



HAL
open science

Contribution à l'optimisation multiobjectif en conception multidisciplinaire

Ouael Mouelhi

► **To cite this version:**

Ouael Mouelhi. Contribution à l'optimisation multiobjectif en conception multidisciplinaire. Automatique / Robotique. INSA de Lyon, 2010. Français. NNT : 2010ISAL0022 . tel-00824460

HAL Id: tel-00824460

<https://theses.hal.science/tel-00824460>

Submitted on 21 May 2013

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.

THÈSE

Présentée devant

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

Pour obtenir

LE GRADE DE DOCTEUR

ÉCOLE DOCTORALE : ÉLECTRONIQUE, ÉLECTROTECHNIQUE,
AUTOMATIQUE

Par

MOUELHI Ouael

**Contribution à l'optimisation multiobjectif en
conception multidisciplinaire**

Soutenue le 17 mars 2010

Jury

M. Bideaux	Président du jury	Professeur, AMPERE, INSA de Lyon
M. Chapurlat	Examineur	Professeur, LGI2P, Ecole des Mines d'Ales
M. Couturier	Codirecteur de Thèse	Maître Assisant, LGI2P, Ecole des Mines d'Ales
M. Gandibleux	Rapporteur	Professeur, LINA, Université de Nantes
M. Lutz	Rapporteur	Professeur, FEMTO, Université de Franche-Comté
M. Redarce	Directeur de Thèse	Professeur, AMPERE, INSA de Lyon
M. Wippler	Membre Invité	Responsable technique, Ingénierie Système

Remerciements

Les travaux présentés dans ce mémoire ont été menés au laboratoire Ampère de l'INSA de Lyon et au Laboratoire de Génie Informatique et d'Ingénierie de Production (LGI2P) de l'Ecole des Mines d'Alès.

J'exprime une profonde gratitude à M. Laurent Nicolas et M. Michel Vasquez pour m'avoir accueilli au sein de leurs établissements respectifs.

Je voudrais remercier les personnes qui ont honoré mon jury de thèse. Cela a été une grande fierté pour moi que M. Eric Bideaux préside ce jury.

Je remercie M. Xavier Gandibleux et M. Philippe Lutz de m'avoir fait l'honneur d'être les rapporteurs de cette thèse. Leurs remarques critiques, justes et avisées et leurs commentaires très pertinents durant la soutenance m'ont permis de clarifier certains points de ce manuscrit.

Merci à M. Vincent Chapurlat et M. Jean-Luc Wippler d'avoir accepté d'être examinateurs de cette thèse, et d'associer à ce travail leurs expertises respectives. J'éprouve un profond respect pour leur travail et leur parcours, et surtout pour leurs relations humaines.

Je tiens à remercier tout particulièrement M. Tanneguy Redarce, mon directeur -sans qui ce projet n'aurait surement pas vu le jour- pour avoir supervisé mes travaux tout au long de ces trois années.

C'est envers M. Pierre Couturier, mon co-directeur de thèse, que je suis le plus redevable. Je ne le remercierai jamais assez pour sa présence, sa patience, ses remarques et conseils aussi subtils que constructifs. Qu'il trouve ici le témoignage de ma plus grande gratitude.

Que l'ensemble des chercheurs et doctorants du LGI2P, ainsi que son personnel, soient aussi remerciés pour l'ambiance agréable qu'ils contribuent à maintenir au sein du laboratoire. Merci à ceux que je croisais quotidiennement et qui s'attardaient pour échanger quelques mots, notamment à la salle café.

Je remercie tout particulièrement Michel, Jacky, Françoise, Sami, Abdelhak, Gladys, Fady, Nicolas... pour leur amitié. Sans oublier les anciens : Saber, Kamel et Désiré.

Je remercie infiniment mes parents qui ont été présents, à chaque instant. Leur irremplaçable et inconditionnel soutien m'a permis d'écarter les doutes, de soigner les blessures et de partager mes joies. Cette thèse est un peu la leur, aussi... J'espère qu'ils seront toujours là chaque fois que j'aurai besoin d'eux.

« At last but not least », je remercie Chourouk qui a su me supporter, m'encourager et me prodiguer les bons conseils pendant les moments les plus difficiles de cette thèse.

INSA Direction de la Recherche - Ecoles Doctorales – Quadriennal 2007-2010

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	<p>CHIMIE DE LYON http://sakura.cpe.fr/ED206</p> <p>M. Jean Marc LANCELIN</p> <p>Insa : R. GOURDON</p>	<p>M. Jean Marc LANCELIN Université Claude Bernard Lyon 1 Bât CPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 Fax : lancelin@hikari.cpe.fr</p>
E.E.A.	<p>ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE http://www.insa-lyon.fr/eea</p> <p>M. Alain NICOLAS Insa : C. PLOSSU ede2a@insa-lyon.fr Secrétariat : M. LABOUNE AM. 64.43 – Fax : 64.54</p>	<p>M. Alain NICOLAS Ecole Centrale de Lyon Bâtiment H9 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60 97 Fax : 04 78 43 37 17 eea@ec-lyon.fr Secrétariat : M.C. HAVGOUDOUKIAN</p>
E2M2	<p>EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION http://biomserv.univ-lyon1.fr/E2M2</p> <p>M. Jean-Pierre FLANDROIS Insa : H. CHARLES</p>	<p>M. Jean-Pierre FLANDROIS CNRS UMR 5558 Université Claude Bernard Lyon 1 Bât G. Mendel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cédex Tél : 04.26 23 59 50 Fax 04 26 23 59 49 06 07 53 89 13 e2m2@biomserv.univ-lyon1.fr</p>
EDISS	<p>INTERDISCIPLINAIRE SCIENCES- SANTE</p> <p>Sec : Safia Boudjema M. Didier REVEL Insa : M. LAGARDE</p>	<p>M. Didier REVEL Hôpital Cardiologique de Lyon Bâtiment Central 28 Avenue Doyen Lépine 69500 BRON Tél : 04.72.68 49 09 Fax :04 72 35 49 16 Didier.revel@creatis.uni-lyon1.fr</p>
INFOMATHS	<p>INFORMATIQUE ET MATHEMATIQUES http://infomaths.univ-lyon1.fr</p> <p>M. Alain MILLE Secrétariat : C. DAYEYAN</p>	<p>M. Alain MILLE Université Claude Bernard Lyon 1 LIRIS - INFOMATHS Bâtiment Nautilus 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72. 44 82 94 Fax 04 72 43 13 10 infomaths@bat710.univ-lyon1.fr - alain.mille@liris.cnrs.fr</p>
Matériaux	<p>MATERIAUX DE LYON</p> <p>M. Jean Marc PELLETIER Secrétariat : C. BERNAVON 83.85</p>	<p>M. Jean Marc PELLETIER INSA de Lyon MATEIS Bâtiment Blaise Pascal 7 avenue Jean Capelle 69621 VILLEURBANNE Cédex Tél : 04.72.43 83 18 Fax 04 72 43 85 28 Jean-marc.Pelletier@insa-lyon.fr</p>
MEGA	<p>MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE</p> <p>M. Jean Louis GUYADER Secrétariat : M. LABOUNE PM : 71.70 –Fax : 87.12</p>	<p>M. Jean Louis GUYADER INSA de Lyon Laboratoire de Vibrations et Acoustique Bâtiment Antoine de Saint Exupéry 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél :04.72.18.71.70 Fax : 04 72 43 72 37 mega@lva.insa-lyon.fr</p>
ScSo	<p>ScSo*</p> <p>M. OBADIA Lionel Insa : J.Y. TOUSSAINT</p>	<p>M. OBADIA Lionel Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.69.72.76 Fax : 04.37.28.04.48 Lionel.Obadia@univ-lyon2.fr</p>

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Résumé

Ce travail de recherche entre dans le cadre de la conception pluridisciplinaire de systèmes techniques. Son contexte est le processus support à l'optimisation en conception organique suivant l'approche d'Ingénierie Système.

La problématique est l'affectation des variables et le choix de composants en tenant compte des possibilités de couplage entre les disciplines et des objectifs multiples à satisfaire. L'objectif poursuivi est un apport méthodologique et outillé pour l'aide à l'optimisation multiobjectif et multidisciplinaire.

La contribution comporte deux volets principaux. D'une part, deux méthodes de recherche basées sur les méta-heuristiques permettent de déterminer rapidement un ensemble des solutions efficaces. D'autre part, l'application de méthodes de discrimination fournit aux concepteurs une aide à l'interprétation des ensembles de solutions efficaces dégagés dans la phase d'optimisation.

Nos propositions sont illustrées par deux exemples de conception en robotique : dimensionnement d'un robot à deux degrés de liberté et choix de motoréducteurs pour la réalisation d'un robot désempileur pour l'industrie textile. Il est ainsi possible aux concepteurs d'analyser leurs marges de manœuvre dans l'affectation des variables de conception ou dans le choix de composants tout en préservant la satisfaction du client final.

Mots Clés : Ingénierie Système, Mécatronique, Optimisation multiobjectif, Optimisation multidisciplinaire.

Thèse préparée dans les laboratoires :

AMPÈRE, INSA de Lyon, 20 Avenue Albert Einstein, 69621, Villeurbanne Cedex.

LGI2P, Ecole des Mines d'Alès, Parc Scientifique Georges Besse, 30035 Nîmes Cedex.

Summary

Title:

Contribution to multiobjective optimization for multidisciplinary design

This research is related to the multidisciplinary design of complex systems. We focus on the optimization activity of the embodiment design phase according to “System Engineering” formalism. In such an activity, developers may encounter some difficulties mainly due to a coupling between the disciplines and a multiplicity of the design objectives.

To provide designers with a methodology and tools to help them when choosing the product architecture and components, we are interested in Multidisciplinary Design Optimization (MDO) methods as well as in multiobjective optimization.

Our contribution aimed at reducing the duration of the optimization activity and has two main issues. On the one hand, two research methods based on meta-heuristics can quickly determine an efficient solutions set. On the other hand, the application of discrimination methods provides designers with a help tool for analyzing the set of computed efficient solutions.

Our proposals are illustrated by two examples in robotics: design of a two degrees of freedom robot and choice of DC gearmotors for the realization of a ply gripper robot for the textile industry. This allows designers to analyze their leeway in the allocation of design variables or the choice of components while preserving the end-user satisfaction.

Key words: System Engineering, Mechatronics, Multiobjective optimization, Multidisciplinary Design Optimization.

Sommaire

Remerciements	1
Summary	4
Sommaire	5
Liste des figures.....	9
Liste des tableaux	12
Introduction générale	13
Chapitre 1 : Conception de systèmes multidisciplinaires en « Ingénierie Système »	17
Introduction	18
I. La conception comme processus	18
1. Modèles du processus de conception	21
2. Synthèse.....	26
II. Ingénierie Système	29
1. Notions de système.....	29
2. Le cadre de l'Ingénierie Système.....	32
3. Les processus de Ingénierie Système	36
A. Processus Ingénierie.....	37
B. Processus Support Ingénierie.....	40
4. Synthèse.....	40
Conclusion	41
Chapitre 2 : Optimisation en conception multidisciplinaire	45
Introduction	46
I. Optimisation en conception multidisciplinaire	46
1. Définitions et terminologie.....	46
2. Problématique	47
3. Formulation d'un problème OMD	48

II. Stratégies d'optimisation multidisciplinaire	50
1. Stratégies mono-niveau.....	53
A. Multi-Discipline Feasible (MDF).....	53
B. All-At-Once (AAO)	54
C. Individual-Discipline Feasible (IDF).....	55
2. Stratégies multi-niveaux.....	57
A. Collaborative Optimization (CO).....	58
B. Collaborative SubSpace Optimization (CSSO)	58
C. Bi-Level Integrated System Synthesis (BLISS)	58
D. Disciplinary Interaction Variable Elimination (DIVE).....	58
Conclusion	59
Chapitre 3 : Approches pour la résolution de problèmes d'optimisation multiobjectif	61
Introduction	62
I. Approches de résolution d'un problème multiobjectif	63
1. Approche mono-objectif.....	65
2. Approches multiobjectifs	66
A. Approche a posteriori	67
B. Approche progressive.....	67
3. Choix d'une approche.....	68
II. Méthodes d'optimisation	69
1. Méthodes complètes.....	70
A. La recherche arborescente orientée (ou Direct Tree Search).....	71
B. La Programmation Dynamique (ou Dynamic Programming)	71
C. La programmation linéaire (PL) (ou Linear Programming).....	71
2. Méthodes incomplètes.....	72
A. Algorithmes évolutionnaires.....	74
B. Recherche locale.....	75

3. Choix de méthodes	78
Conclusion	78
Chapitre 4 : Contribution à la résolution d'un problème d'optimisation multiobjectif en conception multidisciplinaire	81
Introduction	82
I. Optimisation avec pondération des objectifs	83
1. Description de la méthode de recherche	84
2. Mécanismes de l'algorithme de recherche.....	86
II. Optimisation sans pondération des objectifs	89
1. Description de la méthode de recherche.....	89
2. Mécanismes de l'algorithme de recherche.....	93
III. Cartes auto-organisatrices	98
Conclusion	100
Chapitre 5 : Aide au choix de variables de conception : Deux cas d'illustration	103
Introduction	104
I. Optimisation multiobjectif en conception d'un robot à deux degrés de liberté	105
1. Problématique	105
A. Cahier des charges	105
B. Architecture du robot	106
C. Variables et objectifs de conception :	107
D. Formalisation du problème:	108
2. Outil d'optimisation multiobjectif pour le choix d'architecture	110
A. Méthode tabou continu avec pas adaptatifs.....	111
B. Méthode de recherche hybride.....	114
3. Bilan	122
II. Application des outils de discrimination en conception d'un robot désempileur pour l'industrie textile	123
1. Modèle du robot	123

2. Simulations	127
A. Constitution de la base d'apprentissage	127
B. Exploitation des résultats	130
C. Discrimination avec des méthodes d'apprentissage	131
a) Apprentissage avec les cartes de Kohonen	131
b) Apprentissage incrémental avec la méthode IGNG	135
c) IGNG vs Kohonen	139
Conclusion	140
Conclusion Générale et Perspectives	141
Références bibliographiques	145
Annexe A : Méthodes de discrimination utilisées	154
1 Cartes de Kohonen	154
2 Incremental Growing Neural Gas (IGNG)	155
Annexe B : Modèle du robot à deux degrés de liberté	158
1 Architecture	158
2 Variables de conception	159
3 Domaines des variables	159
4 Objectifs de conception	161
5 Contraintes de conception	163
Annexe C : Modèle du robot désempileur	164
1 Description	164
2 Modèle cinématique	165
3 Modèle dynamique	166
4 Modélisation des ensembles variateur-actionneur	167
5 Modélisation des liaisons	168
6 Données	168

Liste des figures

Figure 1 : Un processus est un ensemble d'activités de transformation.....	19
Figure 2 : Types de conception ; regroupement par connaissance ou non du problème, de la solution et de la méthodologie de résolution	21
Figure 3 : Les phases du processus de conception, et les livrables de Pugh	22
Figure 4 : Les micro-stratégies de Purcell.....	23
Figure 5 : Les domaines de la conception de Yannou [Yannou'98].....	24
Figure 6 : Modèle co-évolutif de Lonchamp : Co-évolution entre le domaine du problème et le domaine de la solution [Lonchamp'04].....	25
Figure 7 : Mise en parallèle des phases de conception et du processus de décision lors de l'affinement de l'objet à concevoir [NASA handbook]	26
Figure 8 : L'évolution des modèles du processus de conception [Blessing'96].....	27
Figure 9 : Typologies et objectifs des modèles [Penalva'97]	27
Figure 10 : Prise en compte des activités du cycle de vie lors du développement du système à concevoir	32
Figure 11 : Décomposition en blocs systèmes du système à concevoir.....	33
Figure 12 : Les composantes de la mise en œuvre parallélisée, intégrée et simultanée du développement d'un système.....	34
Figure 13 : L'IS , cadre structurant toutes les activités du cycle de développement	34
Figure 14 : Les activités des processus de développement	35
Figure 15 : Cycle de développement de systèmes	35
Figure 16 : Processus techniques d'IS : « Phases » de conception et « activités support »	37
Figure 17 : Processus d'ingénierie ; distinction entre domaines du problème et de la solution	38
Figure 18 : Optimisation en conception organique.....	42
Figure 19 : Exemple de couplage entre les disciplines D1 et D2 : faible à gauche et fort à droite	47
Figure 20 : Approche standard d'optimisation.....	48
Figure 21 : Bloc d'analyse détaillé pour un système à trois disciplines, avec un couplage faible entre D1 et D2 et un couplage fort entre D2 et D3	49
Figure 22 : Bloc d'analyse détaillé pour un système, avec couplage fort entre disciplines D1, D2 et D3.	50

Figure 23 : Processus de résolution d'un problème d'optimisation multidisciplinaire	52
Figure 24 : Processus d'un problème d'optimisation multidisciplinaire avec reformulation	52
Figure 25 : Approche MDF	54
Figure 26 : Approche AAO	55
Figure 27 : Approche IDF	56
Figure 28 : Modèle résumant les approches multi-niveaux.....	57
Figure 29 : Approches et exemples de méthodes de résolution d'un problème OMD.....	59
Figure 30 : Illustration de deux ensembles de solutions dans l'espace des objectifs ; l'ensemble des solutions non dominées étant représenté en rouge	63
Figure 31 : Approches en optimisation multiobjectif.....	65
Figure 32 : Approches d'optimisation mono-objectif ; exemples de méthodes complètes	70
Figure 33 : Approches et exemples de méthodes de résolution d'un problème d'optimisation ;.....	74
Figure 34 : Algorithme de recherche tabou avec détermination de solutions locales efficaces	86
Figure 35 : Maillage par la méthode de l'hypercube latin.....	91
Figure 36 : Initialisation de l'espace de recherche et résultat de la phase 1 dans le cas d'un problème à deux variables ; à gauche : Espace de recherche; PMVi : Pas Maximal de discrétisation de la Variable i ; à droite : Front de Pareto déterminé après la première phase de recherche exhaustive...	92
Figure 37 : Principe de base de l'algorithme de recherche hybride avec détermination du front de Pareto	93
Figure 38 : Choix des paramètres de gestion de la liste de Pareto	97
Figure 39 : Optimisation en conception organique.....	104
Figure 40 : Robot à deux degrés de liberté [Cirrus'98]	106
Figure 41 : Ensemble des points à atteindre par l'extrémité du robot formant un petit M	107
Figure 42 : Entrées, sorties et domaines du processus de conception organique du robot à deux degrés de liberté.....	108
Figure 43 : Regroupement des courbes $FO = f(TC)$ ayant la même allure ; FO : valeur de la Fonction Objectif ; TC : Temps de Calcul	112
Figure 44 : $NP=f(DT)$; NP : Nombre de solutions dans le front de Pareto ; DT : Durée Tabou	112
Figure 45 : Variables de conception du robot à deux degrés de liberté.....	117
Figure 46 : Objectifs de conception du robot à deux degrés de liberté.....	117
Figure 47 : Exemple d'une solution de la carte de Kohonen représentant un ensemble de solutions semblables du front de Pareto	119

Figure 48 : Projection sur les axes « poids » et « gain proportionnel de l'axe 2 » des fronts obtenus avec les algorithmes HS (Hybrid Search), ES (recherche exhaustive), ainsi que le front obtenu après la phase de diversification de HS : SNE.....	120
Figure 49 : Robot dépileur pour l'industrie textile [Rakotoarisoa'93].....	123
Figure 50 : Trajectoire cinématique du point de contact	125
Figure 51 : Trajectoire cinématique du 1er axe.....	126
Figure 52 : Trajectoire cinématique du 2ème axe	126
Figure 53 : Trajectoire cinématique du 3ème axe.....	126
Figure 54 : Clusters des moteurs en 3D.....	130
Figure 55 : Domaines de validité (clusters) des moteurs	132
Figure 57 : Critère d'arrêt de l'algorithme IGNG	136
Figure 58: Voisinage 2D du neurone central (Hexagonal à gauche ou rectangulaire à droite)	154
Figure 59: Mise à jour du neurone gagnant (BMU) et de ses voisins directs	155
Figure 61 : Mise à jour des neurones gagnant : 2ème cas	156
Figure 62 : Algorithme Incremental Growing Neural Gas	157
Figure 63 : Architecture du robot à deux degrés de liberté [Cirrus'98]	158
Figure 64 : Ensemble des points à atteindre par l'extrémité du robot formant un petit M	159
Figure 65 : Points à atteindre, workspace et localisation de la base	160
Figure 66 : Exemple de solutions	160
Figure 67: Epaisseur des bras	161
Figure 68: Asservissement de l'axe	162
Figure 69: Description du robot en phase de préhension	164
Figure 70: Description du robot pour le modèle dynamique	166
Figure 71: Modélisation de l'ensemble variateur-actionneur.....	167

Liste des tableaux

Tableau 1 : Classification des conceptions en fonction de la connaissance et des méthodologies de résolution [Cirrus'98].....	20
Tableau 2 : Avantages et inconvénients de l'approche de recherche avec pondération des objectifs : Algorithme tabou avec pas adaptatifs et constitution du front de Pareto.....	89
Tableau 3 : Paramètres de mise à jour de la liste de Pareto	95
Tableau 4 : Variables continues : pas minimaux et maximaux et leurs domaines.....	110
Tableau 5 : Nombre de solutions du front final et durée totale d'exécution en fonction du nombre de phases d'intensifications.....	114
Tableau 6 : Bilan des trois méthodes de recherche.....	122
Tableau 7 Discrétisation des variables de conception	128
Tableau 8 : Liste des moteurs retenus sur le 1er axe	129
Tableau 9 : Liste des réducteurs retenus sur le 1er axe	130
Tableau 10 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode de Kohonen avec des prototypes de la base de tests	134
Tableau 11 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode Kohonen avec des spécifications techniques incomplètes (sans la valeur du pic de puissance) de la base de tests	135
Tableau 12 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode IGNG avec des prototypes de la base de tests.....	137
Tableau 13 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode IGNG avec des spécifications techniques incomplètes (sans la valeur du pic de puissance) de la base de tests	137
Tableau 14 : Matrice de confusion : Interrogations avec des prototypes de la base de test des cartes obtenues avec la méthode IGNG	138
Tableau 15 : Domaines des variables cinématiques.....	160
Tableau 16 : Matériaux utilisés et leurs caractéristiques	161

Introduction générale

Pour satisfaire une demande fluctuante et un haut niveau de qualité et de service, les industriels cherchent à développer le plus rapidement possible de nouveaux produits présentant de nouvelles fonctionnalités. Cela entraîne une complexification croissante de leur conception. En effet, le processus de conception nécessite alors le concours de nombreux acteurs dans des métiers variés. Il doit répondre à des contraintes techniques, économiques, légales, environnementales et de sûreté de fonctionnement toujours plus exigeantes. En particulier, lors de la conception de systèmes mécatroniques, l'intégration des ingénieries métiers (mécanique, électronique, automatique, informatique...) est nécessaire dès les phases amonts de ce processus. De plus, au fur et à mesure de l'avancement dans les phases de conception, de nouvelles contraintes sont dégagées et peuvent impacter tous les niveaux de conception à n'importe quel moment du cycle de vie.

Ce mémoire regroupe les travaux de recherche dont les objectifs consistent, d'une part, à situer des modèles reconnus de conception de produits techniques dans le cadre de l'Ingénierie Système et, d'autre part, à proposer des méthodes et outils aidant le concepteur humain dans certaines tâches fastidieuses et répétitives, afin qu'il se consacre davantage à celles nécessitant ses capacités cognitives. La contribution de cette thèse constitue un apport méthodologique et outillé pour l'aide à l'optimisation multiobjectif en phase de conception organique de systèmes multidisciplinaires dans les cas où les problèmes peuvent être posés dans un formalisme numérique.

Le premier chapitre rend compte de la richesse des modèles de conception séquentiels, simultanés, intégrés et co-évolutifs. Chaque modèle, bien que présentant des vues intéressantes du processus de conception, ne couvre qu'une partie des vues de l'ensemble des acteurs (managers, ingénieurs...).

Des travaux de l'INCOSE (International Council on Systems Engineering) et de l'AFIS (Association Française d'Ingénierie Système), permettent de regrouper l'ensemble des aspects de la conception autour de l' « Ingénierie Système ». Elle propose à l'usage de toutes les parties prenantes un cadre fédérateur et une terminologie commune couvrant les phases d'analyse du besoin et de conception tout au long du cycle de vie du système incluant le produit.

Nous situant dans ce cadre, nous constatons que la complexité du système multidisciplinaire à concevoir (à faire) se répercute sur celle du système de conception (pour faire). En effet, les connaissances requises en conception multidisciplinaire sont réparties entre différents domaines de la science et de l'ingénierie de sorte qu'une seule personne n'est en général pas capable de posséder toutes les expertises requises. Il est alors nécessaire d'impliquer des experts de métiers différents, ayant leurs propres méthodes et terminologies. Ces différents points de vue métier sur le système doivent s'accorder pour générer des solutions compatibles avec les règles et objectifs de chaque métier et respectant les objectifs globaux de conception définis par les besoins du client.

Les interactions entre les différents phénomènes physiques, les antagonismes entre qualité, sûreté de fonctionnement, coûts et délais à maîtriser font que plusieurs experts peuvent être amenés à modifier des paramètres techniques pour les besoins de certaines disciplines à l'insu d'autres. En outre, le nombre d'interactions peut être très grand selon le type du problème.

Toutes ces interactions se répercutent sur l'ensemble des processus de conception ainsi que sur leurs activités support. Plus particulièrement, en conception organique, l'activité support d'optimisation peut devenir relativement longue, fastidieuse et coûteuse.

La problématique traitée dans ce travail de recherche est celle de la prise en compte du couplage entre les variables et de la multiplicité des objectifs de conception. Ces aspects sont respectivement traités dans les deuxième et troisième chapitres.

Ainsi, le deuxième chapitre traite de l'optimisation en conception multidisciplinaire, aussi connue sous l'acronyme MDO (pour Multidisciplinary Design Optimization). L'optimisation globale du système à concevoir ne se limite pas à l'optimisation de ses composantes mais plutôt, du fait du couplage (faible ou fort) entre les disciplines, à une optimisation du système global nécessitant des compromis entre les disciplines. Afin de lever ces couplages, plusieurs stratégies existent. Elles sont classées en deux grandes familles : mono et multi-niveaux.

Le troisième chapitre traite de l'optimisation à objectifs multiples. Nous rappelons quelques approches, parmi les plus répandues, permettant de résoudre la problématique de cette multiplicité d'objectifs. En effet, deux grandes familles d'approches existent. Dans la première, différentes méthodes proposent des techniques de pondération des objectifs en fonction des préférences des décideurs et des données du problème à résoudre. Le problème est ensuite appréhendé par des méthodes classiques d'optimisation mono-objectif. Dans la deuxième approche, les objectifs ne sont pondérés à aucun moment du processus d'optimisation.

Dans le quatrième chapitre, nous présentons notre contribution à la problématique de l'affectation de valeurs aux variables en conception pluridisciplinaire et multiobjectif. Deux algorithmes d'optimisation capables de fournir aux concepteurs, en un temps acceptable, un ensemble de solutions efficaces (non dominées au sens de Pareto) sont introduits et analysés. Un outil de discrimination des solutions calculées est proposé pour en faciliter l'interprétation.

Le premier est applicable dans le cas où les concepteurs peuvent exprimer des préférences afin de classer les objectifs de conception et d'agrèger ainsi les fonctions objectifs. Il s'agit d'une adaptation d'une méthode de recherche locale : tabou de Glover, permettant de prendre en compte des variables mixtes et de dégager la meilleure solution trouvée ainsi que l'ensemble des solutions efficaces lors de l'exploration de l'espace des solutions.

Le second algorithme est une hybridation entre recherches exacte et approchée. Il permet de déterminer le front de Pareto sans avoir à agréger les objectifs du problème. En effet, notre but premier est de proposer rapidement aux concepteurs un ensemble de solutions afin qu'ils choisissent parmi des solutions au lieu d'avoir à choisir entre des valeurs d'objectifs.

Du fait de l'absence a priori de connaissance globale sur la répartition en famille des solutions possibles, les outils de discrimination envisagés reposent sur des cartes auto-organisatrices. Les cartes auto-organisatrices de Kohonen ont l'intérêt de permettre une projection des solutions sur des cartes visualisables en 2D. La contrainte d'imposition au départ de la topologie de l'espace de lecture des solutions peut entraîner un manque de précision dans la restitution aux concepteurs du résultat de la discrimination. Aussi avons-nous envisagé des méthodes plus flexibles et plus précises telles que la méthode Incremental Growing Neural Gas.

Notre contribution au support d'aide à l'optimisation multiobjectif en conception organique de produits multidisciplinaires à travers deux exemples en robotique fera l'objet du cinquième chapitre.

Dans le premier exemple, conception d'un robot planaire, les variables disciplinaires étant couplées, un choix de méthode MDO permet de lever ce couplage. Dans le but d'optimiser les choix des variables de conception, les deux algorithmes d'optimisation introduits dans le Chapitre 4 sont appliqués afin d'atteindre l'ensemble des solutions efficaces. Ces solutions sont interprétées en utilisant les cartes de Kohonen.

Le second exemple est issu du travail de conception d'un robot désempileur pour l'industrie textile. Etant composé de trois axes couplés, le choix des variables de conception et des composants de chaque axe influence fortement les choix des variables et des composants des deux autres. Le but de cette étude est d'apporter aux concepteurs un outil d'aide à l'interprétation d'un grand ensemble de solutions multidimensionnelles. C'est pourquoi nous mettons en évidence l'apport de l'application des méthodes de discrimination basées sur les cartes auto-organisatrices introduites au chapitre 4.

Chapitre 1 : Conception de systèmes multidisciplinaires en « Ingénierie Système »

Introduction

Le premier chapitre rend compte de la richesse des travaux visant à définir la conception. Nous distinguons quatre grandes familles de modèles : séquentiels, simultanés, intégrés et co-évolutifs. Chaque modèle, bien que présentant des vues intéressantes du processus de conception, ne couvre qu'une partie des vues des acteurs (managers, ingénieurs...).

En conception de systèmes multidisciplinaires, la complexité du système à faire se répercute sur celle du système pour faire. En effet, les connaissances requises en conception multidisciplinaire sont réparties entre différents domaines de la science et de l'ingénierie de sorte qu'une seule personne n'est en général pas capable de posséder toutes les expertises requises. Un cadre fédérateur, fruit des travaux de l'INCOSE et de l'AFIS, permet de regrouper l'ensemble des aspects de la conception autour de l'« Ingénierie Système ». Elle propose à l'usage de toutes les parties prenantes un modèle générique et une terminologie commune couvrant les phases d'analyse du besoin et de conception tout au long du cycle de vie du produit.

I. La conception comme processus

De nombreuses définitions de la conception existent. S'appuyant sur celle du cogniticien Simon [Simon'73], la conception consiste en « un passage d'une situation initiale d'insatisfaction à une situation objective, dans laquelle cette insatisfaction est résolue », Lonchamp [Lonchamp'04] considère que « le processus de conception est une démarche de résolution d'un problème ». Nous allons préciser les concepts et notions attachées à cette définition pour mieux situer le cadre de notre travail.

Un processus est un ensemble d'activités qui transforment des éléments d'entrée, ici les besoins du client en éléments de sortie, un produit, en utilisant des ressources et en satisfaisant des contraintes (Figure 1).

