose sur une représentation de la cellule comme une source de tension idéale avec une résistance interne constante [33]. Le dimensionnement du pack batterie revient alors à un assemblage de cellules en série N_s et en parallèle N_p , illustré par la figure III.9. Le regroupement des cellules en modules n'est pas pris en compte. Le modèle est clair et l'information complète. Les paramètres utilisés sont pertinents avec les spécifications. Cependant, le contexte est celui d'un prototype de moto électrique de course et le nombre d'éléments modélisés est trop grand. D'abord, on sélectionne les parties qui correspondent bien à la stricte représentation du problème initial. Cela revient à éliminer les expressions liées au volume, à la masse, au coût et au comportement thermique de la batterie. Ensuite, on adapte le modèle en modifiant les bornes des domaines de façon à couvrir l'espace de conception du problème établi dans le tableau des spécifications. Finalement, on obtient le MI suivant :

$$X = (C_c, U_c, U_{c,min}, R_{int}, I_{max}, N_s, N_p)$$

$$D = (\{0, 300\}, \{0, 4\}, \{0, 3\}, \{0, 200\}, \{0, 20\}, \{1, 2000\}, \{1, 2000\})$$

sujet à.

$$c_1 : E_{batt} \geq E_{min} \quad (III.4)$$

$$c_2 : U_c I_{max} N_s N_p \geq P_{peak} \quad (III.5)$$

$$c_3 : U_c N_s \leq U_{batt,max} \quad (III.6)$$

$$c_4 : N_s (U_{c,min} - R_{int}(I_{batt,max})/N_p) \geq U_{batt,min} \quad (III.7)$$

Avec :

$$E_{batt} = N_s N_p C_c U_c \quad (III.8)$$

La contrainte c_1 correspond à l'énergie (ou capacité) minimale E_{min} nécessaire à l'usage prévu pour le système. c_2 assure que le pack peut générer le pic de puissance attendu P_{peak} , par exemple pour une accélération de 0 à 100 km.h-1 du véhicule. c_3 donne la borne supérieure $U_{batt,max}$ du voltage de la batterie. Finalement, les composants de la voiture requièrent un voltage minimal pour fonctionner $U_{batt,min}$. Celui-ci doit être garanti même lorsque la batterie est déchargée (tension à $U_{c,min}$) et que la demande de courant du moteur $I_{batt,max}$ est maximale (contrainte c_4). Ces 4 expressions expriment les deux fonctions du problème principal : stocker et délivrer de l'énergie électrique. En plus de ces contraintes, on peut ajouter la contrainte de table induite par le catalogue de cellules C_{table} (III.9). Ce modèle est le *modèle initial* de notre problème de conception.

$$C_{table} = \begin{pmatrix} n_{cell} & C_c & U_c & R_{int} & U_{c,min} & I_{max} \\ 1 & 230 & 3,2 & 0,3 & 2,5 & 2,00 \\ 2 & 280 & 3,2 & 0,25 & 2,5 & 2,00 \\ 3 & 105 & 3,2 & 0,5 & 2,5 & 3,00 \\ 4 & 3,45 & 3,6 & 38 & 2,5 & 2,90 \\ 5 & 3,5 & 3,65 & 40 & 2,5 & 2,86 \\ 6 & 3,2 & 3,6 & 30 & 2,5 & 3,13 \\ 7 & 62 & 3,7 & 0,8 & 2,75 & 1,00 \\ 8 & 116 & 3,7 & 0,7 & 2,75 & 1,00 \\ 9 & 132 & 3,7 & 0,4 & 2,75 & 1,00 \\ 10 & 5 & 3,63 & 0,25 & 2,5 & 3,2 \end{pmatrix} \quad (III.9)$$

III.4.2.a Qualification du modèle initial

- La précision du modèle est égale à la précision des données du fournisseur, car aucun des paramètres n'est incertain et les expressions ne sont pas empiriques.
- La complexité est donnée dans le tableau III.7. Le nombre de V_{co} apparentes est de 7. Cependant, la table de contrainte du catalogue de cellule réduit ce nombre à 3. E_{batt} est la seule V_p représentée.

TABLE III.7 – Complexité du modèle initial (MI) du pack batterie

n_{vco}	n_{eq}	n_{ineq}	n_{vp}	$[d]$	c_{tr}	$T(\mathcal{P})$
3(7)	0	4	1	Discret	NL,T	$\approx 4e^7$ ($\epsilon = 10^{-3}$)

- Le MI considère le flux d'énergie électrique, les cellules, les fonctions "stocker de l'énergie", "délivrer de l'énergie" et "respecter un catalogue fournisseur". Le calcul de la complétude donne $C_P = 5/23$.
- L'étendue est estimée à partir des domaines des variables de conception dans les spécifications du concept. Les V_{co} considérées par le MI ont les mêmes domaines que les spécifications. Il suffit donc de compter le nombre de variables représentées : $C_c, Type, Form, N_s, N_p$. Ce qui donne $E_T = 5/15$.
- Le modèle est implémenté dans le langage Minibex⁵ pour vérifier l'existence de solution. Le modèle admet des solutions, quelques-unes sont données dans le tableau III.8. Ces valeurs ne sont pas réalistes car les domaines ne sont pas assez restreints, elles sont néanmoins cohérentes avec l'état du modèle.

TABLE III.8 – Quelques solutions du modèle initial de la batterie

Référence	N_s	N_p
EVE 18650	216	2000
PANASONIC NCR18650GA	95	4
EVE LF23	250	1500

III.4.3 Agrégation des modèles complémentaires

Une fois le MI construit, nous devons identifier et agréger les MC pour les fonctions, les composants et les flux restants. À partir des différents points de vue (diagrammes SysML) sur le concept on peut lister ces éléments :

- Composants : Système de refroidissement, Carter, BMS
- Fonctions : Coût, Masse, Volume, Température, Durée de vie, Impact environnemental, Autonomie
- Flux : Thermique, électrique.

Au fur et à mesure de l'identification et de l'adaptation de chaque MC, l'implémentation du modèle initial en langage Minibex est complétée. Le but est de vérifier que l'ajout d'une variable ou d'une expression ne rend pas le problème incohérent ou insoluble. Pour le moment, on s'assure systématiquement de l'existence de solutions et du réalisme des valeurs obtenues sans considérer l'exactitude.

5. <http://www.ibex-lib.org/>

III.4.3.a Contraintes de masse, de volume et de coût

Nous n'avons pas pu identifier de modélisation adaptée à notre cadre pour le Carter et le BMS. Ces éléments pourront être considérés pendant la phase de conception détaillée. Les exigences sur la masse et le volume sont modélisées dans la référence déjà utilisée pour construire le MI [137]. Ces expressions sont cohérentes avec le MI par construction. Réutiliser ces expressions permet de limiter le risque que le modèle global ne devienne incohérent. M_{batt} et V_{batt} sont respectivement la masse et le volume total de la batterie. Ensuite, les contraintes c_5 et c_6 qui expriment respectivement la masse maximale $M_{batt,max}$ et le volume maximal $V_{batt,max}$. Les expressions de M_{batt} et V_{batt} seront construites une fois l'ensemble des composants considéré.

$$c_5 : M_{batt} \leq M_{batt,max} \quad (III.10)$$

$$c_6 : V_{batt} \leq V_{batt,max} \quad (III.11)$$

En revanche, le modèle de coût n'est pas adapté car les prix des cellules utilisées dans le catalogue fournisseur sont trop approximatifs. Pour ce cas d'étude, on s'appuie plutôt sur le coût par kilowattheure moyen J_c (€/kWh) pour chaque type de batterie (NMC et LFP). Le prix est ensuite estimé à partir de la masse totale des cellules pour donner la contrainte c_7 :

$$c_7 : N_s N_p M_c J_c \leq J_{max} \quad (III.12)$$

Avec $J_c = 70$ €/kWh pour les cellules LFP prismatiques et $J_c = 100$ €/kWh pour les cellules NMC prismatiques. Le prix des cellules NMC cylindriques est en moyenne 15% plus élevé que les cellules NMC prismatiques pour la même densité d'énergie [28]. Les expressions des MC précédents sont simples, la cohérence avec le MI est immédiate.

III.4.3.b Impact environnemental

La modélisation de l'impact environnemental d'une batterie en conception est moins immédiat. On cherche à estimer l'impact à la fabrication et à l'utilisation pour les 4 indicateurs : GWP , AC , EU et RD . Le travail de [58] propose une estimation du GWP en se basant sur l'impact massique à la fabrication et l'impact du kilowattheure dans le pays d'utilisation. E_{ut} est l'énergie dépensée pendant la phase d'utilisation pour déplacer la batterie seule (équation III.14). E_{r_i} sont les pertes d'énergie due à l'efficacité interne de la batterie (équation III.15). Ce deux paramètres sont estimés à partir de l'autonomie du véhicule électrique REV pour une consommation d'énergie massique donnée k_{avg} ($Wh.km^{-1}.kg^{-1}$) (équation III.13). Puis, en ajoutant l'impact environnemental à la fabrication des cellules (= $GWP_{manuf} N_c N_p M_c$) on obtient le GWP total avec l'expression III.16.

TABLE III.9 – Coefficients pour le calcul des indicateurs environnementaux (données EcoInvent)

I	Unités	I_{manuf} (1kg LFP)	I_{manuf} (1kg NMC)	I_{elec} (1kWh France)
GWP	kg.co2-eq	8.33	15.72	0.0855
AC	kg.So2-eq	4.84e-2	2.014e-1	4.36e-4
ET	kg.PO4 ⁻⁻⁻ -eq	3.36e-2	5.14e-2	1.396e-4
RD	kg.Sb-eq	1.644e-4	3.68e-3	3.42e-6

$$REV = \frac{REV_{init} + REV_{end}}{2} \quad (III.13)$$

$$E_{ut} = \frac{0.3M_{batt}E_{batt}LCD}{REV(M_{base} + M_{batt})} \quad (III.14)$$

$$E_{ri} = \frac{0.1E_{batt}LCD}{R_{ev}} \quad (III.15)$$

$$GWP = (E_{ut} + E_{ri})GWP_{elec} + N_c N_p M_c GWP_{manuf} \quad (III.16)$$

REV_{init} est l'autonomie initiale (équation III.17) du véhicule et REV_{end} l'autonomie finale. REV_{end} sera donnée avec le calcul de la durée de vie (III.24). Les trois autres indicateurs évoqués sont obtenus de la même façon en généralisant cette équation III.18 à tout indicateur I dont on connaît l'impact à la fabrication massique I_{manuf} et l'impact au kilowattheure I_{elec} . Les exigences limitant chaque impact sont modélisées par les contraintes c_8 . Les valeurs des constantes du modèle sont données dans le tableau III.9. Elles ont été obtenues à partir de la base de données EcoInvent (Avril 2022). Ces modélisations s'appuient sur deux hypothèses importantes : 30% de l'énergie utilisée peut-être allouée à la masse dans E_{ut} [202] et l'impact à la fabrication est principalement dû aux cellules et à leur masse.

$$REV_{init} = \frac{E_{batt}}{k_{avg}(M_{base} + M_{batt})} \quad (III.17)$$

$$I = (E_{ut} + E_{ri})I_{elec} + N_c N_p M_c I_{manuf} \quad (III.18)$$

$$c_{8a} : GWP \leq GWP_{max} \quad (III.19)$$

$$c_{8b} : AC \leq AC_{max} \quad (III.20)$$

$$c_{8c} : ET \leq ET_{max} \quad (III.21)$$

$$c_{8d} : RD \leq RD_{max} \quad (III.22)$$

III.4.3.c Durée de vie

De nombreux modèles semi-empiriques existent dans la littérature récente pour estimer les deux définitions de la durée de vie [190, 32, 158, 187, 100]. Ces modèles distinguent la durée de vie cyclique, liée au nombre de cycles de charge et de décharge, et la durée de vie calendaire, liée au vieillissement

de la batterie pendant le stationnement du véhicule [147]. Pour simplifier notre travail, nous considérons simplement la durée de vie cyclique N_{cy} que l'on peut traduire en durée de vie kilométrique LCD à partir de l'autonomie du véhicule REV . Les modélisations répertoriées dans [35] (équations III.23) permettent de considérer les deux chimies de cellules du catalogue (NMC [189] et LFP [158]). Les hypothèses et limites des modèles sont bien établies par l'auteur et ne sont pas contradictoires avec notre concept.

$$Q_{cy} = \begin{cases} f_{NMC}(T, D_{od}, N_{cy}, C - rate, Cc) & \text{cas NMC} \\ f_{LFP}(T, D_{od}, N_{cy}, C - rate, Cc) & \text{cas LFP} \end{cases} \quad (\text{III.23})$$

$$REV_{end} = (0.9 - Q_{cy})REV \quad (\text{III.24})$$

$$LCD = \frac{N_{cy}(REV + REV_{end})D_{od}}{2} \quad (\text{III.25})$$

$$c_9 : Q_{cy} < 0.2 \quad (\text{III.26})$$

Q_{cy} est la perte de capacité de la batterie due au vieillissement cyclique. On considère généralement que la diminution totale de capacité ne doit pas dépasser 20%, autrement la batterie est considérée obsolète (contrainte III.26). Cette modélisation permet de donner une estimation de N_{cy} . On considère généralement que 10% de l'énergie de la batterie est perdue par l'effet joule dans le calcul de l'autonomie finale REV_{end} [19] (équation III.24). LCD est calculée à partir de l'autonomie moyenne REV dans l'expression III.25. Le modèle f_{LFP} n'est valable que pour certaines valeurs du taux de décharge $C - rate$. Nous considérons ici le taux de décharge $C/2$ comme point de fonctionnement pour estimer le vieillissement. L'expression est fonction de la température d'utilisation T et de la profondeur de décharge moyenne. La température de la batterie pour une décharge ou une charge à $C/2$ est estimée à partir du modèle thermique décrit dans les sections suivantes. Les expressions complètes de f_{NMC} et f_{LFP} sont données en annexe D.4.2

Limites du modèle de vieillissement

Cette modélisation de la durée de vie repose sur des expressions empiriques obtenues à partir de cellules dont les caractéristiques sont différentes des cellules du catalogue fournisseur. En particulier, les capacités et les conditions d'utilisation sont différentes. De plus, le point de fonctionnement $C/2$ est un compromis entre validité du modèle et usage moyen, il ne reflète pas une utilisation réelle de la batterie. En somme, le modèle ne permet qu'une estimation grossière de la durée de vie. La réutilisation de ces expressions dans le modèle doit être soumise à un point d'arbitrage. D'un côté, l'agrégation risque de diminuer la précision et l'exactitude du modèle. De l'autre, ne pas considérer la durée vie limiterait la complétude et l'étendue du modèle final. Nous avons choisi de maintenir cette modélisation de la durée de vie cyclique de la batterie.

III.4.3.d Modèle thermique et système de refroidissement

Les modèles thermiques rencontrés dans la littérature utilisent globalement le même niveau de modélisation pour le flux thermique émis par les cellules [154, 78, 98, 136]. Sous les hypothèses d'un régime permanent et d'une répartition uniforme de la température à la surface des cellules, on estime la chaleur générée q_c à partir de leur résistance interne R_{int} et pour une puissance donnée P (équation III.27). On en déduit les expressions III.28 et III.29. La première donne la chaleur totale générée par la batterie Q . La deuxième donne l'augmentation de la température interne de la batterie sans refroidissement Δ_T pour une durée t . λ_c est la capacité thermique spécifique de la cellule. Ce modèle permet d'estimer l'augmentation de température pour différents points de fonctionnements. Un point de fonctionnement est caractérisé par un taux de décharge $C - rate$, un temps d'utilisation t et la puissance correspondante P .

$$q_c(P) = R_{int} \left(\frac{P}{U_c} \right)^2 \quad (\text{III.27})$$

$$Q(P, t) = q_c N_c N_p t \quad (\text{III.28})$$

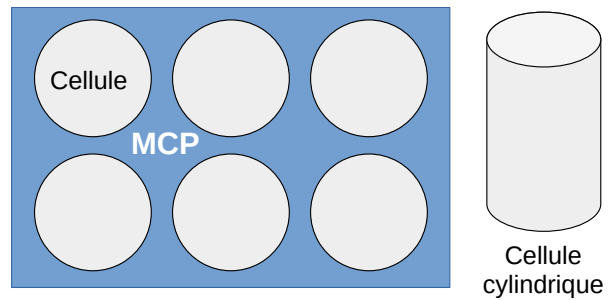
$$\Delta_T(P, t) = \frac{Q}{N_s N_p M_c \lambda_c} \quad (\text{III.29})$$

Refroidissement passif

Notre concept considère deux types de refroidissement : le refroidissement passif, avec un matériau à changement de phase (MCP), et un refroidissement actif à partir de la circulation d'un fluide (air, eau ou mélange spécifique) [154]. L'utilisation d'un MCP n'est pertinent que pour les cellules cylindriques qui ont une résistance interne importante [129]. Il permet de stabiliser la température interne du système. L'intervalle de température correspondant au changement de phase doit être choisi pour éviter aux cellules d'atteindre des températures critiques (généralement au-delà de 55°C [154]). Le MCP peut être modélisé comme une quantité de matière à ajouter au système entre les cellules cylindriques comme sur la figure III.4.3.d. Cette quantité de matière est représentée par le ratio τ_{mcp} de la masse du MCP (M_{mcp}) sur la masse de la batterie (M_{batt}) [137] (équation III.30). Le MCP modifie la capacité thermique globale du système et donc l'augmentation de température Δ_T (équation III.31). Le modèle utilisé est celui de [68]. La variable de conception du système de refroidissement passif est le ratio τ_{mcp} . Le MCP retenu pour notre problème est la n-paraffine 5838 [193].

$$\tau_{mcp} = \frac{M_{mcp}}{N_c N_p M_c} \quad (\text{III.30})$$

$$\Delta_T(P, t) = \frac{Q}{N_s N_p M_c (\lambda_c + \tau_{mcp} \lambda_{mcp})} \quad (\text{III.31})$$



Pour dimensionner la quantité de MCP, nous considérons une décharge constante en C/3 de la batterie, c'est-à-dire une puissance délivrée de $P_{C/3} = E_{batt}/3$ pendant $t_{C/3} = 3$ (h). Ce point de fonctionnement a notamment été utilisé pour dimensionner le système de refroidissement dans [98]. Le modèle thermique appliqué à $P_{C/3}$ et $t_{C/3}$ permet d'évaluer la température atteinte $T_{C/3}$ (équation III.32). Avec l'inégalité III.33, le système est contraint à rester dans l'intervalle de fonctionnement optimal (entre 15 et 35 °C [129]) .

$$T_{C/3} = T_{amb} + \Delta_T(P_{C/3}, t_{C/3}) \quad (\text{III.32})$$

$$c_{10} : T_{C/3} \leq 35 \quad (\text{III.33})$$

Refroidissement actif

La modélisation fine du système de refroidissement actif est complexe. Le modèle le plus adapté à notre échelle est celui proposé dans le projet BatPac [98]. Le système de refroidissement est caractérisé par sa capacité de refroidissement q_{cool} (W). Le système est ensuite choisi dans le catalogue III.10. L'auteur a choisi des systèmes qui utilisent comme fluide de transfert de chaleur un mélange éthylène-glycol/eau. Cependant, comme les systèmes sont ici dimensionnés par leurs capacités de refroidissement maximales, il est envisageable d'utiliser un fluide différent en dimensionnant correctement le système pour respecter cette capacité maximale. En considérant le système de refroidissement actif, l'expression de la chaleur Q devient : $Q = q_c N_c N_p t - q_{cool} t$

TABLE III.10 – Catalogue de systèmes de refroidissement

Référence	Capacité q_{cool} (W)	Masse M_{cool} kg	Volume C_{cool} l	Coût J_{cool} €
Refrig 0	0	0	0	0
Refrig 1	500	2	0.8	30
Refrig 2	1000	3	1.2	80
Refrig 3	3000	5	2	120
Refrig 4	6000	7	2.8	200

Le système de refroidissement actif est dimensionné pour permettre à la batterie de supporter une charge rapide. Pour cela, on considère le cas extrême d'une recharge rapide totale de la batterie sur une borne délivrant une puissance de P_{charge} (kW). Le temps de recharge de la batterie dans ces conditions est $t_{charge} = E_{batt}/P_{charge}$ (h). Pour notre modèle, la température de la batterie en charge rapide T_{charge} ne doit pas dépasser 45 °C afin d'éviter des dégradations trop importantes (équations III.34 et III.35) [78].

$$T_{charge} = T_{amb} + \Delta_T(P_{charge}, t_{charge}) \quad (\text{III.34})$$

$$c_{11} : T_{charge} \leq 45 \quad (\text{III.35})$$

Masse et volume total de la batterie

À partir des modèles complémentaires précédents on peut obtenir les expressions définitives de la masse et du volume. On a les masses de l'ensemble des cellules avec $M_{cell} = N_c N_p M_c$, du refroidissement actif avec M_{cool} et du refroidissement passif $M_{mcp} = N_c N_p M_c \tau_{mcp}$. On déduit de la même façon le volume de ces composants. Pour estimer la masse et le volume total de la batterie, on suppose que le BMS et le carter représente 35% de la masse totale et 50% du volume total en se basant sur les différentes données de la littérature [105, 19, 98]. On obtient respectivement les expressions suivantes avec la masse en kilogrammes et le volume en litres :

$$M_{batt} = (N_s N_p M_c (1 + \tau_{mcp}) + M_{cool}) / 0.65 \quad (\text{III.36})$$

$$V_{batt} = (N_s N_p V_c (1 + 1000 \frac{\tau_{mcp}}{\rho_{mcp}}) + V_{cool}) / 0.5 \quad (\text{III.37})$$

III.4.4 Synthèse du modèle

III.4.4.a Implémentation

La complexité du modèle complet est résumée dans le tableau III.11. Le nombre de degrés de liberté n_{vco} est réduit à 5 par la présence de contraintes de tables. Le nombre d'égalités dépend de l'implémentation. Pour que le code soit clair, le concepteur doit expliciter les expressions et variables auxiliaires. Le langage Minibex et les algorithmes d'exploration que nous avons développé pour ce travail au chapitre IV sont adaptés pour l'implémentation de ce modèle. Le code Minibex complet est donné en annexe B.3.

TABLE III.11 – Complexité du modèle initial (MI) du pack batterie

n_{vco}	n_{eq}	n_{ineq}	n_{vp}	$[d]$	c_{tr}	$T(\mathcal{P})$
5 (17)	≈ 14	11	10	Mixte	NL,T,CD	$\approx 2^{11}$ ($\epsilon = 10^{-3}$)

III.4.4.b Mesure de l'exactitude

Le développement d'un prototype comme référence n'est pas envisageable, il faut donc utiliser des architectures existantes ou référencées dans littérature pour évaluer notre modèle. Les données constructeur sont difficiles à obtenir, peu de batteries existantes sont bien détaillées dans la littérature. Pour valider notre approche, nous nous sommes basés sur les benchmarks du site⁶. Les caractéristiques du véhicule de référence sont données dans le tableau III.12. Les données de référence ne permettent pas de vérifier les modèles de refroidissement. En revanche, il est précisé que le système de refroidissement réel est capable de maintenir une température de 25°C dans la batterie.

Pour faire la comparaison, on considère les points de fonctionnement décrits dans le benchmark (Tableau III.13a) avec une profondeur de décharge de 100% à chaque cycle. Les V_{co} sont fixés avec les valeurs de références, les V_{co} du système de refroidissement sont ignorées. Le modèle est ensuite résolu pour les différentes valeurs de T et de N_{cy} et les valeurs obtenues sont présentées dans le tableau de III.13b. En dehors des points de fonctionnement, le modèle donne les bonnes valeurs pour les caractéristiques globales simples de la batterie : E_{batt} et M_{batt} . Les valeurs attendues sur LCD et REV pour 1000 cycles

6. <https://pushevs.com/>

TABLE III.12 – Caractéristiques des cellules et de la batterie de référence utilisées pour le calcul de l’exactitude

Variable	Unité	Valeur
N_s	-	96
N_p	-	2
C_c	Ah	64.6
V_c	l	0.467
Type	-	NMC
Form	-	Prismatique
U_{cmin}	V	2.5
U_{cmax}	V	4.2
U_c	V	3.6
M_c	kg	0.9649
R_{int}	Ohm	0.8
I_{max}	A	128
E_{batt}	kWh	45,34
M_{batt}	kg	185.26

à 25 °C sont légèrement surestimées par rapport à la réalité. L’exactitude est mesurée avec l’expression III.2 sur les valeurs de Q_{cy} . L’erreur maximale est de 0.0062 en valeur absolue soit 38% en valeur relative. Globalement, le modèle est optimiste par rapport à la réalité.

TABLE III.13 – Comparaison des valeurs des différents points de fonctionnement du pack batterie entre la référence (gauche) et le modèle (droite)

(a) Référence					(b) Modèle				
\tilde{T}	\tilde{N}_{cy}	\tilde{Q}_{cy}	\tilde{LCD}	\tilde{REV}	T	N_{cy}	Q_{cy}	LCD	REV
-	0	0	0	≈ 250	-	0	0	0	262.025
25	200	0.06	-	-	25	200	0.045	-	-
25	1000	0.16	>225000	200	25	1000	0.098	262025	210.054
25	1400	0.2	-	-	25	1400	0.12	-	-
45	200	0.08	-	-	45	200	0.08	-	-
45	1000	0.22	-	-	45	1000	0.173	-	-
45	1400	0.28	-	-	45	1400	0.20	-	-

III.4.4.c Pouvoir décisionnel global

Finalement, le pouvoir décisionnel global de notre modèle est estimé complètement :

1. Adéquation au concept

- **E_T = 7.5/12.** Les domaines prévus par la spécification des concepts sont globalement respectés, sauf le choix entre deux MCP pour le refroidissement passif. Chaque V_{co} manquante réduit de 1 la valeur de l’étendue.
- **C_P = 19/23.** Les éléments que nous n’avons pas pu modéliser sont : la répartition en module, le carter, le BMS et la durée de vie calendaire.

2. Résolution

- C_X est donné dans le tableau III.11
- $P_R = 10^{-3}$. Des expressions empiriques interviennent dans notre modèle pour le calcul du coût, du GWP et de la durée de vie cyclique. Ce sont les plus grandes sources d'imprécision du modèle. Les données dans les travaux à l'origine de ces modèles complémentaires ne nous permettent pas d'envisager une meilleure précision relative que 10^{-3} .

3. Validité

- $C_H = \mathbf{Vrai}$. La cohérence a été vérifiée pour l'agrégation de chaque MC. Les résolutions précédentes permettent de s'assurer définitivement de la vraisemblance des résultats et de la cohérence globale.
- $E_X(Q_{cy}) = \mathbf{0.0062}$. L'évaluation de l'exactitude indique que notre modèle sous-évalue le vieillissement cyclique des cellules. Il faudra prendre en compte cet aspect lors de l'interprétation des résultats.

Ces informations permettent d'apprécier la capacité de notre modèle à répondre au concept sur les trois aspects : adéquation, résolution et validité. Premièrement, le modèle est cohérent et retourne des solutions réalistes. L'étendue et la complétude quantifient la proportion d'éléments manquant au modèle. En convertissant les rapports en pourcentage, on obtient que notre modélisation couvre 73% du problème. Certains paramètres, comme la durée de vie, ne peuvent être évalués qu'avec une précision grossière. Globalement, les résultats permettent de comparer les architectures faisables avec une vision optimiste des performances. Une expertise plus poussée du système batterie permettrait d'améliorer le modèle, par exemple sur le choix des points de fonctionnement. Néanmoins, connaître le pouvoir décisionnel du modèle autorise le concepteur à faire des choix à partir de ses solutions architecturales.

III.5 Bilan du chapitre

« Comment formaliser un modèle CSP adapté pour la conception préliminaire d'un système complexe ? » Formaliser un modèle mathématique est un processus complexe et exigeant. Notre réponse à la question de recherche initiale est double. D'une part, il faut appliquer une méthodologie complète et efficace pour n'omettre aucun aspect du concept. D'autre part, en s'appuyant sur les bonnes métriques le concepteur peut caractériser le modèle sur son adéquation avec le concept, sa validité mathématique et enfin son apport décisionnel global.

Dans ce chapitre, nous avons revu le cadre existant pour la formalisation de modèle de dimensionnement. Les limites observées nous ont conduit à développer une nouvelle démarche dédiée à la modélisation de CSP pour la conception architecturale. Notre approche est centrée sur la capacité du modèle à faciliter la prise de décision, aussi appelée pouvoir décisionnel. Ce dernier repose sur 6 caractéristiques : la complexité, la complétude, l'étendue, la cohérence, la précision et l'exactitude. L'évaluation de ces aspects au fur et à mesure de la formalisation assure deux points essentiels. D'une part, l'adéquation du modèle avec le besoin et le concept. D'autre part, la cohérence mathématique du modèle et de son implémentation. Le processus a été éprouvé dans son ensemble avec le concept d'un pack de batterie pour véhicule électrique.

Finalement, la modélisation du niveau global du système (MI) vers le niveau local des composants (MC) favorise la réutilisation du modèle pour des concepts proches.

Notre proposition est conçue pour guider le concepteur dans l'étape de formalisation. Certains aspects de notre approche montrent des limites et ne sont pas pertinents dans tous les contextes. Les systèmes complexes n'adoptent pas nécessairement une hiérarchie structurelle et fonctionnelle claire. Dans ces conditions, le modèle initial peut être délicat à identifier sans inclure l'ensemble des constituants. En pratique, l'expertise de l'utilisateur est une condition nécessaire à la qualité du résultat, notamment pour repérer les incohérences. Notre définition du pouvoir décisionnel est aussi très adaptée à des modèles type CSP. Le temps nécessaire pour appliquer notre méthodologie reste fonction de la complexité du problème et des connaissances du concepteur.

CADRE POUR L'EXPLORATION PARTIELLE D'ESPACE DE RECHERCHE DE CSP MIXTES NON-LINÉAIRES

IV.1 Introduction

Dans ce chapitre, nous présentons un cadre pour explorer efficacement l'espace de recherche de problèmes de satisfaction de contraintes (CSP) en conception préliminaire. Ce cadre s'appuie sur le développement d'algorithmes Anytime Branch and Prune (AB&P) adaptés à la résolution de CSP mixtes non-linéaires. En particulier, nous cherchons à identifier quelle stratégie de recherche (combinaison de stratégies de bisection et de sélection) répond le mieux à ce contexte. Nous verrons que certaines caractéristiques de nos algorithmes doivent être adaptées pour que les performances restent pertinentes dans notre contexte d'application. Au préalable, plusieurs indicateurs de performances seront introduits pour supporter l'évaluation des différents algorithmes décrits. Les expérimentations porteront sur un problème unitaire, simple et modulable, puis sur un benchmark de problèmes de conception issues principalement de la littérature.

IV.1.1 Motivation

Notre travail est d'abord motivé par plusieurs constats sur les approches classiques en conception préliminaire. Premièrement, la plupart des méthodes de résolution de modèles gros grains reposent sur l'optimisation d'objectifs basés sur les variables de performances (V_p). Or ces modèles sont formalisés à partir d'hypothèses et d'incertitudes qui peuvent limiter fortement le caractère réellement "optimal" des solutions issues de la résolution. De plus, nous avons vu dans le chapitre III que ces modèles sont difficilement en complète adéquation avec le contexte de conception. Certaines exigences peuvent être omises de la modélisation ainsi que certains composants ou comportements physiques. Les solutions issues de tels modèles ne sont optimales qu'au sens mathématique. En sachant cela, le concepteur peut vouloir s'intéresser à certaines architectures réalisables non-optimales mais dont les performances restent pertinentes. D'autre part, le nombre croissant de critères à prendre en compte pour l'optimisation augmente la difficulté d'identifier le ou les meilleurs compromis entre ces critères. C'est notamment le cas en éco-conception où les multiples indicateurs forment autant d'objectifs à minimiser [29, 171]. Finalement, les modèles gros grains en conception ont généralement un espace de recherche avec un grand nombre

de solutions à considérer, en particulier pour les problèmes numériques. Une génération exhaustive des solutions n’est donc pas raisonnable, aussi bien en termes de temps de calcul que pour l’analyse de ces solutions.

L’exploration partielle est aujourd’hui peu exploitée, mais elle représente une piste intéressante pour traiter ces problèmes de conception préliminaire. Le principe est de générer un échantillon de solutions suffisamment grand et bien réparties dans l’espace de recherche pour être représentatif de l’espace réalisable complet. La génération de ces solutions ne fait intervenir qu’un seul mécanisme de discrimination sur les performances : les seuils définis par les exigences du cahier des charges.

IV.1.2 Contexte et objectif

Les problèmes traités dans ce chapitre sont des modèles de dimensionnement pour la conception préliminaire de systèmes complexes. Ce sont des CSP numériques ou mixtes avec des contraintes non-linéaires ou des contraintes de tables. Les modèles sont composés au maximum de quelques dizaines de variables de conception et de contraintes. Ces problèmes sont caractérisés par des espaces réalisables très grands avec beaucoup de solutions. Par exemple, on considère le problème simple du design d’une poutre, proposé dans [130] (appelé *beam_osy*). Ce CSP est composé de 4 variables continues, et de 3 contraintes d’inégalités. L’exploration complète de ce problème avec une précision absolue grossière de 10^{-1} pour les 4 variables conduit à 445332 solutions distinctes. Nous cherchons à traiter ce type de problèmes par une approche d’exploration partielle efficace et adaptée aux CSP en conception. Les algorithmes AB&P sont de bons candidats pour l’exploration de CSP numériques. La stratégie de sélection Depth and Most Distant First Search (DMDFS) introduite dans [26] permet d’obtenir une bonne diversité, mais demande des temps de calcul importants. De plus, l’implémentation proposée par l’auteur ne tient pas compte des particularités des problèmes de conception. Par exemple, les opérateurs ne tiennent pas compte des unités des variables et des différences d’ordre de grandeur entre les dimensions. Il est nécessaire de développer un cadre plus précis pour appliquer correctement des algorithmes AB&P dans notre contexte de dimensionnement.

La première partie de ce chapitre est consacrée à la définition de l’efficacité d’un algorithme d’exploration partielle. Trois indicateurs sont introduits pour mesurer la diversité au sein des résultats. Ces indicateurs vont permettre de comparer l’efficacité des stratégies de recherche et de bisection évoquées. Nous nous intéresserons aussi aux autres éléments qui influent sur l’exploration. Notamment les méthodes de bisection et la précision des variables. Pour implémenter nos algorithmes, nous avons utilisé la bibliothèque d’analyse par intervalles c++ IBEX¹ [88]. Les modèles CSP sont écrits en langage Minibex. Ce langage est déclaratif, le script suivant donne l’exemple d’implémentation du problème de conception *beam_osy* décrit dans [22].

1. <http://www.ibex-lib.org/>

Exemple de script Minibex sur le problème *beam_osy* [22]

```

Constants

Kg = 16;
E = 2e4;
Q = 50;
P = 600;
l = 200;
My = 30000;
Mz = 2500;

Variables

x1 in [10,80];
x2 in [10,50];
x3 in [0.9,50];
x4 in [0.9,5];

Constraints

Wy = (x3*(x1-2*x4)^3+2*x2*x4*(4*x4^2+3*x1*(x1-2*x4)))/(6*x1);
Wz = ((x1-2*x4)*(x3^3)+2*x4*x2^3)/(6*x2);
My/Wy + Mz/Wz <=Kg;
f1 = 2*x2*x4+x3*(x1-2*x4);
f1 <= 300;
I = (x3*(x1-2*x4)^3+2*x2*x4*(4*x4^2+3*x1*(x1-2*x4)))/12;
f2 = P*l^3/(48*E*I);
f2 <= 0.015;

End

```

IV.2 Indicateurs de qualité et profils de performance

Dans cette section, nous proposons une méthode pour évaluer et comparer l'efficacité des algorithmes AB&P. Le résultat d'une exploration partielle d'un espace de recherche n'est pas binaire car l'évaluation du temps de calcul n'est pas suffisante pour conclure sur les performances d'un algorithme. Il s'agit plutôt de mesurer leur capacité à générer des résultats suffisamment diversifiés pour un critère d'arrêt donné. Le critère d'arrêt est défini pour une précision ϵ donnée, par exemple avec un nombre maximal de solutions ou un temps de calcul.

Parmi les quatre propriétés introduites au chapitre II.3.4.a pour définir la diversité d'un résultat en optimisation multi-objectifs [104, 4], seules trois sont pertinentes pour qualifier le résultat d'une exploration partielle : l'uniformité, la couverture et la cardinalité. Les figures IV.1 illustrent trois explorations différentes du problème *cerceau*. Le résultat de la première exploration (figure IV.1a) respecte le nombre de solutions (cardinalité) suffisant pour représenter ce problème (ici 100 solutions), en revanche les solutions ne sont pas réparties uniformément et ne couvrent pas tout le problème. Pour la figure IV.1b, l'échantillon est uniforme mais ne respecte pas la cardinalité et la couverture du problème. Enfin, l'ex-

ploration partielle de la figure IV.1c couvre bien les limites de l'espace réalisable mais ne respecte pas les deux autres critères de diversité.

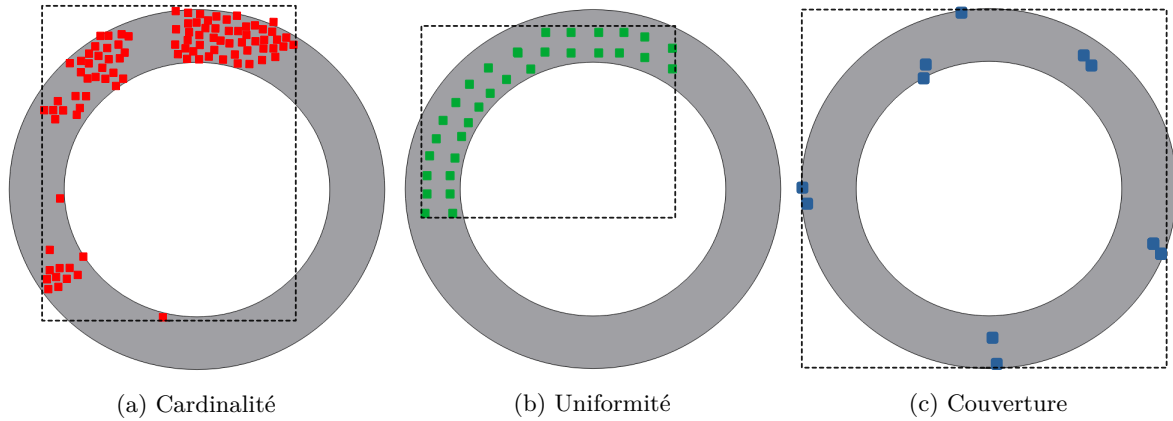


FIGURE IV.1 – Comparaison des propriétés attendues pour le résultat d'une exploration partielle.

