Résolution par satisfaction de contraintes appliquée à l'aide à la décision en conception architecturale

Raphael Chenouard

▶ **To cite this version:**

Raphael Chenouard. Résolution par satisfaction de contraintes appliquée à l'aide à la décision en conception architecturale. Modélisation et simulation. Arts et Métiers ParisTech, 2007. Français. NNT : 2007ENAM0041 . tel-00486662

HAL Id: tel-00486662
https://pastel.hal.science/tel-00486662

Submitted on 26 May 2010

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.

N°: 2007 ENAM 41

Ecole doctorale n° 432 : Sciences des Métiers de l'Ingénieur

T H È S E

pour obtenir le grade de

Docteur
de
l'École Nationale Supérieure d'Arts et Métiers

Spécialité "Informatique – traitement du signal"

présentée et soutenue publiquement
par

Raphaël CHENOUARD

le 17 décembre 2007

RESOLUTION PAR SATISFACTION DE CONTRAINTES APPLIQUEE À L'AIDE À LA DECISION EN CONCEPTION ARCHITECTURALE

Directeur de thèse : ***Patrick SEBASTIAN***

Co-directeur de la thèse : ***Laurent GRANVILLIERS***

Jury :

M. Bernard YANNOU, Professeur, lab. GI, Ecole centrale de Paris.....	Président
M. Michel ALDANONDO, Professeur, ENSTIMAC, Albi.....	Rapporteur
M. Gérard VERFAILLIE, Maître de recherches HDR, ONERA, centre de Toulouse	Rapporteur
M. Patrick SEBASTIAN, Maître de conférence HDR, TREFLE, site ENSAM, Université de Bordeaux I	Directeur de thèse
M. Laurent GRANVILLIERS, Professeur, LINA, Université de Nantes.....	Co-directeur de thèse
M. Jean-Pierre NADEAU, Professeur, TREFLE, site ENSAM, centre de Bordeaux.....	Examinateur

Laboratoire de Transferts Écoulements Fluides Énergétique
ENSAM, CER de Bordeaux

L'ENSAM est un Grand Etablissement dépendant du Ministère de l'Education Nationale, composé de huit centres :
AIX-EN-PROVENCE ANGERS BORDEAUX CHÂLONS-EN-CHAMPAGNE CLUNY LILLE METZ PARIS

Remerciements

Je tiens tout d’abord à remercier mes encadrants de thèse sans qui rien n’aurait été possible : M. Patrick Sébastian, directeur de thèse et M. Laurent Granvilliers, co-directeur de thèse. Merci à tous les deux pour vos qualités humaines, votre disponibilité, le recul et les conseils avisés que vous m'avez donnés sans retenue durant ces trois années.

J’adresse également mes remerciements aux personnes qui ont accepté de participer à mon jury de thèse :

- M. Michel ALDANONDO, professeur à l'école des Mines d'Albi-Carmaux et Gérard VERFAILLIE, maître de recherche HDR à l'Onera de Toulouse, qui m'ont fait l'honneur d'être les rapporteurs de ma thèse.
- M. Bernard YANNOU, professeur à l'école centrale de Paris, pour avoir accepté de présider ce jury et M. Jean-Pierre NADEAU, professeur à l’ENSAM de Bordeaux, pour avoir accepté de faire partie de mon jury de thèse et de l'intérêt qu'il a porté sur mon travail.

Je tiens à remercier toutes les personnes des laboratoires TREFLE et LINA, qui m’ont soutenu et supporté durant cette période de trois ans. Je pense plus particulièrement aux permanents et aux doctorants de l'équipe énergétique et conception et de l'équipe méthodes ensemblistes pour l'optimisation.

Enfin, je tiens à remercier ma famille et ma belle-famille pour leur soutien inébranlable.

A Céline et mes parents.

Table des matières

CHAPITRE I - Introduction générale....1
I.1 - Positionnement de la thèse....3
I.2 - Contexte de la thèse et travaux précédents....4
I.2.1 - Projet RNTL CO2....4
I.2.2 - Outils et travaux existants autour de la conception par contraintes....5
I.3 - Organisation du document....7
CHAPITRE II - État de l'art....9
II.1 - La conception préliminaire....10
II.1.1 - Le processus de conception d'un système mécanique....10
II.1.2 - La phase de recherche de concepts....11
II.1.3 - La phase de conception architecturale....11
II.1.3.1 - Aide à la décision en conception architecturale....12
II.1.3.2 - Modèle en conception architecturale....14
II.2 - La programmation par contraintes....18
II.2.1 - Les principes de base sur les domaines continus....21
II.2.1.1 - Les erreurs d'arrondis....22
II.2.1.2 - Le calcul par intervalles....23
II.2.2 - La propagation de contraintes....26
II.2.2.1 - Les techniques de consistance locale....27
II.2.3 - Les algorithmes de recherche....29
II.2.3.1 - Les heuristiques de recherche....32
II.2.3.2 - L'exploration des domaines des variables....33
CHAPITRE III - Élaboration et exploitation d'un modèle pendant la phase de conception architecturale....35
III.1 - Introduction....36
III.1.1 - Objectifs et contexte de ce type de modèle....36
III.2 Modélisation et aide à la décision en conception architecturale....37
III.2.1 - Adaptation de modèles....39
III.3 - Connaissance formalisée pendant la phase de recherche de concepts....41
III.3.1 - Connaissance formalisée à l'aide du cahier des charges fonctionnel....45
III.4 - Connaissance formalisée pendant la phase de conception architecturale....46
III.5 - Exploitation du modèle et des solutions....49
III.6 - Pourquoi utiliser le paradigme des CSP?....51
III.7 - Points importants pour la réalisation d'un outil d'aide à la décision en conception architecturale....52
III.7.1 - Types des variables en conception....52
III.7.2 - Importance des variables de conception architecturale....54
III.7.3 - Variable explicite ou alias....57
III.7.4 - Précision des variables....58
III.7.5 - Représentation des phénomènes physiques....60
III.7.6 - Gestion des situations de vie....62
III.8 – Bilan....66
CHAPITRE IV - Concepts et apports en programmation par contraintes....67
IV.1 – Etude de l'existant....68
IV.2 – Nouveaux concepts....69
IV.2.1 – Gestion de la précision....69
IV.2.2 – Heuristiques de recherche....72
IV.2.2.1 – Variables de conception....72
IV.2.2.2 – Solution de conception....75
IV.2.2.3 – Apprentissage de la sensibilité....76

IV.2.3 – Contraintes par morceaux....78
IV.3 – Benchmarking....81
IV.4 - Bilan....83
CHAPITRE V - Étude de problèmes de conception....85
V.1 - Introduction....86
V.2 Problèmes de conception....87
V.1.1 - Circuit d'une pompe et de réservoirs....87
V.2.2 - Système de refroidissement d'apéritif....93
V.2.3 - Système de conditionnement d'air sans ailette....98
V.2.4 - Système de conditionnement d'air avec ailettes....107
V.3 - Analyse des performances de résolution....111
V.3.1 – Gestion de la précision des variables auxiliaires....111
V.3.2 - Analyse du ratio de l'heuristique sur les variables de conception....113
V.3.3 – Comparaison globale des performances des heuristiques de recherche....115
V.4 – Bilan....119
CHAPITRE VI - Conclusion et perspectives....121
VI.1 - Contribution apportée....122
VI.2 - Perspectives et futures évolutions....124
Références bibliographiques....129
Annexes....139

Liste des figures

Figure 1 : Modèle simplifié du processus de conception d'un produit industriel....11
Figure 2 : Évolution des coûts engagés et des coûts de développement en fonction de l'avancement dans le cycle de vie d'un produit....14
Figure 3 : Organisation globale des modèles en conception architecturale....17
Figure 4 : Conception préliminaire et combinatoire des alternatives....18
Figure 5 : Représentation de la fonction f(x) = x2 sur l'intervalle [-0,5;1]....25
Figure 6 : Extrait d'arbre de recherche du problème des 6 reines....31
Figure 7 : Cheminement des modèles en conception préliminaire....38
Figure 8 : Exemple de diagramme FAST d'un aspirateur ménager....42
Figure 9: Exemple d'organigramme technique pour un aspirateur ménager....43
Figure 10 : Exemple d'extrait de bloc diagramme fonctionnel pour un aspirateur ménager....44
Figure 11 : Organigramme technique définissant la décomposition organique d'un système de conditionnement d'air dans un avion....47
Figure 12 : Bloc diagramme fonctionnel et phénomènes physiques identifiés pour un échangeur de chaleur dans un système de conditionnement d'air d'avion....47
Figure 13 : Classification des variables dans un modèle de conception architecturale....54
Figure 14 : Expression du facteur de Fanning en fonction du nombre de Reynolds pour l'écoulement d'un fluide dans un tube commercial....61
Figure 15 : Situations de vie d'un produit pendant son cycle de vie....62
Figure 16 : Exemple de situation de vie de fonctionnement pour un système de conditionnement d'air dans un avion....63
Figure 17 : Exemple de réductions pour une contrainte par morceaux de la contrainte globale piecewise....81
Figure 18 : Schéma d'un circuit d'eau comprenant une pompe et trois réservoirs....89
Figure 19 : Solutions du circuit d'eau en considérant les 2 variables de conception....92
Figure 20 : schéma du refroidisseur d'apéritif....94

Figure 21 : Organigramme Technique étendu du doseur.....95
Figure 22 : Solutions en fonction du nombre et de l'espacement des ailettes.....97
Figure 23 : Solutions en fonction du matériau utilisé et de l'espacement des ailettes.....97
Figure 24 : Solutions en fonction des diamètres normalisés des tubes et de l'espacement des ailettes.....97
Figure 25 : Solutions en fonction des largeurs normalisées et de l'espacement des ailettes.....97
Figure 26 : Bloc diagramme fonctionnel d'un système de conditionnement d'air dans un avion. 100
Figure 27 : Exemple d'échangeur thermique formé d'ailettes entre les plaques (intercalaires). 101
Figure 28 : Organigramme technique étendu du système de conditionnement d'air sans ailette. 102
Figure 29 : Variables de décision liées au système de conditionnement d'air : masse et volume de l'échangeur.....105
Figure 30 : Évolution de la température du flux d'air principal dans le SCA.....105
Figure 31 : Évolution de la température du flux d'air principal dans le SCA.....106
Figure 32 : Organigramme technique étendu du système de conditionnement d'air avec ailettes.....107
Figure 33 : 4 familles de surfaces d'échanges utilisées dans la conception des échangeurs thermiques compacts.....108
Figure 34 : Solutions du système de conditionnement d'air en considérant la masse en fonction du volume de l'échangeur, ainsi que les familles de surfaces d'échange pour le côté air principal.....109
Figure 35 : Répartition des solutions en considérant le type de surface d'échange pour le côté air principal et la masse de l'échangeur.....110
Figure 36 : Répartition des solutions en considérant le type de surface d'échange pour le côté air dynamique et la masse de l'échangeur.....111
Figure 37 : Circuit de la pompe et des réservoirs : comparaison du nombre de découpages des domaines en fonction du ratio R de l'heuristique de choix des variables favorisant les VCo.114
Figure 38 : Système de conditionnement d'air : comparaison du nombre de découpages des domaines en fonction du ratio R de l'heuristique de choix des variables favorisant les VCo.114
Figure 39 : Exemples de classes d'objets composant la classe solver d'Elisa représentée suivant la notation UML.....161
Figure 40 : Diagramme de classe UML de la représentation d'un modèle au sein d'Elisa.....162
Figure 41 : Diagramme de classe UML de la représentation d'une contrainte au sein d'Elisa. 163
Figure 42 : Diagramme de classe UML de la représentation d'une variable au sein d'Elisa...163
Figure 43 : Diagramme de classe UML de la représentation d'une réduction au sein d'Elisa. 164
Figure 44 : Diagramme de classe UML de la représentation de la stratégie de choix de variables au sein d'Elisa.....164

Liste des tableaux

Table 1 : Résultats de l'évaluation de f(77617, 33096) en considérant différentes précisions de calculs.....23
Table 2 : Comparatif des performances des heuristiques RR/SF/LS.....82
Table 3 : Caractéristiques principales des problèmes étudiés.....87
Table 4 : Variables de conception du circuit d'eau.....90

Table 5 : Données constructeur à propos du débit, de la hauteur manométrique et du rendement de la pompe.....90
Table 6 : Données constructeur à propos du débit et NPSH requis de la pompe.....91
Table 7 : Variables de conception du refroidisseur/doseur.....96
Table 8 : Variables de conception du SCA sans ailette.....102
Table 9 : Variables de conception du SCA avec ailettes.....108
Table 10 : Comparatif des résultats obtenus en fonction de la précision des variables.....112
Table 11 : Abréviations et descriptions des heuristiques de recherche utilisées.....116
Table 12 : Comparatifs des heuristiques de recherche sur des problèmes de conception.....117

Liste des algorithmes

Algorithme 1 : Algorithme de branch-and-prune.....30
Algorithme 2 : Calcul de la précision relative d'un intervalle.....70
Algorithme 3 : Algorithme de choix des variables de conception d'abord.....73
Algorithme 4 : Algorithme de choix des variables favorisant les variables de conception.....74
Algorithme 5 : Algorithme de backtrack revenant à la dernière VCo découpée.....76
Algorithme 6 : Algorithme de choix des variables apprenant la sensibilité des variables.....78

CHAPITRE I

INTRODUCTION GÉNÉRALE

La phase de conception préliminaire d'un produit est une phase critique du processus de conception. Les principales caractéristiques structurantes et les premières estimations de performances du produit y sont définies. Les modèles utilisés, leur qualité et leur résolution sont alors essentiels pour éviter les risques de mauvaise conception. Les outils disponibles pour le concepteur sont peu nombreux et sont souvent inadaptés au contexte de la conception préliminaire. L'utilisation du paradigme des problèmes de satisfaction de contraintes (CSP) pour exprimer les modèles de conception préliminaire est encore une approche récente. L'idée principale du paradigme des CSP réside dans le fait que l'utilisateur modélise les propriétés d'un système à l'aide de contraintes et un solveur générique les résout. Les contraintes représentent des relations entre variables définissant les associations de valeurs possibles. L'utilisation des CSP en conception préliminaire offre de nombreux avantages par rapport aux autres outils disponibles, dont une grande souplesse dans l'expression de la connaissance et la modification des modèles. Cependant, la généricité de l'approche et du formalisme rend parfois difficile l'expression de connaissances spécifiques et la résolution des modèles n'est pas toujours efficace.

Lors de la formalisation de la connaissance au sein d'un solveur de CSP, un concepteur rencontre généralement des problèmes à plusieurs niveaux. Le concepteur est limité par le formalisme propre au solveur, qui reste générique et ne permet pas d'exprimer certaines spécificités des problèmes de conception. Dans la plupart des solveurs, il ne peut pas définir de composants. Il ne peut pas distinguer les types de variables et aucune hiérarchisation au niveau des variables comme des contraintes n'est possible. De même, certaines notions liées à la résolution au sein du solveur lui échappent. Il n'est pas toujours possible pour le concepteur de définir des précisions pertinentes pour toutes les variables d'un modèle. Une précision globale au modèle formalisé est souvent inappropriée aux modèles issus de la conception préliminaire. Ainsi, l'efficacité de la résolution n'est pas toujours assurée, ce qui ne permet pas de traiter des modèles complexes et dont l'espace de recherche est trop grand. Pour un concepteur, un solveur de contraintes ne représente alors qu'un outil mathématique parmi d'autres, qui permet cependant de résoudre des problèmes exprimés de manière générique. C'est pourtant aussi cet aspect qui intéresse le concepteur, car il exprime la connaissance sans définir comment la traiter. Ainsi, il est plus facile de la manipuler et de la réutiliser, bien que la plupart des solveurs actuels ne soient pas en mesure de permettre de capitaliser la

connaissance exprimée.

I.1 – Positionnement de la thèse

Actuellement, la plupart des problèmes de conception architecturale ne peuvent être résolus efficacement à l'aide des solveurs de contraintes. Même sur des modèles simples, les solutions calculées ne correspondent pas aux attentes d'un concepteur et il lui est difficile d'exploiter de manière pertinente ces résultats pour prendre des décisions importantes sur l'architecture globale d'un produit. Lorsque des problèmes de conception complexes doivent être traités par un solveur de contraintes, comme, par exemple, dans le cas du dimensionnement d'un système de conditionnement d'air, le processus de résolution des solveurs est inefficace et aucune solution n'est calculée au bout de plusieurs heures, contrairement à d'autres approches qui tirent partie des spécificités des problèmes de conception [Seb07].

Le principal problème d'un solveur CSP est qu'il ne tient pas compte des besoins du concepteur. Quelques stratégies peuvent être utilisées pour explorer l'espace de recherche, mais elles ne tiennent pas compte de la connaissance du concepteur par rapport à un produit et au modèle à résoudre. La conception architecturale est une phase du processus de conception qui regroupe des informations hétérogènes sur un produit. Les incertitudes sur les composants et les caractéristiques d'un produit sont encore nombreuses puisqu'il s'agit d'une des premières phases de conception. Les modèles formalisés pendant cette phase sont alors souvent adaptés pour prendre en compte l'apport décisionnel qu'ils représentent pour le concepteur. Pour cela, le concepteur doit évaluer la précision, l'exactitude, la parcimonie et la spécialisation du modèle qu'il manipule. La phase de formalisation d'un modèle est alors fondamentale pour la bonne exploitation du modèle et de ses solutions. Nous avons donc étudié cette phase de modélisation qui nous semble fondamentale pour la résolution efficace d'un problème à l'aide d'un solveur CSP.

Ce travail de thèse est basé sur deux domaines principaux que sont la conception préliminaire en ingénierie mécanique et la programmation par contraintes. L'objectif est d'apporter à ces deux domaines des moyens de mieux inter-opérer pour améliorer la résolution des problèmes de conception et améliorer la qualité des solutions calculées.

Nous avons alors recensé et observé les méthodes utilisées actuellement pour formaliser et résoudre les modèles issus de la phase de conception préliminaire, ce qui nous a permis de proposer de nouvelles approches, que nous avons implémentées dans un solveur de CSP numériques. Nous présentons donc de nouvelles heuristiques de recherche, qui prennent en compte certaines spécificités des problèmes de conception, avec entre autres, les propriétés des variables de conception. Nous présentons aussi les besoins et les problèmes des concepteurs en ce qui concerne la gestion de la précision associée aux variables. Certaines sont fixées par le cahier des charges, alors que d'autres sont difficiles à évaluer à cause de la complexité des phénomènes pris en compte et de leur relations associées dans le modèle. Ces mêmes phénomènes nécessitent dans certains cas de définir un nouveau type de contrainte définie par morceaux. Nous l'avons alors implémenté comme une contrainte globale. De plus, nous avons appliqué ces nouveaux algorithmes et ces nouveaux concepts de modélisation à des problèmes réels, de complexité variable. Nos algorithmes ont alors permis des gains significatifs en temps de résolution, mais aussi en comparaison du nombre de solutions de conception calculées et mises en évidence.

I.2 - Contexte de la thèse et travaux précédents

I.2.1 - Projet RNTL CO2

Le projet RNTL CO2[1] (Réseau National des Technologies Logicielles : projet COnception par COntraintes) qui s'est achevé en juillet 2004, avait pour objectif de développer un environnement de calculs et d'aide à la décision en conception. Ce projet regroupait six laboratoires français : quatre entités académiques (LINA, LIPSI-ESTIA, LIP VI, TREFLE) et deux entités industrielles (DASSAULT-Aviation, CRIL Technology). Ce projet a trouvé sa concrétisation dans le développement d'une librairie au sein du LINA et d'un solveur de Problèmes par Satisfaction de Contraintes numériques *Constraint Explorer* au sein de la société Dassault Aviation [Zim04]. L'équipe du TREFLE a développé de son côté une démarche d'analyse préalable, d'aide à la recherche de caractéristiques structurantes d'un problème de conception en ingénierie mécanique et énergétique [Sca04a, Sca04b] et d'aide à l'expression des contraintes de conception [Ver04]. Ce projet a aussi permis de tester ces

1 http://co2.criltechnology.com/IndexCO2.htm

méthodes et ces outils sur des problèmes de conception réels[2], comme, par exemple, la conception d'un double réducteur ou d'un joint labyrinthique.

Les algorithmes de résolution utilisés au sein de Constraint Explorer sont aussi présents dans la librairie Elisa et le solveur realpaver [Gra06]. Ils sont tous les deux publiés sous une licence libre, ce qui permet d'avoir accès à leur code. C'est au sein de ces deux outils que les algorithmes du chapitre IV ont été développés.

I.2.2 - Outils et travaux existants autour de la conception par contraintes

Les CSP permettent de modéliser de manière assez naturelle les problèmes de conception préliminaire. Les paramètres de conception sont représentés par des variables, alors que les spécifications d'un problème de conception sont exprimées au travers des contraintes d'un CSP [Jan90]. Par exemple, CADRE [Hua96] et IDIOM [Lot98, Smi96] utilisent les techniques de résolution de CSP sur des contraintes décrivant des paramètres géométriques pour améliorer des plans décrivant la disposition d'un plancher d'appartement.

De plus, les CSP ont été utilisés très tôt dans le cadre de la conception collaborative [Ser91, Dar94, Lot99]. Ils permettent de propager les changements effectués entre les différents partenaires d'un même projet. La prise en compte de catalogues et de restrictions de domaines remplacent alors la proposition de valeurs uniques par les collaborateurs d'un même projet. Les CSP sont aussi utilisés pour résoudre des problèmes de conflits et d'aide à la négociation [Bow93, Khe94, Bah95, Kle97].

Quelques années plus tard, les CSP ont été utilisés dans le contexte de la phase de recherche de concepts [OSu01]. Cette utilisation est à rapprocher de celle faite en configuration [Stu97, Far02]. Les concepts ou composants sont décrits principalement à l'aide de variables discrètes. Le concepteur dispose de catalogues, dans lesquels il doit choisir des composants. Le reste du modèle décrit les contraintes liées à l'arrangement des composants et la satisfaction de contraintes liées aux performances globales du produit. La structure des solutions obtenue change en fonction des choix des composants, c'est pourquoi plusieurs dérivations du paradigme CSP sont généralement utilisées : les CSP dynamiques [Mit90] et les CSP composites [Sab96]. Dans le premier cas, les variables et les contraintes ont un état actif/inactif, qui est défini en fonction de contraintes d'activation. Dans le second cas, les

2 http://co2.criltechnology.com/public/experimentation.htm

variables peuvent avoir pour domaines des sous-problèmes, ce qui permet de hiérarchiser les variables et leur domaine en fonction des composants d'un produit.

La configuration d'un produit est un processus interactif, qui prend souvent en compte des composants ou des caractéristiques optionnels ou à choisir dans un catalogue [Jun06]. Lors d'une résolution interactive, c'est l'utilisateur et non la machine qui fait des choix pour restreindre l'espace de recherche du problème [Gel96, Oui02, Van06]. L'utilisateur ajoute ou retire des contraintes pour exprimer ses choix ou ses critères. La cohérence des valeurs des domaines doit alors être calculée rapidement à chaque changement effectué par l'utilisateur [Ami02, Far04].

Cependant, l'utilisation de variables sur les domaines continus a rapidement été nécessaire [Sam96, Ben01, Gel03]. Les problèmes définis sont alors continus ou mixtes : les variables discrètes représentent certains choix de valeurs normalisées ou de composants et les variables continues expriment certaines dimensions ou certaines propriétés des phénomènes physiques. Ces problèmes prennent alors en compte de nouvelles connaissances liées au produit, comme la description de phénomènes physiques [Mul02, Hao05, Mer05, Vu05]. Cependant, le choix de composants et leur arrangement restent l'un des principaux objectifs. La gestion de l'incertitude [Ben00] pendant la phase de recherche de concepts est alors traitée en se basant soit sur les CSP valués [Sch97], soit sur les CSP mixtes au sens de Fargier [Far96].

D'autres travaux en conception architecturale ont permis de montrer l'intérêt de l'utilisation des CSP [Tho94, Fis00, Saw01, Zim01, Had02, Fis04, Sca04a, Ver04, Var05, Arb06, HoK06, Mes06]. Les problèmes de conception traités sont alors plus complexes et les variables liées au produit sont en grande partie définies sur des domaines de valeurs continues, contrairement à la configuration ou à la phase de recherche de concepts. L'espace de recherche lié au produit est alors nettement plus grand et le processus de résolution s'avère peu performant. L'élaboration des modèles et la connaissance qu'ils représentent est alors un travail important pour obtenir des solutions pertinentes [Ver04, Sca04a].

I.3 - Organisation du document

Le chapitre 2 présente un état de l'art des deux domaines d'étude de la thèse. Le

chapitre 3 décrit comment un modèle de conception préliminaire est élaboré et met en évidence quelques notions importantes pour le concepteur. Le chapitre 4 présente les contributions apportées en programmation par contraintes pour résoudre des problèmes de conception mécanique. Le chapitre 5 regroupe quelques exemples de problèmes de conception mécanique, qui sont utilisés comme *benchmarks* pour tester les différents algorithmes de résolution.

CHAPITRE II

ÉTAT DE L'ART

II.1 - La conception préliminaire

La conception préliminaire est une phase du processus de conception (voir figure 1), qui se situe dans les premières phases du cycle de vie d'un produit industriel [Pah96]. Le cycle de vie d'un produit correspond aux différentes étapes que rencontre un produit : expression d'un besoin fonctionnel, conception préliminaire, conception détaillée, réalisation/conformité, vente et fin de vie. Le cahier des charges fonctionnel (CdCF) traduit le besoin fonctionnel et sert de base à la conception préliminaire, qui vise alors à définir l'architecture globale qui sera détaillée par la suite. Le processus de conception est achevé lorsque le dossier de définition du produit est établi en fin de conception détaillée. Il décrit exhaustivement toutes les caractéristiques et les performances du produit. Ensuite des prototypes du produit sont réalisés et leur conformité (au besoin initial, aux normes, etc.) est validée, avant de passer à sa phase d'utilisation par le client et, pour finir sa fin de vie (recyclage, etc.) qui termine le cycle de vie du produit.

Assez peu d'outils de traitement numérique des modèles de dimensionnement et de validation des concepts, aident le concepteur dans les phases amonts du processus de conception [Yan03]. Pourtant ces phases préliminaires sont primordiales d'un point de vue économique et stratégique dans le développement d'un produit. La conception de produit vise à satisfaire autant que possible les objectifs définis dans le cahier des charges. Il est souvent difficile de remplir tous ces objectifs et des compromis sont nécessaires.

II.2.1 - Le processus de conception d'un système mécanique

Le processus de conception est jalonné de plusieurs phases [Pah96] que l'on peut généralement séquencer ainsi (voir figure 1) : expression du besoin, recherche de concepts, conception architecturale (pré-dimensionnement : choix d'architecture, de structure et de topologie du produit), conception détaillée. Résoudre un problème de conception peut alors être considéré comme [Jan90,Sca04a] : partir des spécifications fonctionnelles d'un produit et en construire une représentation respectant :

- des lois physiques,
- des données impératives provenant de la phase de réalisation ou de fabrication du produit :

matériaux ou ressources disponibles par exemple, limitations de coûts, de temps, etc.,

- la prise en compte de critères portant sur le produit (robustesse, coût, simplicité) ou sur sa réalisation (coût de fabrication, gestes métiers, etc.).

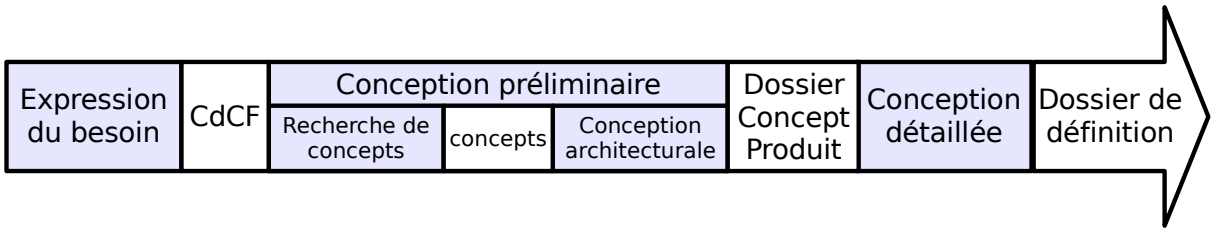


Figure 1 : Modèle simplifié du processus de conception d'un produit industriel.

La conception détaillée (calculs et simulation, plans détaillés, dimensionnement complet, etc.) est souvent bien plus longue et coûteuse que les phases précédentes. Beaucoup d'efforts y sont souvent consacrés aux dépens de la conception préliminaire. Ainsi, peu d'outils d'exploration de concepts ou d'architectures d'un produit existent, contrairement aux logiciels de CAO qui sont très utilisés dans les bureaux d'études. Les logiciels de CAO permettent surtout de définir une représentation géométrique détaillée d'un produit et de ses composants, et offrent la possibilité de faire de la simulation numérique en se basant sur cette géométrie.

De son côté, la conception préliminaire est composée de deux phases : la recherche de concepts et la conception architecturale. Ces phases se basent toutes les deux sur le cahier des charges fonctionnel établi lors de la phase d'analyse du besoin.

II.2.2 - La phase de recherche de concepts

Elle est basée sur l'exploration de concepts ou principes permettant de répondre aux besoins fonctionnels énoncés dans le cahier des charges fonctionnel. Des méthodes liées à la créativité peuvent être utilisées pour trouver des concepts innovants satisfaisant les besoins fonctionnels. Elle aboutit donc à une structure fonctionnelle répondant au besoin identifié, définissant alors un principe de solution.

II.2.3 - La phase de conception architecturale

L'enjeu de cette phase est d'orienter, le plus tôt possible, la conception vers les meilleurs concepts de solution en étudiant leur faisabilité (physique, économique, etc.)

[Yao97]. Les connaissances rassemblées lors de la phase de recherche de concepts sont alors synthétisées dans des modèles permettant d'évaluer la viabilité des architectures possibles. Lors de cette synthèse, apparaissent des formes, des arrangements de composants, des matériaux, etc. Les modèles ainsi définis sont souvent hétérogènes et peuvent s'avérer complexes à traiter, mais ils doivent tenir compte de l'imprécision et des incertitudes existant à ce stade peu avancé de la conception. Ainsi, la phase de conception architecturale vise à réaliser des choix : de technologies, de dimensions principales (grandes dimensions, dimensions de caractéristiques clés : *key characteristics* [Tho99]), de principaux composants, de structure (par exemple : nombre d'étages, agencements) et de topologies (liaisons entre composants, sens et direction des flux). Ces choix aboutissent à une architecture produit principale ou une configuration optimale (du point de vue du modèle traité) par rapport aux critères définis dans le cahier des charges fonctionnel.

II.2.3.1 - Aide à la décision en conception architecturale

La diversité, voire l'hétérogénéité de la connaissance à synthétiser pendant la phase de conception architecturale rend cette phase difficile à aborder. Le concepteur est souvent confronté au début de la conception préliminaire à une difficulté de recensement des connaissances nécessaires et suffisantes pour réaliser cette phase. De plus, il doit prendre à plusieurs reprises des décisions a priori dans un contexte de connaissances imprécises. Pour réduire l'ambiguïté et l'incertitude dans les premières phases du processus de conception, il faut aider le concepteur à écrire un modèle approprié au problème de conception, afin d'aider ensuite à la prise de décision [Sca04b, Ver04]. L'aide à la décision en conception architecturale passe alors par l'utilisation d'outils ou de méthodologies aidant le concepteur à formaliser la connaissance relative à un produit, tout en évaluant cette connaissance, pour estimer les risques encourus et ainsi prendre de « bonnes » décisions.

La hiérarchisation des connaissances est alors primordiale pour ne considérer que les informations pertinentes à ce stade peu avancé de la conception d'un produit [Sca04a]. Cela permet de prendre en compte un maximum d'informations utiles sur le produit sans surcharger les modèles inutilement. La qualité des résultats obtenus est alors améliorée aussi bien du point de vue des performances de résolution que des solutions elles-mêmes. La hiérarchisation des connaissances liées au produit permet ainsi de trier les critères selon leur pertinence, ce

qui permet d'aider le concepteur à mieux évaluer les architectures mises en évidence par le processus de résolution.

Cependant, l'optimisation de la solution retenue ne se fait généralement pas lors de la résolution, mais à posteriori. En effet, les critères de validité d'une solution peuvent évoluer durant le processus de conception et le contexte précoce de la conception préliminaire induit un certain nombre d'incertitudes sur le produit et ses composants. Il est alors plus facile pour le concepteur de considérer un ensemble de solutions, qu'il peut trier suivant les critères qu'il considère à un moment donné. Il faut bien prendre en compte le fait qu'un modèle au sens mathématique du terme ne traduit qu'une partie des connaissances d'un concepteur. « L'optimisation » est alors réalisée à posteriori en prenant en compte des connaissances n'ayant pas été formalisées.

Tout au long du processus de conception, les concepteurs et/ou les décideurs sont amenés à prendre des décisions stratégiques par rapport à un produit. Ces décisions sont soumises à l'appréciation des risques (à tout niveau) encourus. Le risque de conception correspond au fait que le produit conçu ne réponde pas à l'ensemble des exigences. Ces exigences peuvent être très diverses suivant les contextes :

- Spécifications du produit : les directives du cahier des charges ne correspondent pas aux attentes du client.
- Marché : le produit arrive trop tard, les coûts objectifs sont dépassés, le produit ne correspond pas aux critères du marché.
- Technique : (1) la conception proposée ne respecte pas les spécifications, (2) non-compatibilité des interfaces [Cav95], (3) degré de complexité élevé.
- Délais : remise en cause des décisions, reconception.
- Financier.
- Ressources : les ressources de développement ne sont pas disponibles.
- Risque environnemental : le produit, ses composants, les procédés affectent l'environnement.

Les « risques produit » sont principalement supportés par l'utilisateur final durant la phase d'utilisation du produit. Les « risques programme » sont surtout associés aux dérives

des objectifs de performances, de coût, de délai [Cav95]. Les risques encourus peuvent être estimés tout au long du processus de conception en se basant sur les modèles établis à chaque étape, en considérant l'apport qu'ils fournissent pour aider à la prise de décision.

Les décisions prises pendant la phase de conception préliminaire ont bien souvent un impact capital sur le reste du cycle de développement d'un produit. En effet, les décisions prises à la fin de la conception préliminaire engagent généralement plus de 70% des coûts généraux relatifs à un produit, comme le montre la figure suivante :

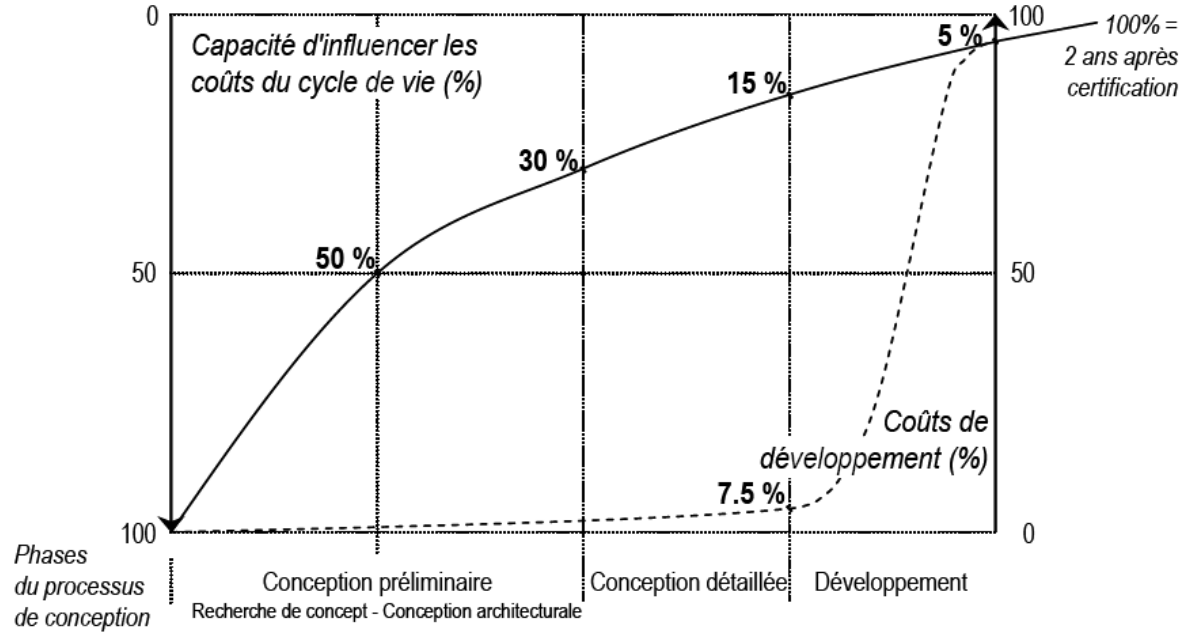


Figure 2 : Évolution des coûts engagés et des coûts de développement en fonction de l'avancement dans le cycle de vie d'un produit [Ber88].

Sur la Figure 2, la courbe en trait pointillé représente l'évolution des coûts de développement du produit durant les phases du processus de conception : les phases de conception préliminaire et détaillée représentent seulement 7.5% du coût global du projet jusqu'à l'industrialisation. L'autre courbe indique la marge de manœuvre financière restant à la fin de chaque phase pour la totalité des coûts du cycle de vie du produit; les choix réalisés durant chaque phase en figeant une partie.

II.2.3.2 - Modèle en conception architecturale

Durant les premières phases du processus de conception, les données sont imprécises, et il est difficile d'utiliser des outils informatiques permettant de faire des calculs exacts et

précis [Ant01]. Les méthodes assistées par ordinateur comme la modélisation solide, l'optimisation, l'analyse de mécanisme, etc., demandent toute une représentation très précise des objets. Dans le cadre de la conception préliminaire, les concepteurs doivent alors faire beaucoup d'hypothèses [Cha92] et restreignent le champ de leurs investigations.

Les outils d'analyse conventionnels ne sont pas adaptés pour étudier une solution de conception incomplète [Saw01]. Ainsi, les concepteurs doivent souvent adopter une démarche empirique du type "essai-erreur" afin de déterminer les valeurs des paramètres de conception. Ils se limitent alors souvent à évaluer les performances d'une solution souvent arbitrairement valuée. Peu d'outils et méthodes s'orientent vers la synthèse automatique ou semi-automatique de concepts [Yan01].

Pendant le stade de la conception préliminaire, des décisions importantes sont prises par les concepteurs/décideurs alors que tous les éléments, permettant de faire des choix raisonnés, ne sont pas accessibles. Des outils de validation de la conception existent pour des phases plus avancées, lorsque la définition du produit est plus complète. La définition d'un modèle permettant de prendre de telles décisions est alors déterminant pour limiter les risques de mauvaise conception.

Les modèles utilisés synthétisent la connaissance sur le produit à chaque étape du processus de conception. Cependant, ces modèles ne peuvent représenter parfaitement le monde réel et le produit (qui d'ailleurs n'existe pas encore) [Pow05]. C'est précisément la particularité d'un modèle. Les modèles sont définis pour représenter des concepts réels ou abstraits. Des simulations ou des expérimentations permettent de les valider si nécessaire. Ils sont utiles lorsqu'un système n'existe pas, quand des expérimentations ne sont pas possibles ou trop coûteuses. L'utilisation des modèles en conception est importante et permet d'aider le concepteur à prendre des décisions. En effet, un concepteur définit des modèles pour répondre à ses besoins décisionnels, les modèles permettant de prédire le comportement d'un produit, son coût, etc. Par la suite, nous considèrerons la définition suivante pour la notion de modèle :

Définition (modèle) : *Un modèle est une abstraction du monde réel ou d'un concept abstrait, qui tient compte d'un nombre fini d'informations. Ce nombre limité d'informations implique de faire des hypothèses simplificatrices quant à la description du sujet du modèle. Il est alors intrinsèquement relié à des niveaux d'exactitude et de précision. Un modèle est défini dans un*

but précis, qui détermine les compromis à effectuer entre la capacité d'un modèle à produire une aide à la décision et son intelligibilité.

L'intelligibilité d'un modèle correspond au fait qu'il soit à la fois expressif et compréhensible pour un concepteur, aussi bien pour chacun de ses éléments le composant, que dans sa globalité. Nous utilisons alors les définitions suivantes pour les termes introduits ici :

Définition (expressivité) : *On parle d'expressivité à propos de variables ou de tout ce qui est associé à une valeur dans un modèle. Une valeur est expressive, si la quantité associée à cette valeur est riche d'informations pour le concepteur.*

Définition (compréhensibilité) : *On parle de compréhensibilité à propos des relations exprimées dans un modèle. Un modèle est compréhensible pour un concepteur, s'il peut s'approprier ces relations et les faire évoluer, les modifier ou les éliminer. La compréhensibilité concerne aussi les ensembles de relations structurées ou agencées que le concepteur doit pouvoir transformer, modifier ou éliminer.*

Il est alors important d'évaluer la qualité d'un modèle par rapport au produit ou à la réalité qu'il représente, ainsi que par rapport aux besoins décisionnels auxquels il doit répondre. La qualification décisionnelle d'un modèle permet d'estimer dans quelle mesure il permet de prendre des décisions en accord avec ses besoins et un contexte décisionnel. Lorsque l'adéquation n'est pas suffisante, le concepteur peut adapter son modèle et limiter autant que possible les risques liés aux décisions à prendre. Ainsi, l'évaluation de l'état décisionnel d'un modèle peut aider le concepteur à gérer les risques qu'il encourt en fonction des décisions qu'il prend et de la connaissance qu'il représente et maîtrise.

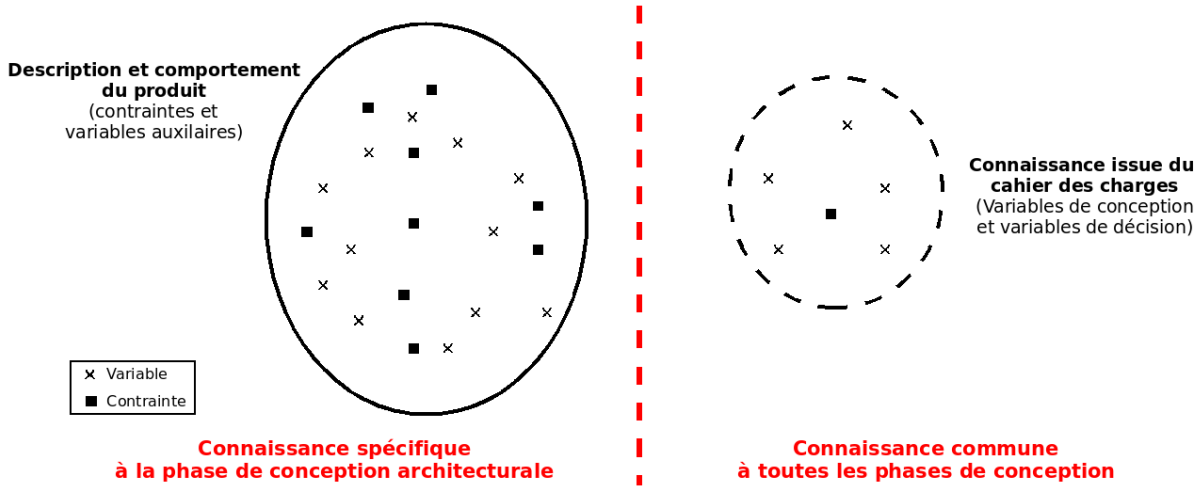


Figure 3 : Organisation globale des modèles en conception architecturale.

La figure 3 illustre l'organisation de la connaissance dans les modèles en conception architecturale. La connaissance regroupée au sein d'un modèle de conception architecturale est basée sur la description du comportement du produit (partie gauche) et sur le cahier des charges (partie droite). Cette dernière regroupe l'ensemble des caractéristiques clés du produit suite à l'analyse fonctionnelle et à l'étude des concepts qui a été faite pendant la phase de recherche de concepts. La connaissance exprimée dans cette partie est commune et partagée par toutes les phases de conception. Seulement, elle ne décrit que l'aspect fonctionnel du produit, sans décrire certains aspects géométriques ou d'autres aspects liés au comportement physique du produit ou de ses composants.

La partie issue du cahier des charges est plus facile à définir, car elle se base sur les phases précédentes du processus de conception et sur l'expérience qu'a le concepteur du cycle de vie complet du produit. Les éléments qui la composent sont généralement peu concernés par la phase d'adaptation de modèle. Par contre, la description du comportement du produit est souvent délicate à établir. Le concepteur doit prendre en compte des modèles décrivant les composants, les phénomènes physiques identifiés, etc. Le concepteur est alors obligé de synthétiser la connaissance pour la définir de manière cohérente dans un modèle global. En effet, la description du comportement du produit ne doit pas être comparable aux simulations menées pendant la phase de conception détaillée. Les incertitudes sur le produit et le temps nécessaire pour établir et résoudre de tels modèles ne sont pas adaptés à la phase de conception architecturale.

La figure 4 illustre la complexité de la phase de conception préliminaire, avec à la fois

les choix que doit faire le concepteur au niveau des concepts de fonctionnement, mais aussi lors de la modélisation d'un même concept. Les choix du concepteur ne s'arrêtent pas là, puisqu'il doit ensuite choisir une architecture parmi les solutions obtenues. Dans l'idéal, il faudrait étudier chaque alternative de concepts de fonctionnement, ainsi que chaque alternative de modélisation, mais dans un contexte industriel, le concepteur n'a pas le temps. Il fait alors des choix à priori, basés sur sa propre expérience et sur les besoins décisionnels liés au modèle qu'il doit définir. Pour le choix d'une alternative de dimensionnement ou d'une architecture, le concepteur peut s'aider de méthodes permettant de trier et de discriminer les solutions les unes par rapport aux autres, par rapport aux critères du cahier des charges et à une analyse de robustesse [Sca04a].

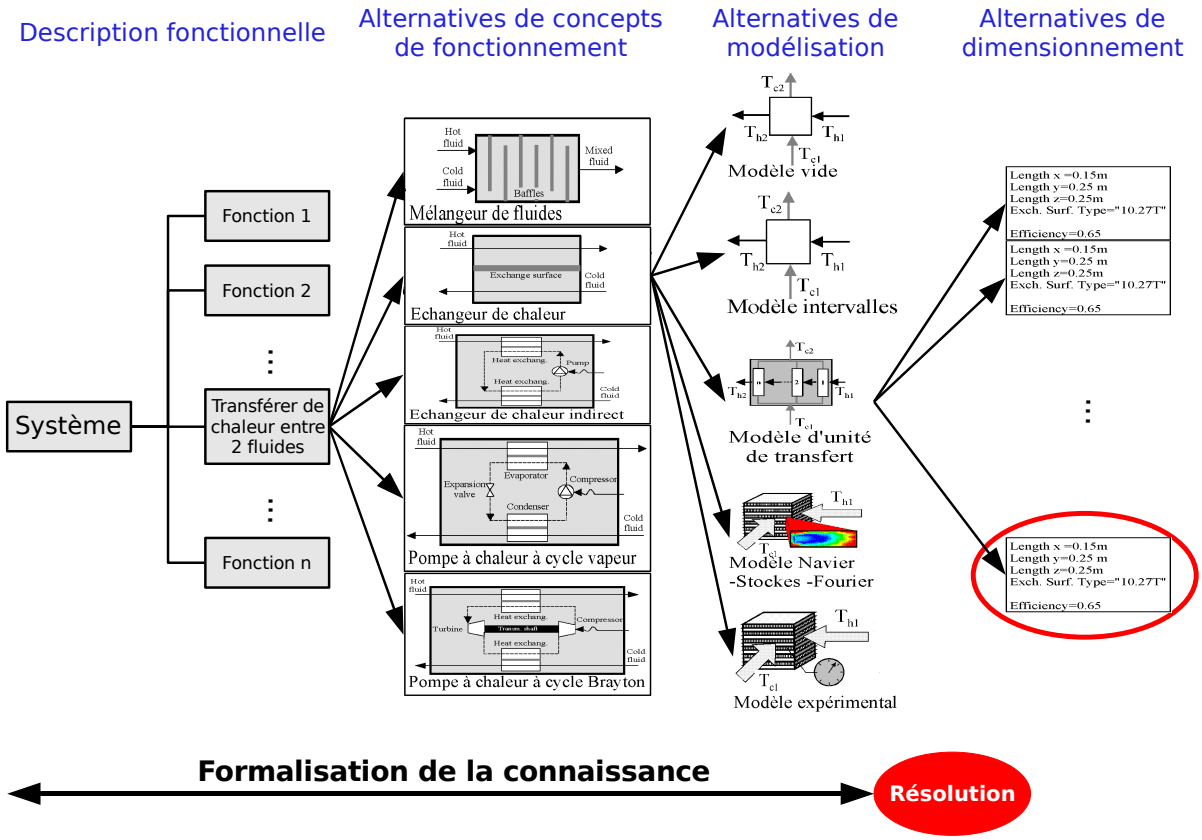


Figure 4 : Conception préliminaire et combinatoire des alternatives.

II.2 – La programmation par contraintes

La programmation par contraintes est un paradigme expressif et puissant qui permet de représenter, résoudre et raisonner sur de nombreux problèmes complexes. Elle est généralement basée sur l'utilisation du formalisme des problèmes de satisfaction de

contraintes (CSP) pour énoncer des problèmes. Elle regroupe l'ensemble des techniques liées à la résolution/satisfaction d'un ensemble de contraintes. Un CSP est traité à l'aide d'un solveur qui calcule un ensemble de solutions, qui explore l'espace de recherche défini par les domaines de variables, tout en s'assurant que les contraintes formulées restent cohérentes.

Les premiers travaux portant sur des techniques de satisfaction de contraintes ont principalement été amenés par le domaine de l'infographie [Mon74, Wal75]. Le concept de contrainte est une notion omniprésente dans notre environnement, une contrainte définissant une restriction sur un ou plusieurs éléments (abstraits ou concrets). Une contrainte peut définir une relation entre deux points d'une figure géométrique, si nous restons dans le domaine de l'infographie, mais elle peut tout aussi bien exprimer une limite de coût pour un produit, la déformation d'un objet suite à l'application d'un effort, etc. On peut définir une contrainte dès qu'il s'agit de restreindre un degré de liberté lié à l'évolution d'une variable caractéristique d'un objet. La difficulté est alors de pouvoir exprimer les restrictions relatives à un sujet d'étude de manière à utiliser efficacement un solveur générique de CSP, comme lors de la description d'un produit et de toutes les contraintes à prendre en compte pour étudier sa faisabilité. La généricité du paradigme des CSP permet de définir un cadre général pour la formalisation des problèmes. Ils sont décrits par des contraintes, qui s'appliquent à des variables associées à un domaine, qui représente l'ensemble de valeurs possible d'un domaine de calcul.