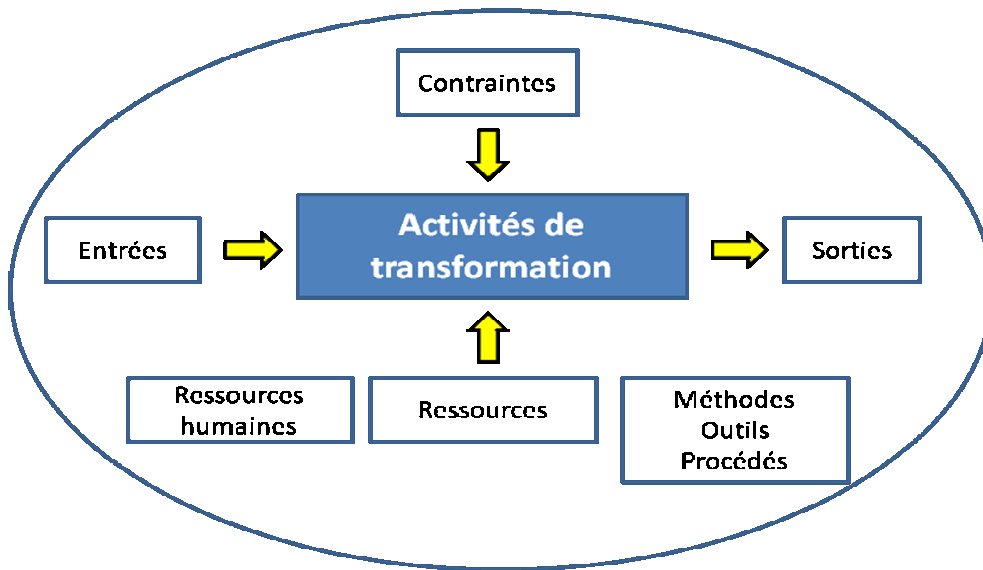


Figure 1 : Un processus est un ensemble d'activités de transformation

Au sens de l'AFNOR [Normes Françaises'90], un produit est défini comme un objet matériel ou logiciel. Dans cette étude le produit est décrit par l'ensemble des informations qui constituent le dossier de conception d'un ensemble cohérent de composants technologiques ou d'un système technique pouvant inclure des éléments matériels ou logiciels.

La résolution d'un problème de conception est en général collective, œuvre de plusieurs acteurs de différents métiers. C'est une activité coordonnée visant à atteindre un objectif commun aux agents coopérants [Coelho'09].

Un problème de conception est en général un problème ouvert qui admet un nombre non fini et non prévisible de solutions. Il peut être mal défini, car l'état initial, l'état but et les opérateurs qui permettent de passer de l'un à l'autre sont en général non identifiés préalablement. Certaines données peuvent être déterminées durant le processus et peuvent donc changer la connaissance du problème.

La conception suppose une clarification des besoins ou insatisfactions du demandeur (initialement le client), une recherche de solutions et une mise en œuvre de ces solutions. Elle s'appuie sur des connaissances explicites ou implicites et sur des méthodologies de résolution.

Le niveau de disponibilité de connaissances ou de méthodologies nécessaires à la conception, peut permettre de distinguer différents types de conception. Ainsi Cirrus [Cirrus'98], se basant sur la possession ou non possession par les concepteurs des connaissances requises et des méthodologies à appliquer propose la classification du Tableau 1.

Type de conception	Base de connaissance	Méthodologie de résolution
Créative	Inconnue	Inconnue
Innovante	Connue	Inconnue
Routinière	Connue	Connue

Tableau 1 : Classification des conceptions en fonction de la connaissance et des méthodologies de résolution [Cirrus'98]

Scaravetti, dans [Scaravetti'04], a introduit une classification plus détaillée basée sur la part de connaissances nouvelles nécessaire aux concepteurs, où il distingue entre:

- la conception routinière qui fournit une solution à un nouveau problème en se basant totalement sur des solutions existantes dans la plupart des cas cataloguées,
- la re-conception qui répond à de nouvelles exigences en partant d'une solution déjà existante dans le but d'améliorer ses performances ou de la corriger,
- la conception adaptative qui permet de garder les fonctionnalités initiales en adaptant l'architecture suite à l'apparition de nouvelles exigences,
- le « variant design » qui permet de décliner un produit existant en une gamme,
- la conception innovante qui permet de combiner des éléments ou des connaissances existants afin de développer un nouveau produit,
- la conception stratégique permettant de développer des connaissances nouvelles : des résultats ou des applications produit de la recherche fondamentale la conception créative où l'objet à concevoir est défini d'une façon abstraite et où la connaissance et les méthodologies de résolution sont jusque là inconnues en conception.

Afin d'enrichir la classification de Cirrus, en tenant compte des constatations de Scaravetti, nous proposons une décomposition de l'axe « Connaissance » en trois catégories : « A construire » (les connaissances sont à découvrir), « A s'approprier » (les connaissances existent mais ne sont pas encore maîtrisées par le groupe de concepteurs concernés), et « Connues » (les connaissances sont acquises ou facilement accessibles et exploitables).

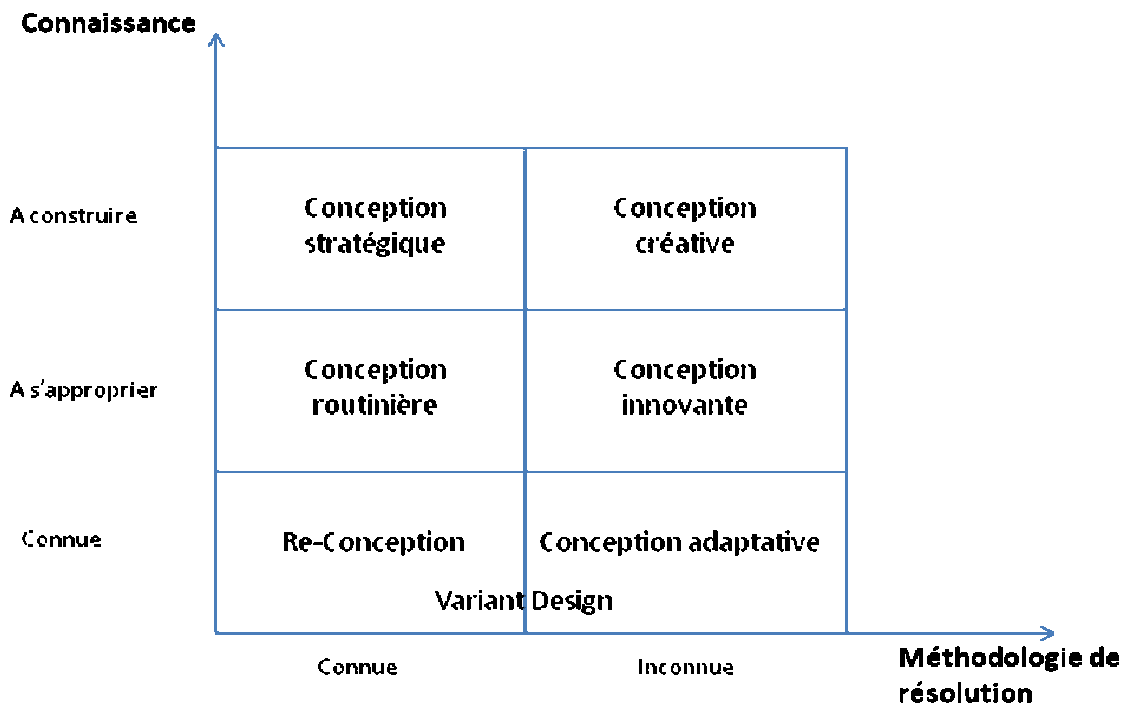


Figure 2 : Types de conception ; regroupement par connaissance ou non du problème, de la solution et de la méthodologie de résolution

Ainsi nous pouvons préciser que le processus de conception est un ensemble d'activités collectives qui transforment progressivement des besoins pouvant être redéfinis au cours du temps en éléments de représentations du produit solution en utilisant des ressources, connaissances et méthodes disponibles ou bien à construire et en satisfaisant des contraintes, limites à la liberté des concepteurs.

Nous décrivons plus en détail quelques modèles de conception issus de la littérature au paragraphe suivant.

1. Modèles du processus de conception

Un modèle désigne « toute représentation ou transcription abstraite d'une réalité concrète » [Durand'79]. Cette représentation dépend de l'état de l'objet en cours de transformation dans le processus de conception, de l'objectif fixé au modelleur, ainsi que de l'utilisateur final du modèle (Figure 9).

En effet, le processus de conception peut être considéré comme une « conversation réflexive » et est assimilé à un cycle de transformations (ou mouvements) pratiquées par les concepteurs sur l'objet à concevoir avec des activités d'expérimentation exploratoire et d'évaluation.

Certains modèles de conception sont orientés selon une décomposition temporelle de ce processus. Parmi eux, Lonchamp distingue les modèles basés sur la notion de phases séquentielles. Une phase étant une partie du processus de conception délimitée par le dégagement d'un livrable.

Parmi ces modèles, nous trouvons ceux d'Ulrich et Eppinger [Ulrich'00], d'Ullman [Ullman'02], l'Analyse de la Valeur et l'Analyse Fonctionnelle [Normes Françaises 90] ; Total Design de [Pugh'91]...

La Figure 3 illustre le modèle de Pahl [Pahl'96], très cité dans la littérature, avec ses phases de conception et les attendus de fin de chaque phase.

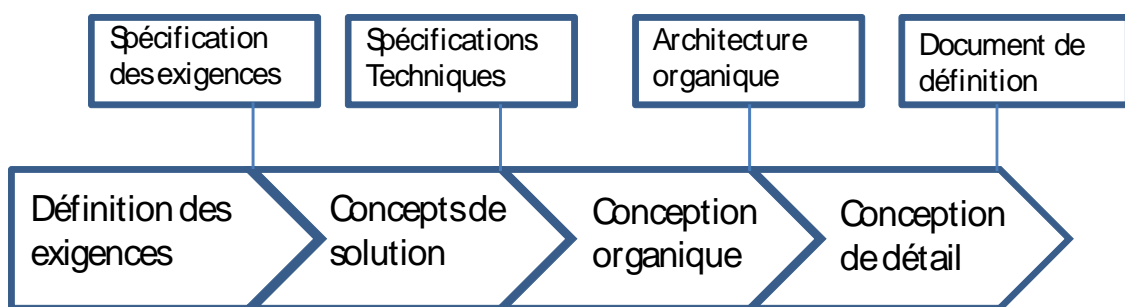


Figure 3 : Les phases du processus de conception, et les livrables de Pugh

Suh a dégagé dans [Suh'90] des phases communes à la plupart des processus de conception :

- Connaître et comprendre les besoins du client ;
- Définir le problème à résoudre pour satisfaire ces besoins ;
- Conceptualiser la solution ;
- Effectuer l'analyse (de la solution), pour optimiser la solution proposée ;
- Vérifier la conception obtenue, pour voir si elle répond aux besoins clients initiaux ;

Très utiles pour suivre l'état d'avancement du développement du produit, ces modèles basés sur un phasage séquentiel présente quelques inconvénients majeurs :

- Ils ne représentent pas suffisamment la dynamique du processus (affinement progressif des solutions, impact en amont de décisions prises au cours des phases successives).
- En étant unidimensionnels (selon l'axe du temps), ils ne présentent pas de vue permettant de décrire la diversité des activités intervenant dans le processus de conception.

C'est pourquoi une seconde catégorie de modèles, basée sur la notion d'activité, décompose plus finement (qu'une phase) la conception. Le processus de conception devient alors un enchaînement d'activités.

L'intérêt de ce type de décomposition est d'associer à chaque activité, appelée aussi tâche, les acteurs impliqués dans sa réalisation. Cette décomposition permet une granularité plus fine de l'axe temporel.

Les modèles les plus connus de cette catégorie sont celui de Purcell [Purcell'94], de Micaelli [Micaelli'02] et d'Ahmed (approche EVAD) [Ahmed'02]... La Figure 4 montre un exemple d'une démarche de conception selon Purcell qui utilise le terme « micro-stratégie » pour ce que nous appelons « activité ». On distingue bien les phases, telles que l'analyse du problème, la proposition de la solution (conceptualisation), l'analyse de cette solution... On retrouve également des micro-stratégies décrivant les activités à mener lors de chaque phase. Ce modèle - de Purcell - présenté comme une liste de tâches, ne fait d'hypothèses ni sur le nombre ni sur l'enchaînement de ces tâches. Les réponses aux questions (micro-stratégies) peuvent donc se chevaucher ou être menées en parallèle.

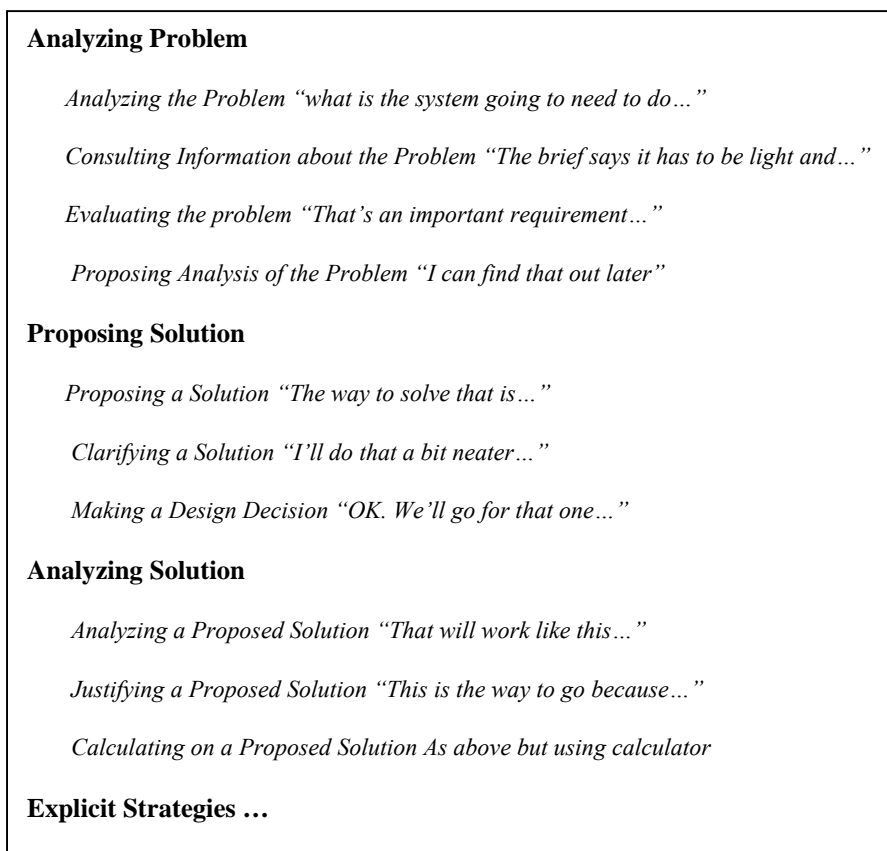


Figure 4 : Les micro-stratégies de Purcell

Cette décomposition en activités enchevêtrées permet de mettre en évidence la dynamique du processus de conception.

Certains travaux, tels que les niveaux d'abstraction de Purcell ou bien sujet/sous-sujet de Micaelli, complètent la représentation de l'axe temporel par une représentation d'une autre dimension non parallèle. Lonchamp utilise le terme « domaine » pour décrire ces différents axes de décomposition du processus de conception. Les termes « monde », « espace » ou encore « dimension » peuvent être trouvés dans la littérature. Le processus de conception est présenté comme « une évolution parallèle et/ou itérative de différents domaines, mondes ou espaces » [Lonchamp't04].

Parmi ces modèles, nous trouvons l'Axiomatic Design de Suh, les approches d'Europe du nord dont celle de Mortensen [Mortensen'99], Function Behavior Structure de Gero [Gero'02], et Design Rationale de Regli [Regli'00]... Dans l'approche valorique (voir Figure 5) [Yannou'98], Yannou distingue entre les mondes du modèle où on trouve l'espace subjectif des besoins et l'espace explicite physique, et le monde réel où on trouve un espace perceptuel subjectif par opposition à l'espace fonctionnel qui est explicite.

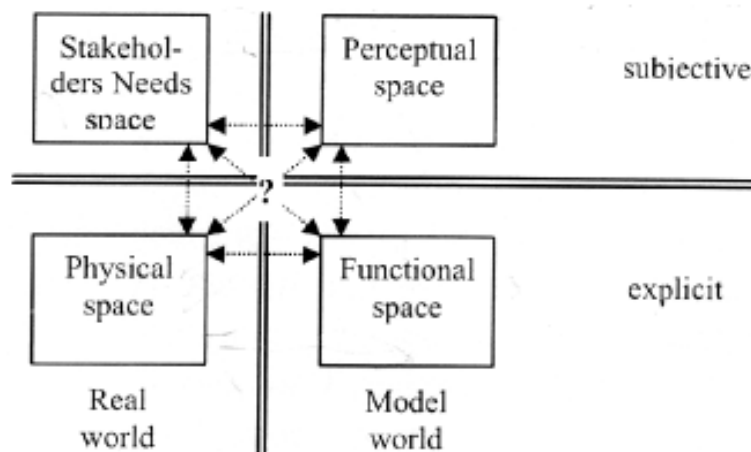


Figure 5 : Les domaines de la conception de Yannou [Yannou'98]

D'autres exemples de modèles distinguent entre les mondes abstrait et concret, et entre les domaines « Process », « Function », « Organ », « Constructional »...

Selon Longchamp, lors des différentes itérations du processus de conception, les domaines besoins et solutions évoluent simultanément vers une solution finale (Figure 6). Les domaines

du problème et de la solution progressent et s'affinent jusqu'à l'épuisement des problèmes posés (solution obtenue) ou des solutions proposées (impasse).

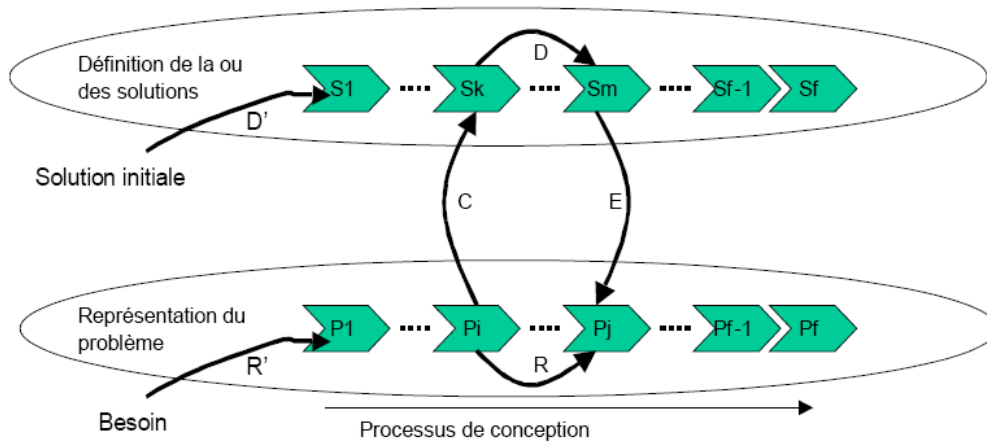


Figure 6 : Modèle co-évolutif de Lonchamp : Co-évolution entre le domaine du problème et le domaine de la solution [Lonchamp'04]

Les activités définies par Lonchamp pour son modèle co-évolutif, où les S_i et P_i représentent les domaines des solutions et des problèmes, sont :

- R : Reformulation du problème
- C : Conjecture
- D : Définition
- E : Evaluation

L'exemple du modèle dynamique en spirale (Figure 7) [NASA handbook] nous paraît intéressant car il complète la notion de co-évolution par celle de décisions d'implémentation (impact du processus de conception sur d'autres processus pouvant être lancés en parallèle) et de convergence vers un produit final par réduction de la granularité de résolution. Le modèle illustre les processus de conception et de décision se déroulant en parallèle, pour l'objet à concevoir et pour ses composantes.

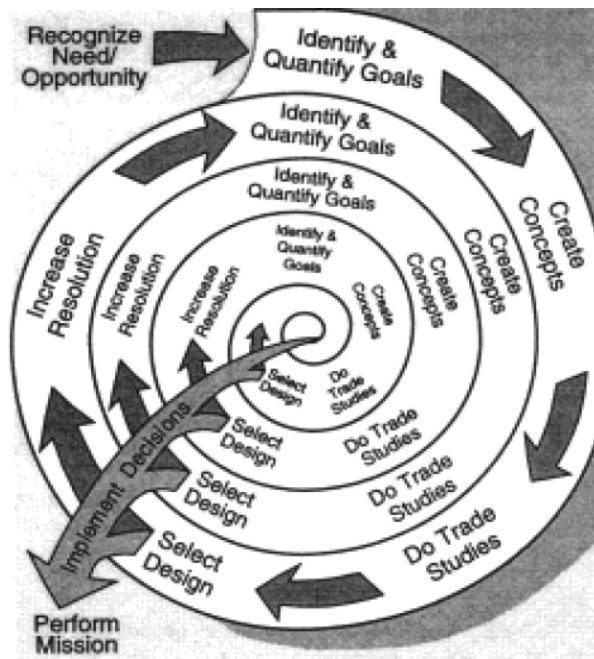


Figure 7 : Mise en parallèle des phases de conception et du processus de décision lors de l'affinement de l'objet à concevoir [NASA handbook]

Scaravetti insiste dans [Scaravetti'04] sur la notion de conception intégrée. Dans [Lonchamp't'04] et [Scaravetti'04], on parle d'intégration du cycle de conception dans le cycle de vie global du produit. C'est-à-dire qu'aux besoins initiaux du client, il faut ajouter les besoins des parties prenantes lors de la fabrication, de l'exploitation, de la maintenance, et du retrait du marché.

2. Synthèse

Au cours du temps, les modèles de processus de conception sont passés par trois grandes étapes :

- Processus de conception séquentiels : Une tâche ne commence qu'après l'achèvement total de celle qui la précède.
- Processus de conception simultanés : Différentes tâches peuvent se chevaucher, commencer ou se terminer en même temps...
- Processus de conception intégrés : Dans ce cas, le modèle intègre les différents acteurs intervenant tout au long du cycle de vie prenant en charge leurs interactions (diffusion de documents, conversations...).

Cette évolution est schématisée dans [Blessing'96] comme donné dans la Figure 8 :

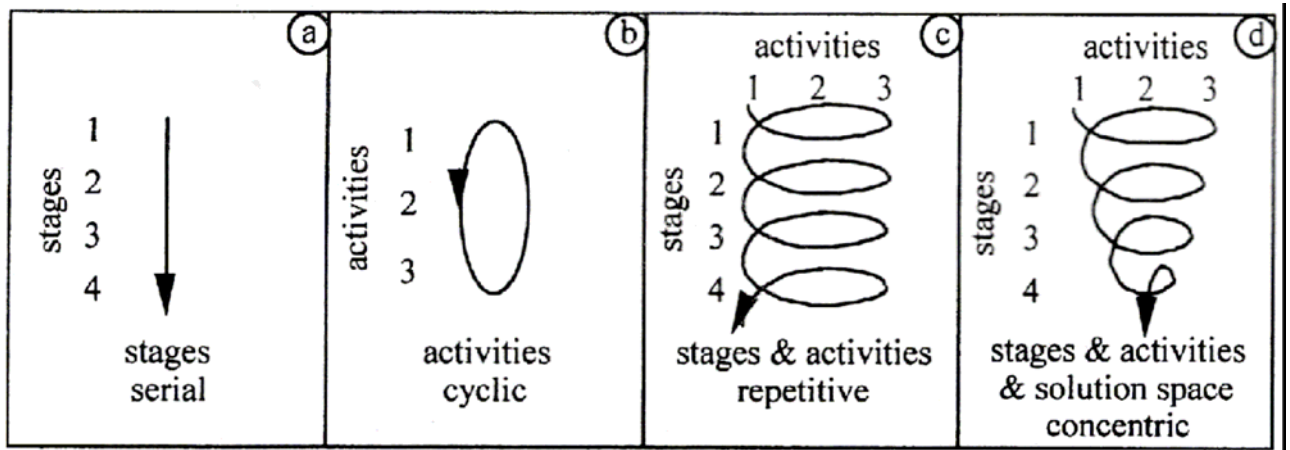


Figure 8 : L'évolution des modèles du processus de conception [Blessing'96]

Dans la Figure 8, le premier cadre (a) symbolise l'enchaînement strict des phases de conception au cours du temps. Les modèles basés sur les activités sont schématisés dans (b) par un cycle itératif d'activités. (c) schématise un flux répétitif à la fois des phases et des activités. (d) représente la répétition des activités et des phases mais avec une convergence qui correspond, selon Lonchamp, à la réduction de l'ensemble des solutions possibles.

Type du modèle	Modèle cognitif	Modèle normatif		Modèle prédictif ou prospectif	
		Modèle prescriptif	Modèle constructif	Modèle formel	Modèle analytique
Usages du modèle	Analyser Comprendre Identifier l'environnement Explorer Simuler Valider les concepts opérationnels	Formaliser le problème et les besoins Prescrire les exigences	Construire les architectures fonctionnelles et physiques	Prévoir Valider Prouver des comportements Simuler	Estimer la performance la fiabilité et la SdF Simuler
Objectifs du modèle	Fournir une représentation d'un système existant qui mette en évidence les <u>propriétés intéressantes</u> de ce système	Fournir une représentation d'un système à créer qui mette en évidence les <u>propriétés souhaitées</u> de ce système		A partir des connaissances que l'on a d'un état du système, déduire son comportement dans des situations nouvelles	

Figure 9 : Typologies et objectifs des modèles [Penalva'97]

Cette évolution des modèles du processus de conception est due, en grande partie, à la complexification des produits à concevoir (en associant notamment les services à fournir au

cours de la vie du produit). Ainsi, afin de prévoir des problèmes potentiels pouvant survenir pendant les phases aval du cycle de vie, les processus de conception ont tendance à intégrer dans le cycle de développement tous les acteurs intervenant durant le cycle de vie pour mener une résolution interdisciplinaire et collective du problème de conception.

Comme tout modèle, un modèle de conception est caractérisé par ses objectifs et ses utilisateurs (Figure 9).

Ainsi, les modèles prescriptifs axés sur la notion de phase permettent-ils de renseigner le chef de projet et le client sur l'état d'avancement du travail de conception par rapport aux jalons principaux. Les modèles prescriptifs, basés sur la notion d'activité permettent d'avoir une décomposition en tâches à planifier et des ressources à affecter. Les vues par domaine décrivent le processus suivant plusieurs axes non parallèles à celui du temps. Ainsi, les utilisateurs des modèles cognitifs peuvent-ils avoir des informations sur les domaines métiers associés, les approvisionnements nécessaires, la documentation disponible, le niveau de qualité, des coûts, de l'innovation... Les modèles prospectifs (formel ou analytique) mis en œuvre dans les activités d'ingénierie permettent de simuler et valider des attentes ou des comportements du système. Nous pouvons dire que chaque modèle (cognitif, normatif ou prospectif) de conception apporte une vue indispensable et incomplète du processus de conception en se limitant à certains domaines, avec son propre niveau de granularité.

Afin d'unifier ces approches, certaines communautés rassemblent, dans un même référentiel, un ensemble de modèles afin de couvrir une grande partie, des aspects de la conception. Ceci, dans le but d'uniformiser les langages et les pratiques dans ce domaine...

La communauté OMG, par exemple, a rassemblé un ensemble de modèles couvrant une grande partie du cycle de développement de produits dans un référentiel commun appelé SysML « System Modeling Language ». SysML n'est autre que l'adaptation d'UML « Unified Modeling Language » de la conception logicielle à la conception de produits.

La modélisation SysML, comme UML, offre une panoplie de modèles qui prodiguent un très bon outil de communication via ses représentations compréhensibles par les acteurs de la conception quelque soit le domaine d'expertise. L'inconvénient de cette démarche est, à notre sens, que ces modèles sont orientés vers l'objet à concevoir. Nous trouvons une incomplétude des points de vue démarche à suivre, management de la conception...

Une autre approche, fruit des travaux de l'INCOSE (International Council on Systems Engineering) et de l'AFIS (Association Française d'Ingénierie Système), regroupe l'ensemble

des aspects de la conception autour de l' « Ingénierie Système ». Elle propose un cadre fédérateur et une terminologie commune couvrant l'analyse du besoin et la conception du produit pendant l'ensemble de son cycle de vie à l'usage de toutes les parties prenantes. Parce que ce cadre nous permet d'intégrer et de lier les différents concepts et notions abordées jusqu'à présent, nous présentons l'ingénierie système au paragraphe suivant.

II. Ingénierie Système

L'Ingénierie Système étant centrée sur la notion de système, il nous paraît nécessaire de définir ce qu'est un système. Puis nous introduisons les notions de système complexe et de système multidisciplinaire.

1. Notions de système

Systeme :

Dans le domaine des sciences de l'ingénieur un système peut être défini comme « une entité technique », formée d'« un ensemble d'éléments en interrelations mutuelles » [Durand'79], et en interaction dynamique [De Rosnay'75].

L'Association Française de l'Ingénierie Système (AFIS) définit un système comme « un ensemble d'éléments en interaction entre eux et avec l'environnement, intégré pour rendre à son environnement les services correspondant à sa finalité ».

Bien définir un système en IS commence donc par répondre aux questions « pourquoi ? », « comment ? » et « combien ? ». Les réponses représentent, respectivement, la finalité, la mission et les objectifs du système. Par la suite, doivent être explicités les scénarios opérationnels ainsi que les fonctions et les constituants, et surtout le périmètre (ou bien environnement) du système à concevoir.

Lorsqu'un élément du système peut être considéré comme un système à part entière, c'est-à-dire que l'on peut lui définir une finalité, une mission et un objectif propre..., cet élément est alors qualifié de « sous-système ».

Systeme complexe :

Un système est qualifié de complexe, selon [Lemoigne'90], « si on le tient pour irréductible à un modèle fini, aussi compliqué, stochastique, sophistiqué que soit ce modèle, quelle que soit sa taille, le nombre de ses composants ou l'intensité de leurs interactions ».

[Meinadier'98] distingue deux types de complexité du système :

- La complexité statique (désignée par complexité spatiale dans [Braesch'95]) qui est liée à l'architecture du système, à savoir, le nombre de fonctions, de composantes, de relations, etc. existantes. [Morel'92] propose par exemple d'évaluer la complexité statique au moyen de mesures théoriques ou métrique telles que la complexité visuelle, la complexité structurelle et la complexité textuelle.
- La complexité dynamique qui est liée à la dynamique des interactions entre les sous-systèmes et les composants, ainsi qu'à l'existence des relations bouclées entre les fonctions.

Plusieurs concepts intéressants apparaissent [Morin'77] : l'auto-organisation (liée essentiellement à la structure du système), l'émergence (le comportement du tout ne peut en effet être simplement déduit du comportement des composants). En se basant sur [Penalva'97], est considérée comme complexe toute situation qui présente pour un observateur des difficultés de compréhension, d'anticipation ou de maîtrise.

Conception de systèmes multidisciplinaires :

Un système multidisciplinaire est le produit d'une conception pluridisciplinaire ou interdisciplinaire. Nous avons identifié les caractéristiques de la conception des systèmes multidisciplinaires qui peuvent être sources de complexité :

- **Multi-compétences :**

Les connaissances requises pour faire la conception multidisciplinaire sont réparties entre différents domaines de la science et l'ingénierie de sorte qu'une seule personne n'est pas capable de posséder toutes les expertises requises.

- **Multi-vues :**

L'implication d'experts de divers métiers crée une différence de points de vue sur le système à concevoir et des besoins d'échanges et de coopération.

- **Multi-langages :**

Les membres d'une équipe multidisciplinaire, collaborant les uns avec les autres afin d'atteindre un objectif commun, sont souvent de cultures scientifiques différentes et

utilisent leur propre langage. En conséquence, il devient difficile pour les participants de communiquer, de se comprendre et de régler les conflits éventuels.