On note Q_c la cardinalité d'un sous-ensemble de solutions. Par défaut, on considère comme critère d'arrêt un nombre maximal de solutions $\phi = Q_c^{max}$. Le temps de calcul t nécessaire pour générer ces Q_c^{max} solutions est un premier indicateur de performance. Dans ce cadre, nous cherchons à construire des grandeurs adaptées à l'évaluation de la diversité d'un sous-ensemble de solutions de taille Q_c^{max} .

Les opérateurs de mesure de la diversité les plus utilisés en optimisation se basent souvent sur un ou plusieurs points de référence. Par exemple l'hypervolume, l' ϵ -indicateur ou l'Inverted Generational Distance (IGD) [151, 169, 4]. Ces points de références sont liés aux objectifs et représentent des solutions idéales ou des approximations du front de Pareto. Dans une démarche d'exploration partielle, on considère que l'on ne peut se rattacher à ces points de références, souvent indisponibles en amont de la résolution. On ne dispose alors que des bornes des domaines initiaux des variables comme repères. Les indicateurs de qualité pour l'exploration ne peuvent s'appuyer que sur ces valeurs pour être construits. De plus, la diversité en optimisation ne porte que sur l'espace des objectifs. Dans notre cas, nous recherchons de la diversité à la fois dans l'espace des variables de conception (V_{co}) et dans l'espace des variables de performance (V_p).

IV.2.1 Construire un indicateur de diversité

Indicateur de qualité

Un indicateur de qualité Q est une fonction qui associe à un ensemble de solutions $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_p\}$, un réel $q \in \mathbb{R}$ [206].

Pour estimer la diversité d'un sous-ensemble, on distingue la mesure de l'uniformité et de la couverture. La première étape est d'identifier ou de créer une définition de distance adaptée [4]. Dans le

cadre de la conception, les dimensions sont associées à des grandeurs réelles (par exemple un coût et une puissance). Les notions de distances classiques qui additionnent les dimensions entre elles ne sont pas pertinentes [161]. La distance de Hausdorff d_h , définie dans le premier chapitre, évite d'additionner ou soustraire des grandeurs d'unités différentes [163]. Pour comparer deux distances de Hausdorff entre elles, on se ramène à une valeur adimensionnée et définie entre 0 et 1. On définit alors la distance de Hausdorff normalisée \overline{d}_h . Soit $[\mathbf{z}]$ et $[\mathbf{y}]$ deux boîtes incluses dans l'espace de recherche $[\mathbf{x}]^0$ de \mathbb{I}^n , la distance de Hausdorff normalisée entre ces boîtes est donnée par IV.1.

$$\overline{d}_h([\mathbf{z}], [\mathbf{y}]) = \max\left\{\frac{d_h([z_i], [y_i])}{w([x_i]^0)} : 1 \leq i \leq n\right\} \quad (\text{IV.1})$$

Pour faciliter leur interprétation et la comparaison des sous-ensembles de solutions, les indicateurs sont construits autour de notions intuitives (distance moyenne, volume) et définis avec des échelles de valeur entre 0 et 1. Il est aussi souhaitable que les expressions de ces indicateurs soient basées sur des fonctions croissantes avec la diversité. Certains indicateurs existants s'appuient sur le découpage de l'espace en s [37, 104] pour ensuite compter le nombre d'entités par clusters. Bien que très intuitifs, leurs complexités de calcul augmentant rapidement avec le nombre de dimensions, ces indicateurs n'ont donc pas été retenus. La liste suivante synthétise les attendus :

1. Notion intuitive ;
2. Échelle de valeur entre 0 et 1 ;
3. Fonction croissante corrélée à l'augmentation de la diversité ;
4. Temps de calcul raisonnable.

IV.2.2 Uniformité

Un sous-ensemble de solutions est réparti uniformément dans l'espace lorsque les solutions sont équidistantes les unes des autres. On peut évaluer cette grandeur en calculant la distance minimale moyenne entre les solutions. Pour un sous-ensemble de solutions \mathcal{S} de taille p , on définit la distance minimale moyenne D_{min} par l'équation IV.3.

$$D_{min}(\mathcal{S}) = \frac{1}{p} \sum_{i=1}^p \min\{\overline{d}_h(\mathbf{s}_i, \mathbf{s}_j) : 1 \leq j \leq p, j \neq i\} \quad (\text{IV.2})$$

$$D_{min}^*(\mathcal{S}) = p \times D_{min}(\mathcal{S}) \quad (\text{IV.3})$$

La distance minimale moyenne n'est pertinente que pour comparer des sous-ensembles de solutions de même cardinalité. C'est aussi un indicateur qui, utilisé tel quel, décroît au fur et à mesure que la taille du sous-ensemble de solutions augmente. On préfère le substituer par l'expression D_{min}^* (IV.3) dont l'évolution est quasi monotone avec l'augmentation du nombre de solutions et de la diversité. Le calcul de $D_{min}^*(\mathcal{S})$ n'est pas suffisant, on complète cette estimation de l'uniformité par le calcul de la distance moyenne entre les solutions, noté D_{moy} (équation IV.4). D_{moy} est préféré au calcul de l'écart-type par souci d'interprétation des résultats.

$$D_{moy}(\mathcal{S}) = \frac{1}{p} \sum_{i=1}^p \frac{1}{p} \sum_{j=1}^p \overline{d}_h(\mathbf{s}_i, \mathbf{s}_j) \quad (\text{IV.4})$$

Les valeurs de D_{min}^* et D_{moy} sont ramenées entre 0 et 1 à partir de l'évaluation d'un sous-ensemble de solutions de référence \mathcal{S}_{ref} . Comme nous n'avons pas accès à la solution complète pour établir cette référence, \mathcal{S}_{ref} est classiquement obtenue, pour chaque indicateur, avec le sous-ensemble de solutions donnant la plus grande valeur pour un problème donné. Finalement, l'uniformité d'un sous-ensemble \mathcal{S} de solution est évaluée par le couple d'indicateurs noté $(Q_a(\mathcal{S}), Q_d(\mathcal{S}))$ dont les valeurs respectives sont données par les expressions IV.5 et IV.6.

$$Q_a(\mathcal{S}) = \frac{D_{min}^*(\mathcal{S})}{D_{min}^*(\mathcal{S}_{ref})} \quad (\text{IV.5})$$

$$Q_d(\mathcal{S}) = \frac{D_{moy}(\mathcal{S})}{D_{moy}(\mathcal{S}_{ref})} \quad (\text{IV.6})$$

Soit \mathcal{S}_1 et \mathcal{S}_2 , deux sous-ensembles de solutions obtenues par deux stratégies d'exploration partielles différentes. On considère que \mathcal{S}_1 est plus uniforme que \mathcal{S}_2 si $Q_a(\mathcal{S}_1) \geq Q_a(\mathcal{S}_2)$ et $Q_d(\mathcal{S}_1) \geq Q_d(\mathcal{S}_2)$.

IV.2.3 Couverture

La couverture traduit la capacité d'un sous-ensemble de solutions à atteindre la frontière de l'espace réalisable. Pour un CSP numérique, cette frontière est matérialisée par l'enveloppe de l'ensemble complet des solutions. Ainsi, pour comparer la couverture de deux sous-ensembles de solutions, on peut évaluer leurs enveloppes respectives. L'indicateur Q_h repose sur le calcul du volume de l'enveloppe de $V(\mathcal{S})$ (équation IV.7). Pour ce calcul, si un membre $[S_i]$ de $\square\mathcal{S}$ est dégénéré, celui-ci est ramené à un intervalle de largeur ϵ_i , soit $[S_i] \approx [S_i - \epsilon_i/2, S_i + \epsilon_i/2]$. On normalise ensuite cette grandeur à partir de la valeur du volume de l'enveloppe d'une solution de référence \mathcal{S}_{ref} . Q_h est finalement obtenue par l'équation IV.8. La définition de Q_h peut être assimilée à la définition de l'*overall spread* dans [67].

$$V(\mathcal{S}) = v(\square\mathcal{S}) \quad (\text{IV.7})$$

$$Q_h(\mathcal{S}) = \frac{V(\mathcal{S})}{V(\mathcal{S}_{ref})} \quad (\text{IV.8})$$

Soit \mathcal{S}_1 et \mathcal{S}_2 , deux sous-ensembles de solutions obtenues par deux stratégies d'exploration partielles différentes. On considère que \mathcal{S}_1 a une meilleure couverture de l'espace de recherche que \mathcal{S}_2 si $Q_h(\mathcal{S}_1) \geq Q_h(\mathcal{S}_2)$.

IV.2.4 Précisions sur l'utilisation des indicateurs

Les indicateurs ont deux vocations : comparer les performances des algorithmes AB&P ou étudier l'évolution de la diversité pour une exploration particulière. Le premier cas est l'objet de ce chapitre et le second cas sera davantage développé dans le chapitre suivant (V). Les définitions de Q_a , Q_h et Q_d demandent d'identifier une valeur de référence \mathcal{S}_{ref} . Cette référence a surtout pour objectif de ramener les valeurs des indicateurs entre 0 et 1. Selon l'utilisation des indicateurs, \mathcal{S}_{ref} n'est pas déterminée de la même façon. Ainsi, lorsqu'il s'agit de comparer plusieurs sous-ensembles sur un problème donné, \mathcal{S}_{ref} correspond au résultat donnant la meilleure valeur absolue pour l'indicateur considéré. Par exemple, pour Q_h , \mathcal{S}_{ref} est le sous-ensemble maximisant le volume de l'enveloppe $V(S)$. Lorsque l'on regarde uniquement l'évolution d'un indicateur pour une exploration, \mathcal{S}_{ref} est simplement le sous-ensemble formé par l'ensemble des solutions générées.

IV.3 Les stratégies de sélection pour des algorithmes Anytime Branch and Prune

Dans l'état de l'art, nous avons présenté un algorithme AB&P et ses composants principaux : la stratégie, la méthode de bisection et la stratégie de sélection. La stratégie de sélection désigne la zone à explorer après chaque backtrack, c'est le composant d'un algorithme AB&P ayant le plus d'influence sur la diversité du résultat. La fonction de tri ρ_{md} de DMDFS s'appuie sur le calcul de la distance minimale entre les zones restant à explorer et les solutions déjà calculées. Dans cette section, nous cherchons de nouvelles stratégies plus efficaces que DMDFS. En particulier, des stratégies dont la complexité est moins dépendante de la taille du problème et du nombre de solutions à calculer. Dans un second temps, ces stratégies sont adaptées à l'exploration de problèmes réels de conception.

IV.3.1 Stratégie hybride DFS/BestFS générique

L'algorithme 3 est une version générique d'un algorithme AB&P avec une stratégie hybride DFS/BestFS. Un algorithme AB&P parcourt un CSP $\mathcal{P} = \langle X, [\mathbf{x}]^0, \mathcal{C} \rangle$ jusqu'à ce que le critère d'arrêt ϕ soit vérifié où que l'espace des solutions ait été entièrement exploré. Il traite les boîtes non-explorées grâce à deux structures de donnée : une liste L définie comme une pile et un tas M pour maintenir un arbre trié selon un critère ρ . La stratégie de sélection est déterminée par la gestion de ces deux structures. Entre chaque *backtrack*, la liste L permet de parcourir l'arbre de recherche binaire en profondeur jusqu'à atteindre une ϵ -solution ou une boîte vide. On désigne cette phase par *descente DFS*. L'ordre d'insertion de x_1 et x_2 dans L est discuté au paragraphe IV.4.4. La ligne 5 : permet de traiter le cas où la liste L a été entièrement explorée sans trouver de solution. La méthode $\text{Update}_\rho(L, M, S)$ est l'opérateur générique de tri des boîtes $[\mathbf{x}]$ de M selon les valeurs de $\rho([\mathbf{x}])$. C'est généralement l'opération limitante de l'algorithme 3. Le caractère hybride de l'algorithme AB&P est matérialisé par la distinction entre les deux structures L et M .

Algorithme 3 Expression générique d'un algorithme AB&P avec une stratégie de sélection BestFS

Entrées : CSP $\mathcal{P} = \langle X, [\mathbf{x}]^0, \mathcal{C} \rangle$, précision $\epsilon > 0$, fonction ρ , critère d'arrêt ϕ

Sortie : pavage partiel \mathcal{S} de l'ensemble $\mathcal{S}_{\mathcal{P}}$

```

1:  $M \leftarrow \{[\mathbf{x}]^0\}$ ;
2:  $L \leftarrow \emptyset$ ;
3:  $\mathcal{S} \leftarrow \emptyset$ ;
4: tant que  $((M \neq \emptyset) \text{ ou } (L \neq \emptyset))$  et  $(\phi \text{ est faux})$  faire
5:   si  $L = \emptyset$  alors                                     ▷ Sélectionne la prochaine boîte à explorer
6:      $[\mathbf{x}] \leftarrow \text{Extraire}(M)$ ;
7:   sinon
8:      $[\mathbf{x}] \leftarrow \text{Extraire}(L)$ ;
9:   fin si
10:   $\text{Contracter}_{\mathcal{C}}([\mathbf{x}])$ ;                                     ▷ Réduction de la boîte explorée
11:  si  $[\mathbf{x}]$  n'est pas vide alors
12:    si  $w([\mathbf{x}]) \leq \epsilon$  alors
13:       $\mathcal{S} \leftarrow \mathcal{S} \cup \{[\mathbf{x}]\}$ ;                               ▷ Une  $\epsilon$ -solution est trouvée
14:       $\text{Update}_{\rho}(L, M, \mathcal{S})$ ;                                     ▷ Maintien de  $L$  et  $M$ 
15:    sinon
16:       $\{[\mathbf{x}_1], [\mathbf{x}_2]\} \leftarrow \text{Diviser}([\mathbf{x}])$ ;                 ▷ Bisection de la boîte explorée
17:       $L \leftarrow \{[\mathbf{x}_1], [\mathbf{x}_2]\}$ ;
18:    fin si
19:  fin si
20: fin tant que
    
```

La figure IV.2 illustre les mécanismes de l'algorithme en deux dimensions. Les numéros indiquent l'ordre dans lequel les boîtes sont explorées. Les descentes DFS alternent bisections et contractions. Après un backtrack et de deux descentes DFS, 6 boîtes non-explorées sont générées. Le rôle de la stratégie de sélection, et donc celui de Update, est de choisir parmi ces boîtes laquelle doit être considérée pour la prochaine descente.

Selon la définition de la fonction de tri de M dans Update, ρ , on peut distinguer deux types de stratégie DFS/BestFS. Dans le premier cas, le calcul de $\rho([\mathbf{x}])$ dépend de l'état de l'exploration et doit être recalculé dès que cet état change. On désigne ce type de stratégie comme *dynamique*. Le second cas est appelé *statique*, la fonction ρ est indépendante de l'état de l'exploration, sa valeur reste constante. Cette différenciation dynamique/statique est évoquée dans [90]. Update est définie de deux manières selon ces types.

IV.3.1.a Stratégie DFS/BestFS dynamique

Pour une stratégie dynamique, le calcul de $\rho([\mathbf{x}])$ d, par exemple, des autres boîtes non-explorées, du nombre d'itérations, de la qualité du résultat ou du nombre solutions. C'est le cas de la stratégie DMDFS [26], car le calcul de $\rho_{ma}([\mathbf{x}])$ dépend des solutions déjà calculées. À chaque nouvelle solution obtenue, le critère doit être recalculé pour toutes les boîtes de M en plus de celles de L . La méthode Update définie pour les stratégies dynamiques est donnée dans l'algorithme 4. Soit q le coût de l'évaluation $\rho([\mathbf{x}])$, m la taille de la structure M et l la taille de la structure L . La complexité de Update pour une fonction

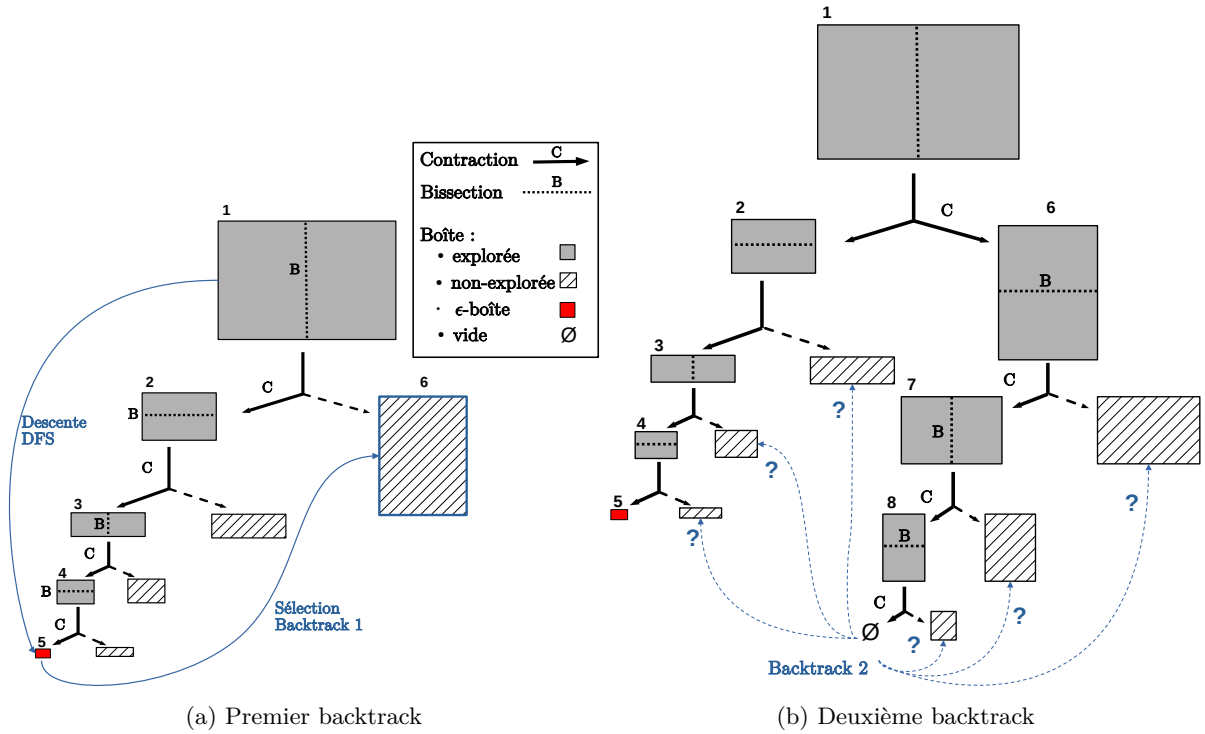


FIGURE IV.2 – Quelle boîte visiter après un backtrack pour maximiser efficacement la diversité?

de tri ρ donnée est alors en $O((m + l)(q + \log(m + l)))$. Pour DMDFS, on peut réduire la complexité en évaluant les boîtes de M à chaque nouvelle solution générée et non à chaque division. Ce qui donne une complexité en $O(n(sl + m) + (m + l)\log(m + l))$, avec s le nombre de solutions déjà obtenues (taille de S) et n le nombre de dimensions du problème. On voit que cette complexité augmente rapidement avec le nombre de solutions calculées (s) et la profondeur de la descente DFS (l).

Algorithme 4 Méthode Update pour une stratégie DFS/BestFS dynamique (algorithme 3)

Entrées : boîtes non-explorées (L, M) , ϵ -boîtes \mathcal{S} , fonction ρ

Sortie : (L, M)

- 1: **pour** $[x] \in M$ **faire**
 - 2: Evaluer $_{\rho}([x], M, \mathcal{S})$;
 - 3: **fin pour**
 - 4: Trier(M);
 - 5: **tant que** $L \neq \emptyset$ **faire**
 - 6: $[x] \leftarrow$ Extraire(L);
 - 7: Evaluer $_{\rho}([x], M, \mathcal{S})$;
 - 8: $M \leftarrow$ Insérer($[x]$);
 - 9: **fin tant que**
-

IV.3.1.b Stratégie DFS/BestFS statique

L'algorithme 5 donne le pseudo-code de la méthode Update pour une stratégie de sélection DFS/BestFS statique. La valeur de $\rho([\mathbf{x}])$ reste constante au cours de l'exploration et les boîtes non-explorées ne sont évaluées qu'une seule fois. En conservant les notations précédentes, la complexité de Update pour une stratégie statique est en $O(l(q + \log(l + m)))$. Son coût est réduit par rapport à une stratégie dynamique et surtout indépendant de la taille des structures de donnée M , L et \mathcal{S} . Nous avons donc privilégié le développement de stratégies de sélection DFS/BestFS statiques. Partant de là, il s'agit de trouver la fonction ρ qui permet d'améliorer l'efficacité par rapport à DMDFS sans trop dégrader la diversité du résultat.

Algorithme 5 Méthode Update pour une stratégie DFS/BestFS statique (algorithme 3)

Entrées : boîtes non-explorées (L, M) , ϵ -boîtes \mathcal{S} , fonction ρ
Sortie : (L, M)

```

1: tant que  $L \neq \emptyset$  faire
2:    $[\mathbf{x}] \leftarrow \text{Extraire}(L)$ ;
3:    $\text{Evaluer}_\rho([\mathbf{x}], M, \mathcal{S})$ ;
4:    $M \leftarrow \text{Insérer}([\mathbf{x}])$ ;
5: fin tant que

```

IV.3.2 Nouvelles stratégies de sélection

Les stratégies de sélection proposées dans les paragraphes suivants sont construites sur le schéma d'un algorithme AB&P avec une stratégie hybride DFS/BestFS statique. Chaque stratégie est définie par une fonction de tri $\rho : \mathbb{I}^n \rightarrow \mathbb{R}$ des boîtes de M .

IV.3.2.a Depth and Largest First Search ou DLFS

L'heuristique *Depth and Largest First Search* (DLFS) explore en priorité les boîtes de plus grand volume. Le principe de DLFS est que des boîtes volumineuses sont davantage susceptibles de contenir des solutions et que ces solutions seront a priori bien différentes des solutions déjà calculées. Cette méthode de sélection de boîtes est évoquée dans les travaux récents de [43]. Pour une boîte $[\mathbf{x}] = ([x_1] \times [x_2] \times \dots \times [x_n])$, on définit $\rho_l : \mathbb{I}^n \rightarrow \mathbb{R}$ la fonction associée à DLFS par :

$$\rho_l([\mathbf{x}]) = v([\mathbf{x}]) \tag{IV.9}$$

La structure M est triée par ordre décroissant des valeurs de $\rho_l([\mathbf{x}])$. En pratique, si un intervalle $[x_i]$ de $[\mathbf{x}]$ est dégénéré, celui est ramené à un intervalle de largeur ϵ_i , la précision pour cette dimension. Dans ce cas $[x_i]$ devient $[x_i - \epsilon_i/2, x_i + \epsilon_i/2]$ pour le calcul de $\rho_l([\mathbf{x}])$. Pour une boîte de dimension n , le calcul de ρ_l a un coût en $O(n)$ et la méthode Update_{ρ_l} correspondante une complexité en $O(l(n + \log(m + l)))$.

IV.3.2.b Depth and Perimeter First Search ou DPFS

Sur le même principe, nous proposons la stratégie *Depth and Perimeter First Search* (DPFS). Celle-ci ordonne les boîtes de M dans un ordre décroissant selon leurs périmètres. DPFS a pour avantage par rapport à DLFS d'être moins impactée par les dimensions dégénérées des boîtes à évaluer. On définit la fonction de tri ρ_p pour DPFS dans l'équation IV.10. La complexité de DPFS est du même ordre de grandeur que DLFS.

$$\rho_p([\mathbf{x}]) = \sum_i^n w([x_i]) \tag{IV.10}$$

IV.3.2.c Interleaved Depth First Search ou IDFS

Interleaved Depth First Search (IDFS) est une heuristique présentée par [114] pour le parcours d'arbres de recherche. L'objectif initial est d'éviter la séquentialité d'une stratégie DFS qui, pour certains problèmes, peut entraîner l'exploration inutile de larges zones sans solutions. IDFS repose sur la recherche parallèle des nœuds de profondeurs identiques dans l'arbre initial. Lors d'un retour en arrière, l'heuristique explore en priorité le nœud non-exploré de plus faible profondeur dans l'arbre. La stratégie correspondante pour un algorithme AB&P alterne donc sélection en largeur et parcours en profondeur. Pour réaliser IDFS, la fonction de tri ρ_i (équation IV.11) renvoie la profondeur du nœud de l'arbre correspondant à la boîte à explorer. La liste M est ensuite ordonnée par ordre croissant des valeurs de $\rho_i([\mathbf{x}])$.

$$\rho_i([\mathbf{x}]) = \text{profondeur}([\mathbf{x}]) \tag{IV.11}$$

La profondeur de chaque boîte est une propriété statique et cette information peut être mémorisée pour un coût constant ($O(1)$). Il en résulte une complexité pour Update_{ρ_i} en $O(l(1 + \log(m + l)))$.

IV.3.2.d Random and Depth First Search ou RDFS

Une heuristique de parcours pseudo-aléatoire de l'arbre de recherche peut être construite sur le même principe. La fonction de tri ρ_r attribue alors à chaque boîte non-explorée un coefficient réel aléatoire entre 0 et 1. *Random and Depth First Search* (RDFS) est introduite essentiellement pour comparer les approches précédentes avec une approche aléatoire. L'opérateur de génération d'un réel pseudo-aléatoire a un coût constant ($\text{rand}()$ en $c++$), ce qui donne une complexité pour Update_{ρ_r} en $O(l(1 + \log(m + l)))$. La figure IV.3 reprend les heuristiques statiques présentées plus haut et explicite le choix de la boîte à explorer après le second backtrack.

IV.3.3 Adimensionnement des critères de tri

Dans le contexte de problèmes de dimensionnement en conception, il existe deux limites aux définitions précédentes des fonctions de tri. Premièrement, Les variables d'un CSP pour un problème de conception sont liées à des grandeurs réelles (masse, distance, énergie, etc.) avec des unités différentes. Or, plusieurs des stratégies de sélection proposées reposent sur la comparaison de ces valeurs. Par exemple, pour DMDFS, le critère de tri est basé sur la distance de Hausdorff dont le résultat est homogène à l'unité

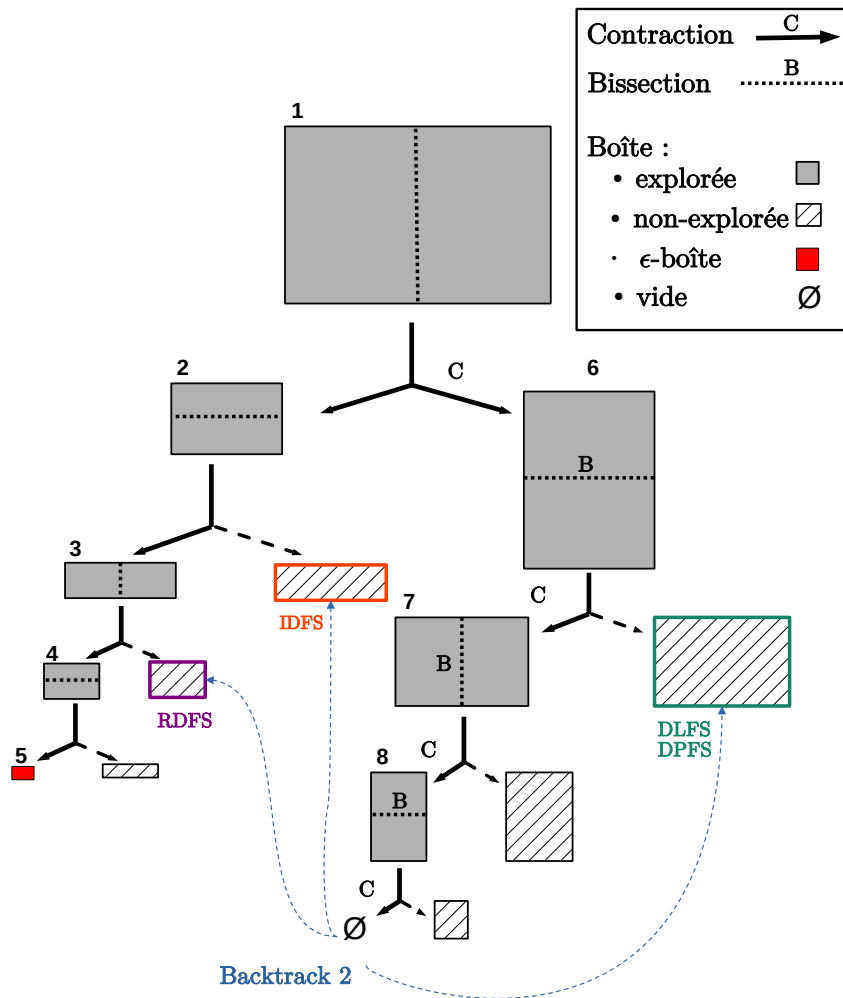


FIGURE IV.3 – Illustration du comportement des stratégies de sélection après deux parcours en profondeur

d'un des domaines de valeurs (cf expression II.6). Entre deux boîtes non-explorées différentes, leurs critères respectifs peuvent ne pas être homogènes entre eux. On peut aussi noter que le calcul du périmètre défini dans DPFS implique d'additionner des valeurs d'unités différentes.

Deuxièmement, l'ordre de grandeur des valeurs peut être très différent d'une variable à une autre. Par exemple, dans le problème *batterie*, la proportion massique du matériau de changement de phase est défini sur l'intervalle $[0.05, 0.16]$ (sans unités) et le nombre de cellules en série sur l'intervalle $[1, 2000]$ (sans unité). Les stratégies comme DMDFS vont avoir tendance à diversifier davantage les solutions dans le sens des domaines les plus grands. Pour éviter de trier des valeurs avec des unités ou des ordres grandeur différents, les domaines de valeurs doivent être adimensionnés pour le calcul de distances, de volumes ou de périmètres. Cette opération est réalisée à partir de la boîte initiale $[\mathbf{x}]^0$ du problème. L'expression normalisée des fonctions de tri de DLFS et DPFS est donnée par l'équation IV.12. Pour DMDFS, on utilise la distance de Hausdorff normalisée \bar{d}_h (équation IV.1) plutôt que la distance de Hausdorff.

$$\bar{\rho}([\mathbf{x}]) = \frac{\rho([\mathbf{x}])}{\rho([\mathbf{x}]^0)} \quad (\text{IV.12})$$

IV.4 Les autres composants impactants d'un algorithme AB&P pour l'exploration partielle

IV.4.1 Choix de la variable à diviser

Après la stratégie de sélection, la stratégie de bisection (Diviser dans l'algorithme 3) est l'opérateur ayant le plus d'influence sur la qualité finale du résultat. Les trois méthodes de bisection introduites dans le chapitre II sont considérées : Round Robin (RR), Smear Sum Relative (SSR) et Largest-First (LF). La méthode du tourniquet RR est conditionnée par l'ordre dans lequel les domaines des variables sont déclarés dans le modèle CSP. Le choix du domaine i à diviser dépend uniquement de la dimension découpée lors de la bisection précédente. SSR s'appuie sur la sensibilité numérique relative des variables du problème [178]. La stratégie de bisection LF découpe en priorité selon la dimension la plus large. En théorie, LF minimise le nombre d'itérations nécessaires pour la descente DFS, en pratique cela dépend davantage de l'efficacité et de la forme de l'espace de recherche. Aussi, le domaine le plus large est celui qui, a priori, a été le moins découpé lors des descentes DFS précédentes. Finalement, LF semble être une approche pertinente pour la recherche de diversité puisqu'un domaine large contient potentiellement beaucoup de valeurs différentes pour la variable correspondante. Cependant, pour les mêmes raisons évoquées dans les paragraphes précédents sur les critères de tri et les indicateurs de qualités, l'expression LF doit être adimensionnée.

IV.4.1.a Adimensionnement de la largeur de la boîte à diviser

Nous proposons deux stratégies de bisection largest-first à partir de deux définitions de la largeur adimensionnées d'une boîte. L'objectif est d'éviter de diviser trop intensément les domaines les plus larges en valeur absolue et de comparer des valeurs d'unités différentes. Deux expressions différentes de la largeur d'un intervalle sont considérées. Soit $[x]$ un composant d'une boîte à bissecter et $[x]^0 =$

$[\underline{x}^0, \bar{x}^0]$ son domaine initial. La première expression, $w_\epsilon([x])$, s'exprime à partir du milieu de la boîte initiale et la seconde $w_0([x])$ est construite à partir de la largeur de $[x]^0$. On note respectivement LF_E et LF_I les méthode de bisection correspondant à chaque expression de la largeur. LF_E en particulier, est directement lié à la nouvelle définition de la précision introduite dans les paragraphes suivants IV.4.3.b.

$$w_\epsilon([x]) = \frac{2w([x])}{|\underline{x}^0| + |\bar{x}^0|} \qquad w_0([x]) = \frac{w([x])}{w([x]^0)} \qquad (\text{IV.13})$$

IV.4.2 Distinction des types de variables

Nous avons vu précédemment que les variables des problèmes de conception peuvent être réparties en trois catégories : conception, performance et auxiliaire (section II.1.5). En conception préliminaire, le concepteur recherche de la diversité sur les variables de conception pour évaluer le plus d'architectures réalisables différentes. Lorsque que le problème de conception comprend plusieurs critères de performances, le concepteur peut vouloir aussi de la diversité sur les valeurs de ces variables pour considérer l'ensemble des compromis possible. On distingue ainsi la diversité des solutions dans l'espace de conception et dans l'espace des objectifs. Cette différenciation des variables peut être prise en compte lors de l'exploration de l'espace de recherche. En optimisation multi-objectif, la diversité est principalement recherchée dans l'espace des objectifs pour bien répartir les solutions au plus proche du front de Pareto.

En exploration partielle, on peut envisager de la diversité dans les deux espaces à la fois. Il suffit de limiter les opérateurs ayant un impact sur la diversité du résultat final aux variables de conception et de performance. Ce travail a déjà été réalisé en particulier pour les stratégies de bisection dans un contexte différent à l'exploration partielle [25]. L'objectif de ces travaux était de privilégier la bisection sur les variables de conception au début de la descente en profondeur. On peut facilement étendre cette approche en donnant la priorité à la fois aux variables de conception et aux variables de performances. Pour les stratégies de sélection de type BestFS, cela revient à retirer les variables auxiliaires du calcul du critère de choix des boîtes non-explorées.

Algorithme 6 Algorithme de priorisation des variables à bissecter

Entrées : boîte $[\mathbf{x}]$, liste d'entiers N

Sortie : dimension i

- 1: $[\mathbf{x}]_a \leftarrow \{[x_i] \in [\mathbf{x}], i \in N\}$;
 - 2: $[\mathbf{x}]_b \leftarrow \{[x_i] \in [\mathbf{x}], i \notin N\}$;
 - 3: **si** $\exists i \in N, w([x_i]) \geq \epsilon$ **alors**
 - 4: $i \leftarrow SelectionDimension([\mathbf{x}]_a)$; ▷ Choix de la dimension à bissecter en priorité
 - 5: **sinon**
 - 6: $i \leftarrow SelectionDimension([\mathbf{x}]_b)$;
 - 7: **fin si**
-

On note $[\mathbf{x}]_{vco}$, la boîte composée uniquement des intervalles de $[\mathbf{x}]$ associées aux V_{co} et aux V_p . Pour une fonction de tri ρ et une boîte à évaluer $[\mathbf{x}]$, lors du calcul de $\rho[\mathbf{x}]$ dans l'algorithme 3 on substitue $[\mathbf{x}]$ par $[\mathbf{x}]_{vco}$. De la même manière, pour la fonction β d'un bissecteur, on détermine la dimension i à diviser en

évaluant $\beta([\mathbf{x}]_{vco})$ tant que ses dimensions sont bissectables. En pratique, il est possible d'aller plus loin en laissant le concepteur choisir un sous-ensemble arbitraire de variables $N \leq n$ au sein desquelles il souhaite diversifier efficacement ses solutions. Dans cas, $[\mathbf{x}]_a$ est la boîte composée de la collection d'intervalles d'indices i de $[\mathbf{x}]$ et $[\mathbf{x}]_b$ son complémentaire dans $[\mathbf{x}]$. Le mécanisme correspondant à cette discrimination pour la bissection des dimensions est donné dans l'algorithme 6. Pour faciliter la réutilisation de modèles CSP en Minibex, nous avons enrichi le langage pour définir les variables pour la fonction de sélection et pour la fonction de bissection. Le script suivant donne un exemple à partir du problème batterie du chapitre III où l'on divise en priorité les domaines des variables de conception C_c , N_s , N_p , τ_{mcp} et q_{cool} . Dans cet exemple on néglige la diversité dans l'espace des V_p (donné par J_{batt} , M_{batt} , V_{batt} et GWP) :

Définition des variables pour les stratégies de sélection et de bissection dans Minibex

```

variables
Cc in [0,300];           // capacit cellule (Vco)
Ns in [1,2000];         // nombre cellules en s ries (Vco)
Np in [1,2000];         // nombre cellules en parall les (Vco)
t_mpc in [0,0.15];     // quantit de mat riau de refroidissement (Vco)
q_cool in [0,6000];    // puissance de refroidissement actif (Vco)
Jbatt in [0,10000];    // prix batterie (Vp)
Mbatt in [0,600];     // masse batterie (Vp)
Vbatt in [0,0.4];     // volume batterie (Vp)
GWP in [0,10000];     // impact GWP batterie (Vp)
/*
// VARIABLES SELECTION
diversity
Cc,Ns,Np,t_mpc,q_cool
// VARIABLES BISSECTION
bisection
Cc,Ns,Np,t_mpc,q_cool
*/

```

IV.4.3 Gestion de la précision des variables

IV.4.3.a Limites rencontrées par les définitions classiques

Pour les problèmes de conception, la précision doit être considérée différemment selon que l'on considère une variable de conception, une variable de performance ou une variable auxiliaire [25]. Idéalement, seule la précision des variables de conception doit être définie et celle des autres variables devrait être seulement déduite à partir de leurs expressions. Dans certains, cas, il peut être nécessaire de définir la précision des variables de performance et auxiliaires. Cela peut, par exemple, permettre d'accélérer la convergence même si ce phénomène est difficilement prévisible. Les définitions classiques de la précision (paragraphe II.3.2.d) présentent certaines limites.

- Une précision absolue nécessite de définir une valeur pour chaque variable. La précision devient un paramètre de résolution à part entière. Ce travail est complexe et peut créer des déséquilibres pendant l'exploration s'il est mal réalisé. Les dimensions avec une trop grande précision sont trop intensément explorées, et à l'inverse, une précision trop faible diminue la confiance dans le résultat.

- Une précision relative entraîne, pour certains domaines de valeurs, une répartition et une taille inégales des ϵ -boîtes dans l'espace de recherche. Par exemple, sur la figure IV.4 on a résolu partiellement le problème du *cerceau* (II.13) avec $(x, y) \in [0, 4] \times [0, 4]$ pour une précision relative de 10^{-1} . Plus les solutions sont proches de 0 dans une dimension, plus la taille des ϵ -boîtes est faible selon cette dimension. Ce phénomène a aussi pour conséquence d'augmenter le nombre d'itérations nécessaire à l'exploration des zones de l'espace de recherche où la largeur requise pour les ϵ -boîtes est la plus faible. Pour certaines stratégies, on peut aussi observer une densification dans les zones de plus faible valeur absolue.