Définition (Contrainte) : *Une contrainte c s'applique à un ensemble de variables* $x_1,...,x_k$*, noté var(c), associé à leurs domaines respectifs* $d_1,...,d_k$*. La contrainte c est alors égale à un sous-ensemble du produit cartésien des domaines des variables :* $c(x_1,...,x_k) \subseteq d_1 \times \cdots \times d_k$.

Une contrainte peut être définie en extension (tous les tuples ou l'ensemble des valeurs autorisées), ou en intention (expression synthétique de la restriction sur les domaines des variables). C'est en général cette dernière approche qui est utilisée, en particulier pour les problèmes de grande taille, car elle permet d'exprimer un problème de manière plus concise en décrivant uniquement les variables, les domaines et les relations existantes entre les variables d'un problème. C'est aussi cette approche qui est utilisée pour décrire des CSP numériques, car il n'est pas possible d'énumérer toutes les combinaisons de valeurs possibles pour des intervalles sur les nombres réels.

Définition (CSP) : *Un problème de satisfaction de contraintes (CSP) est constitué de trois*

entités :

- *Un ensemble de variables : $X=\{x_1,...,x_n\}$,*
- *Un ensemble de domaines associés aux variables : $D=\{d_1,...,d_n\}$.*
- *Un ensemble de contraintes décrivant les restrictions appliquées aux domaines des variables : $C=\{c_1,...,c_m\}$,*

Initialement, le paradigme des CSP est basé sur des domaines discrets (ensembles finis de valeurs, par exemple : dimensions normalisées, identifiant de composants à choisir, etc.). Ils correspondent aux valeurs qu'il est possible d'affecter aux variables. Cependant d'autres types de domaines peuvent être considérés pour représenter des problèmes réels, comme des domaines continus représentés par des intervalles (par exemple : dimensions, grandeurs physiques, etc.).

Les CSP basés sur des domaines continus sont généralement appelés CSP numériques. En conception architecturale, ces deux types de domaines sont utilisés et dissocient généralement des variables issues du cahier des charges (domaine discret) et des variables spécifiques à la description du comportement physique du produit. Les domaines discrets sont alors principalement utilisés pour décrire des identifiants de composants à choisir, des dimensions ou grandeurs issues de catalogues, etc. alors que les domaines continus sont plus appropriés pour la description de dimensions, de variables d'état liées à des grandeurs physiques (pressions, températures, etc.). Lorsque plusieurs types de domaines sont utilisés conjointement dans un CSP, on parle de CSP mixte. Le traitement d'un problème mixte est encore difficile au vu des techniques actuelles [Gel03]. Par la suite, les variables discrètes, présentes dans les problèmes de conception, seront traitées comme des variables continues raffinées au fur et à mesure à l'aide des valeurs discrètes de leur domaine initial.

Une contrainte est considérée comme satisfaite par des domaines de variables, lorsque ces domaines sont cohérents avec la restriction définie par la contrainte. D'un point de vue ensembliste, le produit cartésien des k domaines de variables est inclus dans l'image de la relation ρ_c représentant une contrainte c :

$$d_1 \times \cdots \times d_k \subseteq \rho_c(x_1,...,x_k).$$

On note alors $c(d_1,...,d_k)$ le fait que c est satisfaite par les domaines des variables et

$\neg c(d_1,...,d_k)$ dans le cas contraire. Une solution d'un CSP est représentée par un tuple de valeurs associées aux variables, telles que toutes les contraintes d'un problème sont satisfaites. Lorsqu'un problème ne dispose pas de solution, il est qualifié d'inconsistant.

Un CSP peut être considéré comme une expression logique correspondant à une conjonction de toutes les contraintes de l'ensemble C. Les contraintes peuvent alors être vues comme des prédicats, ce qui permet, dans certains cas, de traiter des problèmes de satisfiabilité de formules logiques [Ben04, Mar06].

Deux CSP sont dits équivalents s'ils ont le même ensemble de solutions (et donc le même ensemble de variables) :

$$Sol(CSP_1) = Sol(CSP_2) \rightarrow CSP_1 \equiv CSP_2.$$

Cette équivalence entre problèmes est la base de la phase de propagation de contraintes qui permet, quand elle est complète, de remplacer un problème initial par un problème équivalent de plus petite taille.

La résolution d'un CSP fait généralement appel à deux types d'algorithmes : les algorithmes de propagation basés sur des opérateurs de réduction et des algorithmes de recherche. Ces deux types d'algorithmes sont suffisamment génériques pour s'appliquer à tout type de problèmes, à partir du moment où ils respectent le formalisme des CSP. Dans le cadre des CSP numériques, la résolution complète d'un problème est assurée par des techniques de recherche exhaustive, qui calcule un ensemble de solutions définies comme des produits cartésiens d'intervalles, appelés boîtes. La phase de recherche est accélérée à l'aide d'algorithmes de propagation de contraintes. La combinaison des techniques de consistances locales à une arithmétique des intervalles rigoureuse permet alors de fournir un encadrement, à une précision fixée, de l'ensemble des solutions [Ben94].

II.2.1 - Les principes de base sur les CSP numériques

Les CSP numériques sont basés sur des contraintes exprimées en intention comme des relations sur des variables définies sur les nombres réels. Le traitement des domaines continus peut se faire suivant deux approches [Ben06] : une discrétisation des domaines continus sur l'ensemble des nombres flottants suivi de calculs algébriques ou une approche basé sur du calcul ensembliste. C'est cette approche que nous développerons par la suite. Elle fait appel à

l'analyse par intervalles et à l'algorithmique qui en découle [Moo66]. Cependant, les domaines continus sont représentés par des intervalles dont les bornes sont des nombres réels, ce qui pose de nombreux problèmes quant à leur représentation informatique et leur utilisation dans les calculs [Ric68, Wan74]. Les nombres réels ne peuvent être tous définis à l'aide d'une représentation basée sur un nombre fini de bits en mémoire. La principale conséquence de cette représentation est l'erreur d'arrondi. A partir de là, le but principal des techniques basées sur les intervalles est de résoudre des problèmes définis comme des relaxations de problèmes continus. Un autre problème important est de prouver l'existence de solutions dans une boîte intervalle, pour garantir les solutions calculées.

II.2.1.1 - Les erreurs d'arrondis

Les intervalles définissant les domaines continus sont bornés par des nombres réels. Ces derniers ne sont pas tous représentables en mémoire, car ils respectent la norme IEEE754 [IEEE754]. Les nombres définis par cette norme sont calculés à l'aide d'un bit de signe, d'une mantisse et d'un exposant:

$$f = (-1)^s \cdot m \cdot 2^E.$$

Ces nombres sont appelés nombres flottants. L'ensemble des nombres flottants est noté $\mathbb{F}$ et est inclus dans l'ensemble des nombres réels $\mathbb{R}$. Lors des calculs, les nombres réels sont arrondis aux nombres flottants les plus proches, ce qui provoque des erreurs de calculs plus ou moins importantes suivant les valeurs et les opérations considérées.

Prenons l'exemple de la fonction de Rump [Rum88] : $f(x, y) = 333.75y^6 + x^2(11x^2 y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + x/(2y)$. A l'aide de l'outil MuPad 2.5.3 sur un processeur Intel Pentium M, nous obtenons les résultats suivants pour l'évaluation de *f(77617, 33096)* décrit dans la table 1.

La valeur de *f(77617; 33096*) est -54767/66192, soit environ -0,8273960599. D'après les résultats de ce tableau, nous voyons clairement les écarts (parfois énormes) avec la solution réelle, qui peuvent avoir lieu à cause des erreurs d'arrondis.

Table 1 : Résultats de l'évaluation de f(77617, 33096) en considérant différentes précisions de calculs.

Précision de 7 chiffres	-2,47588e27
Précision de 16 chiffres	-5,7646075226366 51e17
Précision de 24 chiffres	-134217728,0
Précision de 32 chiffres	-0.84375
Précision de 64 chiffres	-0.8273960599...

II.2.1.2 - Le calcul par intervalles

Le calcul par intervalles [Moo66] permet de faire des calculs sur des ensembles de valeurs, tout en gérant les erreurs d'arrondis. Cependant, les calculs ne se font plus sur des nombres (réels ou flottants), mais sur des intervalles représentés par leur deux bornes.

Définition (Intervalle) : *On considère* $\mathbb{R} \cup \{-\infty, +\infty\}$ *l'ensemble des réels auquel on a ajouté* $+\infty$ *et* $-\infty$*, et la relation* $\leq$ *sur cet ensemble.*

$$\forall a, b \in \mathbb{R} \cup \{-\infty, +\infty\}, [a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}.$$

Pour tous réels a et b, l'intervalle des nombres entre a et b compris, se note [a, b]. On note $\mathbb{I}$ *l'ensemble des intervalles à bornes réelles.*

Pour plus de simplicité, on ne considère que les intervalles fermés, bien qu'il soit possible de généraliser aux autres intervalles [Cle87, Ben97]. Pour un intervalle I, on note inf(I) sa borne inférieure et sup(I) sa borne supérieure.

Le calcul sur les intervalles nécessite de redéfinir les opérateurs arithmétiques usuels, de manière à ce que le calcul par intervalles garantisse la validité des valeurs calculées. Par exemple, pour les quatre opérateurs arithmétiques usuels sur les intervalles à bornes réelles, nous obtenons :

$$[a, b] + [c, d] = [a + c, b + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[a, b]/[c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)], 0 \notin [c, d]$$

Nous pouvons noter que certaines propriétés sur les réels ne sont plus applicables aux intervalles. C'est en particulier le cas de la distributivité de l'addition par rapport à la multiplication.

Les fonctions sur les réels, elles aussi, doivent être redéfinies pour les intervalles. Deux approches sont alors possibles suivant la monotonicité de la fonction sur son domaine de validité :

- elle est monotone : il suffit d'appliquer la fonction aux bornes de l'intervalle de calcul pour obtenir les nouvelles bornes, en les inversant dans le cas d'une fonction décroissante.
- elle n'est pas monotone : il faut faire une étude de variation de la fonction afin d'obtenir ses valeurs minimum et maximum, qui serviront alors à calculer les bornes du nouvel intervalle.

Exemple : Soit la fonction $f(x) = x^2$, étudiée pour x dans l'intervalle I = [-0,5;1].

L'évaluation des bornes de x permet d'obtenir l'encadrement : [0,25;1].

Cet encadrement n'est pas l'intervalle d'évolution de $f(x)$ sur I comme le montre la figure 5. L'étude de la fonction f permet de définir sa borne minimale et sa borne maximale sur I, ici 0 et 1, ce qui nous permet d'obtenir l'image de f sur I en évaluant la fonction pour ces valeurs : [0,1].

Jusqu'à présent, les intervalles considérés ont des bornes réelles et leur représentation sur un ordinateur pose les mêmes problèmes que les nombres réels. Cependant, pour garantir les calculs, les intervalles à bornes réelles sont arrondis par des intervalles englobant dont les bornes sont des nombres flottants. De même pour les nombres réels, nous pouvons les arrondir par des intervalles dont les bornes sont des nombres flottants, ce qui permet de conserver la valeur réelle approximée dans un intervalle d'étude.

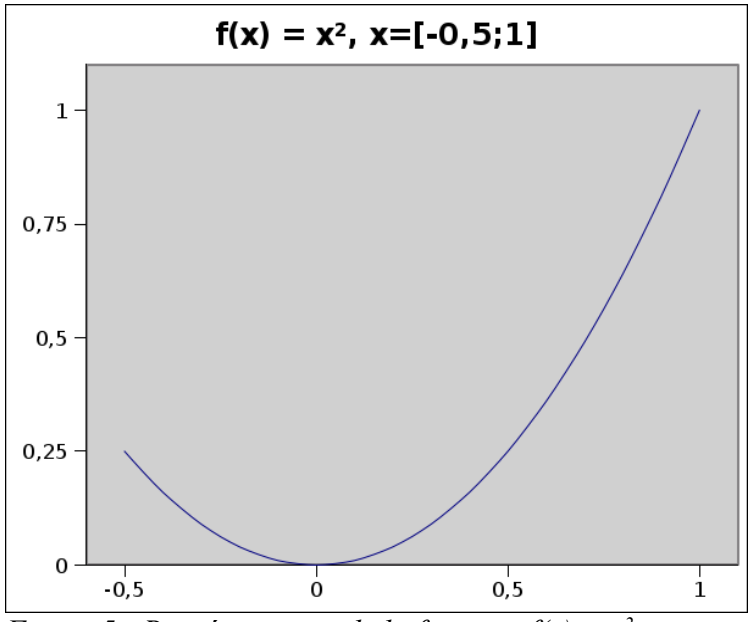


Figure 5 : Représentation de la fonction f(x) = x^2 sur l'intervalle [-0,5;1].

Soit a un nombre réel, on note alors a^+ le plus petit élément de $\mathbb{F}$ qui est plus grand que le nombre réel a, et a^- le plus grand élément de $\mathbb{F}$ qui est plus petit que a. Les opérations arithmétiques précédentes deviennent alors :

$$[a, b] + [c, d] = [(a + c)^- , (b + d)^+]$$

$$[a, b] - [c, d] = [(a - d)^- , (b - c)^+]$$

$$[a, b] \times [c, d] = [min(ac, ad, bc, bd)^- ,max(ac, ad, bc, bd)^+]$$

$$[a, b]/[c, d] = [min(a/c, a/d, b/c, b/d)^- ,max(a/c, a/d, b/c, b/d)^+], 0 \notin [c, d]$$

Intuitivement, on peut approximer un intervalle de réels de deux manières :

- par l'union des plus petits intervalles flottants, pour manipuler des entités calculables,
- ou par un intervalle flottant continu, pour éviter l'explosion combinatoire de l'approche précédente.

C'est généralement cette deuxième solution qui est mise en oeuvre, mais elle nécessite l'utilisation d'un opérateur ou d'une fonction permettant d'obtenir un intervalle englobant sur les nombres flottants d'un intervalle sur les réels.

Définition (*hull*) : *Soit hull la fonction calculant le plus petit élément de l'ensemble des intervalles flottants $\mathbb{I}_\mathbb{F}$ englobant un réel, un intervalle, ou un ensemble de valeurs :*

$$\forall\, a \subseteq \mathbb{R},\ hull(a) = [a^-, a^+]$$

$$\forall\, [a,b] \subseteq \mathbb{I},\ hull([a,b]) = [a^-, b^+]$$

$$\forall\, \{a_1,...,a_n\} \in \mathbb{R}^n,\ hull(\{a_1,...,a_n\}) = hull([min(a_1,...,a_n), max(a_1,...,a_n)])$$

On peut noter que la fonction *hull* est monotone et idempotente. Elle permet de calculer la plus petite approximation extérieure faisant partie de l'ensemble des intervalles flottants pour un réel, un intervalle ou un ensemble fini de valeurs.

Les contraintes sur les réels d'un CSP numérique sont définies comme des équations ou des inéquations. Leur évaluation est généralement associée à une analyse numérique de fonctions sur les réels. Ces fonctions sont étendues aux intervalles en remplaçant chaque variable réelle par l'intervalle flottant la représentant, ce qui est considéré comme l'extension naturelle aux intervalles. Lorsque des variables apparaissent plusieurs fois dans une même contrainte, la relation entre les variables et les intervalles la représentant sont perdus, ce qui conduit à des approximations moins fines de l'intervalle résultant. Des manipulations symboliques peuvent être mises en oeuvre pour diminuer les effets négatifs de l'occurence multiple des variables : factorisations, mise sous forme de Hörner dans les cas de fonctions polynomiales... D'autres extensions aux intervalles peuvent être utilisées, telles que les formes centrées [Sta95], les mises en forme de Taylor [Gou00], ou les extensions en forme de Bernstein [Gar85].

II.2.2 - La propagation de contraintes

L'objectif de la propagation de contraintes est de remplacer un CSP initial par un CSP équivalent, mais avec un espace de recherche plus restreint. La propagation de contraintes permet, à partir d'un ensemble de contraintes et de domaines, d'inférer des contraintes et des domaines plus restrictifs. Le principe d'une inférence est alors de propager les informations contenues dans une contrainte aux contraintes voisines. Ces inférences sont basées sur l'application de techniques de cohérence de domaines, aussi appelées techniques de consistance, qui permettent de filtrer les valeurs ou les ensembles de valeurs qui participent à une solution.

Les techniques de propagation [Mac77] sont une instance du cadre général des itérations chaotiques [Apt99, Mon99]. La propagation de contraintes est souvent appliquée au

travers du calcul d'un point fixe (plus de variations sur les domaines). Elle est basée sur des opérateurs de réduction ou de contraction qui sont monotones et contractants, ce qui garantit la convergence du processus de propagation. Ces opérateurs sont aussi complets, ce qui permet de garantir la complétude de l'ensemble de solutions obtenues. C'est l'enchaînement de l'application de ces opérateurs qui définit un algorithme de propagation [Apt99]. Les opérateurs de réduction implémentent des consistances locales, une consistance globale étant généralement trop coûteuse à appliquer.

II.2.2.1 - Les techniques de consistance

Elles sont aussi appelées techniques de filtrage de domaines. Elles permettent de réduire les domaines des variables en tenant compte de la cohérence des valeurs des domaines entre elles en fonction des contraintes d'un problème. Elles permettent d'éliminer efficacement un grand nombre de valeurs incohérentes d'un domaine par rapport à une contrainte ou à l'ensemble du problème suivant la consistance appliquée. La plus simple est la consistance de borne et la plus utilisée la consistance d'arc ou ses variantes [Mac77, Deb97, Deb01]. Par exemple, nous pouvons considérer la définition suivante de la consistance d'arc :

Définition (Consistance d'arc) : *On dit qu'une contrainte n-aire $c(x_1,...x_n)$ est consistante d'arc, si pour k* dans *{1, ..., n} on a :*

$$\forall a_k \in d_k : \quad \exists a_1 \in d_1, \ldots, \exists a_{k-1} \in d_{k-1}, \exists a_{k+1} \in d_{k+1}, \ldots, \exists a_n \in d_n \textit{ tels que}$$

$$c(a_1, \ldots, a_n)$$

La consistance d'arc permet de vérifier que toutes les valeurs d'un domaine d_k disposent d'au moins une valeur dans les domaines des autres variables, la contrainte *c* étant satisfaite pour ces valeurs. Cette consistance est généralement assez rapide à calculer, mais des consistances plus fines sur les CSP discrets peuvent être appliquées pour éliminer plus de valeurs inconsistantes dans les domaines. On peut ainsi citer la consistance de chemin [Mon74], les k-consistances [Fre78]. En général, les k-consistance (pour k>3) ont une complexité trop élevée pour une utilisation sur la plupart des problèmes.

La consistance d'arc n'est pas applicable en tant que telle dans le cadre des CSP numériques. En effet, l'énumération des valeurs des domaines n'est pas possible sur les intervalles réels. Cependant, la consistance d'arc peut être adaptée et affaiblie pour les

domaines continus en se basant sur l'arithmétique des intervalles flottants et sur le mécanisme de projection des contraintes sur le domaine d'une variable. Les consistances sur les intervalles calculent des ensembles plus larges que ceux définis par la consistance d'arc. Ils ne peuvent que l'approximer à cause du calcul sur des intervalles dont les bornes sont des nombres flottants.

La première consistance issue d'un affaiblissement de la consistance d'arc est la *hull*-consistance (consistance d'enveloppe) ou 2B-consistance [Lho93, Ben95, Ben97] :

Définition (*hull*-consistance) : *Étant donnés une contrainte réelle* $c(x_1,...,x_n)$*, un entier* k dans $\{1,...,n\}$ *et un pavé* $B = I_1 \times \cdots \times I_n$ *de domaines, on dit que la contrainte c est hull-consistante par rapport à la variable* x_k *si et seulement si :*

$$I_k = hull(\pi_k (\rho_c \cap B)),$$

où π_k *correspond à la projection de c sur* x_k *et* ρ_c *correspond à la relation sur les réels associée à c.*

La contrainte *c* est dite *hull*-consistante sur *B* si cette relation est vraie pour tout *k* dans $\{1,...,n\}$.

L'autre consistance sur les intervalles qui est principalement utilisée est la box-consistance (consistance de boîte). Elle est basée sur l'extension aux intervalles des contraintes réelles [Chi97] :

Définition (Box-consistance [Ben99a]) : *Soit c une contrainte réelle n-aire, C une extension aux intervalles de c et* $B = I_1 \times \cdots \times I_n$ *un pavé de domaines. La contrainte c est dite box-consistante par rapport à B et à un entier k* dans $\{1,...,n\}$ *si et seulement si :*

$$I_k = hull(\{a_k \in I_k \mid C(I_1,...,I_{k-1},hull(\{a_k\}),I_{k+1},...,I_n)\}).$$

L'application de la box-consistance permet en général d'obtenir une meilleure approximation de l'ensemble de solutions que la hull-consistance lorsque les contraintes ont une arité supérieure à trois. C'est généralement le cas pour les modèles définis en conception préliminaire qui comportent souvent des contraintes complexes. Afin de rendre la box-consistance plus rapide, elle a été affaiblie en évaluant non plus l'enveloppe canonique de chaque valeur, mais un intervalle de taille φ [Gra99]. Une bonne alternative est alors de combiner ces deux consistances en utilisant des opérateurs de réduction qui implémentent la

hull-consistance pour les contraintes primitives (arités < 3) et des opérateurs qui implémentent la box-consistance pour les autres contraintes. Ce compromis permet en général d'obtenir de meilleurs résultats en considérent les temps de calculs et l'approximation extérieure obtenue des contraintes réelles du problème.

La 3B-consistance est une relaxation de la consistance de chemin, qui vérifie que la hull-consistance peut être appliquée quand le domaine d'une variable est réduit à une de ses bornes dans tout le problème [Lho93]. Nous ne détaillerons pas la notion de kB-consistance, car seules la 2B-consistance (équivalente à la *hull*-consistance) et la 3B-consistance sont utilisables dans des problèmes concrets.

Récemment des travaux basés sur la disjonction constructive sur les intervalles et utilisant la hull-consistance et la 3B-consistance permettent d'obtenir des résultats intéressants [Tro07]. En fait, la *hull*-consistance est appliquée sur un nombre *n* de morceaux, dont on calcule ensuite l'enveloppe englobante à l'aide de l'opérateur *hull*. Ces morceaux sont obtenus en découpant un domaine en *n* parts équitables. Il faut alors noter que le résultats des réductions sur ces morceaux d'un intervalle sont ensuite utilisés pour découper de manière plus pertinente le domaine des variables pendant la phase de recherche.

Nous pouvons noter que les techniques définies pour réduire les domaines continus sont applicables aux domaines discrets. Les techniques de consistance sur les domaines continus sont plus faibles en comparaison des consistances souvent appliquées sur les problèmes discrets, qui exploitent toutes les valeurs des domaines et pas seulement leurs bornes.

II.2.3 - Les algorithmes de recherche

Ils permettent d'explorer exhaustivement l'espace de recherche d'un problème. Cette exploration est généralement représentée sous la forme d'un arbre, où la racine est le problème initial et chaque autre noeud est un problème plus petit. Les algorithmes de recherche définissent la manière dont est généré/exploré cet arbre. La construction de cet arbre se fait en découpant le domaine d'une variable de manière à obtenir plusieurs sous-problèmes à traiter séparément. Un noeud père est donc un problème équivalent à l'union des problèmes représentés par ses noeuds fils.

Il existe alors principalement deux stratégies pour le parcours de l'arbre de recherche : en profondeur DFS (*depth-first search*) et en largeur LFS (*left-first search*). En plus de cette stratégie globale d'exploration de l'arbre de recherche, plusieurs algorithmes de génération/exploration de l'arbre existent. L'approche la plus basique, pour définir et explorer un arbre de recherche, est le *generate&test*. Cet algorithme consiste à énumérer exhaustivement toutes les valeurs possibles pour toutes les variables du problème. Ensuite la consistance des feuilles de l'arbre, qui correspondent alors à des solutions, est vérifiée. Cette approche correspond au pire des cas pour tous les problèmes et est très coûteuse.

Une approche plus judicieuse consiste alors à vérifier la consistance des problèmes à chaque noeud de l'arbre, c'est l'algorithme *backtrack search* [Fre82]. Dès qu'une inconsistance est détectée, on remonte dans l'arbre de recherche pour explorer les prochaines branches. Cela permet d'éviter d'explorer un grand nombre de noeuds inutiles, tout en garantissant la complétude de la recherche. On parle alors principalement d'algorithme de *branch-and-reduce* ou *branch-and-prune* (voir algorithme 1). Différents algorithmes pour les domaines discrets et basés sur des variantes plus ou moins importantes du *backtrack search* sont aussi utilisés, comme : *forward checking, partial look ahead, full look ahead*, etc [Ver95, Pet97, Apt03, Ros06].

Algorithme 1 : Algorithme de branch-and-prune [Van97a].

```
Solve (θ: opérateur de contraction, B: boîte intervalle)
Début
      B_θ := θ(B)
      Si B_θ est vide alors
            Retourner ∅
      Sinon si B_θ est assez précis alors
            Retourner B_θ
      Sinon
            (B_θ^(1), B_θ^(2)):= Split(B_θ)
            Retourner Solve(B_θ^(1)θ,) ∪ Solve(B_θ^(2)θ,)
      Fin si
Fin
```

L'algorithme 1 prend en entrée un ensemble d'opérateurs de contraction et une boîte (un ensemble d'intervalles). Cette dernière est réduite en utilisant l'opérateur de contraction qui lui est associé (ligne 1). Cette nouvelle boîte est retournée si elle est suffisamment précise (ligne 5), c'est-à-dire : la précision requise pour toutes les variables est atteinte. Dans le cas contraire, la boîte réduite est découpée en deux parties et l'algorithme de résolution est appliqué à ces deux sous-boîtes (ligne 8). Un domaine est découpé seulement s'il contient au moins trois nombres flottants, et si sa précision (absolue ou relative) est supérieure à celle fixée pour la variable correspondante.

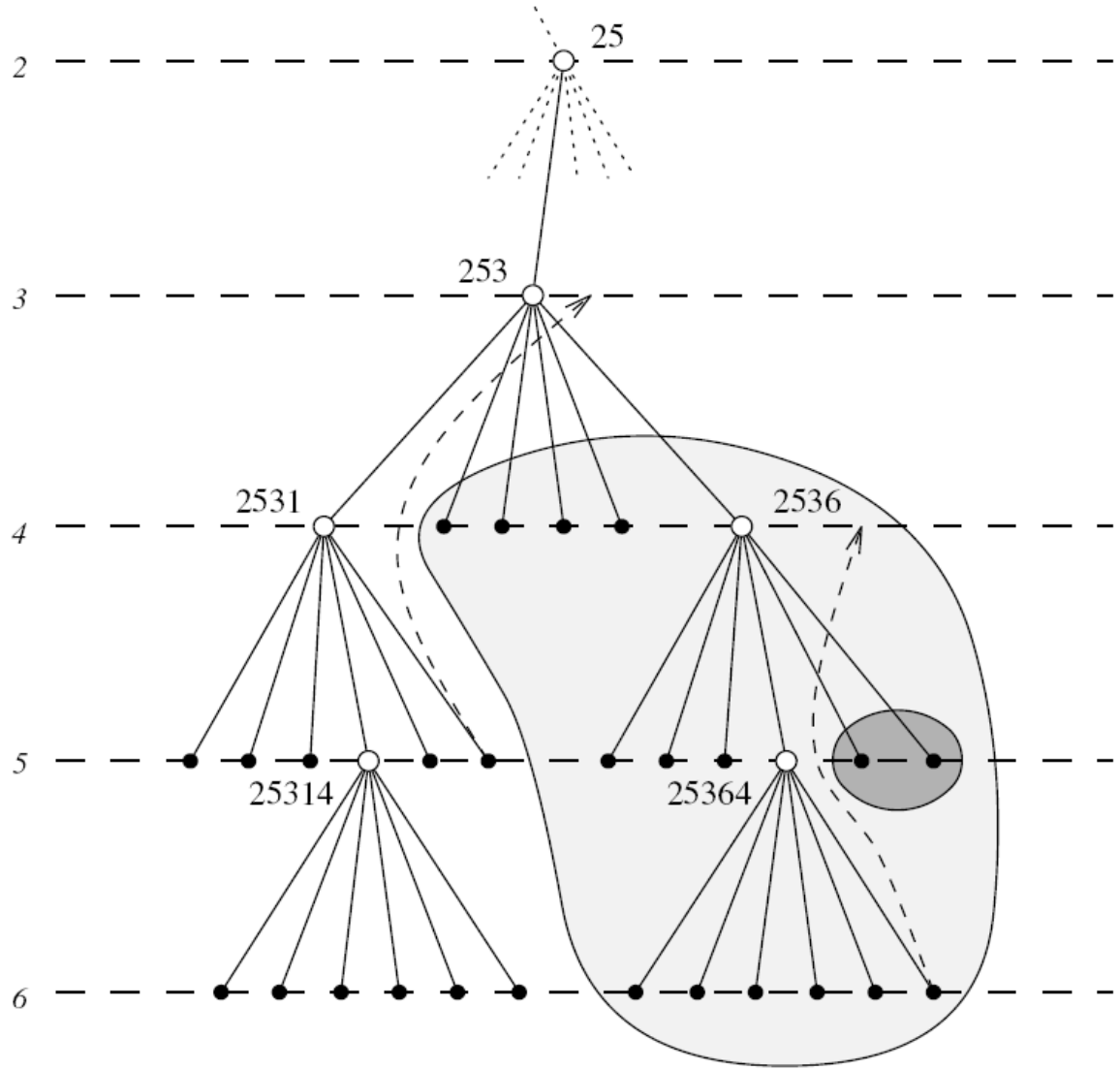


Figure 6 : Extrait d'arbre de recherche du problème des 6 reines [Kon97,Ros06].

La figure 6 représente alors un exemple d'arbre de recherche correspondant au

problème des 6 reines. Cette arbre de recherche a été construit en se basant sur l'algorithme naïf de *backtrack*. Les noeuds noirs correspondent à des solutions lorsqu'ils sont de niveau 6. Dans le cas contraire, une inconsistance a été détectée et le processus d'exploration est revenu en arrière pour explorer d'autres branches encore inexplorées de l'arbre de recherche.

Dans le cadre des CSP numériques, il est important de noter que l'exploration de l'espace de recherche est limité par une précision définie sur les domaines intervalles. Cette précision, définie de manière absolue, correspond à la largeur de l'intervalle à atteindre pour considérer qu'une solution est trouvée.

II.2.3.1 - Les heuristiques de recherche

La génération de nouveaux noeuds dans l'arbre de recherche fait aussi appel à différentes heuristiques liées au choix des variables et à la manière dont leur domaine est découpé ou énuméré. Le choix de la prochaine variable à découper est souvent crucial pour l'efficacité de l'algorithme de résolution.

Une stratégie de tourniquet (ou *Round-Robin*) peut être appliquée : les variables sont choisies tour à tour de manière équitable, tant que leur domaine peut être découpé. Cette heuristique est simple à mettre en oeuvre et souvent la plus robuste, mais elle n'exploite pas la structure du problème ou d'autres informations liées aux variables ou à leur domaine. Ainsi, d'autres heuristiques assez simples à mettre en oeuvre peuvent être utilisées :

- *fail-first* : la variable de plus petit domaine est choisie. Cette heuristique est connue pour être mieux adaptée aux domaines discrets.
- *reduce-first* : la variable de plus grand domaine est choisie. Cette heuristique est connue pour être mieux adaptée aux domaines continus.
- *most-constrained variable* : la variable la plus contrainte du problème est choisie. C'est celle qui apparaît dans le plus de contraintes et qui possède donc le plus d'opérateurs de réduction.

D'autres heuristiques plus complexes tiennent compte de la structure du problème et du réseau de contraintes. Le graphe de contraintes associé à un CSP peut être décomposé en blocs, formant un graphe orienté acyclique (DAG) [Red96,Bli98,Nev06]. Ce DAG définit alors un ordre partiel entre les blocs et donc entre les variables à choisir. Cette approche

s'avère très performante pour les problèmes continus bien contraints (autant de variables que de contraintes) et ne comportant pas de trop grands ensembles de variables couplées.

De même pour les CSP continus, la *maximal smear strategy* utilise l'analyse numérique pour choisir la variable maximisant la *smear function* [Kea95]. Pour chaque variable x_k, la *smear function* est définie par :

$$s_k = max\{max\{|inf(J_{i,j})|, |sup(J_{i,j})|\}.w(x_i)\}, \quad 1 \leq j \leq m,$$

où $J_{i,j} = [|inf(J_{i,j})|, |sup(J_{i,j})|]$ correspond à la (i,j)ème entrée dans l'extension intervalle de la matrice jacobienne du problème et $w(x_i)$ représente la taille du domaine de x_i.

Cette heuristique vise à choisir la variable la plus sensible numériquement dans un problème. Elle peut être très efficace, mais la forme des fonctions associées aux contraintes pose parfois problème pour calculer les gradients des variables. Ils ne représentent alors pas pertinemment la sensibilité numérique de la variable dans le problème. De plus, les problèmes de conception architecturale comportent souvent des variables définies sur des domaines discrets, ce qui tend à fausser cette heuristique.

Aucune de ces heuristiques ne s'avère idéale pour tous les problèmes, et il faudrait bien souvent pouvoir exploiter les spécificités de chaque problème pour obtenir un arbre de recherche optimal.

II.2.3.2 - L'exploration des domaines des variables

Différentes stratégies peuvent être définies suivant le type des domaines. Les domaines continus sont généralement découpés en utilisant la bissection, c'est-à-dire qu'un intervalle est découpé en son milieu afin d'obtenir deux intervalles de taille équivalente. On peut aussi énumérer des petits intervalles, dont la taille correspond à la précision des variables, ce qui s'apparente à l'énumération des valeurs généralement faites sur les domaines discrets. Des valeurs particulières peuvent aussi être choisies au sein des domaines des variables, comme par exemple lors de l'utilisation d'inéquations [Sil99], ou des minima ou maxima de fonctions correspondant aux contraintes.

Les domaines discrets sont généralement énumérés, bien qu'il soit tout à fait possible de les découper à la manière des domaines continus. Lors de l'énumération, la première valeur du domaine est en général choisie, mais on peut aussi choisir la plus grande, la plus petite ou

celle se trouvant au milieu.

CHAPITRE III

ÉLABORATION ET EXPLOITATION D'UN MODÈLE PENDANT LA PHASE DE CONCEPTION ARCHITECTURALE

III.1 - Introduction

Le but de ce chapitre est d'étudier la phase d'élaboration d'un modèle en conception architecturale, ainsi que les objectifs et le contexte d'utilisation d'un tel modèle. Un modèle défini pendant cette phase du processus de conception est issu d'une étude de faisabilité du produit, qui vise à prendre en compte le maximum d'informations pertinentes à ce stade encore peu avancé dans le cycle de vie d'un produit. Cette étude tient compte non seulement de la connaissance définie pendant la phase de conception architecturale, mais aussi de la connaissance identifiée pendant les phases précédentes et, lorsque c'est possible, des phases suivantes. Ainsi, les informations regroupées dans un modèle de conception architecturale sont très variées, ce qui rend souvent complexes et difficiles l'élaboration et la résolution de ces modèles.

III.1.1 - Objectifs et contexte de ce type de modèle

La phase de conception architecturale est destinée à évaluer la faisabilité des concepts et des architectures identifiés pendant la phase de recherche de concepts. Les modèles établis pendant la phase de conception architecturale sont parfois complètement exprimés sous une forme permettant une résolution informatique, afin de simuler le comportement, la géométrie, etc. du futur produit. Ainsi, ces modèles sont en général exprimés sous une forme mathématique. La connaissance rassemblée sur le produit pendant les phases précédentes est alors prise en compte et doit être transcrite sous une forme mathématique compréhensible par l'outil de résolution/simulation utilisé. Les décisions précédentes concernant les choix technologiques, le champ d'application du produit et d'autres informations issues du cahier des charges fonctionnel sont ainsi exprimées au travers de ce modèle, même si le formalisme utilisé par l'outil de résolution n'est pas toujours bien adapté et ne permet pas d'exprimer toute la connaissance disponible.

Certaines connaissances s'expriment naturellement (comme certains critères par exemple), alors que l'expression d'autres connaissances sont sujettes à des imprécisions ou incertitudes, à des inexactitudes et à des hypothèses simplificatrices, ou sont intraduisibles dans un modèle mathématique [Ott94]. Certaines connaissances sont exprimées au sein de diagrammes ou d'organigrammes. Le concepteur doit alors définir un modèle mathématique, traduisant (souvent partiellement) ces données graphiques, en plus de l'approximation

intrinsèque de l'expression de la connaissance au sein d'un modèle.

Le modèle établi doit être en adéquation avec les décisions à prendre à ce stade précoce de la conception d'un produit. Autrement dit, le concepteur doit définir un modèle suffisamment exact, précis et parcimonieux sans être trop spécialisé pour choisir de manière pertinente les dimensions principales, les composants, les matériaux à utiliser, etc. Le modèle défini doit aussi conserver une bonne expressivité et compréhensibilité afin de permettre son exploitation par le concepteur dans le cadre d'un processus dynamique de conception. En effet, le concepteur est souvent obligé d'adapter un modèle lors du processus de conception (le processus de conception est nécessairement itératif), en fonction de ses besoins décisionnels et des évolutions qui ont lieu tout au long de la phase de conception architecturale.

La connaissance représentée au sein de ces modèles n'est pas simplement issue de la phase de conception architecturale, mais elle vient aussi des phases précédentes (expression du besoin et recherche de concepts) et des phases futures du cycle de vie d'un produit, en se basant sur les retours d'expérience et sur la connaissance globale qu'a le concepteur du produit et de son cycle de vie.

III.2 Modélisation et aide à la décision en conception architecturale

La notion de modèle et son élaboration sont fondamentales en conception architecturale. En effet, les modèles considérés pendant cette phase permettent de prendre des décisions importantes par rapport au processus de développement d'un produit (cf. chapitre 1). Ils doivent être définis pour aider à prendre des décisions qui concernent l'architecture principale d'un produit. Ces modèles n'ont pas besoin d'être trop exhaustifs quant aux caractéristiques du produit. L'objectif n'est pas de faire de la simulation intensive du comportement du produit, d'autant plus qu'assez peu de choix sont généralement déjà faits concernant le produit et ses composants. Ces modèles doivent tenir compte des compromis nécessaires pour prendre de bonnes décisions en accord avec le contexte de la conception préliminaire. La plupart des composants ne sont pas encore déterminés et certaines caractéristiques sont encore incertaines, afin, entre autre, de ne pas restreindre trop tôt le champ d'investigation lié à l'utilisation du produit.

La figure 7 décrit les types de modèles utilisés pour élaborer un modèle de conception architecturale. La phase de recherche de concepts fait principalement appel à des modèles conceptuels. Ils sont généralement basés sur une représentation graphique sous forme de

diagrammes ou de graphes. Il permettent de structurer globalement les concepts les uns par rapport aux autres.

Les modèles mathématiques sont issus de l'analyse menée pendant la phase de conception architecturale, lors de l'analyse des concepts et des phénomènes physiques identifiés précédemment. Ces modèles décrivent généralement la géométrie et le comportement physique du produit, de ses composants et de leurs interactions. Chaque modèle mathématique peut décrire un point de vue particulier sur le produit ou un de ses composants. Nous considérons qu'un point de vue correspond au contexte d'exploitation d'un modèle. Chaque point de vue peut être lié à l'expression d'une connaissance métier : fonctionnel, tolérancement, thermique, électrique, etc. Deux modèles différents peuvent décrire le même concept, mais avec un point de vue différent, traduisant une connaissance métier spécifique.

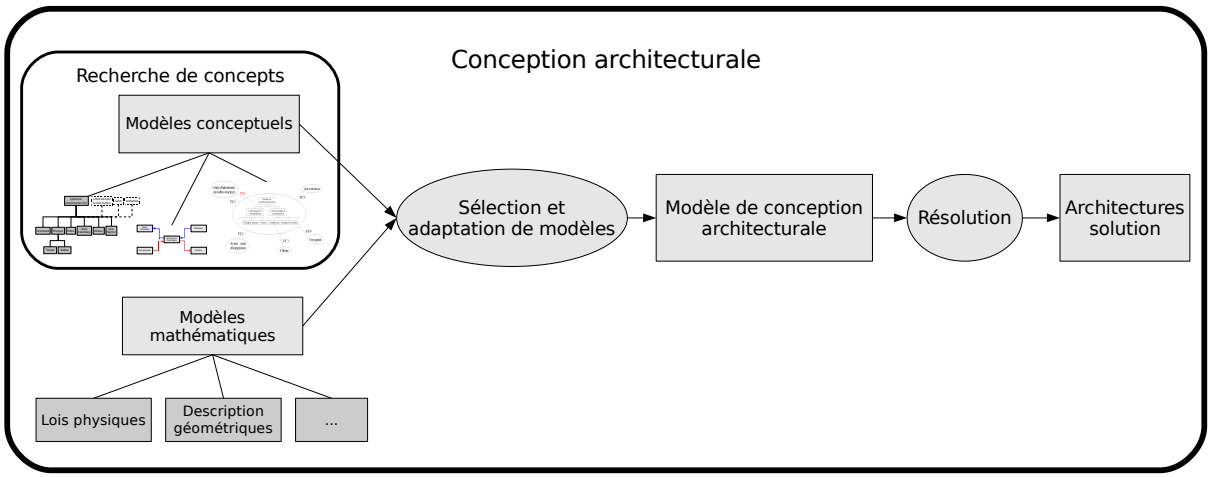


Figure 7 : Cheminement des modèles en conception préliminaire.

Pur élaborer un modèle pertinent, le concepteur essaie de regrouper l'ensemble de la connaissance qu'il a formalisée dans un seul modèle qui permettra d'obtenir des architectures du produit satisfaisant le besoin initial et l'ensemble des contraintes qu'il a ainsi défini. Autrement dit, il fait la synthèse de l'ensemble des points de vue pertinents qu'il a mis en évidence pour définir un unique modèle à résoudre. Cette synthèse résulte du choix des modèles et de leur adaptation qui est effectuée pour correspondre aux attentes décisionnelles du concepteur.

En effet, le concepteur doit évaluer l'apport décisionnel de son modèle ou des sous-modèles qui composent le modèle du produit. L'état décisionnel d'un modèle peut s'estimer en considérant 4 paramètres principaux [Ver04] :

- La **parcimonie** : elle est une mesure inverse de la complexité d'un modèle. Elle croît avec le nombre de variables et leur niveau de couplage.
- L'**exactitude** : elle définit une estimation de l'écart entre le modèle considéré et un modèle de référence ou la réalité, perçue par exemple au travers d'expérimentations.
- La **précision** : elle exprime les imprécisions intrinsèques à certaines variables ou connaissances exprimées dans le modèle. Elle regroupe la prise en compte des imprécisions d'ordre épistémique et des incertitudes liées à la variabilité du produit et de son contexte.
- La **spécialisation** : elle définit le champ d'application du modèle considéré. Elle se base aussi sur les hypothèses simplificatrices utilisées pour exprimer le modèle.

Ces 4 paramètres suffisent pour évaluer dans quelle mesure un modèle correspond à nos attentes décisionnelles, c'est-à-dire dans quelle mesure il aide à prendre des décisions dans un contexte donné. Lorsque l'adéquation n'est pas suffisante, le concepteur doit adapter son modèle.

III.2.1 - Adaptation de modèles

Les méthodes d'adaptation de modèles s'appuient sur des techniques mathématiques ou sur des connaissances liées à leur validation expérimentale et à leur contexte d'exploitation. Elles permettent d'exprimer la même connaissance d'une autre manière que celle déjà utilisée. En effet, la connaissance liée à un composant ou à un phénomène physique peut se représenter de plusieurs façons. Les différentes expressions d'une même connaissance n'ont pas le même état décisionnel (parcimonie, exactitude, précision, spécialisation), ce qui permet d'utiliser la représentation satisfaisant au mieux le besoin du concepteur. Par exemple, des réseaux de neurones peuvent être utilisés pour approcher des phénomènes physiques complexes, comme lors du séchage de matériaux [Hug99] ou encore lors de l'étude de la résistance mécanique de plaques raidies [Fis02]. L'utilisation des méthodes d'adaptation visent à améliorer un ou plusieurs paramètres de l'état décisionnel d'un modèle pour le rendre plus facilement ou mieux exploitable. Néanmoins, l'amélioration d'un paramètre de l'état décisionnel d'un modèle se fait à un moment donné nécessairement au détriment des autres.

Par exemple, les méthodes de réduction de modèles améliorent principalement la parcimonie des modèles, mais en contrepartie elles augmentent leur spécialisation. Ces méthodes permettent de définir peu d'inconnues principales et de déduire toutes les autres à

partir de celles-ci. C'est cette déduction des autres paramètres qui spécialise les modèles en faisant des hypothèses sur les relations existantes entre certains paramètres. Ces hypothèses sont basées sur une connaissance à priori concernant l'objet modélisé [Ryc04]. Elles sont utilisées pour gérer la complexité des modèles de simulation numérique (éléments finis, volumes finis) utilisés en Conception Assistée par Ordinateur. La complexité des modèles est souvent évaluée, dans ce domaine, à la seule taille de ces modèles et plus particulièrement à la taille des matrices ou aux nombres de nœuds des maillages (degrés de liberté). Les méthodes de réduction de modèles augmentent donc la parcimonie des modèles éléments finis en réduisant le nombre des degrés de liberté du modèle [Ryc04, Mad98].

Les méthodes d'approximation de modèles sont parfois liées à l'utilisation des réseaux de neurones artificiels [Dre98] comme des fonctions d'approximation définies par des techniques mathématiques de régression non linéaires. Elles permettent aussi d'accroître la parcimonie d'un modèle, tout en augmentant sa spécialisation. C'est, par exemple, ce qui peut être fait lorsqu'on veut approximer une liste de valeurs expérimentales d'un phénomène physique, sans qu'aucune loi de comportement ne soit définie. Ces valeurs expérimentales nécessitent de restreindre le champ d'application du modèle aux conditions d'expérimentation, mais la précision et l'exactitude de l'approximation dépendent en principe de la validité des données expérimentales. Trois types de modèles d'approximation sont généralement distingués [Hug99, Dre04] : les modèles « boîtes noires », les modèles de connaissance et les modèles « boîtes grises » ou semi physiques. Les modèles de connaissance désignent les modèles mathématiques établis par l'intermédiaire d'une analyse structurelle des phénomènes étudiés et font parfois intervenir des paramètres ajustables mais ayant un sens physique. Un modèle « boîte noire » est un modèle établi dans le but de déterminer une relation entre les variables caractéristiques d'un phénomène. Cette relation est établie uniquement sur la connaissance de différentes valeurs mesurées ou simulées que l'on cherche à approcher en ajustant les paramètres des fonctions choisies. Les réseaux de neurones ont connu un essor important dès lors qu'il a été démontré formellement qu'une des caractéristiques fondamentales des réseaux de neurones était leur propriété d'approximateurs universels parcimonieux [Hor89, Hor94]. La modélisation semi-physique est une représentation intermédiaire entre les types de modélisation précédents. Il peut s'agir, par exemple, d'un modèle phénoménologique simplifié décrivant un phénomène par quelques paramètres ou

caractéristiques principales restant à fixer de manière empirique. Le comportement global est alors approché tout en conservant son caractère phénoménologique.

D'autres méthodes accroissent la parcimonie des modèles tout en diminuant assez peu leur précision et leur exactitude, mais en augmentant leur spécialisation. Les méthodes mixtes s'appuient sur le constat que les modèles physiques intègrent deux types de relations : (i) des lois de conservation et (ii) des lois phénoménologiques. Les lois phénoménologiques sont définies à l'aide de paramètres qui sont imprécis par nature. Ces paramètres imprécis sont tributaires de tous les phénomènes physiques non pris en compte dans l'élaboration du modèle. La prise en compte de ces phénomènes se ferait de toute façon au détriment de la parcimonie du modèle puisqu'elle entraînerait, à terme, une complexification de son expression. La stratégie d'adaptation des méthodes mixtes est donc de gérer conjointement, précision, exactitude et spécialisation du modèle. Dans le domaine de la conception des échangeurs de chaleur ou de matière par exemple, la méthode des Unités de Transfert s'appuie sur une *fuzzification* des paramètres d'intensité des transferts thermiques qui sont définis à l'aide d'intervalles. La largeur de cet intervalle peut être reliée à une mesure du niveau de confiance accordé à la valeur du paramètre. L'adaptation du modèle est obtenue par une intégration analytique à l'aide de réseaux dont chaque élément décrit le fonctionnement d'une unité qui transfère de la chaleur et/ou de la matière. L'intégration analytique n'est possible que grâce à la *fuzzification* du modèle. Elle a aussi pour conséquence de restreindre l'étendue du domaine d'application du modèle qui n'est plus valide que pour certaines topologies de réseaux [Seb03].

III.3 - Connaissance formalisée pendant la phase de recherche de concepts

Cette phase se situe avant la phase de conception architecturale, comme nous avons pu le voir précédemment (cf Chap. 1 et [Pal96]). Elle permet de faire les premiers choix conceptuels et technologiques liés au produit. C'est aussi durant cette phase que certaines parties essentielles du modèle mathématique, établi à la fin de la phase de conception architecturale, sont identifiées. Ainsi, les premières variables et leurs valeurs associées peuvent être mises en évidence à ce stade, en se basant sur les outils de l'Analyse Fonctionnelle (AF).

En effet, l'AF permet au concepteur de décomposer et de hiérarchiser les fonctions

d'un produit relatives à un besoin. Les fonctions de niveaux inférieurs permettent de réaliser la fonction de niveau supérieur à laquelle elles sont reliées [Szy99, Hir01]. Cette décomposition est souvent représentée au sein d'un diagramme FAST (Function Analysis System Technique) ou d'un organigramme technique (OT), ce dernier peut être étendu aux milieux extérieurs. Les blocs fonctionnels de ces organigrammes traduisent une fonction qui s'exprime souvent à l'aide d'un verbe d'action (par exemple : refroidir, transporter, transférer, etc.) auquel est joint un flux (de matière, d'énergie ou de signal) et des compléments précisant la portée de la fonction. L'OT permet de synthétiser les décompositions fonctionnelles effectuées, par exemple dans le cadre de diagrammes FAST pour les fonctions de service d'un produit. L'OT peut être utilisé pour représenter la décomposition fonctionnelle ou organique d'un produit.

La figure 8 représente un exemple de décomposition fonctionnelle au sein d'un diagramme FAST pour un aspirateur ménager. La fonction de service que doit remplir l'aspirateur est de récupérer la poussière du sol. Cette dernière se décompose en 3 sous-fonctions : aspirer la poussière, séparer la poussière de l'air et stocker la poussière. Sur la droite du diagramme sont représentés les composants associés à ces fonctions. Ainsi le sac jetable permet de remplir plusieurs fonctions : laisser passer l'air, retenir la poussière et stocker la poussière.