- **Multi-objectifs :**

Les différents métiers ont leurs propres règles et contraintes. Celles-ci peuvent entrer en conflit avec les objectifs globaux de la conception ou avec des caractéristiques des solutions proposées par les autres métiers.

- **Multi-relations et couplage interdisciplinaire :**

Les interactions entre les différents phénomènes physiques, les antagonismes entre qualité, sûreté de fonctionnement, coûts et délais à maîtriser peuvent faire que certains membres de l'équipe de conception peuvent modifier des paramètres pour leur propre besoins à l'insu des besoins des autres.

- **Intégration interdisciplinaire :**

L'intégration des composants et sous-système, l'élaboration de procédures de vérification des sous-systèmes sont eux-mêmes des problèmes multidisciplinaires qui nécessitent l'imbrication des compétences.

- **Dynamique difficilement prévisible :**

La départementalisation, la distribution géographique des compétences métiers cloisonnent les disciplines et rendent moins prévisible le nombre d'itérations nécessaires pour trouver une solution..

- **Complexité combinatoire :**

Les composantes physiques de chaque produit étant souvent nombreuses, le processus d'évaluation de toutes les configurations possibles s'avère coûteux en temps, même en simulation. De plus, pour ne pas écarter une piste de solution innovante trop tôt, une tendance est à repousser le plus longtemps possible la décision d'abandonner une voie investiguée.

Un système multidisciplinaire est source de complexité, à cause des différences de culture scientifique au sein de l'équipe qui conçoit un tel système. De ce fait, en général, les acteurs de la conception peinent à communiquer et à trouver des accords quant aux choix de conception, chacun ayant leurs propres contraintes métiers et des objectifs pouvant être contradictoires

2. Le cadre de l'Ingénierie Système

Pour l'Ingénierie Système (IS), la conception fait partie d'un cycle de développement, qui lui-même, se répète au cours du cycle de vie du produit pour répondre aux besoins des activités de production, de maintenance d'évolution ou de démantèlement. Ces activités sont mises en œuvre par des systèmes contributeurs portant le même nom (Figure 10). Le développement en parallèle de ces systèmes contributeurs suppose la prise en compte au plus tôt et simultanément de leurs exigences.

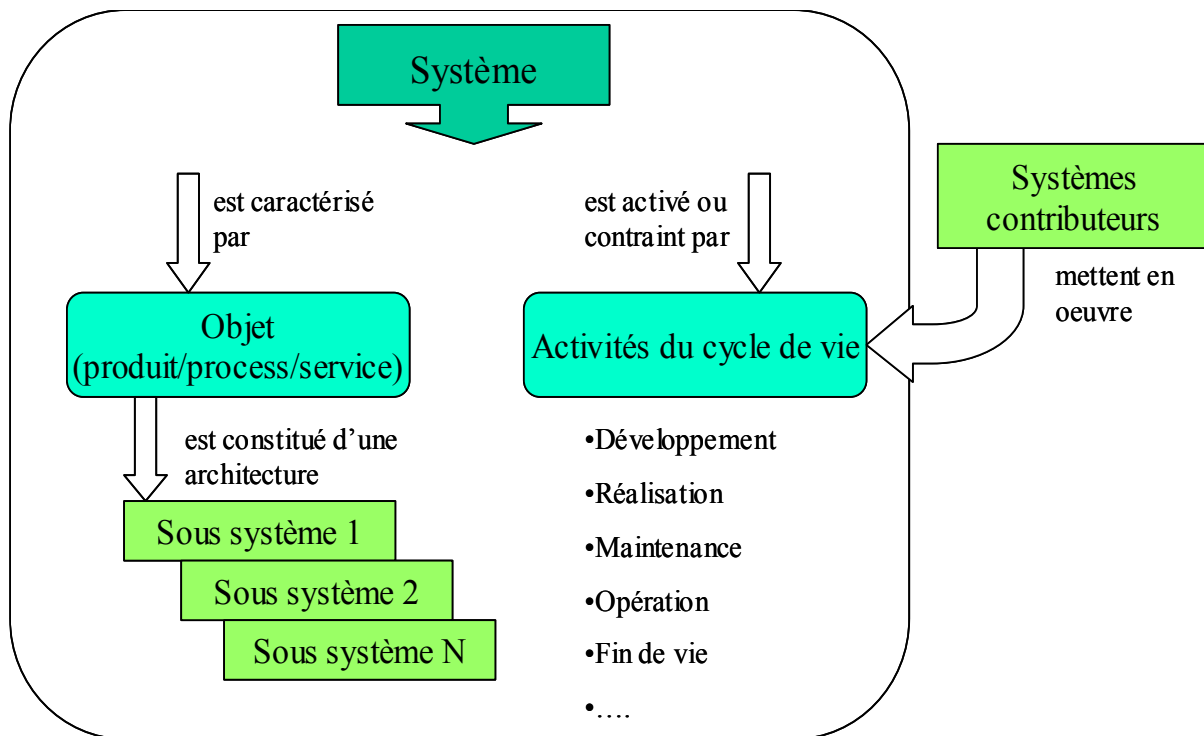


Figure 10 : Prise en compte des activités du cycle de vie lors du développement du système à concevoir

Le développement simultané du système et des activités du cycle de vie est désigné par *Ingénierie simultanée*, et induit l'intégration de différents acteurs du cycle de vie dans le processus de développement : *Ingénierie intégrée*.

En effet, l'ensemble des compétences nécessaires durant le cycle de vie est réuni dès le début du projet : experts du besoin opérationnel, architectes et intégrateurs système, génies techniques et technologiques, spécialistes de maîtrise des risques et sûreté de fonctionnement, experts économiques, ergonomes, spécialistes de production, de maintenance et de recyclage en fin de vie, etc. Les problèmes sont ainsi analysés de manière interdisciplinaire : les raisons

des choix successifs sont justifiées après une évaluation coopérative des solutions alternatives, de leurs impacts et de leurs risques.

D'autre part, un système à concevoir est décomposable en artefacts (produit, procédé ou service), qui sont supportés par des sous-systèmes. Comme le montre la Figure 11, chaque sous-système est considéré comme un système à part entière où les activités de son cycle de vie sont prises en compte lors de son propre développement.

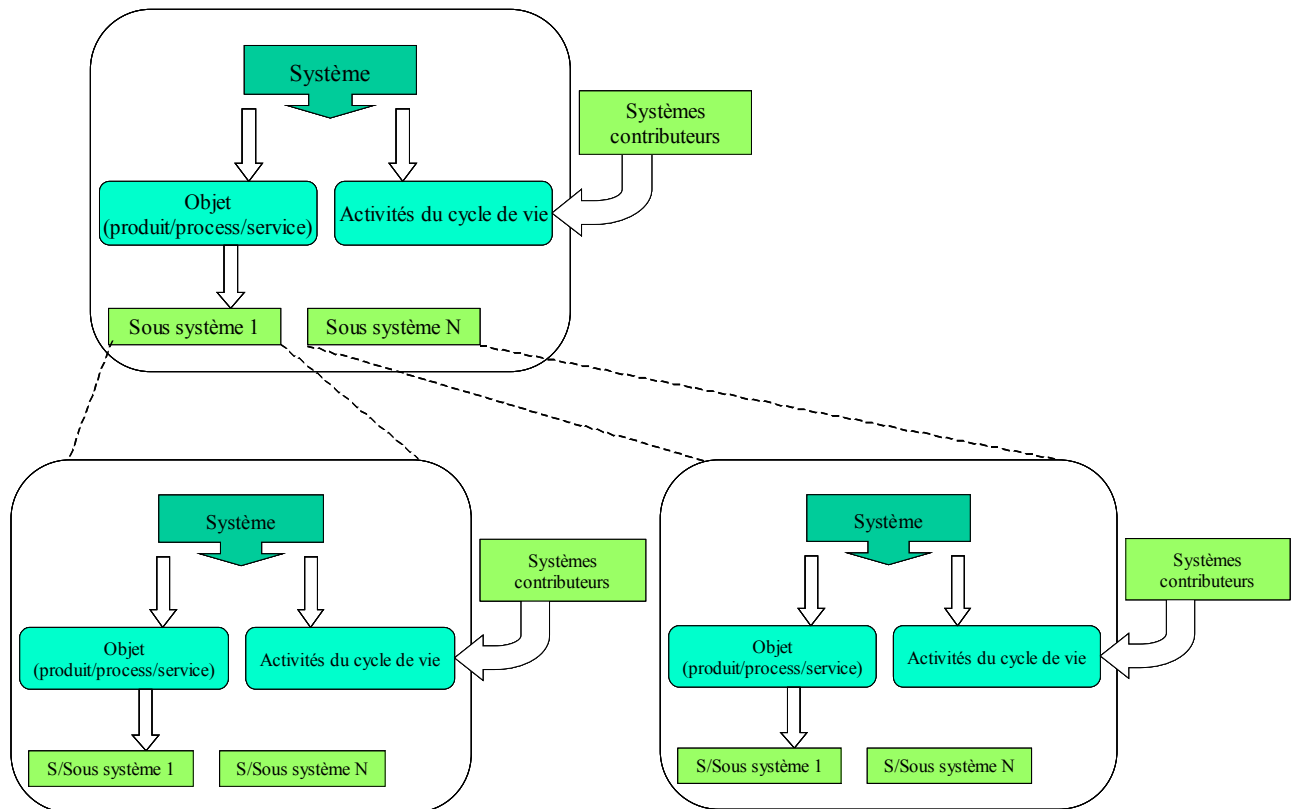


Figure 11 : Décomposition en blocs systèmes du système à concevoir

Le développement en parallèle du système et de l'ensemble de sous-système qualifie le processus de conception d'*ingénierie parallélisée*.

La Figure 12 synthétise l'ensemble des propriétés de l'ingénierie : « parallélisée », « simultanée » et « intégrée ».



Figure 12 : Les composantes de la mise en œuvre parallélisée, intégrée et simultanée du développement d'un système

L'Ingénierie Système (IS) est un cadre de référence supportant les concepts d'ingénierie parallélisée simultanée et intégrée et offrant un cadre structurant l'ensemble des activités de conception. Cette structuration est décrite par la Figure 13 [Faisandier'05b] où l'on peut retrouver les différents domaines du processus de développement de systèmes complexes.

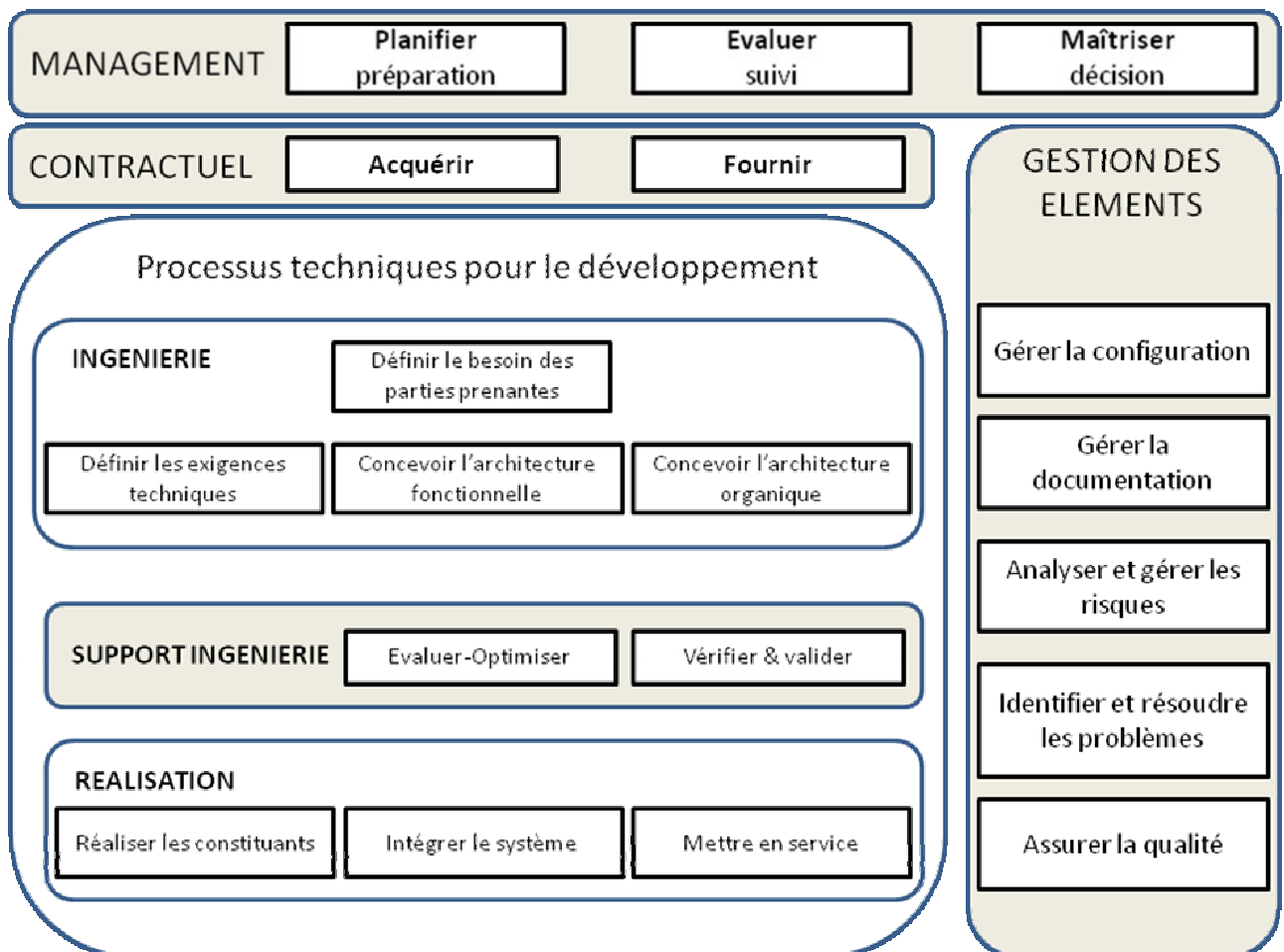


Figure 13 : L'IS , cadre structurant toutes les activités du cycle de développement

L'IS répond au besoin d'avoir un modèle basé sur la notion de domaine (section I) qui fournit à chaque acteur de la conception une vue l'intéressant. Aussi, met-elle en relation les activités destinées à la gestion du projet, la gestion des éléments de conception et des aspects contractuels, et bien évidemment le cycle de développement.

Un processus type de développement en IS « met en œuvre des activités de conception faisant passer du besoin à la spécification des constituants d'un système apportant une solution à ce besoin, des activités de réalisation de ces constituants, des activités d'intégration » Figure 14.

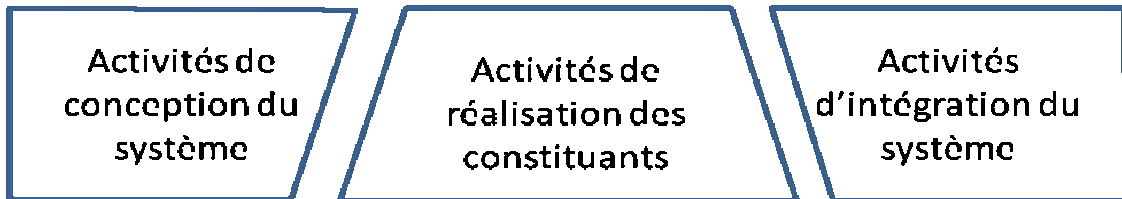


Figure 14 : Les activités des processus de développement

Le système est itérativement décomposé en sous-systèmes et constituants. Ceux-ci sont développés en parallèle selon la logique générale de conception, réalisation et intégration. En final, le système est à son niveau intégré (y compris dans son environnement), vérifié et validé.

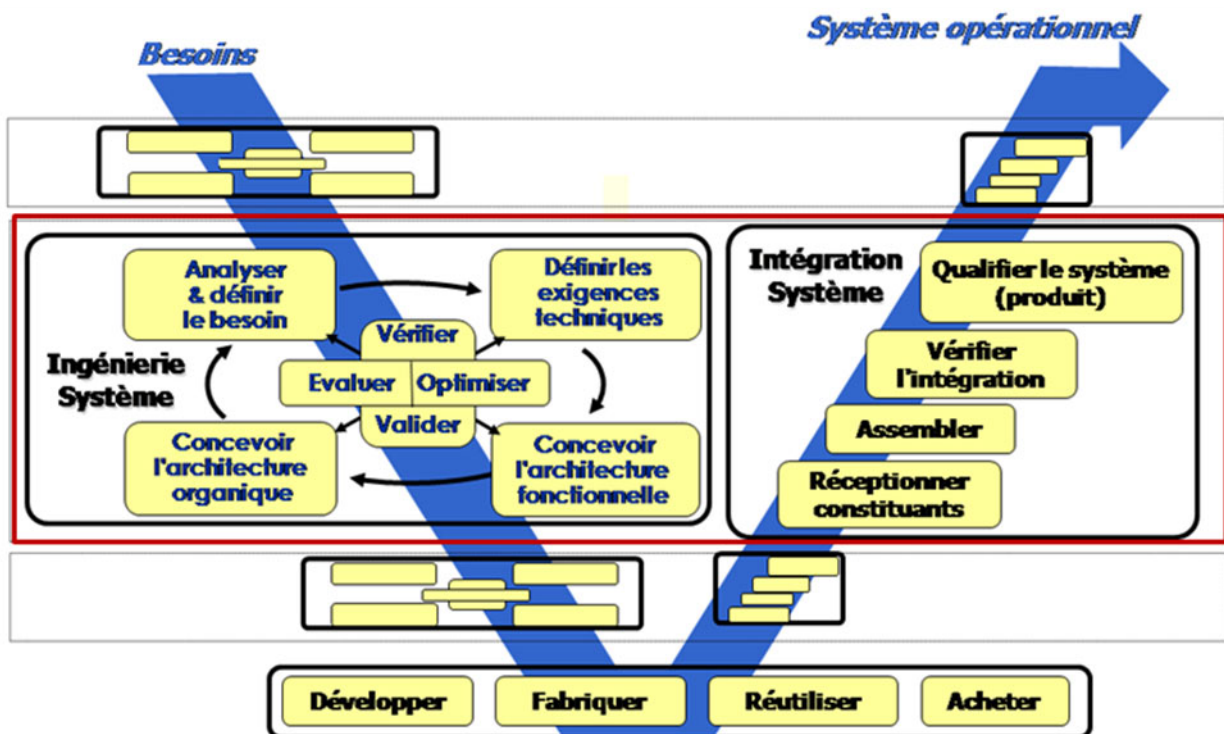


Figure 15 : Cycle de développement de systèmes

Cette logique de décomposition du système est décrite par la branche descendante du cycle de développement en V (activités de conception), alors que sa branche ascendante décrit les processus d'intégration, à savoir, la réception, l'assemblage, la vérification de l'intégration et la qualification. La Figure 15 décrit ce cycle pendant lequel le système est décomposé en sous-systèmes, qui sont à leur tour décomposés, jusqu'à la plus petite entité appelée constituant. Un bloc système est une entité de cette décomposition sur lequel opèrent de façon cohérente les activités de conception et d'intégration (rectangle à angles droits dans la Figure 15).

En analysant le cycle en V de la Figure 15, on retrouve les éléments introduits dans la première partie de ce chapitre et nécessaires au bon déroulement de la conception. En effet, ce modèle offre une vue temporelle guidant le processus de conception des besoins initiaux à la définition du système. Cet axe est décomposé en trois grandes étapes successives : Ingénierie, Réalisation et Intégration.

Chacune de ces étapes comprend un ensemble de phases et/ou de processus définis par des activités. Ceci donne au processus de développement une granularité et un niveau de détail semblables aux modèles basés sur la notion d'activité (cf. section I).

Le déroulement du processus de développement est donc progressif et itératif. A chaque itération affinant le système en train d'être conçu, la définition du besoin et des exigences, les architectures organique et fonctionnelle des blocs systèmes, évoluant en parallèle, sont complétées par de nouvelles exigences et contraintes.

La conception, telle qu'elle est décrite dans la première partie du chapitre, n'évoque pas, ou évoque de façon implicite, l'intégration. Dans le cycle en V, les objectifs de tests et la description de la recette du produit final sont définis pendant le déroulement des activités de la branche descendante. Nous considérons que les besoins d'ingénierie du processus d'intégration (réception, assemblage, vérification du système et sa qualification), délicate pour les produits multidisciplinaires car pouvant nécessiter la réalisation de systèmes dédiés tels que des bancs de tests, doivent être pris en compte plus fortement lors de la phase descendante du cycle en V.

3. Les processus de Ingénierie Système

L'IS est donc définie comme un ensemble de « Processus » qui peuvent être :

- Soit des processus d'ingénierie, correspondant à la notion de « phase » des modèles de conception décrits dans la première partie de ce chapitre,
- Soit des processus support de l'ingénierie, correspondant à certaines « activités » communes à ces mêmes modèles de conception.

L'ensemble de ces processus est appliqué au système à concevoir ainsi qu'à tous les sous-systèmes. On retrouve donc les phases permettant de décrire la conception en termes de livrables. Ainsi définit-on les grandes étapes du projet. On retrouve également la notion d'activité définie dans la première partie de ce chapitre lorsque l'on conçoit les sous-systèmes. La notion de granularité plus fine est présente dans ce modèle par la décomposition du système en sous-systèmes, et du problème en sous-problèmes. On retrouve également les activités qualifiées ici (en IS) de processus support à l'ingénierie.

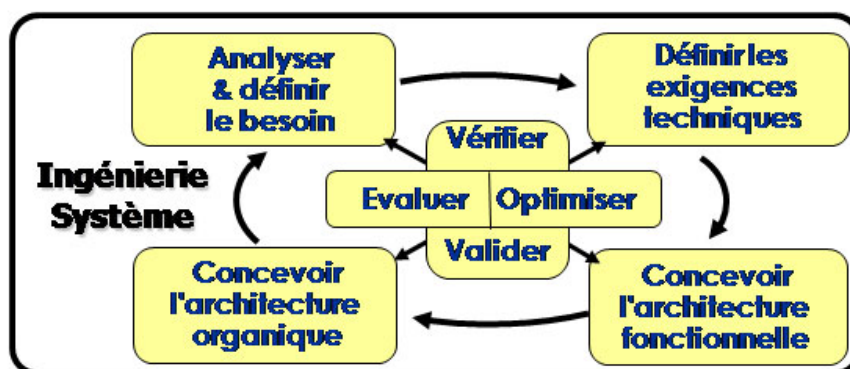


Figure 16 : Processus techniques d'IS : « Phases » de conception et « activités support »

A. Processus Ingénierie

L'IS, en tant que processus, peut être définie comme une transformation d'un besoin en une solution par la mise en œuvre de quatre processus. Les deux premiers, faisant partie du domaine du problème (vue externe : le quoi) où l'on spécifie le système à concevoir, permettent de bien définir le besoin ainsi que les exigences techniques. Ensuite, dans le domaine de la solution (vue interne : le comment), la conception du système passe par les processus de conception des architectures fonctionnelle et organique. Cette distinction nous donne un aperçu supplémentaire sur la notion de domaines que prend en compte l'IS.

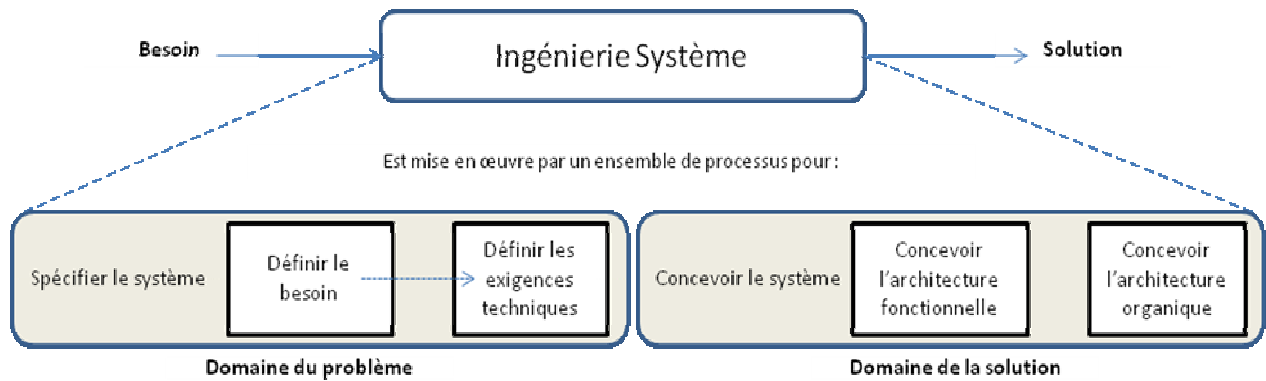


Figure 17 : Processus d'ingénierie ; distinction entre domaines du problème et de la solution

Comme mentionné par ailleurs, ces quatre processus d'ingénierie sont appliqués de manière itérative pour l'ensemble des sous-systèmes qui constituent le système global. Les entrées/sorties de chaque processus servent à améliorer, affiner ou consolider les processus suivants ou bien remettre en cause certains processus précédents. C'est ainsi que les éléments d'ingénierie deviennent de plus en plus matures. Généralement, et comme le montre la Figure 15, plusieurs itérations sont nécessaires afin d'obtenir un ensemble cohérent de besoins, d'exigences techniques, d'architecture fonctionnelle et d'architecture organique.

- Définir le besoin :

Ce processus consiste à décrire, en langage naturel, et d'une façon précise, détaillée et synthétique le besoin des parties prenantes. Il s'agit des besoins formulés par le client quand il s'agit du système global, et des équipes, internes ou externes, responsables du développement des sous-systèmes dans le cas échéant.

En termes de livrable, il s'agit du Cahier de Charges (CdC) du système dans lequel sont décrits les limites physiques, les limites fonctionnelles. Les échanges avec l'extérieur sont décrits dans la Spécification d'Interfaces (SI) du système.

En termes de domaine, ce processus permet de décrire le contexte du système tel qu'il est vu par l'utilisateur.

- Définir les exigences techniques :

Ce processus consiste à transformer le CdC du système validé par l'utilisateur en des exigences techniques et des contraintes. Il s'appuie notamment sur l'analyse des scénarios opérationnels décrits de façon exhaustive afin de générer la Spécification Technique (ST) du système.

Une exigence est définie par l'AFIS comme une « propriété ou caractéristique dont l'obtention est jugée nécessaire pour atteindre un but donné ».

Les exigences peuvent être imposées par l'utilisateur ou bien elles sont induites par les choix de conception, par exemple technologiques, effectués au niveau du bloc sous système père ou lors des itérations précédentes.

En termes de domaine, ce processus permet de décrire le contexte du système tel qu'il est vu par les concepteurs.

- **Conception de l'architecture fonctionnelle :**

Ce processus consiste à trouver une architecture fonctionnelle au système répondant aux exigences et contraintes définies précédemment. En effet, c'est à ce niveau que commence la définition de la solution (domaine de la solution) en apportant une réponse à la question « Comment marche le système ? ». L'architecture fonctionnelle du système (le résultat de cette activité) est une vue dynamique/comportementale du système comprenant les fonctions du système (i.e. les transformations), les flux échangés entre ces fonctions, et la manière dont les fonctions s'enchaînent les unes avec les autres. L'architecture fonctionnelle générée est dite « idéale » car indépendante des constituants, mais pas indépendante de la science. Cette architecture peut également être issue d'un certain nombre de choix successifs et d'activité d'optimisation.

- **Conception de l'architecture organique :**

Ce processus prend en compte les exigences non fonctionnelles (performances, contraintes) pour trouver une architecture organique (physique) portant les fonctions et satisfaisant au mieux les exigences non fonctionnelles. Ce processus apporte une réponse à la question « Comment le système doit faire ce qu'il a à faire ? » en définissant la structure interne telle qu'elle est vue par le concepteur. Une architecture organique montre les constituants (qui exécutent les fonctions) reliés par les liens qui transportent les flux.

Ce processus génère un Dossier de Conception (DD) du bloc système considéré.

D'autres livrables tels que le Dossier Justificatif ou le Plan d'Intégration Validation, constituent une interface avec le processus d'Intégration du même bloc système. L'ensemble de ces processus est supporté par des activités support permettant les Vérification/Validation et Evaluation/Optimisation des solutions choisies au cours de l'Ingénierie.

B. Processus Support Ingénierie

Le but des activités support est de *vérifier* et *valider* les choix qui ont été faits lors de chaque processus afin de s'assurer que l'on a bien conçu, mais aussi de ne pas fournir une solution au problème mais plutôt d'*optimiser* la réponse. L'IS dégage donc quatre activités support essentielles : la vérification, la validation, l'évaluation et l'optimisation.

Vérification

La vérification a pour but de montrer que l'activité a bien été faite, en conformité à son plan de réalisation et qu'elle n'a pas introduit de défaut dans le résultat (ainsi définit-on progressivement le système selon les règles). Elle peut se faire notamment sur les états intermédiaires successifs du produit de l'activité [Faisandier'05b].

Validation

La validation a pour but de montrer que l'activité s'est conformée à son objectif, que le résultat de l'activité répond au besoin pour lequel l'activité a été faite (ainsi définit-on progressivement le bon système). Elle se fait en vérifiant la conformité du produit de l'activité à ses exigences de besoin [Faisandier'05b].

Evaluation

Evaluer c'est déterminer puis analyser les caractéristiques importantes des solutions de façon à :

- Mesurer le niveau de satisfaction par rapport au besoin exprimé ;
- Comparer les solutions candidates entre elles.

Optimisation

Optimiser c'est rechercher le juste produit qui répond au juste besoin, c'est à dire :

- Extraire, après évaluation, la solution qui répond le mieux au problème ;
- Améliorer encore cette solution en utilisant les résultats de l'évaluation.

4. Synthèse

En Ingénierie Système (IS), on considère l'ensemble du cycle de vie du produit lors de la conception. En effet, le processus d'ingénierie est mené en parallèle avec la conception des

systèmes contributeurs (systèmes de réalisation, de maintenance, de mise en service...). Elle intègre également l'ensemble des acteurs des différentes disciplines dès le début du processus de conception. Pour ces raisons, elle permet à chaque domaine de la conception d'avoir une vue synthétique sur le déroulement de la conception.

Lors de ces processus, le système à concevoir est décomposé en sous-systèmes. Le développement de l'ensemble des sous-systèmes se déroule d'une manière parallélisée. Chaque sous-système passe donc par des étapes d'Ingénierie, de Réalisation et d'Intégration. Ces étapes constituent le cycle de développement connu sous le nom de cycle en V.

En ingénierie, et pour chaque sous-système, les besoins sont définis et analysés puis transformés en exigences techniques. Ensuite, les architectures fonctionnelle et organique sont définies. Ces processus d'ingénierie sont supportés par des activités de vérification/validation et d'évaluation/optimisation. Ces activités support servent, à chaque étape à justifier les choix de conception et à converger graduellement dans l'espace des solutions, par l'élimination de celles qui violent les nouvelles contraintes.

Cette réduction progressive du champ d'investigation peut être perçue à la fois positivement et négativement. D'un côté, ce qui est attendu par la conception est la définition de la « meilleure » solution et la réduction progressive du champ d'investigation fait gagner du temps lors des itérations successives. D'un autre côté, le fait d'éliminer des possibilités peut éliminer prématurément des pistes de solutions.

Conclusion

Le processus de conception est complexe quand il s'agit de développer des systèmes où interviennent différents métiers, différentes compétences et par suite différentes contraintes et objectifs non compatibles. Les contraintes et les exigences induites par les concepteurs pouvant ne pas être toutes identifiées dès le début de la conception, plusieurs itérations peuvent être nécessaires afin de trouver une architecture satisfaisant le besoin du client.