Finalement, une précision non adaptée aux domaines du problème impacte directement la qualité de l'exploration partielle du problème. En particulier pour les domaines des variables dont on cherche à diversifier les valeurs.

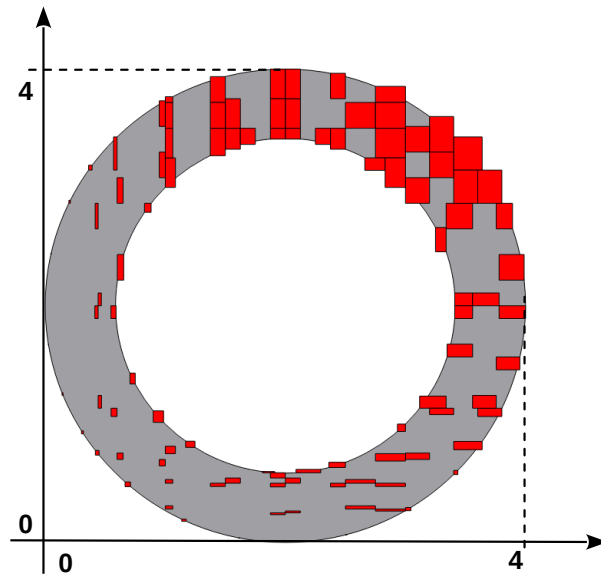


FIGURE IV.4 – Résolution du problème *cerceau* (II.13) défini en $[0, 4] \times [0, 4]$ avec une précision relative de 10^{-1}

IV.4.3.b Proposition de définition de la précision

Face à ces limites, nous proposons une définition de la précision plus adaptée à l'exploration partielle de CSP numériques en conception. Nous proposons d'utiliser une valeur absolue fixe de la précision construite à partir du domaine de validité de la variable pour laquelle elle est définie. Pour une variable x dont le domaine initial est définie par l'intervalle $[x] = [\underline{x}^0, \bar{x}^0]$, un intervalle $[x]$ pour x est considéré comme précis à ϵ près lorsque $w([x])$ vérifie la relation IV.14.

$$w([x]) \leq \epsilon \frac{(|\underline{x}^0| + |\bar{x}^0|)}{2} \tag{IV.14}$$

Cette définition de la précision implique que les bornes des domaines de chaque variable de conception

soient correctement définies en amont de la résolution. En pratique, il peut être intéressant de contracter la boîte initiale du problème une première fois avant de définir les précisions de chaque variable. On utilisera systématiquement cette expression pour les V_{co} . Contrairement à la précision relative, l'expression IV.14 donne une précision de valeur constante, quelle que soit la zone du domaine considérée. En somme, cette définition est équivalente à une précision absolue bien choisie.

IV.4.3.c Précision des variables auxiliaires

Pour les variables dont la valeur est déduite des V_{co} , leur précision peut être grossière pour éviter de générer trop de solutions une fois que les V_{co} ont atteint leurs précisions. C'est le cas pour les variables auxiliaires et les V_p qui ne sont pas des V_{co} .

IV.4.4 Méthode de descente DFS

La méthode de descente DFS désigne l'approche utilisée pour traiter les boîtes issues de la division lors de la phase de descente en profondeur. À chaque bisection, les deux boîtes filles, que l'on différencie ici par les termes *boîte droite* et *boîte gauche*, sont insérées dans la liste maintenue comme une pile. L'ordre d'insertion de ces boîtes dans la liste L est important puisque c'est la boîte insérée en dernier qui sera extraite de L à l'itération suivante. Par défaut, dans un algorithme B&P, l'ordre d'insertion reste le même, par exemple boîte droite puis boîte gauche. Si l'on conserve ce comportement pendant toute l'exploration partielle, le résultat sera déséquilibré vers la "gauche" sur chaque dimension du problème.

Des idées simples permettent d'éviter ce phénomène. Par exemple, alterner entre une descente vers la droite puis une descente vers la gauche. Le changement de direction se fait à chaque nouvelle solution ou lorsqu'une boîte vide est atteinte. Cette méthode d'alternance droite/gauche est désignée ici par *left/right* et est la méthode de descente appliquée par défaut pour nos expérimentations. La figure IV.5 montre l'exploration du problème *cerceau* avec DMDFS et deux méthodes de descente différentes. La méthode de descente "droite" (figure IV.5a) montre des difficultés à atteindre la frontière en bas à gauche de l'espace de recherche (hachurée en bleu). La méthode de descente *left/right* permet de répondre à ce problème (figure IV.5b).

IV.4.5 Gestion des CSP mixtes

IV.4.5.a Variables entières et binaires

Les algorithmes et méthodes décrites précédemment sont développées pour des CSP numériques. Nous pouvons étendre leurs mécanismes pour la résolution de CSP mixtes grâce à deux opérateurs de contraction. Le premier est un contracteur pour les variables entières, il est déjà implémenté dans la bibliothèque IBEX. Cet opérateur réduit un intervalle $[x]$ au plus grand intervalle aux bornes entières $[a, b]$, $(a, b) \in \mathbb{N}^2$ inclut dans $[x]$. Les variables entières et binaires sont gérées à l'aide de cet opérateur.

IV.4.5.b Contrainte de table

Les domaines sous forme de tuples et les tableaux de valeurs sont fréquemment rencontrés dans les problèmes de conception, notamment pour modéliser des choix de composants ou de matériaux. Ces deux

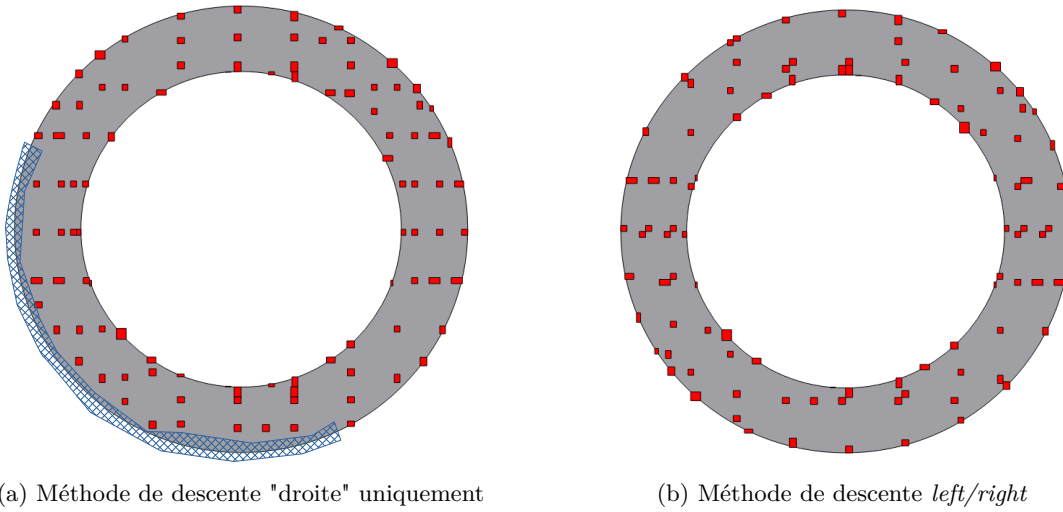


FIGURE IV.5 – Exploration du problème du *cerceau* avec deux méthodes de descente différentes (stratégie DMDFS/RR)

éléments peuvent être modélisés à partir d'une contrainte de table. Une contrainte de table correspond à une matrice des valeurs réalisables de la forme de la matrice IV.15. Chaque ligne i correspond aux valeurs réalisables de la variable x_i et chaque colonne à une instantiation des variables réalisable. Le domaine initial de la variable x_i est l'enveloppe de $\{a_{i1}, a_{i2}, \dots, a_{im}\}$. Soit $[\mathbf{x}]$ une boîte à contracter, le contracteur associé à la table fonctionne en deux étapes :

$$\begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{matrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{km} \end{pmatrix} \quad (\text{IV.15})$$

1. Pour chaque colonne j on vérifie la consistance des a_{ij} avec les domaines $[x_i]$.
2. Pour chaque variable x_i , on calcule l'enveloppe des a_{ij} pour les colonnes j consistantes et on intersecte cette enveloppe avec le domaine de x_i .

Pour illustrer le traitement des variables discrètes, on considère le problème de conception d'une batterie pour voiture électrique formalisé au chapitre III. Le modèle contient deux contraintes de tables pour représenter le choix de la cellule et le choix du système de refroidissement. Le tableau IV.16 donne la matrice correspondant aux paramètres du système de refroidissement. Nous avons ajouté la possibilité d'intégrer directement une contrainte de table dans le langage Minibex, la sémantique est présentée dans le script ci-dessous.

Contrainte de table et script correspondant avec le langage Minibex enrichi

```

variables
q_cool in [0,6000];
M_cool in [0,7];
V_cool in [0,2.8];
J_cool in [0,200];

/* Active cooling system
table
q_cool, M_cool, V_cool, J_cool
0,0,0,0
500,2,0.8,30
1000,3,1.2,80
3000,5,2,120
6000,7,2.8,200
end
*/

```

$$\begin{matrix}
 q_{cool} \\
 M_{cool} \\
 V_{cool} \\
 J_{cool}
 \end{matrix}
 \begin{pmatrix}
 O & 500 & 1000 & 3000 & 6000 \\
 O & 2 & 3 & 5 & 7 \\
 O & 0.8 & 1.2 & 2 & 2.8 \\
 O & 30 & 80 & 120 & 200
 \end{pmatrix}
 \quad (IV.16)$$

IV.4.6 Backtrack forcé

Les opérateurs de contraction ne sont pas toujours efficaces pour réduire le domaine de recherche. Par exemple lorsque les expressions des contraintes contiennent plusieurs fois la même variable ou lorsque beaucoup de variables auxiliaires interviennent dans la modélisation. Dans ce cas, l’algorithme d’exploration risque d’explorer beaucoup de zones non réalisables qui n’ont pas été réduites efficacement. Or, selon le principe initial d’un AB&P avec une stratégie DFS/BestFS, l’exploration ne change pas de zone à explorer tant qu’une solution n’a pas été trouvée. Pour réduire ce phénomène, on peut forcer l’algorithme à explorer une nouvelle zone après plusieurs échecs successifs lors des descentes DFS. Dans l’algorithme 3, si l’exploration d’une zone ne donne rien après 100 descentes en profondeur, on sélectionne une nouvelle de boîte b de M à explorer, on vide la liste L dans M , puis on réinitialise L avec b . Ce *backtrack forcé* n’est utilisé que pour les problèmes les plus délicats à résoudre. Le paramètre associé au backtrack forcé est le nombre d’échecs successifs autorisé, noté χ .

IV.5 Comparaison des stratégies de sélection AB&P : exemples simples

La figure IV.6 reprend les éléments d’un algorithme AB&P et les différents choix d’opérateurs possibles introduits dans ce chapitre. On représente aussi leurs impacts potentiels sur les caractéristiques du résultat en termes de diversité, efficacité (temps de calcul) et de faisabilité. Cette section et la section suivante présentent des expérimentations vouées à identifier les opérateurs les plus pertinents pour l’exploration partielle de problèmes de conception. En particulier sur le choix des mécanismes de sélection et de bisection.

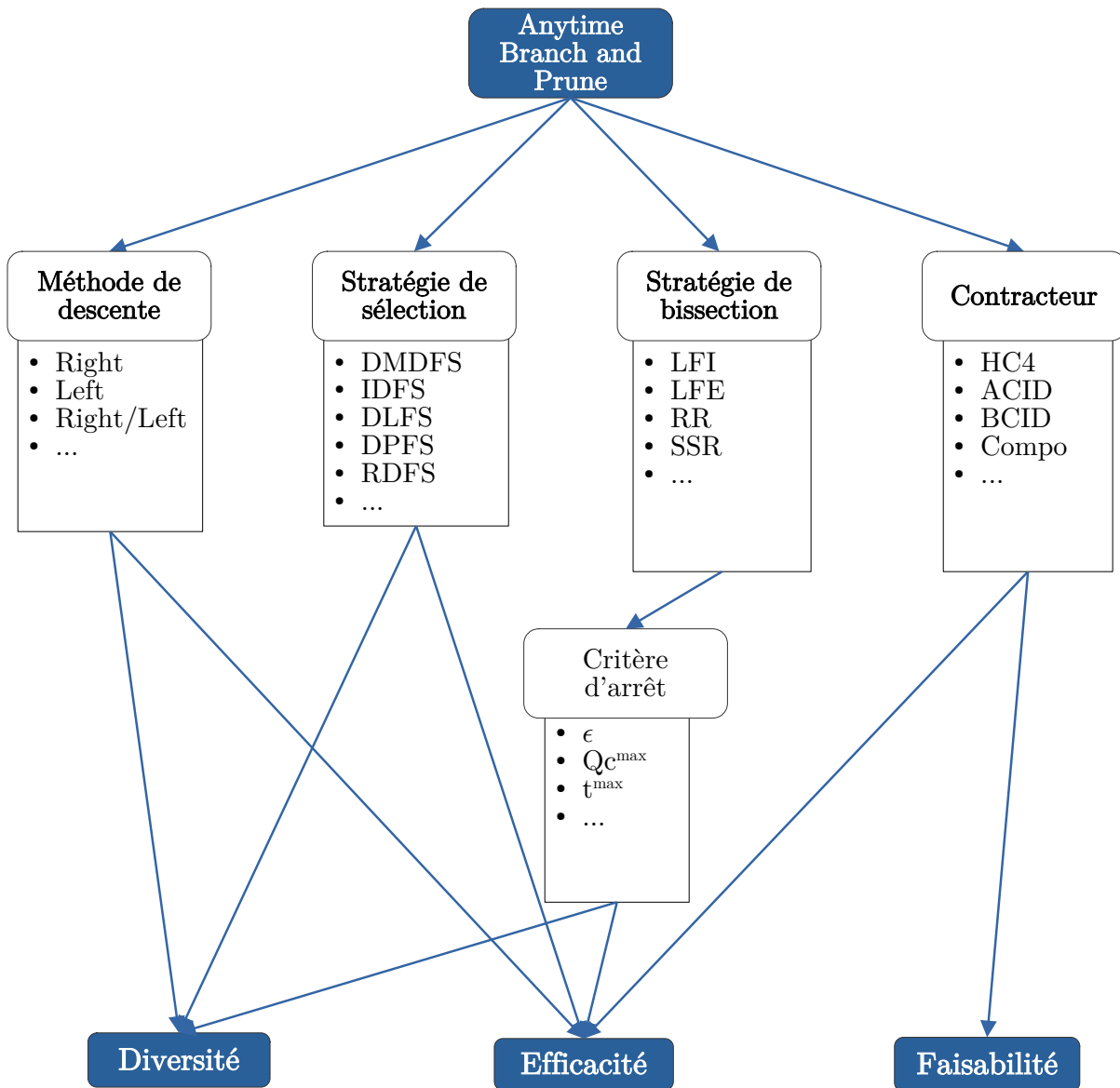


FIGURE IV.6 – Présentation des éléments composant un algorithme AB&P

IV.5.1 Problème des m boules : modèle numérique

Les stratégies de sélection sont comparées à partir du problème des m boules. Il s'agit d'un espace de recherche composé de m sphères disjointes de rayon 1 réparties aléatoirement dans un espace clos de n dimensions. Ce problème a été décrit initialement dans [26]. Chaque boule du problème (appelé ici problème des m boules) forme un espace continu de solutions. Ce problème est évolutif sur le nombre de dimensions et le nombre de boules. Plusieurs formes du problème seront résolues successivement pour évaluer la capacité des stratégies de sélection à placer au moins une solution dans chacun des sous-espaces continus de solutions. Le problème peut prendre la forme du CSP numérique IV.17.

$$X = (x_1, \dots, x_n, y_1, \dots, y_m) \tag{IV.17}$$

avec :

$$\begin{aligned} x_i &\in [-100, 100], \quad i \in (1, \dots, n) \\ y_j &\in [-1, 200\sqrt{n} - 1], \quad j \in (1, \dots, m) \\ \sum_{i=1}^n (x_i - c_{ji})^2 &= (1 + y_j)^2, \quad j \in (1, \dots, m) \end{aligned} \tag{IV.18}$$

$$\prod_{j=1}^m y_j \leq 0 \tag{IV.19}$$

Les coordonnées des centres des boules c_{ji} sont déterminées aléatoirement entre -100 et 100 . $1 + y_i$ est la distance euclidienne entre la solution et la boule j , la borne supérieure de y_j est obtenue à partir de $200\sqrt{n}$, la distance euclidienne maximale dans $[-100, 100]^n$. Il y a plusieurs raisons de s'intéresser à ce problème pour étudier nos stratégies d'exploration partielle :

- Le nombre de variables peut être ajusté simplement pour un passage à l'échelle ;
- La solution idéale d'une exploration partielle est connue : un sous-ensemble de m solutions parfaitement réparties dans chacune des m boules², soit 1 solution par boule ;
- Les caractéristiques du problème peuvent être changées simplement (rayon des boules, bornes des domaines, types des variables)
- Le problème peut être assimilé à un problème de conception dont les x_i sont les variables de conception et les y_i des variables auxiliaires qui permettent la modélisation. On limite la recherche de diversité aux variables x_i .

IV.5.1.a n boules en n dimensions

On considère les cas où le nombre de boules est égal au nombre de dimensions ($n = m$) avec n compris entre 2 et 15. Les 14 problèmes sont résolus par les 5 stratégies DMDFS, DLFS, DPFS, IDFS et RDFS dans des conditions identiques. Pour chaque algorithme AB&P, on utilise le contracteur HC_4 , la stratégie de bisection RR et la méthode de descente *left/right*. Le critère d'arrêt est fixé à $Q_c^{max} = 100$

² lorsque certaines boules ne sont pas parfaitement disjointes, le nombre idéal de solutions peut être légèrement inférieur à m

solutions et la précision des variables x_i à $\epsilon = 10^{-6}$ en valeur relative aux domaines. Pour ces problèmes particuliers, le choix de la dimension à bissecter n'a pas d'impact sur l'exploration. Lorsqu'une solution est obtenue, on peut simplement déterminer la boule "touchée" en identifiant la variable y_j dont la valeur est négative. La précision des variables y_i est fixée à 10^{-1} en valeur absolue. Pour une valeur donnée de n , la répartition aléatoire des boules est conservée d'une résolution à une autre. Les résultats sont donnés dans le tableau IV.1.

TABLE IV.1 – Résultat de l'exploration partielle du problème numérique des m boules ($n=m$) pour différentes valeurs de m .

	DMDFS	DPFS	DLFS	IDFS	RDFS
n	s_{min}	s_{min}	s_{min}	s_{min}	s_{min}
2	2	2	2	2	2
3	3	3	3	3	(1)
4	4	4	4	6	(1)
5	6	6	6	9	(2)
6	6	6	6	56	(1)
7	7	7	7	8	(1)
8	8	8	8	15 (7)	(1)
9	9	9	9	17	(4)
10	10	10	10	19 (9)	(2)
11	11	11	11	92	(2)
12	12	12	12	24 (11)	(1)
13	13	13	13	74	(1)
14	15	14	14	42	(1)
15	16	16	16	43 (14)	(2)

Dans le tableau IV.1, s_{min} est le nombre de solutions nécessaire pour atteindre les n boules. Les valeur entre parenthèse donne le nombre de boules touchées si n n'a pas été atteint. Pour RDFS, s_{min}^* donne uniquement le nombre de boules atteintes, car, en dehors du premier problème ($n = 2$), toutes les boules n'ont pas été atteintes après 100 solutions. Globalement, DMDFS, DPFS, et DLFS permettent d'atteindre les n boules en pratiquement autant de solutions, au plus en $n + 1$ solutions. Les résultats de IDFS dépendant de la disposition initiale des boules dans l'espace de recherche. Plusieurs répartitions aléatoires des boules ont été testées en amont, la plupart du temps IDFS n'atteint pas les n boules en moins de 100 solutions. Pour la répartition proposée ici, dans 4 cas, la dernière boule n'a pas été trouvée par IDFS. La recherche aléatoire RDFS ne permet pas de diversifier suffisamment les solutions pour atteindre plus de 2 boules en moins de 100 solutions.

IV.5.1.b 100 boules en n dimensions

On considère maintenant 100 boules de rayon 1 réparties dans des espaces de dimension 2 à 15. Les conditions d'exploration sont conservées (*HC4*, *LF*, *left/right*) avec les mêmes critères d'arrêt $Q_c^{max} = 100$ et $\epsilon = 10^{-6}$. Les résultats des explorations partielles sont donnés dans le tableau IV.2. En dehors des temps de calcul, DMDFS, DLFS et DPFS sont les stratégies les plus efficaces pour atteindre les 100 boules disjointes dans l'espace de recherche. DMDFS est légèrement plus performante dans ces conditions d'exploration. En revanche, DMDFS est en moyenne 40% plus lente que les trois autres stratégies pour

générer un sous-ensemble de 100 solutions. Dans cet exemple, les stratégies statiques permettent bien d'améliorer la vitesse de convergence sans trop dégrader la diversité des solutions.

TABLE IV.2 – Nombre de boules et temps de calcul moyen pour le problème des 100 boules ($n \in (2, \dots, 15)$)

	DMDFS	DPFS	DLFS	IDFS	RDFS
Nombre de boules	91,8	91,0	91,0	68,4	1,4
Temps (s)	2,8275	2,0503	2,0490	2,0873	0,3977

On constate que RDFS converge rapidement, mais reste très peu efficace pour générer de la diversité sur ce problème particulier. On peut tenter d'expliquer ce phénomène. La première descente en profondeur génère beaucoup de boîtes non explorées et beaucoup d'entre elles ne se dirigent que vers la boule déjà explorée. Par exemple, en dimension 2 pour la précision demandée (10^{-6}), la première descente génère 24 boîtes non explorées, dont 17 autour de la boule déjà atteinte. Ensuite, à chaque nouvelle descente, la proportion de ces boîtes augmente parmi l'ensemble des boîtes non explorées. Ainsi, plus l'exploration avance, plus la probabilité d'explorer une nouvelle boule diminue.

IV.5.1.c Profils de performance et comportement des stratégies de sélection

Dans un second temps, on continue l'exploration jusqu'à 300 solutions pour étudier l'évolution des indicateurs de qualités Q_a et Q_h . Pour chaque taille de problème n et chaque indicateur, on détermine les ensembles de référence \mathcal{S}_{ref} à partir d'une répartition hypothétique parfaite de 100 solutions dans les 100 boules. Le profil de performance de chaque stratégie est obtenu en traçant la moyenne sur l'ensemble des 14 problèmes du benchmark (figure IV.7). La répartition idéale correspond à $Q_c = 100$, $Q_a = 1$ et $Q_h = 1$.

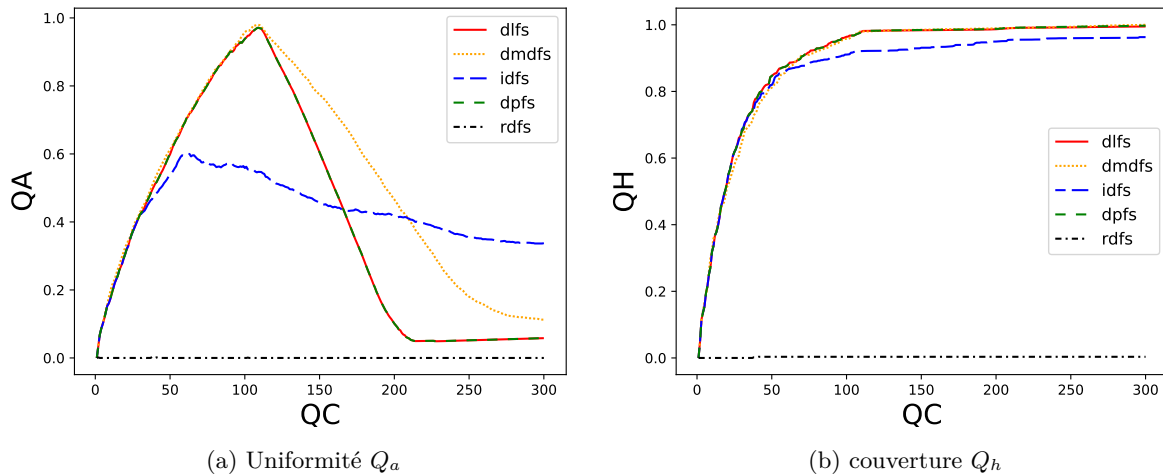


FIGURE IV.7 – Profils de performance de Q_a et Q_h pour chaque stratégie sur le problème des 300 boules ($2 \leq n \leq 15$).

DPFS et DLFS ont le même comportement, c'est le fait de l'absence de dimensions dégénérées dans l'exploration. On retrouve les performances relatives de l'analyse précédente. DMDFS, DPFS et DLFS

atteignent rapidement les 100 zones disjointes, IDFS est moins efficace et RDFS n'est pas adapté. Les profils de performances associés à Q_a et Q_h pendant la totalité de l'exploration peuvent être décrit en deux parties. La première est la phase de *diversification* pendant laquelle on cherche à atteindre chaque zone disjointe de l'espace (Q_a et Q_h augmente). Puis, une fois atteinte la 100^{ième} boule du problème, l'algorithme repasse par les zones déjà explorées, c'est la phase d'*intensification* (Q_a diminue et Q_h atteint sa valeur maximale). On peut expliquer la divergence de comportement des stratégies lors de la phase d'intensification. Le choix des boîtes à explorer avec et DPFS ne dépend pas de la disposition des boules. L'intensification de l'exploration dans les zones déjà explorées est faite de manière régulière. En revanche, l'heuristique de DMDFS n'est plus efficace une fois toutes les zones atteintes. L'exploration ne dépend plus de la disposition des boules dans l'espace et certaines sont visitées plus que d'autres. Ce résultat est confirmé en mesurant l'écart type à la moyenne du nombre de solutions par boules dont la valeur idéale (tableau IV.3). La stratégie IDFS alterne intensification et diversification, cette difficulté est le fait de la répartition aléatoire des boules et du nombre important de zones sans solutions dans l'espace de recherche.

 TABLE IV.3 – Ecart type du nombre de solutions par boules, pour 300 ϵ -boîtes réparties dans 100 boules

	DMDFS	DPFS	DLFS	IDFS	RDFS
Ecart Type	1,69	0,43	0,43	2,98	28,78

IV.5.2 Problème des m boules : modèle mixte

Pour illustrer la capacité des algorithmes AB&P à traiter des problèmes mixtes, le problème des m boules peut être formulé sous la forme IV.20 :

$$X = (x_1, \dots, x_n, y_1, \dots, y_m) \quad (\text{IV.20})$$

subject to :

$$x_i \in [-100, 100], \quad i \in (1, \dots, n)$$

$$y_j \in \{0, 1\}, \quad j \in (1, \dots, m)$$

$$y_j \sum_{i=1}^n (x_i - c_{ji})^2 \leq 1, \quad j \in (1, \dots, m) \quad (\text{IV.21})$$

$$\sum_j y_j = 1 \quad (\text{IV.22})$$

Les y_i sont ici des variables binaires, traitées comme des entiers. La boule touchée par une solution est identifiée par l'indice i pour lequel $y_i = 1$. On réalise les mêmes tests que pour la forme numérique du problème avec les mêmes répartitions de boules dans l'espace. Pour le cas des n boules en n dimensions, on retrouve les mêmes résultats que dans le tableau IV.1.

Les résultats des deux expériences réalisées avec les problèmes avec 100 boules en n dimensions (exploration jusqu'à 100 solutions puis jusqu'à 300 solutions) sont donnés en annexe C.1. On observe les

mêmes comportements que pour le cas numérique. Les performances sont légèrement meilleures pour les trois stratégies DMDFS, DPFS et DLFS mais elles sont dégradées pour IDFS. Les temps de calculs sont améliorés par rapport aux modélisations en CSP numériques. Il est en effet plus rapide de converger sur les domaines entiers $\{0, 1\}$ des y_i que sur les domaines réels.

IV.6 Comparaison des stratégies ABPA : benchmark de problèmes de conception

L'objectif est d'identifier la méthode de recherche la plus prometteuse pour explorer un CSP mixte ou numérique issu d'un problème de conception. Dans le cas du problème des m boules, seule la stratégie de sélection des zones non-explorées a un impact sur la diversité du résultat, le choix de la stratégie de bisection n'a pas d'impact significatif. Pour les problèmes complexes de conception, les domaines de validité sont d'ordres de grandeur différents et les variables ne sont pas soumises aux mêmes types de contraintes. Par conséquent, l'opérateur de bisection impacte fortement sur le résultat final. Pour avoir une réponse complète à la question de recherche, nous devons identifier le couple de stratégies (sélection ; bisection) le plus efficace selon nos indicateurs de qualité. On considère les quatre stratégies de sélection : DMDFS, DLFS, DPFS et IDFS et les quatre stratégies bisection : SSR, RR, LF_E et LF_I .

IV.6.1 Protocole

Les algorithmes sont testés sur un ensemble de 30 problèmes de conception issus de la littérature ou de banques de problèmes MINLP comme *gamslib*³. Les problèmes d'optimisation sont convertis en problèmes d'exploration en transformant l'objectif en une contrainte d'inégalité. La recherche de diversité est limitée à l'espace de conception. Pour chaque problème, on identifie au préalable le contracteur le plus efficace entre *HC4*, *ACID* ou une composition des deux *HC4/ACID*. Les opérateurs basés sur la linéarisation se sont avérées moins efficaces. La méthode de descente utilisée est *left/right*. Le critère d'arrêt principal est le nombre de solutions calculées $Q_c^{max} = 1000$ avec une précision de $\epsilon = 10^{-5}$ en utilisant la définition IV.14. Le temps de résolution est en plus limité à 1200 secondes. Le *backtrack forcé* est appliqué pour 6 problèmes en fixant le nombre d'échecs successibles acceptable à $\chi = 100$. Le tableau donné en annexe C.3 résume, pour chaque problème, ses caractéristiques, les conditions de résolution utilisées et la source. Certains des modèles sont disponibles en annexe.

Les tests sont réalisés sur une machine Linux Core i7-9850H 2.6 GHz (16 GB). Les algorithmes sont implémentés en C++ avec la librairie IBEX⁴. Les modèles ont été implémentés en langage Minibex dont la sémantique est enrichie pour intégrer les contraintes de tables et identifier les variables de conception pour la recherche de diversité. Dans la mesure du possible, le nombre de variables auxiliaires est réduit au minimum pour chaque problème. Un exemple de modélisation avec Minibex est donné en annexe B.3 à partir du problème de batterie décrit dans le III.

3. https://www.gams.com/latest/gamslib_ml/libhtml/index.html#gamslib

4. <http://www.ibex-lib.org/>

L'évaluation des indicateurs de qualités est réalisée a posteriori de chaque résolution. Pour chaque problème et pour chaque indicateur, la solution de référence \mathcal{S}_{ref} utilisée dans le calcul de IV.5, IV.8 et IV.6 est donnée par le meilleur résultat. En plus trois des indicateurs de qualités, on évalue aussi leur somme Σ (équation IV.23). Finalement, pour équilibrer le compromis entre uniformité (Q_a et Q_d) et couverture (Q_h), on considère aussi la somme pondérée Σ^P (IV.24).

$$\Sigma(\mathcal{S}) = Q_a(\mathcal{S}) + Q_h(\mathcal{S}) + Q_d(\mathcal{S}) \quad (\text{IV.23})$$

$$\Sigma^P(\mathcal{S}) = 2Q_a(\mathcal{S}) + 3Q_h(\mathcal{S}) + Q_d(\mathcal{S}) \quad (\text{IV.24})$$

IV.6.2 Résultats globaux

Au total, 600 résolutions partielles sont réalisées. Le problème *transfo_secu* n'a pas pu être résolu en utilisant l'opérateur de bisection RR, les résultats de ce problème sont ignorés pour cette analyse globale. Dans 7 cas, la résolution n'a pas pu atteindre les 1000 solutions de l'exploration partielle en moins de 1200 secondes. En dehors de ces cas, les résolutions mettent en moyenne moins de 60 secondes pour calculer les 1000 solutions.

IV.6.2.a Valeurs de référence

TABLE IV.4 – Nombre de fois où les stratégies de sélection et de bisection ont donné le meilleur résultat pour un problème donné parmi les 30 résolutions

	Q_a	Q_h	Q_d	Σ	Σ^P
DMDFS	14	18	10	13	15
IDFS	2	28	14	4	4
DLFS	7	18	0	6	4
DPFS	7	18	6	6	7
LF_I	14	20	6	11	15
LF_E	8	21	9	6	6
RR	6	13	0	6	4
SSR	2	28	15	6	5

Le tableau IV.4 présente pour chaque composant, le nombre de résolutions pour lesquelles son utilisation a donné le meilleur résultat (\mathcal{S}_{ref}). La meilleure valeur pour un problème peut être atteinte par plusieurs résolutions, c'est notamment le cas pour Q_h . DMDFS est la stratégie qui obtient le plus de résultats de référence sur les indicateurs Q_a, Σ et Σ^P . IDFS permet de se rapprocher de la frontière d'un problème dans davantage de cas que les autres stratégies de sélection. SSR est impliquée dans plus de la moitié des références pour les indicateurs Q_h et Q_d . LF_I semble ici plus efficace sur Q_a, Σ et Σ^P . Aucune stratégie ne permet d'obtenir systématiquement le meilleur résultat sur l'ensemble des problèmes. Dans la table IV.5 on donne pour chaque problème la meilleure combinaison selon les valeurs de Σ^P . Le couple(DMDFS/ LF_I) est le plus performant dans 9 cas. Chaque opérateur apparaît au moins une fois dans le tableau. Parmi les 16 combinaisons possibles, 12 sont représentées.

TABLE IV.5 – Meilleures combinaisons (stratégie) pour chaque problème selon les valeurs de Σ^P

Problème	Sélection	Bissection	Problème	Sélection	Bissection
<i>2bartruss</i>	DLFS	LF_I	<i>disbrake</i>	DMDFS	LF_I
<i>4bartruss</i>	DMDFS	LF_E	<i>frigo</i>	DPFS	LF_I
<i>actuator_exp</i>	DPFS	RR	<i>fuel</i>	DMDFS	LF_I
<i>alkyl</i>	DMDFS	LF_E	<i>gear_exp</i>	DMDFS	RR
<i>beam_huy</i>	DMDFS	LF_I	<i>house</i>	DMDFS	SSR
<i>beam_osy</i>	DPFS	RR	<i>icengine_exp</i>	DMDFS	LF_I
<i>bearing</i>	IDFS	LF_E	<i>launch</i>	IDFS	SSR
<i>bobine</i>	DPFS	LF_I	<i>reducer_exp</i>	DLFS	SSR
<i>bolt_exp</i>	IDFS	LF_I	<i>sample</i>	DPFS	RR
<i>brushless_brisset_10</i>	IDFS	SSR	<i>screw_plb</i>	DPFS	LF_I
<i>brushless_brisset_6</i>	DLFS	LF_E	<i>ship</i>	DMDFS	LF_I
<i>carside_exp</i>	DPFS	LF_I	<i>spring</i>	DMDFS	LF_I
<i>chance</i>	DMDFS	LF_I	<i>vibration</i>	DMDFS	LF_I
<i>chem</i>	DMDFS	LF_I	<i>water_exp</i>	DMDFS	LF_E
<i>disbrake</i>	DMDFS	LF_I	<i>battery</i>	DLFS	SSR

TABLE IV.6 – Moyenne des résultats de l'exploration partielle de l'ensemble des problèmes du benchmark pour chaque stratégie considérée

	t (s)	Q_c	Q_a	Q_h	Q_d	Σ	Σ^P
DMDFS	113,84	982,0	0,727	0,806	0,844	2,377	4,715
DLFS	64,83	991,6	0,811	0,842	0,893	2,546	5,042
DPFS	60,57	992,3	0,777	0,830	0,877	2,484	4,921
IDFS	50,61	993,9	0,735	0,829	0,918	2,482	4,875
LF_I	72,49	993,2	0,869	0,823	0,885	2,577	5,093
LF_E	82,38	992,5	0,765	0,838	0,901	2,503	4,943
RR	49,78	1000	0,803	0,799	0,866	2,468	4,869
SSR	85,20	974,1	0,613	0,847	0,880	2,340	4,647

IV.6.2.b Valeurs moyennes

Le tableau IV.6 donne la moyenne des indicateurs de qualité Q_a , Q_h et Q_d , du nombre de solutions calculées Q_c et des temps de calcul (t). Le tableau IV.7 donne les résultats pour chaque combinaison (sélection ;bissection). Les meilleurs résultats obtenus pour chaque critère sont mis en valeurs en gras.

Globalement, dans le tableau IV.6, les algorithmes obtiennent des résultats comparables. Chaque stratégie semble présenter un léger avantage compétitif sur les autres pour au moins un des indicateurs. IDFS est la stratégie la plus efficace pour explorer rapidement les espaces de recherche et DMDFS converge nettement moins rapidement que les autres stratégies. Ce résultat est en accord avec l'évaluation des complexités des stratégies dans la section IV.3. DLFS obtient les meilleurs résultats globaux selon les valeurs de Σ , Σ^P . IDFS est efficace pour maximiser la distance moyenne entre les solutions (Q_d) et DPFS a des performances proches de DLFS. Côté bissection, LF_I est l'opérateur le plus efficace pour maximiser la diversité finale (Σ et Σ^P). La stratégie de bissection SSR est impliquée dans 5 des 7 résolutions qui n'ont pas atteint le nombre de solutions avant le temps imparti.