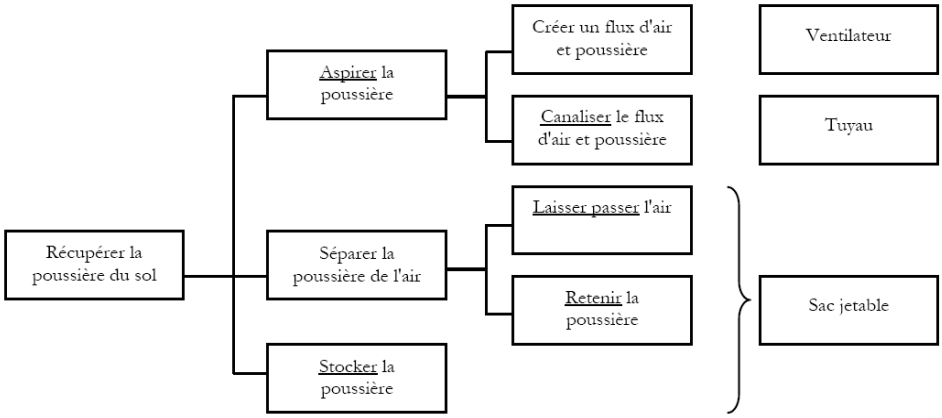


Figure 8 : Exemple de diagramme FAST d'un aspirateur ménager.

Ces mêmes composants apparaissent au sein de l'OT de l'aspirateur dans la figure 9. Sur cet organigramme, les composants aidant à la satisfaction des fonctions de service ou des fonctions contraintes sont regroupées en unités fonctionnelles. A chaque bloc fonctionnel de cet organigramme peut être assimilé des variables caractéristiques. Ces variables sont

directement liées aux blocs fonctionnels et décrivent ainsi les principales caractéristiques des flux, des futurs composants ou des unités fonctionnelles du produit. Ces caractéristiques sont directement en relation avec la satisfaction des fonctions associées au besoin ayant déclenché la conception de ce produit. Ce type d'organigramme permet de hiérarchiser une première fois les fonctions et les variables identifiées au sein du produit.

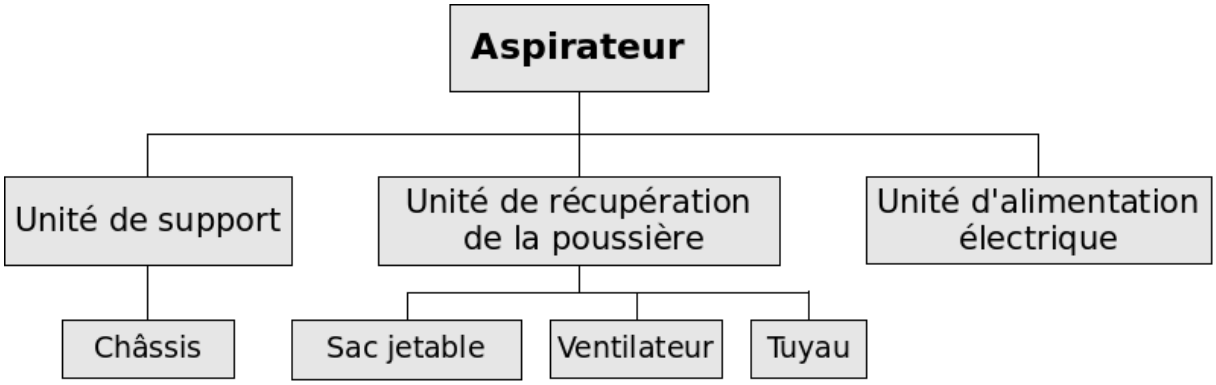


Figure 9: Exemple d'organigramme technique pour un aspirateur ménager.

De même, l'AF permet d'obtenir des informations sur les flux et leur circulation entre les blocs fonctionnels en utilisant un bloc diagramme fonctionnel. Ce dernier permet de mettre en évidence certaines topologies ou contraintes géométriques liées aux flux et à leur circulation. La figure 10 représente un bloc diagramme fonctionnel de l'aspirateur lors de son fonctionnement. Il décrit une topologie correspondant à la fonction de service précédemment définie dans la figure 8. Ainsi, nous pouvons voir que le flux de poussière passe par le tuyau et est arrêté et stocké par le sac, contrairement à l'air qui traverse ensuite le ventilateur.

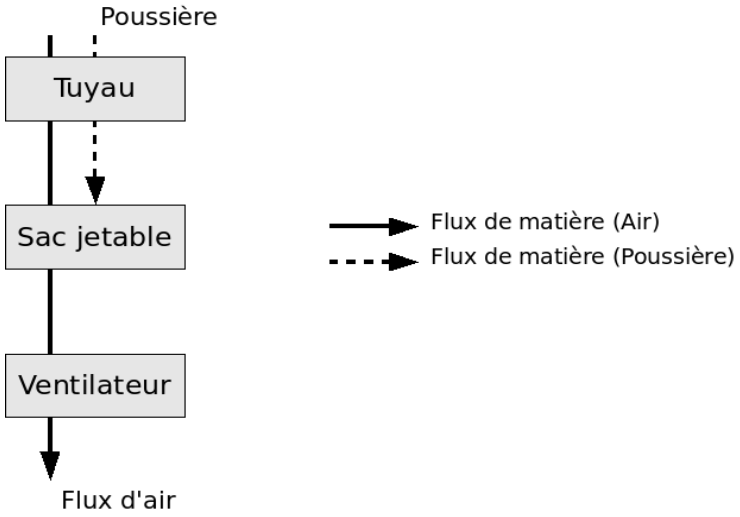


Figure 10 : Exemple d'extrait de bloc diagramme fonctionnel pour un aspirateur ménager.

Ces diagrammes ou ces organigrammes permettent ainsi au concepteur d'obtenir l'ensemble des principales caractéristiques fonctionnelles d'un produit conformément à une hiérarchie et une topologie fonctionnelle basées sur un cahier des charges.

Ces diagrammes sont établis pour chaque situation de vie importante rencontrée par le produit. Les situations de vie les plus importantes sont identifiées par le concepteur au début de la phase de recherche de concepts par retour d'expérience. Elles sont choisies en fonction de leur pertinence par rapport au besoin fonctionnel sur tout le cycle de vie du produit. Il faut alors noter que ces situations de vie peuvent mener à des décompositions fonctionnelles très différentes, en particulier lorsqu'il ne s'agit pas de situations de vie liées au fonctionnement du produit (par exemple : assemblage, recyclage).

A la fin de la phase de recherche de concepts, le concepteur, ayant établi une décomposition fonctionnelle satisfaisante (au moins) pour chaque situation de vie importante, doit tester la faisabilité du ou des concepts correspondants pendant la phase de conception architecturale. Il obtient ainsi une hiérarchie et une topologie de fonctions, mais aussi de variables définissant les caractéristiques clés du produit. Ces variables seront alors utilisées dans les modèles définis en conception architecturale et correspondent la plupart du temps à des variables de conception (voir section suivante) ou à des variables dont les valeurs identifiées traduisent des critères. Quelques autres variables peuvent être définies, simplement parce qu'elles sont utilisées communément pour qualifier ou évaluer d'autres caractéristiques pertinentes pour le concepteur (par exemple des caractéristiques liées aux phénomènes et technologies en relation avec des flux identifiés).

III.3.1 - Connaissance formalisée à l'aide du cahier des charges fonctionnel

Le cahier des charges fonctionnel permet d'exprimer le besoin lié au produit à concevoir. Ce besoin couvre aussi bien les situations de vie liées à l'utilisation que d'autres situations de vie liées à la fabrication, à l'assemblage ou encore au recyclage du produit. Ainsi, de nombreuses restrictions peuvent être définies par rapport au champ d'application d'un produit. Elles définissent principalement certaines caractéristiques à prendre en compte et dont les valeurs servent d'indicateurs concernant la faisabilité du produit par rapport au cahier des charges fonctionnel.

Pour définir cette connaissance liée aux phases suivantes du cycle de vie d'un produit,

le concepteur fait aussi appel à sa propre expérience, à la connaissance des procédés de fabrication de l'entreprise, à des retours d'expérience sur des produits similaires, etc. En particulier, certaines contraintes liées à la fabrication, à l'assemblage et au recyclage peuvent alors être prises en compte si elles s'avèrent pertinentes. Le non respect de certaines de ces informations pendant la phase de conception peut mener à un produit difficile à fabriquer ou assembler ou ne respectant pas une certification ou une norme nécessaire à sa commercialisation. Dans ce cas, tous les efforts effectués jusque là sont vains et la phase de conception architecturale doit alors être réinitialisée afin de satisfaire ces critères supplémentaires qui n'avaient pas été pris en compte. Les allers-retours pendant la phase de conception d'un produit entraînent alors des coûts supplémentaires importants et doivent être minimisés; c'est pourquoi ces informations sont importantes à prendre en compte, lorsqu'elles sont accessibles. Par exemple, certaines pièces de grande taille peuvent avoir des dimensions limitées en considérant les phases de stockage et de transport de la pièce.

Le cahier des charges fonctionnel permet alors de définir le domaine de validité des principales variables d'un modèle définissant un produit. De plus, certaines précisions ou tolérances sur les dimensions, ou sur les performances peuvent aussi être définies. Ces précisions ou tolérances peuvent en effet être issues de la connaissance des machines et des procédés de fabrication utilisés par l'entreprise, définissant la réalité physique du produit et de sa conception.

III.4 - Connaissance formalisée pendant la phase de conception architecturale

L'objectif de la conception architecturale est d'étudier la faisabilité des architectures issues des concepts définis par la phase de recherche de concepts, tout en tenant compte des différents critères du cahier des charges fonctionnel, mais aussi de la réalité physique du produit. Cette phase de conception préliminaire permet de définir l'architecture globale et les caractéristiques principales du produit à concevoir. L'ensemble des caractéristiques du produit est ensuite défini pendant la phase de conception détaillée.

Le concepteur doit ainsi remplacer les blocs fonctionnels, définis dans les diagrammes et organigrammes précédents, par des composants existants ou à dimensionner. Ces composants sont alors principalement définis par les variables (et les valeurs) identifiées pendant la phase de recherche de concepts, puisque ces variables sont directement liées au

besoin fonctionnel. Ces composants ne sont pas tous encore clairement identifiés et peu de connaissances supplémentaires sont disponibles.

Pour les composants existants, les données constructeurs sont souvent accessibles (catalogue ou base de données). Lorsqu'il n'y a pas de choix de composants à faire, les données constructeurs peuvent alors être définies comme des constantes. Dans le cas des composants à dimensionner, ils sont détaillés afin d'étudier leur faisabilité, au même titre que le produit global. Le niveau de détails de ces composants doit alors correspondre avec la décomposition fonctionnelle et la hiérarchisation des fonctions définies pendant la phase de recherche de concepts. Ces composants à dimensionner peuvent ensuite être remplacés par des composants existants si leurs caractéristiques s'avèrent être compatibles. Le composant existant ainsi dimensionné ne l'a été que pour analyser plus profondément ses caractéristiques, l'AF ayant révélé des composants internes au rôle important pour la satisfaction du besoin fonctionnel lié au produit.

La figure 11 illustre ainsi la décomposition organique d'un système de conditionnement d'air dans un avion. L'analyse fonctionnelle effectuée précédemment définit que les principales contraintes de faisabilité sont liées aux échangeurs de chaleur qu'il convient de dimensionner. Les autres composants sont standardisés et nous pouvons utiliser les caractéristiques fournies par leur fabriquant (dimensions, rendement, prix, etc.). Toutes ne seront pas forcément utilisées, cela dépend du contexte d'exploitation de ces données (physique, géométrique, économique, etc.).

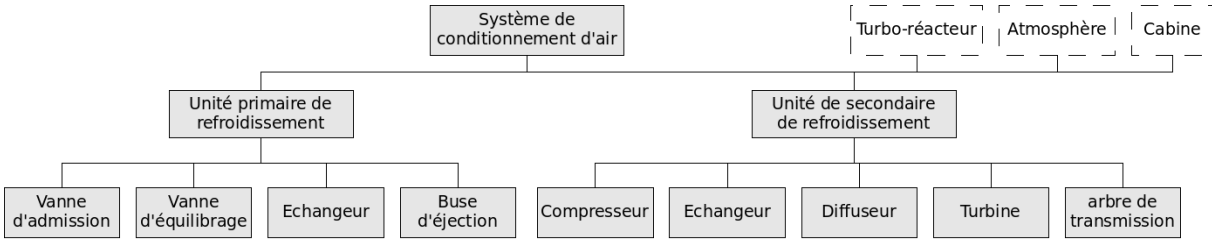
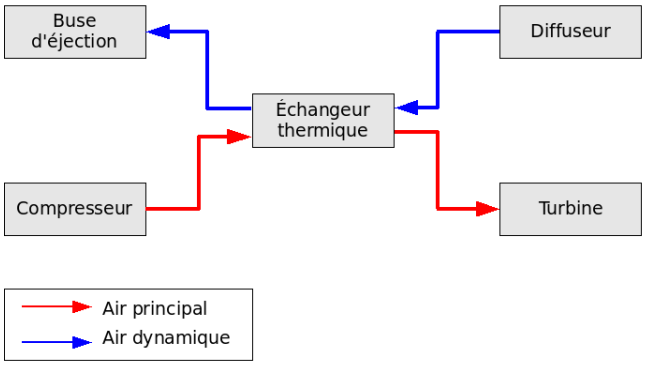


Figure 11 : Organigramme technique définissant la décomposition organique d'un système de conditionnement d'air dans un avion.

Ensuite, le concepteur identifie les phénomènes physiques qui interviennent au sein du produit, entre les blocs fonctionnels ou composants. Il s'appuie alors sur le bloc diagramme fonctionnel (voir figure 12) et hiérarchise les phénomènes en se rapportant à la décomposition fonctionnelle du produit. En se basant sur la connaissance définie par l'AF, il peut définir les plus pertinents et ceux qui peuvent être négligés. En effet, cette hiérarchisation des

phénomènes physiques se base sur l'expérience du concepteur et la relation du phénomène physique avec les flux ayant ou non un rapport direct avec le besoin fonctionnel. La prise en compte de ces phénomènes permet, à terme, de relier, dans le modèle mathématique, les variables de conception aux critères en prenant en compte la topologie et le comportement physique du produit.



Effets physiques fonctionnels	Variables identifiées
- Transfert de chaleur par convection - Transfert de chaleur par conduction	- Température d'entrée/sortie
Effets physiques induits	
- Perte de pression - Encrassement - Déformation des matériaux - Ruine thermique des matériaux - Ruine mécanique des matériaux	- Pression d'entrée/sortie - Épaisseur des particules - Déformations - Température interne maximale - Contrainte mécanique maximale

Figure 12 : Bloc diagramme fonctionnel et phénomènes physiques identifiés pour un échangeur de chaleur dans un système de conditionnement d'air d'avion.

La figure 12 présente ainsi les effets physiques identifiés pour un échangeur de chaleur dans un avion, ainsi que les variables caractéristiques de ces phénomènes. Une première catégorisation des phénomènes et des variables est faite afin de dissocier les effets physiques fonctionnels des effets physiques induits. Les effets induits sont classés/hiérarchisés et certains seront négligés.

Une fois les phénomènes physiques identifiés et hiérarchisés, le concepteur doit définir ceux qui peuvent être négligés et ceux qui doivent être pris en compte pour étudier de manière pertinente et réaliste la faisabilité du produit. Il se base alors principalement sur des retours d'expériences ou sur sa connaissance des phénomènes physiques et des technologies employées.

Ensuite il doit définir les modèles mathématiques qui vont permettre de formaliser la connaissance relative aux composants et à leurs interactions. Un même phénomène physique peut se modéliser de plusieurs manières. Ces différentes modélisations ne sont pas équivalentes du point de vue de la prise de décision. Elles ne représentent pas le phénomène avec la même précision et la même exactitude. Certains types de modélisation obligent à figer certaines caractéristiques, comme des topologies ou structures, ce qui peut alors spécialiser

prématurément la représentation du produit. De même, toutes les modélisations n'ont pas la même complexité de résolution, mais aussi la même intelligibilité.

Par exemple, une modélisation par éléments finis est bien adaptée à de la simulation, mais pas à la conception préliminaire, et ce pour plusieurs raisons, dont :

- en général, le composant à dimensionner n'a pas nécessairement une structure interne prédéfinie, ce qui empêche ce type de modélisation.
- les temps de calculs et le nombre de variables introduites sont contraires au souci de parcimonie du modèle en cours d'élaboration.

D'autres modèles sont des modèles « boîtes noires ». Ils sont principalement utilisés lorsqu'il s'agit de composants existants et que l'AF n'a pas mis en évidence de variables pertinentes dans des niveaux inférieurs de l'OT dans la décomposition fonctionnelle de ces composants. Ce type de modèle définit généralement un composant ou un phénomène physique à l'aide d'une ou plusieurs relations reliant directement les variables d'entrée et de sortie. Ce type de modèle permet de ne pas exprimer le fonctionnement interne d'un composant, en se basant, par exemple, sur des données constructeurs liées au rendement ou à l'efficacité du composant. Dans ce cas, ces données sont souvent fournies avec une précision ou un niveau de confiance établis par le constructeur lui même. Il peut aussi être défini par retour d'expérience lorsqu'il s'agit de phénomènes physiques.

Une fois toutes ces caractéristiques formalisées, le concepteur obtient généralement un modèle global du produit qui lui permet d'établir dans un contexte décisionnel particulier la faisabilité d'un produit.

III.5 - Exploitation du modèle et des solutions

Le concepteur dispose donc d'un modèle mathématique qu'il doit résoudre, généralement à l'aide d'un outil mathématique qui prend parfois la forme d'un solveur. Il n'existe pas d'outil spécialement dédié à la conduite et au développement des phases de conception architecturale et plusieurs alternatives s'offrent à lui :

- un tableur : il offre une facilité d'utilisation, car en général ce type d'outil est souvent utilisé dans l'environnement du concepteur. Les données sont faciles à manipuler et à trier. Les solutions sont représentables à l'aide de graphiques et peuvent facilement être exportées vers d'autres formats.

- un outil de calcul numérique : il offre généralement de très bonnes performances de calcul, c'est pourquoi ce type d'outil est principalement utilisé en simulation numérique. Cependant, il nécessite une phase de programmation spécifique au problème et à l'outil, ainsi que de bonnes connaissances mathématiques.
- un solveur de problèmes de satisfaction de contraintes : il se situe généralement entre les deux outils précédents. Il permet au concepteur d'exprimer facilement la connaissance et ne devrait pas nécessiter de programmation particulière.

Le choix de l'outil utilisé pour traiter un modèle influence aussi l'expression de la connaissance présente dans un modèle, car chaque outil dispose, en général, de son propre langage. Ce langage ne permettant pas d'exprimer toute la connaissance du concepteur, ce paramètre doit être pris en compte lors du choix de l'outil. De plus, chaque outil n'utilise pas les mêmes méthodes de résolution. Par exemple, le tableur ou le solveur de CSP ne laisse pas ou peu le choix de la méthode de résolution au concepteur, alors que la méthode de résolution de l'outil de simulation numérique est programmée. Un tableur sera généralement plus limité : il ne permet d'effectuer qu'une résolution séquentielle d'un modèle. Le solveur CSP dispose quant à lui d'un mécanisme de propagation de contraintes calculant des domaines ou valeurs consistantes pour les variables du problème et un algorithme de recherche complet permettant d'explorer exhaustivement l'espace de recherche sur un domaine d'approximation. Certaines des méthodes de résolution couramment utilisées dans les outils de calcul numérique font appel à une exploration aléatoire de l'espace de recherche basée sur des algorithmes évolutionnaires. Ces méthodes de résolution sont très efficaces, mais ne permettent pas d'obtenir l'ensemble complet des solutions, contrairement à l'approche du solveur CSP [Seb07].

Une fois l'outil choisi et le problème traité, le concepteur dispose d'un ensemble de solutions. Il doit choisir parmi elles l'architecture qui sera détaillée dans les phases suivantes du processus de conception. L'exploitation de l'ensemble des solutions obtenues est alors primordiale, et l'objectif est d'identifier non seulement une solution optimale par rapport à certains critères, mais aussi une solution qui soit robuste du point de vue de la conception. En effet, le modèle utilisé par la résolution est basé sur un certain nombre d'imprécisions, d'inexactitudes et d'hypothèses. Ce choix est alors tout aussi important que le modèle en lui-même et doit tenir compte de l'apport décisionnel de ce dernier, car un modèle est faux par nature : il n'est qu'une représentation partielle de la réalité.

Plusieurs méthodes peuvent être mises en oeuvre pour trier les solutions obtenues. Ces méthodes sont indépendantes de l'outil utilisé pour la résolution et permettent de faire ressortir la ou les solutions les plus pertinentes.

Lorsque les critères à prendre en compte peuvent être pondérés les uns par rapport aux autres, ou plus généralement qu'une relation pertinente peut être établie pour définir une valeur représentative de l'intérêt d'une solution par rapport aux besoins du concepteur, alors une fonction objectif à minimiser (ou à maximiser) peut être définie. Cette méthode est séduisante, puisqu'elle ne fournit qu'une seule solution, mais la validité de la fonction objectif est souvent discutable. Lorsqu'un unique critère est à prendre en compte, alors cette méthode est intéressante, puisque la fonction objectif vise à minimiser (ou maximiser) la valeur du seul critère.

Lorsque plusieurs critères sont à considérer et qu'il est difficile de les hiérarchiser et de les pondérer, la méthode du front de Paréto peut être utilisée. Cette dernière est plus facile à mettre en oeuvre et permet en général d'identifier plusieurs solutions satisfaisantes. Il est alors plus facile d'identifier des groupes de solutions se trouvant dans la même zone du front de Paréto. Ces solutions proches les unes des autres permettent d'identifier des zones que l'on peut supposer plus robustes aux changements qu'une unique solution optimale mise en évidence par une fonction objectif ou se trouvant isolée dans le front de Paréto. Cependant, il faut s'assurer que la proximité des solutions dans le front de Paréto correspond aussi à la proximité des solutions par rapport aux valeurs des variables de conception les définissant.

La notion de robustesse est essentielle pour éviter les risques de mauvaise conception du produit [Rot90]. En effet, une solution, non robuste aux choix effectués dans les phases suivantes de conception, risque de remettre en cause le bon fonctionnement du produit et la satisfaction du besoin initial. Afin de contrôler ces risques, le concepteur doit estimer l'état décisionnel du modèle utilisé pour évaluer les erreurs (précision et exactitude) et les hypothèses simplificatrices liées à l'expression de la connaissance.

III.6 - Pourquoi utiliser le paradigme des CSP?

Un problème de satisfaction de contrainte (CSP) est un problème modélisé sous la forme d'un ensemble de contraintes appliquées à des variables évoluant dans un domaine de variation (voir Chap1 pour plus de détails). Une résolution basée sur les techniques de satisfaction de contraintes permet de calculer l'ensemble complet des solutions d'un problème

sur un domaine d'approximation. La plupart des autres méthodes ne fournissent que des solutions optimales ou qu'un nombre limité de solutions. De nombreuses applications sont développées dans le domaine de l'ingénierie de conception à l'aide d'approches ne permettant qu'une exploration stochastique de l'espace des solutions [Seb07]. Le concepteur n'a donc pas la garantie d'obtenir toutes les solutions potentiellement exploitables de son problème. De plus, les techniques de résolution utilisées pour résoudre les problèmes de satisfaction de contraintes sur des domaines continus fournissent des garanties sur la non existence de solutions, contrairement aux autres approches, dont les méthodes évolutionnaires. Cette certitude sur l'absence de solutions permet au concepteur de mieux percevoir la forme de l'ensemble de solutions du problème. Disposant de l'ensemble complet de solutions, le concepteur peut aussi mieux comprendre les réponses aux variations de son modèle et mieux choisir une solution robuste, en essayant de prévoir les variations pouvant intervenir dans les phases de conception suivantes (tolérances sur certaines dimensions, modification de composants, etc.).

Le paradigme des CSP vise aussi à dissocier l'écriture du modèle et la résolution. Ainsi, le concepteur décrit son produit suivant le formalisme des CSP, le processus de résolution étant générique et applicable à tous les problèmes bien formalisés. Il peut alors se concentrer sur la formalisation de la connaissance et sa compréhensibilité. Les modifications au sein du modèle sont alors plus faciles à effectuer, comparées à d'autres outils comme les tableurs ou les outils de calcul numérique, où l'expression de la connaissance est diluée au milieu des algorithmes de résolution. D'autant plus que la conception est un processus dynamique, où les modifications apportées au modèle peuvent être fréquentes et plus particulièrement lors de la conception préliminaire, où peu de décisions ont encore été prises au sujet du produit. Certaines décisions peuvent avoir lieu pendant cette phase et nécessiter d'importants changements au sein du modèle (choix ou changement de composants, de technologie, de topologie, etc.).

De même, le fait de séparer la modélisation de la résolution, permet de mieux réutiliser la connaissance formalisée. Les contraintes et les variables, peuvent être facilement rapprochées de caractéristiques de composants ou d'interactions entre composants. Ainsi, ces modèles peuvent être intégrés dans un formalisme plus général et de plus haute abstraction, permettant au concepteur de définir la connaissance plus exhaustivement et facilitant sa manipulation. Cette capitalisation de la connaissance permet d'aider à la conception de futurs

produits, ce qui tend à réduire les temps de conception. Ces délais sont souvent critiques dans un contexte industriel et certaines phases préliminaires sont parfois négligées par manque de temps.

III.7 - Points importants pour la réalisation d'un outil d'aide à la décision en conception architecturale

Cette partie a pour objectif de faire une liste aussi exhaustive que possible de notions importantes pour le concepteur lorsqu'il définit et traite un modèle en conception architecturale. Ainsi ces notions, lorsqu'elles ne sont pas maîtrisées, peuvent nuire à la bonne utilisation d'un outil dédié à cette phase du processus de conception.

II.7.1 - Types des variables en conception

En conception, plusieurs types de variables sont mis en évidence tout au long du processus de conception. Nous pouvons dissocier principalement trois types de variables :

- **les variables de conception :** elles définissent, par leurs valeurs, le produit et représentent ses principales caractéristiques. Elles correspondent, la plupart du temps, à des choix de composants, de dimensions principales, de choix de matériaux, de formes ou de topologies.
- **les variables de décision :** elles définissent par leurs valeurs les performances du produit au regard du cahier des charges fonctionnel. Elles sont souvent appelées variables de performance.
- **les variables auxiliaires :** elles représentent toutes les autres variables, qui sont, en général, spécifiques au modèle et à la phase de conception étudiée (voir figure 7). Elles ne sont introduites dans les modèles que pour relier mathématiquement les variables de conception et les variables de décision entre elles. Ce lien est réalisé au travers de l'expression des contraintes. Elles sont généralement nécessaires mais contraires au principe de parcimonie des modèles. Elles sont nécessaires pour les raisons suivantes :
  - Le modèle doit conserver un caractère intelligible (expressif et compréhensible). Il est susceptible d'évoluer au cours du processus de conception (processus itératif) et sera

probablement réutilisé dans d'autres projets de conception. Le concepteur doit pouvoir le faire évoluer rapidement. Il doit donc comprendre sa structure et sa nature dans sa globalité et dans chacun de ses éléments (variables, contraintes, ensemble de contraintes).

- L'expression des connaissances à l'aide de modèles mathématiques traduit des couplages qui ne peuvent pas toujours s'exprimer sans introduire de variables auxiliaires de calculs.

On peut trouver une illustration de l'utilisation des variables auxiliaires dans [Din90, Smi06]. Dans ces travaux, le modèle défini pour représenter une chaîne montage de voitures a nécessité l'introduction de variables auxiliaires booléennes pour modéliser certains choix de séquencement de la chaîne de montage.

La figure 13 illustre la répartition des variables et des contraintes dans un modèle de conception architecturale. Les variables de conception et les variables de décision sont communes à toutes les phases du processus de conception du produit. Les critères issus du cahier des charges fonctionnel sont généralement exprimés au travers de variables, bien que certains se traduisent directement sous forme de contraintes. L'ensemble des variables auxiliaires, ainsi que la plupart des contraintes sont définis spécifiquement dans le contexte de la conception architecturale. Ils traduisent les caractéristiques géométriques, le comportement physique du produit ou de ses composants, etc. en fonction des besoins décisionnels du concepteur. Ils sont nécessaires pour prendre en compte la réalité physique du produit et sa faisabilité, mais nuisent à la parcimonie du modèle.

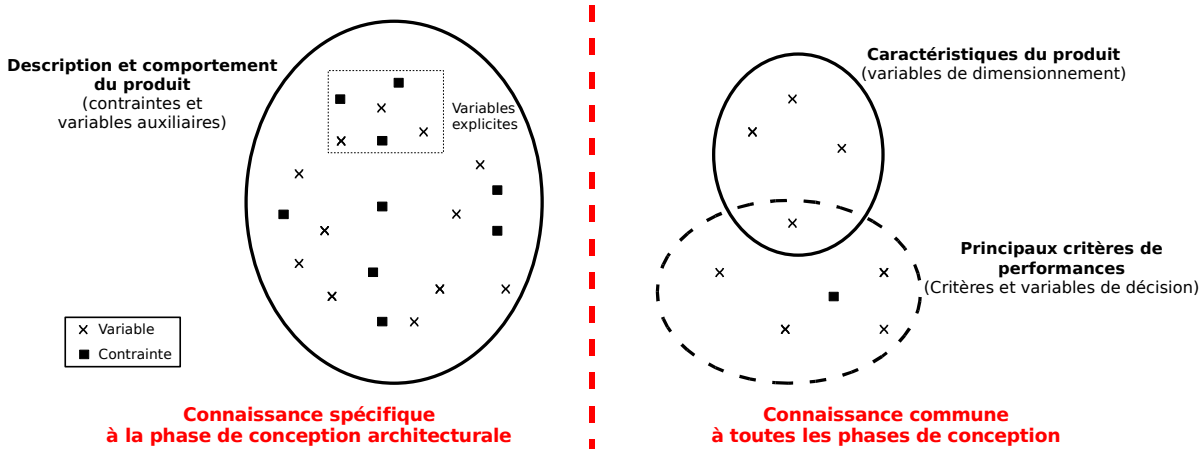


Figure 13 : Classification des variables dans un modèle de conception architecturale.

Le concepteur utilise naturellement la distinction de ces types de variables au sein d'un modèle, ce qui lui permet d''avoir une bonne vision globale du produit. Il peut alors identifier rapidement les variables essentielles, dont celles qui lui serviront pour définir et évaluer un produit. Il peut aussi plus facilement hiérarchiser et trier les variables d'un modèle en fonction de ses besoins. Le concepteur peut ainsi mieux organiser la connaissance et mieux la maîtriser.

III.7.2 - Importance des variables de conception architecturale

Les variables de conception définies dans le cadre de la conception architecturale (ou plus simplement : variables de conception architecturale) sont des variables identifiant les principales caractéristiques structurantes d'un produit [Sca04a]. Pendant la phase de conception architecturale, ce sont principalement ces variables qui doivent être déterminées par le concepteur. Elles sont essentielles, car un changement de leur valeur dans les phases de conception suivantes, nécessite de réinitialiser la phase de conception architecturale pour analyser la faisabilité du produit. Ces variables disposent alors de propriétés particulières au sein d'un modèle et par rapport à un produit :

- Elles sont utilisées dans toutes les situations de vie.
- Elles représentent les caractéristiques clés d'un produit [Tho99, Sca05]. Leurs valeurs influent grandement sur les performances, le coût, le rendement, etc. d'un produit.
- Les modèles définis en conception préliminaire cherchent à relier des variables de conception caractérisant un produit aux variables de décision qui permettent d'évaluer ses performances. Autrement dit, les modèles de conception architecturale sont construits pour relier les variables de conception aux performances du produit.

Ces caractéristiques nous permettent alors de définir plus précisément la notion de variable de conception :

Définition (Variable de conception) : *Les variables de conception (VCo) d'un modèle représentent les principales caractéristiques structurantes d'un produit. Leur domaine de valeurs sont reliés à des choix de composants, des dimensions principales, des choix de matériaux, de formes ou de topologies et permettent de distinguer différentes architectures de produit. Elles sont directement déduites de la phase de recherche de concepts et du contexte défini par le cahier des charges fonctionnel.*

Ces variables sont souvent des variables discrètes (en particulier en ce qui concerne des choix de composants, de matériaux, etc.), mais, par exemple, certaines dimensions peuvent être définies sur des domaines continus.

L'importance des variables de conception architecturale dans un modèle résulte de l'expression de la connaissance globale du concepteur par rapport à un produit [Tho99]. Le concepteur a une vision globale du produit, qui doit évoluer d'un concept à un produit industriel. Cette perception de l'importance des variables de conception englobe nettement plus de connaissances que les informations exprimées au sein des modèles mathématiques. En effet, ces modèles sont spécifiques à chaque phase de conception. Cette vision globale du produit dans tout son cycle de vie permet au concepteur de percevoir des valeurs pertinentes pour ces variables, en se basant sur son expérience et sur des retours d'expérience sur ce type de produit ou de technologie. Ces variables sont aussi souvent reliées à des contraintes dimensionnantes majeures, comme par exemple des contraintes d'encombrement, de poids, de coût, etc.

Les variables de conception sont généralement déterminées grâce à la connaissance définie pendant la phase de recherche de concepts. Cette phase vise à définir les principaux concepts et permet de mettre en évidence des variables et des valeurs pertinentes (voir début de chapitre). Ces variables sont d'ailleurs classées et hiérarchisées en fonction de leur importance fonctionnelle. Cette première analyse du produit d'un point de vue fonctionnel permet d'identifier des valeurs, mais aussi des degrés de liberté ou des niveaux de confiance associés aux variables et à leurs valeurs. Ces considérations tiennent compte des retours d'expériences existants, des effets physiques négligés, de l'incertitude estimée par rapport aux phénomènes physiques pris en compte, etc. Au final, ces variables sont identifiées à l'issue d'une analyse de sensibilité du produit sur tout son cycle de vie.

Au sein de certains modèles, le concepteur considère certaines variables comme des variables de conception architecturale, bien qu'intrinsèquement, ces variables n'en soient pas.

Définition (Pseudo variable de conception) : *Les pseudo variables de conception sont des variables dimensionnantes dans le contexte d'un modèle, mais elles ne correspondent pas initialement à des variables de conception :*

- *Elles sont des variables auxiliaires, qui prennent le rôle de variables de conception à partir du moment où le concepteur peut très facilement relier leurs valeurs à celles de variables de conception.*

– *Elles se substituent à des variables de conception non exprimées dans un modèle.*

Dans des situations de vie particulières, certaines variables de conception ne sont pas exprimées à cause d'hypothèses simplificatrices négligeant, par exemple, l'impact d'un composant. C'est le contexte de la conception préliminaire et le principe de parcimonie des modèles, qui pousse le concepteur à ne pas exprimer toutes les variables de conception du produit quand elles ne sont pas nécessaires. Ainsi, les composants non pertinents, dans une situation de vie donnée, sont décrits par quelques caractéristiques clés utilisées dans le reste du modèle. Les variables supplémentaires et inutiles pour ce composant ne sont alors pas exprimées dans ce contexte. Lorsque ces caractéristiques ne correspondent pas à la définition de variable de conception, mais qu'elle permettent tout de même de dimensionner le composant négligé, nous pouvons parler de pseudo variable de conception. C'est par exemple le cas du rapport de capacité calorifique définie entre deux fluides dans des échangeurs thermiques qui se substitue aux dimensions d'une écope (ouverture permettant l'admission d'air dans une machine), comme nous le verrons dans l'exemple du système de conditionnement d'air dans le chapitre V.

Lors de la résolution d'un modèle de conception architecturale, seules les valeurs des variables de conception ou des pseudo variables de conception architecturale importent vraiment pour le concepteur, puisqu'elles définissent l'architecture globale du produit. Les variables auxiliaires et les variables de décision ne permettent que d'évaluer les performances et la faisabilité du produit. Nous pouvons alors décomposer l'ensemble des variables X d'un modèle en trois sous-ensembles :

– V_{Co} : l'ensemble des variables de conception, ayant pour domaines D_{VCo}.

– V_A : l'ensemble des variables auxiliaires, ayant pour domaines D_{VA}.

– V_D : l'ensemble des variables de décision, ayant pour domaines D_{VD}.

Nous avons donc : $X = V_{Co} \cup V_A \cup V_D$. En considérant C l'ensemble des contraintes relatives à un produit, l'ensemble des solutions d'un problème de conception peut alors s'exprimer comme :

$$\{x \in V_{Co} : \exists\, y \in V_A, \exists\, z \in V_D \mid C(x,y,z)\}.$$

III.7.3 - Variable alias

Les variables alias sont des variables auxiliaires exprimant des calculs intermédiaires définissant des concepts pertinents pour le concepteur, améliorant ainsi l'intelligibilité d'un modèle. Ces variables sont explicitement définies en fonction d'autres variables du modèle. Dans un modèle mathématique, cela se manifeste, par exemple, par le fait qu'elles sont seules dans la partie gauche d'une équation. Pour une variable alias x, on pourra définir :

$$x = f(x_1,...,x_k),$$

où $f(x_1,...,x_k)$ correspond à la définition du calcul intermédiaire utile pour la compréhension et l'expressivité du modèle pour le concepteur. Une variable alias peut être définie en fonction d'autres variables alias, mais dans ce cas, il faut vérifier qu'aucun cycle de dépendance n'existe entre ces variables. En effet, par nature elles peuvent être substituées par leur expression de définition, puisqu'elles sont des expressions de calculs intermédiaires. Elles correspondent à des notions géométriques, physiques qui servent au concepteur à mieux comprendre et évaluer le modèle et ses solutions. Dans le cas de la présence d'un cycle de dépendance, il existe un couplage entre les variables impliquées et certaines d'entre elles correspondent à des variables auxiliaires et non des alias.

Définition (Variable alias) : *Les variables alias sont des variables auxiliaires. Leurs valeurs sont directement associées à une expression d'autres variables. Elles représentent l'expression d'un calcul intermédiaire améliorant l'intelligibilité du modèle.*

De plus, le calcul de ces variables ne doit pas impacter négativement le temps de résolution du modèle. En effet, n'étant que des intermédiaires de calcul, le concepteur ne connaît que très partiellement le domaine de validité de ces variables. Ainsi, l'espace de recherche offert au solveur peut être très important, ce qui nuit fortement aux performances de la résolution. Le coût associé à leur calcul doit alors être minimisé autant que possible, surtout si ces variables s'avèrent être peu importantes par rapport au besoin fonctionnel.

Ces variables pourraient être éliminées du modèle en les substituant par leur expression de définition dans les contraintes où elles apparaissent. Cependant, l'intelligibilité du modèle serait amoindrie et le concepteur ne pourrait modifier facilement le modèle ou encore exploiter correctement les solutions obtenues.

Soit V_E l'ensemble des variables alias et D_E leurs domaines. On peut alors définir :

- $V_E \subseteq V_A \subseteq X$,

– $\forall P_1 = <X,D,C>, \exists P_2 = <X\text{-}V_E, D\text{-}D_E, C'>$ tel que : $P_1 \equiv P_2$.

Le calcul de P_2 peut se faire facilement en remplaçant les variables alias par leur expression de définition dans toutes les contraintes de P_1.

La prise en compte des variables alias au sein d'un modèle est nécessaire dans le cadre de la conception architecturale où les modèles peuvent évoluer plusieurs fois en fonction des décisions qui sont prises pendant cette même phase. Cependant, leur exploration doit être aussi réduite que possible si elles s'avèrent peu importantes dans le fonctionnement du produit.

III.7.4 - Précision des variables

La précision des variables et des calculs effectués au sein d'un solveur n'est pas toujours facile à évaluer pour le concepteur. En effet, la notion de précision peut être comprise dans plusieurs sens :

– l'imprécision théorique liée à un modèle et à l'expression de la connaissance.
– la précision numérique des valeurs calculées.

Ces deux types de précisions cohabitent lors de la résolution d'un modèle et elles doivent être cohérentes afin de ne pas dénaturer les solutions obtenues. Le principal problème rencontré vient souvent du fait de la négligence ou le manque de connaissance du concepteur sur la précision numérique, et en même temps du fait que l'outil qu'il manipule ne lui permet pas de formuler correctement la précision théorique de la connaissance qu'il souhaite exprimer. Se pose alors le problème de la validité des solutions obtenues et de ce qui peut être considéré comme une solution au vu du manque de discernement auquel est confronté le concepteur entre ces deux notions.

Cependant, nous pouvons tout de même définir deux types de précisions qui devraient être présentes au sein de l'outil de résolution :

– une précision absolue, qui définit le rayon maximal d'un intervalle solution (soit la moitié de la largeur) pour une variable définie sur un domaine continu. Pour une variable x et son domaine d_x, sa précision $pa(x)$ se calcule comme : $pa(x) = w(d_x)/2$.
– une précision relative indiquant le niveau de confiance qu'a le concepteur par rapport aux valeurs d'une variable. Pour une variable x et son domaine d_x, sa précision $pr(x)$ se calcule comme :

$$pr = (w(d_x)/2)/mid(d_x), \text{ si } mid(d_x) \neq 0,$$

$$pr = (w(d_x)/2)/sup(|d_x|), \text{ sinon.}$$

Où $w(d_x)$ retourne la taille de l'intervalle (soit $|b-a|$), $mid(d_x)$ retourne le milieu du domaine intervalle et $sup(|d_x|)$ la valeur absolue de la borne supérieure de d_x.

La précision absolue correspond à l'erreur numérique maximale existante pour une valeur (c'est-à-dire le milieu d'un intervalle). Cette notion de précision permet alors de bien exprimer la précision liée à l'ordre de grandeur de variables liées à des quantités d'ordre physique. Cela permet au concepteur de faire facilement le rapprochement entre des variables et des unités (quand elles en ont). Par exemple, pour une variable définie en mètre, on peut vouloir obtenir des solutions de l'ordre du millimètre, c'est-à-dire d'une précision absolue de 0.001 mètre.

La précision relative permet au concepteur de définir la confiance accordée à une valeur réelle (en général le milieu d'un intervalle). Elle s'exprime en pourcentage et est calculée à l'aide d'un rapport basé sur une précision absolue et certaines valeurs pertinentes du domaine de la variable considérée. Le milieu de l'intervalle est choisi s'il est non nul et sinon la borne dont la valeur absolue est la plus petite. Si le domaine d'une variable est réduit à une seule valeur, alors la précision relative est de 0%. Par exemple, une variable dont le domaine est de [90,110] est connue à 10% près, puisque le rayon de cet intervalle est égal à 10% de son milieu. Cette notion est bien adaptée, quand il s'agit de définir des phénomènes physiques, dont les variables ont un domaine de variation important. L'exploitation des phénomènes physiques est généralement basée sur des hypothèses. Les lois qui traduisent ces phénomènes prennent alors en compte les incertitudes ou les imprécisions liées à ces hypothèses en utilisant des paramètres basés sur une précision relative. Elle s'exprime alors au travers d'un niveau de confiance exprimé en pourcentage. Ce pourcentage est important à prendre en compte, lors de la description de modèles phénoménologiques afin de ne pas obtenir des solutions trop « précises » qui n'ont alors pas de réalité physique, ou bien dont on ne peut évaluer l'exactitude ou la faisabilité en rapport avec la connaissance exprimée.

Par exemple, les coefficients d'échanges thermiques h sont généralement liés à une précision intrinsèque de plus ou moins 5% dans des conditions expérimentales de laboratoire. Cependant, ils sont souvent utilisés dans conditions où cette imprécision peut atteindre plusieurs dizaines de %.

III.7.5 - Représentation des phénomènes physiques

La connaissance des phénomènes physiques est souvent formalisée par des modèles phénoménologiques établis à partir de résultats expérimentaux, permettant ainsi de définir une loi de comportement. Pour certains phénomènes, une seule loi permet de les représenter dans tout leur champ d'application. Cependant, il est fréquent que les phénomènes complexes soient observés au filtre de différentes conditions expérimentales, et qu'à chaque condition expérimentale corresponde une loi différente. Les conditions expérimentales font en particulier évoluer le domaine de validité de certaines variables. Un modèle couvrant un domaine de validité plus large est obtenu en raccordant des modèles dont les domaines de validité sont restreints. Par exemple, plusieurs types d'écoulement sont pris en compte pour analyser des phénomènes de dégradation d'énergie dans un fluide : écoulements laminaire, transitoire et turbulent. Ces types d'écoulement sont identifiés par le nombre adimensionnel de Reynolds (<2100, [2100..50000], >50000) et, par exemple, le facteur de Fanning lié aux frottements est exprimé de plusieurs manières selon la valeur du nombre de Reynolds (voir figure 14).

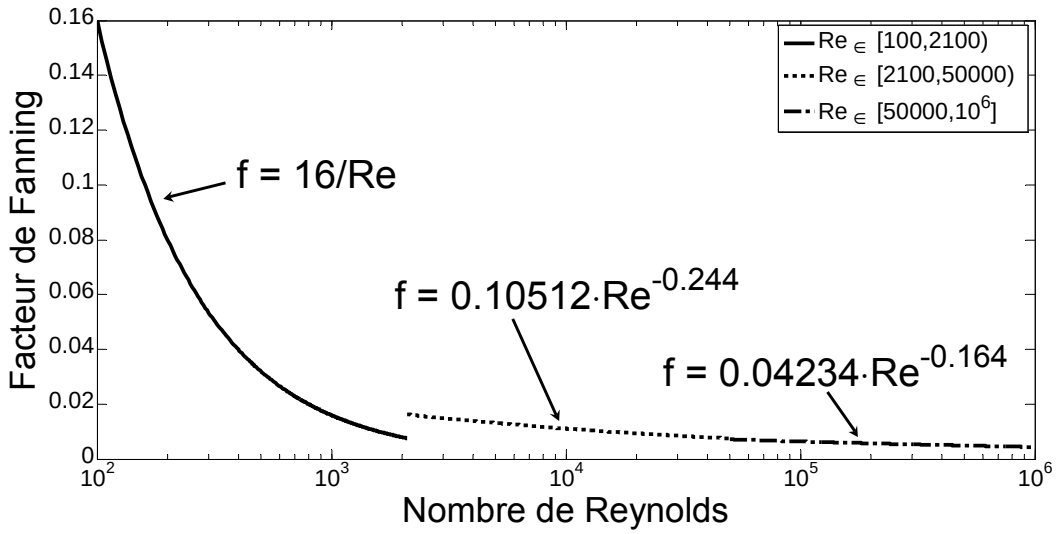


Figure 14 : Expression du facteur de Fanning en fonction du nombre de Reynolds pour l'écoulement d'un fluide dans un tube commercial.

Il faut donc permettre au concepteur de définir un phénomène physique à partir de plusieurs contraintes, chacune ne correspondant qu'à un contexte particulier : une plage de valeurs pertinentes pour une ou plusieurs variables caractéristiques du phénomène. Sans cela,

le concepteur ne peut exprimer correctement la connaissance liée aux phénomènes physiques qu'il doit prendre en compte, ce qui risque de nuire à la validité des résultats, ou bien l'oblige à faire autant de modèles que de plages de valeurs des phénomènes.

L'expression des lois de comportement des phénomènes physiques ne dépend parfois que d'une variable pertinente. Ces comportements se décrivent alors à l'aide de plusieurs contraintes de la manière suivante :

$$x \in [a_1,b_1] \text{ -> } c_1,$$

...

$$x \in [a_k,b_k] \text{ -> } c_k.$$

Un tel phénomène physique est pris en compte dans un modèle comme :

$$\exists\, i \in \{1,...,k\} \ \wedge \nexists\, j \in \{1,...,k\} : x \in [a_i,b_i] \wedge x \in [a_j,b_j] \ \wedge c_i \ \wedge i \neq j.$$

En effet, deux contraintes c_i et c_j ne peuvent être prises en considération en même temps et il faut qu'au moins une contrainte soit active pour appliquer la loi de comportement associée au phénomène physique.

Il est à noter que ces résultats expérimentaux et les valeurs définies par ces modèles sont entachées d'imprécisions même lorsque les conditions expérimentales sont respectées (rapport de la longueur d'écoulement au diamètre d'écoulement, etc.). Elles sont par ailleurs souvent utilisées par les concepteurs en dehors de leur domaine de spécialisation.

III.7.6 - Gestion des situations de vie

Une situation de vie correspond à une étape du cycle de vie ou un cas d'utilisation d'un produit. Il existe plusieurs représentations du cycle de vie, mais nous ne retiendrons que celle de Pahl & Beitz [Pah96]. La figure 15 présente une liste des principales situations de vie existantes pendant le cycle de vie d'un produit [Sca04a].

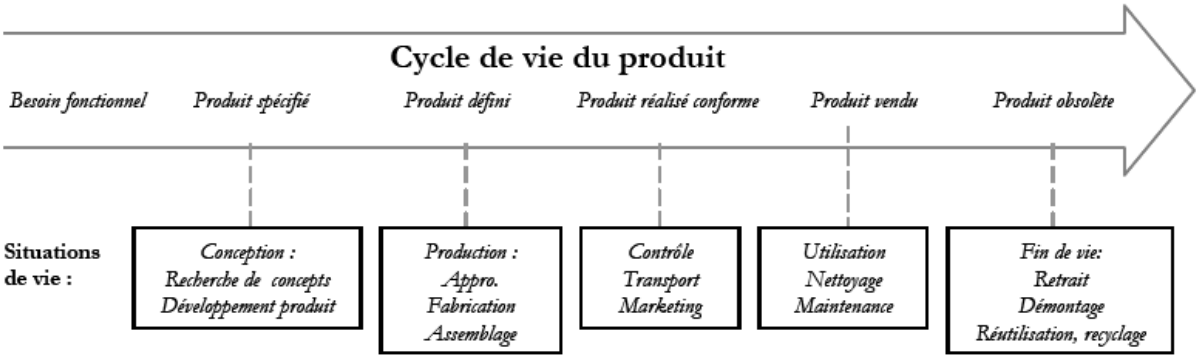


Figure 15 : Situations de vie d'un produit pendant son cycle de vie.