D'après [Matthews'02], la phase de conception préliminaire, pendant laquelle est effectuée la recherche de concept et de solutions candidates au problème, est souvent guidée par les habitudes professionnelles des concepteurs. C'est pour cela qu'il est nécessaire de choisir entre les architectures qui respectent le cahier des charges, les règles métiers, les contraintes de fabrication, celle qui apporte la meilleure valeur ajoutée au produit... D'où des itérations de type « essai-erreur » de l'activité d'optimisation de l'architecture du système à concevoir.

D'après Le Coq [Le Coq'92], les choix de l'architecture et des composants du futur produit, souvent réalisés par une démarche itérative et du type essai-erreur, sont très coûteux en temps.

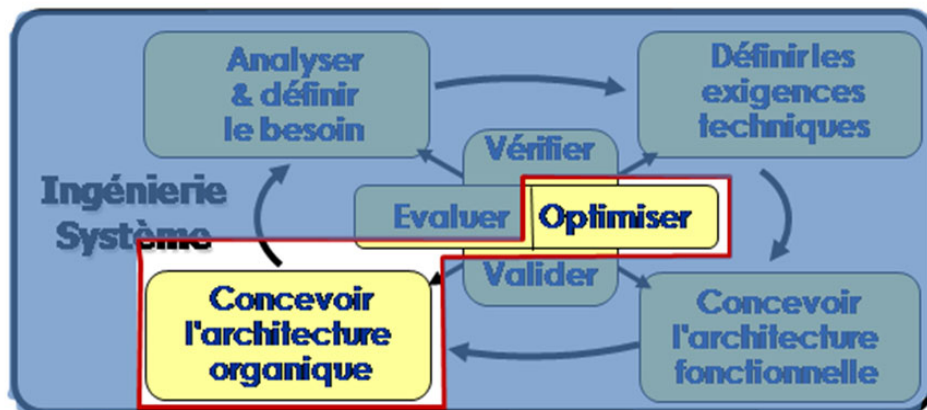


Figure 18 : Optimisation en conception organique

La définition de l'architecture organique du système nécessite de traiter un nombre important de paramètres correspondant aux variables de conception. Dans ce travail, on ne considère à cette étape du cycle de développement, que des variables exprimées numériquement. Dans la plupart des cas, ces variables peuvent prendre plusieurs valeurs.

Dans la conception de systèmes multidisciplinaires, le choix de ces variables peut s'avérer difficile et coûteux en temps pour les raisons suivantes :

- **Continuité/Discontinuité :**

D'une part, certaines variables de conception sont continues. Par exemple les grandeurs physiques : positions, longueurs, vitesse, accélération, force... D'autre part certaines variables, comme par exemple le choix d'un composant dans un catalogue, sont discontinues.

- **Complexité combinatoire :**

En plus des variables de conception du système global, chacun des sous-systèmes possède des variables propres dont le choix doit être optimisé. Le nombre élevé de combinaisons possibles peut devenir coûteux en temps de calcul.

- **Couplage entre les disciplines :**

Certaines variables de conception sont nécessaires à la définition de plusieurs sous-systèmes. Il peut y avoir des appréciations différentes entre les concepteurs de ces sous-systèmes quant aux choix de ces variables, dites couplées. Ce problème est plus apparent lorsque les sous-systèmes sont conçus par des équipes différentes et notamment, de façon interdisciplinaire.

- **Objectifs contradictoires :**

Afin d'optimiser chaque sous-système, chaque équipe a tendance à prendre en compte que ses propres objectifs. Ceci est dû à leur visibilité partielle des variables à instancier et des objectifs à atteindre. Ce qui peut entraîner des situations conflictuelles, si les objectifs à atteindre sont contradictoires.

Dans ce contexte, notre contribution vise à fournir en un temps acceptable, aux concepteurs un ensemble de solutions proche de l'optimum, faisant apparaître une marge de manœuvre et facilitant la recherche d'un compromis.

Chapitre 2 : Optimisation en conception multidisciplinaire

Introduction

Dans le premier chapitre, nous avons introduit le cadre de l'Ingénierie Système regroupant un ensemble de méthodes et couvrant l'ensemble des vues sur le produit depuis l'expression des besoins jusqu'au dossier de conception.

Dans ce cadre nous avons situé les activités de support avec un intérêt particulier pour l'activité support d'optimisation. En effet, nous avons dégagé un ensemble de problématiques liées à cette activité dans le cas d'une conception multidisciplinaire comme par exemple, le couplage entre les disciplines ou la multiplicité des objectifs.

Dans ce chapitre, nous traitons la problématique de couplage entre les disciplines. A cet effet, nous introduisons la notion de formulations d'un problème d'optimisation multidisciplinaire et les principales méthodes de résolution. Nous distinguons deux stratégies permettant d'aborder les problèmes de couplage. Dans la première, appelée optimisation mono-niveau, le système à optimiser est considéré comme étant un seul bloc et on dispose d'un optimiseur unique. Quant à la deuxième, le processus d'optimisation apparaît au niveau de chaque discipline ainsi qu'au niveau du système global, d'où l'appellation de stratégie multi-niveaux.

I. Optimisation en conception multidisciplinaire

1. Définitions et terminologie

Dans ce qui suit, nous explicitons les définitions que nous utilisons.

- Variable disciplinaire ou privée : C'est une variable qui n'est **utilisée que par une seule discipline**, appelée aussi variable locale ou interne.
- Variable partagée : C'est une **entrée globale** du système dont la valeur est imposée à toutes les disciplines. Elle est publique.
- Variable de conception : elle peut être partagée ou disciplinaire.
- Variable de couplage : C'est une variable qui est **calculée par une discipline et utilisée par une autre**. Elle représente une des entrées d'une discipline et l'une des sorties d'une autre. Elle est publique.
- Contrainte : C'est une **limite au choix du concepteur**. En général, elle porte sur des lois physiques, des règles métier ou des limites imposées par les normes.

- Contrainte interdisciplinaire : C'est une **contrainte de cohérence** entre les disciplines.
- Analyse : Analyser, c'est **résoudre** un système d'équations afin de dégager des solutions (valeurs de conception) pouvant garantir une évaluation valide.
- Analyse interdisciplinaire : Elle permet de **résoudre les équations de couplage** entre disciplines.
- Evaluation : Evaluer une solution, c'est **calculer** le résultat de systèmes d'équations **(et non résoudre)**.

2. Problématique

La conception multidisciplinaire implique généralement plusieurs composantes interagissant entre elles et issues de disciplines différentes. L'optimisation globale du système à concevoir ne se limite pas à l'optimisation de ses composantes du fait du couplage (faible ou fort) entre les disciplines. Un problème d'optimisation multiobjectif en conception multidisciplinaire est considéré comme un problème difficile à résoudre.

La Figure 19 représente deux exemples de couplage (faible à gauche et fort à droite) entre deux disciplines D_1 et D_2 .

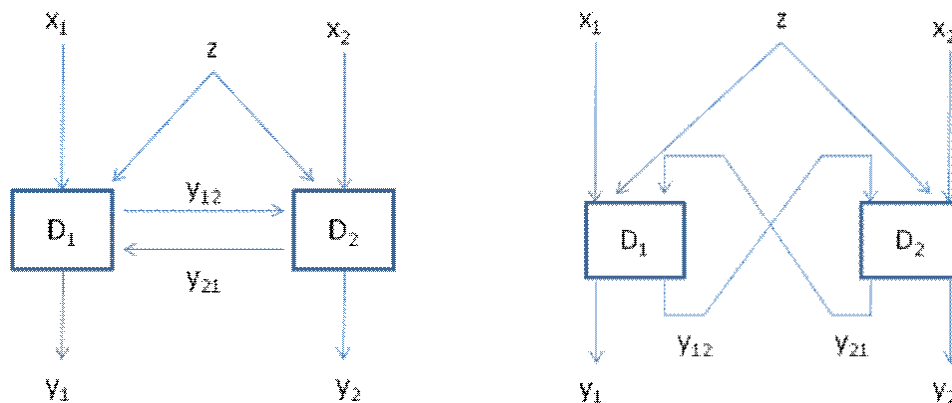


Figure 19 : Exemple de couplage entre les disciplines D_1 et D_2 : faible à gauche et fort à droite

Deux disciplines sont dites couplées lorsque l'analyse d'une composante disciplinaire nécessite un (ou des) résultats d'analyse de l'autre. Ce couplage est dit faible lorsque l'analyse peut être résolue séquentiellement (Figure 19 gauche). Il est dit fort lorsque des résultats d'analyse de chaque composante est nécessaire pour mener l'analyse de chacune d'elles, ce qui rend les calculs séquentiels impossibles (Figure 19 droit).

Plusieurs méthodes permettant de résoudre ce problème de couplage ont été introduites dans la littérature. Dans la section suivante, nous commencerons par énoncer une formulation générale d'un problème d'optimisation multidisciplinaire. Dans la section 4, nous détaillerons quelques exemples de formulations utilisées par des approches différentes afin de pallier le problème de couplage.

3. Formulation d'un problème OMD

Un problème d'optimisation multidisciplinaire (OMD) peut être représenté de la manière simplifiée suivante :

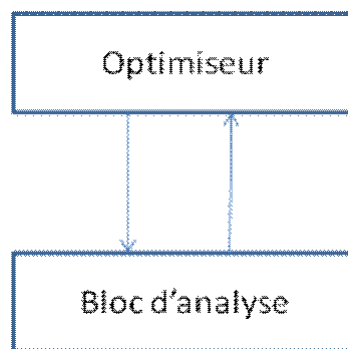


Figure 20 : Approche standard d'optimisation

Dans la Figure 21, nous détaillons le bloc analyse de la Figure 20 afin de montrer l'interaction entre les différentes disciplines. En effet, dans l'exemple qui nous sert d'illustration dans ce chapitre, le système global est composé de trois composantes analysables selon des connaissances dans des domaines disciplinaires différents notés D_1 , D_2 et D_3 , avec un cas de couplage fort et un cas de couplage faible.

Les notations des Figure 21 et Figure 22 sont les suivantes :

- n : nombre de disciplines ($n=3$ dans les cas illustré en Figure 21)
- x_i : est l'ensemble des variables disciplinaires de la discipline i
- y_i : est l'ensemble des sorties de la discipline i . Elles entrent dans le calcul des fonctions objectifs et des contraintes du système,
- y_{ij} : est l'ensemble des variables de couplage, sorties de la discipline i et entrées de la discipline j ,
- X : L'ensemble des variables disciplinaires,

- Z : L'ensemble des variables partagées,
- V : L'ensemble des variables partagées et disciplinaires, $V = Z \cup X$
- Y : L'ensemble des sorties de l'analyse du système,
- M : L'ensemble des variables de couplage
- F fonction objectif
- C contraintes

Le problème d'optimisation (de minimisation, dans l'exemple) est décrit par ce schéma est le suivant :

Minimiser $F(V)$

Sous les contraintes

$C(V) \leq 0$

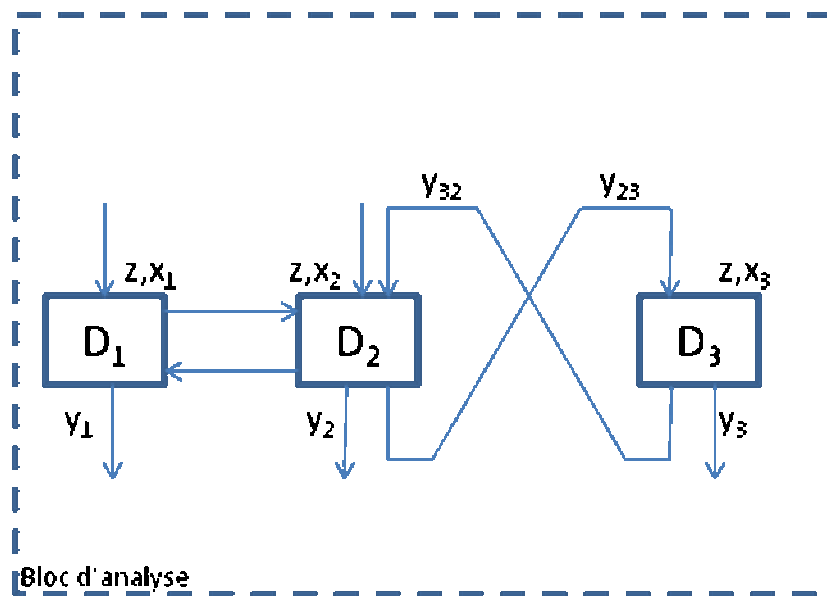


Figure 21 : Bloc d'analyse détaillé pour un système à trois disciplines, avec un couplage faible entre D1 et D2 et un couplage fort entre D2 et D3

Les disciplines D_1 et D_2 sont faiblement couplées, c'est-à-dire qu'au cours de l'analyse d'une solution proposée par l'optimiseur, les valeurs des variables de couplage entre ces deux disciplines peuvent être déterminées à travers des interactions séquentielles.

Par contre, entre les disciplines D_2 et D_3 , le couplage est dit fort. C'est-à-dire que les entrées de chaque discipline dépendent des sorties de l'autre. C'est ce qui rend les variables de

couplage plus difficiles à évaluer et théoriquement incalculables séquentiellement sans hypothèse sur leur valeur. La Figure 22 montre un exemple de couplage fort entre 3 disciplines :

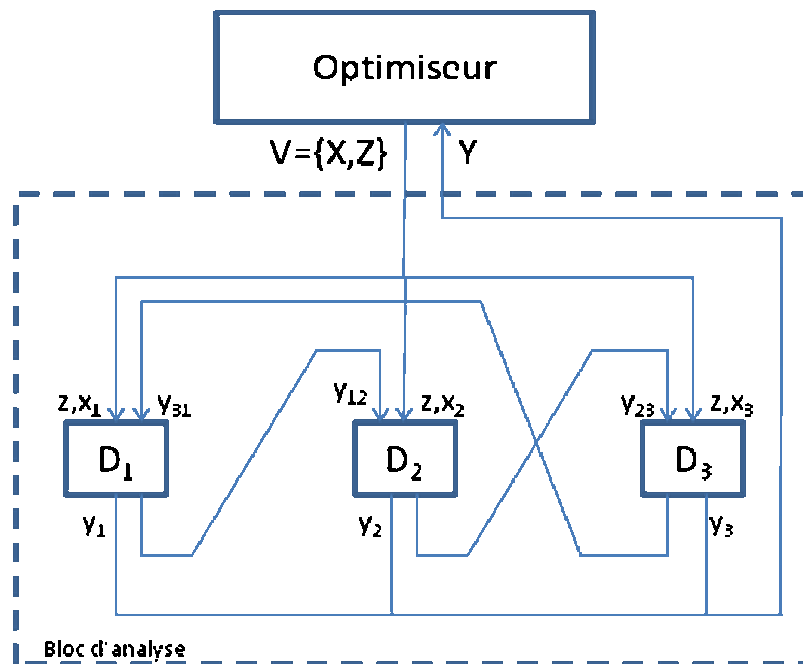


Figure 22 : Bloc d'analyse détaillé pour un système, avec couplage fort entre disciplines D1, D2 et D3.

Les blocs adjacents sont faiblement couplés, c'est-à-dire qu'au cours de l'analyse d'une solution proposée par l'optimiseur, les valeurs des variables de couplage entre deux blocs adjacents peuvent être déterminées à travers des interactions séquentielles.

Mais, du fait du cycle d'attente des résultats, le couplage est fort, c'est-à-dire que les entrées de chaque discipline dépendent des sorties l'une de l'autre.

Dans ce qui suit, nous allons introduire certaines méthodes d'optimisation multidisciplinaire proposées afin de résoudre la problématique du couplage fort entre les disciplines. Dans certaines d'entre elles, la résolution se fait par l'optimiseur et dans d'autres, au cours de l'analyse système.

II. Stratégies d'optimisation multidisciplinaire

La littérature introduit des notions de formulation, d'approche, de méthode ou encore de stratégie de résolution. Nous utiliserons dans ce manuscrit ces termes définis de la manière suivante :

- « Stratégie de résolution ou Approche » : Une stratégie permet de distinguer entre les cas où le système est considéré comme un bloc unique et où le système est décomposé en plusieurs blocs disciplinaires.
- « Méthode de résolution » : Après avoir choisi la stratégie de résolution, les méthodes sont des techniques et algorithmes d'optimisation appliqués afin de résoudre le problème du couplage entre les disciplines.
- « Formulation » : Formuler un problème d'optimisation multidisciplinaire revient à définir toutes les composantes de ce problème, à savoir, les contraintes, les objectifs, les variables publiques et privées et les systèmes d'équations liant les variables.
- « Reformulation » : Reformuler un problème, c'est changer au moins une composante de la formulation courante.
- « Système d'équations » : C'est un ensemble d'équations, appelées quelquefois équations d'état permettant de déterminer les valeurs des sorties du système en fonction des valeurs des variables de conception (ses entrées).

Certains travaux, et notamment dans [Cramer'93], [Flager'07], [Balling'94], [Brown'06], [Carrier'04], [Kodiyalam'98] insistent sur l'importance des différentes formulations d'un même problème. N. M. Alexandrov a introduit dans [Alexandrov'99, 00a, 00b] une classification intéressante des différentes formulations des problèmes d'OMD basée sur l'aspect ouverture/fermeture de l'analyse disciplinaire, des contraintes de conception, et des contraintes interdisciplinaires. Une formulation est dite fermée (stricte) lorsqu'à chaque étape d'optimisation, toutes les contraintes doivent être satisfaites.

D'autres travaux, et notamment dans [Clément'09], [Duranté'04], [Parte'04], [Coelho'09a] portent d'avantage sur les méthodes de résolution. On trouve la notion de stratégie de résolution (mono ou multi-niveau(x)).

Suite à l'analyse des travaux énoncés précédemment, nous distinguons deux façons d'aborder le problème : choisir une stratégie de résolution et ensuite une méthode de résolution en fonction de la nature et des caractéristiques du problème à résoudre et, en fonction de ces choix, formuler le problème en dégageant le système d'équations qui en découle, ou bien : choisir une formulation puis les stratégies et méthodes de résolution les mieux adaptées à cette formulation. La Figure 23 met en évidence ces deux tendances.

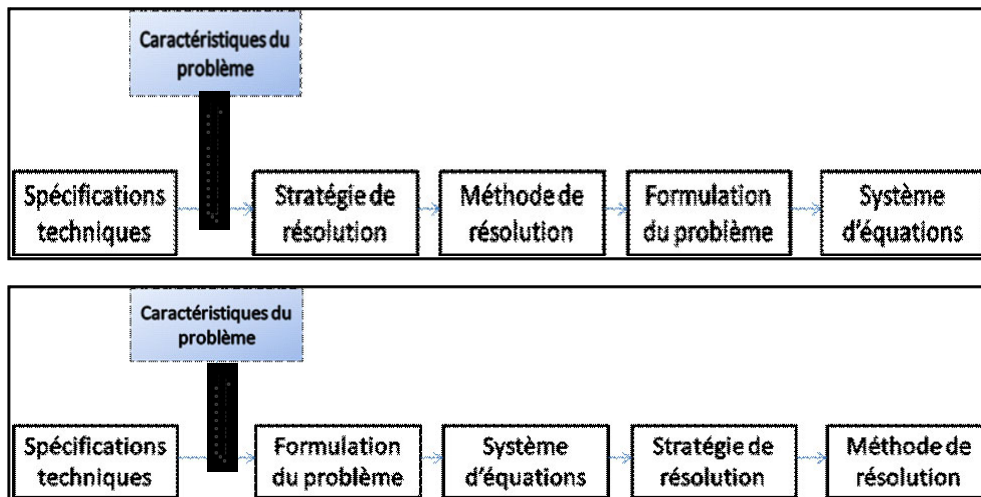


Figure 23 : Processus de résolution d'un problème d'optimisation multidisciplinaire

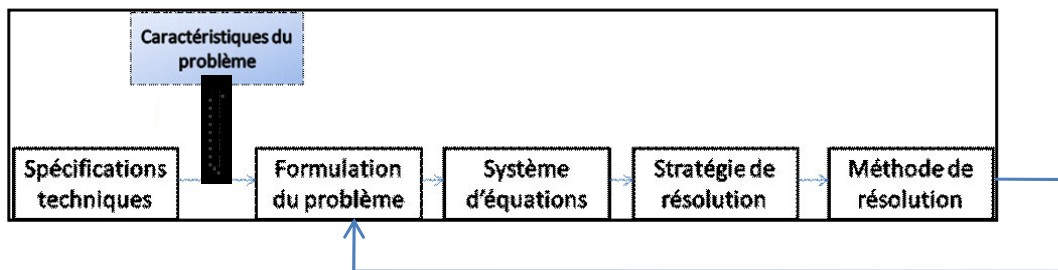


Figure 24 : Processus d'un problème d'optimisation multidisciplinaire avec reformulation

La Figure 23 illustre que reformuler un problème peut permettre de le résoudre différemment et avec un optimiseur plus adapté. En effet, certains optimiseurs sont plus efficaces¹ que d'autres sur certains problèmes. Reformuler un problème c'est redéfinir son système d'équations pour avoir un système équivalent mais étant plus ou moins optimisable. Plus particulièrement, les formulations ayant des paramètres et des modèles multi-niveaux sont plus coûteux et difficiles à résoudre [Coelho'09a].

S'agissant d'une relaxation de contraintes, d'une approximation des fonctions objectifs, ou bien d'une transformation des objectifs en contraintes, la reformulation d'un problème peut nuire à la convergence et à la robustesse de l'algorithme utilisé pour l'optimisation. En effet, la reformulation induit la redéfinition des informations partagées par les différentes disciplines ainsi que le volume d'information échangé. Cette manipulation risque d'élargir la

¹ L'efficacité d'un optimiseur est jugée par rapport à la qualité de la solution obtenue et du temps de calcul nécessaire.

dimension du problème d'optimisation en introduisant de nouvelles variables, de nouvelles équations d'état, ou bien de nouveaux couplages entre les disciplines.

Dans la suite nous détaillons certaines des méthodes OMD (optimisation multidisciplinaire) les plus répandues et utilisées [Clément'09] [Duranté'04], [Parte'04], [Carrier'04], [Hulme'00]. Ces méthodes sont classées suivant la stratégie de résolution ou d'optimisation : globale (mono-niveau) ou disciplinaire (multi-niveaux).

1. Stratégies mono-niveau

Dans les approches mono-niveau, le problème d'optimisation multidisciplinaire est considéré comme un ensemble de paramètres à optimiser. Un optimiseur global et unique assure le processus d'optimisation, et les disciplines sont considérées comme étant des outils d'analyse ou bien d'évaluation (le cas de l'approche AAO). Ces méthodes sont plus appropriées lorsqu'on dispose de toutes les données du problème : variables de conception (disciplinaires, partagées et de couplage) et contraintes (disciplinaires et interdisciplinaires) et équations d'état.

A. Multi-Discipline Feasible (MDF)

Cette approche restreint les variables à optimiser aux variables de conception en excluant les variables couplées. A chaque itération de l'optimisation, une analyse multidisciplinaire est effectuée afin d'assurer la faisabilité de la solution (ou consistance). Cette méthode OMD très utilisée [Hulme'00] présente l'inconvénient majeur d'être très gourmande en temps d'implémentation et d'exécution à cause de l'analyse multidisciplinaire du système global à mener à chaque itération. Elle suppose de centraliser les connaissances et les données.

La méthode MDF ne tient pas compte dans l'optimiseur des variables de couplage qui sont calculées par l'analyseur, et la formulation du problème d'optimisation devient donc :

Minimiser $F(V)$;

- Avec $V = Z \cup X$;

Sous les contraintes disciplinaires

$C(V) \leq 0$;

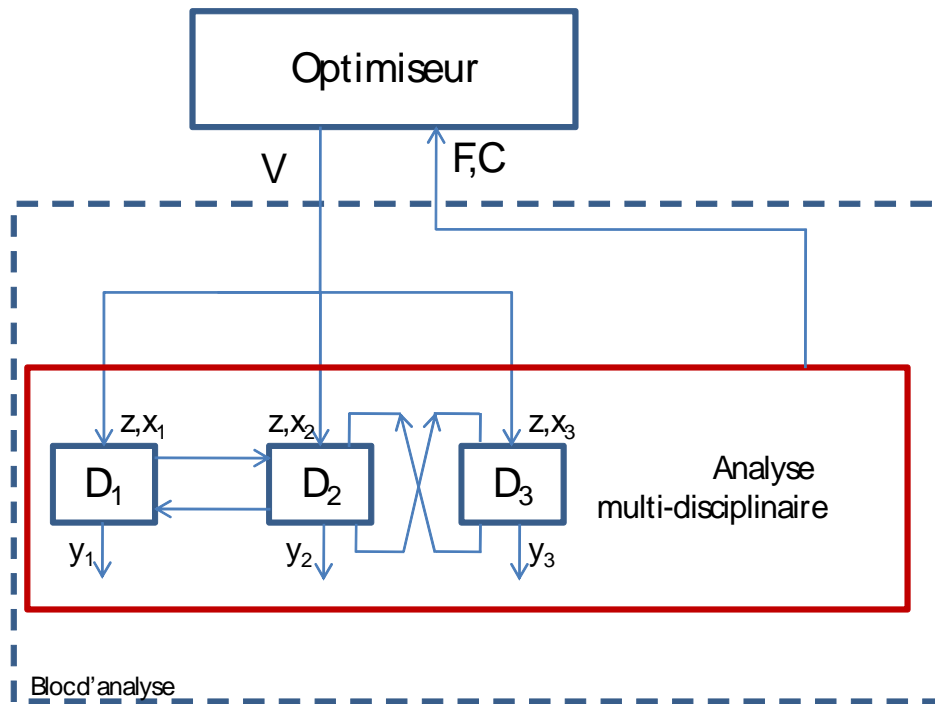


Figure 25 : Approche MDF

B. All-At-Once (AAO)

Contrairement à la méthode MDF, l'approche AAO prend en compte les variables couplées et les intègre à l'ensemble de variables à optimiser. L'évaluation d'une solution ne se fait qu'à la rencontre d'un optimum local ou global [Brown'06].

Dans cette approche, nous parlons bien d'évaluation disciplinaire et non d'analyse disciplinaire car aux contraintes $C(V)$ s'ajoutent des contraintes vérifiant l'égalité entre les variables couplées calculées par le bloc d'analyse et les valeurs choisies par l'optimiseur. Les équations d'état ne sont pas forcément satisfaites à chaque pas de l'optimisation. Elles doivent l'être à la convergence de l'algorithme.

Connue pour sa rapidité de calcul, l'inconvénient majeur de cette méthode est qu'elle complexifie le processus d'optimisation en augmentant le nombre de variables. De plus, la convergence n'est pas assurée, c'est-à-dire que puisque l'évaluation d'une solution ne se fait qu'aux optima, et donc si l'optimum global n'est pas atteint, la faisabilité de la solution n'est pas assurée.

Ayant un grand nombre de variables à optimiser, cette méthode présente également l'inconvénient de ne pas être la plus appropriée dans la phase de conception détaillée des systèmes les plus compliqués. Elle est communément utilisée comme un outil de vérification de solutions obtenues avec des approches multi-niveaux [Shubin'95].

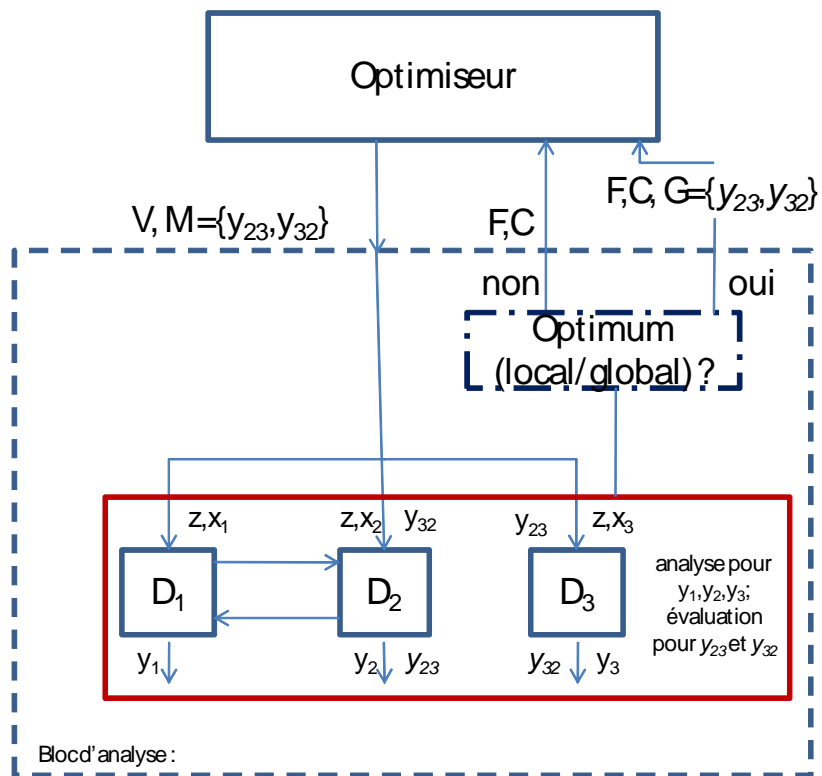


Figure 26 : Approche AAO

Les variables de couplage fort sont rajoutées à l'ensemble des variables d'optimisation. La faisabilité disciplinaire n'est vérifiée qu'à la convergence, c'est-à-dire la rencontre d'un optimum local ou global.

Des contraintes d'égalité sont rajoutées au système afin de vérifier l'adéquation entre les valeurs des variables de couplages proposées par l'optimiseur et celles utilisées par les évaluateurs disciplinaires. La formulation du problème d'optimisation devient donc :

Minimiser $F(V, y_{ij})$;

Sous les contraintes disciplinaires

$C(V, y_{ij}) = 0$;

Et les contraintes de cohérence interdisciplinaires dans le cas d'un optimum

$M - G = 0$; où G est l'ensemble des variables de couplage évaluées à partir des données fournies par l'optimiseur.

C. Individual-Discipline Feasible (IDF)

Cette méthode peut être considérée comme une approche intermédiaire entre les approches AAO et MDF. En effet, la méthode MDF requiert une faisabilité multidisciplinaire, la

méthode AAO ne vérifie la faisabilité de chaque discipline qu'aux optimums (locaux ou globaux), le principe général de la méthode IDF est d'assurer la faisabilité disciplinaire à chaque itération.

Comme pour l'approche AAO, en IDF les variables couplées sont transformées en variables à optimiser. En effet, à chaque convergence de l'optimiseur, les contraintes d'égalités rajoutées sont vérifiées afin de trouver un équilibre : une concordance interdisciplinaire entre l'ensemble des variables couplées.

Le processus d'optimisation de l'approche IDF est plus (resp. moins) complexe que celui de l'approche MDF (resp. AAO), et la résolution est plus (resp. moins) gourmande en coût et en temps que l'approche AAO (resp. MDF).