L'analyse du tableau IV.7, permet de confirmer l'intérêt de l'opérateur de bissection LF_I qui est

TABLE IV.7 – Moyenne des résultats de l'exploration partielle de l'ensemble des problèmes du benchmark pour chaque combinaison de stratégies

Sélection	Bissection	t (s)	Q_c	Q_a	Q_h	Q_d	Σ	Σ^p
DMDFS	LF_I	118,96	972,9	0,869	0,842	0,849	2,560	5,114
DLFS		59,95	1000	0,903	0,833	0,886	2,622	5,190
DPFS		78,70	1000	0,894	0,836	0,884	2,614	5,181
IDFS		32,33	1000	0,811	0,782	0,919	2,512	4,887
DMDFS	LF_E	174,10	970,0	0,746	0,844	0,860	2,450	4,884
DLFS		58,38	1000	0,792	0,834	0,901	2,527	4,988
DPFS		50,99	1000	0,791	0,825	0,901	2,517	4,958
IDFS		46,04	1000	0,730	0,847	0,942	2,520	4,944
DMDFS	RR	68,92	1000	0,776	0,768	0,828	2,373	4,686
DLFS		43,05	1000	0,851	0,821	0,882	2,553	5,045
DPFS		48,95	1000	0,840	0,831	0,889	2,559	5,061
IDFS		38,22	1000	0,746	0,776	0,865	2,387	4,684
DMDFS	SSR	93,38*	985,2	0,517	0,768	0,839	2,124	4,176
DLFS		97,92	966,5	0,699	0,882	0,902	2,483	4,946
DPFS		63,65	969,2	0,583	0,828	0,833	2,243	4,482
IDFS		85,85	975,6	0,653	0,910	0,947	2,509	4,983

impliqué dans les trois meilleurs algorithmes selon les valeurs de Σ^p . En particulier, considérant à la fois Σ et Σ^p , le couple (DPFS/ LF_I) apparaît comme la meilleure combinaison pour assurer une bonne diversité globale du résultat. La combinaison qui retourne le meilleur compromis entre temps de calcul et diversité est (IDFS/ LF_I). (IDFS/SSR) est particulièrement efficace pour maximiser la couverture et garder une bonne distance moyenne entre les solutions. Trois autres combinaisons sont intéressantes sur la diversité globale. (DLFS/RR) et (DPFS/RR) donnent de bonnes performances avec des temps de calculs moindre que (DPFS/ LF_I). L'algorithme AB&P associant DMDFS et SSR est le moins performant sur la diversité. Sur la stratégie SSR, on observe que DMDFS obtient un meilleur temps de calcul moyen que DLFS à cause de valeurs extrêmes sur certains problèmes. Si l'on ne considère pas ces valeurs extrêmes, le temps de calcul moyen de DMDFS est plus important.

IV.6.2.c Discussion

Ces résultats sur l'ensemble des problèmes permettent d'identifier les algorithmes AB&P qui ont le plus de chance de donner une bonne diversité. Les nouvelles stratégies DLFS et DPFS, ainsi que la stratégie de bisection spécifique LF_I donnent de meilleurs résultats que les stratégies existantes pour de nombreux problèmes. En particulier, ils permettent d'améliorer nettement les temps de calcul de DMDFS. Cependant, aucune des combinaisons n'est significativement meilleure sur l'ensemble des problèmes et pour tous les indicateurs. Les résultats dépendent fortement des caractéristiques de chaque problème. Dans un second temps, nous avons différencié les problèmes entre numériques et mixtes pour étudier si le type des problèmes est impactant sur le résultat. Aucune tendance claire n'a pu être mise en évidence pour discriminer davantage les algorithmes entre eux.

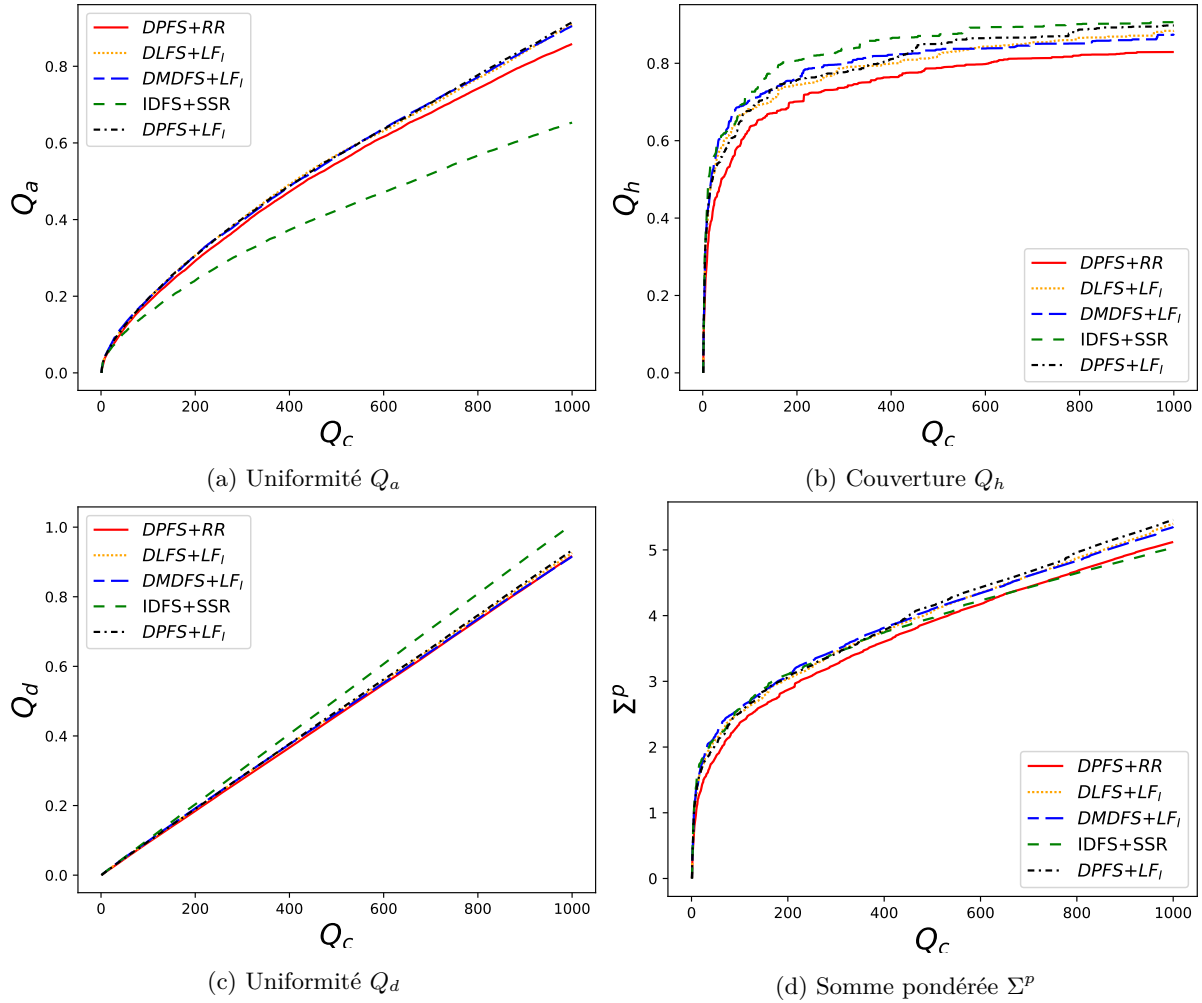


FIGURE IV.8 – Profils de performance des stratégies d’exploration partielle

IV.6.3 Profils de performance

Nous nous intéressons maintenant à l’évolution de la diversité pendant l’exploration. Pour cela, on considère quelques combinaisons de stratégies parmi les plus prometteuses : (DPFS/LF_I), (IDFS/SSR), (DPFS/RR) et (DLFS/LF_I). Auxquelles on ajoute (DMDFS/LF_I) car elle obtient de bons résultats moyens et permet d’atteindre le plus souvent les résultats de références. Les 5 figures IV.8 présentent les profils de performance pour chaque indicateur de qualité et les fonctions d’agrégation. Les profils de Q_h et Σ^p montrent l’importance des premières solutions calculées sur la diversité finale du résultat. En moyenne, les 200 premières solutions permettent d’atteindre entre 70% et 80% de la couverture du problème et plus de 50% de la valeur de Σ^p à 1000 solutions.

Une diversité "idéale" a pu être définie facilement pour les problèmes des n boules. C’est plus délicat pour nos problèmes de conception. Le maximum de diversité déterminé par les indicateurs n’est pas nécessairement atteint après 1000 solutions pour nos problèmes de conception. Tant que Q_a ou Q_h augmentent, la diversité maximale telle que nous l’avons définie n’est pas atteinte. On observe sur l’ensemble

des problèmes qu'après quelques dizaines ou centaines de solutions, chaque nouvelle solution n'augmente pas systématiquement la diversité. En pratique, on cherchera à construire le meilleur compromis entre un nombre de solutions raisonnable et une diversité suffisante à partir d'une méthode simple de post-traitement.

IV.7 Bilan du chapitre

Dans ce chapitre, nous avons établi un cadre pour l'exploration partielle de CSP numériques et mixtes en conception. Ce cadre s'appuie sur un algorithme AB&P associé au paradigme BestFS pour construire des stratégies adaptées à la recherche de solutions diversifiées. La stratégie existante DMDFS repose sur un calcul de distance et des mécanismes dynamiques qui amènent à une forte complexité. Nous avons défini deux nouvelles stratégies et adapté une stratégie existante qui manipulent des propriétés statiques moins coûteuses de l'arbre de recherche. En plus de la stratégie de sélection, les autres composants des algorithmes AB&P ont été étudiés. Notamment l'opérateur de division qui impacte fortement la diversité finale des résultats. Quatre méthodes différentes de bisection ont été considérées pour les expérimentations. Aussi, plusieurs caractéristiques secondaires ont été adaptées pour pouvoir résoudre efficacement des problèmes de dimensionnement en conception. Par exemple, les fonctions d'évaluation liées à des grandeurs physiques ont été normalisées pour que le résultat soit homogène. De la même manière, la précision a été définie relativement aux domaines initiaux pour garantir une instanciation homogène des ϵ -boîtes.

Afin d'évaluer la qualité du résultat d'une exploration partielle, la notion de diversité a été définie autour de trois grandeurs : l'uniformité, la couverture et la cardinalité. Nous avons introduit trois indicateurs pour quantifier l'uniformité et la couverture. L'objectif était de proposer des outils pour comparer nos algorithmes. Nous avons testé les différentes combinaisons possibles de stratégies de sélection et de bisection sur un ensemble de problèmes de conception. Cette étude a permis d'identifier les algorithmes les plus prometteurs. En particulier, elle a prouvé l'efficacité des stratégies de sélection *statiques*. Cependant, pour traiter ces problèmes non-linéaires, le choix des paramètres de résolution reste dépendant des caractéristiques du problème considéré. De plus, les problèmes de dimensionnement bien définis sont rares dans la littérature et notre benchmark est limité.

Les indicateurs de qualité autorisent l'utilisation de profils de performances pour étudier l'évolution de la diversité au fur et à mesure des explorations. L'analyse de ces courbes aide à la définition de critères d'arrêt pertinent pour explorer un problème donné. En pratique, les algorithmes alternent des phases de diversification et d'intensification et le résultat brut d'un algorithme AB&P n'est souvent pas suffisant. Aussi, les variations sur les profils de performances peuvent aider à identifier les solutions qui participent le plus à la diversité globale. Dans le chapitre suivant, nous verrons comment utiliser de manière pertinente les résultats de l'exploration partielle pour construire un sous-ensemble de solutions avec une bonne diversité et une cardinalité raisonnable.

L'EXPLORATION D'ESPACE DE RECHERCHE POUR L'AIDE À LA DÉCISION EN CONCEPTION

V.1 Introduction

V.1.1 Contexte

Dans ce chapitre, nous cherchons à répondre à la dernière question de recherche :

L'exploration partielle diversifiée est-elle un outil pertinent pour traiter des problèmes de conception architecturale ?

Nous allons appliquer notre approche d'exploration partielle à trois cas d'études de conception architecturale et comparer les résultats avec des approches plus classiques d'optimisation multi-objectifs. Les cas d'études sont les suivants : le moteur à combustion interne (*ice*), le moteur à aimants permanents sans encoche (*mapse*) et la batterie (*batterie*). Les deux premiers sont tirés de problèmes connus de la littérature et le troisième est le problème formalisé au chapitre III. Les modèles correspondants sont de type numériques ou mixtes avec des contraintes non-linéaires et au plus quelques dizaines de variables.

V.1.2 Approches d'optimisation utilisées

De nombreuses méthodes et solveurs existent pour l'optimisation de problèmes multi-objectifs sous contraintes. Quelques approches parmi les plus utilisées en ingénierie sont présentées dans les revues [2, 167]. Notre but n'est pas de faire une comparaison exhaustive de ces approches, mais d'analyser les résultats attendus pour positionner nos approches d'exploration. Les solveurs sélectionnés seront utilisés à la manière de boîtes noires représentatives des méthodes d'optimisations en général.

Les algorithmes génétiques comme NSGA-II [38] et NSGA-III [87] sont aujourd'hui largement répandus en conception préliminaire comme solveur d'optimisation multi-objectifs [133, 67, 57, 18, 192, 141, 46]. Ils permettent d'approximer un front de Pareto en maintenant une population d'individus avec

des opérateurs de croisement et de mutation. Leur fonctionnement est décrit dans l’annexe D.1. La bibliothèque python pymoo¹ propose les deux algorithmes NSGA-II et NSGA-III, elle permet notamment de gérer des problèmes numériques ou mixtes. Le tableau V.1 donne les paramètres principaux utilisés pour résoudre nos problèmes d’optimisation. Les paramètres restants (taille de la population, nombre de générations, nombre de directions) seront déterminés en fonction du modèle considéré.

TABLE V.1 – Paramétrage initial des algorithmes NSGA-II et NSGA-III, n est le nombre de variables du problème à résoudre

Paramètre	Valeur
Mutation rate	1/n
Mutation distribution	30
Crossover probability	0.95
Crossover distribution	20

La figure V.2 montre le parallèle entre l’approche d’exploration partielle et l’optimisation. Chaque problème de conception est initialement un problème de satisfaction de contraintes. Dans le cas d’une optimisation, les critères de performance sont traduits en objectifs à optimiser. Pour l’exploration partielle, les critères sont convertis en contraintes d’inégalités (ϵ -contraintes) pour centrer le problème autour de la portion de l’espace qui nous intéresse.

V.1.2.a Hypervolume dans l’espace des objectifs

L’hypervolume HV introduit par [207] est un indicateur de qualité en optimisation multi-objectifs. Il mesure dans l’espace des objectifs le volume dominé par un ensemble de solutions. Il est construit initialement pour évaluer à la fois la convergence et la diversité d’une population approximant le front de Pareto. L’avantage de l’hypervolume est que sa mesure ne repose pas sur une ou plusieurs solutions de références. HV est déterminé à partir du point de l’espace des objectifs le plus éloigné des valeurs idéales. La figure V.1 donne une illustration en deux dimension de l’hypervolume. Le point nadir et le point idéal caractérise l’enveloppe des points du front de Pareto.

Pour ce travail, HV nous permet d’évaluer aussi d’avoir un indicateur supplémentaire pour la qualité des résultats de l’exploration dans l’espace des objectifs. En effet, l’hypervolume évalue la couverture et la diversité d’une population dans les directions préférentielles des V_p .

V.1.3 Protocole expérimental

V.1.3.a Optimisation

Les algorithmes génétiques comme NSGA-III nécessitent un paramétrage adapté pour converger correctement vers le front de Pareto. La première étape est donc de déterminer ce paramétrage, notamment la taille de la population, le nombre de générations ou encore le nombre de directions de références pour NSGA-III. Les paramétrages appliqués pour des problèmes existants de la littérature [87, 192, 46] permettent d’estimer correctement une première fois ces quantités. Puis la qualité du résultat (diversité et

1. <https://pymoo.org/>

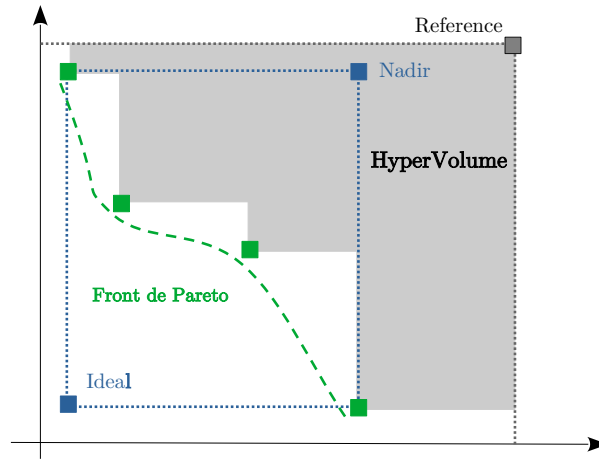


FIGURE V.1 – Représentation de l’hypervolume HV dans un espace en deux dimensions avec 4 solutions (s_1, \dots, s_4) et un point de référence ref .

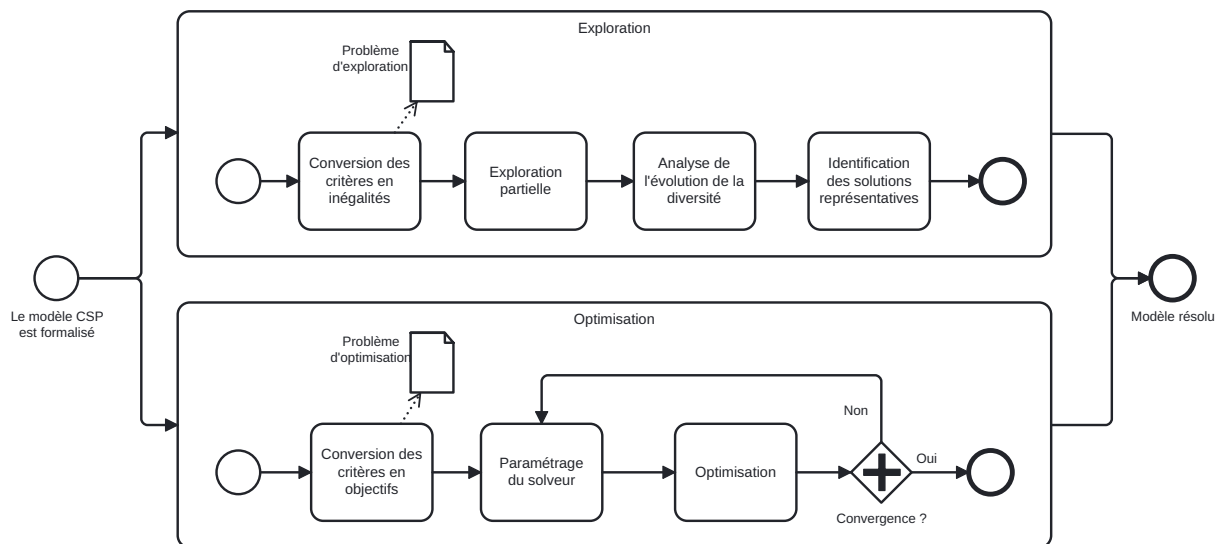


FIGURE V.2 – Protocole de résolution des approches d’exploration et d’optimisation

convergence) est vérifiée à partir des indicateurs Q_h et Q_a introduits au chapitre IV et de l'hypervolume HV défini par [207]. Finalement, les solutions issues de la bonne configuration sont conservées et représentées dans l'espace de conception et dans l'espace des objectifs.

V.1.3.b Exploration partielle

Pour cette étude, les explorations partielles sont réalisées par un algorithme AB&P implémenté avec la stratégie de sélection IDFS et la stratégie de bisection SSR présentées dans le chapitre IV. Cette combinaison d'opérateurs donne de bon résultats en un temps raisonnable. De plus, l'intérêt particulier de la stratégie de sélection IDFS est d'être indépendante des caractéristiques du problème et donc du nombre de V_p qui peut varier selon la formalisation du modèle. La précision pour l'instanciation des variables continues est définie de la même manière qu'au chapitre IV, en utilisant l'expression IV.14. Pour chaque problème, l'objectif est d'obtenir une population de solutions bien réparties dans l'espace de recherche notée \mathcal{S}_{exp} .

Dans la pratique, il est nécessaire de trouver un équilibre entre la diversité et la cardinalité du résultat final. Il faut atteindre une diversité suffisante pour représenter l'espace réalisable tout en conservant une taille raisonnable qui soit facilement gérable par un être humain. Limiter l'exploration partielle au nombre de solutions souhaité ne permet pas de garantir une bonne diversité finale. Nous avons vu dans le chapitre précédent que les stratégies mises en place alternent diversification et intensification. De fait, toutes les solutions générées ne participent pas à la qualité globale du résultat. Le sous-ensemble de solutions issu d'une exploration partielle doit être réduit pour ne conserver que les meilleures solutions.

La résolution est faite en deux temps. Premièrement, le problème est exploré "en excès" en générant un grand nombre de solutions réparties dans l'espace réalisable. Cette résolution est dimensionnée par un critère d'arrêt ϕ comme un temps de calcul t ou un nombre maximal de solutions Q_c^{max} . Deuxièmement, le résultat est réduit à une taille raisonnable (au plus quelques centaines de solutions) en ne conservant que les meilleures solutions. Les solutions sont sélectionnées à partir de l'évolution des indicateurs Q_a et Q_h . Si le problème présente plusieurs V_p , le calcul de l'hypervolume permet d'identifier les quelques solutions qui couvrent le front de Pareto. Finalement, nous proposons la méthode de sélection suivante :

Soit Q_i un indicateur de qualité et λ_i un réel donné. $Q_i(j)$ est la valeur de Q_i pour le sous-ensemble de solutions formé par les j premières solutions. La $j^{\text{ème}}$ solution générée est retenue dans la population finale \mathcal{S}_{exp} si :

$$Q_i(j) - Q_i(j-1) > \lambda_i Q_i(j-1), \quad j \geq 2 \quad (\text{V.1})$$

Cette méthode revient à ne considérer que les solutions qui augmentent de λ au moins un des indicateurs sur la diversité. Selon le problème, différentes valeurs de λ peuvent être considérées pour identifier \mathcal{S}_{exp} . En pratique, l'évolution de HV et Q_h est en escalier et les solutions qui améliorent réellement la frontière du problème sont peu nombreuses. Pour ces deux indicateurs, on pourra choisir $\lambda_i = 0$. Sélectionner les meilleures solutions selon l'indicateur Q_a au fur et à mesure de l'exploration revient à rejeter les individus qui sont trop similaires à des solutions déjà identifiées. Le choix de λ_a pour cet indicateur

dépend du concepteur et du problème.

V.2 La machine à aimants permanents sans encoche

L'étude de ce problème de conception permet de détailler l'application de notre méthode d'exploration partielle et de comparer le résultat obtenu avec celui d'une approche d'optimisation.

V.2.1 Description du modèle

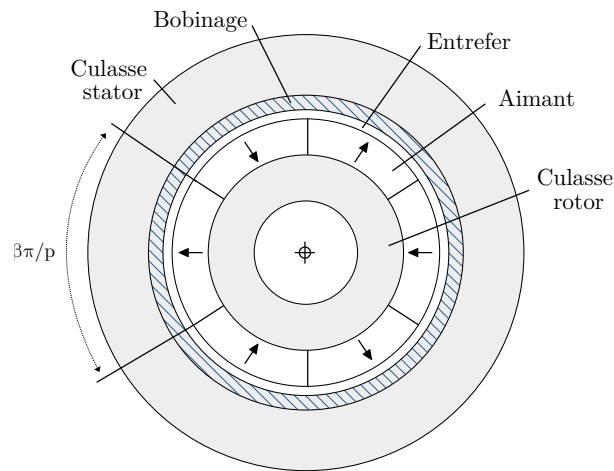


FIGURE V.3 – Schéma du problème du moteur à aimants permanents (*mapse*)

TABLE V.2 – Nomenclature du problème *mapse*

Nom	Description	Type de paramètre
e	Entrefer mécanique (m)	Conception
E	Épaisseur des bobinages (m)	Conception
l_a	Épaisseur de l'aimant (m)	Conception
β	Coefficient d'arc polaire	Conception
p	Nombre de paires de pôle	Conception
V_u	Volume des parties actives (m^3)	Critère
V_a	Volume de l'aimant (m^3)	Critère
C_m	Couple électromagnétique (N.m)	Constante
L	Longueur de la machine (m)	Auxiliaire
P_j	Pertes par effet joule (W)	Critère

Le problème de la machine à aimants permanents sans encoche est proposé dans [99] et représenté sur la figure V.3. Il est utilisé dans plusieurs travaux de recherche en conception préliminaire, [115, 112, 42]. Les leviers de conception réels sont l'épaisseur de l'entrefer e , le coefficient d'arc polaire β , le nombre de paires de pôles p , l'épaisseur de l'aimant l_a et l'épaisseur des bobinages E . La longueur de la machine L décrite comme une V_{co} par l'auteur est en pratique une variable déduite de l'expression

du couple électromagnétique dont la valeur est fixée. Le cahier des charges demande de dimensionner le moteur pour développer un couple C_m de $10Nm$. L'ensemble des exigences données dans [115] font apparaître 13 contraintes d'inégalités, notamment pour restreindre les domaines de variables auxiliaires. Les critères de performance sont le volume des parties actives V_u , le volume de l'aimant V_a et les pertes joules P_j . Le modèle analytique est donné en annexe D.2. Le problème de dimensionnement résultant est ainsi composé de 5 variables de conception, 7 variables auxiliaires et 3 critères de performances (les paramètres principaux sont donnés dans le tableau V.2). Le problème est mixte non-linéaire. Pour encadrer l'exploration de l'espace dans une zone pertinente, les domaines des V_p sont restreints par une borne supérieure tirée des résultats de [115] :

$$V_u \leq 6.5e^{-4}$$

$$V_a \leq 1.5e^{-4}$$

$$P_j \leq 45$$

V.2.2 Résolution

V.2.2.a Optimisation avec NSGA-III

Le problème d'optimisation correspondant compte 3 objectifs pour minimiser les 3 critères de performance V_u , V_a , et P_j . On utilise pour sa résolution l'algorithme génétique NSGA-III. Plusieurs paramétrages sont comparés pour obtenir le résultat le plus satisfaisant. La résolution retenue a été réalisée avec le nombre de directions de référence fixé à 153, la taille de la population à 156 et le nombre de générations à 1000. On obtient 55 solutions après 60 secondes. Le sous-ensemble correspondant est noté \mathcal{S}_{opt} . Les résultats sont donnés dans un graphe de coordonnées parallèles (Figure V.5b). Le calcul de l'hypervolume HV est réalisé avec le point de référence $(V_u, V_a, P_j) = (0.00065, 0.00015, 45)$. On obtient pour \mathcal{S}_{opt} , $HV_{opt} = 3.33e^{-8}$.

V.2.2.b Exploration de l'espace réalisable

On explore l'espace de recherche de *mapse* avec le critère d'arrêt $t = 300$ secondes et la précision de $\epsilon = 10^{-5}$. Le résultat de cette exploration \mathcal{S}_1 contient 1549 solutions. Les profils de performance de la couverture Q_h , de l'uniformité Q_a et de l'hypervolume HV sont présentés dans la figure V.4. L'évolution de Q_h et HV donnent des graphiques en escalier ce qui caractérise les solutions plus importantes pour l'amélioration de ces indicateurs. Les solutions qui améliorent ces deux indicateurs sont sélectionnés avec l'expression V.1 et $\lambda_i = 0$. Le seuil de sélection pour Q_a est fixé à $\lambda_a = 0.005$. Finalement, on obtient un sous-ensemble réduit de solution \mathcal{S}_{exp} composé de 320 solutions bien réparties.

Dans le tableau V.3 on donne les caractéristiques de trois sous-ensembles de solutions. $\mathcal{S}_1(1549)$ est le résultat de la résolution avant réduction, $\mathcal{S}_1(320)$ est le sous-ensemble formé par les 320 premières solutions de l'exploration et \mathcal{S}_{exp} est le sous-ensemble issu de la réduction de \mathcal{S}_1 . \mathcal{S}_{exp} est une représentation de l'espace réalisable plus diversifié que $\mathcal{S}_1(320)$ et avec une cardinalité plus raisonnable que $\mathcal{S}_1(1549)$.

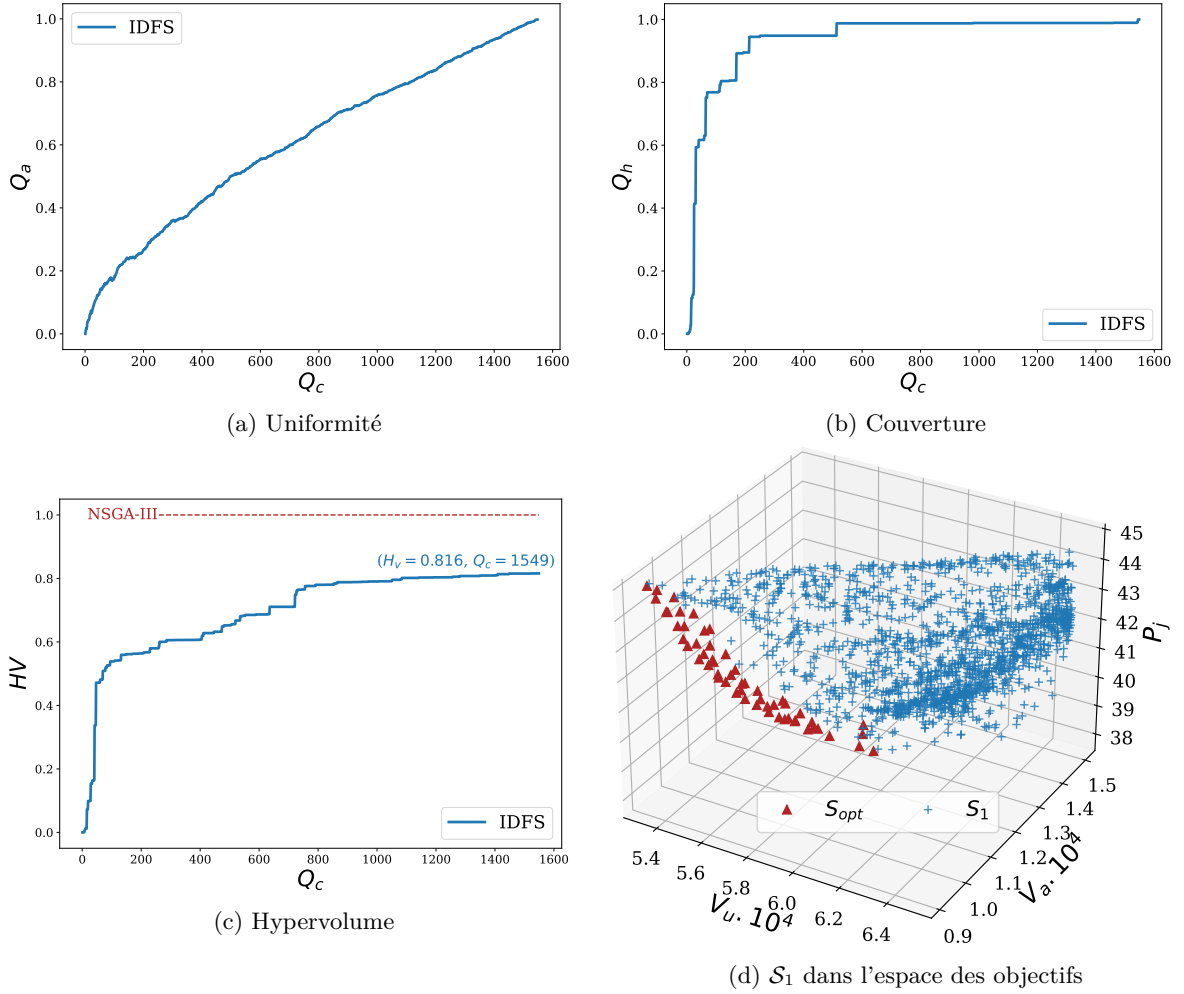


FIGURE V.4 – Profils de performance de l'exploration du problème *mapse* après $t = 300$ secondes et $Q_c = 1549$ solutions. (d) donne les résultats de l'exploration et de l'optimisation dans l'espace des objectifs.

TABLE V.3 – Caractéristiques des sous-ensembles de solutions issus de l'exploration et de l'optimisation du problème *mapse*

\mathcal{S}	Q_c	Q_h	Q_a/Q_c	HV/HV_{opt}
S_1	1549	1	0.0217	0.816
S_1	320	0.948	0.0400	0.606
S_{exp}	320	1	0.0492	0.816
S_{opt}	55	0.0004	0.023	1

V.2.3 Comparaison des résultats

La comparaison entre les sous-ensembles de solutions \mathcal{S}_{exp} et \mathcal{S}_{opt} est supportée par les graphes de coordonnées parallèles (GCP) présentés respectivement sur les figures V.5a et V.5b. Dans l'espace de conception formé par les variables e , E , l_a , et β , l'étendue des valeurs de \mathcal{S}_{exp} est bien plus importante que celle de \mathcal{S}_{opt} . Concrètement, le volume de l'enveloppe de \mathcal{S}_{opt} dans l'espace de conception couvre environ 0.13% de celui de \mathcal{S}_{exp} . La méthode d'optimisation ne révèle qu'une fraction des architectures envisageables quand l'exploration partielle permet d'appréhender l'ensemble du réalisable. Par exemple, les solutions obtenues avec NSGA-III considèrent des architectures de moteur avec une épaisseur de bobinage comprise entre 3 et 4.37 mm. L'exploration montre que des solutions existent avec une épaisseur de bobinage entre 1.61 et 8.62 mm, dont certaines peuvent être pertinentes pour le problème de conception.

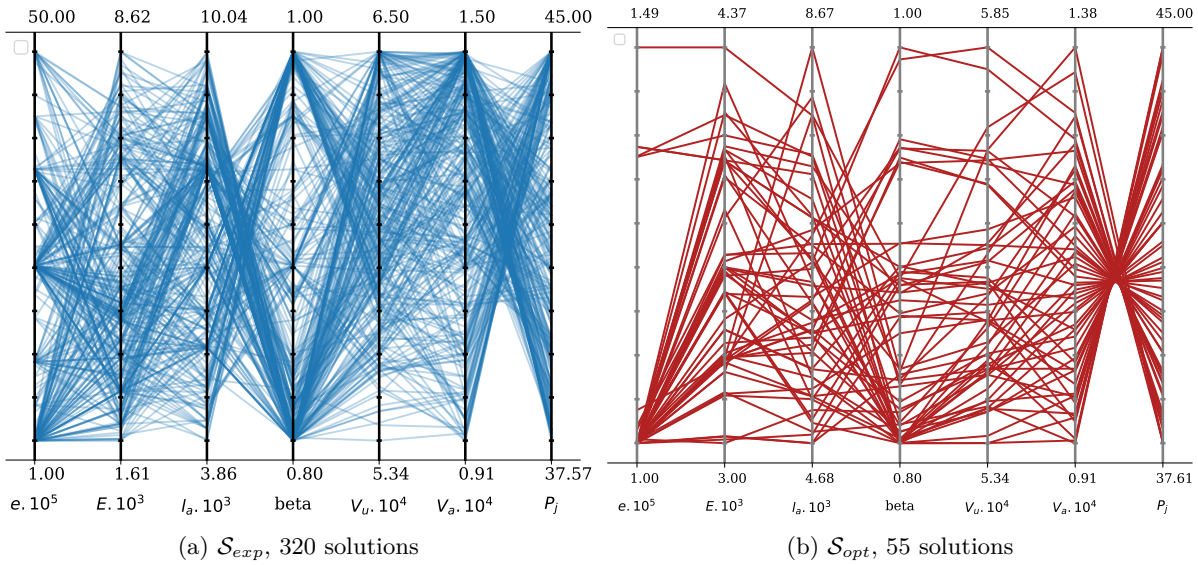


FIGURE V.5 – GCP des différents sous-ensembles de solutions construit après l'exploration partielle (bleu) et l'optimisation (rouge) du problème *mapse*

La représentation de l'espace des objectifs (Figure V.4d) et les valeurs de l'hypervolume HV/HV_{opt} (Tableau V.3) mettent en évidence que l'exploration n'a que partiellement couvert la frontière de Pareto. Ce constat peut s'expliquer par les expressions des critères de performances qui intègrent plusieurs occurrences des variables E , e et l_a . Or, le contracteur utilisé, HC_4 , est peu efficace face à ce type de contrainte [88]. Certaines des architectures de \mathcal{S}_1 sont clairement dominées par \mathcal{S}_{opt} . Cependant, il est possible d'identifier des solutions proches des valeurs optimales et intéressantes à considérer en conception préliminaire.

Pour mettre en évidence ces architectures quasi-optimales, les solutions de \mathcal{S}_{exp} sont filtrées. Ne sont conservées que les solutions incluses dans l'espace des objectifs de \mathcal{S}_{opt} (ou dominées par le point nadir de \mathcal{S}_{opt}), c'est-à-dire avec des performances comparables. Parmi les 87 solutions ainsi identifiées, on met en évidence celles qui sont en dehors de l'espace de conception de \mathcal{S}_{opt} . Le sous-ensemble restant \mathcal{S}_{exp}^* est composé de 79 solutions quasi-optimales et représenté sur les figures V.6. Les architectures de \mathcal{S}_{exp}^* offrent des performances proches de l'optimale avec des caractéristiques de conception très différentes

des solutions de l'optimisation. Le modèle *mapse* n'étant qu'une abstraction de la réalité, il peut être pertinent de considérer ces architectures préliminaires pour la phase de conception détaillée.

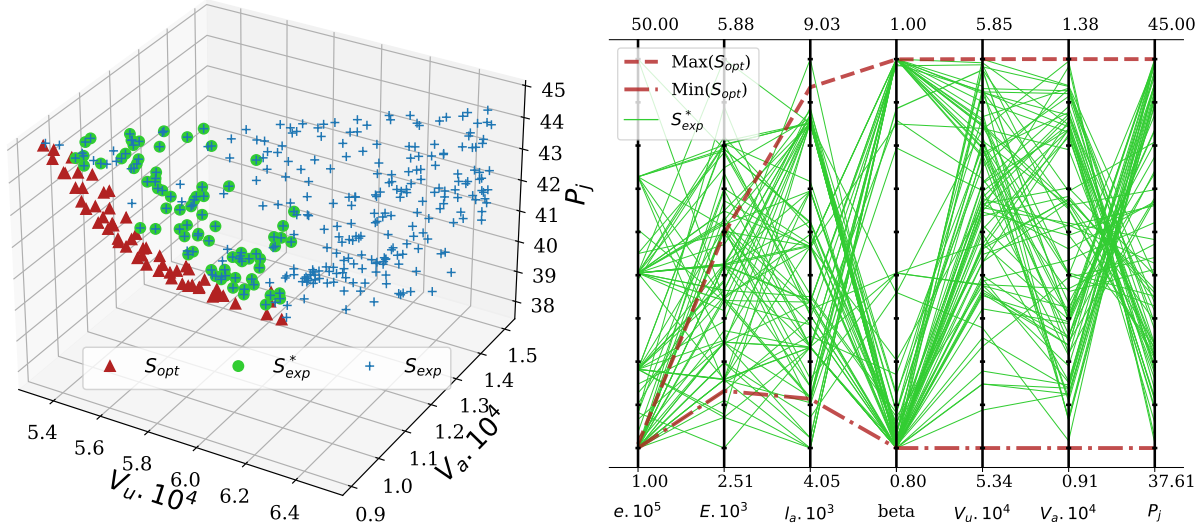


FIGURE V.6 – Représentations des solutions quasi-optimales (vert) dans l'espace des objectifs et dans un GCP

V.3 Le moteur à combustion

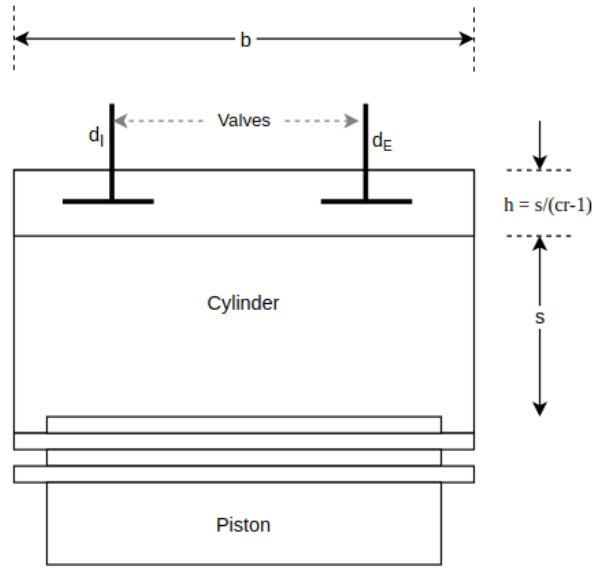
L'objectif de ce cas d'étude est d'appliquer l'approche d'exploration partielle à un cas simple et d'illustrer les possibilités d'interaction pour l'utilisateur.