Les situations de vie peuvent être classées et hiérarchisées suivant au moins trois points de vue [Sca04a] : client, entreprise et conception. Pour chaque situation de vie, des fonctions de service (FS) et des fonctions contraintes (FC) sont identifiées (voir chapitre 1). Elles permettent de définir les principales fonctions que doit remplir le système mécanique dans chaque situation de vie. Nous pouvons alors décomposer une situation de vie en trois parties : les fonctions que le système mécanique doit remplir (FS et FC), un contexte environnemental et une description du produit. Ces trois parties permettent de différencier les différentes phases que rencontre un produit. La figure 16 présente l'exemple d'une situation de vie de fonctionnement pour un système de conditionnement d'air dans un avion. Ce dernier doit remplir la fonction de service « conditionner l'air de la cabine de l'avion ». Cinq fonctions contraintes sont identifiées et permettent de mieux définir le contexte d'utilisation de ce système mécanique en fonction de ses milieux extérieurs et du cahier des charges fonctionnel défini.

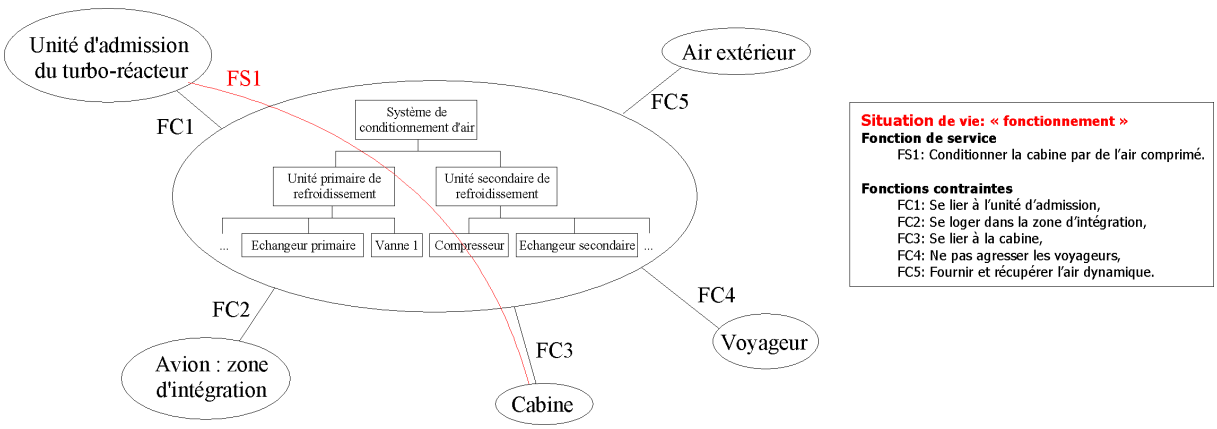


Figure 16 : Exemple de situation de vie de fonctionnement pour un système de conditionnement d'air dans un avion.

Très souvent, les situations de vie sélectionnées par le concepteur en phase de conception architecturale prennent en compte une ou plusieurs phases de fonctionnement du produit conçu, phases qui correspondent à l'exploitation du produit par le client (principaux critères client). Très souvent aussi, l'état du système mécanique ne change pas ou peu (structure organique figée) d'une phase de fonctionnement à l'autre. Par contre, l'environnement du produit diffère pour chacune de ces phases. Elles correspondent alors au fonctionnement du produit dans des environnements différents. Par exemple : les composants d'un avion ne diffèrent pas entre la situation de vie décollage et vol de croisière. Par contre, son environnement (pression, température et vitesse relative de l'air, etc.) évolue beaucoup entre les phases de vols qui correspondent chacune à des phases de fonctionnement différentes.

Nous devons donc décrire chaque situation de vie à l'aide de 3 aspects : contexte fonctionnel, environnemental et d'état. Le contexte fonctionnel détermine les fonctions (FS et FC) que le système doit remplir. Le contexte d'état permet de définir les différents composants du système mécanique et leur arrangement, ainsi que des plages de valeurs pour certaines variables d'état. L'environnement du système mécanique décrit principalement les plages de valeurs pour les variables décrivant l'extérieur du produit et l'entrée/sortie des flux relatifs au produit.

Au final, le concepteur doit étudier le produit dans plusieurs situations de vie avec parfois une structure organique identique, des fonctions changeant peu, mais bien souvent dans des environnements différents. L'étude de plusieurs situations de vie liées à l'utilisation du produit ne nécessite alors que peu de modifications au sein du modèle. Les principales variables qui sont modifiées sont les variables décrivant l'environnement du produit et son contexte d'utilisation.

Cependant, le concepteur est obligé de résoudre autant de modèles que de situations de vie sélectionnées. De nombreux calculs sont alors effectués à l'identique entre plusieurs situations de vie et certaines données ne sont pas réutilisées. Dans un contexte industriel, le traitement de plusieurs situations de vie est souvent trop long et seule la plus prédominante est réellement évaluée. Les autres situations de vie peuvent être évaluées succinctement à partir des résultats obtenus avec la première situation de vie étudiée et des principaux critères spécifiques à chacune d'elles.

Nous pouvons alors essayer de définir un problème de conception global regroupant

toutes les situations de vie à étudier. L'objectif est de fournir au concepteur un problème à résoudre plus riche afin de rendre la résolution plus efficace et éviter de devoir faire autant de modèle que de situations de vie. La séparation du contexte d'utilisation du produit et de sa structure organique permet de mieux envisager les changements existants dans un modèle entre les différentes situations de vie. De plus, le calcul de certaines variables n'est pas spécifique à une situation de vie et les résultats de certaines variables/contraintes peuvent être réutilisés. Un problème de conception peut alors s'exprimer comme :

$$C(VCo, VD^{(0)}, VA^{(0)}) \wedge \left(\bigwedge_{p=1}^{k} S_p(VCo, VD^{(p)}, VA^{(p)}) \right)$$

où C représente l'ensemble des contraintes identiques dans chaque situation de vie étudiée. Nous considérons que 2 contraintes sont identiques si elles représentent la même expression symbolique et si les domaines des variables sont identiques. Les *k* S_p représentent la connaissance spécifique aux *k* situations de vie étudiées. Il faut noter que l'ensemble des variables de conception peut changer pour chaque situation de vie, mais les domaines associés sont toujours les mêmes (ils sont issus de la recherche de concept et/ou du cahier des charges fonctionnel), c'est pourquoi nous ne les indiçons pas. L'ensemble des variables de décision et l'ensemble des variables auxiliaires sont indicées car ils peuvent changer (variables ou domaines) entre chaque situation de vie.

Lorsque nous étudions plus de deux situations de vie pour un même produit, il est possible que nous puissions faire des factorisations supplémentaires entre certaines situations de vie :

$$C(VCo, VD^{(0)}, VA^{(0)}) \wedge$$
$$\left(\bigwedge_{i=1}^{j} \left[C^{(i)}(VCo, VD^{(i)}, VA^{(i)}) \wedge \left(\bigwedge_{p=1}^{k_i} S_{i.p}(VCo, VD^{(i.p)}, VA^{(i.p)}) \right) \right] \right)$$

Chaque $C^{(i)}$ correspond alors à une factorisation de plusieurs situations de vie et les S_{ip} sont les situations de vie factorisées pour un $C^{(i)}$. Nous pouvons utiliser une représentation plus succincte de cette décomposition du problème initial :

$$\bigwedge_{i=0}^{j} \left[C^{(i)}(VCo, VD^{(i)}, VA^{(i)}) \wedge \left(\bigwedge_{p=1}^{k_i} S_{i.p}(VCo, VD^{(i.p)}, VA^{(i.p)}) \right) \right]$$

Nous décomposons en fait le problème suivant les caractéristiques communes aux sous-problèmes de niveaux inférieurs. Les problèmes décomposés peuvent être assimilés à des noeuds ou blocs dans un arbre. Cette décomposition définit, un ordre partiel entre les blocs dont la racine est le problème $C^{(0)}$, qui correspond à l'intersection de toutes les situations de vie étudiées. Cependant, cette décomposition n'est pas unique, puisque des choix différents

de situations de vie, à regrouper et factoriser, définissent des décompositions différentes. Ces choix peuvent être faits en fonction de la taille du problème obtenu par la factorisation. Seule la racine de l'arbre $C^{(0)}$ ne change pas. Il semble évident qu'un trop grand nombre de situations de vie à prendre en compte risque de réduire la taille de $C^{(0)}$.

Ces blocs définissent alors des sous-problèmes plus faciles à résoudre que le système global et dont les résultats influent directement sur les blocs du niveau inférieur qui leur sont attachés. Malheureusement, ces blocs ne sont pas nécessairement bien contraints lorsqu'on les considère séparément. Il faut alors étudier de manière plus importante les stratégies de recherche à employer sur ces différents blocs. Il faut noter que chaque bloc contient un ensemble de variables de conception. Cet ensemble regroupe les variables de conception présentes dans les sous-blocs, mais ces dernières risquent de n'être que très peu contraintes dans un bloc proche de la racine.

Nous pouvons noter que cette décomposition d'un problème peut être assimilée à l'approche objet avec son mécanisme de spécialisation des classes. Le problème se trouvant tout en haut de l'arbre de décomposition est bien le plus général, et chaque niveau supplémentaire définit la spécialisation des situations de vie étudiées par rapport au problème global.

Lorsque les situations de vie étudiées sont très différentes (par exemple : une situation de vie de fonctionnement avec une situation de vie d'assemblage), la décomposition risque de ne pas être efficace. Cette décomposition d'un problème doit surtout permettre au concepteur de prendre conscience des redondances présentes entre les situations de vie qu'il étudie, ainsi que leurs spécificités. Traiter toutes les situations de vie d'un produit dans un même modèle est souvent irréaliste à cause de la complexité globale du problème. Il est préférable de décomposer le problème global en plusieurs sous-problèmes plus faciles à résoudre. Une situation de vie est égale à l'union des blocs de l'arbre de décomposition menant de la racine à la feuille représentant les spécificités de la situation de vie. Il est ainsi facile de prendre en compte plusieurs situations de vie assez proches les unes des autres au sein d'un même modèle, pour réutiliser les calculs communs.

III.8 – Bilan

Dans ce chapitre, nous avons étudié le cheminement nécessaire à l'établissement d'un modèle en conception architecturale. Ce type de modèle fait la synthèse d'une multitude

d'informations sur un produit. En effet, le concepteur exprime la connaissance dont il dispose sur le produit, aussi bien par rapport à la phase de recherche de concepts et à l'étude des phénomènes physiques qui interviennent, que par rapport aux phases de conception suivantes et aux différentes situations de vie que le produit va rencontrer. Le concepteur adapte généralement son modèle afin qu'il réponde au mieux à ses attentes décisionnelles, car la variété des informations réunies et leur apport décisionnel ne sont pas toujours adéquats au contexte de la conception architecturale.

Les variables de conception architecturale, issues du cahier des charges fonctionnel, sont au coeur du modèle établi et représentent généralement ce qu'on appelle les caractéristiques clés du produit [Tho99]. L'objectif du modèle de conception architecturale est de choisir une architecture qui est représentée par les valeurs de ces variables. Les variables de décision sont aussi issues du cahier des charges et permettent d'évaluer les performances des architectures obtenues à l'aide de la résolution du modèle. Le modèle n'a été établi que pour relier les variables de conception aux critères de décision, tout en prenant en compte les phénomènes physiques pertinents, ainsi que d'autres caractéristiques pertinentes du produit.

A l'aide de cette analyse nous avons alors mis en évidence certaines notions importantes pour un concepteur pendant la phase de conception architecturale. Ces notions peuvent être reliées aux besoins logiciels que rencontrent les concepteurs.

CHAPITRE IV

CONCEPTS ET APPORTS EN PROGRAMMATION PAR CONTRAINTES

Cette partie décrit les nouveaux concepts définis en programmation par contraintes pour résoudre des problèmes de conception. Ces concepts permettent de mieux prendre en compte les besoins d'un concepteur au sein d'un solveur CSP afin d'améliorer la résolution et la qualité des solutions calculées. Nous avons alors défini plusieurs heuristiques de recherche spécialisées aux problèmes de conception, mais aussi une contrainte globale pour formuler des phénomènes physiques dont le comportement est défini par plusieurs lois. Nous insistons aussi sur la gestion de la précision au sein des modèles, car elle joue un rôle important pendant la résolution et le concepteur ne sait pas toujours la définir avec pertinence pour toutes les variables. Les incertitudes liées à certains phénomènes nous ont mené à introduire la notion de précision relative.

IV.1 – Etude de l'existant

Pour introduire de nouveaux concepts et les implémenter, nous sommes basés sur la librairie Elisa[3]. Cette librairie a pour objectif de fournir un ensemble de classes C++ permettant de définir des applications faisant appel à une résolution utilisant des techniques de satisfaction de contraintes sur les domaines continus. La plupart des algorithmes et des techniques de résolution pour les CSP numériques, qui a été présentée dans le chapitre II.2, sont présents dans cette librairie. Elle est basée sur une structure objet facilitant l'implémentation de nouveaux concepts et de nouveaux algorithmes.

Les contraintes sont définies comme des égalités ou des inégalités faibles ($\leq$ ou $\geq$). Elles sont basées sur des expressions arithmétiques de variables définies sur des intervalles de réels, qui sont ensuite approximées sur les intervalles flottants [Gou00]. La résolution d'un CSP numérique applique alors un algorithme de type *Branch-and-Prune* [Van97a] utilisant un algorithme classique de *backtrack*.

Des variables discrètes peuvent être définies, mais leur domaine est implémenté par un intervalle raffiné au fur et à mesure, en utilisant les valeurs discrètes du domaine initial.

Une précision absolue peut être fixée pour toutes les variables d'un problème. Cette notion de précision correspond à la taille d'un intervalle (voir chapitre III.7.4). Par défaut elle est définie à 10^{-8}. De même, chaque variable peut avoir sa propre précision absolue. La notion

3 - http://sourceforge.net/elisa

de précision est alors importante pour l'efficacité de la résolution, car les domaines de variables qui ont atteint le seuil qui leur est fixé ne sont plus découpés par l'algorithme de recherche. Elle permet d'arrêter le processus de résolution lorsque toutes les variables ont atteint leur seuil de précision. Lorsqu'une variable discrète est définie, la précision de son domaine intervalle est fixée à 0, afin d'obtenir une unique valeur du domaine discret initial pour chaque solution trouvée.

Durant le processus de résolution, la phase de propagation des opérateurs de réduction/contraction est assurée par un algorithme de point-fixe. Les opérateurs de réduction utilisés implémentent la hull-consistance, la box-consistance et un algorithme de type Newton appliqué aux intervalles [Van98]. Plusieurs stratégies peuvent être choisies par rapport à l'utilisation de ces opérateurs. Dans notre cas, nous utilisons une stratégie qui combine ces trois opérateurs en fonction de la forme des contraintes (voir chapitre II.2.2).

Concernant les algorithmes de recherche, ils sont paramétrés par une stratégie de sélection des variables. Seulement deux heuristiques sont disponibles par défaut : le tourniquet (*Round-Robin*) et la stratégie choisissant le plus grand intervalle (*MaxInterval*). Nous avons alors implémenté l'heuristique choisissant la variable maximisant la *smear function* [Kea95, Mer00] (voir chapitre II.2.3.1). Par défaut, c'est la bissection qui est utilisée pour découper les domaines des variables.

IV.2 – Nouveaux concepts

IV.2.1 – Gestion de la précision

La définition d'une précision pour une variable permet dans le cadre des CSP numériques, de stopper le processus de résolution avant d'atteindre la précision canonique du flottant près, qui est généralement trop importante.

Dans le cadre des problèmes de conception, certaines variables ont un domaine très grand, car elles sont assez mal connues par le concepteur. Ces variables sont souvent associées à des phénomènes physiques interférant avec le comportement du produit. Les lois de comportement de ces phénomènes expriment alors implicitement des imprécisions ou incertitudes, qui correspondent à des hypothèses simplificatrices nécessaires à la formalisation

d'un modèle. Afin de tenir compte de ces imprécisions, nous pouvons utiliser les variables caractéristiques de ces phénomènes. Leurs précisions traduisent alors la confiance du concepteur par rapport à leurs valeurs. Ce niveau de confiance a été établi par des expérimentations et est aussi fonction de l'adéquation entre contexte expérimental et le contexte d'exploitation du modèle. Il s'exprime généralement en pourcentage et traduit une précision relative.

Nous avons implémenté cette notion de précision relative d'une variable comme l'évaluation du rayon de son domaine par rapport à l'ordre de grandeur de ses valeurs. Le rayon du domaine correspond alors à la moitié de sa taille, soit la moitié de sa précision absolue, qui est donc ramenée à l'échelle des valeurs comprises dans ce domaine. Nous évaluons l'ordre de grandeur d'un intervalle en prenant son milieu (mid(I)) s'il n'est pas nul, sa borne supérieure autrement. La précision relative se calcule alors de la manière suivante :

Algorithme 2 : Calcul de la précision relative d'un intervalle

```
GetRelativePrecision (I=[a,b] : un intervalle)
début
      Si a = b alors
            Retourner 0
      Sinon si mid([a,b]) ≠ 0 alors
            Retourner |a-b|/(2*mid([a,b]))
      Sinon
            Retourner |a-b|/(2*|b|)
      Fin si
Fin
```

Ainsi, l'intervalle [0.001,0.002], peut être considéré comme précis en considérant la précision absolue (0.001), contrairement à la précision relative qui dans ce cas est égale à environ 33%. Inversement, l'intervalle [990,1010] est peu précis au regard de la précision absolue, alors que sa précision relative est de 1%. L'évaluation de l'ordre de grandeur d'un intervalle est délicat, lorsque ses bornes ne sont pas du même ordre de grandeur ou que le nombre zéro fait partie de l'intervalle. En effet, le milieu de l'intervalle n'est pas nécessairement représentatif de l'ordre de grandeur moyen de ses valeurs.

De plus, les variables auxiliaires sont moins bien connues par le concepteur. Lorsque le cahier des charges ne leur définit pas de restrictions de domaine et de précisions pertinentes associées, le concepteur ne peut se fier qu'à sa propre expérience. La prise en compte de phénomènes physiques complexes rend difficile la bonne appréciation de ces variables par le concepteur. Il doit alors tenir compte du rôle de ces variables dans le modèle. En effet, il cherche principalement à définir des valeurs pour les variables de conception. L'exploration des variables auxiliaires ne permet que de valider physiquement, géométriquement, économiquement, etc. la réalité du produit. Nous pouvons conseiller au concepteur de se référer à l'ordre de grandeur représenté par les valeurs de ces variables pour définir une précision absolue, ou alors pour définir une précision relative en accord avec les hypothèses effectuées. Il est préférable que ces précisions soient plus larges, pour ne pas explorer trop intensivement ces variables, ce qui risque de nuire à l'efficacité du processus de résolution. De nombreuses solutions supplémentaires peuvent ainsi être calculées alors que les variables de conception ont atteint leur seuil de précision.

Les variables alias sont des intermédiaires de calculs introduites pour faciliter la compréhension du modèle par le concepteur. Le domaine de ces variables est, lui aussi, mal connu par le concepteur. Afin d'éviter des calculs inutiles, l'exploration de leur domaine doit être évitée, puisque, par nature, leurs valeurs sont calculées à partir d'autres variables grâce à la phase de propagation des contraintes. Le plus simple est alors de leur affecter une précision infinie. Elles sont alors perçues dans le processus de résolution comme non découpables, car elles ont atteint leur seuil de précision. Toutefois, certaines variables alias dont l'ordre de grandeur est connu par le concepteur peuvent être exploré pour aider le processus de résolution à réduire l'espace de recherche.

La définition de précisions pertinentes pour les variables est fondamentale pour une résolution efficace. En effet, ces précisions définissent la taille de l'espace de recherche à un instant donné. Cette taille correspond à la combinatoire qui existe entre toutes les valeurs intervalles des domaines. Le nombre de ces valeurs peut être évalué à l'aide de la précision absolue, ce qui permet de calculer la taille de cet espace de recherche :

$$taille(\langle X, D, C\rangle) = \Pi_{i=1}^{n} \frac{w(d_i)}{pa(x_i)}$$, où *w(d$_i$)* calcule la taille de d_i.

Lorsque nous disposons d'une précision relative sur un domaine donné, il est possible de

calculer facilement la précision absolue correspondante.

Pour conclure, nous pouvons considérer que le concepteur doit essayer de ne définir des précisions que sur les variables dont il pense qu'elles sont importantes au sein du modèle et dont il connaît avec suffisamment de discernement leur précision ou leur ordre de grandeur. Cela permet de réduire l'espace de recherche aux seules variables pertinentes d'un modèle pour éviter toute exploration inutile ou inefficace.

IV.2.2 – Heuristiques de recherche

Nous avons vu précédemment que seulement trois heuristiques étaient disponibles au sein de la librairie Elisa. En général, la stratégie de Round-Robin est utilisée, car elle est connue pour être plus robuste. Cependant, les problèmes de conception disposent de spécificités qui ne sont exploitées par aucune de ces trois heuristiques, ce qui rend le processus de résolution inefficace.

IV.2.2.1 – Variables de conception

Dans le chapitre III, nous avons montré l'importance des variables de conception pour le concepteur, lorsqu'il définit un modèle en conception architecturale. En effet, ces variables sont transverses au cycle de vie du produit. En conception architecturale, le concepteur construit son modèle autour de ces variables pour les relier à des critères de performance.

De plus, les incertitudes et les imprécisions liées à des décisions non encore prises ou à l'expression du comportement du produit induisent parfois le fait que certaines parties des modèles soient sous-contraintes (moins de contraintes que de variables) ou que certaines précisions sur les variables sont moins bien connues. Lorsqu'un problème est sous-contraint, beaucoup de solutions sont trouvées, la consistance des domaines étant souvent vérifiée. Ce n'est pas le cas des variables de conception qui sont directement définies en accord avec la connaissance issue du cahier des charges et de la phase de recherche de concepts.

Il nous semble alors important, dans le processus d'exploration de l'espace de recherche, de traduire cette importance des variables de conception. L'exploration précoce des variables de conception permet ainsi de définir au plus tôt une solution de conception, c'est-à-dire, une architecture produit représentée par les valeurs des variables de conception. La phase

de propagation de contraintes permet de vérifier la cohérence des valeurs de ces variables entre elles et avec le reste du modèle. Certaines des variables de conception sont des variables discrètes (choix de caractéristiques : composants, matériaux, topologie), qui permettent de réduire rapidement le contexte d'exploitation des variables auxiliaires en réduisant les incertitudes liées au produit. Les domaines des variables auxiliaires sont alors réduits de manière à satisfaire les contraintes du modèle. Une fois une solution de conception définie, il ne reste plus qu'à vérifier sa faisabilité en explorant l'espace de recherche représenté par les domaines des variables auxiliaires. Nous ne discernons pas ici les variables de décision des variables auxiliaires, car elles ne définissent pas l'architecture du produit, leurs valeurs n'étant que des indicateurs de performance pour le concepteur.

Nous avons alors défini une nouvelle heuristique de recherche permettant de calculer une solution de conception avant d'explorer les domaines des variables auxiliaires (voir algorithme 3).

Algorithme 3 : Algorithme de choix des variables de conception d'abord.

```
VCoFirst (X: ensemble de variables, B: boîte)
Début
      X_d := {x ∈ X : x est une VCo et B_x peut être découpé}
      X_a := {x ∈ X : B_x peut être découpé} - X_d
      Si X_d est vide alors
            x := SelectRoundRobin(X_a)
      Sinon
            x := SelectRoundRobin(X_d)
      Fin si
      Retourner x
Fin
```

Cet algorithme prend en entrée un ensemble de variables et une boîte intervalle associée à ces variables. L'ensemble de variables est décomposé en deux sous-ensembles de variables découpables : les variables de conception et les autres. Tant que l'ensemble des variables de conception n'est pas vide, on choisit une variable dans cet ensemble en utilisant une stratégie de tourniquet. Ensuite, on applique la même stratégie sur l'ensemble des variables auxiliaires.

Cette première heuristique fait l'hypothèse que toutes les variables auxiliaires sont

secondaires et peu importantes pour le processus d'exploration. Cependant, certaines d'entre elles peuvent être associées à des phénomènes physiques ou des caractéristiques se révélant contraignantes pour le dimensionnement du produit. Ainsi, l'exploration de ces variables permet de détecter plus tôt des inconsistances avec l'architecture produit en cours de définition. Il n'est pas facile pour le concepteur d'identifier ces variables, c'est pourquoi nous avons défini une nouvelle heuristique utilisant un ratio pour diversifier la recherche sur l'ensemble des variables auxiliaires. Ce ratio correspond au pourcentage du nombre de découpages à effectuer sur les variables de conception par rapport au nombre de découpages total (voir algorithme 4).

Algorithme 4 : Algorithme de choix des variables favorisant les variables de conception.

```
VCoFirst (X: ensemble de variables, B: boîte, R: ratio entier)
Début
      X_d := {x ∈ X : x est une VCo, B_x peut être découpé}
      X_a := {x ∈ X : B_x peut être découpé} - X_d
      Soit n le nombre de découpages effectués
      Soit n_d le nombre de découpages effectués sur les VCo
      Si X_a est vide ou n = 0 ou 100.n_d < n.R alors
            n_d := n_d + 1
            x := SelectRoundRobin(X_d)
      Sinon
            x := SelectRoundRobin(X_a)
      Fin si
      n := n + 1
      Retourner x
Fin
```

L'algorithme 4 prend en entrée un ensemble de variables, une boîte intervalle associée et un ratio entier. Comme précédemment, l'ensemble de variables est décomposé en deux sous-ensembles de variables découpables : les variables de conception et les autres. On comptabilise alors le nombre total de découpages n et le nombre de ceux effectués sur les variables de conception n_d. Tant qu'il reste des variables de conception à découper et que n_d est inférieur au pourcentage de n défini par le ratio R, une variable de conception est choisi, sinon on choisit une variable auxiliaire. A chaque fois, c'est la stratégie *Round-Robin* qui est

appliquée pour choisir une variable dans un des deux ensembles de variables.

Avec cette nouvelle stratégie, nous pouvons définir une *intensification* sur les variables de conception ou une *diversification* sur les variables auxiliaires. Cette stratégie s'apparente alors à une méta-heuristiques [Blu03], où la *diversification* (ou exploration) désigne un processus visant à récolter de l'information sur le problème et où l'*intensification* (ou exploitation) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Les notions d'intensification et de diversification sont prépondérantes dans la conception des méta-heuristiques, qui doivent atteindre un équilibre délicat entre ces deux phases de recherches. Les deux notions ne sont pas contradictoires, mais complémentaires. Dans notre cas et pour un problème donné, le ratio optimal est aussi difficile à définir. Il ne résulte pas forcément du rapport entre le nombre de variables de conception et le nombre de variables auxiliaires. Le concepteur doit percevoir l'importance des variables auxiliaires dans le modèle pour le définir de manière pertinente.

IV.2.2.2 – Solution de conception

Nous venons de voir l'importance des variables de conception et le rôle qu'elles jouent au sein d'un modèle de conception architecturale. Cependant, plusieurs solutions physiques peuvent exister pour une même solution de conception. Le processus de résolution calcule alors toutes ces solutions qui n'intéressent pas le concepteur. Il ne veut obtenir que des architectures du produit qui valident les contraintes du problème. Nous pouvons alors regrouper ou projeter ces solutions sur les domaines des variables de conception. Dans certains cas, ces solutions font partie d'un continuum de solutions et leur nombre peut être très important, ce qui rend difficile ce regroupement. Il nous semble alors plus pertinent d'éviter d'explorer toutes ces solutions, car une seule solution suffit à valider une architecture produit, définie par les variables de conception.

Afin de calculer uniquement les solutions de conception d'un problème, nous utilisons l'heuristique `VCoFirst` pour choisir les variables à découper. Nous avons spécialisé l'algorithme de *backtrack* pour remonter à la dernière variable de conception découpée, dès qu'une solution est trouvée (voir algorithme 5). Ainsi, nous évitons d'explorer inutilement d'autres solutions correspondant pour le concepteur à la même architecture.

Algorithme 5 : Algorithme de backtrack revenant à la dernière VCo découpée.

```
BacktrackStrategyVCo (St: arbre de recherche)
Début
      CN := noeud courant de St
      Si CN est une solution Alors
            goBackVCo(St)
      Sinon
            goBack(St)
      Fin si
Fin
```

Cet algorithme prend en entrée l'arbre de recherche exploré pendant le processus de résolution. Lorsque le noeud courant correspond à une solution qui vient d'être trouvée, alors nous remontons dans cette arbre jusqu'à la dernière variable de conception découpée, sinon on utilise la stratégie de *backtrack* classique.

IV.2.2.3 – Apprentissage de la sensibilité

Lors de l'exploration de l'espace de recherche d'un CSP numérique, un critère important pour le choix des variables est leur sensibilité numérique. Ce critère est pris en compte par l'heuristique maximisant le *smear function* [Kea95]. Cependant, toutes les formes de contraintes ne sont pas adaptées à cette heuristique basée sur l'évaluation de la matrice jacobienne d'un système d'équations. De plus, les modèles de conception architecturale sont mixtes. La présence de variables discrètes, même si elles sont implémentées comme des variables continues, fausse cette heuristique. Nous verrons aussi par la suite, la notion de contraintes par morceaux, ce qui empêche, là aussi, une évaluation pertinente de la sensibilité des variables dans un modèle. Nous définissons alors la notion de sensibilité numérique dans un problème de satisfaction de contraintes :

Définition (Sensibilité numérique d'une variable) : *La sensibilité numérique d'une variable dans un CSP correspond à l'efficacité de la réduction effectuée sur la taille du problème lorsque cette variable est découpée. Plus une variable est sensible, plus son découpage permet de réduire le problème en fonction de la consistance locale choisie. La sensibilité d'une variable est soumise à son contexte, c'est-à-dire, l'intervalle représentant son domaine,*

l'ensemble de contraintes actives et les opérateurs de réduction utilisés.

Il nous semble difficile de prédire avec exactitude la sensibilité numérique des variables, qui composent les modèles hétérogènes définis en conception architecturale, en se basant sur la connaissance mathématique liée à ces modèles. Cependant, nous pouvons utiliser les constatations relatives aux réductions effectuées sur le problème pour observer la sensibilité d'une variable. Cette observation nous indiquant, dans le contexte précédent, l'efficacité du découpage de la variable choisie. Après avoir observé les réductions effectuées par le découpage de l'ensemble des variables, nous pouvons tenir compte de cette sensibilité pour prédire la variable la plus sensible dans le contexte actuel. Autrement dit, nous pouvons essayer d'apprendre le comportement de la sensibilité numérique des variables du modèle [Gou07]. Le problème est alors de maintenir un ensemble d'observations suffisamment cohérentes avec le contexte présent pour effectuer des prédictions pertinentes.

Nous avons implémenté cette heuristique au travers de l'algorithme 6. Cet algorithme commence par évaluer un coefficient correspondant à la réduction de la taille du problème, suite au découpage du domaine de la dernière variable choisie et à la phase de propagation qui ont été faits depuis. Le coefficient associé à cette variable est alors mis à jour ou simplement ajouté s'il n'était pas présent. Si aucune variable n'avait été découpée précédemment ou que des variables n'ont pas encore été choisies, alors on applique une stratégie de *Round-Robin* parmi ces variables pour retourner celle dont le domaine va être découpé. Une fois que toutes les variables ont été découpées au moins une fois, on se base alors sur le coefficient qu'on leur a affecté pour choisir celle qui a permis d'effectuer la plus grande réduction de la taille du problème.

Algorithme 6 : Algorithme de choix des variables apprenant la sensibilité des variables.

```
LearningSensitivity (X: ensemble de variables, B: boîte)
Début
      Soit lastVar la dernière variable choisie
      Soit sensibleVec un vecteur de (variable, coefficient)
      X_s := {x ∈ X : x apparaît dans sensibleVec}
      Si lastVar n'est pas nulle alors
            coeff_new := evaluateChanges()
            Si  est vide alors
                  Enlever (lastVar, coeff_old) de sensibleVec
            Fin si
            Ajouter (lastVar, coeff_new) à sensibleVec
      Fin si
      Si X − X_s n'est pas vide alors
            lastVar := SelectRoundRobin(X)
      Sinon
            lastVar :=  tel que: (x,coeff_x)  sensibleVec
                        ∄y : (y,coeff_y) ∈ sensibleVec ∧ coeff_x< coeff_y
      Fin Si
      Retourner lastVar
Fin
```

IV.2.3 – Contraintes par morceaux

Les phénomènes physiques établis par des expérimentations peuvent être efficacement représentés par des fonctions par morceaux. Nous nous intéressons plus particulièrement aux fonctions par morceaux, définies uniquement par rapport à un paramètre. Les phénomènes physiques complexes sont en général perçus par les concepteurs au travers de variables pertinentes, telles certains nombres adimensionnels comme le nombre de Reynolds, de Nusselt, etc. en mécanique des fluides. Des travaux précédents décrivent comment définir des fonctions linéaires par morceaux [Ref99], mais nous avons appliqué ce type d'approche à des fonctions non linéaires et nous avons adapté l'approche au contexte de la conception et des phénomènes physiques à décrire. Nous pouvons ainsi prendre en compte ces fonctions par morceaux au sein du formalisme des CSP en définissant une contrainte globale :

$$\texttt{piecewise}(x, I_1 : C_1, \ldots, I_p : C_p)$$

où x est le nom de la variable paramètre, les I_j sont les plages de valeurs pour x, les C_j représentent une contrainte ou un ensemble de contraintes.

Les intervalles I_j ne doivent pas se superposer. La contrainte globale est satisfaite si et seulement si :

- x a un domaine inclus dans un seul I_j.
- le C_j correspondant est satisfait.

Cette contrainte globale est équivalente à la contrainte réifée :

$$\sum_{j=1}^{p} (x \in I_j \wedge C_j) = 1.$$

Nous avons alors défini une nouvelle classe de contraintes, pour exprimer la contrainte globale *piecewise*. Nous avons aussi défini une nouvelle heuristique de découpage des domaines des variables pour prendre en compte les bornes des intervalles I_j pour explorer le domaine de la variable x au plus tôt et ainsi pouvoir appliquer une contrainte C_j. Une fois les bornes exploitée par l'heuristique, le domaine de la variable x est exploré à l'aide de la bissection comme les autres variables.

Nous considérons alors qu'il existe un opérateur de réduction θ_j pour chaque C_j. Nous considérons ensuite le produit cartésien des domaines des variables, représenté par une boîte B. Nous définissons alors un opérateur de réduction pour B en rapport avec la contrainte *piecewise*. Soit B_j la boîte obtenue suite à l'application de θ_j sur B, telle que B_x est remplacé par $B_x \cap I_j$ pour j de 1 à p.
Soit J l'ensemble des index des morceaux actifs pour une contrainte *piecewise* :

$$J := \{j \in \mathbb{N} \ : 1 \leq j \leq p,\ B_x \cap I_j \neq \emptyset\}.$$

Soit K l'ensemble des index des morceaux actifs et consistants :

$$K := \{k \in \mathbb{N} \ : k \in J,\ B^{(k)} \neq \emptyset\}.$$

Il y a ensuite trois cas particuliers dans l'algorithme de réduction :

- Si K est vide, alors la contrainte *piecewise* est violée. B est alors égale à l'ensemble vide et cela englobe le cas de J vide.

- Le domaine de *x* doit être compatible avec les morceaux actifs et consistants. Nous avons donc :

$$B_x := \text{hull}\left(\bigcup_{k \in K} [B_x \cap I_k]\right)$$

où la boîte englobante (hull) d'un ensemble de réel est le plus petit intervalle englobant.

- Le domaine de chaque variable *y* apparaissant dans au moins une C_j peut être réduit en utilisant les contraintes actives. Soit Y l'index défini par :

$$Y := \{j \in \mathbf{N} \ : j \in J,\ y \text{ occurs in } C_j\}.$$

Il y a encore trois cas possibles :

- Si Y est vide, alors B_y n'est pas modifié tant que *y* n'est pas impliqué dans une contrainte active.
- Lorsque Y n'est pas vide, il y a deux cas:
  - Si l'intersection de Y et K est vide, alors *B* est vide. En fait, aucune contrainte active impliquant *y* n'est consistante.
  - Autrement, le domaine de *y* dans *B* est réduit comme suit :

$$B_y := \text{hull}\left(\bigcup_{k \in K} B_y^{(k)}\right)$$

Nous pouvons alors remarquer que cet algorithme implémente une sorte de disjonction constructive. Chaque morceau correspondant à des boîtes réduites, dont l'union est comprise dans la boîte globale réduite. Le calcul de l'enveloppe permet alors de garder des domaines intervalles, comme on peut le voir sur la figure 17.

Cette contrainte globale peut facilement être étendue à plusieurs variables définissant le contexte d'application d'une contrainte ou d'un ensemble de contraintes, mais cela induit une combinatoire supplémentaire entre les intervalles des variables paramétrantes.

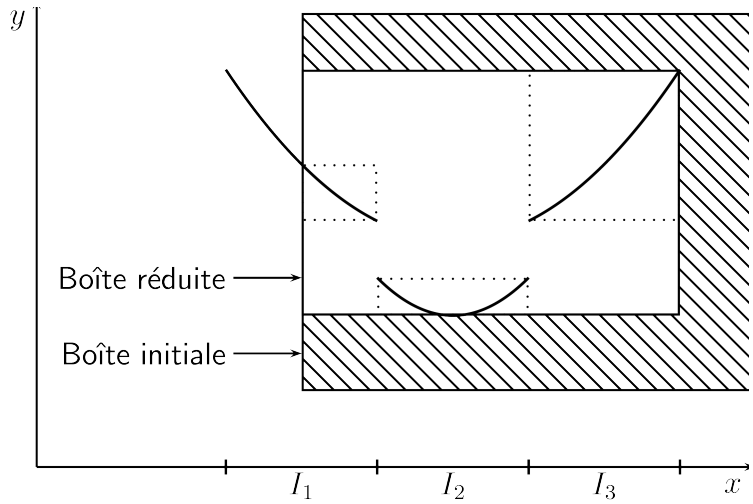


Figure 17 : Exemple de réductions pour une contrainte par morceaux de la contrainte globale piecewise.

IV.3 – Benchmarking

Seules les performances de l'heuristique d'apprentissage de la sensibilité (LS) sont présentées ici. Les autres heuristiques sont évaluées dans le chapitre suivant, car elles n'ont un sens que lorsqu'on traite des problèmes de conception.

Nous avons donc testé cette heuristique sur quelques exemples académiques disponibles avec Elisa et sur les deux premiers problèmes de conception présentés dans le chapitre V. Nous avons comparé ces résultats avec l'heuristique *Round-Robin* (RR) et celle basée sur la maximisation de la *smear function* (SF) (voir table 1).

Les quatre derniers problèmes testés correspondent à des variations du modèle de circuit d'eau présenté dans le chapitre suivant. Pour les test Pompe1 et Pompe3, le catalogue des vitesses de rotation de la pompe n'est pas pris en compte. Pour Pompe3 et Pompe4 nous avons pris en compte une tolérance sur le diamètre du tube en amont de +/- 1mm.

Table 1 : Comparatif des performances des heuristiques RR/SF/LS.

Problème		**Round-Robin**	**SmearFunction**	**LearningSensitivity**
Arm	Solutions	481	578	831
	Découpages	480	625	1094
	Temps (s)	0,1	0,21	0,39
Cyclohexane	Solutions	16	16	16
	Découpages	435	494	996
	Temps (s)	0,2	0,25	0,63
Hunter	Solutions	475	762	476
	Découpages	479	830	476
	Temps (s)	0,44	1,47	0,4
Kinematics1	Solutions	16	16	-
	Découpages	206	171	-
	Temps (s)	0,71	0,65	>300
More_cosnard (17)	Solutions	1	1	1
	Découpages	10	4	5
	Temps (s)	2,62	1,19	1,24
More_cosnard (18)	Solutions	1	1	1
	Découpages	101	11	11
	Temps (s)	19	2,96	3,41
More_cosnard (19)	Solutions	1	1	1
	Découpages	423	26	52
	Temps (s)	73,05	6,98	14,03
More_cosnard (20)	Solutions	1	1	-
	Découpages	660	80	-
	Temps (s)	130	21,19	>300
Doseur	Solutions	895	-	1029
	Découpages	85420	-	7345
	Temps (s)	35,15	>300	11,01
Pompe1	Solutions	6	14	4
	Découpages	51	268	29
	Temps (s)	0,19	1,6	0,34
Pompe2	Solutions	26	-	18
	Découpages	1847	-	242
	Temps (s)	5,62	>300	1,96
Pompe3	Solutions	94	-	58
	Découpages	248	-	186
	Temps (s)	1,92	>300	1,2
Pompe4	Solutions	429	-	279
	Découpages	28873	-	1193
	Temps (s)	108	>300	11,38

L'heuristique LS n'est pas très performante sur les modèles académiques testés, pour lesquels la stratégie SF semble la plus appropriée. Ces modèles sont purement continus et l'évaluation de la matrice jacobienne permet alors de bien cerner la sensibilité de chaque variable. De son côté, la stratégie RR n'est la plus performante que sur le problème Arm, qui est le plus facile à résoudre. Cependant, pour les problèmes de conception LS est plus performante en moyenne de 9 fois par rapport à RR (en considérant le nombre de découpages), alors que SF n'a permis d'obtenir des solutions rapidement que pour un seul problème. La mixité de ces problèmes rend la stratégie SF inefficace, bien que la sensibilité numérique reste un critère important pour le choix des variables.

Sur deux problèmes académiques, LS n'a pas permis d'obtenir une résolution rapide. Elle s'avère encore trop peu robuste pour être utilisée dans des modèles plus complexes, comme ceux décrits dans le chapitre V. Cependant, nous n'évaluons la sensibilité d'une variable que lorsqu'elle a été découpée. Actuellement, le calcul d'un coefficient pour la variable qui a été découpée est basique, mais il est rapide, ce qui permet tout de même d'obtenir des résultats satisfaisants pour 8 des 13 problèmes testés. Nous pensons alors qu'il est possible d'améliorer l'évaluation de ce coefficient en étudiant plus en détail le réseau de contraintes et les réductions effectuées sur chaque domaine après un seul découpage, de manière à inférer à chaque étape la sensibilité de toutes les variables.

IV.4 - Bilan

Dans ce chapitre, nous avons présenté l'implémentation d'algorithmes et de techniques s'insérant dans le processus de résolution des problèmes de satisfaction de contraintes dans la librairie Elisa. Nous avons traduit certaines des notions définies dans le chapitre précédent pour améliorer les performances de résolution des problèmes de conception architecturale, mais aussi la qualité des solutions obtenues. La majorité des résultats de ces algorithmes est présentée dans le chapitre suivant.

Nous avons dans un premier temps introduit plusieurs variantes d'une heuristique visant à favoriser le choix des variables de conception pendant le processus de recherche. Ces heuristiques sont aussi basées sur la stratégie de tourniquet (Round-Robin), ce qui provoque souvent des découpages inutiles, c'est pourquoi nous avons défini une heuristique qui se base sur la sensibilité numérique des variables. Cependant la sensibilité d'une variable est difficile

à évaluer au sein des modèles hétérogènes définis en conception architecturale. Les résultats obtenus ne sont pas suffisamment performants, mais selon nous, des améliorations peuvent être faites pour mieux évaluer cette sensibilité numérique dans un modèle.

La notion de précision nous est aussi apparue comme fondamentale pour la bonne résolution des modèles de conception architecturale. En effet, le concepteur ne connaît pas bien les précisions associées aux variables auxiliaires, ce qui génère la plupart du temps un grand nombre de calculs inutiles. De plus, le concepteur exprime souvent, pour certaines variables, une précision sous la forme d'un niveau de confiance. Ce niveau de confiance permet de définir une précision dynamique en fonction du domaine des variables.

Finalement, nous avons aussi défini une contrainte globale pour traiter les phénomènes physiques définis par plusieurs lois de comportement en fonction d'un paramètre définissant le contexte d'exploitation du phénomène physique.

CHAPITRE V

ÉTUDE DE PROBLÈMES DE CONCEPTION

V.1 - Introduction

Cette partie décrit quelques problèmes de conception utilisés pour valider les algorithmes présentés dans le chapitre précédent. L'étude des systèmes mécaniques décrits dans cette partie, ainsi que l'analyse des modèles représentent une part importante de ce travail de thèse.

Quatre problèmes de conception sont présentés dans ce chapitre. Les deux premiers problèmes sont des problèmes assez simples du point de vue conception. Le premier est un circuit d'eau prenant en compte une pompe et trois réservoirs. La plupart des dimensions sont fixées, cependant la majorité de variables sont continues et la bifurcation dans la partie avale rend tout de même le problème de conception non trivial. Le deuxième problème s'intéresse à la conception d'un échangeur thermique intégré à un doseur/refroidisseur d'apéritif. Le problème de conception est là aussi assez simple, mais il est intéressant dans la mesure où il prend en compte une plus grande proportion de variables discrètes, ainsi qu'un petit catalogue d'ailettes dans la conception de l'échangeur de chaleur. Ces deux modèles ont permis de tester rapidement les algorithmes définis dans le chapitre 3 avant de s'attaquer aux deux derniers modèles qui sont plus proches des modèles traités dans un contexte industriel. Nous nous sommes basés sur des travaux précédents à propos de la conception préliminaire d'un système de conditionnement d'air (SCA) dans un avion [Per02,Seb07]. Afin de simplifier la modélisation, nous avons considéré que l'échangeur de chaleur du SCA ne comportait pas d'ailettes. Une fois que le modèle physique global a été validé sur ce modèle, pour le quatrième problème, nous avons rajouté un catalogue d'ailettes dans l'échangeur [Kay84]. Nous avons ainsi obtenu des solutions comparables à celles calculées à l'aide de méthodes évolutionnaires [Seb07]. Les modèles complets de ces quatre problèmes sont présents en annexe et ont été résolu à l'aide de la librairie Elisa.

Table 1 : Caractéristiques principales des problèmes étudiés.

Problème	**Nombre de variables**	**Nombre de contraintes**	**Particularités**
Circuit d'eau	35 variables : - 3 VCo - 2 VA - 30 alias 19 constantes	40 contraintes : - 30 non linéaires - 3 contraintes *piecewise*	Majoritairement des variables continues. Présence de 3 contraintes *piecewise*.
Doseur/refroidisseur	17 variables : - 5 VCo - 7 VA - 5 alias 5 constantes	14 contraintes : - 4 non linéaires - 3 contraintes *piecewise* (catalogue)	Modèle simplifié du pré-dimensionnement d'un système industriel. Description d'un petit catalogue d'ailettes.
SCA sans ailette	103 variables : - 5 VCo - 12 VA - 86 alias 19 constantes	105 contraintes : - ≈ 80 non linéaires - 4 contraintes *piecewise*	Modèle proche de l'étude du pré-dimensionnement d'un SCA industriel.
SCA avec ailettes	134 variables : - 5 VCo - 62 VA - 67 alias 19 constantes	87 contraintes : - ≈ 80 non linéaires - 2 contraintes *piecewise* (décrivant chacune 48 surfaces d'échange représentées par 56 caractéristiques d'un catalogue)	Modèle similaire à certains pré-dimensionnements de SCA industriels avec un catalogue d'ailettes.

Les problèmes de conception sont présentés en détaillant les enjeux de ces modèles en conception architecturale. Nous présentons brièvement les solutions de conception obtenues. Les résultats sont ensuite analysés et testés par rapport aux différents concepts définis jusqu'ici : variables de conception, heuristiques de recherche, précisions et contrainte *piecewise*.

V.2 Problèmes de conception

V.2.1 - Circuit d'une pompe et de réservoirs

Nomenclature du problème

$\mathbf{A_e}$ Surface intérieure d'un tube (m^2)

D	Diamètre d'un tube (m)
Dr1	Diamètre du tube menant au 1er réservoir aval (m)
Dr2	Diamètre du tube menant au 2ème réservoir aval (m)
H	Hauteur manométrique du circuit (m)
Ha	Hauteur de la conduite verticale amont (m)
Hr1	Hauteur de la conduite verticale avale avant la bifurcation (m)
Hr2	Hauteur de la conduite verticale avale après la bifurcation (m)
Hw	Hauteur d'eau dans le réservoir amont (m)
f	Facteur de Fanning (-)
G	Vitesse massique ($kg/s/m^2$)
g	Accélération de la pesanteur (m/s^2)
L1	Longueur de la conduite horizontale menant au 1er réservoir aval (m)
L2	Longueur de la conduite horizontale menant au 2ème réservoir aval (m)
La	Longueur de la conduite horizontale amont (m)
NPSHrequis	Supplément minimal de pression exprimé en hauteur d'eau qu'il faut ajouter à la pression de saturation à l'entrée de la pompe pour éviter le phénomène de cavitation (m)
NPSHdisponible	Pression disponible exprimée en hauteur d'eau dans le circuit amont (m)
Pamont	Pression dans le réservoir amont (Pa)
Paval	Pression dans les réservoirs avals (Pa)
Psat	Pression de saturation de l'eau dans le circuit amont (Pa)
Pump	Référence de la vitesse de rotation de la pompe (-)
Q	Débit d'eau dans le circuit amont (m^3/h)
R0/D	Rapport entre le rayon du coude et le diamètre du tube (-)
Re	Nombre de Reynolds (-)
S	Section d'un tube (m^2)
V	Vitesse moyenne d'écoulement (m/s)
ΔP	Perte de charge (Pa)
ΔP_{amont}	Perte de charge totales dans le circuit amont (Pa)
δ_0	Angle du coude (°)
ρ	Masse volumique (kg/m^3)
ξ_X	Coefficient de perte de charge (-)

Le problème de conception

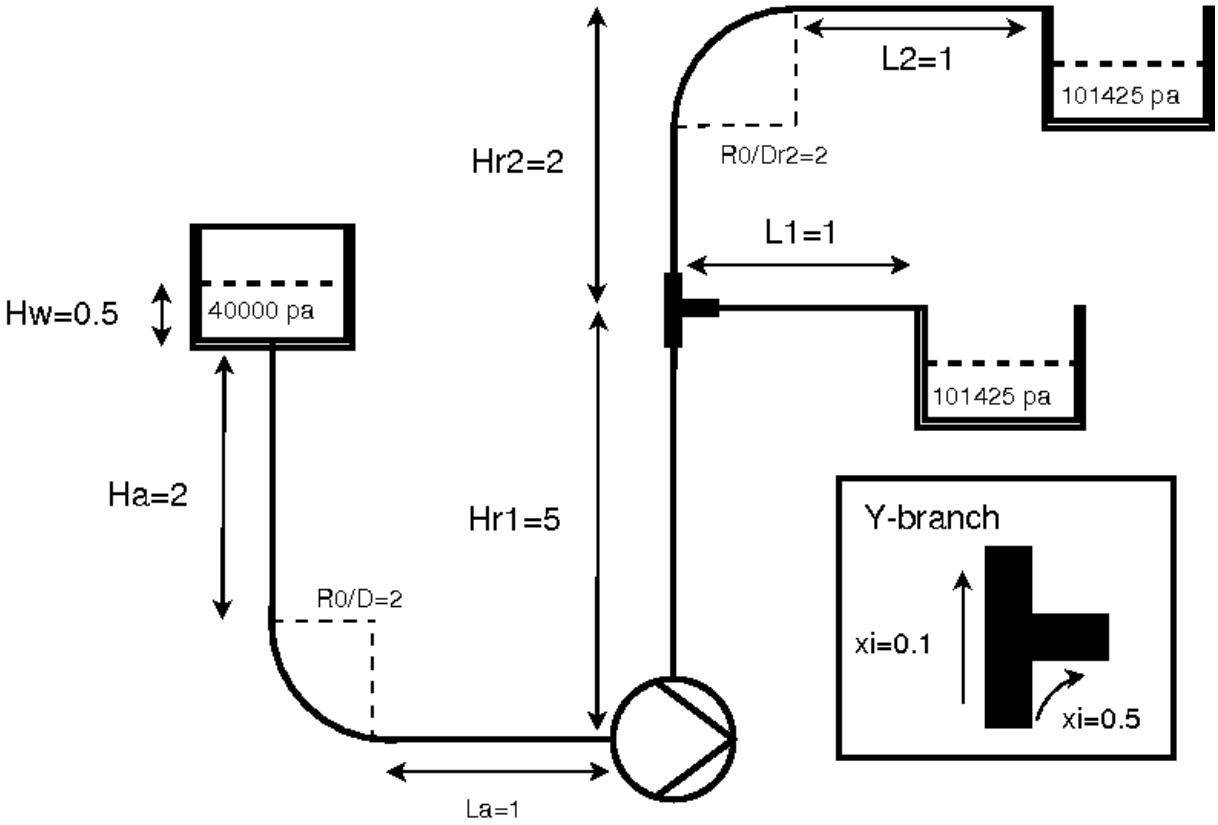


Figure 18 : Schéma d'un circuit d'eau comprenant une pompe et trois réservoirs.