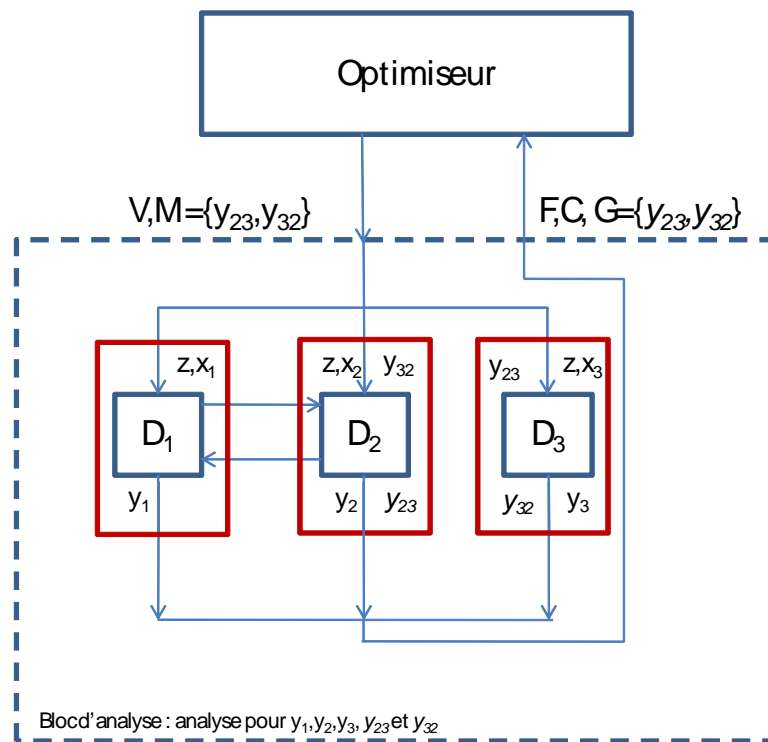


Figure 27 : Approche IDF

Le schéma montre qu'à chaque itération du processus, une analyse disciplinaire est effectuée. La faisabilité d'une solution est assurée par les contraintes d'égalité sur les variables de couplage dont les valeurs sont imposées par l'optimiseur. La formulation du problème d'optimisation devient donc :

Minimiser $F(V, y_{ij})$;

Sous les contraintes disciplinaires

$$C(V, y_{ij}) = < 0 ;$$

Et les contraintes de cohérence interdisciplinaires à chaque itération

$M - G = 0$; où G est l'ensemble des variables de couplage évaluées à partir des données fournies par l'optimiseur.

2. Stratégies multi-niveaux

Les méthodes disciplinaires (ou multi-niveaux) d'optimisation multidisciplinaire font intervenir des optimiseurs au niveau du système ainsi qu'au niveau des sous-systèmes. Les disciplines ne sont plus considérées comme de simples outils d'analyse mais plutôt comme des optimiseurs locaux. Ces approches sont plus adaptées lorsqu'une partie du système global peut (doit) être considéré comme une boîte noire avec des entrées et des sorties ; ceci est en général dû à la confidentialité inter-discipline. Parfois, certaines méthodes mono-niveaux, et notamment la méthode AAO, sont utilisées dans le but de vérifier la pertinence d'une solution obtenue avec une méthode multi-niveaux. La Figure 28 montre le principe général des approches multi-niveaux.

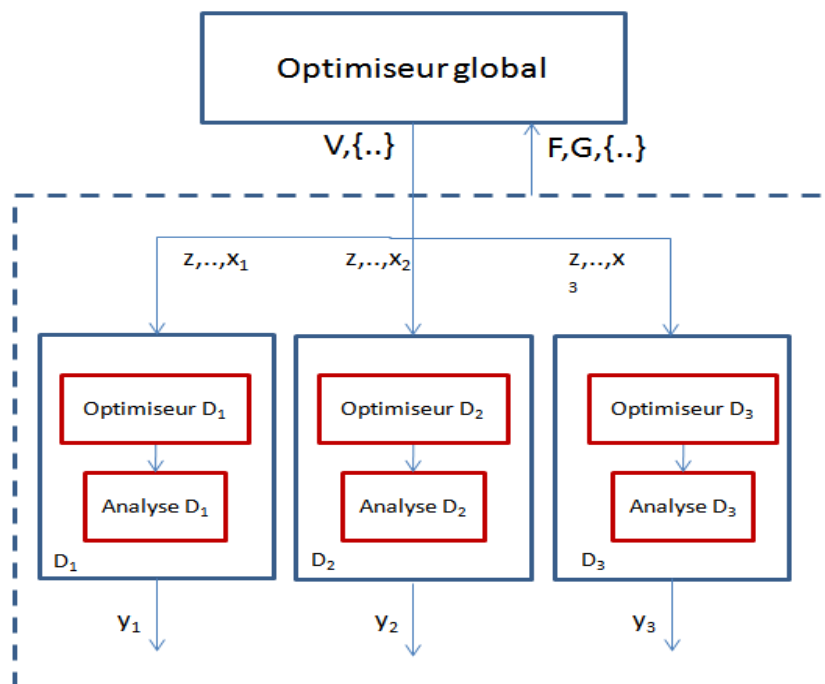


Figure 28 : Modèle résumant les approches multi-niveaux

Le principe général des approches multi-niveaux étant d'avoir un optimiseur global et un optimiseur local pour chaque discipline [Braun'96], plusieurs variantes ont été introduites

dont les plus répandues dans la littérature sont CO, CSSO, BLISS, DIVE... Dans la suite de cette section, nous allons introduire le principe général ainsi que les particularités de certaines de ces méthodes. Plus de détails, d'exemples et de comparaison de ces méthodes peuvent être trouvées dans [Braun'96], [Agte'06], [Yi'07], [Clément'09], [Coelho'09a]...

A. Collaborative Optimization (CO)

Dans cette formulation, l'optimiseur global choisit les valeurs des variables partagées et des variables de couplage. Ensuite chaque optimiseur disciplinaire doit satisfaire contraintes internes et détermine les valeurs des variables disciplinaires de façon à se rapprocher au mieux des valeurs des variables de couplage fixées par l'optimiseur global.

B. Collaborative SubSpace Optimization (CSSO)

Dans cette approche, à chaque optimiseur disciplinaire sont affectées quelques variables de couplage et les équations qui leurs sont rattachées. L'optimiseur global quant à lui est responsable du partage des contraintes dont chaque optimiseur disciplinaire a une responsabilité de satisfaire au mieux en cherchant son (ou ses) optimums.

C. Bi-Level Integrated System Synthesis (BLISS)

D'après Clément [Clément'09] et [Agte'06], cette approche construit un modèle réduit linéaire à partir des valeurs de sorties et de leurs gradients. Une optimisation globale du système n'est effectuée qu'après une optimisation locale pour chaque discipline. Ceci étant, chaque problème ou bien sous-problème d'optimisation est formulé comme un problème d'approximation linéaire.

D. Disciplinary Interaction Variable Elimination (DIVE)

Comme pour l'approche CO, sus-décrite, cette approche définit deux niveaux d'optimisation, l'un au niveau système et l'autre au niveau discipline. Ce qui les différencie, c'est l'ordonnement des différentes étapes d'optimisation. En effet, dans l'approche DIVE, on commence par les optimisations disciplinaires. Après avoir trouvé la solution optimisant l'objectif, les contraintes de couplage sont vérifiées et celles qui ne sont pas satisfaites sont passées à l'optimiseur global suite à une analyse multidisciplinaire. L'optimiseur global, en fonction de ces contraintes détermine les nouvelles valeurs des variables couplées qu'il impose aux optimiseurs disciplinaires.

D'autres approches multi-niveaux pour l'optimisation multidisciplinaires existent (MDOIS, MDA, DAO...). Des définitions détaillées de ces approches peuvent être trouvées dans [Coelho'09a], [Clément'09].

Conclusion

Dans ce chapitre, nous nous sommes intéressés au traitement du problème de couplage de variables en optimisation multidisciplinaire. Nous avons identifié des méthodes de résolution se regroupant en deux catégories : mono-niveau, et multi-niveaux. La Figure 29 permet de synthétiser l'ensemble des approches évoquées dans ce chapitre.

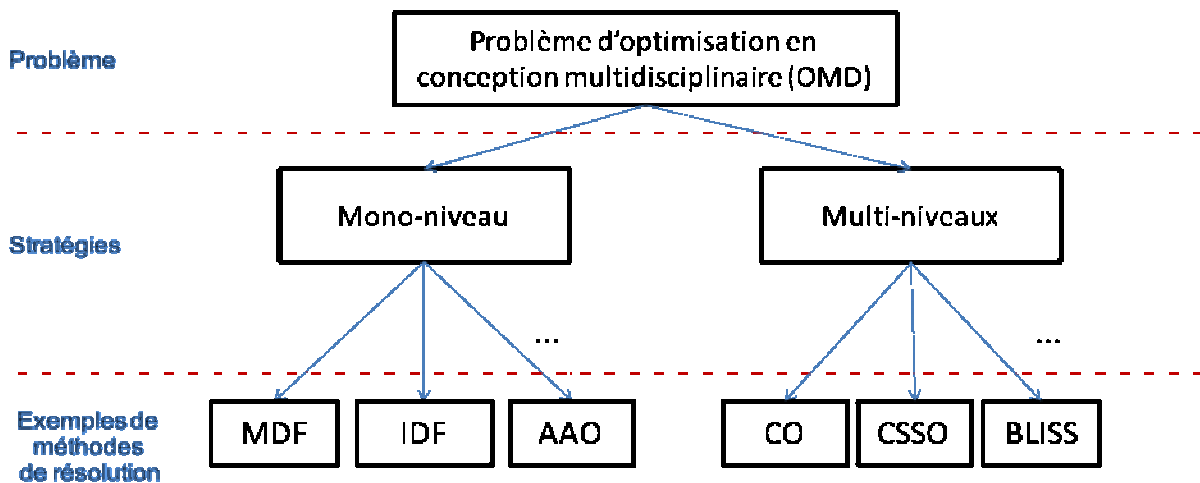


Figure 29 : Approches et exemples de méthodes de résolution d'un problème OMD

Les stratégies mono-niveaux peuvent s'appliquer lorsque l'on dispose de toutes les données du problème : variables, contraintes, et objectifs. Par contre, leur application devient impossible avec la présence de « boîtes noires » internes aux disciplines.

Ces méthodes d'optimisation multidisciplinaires permettent de résoudre la problématique du couplage entre les disciplines. Une fois le couplage levé, la résolution nécessite l'application d'une méthode d'optimisation. Cette problématique est abordée dans le chapitre suivant.

Chapitre 3 : Approches pour la résolution de problèmes d'optimisation multiobjectif

Introduction

L'optimisation numérique s'applique à différents domaines : économie, recherche opérationnelle, mathématiques, informatique, logistique, ingénierie. Ces problèmes peuvent être très difficiles à résoudre et il existe un éventail de méthodes utilisées dans le domaine de la recherche ainsi que dans l'industrie [Ehrgott'04], [Steuer'02].

Dans le cas de la conception organique de système pluridisciplinaire, les critères sont multiples et quelquefois contradictoires. Dans ce chapitre, nous considérons le problème d'optimisation multiobjectif qui confère à une architecture organique de très bonnes (et si possible les meilleures) performances tout en satisfaisant les contraintes. Nous privilégions les méthodes permettant de retarder la prise en compte des avis (préférences) des concepteurs multi-métiers afin d'éviter de restreindre prématurément le domaine de recherche en favorisant, sans raison objective, un critère métier par rapport à un autre. Nous faisons un choix parmi les méthodes généralement connues en mécanique sans pour autant prétendre à un état de l'art exhaustif du domaine qui a connu une forte activité ces dernières années.

Dans la première partie de ce chapitre, sont distinguées deux familles d'approches en optimisation multiobjectif. Dans la première, différentes méthodes reposent sur des techniques d'agrégation de critères en un seul. Le problème est ensuite résolu par des méthodes classiques d'optimisation mono-objectif. Dans la deuxième approche, des réponses sont apportées au problème en prenant en compte l'ensemble des critères. L'approche de Pareto étant souvent utilisée dans le domaine de l'optimisation en conception multidisciplinaire, il nous paraît nécessaire d'approfondir cette notion.

La deuxième partie de ce chapitre est consacrée aux méthodes de résolution d'un problème d'optimisation mono-objectif sur lesquelles on s'appuie en général. Un intérêt particulier est porté aux méthodes de résolution appelées méta-heuristiques qui ont inspiré nos travaux de recherche.

I. Approches de résolution d'un problème multiobjectif

Dans le cadre d'un problème d'optimisation multiobjectif, se pose le problème de la comparaison et du classement de solutions. Quelles sont celles réalisant un bon compromis entre les différents objectifs ?

La difficulté d'un problème multiobjectif est qu'il n'existe pas de définition d'une solution optimale. Le décideur exprime seulement qu'une solution est préférable, mais en général il n'existe pas une solution meilleure que toutes les autres.

Aussi définit-on la notion de dominance. Une solution est jugée meilleure qu'une autre si les valeurs obtenues selon tous les critères sont meilleures. Une solution qui domine une autre par rapport à quelques objectifs et qui est dominée par rapport aux autres, n'est pas comparable à celle-ci.

Une variante de cette approche considère qu'une solution est meilleure qu'une autre si le nombre d'objectifs où elle la domine est supérieur au nombre d'objectifs où elle est dominée par celle-ci.

Dans nos travaux, nous nous intéressons à la première version où la dominance doit être vérifiée sur l'ensemble des objectifs à optimiser.

La Figure 30 montre un ensemble de solutions placées dans l'espace des objectifs O_1 et O_2 , dans le cadre d'un problème de minimisation multiobjectif.

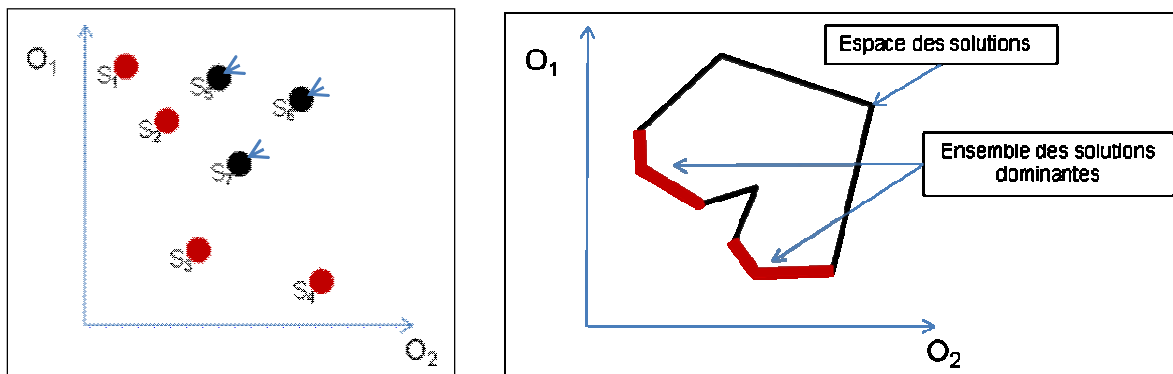


Figure 30 : Illustration de deux ensembles de solutions dans l'espace des objectifs ; l'ensemble des solutions non dominées étant représenté en rouge

D'après la définition de dominance ci-dessus, les solutions 1, 2, 3 et 4 dominent les solutions 5, 6, et 7, dans le cadre gauche. Les solutions de l'ensemble dominant n'étant pas comparables entre elles, représentent des optima et constituent le front de Pareto.

La notion de dominance de Pareto peut donc se traduire comme suit dans un cadre de minimisation :

Si les deux solutions S_1 et S_2 sont définies par les vecteurs de décision $\{S_{11}, S_{12}, \dots, S_{1n}\}$, et $\{S_{21}, S_{22}, \dots, S_{2n}\}$, avec n le nombre d'objectifs, alors une seule des trois propositions suivantes est valable :

S_1 domine S_2 si pour tout i dans $\{1, \dots, n\}$, $S_{1i} < S_{2i}$

S_2 domine S_1 si pour tout i dans $\{1, \dots, n\}$, $S_{2i} < S_{1i}$

S_1 et S_2 sont incomparables si aucune des deux relations précédentes n'est vérifiée.

Une autre nuance de la notion de dominance peut être rajoutée, à savoir la dominance stricte ou faible définie par :

S_1 domine faiblement S_2 si pour tout i dans $\{1, \dots, n\}$, $S_{1i} \leq S_{2i}$, par opposition à la dominance stricte où la supériorité de chaque élément de décision doit être stricte.

La notion de dominance de Pareto s'avère très utile et est utilisée dans le cadre d'un problème d'optimisation multiobjectif en conception multidisciplinaire [Coelho'09a]. On dira qu'une solution est efficace si elle est non dominée. Néanmoins, elle soulève deux critiques principales. En effet, on lui reproche un manque de flexibilité dans la définition de la relation de dominance, mais aussi la grande taille de l'ensemble optimal appelé front de Pareto.

Deux approches de résolution d'un problème d'optimisation multiobjectif peuvent être distinguées dans la littérature [Ehrgott'00], [Ehrgott'04][Roy'93][Roy'05]. La première consiste à ramener le problème à un problème d'optimisation mono-objectif. La deuxième consiste à proposer des solutions en tenant compte de l'ensemble des critères. La Figure 31 montre une classification possible des principales méthodes de résolution que nous présentons dans la suite.

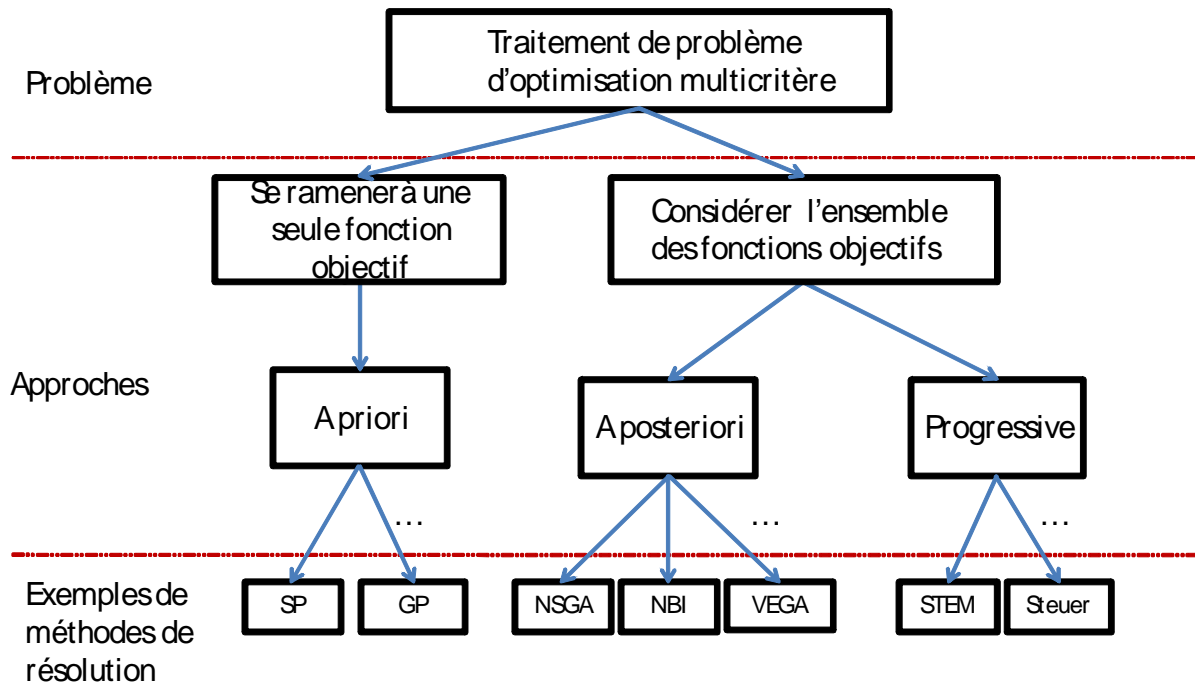


Figure 31 : Approches en optimisation multiobjectif
 SP : Somme Pondérée, GP : Goal Programming, NBI : Normal Boundary Interaction ; VEGA : Vector Evaluator Genetic Algorithm.

1. Approche mono-objectif

Elle caractérise les méthodes dites ‘a priori’ utilisées pour leur simplicité de mise en œuvre et leur grande généralité. En effet, les objectifs du problème d’optimisation sont transformés en une seule fonction objectif. Dans ce cas, le décideur est supposé quantifier a priori l’importance de chaque critère afin de construire une fonction unique. Le processus d’optimisation mono-objectif est ensuite lancé afin de déterminer la solution « optimale ».

Nous donnons ci-dessous des méthodes a priori les plus connues :

La méthode somme pondérée (SP)

Répondue et assez simple à mettre en œuvre, le principe de l’approche somme pondérée est d’affecter aux fonctions objectifs des poids relatifs à l’importance de chacun des objectifs. Une difficulté de cette méthode réside dans le choix des poids en fonction des préférences des experts. Plusieurs travaux issus de la recherche opérationnelle, et notamment la méthode MachBeth [Clivillé’06], proposent des méthodes et des outils permettant de déterminer automatiquement ces poids en fonction du classement, par ordre de préférence des experts, de certaines solutions ou bien d’objectifs.

L'interprétation des poids est en général délicate car elle intègre à la fois la notion d'importance relative des poids et le facteur de normalisation des échelles des critères. La logique d'agrégation est compensatoire ce qui peut masquer une forte insuffisance sur un critère donné. De plus, de faibles variations des valeurs de poids peuvent conduire à des solutions radicalement différentes.

La méthode du but à atteindre (GP : Goal Programming)

Une solution dite utopique S^* étant fixée au départ par le décideur, le problème d'optimisation détermine la meilleure solution : celle qui se rapproche le plus de S^* .

Afin de comparer différentes solutions, plusieurs variantes de cette approche existent. L'une, appelée archimédienne, utilise la somme pondérée afin d'avoir une seule fonction objectif. Une deuxième variante appelée lexicographique, classe les objectifs du problème suivant un ordre de priorités.

La méthode ε -contrainte

Dans cette méthode, on n'optimise qu'un seul objectif jugé le plus important par le décideur, les autres sont transformés en contraintes d'inégalité par rapport à un vecteur seuil ε .

La méthode lexicographique

Dans cette méthode le décideur classe les objectifs par ordre de préférence. Le processus d'optimisation commence par rechercher la meilleure solution en fixant l'objectif prioritaire jusqu'à atteindre un optimum. En cas d'égalité de plusieurs solutions, la comparaison se fait par rapport à l'objectif de priorité inférieure. N'étant pas très efficace quand elle est utilisée seule, cette méthode présente un intérêt lorsqu'elle est utilisée par une variante d'autres approches, comme par exemple la méthode du but à atteindre ou bien les algorithmes évolutionnaires.

D'autres méthodes intégrant notamment, la logique floue, la théorie de l'utilité, la combinaison non linéaire, les fonctions d'acceptabilité... permettent de pondérer a priori des préférences des experts. Le lecteur pourra se référer à l'état de l'art d'Andersson [Andersson'00] pour plus de détails.

2. Approches multiobjectifs

Dans les approches 'a posteriori', les décisions sont prises d'après un ensemble de solutions calculées par un solveur. Dans les approches 'progressives', les processus de décision et d'optimisation sont alternés.

A. Approche a posteriori

Dans ces approches, l'expression des préférences et la pondération des objectifs se fait a posteriori. C'est-à-dire que le processus d'optimisation détermine un ensemble de solutions candidates et la sélection de la meilleure solution ne se fait qu'à la fin du processus. Des approches a priori, notamment la somme pondérée et ε -contrainte, peuvent être appliquées à la fin d'un processus d'optimisation.

Parmi les approches a posteriori, Normal Boundary Interaction (NBI) détermine les solutions optimales par rapport à chaque objectif ; l'ensemble des solutions optimales est défini par l'espace délimité par les segments de droites reliant les solutions optimales.

Les approches a posteriori les plus utilisées sont celles basées sur les algorithmes génétiques, appelés aussi algorithmes évolutionnaires. Différentes formulations de ces approches, comme VEGA (Vector Evaluator Genetic Algorithm), HLGA (Hajela and Lin's Genetic Algorithm), NSGA (Non-dominated Sorting Genetic Algorithm) peuvent être trouvées dans la littérature [Jones'02], [Hajela'90], [Fonseca'93]. Inspirées du principe de la génétique, ces méthodes se basent sur une amélioration des performances des populations grâce à différents types de brassage. La méthode Non-Dominated Sorting Genetic Algorithm II (NSGA II) est une variante des algorithmes génétiques dans laquelle l'auteur définit la relation de dominance suivante [Irisarry'09]. Une solution i domine une solution j si l'une des trois relations suivantes est vérifiée :

- La solution i est faisable alors que la solution j ne l'est pas,
- Les solutions i et j ne sont pas faisables mais la solution i viole moins de contraintes que la solution j ,
- Les deux solutions i et j sont faisables et la solution i domine (au sens de Pareto) la solution j .

B. Approche progressive

Le principe des méthodes de l'approche progressive est de guider l'exploration tout au long du processus d'optimisation. Le but est d'alterner entre le processus de recherche et le processus de décision défini par un classement de solutions, ou bien un choix des poids de pondération des objectifs. A chaque étape, l'ensemble des solutions est analysé afin d'orienter les futures itérations vers les zones les plus intéressantes de l'espace de recherche. C'est pour cela qu'elles sont également appelées méthodes interactives. Les méthodes les plus connues

de cette classe sont la méthode STEM, aussi appelée STEP et celle de Steuer [Coelho'09a], [Jones'02].

La méthode STEM peut être résumée comme étant une suite successive de formulations Min-Max qui minimise la distance entre une solution candidate et une solution théorique S^* . En effet, la prise de décision progressive permet de réduire successivement l'espace des solutions en recentrant les intervalles d'acceptation ainsi que les poids des objectifs. Après une première itération, le décideur peut être satisfait par la solution trouvée. Si ce n'est pas le cas, il peut relaxer certaines contraintes du problème, modifier les poids ou les objectifs eux-mêmes, ou bien rajouter des contraintes supplémentaires au problème en fonction de son analyse de la (ou des) solution(s) retenue(s).

Dans la méthode de Steuer, à chaque itération, un ensemble de solutions est proposé à l'expert qui doit en choisir une ou bien les classer par ordre de préférence. Les poids sont alors automatiquement recalculés afin de minimiser la distance entre l'objectif global et la (ou les) solutions choisies. Cette méthode représente une alternative assez intéressante mais coûteuse en temps de calcul et nécessite des outils très spécifiques et une expertise élevée du problème traité. En effet, n'ayant pas la possibilité de revenir sur les décisions précédentes, un changement de préférences ou bien de décideur impliquerait une relance depuis le début de tout le processus d'optimisation.

Nous classons aussi parmi les approches progressives les méthodes suivantes :

- La théorie des jeux qui est basée sur la recherche d'équilibre entre les objectifs du problème. Les équilibres les plus connus sont celui de Nash et celui de Stackelberg [Coelho'09a].
- La formulation Min-Max qui, au cours de la résolution, minimise le maximum de l'écart relatif entre un critère i et une solution utopique S_i^* , qui peut être irréalisable.

3. Choix d'une approche

Compte tenu de notre hypothèse que les critères métiers s'avèrent souvent difficiles à ordonner entre eux, nous considérons les approches a posteriori comme les plus adaptées aux problèmes d'optimisation multiobjectif en conception organique de systèmes multidisciplinaires.

Dans la section II, nous passons en revue une liste non exhaustive des méthodes d'optimisation les plus répandues pouvant être réparties en deux grandes classes. La première

classe regroupe les méthodes dites complètes ou exactes qui traitent les problèmes ayant un nombre relativement petit de configurations de variables. Ces méthodes garantissent l'obtention de la solution optimale. La deuxième classe regroupe les méthodes incomplètes, appelées aussi méthodes approchées, qui sont mieux adaptées pour la résolution de problèmes plus difficiles. En effet, le principe de ces méthodes est d'effectuer un parcours non systématique de l'espace de recherche et de renvoyer la meilleure solution rencontrée durant la résolution, sans pouvoir garantir son optimalité. Nous portons un intérêt particulier pour les méthodes incomplètes reconnues pour leur exécution rapide : facteur très important en conception.

II. Méthodes d'optimisation

Plusieurs méthodes sont disponibles pour résoudre les problèmes d'optimisation. Une méthode peut s'avérer meilleure qu'une autre pour résoudre un problème donné et moins bonne sur un problème différent. D'après Berthiau, dans [Berthiau'94], la difficulté d'un problème d'optimisation est, dans la plupart des cas, estimée par :

- Le nombre de variables d'optimisation : Pour certains problèmes possédant un nombre élevé de variables, il n'existe pas d'algorithme exact pour le résoudre assez rapidement en évitant l'explosion combinatoire. Ces problèmes sont dits NP-difficiles [Garey'79].
- La continuité des variables d'optimisation : Pour ce type de problèmes, on ne connaît pas, en général, d'algorithme déterminant à coup sûr la solution optimale en un nombre fini d'itérations.

La plupart des méthodes de résolution possède en général une stratégie d'exploration de l'espace de recherche et une stratégie d'exploitation permettant la convergence rapide de l'algorithme de recherche. Les diverses méthodes peuvent être réparties en deux grandes catégories : les méthodes complètes et les méthodes incomplètes. Nous accordons un intérêt particulier à la deuxième catégorie de méthodes du fait de leur rapidité d'exécution.

Le lecteur pourra trouver d'autres classifications de ces méthodes dans [Berthiau'94] où les méthodes sont distinguées par rapport à leur utilisation ou non d'éléments stochastiques, ou bien leur précision, ou encore, leur philosophie sous-jacente, ...

Le terme **configuration** est utilisé par la communauté Recherche Opérationnelle pour décrire, à une itération donnée, les valeurs utilisées afin d'instancier des variables d'optimisation, ceci pour éviter de prêter à confusion avec le terme « solution » utilisé dans le deuxième chapitre.

A notre sens, une solution est une configuration où toutes les variables sont instanciées. En effet, certains algorithmes de recherche décomposent le problème en sous-problèmes. Si un problème est défini par des variables à déterminer V , des contraintes à satisfaire C et des fonctions objectifs à optimiser F , un sous problème est défini par un ensemble de variables inclus dans V , un ensemble de contraintes inclus dans C et un ensemble de fonctions objectifs inclus dans F . Dans ce cas, la résolution du problème revient à résoudre l'ensemble des sous-problèmes qui ne sont pas forcément disjoints.

Un autre cas, où la notion de configuration peut être utilisée, c'est quand l'algorithme divise l'espace de recherche en sous-espaces. Dans ce cas, certaines variables d'optimisation ne sont pas instanciées ou bien sont instanciées à des valeurs par défaut.

Nous définirons donc le terme « configuration » par une instanciation des variables d'optimisation à un instant donné du processus d'optimisation. Le terme « solution » sera utilisé lorsque nous serons certains que toutes les variables du problème sont instanciées.

1. Méthodes complètes

Le principe essentiel des méthodes complètes, aussi appelées méthodes exactes, consiste à énumérer, souvent implicitement, l'ensemble des configurations de l'espace de recherche et à retourner ainsi la ou les solutions optimales, et ce sans aucune stratégie d'exploitation.

Ces méthodes garantissant l'obtention de la solution optimale, se limitent à la résolution des problèmes de taille réduite : définis par un petit nombre de variables d'optimisation ou par des domaines réduits pour ces variables.

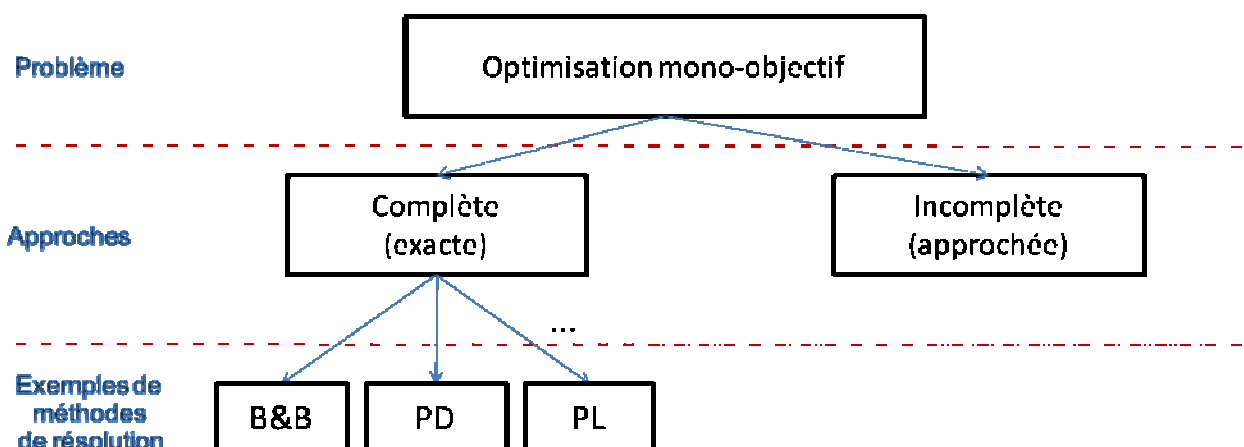


Figure 32 : Approches d'optimisation mono-objectif ; exemples de méthodes complètes
 B&B : Branch and Bound ; PD : Programmation Dynamique ; PL : Programmation Linéaire.