V.3.1 Description du problème

TABLE V.4 – Nomenclature du problème *ice*

Nom	Description	Type de paramètre
b	Diamètre des pistons (mm)	Conception
c_r	Ratio de compression	Conception
d_I	Diamètre de la valve d'admission (mm)	Conception
d_E	Diamètre de la valve d'éjection (mm)	Conception
w	Nombre de tours par minute (rpm)	Conception
h	Hauteur de la chambre de soupape (mm)	Auxiliaire
s	Parcours du piston (mm)	Auxiliaire
N_c	Nombre de cylindres	Constante
PwB	Puissance par unité de volume (kW/l)	Critère
$ISFC$	Consommation d'essence (g/kWh)	Critère

Le problème de conception d'un moteur à combustion interne (*ice*) est décrit dans [134] par un modèle gros grain dont l'objectif initial est de maximiser la puissance par unité de volume PwB . Le


 FIGURE V.7 – Schéma du moteur à combustion interne à tête plate (*ice*)

deuxième critère de performance du problème est la consommation d'essence notée $ISFC$. Les paramètres principaux sont décrits dans le tableau V.4 et représentés sur la figure V.7. Le modèle proposé par l'auteur est composé de 5 variables de conception, 2 variables auxiliaires, 2 critères de performances et 9 contraintes d'inégalités. C'est un CSP numérique non-linéaire avec une contrainte par morceaux. Les domaines des V_{co} et les expressions du modèle complet sont donnés en annexe D.3. L'espace de recherche initial est grossièrement restreint par une borne supérieure sur $ISFC$ (Equation V.2) et une borne inférieure sur PwB (Equation V.3). Comment identifier quelques solutions préliminaires candidates pour la conception détaillée ?

$$c_{10} : ISFC \leq 230.5 \quad (V.2)$$

$$c_{11} : PwB \geq 13000 \quad (V.3)$$

V.3.2 Première exploration de l'espace réalisable

Nous ne disposons d'aucunes connaissances sur l'espace réalisable en dehors des bornes sur les domaines des variables. Le premier objectif est de comprendre cet espace pour identifier des zones intéressantes à explorer en détail. Pour cela, le problème initial est exploré une première fois de manière grossière avec un algorithme AB&P (IDFS/SSR). Cette résolution partielle est limitée par un temps de calcul $t = 60s$ et une précision relative aux domaines initiaux de 10^{-3} . On obtient un sous-ensemble \mathcal{S}_a composé de 6559 solutions. Les profils de performance de Q_h et HV donnés en annexe D.3 montrent que l'on atteint rapidement la frontière du problème. La figure V.8b représente les solutions dans l'espace des objectifs et la figure V.8a donne le GCP de quelques solutions choisies pour matérialiser l'enveloppe de

l'espace réalisable.

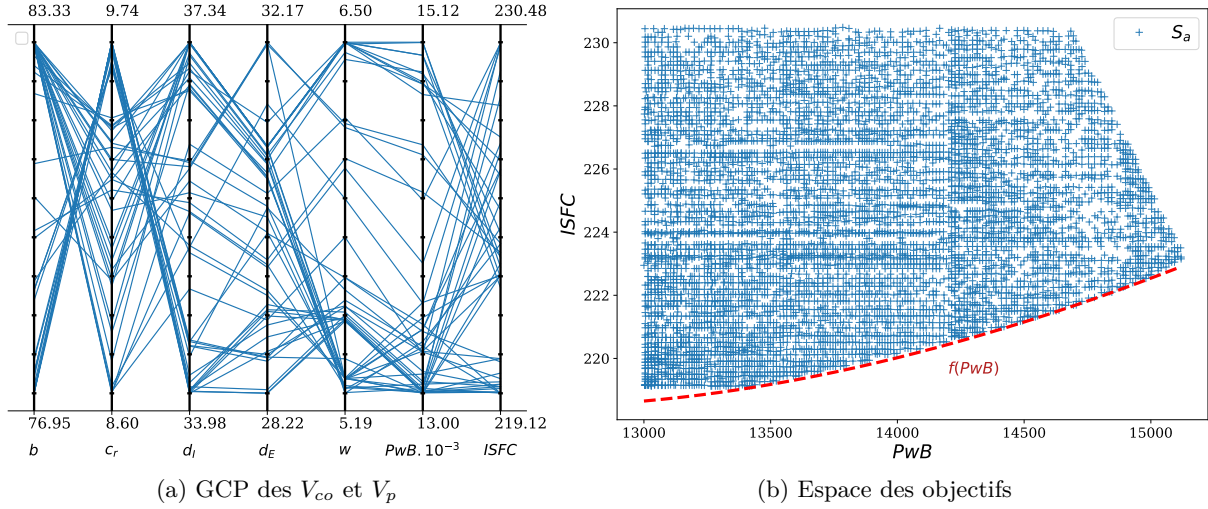


FIGURE V.8 – Résultat de la première exploration du problème *ice* \mathcal{S}_a ($t = 60s, Q_c = 5499$)

La représentation des solutions dans l'espace des objectifs sur la figure V.8b fait apparaître une frontière dans les directions des V_p . Celle-ci correspond au front de Pareto du problème d'optimisation bi-objectif équivalent : maximiser PwB et minimiser $ISFC$. La frontière est approchée en partie par la fonction V.4 tracée en rouge sur cette figure. Une grande partie de l'espace réalisable du problème initial est très éloigné de cette frontière. À partir des résultats de cette première résolution, nous choisissons de restreindre l'espace réalisable plus proche du front de Pareto. Pour cela, nous pouvons ajouter la contrainte d'inégalité $ISFC \leq f(PwB) + \epsilon$, avec ϵ un réel positif.

$$f(PwB) = 125.7 \log(PwB)^2 - 2372.24 \log(PwB) + 11410.85 \tag{V.4}$$

V.3.3 Exploration du problème réduit

V.3.3.a Reformulation du problème *ice*

Le modèle analytique initial de *ice* D.3 est complété par la contrainte c_{12} V.5 afin de garantir une tolérance de $1g/kWh$ sur la consommation d'essence pour une puissance par unité de volume donnée. L'espace réalisable sur la figure (figure V.8b) montre qu'en dessous de $PwB = 13400kW/l$, la consommation ne diminue plus. La borne inférieure sur PwB est fixée à cette valeur. À partir de valeurs extrêmes sur les V_{co} et les V_p identifiées lors de la première exploration, les domaines initiaux des variables sont réduits. Cette étape permet d'affiner la définition de la précision relative aux bornes des domaines (IV.14). Finalement, les domaines des paramètres principaux du nouveau problème sont donnés dans le tableau V.5. Nous cherchons une représentation de cet espace de recherche. La première approche envisageable est de filtrer les solutions générées lors de la résolution initiale. Cependant, cette méthode ne garantit pas la représentativité de l'espace réalisable du problème restreint. Il est préférable de réaliser une nouvelle exploration avec une précision plus fine pour obtenir un résultat fiable.

TABLE V.5 – Nouveaux domaines initiaux des V_{co} du problème *ice*

Nom	Domaines
b	[76, 84]
c_r	[8, 10]
d_I	[33, 38]
d_E	[28, 33]
w	[5, 6.5]
$ISFC$	[219, 230]
PwB	[13400, 16000]

$$c_{12} : ISFC \leq f(PwB) + 1 \quad (V.5)$$

L'exploration est réalisée pour le problème *ice* restreint avec une précision fixée à 10^{-5} et un temps de calcul initial de 60 secondes. On obtient 680 solutions. L'objectif est d'obtenir un échantillon représentatif de l'espace de recherche et donc de sélectionner les meilleures comme pour le problème *mapse*. Pour garantir une bonne qualité de l'échantillon final, on poursuit la résolution pour générer a minima 2000 solutions (*start and stop*). Les 1320 solutions restantes sont générées en 131 secondes pour un temps de calcul total de 191 secondes. Le sous-ensemble de 1200 solutions est ensuite réduit à partir de notre méthode de sélection (expression V.1). Avec le paramétrage $\lambda = 0$ pour Q_h et HV , $\lambda = 0.01$ pour Q_a , 246 architectures sont sélectionnées pour former le sous-ensemble \mathcal{S}_b , représentatif de l'espace du problème restreint.

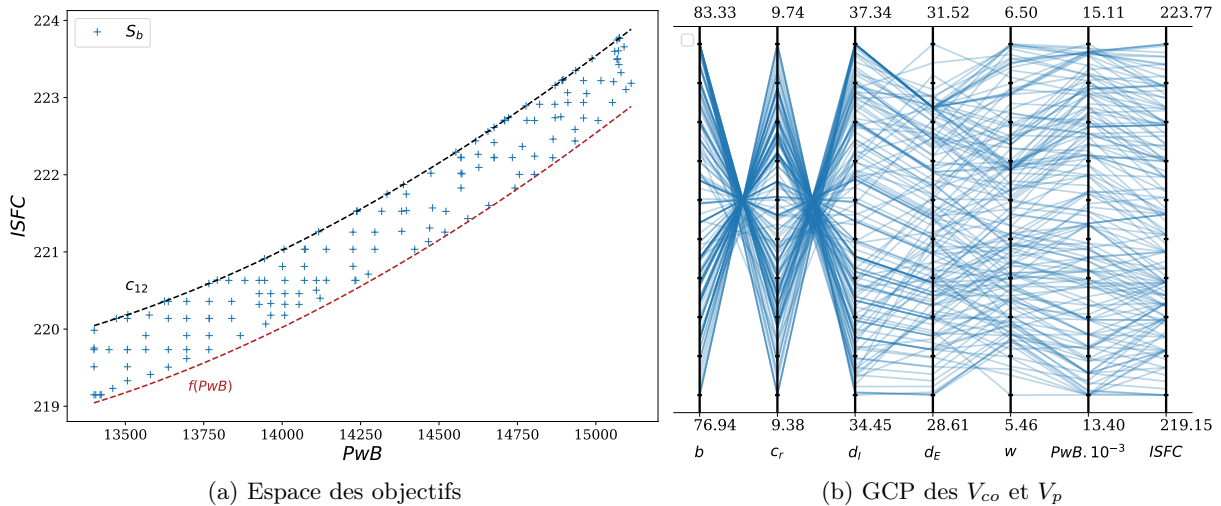

 FIGURE V.9 – Les 246 meilleures solutions issues de l'exploration de *ice* restreint \mathcal{S}_b , ($Q_c = 2000$, $t = 191s$)

TABLE V.6 – Comparaison de deux architectures possibles pour le problème *ice*

Performance	b	c_r	d_I	d_E	w	PwB	$ISFC$
Sous-optimale	82,19	9,50	36,37	31,04	6,26	14675,60	222,42
Optimale	80,27	9,59	35,97	29,85	6,12	14474,04	221,16

V.3.4 Comparaison avec les résultats de l'optimisation du problème réduit

Le problème *ice* réduit peut être converti en un problème d'optimisation en convertissant les deux critères $ISFC$ et PwB en objectifs : maximiser PwB et minimiser $ISFC$. Le front de Pareto de ce CMOP est approché avec NSGA-II et les paramètres de résolution suivants : taille de la population : 200, nombre de générations : 1000. Le résultat \mathcal{S}_{opt} est composé de 191 individus représentés par des triangles rouges sur la figure V.10a. Sur cette même figure on retrouve le résultat de l'exploration partielle \mathcal{S}_b . On constate que dans l'espace des objectifs, \mathcal{S}_b recouvre efficacement le front de \mathcal{S}_{opt} . Dans l'espace de conception, le volume de l'enveloppe de \mathcal{S}_b est 2.3 fois plus important que celui de \mathcal{S}_{opt} . L'exploration met donc en valeur des architectures différentes de celles du front de Pareto. Pour les mettre en valeur, on construit \mathcal{S}_b^* , le sous-ensemble des solutions de \mathcal{S}_b qui n'appartiennent pas au domaine de conception de \mathcal{S}_{opt} . \mathcal{S}_{opt} et \mathcal{S}_b^* sont représentés dans un GCP sur la figure V.10a. Les 20 solutions de \mathcal{S}_b^* sont dominées par \mathcal{S}_{opt} , mais représentent des alternatives viables pour une étude de faisabilité en conception préliminaire. En particulier au regard des hypothèses importantes sur lesquelles repose le modèle *ice* (par exemple un cycle adiabatique parfait). Dans le tableau V.6, on présente deux alternatives avec une puissance PwB comparable. La première est sous-optimale issue de l'exploration et la deuxième est issue de NSGA-II. Cet exemple montre qu'en dégradant légèrement le critère $ISFC$ le concepteur peut s'autoriser un design de chambre de combustion avec des diamètres de piston et de valves différents.

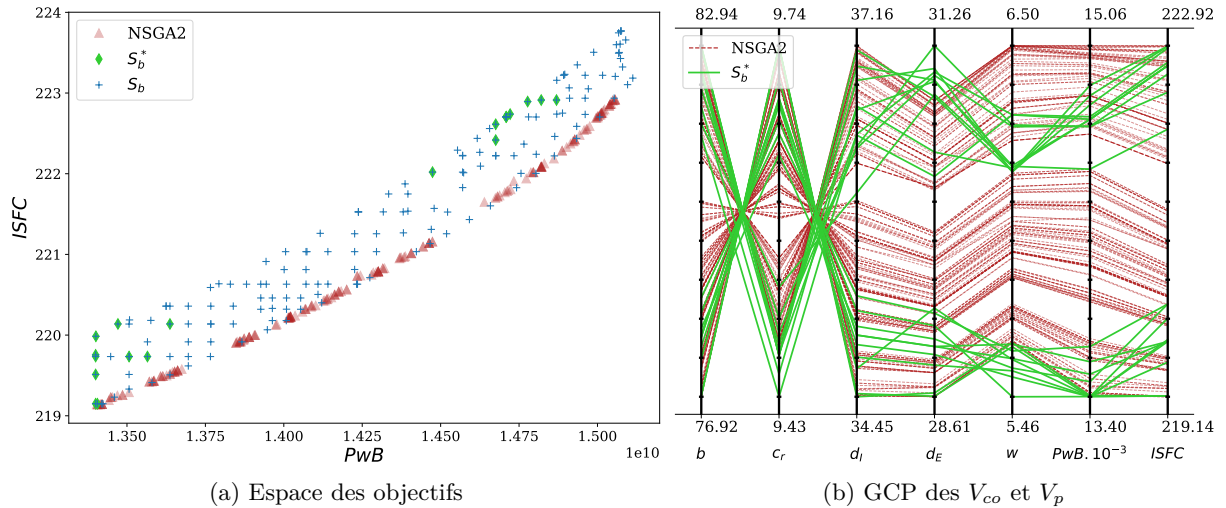


FIGURE V.10 – Comparaison des résultats de l'exploration (vert) et de l'optimisation (rouge) du problème *ice*

Finalement, sans connaissances préalables sur l'espace de conception ou le front de Pareto du problème *ice*, l'approche d'exploration partielle nous a permis d'interagir avec le modèle pour obtenir des solutions

dans une zone précise de l'espace. Une sélection de solutions bien réparties donne l'occasion au concepteur de comparer quelques architectures représentatives du réalisable avant la phase de conception détaillée. Sur le problème *ice*, le résultat de l'exploration partielle atteint correctement la frontière de Pareto dans l'espace des objectifs. Dans l'espace de conception, l'étendue des valeurs des V_{co} permet d'accéder à des solutions dominées, mais pertinentes dans un contexte de conception préliminaire.

V.4 Le pack Batterie pour véhicule électrique

Dans cette section, nous proposons de traiter le modèle (*batterie*) développé au chapitre II et issu du problème de conception du pack batterie li-ion introduit au chapitre III. La nomenclature du problème est donnée dans le tableau III.4 et le catalogue de cellules dans le tableau B.1. En amont de la résolution, plusieurs paramètres doivent être déterminés sur le modèle. Notamment les caractéristiques du véhicule et quelques points de fonctionnement. Ces valeurs sont estimées à partir de données sur les voitures électriques les plus récentes des constructeurs français.

La masse du véhicule sans la batterie et avec passager est fixée à $M_{base} = 1200$ kg. La batterie doit pouvoir assurer une consommation d'énergie massique moyenne $k_{elec} = 0,110$ Wh.km⁻¹kg⁻¹ et une durée de vie kilométrique $LCD = 300000$ km. La valeur de k_{elec} correspond à la un véhicule de 1,5 tonnes avec une consommation kilométrique moyenne sur l'année de 165 Wh.km-1. La voiture est alimentée par un moteur de 100 kW, on pourra en déduire $I_{batt,max}$ dans la contrainte III.7. Le pic de puissance P_{peak} est estimé pour une accélération de la voiture de 0 à 100 km.h-1 (27.78 m.s-1) en 10 secondes sur route droite sans pente. Le détail du calcul est donné en annexe D.2. On donne aussi en annexe les expressions empiriques utilisées pour le calcul du vieillissement (D.4.2). La profondeur de décharge à chaque cycle D_{od} est fixée à 50% pour un usage moyen. Finalement, le tableau V.7 résume l'ensemble des paramètres nécessaires à l'application du modèle développé dans le chapitre III.

TABLE V.7 – Paramètres fixés pour le problème *batterie* avant la résolution du modèle

Constante	Valeur	Unité
LCD	300000	km
k_{avg}	110	Wh.km ⁻¹ .kg ⁻¹
P_{motor}	100000	W
M_{base}	1200	kg
D_{od}	50	%
$M_{max,batt}$	700	kg
$V_{max,batt}$	0,5	m ³
$J_{max,batt}$	10000	€
GWP	10000	kg.co2-eq
AC	100	kg.So2-eq
EU	30	kg.PO4 ⁻⁻⁻ -eq
RD	2	kg.Sb-eq

V.4.1 Exploration de l'espace réalisable du problème *batterie*

Le modèle Minibex complet est donné dans l'annexe B.3. Comme pour le problème du moteur thermique, l'exploration partielle du problème doit d'abord nous permettre de comprendre l'espace réalisable disponible. On utilise l'algorithme AB&P avec la stratégie de sélection IDFS et la stratégie de bisection SSR. La précision est fixée à $\epsilon = 10^{-3}$ et le temps de calcul à 30 secondes. 2746 solutions sont obtenues sur les 2^{11} théoriques estimées lors de l'évaluation de la complexité au chapitre III. L'évolution des indicateurs est tracée sur les courbes V.11a, V.11b et V.11c.

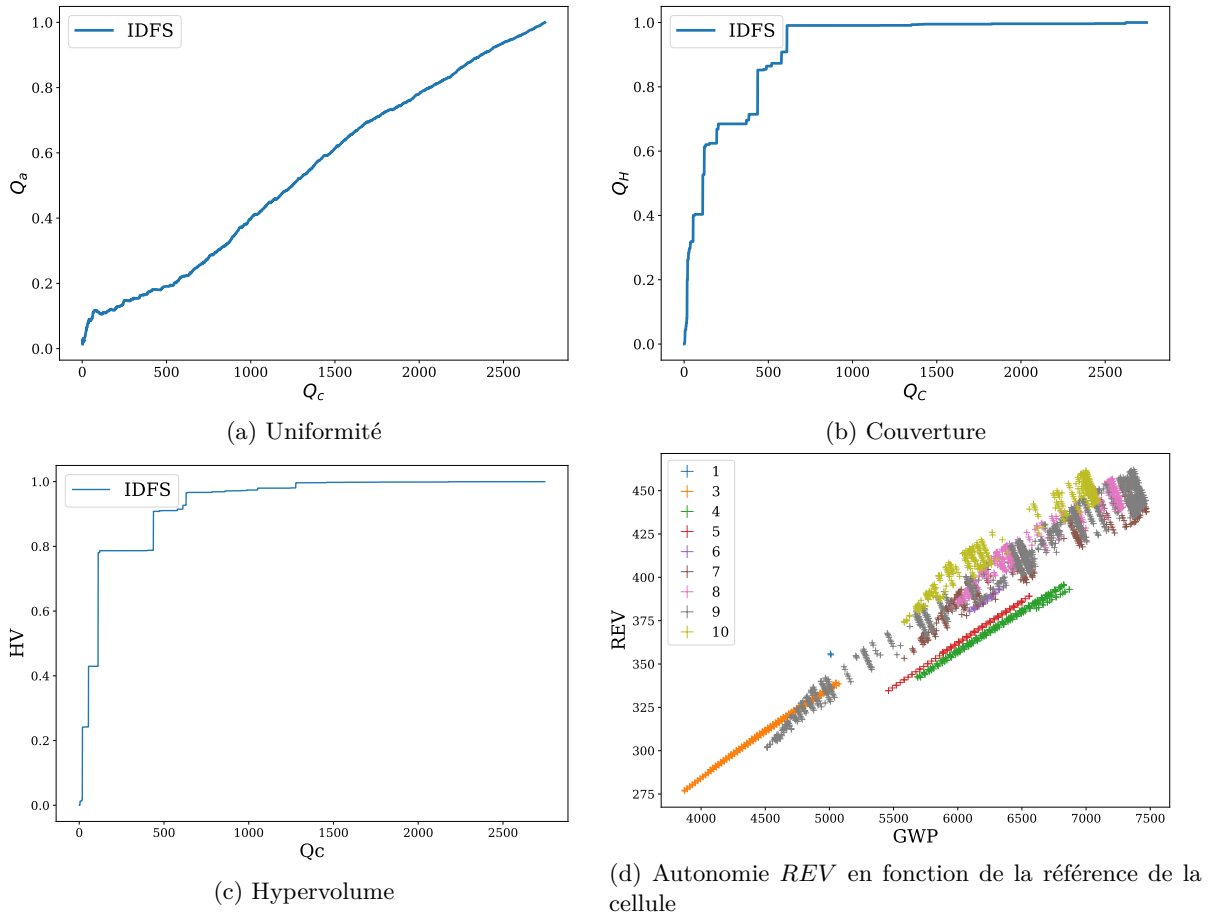


FIGURE V.11 – Résultats de l'exploration du problème *batterie* après $t = 30$ secondes et $Q_c = 2746$ solutions.

Les profils des indicateurs Q_h et HV montrent que l'on atteint 95% de la frontière de l'espace réalisable après quelques centaines de solutions. Le profil de Q_a est singulier avec un ralentissement de l'augmentation de l'uniformité sur l'intervalle $Q_c = [200, 600]$. Ce phénomène peut s'expliquer par le caractère fortement discret du problème *batterie* alors que la distance de Hausdorff utilisée par Q_a est plus adaptée aux domaines continus. La figure V.11d montre l'ensemble des solutions calculées dans l'espace des V_p REV et GWP . Les points sont différenciés selon leur référence n_{cell} dans le catalogue fournisseur. La cellule $n_{cell} = 2$ n'offre pas de solutions architecturales à notre problème. Les cellules cylindriques

($n_{cell} = \{7, 8, 9, 10\}$) permettent plus d'architectures différentes car le refroidissement passif donne un levier de conception supplémentaire.

V.4.2 Sélection de quelques architectures représentatives

Le résultat est réduit à un nombre de solutions raisonnable en utilisant notre méthode de sélection. Avec $\lambda_a = 0.01$ pour Q_a et $\lambda = 0$ pour Q_h et HV on obtient un sous-ensemble \mathcal{S}_{exp} de 204 architectures. Ces architectures sont ensuite représentées dans l'espace de conception et dans l'espace des V_p dans les GCP V.12. Pour ce problème, nous avons généré les graphiques avec Plotly² pour faciliter l'interaction et l'analyse des résultats. Sur cette figure, le paramètre (n_refrig) est la référence de système de refroidissement actif dans le tableau III.10 et la variable (n_cell) est la référence de cellule dans le tableau B.1.

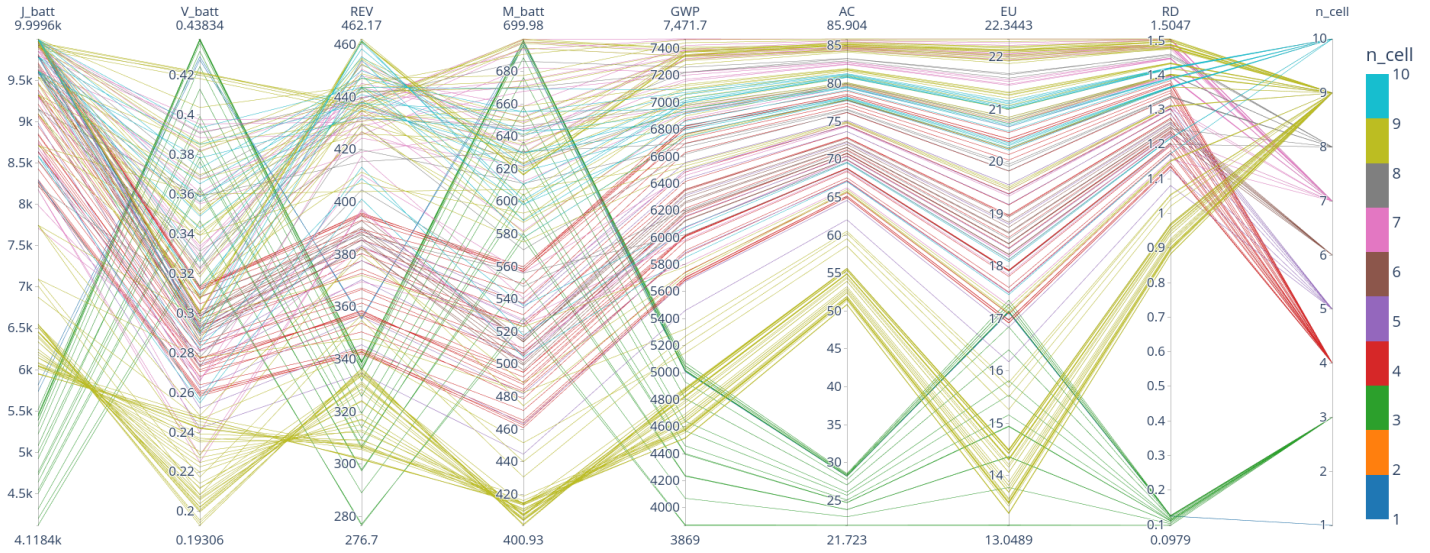
V.4.3 Analyse de l'espace réalisable

On utilise deux codes couleurs pour mettre en valeur les différentes références de cellules dans la figure V.12a et l'impact des émissions de CO2 dans la figure V.12b. L'échantillon \mathcal{S}_{exp} couvre tout l'espace réalisable et permet de reconnaître des tendances entre les alternatives. En interagissant avec les représentations, le concepteur peut parcourir le sous-ensemble de solutions. Par exemple, dans l'espace des V_p , la cellule prismatique $n_{cell} = 3$ donne les architectures les moins impactantes sur l'ensemble des impacts environnementaux, mais au prix d'un ratio autonomie/masse faible. La figure D.4 en annexe met en valeurs ces architectures dans l'espace de conception. De la même façon, seules les architectures composées des cellules cylindriques $n_{cell} = \{7, 8, 9, 10\}$ garantissent une autonomie moyenne REV supérieure à 400 km (GCP en annexe D.4). En revanche, ces batteries avec une grande autonomie nécessitent au moins 8000 € d'investissement et ont un impact important sur l'environnement. Finalement, dans la figure V.13 l'espace des V_p est restreint pour considérer les architectures qui offrent au minimum 300 km d'autonomie et ont un impact limité pour les quatre indicateurs ACV. Quelques packs batterie parmi ces solutions sont données dans le tableau V.8. À ce stade, le concepteur peut comprendre et justifier le choix de ces architectures en les positionnant dans l'espace réalisable exploré.

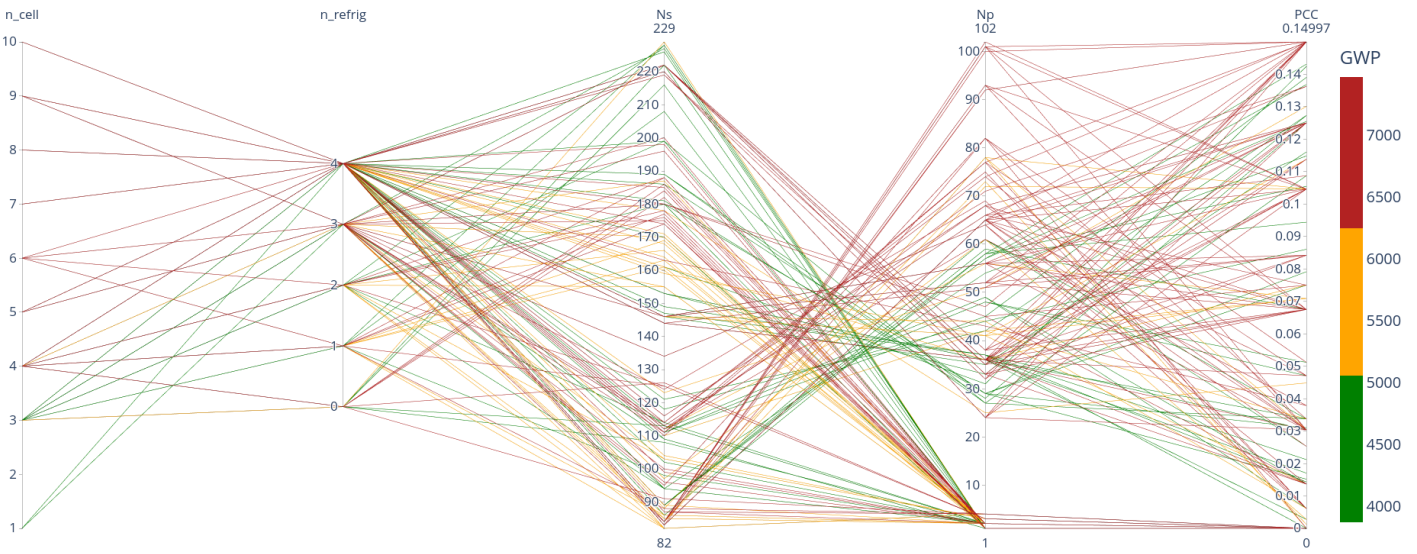
TABLE V.8 – Quelques architectures pertinentes issues de l'exploration partielle du problème *batterie*

n_{cell}	V_{co}				V_p							
	N_s	N_p	τ_{mcp}	n_{refrig}	J_{batt}	M_{batt}	V_{batt}	REV	GWP	AC	RD	EU
1	109	1	0	3	5735,69	695,23	0,42779	355,872	5007,95	28,184	0,12488	17,114
1	109	1	0	4	5815,69	698,30	0,42939	355,287	5012,05	28,205	0,12504	17,121
3	102	2	0	4	4998,09	632,18	0,39564	313,918	4568,89	25,690	0,11457	15,538
3	228	1	0	2	5442,57	699,13	0,43833	338,883	5051,88	28,435	0,12587	17,276
3	108	2	0	0	5080,33	657,97	0,41299	325,901	4808,07	27,051	0,12013	16,403
3	208	1	0	1	4922,17	636,67	0,39929	319,377	4640,48	26,102	0,11610	15,813
9	199	27	0,0337	4	6389,70	403,83	0,20656	327,388	4777,30	54,326	0,94670	14,160

2. <https://plotly.com/python/>



(a) GCP de l'espace des V_p



(b) GCP de l'espace des V_{co}

FIGURE V.12 – GCP de l'échantillon des 204 architectures sélectionnées après l'exploration partielle du CSP de la batterie

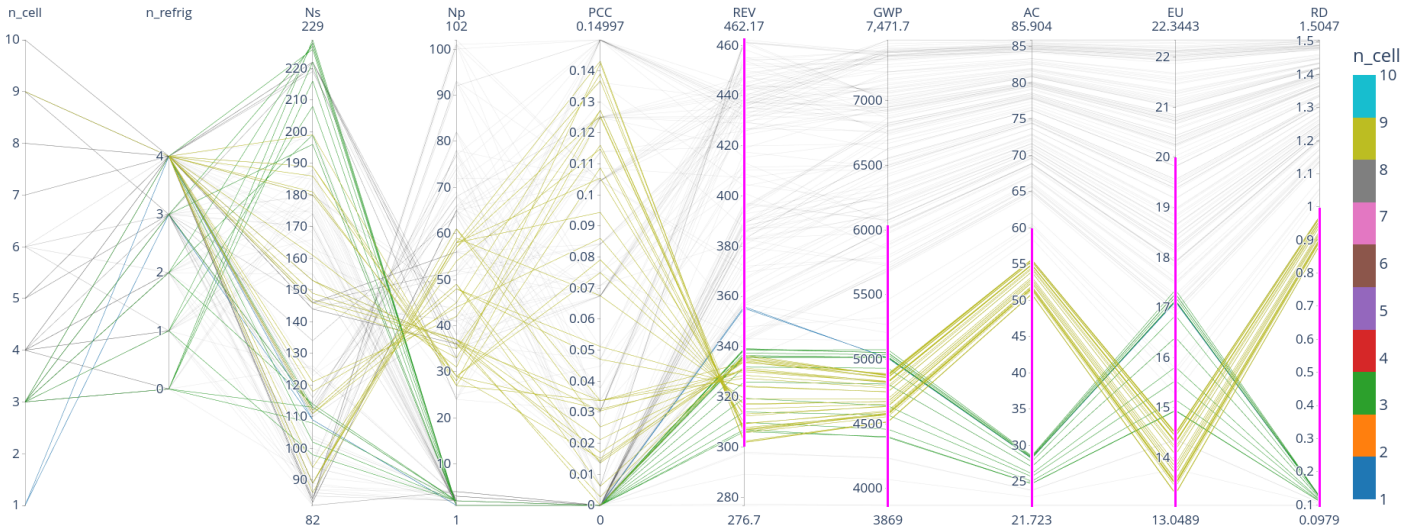


FIGURE V.13 – Restriction de l’espace réalisable à certaines plages de valeurs sur l’autonomie et les indicateurs ACV (barres verticales roses).

V.5 Bilan du chapitre

Dans ce chapitre, nous avons traité 3 problèmes de conception avec deux intentions. D’une part, proposer une méthode pour explorer partiellement et efficacement un espace de recherche. D’autre part, positionner l’approche d’exploration partielle par rapport à une approche plus classique basée sur des algorithmes d’optimisation connus.

Notre démarche est divisée en deux temps : exploration puis sélection. L’exploration consiste à générer un grand nombre de solutions bien réparties dans l’espace. Ce premier résultat permet d’apprécier les zones faisables. Le concepteur peut ensuite interagir avec le modèle pour concentrer la résolution sur une zone d’intérêt. Dans le problème *ice* nous avons pu identifier une zone proche des valeurs cibles des V_p et ajouter au problème initial une contrainte adaptée pour une seconde exploration. La complexité des problèmes et les limites des algorithmes AB&P ne permettent pas d’obtenir immédiatement le meilleur compromis entre diversité et nombre de solutions. Une étape de sélection est alors nécessaire pour dégager du résultat de l’exploration le sous-ensemble de solutions attendu. La qualité de l’échantillon final dépend de l’efficacité de la première exploration. Il est donc crucial d’utiliser les algorithmes d’exploration partielle les plus efficaces comme ceux présentés au chapitre IV.

Pour positionner notre méthode, nous avons comparé les résultats de l’exploration des modèles *mapse* et *ice* avec des approches d’optimisation multi-objectifs classiques. Pour chaque problème, les résultats de l’exploration offrent une meilleure image de l’ensemble réalisable. Cet aspect est important dans le contexte d’une étude de faisabilité. Les architectures obtenues sont mieux réparties, à la fois dans l’espace de conception et dans l’espace des objectifs. L’exploration a permis d’identifier quelques solutions dominées, mais pertinentes à considérer pour une conception détaillée. Cependant, nos algorithmes montrent leurs limites pour couvrir efficacement des frontières définies par des expressions mal supportées par les

opérateurs de contraction (par exemple sur le problème *mapse*). Dans un dernier temps, le problème *batterie* formalisé au III a été exploré pour montrer que l'exploration est une approche souple et interactive. Nous avons notamment couplé les résultats de l'exploration à des outils d'analyse. Le concepteur a alors accès à tous les éléments de compréhension pour justifier ses décisions avant la phase de conception détaillée.

Les perspectives de l'exploration partielle sont nombreuses. Au-delà de l'amélioration de l'efficacité des différents opérateurs (stratégie de sélection, stratégie de bisection, contracteurs), l'exploration donne la possibilité d'interagir efficacement avec le modèle. Par exemple en définissant des contraintes adaptées à la forme de l'espace de recherche. C'est aussi une approche plus explicite qui permet de réduire l'effet boîte noire des méthodes d'optimisation. Finalement, l'exploration partielle supporte bien les mécanismes "Start and Stop", c'est-à-dire arrêter et reprendre l'exploration à n'importe quel instant. L'étape suivante est le développement d'un outil interactif d'exploration pour la conception préliminaire.

CONCLUSION

Dans ce travail de recherche, nous avons développé de nouveaux outils pour la modélisation et la résolution de problème de dimensionnement en conception préliminaire. Il s'agissait, d'une part, de répondre à l'enjeu du nombre croissant d'exigences et de critères dans les cahiers des charges, et d'autre part de remettre l'expertise du concepteur au centre des choix architecturaux. Les thématiques principales abordées sont l'aide à la décision (*Decision Making*), la formalisation de modèle (*Modeling*) et l'exploration d'espace de conception (*Design Space Exploration*). Le premier chapitre (II) nous a permis de faire un état de l'existant et de préciser le cadre choisi pour cette thèse. Nous nous sommes intéressés aux modèles de dimensionnement gros grains pour la phase de conception architecturale, et plus particulièrement au paradigme des problèmes de satisfaction de contraintes numériques ou mixtes. Pour la résolution de ces problèmes, nous avons orienté nos travaux vers l'exploration partielle diversifiée, domaine encore peu étudié dans la littérature. Le but de l'exploration partielle diversifiée est d'obtenir un échantillon de solutions de taille raisonnable qui représente correctement chaque zone de l'espace réalisable. Le cas d'application principal de notre étude concerne la conception d'une batterie Li-ion pour un véhicule électrique de tourisme.

VI.1 Contributions et limites

Trois questions de recherches ont été soulevées dans l'introduction. Les chapitres III, IV et V répondent respectivement à chacune des questions. Les paragraphes suivants reprennent les éléments de réponses principaux et discutent de leurs limites.