Nous considérons, pour ce problème, un circuit d'eau composé d'un réservoir en amont, de deux réservoirs en aval et d'une pompe (voir figure 18). L'objectif de ce problème de dimensionnement est de définir le diamètre des conduites après la bifurcation dans la partie en aval de la pompe. Les autres caractéristiques du circuit sont déjà choisies (dimensions des conduites, conditions au sein des réservoirs). Le fluide qui traverse la pompe est de l'eau à 20°C de densité 1000kg/m3 et de viscosité dynamique 0.001 kg/m/s. Nous utilisons les caractéristiques d'une pompe monocellulaire centrifuge entraînée à 1450 tr/mn, dont les caractéristiques sont fournies par le fabriquant. Nous utilisons cinq vitesses de rotation différentes de la pompe.

Table 2 : Variables de conception du circuit d'eau.

Variable de conception	Domaine	Précision	Détails
Dr1	[0,01..0,1]	0,001	Diamètre de tube vers le 1er réservoir (m)
Dr2	[0,01..0,1]	0,001	Diamètre de tube vers le 2nd réservoir (m)
Pump	{1,2,3,4,5}	-	Choix de la vitesse de rotation de la pompe

A l'aide des données du fabriquant, nous avons pu définir la hauteur manométrique en fonction d'un débit d'eau compris entre 17 et 96 m^3/h pour chaque vitesse de rotation.

Table 3 : Données constructeur à propos du débit, de la hauteur manométrique et du rendement de la pompe.

Q (m3/h)	0	17	25	44	60	76	86.5	96
H (m)	21,6	21,6	21,5	21	19	16,1	13,6	10,5
□		0,35	0,47	0,7	0,72	0,67	0,58	0,46

Par interpolation, nous obtenons la fonction suivante :

$$H = -1.1763e^{-5}(3600 \cdot Q)^3 - 2.2052e^{-4}(3600 \cdot Q)^2 + 1.4384e^{-2}(3600 \cdot Q) + 21.554$$

Le débit d'arrivée d'eau dans les deux réservoirs amont doit être équivalent afin d'assurer le bon approvisionnement de chacun. Pour éviter des pertes de charge supplémentaires, la somme des deux sections des tubes après la bifurcation doit être égale à la section des tubes du circuit amont. Les principales contraintes prises en compte, en rapport avec des phénomènes physiques, sont les frottements entraînant des pertes de charges dans les conduites et le phénomène de cavitation dans la pompe :

- La perte de charge dans une conduite s'exprime en fonction, du facteur de Fanning qui est un coefficient de frottement liquide moyen équivalent, du rapport de la surface de la conduite à sa section, de la densité du fluide et de sa vitesse massique (rapport du débit massique à la section d'écoulement) :

  $$\Delta P = f \frac{A_e}{S} \frac{G^2}{2\rho}$$

  Pour le calcul du facteur de fanning de conduites cylindriques, nous utilisons des lois qui ont été déterminées statistiquement pour des tubes commerciaux (voir III.7.5) :

  $$\text{si } Re \in [100, 2100] \text{ alors } f = 16/Re,$$

$$\text{sinon si } Re \in [2100, 50000] \text{ alors } f = 0,10512 Re^{-0,244},$$

$$\text{sinon si } Re \in [50000, 10^6] \text{ alors } f = 0,04234 Re^{-0,164}.$$

- La perte de charge dans un coude cylindrique lisse s'exprime en fonction de la vitesse moyenne d'écoulement du fluide dans la conduite, de sa masse volumique et d'un coefficient de pertes de charge singulières. Le coefficient de pertes de charge est fonction de l'angle du coude et du rapport entre son rayon intérieur et son diamètre hydraulique [Ide69] :

$$\Delta P = \xi \frac{\rho V^2}{2},$$

$$\xi = \xi_M + \xi_f,$$

$$\xi_M = A_1 \cdot B_1,$$

$$\xi_f = 0.0175 \cdot 4f \frac{R_0}{D} \delta_0.$$

Dans notre configuration ξ_M vaut 0,15, A1 et B1 étant obtenus à l'aide de valeurs expérimentales. A1 est défini en fonction de δ_0, qui ici vaut 90°, ce qui nous permet de fixer A1 à 1. B1 est lui obtenu en fonction du rapport R_0/D qui ici vaut 2, ce qui nous permet de fixer B1 à 0,15.

- la pression de saturation de l'eau qui alimente la pompe à 13°C (conditions hivernales) vaut alors :

$$P_{sat} = 1496.13 \text{ Pa}.$$

- Le calcul du NPSH (Net Positive Suction Head) requis est obtenu en se basant sur les données constructeur.

Table 4 : Données constructeur à propos du débit et NPSH requis de la pompe.

Q (m3/h)	17	25	44	60	76	86.5	91	96
NPSH requis (m)	1,08	1,09	1,3	1,7	2,5	3,3	3,9	4,7

Ces données peuvent être utilisées dans un modèle mathématique en utilisant la fonction interpolée suivante :

$$NPSH_{requis} = 1.2144e^{-5}(Q \cdot 3600)^3 - 1.2301e^{-3}(Q \cdot 3600)^2 + 4.9136e^{-2}(Q \cdot 3600) + 0.49957$$

- Le NPSH disponible est évalué à l'aide de la contrainte suivante :

$$NPSH_{dispo} = \frac{P_{amont} P_{sat}}{\rho \cdot g} + (Ha + 2D) - \frac{\Delta P_{amont}}{\rho \cdot g}$$

La comparaison du NPSH requis et du NPSH disponible permet de vérifier la non

présence du phénomène de cavitation dans le système.

Les critères permettant de valider les diamètres à déterminer sont : la non cavitation dans la pompe et l'efficacité de celle-ci par rapport au débit d'eau.

Les solutions

Le modèle du circuit de pompe permet d'obtenir un ensemble de six solutions de conception comme présenté sur la figure 19 :

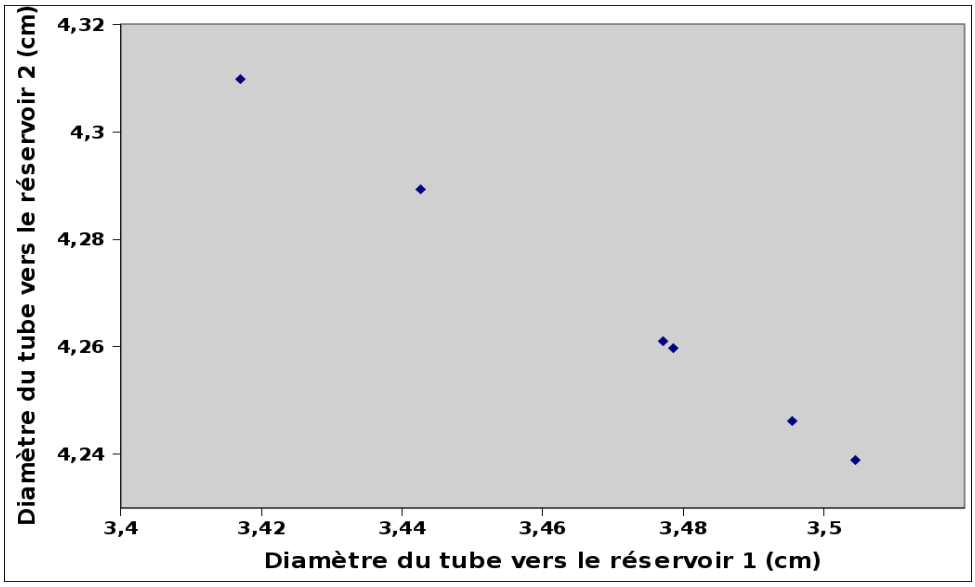


Figure 19 : Solutions du circuit d'eau en considérant les 2 variables de conception.

Deux de ces solutions sont très proches. Elles sont un artéfact de calcul lié à la gestion de la précision par le solveur ainsi qu'au caractère aléatoire du découpage des domaines. Ces deux solutions ne correspondent en réalité qu'à une seule solution de conception. Pour éliminer une des solutions redondantes, il suffit d'accroître la précision du calcul afin que les opérateurs de réduction puissent éliminer la quasi-solution. Les cinq solutions restantes correspondent aux cinq vitesses de rotation effectives de la pompe. Les résultats sont obtenus en quelques dixièmes de secondes. Cette application est intéressante pour un concepteur dans la mesure où la résolution du problème couplé avec présence d'un embranchement dans un circuit ne nécessite pas de programmation par rapport à d'autres outils et ce modèle est résolu quasi instantanément une fois que les contraintes sont exprimées.

V.2.2 – Échangeur d'un système de refroidissement d'apéritif

Nomenclature du problème

A	Aire de la surface d'échange (m^2)
ail	Référence du choix de la longueur des ailettes (-)
C_p	Capacité calorifique de l'apéritif (J/(kg.K))
diam	Référence du choix du diamètre du tube (-)
e	Espace entre ailettes (mm)
f_i	Coefficient de rendement d'ailette (-)
h	Coefficient d'échange convectif ($W/(m^2.K)$)
mater	Référence du choix de matériaux pour les ailettes (-)
N	Nombre d'ailettes dans le doseur
Nu	Nombre de Nusselt (-)
NUT	Nombre d'unité de transfert (-)
t	Temps de refroidissement (s)
T_f	Température finale (K)
T_i	Température initiale (K)
T_{sat}	Température de saturation de l'ammoniac (K)
V	Volume du doseur (cl)
ε	Efficacité du doseur (-)
η	Rendement d'ailette (-)
λ	Conductivité d'ailette (W/m/K)
ρ	Masse volumique de l'apéritif (kg/m^3)

Le problème de conception

Ce système mécanique a pour objectif de refroidir une dose d'apéritif pour arriver à 8°C en moins de 25s (voir figure 20). Ce système autonome permet, à partir d'une réaction chimique, de produire, au choix, de la chaleur ou du froid. Contrairement à un réfrigérateur, ce procédé ne comporte pas de compression frigorifique mais utilise, en revanche, une solution de sel pour ses facultés d'adsorption de l'ammoniac.

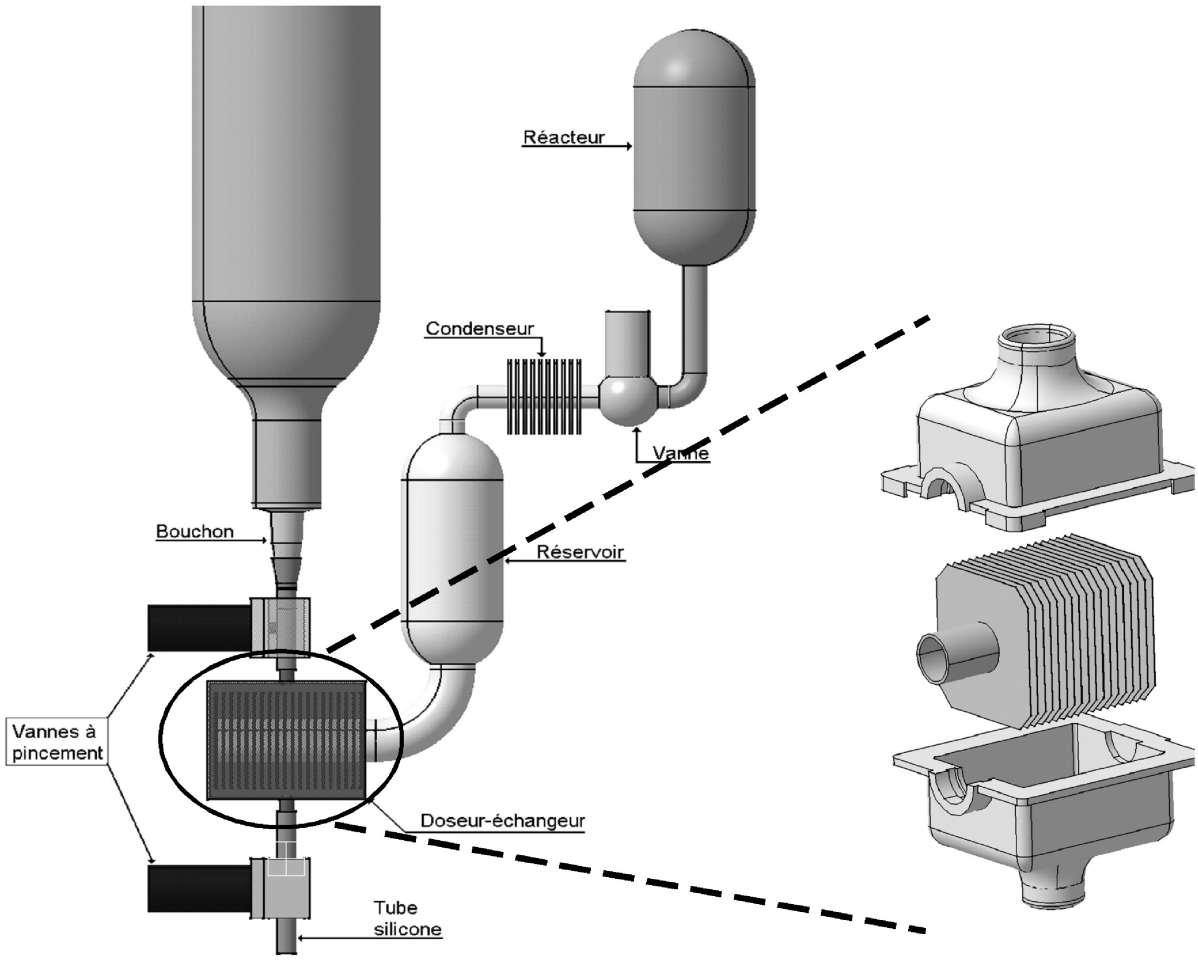


Figure 20 : schéma du refroidisseur d'apéritif.

En fonctionnement, l'ouverture de la vanne provoque une chute de pression et l'ammoniac, se trouvant dans le tube du doseur-échangeur, se vaporise pour rejoindre le réacteur et y être adsorbé par la solution de sel. Cette évaporation entraîne une production de froid : c'est le principe de la réfrigération. Quand le réactif est saturé, la réaction d'adsorption s'arrête ainsi que la production frigorifique. Il faut alors recharger le système en chauffant le sel réactif. L'apport de chaleur induit la séparation du sel et de l'ammoniac gazeux qui rejoint alors son réservoir en se liquéfiant dans le condenseur. Nous nous concentrons alors sur la partie liée au dosage/refroidissement de l'apéritif et nous étudions les échanges de chaleur entre l'ammoniac et l'apéritif.

La figure 21 présente l'organigramme technique étendu aux milieux extérieurs du doseur. Les flux fonctionnels correspondent aux transferts d'énergie calorifique entre l'apéritif et l'ammoniac. L'apéritif transfère de la chaleur aux ailettes, qui la transfèrent ensuite au tube,

qui finit par la transmettre à l'ammoniac. Nous supposons alors que le doseur-échangeur est calorifugé, et que les pertes d'énergie entre les corps 1 et 2 vers l'air extérieur sont négligeables. Nous appliquons alors une loi de conservation de l'énergie dans notre système. Pour cela, nous étudions les transferts d'énergie au niveau des ailettes et nous utilisons une loi de comportement thermique, prenant en compte le gradient de température sur les ailettes selon leur épaisseur et leur matériau.

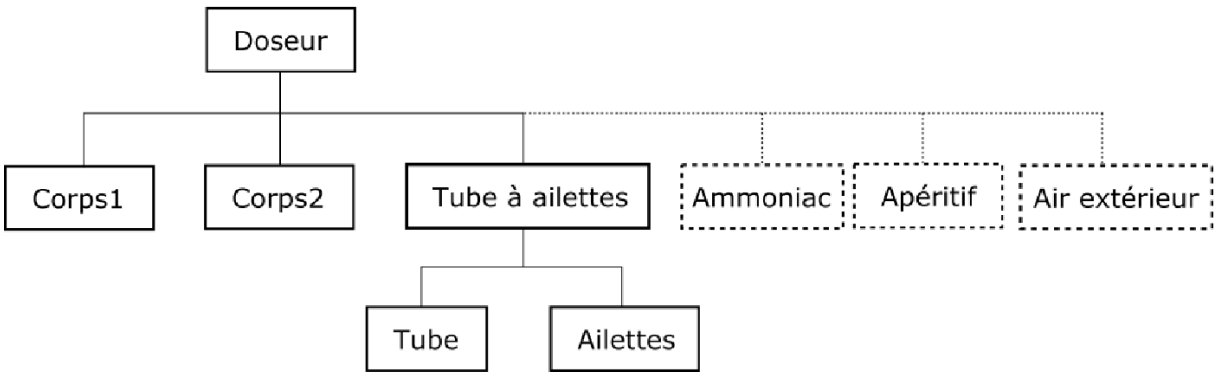


Figure 21 : Organigramme Technique étendu du doseur.

Dans le cadre du doseur-échangeur, nous pouvons alors exprimer la loi de conservation de l'énergie en exprimant le nombre d'unité de transfert (NUT) du système :

$$NUT = \frac{h \cdot A \cdot \eta \cdot t}{V \cdot \rho \cdot C_p}$$

où h est le coefficient d'échange convectif, A est l'aire représentant la surface d'échange, η le rendement d'ailette, V le volume du doseur, ρ la masse volumique de l'apéritif et Cp sa chaleur massique (en J/(kg · K)).

Nous devons donc exprimer h, tout en prenant en compte l'hypothèse que le fluide est quasi-immobile entre les ailettes, du fait de leur proximité et de la viscosité de l'apéritif ($e < 4 \cdot 10{-}3$ m et $\mu > 2 \cdot 10{-}3$ kg/(m · s)) :

$$h = Nu\frac{\lambda}{2e}$$

D'après nos hypothèses et la connaissance du concepteur par rapport à ce type de phénomènes physiques, nous pouvons prendre en compte un nombre de Nusselt de l'ordre de 4 pour effectuer les calculs.

Ensuite nous déterminons l'efficacité ε de notre système à l'aide des relations :

$$\epsilon = \frac{T_f - T_{sat}}{T_i - T_{sat}} \text{ et } \epsilon = e^{-Nut},$$

Dans l'exemple du doseur-échangeur, nous utilisons un catalogue de matériaux, de tubes et d'ailettes. Ainsi, nous disposons de trois VCo correspondant aux références des catalogues, qui nous permettent ensuite d'exprimer A et λ, qui sont alors des VA. Le nombre d'ailettes (N) et leur espacement (e) sont des caractéristiques dimensionnantes et donc des VCo. T_i, T_f, T_{sat}, etc. sont toutes des VA, dont certaines sont fixées pour définir le contexte d'utilisation.

Table 5 : Variables de conception du refroidisseur/doseur.

Variable de conception	Domaine	Précision	Détails
mater	{1,2}	-	choix du matériau des ailettes
diam	{1,2}	-	choix du diamètre du tube
ail	{1,2}	-	choix de la longueur des ailettes
N	{5,6,...,20}	-	nombre d'ailettes
e	[1..4]	-	espace entre ailettes (mm)

Nous obtenons alors un modèle (voir annexes) comportant 14 contraintes, dont 6 qui décrivent un catalogue d'ailettes et de tubes. Nous voyons aussi que certaines VA comme h, A, f_i et η s'expriment directement en fonction des autres variables du modèle. Nous pouvons alors les définir comme des variables alias. Le volume du doseur V, le volume de la dose à refroidir, la température initiale T_i et la température finale T_f sont fixées par les exigences du cahier des charges.

Les solutions

L'échangeur du doseur/refroidisseur d'apéritif a été traité avec succès en utilisant les algorithmes détaillés dans le chapitre précédent. Nous obtenons alors 144 solutions présentées sur les figures 22, 23, 24 et 25.

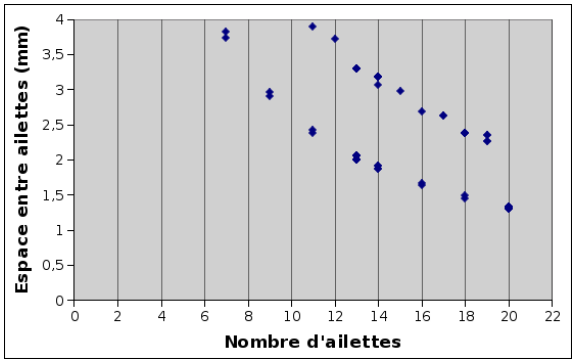


Figure 22 : Solutions en fonction du nombre et de l'espacement des ailettes.

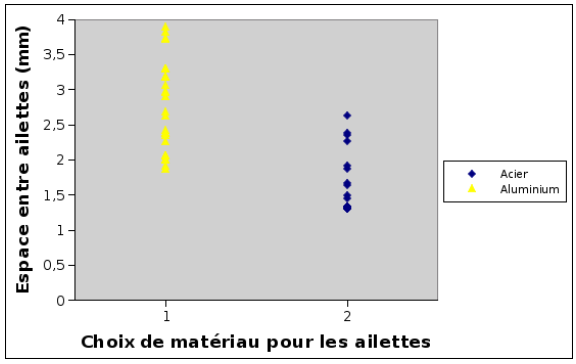


Figure 23 : Solutions en fonction du matériau utilisé et de l'espacement des ailettes.

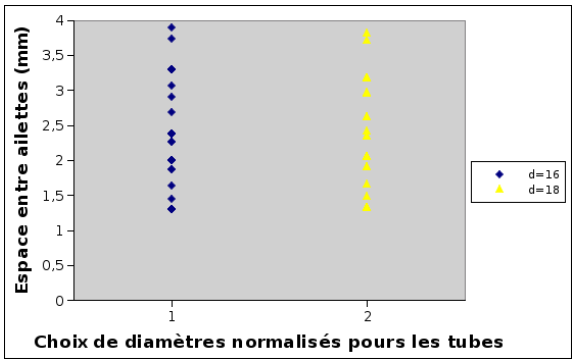


Figure 24 : Solutions en fonction des diamètres normalisés des tubes et de l'espacement des ailettes.

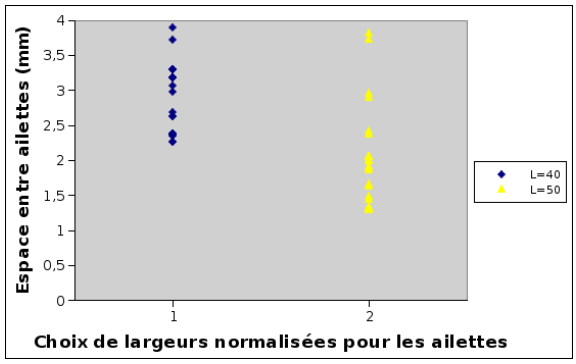


Figure 25 : Solutions en fonction des largeurs normalisées et de l'espacement des ailettes.

Ces quatre figures permettent de visualiser l'espace de solutions selon des perspectives différentes. Sur la figure 22, nous pouvons voir que l'espace de solution est divisé en deux. Nous pouvons identifier une première famille de solutions dont l'espacement entre ailettes ne descend pas en dessous de 2 mm et n'utilise que de 11 à 19 ailettes, alors que pour la deuxième famille 7 à 20 ailettes sont utilisées et sont espacées de 1,2 à 4 mm. Les figures 23 et 24 nous permettent de supposer que le choix de matériau et le diamètre du tube ne permettent pas d'identifier ces familles, contrairement au largeurs normalisées des ailettes dont la répartition correspond à celle de la figure 22. Le diamètre du tube ne semble pas déterminant dans la conception de cet échangeur, puisqu'autant de solutions sont trouvées pour les deux diamètres disponibles. Par contre, le choix de matériau est important, puisqu'avec l'aluminium les ailettes sont globalement plus espacées qu'avec l'acier. Sur cet exemple, nous observons une fois de plus la présence de quasi-solutions liées à la gestion par le solveur des précisions et des découpages.

V.2.3 - Système de conditionnement d'air sans ailette

Nomenclature du problème

Variables :

A	Aire totale des surfaces d'échange (m^2)
A_p	Aire d'un côté de plaque (m^2)
Cp	Capacité calorifique de l'air (J/kg.K)
CPam	Coefficient de Perte de charge en entrée de l'échangeur (-)
CPd	Coefficient de Perte de charge dynamique de l'échangeur (-)
CPr	Coefficient de Perte de charge régulière de l'échangeur (-)
CPav	Coefficient de Perte de charge en sortie de l'échangeur (-)
G	Débit massique de l'air ($kg/s/m^2$)
h	Coefficient de transfert convectif ($W/m^2/K$)
J	Coefficient de Colburn (-)
k_p	Conductivité de plaque (W/m/K)
L_x	Largeur de l'échangeur de chaleur (m)
L_y	Longueur de l'échangeur (m)
L_z	Hauteur de l'échangeur (m)
Nut	Nombre d'unité de transfert (-)
p2	Pression d'entrée de l'air principal dans l'échangeur (Pa)
p3	Pression de sortie de l'air principal dans l'échangeur (Pa)
pei_c	Pression d'entrée de l'air dynamique dans l'échangeur (Pa)
peo_c	Pression de sortie de l'air dynamique dans l'échangeur (Pa)
Pr	Nombre de Prandtl (-)
q_{ma}	Débit d'air principal dans l'échangeur (kg/s)
q_{ra}	Débit d'air dynamique dans l'échangeur (kg/s)
rh	Espace entre plaque (m)
R_v	Rayon de la vanne d'équilibrage (m)
T0	Température dans le turbo-réacteur (K)
T2	Température d'entrée de l'air principal dans l'échangeur (K)
T3	Température de sortie de l'air principal dans l'échangeur (K)
Tei_c	Température d'entrée de l'air dynamique dans l'échangeur (K)
Teo_c	Température de sortie de l'air dynamique dans l'échangeur (K)
t_p	Épaisseur de plaque (m)
ε	Efficacité de l'échangeur (-)

ΔPe	Perte de charge dans l'échangeur (Pa)
λ	Rapport de capacités calorifiques massiques (-)
ρ	Masse volumique de l'air (kg/m^3)
τ	Rapport de débit massique coté froid/côté chaud (-)

Indices

c	Côté air dynamique de l'échangeur
h	Côté air principal de l'échangeur

Le problème de conception

L'objectif de ce problème est d'étudier la faisabilité d'un système de conditionnement d'air dans un avion, tout en prenant en compte, dès la phase de conception préliminaire, les phénomènes physiques principaux liés aux transformations thermodynamiques, aux transferts thermiques et aux dégradations de l'énergie. La conception d'un système de conditionnement d'air dans un avion est soumise à de nombreuses contraintes critiques. En effet, l'air fourni à la cabine ne doit pas nuire à la santé et au bien-être des personnes à bord, mais les performances de l'avion ne doivent pas être non plus trop dégradées afin de garantir son bon fonctionnement (augmentation de la traînée, diminution de la poussée, encombrement et poids du système). Tous ces critères sont difficiles à hiérarchiser, d'autant plus que certains évoluent de manière contradictoire.

Le système de conditionnement d'air est étudié dans une seule situation de vie correspondant au vol de croisière. Les conditions atmosphériques sont alors fixées : l'altitude est de 10500 mètres, la vitesse de l'avion est de Mach 0,8. La figure 26 représente les composants et les flux qui traversent le système de conditionnement d'air pendant la situation de vie étudiée. Le modèle complet est présenté dans les annexes. Deux principaux flux sont mis en évidence : le flux d'air principal pris sur le turbo-réacteur qui va dans la cabine et le flux d'air dynamique permettant de refroidir le flux d'air principal dans l'échangeur de chaleur.

Ce type de système de conditionnement d'air est basé sur un cycle de Joule-Brayton, qui diffère du cycle compression de vapeur des systèmes de refroidissement plus classiques, comme dans les réfrigérateurs par exemple. Les technologies à compression de vapeur sont encore peu utilisées du fait des contraintes liées à la masse du système. Ce système prend de

l'air depuis un ou plusieurs turbo-réacteurs de l'avion en sortie de compresseurs. Une fois prélevé, le flux d'air est comprimé une nouvelle fois pour accroître sa température et faciliter le rejet de l'énergie calorifique dans l'atmosphère. L'extraction de la chaleur est réalisée dans un échangeur thermique. L'échangeur utilise un flux d'air froid provenant de l'atmosphère pour refroidir l'air principal. Une turbine diminue ensuite la pression et la température de l'air principal, tout en transmettant de l'énergie mécanique au compresseur à l'aide d'un arbre de transmission. Cependant, l'air principal est parfois trop froid pour être envoyé en tant que tel dans la cabine ; une vanne permet de réguler la température de l'air en mélangeant l'air en sortie de turbine avec de l'air chaud prélevé du turbo-réacteur. Le compresseur, la turbine et l'arbre de transmission forment une turbo-machine. Avec l'échangeur, cette turbo-machine est souvent appelée *bootstrap* en raison de sa topologie qui rappelle celle d'une boucle. Il est important de noter que cette configuration reste simplifiée au regard des architectures, aujourd'hui les plus utilisées, qui intègrent trois échangeurs supplémentaires et un extracteur d'eau pour déshumidifier le flux d'air.

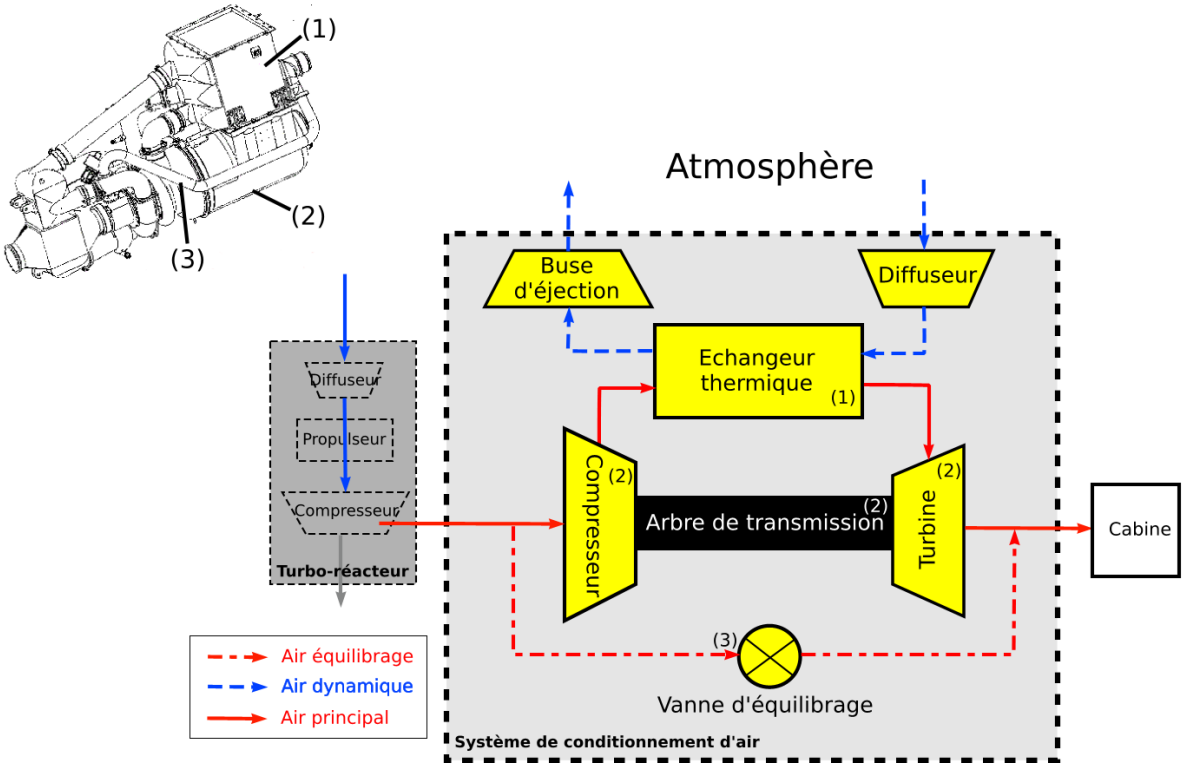


Figure 26 : Bloc diagramme fonctionnel d'un système de conditionnement d'air dans un avion.

L'analyse du fonctionnement du système et de son processus de conception ont révélé

l'importance et la criticité du rôle de l'échangeur thermique. Ce composant a une masse importante (≈50% de la masse total du système), est encombrant et conditionne directement les performances du système. Dans le cadre d'un bureau d'étude, sa structure interne et ses performances ne sont pas toujours suffisamment maîtrisées par les architectes systèmes, qui ne n'étudient pas les ailettes et leur arrangement. Nous avons donc développé un modèle détaillant le système de conditionnement d'air jusqu'au niveau des formes et des géométries internes à l'échangeur. Par contre, les autres composants sont moins détaillés et leur fonctionnement est envisagé à l'aide d'un modèle boîte noire, car ils sont moins déterminant et mieux connus par rapport à l'échangeur.

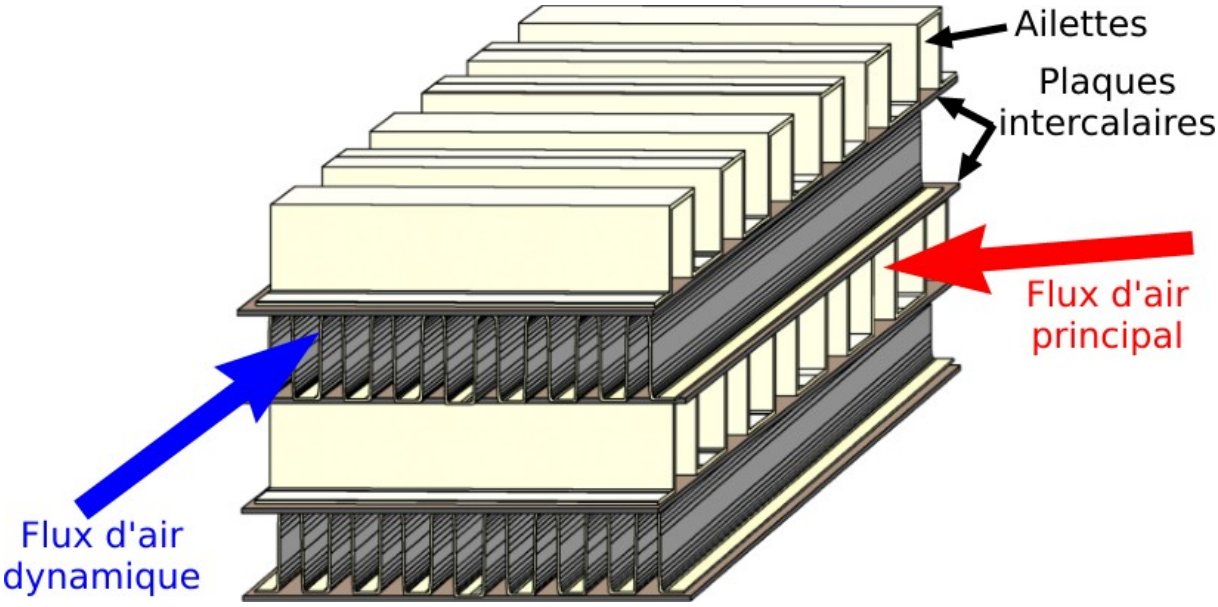


Figure 27 : Exemple d'échangeur thermique formé d'ailettes entre les plaques (intercalaires).

Les caractéristiques de l'échangeur sont des caractéristiques clés du système global. L'échangeur est décrit par ses dimensions principales (Lx, Ly, Lz) et par l'écart entre plaques (rh). Dans un premier temps, nous ne considérons pas d'ailettes entre les plaques de l'échangeur. Des rapports définis entre la hauteur, la largeur et la longueur de l'échangeur sont fixés pour garantir que l'échangeur soit bien proportionné. Par la suite, seule la largeur de l'échangeur est utilisée comme variable de conception, les autres étant déduites à partir de ses valeurs.

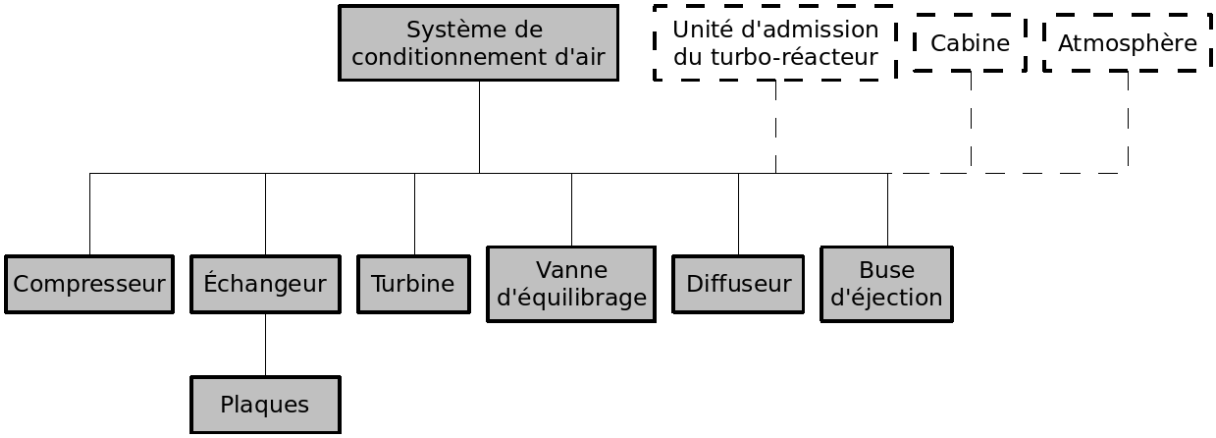


Figure 28 : Organigramme technique étendu du système de conditionnement d'air sans ailette.

Le ratio τ représentant le rapport entre le débit d'air dynamique et d'air principal qui traversent le système est pris en compte comme une pseudo variable de conception. En effet, τ est directement lié à la section de l'écope qui sert à prélever de l'air à l'extérieur de l'avion pour alimenter le circuit d'air dynamique. Comme la section de l'écope n'est pas utilisée dans le reste du modèle, il n'est pas utile de la prendre en compte explicitement. Ainsi τ est relié à une dimension principale du système de conditionnement d'air. Le débit d'air principal *q,* pris sur le turbo-réacteur, est aussi directement associé à la capacité nominale de l'échangeur. Cependant, *q* est défini comme une constante, car le débit d'air à amener dans la cabine est fixé par le cahier des charges et la situation de vie étudiée.

Table 6 : Variables de conception du SCA sans ailette.

Variable de conception	Domaine	Précision	Détails
L_x	[0,3..0,8]	0,01	largeur de l'échangeur de chaleur (m)
rh_h	[0,01..0,1]	0,0001	espace entre plaques côté chaud (m)
rh_c	[0,01..0,1]	0,0001	espace entre plaques côté froid (m)
tau	[0,8..2]	0,01	rapport des débits (côté froid/côté chaud)
R_v	[0.01..0,2]	0,01	rayon de la vanne (m)

Le fonctionnement des composants du système de conditionnement d'air est décrit en utilisant des bilans d'énergie, basés sur les températures et les pressions en entrée et en sortie de chaque composant. L'échangeur de chaleur est plus détaillé. Les modèles de transfert thermique dans l'échangeur s'appuient sur une description de type nombre d'unités de

transfert/efficacité :

$$Nut = 1/(\frac{1}{h_h \cdot A_h} + \frac{t_p}{k_p \cdot A_p} + \frac{1}{h_c \cdot A_c})/(q_{ma} \cdot Cpi_h)$$

où h_h le coefficient de transfert convectif pour les plaques côté chaud, A_h la surface totale des plaques côté chaud, t_p l'épaisseur d'une plaque, k_p la conductivité d'une plaque, A_p la demie surface totale des plaques, h_c le coefficient de transfert convectif pour les plaques côté froid, A_c la surface totale des plaques côté froid, q_{ma} le débit d'air principal dans l'échangeur et Cpi_h la capacité calorifique de l'air en entrée de l'échangeur. Le coefficient de transfert convectif pour le côté chaud et pour le côté froid sont alors exprimés à l'aide des contraintes suivantes :

$$h_h = J_h \cdot e^{-\frac{2}{3} \cdot ln(Pr_h)} \cdot G_h \cdot Cp_h,$$

$$h_c = J_c \cdot e^{-\frac{2}{3} \cdot ln(Pr_c)} \cdot G_c \cdot Cp_c.$$

où J_h et J_c sont les coefficients de Colburn pour le côté chaud et froid, Pr_h et Pr_c les nombres de Prandtl pour le côté chaud et froid, Cp_h et Cp_c les capacités calorifiques moyennes de l'air principal et dynamique dans l'échangeur.

Le coefficient Colburn et le nombre de Prandtl, sont définis à partir des nombres de Standton, Nusselt et Reynolds. Ces nombres sont adimensionnels et sont introduits dans le modèle, car ils permettent de caractériser le couplage entre les phénomènes physiques liés à la mécanique des fluides et aux transferts thermiques.

L'efficacité de l'échangeur thermique se définit comme :

$$\epsilon = \frac{T_2 - T_3}{T_2 - T_0}$$

Cette efficacité peut se relier au nombre d'unité de transfert, caractéristiques des dimensions de l'échangeur :

$$\epsilon = 1 - e^{\lambda \cdot e^{0.22 \cdot log(Nut)} \cdot (e^{-\frac{1}{\lambda} \cdot e^{0.78 \cdot log(Nut)}} - 1)}.$$

Le fait qu'il n'y ait pas d'ailettes entre les plaques, nécessite de considérer des espaces entre plaques assez faibles, pour compenser la surface d'échange qu'auraient pu offrir des ailettes. Ce modèle est donc assez peu réaliste d'un point de vue technologique, mais reste tout de même cohérent et il permet de tester la faisabilité du système. L'ajout d'ailettes permettra de réduire le nombre de plaques, ce qui tendra à réduire la masse du système; les surfaces à

ailettes ayant une masse propre importante, elles ne permettent pas néanmoins de réduire drastiquement la masse de l'échangeur.

Afin d'évaluer le niveau de dégradation de l'énergie dans le système, nous calculons la chute de pression de l'air circulant dans l'échangeur. Ces chutes de pression sont des pertes de charge que nous évaluons pour le côté chaud et le côté froid. Nous disposons de l'expression suivante [Ros85] :

$$\Delta Pe_h = \frac{G_h^2}{2\rho_2}(CPam_h + CPd_h + CPr_h + CPav_h) \text{ et}$$

$$\Delta Pe_c = \frac{G_c^2}{2\rho_{ic}}(CPam_c + CPd_c + CPr_c + CPav_c).$$

Ces pertes de charges sont par définition égales à la différence de pression de l'air entrant et sortant de l'échangeur. Le bilan de pression de chaque côté de l'échangeur s'écrit :

$$\Delta Pe_h = p_2 - p_3 \text{ et } \Delta Pe_c = pei_c - peo_c$$

Nous utilisons aussi un bilan énergétique pour calculer les températures de sortie de l'échangeur :

$$T_3 = T_2 - \epsilon(T_2 - Tei_c),$$

$$Teo_c = \frac{q_{ma} \cdot Cpi_h}{q_{ra} \cdot Cpi_c}(T_2 - T_3) + Tei_c.$$

Les solutions

Un ensemble de 6644 solutions de conception a été calculé en utilisant l'heuristique favorisant les variables de conception. Nous avons cherché à obtenir des systèmes de conditionnement d'air comprenant des échangeurs dont la masse était comprise en 10 et 40 kg. Les deux principaux critères de performance du système de conditionnement d'air (volume et masse de l'échangeur) permettent de visualiser l'ensemble de solutions comme le montre la figure 29.

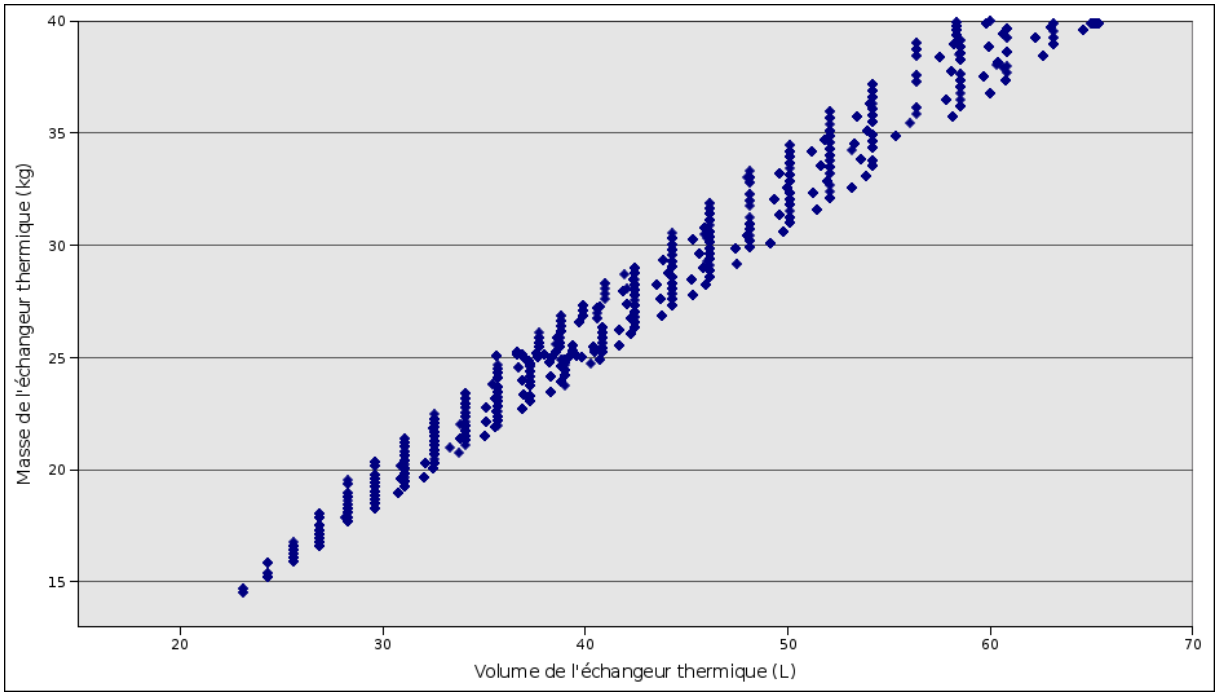


Figure 29 : Variables de décision liées au système de conditionnement d'air : masse et volume de l'échangeur.

Il est intéressant de considérer l'évolution de la température dans le système de conditionnement d'air que nous obtenons. Pour les figures 30 et 31, nous avons représenté les bornes supérieures et inférieures des champs de températures pour le côté flux d'air principal et pour le côté flux d'air dynamique.

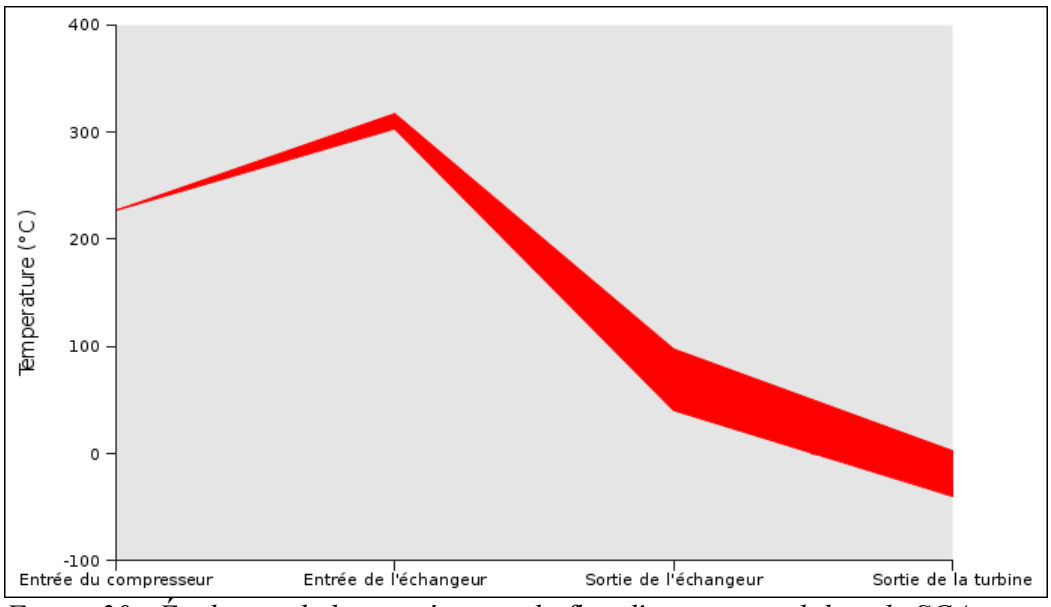


Figure 30 : Évolution de la température du flux d'air principal dans le SCA.