Selon Laporte [Laporte'92], les méthodes complètes les plus connues sont :

A. La recherche arborescente orientée (ou Direct Tree Search)

Fortement inspirée du principe de « diviser pour mieux régner » ou bien « diviser et régner », consiste en une construction séquentielle des solutions au moyen d'un arbre de recherche par Séparation et Evaluation (Branch and Bound) [Larsen'00]. Cet algorithme permet d'effectuer une recherche dynamique arborescente où le problème initial est décomposé en sous-problèmes (appelés aussi sous-espaces) en intégrant à chaque fois de nouvelles contraintes.

B. La Programmation Dynamique (ou Dynamic Programming)

C'est une technique algorithmique qui permet de résoudre une catégorie particulière des problèmes d'optimisation sous contrainte. Inventée, à la fin des années 1940 par le professeur Richard Bellman, elle s'applique à des problèmes d'optimisation dont la fonction objectif se décrit comme la somme de fonctions monotones non-décroissantes.

C'est une amélioration ou une adaptation de la méthode diviser et régner dont le principe de base est que la résolution d'un problème dépend de la résolution de ses sous-problèmes.

Contrairement à la méthode diviser et régner, dans la programmation dynamique les sous-solutions peuvent interagir entre elles et se superposer. C'est-à-dire qu'un sous-problème peut être inclus dans deux sous-problèmes différents.

Une seconde différence entre ces deux méthodes est que la méthode diviser et régner est récursive dans sa construction de l'arbre de recherche : les calculs se font de haut en bas. Tandis que la programmation dynamique est une méthode dont les calculs se font de bas en haut. On commence par résoudre les plus petits sous-problèmes (avec un petit nombre de variables d'optimisation), et en combinant leurs solutions, on obtient les solutions des sous-problèmes les englobant.

C. La programmation linéaire (PL) (ou Linear Programming)

C'est une méthode d'optimisation des problèmes les plus faciles : où toutes les contraintes sont linéaires. Elle s'applique à plusieurs problèmes réels de recherche opérationnelle et un grand nombre d'algorithmes s'en inspirent pour la résolution d'autres problèmes d'optimisation.

La PL traite des problèmes à variables réelles, et dans le cas où certaines variables du problème sont discrètes, on parle alors de programmation linéaire en nombres entiers (PLNE).

Un problème de PLNE n'est pas un programme linéaire dans le sens où son domaine de réalisabilité n'est pas un polyèdre mais un ensemble discret de points. Il peut être formulé comme un problème de PL auquel on ajoute des contraintes sur les domaines d'appartenance des variables discrètes. Le problème est alors dit du type linéaire mixte (ou à variables mixtes).

Plusieurs techniques de PLNE ont été proposées dans la littérature afin de traiter les divers problèmes d'optimisation. Les plus répandues d'entre elles sont : la technique de partitionnement d'ensembles (Set partitioning method), celle de génération de colonnes (Column generation method) et la méthode du simplexe [Battiti'94].

L'avantage de la PLNE par rapport à la PL est qu'elle englobe plus de problèmes et qu'il existe déjà plusieurs techniques et solveurs, ayant fait leurs preuves sur des cas pratiques.

2. Méthodes incomplètes

Les méthodes incomplètes, appelées aussi méthodes approchées, effectuent un parcours non systématique de l'espace de recherche et renvoient la ou les meilleures solutions rencontrées durant la résolution. Essentiellement axées vers l'exploitation de l'espace déjà visité, elles visent à faire converger, le plus rapidement possible, l'algorithme vers une bonne solution, sans pour autant garantir l'optimalité. Ces méthodes sont dites heuristiques. Une heuristique est un algorithme intuitif qui permet de déterminer une bonne solution assez rapidement à un problème difficile sans pouvoir garantir son optimalité. Quand cet algorithme est applicable sur une famille de problèmes différents mais pouvant être formalisés de manière semblable, il est appelé méta-heuristique.

D'autres méthodes, utilisent leurs capacités d'exploration sans avoir de stratégie d'exploitation, comme par exemple Monté-Carlo, [Fishman'95]. En effet, utilisant des techniques probabilistes, cette méthode recherche l'optimum en utilisant une heuristique d'essai-erreur, tout en générant un ensemble de solutions successives suivant une loi statistique.

Dans cette section, nous nous intéressons aux méta-heuristiques. Plus générales que les heuristiques spécifiques, les méthodes les plus répandues de cette classe sont : Le recuit simulé, les algorithmes évolutionnaires et la méthode tabou. Avant de décrire ces trois méthodes, il nous paraît nécessaire de définir certaines notions.

Définitions :

Soit un espace de recherche défini sur un problème d'optimisation,

- On appelle **voisinage** toute application associant à toute solution S un ensemble de solutions voisines $N(S)$. Le voisinage de S représente donc l'ensemble des éléments $N(S)$ potentiellement accessibles lors du processus d'optimisation.
- Le parcours de ce voisinage est décrit par une **heuristique de mouvement** qui décide, à partir de S , de la prochaine solution S' de $N(S)$ à explorer. Cette transition, appelée **mouvement**, est définie par l'opération qui consiste à modifier une solution S en une autre solution S' de $N(S)$.
- **Instancier** (resp. **désinstancier**) une variable revient à lui affecter une valeur (respectivement lui remettre sa valeur par défaut). La **réinstanciation** est un processus de désinstanciation suivi d'une instanciation.
- Une solution S est un **optimum local** par rapport à un voisinage $N(S)$ si la condition $F(S) < F(S')$ (où F est la fonction objectif à minimiser) est vérifiée pour toute solution voisine S' de $N(S)$.

Nous allons, par la suite, nous intéresser aux méthodes de recherche se distinguant par leurs stratégies d'exploration respectives pouvant être regroupées en deux catégories. Les premières sont basées sur l'exploration de l'espace avec plusieurs configurations simultanément. Les méthodes les plus connues de cette catégorie sont les algorithmes évolutionnaires. Tout au long de la recherche, plusieurs mécanismes, tels que la mutation, le croisement... sont appliquées à l'ensemble de configurations, appelé population, afin de converger vers de meilleures solutions.

Dans la deuxième catégorie, les méthodes se basent sur l'exploration du voisinage de la solution courante afin de guider la recherche progressivement vers l'optimum. C'est pour cela qu'elles sont souvent appelées méthodes de voisinage. Les méthodes les plus répandues de cette catégorie sont le Recuit simulé, la méthode de descente (Hill Climbing), les colonies de fourmis, la recherche tabou...

La Figure 33 met en évidence les différentes approches permettant de résoudre un problème d'optimisation mono-objectif. La liste des méthodes évoquées n'est nullement exhaustive, mais elle regroupe les méthodes d'optimisation les plus répandues.

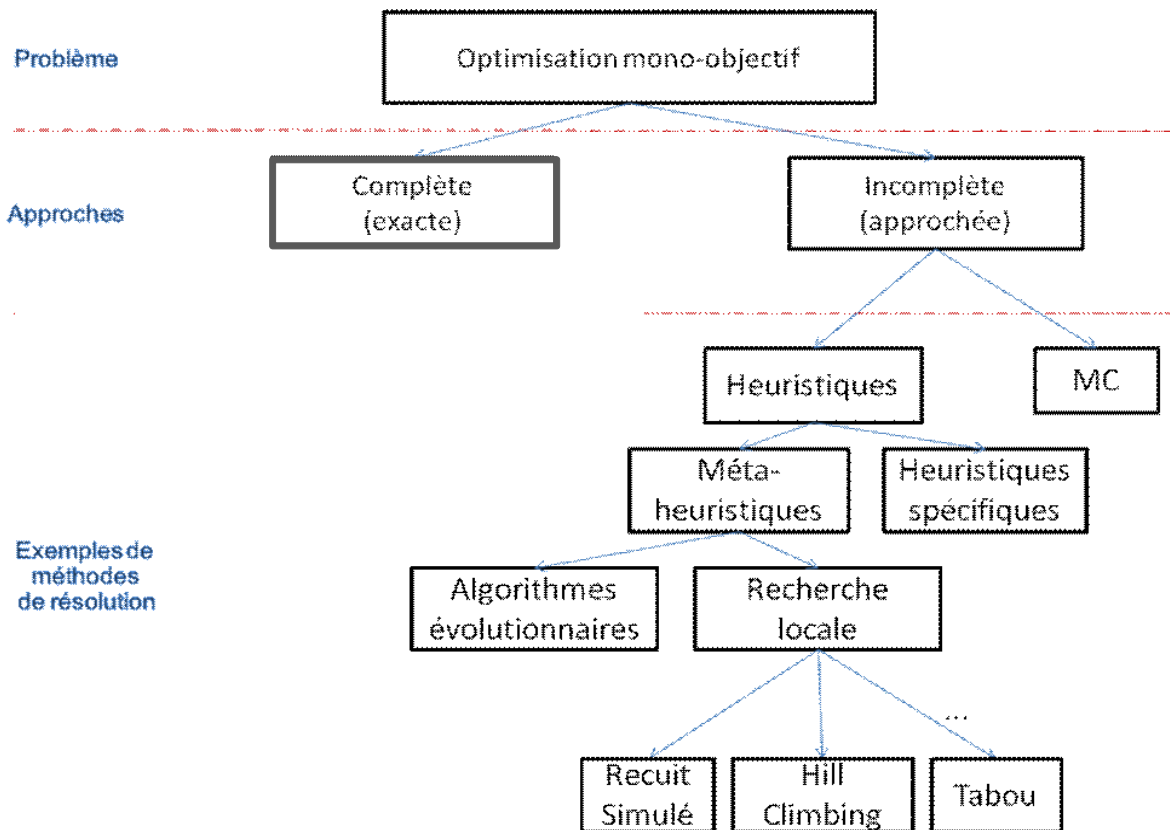


Figure 33 : Approches et exemples de méthodes de résolution d'un problème d'optimisation ; MC : Monté Carlo.

A. Algorithmes évolutionnaires

Aussi connus sous le nom d'algorithmes évolutifs, les algorithmes évolutionnaires regroupent un ensemble de méthodes ayant les mêmes caractéristiques telles que la programmation génétique, les stratégies d'évolution et les algorithmes génétiques : classe la plus répandue dans la littérature [Jones'02]. Fortement inspiré de la théorie de Darwin sur l'évolution, le principe des algorithmes évolutionnaires est de retenir à chaque itération, un ensemble de solutions candidates, de choisir les meilleures d'entre elles et de les manipuler par des opérateurs afin d'en dériver un meilleur ensemble de solutions.

Plus précisément, à partir d'un ensemble de solutions, appelé population, une opération de sélection vise à retenir les individus, ou bien solutions, les plus performantes. A cet ensemble d'individus retenu, sont appliqués des opérateurs génétiques, tels que la mutation ou le croisement. Ces manipulations visent à améliorer les performances des individus. Les processus de sélection et de manipulation constituent une génération d'un point de vue génétique et une itération d'un point de vue algorithmique. Ces itérations sont répétées

jusqu'à atteindre un certain critère d'arrêt pouvant être, par exemple, le nombre de générations.

B. Recherche locale

Une méthode de Recherche Locale (RL) est un processus itératif basé sur l'amélioration d'une ou plusieurs solutions initiales [Battiti'94], par le remplacement de la solution courante par une solution voisine, obtenue par une suite de variations locales dites mouvements. Il en ressort que l'utilisation d'un algorithme de recherche locale nécessite la définition d'une fonction de voisinage et d'une stratégie de son exploration.

En recherche locale classique, un mouvement à partir d'une configuration, consiste à affecter une valeur différente à une variable préalablement instanciée à une valeur différente. Le choix de la variable à réinstancier est fait par une heuristique de mouvement : fonction permettant de choisir la meilleure solution voisine à la solution courante.

Cependant, une heuristique de mouvement peut choisir la variable à réinstancier tout en désinstanciant un certain nombre d'autres variables et en ne violant pas les contraintes du problème. Dans ce cas, la configuration choisie est celle pour laquelle le nombre de désinstanciations nécessaires est minimal, afin de mieux s'approcher d'une solution complète (où toutes les variables sont instanciées). Cet exemple d'heuristique est utilisé dans les méthodes de construction progressive d'une solution tout en gardant la solution consistante à chaque itération ; c'est-à-dire, ne violant aucune contrainte.

Contrairement au mouvement classique, où la réparation s'effectue uniquement sur une seule variable, ce mouvement (instanciation + désinstanciation(s)) qu'on vient de définir est multi-attributs. Il porte, par sa sémantique, sur plusieurs variables et permet ainsi une exploration plus large de l'espace de recherche. C'est pour cette raison que cette recherche est moins locale qu'une recherche locale classique.

Pour surmonter l'obstacle des optima locaux, une idée s'est montrée très fructueuse, au point qu'elle est à la base de toutes les méthodes de voisinage (méthode tabou, recuit simulé...) : Il s'agit d'autoriser, de temps en temps, des mouvements de remontée, autrement dit, d'accepter une dégradation temporaire de la valeur de la fonction objectif. Un mécanisme de contrôle des dégradations, spécifique à chaque méta-heuristique, permet d'éviter la divergence du procédé. Il devient dès lors possible de s'extraire du piège que représente l'optimum local, pour partir explorer une autre région qui pourrait être prometteuse.

- ***Méthode de la descente (Hill-Climbing)***

Inspirée de l'escalade en montagne, le principe du Hill-Climbing stipule que pour atteindre le sommet le plus proche il suffit de prendre le chemin qui monte le plus à partir de la position initiale. Ce choix est réitéré jusqu'à ce qu'on atteigne un « pic » local ou global. Donc, d'une manière générale, l'algorithme de la descente choisit à chaque itération la configuration qui minimise le coût ou bien celle qui maximise le profit dans le voisinage direct de la configuration courante. Cette opération est répétée jusqu'à ce que l'on trouve une configuration qui soit meilleure que tout son voisinage.

- ***Recuit simulé***

Le recuit simulé (Simulating Annealing) a été présenté initialement par Kirkpatrick dans [Kirkpatrick'83]. Il s'agit d'une méthode de résolution issue de la physique statistique et qui s'inspire fortement des techniques du recuit thermique. Plutôt que de chercher la ou les solution(s) optimale(s), le recuit simulé se contente de chercher une bonne solution réalisable. Dès lors, le problème est de trouver un meilleur compromis entre le temps de calcul et la qualité de la solution approchée.

Partant de ce principe, l'innovation apportée par Kirkpatrick et qui la distingue de la méthode de la descente est le fait d'accepter des détériorations ponctuelles sur la solution courante dans le but d'éviter le piégeage par les optimums locaux et de converger vers l'optimum global. En effet, à chaque itération, toutes les configurations voisines de la courante sont évaluées. Celle qui apporte le plus d'amélioration est choisie. Dans le cas d'un optimum local (la configuration actuelle est meilleure que tout son voisinage), la configuration apportant la dégradation la moins importante est choisie.

- ***Recherche Tabou***

Comme les méthodes de recherche locale, l'algorithme de recherche tabou [Glover'89], [Glover'95] utilise un schéma itératif visant à améliorer à chaque itération la solution courante. Inspiré du fonctionnement de la mémoire humaine, le principe de la méthode tabou est le suivant : on part d'une solution initiale réalisable et on visite à chaque itération la meilleure solution située dans le voisinage de la solution courante jusqu'à ce qu'un critère d'arrêt soit rencontré. Pour éviter le bouclage sur les solutions déjà explorées, les solutions récemment examinées sont temporairement déclarées taboues ou interdites pour un certain nombre d'itérations. Ces solutions sont alors placées dans une structure de données adéquate : la liste *tabou*, qui peut être, selon R. Battiti et G. Tecchiolli [Battiti'94], stricte ou non stricte.

Le statut tabou d'une solution peut être annulé si certaines conditions sont rencontrées. Un exemple de cette situation est quand une solution taboue est meilleure que les solutions rencontrées.

D'autres principes de cette méthode de recherche lui permettent de favoriser une bonne exploration de l'espace de recherche comme celui de la Diversification et de l'Intensification.

- ***Gestion de la liste tabou***

Etant donné qu'un mouvement est multi-attributs, les risques d'occurrence de cycles dans l'espace de recherche sont plutôt importants. Afin d'éviter la production de tels cas, une liste tabou va contenir, tout au long de la recherche, des mouvements interdits qui seront classés tabous durant une certaine durée DT.

- ***Diversification***

Le but de la diversification est de permettre à la recherche locale d'échapper des optima locaux de l'espace de recherche. Plusieurs mécanismes permettent d'effectuer cette fonction sans qu'ils puissent toutefois garantir son accomplissement. L'un des mécanismes qui peut servir à la diversification est la détection de nogoods [Habet'04]. Un nogood est, par définition, une configuration qui ne peut être étendue à une solution. Donc, il serait plus approprié de considérer les éléments d'une telle configuration comme non opportuns à la recherche.

- ***Intensification***

L'intensification a pour but de focaliser la recherche autour des zones de l'espace des configurations jugées pertinentes. Plusieurs schémas d'intensification sont possibles. Toutefois, cette tâche est accomplie par une relance de l'exploration tabou à partir des éléments de la ou des meilleures configurations trouvée S^* . Ainsi, si au bout d'un certain nombre d'itérations ou en cas de saturation tabou (i.e. toutes les configurations voisines sont taboues), leur caractère tabou est levé et la recherche est réinitialisée à partir de S^* . Cette relance est effectuée sans une redéfinition de la liste *tabou* qui reste non stricte. En effet, l'utilisation d'une liste stricte, dans le cas d'un mouvement multi-attributs, conduirait rapidement la recherche, une nouvelle fois, à une saturation *tabou*.

Mis à part la possibilité de repartir des bonnes solutions déjà rencontrées ; le mécanisme d'intensification peut s'effectuer en reconstruisant une solution de départ qui tente de combiner des attributs qui ont été présents souvent dans les meilleures configurations ; ou

bien de fixer à certaines variables les valeurs qui ont été souvent trouvées dans les meilleures solutions sauvegardées.

3. Choix de méthodes

Dans un contexte d'optimisation multiobjectif, plusieurs méthodes ont été adaptées à cet effet. Comme nous l'avons explicité précédemment (cf. section I1.), différentes techniques permettent de ramener l'ensemble des objectifs à optimiser en un seul. Néanmoins, plusieurs auteurs ont introduit des variantes de ces méthodes pouvant être appliquées dans un contexte multiobjectif et sans recours à une pondération des multiples objectifs (cf section I 2.). La plupart de ces méthodes sont basées sur la notion de dominance, souvent celle de Pareto.

En général, les méthodes de recherche locale, nécessitant une heuristique de mouvement pour le choix de la meilleure solution voisine, sont difficilement applicables à une optimisation multiobjectif sans pondération des objectifs. Néanmoins, certains auteurs [Hajela'90], [Coello'96] ont proposé des techniques permettant de choisir la solution à retenir en fonction des propriétés mathématiques de celle-ci. En effet, en fonction de la stratégie de résolution, l'algorithme recherche choisit les zones de l'espace de recherche suivant la densité et le voisinage des solutions faisables. Des méthodes hybrides de méta-heuristiques multiobjectifs ont aussi été proposées par exemple dans [Barichard'03], [Gandibleux'02], avec l'étude de l'impact des solutions prometteuses et la question des solutions 'potentiellement' Pareto.

Pour garantir de fournir une solution en un temps fini nous privilégions les méthodes de résolution incomplètes. L'approche Tabou sera plus particulièrement retenue du fait qu'elle permet, dans ses versions simples, une interprétation du cheminement de la recherche dans l'espace des variables de conception contrairement aux algorithmes génétiques dans lesquels la progression de la recherche par croisement/mutation est moins intuitive.

Conclusion

Dans ce chapitre nous avons examiné la problématique de multiplicité des objectifs, souvent rencontrée en optimisation multidisciplinaire. Dans une première approche, le problème d'optimisation multiobjectif est ramené à un problème d'optimisation mono-objectif par l'agrégation des fonctions objectifs dans le but d'appliquer des méthodes de résolution connues pour leur efficacité. Cette agrégation peut s'effectuer avant, après ou en cours du processus d'optimisation. Ces trois possibilités sont dites a priori, a posteriori ou progressive.

Le problème devenu mono-objectif, plusieurs méthodes d'optimisation peuvent être appliquées. Une première famille de méthodes, dite complètes ou exhaustives, énumère, souvent implicitement, toutes les configurations possibles afin d'en choisir la meilleure. Ces méthodes garantissent l'optimalité du résultat mais présentent l'inconvénient d'être coûteuses en temps, vu la difficulté des problèmes traités. Une deuxième famille de méthodes regroupe essentiellement les méta-heuristiques. Souvent inspirées de mécanismes naturels (fourmis, abeilles, génétique humaine, cerveau...), ces méthodes permettent de déterminer une bonne solution au problème en un temps relativement court et acceptable, sans garantie de l'optimalité. Le problème principal de méta-heuristiques est qu'elles ne sont pas généralisables. En effet, leur efficacité ne peut être prouvée qu'expérimentalement, et sur une famille de problèmes ayant les mêmes caractéristiques.

Une deuxième approche de résolution du problème de multiplicité des objectifs considère l'ensemble des critères. Celles basées sur la notion de dominance de Pareto (qui ne compare ni oppose ces objectifs) nous paraissent plus intéressantes vu le cadre de notre travail de recherche. En effet, les critères métiers en conception pluridisciplinaire sont spécifiques aux disciplines et peuvent être contradictoires. C'est pour cette raison qu'il peut être plus judicieux d'exploiter des préférences entre critères seulement a posteriori. Les considérer trop tôt peut conduire à restreindre l'espace de recherche en privilégiant un métier plutôt qu'un autre.

Pour fournir des solutions efficaces en un temps fini nous privilégions les méthodes de résolution incomplètes. L'approche Tabou sera plus particulièrement retenue du fait qu'elle permet, dans ses versions simples, une interprétation du cheminement de la recherche dans l'espace des variables de conception contrairement aux algorithmes génétiques pour lesquels la progression de la recherche par croisement/mutation est moins intuitive.

Dans le chapitre 4 nous présentons notre contribution consistant en des méthodes de construction de front de Pareto et en une proposition d'outils d'analyse de ces fronts. Le principe de recherche de solutions efficaces est de fouiller l'espace des variables de conception en permettant une bonne exploration de cet espace et en examinant plus systématiquement les zones prometteuses pour identifier de très bonnes solutions en un temps acceptable par les concepteurs.

Chapitre 4 : Contribution à la résolution d'un problème d'optimisation multiobjectif en conception multidisciplinaire.

Introduction

Notre apport scientifique se situe au niveau du support d'ingénierie en optimisation pendant le processus de conception organique de produits multidisciplinaires. Le contexte est celui d'un processus de conception dont la complexité résulte des caractères (Voir Chapitre 1) :

- Multi-objectifs
- Multi-compétences
- Multi-relationnel avec couplage
- Mixité des variables
- Combinatoire

La problématique est celle du choix de variables de conception à déterminer par des concepteurs de cultures scientifiques différentes et pouvant de ce fait rencontrer des difficultés à dégager un compromis entre des valeurs possibles de variables disciplinaires. L'objectif est d'une part de proposer des méthodes d'optimisation adaptées au contexte de conception multidisciplinaire capables de fournir aux concepteurs, en un temps acceptable, un ensemble de solutions efficaces et d'autre part de choisir un outil de discrimination des solutions en facilitant l'analyse.

Pour couvrir un plus grand nombre de situations deux méthodes d'optimisation sont proposées [Mouelhi'09b]. La première, illustrée dans la section 1 de ce chapitre, est applicable dans le cas où les concepteurs peuvent exprimer a priori des préférences afin de classer les objectifs de conception et d'agréger leurs fonctions objectifs. Il s'agit d'une adaptation d'une méthode de recherche locale : tabou (cf. Chapitre 3) permettant de prendre en compte des variables mixtes et de dégager la meilleure solution trouvée ainsi que l'ensemble des solutions efficaces (non dominées au sens de Pareto) lors de l'exploration de l'espace des solutions.

La deuxième méthode applicable dans le cas que nous privilégions où les préférences sont exploitées a posteriori est une hybridation entre recherches exacte et approchée. Elle permet de déterminer le front de Pareto sans avoir à agréger les objectifs du problème. En effet, notre but premier est de proposer rapidement aux concepteurs un ensemble de solutions afin qu'ils choisissent a posteriori parmi des solutions efficaces.

Du fait de l'absence de connaissance globale sur la répartition en famille des solutions possibles, les outils de discrimination envisagés reposent sur des cartes auto-organisatrices. Les cartes auto-organisatrices de Kohonen [Kohonen'01] ont l'intérêt de permettre une projection des solutions sur des cartes visualisables en 1D, 2D ou 3D. La contrainte d'imposition au départ de la topologie de l'espace de lecture des solutions peut entraîner un manque de précision dans la restitution aux concepteurs du résultat de la discrimination. Aussi avons-nous envisagé des méthodes plus souples telles que l'IGNG d'Ennaji et Prudent dans [Ennaji'04] et [Ennaji'05].

I. Optimisation avec pondération des objectifs

Dans cette partie, nous nous intéressons au cas où un ordre de préférence pour les objectifs du problème peut être fixé a priori par les concepteurs. Dans ce cas l'optimisation est décrite sous forme d'un problème de minimisation (ou maximisation) d'une fonction objectif.

Comme rappelé au Chapitre 3, plusieurs choix d'optimisation sont possibles. Notre but étant de déterminer la meilleure solution possible en un temps de calcul fixé a priori, les méta-heuristiques sont privilégiées. En effet, ces méthodes permettent d'explorer efficacement l'espace des solutions et de guider la recherche afin de converger vers les solutions optimales. Dans le contexte, de notre travail, à savoir, l'optimisation multiobjectif, la fonction objectif peut présenter plusieurs irrégularités et discontinuités, ce qui peut engendrer un nombre important d'optimums locaux. Notre choix s'est donc porté sur la méthode de recherche tabou, réputée pour sa capacité à éviter ces optimums locaux.

Dans le but de prendre en compte des variables continues, nous avons adapté la méthode de recherche tabou à notre contexte de travail. En général, la recherche tabou, comme pour la majorité des méta-heuristiques classiques [Siarry'98], nécessite un ensemble de variables discrètes et détermine la meilleure solution rencontrée lors de l'exploration de l'espace des solutions. Nous avons donc dû apporter une modification à l'algorithme de base de [Glover'95] pour prendre en compte les variables continues. L'apport proposé consiste en deux adaptations de la recherche « tabou » classique.

Pour l'optimisation d'une fonction continue, une approche courante est de discrétiser le domaine de chaque variable. La difficulté majeure réside dans la détermination optimale du pas de discrétisation [Berthiau'94].

Le choix de ces pas de discrétisation est un compromis entre deux situations extrêmes :

- si le pas est trop petit, on n'explore qu'une région limitée de l'espace des variables, et il y a, en général, un risque de piégeage par un optimum local. Le choix de la méthode tabou permet d'éviter cela ;
- si le pas est trop grand, il devient impossible de suivre les variations fines de la fonction objectif.

Plusieurs auteurs, et notamment Siarry, s'intéressent à la problématique de la discrétisation des variables. Un de ses exemples d'adaptation de la recherche tabou à des variables continues [Chelouah'00] s'intitule « Enhanced Continuous Tabu Search ». Suivant cette approche chaque solution est considérée comme « une boule » centrée sur une solution... D'après ce même auteur, la meilleure solution peut consister à élaborer une topologie adaptative.

Partant de ce constat, et afin de prendre en compte les variables continues, nous avons utilisé un mécanisme d'adaptation dynamique des pas de discrétisation. Ce mécanisme appelé « pas adaptatifs » est inspiré de la famille générale des méthodes de région de confiance en optimisation [Coelho'09b].

Pour extraire en cours de recherche des solutions efficaces (non dominées au voisinage du chemin suivi vers l'optimum), un processus de mémorisation de l'ensemble de solutions non dominées rencontrées tout au long de la recherche de l'optimum, est introduit. En effet, indépendamment de la fonction objectif, chaque solution rencontrée est ajoutée au front de Pareto déjà constitué si elle n'est dominée par aucune solution de celui-ci.

1. Description de la méthode de recherche

Les méta-heuristiques, en général, commencent l'exploration de l'espace de recherche par une solution initiale qu'elles tentent d'améliorer à travers les mouvements successifs.

Dans notre approche [Mouelhi'09b], nous avons utilisé le principe de la version initiale de la recherche tabou, à savoir, choisir une solution initiale d'une manière aléatoire. Cette solution ainsi que son évaluation sont sauvegardées comme étant optimales au début de la recherche.

Le cœur de la méthode consiste à guider la recherche vers la meilleure solution atteignable à partir de la solution en cours. A chaque itération, le voisinage de la solution courante est évalué et la meilleure solution choisie. Comme pour les méthodes de recherche locale notre approche permet de retenir la meilleure solution voisine même si celle-ci n'améliore pas la solution courante afin d'éviter le piégeage par les optimums.

Dans le but d'analyser le comportement de l'algorithme et de reconstituer, a posteriori le chemin suivi par la recherche, nous avons gardé en mémoire toute la solution.

Ainsi, toutes les solutions du voisinage sont ajoutées automatiquement à la liste Tabou. Ce qui représente une différence entre notre proposition et l'algorithme tabou classique, où on ne retient dans la liste tabou que le mouvement ayant permis d'atteindre la solution. Dans ce dernier cas, il faut juste retenir la variable ré-instanciée ainsi que le sens du mouvement (+ ou -).

Ce choix est coûteux en temps de calcul et en espace mémoire puisqu'il rajoute une fonctionnalité supplémentaire à la méthode de base, à savoir l'historique. Mais il est très constructif quant à l'analyse des paramètres du tabou et du comportement de l'algorithme de recherche lors de l'exploration de l'espace.

Ensuite, dans l'optique de donner plus de flexibilité et de liberté aux concepteurs dans leurs choix, un front de Pareto est construit pendant la recherche d'un optimum de la fonction objectif. Ceci permet également aux concepteurs de comparer la solution optimale proposée avec l'ensemble de solutions non dominées. Ils ont ainsi des alternatives à la solution optimale sans avoir à relancer l'algorithme d'optimisation.

Pour ce faire, chaque solution visitée est rajoutée au front de Pareto si elle n'est pas dominée. Lors de cette vérification, nous ne prenons plus en compte les valeurs de la fonction objectif (obtenue par pondération des fonctions objectifs) mais plutôt les valeurs des fonctions objectifs que l'on compare deux à deux.

La solution retenue est ensuite comparée à la meilleure solution obtenue jusque là et est mémorisée si son évaluation est meilleure.

Le critère d'arrêt est atteint quand on se trouve soit dans une situation où la solution courante est entourée, au sens du voisinage, par un ensemble de solutions taboues soit quand le temps de calcul est atteint. Ce seuil doit être choisi au préalable.

La Figure 34 donne l'algorithme proposé.