VI.1.1 Comment formaliser un modèle mathématique adapté pour la conception préliminaire d'un système complexe ?

Le chapitre III propose une nouvelle méthode pour formaliser un modèle de dimensionnement dédié à l'étude de faisabilité en conception. Les différentes caractéristiques d'un modèle, définies dans la littérature, ont été reformulées et synthétisées dans la notion de pouvoir décisionnel. Cette notion permet d'estimer à la fois l'adéquation avec le concept et la validité mathématique et informatique. Le rôle et les méthodes de calculs des six caractéristiques décisionnelles (complétude, cohérence, exactitude, précision, complexité, étendue) ont été revus. La démarche globale s'appuie sur une modélisation en deux temps. Premièrement, la construction d'un modèle initial pour répondre spécifiquement au besoin principal. Deuxièmement, l'agrégation des modèles liés aux éléments complémentaires du concept. Pour guider ce travail, la composition du modèle a été anticipée en amont dans un tableau des spécifications du concept.

Ce tableau regroupe l'ensemble des constituants du modèle tels que prévu par le concept. La dernière étape consiste à implémenter le modèle complet et analyser son pouvoir décisionnel global. Notre méthodologie permet de formaliser un modèle dont la capacité à orienter les choix de conception est maîtrisée selon trois aspects : l'adéquation au concept, les caractéristiques de résolutions et la validité. La définition d'un modèle initial facilite sa réutilisation pour la formalisation d'un concept proche. La démarche a été éprouvée sur le concept de la batterie.

La méthodologie proposée est conçue pour guider le concepteur pendant la modélisation. Pour certains problèmes de conception, il peut être nécessaire d'adapter les étapes au contexte. Par exemple, il n'est pas toujours possible d'identifier clairement un problème principal et un modèle initial pour initier la démarche. Dans ce cas, il peut être pertinent de structurer l'agrégation autour d'un modèle complémentaire particulier. En pratique, l'expertise de l'utilisateur joue un rôle clé dans la réussite du processus de formalisation. La notion de pouvoir décisionnel repose sur des caractéristiques bien établies des modèles. Néanmoins, les mesures proposées pour évaluer ces grandeurs sont assez spécifiques au cadre de la programmation par contraintes, notamment le tableau de la complexité. Les expressions de l'étendue et de la complétude nécessitent une définition rigoureuse du concept et des paramètres du problème. Entre autres, il n'est pas toujours possible d'identifier les bornes des variables de conception. Finalement, le calcul de l'exactitude est largement dépendant des valeurs de référence disponibles. On pourra tester la démarche sur d'autres problèmes de conception.

VI.1.2 Comment explorer efficacement l'espace réalisable d'un CSP numérique ou mixte ?

La contribution globale du chapitre IV est d'introduire un cadre pour l'exploration partielle diversifiée comme outil d'aide à décision en conception préliminaire. Les algorithmes anytime branch and prune existants sont peu efficaces dès que la taille des problèmes augmente. Les trois stratégies proposées DLFS, DPFS et IDFS sont reliées à des propriétés constantes de l'arbre de recherche. Ce comportement appelé « statique » réduit significativement la complexité de calcul par rapport à des stratégies dynamiques comme DMDFS et MDFFS. Pour traiter des problèmes de conception avec ces algorithmes, nous avons identifié plusieurs aspects à adapter. En particulier, les opérateurs de tri et de bisection liés à des grandeurs réelles ont été adimensionnés, et une nouvelle définition de la précision a été proposée pour générer des sous-ensembles de solutions cohérents. D'autres éléments secondaires, mais impactants, ont été abordés comme la gestion des tables de contraintes, les méthodes de descente et le retour en arrière forcé. Dans l'ensemble, l'efficacité des AB&P dépend principalement du choix des stratégies de sélection et de bisection. Pour tenter d'y répondre, nous avons introduit trois indicateurs de qualité qui évaluent la diversité au sein d'un sous-ensemble de solutions. Ils nous ont d'abord permis d'analyser les comportements des stratégies de recherche possibles pour chaque combinaison de stratégie de sélection/stratégie de bisection sur un problème évolutif simple. Puis, nous avons pu identifier les algorithmes les plus prometteurs par une expérimentation globale sur un benchmark de problèmes de conception existants. En pratique, la complexité et l'hétérogénéité des modèles rencontrés demande d'adapter le solveur à chaque problème.

Nous avons noté quelques limites à notre cadre de résolution. Les stratégies de recherche proposées restent peu adaptatives et demandent des problèmes bien construits. En particulier, les problèmes avec des bornes mal définies sont difficilement solvables avec nos algorithmes et les modèles incomplets n'ont pu être inclus dans les expérimentations. De fait, le benchmark utilisé C.3 est limité à une trentaine de modèles simples. Les variables discrètes des problèmes mixtes ont été traitées avec des contraintes sur les tables de valeurs et les entiers. Cependant, les opérateurs de tri et de divisions et les indicateurs de qualité utilisent des distances définies sur des ensembles continus. La gestion des domaines finis peut être améliorée en intégrant des distances spécifiques sur variables discrètes. En particulier lorsque les valeurs d'un domaine correspondent à des caractéristiques de composants sur étagère. Finalement, les indicateurs sur la diversité peuvent être améliorés. Par exemple, le volume de l'enveloppe Q_h donne une vision caricaturale de la frontière de l'espace réalisable. L'enjeu est de trouver des mesures plus précises qui ne demandent pas des temps de calcul trop grands.

VI.1.3 L'exploration partielle diversifiée est-elle un outil pertinent pour traiter des problèmes de conception architecturale ?

Enfin, l'objectif du chapitre V était de proposer une démarche d'exploration partielle sur différents cas d'application. Les algorithmes introduits au chapitre IV permettent une exploration partielle efficace de l'espace de recherche. Cependant, il persiste des phases d'intensification pendant la résolution. Le premier résultat de l'exploration partielle ne donne donc pas immédiatement le compromis souhaité entre diversité et cardinalité. Néanmoins, en explorant l'espace avec suffisamment de solutions, on peut atteindre la diversité recherchée. Le résultat peut ensuite être réduit intelligemment pour que le concepteur ait un nombre d'architectures raisonnable à analyser. La méthode d'exploration/sélection proposée a été éprouvée sur le cas d'application d'un actionneur électromagnétique. Nous avons pu construire la représentation souhaitée de l'espace réalisable. Le deuxième exemple avait pour rôle d'illustrer les possibilités d'interactions avec l'utilisateur lors de la résolution. Quelques solutions pertinentes ont pu être générées dans une zone précise de l'espace de recherche. Pour ces deux premiers cas d'étude, les résultats ont été positionnés avec les solutions des problèmes multi-objectifs équivalents. Enfin, nous avons appliqué cette démarche à notre modèle de batterie li-ion. Ce dernier cas nous a permis d'illustrer comment analyser des résultats en considérant à la fois l'espace de conception et l'espace des objectifs avec des outils de post-traitement efficaces. L'exploration partielle autorise une meilleure compréhension de l'espace réalisable global et permet d'identifier des solutions proches de l'optimale dont les caractéristiques restent intéressantes pour la conception préliminaire.

L'algorithme d'exploration utilisé dans ce chapitre a montré quelques limites. Par exemple, dans le problème du moteur à aimants permanents la frontière du front de Pareto était faiblement recouverte par l'échantillon généré. Dans certains cas, il peut s'avérer difficile pour le concepteur d'identifier les zones mal explorées de l'espace. Il peut être pertinent de mettre en place une analyse des zones de l'espace de recherche faiblement explorées pour faciliter ce travail. On peut aussi noter que la sémantique de Minibex n'est pas adaptée à la résolution de problèmes de dimensionnement. En particulier, les notions

de composants, flux ou fonctions ne sont pas présents. La diffusion de nos outils nécessitera quelques étapes supplémentaires d'implémentation et de mises en forme.

VI.2 Perspectives

1. La notion de pouvoir décisionnel pourra être éprouvée sur d'autres cas d'applications pour déterminer plus précisément ses implications sur la réduction et l'adaptation des modèles. Par exemple, pour un modèle donné, comment l'évaluation de sa complexité peut influencer le choix du solveur et les paramètres de résolution. De la même manière, il faut clarifier le rôle des différentes caractéristiques du pouvoir décisionnel sur l'analyse des solutions et sur les décisions des concepteurs. On peut envisager le développement d'un indice de confiance dans les décisions prises à partir d'un modèle dont le pouvoir décisionnel est connu.
2. De nombreuses perspectives existent pour améliorer les stratégies des algorithmes AB&P. Nous avons envisagé le développement de stratégies de recherche qui adaptent la fonction de tri statique en fonction de seuils sur la diversité atteinte. Par exemple, maximiser le volume de l'espace de recherche dans un premier temps, puis intensifier la diversité à l'intérieur de l'espace réalisable dans un second temps. Il est aussi possible de construire des fonctions de tri pour différencier la diversité dans l'espace de conception et dans l'espace des objectifs.
3. Quatre indicateurs ont été utilisés pour évaluer la qualité de nos résultats : Q_a , Q_h et Q_d pour la diversité globale, et HV pour la diversité dans l'espace des objectifs. Les indicateurs de qualité ont été largement étudiés en optimisation multi-objectifs [161, 104, 4, 89]. Il peut être pertinent de s'appuyer sur ces travaux pour approfondir ces outils. Par exemple, en utilisant différentes définitions de la distance entre les solutions (Euclidienne, Minkowski). La discussion du chapitre V a aussi mis en évidence la nécessité de pouvoir évaluer la diversité sur des domaines discrets.
4. L'algorithme B&P utilisé ne tient pas compte des boîtes intérieures ou de la preuve numérique. Cette question est complexe à traiter pour des problèmes mixtes non-linéaires. Cependant, dans certains cas, il peut être pertinent d'identifier rapidement les boîtes intérieures réalisables dans un espace de recherche. Plusieurs travaux peuvent servir de point de départ [185, 63, 64, 109]. Dans une autre mesure, nos algorithmes pourraient être combinés à des méthodes de recherche locales pour accélérer la convergence. Par exemple, lorsque les opérateurs de contractions comme $HC4$ deviennent inefficaces à partir d'une certaine profondeur de l'arbre de recherche.
5. Dans le chapitre V, nous avons présenté une méthode pour identifier un échantillon de solutions correctement réparties dans l'espace de recherche. D'autres approches de post-traitement peuvent s'avérer intéressantes pour participer à l'analyse des résultats de l'exploration. Les méthodes de clustering sont de bons candidats : k-means [120, 150], grid [56, 191] ou autre [122, 145]. Le clustering permettrait d'identifier clairement les volumes de l'espace réalisable représentés par les solutions de

l'exploration.

6. Nous avons vu que le langage Minibex n'est pas adapté au traitement de problème de dimensionnement. De plus, le projet IBEX n'est plus maintenu depuis 2022. Un travail conséquent d'adaptation et d'implémentation de nos contributions dans un support dédié à la conception peut être envisagé. Par exemple avec le langage DEPS [199].
7. Enfin, la perspective portée par le projet OSCAR est la prise en compte du cycle de vie dans le processus de conception. En introduisant la notion d'*architecture de cycle de vie*, nous pouvons envisager d'utiliser notre approche pour générer des cycles de vie représentatifs pour un concept donné. L'objectif global est de mettre en parallèle l'exploration des architectures du produit et l'exploration des cycles de vie correspondant. Le livrable doit être une méthodologie outillée, exhaustive et rigoureuse d'éco-conception. Le processus correspondant devra être compréhensible pour l'ensemble des parties prenantes. Aussi, la formalisation de modèles de dimensionnement pour les architectures de cycles de vie est un domaine inédit. Il implique de faire dialoguer deux domaines différents : l'Analyse de Cycle de Vie et l'Ingénierie Système.

VI.3 Contributions

Article de conférence

- de Latour, T., Chenouard, R., Granvilliers, L., 2021. Stratégies de résolution anytime de problèmes de satisfaction de contraintes numériques. *Journées Francophones de Programmation par Contraintes (JFPC)*, p. 10.

Abstract : A Branch-and-Prune algorithm computes a paving of the solution set of a numerical constraint satisfaction problem to a given precision. An anytime algorithm prematurely stops the computation to a given criterion (for example a maximum running time). This article revises anytime Branch-and-Prune algorithms and introduces two new search strategies and three quality indicators to compare them. The evaluation of their complexity and an experimental analysis enabled to prove the efficiency of those new strategies.

Article de journal

- de Latour, T., Chenouard, R., Granvilliers, L., 2023. Partial design space exploration strategies applied in preliminary design. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. p. 16.

Abstract : During preliminary phases in product design, on the basis of strong physical hypotheses (e.g. isotherm, steady state), physical and functional requirements can be expressed as coarse-grained constraint-based models on a few degrees of freedom, possibly including several design criteria to optimize. Such models are usually handled by multi-objective optimization solvers in order to find design solutions giving the best trade-offs between design criteria. Another approach developed in this paper is to partially explore all the areas of the design space using an anytime interval branch-and-prune algorithm called IDFS such that the design criteria are converted into so-called -constraints. The expected result is a sample of solutions diversified in both the objective space and the design space. Several quality indicators are introduced in order to measure this diversity and compare IDFS with two state-of-the-art multi-objective optimization solvers NSGA-II and NSGA-III on three real-world case studies. The results show that IDFS is able to identify new close-to-optimal designs and permits a better understanding of the design space. This framework provides a promising alternative tool for decision making, in particular for integrating interaction in the preliminary design process.

ANNEXE CHAPITRE 1

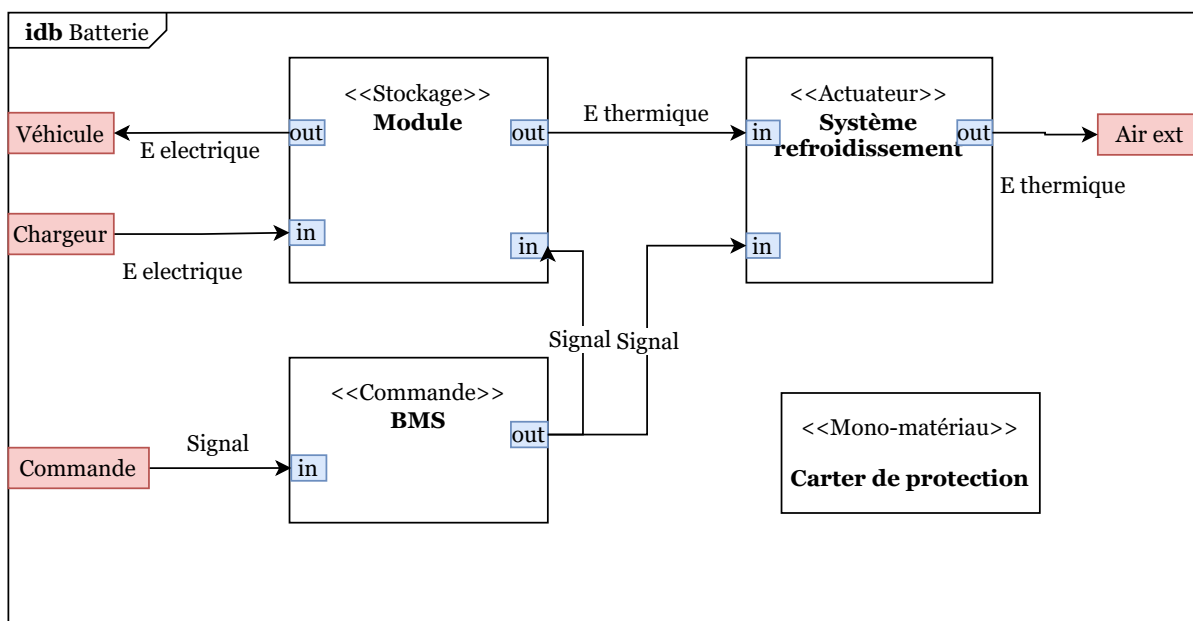


FIGURE A.1 – Modélisation de l'état interne de la batterie (*batterie*) avec le digramme SysML IBD

ANNEXE CHAPITRE 2

B.1 Agrandissement des BPMN du processus de formalisation

B.2 Catalogue de références de cellules li-ion LFP et NMC pour le problème batterie

TABLE B.1 – Catalogue fournisseur de cellule Li-ion pour véhicule électrique. (Prism = prismatique; Cyl = cylindrique.)

	Caractéristiques	Type	Form	C_c	U_c	V_c	M_c	R_{int}	U_{max}	U_{min}	I_{max}
n_{cell}	Référence	-	-	Ah	V	l	kg	mΩ	V	V	C-rate (h^{-1})
1	EVE LF230	LFP	Prism	230	3,2	1,94	4,10	0,3	3,65	2,5	2,00
2	EVE LF150	LFP	Prism	150	3,2	1,35	2,95	0,34	3,65	2,5	2,00
3	EVE LF105	LFP	Prism	105	3,2	0,96	1,98	0,5	3,65	2,5	3,00
4	CATL NMC 62	NMC	Prism	62	3,7	0,41	0,96	0,8	4,2	2,75	1,00
5	CATL NMC 116	NMC	Prism	116	3,7	0,72	1,72	0,7	4,2	2,75	1,00
6	CATL NMC 132	NMC	Prism	132	3,7	0,84	1,88	0,4	4,2	2,75	1,00
7	NCR18650GA	NMC	Cyl	3,45	3,6	0,02	0,05	38	4,2	2,5	2,90
8	LG CHEM MJ1	NMC	Cyl	3,5	3,65	0,02	0,05	40	4,2	2,5	2,86
9	EVE 18650	NMC	Cyl	3,2	3,6	0,02	0,046	30	4,2	2,5	3,13
10	LG INR21700 M50	NMC	Cyl	5	3,63	0,0246	0,069	0,25	4,2	2,5	3,2

B.3 Modèle minibex du problème de conception de batterie décrit dans [137]

```
constants
```

```
Ts = 49;
```

```
Tl = 52;
```

```
M_base = 1200; //
```

```
Kelec = 105; // Wh/km/kg weight power consumption
```

```
Pcharge = 150000;
```

```
//GWP
```

```
Pteff = 90; //Plug-to-Tank efficieNsy q_PtT
```

```
INTEff = 90; //Internal efficieNsy n_ri
```

```
Gelec = 0.0855; //GWP value for the production of 1 kW.h of electricity
```

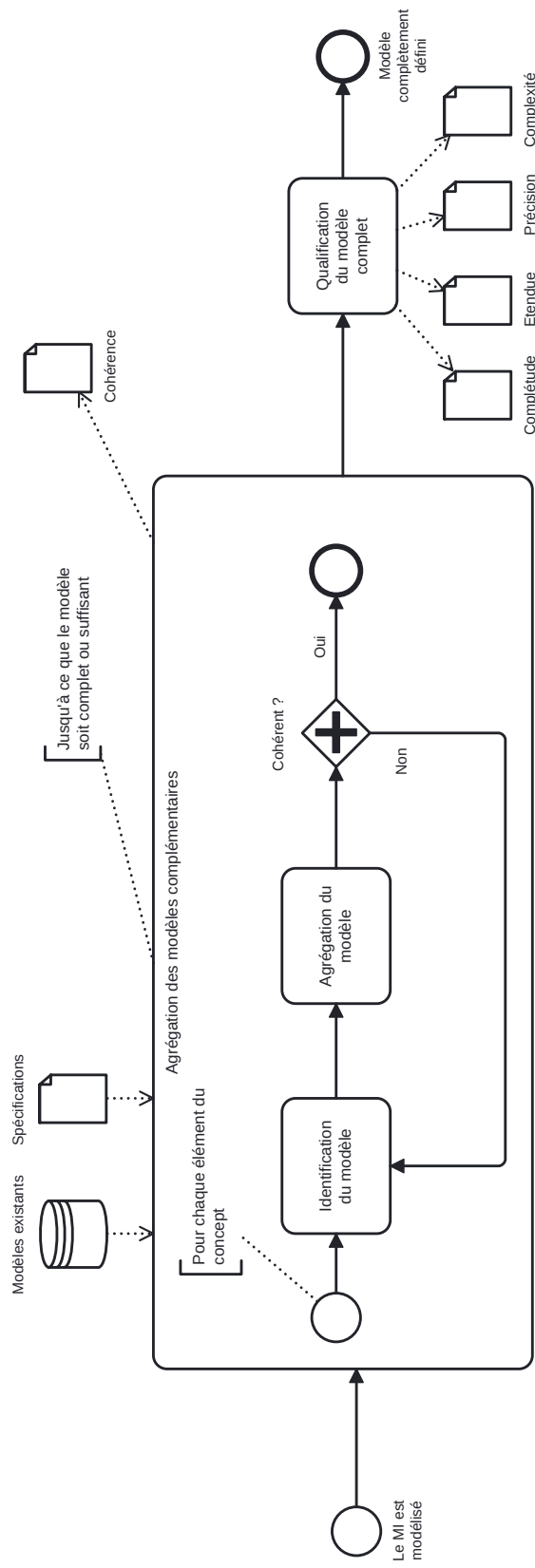


FIGURE B.1 – Phase 3 d'agrégation des modèles complémentaires (MC)

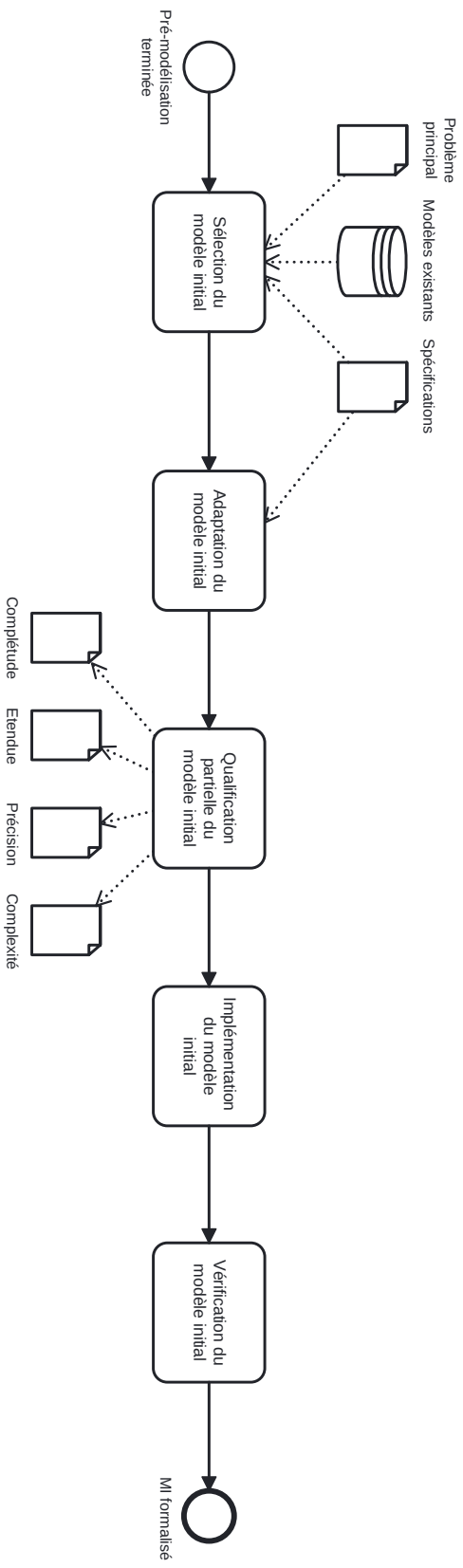


FIGURE B.2 – Phase 2 de formalisation du modèle initial

```

//
MANUF_lfp = 8.33; // GWP value for the manufacturing of 1 kg of nmc cell
MANUF_nmc = 17.12; // GWP value for the manufacturing of 1 kg of lfp cell
Dod = 0.5;

// Thermal characteristics
Cp_cell_nmc = 1040; // heat capacity of NMN cell J/kg.K
Cp_cell_lfp = 1130; // heat capacity of LFP cell J/kg.K

//cost
L = 189000; // latente heat of the PCM
Cp_pcc = 2100; // Heat capacity of the CPM
rho_cc = 912; // density of the PCM

//WLTP
dt = 3600; // cycle duration

variables

// design variables
n_cell in [1,10];
Ns in [1,2000]; // number of battery cells connected in series
Np in [1,2000]; // number of battery cells connected in parallel (Module)
t_mpc in [0,0.15]; // proportion of MPC
n_ref in [0,4];

// cell specifications
Cc in [1,290]; // capacity of the battery cell Cbc [Ah]
Vc in [0,10]; //volume (L)
Type in [0,1]; // NMC/LFP
Form in [0,1]; // NMC/LFP
Ucmin in [2.5,2.75]; // Minimum voltage
Ucmax in [3.6,4.2]; // Maximum voltage
Uc in [3.2, 3.7]; //Nominal Voltage
Mc in [0,10]; // Kg mass cell
Rint in [0,100]; // Internal resistance of the cell (mOhm)
Imax in [0,500]; // Peak discharge current (h-1)

// cooling system specs
qcool_max in [0,6000];
add_mass in [0,7];
add_vol in [0,2.8];
add_cost in [0,200];

// criteria variables
Jbatt in [0,10000]; // battery price
Mbatt in [0,600]; // mass battery
Vbatt in [0,0.4]; // mass battery
REV_end in [250,1000];
E_batt in [30000,200000];
GWP in [0, 10000];

```

```

LCD_end [250000,500000]; // - Time in cycle
Ncy in [1000,3000];
Qcy in [0,0.1];
T_c3 in [0,35];

/* Resolutions parameters

design
n_cell, Ns, Np, t_mpc, n_ref

criteria
Jbatt, Mbatt, Vbatt, REV_end, E_batt, GWP, LCD_end, Ncy

// Tables constraints

// Table for cells
table
n_cell, Cc, Uc, Vc, Mc, Type, Rint, Umax, Umin, Imax, Form
1, 230, 3.2, 1.944, 4.10, 1, 0.3, 3.65, 2.5, 460, 1
2, 150, 3.2, 1.35, 2.95, 1, 0.34, 3.65, 2.5, 300, 1
3, 105, 3.2, 0.956, 1.98, 1, 0.5, 3.65, 2.5, 315, 1
4, 62, 3.7, 0.414, 0.96, 0, 0.8, 4.2, 2.75, 124, 1
5, 116, 3.7, 0.721, 1.72, 0, 0.7, 4.2, 2.75, 232, 1
6, 132, 3.7, 0.842, 1.88, 0, 0.4, 4.2, 2.75, 264, 1
7, 3.45, 3.6, 0.017, 0.05, 0, 35, 4.2, 2.5, 10, 0
8, 3.5, 3.65, 0.017, 0.05, 0, 38, 4.2, 2.5, 10, 0
9, 3.2, 3.6, 0.017, 0.046, 0, 30, 4.2, 2.5, 10, 0
10, 5, 3.63, 0.0246, 0.069, 0, 25, 4.2, 2.5, 16, 0

// Table for active cooling system
table
n_ref, qcool_max, add_mass, add_vol, add_cost
0, 0, 0, 0, 0
1, 500, 2, 0.8, 30
2, 1000, 3, 1.2, 80
3, 3000, 5, 2, 120
4, 6000, 7, 2.8, 200

end
*/

constraints

Ppeak = (M_base+Mbatt)*(27.8^2)/8.1;

// Layout
integer(n_cell);
integer(n_ref);
integer(Ns);
integer(Np);

// Energy Performances

```

```

E_batt = Ns*Np*Cc*Uc;
E_batt >= 30000; // Energy Battery c1

// Peak Performances
Imax*Ns*Np*Uc >= Ppeak;
Ns*Uc <=800; // c5
Ns*Uc >=350; // c3
Ns*(Ucmin-Rint*(Ppeak/(Ns*Np*Uc))/1000/Np) >= 34; //c4

// Weight
Mbatt = Ns*Np*Mc*(1+t_mpc)+add_mass; //C5

// Volume
Vbatt = Ns*Np*Vc/1000+t_mpc*Ns*Np*Mc/rho_cc+add_vol/1000;
Vbatt <= Vol_max; //C6
// Autonomy
REV = Ns*Np*Cc*Uc/(Kelec*(M_base+Mbatt)/1000);
REV_end = REV*(0.9-Qcy);
REV >= 300;

// Life duration
LCD = Ncy*(REV+REV_end)*(Dod/100)/2;
LCD_end = Ncy*(REV+REV_end)*(Dod/100)/2;
LCD >= 300000;

// Price
Jbatt = (1-Type)*(Form*115+(1-Form)*100)*E_batt/1000 + Type*70*E_batt/1000 + add_cost;

// t_mpc = 0 for prismatic or pouch cells
t_mpc <= (1-Form)*0.15;

// // Thermal model at c/2 for ageing
Cp = (Ns*Np*Mc*Cp_cell_nmc+t_mpc*Ns*Np*Mc*Cp_pcc)/Mbatt;
Pcell_c2 = E_batt/2/(Ns*Np);
Icell_c2 = Pcell_c2/Uc;
qcell_c2 = (Icell_c2)^2*(Rint/1000);
qcell_c2*Ns*Np*(2*dt) - (Mbatt*Cp)*dTmax <= qcool_max*2*dt;

// // Thermal model at sustained speed (C/3)
Pcell_c3 = E_batt/3/(Ns*Np);
Icell_c3 = Pcell_c3/Uc;
qcell_c3 = (Icell_c3)^2*(Rint/1000);
H = (qcell_c3*Ns*Np)*(3*dt)/Mbatt + Cp*Tamb;
H*(1-Form) <= d1*Cp*Ts ;
T_c3 = (H/Cp)*(1-Form);
T_c3 <= 35; //C12

// // Thermal model for fast charging
Pcell_peak = Pcharge/(Ns*Np);
Icell_peak = Pcell_peak/Uc;
qcell_peak = (Icell_peak)^2*(Rint/1000);
Qpack_peak = qcell_peak*Ns*Np-qcool_max;

```

```

Tamb + Qpack_peak*(dt*E_batt/Pcharge)/(Mbatt*Cp) <=45;

//GWP
E_batt_gwp = E_batt/REV*0.3*Mbatt*LCD/((Pteff/100)*(M_base + Mbatt));
E_ri_gwp = (1-INTeff/100)*(E_batt)*LCD/REV;
GWP = E_batt_gwp*Gelec/1000+E_ri_gwp*Gelec/1000+Mbatt*(Type*MANUF_lfp+(1-Type)*MANUF_nmc);

//Ageing
T_c2 = Tamb + (qcell_c2*Ns*Np-qcool_max)*(2*dt)/(Mbatt*Cp);
T =max(Tamb,T_c2)+273.15;
Qcy = Type*(30330*exp((-31500)/(8.314*(T)))*((Dod*Ncy*Cc)^(0.552)))/100 +
(1-Type)*((137+420+9610*(0.75-Dod)^3)*exp(-22406/(8.314*T))*(Dod*Cc*Ncy)^0.48)/100 ;
Qcy < 0.1; // C9/

end

```

ANNEXE CHAPITRE 3

C.1 Résultats de l'exploration des problèmes des m boules sous la forme d'un CSP mixte

	DMDFS	DPFS	DLFS	IDFS	RDFS
nombre de boules	93,6	93,8	93,8	61,2	1,3
temps (s)	2,39	1,42	1,41	1,42	0,2128

TABLE C.1 – Nombre de boules et temps de calcul moyen pour le problème des 100 boules ($n \in (1, \dots, 15)$) (forme CSP mixte IV.20)

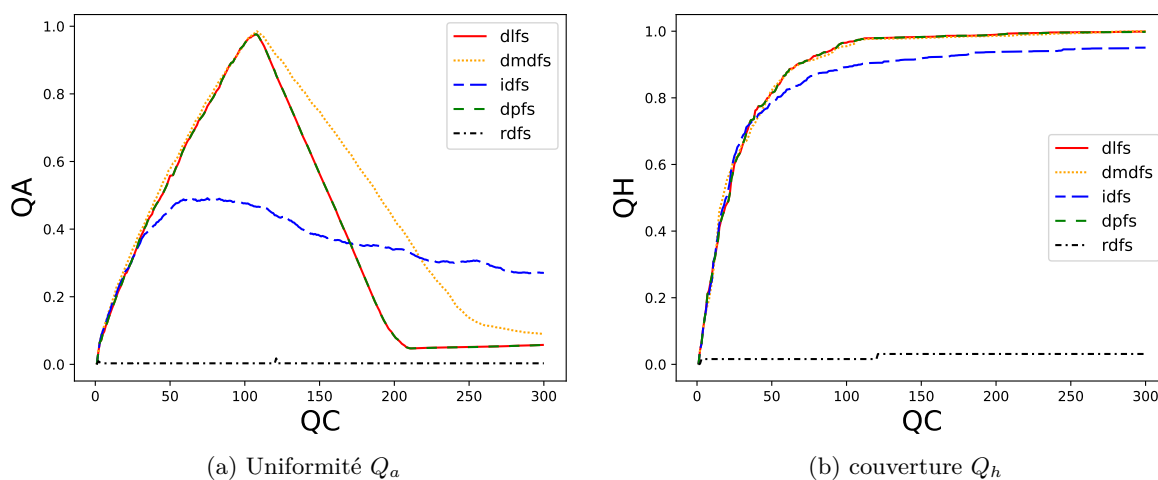


FIGURE C.1 – Profil de performance pour Q_a sur l'ensemble des problèmes des 100 boules (forme CSP mixte IV.20)

	DMDFS	DPFS	DLFS	IDFS	RDFS
Écart Type	1,84	0,41	0,41	3,48	28,6

TABLE C.2 – Écart type à la moyenne du nombre de solutions par boules, pour 300 solutions réparties dans 100 boules (forme CSP mixte IV.20)

C.2 Problèmes du benchmark

TABLE C.3 – Benchmark de 30 problèmes de conception (num = numérique; mix = mixte)

Nom	Vco	Contraintes	Type	Contracteur	Descente	χ	Source
2bartruss	3	3	num	<i>HC4/ACID</i>	L/F	-	[22]
4bartruss	4	10	num	<i>HC4/ACID</i>	L/F	-	[22]
<i>mapse</i>	5	15	num	<i>HC4/ACID</i>	L/F	-	[99, 115]
alkyl	7	14	num	<i>HC4/ACID</i>	L/F	-	gamslib
beam_huy	4	7	num	<i>HC4/ACID</i>	L/F	-	[80]
beam_osy	4	3	num	<i>HC4/ACID</i>	L/F	-	[130]
bearing	8	13	num	<i>HC4/ACID</i>	L/F	-	gamslib
bobine	5	6	mix	<i>HC4/ACID</i>	L/F	-	
bolt	10	14	mix	<i>HC4/ACID</i>	L/F	-	[198]
brushless_10	11	6	num	<i>HC4/ACID</i>	L/F	100	[17]
brushless_6	6	6	num	<i>HC4/ACID</i>	L/F	100	[17]
carside	7	10	num	<i>HC4/ACID</i>	L/F	-	[87]
chance	3	5	num	<i>HC4/ACID</i>	L/F	-	gamslib
chem	7	8	num	<i>HC4</i>	L/F	-	gamslib
disbrake	4	7	num	<i>HC4</i>	L/F	-	[146]
refrigeration	14	15	num	<i>HC4/ACID</i>	L/F	-	gamslib
fuel	9	19	mix	<i>HC4</i>	L/F	-	gamslib
gear	5	6	mix	<i>HC4/ACID</i>	L/F	-	[197]
house	4	8	num	<i>HC4/ACID</i>	L/F	-	gamslib
<i>ice</i>	6	15	mix	<i>HC4/ACID</i>	L/F	-	[134]
launch	19	18	num	<i>HC4/ACID</i>	L/F	100	gamslib
reducer	7	9	mix	<i>HC4</i>	L/F	100	[67]
sample	4	3	num	<i>HC4/ACID</i>	L/F	-	gamslib
screw_plb	8	10	mix	<i>HC4/ACID</i>	L/F	-	[134]
ship	11	18	num	<i>HC4/ACID</i>	L/F	100	gamslib
spring	3	7	mix	<i>HC4/ACID</i>	L/F	-	gamslib
transfo_secu	11	19	mix	<i>HC4</i>	L/F	100	[176]
vibration	5	5	num	<i>HC4/ACID</i>	L/F	-	[195]
water	3	10	num	<i>HC4/ACID</i>	L/F	-	[39]
<i>batterie</i>	5	24	mix	<i>HC4/ACID</i>	L/F	-	III

ANNEXE CHAPITRE 4

D.1 Fonctionnement des algorithmes NSGA

D.1.1 Les algorithmes évolutionnistes

Lorsque les objectifs sont considérés séparément dans un MOP, le problème ne peut être résolu de manière déterministe, en particulier au-delà de deux objectifs. On utilise plutôt des métaheuristiques [167]. Il s'agit d'approcher le front de Pareto dans son ensemble avec une population d'individus bien réparties. Ces algorithmes sont appelés évolutionnistes, ils font évoluer la population d'individus à l'aide d'opérateurs de sélection et de variation. Les individus sont évalués et comparés d'une génération à l'autre pour assurer une convergence vers les solutions optimales. Les mécanismes des différentes heuristiques existantes sont souvent inspirées de phénomènes biologiques réels. On distingue 5 familles d'algorithmes évolutionnistes : les algorithmes génétiques (GA), les stratégies d'évolutions, la programmation évolutionnaire, l'évolution différentielle et la programmation génétique.

D.1.2 Les algorithmes génétiques

Les GAs sont inspirés du principe de la sélection naturelle, ce sont les plus approches les plus utilisées dans le domaine de la conception. Un GA fait évoluer la population d'individu à travers une succession de générations (figure D.1). D'une génération à une autre, les individus peuvent subir trois opérations : sélection, croisement et mutation. L'étape de sélection permet d'identifier les meilleurs individus. Le croisement mélange les propriétés des individus choisis. La mutation est une altération aléatoire d'une des caractéristiques d'un individu. Un GA est caractérisé par les mécanismes de ses trois opérateurs.

NSGA-II

L'algorithme NSGA-II (Non-dominated Sorting Genetic Algorithm) a été introduit par [38]. La figure D.2 décrit le fonctionnement spécifique de NSGA-II. Une des particularités de l'algorithme est d'utiliser la *crowding distance* (Distance de Manhattan entre les individus) pour maintenir une population bien répartie sur le front de Pareto. Après la phase de croisement et de mutation, les populations mère (P_t) et fille (Q_t) sont regroupées et triées selon leur dominance (distance aux objectifs). Ensuite, la meilleure moitié du regroupement est sélectionnée pour la génération suivante. Les individus indécis (F_3 dans la figure D.2) sont alors choisis à partir de la *crowding distance*.