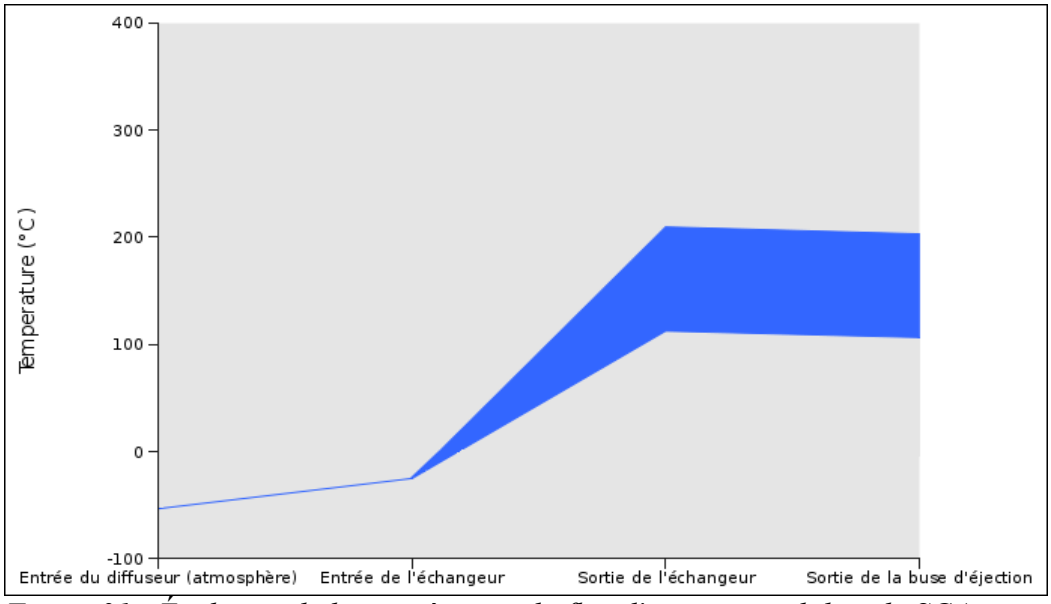


Figure 31 : Évolution de la température du flux d'air principal dans le SCA.

Les températures d'entrée de l'échangeur ne varient presque pas, car leur évaluation dépend des composants standards que nous avons choisis. Ils sont principalement définis par des constantes (leur rendement) et des bilans d'énergie. Les champs de températures obtenus en sortie de l'échangeur et du système représentent alors la variabilité de la température sur l'ensemble des solutions obtenues. Cette vision de l'espace de solutions permet au concepteur de percevoir l'évolution des températures dans le système et ainsi de mieux comprendre l'espace des solutions. Il peut alors choisir une solution qui soit parmi les plus optimales en considérant les deux principaux critères (la masse et le volume de l'échangeur) et en même temps robuste, en vérifiant son positionnement par rapport aux températures au sein du système, en tenant compte de la variabilité des composants dans les phases de conception suivantes.

V.2.4 - Système de conditionnement d'air avec ailettes

Le problème de conception

Nous considérons ici les mêmes contraintes et les mêmes critères que dans le problème précédent, sauf que nous considérons un échangeur composé de plaques et d'ailettes (voir figure 32). L'ajout d'ailettes permet de faire varier les échanges de chaleur en fonction de leur arrangement et de leur dimension. Le matériau utilisé est l'aluminium, ou l'inconel si la

température de l'air est supérieure à 473 K.

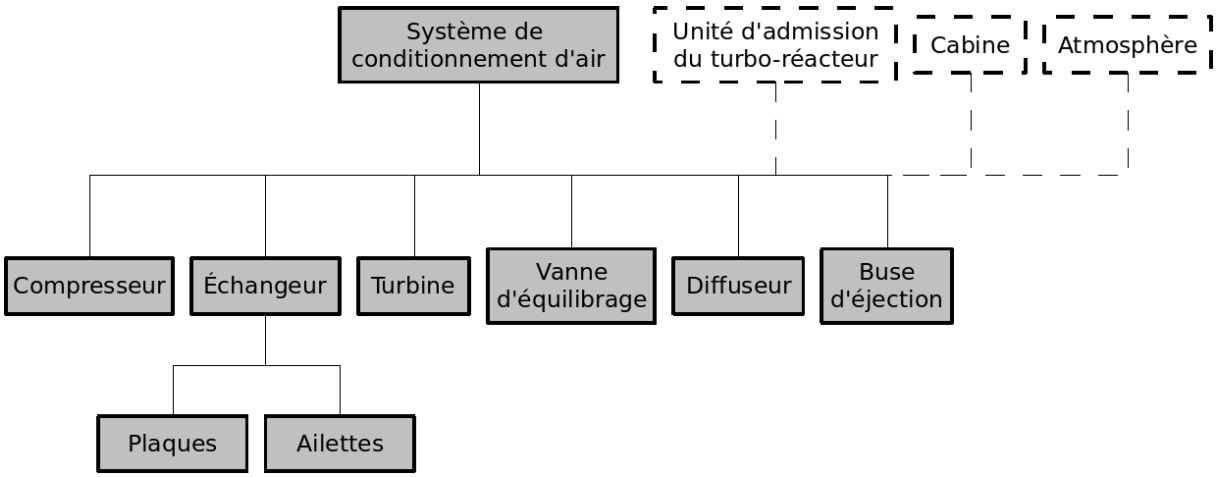


Figure 32 : Organigramme technique étendu du système de conditionnement d'air avec ailettes.

Nous utilisons alors les données présentes dans le livre de Kays et London [Kay84] pour définir les caractéristiques relatives aux ailettes et aux transferts de chaleur. Nous disposons alors de 48 surfaces d'échanges différentes, que nous pouvons regrouper en quatre familles (voir figure 33) : *wavy* (ondulantes), *strip (à pas décalé)*, *louvered (louvoyantes)* et *plain (droites)*. Chaque surface d'échange est caractérisée par un espacement entre ailettes, la taille d'ailette, le positionnement vertical ou incliné des ailettes, etc.

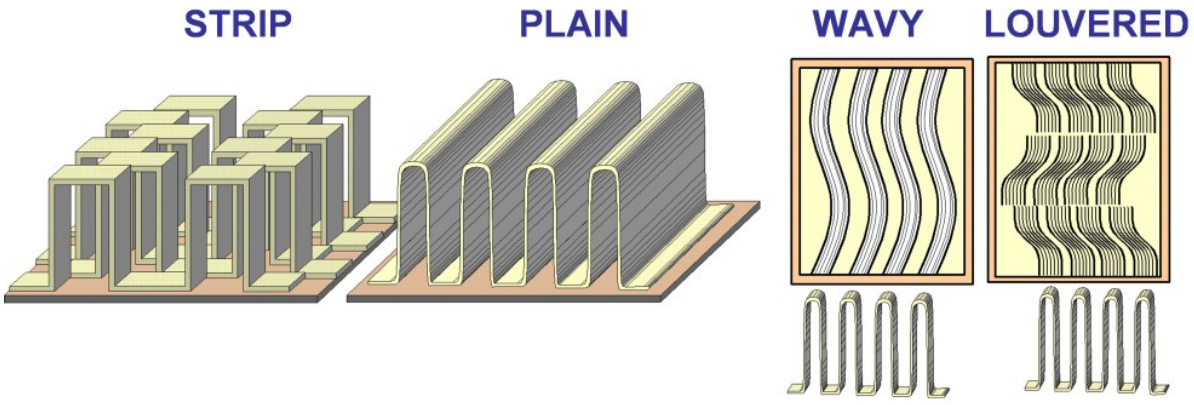


Figure 33 : 4 familles de surfaces d'échanges utilisées dans la conception des échangeurs thermiques compacts [Kay84].

De plus, certains nombres adimensionnels comme le coefficient de Colburn ou le facteur de Fanning sont obtenus à partir de l'interpolation d'une série de valeurs dépendantes du nombre de Reynolds. De même pour les coefficients de pertes de charges en entrée et en sortie de l'échangeur qui sont basés sur des interpolations dépendantes du nombre de

Reynolds et de la porosité relative aux ailettes, les coefficients de l'interpolation sont issus de la base de données de Kays & London.

Table 7 : Variables de conception du SCA avec ailettes.

Variable de conception	Domaine	Précision	Détails
L_x	[0,3..0,8]	0.01	largeur de l'échangeur de chaleur (m)
$exSurf_h$	{1,2,...,48}	-	choix de la surface d'échange
$exSurf_c$	{1,2,...,48}	-	choix de la surface d'échange
tau	[0,8..2]	0.01	rapport des débits (côté froid/côté chaud)
R_v	[0.01..0.2]	0.01	rayon de la vanne (m)

Les solutions

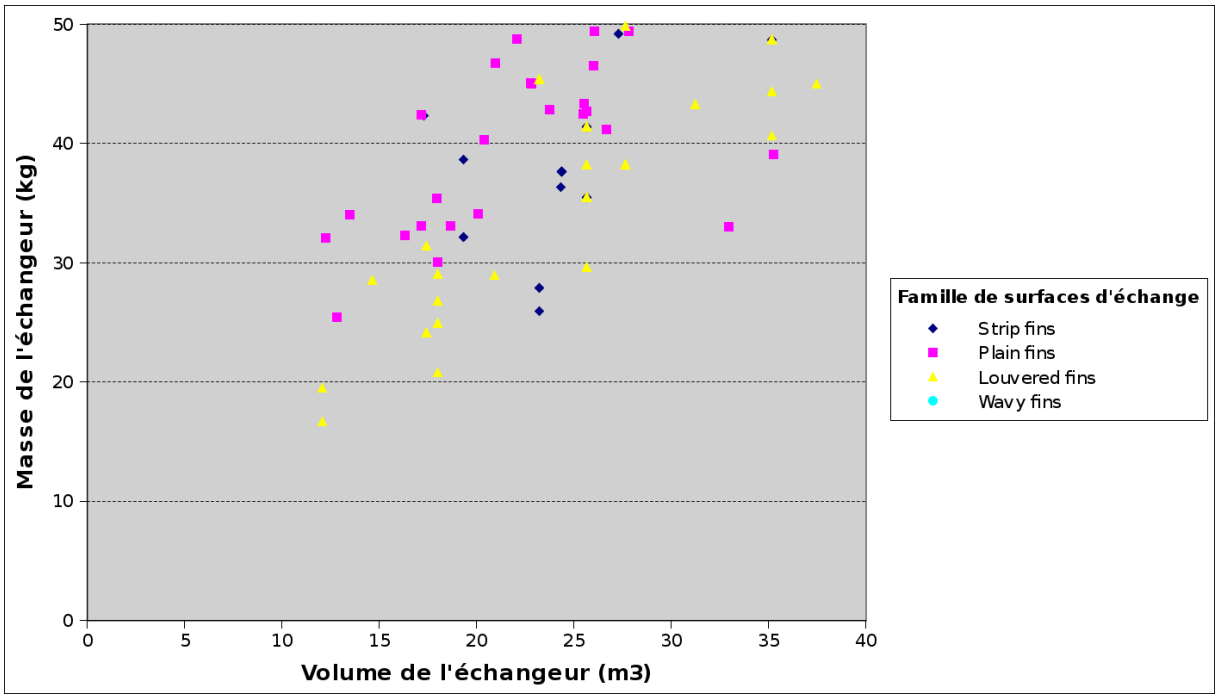


Figure 34 : Solutions du système de conditionnement d'air en considérant la masse en fonction du volume de l'échangeur, ainsi que les familles de surfaces d'échange pour le côté air principal.

Un ensemble de 1254 solutions de conception a été calculé pour ce problème. Cet ensemble est illustré sur la figure 34, qui, comme pour le problème précédent, représente les solutions en se basant sur les deux critères principaux que sont la masse et le volume de

l'échangeur. Il est intéressant de comparer la répartition des solutions dans ces deux derniers problèmes qui ne diffèrent que par la présence ou non d'ailettes entre les plaques de l'échangeur. D'une part, les solutions dans ce problème sont plus éparses à cause de la prise en compte d'un catalogue d'ailettes discrétisant un peu plus le problème. D'autre part, nous obtenons des échangeurs de masse sensiblement identique, mais leur volume est bien inférieur (environ d'un facteur 1/2). En effet, les ailettes permettent de mieux freiner l'air, ce qui augmente significativement les échanges de chaleur entre l'air principal et l'air dynamique.

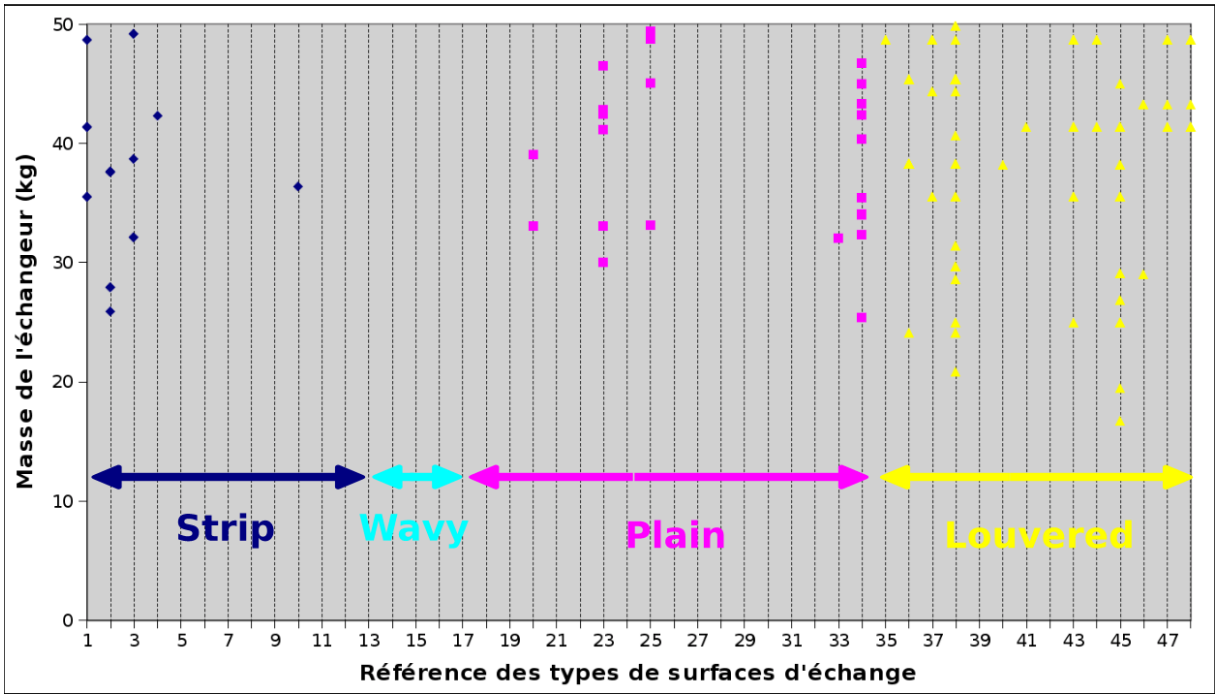


Figure 35 : Répartition des solutions en considérant le type de surface d'échange pour le côté air principal et la masse de l'échangeur.

La figure 34 permet aussi de constater qu'aucune solution n'est basée sur des surfaces d'échange de la famille *wavy* pour le côté air principal. La figure 35 représente quant à elle les solutions en considérant les surfaces d'échanges utilisées pour le côté air principal et la masse des échangeurs obtenus. Cette figure, nous montre encore qu'aucune solution n'utilise une surface d'échange de type *wavy*. Cette absence de solution se justifie par le faible nombre de surfaces d'échange de cette famille recensées dans la base de données de Kays et London. Nous constatons aussi qu'une bonne partie des surfaces d'échange disponibles ne sont pas compatibles avec nos contraintes, puisque toutes ne participent pas à des solutions. Nous voyons aussi plus facilement la famille de surface d'échange qui semble la plus intéressante au regard du critère de la masse, dans notre cas il s'agit de la famille *louvered*.

La figure 36 représente l'ensemble des solutions en considérant cette fois-ci le côté air dynamique de l'échangeur. Le nombre de surfaces d'échange ne permettant pas d'obtenir des solutions est plus petit. D'ailleurs, les quatre familles de surface d'échange permettent d'obtenir des solutions, mais la famille des *louvered* semble encore la plus intéressante concernant la masse de l'échangeur. Ces résultats confirment certaines connaissances que le concepteur peut avoir par rapport à l'importance de l'échangeur dans le système et plus particulièrement le côté air principal.

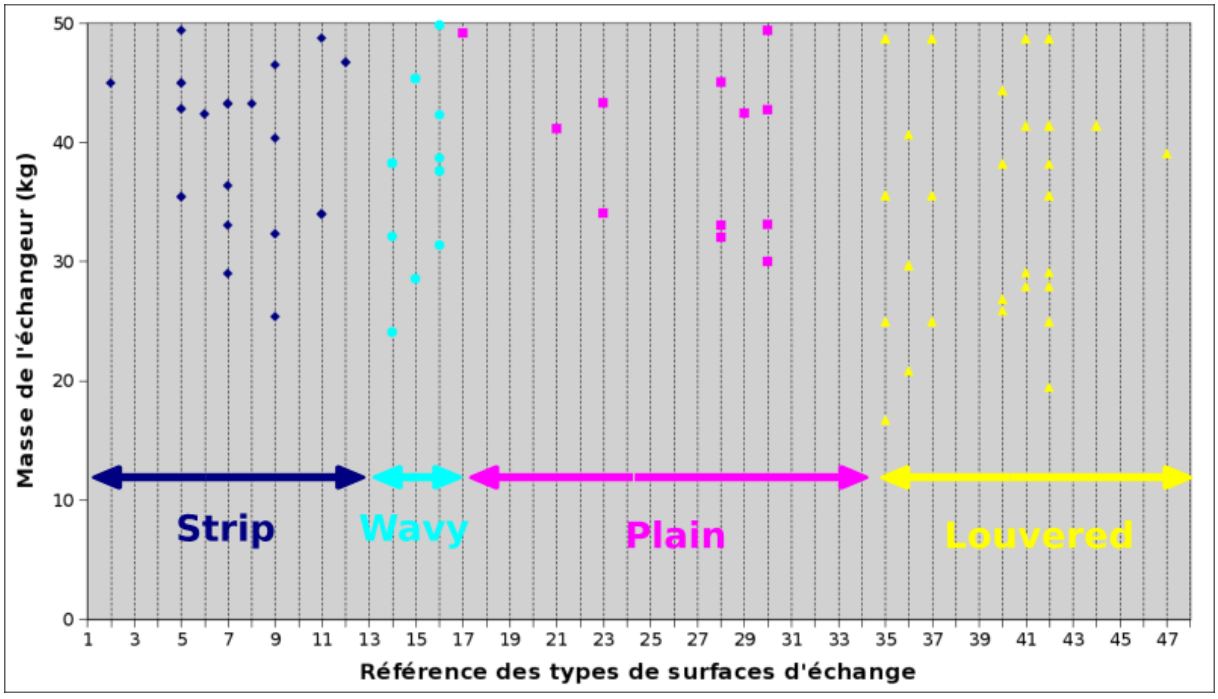


Figure 36 : Répartition des solutions en considérant le type de surface d'échange pour le côté air dynamique et la masse de l'échangeur.

V.3 Analyse des performances de résolution

Dans cette partie, nous présentons les performances de résolution obtenues sur les problèmes traités à l'aide des nouveaux concepts introduits pour utiliser les spécificités des problèmes de conception architecturale. Nous comparons alors les heuristiques de recherche, les problèmes de précision en considérant les temps de calculs et le nombre de découpages total, mais aussi la qualité des solutions pour le concepteur. Nous pouvons noter que le modèle du système de conditionnement d'air prenant en compte des surfaces d'échange

ailetées n'a pu être résolu en un temps raisonnable (plusieurs heures) qu'en utilisant l'heuristique ne calculant que des solutions de conception.

V.3.1 – Gestion de la précision des variables auxiliaires

Nous avons fait des tests par rapport à la gestion de la précision au sein du modèle représentant le circuit d'eau avec une pompe et des réservoirs (voir chapitre IV.1). Les résultats de ces tests sont présentés dans le tableau suivant et ont été obtenus à l'aide de realpaver [Gra06] :

Table 8 : Comparatif des résultats obtenus en fonction de la précision des variables.

$P_{Dr1,Dr2}$	P_Q	P_H	P_e	Nb boîtes	Nb découpages	Temps
10^{-8}	10^{-8}	10^{-8}	10^{-8}	0	$>10^7$	>3600
10^{-8}	10^{-8}	10^{-8}	∞	331	1814	1,2
10^{-3}	10^{-8}	10^{-8}	∞	$>10^7$	$>10^7$	>3600
10^{-3}	10^{-8}	1%	∞	71556	72 711	58,1
10^{-3}	10^{-7}	1%	∞	6259	6743	7,4
10^{-3}	10^{-6}	1%	∞	678	977	1,9
10^{-3}	10^{-5}	1%	∞	86	339	0,2
10^{-3}	10^{-4}	1%	∞	15	244	0,2
10^{-3}	10^{-3}	1%	∞	14	227	0,2
10^{-3}	10^{-2}	1%	∞	119	402	0,6
10^{-3}	10^{-1}	1%	∞	302	618	1,0

La stratégie Round-Robin classique est utilisée pour obtenir ces valeurs. P_x définit la précision utilisée pour une variable x (h représente la hauteur manométrique, alors que e représente l'ensemble des variables alias). Ce modèle comporte trois variables de conception (Pump un entier, D_{r1} et D_{r2}) et seulement deux variables auxiliaires sont définies (Q et H); les autres variables sont toutes considérées comme des alias. Les autres colonnes définissent le nombre total de boîtes solutions calculées, le nombre total de découpages sur les domaines des variables et le temps de calcul en secondes.

La première ligne permet de montrer qu'en utilisant une précision globale de 10^{-8} pour toutes les variables, le processus de résolution est très inefficace. La deuxième ligne montre

que le fait de ne pas découper les variables alias (précision infinie) permet de calculer assez efficacement un ensemble de solutions au problème. La troisième ligne exprime le fait que découper les variables de conception à leur précision pertinente de 10^{-3} est inefficace si la précision des variables auxiliaires n'est pas gérée de manière cohérente. La connaissance du concepteur permet d'associer une précision relative de 1% à la hauteur manométrique H. Les lignes suivantes montrent alors les effets de la variation de la précision du débit dans le circuit Q. Les meilleurs résultats sont obtenus pour des précisions pour Q de 10^{-4} ou 10^{-3}. Ces résultats sont cohérents avec l'ordre de grandeur du domaine de Q et des connaissances du concepteur sur ces valeurs et leur réalité physique.

V.3.2 – Analyse du ratio de l'heuristique sur les variables de conception

Les résultats présents sur la figure 37 sont obtenus en se basant sur le circuit de la pompe et des réservoirs qui a été traité avec realpaver. Nous utilisons alors les précisions optimales définies précédemment (14 solutions, 227 découpages). Seul le nombre de découpages des domaines est ici étudié, car le nombre de solutions ne varie pas. Le temps de résolution est proportionnel au nombre de découpages et ce nombre n'est pas sujet aux variabilités liées à l'ordinateur et au système d'exploitation utilisé.

D'après cette figure, le nombre de découpages est nettement plus petit en considérant les heuristiques de recherche favorisant les VCo. Le meilleur résultat est obtenu pour R=80%, avec moitié moins de découpages sur les domaines des variables.

La figure 38 regroupe les résultats obtenus sur le système de conditionnement d'air sans ailettes résolus au sein d'Elisa. Nous considérons à nouveau le nombre de découpages des domaines suivant le ratio R en favorisant les VCo. La forme de la courbe obtenue montre un comportement similaire à l'exemple précédent. Cependant, le nombre de découpages diminue de manière nettement plus importante, puisqu'au delà de R=70%, le nombre de découpages est de l'ordre de quarante fois inférieur à la stratégie de Round-Robin. Le meilleur résultat est obtenu pour R=80% avec 22255 découpages.

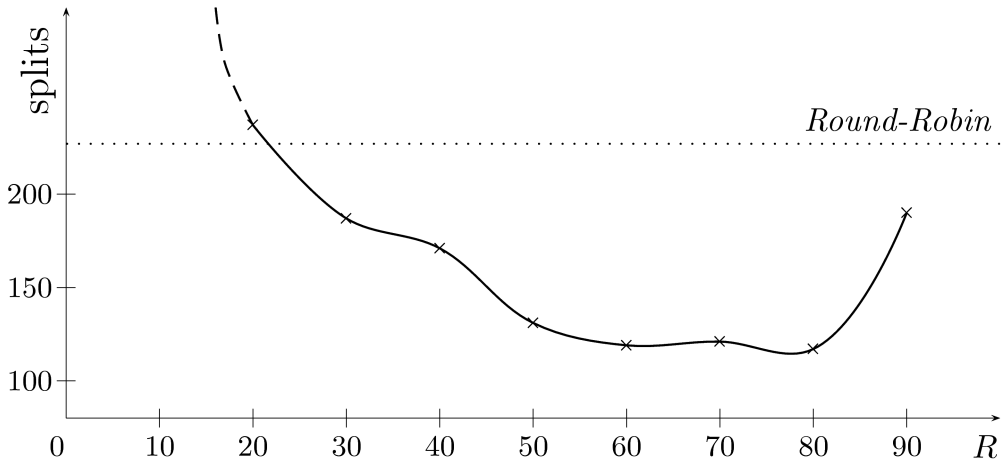
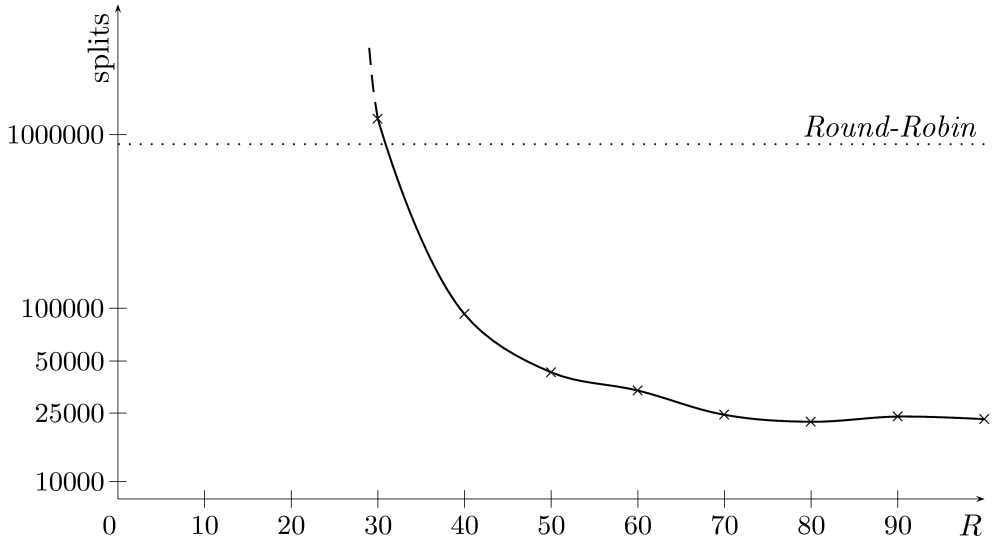


Figure 37 : Circuit de la pompe et des réservoirs : comparaison du nombre de découpages des domaines en fonction du ratio R de l'heuristique de choix des variables favorisant les VCo.



Figur

e 38 : Système de conditionnement d'air : comparaison du nombre de découpages des domaines en fonction du ratio R de l'heuristique de choix des variables favorisant les VCo.

Favoriser les variables de conception pendant la phase d'exploration de l'espace de recherche est important sur les modèles testés. Cette capacité du concepteur à pouvoir classer les variables permet d'améliorer significativement les performances de résolution. L'utilisation d'un ratio R adapté permet d'affiner l'importance à accorder aux variables de conception dans

un modèle où des variables auxiliaires ont été introduites spécifiquement au modèle défini. Cependant, il est difficile de définir ce ratio de manière optimale.

Nous pouvons noter que la stratégie de sélection des variables Round-Robin est similaire à un ratio R_{rr} que l'on peut qualifier de « naturel », correspondant au rapport entre le nombre de variables de conception et le nombre de variables à découper au total (on exclut ici les variables alias puisqu'on leur a défini une précision infinie). Dans le cas du circuit de la pompe et des réservoirs, ce ratio est égal au début de la résolution à 50% et pour le système de conditionnement d'air sans ailette à environ 29%. Le fait de promouvoir les variables de conception change le choix prédéfini qui a lieu au sein de la stratégie Round-Robin (on commence par les VCo), ce qui explique que pour des ratios identiques, les performances ne soient pas les mêmes. Comme les VCo sont importantes par nature dans les modèles de problèmes de conception, le ratio optimal R_{opt} est naturellement plus grand que R_{rr}.

Nous pouvons déduire trivialement que plus le ratio naturel R_{rr} est petit, plus le modèle comporte de variables auxiliaires en comparaison des VCo. Parmi ces variables auxiliaires, la plupart ne sont pas intéressantes à découper fréquemment. Tant qu'une stratégie Round-Robin est appliquée à chaque type de variables et qu'aucune information ne permet d'identifier les variables auxiliaires les plus pertinentes au sein des modèles de conception, il vaut mieux définir un ratio élevé. Nous pourrions utiliser d'autres stratégies, comme celle basée sur la *smear function* ou basée sur l'heuristique d'apprentissage que nous avons définie, mais, ces heuristiques ne semblent pas aussi robustes que la stratégie de Round-Robin (voir IV.4). Autrement dit, il vaut mieux éviter de diversifier la recherche sur les variables auxiliaires et préférer une intensification de la recherche sur les variables de conception. C'est en particulier le cas, dans le cadre de problèmes comportant beaucoup de variables auxiliaires et où la diversification risque de ne pas être très intéressante et efficace pour le processus de résolution.

Nous pouvons aussi noter une différence importante entre l'heuristique Round-Robin et celle favorisant les variables de conception. Le ratio défini pour cette dernière est constant tout au long du processus de résolution tant qu'il reste des domaines de variables à découper parmi les VCo et parmi les VA. Le ratio R_{rr} de la stratégie Round-Robin est constant tant qu'aucune variable n'a atteint sa précision. Une fois cette étape passée, R_{rr} évolue dynamiquement en fonction des VCo et des VA qui atteignent leur seuil de précision.

V.3.3 – Comparaison globale des performances des heuristiques de recherche

Afin de comparer plus facilement les performances offertes par les différentes heuristiques de recherche disponibles pour un concepteur, nous nous sommes basés sur 3 des problèmes présentés dans ce chapitre : l'échangeur du doseur d'apéritif, le circuit de la pompe et le système de conditionnement d'air avec ailettes. Ces modèles comportent des contraintes définissant des catalogues de composants/matériaux (implémentés par des contraintes *piecewise*) et sont composés de variables continues et discrètes. Le problème du circuit d'eau comprend en plus trois contraintes *piecewise* pour décrire le calcul du facteur de Fanning en fonction du nombre de Reynolds (voir III.7.5 et V.2.1). Le catalogue d'ailettes du SCA est ici réduit pour diminuer les temps de calculs, qui sont conséquents pour évaluer les 48 surfaces d'échange disponibles pour chaque côté de l'échangeur (>6 heures), sans pour autant changer le comportement des heuristiques. Les résultats sont obtenus à l'aide d'Elisa et sont regroupés dans le tableau 10. Nous limitons le temps de résolution à 30 minutes. La présence de "0" et "?" indique que le processus de résolution ne s'est pas achevé durant les 30 minutes de calculs. Les heuristiques de recherche utilisées sont présentées dans le tableau 9.

Table 9 : Abréviations et descriptions des heuristiques de recherche utilisées.

Abréviation	Description
RR	C'est l'heuristique Round-Robin qui applique une stratégie de tourniquet sur l'ensemble de variables
LS	C'est la stratégie d'apprentissage de la sensibilité des variables.
VF	C'est la stratégie VCoFirst calculant les solutions pour les variables de conception avant d'explorer les variables auxiliaires. Par défaut au sein de ces deux ensembles de variables, la stratégie RR est appliquée.
VF(LS)	C'est la stratégie VF pour laquelle LS est appliquée à la place de RR sur l'ensemble des VCO et sur l'ensemble des VA.
VO	C'est la stratégie VCoOnly qui ne calcule que des solutions de conception. Elle est basée sur VF et une stratégie de *backtrack* adaptée qui évite d'explorer des solutions de conception similaires.
VO(LS)	C'est la stratégie VO basée non plus sur VF, mais sur VF(LS).
+B	On tient compte des bornes des intervalles définis dans une contrainte *piecewise*. Le processus d'exploration commence alors par les variables qui paramètrent les contraintes *piecewise* en les découpant sur ces bornes.

D'après les résultats présentés dans le tableau 10, la stratégie RR est la moins efficace

lorsque les bornes des contraintes *piecewise* ne sont pas prises en compte. Les choix dans les catalogues sont faits tardivement, alors qu'ils restreignent fortement l'espace de recherche et un grand nombre de calculs sont faits inutilement.

Table 10 : Comparatifs des heuristiques de recherche sur des problèmes de conception.

Problème	**Stratégies**	Solutions	Découpages	Temps (s)
Doseur	RR	895	85420	35,35
	LS	1029	7345	11,01
	VF	1481	1032	1,11
	VF(LS)	898	1301	1,05
	RR+B	962	1545	1,15
	LS+B	960	1911	1,4
	VF+B	1056	1308	0,86
	VF(LS)+B	917	1076	0,84
	VO+B	186	377	0,26
	VO(LS)+B	125	279	0,2
Pompe	RR	252	5722	23,69
	LS	229	1080	5,92
	VF	244	1415	12,77
	VF(LS)	238	2435	13,9
	RR+B	220	707	4,7
	LS+B	236	805	1,07
	VF+B	233	1601	12,95
	VF(LS)+B	241	1471	12,97
	VO+B	35	1112	9,67
	VO(LS)+B	23	1093	10,19
SCA	RR	0	?	>1800
	LS	0	?	>1800
	VF	3957	14694	447,42
	VF(LS)	3772	8054	436,72
	RR+B	0	?	>1800
	LS+B	3588	6839	380,93
	VF+B	4296	7760	422,23
	VF(LS)+B	5067	9717	544,72
	VO+B	10	524	21,74
	VO(LS)+B	7	777	41,03

La stratégie LS et la stratégie VF permettent d'améliorer les temps de calculs et le nombre de découpages, mais pas nécessairement le nombre de solutions calculées. Les références de choix de composants dans les catalogues sont des VCo, ce qui explique les bonnes performances de VF, puisqu'elles sont valuées avant d'explorer les VA. Nous pouvons tout de même noter que pour le problème du SCA, LS comme RR n'ont pas permis d'obtenir la moindre solution en 30 minutes de calculs.

La prise en compte des bornes des contraintes *piecewise* permet globalement d'améliorer toutes les stratégies. C'est pour RR que l'impact est le plus important, ce qui s'explique par le choix dans les catalogues qui sont effectués avant de parcourir le reste des domaines des variables. Pour VF, les gains sont moins importants, puisque ces choix sont déjà favorisés sans la prise en compte des bornes (les choix dans les catalogues sont des VCo).

Les variations du nombre de solutions entre les heuristiques s'expliquent principalement par l'exploration différente des domaines des variables, qui ne sont ensuite pas réduits de la même manière. Il s'agit cependant du même espace de solutions. Il est simplement découpé différemment.

La stratégie VO ne calcule que les solutions de conception des problèmes en se basant sur VF, alors que VO(LS) se base sur VF(LS). Cette dernière heuristique est plus performante que VO et la réduction du nombre de solutions et de découpages s'expliquent par le fait que LS est plus performante que RR sur ces problèmes. Le choix des variables par LS permet d'atteindre plus rapidement leur seuil de précision pour trouver une première solution validant l'architecture produit. Cependant, lorsque toutes les solutions numériques sont calculées, l'application de LS à la place de RR n'est pas toujours bénéfique (avec ou sans prise en compte des bornes). Nous expliquons cela par la difficulté de LS à maintenir un ensemble de coefficients pertinents pour évaluer la sensibilité des variables, en particulier lorsque l'on vient de trouver une solution et qu'on effectue des retours en arrière importants dans l'arbre de recherche.

Les stratégies VO et VO(LS) permettent d'obtenir les meilleurs résultats pour le concepteur. Il obtient ainsi l'ensemble des solutions de conception, même si nous avons précédemment vu que quelques quasi-solutions sont tout de même calculées. Le concepteur peut manipuler plus facilement l'espace des solutions et choisir parmi elles l'architecture à

développer en phase de conception détaillée.

En comparaison, les autres heuristiques calculent un nombre de solutions bien plus important, à cause des découpages sur les VA (la plupart inutiles). Ces découpages provoquent une duplication des solutions de conception, car pour une même architecture plusieurs solutions sont calculées. Ces découpages sont nécessaires pour atteindre les précisions définies par le concepteur.

Pour le concepteur, la qualité des solutions obtenues et le temps de résolution sont des critères importants lors de l'utilisation d'un outil de traitement. Le concepteur évalue la qualité de l'ensemble de solutions obtenues, en considérant la complétude de l'ensemble calculé et en considérant la quantité de solutions de conception dupliquées. Sur les trois problèmes testés, VO est plus intéressante du point de vue de la qualité des solutions sont plus faibles que les autres heuristiques, sauf pour le problème de la pompe. VO est plus performante, car tout l'espace de recherche offert par les domaines des VA n'est pas exploré. Le petit nombre de VA dans le modèle de la pompe (seulement deux) est peu pénalisant et RR+B s'avère ici plus efficace que VO. Sur des problèmes de conception intégrant plus de variables auxiliaires (problèmes industriels) comme le problème du SCA, les performances de RR ou RR+B ne sont pas suffisantes pour permettre une résolution efficace.

V.4 – Bilan

Dans ce chapitre, nous avons présenté quatre problèmes de conception. Deux problèmes plus simples ont permis d'affiner les algorithmes et de faire de nombreux tests, alors que les deux derniers problèmes sont plus difficiles à résoudre. Pour le dernier problème, plusieurs heures de calculs sont nécessaires pour obtenir toutes les solutions de conception. Les résultats sont tout de même entachés de quasi-solutions qui sont liées à des problèmes de précisions et de découpages des variables. Cependant, la stratégie de résolution VO ne calcule que des solutions au sens de la conception architecturale; elle permet de diminuer le nombre de solutions numériques calculées, ce qui facilite leur exploitation par le concepteur.

Nous avons aussi analysé les stratégies proposées en regardant les performances de résolution obtenues sur les problèmes testés. La stratégie d'apprentissage de la sensibilité

numérique semble prometteuse et permet parfois de diminuer le nombre de solutions calculées. La stratégie favorisant les variables de conception permet généralement d'améliorer les performances de résolution de manière significative, en particulier pour des modèles comportant beaucoup de variables auxiliaires. Il ressort cependant que le ratio R de cette heuristique est difficile à fixer pour un problème donné. Pour des problèmes durs à résoudre et comportant un grand nombre de solutions, il est plus pertinent de ne calculer que les solutions de conception. Aucun test n'a été fait sans la gestion de la contrainte globale *piecewise*, mais, sans elle, les lois de comportement physique ou le choix de composants s'avèrent souvent impossibles à exprimer. La prise en compte des bornes des intervalles définis dans cette contrainte globale permet d'améliorer significativement les performances de résolution, en particulier lorsque nous ne favorisons pas les variables de conception.

CHAPITRE VI

CONCLUSION

Nous avons vu qu'un solveur de CSP constitue un outil à priori plus souple que les autres outils utilisés en conception préliminaire (tableurs ou outils de simulation numérique), en particulier lors de modifications fréquentes concernant les caractéristiques du produit ou de ses composants. Ce type d'outil garantit en plus la présence de zones de non faisabilité, ce qui permet de définir une approximation de l'ensemble complet des solutions, et non un ensemble de solutions obtenues de manière stochastique. Cependant, l'utilisation des solveurs de CSP pendant la phase de conception architecturale est encore difficile dans un contexte industriel, car les solveurs ont encore des défauts qui nuisent à leur utilisation. L'expression d'une connaissance au sein d'un CSP bien défini et adapté à une résolution efficace reste un problème ouvert. Les algorithmes de résolution utilisés pour les CSP numériques ne sont pas encore efficaces sur des problèmes de conception complexes. Les variables sont définies sur des domaines discrets ou continus. Le traitement des problèmes mixtes est encore difficile et peu de solveurs sont efficaces sur ce type de problèmes.

De plus, le concepteur n'est pas familiarisé avec ce type d'outil qui ne permet pas d'exprimer certains concepts familiers et reste limité par le formalisme utilisé. Nous avons ainsi montré l'intérêt de définir des variables de conception, ainsi que des alias.

VI.1 - Contribution apportée

Nous avons étudié les différentes phases d'élaboration d'un modèle en conception architecturale. Cette analyse nous a permis de mettre en évidence certaines spécificités liées à ces modèles, comme, par exemple, la hiérarchisation des variables et leur importance pour le concepteur. En effet, les variables de conception architecturale permettent d'identifier l'architecture d'un produit. Nous avons alors défini plusieurs heuristiques de recherche favorisant l'exploration des variables de conception au détriment des variables auxiliaires afin de calculer un ensemble de solutions le plus proche possible de l'ensemble des solutions de conception. Ces heuristiques ont permis de réduire significativement les temps de résolution des modèles traités. Les variables de conception font partie des variables les plus sensibles numériquement dans le modèle. Cette caractéristique des variables de conception s'explique par leurs propriétés qui découlent des méthodes qui ont permis de les identifier en conception préliminaire. En effet, le modèle du produit en conception architecturale n'a été établi que pour définir des solutions pour ces variables. Le concepteur n'a introduit des variables

auxiliaires que pour relier les variables de conception aux performances du produit (variables de décision), qui lui permettent d'évaluer les critères définis dans le cahier des charges. Ainsi, une intensification de la recherche sur les variables de conception permet de converger plus rapidement vers les solutions du modèle.

De plus, nous avons mis en évidence les ambiguïtés présentes lors de la définition des précisions associées aux variables. Ces précisions sont fondamentales pour le solveur, car elles définissent la taille de l'espace de recherche et le seuil à atteindre pour chaque domaine pour considérer une solution. Cependant, le concepteur ne peut pas définir toutes ces précisions avec pertinence, car il ne maîtrise pas toutes les variables auxiliaires présentes dans un modèle. Des précisions mal définies risquent de nuire à l'efficacité du processus de résolution, mais elles risquent aussi de nuire à la qualité des solutions obtenues. En effet, la phase de propagation des contraintes calcule l'ensemble des domaines des variables de manière cohérente. Ainsi, si une seule variable est associée à une précision trop importante, son domaine est découpé jusqu'à ce seuil et les domaines des autres variables sont réduits en conséquence. Dans le cadre de la conception architecturale, cela rend difficile l'exploitation des solutions par le concepteur, les calculs sur les variables auxiliaires réduisant trop les domaines des variables de conception par propagation. Cela augmente aussi parfois le nombre des solutions, lorsque plusieurs solutions sont obtenues pour les variables auxiliaires et correspondent à une même combinaison de valeurs pour les variables de conception.

Pour pouvoir prendre en compte certains phénomènes physiques, nous avons défini une contrainte globale implémentant une contrainte définie par morceaux. En effet, certains phénomènes physiques définis de manière expérimentale sont décrits par des lois de comportement spécifiques à un contexte d'exploitation. Ce contexte est souvent exprimé par les valeurs d'une variable représentative du phénomène. Ainsi, nous avons défini une contrainte globale qui comporte plusieurs contraintes, chacune étant associée à un intervalle faisant partie du domaine initial de la variable paramétrant le phénomène. Dès que le domaine de cette variable est inclus dans un des intervalles définis, alors la ou les contraintes correspondantes sont appliquées. Nous avons utilisé les bornes de ces intervalles pour explorer le domaine de la variable correspondante, à la place de la bissection utilisée sur les autres domaines des variables. En effet, cela permet d'obtenir plus rapidement les intervalles souhaités pour l'activation des contraintes.

En dehors des implémentations que nous avons réalisées au sein d'un solveur CSP,

nous avons mis en évidence d'autres concepts importants en conception architecturale. Leur prise en compte permettrait d'améliorer l'utilisation de ces solveurs. Par exemple, la prise en compte de la notion de situation de vie permettrait au concepteur de concevoir mieux et plus rapidement des produits.

Les concepts introduits en programmation par contraintes ont été testés sur des modèles de conception architecturale de plusieurs produits. Parmi eux, deux systèmes plus basiques ont servi à faire les premiers tests, alors que les deux autres modèles testés ont vraiment permis de montrer l'avantage de nos stratégies et de nos algorithmes. Par la suite, notre approche devra être validée sur un plus grand nombre plus de problèmes de conception, mais cela nécessite un investissement très conséquent en temps.

VI.2 - Perspectives et futures évolutions

Les solveurs basés sur les CSP et qui utilisent l'analyse par intervalles sont les seuls outils numériques disponibles qui permettent d'explorer des espaces de solutions avec des garanties mathématiques sur la non présence de solutions. Les mécanismes de consistance et de propagation de contraintes permettent de calculer une approximation extérieure de l'espace de solutions. A l'aide de l'arithmétique des intervalles, les calculs effectués garantissent la conservation des solutions dans l'enveloppe calculée, bien que celle-ci puisse être une approximation grossière de l'espace final des solutions. Le concepteur a cependant la garantie que les zones de l'espace de recherche éliminées ne comportent pas de solution. Ces informations sont souvent importantes pour le concepteur qui peut tout de suite prendre conscience de la zone de faisabilité de son problème. Ces méthodes ont ainsi un avantage décisif en conception par rapport à d'autres méthodes d'exploration globale plus classiques.

Aujourd'hui, les problèmes rencontrés dans l'utilisation des CSP en conception sont liés à l'efficacité du traitement numérique du point de vue temps de calcul, mais aussi du point de vue des interactions avec l'utilisateur. Ce dernier n'a aucun contrôle pendant le processus de résolution. Nous avons défini une heuristique permettant d'améliorer le contrôle sur les variables de conception et sur les variables auxiliaires, mais nous n'avons pas encore exploité les propriétés des variables de décision. Ces dernières sont fondamentales pour le concepteur lorsqu'il considère une solution ou un ensemble de solutions. C'est à l'aide de ces critères qu'il détermine si une zone de l'espace de recherche est intéressante.

Nous pouvons alors définir plusieurs perspectives qui font suite à ce travail :

1. Les méthodes d'exploration des VA ne sont pas forcément adaptées à leur signification dans les modèles de conception. Pour ces variables, on cherche à converger rapidement vers des valeurs consistantes et non à élaguer au fur et à mesure leur domaine. Nous pensons améliorer l'heuristique d'apprentissage de la sensibilité numérique des variables que nous avons défini. Une autre évolution de ce travail de thèse est l'utilisation d'une approche évolutionnaire pour explorer les VA. L'utilisation d'un algorithme génétique permet de converger vers des valeurs consistantes dans un espace de recherche souvent très vaste. Les algorithmes de recherche utilisés dans les solveurs CSP sont parfois inefficaces, car ils doivent découper et réduire certaines variables jusqu'à des précisions importantes, nécessitant alors un très grand nombre d'itérations.
2. Par ailleurs, les espaces de solutions fournis par les solveurs ne sont pas suffisamment adaptés aux besoins des utilisateurs en conception. Le contrôle du processus de résolution et donc l'interaction entre le concepteur et le solveur permettraient d'améliorer à la fois la résolution et l'exploitation de l'espace des solutions. Ce travail d'évolution des stratégies de résolution par des approches de type CSP pour la conception interactive existe déjà, mais se base principalement sur les choix de l'utilisateur pour explorer l'espace de recherche [Gel96, Far04]. Nous pensons que des heuristiques de recherche peuvent être combinées avec les choix de l'utilisateur en se basant sur l'importance des variables de décision pour le concepteur. La connaissance du concepteur liée à l'exploration des domaines et des valeurs des variables de décision peut être utilisée pour ajuster dynamiquement le processus de résolution à ses attentes. Le traitement d'un problème de conception est intrinsèquement un processus dynamique même dans ses phases numériques. Toutes les connaissances liées aux problèmes de conception ne sont pas formalisables en amont de la résolution numérique du problème. Cette inadéquation entre le besoin et le traitement numérique est à l'origine de calculs inutiles qui dégradent les performances du solveur et induisent des difficultés dans l'exploitation et l'interprétation des résultats.
3. La robustesse des solutions obtenues doit aussi être améliorée. Le concepteur obtient des solutions pour lesquelles une seule situation de vie du produit a été satisfaite. Il faut alors permettre au concepteur de définir la robustesse des solutions qu'il a obtenues par rapport à des critères supplémentaires en relation avec les restrictions

définies par d'autres situations de vie. De plus, nous n'avons aucune certitude quant à la validité des solutions obtenues du fait des évolutions du problème de conception, certains composants pouvant changer dans les phases de conception suivantes. Le concepteur ne connaît pas la marge d'erreur dont il dispose, quant à la faisabilité du produit ou quant à la satisfaction des critères de performance. Autrement dit, il n'a pas d'indication sur le rapport de sensibilité existant entre les variables structurantes d'un produit et ses performances.

4. La notion de robustesse est aussi à prendre en compte d'un point de vue numérique. En effet, les techniques de résolution par intervalles permettent de calculer des approximations extérieures de solutions ou d'ensemble de solutions, mais, pour des problèmes hétérogènes comme ceux définis en conception, il est difficile de prouver leur existence dans une boîte d'intervalles [Mic01, Kea02, Kea03]. Lorsque des continuums de solutions sont calculés nous pouvons alors effectuer des approximations intérieures pour définir des solutions garanties [Vu02]. Le calcul d'approximation intérieure est aussi nécessaire lorsque des quantificateurs universels sont définis pour certaines variables [Ben99b, Gol06]. Les variables quantifiées ne sont pas encore présentes dans les modèles de conception que nous traitons. Pour le dimensionnement de certains produits, les contraintes liées à la géométrie et aux tolérances sur les dimensions sont déterminantes par rapport aux performances du produit. Ces tolérances s'expriment alors principalement comme des quantificateurs universels sur des dimensions.
5. L'évolution du système étudié et de l'environnement d'utilisation du solveur peuvent aussi être envisagés afin d'appliquer cette approche dans des contextes plus réalistes d'un point de vue conception. L'étude d'un SCA prenant en compte le changement de phase avec condensation dans les échangeurs de chaleur permettrait d'obtenir des solutions plus proches des études menées dans un cadre industriel. Cela implique aussi de tenir compte des phénomènes de givrage et de filtration de l'eau présente dans l'air. Les modèles de ces phénomènes ont déjà été établis au sein d'autres outils numériques dans des travaux récents au sein de l'équipe Énergétique et conception du TREFLE, mais ils n'ont pas été intégrés dans un modèle CSP à cause des difficultés de résolution actuellement rencontrées.

L'équipe Énergétique et conception du TREFLE développe, dans le cadre d'une ERT (Équipe de Recherche Technologique) IMPACT, des travaux autour du développement de concepts de turbo-moteurs innovants. Ces travaux sont confrontés à des difficultés d'exploration d'espaces de solutions liées à des modèles complexes de turbines à haute pression. Les contraintes à prendre en compte amènent de nouveaux besoins de modélisation et de résolution, liés par exemple au tolérancement géométrique et à des modèles thermo-mécaniques d'ordre réduit.