Algorithm 1: Tabu Search

```
begin
   $S \leftarrow$  choose randomly;
   $e \leftarrow$  evaluate ( $S$ );
   $S^* \leftarrow S$ ;
   $e^* \leftarrow e$ ;
  add ( $S$ , Tabu list);
  add ( $S$ , Pareto front);
  repeat
     $N(S) \leftarrow$  neighborhood ( $S$ );
    for each  $S_i \in N(S)$  do
      add ( $S_i$ , Tabu list);
       $e_i \leftarrow$  evaluate ( $S_i$ );
      if not dominated ( $S_i$ ) then
        add ( $S_i$ , Pareto front);
     $\{S, e\} \leftarrow$  best evaluation ( $N(S)$ );
    if  $e \leq e^*$  then
       $S^* \leftarrow S$ ;
       $e^* \leftarrow e$ ;
  until Stop criterion;
end;
```

Figure 34 : Algorithme de recherche tabou avec détermination de solutions locales efficaces

La Figure 34, ne schématise pas tous les mécanismes de notre algorithme. Pour des raisons de clarté, nous avons choisi d'expliquer textuellement ces mécanismes.

2. Mécanismes de l'algorithme de recherche

Pas adaptatifs

La méthode de recherche tabou classique nécessite des variables d'optimisation discrètes. L'algorithme proposé prend en compte le cas des variables mixtes. Nous avons utilisé un mécanisme de pas adaptatif afin de pallier la problématique des variables continues.

Cette méthode se rapproche de la famille générale des méthodes de région de confiance en optimisation [Coelho'09b]. Son principe est de calculer la différence d'évaluations Delta entre deux solutions voisines que l'on compare avec les seuils Sigma_{\min} et Sigma_{\max} fixés au préalable :

- Si Delta est supérieur à $\text{Sigma}_{\text{max}}$, on considère que les solutions sont assez éloignées. Dans ce cas, la solution trouvée est rejetée, les pas des variables continues sont réduits, et on réévalue les nouvelles solutions voisines de l'ancienne solution courante.
- Si Delta est inférieur ou égal à $\text{Sigma}_{\text{max}}$ et Delta est supérieur à $\text{Sigma}_{\text{min}}$, on passe à l'itération suivante.
- Si Delta est inférieur à $\text{Sigma}_{\text{min}}$, on passe à l'étape suivante en augmentant les pas de discrétisation.

Mécanismes d'intensification et de diversification

Comme nous l'avons expliqué dans le Chapitre 3, le principe de la diversification est de relancer la recherche à partir d'une solution initiale différente. Partant de ce principe, une diversification est lancée à chaque fois qu'une solution se retrouve entourée par un voisinage tabou.

A chaque fois qu'une solution est meilleure que toutes les solutions voisines, il s'agit d'un optimum (local ou global). Cette solution est retenue comme étant une piste prometteuse pour de prochaines recherches. En effet, après un certain nombre de diversifications, un processus d'intensification est lancé. Il s'agit de reprendre l'ensemble des optimaux locaux rencontrés au cours de la recherche et de fouiller autour d'eux. Dans notre cas, on fixe les pas des variables continues à leurs valeurs minimales. L'algorithme se remet à explorer successivement les voisinages des optimaux locaux avec plus de précision [Mouelhi'09b].

Paramétrage de l'algorithme

Comme pour toute méta-heuristique, la difficulté de l'algorithme réside dans le choix des paramètres de l'algorithme utilisé. Ces paramètres dépendent fortement du problème d'optimisation et ne peuvent être généralisés. D'ailleurs, c'est pour cette raison qu'on ne peut pas affirmer qu'une méta-heuristique est meilleure qu'une autre d'une façon générale. L'efficacité d'une méta-heuristique est toujours liée à un problème ou à une famille de problèmes similaires. Dans notre cas (l'algorithme tabou), les paramètres prépondérants sont : le temps de recherche alloué, la durée d'interdiction tabou, et le choix des pas d'adaptation...

Pour le premier paramètre, le temps de calcul, d'une manière générale plus l'algorithme explore l'espace de recherche, plus il a de chances de trouver la solution optimale. Cela est correct, sauf que notre objectif premier est de trouver la meilleure solution possible, le plus

rapidement possible. En d'autres termes, il ne faut pas laisser l'algorithme explorer longtemps l'espace et que la solution trouvée n'apporte qu'une amélioration mineure par rapport à la meilleure solution précédente.

Le deuxième paramètre, la durée tabou, consiste à choisir le nombre d'itérations durant lesquels une solution reste tabou (interdite). Ce mécanisme servant à éviter le bouclage de la recherche autour de l'optimum local, doit permettre une relaxation de cette contrainte lorsque c'est nécessaire.

Quant au troisième paramètre, l'adaptation du pas consiste à choisir les valeurs initiales des pas de discrétisation des variables continues, le seuil (Σ) de la différence entre les évaluations des solutions voisines, et le rapport ancien pas/nouveau pas. Les pas minimal et maximal peuvent être fixés au préalable en fonction de la précision désirée sur les variables continues et sur les objectifs. Si l'on cherche à déterminer la longueur d'une poutre de charpente, par exemple, choisir un pas minimum inférieur à un millimètre augmenterait inutilement le temps de recherche. Afin d'analyser les causes d'éventuelle perte de temps en fouillant excessivement une zone de l'espace, l'analyse des résultats nécessite une reconstruction de la trajectoire choisie par la recherche.

Les nombres de diversifications et d'intensifications sont deux paramètres importants et ils sont déterminés expérimentalement. Il s'agit de fixer au début de la recherche, le ratio nombre de diversifications/nombre d'intensifications. En effet, quand ce ratio est élevé - beaucoup de phases de diversification- l'algorithme explore un grand nombre de sous-espaces de recherche. Chaque sous-espace étant déterminé par le parcours de l'algorithme et en fonction de la solution initiale. Par contre, diversifier souvent la recherche empêche l'algorithme de fouiller d'une manière précise les zones des sous-espaces jugées prometteuses.

Les paramètres que l'on vient de citer dépendent fortement du problème à résoudre. De plus, changer la valeur d'un des paramètres peut modifier l'influence d'autres paramètres sur l'allure de la fonction coût. Il est donc nécessaire, en fonction du problème traité, de choisir a priori un compromis entre le temps de calcul, la part d'exploration de l'espace de recherche et la précision des solutions quant aux variables continues.

En plus de cette difficulté de réglage des paramètres, et comme nous l'avons précisé dans le Chapitre 3, appliquer un algorithme de recherche tabou nécessite de définir une heuristique de mouvement permettant de déterminer la solution à choisir dans un voisinage. Ce principe

nécessite donc la définition d'une fonction objectif résultant par agrégation, ou pondération des objectifs de conception. Dans le cas de la conception pluridisciplinaire il n'est pas évident a priori de classer entre eux les objectifs poursuivis par chaque acteur disciplinaire.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Rapidité d'exécution • Solutions individuelles de très bonne qualité 	<ul style="list-style-type: none"> • Difficultés de paramétrage • Nécessité d'une pondération des objectifs • Exploration incomplète de l'espace de recherche

Tableau 2 : Avantages et inconvénients de l'approche de recherche avec pondération des objectifs : Algorithme tabou avec pas adaptatifs et constitution du front de Pareto

C'est pour pallier les inconvénients de cette approche que nous avons proposé un algorithme hybride (entre méthodes exactes et approchées) prenant en compte cette difficulté et se basant sur la notion de dominance de Pareto (cf. Chapitre 3).

II. Optimisation sans pondération des objectifs

Dans cette section, nous nous intéressons aux problèmes dans lesquels un ordre de préférence pour les objectifs n'est pas fixé a priori par les concepteurs. Ce que nous visons c'est de proposer aux concepteurs un ensemble de solutions efficaces sur lesquels ils mèneront une analyse sans avoir à exprimer une hiérarchie entre les critères.

La méthode que nous proposons vise donc à déterminer, en un temps acceptable par les concepteurs, un ensemble de solutions efficaces (non dominées) : le front de Pareto. Dans cette proposition, nous prenons en compte la multiplicité des objectifs ainsi que la mixité des variables.

1. Description de la méthode de recherche

La méthode est inspirée d'un certain nombre de mécanismes de recherche exacte et de méta-heuristique. Pour cette raison, nous l'appellerons dans la suite méthode de recherche hybride [Mouelhi'09a].

Dans le but de déterminer l'ensemble de solutions non dominées, la méthode proposée explore en un temps fixé au préalable plusieurs zones jugées prometteuses de l'espace des solutions. Elle se décompose en deux phases ; dans la première, la méthode détermine les zones à explorer d'une manière plus fine et plus précise dans la deuxième phase.

La première phase peut être identifiée comme une phase de diversification. En effet, dans le but de choisir l'ensemble des configurations (possibilités) à évaluer, une stratégie d'exploration grossière de l'espace global est mise en place. Pour cela, et en fonction de la taille du problème, reflétée par le nombre de variables et par la taille de leurs domaines, un certain nombre de configurations doivent être évaluées. La stratégie revient donc à répondre à la question : comment choisir les configurations initiales ?

Afin de répondre à cette question nous devons prendre en compte la mixité des variables en discrétisant les domaines des variables continues. Quant aux domaines des variables discrètes, les approches possibles sont les suivantes :

- Les examiner toutes si le nombre de variables et la taille de leurs domaines ne sont pas très élevés. Le cas échéant, l'exploration peut les couvrir partiellement :
- Choisir des valeurs représentatives après les avoir regroupés, dans le cas où cela est possible. Ce choix de regroupement revient aux experts métiers.
- Choisir des valeurs tirées aléatoirement.

Cette opération de discrétisation est un échantillonnage de l'espace d'entrée du problème. Plusieurs techniques, appelées techniques de maillage apportent une réponse à cette problématique. L'une d'elle, la méthode de l'hypercube latin permet de trouver simplement un maillage avec « une variance plus faible que... le simple tirage de Monte-Carlo » [Coelho'09b].

Le principe de cette méthode est de trouver un maillage n'exigeant pas un nombre de configurations égal au produit du nombre de discrétisations de tous les domaines. En effet le principe est de réduire la matrice (dans le cas de deux variables), appelée carré latin, de façon à ce que chaque configuration soit isolée sur une ligne et sur une colonne. Le carré latin étant formé par les valeurs prises par l'une des variables en lignes et par l'autre variable en colonnes.

La Figure 35 illustre l'application de la méthode sur un exemple à deux variables et où un maillage régulier sera constitué par $N*N$ configurations, où N est le nombre de discrétisations des domaines des variables.

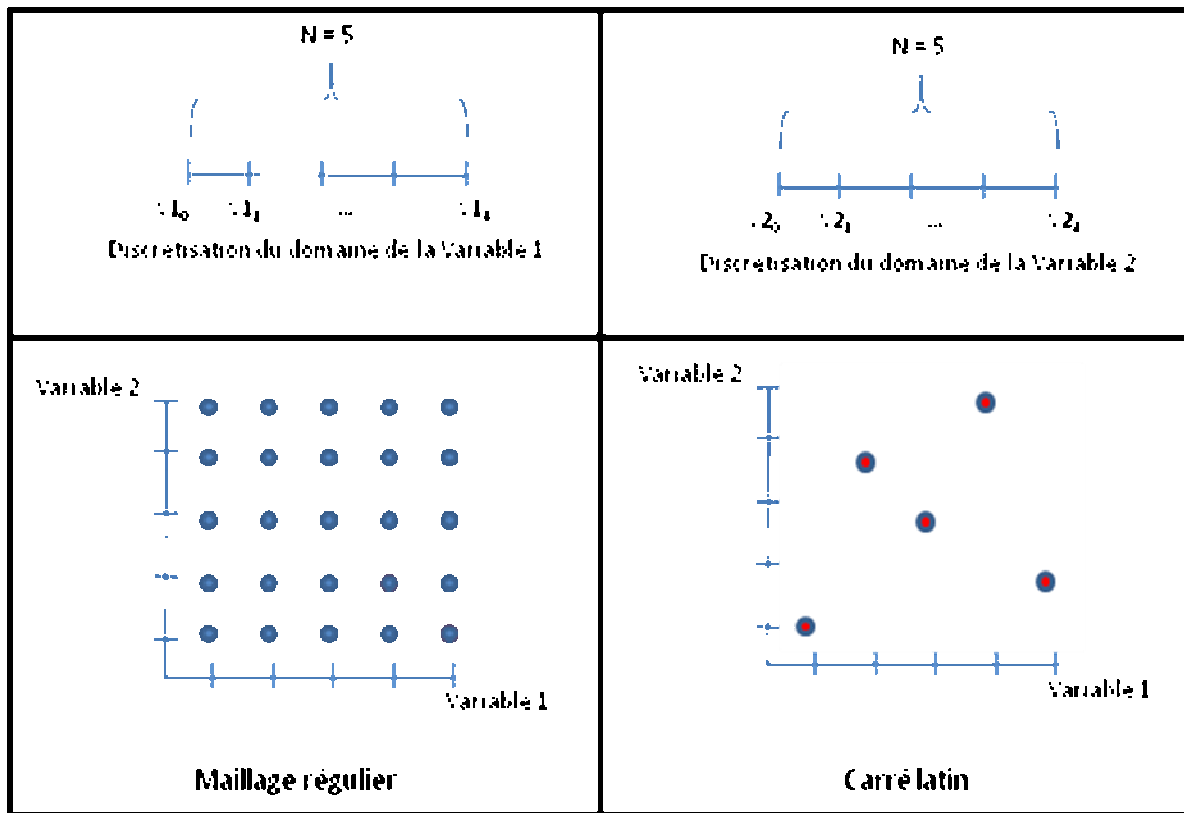


Figure 35 : Maillage par la méthode de l'hypercube latin

Après avoir choisi la stratégie d'échantillonnage, une recherche exacte est lancée afin de déterminer l'ensemble des zones prometteuses de l'espace qui sont représentées par des solutions non dominées au sens de Pareto. Cette recherche exacte peut être effectuée par une méthode de recherche exhaustive. Dans nos travaux, une recherche arborescente orientée (cf. Chapitre 3, section 2) a été privilégiée à cause de son principe de recherche permettant d'arrêter l'itération en cours dès que l'une des contraintes du problème est violée. En effet, en construisant séquentiellement des solutions au moyen d'un arbre de recherche par Séparation et Evaluation, le problème initial est décomposé en sous-problèmes (appelés aussi sous-espaces) en intégrant à chaque fois de nouvelles contraintes.

Afin de mieux illustrer notre approche, nous avons pris l'exemple d'un problème à deux variables. La Figure 36 (cadre de gauche) montre l'espace des entrées formé par les configurations possibles des deux variables. La distance entre les projections de deux points sur un axe représente le pas de discrétisation de la variable représentée par ce même axe (PMV1 et PMV2).

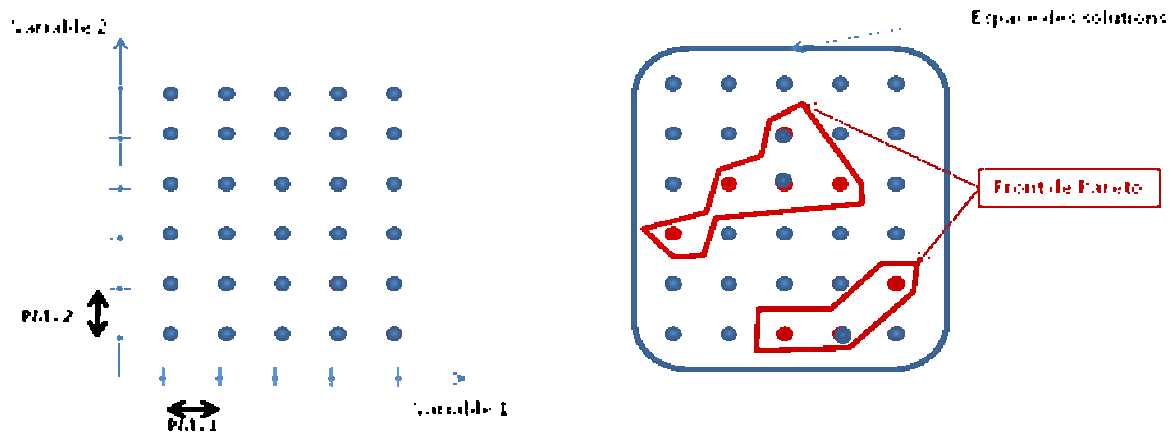


Figure 36 : Initialisation de l'espace de recherche et résultat de la phase 1 dans le cas d'un problème à deux variables ; à gauche : Espace de recherche; PMVi : Pas Maximal de discrétisation de la Variable i ; à droite : Front de Pareto déterminé après la première phase de recherche exhaustive.

Lors de la deuxième phase, les zones prometteuses de l'espace représentées par les solutions non dominées dégagées lors de la première phase, sont explorées d'une manière plus fine et précise.

Pour ce faire, un mécanisme d'exploration des voisinages de ces solutions, semblable à celui des méthodes de recherche locale, est mis en place. En effet, lors des différentes itérations de cette phase, les pas de discrétisation des variables continues sont réduits et une exploration des nouveaux voisinages des solutions du front de Pareto est lancée.

Une itération consiste en une, et une seule, diminution de tous les pas de discrétisation des variables et une évaluation des voisinages de toutes les solutions du front de Pareto dégagées lors des itérations précédentes. Toutes les solutions non dominées rencontrées lors de ces itérations successives, que l'on peut comparer aux phases d'intensification des méthodes de recherche locale, sont rajoutées au front de Pareto.

Un mécanisme de liste tabou, semblable à celui mis en place dans la première section de ce chapitre, est utilisé afin d'éviter le piègeage par les optimums locaux.

Les principes de la méthode proposée sont illustrés par l'algorithme de recherche hybride HS (pour *Hybrid Search*) [Mouelhi'09a] de la Figure 37.

Algorithm : Hybrid Search

```

begin
  Steps ← max (Steps);
  Pareto ← non-dominated solutions (exhaustive search);
  add (Pareto, Tabu list);
  repeat
    Steps ← reduce (Steps);
    for each  $S_i$  in Pareto front do
      { $N(S)$ } ← neighborhood ( $S$ );
      for each  $S_j \in N(S)$  do
        if  $S_j$  not dominated and not tabu then
          | add ( $S_j$ , Pareto);
          | add ( $S_j$ , Tabu);
        update (Pareto);
  until Stop criterion;
end;

```

Figure 37 : Principe de base de l'algorithme de recherche hybride avec détermination du front de Pareto

2. Mécanismes de l'algorithme de recherche

Après la définition du maillage initial de l'espace des variables de conception continues, la phase de diversification est lancée à travers la recherche exhaustive du front de Pareto qui sera copié dans la liste tabou.

Une fois cette phase de diversification terminée, plusieurs phases d'intensifications sont lancées successivement. En effet, chaque itération de la boucle principale de notre algorithme représente une étape d'intensification.

Lors de chaque étape, et pour explorer d'une manière plus précise et plus fine, ces zones prometteuses, les pas de discrétisation sont réduits et une exploration du voisinage de chaque solution du front de Pareto est lancée.

Choix du nombre de phases d'intensification

Puisque les pas de discrétisation minimaux et maximaux sont fixés à l'avance par les concepteurs, choisir le nombre de phases d'intensification revient à fixer le nombre de réductions à appliquer sur les pas de discrétisation. Dans le cas d'une seule phase d'intensification, on convient de choisir la valeur du pas minimum dès le début du processus de recherche locale.

Peu de phases d'intensification implique un passage rapide vers des pas de discrétisations très petits, et donc un niveau de détail très élevé pour les variables de conception, ce qui augmente le temps de recherche puisque la propagation de la recherche dans les zones prometteuses se fait lentement.

Par contre, choisir un nombre élevé de phases d'intensification implique une rapidité d'exécution puisque l'algorithme élague assez rapidement des parties de l'espace de recherche. Ceci s'explique par le fait que la vérification du front de Pareto se fait plus fréquemment et que des zones de l'espace sont abandonnées dès que l'algorithme trouve dans une autre zone une solution plus efficace

Au cours de la recherche, chaque solution visitée est rajoutée dans la liste tabou, et si elle n'est pas dominée, elle est rajoutée au front de Pareto.

Gestion de la liste tabou

Comme pour le premier algorithme (décrit dans la première section de ce chapitre), nous avons choisi de mémoriser toute la solution dans la liste tabou et pas seulement le mouvement qui y a amené. Mais encore plus, cette fois ce choix est obligatoire puisque l'exploration de l'espace se fait de plusieurs endroits différents. Nous désirons donc interdire l'exploration de sous-espaces déjà accédés par d'autres processus de recherche.

Gestion de la liste Pareto

Plusieurs versions du mécanisme de gestion de la liste Pareto ont été proposées, implémentées et testées. Ces variantes portent principalement sur « quand ? » mettre à jour la liste complète, c'est-à-dire supprimer les solutions dominées par la (ou les) dernière(s) solution(s) rajoutée(s), et sur « où » rajouter une solution non dominée dans la liste. Ces quatre possibilités sont résumées dans le Tableau 3.

Quand ?	Immédiatement après l'ajout	A la fin de la phase d'intensification en cours
Où ?	Juste après la solution courante	A la fin de la liste de Pareto

Tableau 3 : Paramètres de mise à jour de la liste de Pareto

Pour répondre à la première interrogation « quand ? », deux possibilités s'offrent à nous :

- Dans la première possibilité, à chaque fois qu'une solution est rajoutée au front, elle est directement comparée à l'ensemble des solutions qui y étaient. L'algorithme compare donc chaque solution avec celles déjà contenues dans le front de Pareto jusqu'à ce qu'on trouve une solution la dominante. Le cas échéant elle est rajoutée à la liste. Tout en vérifiant cette condition, toutes les solutions dominées par la dernière sont alors supprimées.
- Dans la deuxième possibilité, chaque solution qui n'est pas dominée par la solution courante, qui y a amené grâce à un mouvement, et figurant déjà dans la liste de Pareto est rajoutée automatiquement. A la fin de chaque phase, tout le front est mis à jour en comparant toutes les solutions deux à deux et en supprimant les solutions dominées.

Nous avons remarqué que ce processus élague rapidement des zones de l'espace de recherche qui peuvent s'avérer intéressantes. En effet, ce mécanisme permettrait de supprimer des solutions du front sans avoir évalué leurs voisinages.

La deuxième version est plus lente que la précédente à cause de la taille du front et à cause du fait qu'à chaque mise à jour, il faut comparer toutes les solutions du front deux à deux. Néanmoins, la qualité des fronts obtenus est meilleure.

On estime qu'un front FA est meilleur qu'un front FB lorsqu'il y'a plus de solutions de FA qui dominent des solutions de FB, et lorsque l'espace des solutions couvert par FA est plus large que celui couvert par FB.

En ce qui concerne la deuxième interrogation : « où ? » rajouter la solution ? Deux possibilités s'offrent également à nous :

- Dans la première possibilité, une solution non dominée est rajoutée directement après la solution courante. Ce qui implique qu'elle sera la prochaine solution dont l'algorithme va explorer le voisinage. Partant de ce principe, l'algorithme explore d'une manière efficace chaque sous-espace jusqu'à ce qu'il arrive à un optimum local, où tout le voisinage est soit tabou, soit dominé par la solution courante.
- Dans la deuxième possibilité, une solution non dominée est rajoutée à la fin de la liste. C'est-à-dire que l'exploration des voisinages des solutions voisines ne commence qu'après l'évaluation des voisinages de toutes les solutions placées dans la liste au début de l'itération en cours. Plus concrètement, soit S l'ensemble de solutions placées dans la liste de Pareto au début d'une itération bien déterminée, et soit la fonction V qui détermine les solutions voisines de S . Le principe de cette version est que l'exploration de $V(V(S))$ ne commence que si on a évalué toutes les solutions de $V(S)$.

La première version peut être comparée à l'algorithme Multi-Start Tabou Search où, on sélectionne un ensemble de solutions au départ et on lance une recherche de type tabou à partir de chacune.

La deuxième version est inspirée des algorithmes évolutionnaires où, à chaque itération de l'algorithme, un ensemble de solutions est généré, puis évalué, et les meilleures solutions sont retenues. Contrairement aux méthodes de recherche locale où une seule solution est retenue à chaque itération, cette version consiste à « zoomer » progressivement sur plusieurs zones de l'espace d'une manière quasi-simultanée.

Après des expérimentations sur un cas particulier –qui sera développé dans le Chapitre 5 de ce manuscrit- nous avons analysé le comportement de l'algorithme avec les différentes possibilités de paramétrage de la liste Pareto [Mouelhi'09c]. La Figure 38 synthétise nos conclusions quant au choix de ces paramètres.

Quand ?	A la fin de la phase	<ul style="list-style-type: none"> • Equilibre entre exploration et exploitation • Bonne couverture de l'espace de recherche • Temps d'exécution moyen 	<ul style="list-style-type: none"> • Exploration grossière • Très bonne couverture de l'espace de recherche • Temps d'exécution élevé (s'il n'est pas fixé au début)
	Immédiatement	<ul style="list-style-type: none"> • Exploration excessive de peu de zones de l'espace de recherche • Rapidité d'exécution 	<ul style="list-style-type: none"> • Equilibre entre exploration et exploitation • Bonne couverture de l'espace de recherche • Temps d'exécution moyen
		Immédiatement	A la fin de la liste

Ou ?

Figure 38 : Choix des paramètres de gestion de la liste de Pareto

La version finale que nous avons retenue pour notre algorithme [Mouelhi'09d] est celle où les mises à jour de la liste se font à la fin de chaque phase. Cette version est consolidée par le rajout des solutions à la fin du front. Ces choix permettent d'évaluer, à chaque itération, les voisinages de toutes les solutions du front avant d'évaluer les voisinages de leurs solutions voisines et non dominées. Ce qui permet d'explorer d'une façon identique, le plus possibles de zones de l'espace de solutions ; et par suite de ne pas consacrer la majeure partie du temps imparti à la fouille d'un nombre réduit de zones.

De plus, les concepteurs, une fois qu'ils arrêtent leur choix sur une solution bien déterminée, un processus de fouille précise peut être appliqué sur le voisinage de cette solution d'une manière plus exhaustive...

Critère d'arrêt de l'algorithme

Le choix du temps d'exécution comme critère d'arrêt de l'algorithme a été choisi pour limiter le temps d'exécution de l'algorithme en cas de non-convergence. En effet, la phase de diversification initiale, permet de déterminer un ensemble de solutions non dominés dont le nombre dépend fortement de la nature du problème. Aussi, elle dégagera un nombre très faible voir nul quand le problème est très contraint voire non faisable. Un nombre élevé sera

dégagé quand les contraintes sont plus relâchées et les objectifs sont décrits par des intervalles d'acceptation et non par des valeurs fixes.

Dans ce deuxième cas, où le nombre de solutions de départ peut être élevé, il y a un risque élevé d'une explosion combinatoire puisque le nombre de solutions évaluées est multiplié par le nombre de solutions voisines qui est égal au nombre de variables * 2 à chaque itération ; chaque solution explorée induit l'évaluation de toutes ses voisines, et les solutions non dominées sont rajoutées au front de Pareto.

En conclusion, les avantages de l'approche hybride d'optimisation proposée peuvent être résumés comme suit :

- Ne nécessite pas de choix de préférences (poids) pour les objectifs de conception
- Très bonne exploration de l'espace des solutions.
- Temps d'exécution choisi a priori par les concepteurs.
- Equilibre entre précision sur les valeurs des variables et étendue du front : Par le choix simple du nombre de phases d'intensification
- Facilité de paramétrage

Une limitation de cette méthode est qu'elle est plus lente qu'la recherche tabou classique. Cette lenteur est compensée par l'exploration de plus de sous-espaces de recherche.

Dans la section suivante, nous présentons les cartes auto-organisatrices. Ces méthodes sont généralement utilisées pour la discrimination. Notre intérêt pour ces cartes s'articule autour de leur capacité à aider les concepteurs lors de l'interprétation des fronts de Pareto dégagés par les méthodes de recherche présentées plus haut ; ces ensembles étant en général volumineux et multidimensionnels.

III. Cartes auto-organisatrices

Afin d'interpréter et d'évaluer les fronts de Pareto obtenus, nous sommes intéressés à des méthodes et outils qui offrent aux concepteurs la possibilité de visualiser les fronts obtenus. En effet, vu la multidisciplinarité des concepteurs, vu le grand nombre de variables de conception et des objectifs du problème, nous désirions offrir aux concepteurs un visuel de

l'ensemble des solutions du front, des valeurs retenues pour les variables et des valeurs calculées pour les objectifs.

Cette étude nous a amenés vers les méthodes de discrimination et nous avons donc identifié un ensemble de méthodes [Oja'01] tels que : le clustering flou, LBG / LBG-U, Hard Competitive Learning, Growing Grid, Competitive Hebbian Learning, Kohonen maps...

L'aspect visuel ayant une grande importance dans le contexte de notre travail, notre intérêt s'est particulièrement porté sur les cartes de Kohonen reconnues pour leur capacité à représenter, en deux dimensions, des données initialement placées dans un espace de dimensions supérieur à deux [Kohonen'01].

Les cartes de Kohonen, qui sont des cartes auto-organisatrices, permettent de visualiser simplement ces représentants sur une carte de dimensions réduites par rapport à celles de l'espace des données. Elles fournissent des méthodes de quantification vectorielle qui placent un nombre fini de représentants dans une distribution de données tout en conservant ses propriétés de voisinage et de densité. Elles utilisent des réseaux de neurones artificiels basés sur des processus d'apprentissage non supervisés. Pour voir la description détaillée de la méthode [Vesanto'99], le lecteur pourra se référer à [Mouelhi'08]. Cela est d'un apport qualitatif considérable puisque les concepteurs disciplinaires peuvent avoir un ensemble d'informations pertinentes tels que :

- Domaine d'utilisation des composants
- Pour chaque solution, les valeurs des variables et des objectifs de conception
- Clusters de solutions ayant des valeurs (variables et/ou objectifs) proches
- Influence des variations de certaines variables sur les objectifs de conception : Une variation majeure de la valeur d'une variable peut n'avoir aucune influence sur les choix technologiques. Alors qu'une légère variation peut faire basculer la solution, c'est le cas de solutions se trouvant sur la frontière entre deux clusters voisins...

Bien que les cartes de Kohonen présentent un avantage considérable à travers cet aspect, visuel, elles ont l'inconvénient de ne pas être incrémentales. C'est-à-dire, qu'une fois une carte de Kohonen est générée, la modification de l'ensemble de données de départ n'aura aucune incidence sur sa structure finale. En d'autres termes, afin d'incorporer de nouvelles connaissances dans la distribution initiale des données, le processus d'apprentissage

permettant de générer les cartes est relancé depuis le début. C'est pour cette raison que nous nous sommes intéressés aux méthodes de clustering incrémentales dont la structure de la carte évolue au cours du processus. Les méthodes les plus connues sont :

- Incremental Growing Neural Gas (IGNG) [Ennaji'04], [Ennaji'05],
- Incremental Grid Growing [Blackmore'95],
- Adaptative Hierarchical Incremental Grid Growing [Merkel'01], et
- An incremental Hierarchical Clustering [Ribert'99].

Notre choix s'est porté sur la méthode IGNG, celle-ci étant, à notre connaissance, la plus récente des méthodes incrémentales de discrimination. L'application de cette méthode nous a permis d'avoir une meilleure représentation des solutions du front de Pareto. De plus, son aspect incrémental a été vérifié au cours de notre étude [Mouelhi'08]. C'est-à-dire que l'apprentissage sur les données peut être repris lors de l'introduction de nouvelles solutions dans l'ensemble de départ. Cette méthode présente l'inconvénient de ne pas fournir un visuel aux concepteurs, vu la complexité et l'irrégularité de la structure des cartes générées.