Le parcours de l'espace de recherche par NSGA-II est déterminé par plusieurs paramètres, les principaux sont la taille de la population et le nombre maximal de générations. Il faut ajuster ces deux quantités

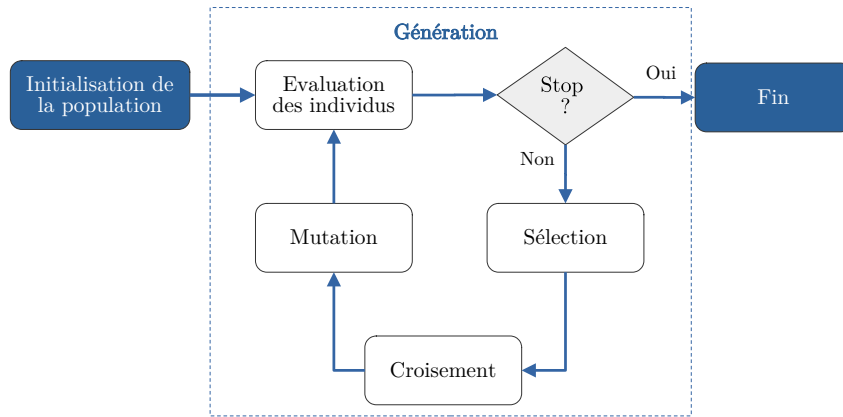


FIGURE D.1 – Fonctionnement générique d'un algorithme génétique

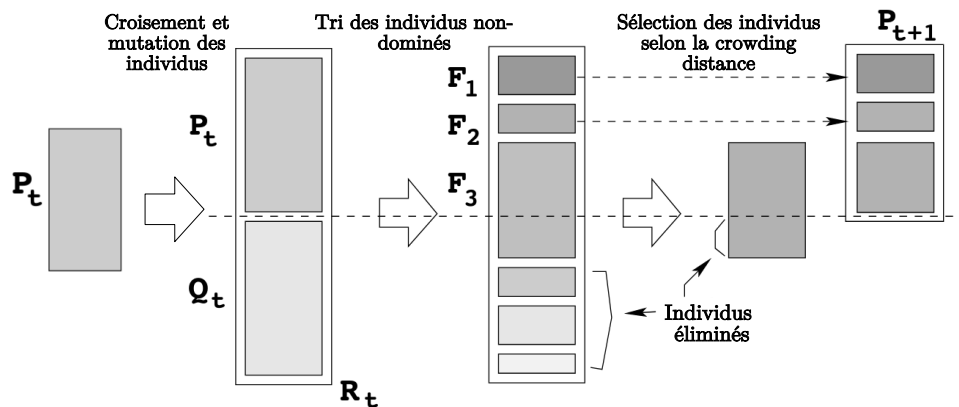


FIGURE D.2 – Schéma du fonctionnement de NSGA-II [38]

en fonction de la complexité du problème (nombre de variables et nombres d'objectifs). D'autres paramètres peuvent s'avérer importants pour faciliter la convergence : la tolérance, le taux de mutation, la probabilité de croisement et les indices de distribution de mutation et de croisement. Les méthodes pour choisir ces paramètres ont été évoqués dans [192].

NSGA-III

Lorsque le nombre d'objectifs dépasse 2, NSGA-II rencontre des difficultés pour converger tout en gardant une bonne diversité entre les solutions pour représenter tout le front de Pareto. Il est alors préférable d'utiliser NSGA-III pour les problèmes avec entre 3 et 15 objectifs [87]. NSGA-III utilise un mécanisme de diversification des solutions dans l'espace des objectifs différent de NSGA-II. La méthode de la *crowding distance* est remplacé par un mécanisme de discrimination à partir de directions créées entre la solution idéale et un ensemble de points de référence. Ces points de référence peuvent être définis par l'utilisateur ou générés au moyen d'une approche spécifique comme Das and Denni's [34]. Ainsi, lors de l'étape de sélection des individus au sein de F_3 dans le schéma D.2, les individus les plus proches des directions sous-représentées en nombre de solutions sont choisis en priorité. À la fin de la résolution, chaque direction doit être représentée par une solution non-dominée. Le nombre de points de référence n_{ref} est un paramètre supplémentaire de la résolution qui peut être estimé par la formule donnée dans [87] (équation D.1).

$$n_{ref} = \binom{M + p - 1}{p} \quad (\text{D.1})$$

Où M est le nombre d'objectifs et p un entier donné. On choisit ensuite la taille de la population comme le multiple de 4 immédiatement supérieur à n_{ref} . Les algorithmes évolutionnistes ont largement été utilisés pour la résolution de modèles gros grains multi-objectifs dans plusieurs domaines d'application [67, 18, 192, 79, 167, 46].

D.2 Problème *mapse*

CSP : $X = (e, E, la, \beta, P)$

$$[e] = [0.00001, 0.005]$$

$$[E] = [0.001, 0.05]$$

$$[la] = [0.003, 0.05]$$

$$[\beta] = [0.8, 1]$$

$$[P] = [1, 10]$$

Sujet à :

$$c_1 : D - 2(la + e) \geq 0$$

$$c_2 : 1e5 \leq J_{cu} \leq 1e7$$

$$c_3 : 0.01 \leq D \leq 0.5$$

$$c_4 : 1 \leq \lambda \leq 2.5$$

$$c_5 : 0.01 \leq K_f \leq 0.5$$

$$c_6 : 0.001 \leq C \leq 0.05$$

$$c_7 : 0.1 \leq B_e \leq 1$$

Avec :

$$J_{cu} = \sqrt{10^{11}/0.7E}$$

$$D = 0.1P/\pi$$

$$\lambda = 0.015\pi D^2(D + E)B_e\sqrt{7^{11}\beta E}$$

$$K_f = 1.5P\beta(e + E)/D$$

$$B_e = 1.8la/(D \log(D + 2E)/(D - 2(la + e)))$$

$$C = D\pi\beta B_e/(6P)$$

Critères :

$$Vu = \pi(D/\lambda)(D + E - e - la)(2C + E + e + la)$$

$$Va = \pi\beta la(D/\lambda)(D - 2e - la)$$

$$Pj = 0.018e^{-6}\pi(D/\lambda)(D + E)10^{11}$$



D.3 Problème *ice*

CSP : $X = (b, c_r, d_I, d_E, w)$

$$[b] = [0, 100]$$

$$[c_r] = [0, 20]$$

$$[d_I] = [0, 50]$$

$$[d_E] = [0, 50]$$

$$[w] = [0, 6.5]$$

Sujet à :

$$c_1 : b - 83.33 \leq 0$$

$$c_2 : 76.9 - b \leq 0$$

$$c_3 : d_I + d_E - 0.82b \leq 0$$

$$c_4 : 0.83d_I - d_E \leq 0$$

$$c_5 : d_E - 0.89d_I \leq 0$$

$$c_6 : 215.46 - d_I^2 \leq 0$$

$$c_7 : c_r - 13.2 + 0.0045b \leq 0$$

$$c_8 : 2b^3/1.18348e^6 - 1.3 < 0$$

$$c_9 : 2b^3/1.18348e^6 - 0.7 > 0$$

Avec :

$$FMEP = 4.826(c_r - 9.2) + (7.97 + 2.99e^5wb^{-2} + 1.36e7bw^2)$$

$$\eta_t = 0.8595(1 - c_r^{-0.33}) - S_v \sqrt{1.5/w}$$

$$\eta_v = \eta_{vb}(1 + 5.96e^{-3}w^2)/((1 + 55.789/0.44)$$

$$\times (w/(d_I^2))^2$$

$$\eta_{vb} = \begin{cases} 0.067 - 0.038e^{w-5.25}, & w \geq 5.25, \\ 0.637 + 0.13w - 0.014w^2 \\ + 0.00066w^3, & w < 5.25 \end{cases}$$

$$S_v = 0.83(8 + 4c_r + 1.0134e^{-5}(c_r - 1)b^3)/((2 + c_r)b)$$

Critères :

$$ISFC = 81.8964/(0.8595(1 - c_r^{-0.33}) + (0.83((8 + 4c_r) +$$

$$1.5(c_r - 1)(4\pi/1.859e^6)b^3)/(2 + c_r)b))$$

$$PwB = w[FMEP - 3688\eta_t\eta_v]/120$$

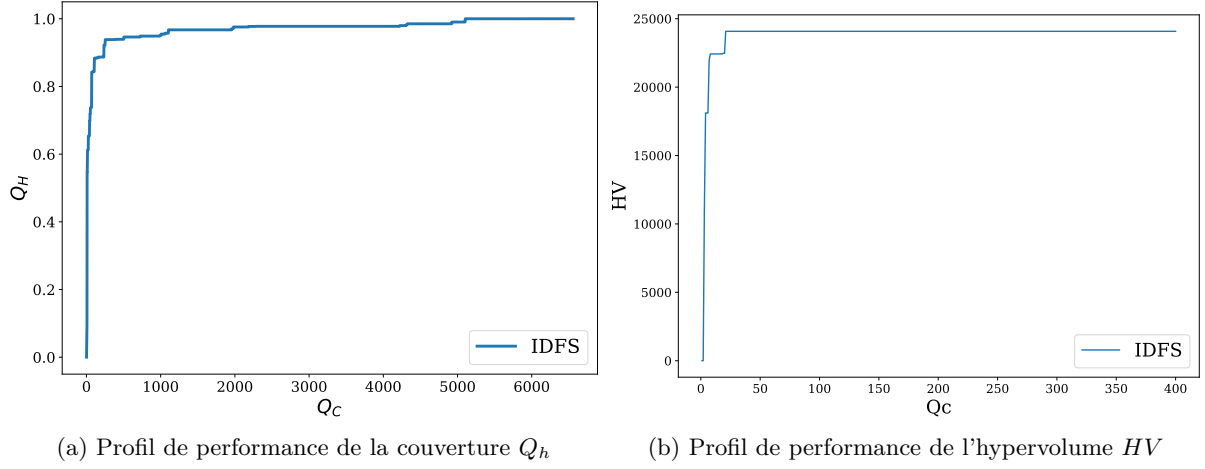


FIGURE D.3 – Couverture et hypervolume de la première exploration du problème *ice* ($t^{max} = 60s$), \mathcal{S}_a , 5499 solutions

D.4 Problème *batterie*

D.4.1 Calcul de P_{peak}

Le pic de puissance P_{peak} est calculé pour une accélération de la voiture de 0 à 100 km.h-1 ($v = 27.78$ m.s-1) en 10 secondes sur route droite sans pente avec l'expression D.2.

$$\begin{aligned}
 P_{peak} &= v(F_{fr} + F_{ta} + F_{acc}) & (D.2) \\
 F_{fr} &= gC_{rr}(M_{base} + M_{batt}) \\
 F_{ta} &= 0.5\rho AC_t v \\
 F_{acc} &= v(M_{base} + M_{batt})/10
 \end{aligned}$$

Avec la force d'accélération F_{acc} , la force de traînée aérodynamique F_{ta} et la force de frottement au sol F_{fr} . On donne le coefficient de résistance au roulement $C_{rr} = 0.012$, g l'accélération de la pesanteur, $\rho = 1,225$ kg.m⁻³, C_t le coefficient de traînée et A la surface frontale de la voiture considérée égale à 2 m².

D.4.2 Calcul du vieillissement cyclique

Pour le calcul de la diminution de la capacité Q_{cy} due au vieillissement cyclique, on distingue les chimies NMC et LFP [35]. Les expressions sont tirées des travaux de [189] pour les cellules li-ion LFP et de [32] pour les NMC. On obtient respectivement les relations D.3 et D.4.

$$Q_{cy}(NMC) = 30330e^{-31500/(8.314T_{C/2})}Ah^{0.552}/100 \quad (D.3)$$

$$Q_{cy}(LFP) = ((557 + 9610(0.75 - Dod)^3)e^{-22406/(8.314T_{C/2})})Ah^{0.48}/100 \quad (D.4)$$

Avec $T_{C/2}$ la température interne des cellules pour le point de fonctionnement choisi, ici une décharge de $C - rate = C/2$. Ah correspond à la quantité d'ampère-heure ayant traversé chaque cellule pendant N_{cy} cycles à une profondeur de décharge de D_{od} %, soit $Ah = D_{od}C_cN_{cy}$.

D.4.3 Post-traitement de l'exploration (*batterie*)

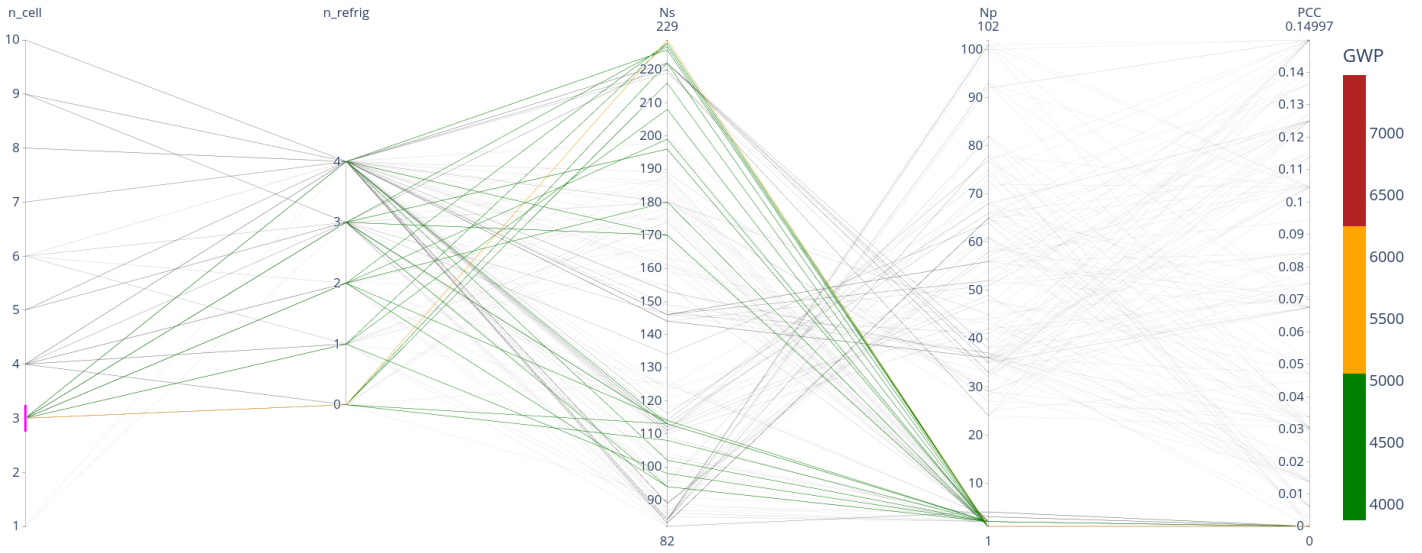


FIGURE D.4 – Mise en évidence des architectures de la référence de cellule $n_{cell} = 3$

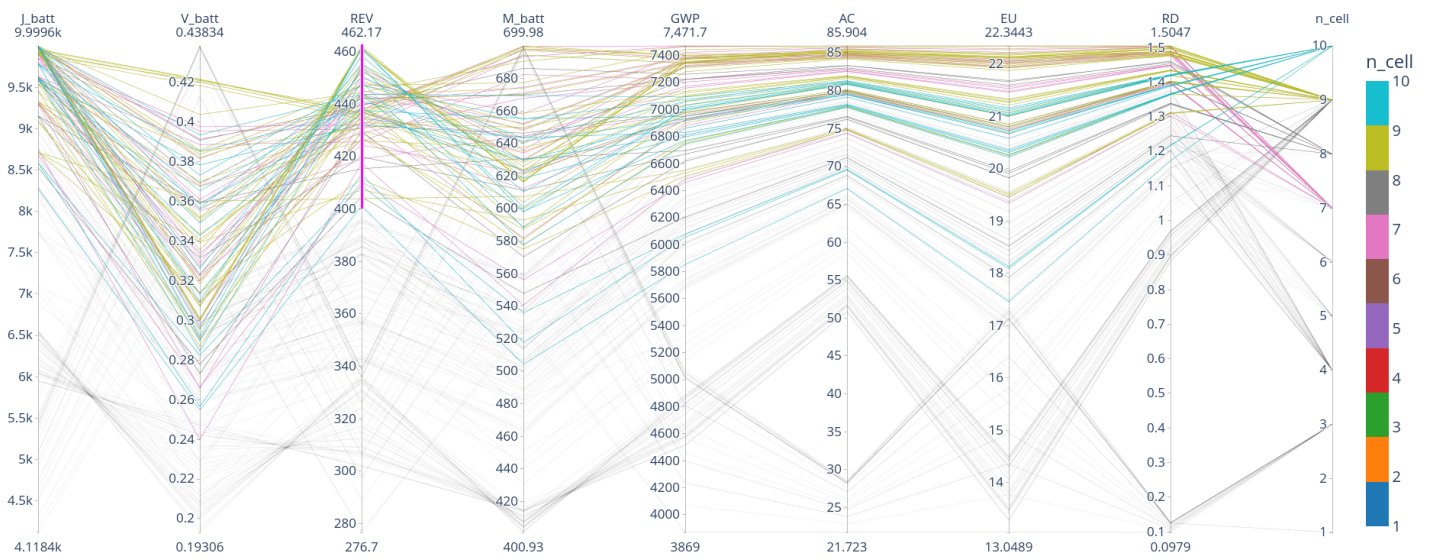


FIGURE D.5 – Mise en évidence des architectures avec une autonomie supérieure à $REV \geq 400$ km

Références bibliographiques

- [1] Aditya AKUNDI et Viviana LOPEZ, « A Review on Application of Model Based Systems Engineering to Manufacturing and Production Engineering Systems », in : *Procedia Computer Science*, Big Data, IoT, and AI for a Smarter Future 185 (1^{er} jan. 2021), p. 101-108, ISSN : 1877-0509, DOI : 10.1016/j.procs.2021.05.011, (visité le 15/02/2023).
- [2] Johan ANDERSSON, *A Survey of Multiobjective Optimization in Engineering Design*, Technical Report LiTH-IKP- R-1097, Linköping University, Linköping, 2000.
- [3] Ignacio ARAYA, Gilles TROMBETTONI et Bertrand NEVEU, « A Contractor Based on Convex Interval Taylor », in : *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Lecture Notes in Computer Science, Berlin, Heidelberg : Springer, 2012, p. 1-16, ISBN : 978-3-642-29828-8, DOI : 10.1007/978-3-642-29828-8_1.
- [4] Charles AUDET et al., « Performance indicators in multiobjective optimization », in : *European Journal of Operational Research* 292 (1^{er} nov. 2020), DOI : 10.1016/j.ejor.2020.11.016.
- [5] Xavier BAGUENARD, « Propagation de contraintes sur les intervalles : Application à l'étalonnage des robots », Thèse de doctorat, Angers, France : Université d'Angers, 1^{er} jan. 2005, URL : <http://www.theses.fr/2005ANGE0043> (visité le 09/08/2021).
- [6] Anthony BARRÉ et al., « A review on lithium-ion battery ageing mechanisms and estimations for automotive applications », in : *Journal of Power Sources* 241 (nov. 2013), p. 680-689, ISSN : 03787753, DOI : 10.1016/j.jpowsour.2013.05.040, (visité le 27/01/2022).
- [7] F. BENHAMOU et al., « Revising Hull and Box Consistency. », in : *International Conference on Logic Programming* (1999), p. 230-244, DOI : 10.7551/mitpress/4304.003.0024.
- [8] Frédéric BENHAMOU et Laurent GRANVILLIERS, « Chapter 16. Continuous and Interval Constraints », in : *Foundations of Artificial Intelligence 2* (déc. 2006), DOI : 10.1016/S1574-6526(06)80020-9.
- [9] Frédéric BENHAMOU et William J. OLDER, « Applying interval arithmetic to real, integer, and boolean constraints », in : *The Journal of Logic Programming* 32.1 (1997), p. 1-24.
- [10] Callie BERLINER et James A. BRIMSON, *Cost Management for Today's Advanced Manufacturing : The CAM-I Conceptual Design*, Harvard Business School Press, 1988, 280 p., ISBN : 978-0-87584-197-7, Google Books : QfZTAAAAMAAJ.
- [11] Gurbakhash Singh BHANDER, Michael HAUSCHILD et Tim MCALOONE, « Implementing life cycle assessment in product development », in : *Environmental Progress* 22.4 (2003), p. 255-267, ISSN : 1547-5921, DOI : 10.1002/ep.670220414, (visité le 23/06/2022).
- [12] B. BLANCHARD et W. FABRYCKY, *Systems Engineering and Analysis, Third Edition*, Pearson International, Systems Engineering and Analysis, 1^{er} jan. 1998.
- [13] Pierre BLONDEL, « Estimation de l'état interne d'une batterie lithium-ion à l'aide d'un modèle électrochimique », Thèse de doctorat, Université de Lorraine, 2019.

-
- [14] Pierre BONAMI, Mustafa KILINÇ et Jeff LINDEROTH, « Algorithms and Software for Convex Mixed Integer Nonlinear Programs », in : *Mixed Integer Nonlinear Programming*, sous la dir. de Jon LEE et Sven LEYFFER, The IMA Volumes in Mathematics and its Applications, New York, NY : Springer, 2012, p. 1-39, ISBN : 978-1-4614-1927-3, DOI : 10.1007/978-1-4614-1927-3_1.
- [15] Houda BOUYARMANE, Mehdi EL AMINE et Mohammed SALLAOU, « Consideration of behavioral models accuracy in the evaluation of a solar collector support performance during embodiment phase », in : *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), mars 2022, p. 1-5, DOI : 10.1109/IRASET52964.2022.9737745.
- [16] Oussama BRAYDI, Pascal LAFON et Rafic YOUNES, « On the formulation of optimization problems under uncertainty in mechanical design », in : *International Journal on Interactive Design and Manufacturing (IJIDeM)* 13.1 (mars 2019), p. 75-87, ISSN : 1955-2513, 1955-2505, DOI : 10.1007/s12008-018-0492-6, (visité le 26/01/2023).
- [17] S. BRISSET et P. BROCHET, « Analytical model for the optimal design of a brushless DC wheel motor », in : *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering* 24.3 (1^{er} sept. 2005), p. 829-848, ISSN : 0332-1649, DOI : 10.1108/03321640510612952, (visité le 29/06/2022).
- [18] Alexander E. I. BROWNLEE et Jonathan A. WRIGHT, « Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation », in : *Applied Soft Computing* 33 (1^{er} août 2015), p. 114-126, ISSN : 1568-4946, DOI : 10.1016/j.asoc.2015.04.010, (visité le 10/05/2022).
- [19] Maria Leonor CARVALHO, Andrea TEMPORELLI et Pierpaolo GIRARDI, « Life Cycle Assessment of Stationary Storage Systems within the Italian Electric Network », in : *Energies* 14.8 (7 avr. 2021), p. 2047, ISSN : 1996-1073, DOI : 10.3390/en14082047, (visité le 11/05/2023).
- [20] Gilles CHABERT et Luc JAULIN, « Contractor programming », in : *Artificial Intelligence* 173.11 (2009), p. 1079-1100, ISSN : 0004-3702, DOI : 10.1016/j.artint.2009.03.002.
- [21] Gang CHEN et al., « Comprehensive sizing and optimization method for series-hybrid unmanned convertiplane », in : *Chinese Journal of Aeronautics* 34.4 (avr. 2021), p. 387-402, ISSN : 10009361, DOI : 10.1016/j.cja.2020.07.016, (visité le 26/01/2022).
- [22] F. Y. CHENG et X. S. LI, « Generalized Center Method for Multiobjective Engineering Optimization », in : *Engineering Optimization* 31.5 (1^{er} mai 1999), p. 641-661, ISSN : 0305-215X, DOI : 10.1080/03052159908941390, (visité le 27/01/2023).
- [23] R. CHENOUEARD, L. GRANVILLIERS et P. SEBASTIAN, « Search heuristics for constraint-aided embodiment design », in : *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23.2 (mai 2009), p. 175-195, ISSN : 0890-0604, 1469-1760, DOI : 10.1017/S0890060409000055, (visité le 09/08/2021).

-
- [24] R. CHENOUEARD, L. GRANVILLIERS et P. SÉBASTIAN, « Solving an Air Conditioning System Problem in an Embodiment Design Context Using Constraint Satisfaction Techniques », in : *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science 4741/2007 (sept. 2007), p. 18-32, DOI : 10.1007/978-3-540-74970-7_4.
- [25] Raphaël CHENOUEARD, « Résolution par satisfaction de contraintes appliquée à l'aide à la décision en conception architecturale », Thèse de doctorat, Ecole nationale supérieure d'arts et métiers - ENSAM, déc. 2007, URL : <https://tel.archives-ouvertes.fr/tel-00482208>.
- [26] Raphaël CHENOUEARD, Alexandre GOLDSZTEJN et Christophe JERMANN, « Search Strategies for an Anytime Usage of the Branch and Prune Algorithm », in : *International Joint Conference on Artificial Intelligence*, United States, 2009, p. 468-473, URL : <https://hal.archives-ouvertes.fr/hal-00421506>.
- [27] G. CHIANDUSSI et al., « Comparison of multi-objective optimization methodologies for engineering applications », in : *Computers & Mathematics with Applications* 63.5 (1^{er} mars 2012), p. 912-942, ISSN : 0898-1221, DOI : 10.1016/j.camwa.2011.11.057, (visité le 02/05/2022).
- [28] Rebecca E. CIEZ et J.F. WHITACRE, « Comparison between cylindrical and prismatic lithium-ion cell costs using a process based cost model », in : *Journal of Power Sources* 340 (fév. 2017), p. 273-281, ISSN : 03787753, DOI : 10.1016/j.jpowsour.2016.11.054, (visité le 01/02/2022).
- [29] François CLUZEL, « Eco-design implementation for complex industrial system : From scenario-based LCA to the definition of an eco-innovative R&D projects portfolio », Thèse de doctorat, Ecole Centrale Paris, sept. 2012, URL : <https://tel.archives-ouvertes.fr/tel-00760580>.
- [30] Jared L. COHON et David H. MARKS, « A review and evaluation of multiobjective programming techniques », in : *Water Resources Research* 11.2 (1975), p. 208-220, ISSN : 1944-7973, DOI : 10.1029/WR011i002p00208, (visité le 17/02/2023).
- [31] Arnaud COLLIGNAN, « Méthode d'optimisation et d'aide à la décision en conception mécanique : Application à une structure aéronautique », Thèse de doctorat, Université Bordeaux 1, 2012, URL : <https://theses.hal.science/tel-00662303>.
- [32] Andrea CORDOBA-ARENAS et al., « Capacity and power fade cycle-life model for plug-in hybrid electric vehicle lithium-ion battery cells containing blended spinel and layered-oxide positive electrodes », in : *Journal of Power Sources* 278 (15 mars 2015), p. 473-483, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2014.12.047, (visité le 11/02/2022).
- [33] Yassir DAHMANE, « Gestion d'énergie optimisée étendue véhicules infrastructures », Thèse de doctorat, Ecole centrale de Nantes, 16 déc. 2020, URL : <https://www.theses.fr/2020ECDN0047> (visité le 09/08/2021).
- [34] Indraneel DAS et J. E. DENNIS, « Normal-Boundary Intersection : A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems », in : *SIAM Journal on Optimization* 8.3 (août 1998), p. 631-657, ISSN : 1052-6234, DOI : 10.1137/S1052623496307510, (visité le 29/03/2023).

-
- [35] Michele DE GENNARO et al., « A case study to predict the capacity fade of the battery of electrified vehicles in real-world use conditions », in : *Case Studies on Transport Policy* 8.2 (1^{er} juin 2020), p. 517-534, ISSN : 2213-624X, DOI : 10.1016/j.cstp.2019.11.005, (visité le 14/01/2022).
- [36] T. DEAN et M. BODDY, « An Analysis of Time-Dependent Planning », in : *AAAI*, 7th National Conference on Artificial Intelligence, St. Paul, MN, 1988, p. 7.
- [37] K. DEB et S. JAIN, « Running performance metrics for evolutionary multi-objective optimizations », in : *CS'2002*, Computer Science, 2002, URL : <https://www.semanticscholar.org/paper/Running-performance-metrics-for-evolutionary-Deb-Jain/3a27ac23c1248ccc32d99ba87e33c9ceea9c10b4>.
- [38] K. DEB et al., « A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization : NSGA-II », in : *Parallel Problem Solving from Nature PPSN VI*, t. 1917, Berlin, Heidelberg : Springer Berlin Heidelberg, 2000, p. 849-858, ISBN : 978-3-540-41056-0, DOI : 10.1007/3-540-45356-3_83, (visité le 28/04/2022).
- [39] Kalyanmoy DEB, « Multi-Objective Optimization Using Evolutionary Algorithms : An Introduction », in : (2011), p. 24.
- [40] R. DECHTER et J. PEARL, « Generalized Best-First Search Strategies and the Optimality of A* », in : *Journal of the ACM* 32.3 (1985), DOI : 10.1145/3828.3830.
- [41] Reynard de VRIES, Malcom BROWN et Roelof VOS, « Preliminary Sizing Method for Hybrid-Electric Distributed-Propulsion Aircraft », in : *Journal of Aircraft* 56.6 (nov. 2019), p. 2172-2188, ISSN : 1533-3868, DOI : 10.2514/1.C035388, (visité le 26/01/2022).
- [42] Sephora DIAMPOVESA, « Modèles de synthèse pour la conception en génie électrique. Application à l'électrification des véhicules. », Thèse de doctorat, Université Technologique de Compiègne, 2021, URL : <https://hal.science/hal-01933261>.
- [44] Yahong DONG et al., « Developing Conversion Factors of LCIA Methods for Comparison of LCA Results in the Construction Sector », in : *Sustainability* 13.16 (16 jan. 2021), ISSN : 2071-1050, DOI : 10.3390/su13169016.
- [45] Matthieu DUBARRY, Nicolas VUILLAUME et Bor Yann LIAW, « From single cell model to battery pack simulation for Li-ion batteries », in : *Journal of Power Sources* 186.2 (15 jan. 2009), p. 500-507, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2008.10.051, (visité le 27/01/2022).
- [46] João A. DURO et al., « Methods for constrained optimization of expensive mixed-integer multi-objective problems, with application to an internal combustion engine design problem », in : *European Journal of Operational Research* (6 sept. 2022), ISSN : 0377-2217, DOI : 10.1016/j.ejor.2022.08.032, (visité le 27/10/2022).
- [47] Mehdi EL AMINE, « Intégration de la maturité des connaissances dans la prise de décision en conception mécanique : application à un système solaire. », Thèse de doctorat, Paris : École Nationale Supérieure d'Arts et Métiers, 2016, URL : <https://pastel.hal.science/tel-01264683>.
- [48] Alain FAISANDIER, « Ingénierie des besoins et des exigences d'un système complexe », in : *Management et ingénierie de l'innovation* (mai 2014), DOI : 10.51257/f-0270, (visité le 26/01/2023).

-
- [49] Xavier FISCHER, « Stratégie de Conduite du Calcul pour l'aide à la décision en conception mécanique intégrée ; application aux appareils à pression », Thèse de doctorat, Paris : École Nationale Supérieure d'Arts et Métiers, 2000, URL : <http://rgdoi.net/10.13140/RG.2.2.19080.11529> (visité le 09/08/2021).
- [50] E. FITAN, F. MESSINE et B. NOGAREDE, « The Electromagnetic Actuator Design Problem : A General and Rational Approach », in : *IEEE Transactions on Magnetics* 40.3 (mai 2004), p. 1579-1590, ISSN : 0018-9464, DOI : 10.1109/TMAG.2004.827183, (visité le 09/08/2021).
- [51] Kevin FORSBERG et Harold MOOZ, « The Relationship of System Engineering to the Project Cycle », in : *INCOSE International Symposium* 1.1 (1991), p. 57-65, ISSN : 2334-5837, DOI : 10.1002/j.2334-5837.1991.tb01484.x, (visité le 17/03/2023).
- [52] Robert FOURER, David M. GAY et Brian W. KERNIGHAN, *AMPL Reference Manual*, AMPL, 2003.
- [53] Eugene C. FREUDER, « In Pursuit of the Holy Grail », in : *Constraints* 2.1 (1^{er} avr. 1997), p. 57-61, ISSN : 1572-9354, DOI : 10.1023/A:1009749006768, (visité le 18/07/2023).
- [54] Eugene C. FREUDER, « Synthesizing constraint expressions », in : *Communications of the ACM* 21.11 (1^{er} nov. 1978), p. 958-966, ISSN : 0001-0782, DOI : 10.1145/359642.359654, (visité le 27/05/2023).
- [55] GAMS, *GAMS Reference Manual*, GAMS Development Corporation, 2020.
- [56] Guojun GAN, Chaoqun MA et Jianhong WU, « 12. Grid-Based Clustering Algorithms », in : *Data Clustering : Theory, Algorithms, and Applications*, Society for Industrial and Applied Mathematics, jan. 2007, p. 209, ISBN : 978-0-89871-834-8, DOI : 10.1137/1.9780898718348, (visité le 11/01/2022).
- [58] J. GARCIA et al., « A novel approach for global environmental performance evaluation of electric batteries for hybrid vehicles », in : *Journal of Cleaner Production* (juill. 2017), p. 406-417, ISSN : 09596526, URL : <https://linkinghub.elsevier.com/retrieve/pii/S0959652617307436> (visité le 09/08/2021).
- [59] E. GELLE et Boi FALTINGS, « Solving Mixed and Conditional Constraint Satisfaction Problems », in : *Constraints* 8 (avr. 2003), DOI : 10.1023/A:1022394531132.
- [60] E. GELLE et al., « Constraint Satisfaction Methods for Applications in Engineering », in : *Engineering with Computers* 16 (sept. 2000), DOI : 10.1007/PL00007190.
- [61] John GERO et Udo KANNENGIESSER, « The Situated Function-Behaviour-Structure Framework », in : *Design Studies* 25 (juill. 2004), p. 373-391, DOI : 10.1016/j.destud.2003.10.010.
- [62] Fred GLOVER, Arne LØKKETANGEN et David L. WOODRUFF, « Scatter Search to Generate Diverse MIP Solutions », in : *Computing Tools for Modeling, Optimization and Simulation : Interfaces in Computer Science and Operations Research*, sous la dir. de Manuel LAGUNA et José Luis González VELARDE, Operations Research/Computer Science Interfaces Series, Boston, MA : Springer US, 2000, p. 299-317, ISBN : 978-1-4615-4567-5, DOI : 10.1007/978-1-4615-4567-5_17, (visité le 14/02/2023).

-
- [63] Alexandre GOLDSZTEJN et Laurent GRANVILLIERS, « A New Framework for Sharp and Efficient Resolution of NCSP with Manifolds of Solutions », in : *Constraints* 15 (jan. 2008), p. 190-212, DOI : 10.1007/s10601-009-9082-3.
- [64] Alexandre GOLDSZTEJN et Luc JAULIN, « Inner Approximation of the Range of Vector-Valued Functions », in : *Reliable Computing* 14 (1^{er} juin 2010).
- [65] Nelson GRIMA et Néstor CAMPOS, « A farewell to glaciers : Ecosystem services loss in the Spanish Pyrenees », in : *Journal of Environmental Management* 269 (1^{er} sept. 2020), ISSN : 0301-4797, DOI : 10.1016/j.jenvman.2020.110789.
- [66] Jeroen B. GUINEE, « Handbook on life cycle assessment operational guide to the ISO standards », in : *The International Journal of Life Cycle Assessment* 7.5 (1^{er} sept. 2002), p. 311-313, ISSN : 1614-7502, DOI : 10.1007/BF02978897, (visité le 15/05/2023).
- [67] S. GUNAWAN et al., « Quality-Assisted Multi-Objective Multidisciplinary Genetic Algorithms », in : *AIAA Journal* 41.9 (sept. 2003), p. 1752-1762, ISSN : 0001-1452, 1533-385X, DOI : 10.2514/2.7293, (visité le 27/10/2022).
- [68] S. Al HALLAJ et J. R. SELMAN, « A Novel Thermal Management System for Electric Vehicle Batteries Using Phase-Change Material », in : *Journal of The Electrochemical Society* 147.9 (2022), p. 3231, ISSN : 00134651, DOI : 10.1149/1.1393888, (visité le 25/10/2022).
- [69] Peter T. HARRIS et al., « A marine plastic cloud - Global mass balance assessment of oceanic plastic pollution », in : *Continental Shelf Research* 255 (15 fév. 2023), ISSN : 0278-4343, DOI : 10.1016/j.csr.2023.104947.
- [70] Chris HARTMANN, « Automatisation de la synthèse d'architectures appliquée aux aéronefs à voilure tournante », Thèse de doctorat, Nantes : Ecole centrale de Nantes, 2018, URL : <https://theses.hal.science/tel-01964251>.
- [71] Chris HARTMANN et al., « Formulation of a design problem for computational pre-design », in : *Virtual Concept Workshop 2016*, Bordeaux, France : ID3, mars 2016, URL : <https://hal.archives-ouvertes.fr/hal-01690831> (visité le 13/10/2021).
- [72] M. HAUSCHILD, J. JESWIET et L. ALTING, « From Life Cycle Assessment to Sustainable Production : Status and Perspectives », in : *CIRP Annals* 54.2 (1^{er} jan. 2005), p. 1-21, ISSN : 0007-8506, DOI : 10.1016/S0007-8506(07)60017-1, (visité le 12/05/2023).
- [73] Emmanuel HEBRARD et al., « Finding Diverse and Similar Solutions in Constraint Programming. », in : *Proceedings of the National Conference on Artificial Intelligence*, t. 1, 1^{er} jan. 2005, p. 377.
- [74] Juha HEINONEN, *Lectures on Analysis on Metric Spaces*, Mathematics Department East Hall, University of Michigan, Ann Arbor, USA : Springer New York, NY, 1^{er} jan. 2001, 141 p., ISBN : 978-1-4612-6525-2, DOI : 10.1007/978-1-4613-0131-8.
- [75] Gabrielle HEYVAERT, « Le Pacte vert pour l'Europe : la longue route vers la neutralité carbone », in : *La Revue de l'Énergie*, EUROPE 648 (2020), URL : <https://www.larevuedelenergie.com/>.