Dans l'équipe MEO du Lina, plusieurs types de travaux en liens avec la conception sont actuellement en cours :

- R. Soto travaille sur les langages de modélisation pour les CSP et en particulier des langages CSP objets et leur résolution dans des solveurs CSP.
- T. Douillard travaille sur le traitement de formules logiques prenant en compte des opérateurs disjonctifs.
- N. Berger travaille sur la résolution efficace de problèmes mixtes. Une approche possible est alors de faire coopérer les solveurs traitant les domaines discrets et ceux traitant les domaines continus, tout en définissant les traitements possibles pour les contraintes mixtes.

Le développement de méthodes et d'outils de résolution robustes pour le traitement de problèmes numériques complexes en conception est une perspective globale à ce travail. L'amélioration des performances de résolution et la prise en compte de la connaissance et des besoins du concepteur sont alors fortement liées.

RÉFÉRENCES BIBLIOGRAPHIQUES

[Ami02] AMILHASTRE J., FARGIER H. et MARQUIS P., "Consistency restoration and explanations in dynamic CSPs - application to configuration". Artificial Intelligence, 135, 199-234, 2002.

[Ant 01] ANTONSSON E.K., "Imprecision in Engineering Design", Engineering Design Research Laboratory, California Institute of Technology, Pasadena, 2001.

[Apt99] APT K. R., "The Essence of Constraint Propagation". Theoretical Computer Science, 221(1-2) :179–210, 1999.

[Apt03] APT K. R., "Principles of Constraint Programming". Cambridge University Press, 2003.

[Arb06] Arbaoui A., "Aide à la décision pour la définition d'un système éolien : adéquation au site et à un réseau faible". Thèse de doctorat, Ensam, Bordeaux. 2006.

[Bah95] BAHLER D., DUPONT C. et BOWEN J., "Mixed quantitative/qualitative method for evaluating compromise solutions to conflicts in collaborative design". Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 9(4):325–336, 1995.

[Ben94] BENHAMOU F., MCALLESTER D. et VAN HENTENRYCK P., "CLP(Intervals) Revisited". In International Symposium on Logic Programming, pages 124–138, Ithaca, NY, USA, 1994. MIT Press.

[Ben95] BENHAMOU F., "Interval constraint logic programming". In Andreas Podelski, editor, Constraint Programming : Basics and Trends, volume 910 of Lecture Notes in Computer Science, pages 1–21. Springer-Verlag, 1995.

[Ben97] BENHAMOU F. et OLDER W.J., "Applying Interval Arithmetic to Real", Integer and Boolean Constraints. Journal of Logic Programming, 32(1) :1–24, 1997.

[Ben99a] BENHAMOU F., GOUALARD F., GRANVILLIERS L. et PUGET J.-F., "Revising Hull and Box Consistency". In D. De Schreye, editor, Proceedings of International Conference on Logic Programming, pages 230–244, Las Cruces, USA, 1999. The MIT Press.

[Ben99b] BENHAMOU F., GOUALARD F., LANGUENOU E. et CHRISTIE M., "An Algorithm to Compute Inner approximations of Relations for Interval Constraints". In proceedings of the third Int. Ershov memorial Conference on Perspectives of system informatics (PSI'99), LNCS 1755, Springer, ISBN 3-540-67102-1, 1999.

[Ben00] BENSANA E., MULYANTO T. et VERFAILLIE G., "Dealing with Uncertainty in Design and Configuration Problems". Dans Workshop on Configuration 14th European Conference on Artificial Intelligence, Berlin, Allemagne, Août 2000.

[Ben01] BENSANA E. "A constraint based environment to solve configuration problems : application to haleuav design". In CEAS Conference on Multidisciplinary Aircraft Design and Optimisation, Cologne (Germany), June 2001.

[Ben04] BENNACEUR H., "A comparison between CSP and SAT techniques", CONSTRAINT Journal, 9 (2) : 123-138, 2004.

[Ben06] BENHAMOU F. et GRANVILLIERS L., “Continuous and Interval Constraints". In Rossi F., Van Beek P. et Walsh T., editors, Handbook of constraint programming, chapitre 16. Pages 571-603, Elsevier, 2006.

[Ber88] BERLINER C. et BRIMSON J.A., "Cost management for today's advanced manufacturing : The CAM-I conceptual design", Harvard Business School Press, 1988.

[Bli98] BLIEK C., NEVEU B. et TROMBETTONI G., "Using Graph Decomposition for Solving Continuous CSPs". In: CP'98. 1998. Pisa, Italy.

[Blu03] BLUM C. et ROLI A., "Metaheuristics in combinatorial optimization : Overview and conceptual comparison", ACM Computing Surveys, 35(3), pages 268-308, 2003.

[Bow93] BOWEN J. et BAHLER D., "Constraint-based software for concurrent engineering". IEEE Computer, 26(1):66–68, 1993.

[Cav95] CAVAILLES J., "Méthodes de management de programme", 2ème édition, DGA- Teknea , 1995.

[Cha92] CHAKRABARTI A., BLIGH T.P. et HOLDEN T., "Towards a decision-support framework for the embodiment phase of mechanical design", Artificial Intelligence in Engineering, n°7, 1992.

[Chi97] CHIRIAEV D. et WALSTER G.W., "Interval arithmetic specifications". Manuscript J3/97-199 for ANSI X3J3, juillet 1997.

[Cle87] CLEARY J.G., "Logical Arithmetic". Future Computing Systems, 2(2) :125–149, 1987.

[Dar94] DARR T.P. et BIRMINGHAM W.P., "Automated design for concurrent engineering". IEEE Expert, Intelligent Systems & their Applications, 9(5):35–42, 1994.

[Deb01] DEBRUYNE R. et BESSIERE C., "Domain Filtering Consistencies", in Journal of Artificial Intelligence Research, Vol. 14, p. 205-230, 2001.

[Din90] DINCBAS M., SIMONIS H., VAN HENTENRYCK P., "Solving Large Combinatorial Problems in Logic Programming". J. Log. Program. 8(1):75-93 (1990).

[Dre98] DREYFUS G., "les Réseaux de neurones", Mécanique Industrielle et Matériaux, n°51 ,1998.

[Dre04] DREYFUS G., MARTINEZ J.-M., SAMUELIDES M., GORDON M.B., BADRAN F., THIRIA S. et HERAULT L., "Réseaux de neurones, Méthodologie et applications". Eyrolles, 2ème édition (2004).

[Far96] FARGIER H., LANG J. et SCHIEX T., "Mixed Constraint Satisfaction : A Framework for Decision Problems under Incomplete Knowledge". In AAAI-96, pages 175–180. AAAI Press, 1996.

[Far02] FARGIER H. et HENOCQUE L., "Configuration à base de contraintes". Technical Report 0010.2002, Laboratoire des Sciences de l'Information et des Systèmes, LSIS, (UMR CNRS 6168), september 2002.

[Far04] FARGIER H., VILAREM M.-C., "Compiling CSPs into tree-driven automata for interactive Solving". Constraint. 9:263-287, 2004.

[Fis00] FISCHER X., "Stratégie de conduite du calcul pour l'aide à la décision en conception mécanique intégrée; application aux appareils à pression", Thèse de doctorat, ENSAM, décembre 2000.

[Fis02] FISCHER X., SEBASTIAN P., NADEAU J.P. et JOYOT P., "Decision support in integrated mechanical design through qualitative constraints". In Integrated Design and Manufacturing in Mechanical Engineering, Chedmail P., Cognet G., Fortin C., Mascle C., Pegna J. (eds), Kluwer, 35-42, 2002.

[Fis04] FISCHER X., SEBASTIAN P., NADEAU J.-P. et ZIMMER L., "Constraint based Approach Combined

with Metamodeling Techniques to Support Embodiment Design". In: SCI'04. 2004. Orlando, USA.

[Fre78] FREUDER E.C., "Synthesizing Constraint Expressions". Communications of the ACM, 21(11) :958–966, 1978.

[Fre82] FREUDER E.C. , "A sufficient condition for backtrack-free search". Journal of the ACM, 29(1):24–32, 1982.

[Gar85] GARLOFF J., "Convergent bounds for the range of multivariate polynomials". In K. NICKEL, réd., Interval Mathematics, pages 37–56. Springer Verlag, 1985.

[Gel96] GELLE E. et WEIGEL R. "Interactive configuration using constraint satisfaction techniques". In Artificial Intelligence and Manufacturing Research Planning Workshop, AAAI Technical Report FS-96-03, pp 37-44, 1996.

[Gel03] GELLE E. et FALTINGS B., "Solving Mixed and Conditional Constraint Satisfaction Problems". Constraints 2003; 8:107–141.

[Gol06] GOLDSZTEJN A. et JAULIN L., "Inner and outer approximations of existentially quantified equality constraints". In Poceedings of CP2006, LNCS 4204, pages 198-212, 2006.

[Gou00] GOUALARD F., "Langages et environnements en programmation par contraintes d'intervalles". Thèse de doctorat, Institut de Recherche en Informatique de Nantes, Juillet 2000.

[Gou07] GOUALARD F. et JERMANN C., "A Reinforcement Learning Approach to Interval Constraint Propagation". Accepté pour publication dans CONSTRAINTS, 2007.

[Gra99] GRANVILLIERS L., GOUALARD F. et BENHAMOU F., "Box Consistency through Weak Box Consistency". Proceedings of the eleventh IEEE International Conference on Tools with Artificial Intelligence. IEEE Computer Society. pages 373—380. 1999.

[Gra06] GRANVILLIERS L. et BENHAMOU F., "Algorithm 852: Realpaver: An interval solver using constraint satisfaction techniques". ACM TOMS 2006; 32(1):138–156.

[Had02] HADJ-HAMOU K. "Contribution à la conception de produits à forte diversité et de leur chaîne logistique : une approche par contraintes". Thèse de doctorat, Institut National Polytechnique de Toulouse, France, 2002.

[Hao05] HAO F. et MERLET J.-P., "Multi-criteria optimal design of parallel manipulators based on interval analysis". Mechanism and Machine Theory, 40(2):151--171, 2005.

[Hir01] HIRTZ J., STONE R., MCADAMS D., SZYKMAN S. et WOOD K., "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts", Research in Engineering Design, 2001, 13(2):65-82.

[HoK06] HO KON TIAT V., "Aide à la décision pour la conception préliminaire de procédés d'évaporation flash". Thèse de doctorat. Université de Bordeaux I. 2006.

[Hor89] HORNIK K., STINCHCOMBE M. et WHITE H., "Multilayer feedforward network are universal approximates", Neural Networks, volume 2 : 359-366, 1989.

[Hor94] HORNIK K., STINCHCOMBE M., WHITE H. et AUER P., "Degree of Approximation Results for

Feedforward Networks Approximating Unknow Mappings and Their Derivates", Neural Computation, volume 6, 1994.

[Hua96] HUA K., FALTINGS B.V. et SMITH I.F.C., "CADRE : Casebased geometric design". AI in Engineering, pages 171–183, 1996.

[Hug99] HUGGET A., SEBASTIAN P. et NADEAU J.-P., "Global Optimization of a Dryer by using Neural Networks and Genetic Algorithms", AIChE Journal, Vol 45, No 6, pp 1227-38, 1999.

[Ide69] IDEL'CIK I.E., "Memento des pertes de charge". Dans Collection de la Direction des Etudes et Recherches d'Electricité de France, 1969.

[IEEE754] IEEE Standard for Binary Floating-Point Arithmetic. IEEE Std 754-1985 (Reaffirmed 1990). IEEE Computer Society.

[Jan90] JANSSEN P., "Aide à la conception : Une approche basée sur la satisfaction de contraintes", Thèse de doctorat, Université de Montpellier, février 1990.

[Jun06] JUNKER U., "Configuration". Handbook of constraint programming, chapitre 24. 2006.

[Kay84] KAYS W.M. et LONDON A.L., "Compact Heat Exchangers". Mc Graw-hill book company, 1984.

[Kea95] KEARFOTT R.B., "A Fortran 90 Environment for Research and Prototyping of Enclosure Algorithms for Nonlinear Equations and Global Optimization". ACM Trans. Math. Soft.. 21(1): 63-78, 1995.

[Kea02] KEARFOTT R.B., "On Existence and Uniqueness Verification for Non-Smooth Functions", Reliable Computing. 8 (4), pp. 267—282, 2002

[Kea03] KEARFOTT R.B. et DIAN J., "Existence Verification for Higher Degree Singular Zeros of Complex Nonlinear Systems", SIAM J. Numer. Anal.41 (6), pp. 2350-2373, 2003.

[Khe94] KHEDRO T. et GENESERETH M.R., "Progressive Negotiation for Resolving Conflicts Among Distributed Heterogeneous Cooperating Agents", National Conference on Artificial Intelligence, Seattle, WA, July 1994.

[Kle97] KLEIN M., "An exception handling approach to enhancing consistency, completeness and correctness in collaborative requirements capture". Concurrent Engineering: Reasearch and Applications, 5(1):73–80, 1997.

[Kon97] KONDRAK G. et VAN BEEK P., "A theoretical evaluation of selected backtracking algorithms". Artificial Intelligence, 89:365–387, 1997.

[Lho93] LHOMME O., "Consistency Techniques for Numeric CSPs", In Proceedings of IJCAI–93, Morgan Kaufmann, San Mateo, CA, 1993, pp. 232–238.

[Lot98] LOTTAZ C., STALKER R.A. et SMITH I.F.C., "Constraint solving and preference activation for interactive design". Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI-EDAM), 12(1):13–27, 1998.

[Lot99] LOTTAZ C., CLEMENT D., SMITH I.F.C. et FALTINGS B.V., "Constraint-based support for collaboration in design and construction". Computing in Civil Engineering, 13(1):23–35, 1999.

[Mac77] MACKWORTH A.K., "Consistency in Networks of Relations". Artificial Intelligence, 8(1) :99–118,

1977.

[Mad98] MADELAINE B., "Détermination d'un modèle dynamique pertinent pour la commande : de la réduction à la construction", Thèse de doctorat de l'école Nationale Supérieure de l'aéronautique et de l'Espace, 1998.

[Mar06] MARIOTT K, STUCKEY P.J. et WALLACE M., "Constraint Logic Programming". Handbook of Constraint Programming, chapitre 12, 2006.

[Mer00] MERLET J.-P., "ALIAS: an interval analysis based library for solving and analyzing system of equations". In *SEA*, Toulouse, 14-16 Juin 2000.

[Mer05] J-P. Merlet. Applications of interval analysis in robotics. In *Interval methods and their applications*, Copenhagen, 25-27, 2005.

[Mes06] MESSINE F. et NOGAREDE B., "The Electromagnetical Actuators Design Problem: a General and Rational Approach". Dans : IEEE Transactions on Magnetics, IEEE, V. 42 N. 12, p. 3847-3853, 2006.

[Mic01] MICHEL C., RUEHER M. et LEBBAH Y., "Solving constraints over floating point numbers". In proceedings of the 7th International Conference on Principles and Practice of Constraint Programing 2001 (CP'2001), LNCS 2239 (Toby Walsh Ed., Springer Verlag), pp 524-538, 2001.

[Mit90] MITTAL S. et FALKENHEINER B. "Dynamic Constraint Satisfaction Problems". In T. Dietterich and W. Swartout, editors, Proceedings of National Conference on Artificial Intelligence, pages 25–32, Boston, USA, 1990. AAAI Press.

[Mon99] MONFROY E. et RETY J.H., "Chaotic Iteration for Distributed Constraint Propagation". In J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G. Lamont, editors, Proceedings of ACM Symposium on Applied Computing, pages 19–24, San Antonio, Texas, USA, 1999. ACM Press.

[Mon74] MONTANARI U., "Networks of Constraints : Fundamental Properties and Applications to Picture Processing". Information Science, 7(2) :95–132, 1974.

[Moo66] MOORE R.E., "Interval Analysis". Prentice-Hall, Englewood Cliffs, NJ, 1966.

[Mul02] MULYANTO T., "Utilisation des techniques de programmation par contraintes pour la conception d'avions". Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France.

[Nev06] NEVEU B., CHABERT G. et TROMBETTONI G., "When Interval Analysis Helps Interblock Backtracking". In: CP'06. 2006. Nantes, France.

[OSu01] O'SULLIVAN B., "Constraint-Aided Conceptual Design". Professional Engineering Publishing, ISBN: 1-86058-335-0, 2001.

[Ott94] OTTO K.N. et ANTONSSON E.K., "Modelling Imprecision in Product Design", Proceedings of the Third IEEE International Conference on Fuzzy Systems FUZZ-IEEE '94, 346-351, 1994.

[Oui02] OUIS S., NARENDRA J. et BOIZUMAULT P., "COINS: a constraint-based interactive solving system", ICLP'02 12th Workshop on Logic Programming Environments (WLPE'02), pp. 31 - 46, 2002

[Pah96] PAHL G. et BEITZ W., "Engineering Design: A Systematic Approach". ISBN 3-540-19917-9, Springer-

Verlag Berlin Heidelberg, 1996.

[Per02] PEREZ-GRANDE I. et LEO T.J., "Optimization of a commercial aircraft environmental control system", Applied Thermal Engineering, Vol. 22, pp 1885–1904, 2002.

[Pet97] PETRIE C.J., JEON H. et CUTKOSKY M.R., "Combining constraint propagation and backtracking for distributed engineering". In Constraints & Agents Workshop at AAAI'97, pages 76–82, Providence, Rhode Island. AAAI Press, 1997.

[Pow05] POWER D.J. et SHARDA R., "Model-driven decision support systems: Concepts and research directions", Decision Support Systems, 2005.

[Red96] REDDY S.Y., FERTIG K.W. et SMITH D.E., "Constraint Management Methodology for Conceptual Design Tradeoff Studies". Design Theory and Methodology Conference. 1996. Irvine, CA.

[Ref99] REFALO P., "Tight Cooperation and Its Application in Piecewise Linear Optimization". In: CP'99. 1999. p. 375–389.

[Ric68] RICHARDSON D., "Some Unsolvable Problems Involving Elementary Functions of a Real Variable". Journal of Symbolic Logic 1968. 33(4):514–520.

[Ros85] ROHSENOW W.M., HARTNETT J.P. et GANIC E.N., "Handbook of heat transfer fundamentals (2nd edition)". New York, McGraw-Hill Book Co., 1985.

[Ros06] ROSSI F., VAN BEEK P. et WALSH T., "Handbook of Constraint Programming", Elsevier, 2006.

[Rot90] ROTHWELL R. et GARDINER P., "Robustness and Product Design Families", Design Management: A Handbook of Issues and Methods. 1990. pp. 279-292. (Oakley, M., ed.), Basil Blackwell Inc., Cambridge, MA.

[Rum88] RUMP S.M., "Algorithms for verified inclusions—theory and practice". In R. Moore, ed. , Reliability in Computing, p. 109–126. Academic Press. 1988.

[Ryc04] D. RYCKELYNCK, "A priory hyper-reduction method: an adaptative approach", J Comput Phys 202, pp. 346–366, 2004.

[Sab96] SABIN D. et FREUDER E.C., "Configuration as Composite Constraint Satisfaction". In AAAI-96 Fall Symposium on Configuration, pages 28–36, 1996.

[Sam96] SAM-HAROUD D. et FALTINGS B., "Consistency Techniques for Continuous Constraints". Constraints 1996. 1:85-118.

[Saw01] SAWADA H., YAN X.-T., "Applying generic constraint solving techniques in providing insights into engineering design", Proceedings of International Conference on Engineering Design, Glasgow, august 2001.

[Sca04a] SCARAVETTI D., "Formalisation préalable d'un problème de conception pour l'aide à la décision en conception préliminaire ", Thèse de doctorat. Université de Bordeaux I. 2004.

[Sca04b] SCARAVETTI D., NADEAU J.-P., SEBASTIAN P. et PAILHES J., "Aided Decision-Making for an Embodiment Design Problem". International IDMME. Bath, UK, 2004.

[Sca05] SCARAVETTI D., NADEAU J.P., PAIHLES J. et SEBASTIAN P., "Structuring of embodiment design

problem based on the product lifecycle". International Journal of Product Development 2005. 2(1-2):47–70. ISSN: 1477-9056. Inderscience, Geneve.

[Sch97] SCHIEX T., FARGIER H. et VERFAILLIE G., "Problème de satisfaction de contraintes valué". Revue d'Intelligence Artificielle, 11(3):339–373, 1997.

[Seb03] SEBASTIAN P., "Approximation et Intégration de Modèles pour l'Aide à la Décision en Génie des Procédés", Mémoire d'Habilitation à diriger les recherches, Ecole Nationale Supérieure d'Arts et Métiers, 2003.

[Seb07] SEBASTIAN P., CHENOUARD R., NADEAU J.-P. et FISCHER X., "The Embodiment Design Constraint Satisfaction Problem of the BOOTSTRAP facing interval analysis and Genetic Algorithm based decision support tools", IJIDEM, 1 (2), pages 99-106, 2007.

[Ser91] SERRANO D., "Constraint-based concurrent design". Systems Automation: Research and Application, 1(3):287–304, 1991.

[Sil99] SILAGHI M.C., SAM-HAROUD D. et FALTINGS B., "Intelligent Domain Splitting for CSPs with Ordered Domains". Proc. of CP'99. LNCS 1713. Springer Verlag, pages 488-489. 1999.

[Smi96] SMITH I.F.C., STALKER R.A. et LOTTAZ C., "Creating design objects from cases for interactive spatial composition". In Gero, J. S. and Sudweeks, F., editors, AI in Design, pages 97–116, Boston MA. Kluwer Academic Publishers.

[Smi06] SMITH B., "Modelling". In Rossi F., Van Beek P. et Walsh T., editors, Handbook of Constraint Programming, chapitre 11, pages 377–404, Elsevier, 2006.

[Sta95] STAHL V., "Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations". PhD thesis, University of Linz, Austria, 1995.

[Stu97] STUMPTER M., "An overview of knowledge-based configuration", AI-Communications, pp 111-125, 1997.

[Szy99] SZYKMAN S., RACZ J.W. et SRIRAM R.D., "The Representation of Function in Computer-based Design". Proceedings of the 1999 ASME Design Engineering Technical Conferences (11th International Conference on Design Theory and Methodology). DETC99/DTM-8742, 1999.

[Tho94] THORNTON A.C., "A support tool for constraint processes in embodiment design", ASME Design Theory and Methodology Conference, Minneapolis, september 1994.

[Tho99] THORNTON A.C., "A Mathematical Framework for the Key Characteristic Process". Research in Engineering Design 1999. 11(3):145–157.

[Tro07] TROMBETTONI G. et CHABERT G., "Constructive Interval Disjunction". Proc. of CP, constraint programming, LNCS 4741, Springer, p. 635-650, 2007.

[Van92] VAN HENTENRYCK P., DEVILLE Y. et TENG C.-M., "A generic arc-consistency algorithm and its specialisation". Artificial Intelligence, 57 :292–321, 1992.

[Van97a] VAN HENTENRYCK P., MCALLESTER D. et KAPUR D., "Solving Polynomial Systems Using Branch and Prune Approach". SIAM Journal on Numerical Analysis 1997. 34(2):797–827.

[Van97b] VAN HENTENRYCK P., MICHEL L. et DEVILLE Y., "Numerica - A Modelling Language for Global Optimization". MIT Press, Cambridge, Massachusetts, 1997.

[Van98] VAN HENTENRYCK P., MICHEL L. et BENHAMOU F., "Newton: Constraint Programming over Nonlinear Constraint". Science of Computer Programming, 30, 83-118, 1998.

[Van05] VAN HENTENRYCK P. et MICHEL L. "Constraint-based local search". Cambridge, Mass. : MIT Press, 2005.

[Van06] VAN OUDENOVE T., GABORIT P., ALDONONDO M. et VAREILLES E., "CSP dynamiques en configuration". JFPC06, 206.

[Var05] VAREILLES E., "Conception et approches par propagation de contraintes : contribution à la mise en œuvre d'un outil d'aide interactif". Thèse de doctorat. Ecole des mines d'Albi Carmaux, 2005.

[Ver95] VERFAILLIE G. et SCHIEX T., "Maintien de solution dans les problèmes de dynamiques de satisfaction de contraintes : bilan de quelques approches". Revue d'Intelligence Artificielle, 9 :269–309, 1995.

[Ver04] VERNAT Y., "Formalisation et qualification de modèles par contraintes en conception préliminaire". Thèse de doctorat. ENSAM, 2004.

[Vu02] VU X.-H., SAM-HAROUD D. et SILAGHI M., "Approximation Techniques for Nonlinear Problems with Continuum of Solutions". In: SARA'02. 2002. Kananaskis, Canada.

[Vu05] VU X.-H., "Rigorous Solution Techniques for Numerical Constraint Satisfaction Problems". Thèse de doctorat. EPFL, Lausane, 2005.

[Wal75] WALTZ D., "Generating semantic descriptions from drawings of scenes with shadows". In P. H. W, editor, The Psychology of Computer Vision, pp. 19—91. McGraw-Hill, New York, NY, 1975.

[Wan74] WANG P.S., "The Undecidability of the Existence of Zeros of Real Elementary Functions". Journal of the ACM 1974. 21(4):586-589.

[Yan01] Bernard YANNOU. "Péconception de produits", Mémoire d'Habilitation à Diriger des Recherches, INPG, juin 2001.

[Yan03] YANNOU B., SIMPSON T.W. et BARTON R.R., "Towards a conceptual design explorer using metamodeling approaches and constraint programming". In Proceedings of DETC'03 : ASME International 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conferences, Chicago, Illinois, September 2003.

[Yao97] YAO Z. et JOHNSON A.L., "On estimating the feasible solution space of design". Computer-Aided Design 1997. 29(9):649–655.

[Zim01] ZIMMER L. et ZABLIT P., "Global Aircraft Predesign based on Constraint Propagation and Interval Analysis". In: CEAS-MADO'01. 2001. Koln, Germany.

[Zim04] ZIMMER L, ANGLADA A., CHRISTIE M. et GRANVILLIERS L., "Constraint Explorer: a Modelling and Sizing Tool for Engineering Design", invited session on Metamodelling and Constraint Based Problem Solving for Embodiment Design Support Systems in SCI, Orlando, 2004.

ANNEXES

A - Modèles des problèmes de conception

Circuit d'une pompe et de réservoirs (voir V.2.1)

Constantes

Nom	**Description**
$P_{amont} = 40000$	Pression dans le réservoir amont (Pa)
$P_{aval} = 101325$	Pression dans les réservoirs avals (Pa)
$R_0/D = 2$	Rapport du rayon du coude sur le diamètre du tube (m)
$L_1 = 1$	Longueur de la conduite en aval (m)
$L_2 = 1$	Longueur de la conduite en aval (m)
$\rho = 1000$	Masse volumique de l'eau (kg/m^3)
$T = 13$	Température de l'eau (°C)
$\mu = 0.001$	Viscosité dynamique de l'eau (kg/(m.s))
$g = 9.81$	Accélération de la pesanteur (m/s^2)
$L_a = 1$	Longueur de la conduite en amont (m)
$D = 0.055$	Diamètre des tubes dans le circuit amont (m)
$H_a = 2$	Longueur de la conduite en amont (m)
$H_w = 0.5$	Hauteur d'eau dans le réservoir amont (m)
$H_{r1} = 5$	Longueur de la conduite avant la bifurcation en aval(m)
$H_{r2} = 2$	Longueur de la conduite après la bifurcation en aval (m)
$S = \frac{\pi}{4}D^2$	Section de la conduite en amont (m^2)
$A_{e1} = \pi D H_a$	Surface de la conduite en amont (m^2)
$A_{e2} = \pi D L_a$	Surface de la conduite en amont (m^2)
$A_{e3} = \pi D H_{r1}$	Surface de la conduite en aval avant la bifurcation (m^2)
$\xi_1 = 0.5$	Coefficient de pertes de charge (-)
$\xi_3 = 1$	Coefficient de pertes de charge (-)
$\xi_4 = 0.5$	Coefficient de pertes de charge (-)
$\xi_5 = 0.1$	Coefficient de pertes de charge (-)
$P_{sat} = e^{(23.3265 - \frac{3802.7}{T+273.18} - \frac{472.68}{T+273.18})^2}$	Pression de saturation de l'eau dans la pompe (Pa)

Variables de conception

Nom	Précision	Description
$D_{r1} : real = [0.01, 0.1]$	0.001	Diamètre du tube menant au réservoir 1 (m)
$D_{r2} : real = [0.01, 01]$	0.001	Diamètre du tube menant au réservoir 2 (m)
$Pump : int = [1, 5]$	-	Référence des vitesses de rotation de la pompe (-)

Variables auxiliaires

Nom	Précision	Description
$Q : real = [17/3600, 96/3600]$	0.0001	Débit volumique d'eau dans la pompe (m^3/s)
$H : real = [2.5, 21.6]$	1%	Hauteur manométrique total de la pompe (m)
$f_1 : real = [-10, 10]$	?	Facteur de fanning (-)
$f_2 : real = [-10, 10]$	?	Facteur de fanning (-)
$f_3 : real = [-10, 10]$	?	Facteur de fanning (-)

Variables alias

Nom	Contrainte	Description
Q_{r1}	$Q_{r1} = \frac{Q}{2}$	Débit d'eau vers le réservoir 1 (m^3/s)
Q_{r2}	$Q_{r2} = Q_{r1}$	Débit d'eau vers le réservoir 2 (m^3/s)
V	$V = \frac{Q}{S}$	Vitesse d'écoulement de l'eau en amont (m/s)
S_{r1}	$S_{r1} = \frac{\pi}{4} D_{r1}^2$	Section de la conduite vers le réservoir 1 (m^2)
S_{r2}	$S_{r2} = \frac{\pi}{4} D_{r2}^2$	Section de la conduite vers le réservoir 2 (m^2)
Re_1	$Re_1 = \frac{\rho V D}{\mu}$	Nombre de Reynlods (-)
Re_2	$Re_2 = \frac{\rho \frac{Q_{r1}}{S_{r1}} D_{r1}}{\mu}$	Nombre de Reynlods (-)
Re_3	$Re_3 = \frac{\rho \frac{Q_{r2}}{S_{r2}} D_{r2}}{\mu}$	Nombre de Reynlods (-)
$NPSH_d$	$NPSH_d = \frac{P_{amont} - P_{sat}}{\rho g} + (H_a + 2D) - \frac{\Delta P_{amont}}{\rho g}$	NPSH disponible (m)
$NPSH_r$	$NPSH_r = 1.2144 * 10^{-5} * (3600 * Q)^3 - 1.2301 * 10^{-3} * (3600 * Q)^2 + 4.9136 * 10^{-2} * (3600 * Q) + 0.49957$	NPSH requis (m)
ΔP_{amont}	$\Delta P_{amont} = \Delta P_0 + \Delta P_1 + \Delta P_2 + \Delta P_3$	Pertes de charge totales en

		amont (Pa)
ΔP	$\Delta P = \Delta P_{amont} + \delta P_4 + \Delta P_5 + \Delta P_6 + \Delta P_7$	Pertes de charge totales (Pa)
ΔP_0	$\Delta P_0 = \xi_1 \frac{\rho V^2}{2}$	Pertes de charge en entrée de conduite en amont (Pa)
ΔP_1	$\Delta P_1 = f_1 \frac{A_{e1} \rho Q^2}{2S^3}$	Pertes de charge dans la conduite verticale en amont (Pa)
ΔP_2	$\Delta P_2 = \xi_2 \frac{\rho V^2}{2}$	Pertes de charge dans le coude en amont (Pa)
ΔP_3	$\Delta P_3 = f_1 \frac{A_{e2} \rho Q^2}{2S^3}$	Pertes de charge dans la conduite horizontale en amont (Pa)
ΔP_4	$\Delta P_4 = f_1 \frac{A_{e3} \rho Q^2}{2S^3}$	Pertes de charge dans la conduite verticale en aval (Pa)
ΔP_5	$\Delta P_5 = \xi_4 \frac{\rho Q_{r1}^2}{2S_{r1}^2}$	Pertes de charge dans la bifurcation vers le réservoir 1 (Pa)
ΔP_6	$\Delta P_6 = f_2 \frac{A_{e4} \rho Q_{r1}^2}{2S_{r1}^3}$	Pertes de charge dans la conduite horizontale vers le réservoir 1 (Pa)
ΔP_7	$\Delta P_7 = \xi_3 \frac{\rho Q_{r1}^2}{2S_{r1}^2}$	Pertes de charge en sortie de conduite dans le réservoir 1 (Pa)
ΔP_8	$\Delta P_8 = \xi_5 \frac{\rho Q_{r2}^2}{2S_{r2}^2}$	Pertes de charge dans la bifurcation vers le réservoir 2 (Pa)
ΔP_9	$\Delta P_9 = f_3 \frac{A_{e5} \rho Q_{r2}^2}{2S_{r2}^3}$	Pertes de charge dans la conduite vertical vers le réservoir 2 (Pa)
ΔP_{10}	$\Delta P_{10} = \xi_6 \frac{\rho Q_{r2}^2}{2S_{r2}^2}$	Pertes de charge dans le coude en aval (Pa)
ΔP_{11}	$\Delta P_{11} = f_3 \frac{A_{e6} \rho Q_{r2}^2}{2S_{r2}^3}$	Pertes de charge dans la conduite horizontale vers le réservoir 2 (Pa)
ΔP_{12}	$\Delta P_{12} = \xi_3 \frac{\rho Q_{r2}^2}{2S_{r2}^2}$	Pertes de charge en sortie de conduite dans le réservoir 2 (Pa)
ξ_2	$\xi_2 = 0.15 + 0.0175 \times 4 f_1 \frac{R_0}{D} 90$	Coefficient de pertes de charge dans le coude en amont (-)

ξ_6	$\xi_6 = 0.15 + 0.0175 \times 4 f_3 \frac{R_0}{D_{r1}} 90$	Coefficient de pertes de charge dans le coude aval (-)

Contraintes

Relation	Description
piecewise($pump$, $[1,1] : H = -6.32734215 * 10^{-6} * (Q0 * 3600)^3 - 1.19161754914 * 10^{-3} * (Q0 * 3600)^2 + 6.580153759339 * 10^{-2} * (Q0 * 3600) + 17.80804203068717$, $[2,2] : H = -6.32734215 * 10^{-6} * (Q0 * 3600)^3 - 1.19161754914 * 10^{-3} * (Q0 * 3600)^2 + 6.580153759339 * 10^{-2} * (Q0 * 3600) + 18.80804203068717$, $[3,3] : H = -6.32734215 * 10^{-6} * (Q0 * 3600)^3 - 1.19161754914 * 10^{-3} * (Q0 * 3600)^2 + 6.580153759339 * 10^{-2} * (Q0 * 3600) + 19.80804203068717$, $[4,4] : H = -6.32734215 * 10^{-6} * (Q0 * 3600)^3 - 1.19161754914 * 10^{-3} * (Q0 * 3600)^2 + 6.580153759339 * 10^{-2} * (Q0 * 3600) + 20.80804203068717$, $[5,5] : H = -6.32734215 * 10^{-6} * (Q0 * 3600)^3 - 1.19161754914 * 10^{-3} * (Q0 * 3600)^2 + 6.580153759339 * 10^{-2} * (Q0 * 3600) + 21.80804203068717$)	Hauteur manométrique produite par la pompe pour 5 vitesses de rotation
piecewise(Re_1, $[200, 2100] : f_1 = \frac{16}{Re_1}$, $[2100, 50000] : f_1 = 0.10512 * e^{-0.244 * ln(Re_1)}$, $[50000, 10^6] : f_1 = 0.04234 * e^{-0.164 * ln(Re_1)}$)	Expression du facteur de Fanning pour le circuit amont
piecewise(Re_2, $[200, 2100] : f_2 = \frac{16}{Re_2}$, $[2100, 50000] : f_2 = 0.10512 * e^{-0.244 * ln(Re_2)}$, $[50000, 10^6] : f_2 = 0.04234 * e^{-0.164 * ln(Re_2)}$)	Expression du facteur de Fanning pour le circuit aval vers le réservoir 1
piecewise(Re_3, $[200, 2100] : f_3 = \frac{16}{Re_3}$ $[2100, 50000] : f_3 = 0.10512 * e^{-0.244 * ln(Re_3)}$, $[50000, 10^6] : f_3 = 0.04234 * e^{-0.164 * ln(Re_3)}$)	Expression du facteur de Fanning pour le circuit aval vers le réservoir 2
$S_{r1} + S_{r2} = S$	
$NPSH_r \leq NPSH_d$	Condition de non cavitation de la

	pompe
$\Delta P_5 + \Delta P_6 + \Delta P_7 = \Delta P_8 + \Delta P_9$ $+\Delta P_10 + \Delta P_11 + \Delta P_12 + \rho g H_{r2}$	Bilan d'énergie dans le circuit aval

Échangeur d'un système de refroidissement d'apéritif (voir V.2.2)

Constantes

Nom	Description
$dose = 0.04$	Volume de la dose à refroidir (l)
$V = 0.06$	Volume du doseur (l)
$eail = 0.0005$	Épaisseur des ailettes (m)
$T_i = 20$	Température initiale de l'apéritif (°C)
$T_f = 8$	Température finale de l'apéritif (°C)

Variables de conception

Nom	Précision	Description
$mater : int = \{1, 2\}$	-	Type de matériau (-)
$diam : int = \{1, 2\}$	-	Diamètre de tube (-)
$ail : int = \{1, 2\}$	-	Longueur d'ailette (-)
$e : real = [0.001, 0.004]$	0.001	Espace entre les ailettes (m)
$N : int = [5, 20]$	-	Nombre d'ailettes (-)

Variables auxiliaires

Nom	Précision	Description
$\lambda : int = [1, 200]$	-	Conductivité des ailettes (W/m/K)
$t : real = [11, 15]$	0.1	Temps de refroidissement (s)
$d : int = [0, 50]$	-	Diamètre du tube (mm)
$L : int = [0, 50]$	-	Côté de l'ailette (mm)
$T_{sat} : real = [-15, 2]$	0.1	Température de saturation (°C)

Variables alias

Nom	Contrainte	Description

A	$A = N \cdot (2 \cdot Aail + \pi \cdot e \cdot d) \cdot \frac{dose}{V}$	Surface de l'échangeur (m^2)
$Aail$	$Aail = (L^2 - \frac{\pi}{4} \cdot d^2)$	Surface d'une demi ailette (m^2)
fi	$fi = \frac{L-d}{2} \cdot \sqrt{\frac{\frac{2 \cdot h}{lamda}}{eail}}$	Coefficient de rendement d'ailette (-)
η	$\eta = \frac{\frac{e^{2 \cdot fi} - 1}{e^{2 \cdot fi} + 1}}{fi}$	Rendement d'ailette (-)
h	$h = \frac{1.2}{e}$	Coefficient d'échange du doseur (W/m^2/K)

Contraintes

Relation	**Description**
$V = e \cdot Aail \cdot N$	Volume du doseur
$V \geq dose$	Condition de remplissage
$T_f = T_{sat} + (T_i - T_{sat}) \cdot e^{-\frac{\frac{h \cdot A \cdot t \cdot \eta}{3.9 \cdot 10^6}}{dose}}$	Pour notre apéritif : $\rho \cdot C_p = 3.9 \cdot 10^6$
$Piecewise(diam,$ $[1,1] : d = 16,$ $[2,2] : d = 18)$	Diamètres normalisés pour les tubes
$Piecewise(mater,$ $[1,1] : \lambda = 200,$ $[2,2]\lambda = 20)$	Matériaux des ailettes : 1 = acier, 2 = inox
$Piecewise(ail,$ $[1,1] : L = 40,$ $[2,2] : L = 50)$	Largeurs normalisées pour les ailettes

Système de conditionnement d'air sans ailette (voir V.2.3)

Constantes

Nom	**Description**
$Cp = 1000$	Chaleur spécifique à pression constante de l'air (J/(kg.K))
$r = 287$	Constante des gaz parfaits rapportée à la masse molaire de l'air (J/(kg.K))
$\gamma = 1.4$	Rapport de capacité thermique (-)
$Z = 10500$	Altitude de vol (m)
$M = 0.8$	Vitesse de l'avion (m/s)
$TC_{TR} = 8$	Taux de compression du turbo-réacteur (-)

$\eta_{TRd} = 0.9$	Rendement du diffuseur du turbo-réacteur (-)
$\eta_{TRc} = 0.8$	Rendement du compresseur du turbo-réacteur (-)
$\eta_C = 0.75$	Rendement du compresseur (-)
$\eta_{AT} = 0.95$	Rendement de l'arbre de transmission (-)
$\eta_t = 0.8$	Rendement de la turbine (-)
$\eta_{NO} = 0.9$	Rendement de la buse d'éjection (-)
$\eta_{DI} = 0.9$	Rendement du diffuseur (-)
$T_5 = 278.15$	Température de l'air en sortie du SCA (K)
$p_5 = 85000$	Pression de l'air en sortie du SCA (Pa)
$q = 0.7$	Débit d'air prélevé sur le turbo-réacteur (kg/s)
$kp = 20$	Conductivité des plaques (W/(m.K))
$tp = 0.001$	Épaisseur des plaques (m)
$T_a = 288.2 - 0.00649 * Z$	Température atmosphérique (K)
$p_a = 101290 * (\frac{T_a}{288.08})^{5.256}$	Pression atmosphérique (Pa)
$T_0 = T_a * (1 + \frac{M^2 * (\gamma - 1)}{2})$	Température en sortie du diffuseur du turbo-réacteur (K)
$p_0 = p_a * (\eta_{TRd} * \frac{M^2 * (\gamma - 1)}{2} + 1)^{\frac{\gamma}{\gamma - 1}}$	Pression en sortie du diffuseur du turbo-réacteur (Pa)
$T_1 = T_0 * (1 + \frac{1}{\eta_{TRc}} * ((\frac{p1}{p0})^{\frac{\gamma - 1}{\gamma}} - 1))$	Température en entrée du SCA (K)
$p_1 = TC_{TR} * p_0$	Pression en entrée du SCA (Pa)
$pi_{Dc} = p_a$	Pression en entrée du diffuseur (Pa)
$Ti_{Dc} = T_a$	Température en entrée du diffuseur (K)
$po_{Dc} = pi_{Dc} * (\eta_{DI} * M^2 * \frac{\gamma - 1}{2} + 1)^{\frac{\gamma}{\gamma - 1}}$	Pression en sortie du diffuseur (Pa)
$To_{Dc} = Ti_{Dc} * (1 + M^2 * \frac{\gamma - 1}{2})$	Température en sortie du diffuseur (K)
$po_{Bc} = p_a$	Pression en sortie de la buse d'éjection (Pa)
$\nu = \frac{r * T_1}{p_1}$	Volume massique (m^3/kg)
$p_4 = p_5$	Pression en sortie de la turbine (Pa)

Variables de conception

Nom	**Précision**	**Description**
$L_x : real = [0.01, 0.1]$	0.01	Largeur de l'échangeur thermique (m)
$\tau : real = [0.8, 2]$	0.01	Rapport des débits (-)
$rh_h : real = [0.001, 0.01]$	0.0001	Espace entre plaques côté chaud (m)

$rh_c : real = [0.001, 0.01]$	0.0001	Espace entre plaques côté froid (m)
$Rv : real = \{0.01, 0.02, ..., 0.2\}$	-	Rayon de la vanne d'équilibrage (m)

Variables auxiliaires

Nom	Précision	Description
$p_2 : real = [0, 10^6]$	5%	Pression en entrée de l'échangeur côté chaud (Pa)
$p_3 : real = [0, 10^6]$	5%	Pression en sortie de l'échangeur côté chaud (Pa)
$p_{eic} : real = [0, 10^6]$	5%	Pression en entrée de l'échangeur côté froid (Pa)
$p_{eoc} : real = [0, 10^6]$	5%	Pression en sortie de l'échangeur côté froid (Pa)
$T_2 : real = [0, 1000]$	5%	Température en entrée de l'échangeur côté chaud (K)
$T_3 : real = [0, 1000]$	5%	Température en sortie de l'échangeur côté chaud (K)
$T_{eic} : real = [0, 1000]$	5%	Température en entrée de l'échangeur côté froid (K)
$T_{eoc} : real = [0, 1000]$	5%	Température en sortie de l'échangeur côté froid (K)
$\Delta P_{eh} : real = [-\infty, +\infty]$	10%	Pertes de charge dans l'échangeur côté chaud (Pa)
$\Delta P_{ec} : real = [-\infty, +\infty]$	10%	Pertes de charge dans l'échangeur côté froid (Pa)
$\epsilon : real = [0, 1]$	5%	Efficacité de l'échangeur (-)
$q_{ra} : real = [0, +\infty]$	0.001	Débit d'air côté froid (kg/s)
$q_{ma} : real = [0, +\infty]$	0.001	Débit d'air côté chaud (kg/s)
$p_4 : real = [0, 10^6]$	5%	Pression en sortie de turbine (Pa)
$T_4 : real = [0, 1000]$	5%	Température en sortie de turbine (K)
$f_h : real = [-10, 10]$	∞	Facteur de Fanning côté chaud (-)
$f_c : real = [-10, 10]$	∞	Facteur de Fanning côté froid (-)
$Nu_h : real = [-\infty, +\infty]$	∞	Nombre de Nusselt côté chaud (-)
$Nu_c : real = [-\infty, +\infty]$	∞	Nombre de Nusselt côté froid (-)
$\alpha : int = [10, 80]$	-	Angle d'ouverture de la vanne d'équilibrage (°)

Variables alias

Nom	Contrainte	Description
L_y	$L_y = L_x$	Longueur de l'échangeur (m)
L_z	$L_z = 0.25 * L_x$	Hauteur de l'échangeur (m)
A	$A = A_h + A_c$	Surface d'échange totale (m^2)
A_h	$A_h = L_x * L_y * n$	Surface d'échange côté chaud (m^2)

A_c	$A_c = L_x * L_y * n$	Surface d'échange côté froid (m^2)
A_p	$A_p = L_x * L_y * n$	Surface d'un côté de plaque (m^2)
Af_h	$Af_h = rh_h * L_y * (n+1)/2$	Surface frontale côté chaud (m^2)
Af_c	$Af_c = rh_c * L_x * (n+1)/2$	Surface frontale côté froid (m^2)
n	$n = \frac{2*L_z}{rh_c+rh_h+2*t_p} - (rh_c + rh_h)$	Nombre de plaques (-)
G_h	$G_h = \frac{q_{ma}}{Af_h}$	Vitesse massique côté chaud ($kg/(s.m^2)$)
G_c	$G_c = \frac{q_{ra}}{Af_c}$	Vitesse massique côté froid ($kg/(s.m^2)$)
μ_h	$\mu_h = \frac{-2.15*10^{-5}-2.225*10^{-9}*(T_3+T_2)}{2} + \frac{1.725*10^{-6}*(\sqrt{T_3}+\sqrt{T_2})}{2}$	Viscosité de l'air côté chaud (kg/(m.s))
μ_c	$\mu_c = \frac{-2.15*10^{-5}-2.225*10^{-9}*(Te_{oc}+Te_{ic})}{2} + \frac{1.725*10^{-6}*(\sqrt{Te_{oc}}+\sqrt{Te_{ic}})}{2}$	Viscosité de l'air côté froid (kg/(m.s))
ρ_2	$\rho_2 = \frac{p_2}{r*T_2}$	Masse volumique de l'air en entrée de l'échangeur côté chaud (kg/m^3)
ρ_3	$\rho_3 = \frac{p_3}{r*T_3}$	Masse volumique de l'air en sortie de l'échangeur côté chaud (kg/m^3)
ρ_{ic}	$\rho_{ic} = \frac{pe_{ic}}{r*Te_{ic}}$	Masse volumique de l'air en entrée de l'échangeur côté froid (kg/m^3)
ρ_{oc}	$\rho_{oc} = \frac{pe_{oc}}{r*Te_{oc}}$	Masse volumique de l'air en sortie de l'échangeur côté froid (kg/m^3)
Cp_{oh}	$Cp_{oh} = 1003.7 + 6.8*10^{-2}*(T_3 - 273.15) + 2.22*10^{-4}*(T_3 - 273.15)^2$	Capacité calorifique de l'air en sortie de l'échangeur côté chaud (J/(kg.K))
Cp_{ih}	$Cp_{ih} = 1003.7 + 6.8*10^{-2}*(T_2 - 273.15) + 2.22*10^{-4}*(T_2 - 273.15)^2$	Capacité calorifique de l'air en entrée de l'échangeur côté chaud (J/(kg.K))
Cp_h	$Cp_h = \frac{Cp_{oh}+Cp_{ih}}{2}$	Capacité calorifique moyenne côté chaud (J/(kg.K))

Cp_{oc}	$Cp_{oc} = 1003.7 + 6.8 * 10^{-2} * (Te_{oc} - 273.15) + 2.22 * 10^{-4} * (Te_{oc} - 273.15)^2$	Capacité calorifique de l'air en sortie de l'échangeur côté froid
Cp_{ic}	$Cp_{ic} = 1003.7 + 6.8 * 10^{-2} * (Te_{ic} - 273.15) + 2.22 * 10^{-4} * (Te_{ic} - 273.15)^2$	Capacité calorifique de l'air en entrée de l'échangeur côté froid (J/(kg.K))
Cp_c	$Cpc = \frac{Cp_{oc}+Cp_{ic}}{2}$	Capacité calorifique moyenne côté froid (J/(kg.K))
m_{ex}	$m_{ex} = V_p * 7800$	Masse de l'échangeur (kg)
V	$V = L_x * L_y * L_z$	Volume de l'échangeur (m^3)
V_p	$V_p = \frac{A}{2} * t_p$	Volume des plaques de l'échangeur (m^3)
C_h	$C_h = \frac{q_{ma}}{Af_h * \rho_2}$	Vitesse de l'air côté chaud (m/s)
C_c	$C_c = \frac{q_{ra}}{Af_c * \rho_{ic}}$	Vitesse de l'air côté froid (m/s)
λ	$\lambda = \frac{q_{ra} * Cp_{ic}}{q_{ma} * Cp_{ih}}$	Rapport de débits de capacités calorifiques (-)
Nut	$Nut = \frac{q_{ma} * Cp_{ih}}{\frac{1}{h_h * A_h} + \frac{t_p}{k_p * A_p} + \frac{1}{h_c * A_c}}$	Nombre d'unités de transfert de l'échangeur (-)
h_h	$h_h = J_h * G_h * Cp_h * Pr_h^{-2/3}$	Coefficient de transfert convectif côté chaud (W/(m2.K))
h_c	$h_c = J_c * G_c * Cp_c * Pr_c^{-2/3}$	Coefficient de transfert convectif côté froid (W/(m2.K))
J_h	$J_h = St_h * Pr_h^{0.33}$	Coefficient de Colburn côté chaud (-)
J_c	$J_c = St_c * Pr_c^{0.33}$	Coefficient de Colburn côté froid (-)
Pr_h	$Pr_h = \frac{1}{2} * (1.65 - 5.4 * 10^{-4} * (T_3 + T_2) + 5.0 * 10^{-7} * (T_3^2 + T_2^2))$	Nombre de Prandtl côté chaud (-)
Pr_c	$Pr_c = \frac{1}{2} * (1.65 - 5.4 * 10^{-4} * (Te_{oc} + Te_{ic}) + 5.0 * 10^{-7} * (Te_{oc}^2 + Te_{ic}^2))$	Nombre de Prandtl côté froid (-)
St_h	$St_h = \frac{Nu_h}{Re_h * Pr_h}$	Nombre de Standton côté chaud (-)
St_c	$St_c = \frac{Nu_c}{Re_c * Pr_c}$	Nombre de Standton côté froid (-)