-
- [76] Thomas HOUDENOVE et al., « CSP dynamiques en configuration », in : Journée Francophone de la Programmation par Contrainte (JFPC), 1^{er} jan. 2006.
- [77] Xiaosong HU, Shengbo LI et Huei PENG, « A comparative study of equivalent circuit models for Li-ion batteries », in : *Journal of Power Sources* 198 (15 jan. 2012), p. 359-367, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2011.10.013.
- [78] C. HUBER et R. KUHN, « Thermal management of batteries for electric vehicles », in : *Advances in Battery Technologies for Electric Vehicles*, sous la dir. de Bruno SCROSATI, Jürgen GARCHE et Werner TILLMETZ, Woodhead Publishing Series in Energy, Woodhead Publishing, 1^{er} jan. 2015, p. 327-358, ISBN : 978-1-78242-377-5, DOI : 10.1016/B978-1-78242-377-5.00013-3, (visité le 08/03/2022).
- [79] X. HUIN et al., « Optimization Based Methodology for Hybrid Electric Vehicle Dedicated Internal Combustion Engine Design », in : *Decision and Information Technologies (CoDIT)*, 7th International Conference on Control, 2020, DOI : 10.1109/CoDIT49905.2020.9263821.
- [80] Truong Hoang Bao HUY et al., « Multi-Objective Search Group Algorithm for engineering design problems », in : *Applied Soft Computing* 126 (sept. 2022), p. 109287, ISSN : 15684946, DOI : 10.1016/j.asoc.2022.109287, (visité le 26/01/2023).
- [82] Linnea INGMAR et al., « Modelling Diversity of Solutions », in : *Proceedings of the AAAI Conference on Artificial Intelligence* 34.02 (02 3 avr. 2020), p. 1528-1535, ISSN : 2374-3468, DOI : 10.1609/aaai.v34i02.5512.
- [83] IPCC, *Climate change 2021 : The physical science basis. Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change*, sous la dir. de V. MASSON-DELMOTTE et al., Cambridge, UK and New York, NY, USA : Cambridge University Press, 2021, DOI : 10.1017/9781009157896.
- [84] Engin IPEK et al., « Efficient architectural design space exploration via predictive modeling », in : *ACM Transactions on Architecture and Code Optimization* 4.4 (jan. 2008), p. 1-34, ISSN : 1544-3566, 1544-3973, DOI : 10.1145/1328195.1328196.
- [85] ISO, *14006 :2020*, ISO, 2020, URL : <https://www.iso.org/fr/standard/72644.html> (visité le 08/05/2023).
- [86] ISO, *14040 :2006*, ISO, 12 août 2014, URL : <https://www.iso.org/fr/standard/37456.html> (visité le 12/05/2023).
- [87] Himanshu JAIN et Kalyanmoy DEB, « An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II : Handling Constraints and Extending to an Adaptive Approach », in : *IEEE Transactions on Evolutionary Computation* 18.4 (août 2014), p. 602-622, ISSN : 1941-0026, DOI : 10.1109/TEVC.2013.2281534.
- [88] L. JAULIN, O. KIEFFER M.and Didrit et E. WALTER, *Applied Interval Analysis*, 1^{re} éd., Springer London Ltd, août 2001, 398 p., ISBN : 1-85233-219-0, URL : <https://hal.archives-ouvertes.fr/hal-00845131>.

-
- [89] A. JESUS, L. PAQUETE et A. LIEFOOGHE, « A model of anytime algorithm performance for bi-objective optimization », in : *Journal of Global Optimization* (mai 2020), DOI : 10.1007/s10898-020-00909-9.
- [90] Alexandre JESUS et al., « Algorithm selection of anytime algorithms », in : 2020 Genetic and Evolutionary Computation Conference, Cancùn, Mexico : Association for Computing Machinery, juin 2020, p. 850-858, ISBN : 978-1-4503-7128-5, DOI : 10.1145/3377930.3390185.
- [91] Leslie Pack KAEHLING, « Associative reinforcement learning : A generate and test algorithm », in : *Machine Learning* 15.3 (1^{er} juin 1994), p. 299-319, ISSN : 1573-0565, DOI : 10.1007/BF00993348, (visité le 18/07/2023).
- [92] Eunsuk KANG, Ethan JACKSON et Wolfram SCHULTE, « An Approach for Effective Design Space Exploration », in : *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, sous la dir. de Radu CALINESCU et Ethan JACKSON, Lecture Notes in Computer Science, Berlin, Heidelberg : Springer, 2011, p. 33-54, ISBN : 978-3-642-21292-5, DOI : 10.1007/978-3-642-21292-5_3.
- [93] Gio KAO, Edward SEWELL et Sheldon JACOBSON, « A branch, bound, and remember algorithm for the 1|ri|ti scheduling problem », in : *Journal of Scheduling* 12 (avr. 2009), p. 163-175, DOI : 10.1007/s10951-008-0087-3.
- [94] Y. L. KARNAVAS et al., « Modelling and multiobjective optimization analysis of a permanent magnet synchronous motor design : Modelling and Multi-Objective Optimization Analysis of a PMSM Design », in : *International Journal of Numerical Modelling : Electronic Networks, Devices and Fields* 30.6 (nov. 2017), e2232, ISSN : 08943370, DOI : 10.1002/jnm.2232, (visité le 11/03/2022).
- [95] R. B. KEARFOTT, « A Fortran 90 environment for research and prototyping of enclosure algorithms for nonlinear equations and global optimization », in : *ACM Transactions on Mathematical Software* 21.1 (1^{er} mars 1995), p. 63-78, ISSN : 0098-3500, DOI : 10.1145/200979.200991, (visité le 10/02/2023).
- [96] G. A. KEOLEIAN et D. MENEREY, *Life cycle design guidance manual. Environmental requirements and the product system. Final report*, PB-93-164507/XAB, National Pollution Prevention Center, Ann Arbor, MI (United States), 1^{er} jan. 1993, URL : <https://www.osti.gov/biblio/5886794> (visité le 30/03/2023).
- [97] Jaouad KHALFI et al., « Box-Jenkins Black-Box Modeling of a Lithium-Ion Battery Cell Based on Automotive Drive Cycle Data », in : *World Electric Vehicle Journal* 12.3 (3 sept. 2021), p. 102, ISSN : 2032-6653, DOI : 10.3390/wevj12030102, (visité le 15/02/2023).
- [98] Kevin KNEHR et al., *Battery Performance and Cost Modeling for Electric-Drive Vehicles : A Manual for BatPaC v5.0*, ANL/CSE-22/1, 1877590, 176234, 1^{er} juill. 2022, ANL/CSE-22/1, 1877590, 176234, DOI : 10.2172/1877590, (visité le 27/10/2022).
- [99] A. KONE, B. NOGAREDE et M. LAJOIE MAZENC, « Le dimensionnement des actionneurs électriques : un problème de programmation non linéaire », in : *Journal de Physique III* 3.2 (1993), p. 285-301, DOI : 10.1051/jp3:1993132.

-
- [100] Amelie KRUPP et al., « Calendar aging model for lithium-ion batteries considering the influence of cell characterization », in : *Journal of Energy Storage* 45 (1^{er} jan. 2022), ISSN : 2352-152X, DOI : 10.1016/j.est.2021.103506, (visité le 27/01/2022).
- [101] V. KRYSSANOV, H. TAMAKI et K. UEDA, « Synthesis and emergence in engineering design problem solving », in : *Proceedings First International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, Proceedings First International Symposium on Environmentally Conscious Design and Inverse Manufacturing, fév. 1999, p. 690-695, DOI : 10.1109/ECODIM.1999.747699.
- [102] Gang LEI et al., « A Review of Design Optimization Methods for Electrical Machines », in : *Energies* 10.12 (12 déc. 2017), ISSN : 1996-1073, DOI : 10.3390/en10121962.
- [103] Olivier LHOMME, « Consistency Techniques for Numeric CSPs. », in : *IJCAI*, t. 1, jan. 1993, p. 232-238.
- [104] M. LI et X. YAO, « Quality Evaluation of Solution Sets in Multiobjective Optimisation : A Survey », in : *ACM Computing Surveys* 52 (mars 2019), p. 1-38, DOI : 10.1145/3300148.
- [105] Hendrik LÖBBERDING et al., « From Cell to Battery System in BEVs : Analysis of System Packing Efficiency and Cell Types », in : *World Electric Vehicle Journal* 11.4 (10 déc. 2020), p. 77, ISSN : 2032-6653, DOI : 10.3390/wevj11040077, (visité le 01/02/2022).
- [106] A. K. MACKWORTH, « Consistency in networks of relations », in : *Artificial Intelligence* 8.1 (1977), p. 99-118, ISSN : 0004-3702, DOI : 10.1016/0004-3702(77)90007-8.
- [107] Brahim MAHIDDINI, « Contribution à l'amélioration de la conception préliminaire d'un produit : Approche basée sur l'implémentation des techniques d'optimisation », Thèse de doctorat, Paris : École Nationale Supérieure d'Arts et Métiers, 2020.
- [108] Arnaud MALAPERT, « Techniques d'ordonnancement d'atelier et de fournées basées sur la programmation par contraintes », Thèse de doctorat, École Nationale Supérieure des Techniques Industrielles et des Mines de Nantes, 1^{er} jan. 2011, URL : <https://www.theses.fr/2011NANT2042> (visité le 09/02/2023).
- [109] Benjamin MARTIN et al., « On Continuation Methods for Non-Linear Bi-Objective Optimization : Certified Interval-Based Approach », in : *Journal of Global Optimization* 64 (mai 2014), DOI : 10.1007/s10898-014-0201-3.
- [110] Joaquim R. R. A. MARTINS et Andrew B. LAMBE, « Multidisciplinary Design Optimization : A Survey of Architectures », in : *AIAA Journal* 51.9 (2013), p. 2049-2075, ISSN : 0001-1452, DOI : 10.2514/1.J051895, (visité le 04/01/2022).
- [111] Jad MATAR, « Une méthode d'aide à la décision en conception architecturale : proposition d'une approche outillée », Thèse de doctorat, Ecole centrale de Nantes, 1^{er} jan. 2013, URL : <http://www.theses.fr/2013ECDN0022> (visité le 09/08/2021).

-
- [112] Issam MAZHOUD et al., « The electromagnetic actuator design problem : an adapted interval global optimization algorithm using model reformulation and constraint propagation », in : *18th International Conference on the Computation of Electromagnetic Fields (COMPUMAG2011)*, Sydney, Australia, juill. 2011, URL : <https://hal.archives-ouvertes.fr/hal-00636331> (visité le 31/08/2021).
- [113] Martin MECKESHEIMER, « A framework for metamodel-based design : subsystem metamodel assessment and implementation issues », Thèse de doctorat, The Pennsylvania State University, 15 août 2001, URL : <https://www.semanticscholar.org/paper/A-FRAMEWORK-FOR-METAMODEL-BASED-DESIGN%3A-SUBSYSTEM-Meckesheimer/6e405e270cfe310ca555b62a80ff0a774d8bbd35> (visité le 19/04/2023).
- [114] P. MESEGUER, « Interleaved Depth-First Search », in : *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'97*, Nagoya, Japan, 1997, p. 1382-1387, DOI : 10.5555/1622270.1622355.
- [115] F. MESSINE, B. NOGAREDE et J.-L. LAGOUANELLE, « Optimal design of electromechanical actuators : a new method based on global optimization », in : *IEEE Transactions on Magnetics* 34.1 (jan. 1998), p. 299-308, ISSN : 1941-0069, DOI : 10.1109/20.650361.
- [116] S. MITTAL et Brian FALKENHAINER, « Dynamic Constraint Satisfaction Problems », in : AAAI Conference on Artificial Intelligence, 29 juill. 1990, URL : <https://www.semanticscholar.org/paper/Dynamic-Constraint-Satisfaction-Problems-Mittal-Falkenhainer/f8027afc65b159ced7d030ac0655b8> (visité le 03/07/2023).
- [117] Ugo MONTANARI, « Networks of constraints : Fundamental properties and applications to picture processing », in : *Information Sciences* 7 (jan. 1974), p. 95-132, ISSN : 00200255, DOI : 10.1016/0020-0255(74)90008-5, (visité le 10/02/2023).
- [118] Ramon E. MOORE, R. Baker KEARFOTT et Michael J. CLOUD, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, jan. 2009, ISBN : 978-0-89871-669-6, DOI : 10.1137/1.9780898717716, (visité le 09/08/2021).
- [119] D. R. MORRISON et al., « Branch-and-bound algorithms : A survey of recent advances in searching, branching, and pruning », in : *Discrete Optimization* 19 (2016), p. 79-102, ISSN : 1572-5286, DOI : 10.1016/j.disopt.2016.01.005.
- [120] Michal MOSHKOVITZ et al., « Explainable k-Means and k-Medians Clustering », in : *Proceedings of the 37th International Conference on Machine Learning*, International Conference on Machine Learning, PMLR, 21 nov. 2020, p. 7055-7065, URL : <https://proceedings.mlr.press/v119/moshkovitz20a.html> (visité le 11/03/2022).
- [121] Clemens MÜNZER, « Constraint-Based Methods for Automated Computational Design Synthesis of Solution Spaces », Thèse de doctorat, ETH Zurich, 2015, 1 Band, DOI : 10.3929/ETHZ-A-010603411, (visité le 09/08/2021).
- [122] Fionn MURTAGH et Pedro CONTRERAS, « Algorithms for hierarchical clustering : an overview », in : *WIREs Data Mining and Knowledge Discovery* 2.1 (jan. 2012), p. 86-97, ISSN : 1942-4787, 1942-4795, DOI : 10.1002/widm.53, (visité le 03/01/2022).

-
- [123] P NELSON et al., *Modeling the performance and cost of lithium-ion batteries for electric-drive vehicles*. ANL-11/32, 1027714, 20 oct. 2011, ANL-11/32, 1027714, DOI : 10.2172/1027714, (visité le 09/08/2021).
- [124] Nicholas NETHERCOTE et al., « MiniZinc : Towards a Standard CP Modelling Language », in : *Principles and Practice of Constraint Programming – CP 2007*, sous la dir. de Christian BESSIÈRE, t. 4741, Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 529-543, ISBN : 978-3-540-74969-1, DOI : 10.1007/978-3-540-74970-7_38, (visité le 09/08/2021).
- [125] A. NEUMAIER, *Interval Methods for Systems of Equations*, Encyclopedia of Mathematics and its Applications, Cambridge : Cambridge University Press, 1991, ISBN : 978-0-521-33196-8, DOI : 10.1017/CB09780511526473, (visité le 09/02/2023).
- [126] Bertrand NEVEU, Gilles TROMBETTONI et Ignacio ARAYA, « Adaptive constructive interval disjunction : algorithms and experiments », in : *Constraints* 20.7 (oct. 2015), p. 452, DOI : 10.1007/s10601-015-9180-3, (visité le 18/11/2021).
- [127] Bertrand NEVEU, Gilles TROMBETTONI et Ignacio ARAYA, « Node selection strategies in interval Branch and Bound algorithms », in : *Journal of Global Optimization* 64.2 (fév. 2016), p. 289-304, ISSN : 0925-5001, 1573-2916, DOI : 10.1007/s10898-015-0375-3, (visité le 09/08/2021).
- [128] W-L OBERKAMPF, C HIRSCH et T-G TRUCANO, *Verification, validation, and predictive capability in computational engineering and physics*. SAND2003-3769, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), 1^{er} fév. 2003, DOI : 10.2172/918370.
- [129] A. G. OLABI et al., « Battery thermal management systems : Recent progress and challenges », in : *International Journal of Thermofluids* 15 (1^{er} août 2022), ISSN : 2666-2027, DOI : 10.1016/j.ijft.2022.100171.
- [130] Andrzej OSYCZKA, « 7 - Multicriteria optimization for engineering design », in : *Design Optimization*, sous la dir. de John S. GERO, Academic Press, 1^{er} jan. 1985, p. 193-227, ISBN : 978-0-12-280910-1, DOI : 10.1016/B978-0-12-280910-1.50012-X.
- [131] Dale K PACE, « Modeling and Simulation Verification and Validation Challenges », in : *JOHNS HOPKINS APL TECHNICAL DIGEST* 25.2 (2004).
- [132] G. PAHL et al., *Engineering Design, A Systematic Approach*, 3rd, Springer London, 2007, 617 p., ISBN : 978-1-84628-318-5.
- [133] Maurizio PALESI et Tony GIVARGIS, « Multi-objective design space exploration using genetic algorithms », in : *Hardware/Software Codesign - Proceedings of the International Workshop*, 1^{er} fév. 2002, p. 67-72, ISBN : 978-1-58113-542-8, DOI : 10.1145/774789.774804.
- [134] P. PAPALAMBROS et D. WILDE, *Principles of Optimal Design : Modeling and Computation*, 3rd edition, Modelling and computation, jan. 2017, 67-73, ISBN : 978-1-107-13267-2, DOI : 10.1017/97811316451038.
- [135] *Paris Agreement*, avec la coll. de Conférence des PARTIES, 2015, URL : <https://heinonline.org/HOL/Page?handle=hein.journals/intlm55&id=803&div=&collection=>.

-
- [136] Shrikant R. PATIL, Bhaskara Rao LOKAVARAPU et Hareesh Karonnin THALIYANVEEDU, « Optimization of battery cooling system used in electric vehicles », in : *Journal of Energy Storage* 58 (1^{er} fév. 2023), ISSN : 2352-152X, DOI : 10.1016/j.est.2022.106299.
- [137] Louis PELLETIER et al., « Sizing of a Battery Pack Based on Series/Parallel Configurations for a High-Power Electric Vehicle as a Constrained Optimization Problem », in : *IEEE Transactions on Vehicular Technology* 69.12 (déc. 2020), p. 14150-14159, ISSN : 1939-9359, DOI : 10.1109/TVT.2020.3004010.
- [138] Nicolas PERRY, Mehdi El AMINE et Jérôme PAILHÈS, « Exploring design space in embodiment design with consideration of models accuracy », in : *CIRP Annals* 64.1 (2015), p. 181-184, ISSN : 00078506, DOI : 10.1016/j.cirp.2015.04.021, (visité le 09/08/2021).
- [139] Linn PERSSON et al., « Outside the Safe Operating Space of the Planetary Boundary for Novel Entities », in : *Environ. Sci. Technol.*, Environmental Science & Technology 56.3 (1^{er} fév. 2022), p. 1510-1521, ISSN : 0013-936X, DOI : 10.1021/acs.est.1c04158, (visité le 08/05/2023).
- [140] Thierry PETIT et Andrew C. TRAPP, « Assessing Both Solution Diversity and Solution Quality in Constraint Programming », in : *2015 INFORMS Computing Society Conference*, 14th INFORMS Computing Society Conference full program proc. Richmond, United States, déc. 2015, URL : <https://hal.archives-ouvertes.fr/hal-01238448> (visité le 12/05/2022).
- [142] Laura PICHERAL, « Contribution à la conception préliminaire robuste en ingénierie de produit », Thèse de doctorat, Grenoble : École doctorale Ingénierie - matériaux mécanique énergétique environnement procédés production, 2014.
- [143] Thomas PLANÈS et al., « Modeling and Design Optimization of an Electric Environmental Control System for Commercial Passenger Aircraft », in : *Aerospace* 10 (8 mars 2023), p. 260, DOI : 10.3390/aerospace10030260.
- [144] Alexandre POPOFF, « Towards a better consideration of users during the ecodesign process : proposal of methodological elements for enriching an early optimization approach », Thèse de doctorat, Université de Toulon, déc. 2017, URL : <https://tel.archives-ouvertes.fr/tel-02023924>.
- [145] Irina RADEVA, « Multi-Criteria Models for Clusters Design », in : *Cybernetics and Information Technologies* 13 (1^{er} mars 2013), DOI : 10.2478/cait-2013-0003.
- [146] T. RAY et K. LIEW, « A Swarm Metaphor for Multiobjective Design Optimization. », in : *Engineering Optimization* 34 (1^{er} mars 2002), p. 141-153, DOI : 10.1080/03052150210915.
- [147] Eduardo REDONDO-IGLESIAS, Pascal VENET et Serge PELISSIER, « Modelling Lithium-Ion Battery Ageing in Electric Vehicle Applications-Calendar and Cycling Ageing Combination Effects », in : *Batteries* 6.1 (mars 2020), p. 14, DOI : 10.3390/batteries6010014, (visité le 07/09/2021).
- [148] *Règlement (UE) 2019/631 en ce qui concerne le renforcement des normes de performance en matière d'émissions de CO2 pour les voitures particulières neuves et les véhicules utilitaires légers neufs conformément à l'ambition accrue de l'Union en matière de climat*, avec la coll. de Parlement EUROPÉEN, 2021, URL : <https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:52021PC0556> (visité le 12/05/2023).

-
- [149] Jérémie RÉGNIER, « Conception de systèmes hétérogènes en Génie Électrique par optimisation évolutionnaire multicritère », Thèse de doctorat, École nationale supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications, 18 déc. 2003, URL : <http://ethesis.inp-toulouse.fr/archive/00000174/> (visité le 20/09/2021).
- [150] Sundar RENGASAMY et Punniyamoorthy MURUGESAN, « K-means – Laplacian clustering revisited », in : *Engineering Applications of Artificial Intelligence* 107 (1^{er} jan. 2022), p. 104535, ISSN : 0952-1976, DOI : 10.1016/j.engappai.2021.104535, (visité le 17/12/2021).
- [151] Nery RIQUELME-GRANADA, Christian VON LÜCKEN et Benjamin BARAN, *Performance metrics in multi-objective optimization*, 20 oct. 2015, DOI : 10.1109/CLEI.2015.7360024.
- [152] Johan ROCKSTRÖM et al., « Planetary Boundaries : Exploring the Safe Operating Space for Humanity », in : *Ecology and Society* 14.2 (18 nov. 2009), ISSN : 1708-3087, DOI : 10.5751/ES-03180-140232, (visité le 08/05/2023).
- [153] Francesca ROSSI, Peter van BEEK et Toby WALSH, *Handbook of Constraint Programming*, USA : Elsevier Science Inc., juill. 2006, 978 p., ISBN : 978-0-08-046380-3.
- [154] Rami SABBAH et al., « Active (air-cooled) vs. passive (phase change material) thermal management of high power lithium-ion packs : Limitation of temperature rise and uniformity of temperature distribution », in : *Journal of Power Sources*, Selected papers from the International Workshop on Degradation Issues in Fuel Cells 182.2 (1^{er} août 2008), p. 630-638, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2008.03.082, (visité le 25/10/2022).
- [155] Nikolaos V. SAHINIDIS, « BARON : A general purpose global optimization software package », in : *Journal of Global Optimization* 8.2 (1^{er} mars 1996), p. 201-205, ISSN : 1573-2916, DOI : 10.1007/BF00138693, (visité le 13/02/2023).
- [156] Apurba SAKTI et al., « A techno-economic analysis and optimization of Li-ion batteries for light-duty passenger vehicle electrification », in : *Journal of Power Sources* 273 (jan. 2015), p. 966-980, ISSN : 03787753, DOI : 10.1016/j.jpowsour.2014.09.078, (visité le 09/08/2021).
- [157] Yasmine SALEHY, « Knowledge model and design space exploration of sustainable refrigeration systems : Application to the integrated design of supermarket refrigeration systems », Thèse de doctorat, Université Paris-Saclay, 25 jan. 2022, URL : <https://theses.hal.science/tel-03625931> (visité le 19/01/2023).
- [158] E. SARASKETA-ZABALA et al., « Cycle ageing analysis of a LiFePO₄/graphite cell with dynamic model validations : Towards realistic lifetime predictions », in : *Journal of Power Sources* 275 (1^{er} fév. 2015), p. 573-587, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2014.10.153, (visité le 09/02/2022).
- [159] Robert G. SARGENT, « Simulation Model Validation », in : *Simulation and Model-Based Methodologies : An Integrative View*, sous la dir. de Tuncer I. ÖREN, Bernard P. ZEIGLER et Maurice S. ELZAS, NATO ASI Series, Berlin, Heidelberg : Springer, 1984, p. 537-555, ISBN : 978-3-642-82144-8, DOI : 10.1007/978-3-642-82144-8_19.

-
- [160] Robert G. SARGENT, « Verification and validation of simulation models », in : *2007 Winter Simulation Conference*, 2007 Winter Simulation Conference, déc. 2007, p. 124-137, DOI : 10.1109/WSC.2007.4419595.
- [161] Serpil SAYIN, « Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming », in : *Mathematical Programming* 87.3 (1^{er} mai 2000), p. 543-560, ISSN : 1436-4646, DOI : 10.1007/s101070050011, (visité le 20/12/2022).
- [162] Dominique SCARAVETTI, « Formulation préalable d'un problème de conception, pour l'aide à la décision en conception préliminaire », Thèse de doctorat, ENSAM Bordeaux, 2004, 205 p., URL : <https://tel.archives-ouvertes.fr/tel-00008800>.
- [163] O. SCHUTZE et al., « Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization », in : *IEEE Transactions on Evolutionary Computation* 16.4 (août 2012), p. 504-522, ISSN : 1941-0026, DOI : 10.1109/TEVC.2011.2161872.
- [164] Nicolas SCHWIND et al., « Representative Solutions for Multi-Objective Constraint Optimization Problems », in : The 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16), KR, 1^{er} avr. 2016, DOI : 10.11517/pjsai.JSAI2015.0_4E14.
- [165] Aditya A. SHAH et al., « Combining Mathematical Programming and SysML for Automated Component Sizing of Hydraulic Systems », in : *Journal of Computing and Information Science in Engineering* (2010), p. 16, DOI : 10.1115/1.4007764.
- [166] Will STEFFEN et al., « Planetary boundaries : Guiding human development on a changing planet », in : *Science* 347.6223 (13 fév. 2015), p. 1259855, ISSN : 0036-8075, 1095-9203, DOI : 10.1126/science.1259855, (visité le 16/10/2021).
- [167] Ryan H. STEWART, Todd S. PALMER et Bryony DUPONT, « A survey of multi-objective optimization methods and their applications for nuclear scientists and engineers », in : *Progress in Nuclear Energy* 138 (1^{er} août 2021), p. 103830, ISSN : 0149-1970, DOI : 10.1016/j.pnucene.2021.103830, (visité le 12/05/2022).
- [168] O. M. G. SysML, *OMG Systems Modeling Language (OMG SysMLTM)*, 2010, URL : <https://sysml.org/>.
- [169] Ryoji TANABE et Hisao ISHIBUCHI, « An Analysis of Quality Indicators Using Approximated Optimal Distributions in a Three-dimensional Objective Space », in : *IEEE Transactions on Evolutionary Computation* 24.5 (oct. 2020), p. 853-867, ISSN : 1089-778X, 1089-778X, 1941-0026, DOI : 10.1109/TEVC.2020.2966014, arXiv : 2009.12788 [cs], (visité le 01/07/2022).
- [170] R. TARJAN, « Depth-First Search and Linear Graph Algorithms », in : *SIAM Journal on Computing* 1.2 (1972), p. 146-160, DOI : 10.1137/0201010.
- [171] Nicolas TCHERTCHIAN, « Study of methodological approaches for improving environmental performance in complex system design », Thèse de doctorat, Ecole Centrale Paris, juill. 2013, URL : <https://tel.archives-ouvertes.fr/tel-00969164>.
- [172] B.H. THACKER et al., *Concepts of Model Verification and Validation*, LA-14167-MS, United States : Los Alamos National Laboratory, oct. 2004, p. 41.

-
- [173] Noel TITUS et Karthik RAMANI, « Design Space Exploration using Constraint Satisfaction », Configuration Workshop at the 19 », in : *th International Joint Conference on Artificial Intelligence - IJCAI*, 2005, p. 31-36.
- [174] Kenan TOKER et al., « Design, Optimization and Experimental Study of Axial and Hub BLDC Motors in-Wheel Application for Light Electric Vehicles », in : *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), juin 2022, p. 354-359, DOI : 10.1109/MELECON53508.2022.9842975.
- [175] Hélène TOUSSAINT, *IBM ILOG CPLEX Optimization Studio : Une introduction à OPL et Cplex Studio IDE*, 2016.
- [176] Tuan Vu TRAN, « Problèmes combinatoires et modèles multi-niveaux pour la conception optimale des machines électriques », Thèse de doctorat, Lille, France : Ecole Centrale de Lille, 2011, URL : <https://theses.hal.science/tel-00425590v3>.
- [177] G. TROMBETTONI et G. CHABERT, « Constructive Interval Disjunction », in : *International Conference on Principles and Practice of Constraint Programming*, CP'07 - 13th International Conference on Principles and Practice of Constraint Programming, 2007, CP, 2007, DOI : 10.1007/978-3-540-74970-7_45.
- [178] Gilles TROMBETTONI et al., « Inner Regions and Interval Linearizations for Global Optimization », in : *AAAI - XXV*, Association for the Advancement of Artificial Intelligence, t. 25, Palo Alto, California USA : AAAI Press, 2011, p. 7.
- [179] P. VAN HENTENRYCK, D. MCALLESTER et D. KAPUR, « Solving Polynomial Systems Using a Branch and Prune Approach », in : *SIAM Journal on Numerical Analysis* 34 (1997), DOI : 10.1137/S0036142995281504.
- [180] Pascal VAN HENTENRYCK, *Constraint satisfaction in logic programming*, Cambridge, MA, USA : MIT Press, juill. 1989, 224 p., ISBN : 978-0-262-08181-8.
- [181] Elise VAREILLES, « Conception et approches par propagation de contraintes : contribution à la mise en oeuvre d'un outil d'aide interactif », Thèse de doctorat, Institut National Polytechnique de Toulouse, 2005.
- [182] Élise VAREILLES, Michel ALDANONDO et Paul GABORIT, « How to take into account piecewise constraints in constraint satisfaction problems », in : *Engineering Applications of Artificial Intelligence* 22.4 (1^{er} juin 2009), p. 778-785, ISSN : 0952-1976, DOI : 10.1016/j.engappai.2009.01.004.
- [183] Y. VERNAT, « Formalizing and qualifying models for constraint-based embodiment design », Thèse de doctorat, ENSAM Bordeaux, nov. 2004.
- [184] Y. VERNAT, J.-P. NADEAU et P. SÉBASTIAN, « Formalization and qualification of models adapted to preliminary design », in : *International Journal on Interactive Design and Manufacturing (IJ-DeM)* 4.1 (fév. 2010), p. 11-24, ISSN : 1955-2513, 1955-2505, DOI : 10.1007/s12008-009-0081-9, (visité le 09/08/2021).
- [185] Xuan-Ha VU, « Rigorous Solution Techniques for Numerical Constraint Satisfaction Problems », Thèse de doctorat, Lausanne : École Polytechnique fédérale de Lausanne, 2005.

-
- [186] David D. WALDEN et Garry J. ROEDLER, *Systems engineering handbook : a guide for system life cycle processes and activities*, sous la dir. de Kevin FORSBERG et al., 4th edition, Hoboken, New Jersey : Wiley, 2015, 1 p., ISBN : 978-1-118-99941-7.
- [187] Dai WANG et al., « Quantifying electric vehicle battery degradation from driving vs. vehicle-to-grid services », in : *Journal of Power Sources* 332 (15 nov. 2016), p. 193-203, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2016.09.116, (visité le 09/02/2022).
- [188] Jian WANG, Shanming LUO et Deyu SU, « Multi-objective optimal design of cycloid speed reducer based on genetic algorithm », in : *Mechanism and Machine Theory* 102 (1^{er} août 2016), p. 135-148, ISSN : 0094-114X, DOI : 10.1016/j.mechmachtheory.2016.04.007, (visité le 25/04/2023).
- [189] John WANG et al., « Cycle-life model for graphite-LiFePO₄ cells », in : *Journal of Power Sources* 196.8 (15 avr. 2011), p. 3942-3948, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2010.11.134, (visité le 25/01/2022).
- [190] John WANG et al., « Degradation of lithium ion batteries employing graphite negatives and nickel-cobalt-manganese oxide + spinel manganese oxide positives : Part 1, aging mechanisms and life estimation », in : *Journal of Power Sources* 269 (10 déc. 2014), p. 937-948, ISSN : 0378-7753, DOI : 10.1016/j.jpowsour.2014.07.030, (visité le 25/01/2022).
- [191] Lang WANG et Haiqing LI, « Clustering algorithm based on grid and density for data stream », in : MATERIALS SCIENCE, ENERGY TECHNOLOGY, AND POWER ENGINEERING I : 1st International Conference on Materials Science, Energy Technology, Power Engineering (MEP 2017), Hangzhou, China, 2017, p. 020202, DOI : 10.1063/1.4982567, (visité le 15/11/2022).
- [192] Qi WANG et al., « Parameterization of NSGA-II for the Optimal Design of Water Distribution Systems », in : *Water* 11.5 (9 mai 2019), p. 971, ISSN : 2073-4441, DOI : 10.3390/w11050971, (visité le 20/01/2022).
- [193] Adil WAZEER et al., « Phase change materials for battery thermal management of electric and hybrid vehicles : A review », in : *Energy Nexus* 7 (1^{er} sept. 2022), p. 100131, ISSN : 2772-4271, DOI : 10.1016/j.nexus.2022.100131, (visité le 25/10/2022).
- [194] Sabine WOLNY et al., « Thirteen years of SysML : a systematic mapping study », in : *Software and Systems Modeling* 19.1 (1^{er} jan. 2020), p. 111-169, ISSN : 1619-1374, DOI : 10.1007/s10270-019-00735-y, (visité le 15/02/2023).
- [195] Jin WU et Shapour AZARM, « Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set », in : *Journal of Mechanical Design - J MECH DESIGN* 123 (1^{er} mars 2001), DOI : 10.1115/1.1329875.
- [196] Nansi XUE et al., « Design of a lithium-ion battery pack for PHEV using a hybrid optimization method », in : *Applied Energy* 115 (15 fév. 2014), p. 591-602, DOI : 10.1016/j.apenergy.2013.10.044.
- [197] Takao YOKOTA, Takeaki TAGUCHI et Mitsuo GEN, « A solution method for optimal weight design problem of the gear using genetic algorithms », in : *Computers & Industrial Engineering* 35.3-4 (déc. 1998), p. 523-526, ISSN : 03608352, DOI : 10.1016/S0360-8352(98)00149-1, (visité le 25/10/2022).

-
- [198] P. A. YVARS, P LAFON et L ZIMMER, « Optimization of mechanical system : Contribution of constraint satisfaction method », in : *2009 International Conference on Computers & Industrial Engineering*, Industrial Engineering (CIE39), Troyes, France : IEEE, juill. 2009, p. 1379-1384, ISBN : 978-1-4244-4135-8, DOI : 10.1109/ICCIE.2009.5223678, (visité le 09/08/2021).
- [199] P. A. YVARS et L ZIMMER, « DEPS : un langage pour la specification de problemes de conception de systemes », in : *MOSIM'14*, Conférence Francophone de Modélisation, Optimisation et Simulation, Nancy, France, nov. 2014.
- [200] Pierre-Alain YVARS et Laurent ZIMMER, « A Model-based Synthesis approach to system design correct by construction under environmental impact requirements », in : *Procedia CIRP* 103 (2021), p. 85-90, ISSN : 22128271, DOI : 10.1016/j.procir.2021.10.013, (visité le 10/12/2021).
- [201] Pierre-Alain YVARS et Laurent ZIMMER, « Towards a correct by construction design of complex systems : The MBSS approach », in : *Procedia CIRP* 109 (2022), p. 269-274, ISSN : 22128271, DOI : 10.1016/j.procir.2022.05.248, (visité le 30/06/2022).
- [202] Mats ZACKRISSON et al., « Life cycle assessment of lithium-air battery cells », in : *Journal of Cleaner Production* 135 (1^{er} nov. 2016), p. 299-311, ISSN : 0959-6526, DOI : 10.1016/j.jclepro.2016.06.104, (visité le 11/05/2023).
- [203] Gustavo R. ZAVALA et al., « A survey of multi-objective metaheuristics applied to structural optimization », in : *Structural and Multidisciplinary Optimization* 49.4 (1^{er} avr. 2014), p. 537-558, ISSN : 1615-1488, DOI : 10.1007/s00158-013-0996-4, (visité le 14/04/2022).
- [204] S. ZILBERSTEIN, « Using Anytime Algorithms in Intelligent Systems », in : *AI Magazine* 17 (1996), p. 73-83.
- [205] Laurent ZIMMER et al., « Constraint Explorer A Modelling and Sizing Tool for Engineering Design », in : 8th World Multi-Conference on Systemics, Cybernetics and Informatics, t. 15, Orlando, Florida, USA, juill. 2004.
- [206] E. ZITZLER et al., « Performance Assessment of Multiobjective Optimizers : An Analysis and Review », in : *IEEE Transactions on Evolutionary Computation* 7 (mai 2003), p. 117-132, DOI : 10.1109/TEVC.2003.810758.
- [207] Eckart ZITZLER et Lothar THIELE, « Multiobjective optimization using evolutionary algorithms — A comparative case study », in : *Parallel Problem Solving from Nature — PPSN V*, sous la dir. d'Agoston E. EIBEN et al., réd. par G. GOOS, J. HARTMANIS et J. van LEEUWEN, t. 1498, Berlin, Heidelberg : Springer Berlin Heidelberg, 1998, p. 292-301, ISBN : 978-3-540-49672-4, DOI : 10.1007/BFb0056872, (visité le 09/08/2021).
- [208] Ghassan ZUBI et al., « The lithium-ion battery : State of the art and future perspectives », in : *Renewable and Sustainable Energy Reviews* 89 (1^{er} juin 2018), p. 292-308, ISSN : 1364-0321, DOI : 10.1016/j.rser.2018.03.002, (visité le 25/03/2022).

Formalisation et exploration partielle de problèmes de satisfaction de contraintes en conception préliminaire

Mot clés : Exploration partielle diversifiée, CSP, Modélisation, Aide à la décision, Diversité, Conception préliminaire

Résumé : Les décisions prises pendant la conception préliminaire de systèmes complexes comme la voiture électrique jouent un rôle crucial sur ses performances finales. La conception préliminaire repose sur une modélisation mathématique gros grains du système pour étudier la faisabilité des alternatives réalisables. La formalisation et la résolution de ces modèles sont des étapes critiques pour la prise de décision. La première partie de cette thèse propose une démarche progressive pour construire un modèle mathématique dans un contexte de conception. L'objectif est d'obtenir un modèle en adéquation avec le besoin initial et dont le concepteur maîtrise la capacité à aider la prise de décision. Pour résoudre les modèles obtenus, nous proposons de les considérer sous l'angle des Problèmes de Satisfaction de Contraintes

(CSP) et d'explorer intelligemment leur espace de recherche dans une approche d'exploration partielle diversifiée. Il s'agit d'obtenir un échantillon de solutions bien réparties dans l'espace de recherche pour représenter fidèlement l'ensemble des alternatives de conception avec un nombre raisonnable de solutions. Le cadre algorithmique développé repose sur le calcul ensembliste et le parcours d'un arbre de recherche avec des opérateurs de division, réduction et sélection. Les stratégies d'exploration introduites montrent des performances intéressantes vis-à-vis de l'existant. Ce cadre est ensuite appliqué à des cas d'application en conception comme la batterie de véhicule électrique pour mettre en évidence ses intérêts et ses limites comme outil d'aide à la décision.

Modeling and partial design space exploration applied in preliminary design

Keywords: Modeling, Decision-making, Design space exploration, CSP, Anytime branch-and-prune algorithm, Preliminary design, diversity

Abstract: Decisions made during the preliminary design of complex systems such as electric vehicles play a crucial role in their final performance. The preliminary design relies on a coarse-grained mathematical modeling of the system to study the feasibility of viable alternatives. The formalization and resolution of these models are critical steps in the decision-making process. The first part of this thesis proposes a progressive approach to constructing a mathematical model in a design context. The objective is to obtain a model that aligns with the initial requirements and that the

designer has the ability to use effectively in decision-making. To solve the obtained models, we propose to consider them from the perspective of Constraint Satisfaction Problems (CSP) and a diversified partial exploration approach. The aim is to obtain a sample of well-distributed solutions in the search space. The developed algorithmic framework is based on a search tree algorithm using division, reduction, and selection operators. This framework is applied to design applications such as electric vehicle batteries to highlight its strengths and limitations as a decision support tool.