Re_h	$Re_h = \frac{4*rh_h*G_h}{\mu_h}$	Nombre de Reynolds côté chaud (-)
Re_c	$Re_c = \frac{4*rh_c*G_c}{\mu_c}$	Nombre de Reynolds côté froid (-)
Kc_h	$Kc_h = 0.00505236835109 * \sigma_h^2 - 0.00414707431984 * \sigma_h + 0.00347507173062 * log^2(Re_h) - 0.08548307647633 * \sigma_h^2 + 0.06740608329495 * \sigma_h - 0.09241949837272 * log(Re_h) - 0.18282301765817 * \sigma_h^2 - 0.17962391485785 * \sigma_h + 1.00333194877608$	Coefficient de pertes de charge en sortie de l'échangeur côté chaud (-)
Kc_c	$Kc_c = 0.00505236835109 * \sigma_c^2 - 0.00414707431984 * \sigma_c + 0.00347507173062 * log^2(Re_c) - 0.08548307647633 * \sigma_c^2 + 0.06740608329495 * \sigma_c - 0.09241949837272 * log(Re_c) - 0.18282301765817 * \sigma_c^2 - 0.17962391485785 * \sigma_c + 1.00333194877608$	Coefficient de pertes de charge en sortie de l'échangeur côté froid (-)
Ke_h	$Ke_h = -0.00496672650332 * \sigma_h^2 0.00113607587171 * \sigma_h - 0.00001379297260 * log^2(Re_h) + 0.06612031387891 * \sigma_h^2 + 0.03340063900613 * \sigma_h - 0.00178687092114 * log(Re_h) + 0.96233612367662 * \sigma_h^2 - 2.55595501972796 * \sigma_h + 1.01310287017856$	Coefficient de pertes de charge en entrée de l'échangeur côté chaud (-)
Ke_c	$Ke_c = -0.00496672650332 * \sigma_c^2 + 0.00113607587171 * \sigma_c - 0.00001379297260 * log^2(Re_c) + 0.06612031387891 * \sigma_c^2 + 0.03340063900613 * \sigma_c - 0.00178687092114 * log(Re_c) + 0.96233612367662 * \sigma_c^2 - 2.55595501972796 * \sigma_c + 1.01310287017856$	Coefficient de pertes de charge en entrée de l'échangeur côté froid (-)
pi_{Bc}	$pi_{Bc} = pe_{oc}$	Pression d'entrée de la buse d'éjection (Pa)
Ti_{Bc}	$Ti_{Bc} = Te_{oc}$	Température d'entrée de la buse d'éjection (K)
To_{Bc}	$To_{Bc} = Ti_{Bc} * (1 + \eta_{NO} * (-1 + (\frac{po_{Bc}}{pi_{Bc}})^{\frac{\gamma-1}{\gamma}}))$	Température de sortie de la buse d'éjection (K)
q_{mav}	$q_{mav} = q - q_{ma}$	Débit d'air passant par la vanne d'équilibrage (kg/s)

ξ_α	$\xi_\alpha = e^{(4.03108e^{-4} * \alpha^2 + 8.50089e^{-2} * \alpha - 1.59295}$	Coefficient de pertes de charge de la vanne (-)
$CPam_h$	$CPam_h = Kc_h + 1 - (\sigma_h^2)$, $CPd_h = 2 * (\frac{\rho_2}{\rho_3} - 1)$	Coefficient de pertes de charge amont côté chaud (-)
CPd_h	$CPd_h == 2 * (\frac{\rho_2}{\rho_3} - 1)$	Coefficient de pertes de charge dynamique côté chaud (-)
CPr_h	$CPr_h = f_h * \frac{A_h}{Ac_h} * (2 * \frac{\rho_2}{\rho_2 + \rho_3})$	Coefficient de pertes de charge régulière côté chaud (-)
$CPav_h$	$CPav_h = (Ke_h + \sigma_h^2 - 1) * \frac{rho2}{rho3}$	Coefficient de pertes de charge aval côté chaud (-)
$CPam_c$	$CPam_c = Kc_c + 1 - \sigma_c^2$, $CPd_c = 2 * (\frac{\rho_{ic}}{\rho_{oc}} - 1)$	Coefficient de pertes de charge amont côté froid (-)
CPd_c	$CPd_c == 2 * (\frac{\rho_{ic}}{\rho_{oc}} - 1)$	Coefficient de pertes de charge dynamique côté froid (-)
CPr_c	$CPr_c = f_c * \frac{A_c}{Ac_c} * (2 * \frac{\rho_{ic}}{\rho_{ic} + \rho_{oc}})$	Coefficient de pertes de charge régulière côté froid (-)
$CPav_c$	$CPav_c = (Ke_c + \sigma_c^2 - 1) * \frac{\rho_{ic}}{\rho_{oc}}$	Coefficient de pertes de charge aval côté froid (-)

Contraintes

Relation	**Description**
$piecewise(Re_h,$ $[200, 1000] : f_h = \frac{J_h}{0.27},$ $[1000, 40000] : f_h = \frac{J_h}{5.39 * 10^{-6} * Re_h + 0.264615},$ $[40000, 1000000] : f_h = \frac{J_h}{0.47})$	Expression du facteur de Fanning pour le côté chaud de l'échangeur
$piecewise(Re_c,$ $[200, 1000] : f_c = \frac{J_c}{0.27},$ $[1000, 40000] : f_c = \frac{J_c}{5.39 * 10^{-6} * Re_c + 0.264615},$ $[40000, 1000000] : f_c = \frac{J_c}{0.47})$	Expression du facteur de Fanning pour le côté froid de l'échangeur
$piecewise(Re_h,$ $[200, 2100] : Nu_h = 1.86 * (\frac{Pr_h * Re_h * 2 * rh_h}{L_x})^{0.33},$ $[2100, 8000] : Nu_h = 0.116 * (Re_h^{0.66} - 125) * Pr_h^{0.33},$ $[8000, 10000] : Nu_h = \frac{10000 - Re_h}{10000 - 8000} * 0.116 *$ $(Re_h^{0.66} - 125) * Pr_h^{0.33} + \frac{Re_h - 8000}{10000 - 8000} * 0.023 *$ $Re_h^{0.8} * Pr_h^{0.33},$	Expression du nombre de Nusselt pour le côté chaud de l'échangeur

$[10000, 1000000] : Nu_h = 0.023 * Re_h^{0.8} * Pr_h^{0.33}$)	
$piecewise(Re_c,$ $[200, 2100] : Nu_c = 1.86 * (\frac{Pr_c * Re_c * 2 * rh_c}{L_x})^{0.33},$ $[2100, 8000] : Nu_c = 0.116 * (Re_c^{0.66} - 125) * Pr_c^{0.33},$ $[8000, 10000] : Nu_c = \frac{10000 - Re_c}{10000 - 8000} * 0.116 *$ $(Re_c^{0.66} - 125) * Pr_c^{0.33} + \frac{Re_c - 8000}{10000 - 8000} * 0.023 *$ $Re_c^{0.8} * Pr_c^{0.33},$ $[10000, 1000000] : Nu_c = 0.023 * Re_c^{0.8} * Pr_c^{0.33}$)	Expression du nombre de Nusselt pour le côté froid de l'échangeur
$\tau = \frac{q_{ra}}{q_{ma}}$	Expression du rapport des débits dynamique et principal
$\epsilon = 1 - e^{\lambda * Nut^{0.22} * (e^{\frac{-Nut^{0.78}}{\lambda}} - 1)}$	Expression de l'efficacité de l'échangeur
$\epsilon = \frac{T_2 - T_3}{T_2 - T_0}$	
$T_3 = T_2 - \epsilon * (T_2 - Te_{ic})$	Bilan d'énergie dans l'échangeur côté chaud
$\Delta Pe_h = p_2 - p_3$	Expression des pertes de charge dans l'échangeur côté chaud
$\Delta Pe_h = \frac{G_h^2}{2 * \rho_2} * (CPam_h + CPd_h + CPr_h + CPav_h)$	
$Te_{oc} = \frac{q_{ma} * Cp_{ih}}{q_{ra} * Cp_{ic}} * (T_2 - T_3) + Te_{ic}$	Bilan d'énergie dans l'échangeur côté froid
$\Delta Pe_c = pe_{ic} - pe_{oc}$	Pertes de charge dans l'échangeur côté froid
$\Delta Pe_c = \frac{G_c^2}{2 * \rho_{ic}} * (CPam_c + CPd_c + CPr_c + CPav_c)$	
$q * Cp * T_5 == q_{ma} * Cp * T_4 + q_{mav} * Cp * T_1$	Bilan d'énergie et de quantité de mouvement dans la vanne d'équilibrage
$p1 - p4 == \xi_\alpha * \nu * \frac{q_{mav}^2}{2 * (\pi * Rv^2)^2}$	

Système de conditionnement d'air avec ailettes (voir V.2.4)

Constantes

Nom	**Description**
$Cp = 1000$	Chaleur spécifique à pression constante de l'air (J/(kg.K))
$r = 287$	Constante des gaz parfaits rapportée à la masse molaire de l'air (J/(kg.K))
$\gamma = 1.4$	Rapport de capacité thermique (-)
$Z = 10500$	Altitude de vol (m)
$M = 0.8$	Vitesse de l'avion (m/s)

$TC_{TR} = 8$	Taux de compression du turbo-réacteur (-)
$\eta_{TRd} = 0.9$	Rendement du diffuseur du turbo-réacteur (-)
$\eta_{TRc} = 0.8$	Rendement du compresseur du turbo-réacteur (-)
$\eta_C = 0.75$	Rendement du compresseur (-)
$\eta_{AT} = 0.95$	Rendement de l'arbre de transmission (-)
$\eta_t = 0.8$	Rendement de la turbine (-)
$\eta_{NO} = 0.9$	Rendement de la buse d'éjection (-)
$\eta_{DI} = 0.9$	Rendement du diffuseur (-)
$T_5 = 278.15$	Température de l'air en sortie du SCA (K)
$p_5 = 85000$	Pression de l'air en sortie du SCA (Pa)
$q = 0.7$	Débit d'air prélevé sur le turbo-réacteur (kg/s)
$kp = 20$	Conductivité des plaques (W/(m.K))
$tp = 0.001$	Épaisseur des plaques (m)
$T_a = 288.2 - 0.00649 * Z$	Température atmosphérique (K)
$p_a = 101290 * (\frac{T_a}{288.08})^{5.256}$	Pression atmosphérique (Pa)
$T_0 = T_a * (1 + \frac{M^2 * (\gamma - 1)}{2})$	Température en sortie du diffuseur du turbo-réacteur (K)
$p_0 = p_a * (\eta_{TRd} * \frac{M^2 * (\gamma - 1)}{2} + 1)^{\frac{\gamma}{\gamma - 1}}$	Pression en sortie du diffuseur du turbo-réacteur (Pa)
$T_1 = T_0 * (1 + \frac{1}{\eta_{TRc}} * ((\frac{p1}{p0})^{\frac{\gamma - 1}{\gamma}} - 1))$	Température en entrée du SCA (K)
$p_1 = TC_{TR} * p_0$	Pression en entrée du SCA (Pa)
$pi_{Dc} = p_a$	Pression en entrée du diffuseur (Pa)
$Ti_{Dc} = T_a$	Température en entrée du diffuseur (K)
$po_{Dc} = pi_{Dc} * (\eta_{DI} * M^2 * \frac{\gamma - 1}{2} + 1)^{\frac{\gamma}{\gamma - 1}}$	Pression en sortie du diffuseur (Pa)
$To_{Dc} = Ti_{Dc} * (1 + M^2 * \frac{\gamma - 1}{2})$	Température en sortie du diffuseur (K)
$po_{Bc} = p_a$	Pression en sortie de la buse d'éjection (Pa)
$\nu = \frac{r * T_1}{p_1}$	Volume massique (m^3/kg)
$p_4 = p_5$	Pression en sortie de la turbine (Pa)

Variables de conception

Nom	Précision	Description
$L_x : real = [0.01, 0.1]$	0.01	Largeur de l'échangeur thermique (m)
$\tau : real = [0.8, 2]$	0.01	Rapport de débits (-)

$exSurf_h : int = [1, 48]$	-	Surfaces d'échange pour le côté chaud (-)
$exSurf_c : int = [1, 48]$	-	Surfaces d'échange pour le côté froid (-)
$Rv : real = \{0.01, 0.02, ..., 0.2\}$	-	Rayon de la vanne d'équilibrage (m)

Variables auxiliaires

Nom	Précision	Description
$p_2 : real = [0, 10^6]$	5%	Pression en entrée de l'échangeur côté chaud (Pa)
$p_3 : real = [0, 10^6]$	5%	Pression en sortie de l'échangeur côté chaud (Pa)
$p_{eic} : real = [0, 10^6]$	5%	Pression en entrée de l'échangeur côté froid (Pa)
$p_{eoc} : real = [0, 10^6]$	5%	Pression en sortie de l'échangeur côté froid (Pa)
$T_2 : real = [0, 1000]$	5%	Température en entrée de l'échangeur côté chaud (K)
$T_3 : real = [0, 1000]$	5%	Température en sortie de l'échangeur côté chaud (K)
$T_{eic} : real = [0, 1000]$	5%	Température en entrée de l'échangeur côté froid (K)
$T_{eoc} : real = [0, 1000]$	5%	Température en sortie de l'échangeur côté froid (K)
$\Delta P_{eh} : real = [-\infty, +\infty]$	10%	Pertes de charge dans l'échangeur côté chaud (Pa)
$\Delta P_{ec} : real = [-\infty, +\infty]$	10%	Pertes de charge dans l'échangeur côté froid (Pa)
$\epsilon : real = [0, 1]$	5%	Efficacité de l'échangeur (-)
$q_{ra} : real = [0, +\infty]$	0.001	Débit d'air côté froid (kg/s)
$q_{ma} : real = [0, +\infty]$	0.001	Débit d'air côté chaud (kg/s)
$p_4 : real = [0, 10^6]$	5%	Pression en sortie de turbine (Pa)
$T_4 : real = [0, 1000]$	5%	Température en sortie de turbine (K)
$\alpha : int = [10, 80]$	-	Angle d'ouverture de la vanne d'équilibrage (°)
$\rho_{ex} : int = [1000, 10000]$	-	Masse volumique des ailettes (kg/m^3)
$kw : int = [10, 1000]$	-	Conductivité d'ailette (W/(m.K))

$b_h, rh_h, \delta_h, \beta_h : real = [-\infty, +\infty]$	∞	Caractéristiques des ailettes côté chaud
$b_c, rh_c, \delta_c, \beta_c : real = [-\infty, +\infty]$	∞	Caractéristiques des ailettes côté froid
$JS_{h1}, JS_{h2}, JS_{h3} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation du nombre de Colburn côté chaud (-)
$JS_{c1}, JS_{c2}, JS_{c3} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation du nombre de Colburn côté froid (-)
$fS_{h1}, fS_{h2}, fS_{h3} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation du facteur de Fanning côté chaud (-)
$fS_{c1}, fS_{c2}, fS_{c3} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation du facteur de Fanning côté froid (-)
$KcS_{h1}, KcS_{h2}, KcS_{h3}, KcS_{h4}, KcS_{h5}, KcS_{h6}, KcS_{h7}, KcS_{h8}, KcS_{h9} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation pour le calcul de Kc côté chaud (-)
$KcS_{c1}, KcS_{c2}, KcS_{c3}, KcS_{c4}, KcS_{c5}, KcS_{c6}, KcS_{c7}, KcS_{c8}, KcS_{c9} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation pour le calcul de Kc côté froid (-)
$KeS_{h1}, KeS_{h2}, KeS_{h3}, KeS_{h4}, KeS_{h5}, KeS_{h6}, KeS_{h7}, KeS_{h8}, KeS_{h9} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation pour le calcul de Ke côté chaud (-)
$KeS_{c1}, KeS_{c2}, KeS_{c3}, KeS_{c4}, KeS_{c5}, KeS_{c6}, KeS_{c7}, KeS_{c8}, KeS_{c9} : real = [-\infty, +\infty]$	∞	Coefficients d'interpolation pour le calcul de Ke côté froid (-)

Variables alias

Nom	Contrainte	Description
L_y	$L_y = L_x$	Longueur de l'échangeur (m)
L_z	$L_z = 0.25 * L_x$	Hauteur de l'échangeur (m)
α_h	$\alpha_h = \frac{(b_h * \beta_h)}{(b_h + b_c + 2 * \delta_h)}$	Compacité côté chaud (m^{-1})
α_c	$\alpha_c = \frac{(b_c * \beta_c)}{(b_h + b_c + 2 * \delta_c)}$	Compacité côté froid (m^{-1})
σ_h	$\sigma_h = \alpha_h * rh_h$	Porosité côté chaud (-)
σ_c	$\sigma_c = \alpha_c * rh_c$	Porosité côté froid (-)
A_w	$A_w = L_x * L_y * n$	Surface d'échange des plaques (m^2)
A_h	$A_h = \alpha_h * L_x * L_y * L_z$	Surface d'échange total des ailettes côté chaud (m^2)
A_c	$A_c = \alpha_c * L_x * L_y * L_z$	Surface d'échange côté froid (m^2)

Ac_h	$Ac_h = \sigma_h * Af_h$	Surface frontale côté chaud (m^2)
Ac_c	$Ac_c = \sigma_c * Af_c$	Surface frontale côté froid (m^2)
Af_h	$Af_h = L_x * L_z$	Section frontale côté chaud (m^2)
Af_c	$Af_c = L_y * L_z$	Section frontale côté froid (m^2)
$\eta 0_h$	$\eta 0_h = 1 - \frac{Af_h}{A_h} * (1 - \eta f_h)$	Efficacité de la surface d'échange côté chaud (-)
$\eta 0_c$	$\eta 0_c = 1 - \frac{Af_c}{A_c} * (1 - \eta f_c)$	Efficacité de la surface d'échange côté froid (-)
n	$n = \frac{Lz}{b_h + b_c + 2 * \delta_h}$	Nombre de plaques (-)
G_h	$G_h = \frac{q_{ma}}{Ac_h}$	Vitesse massique côté chaud ($kg/(s.m^2)$)
G_c	$G_c = \frac{q_{ra}}{Ac_c}$	Vitesse massique côté froid ($kg/(s.m^2)$)
μ_h	$\mu_h = \frac{-2.15 * 10^{-5} - 2.225 * 10^{-9} * (T_3 + T_2)}{2} + \frac{1.725 * 10^{-6} * (\sqrt{T_3} + \sqrt{T_2})}{2}$	Viscosité de l'air côté chaud (kg/(m.s))
μ_c	$\mu_c = \frac{-2.15 * 10^{-5} - 2.225 * 10^{-9} * (Te_{oc} + Te_{ic})}{2} + \frac{1.725 * 10^{-6} * (\sqrt{Te_{oc}} + \sqrt{Te_{ic}})}{2}$	Viscosité de l'air côté froid (kg/(m.s))
ρ_2	$\rho_2 = \frac{p_2}{r * T_2}$	Masse volumique de l'air en entrée de l'échangeur côté chaud (kg/m^3)
ρ_3	$\rho_3 = \frac{p_3}{r * T_3}$	Masse volumique de l'air en sortie de l'échangeur côté chaud (kg/m^3)
ρ_{ic}	$\rho_{ic} = \frac{pe_{ic}}{r * Te_{ic}}$	Masse volumique de l'air en entrée de l'échangeur côté froid (kg/m^3)
ρ_{oc}	$\rho_{oc} = \frac{pe_{oc}}{r * Te_{oc}}$	Masse volumique de l'air en sortie de l'échangeur côté froid (kg/m^3)
Cp_{oh}	$Cp_{oh} = 1003.7 + 6.8 * 10^{-2} * (T_3 - 273.15) + 2.22 * 10^{-4} * (T_3 - 273.15)^2$	Capacité calorifique de l'air en sortie de l'échangeur côté chaud (J/(kg.K))
Cp_{ih}	$Cp_{ih} = 1003.7 + 6.8 * 10^{-2} * (T_2 - 273.15) + 2.22 * 10^{-4} * (T_2 - 273.15)^2$	Capacité calorifique de l'air en entrée de l'échangeur côté

		chaud (J/(kg.K))
Cp_h	$Cp_h = \frac{Cp_{oh}+Cp_{ih}}{2}$	Capacité calorifique moyenne côté chaud (J/(kg.K))
Cp_{oc}	$Cp_{oc} = 1003.7 + 6.8 * 10^{-2} * (Te_{oc} - 273.15) + 2.22 * 10^{-4} * (Te_{oc} - 273.15)^2$	Capacité calorifique de l'air en sortie de l'échangeur côté froid
Cp_{ic}	$Cp_{ic} = 1003.7 + 6.8 * 10^{-2} * (Te_{ic} - 273.15) + 2.22 * 10^{-4} * (Te_{ic} - 273.15)^2$	Capacité calorifique de l'air en entrée de l'échangeur côté froid (J/(kg.K))
Cp_c	$Cpc = \frac{Cp_{oc}+Cp_{ic}}{2}$	Capacité calorifique moyenne côté froid (J/(kg.K))
m_{ex}	$m_{ex} = \rho_{ex} * Lx * Ly * ((\frac{\sigma_h * \delta_h}{rh_h} + \frac{\sigma_c * \delta_c}{rh_c}) * Lz + n * tw)$	Masse de l'échangeur (kg)
V	$V = L_x * L_y * L_z$	Volume de l'échangeur (m^3)
V_p	$V_p = \frac{A}{2} * t_p$	Volume des plaques de l'échangeur (m^3)
C_h	$C_h = \frac{q_{ma}}{Ac_h * \rho_2}$	Vitesse de l'air côté chaud (m/s)
C_c	$C_c = \frac{q_{ra}}{Ac_c * \rho_{ic}}$	Vitesse de l'air côté froid (m/s)
λ	$\lambda = \frac{q_{ra} * Cp_{ic}}{q_{ma} * Cp_{ih}}$	Rapport de débits de capacités calorifiques (-)
Nut	$Nut = \frac{q_{ma} * Cp_{ih}}{\frac{1}{\eta 0_h * h_h * A_h} + \frac{tw}{kw * Aw} + \frac{1}{\eta 0_c * h_c * A_c}}$	Nombre d'unités de transfert de l'échangeur (-)
h_h	$h_h = J_h * G_h * Cp_h * Pr_h^{-2/3}$	Coefficient de transfert convectif côté chaud (W/(m2.K))
h_c	$h_c = J_c * G_c * Cp_c * Pr_c^{-2/3}$	Coefficient de transfert convectif côté froid (W/(m2.K))
J_h	$J_h = e^{JS_{h1} * log^2(Re_h) + JS_{h2} * log(Re_h) + JS_{h3}}$	Coefficient de Colburn côté chaud (-)
J_c	$J_c = e^{JS_{c1} * log^2(Re_c) + JS_{h2} * log(Re_c) + JS_{h3}}$	Coefficient de Colburn côté froid (-)
f_h	$f_h = e^{fS_{h1} * log^2(Re_h) + fS_{h2} * log(Re_h) + fS_{h3}}$	Facteur de Fanning côté chaud (-)
f_c	$f_c = e^{fS_{c1} * log^2(Re_c) + fS_{h2} * log(Re_c) + fS_{h3}}$	Facteur de Fanning côté froid (-)

Pr_h	$Pr_h = \frac{1}{2} * (1.65 - 5.4 * 10^{-4} * (T_3 + T_2) + 5.0 * 10^{-7} * (T_3^2 + T_2^2))$	Nombre de Prandtl côté chaud (-)
Pr_c	$Pr_c = \frac{1}{2} * (1.65 - 5.4 * 10^{-4} * (Te_{oc} + Te_{ic}) + 5.0 * 10^{-7} * (Te_{oc}^2 + Te_{ic}^2))$	Nombre de Prandtl côté froid (-)
Re_h	$Re_h = \frac{4 * rh_h * G_h}{\mu_h}$	Nombre de Reynolds côté chaud (-)
Re_c	$Re_c = \frac{4 * rh_c * G_c}{\mu_c}$	Nombre de Reynolds côté froid (-)
Kc_h	$Kc_h = KcS_{h1} * \sigma_h^2 + KcS_{h2} * \sigma_h + KcS_{h3} * log^2(Re_h) + KcS_{h4} * \sigma_h^2 + KcS_{h5} * \sigma_h + KcS_{h6} * log(Re_h) + KcS_{h7} * \sigma_h^2 + KcS_{h8} * \sigma_h + KcS_{h9}$	Coefficient de pertes de charge en entrée de l'échangeur côté chaud (-)
Kc_c	$Kc_c = KcS_{c1} * \sigma_c^2 + KcS_{c2} * \sigma_c + KcS_{c3} * log^2(Re_c) + KcS_{c4} * \sigma_c^2 + KcS_{c5} * \sigma_c + KcS_{c6} * log(Re_c) + KcS_{c7} * \sigma_c^2 + KcS_{c8} * \sigma_c + KcS_{c9}$	Coefficient de pertes de charge en entrée de l'échangeur côté froid (-)
Ke_h	$Ke_h = KeS_{h1} * \sigma_h^2 + KeS_{h2} * \sigma_h + KeS_{h3} * log^2(Re_h) + KeS_{h4} * \sigma_h^2 + KeS_{h5} * \sigma_h + KeS_{h6} * log(Re_h) + KeS_{h7} * \sigma_h^2 + KeS_{h8} * \sigma_h + KeS_{h9}$	Coefficient de pertes de charge en sortie de l'échangeur côté chaud (-)
Ke_c	$Ke_c = KeS_{c1} * \sigma_c^2 + KeS_{c2} * \sigma_c + KeS_{c3} * log^2(Re_c) + KeS_{c4} * \sigma_c^2 + KeS_{c5} * \sigma_c + KeS_{c6} * log(Re_c) + KeS_{c7} * \sigma_c^2 + KeS_{c8} * \sigma_c + KeS_{c9}$	Coefficient de pertes de charge en sortie de l'échangeur côté froid (-)
pi_{Bc}	$pi_{Bc} = pe_{oc}$	Pression d'entrée de la buse d'éjection (Pa)
Ti_{Bc}	$Ti_{Bc} = Te_{oc}$	Température d'entrée de la buse d'éjection (K)
To_{Bc}	$To_{Bc} = Ti_{Bc} * (1 + \eta_{NO} * (-1 + (\frac{po_{Bc}}{pi_{Bc}})^{\frac{\gamma-1}{\gamma}}))$	Température de sortie de la buse d'éjection (K)
q_{mav}	$q_{mav} = q - q_{ma}$	Débit d'air passant par la vanne d'équilibrage (kg/s)
ξ_α	$\xi_\alpha = e^{(4.03108e^{-4} * \alpha^2 + 8.50089e^{-2} * \alpha - 1.59295)}$	Coefficient de pertes de charge de la vanne (-)
$CPam_h$	$CPam_h = Kc_h + 1 - (\sigma_h^2)$, $CPd_h = 2 * (\frac{\rho_2}{\rho_3} - 1)$	Coefficient de pertes de charge amont côté chaud (-)
CPd_h	$CPd_h == 2 * (\frac{\rho_2}{\rho_3} - 1)$	Coefficient de pertes de charge dynamique côté chaud (-)
CPr_h	$CPr_h = f_h * \frac{A_h}{Ac_h} * (2 * \frac{\rho_2}{\rho_2 + \rho_3})$	Coefficient de pertes de charge régulière côté chaud

		(-)
$CPav_h$	$CPav_h = (Ke_h + \sigma_h^2 - 1) * \frac{rho2}{rho3}$	Coefficient de pertes de charge aval côté chaud (-)
$CPam_c$	$CPam_c = Kc_c + 1 - \sigma_c^2$, $CPd_c = 2 * (\frac{\rho_{ic}}{\rho_{oc}} - 1)$	Coefficient de pertes de charge amont côté froid (-)
CPd_c	$CPd_c == 2 * (\frac{\rho_{ic}}{\rho_{oc}} - 1)$	Coefficient de pertes de charge dynamique côté froid (-)
CPr_c	$CPr_c = f_c * \frac{A_c}{Ac_c} * (2 * \frac{\rho_{ic}}{\rho_{ic}+\rho_{oc}})$	Coefficient de pertes de charge régulière côté froid (-)
$CPav_c$	$CPav_c = (Ke_c + \sigma_c^2 - 1) * \frac{\rho_{ic}}{\rho_{oc}}$	Coefficient de pertes de charge aval côté froid (-)
m_h	$m_h = \sqrt{\frac{2*h_h}{kw*\delta_h}}$	(m^{-1})
m_c	$m_c = \sqrt{\frac{2*h_c}{kw*\delta_c}}$	(m^{-1})
ml_h	$ml_h = \frac{m_h*b_h}{2}$	Efficacité d'ailette côté chaud (-)
ml_c	$ml_c = \frac{m_c*b_c}{2}$	Efficacité d'ailette côté froid (-)

Contraintes

Relation	**Description**
$Piecewise(ExSurf_h,$ $[1,1] : JS_{h1} = 0.0314 \wedge JS_{h2} = -0.8534 \wedge \ldots$ $\vdots$ $[48,48] : JS_{h1} = 0.0369 \wedge JS_{h2} = -0.8995 \wedge \ldots$	Catalogue d'ailettes pour le côté chaud
$Piecewise(ExSurf_c,$ $[1,1] : JS_{c1} = 0.0314 \wedge JS_{c2} = -0.8534 \wedge \ldots$ $\vdots$ $[48,48] : JS_{c1} = 0.0369 \wedge JS_{c2} = -0.8995 \wedge \ldots$	Catalogue d'ailettes pour le côté froid
$\tau = \frac{q_{ra}}{q_{ma}}$	Expression du rapport des débits dynamique et principal
$\epsilon = 1 - e^{\lambda * Nut^{0.22} * (e^{\frac{-Nut^{0.78}}{\lambda}} - 1)}$	Expression de l'efficacité de l'échangeur
$\epsilon = \frac{T_2 - T_3}{T_2 - T_0}$	
$T_3 = T_2 - \epsilon * (T_2 - Te_{ic})$	Bilan d'énergie dans l'échangeur côté chaud
$\Delta Pe_h = p_2 - p_3$	Expression des pertes de charge

$\Delta Pe_h = \frac{G_h^2}{2*\rho_2} * (CPam_h + CPd_h + CPr_h + CPav_h)$	dans l'échangeur côté chaud
$Te_{oc} = \frac{q_{ma}*Cp_{ih}}{q_{ra}*Cp_{ic}} * (T_2 - T_3) + Te_{ic}$	Bilan d'énergie dans l'échangeur côté froid
$\Delta Pe_c = pe_{ic} - pe_{oc}$	Pertes de charge dans l'échangeur côté froid
$\Delta Pe_c = \frac{G_c^2}{2*\rho_{ic}} * (CPam_c + CPd_c + CPr_c + CPav_c)$	
$q * Cp * T_5 == q_{ma} * Cp * T_4 + q_{mav} * Cp * T_1$	Bilan d'énergie et de quantité de mouvement dans la vanne d'équilibrage
$p1 - p4 == \xi_\alpha * \nu * \frac{q_{mav}^2}{2*(\pi*Rv^2)^2}$	

B – Organisation des classes dans la librairie Elisa

La librairie Elisa a pour objectif de fournir un ensemble de classes permettant de définir des applications faisant appel à une résolution utilisant des techniques de satisfaction de contraintes sur les domaines continus. Des variables discrètes peuvent être définies, leur domaine est alors implémenté par un intervalle raffiné au fur et à mesure, en utilisant les valeurs discrètes du domaine initial. L'organisation des classes d'Elisa permet de définir facilement de nouvelles stratégies ou de nouveaux concepts à plusieurs endroits clés au sein d'un solveur CSP.

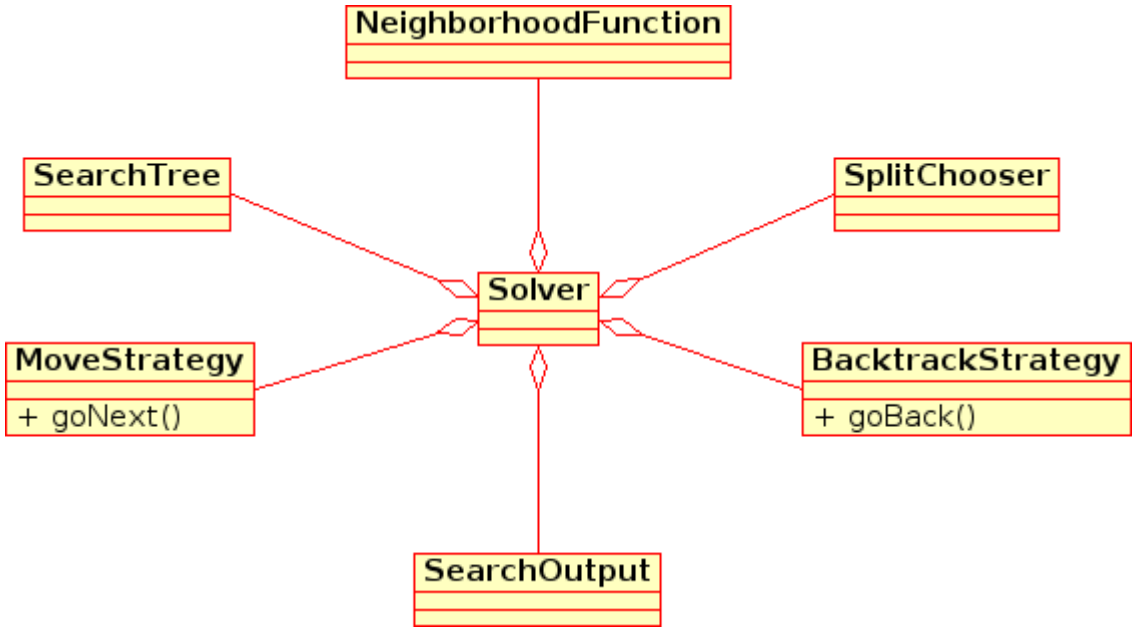


Figure 39 : Exemples de classes d'objets composant la classe solver d'Elisa représentée suivant la notation UML.

La figure 1 présente quelques classes liées au solver défini dans Elisa. La classe `SearchTree` correspond à l'arbre de recherche, qui est généré pour chaque problème traité. La classe `NeighborhoodFunction` définit la manière dont les domaines des variables, choisies par le `SpliChooser`, sont explorés. La classe `BacktrackStrategy` définit la stratégie de retour en arrière dans l'arbre de recherche, lorsqu'une inconsistance est détectée ou une solution trouvée. La classe `MoveStrategy` définit comment explorer l'arbre de recherche. Enfin, la classe `SearchOutput` permet de définir l'export des solutions. Ces classes d'objets peuvent être spécialisées pour définir des stratégies spécifiques à chacun de ces concepts.

La classe `Solver` s'applique à des objets de type `Model`. La figure 2 définit la hiérarchie des types de modèles existants. Un objet de la classe `Model` a pour représentation un objet de type `ModelRep`. Cette dernière classe peut se spécialiser soit en `ConstraintHandler` pour encapsuler des contraintes, soit en `CompoundModel`, puis `AndModel`, qui permet ainsi de représenter des expressions correspondant à des conjonctions de contraintes. Il est possible de spécialiser la classe `CompoundModel` pour définir, par exemple, une nouvelle classe `OrModel` qui représenterait une disjonction de contraintes.

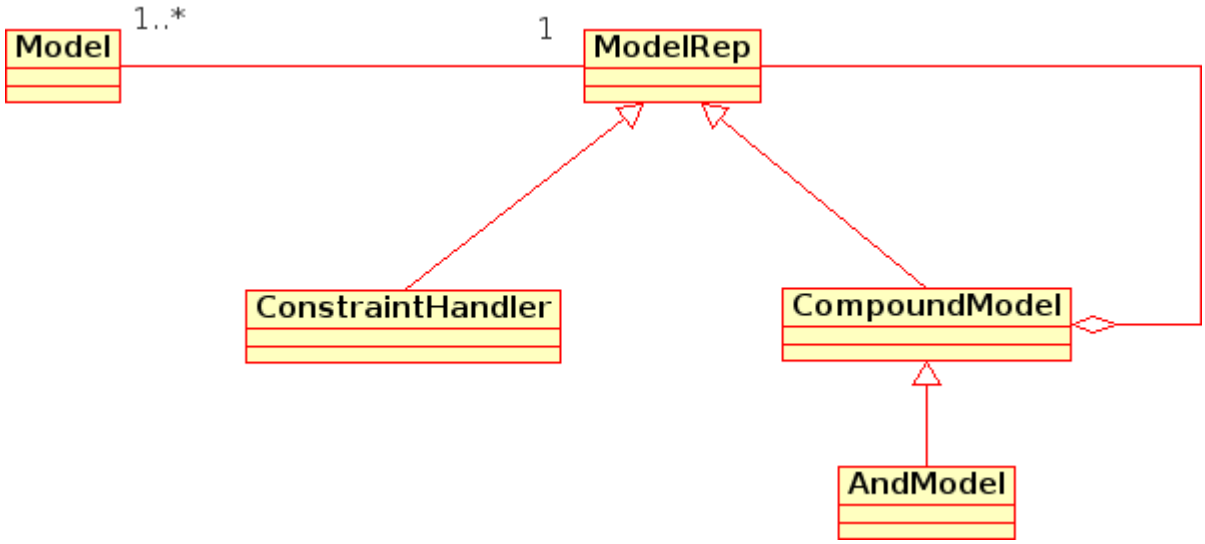


Figure 40 : Diagramme de classe UML de la représentation d'un modèle au sein d'Elisa.

Les contraintes, encapsulées dans les objets de la classe `ConstraintHandler`, sont définies uniquement sur les variables réelles, comme le montre la figure 3. Cependant, la classe `Constraint` peut être spécialisée pour définir de nouveaux types de contraintes. La représentation des contraintes dissocie la partie représentation symbolique de la partie liée à l'évaluation des expressions mathématiques, qui sont alors, soit des égalités, soit des inégalités (<= ou >=). Ainsi, chaque objet de type `Constraint` comporte une `ConstraintRep` définissant son type (`RealConstraintRepEq`, `RealConstraintRepSup`, `RealConstraintRepInf`).

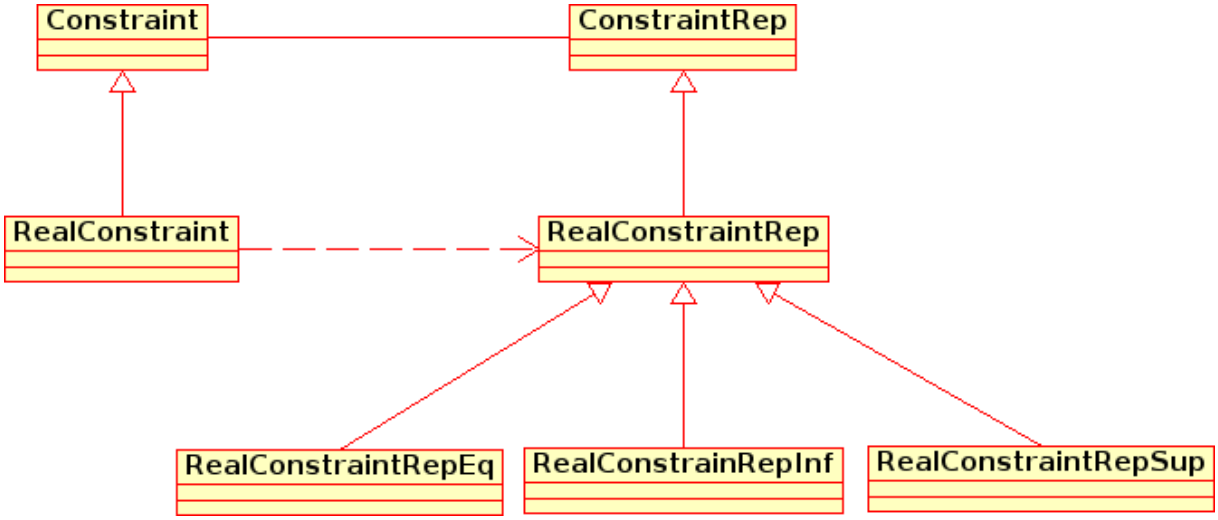


Figure 41 : Diagramme de classe UML de la représentation d'une contrainte au sein d'Elisa.

La représentation des variables, composant les expressions arithmétiques des contraintes, dissocie aussi le symbole de la variable et sa implémentation au sein de l'arbre de recherche lors de la résolution (voir figure 4). On peut alors noter qu'une variable entière de type `IntVar` est en fait une variable réelle spécialisée, ce qui permet de définir des contraintes « mixtes » au travers des objets de type `RealConstraint`.

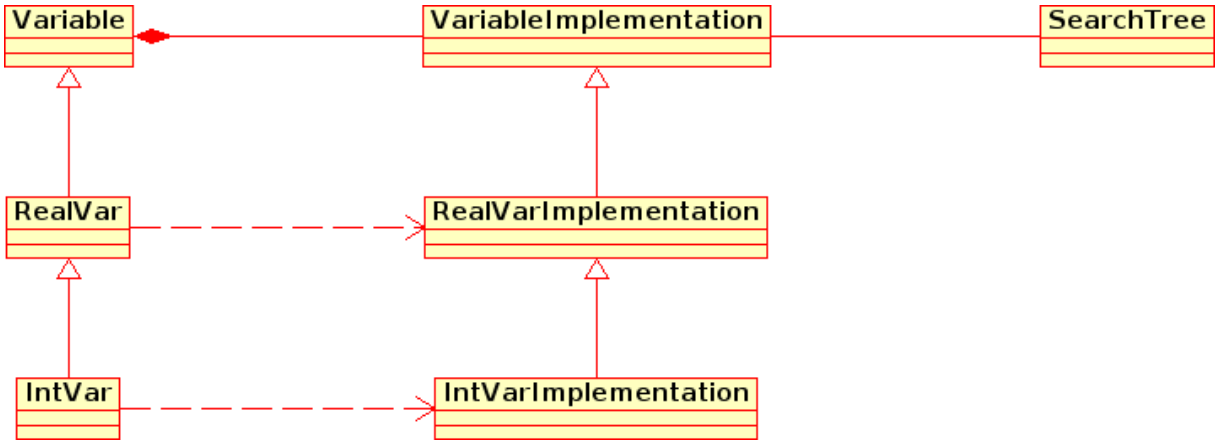


Figure 42 : Diagramme de classe UML de la représentation d'une variable au sein d'Elisa.

La classe `Solver` comporte aussi un ensemble de réductions permettant d'appliquer des opérateurs de réduction définis à partir des contraintes du modèle (voir figure 5). La classe `FixedPointReduction` permet de définir un ensemble d'opérateurs qui sont appliqués selon la méthode du point fixe. Les opérateurs de réduction issus directement des contraintes sont les objets de type `HullReduction`, appliquant la *hull*-consistance, les objets de type `BoxReduction`, appliquant la box-consistance et les objets de type

`NewtonReduction`, appliquant une algorithme de type Newton adapté aux intervalles.

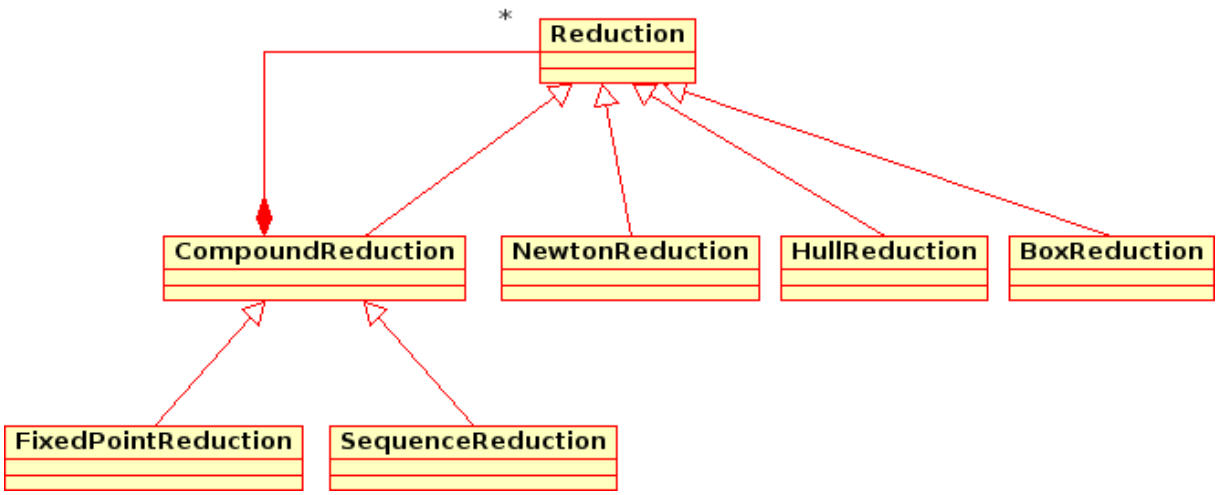


Figure 43 : Diagramme de classe UML de la représentation d'une réduction au sein d'Elisa.

La classe `SplitChooser` permet de relier un ensemble de variables de type `SetVariables` et une stratégie de sélection de ces variables pour le découpage de leur domaine. Les heuristiques de recherche sont alors définies en spécialisant la classe `ChoiceStrategy`. La classe `SplitChooser` définit aussi la manière dont les domaines sont découpés.

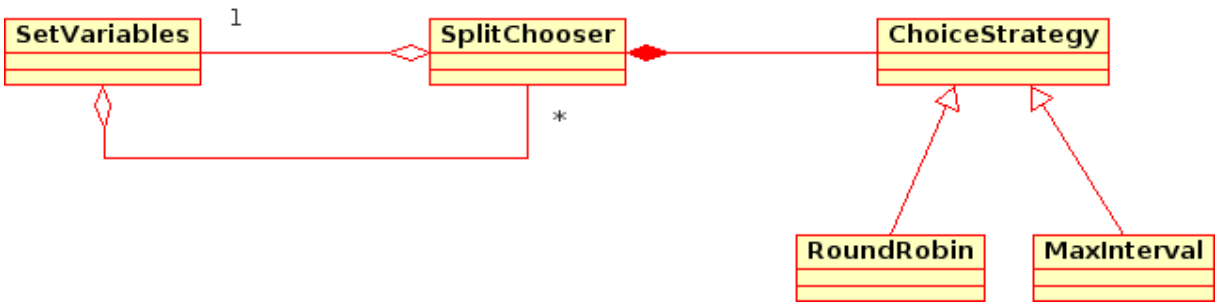


Figure 44 : Diagramme de classe UML de la représentation de la stratégie de choix de variables au sein d'Elisa.

RESOLUTION PAR SATISFACTION DE CONTRAINTES APPLIQUEE A L'AIDE À LA DECISION EN CONCEPTION ARCHITECTURALE

RESUME: La conception architecturale de systèmes mécaniques et énergétiques permet d'étudier la faisabilité de concepts préalablement choisis pendant la phase de recherche de concepts. Elle fait partie de la phase de conception préliminaire et son achèvement vise à définir l'architecture globale qui sera étudiée pendant la phase de conception détaillée d'un produit. Les principales caractéristiques d'un produit (dimensions, choix de composants, de formes, de topologies ou de matériaux) sont alors fondamentales à ce stade de la conception. La conception architecturale nécessite aussi la prise en compte des comportements physiques et des interactions du produit, de ses composants et des milieux extérieurs, de critères économiques, environnementaux, etc. suivant les exigences du cahier des charges fonctionnel. Le concepteur définit un modèle mathématique exprimant cette connaissance liée au produit.
Ce modèle peut être traité à l'aide d'un solveur de Problèmes de Satisfaction de Contraintes numériques (CSP). Ce type de solveur traite de manière générique des problèmes formulés à l'aide de contraintes, de variables et de domaines. Nous proposons de nouveaux algorithmes et de nouvelles heuristiques qui prennent en compte des besoins spécifiques au concepteur en conception préliminaire, comme : la classification des variables, les précisions relatives aux valeurs de certaines variables, les contraintes par morceaux. Les études de faisabilité de plusieurs systèmes mécaniques et énergétiques ont été réalisées avec succès. Les nouveaux algorithmes ont permis de diminuer significativement les temps de calculs, mais ils ont aussi permis d'améliorer la qualité des solutions calculées au regard des besoins du concepteur.

Mots-clés: *Programmation par contraintes, conception architecturale, conception préliminaire, CSP, analyse par intervalles, heuristique de recherche, variable de conception*

CONSTRAINT SATISFACTION SOLVING APPLIED TO DECISION SUPPORT FOR EMBODIMENT DESIGN

ABSTRACT: The embodiment and conceptual design phases of mechanical and energetic systems are concerned with preliminary design. The embodiment design phase aims at investigating the feasibility of concepts emerging from conceptual design. Embodiment achieves to make a decision concerning the main features of the architecture of a product, which will be detailed in the next design phase. The main characteristics of the product are relating to dimensions, choices of components, shapes, topologies or materials. Decisions can better be supported by carrying out models which mainly take into consideration functional specifications and physical behaviors, namely a mathematical definition of the designer knowledge.
These models may be processed using a numerical Constraint Satisfaction Problem (CSP) solver. CSP solvers are based on generic algorithms suitable for the numerical treatment of problems regarded as sets of constraints, variables and domains. Through these models, designers aim at investigating design architectures, i.e. they have to match up values with the main variables of the problem (design variables and decision variables). We propose several new heuristics and algorithms taking into account some specific difficulties relating to preliminary design problems. This approach concerns the classification and relative precision of variables and piecewise constraints. The feasibility studies of several mechanical and energetic systems have been investigated. These new algorithms drastically decrease the calculation time of the numerical treatment, and also improves the quality of the computational results.

Keywords: *Constraint programming, embodiment design, preliminary design, CSP, interval analysis, search heuristic, design variable*