Les deux méthodes de discrimination sélectionnées sont donc complémentaires. En effet, l'utilisation des cartes de Kohonen dans le contexte de notre travail peut être considérée comme un apport qualitatif. Leur utilisation permet aux concepteurs d'apprécier l'ensemble des solutions du front de Pareto et d'avoir des diverses informations intéressantes que l'on illustre dans le prochain chapitre.

Quant à l'utilisation de la méthode IGNG, elle peut être considérée comme un apport quantitatif. Cette méthode crée des représentants aux données d'entrée (le front de Pareto) avec, dans les cas que nous avons étudiés, une erreur moyenne inférieure à celle des représentants de la carte de Kohonen. Elle permet également l'adaptation de la topologie de la carte dans le cas d'un rajout, a posteriori, de nouvelles connaissances.

Ces deux méthodes de discrimination sont détaillées dans l'Annexe A, et la contribution de leur utilisation sur des exemples est présentée dans le Chapitre 5.

Conclusion

Nous avons proposé un outil pour aider les concepteurs lors de l'optimisation du choix des variables en conception organique. Cet outil propose deux fonctionnalités, à savoir, la

détermination du meilleur ensemble de solutions efficaces en un temps fixé au début, et l'interprétation de ces fronts de Pareto.

D'un côté, cet outil permet de déterminer, en un temps fixé par les concepteurs, l'ensemble des solutions efficaces du problème. Pour ce faire, deux méthodes de recherche permettent de choisir les variables de conception de manière à optimiser les objectifs de conception, tout en respectant les contraintes. Les deux méthodes proposées permettent de tenir compte des deux possibilités où une préférence hiérarchique peut ou pas être fixée a priori pour les multiples objectifs à optimiser (géométriques, techniques, énergétiques, économiques, etc.).

D'un autre côté, il permet, à l'aide des méthodes de discrimination, de fournir une aide à l'interprétation de ces ensembles de solutions, souvent volumineux et multidimensionnels.

Lors de ces travaux, plusieurs pistes nous sont parues intéressantes pour de travaux futurs d'amélioration des algorithmes proposés.

Dans un premier temps, il nous paraît intéressant de proposer une alternative aux concepteurs lors des choix d'un ordre hiérarchique pour les objectifs de conception. En effet, plusieurs méthodes permettant d'aider les concepteurs lors du choix des poids de la fonction objectifs, telles que la méthode MachBeth, existent. Une piste intéressante serait d'automatiser la pondération de la fonction objectif de l'algorithme tabou proposé afin de permettre aux concepteurs de faire des choix, en fonction de ses préférences entre des solutions concrètes et non entre des valeurs des poids des objectifs pouvant paraître insignifiants au début du processus.

De plus, les deux méthodes d'optimisation proposées permettent de fixer ou non ces poids a priori. Cela nous paraît intéressant de proposer une méthode itérative permettant aux concepteurs d'arrêter la recherche de solution à n'importe quel moment, et de réorienter la recherche. Pour la première méthode de recherche, ils modifieraient les poids de la fonction objectif. Pour la deuxième, l'interprétation du front de Pareto à travers des cartes de Kohonen permettrait de redéfinir les domaines des variables, de relâcher certaines contraintes du problème...

Une deuxième piste s'est dégagée lors de nos travaux concernant le problème de discrétisation des domaines des variables continues. En effet, il est toujours difficile de choisir les pas de discrétisation. Il nous paraît intéressant de choisir les limites des intervalles de discrétisation avec un mécanisme semblable à celui des cartes auto-organisatrices en général, et celui de la

méthode IGNG en particulier. Il s'agit de choisir des valeurs au hasard dans les intervalles et de placer des représentants pour les configurations ayant des évaluations proches...

Une troisième piste concerne l'exploration de l'espace de recherche par les algorithmes proposés. Une piste intéressante pour des travaux futurs serait de créer un réseau IGNG au fur et à mesure de l'exécution de l'algorithme. Ceci permettrait d'orienter la recherche vers les zones de l'espace suivant les besoins des concepteurs. Des représentants seront placés partout où la recherche a été effectuée afin de pouvoir informer les concepteurs sur les qualités des solutions dans de nombreuses zones de l'espace même non intéressantes. Ceci est motivé par le fait que lors de futures phases du processus de conception, ces mêmes zones pourraient contenir des solutions efficaces. Ceci serait causé, par exemple, par les changements des contraintes du problème...

Chapitre 5 : Aide au choix de variables de conception : Deux cas d'illustration

Introduction

Que ce soit dans les domaines de la micro robotique, ou de la production, la maîtrise des interactions entre composants d'un produit ou entre ce produit et son environnement fédère des compétences multiples [Kopacek'09], [Xie'09]. La conception de robots est un exemple de conception multidisciplinaire intégrant la connaissance de plusieurs métiers : mécanique, mathématiques, contrôle, électricité, informatique, etc. Un robot peut être considéré comme un système complexe car il présente des caractéristiques des « sources de complexité » rappelées au Chapitre 1.

Selon le modèle du cycle de développement de l'Ingénierie Système nous nous plaçons dans le cadre du processus de conception organique (cf. Chapitre 1, Section II) et nous nous intéressons plus particulièrement à l'activité support d'optimisation de ce processus.

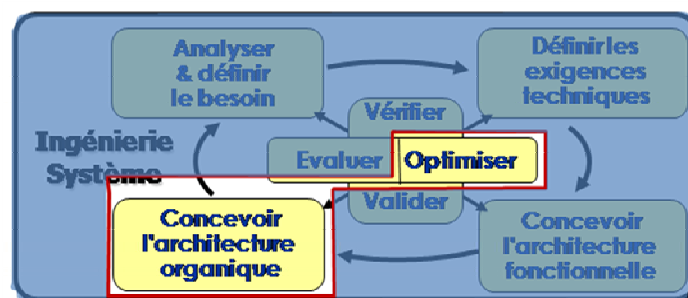


Figure 39 : Optimisation en conception organique

Nous illustrons notre contribution au support d'aide à la sélection d'architecture à travers deux exemples de conception en robotique.

Dans le premier exemple, conception d'un robot planaire, nous proposons un outil pour le choix des variables de conception dans un cadre multiobjectif et multidisciplinaire. Les variables disciplinaires étant couplées, une des méthodes de MDO, vue au Chapitre 2 est utilisée afin de traiter ce couplage. Dans le but d'optimiser les choix de ces variables, les deux algorithmes d'optimisation introduits au Chapitre 4 sont appliqués afin de déterminer l'ensemble des solutions efficaces (c.-à-d. non dominées au sens de Pareto). Ces solutions seront interprétées en utilisant les cartes de Kohonen.

Le second exemple est issu du travail de conception d'un robot désempileur pour l'industrie textile [Rakotoarisoa'93]. Etant composé de trois axes corrélés (ou couplés), le choix des

variables de conception et des composants de chaque axe influence fortement les choix des variables et des composants des deux autres. Le but de cette étude est d'apporter aux concepteurs un outil d'aide à l'évaluation et à l'interprétation d'un grand ensemble de solutions multidimensionnelles. Nous mettons aussi en évidence l'apport de l'application des méthodes de discrimination basées sur les cartes auto-organisatrices et citées en Section 3 du Chapitre 4.

I. Optimisation multiobjectif en conception d'un robot à deux degrés de liberté

Le besoin exprimé par le cahier des charges client est de développer un système capable de déplacer une charge variable sur différents points d'un plan de travail. Cette problématique a servi d'exemple à Cirrus [Cirrus'98] dans le cadre de son mémoire de PhD portant sur l'optimisation des choix de stratégies (ou méthodologies) de conception de produits. Nous reprenons cet exemple pour son caractère pédagogique et nous réutiliserons des modèles ainsi que des données techniques fournies par Cirrus. Dans cet exemple, les disciplines concernées dans la conception de tels robots sont la cinématique, la résistance des matériaux, l'automatique...

Dans cette section, nous supposons avoir choisi le type d'architecture du robot. Le problème du choix des variables de conception est formalisé en termes d'un problème d'optimisation multidisciplinaire (OMD). Dans le but de choisir la (ou les) meilleure(s) architectures satisfaisant au mieux les contraintes du problème et optimisant ses objectifs multiples, les méthodes introduites au Chapitre 4 sont appliquées et les résultats de leurs exécutions discutés.

1. Problématique

Nous commencerons par rappeler le cahier des charges et donnerons les objectifs et contraintes du problème, puis nous donnerons l'architecture retenue et formalisons le problème du choix des valeurs des variables de conception en terme d'optimisation multi-objectifs et multidisciplinaire.

A. Cahier des charges

Le client exprime à travers le cahier des charges les fonctionnalités du robot à concevoir. Dans notre cas, le client définit l'espace de travail à travers l'ensemble des points à atteindre

par l'extrémité du robot, la valeur de la charge à transporter, le temps de parcours et la précision du positionnement ainsi que le dépassement maximum du point à atteindre pendant un transfert.

Lors des processus d'analyse des besoins, de la spécification des exigences techniques et de la définition de l'architecture fonctionnelle, de nouvelles exigences techniques ont été dégagées par les concepteurs comme, par exemple : garder la flèche de l'extrémité du système à une faible valeur par rapport à la longueur des bras du robot pour assurer une rigidité à la structure, préciser les limites de vitesse et d'accélération des pièces en mouvement.

B. Architecture du robot

La proposition d'architecture du robot suit la définition de l'architecture fonctionnelle. Nous nous situons alors au processus de conception organique où le problème est défini en termes d'affectation de valeurs aux variables de conception.

Comme la montre la Figure 40, l'architecture générale retenue, un robot planaire à deux degrés de liberté imaginé par les concepteurs, est composé de deux segments ou axes. Le premier axe est fixé au plan de travail par sa base d'un côté, et relié de l'autre côté au deuxième axe qui porte sur son extrémité la charge à déplacer.

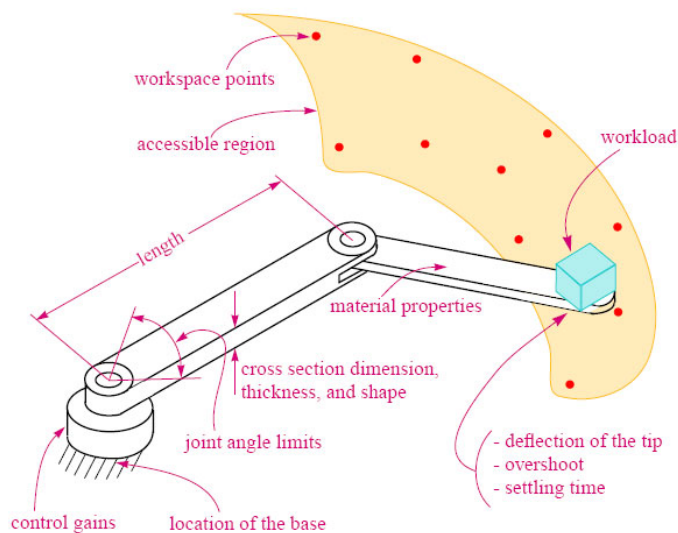


Figure 40 : Robot à deux degrés de liberté [Cirrus'98]

Dans le cahier des charges, la répartition des points, devant être atteints par l'extrémité du robot et formant l'espace de travail, décrit un petit M (« small M ») comme le montre la Figure 41.

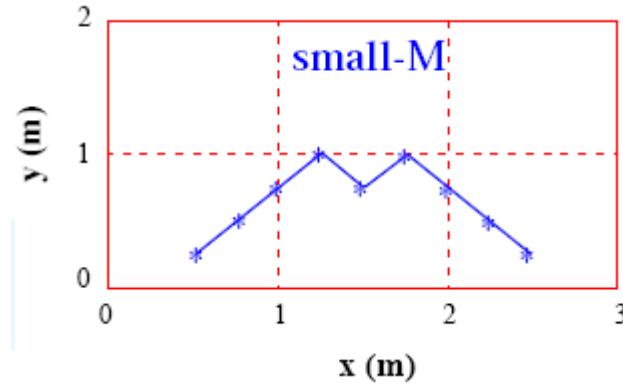


Figure 41 : Ensemble des points à atteindre par l'extrémité du robot formant un petit M

C. Variables et objectifs de conception :

Afin de définir l'architecture décrite ci-dessus, les variables de conception sont regroupées par rapport aux points de vue disciplinaires : cinématique, structure des matériaux et dynamique.

- Les variables cinématiques : la localisation de la base du robot dans le plan de travail (X_{base} et Y_{base}), les longueurs des bras (L_1 et L_2), et le type des angles (configuration gauche ou droite du « coude du robot »: Or).
- Les variables structurelles : le type de matériel (mat), les sections (D_1 et D_2), le rapport entre l'épaisseur et le diamètre des bras (Thickness : Th). Ces variables et les longueurs des bras et doivent être choisis de manière à préserver la solidité du robot et de réduire son poids.
- Les variables dynamiques : le temps de réponse (Settling Time : ST) et le dépassement maximum du système (Overshoot : ov).

L'objectif des équipes de concepteurs est de choisir la meilleure combinaison de variables de conception afin d'optimiser les objectifs suivants :

- Maximiser la région accessible (domaine cinématique) : l'ensemble des points dans l'espace accessible par la pointe du robot est appelé la région accessible (AR).
- Minimiser la flèche (domaine structure) : la flèche des bras (Deflection : Def) est un indicateur de la rigidité du robot.
- Minimiser les gains de commande (domaine dynamique) : la loi de commande doit garantir les performances du robot, en ce qui concerne la stabilité et la précision. Les gains de commande reflètent l'effort requis par le système de contrôle pour fournir l'accélération et la vitesse souhaitées et la précision des performances. De grands

gains de commande impliquent des actionneurs puissants qui augmentent le coût global du système. Les gains de commande sont définis pour chaque axe sous la forme de gains proportionnel (KP_1 et KP_2) et de gains dérivatif (KD_1 et KD_2).

- Minimiser le poids total du robot (Weight : W) impacte la puissance des moteurs et des composants électroniques de puissance qui entrent dans le prix du robot.

Les choix des concepteurs sont limités car ils doivent respecter les contraintes suivantes :

- L'ensemble des points de l'espace de travail (Workspace) doivent être atteignables par l'extrémité du robot.
- Les longueurs des bras et la position de la base du robot sont limitées.
- Des valeurs d'acceptation sont fixées pour les objectifs du problème.
- Le moment de flexion de chaque segment est inférieur à la limite d'élasticité.

Dans la suite, le problème est formalisé en termes d'optimisation multidisciplinaire.

D. Formalisation du problème:

La Figure 42 représente les entrées et les sorties du processus de conception organique. En effet, les différentes équipes de conception des domaines cinématique, dynamique et résistance des matériaux, devront proposer des valeurs des variables de conception optimales (ou proches de l'optimum) par rapport aux objectifs et contraintes.

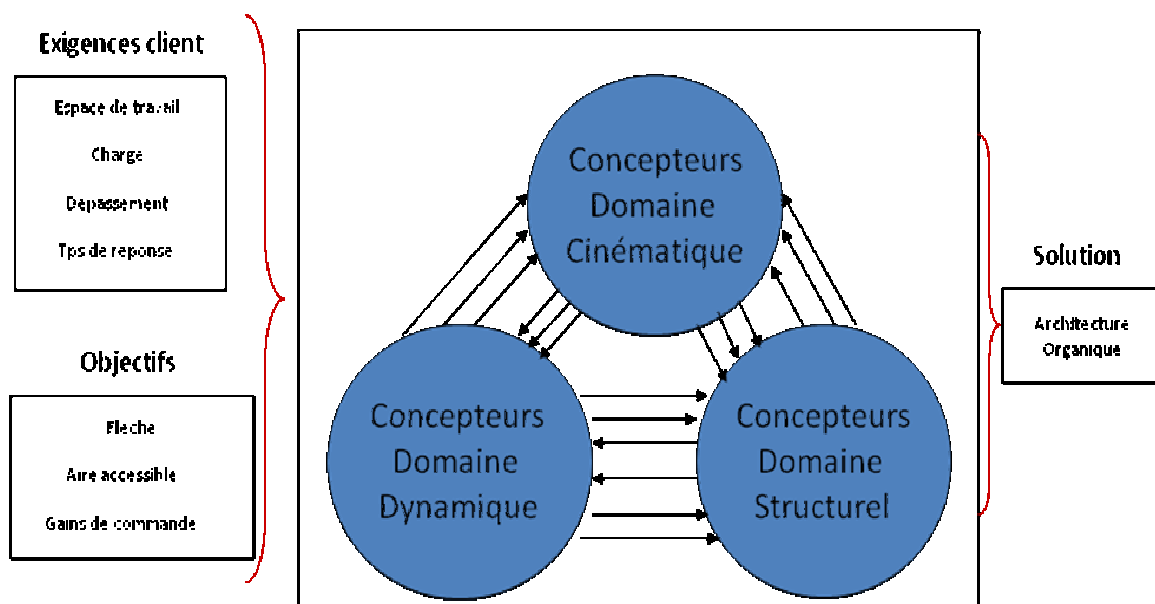


Figure 42 : Entrées, sorties et domaines du processus de conception organique du robot à deux degrés de liberté

Les variables sont de nature mixte. En effet, certaines sont continues telles que les longueurs des bras, les coordonnées de la base..., d'autres sont discrètes, comme les matériaux, l'orientation des bras...

Dans le Chapitre 2, nous avons vu que le problème de couplage peut être résolu par l'une des deux stratégies mono ou multi-niveaux. Dans cet exemple, nous disposons de toutes les données techniques concernant le robot d'où l'application des méthodes mono-niveau est privilégiée. En adoptant cette stratégie, diverses méthodes peuvent être appliquées, telles que AAO, IDF ou MDF.

Le nombre des variables étant faible, nous optons pour la méthode AAO dont le principe est de considérer toutes les disciplines au même niveau. Dans ce cas, le choix de toutes les variables de conception est fait par l'optimiseur multidisciplinaire.

Partant de ces choix, notre problème consiste donc à trouver :

$$F^* = \min_D F(V) ,$$

Sous contraintes :

$C(V)$

Où :

- $V = \{X_{base}, Y_{base}, L_1, L_2, D_1, D_2, Th, Mat, ST, Or\}$: Vecteur composé des 10 variables d'optimisation, avec $\{L_1, L_2\}$ variables de couplage entre les domaines cinématique, structurel et dynamique, $\{D_1, D_2, Th, Mat\}$ variables de couplage entre les domaines dynamique et structurel, $\{X_{base}, Y_{base}\}$ variables partagées, $\{Or\}$ variable disciplinaire associée au domaine cinématique, $\{ST\}$ variable disciplinaire du domaine dynamique.
- $D = \{d_1, d_2, \dots, d_3\}$: Domaine hyper rectangulaire de V ; $d_i =$ domaine de v_i ,
- $F(V) = \{Def, AR, W, KP_1, KD_1, KP_2, KD_2\}$: Ensemble des 7 objectifs, et
- $C(V) = \{c_1(X), c_2(X), \dots, c_k(X)\}$: Ensemble des k contraintes du problème.

Les fonctions d'analyse servant à calculer les objectifs en fonction des variables ainsi que les contraintes du problème sont détaillées dans l'Annexe B.

Dans cette section, nous avons défini l'architecture organique du robot à deux degrés de liberté. A l'aide de la méthode AAO, nous avons levé le couplage existant entre les différentes disciplines et le problème a pu être défini comme un problème d'optimisation multiobjectif

contraint. Dans la section 2, nous allons appliquer les méthodes d'optimisations proposées dans le Chapitre 4 afin de trouver le front de Pareto correspondant à l'ensemble des solutions efficaces de ce problème.

2. Outil d'optimisation multiobjectif pour le choix d'architecture

Le problème de choix d'architecture étant formulé comme un problème d'optimisation multiobjectif contraint, deux stratégies d'optimisation s'offrent à nous. Dans la première, les différents objectifs du problème peuvent être agrégés et une fonction objectif résultante nous permet d'appliquer la méthode de recherche tabou avec pas adaptatifs.

La seconde stratégie, consiste à dégager l'ensemble des solutions efficaces afin de permettre aux concepteurs de faire des choix dans l'espace des solutions plutôt que de fixer un ordre de préférence hiérarchique pour les objectifs du problème. Dans ce cas, la méthode de recherche hybride est appliquée à l'exemple étudié dans cette partie.

Dans les deux parties nous donnerons les détails sur les paramétrages des deux algorithmes. Nous décrirons brièvement l'application des méthodes de discrimination, introduites dans le Chapitre 4, afin d'interpréter les fronts de Pareto obtenus par les deux approches. L'apport principal de l'application de ces méthodes sera étudié plus en profondeur dans la section II de ce chapitre.

Afin de pouvoir appliquer les méthodes de recherche, nous discrétisons les domaines des variables continues. Le Tableau 4 illustre les valeurs des pas minimaux et maximaux choisis ainsi que les valeurs minimales et maximales définissant les domaines de ces variables.

Variable	Pas minimal	Pas maximal	Valeur minimale	Valeur maximale
X_{base} (cm)	0.1	10	$X_0 - L_0/2$	$X_0 + L_0/2$
Y_{base} (cm)	0.1	10	$Y_0 - L_a/2$	$Y_0 + L_a/2$
L_1 (cm)	0.1	10	0.1	200
L_2 (cm)	0.1	10	0.1	200

Tableau 4 : Variables continues : pas minimaux et maximaux et leurs domaines

Où : X_0 , Y_0 , La et Lo représentent respectivement l'abscisse du centre, l'ordonnée du centre, la largeur et la longueur du rectangle circonscrit à l'espace de travail.

En faisant ces choix, et en considérant les valeurs des pas minimales pour la discrétisation des domaines des variables continues, on dispose de plus de 325 milliards de configurations possibles à évaluer. Ce nombre est obtenu par la multiplication des tailles des domaines de toutes les variables.

Les algorithmes ont été implémentés en langage Java, dans un environnement Eclipse. Les diverses simulations ont été effectuées sur une machine équipée d'un processeur à 3 GHz de fréquence et à 2 Go de mémoire RAM.

A. Méthode tabou continu avec pas adaptatifs

Dans cette première simulation, la fonction objectif a été fixée comme étant la somme normalisée des valeurs des objectifs à minimiser et de l'opposé des valeurs des objectifs à maximiser. On cherche donc à minimiser la somme de la flèche, du poids total, des gains de commande, du poids total et de l'opposé de l'aire accessible ce qui revient à minimiser F , avec $F = Def / \Delta_{Def} + W / \Delta_W + KP_1 / \Delta_{KP1} + KD_1 / \Delta_{KD1} + KP_2 / \Delta_{KP2}$

$$+ KD_2 / \Delta_{KD2} - AR / \Delta_{AR}.$$

Où les Δ représentent les différences entre les valeurs maximales et minimales pouvant être prises par les variables. Ces valeurs sont définies par les experts disciplinaires et représentent les intervalles d'acceptation pour chaque discipline.

Paramètres de l'algorithme tabou

Afin de fixer le réglage des paramètres de l'algorithme, une étude approfondie a été menée. Les paramètres des pas adaptatifs ayant été choisis par les concepteurs (experts) en fonction du contexte du problème, de la même manière pour les deux algorithmes (Tableau 4), nous passons aux paramètres suivants, à savoir, la durée pendant laquelle une solution reste taboue, et le temps de calcul maximum de l'algorithme.

Nous avons fait varier la Durée Tabou (DT) entre 1000 et 50000 itérations avec un pas de 1000. La Figure 43 illustre la valeur de la Fonction Objectif (FO) en fonction du Temps de Calcul (TC) pour différents ensembles de tracés représentant le même comportement.

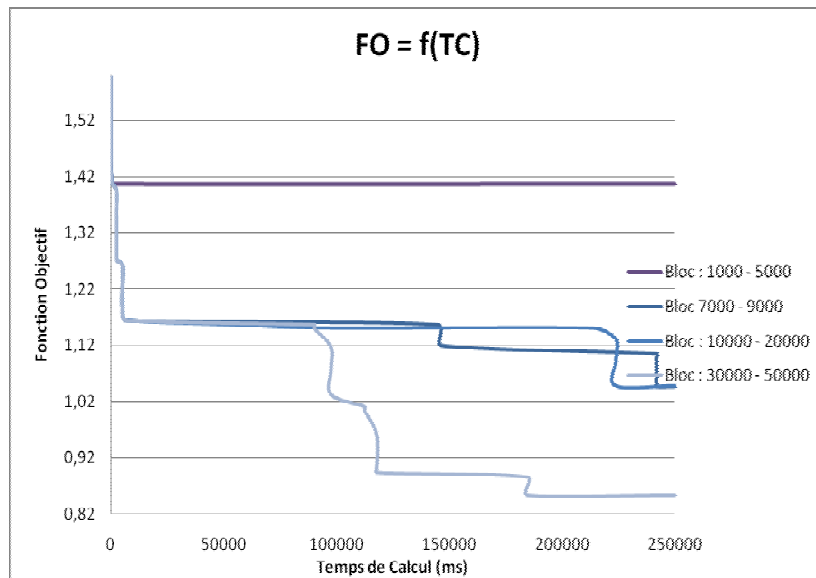


Figure 43 : Regroupement des courbes FO = f(TC) ayant la même allure ; FO : valeur de la Fonction Objectif ; TC : Temps de Calcul

Nous remarquons qu'en choisissant une valeur grande pour DT, nous arrivons à améliorer, relativement vite, la valeur FO. Ceci s'explique par le fait qu'une DT élevée empêche l'algorithme de recherche de revisiter des solutions déjà visitées.

Cette conclusion s'est confirmée après avoir tracé le nombre de solutions dans le front de Pareto en fonction de DT (Figure 44).

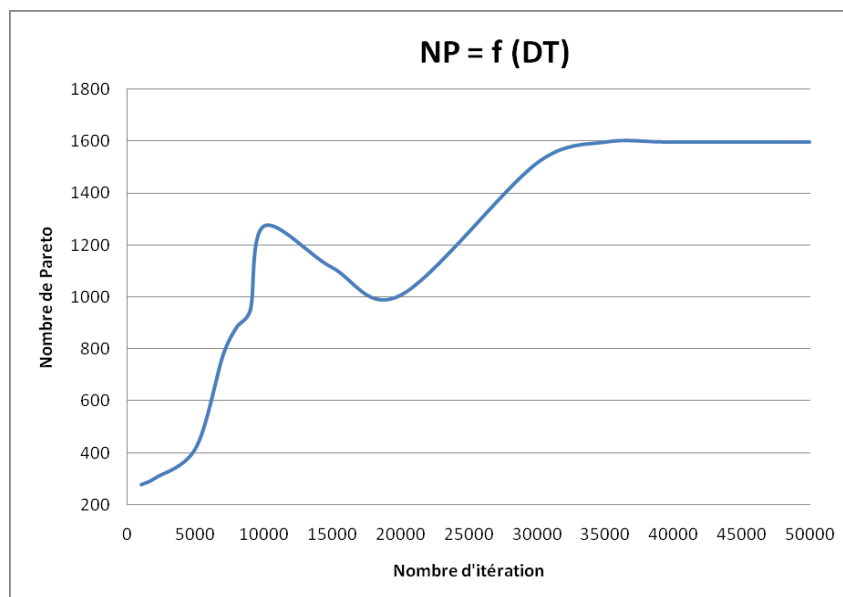


Figure 44 : NP=f(DT) ; NP : Nombre de solutions dans le front de Pareto ; DT : Durée Tabou

Dans la courbe de la Figure 44, on peut remarquer que jusqu'à $DT=10000$, l'algorithme ne fait qu'ajouter des solutions au front. La chute suivant le pic atteint à $DT=10000$ s'explique par le fait qu'il trouve une (ou des) solution(s) qui dominent un grand nombre de solutions du front, d'où la baisse de sa taille. Après avoir élagué un certain nombre de solutions dominées, l'algorithme recommence à rajouter des solutions pour $DT=20000$, pour se stabiliser vers $DT=35000$.

Quant au nombre de phases de diversification et d'intensification, nous avons préféré alterner entre ces deux mécanismes. En effet, à chaque fois que la recherche est bloquée par une solution entourée de mouvements interdits (tabous), un mécanisme d'intensification ou de diversification est lancé. Le nombre de cas similaires (blocages) dépend fortement de la dimension du problème. Dans notre cas, le risque est faible car nous avons 10 dimensions (variables de conception) et donc chaque solution possède 20 voisines...

Une fois, les paramètres de l'algorithme tabou fixés, nous nous sommes intéressés à l'étude de l'influence des variables de conception. En effet, nous avons fait varier les variables telles que la charge à transporter, et nous avons étudié l'influence de chaque variable sur les objectifs de conception. Par exemple, nous en avons déduit que l'aire atteignable et les sections des bras sont proportionnels à la charge à déplacer, contrairement à la flèche ; que la longueur des bras (somme des longueurs des deux bras) est constante, quel que soit la charge...

La recherche avec l'algorithme tabou avec pas adaptatifs et constitution du front de Pareto a été lancée pendant 250 secondes. Le front dégagé contient 1600 solutions que nous estimons être de bonne qualité². Néanmoins, en plus de la difficulté de réglage des paramètres, la fonction objectif utilisée est la simple pondération uniforme des objectifs (pas de critère préférentiel). La diversité des choix des poids des différents objectifs peut représenter une limite de cet algorithme (voir Section 1.1 du Chapitre 3).

C'est dans ce contexte que nous avons appliqué l'algorithme hybride de recherche, proposé dans le Chapitre 4 nous permettant d'alléger les réglages des paramètres et de ne pas imposer une agrégation a priori des objectifs (Chapitre 3).

² Dans notre étude, la qualité d'un front de Pareto est appréciée par rapport à son étendue dans l'espace des solutions et aux valeurs des objectifs atteints.

B. Méthode de recherche hybride

Afin de choisir les paramètres de l'algorithme hybride (HS : Hybrid Search), plusieurs simulations ont été effectuées tout en faisant varier le nombre de phases d'intensification, et la manière de gérer la liste de Pareto.

Nombre de phases d'intensification

Pour cela, nous avons fait varier le nombre d'intensifications entre 1 et 5 phases. Le Tableau 5 illustre, pour chaque simulation « i » (i = nombre de phases d'intensification), le nombre de solutions retenues dans le front final ainsi que le temps de calcul. Sachant que la première phase (recherche exhaustive avec pas maximaux) dure 520s et 3099 solutions sont retenues dans le front.

Nb discrétisations	Nb de solutions du front	Durée totale d'exécution
1	1656	20833
2	1605	1783
3	1324	1429
4	1282	1448
5	841	1035

Tableau 5 : Nombre de solutions du front final et durée totale d'exécution en fonction du nombre de phases d'intensifications

Nous avons, en premier temps, comparé les valeurs minimales et maximales des paramètres et des critères obtenues par les cinq algorithmes ; ces valeurs sont quasiment les mêmes à deux exceptions près. Nous supposons alors que les zones de l'espace des solutions visitées par les différents algorithmes sont les mêmes et que, tous les algorithmes arrivent à trouver les solutions « optimales » en considérant les critères un à un (les optima locaux).

En un deuxième temps, nous avons comparé deux à deux l'ensemble de ses fronts. En effet, ces comparaisons ont été effectuées par l'analyse visuelle des fronts à l'aide des cartes de Kohonen (outil de visualisation).

Nous avons ainsi conclu que la version 2 (avec 2 discrétisations) représente le meilleur rapport « temps de recherche/qualité du front obtenu ». En effet, il est beaucoup plus rapide que l'algorithme 1 et arrive à trouver de nombreuses solutions qui ne sont pas retrouvées par les algorithmes 3, 4, et 5.

Il faut noter que le nombre de solutions dans le front final ne peut pas refléter la qualité de ce front. En effet, l'algorithme de recherche pourrait trouver un nombre élevé de solutions non dominées mais très proches d'un point de vue des valeurs des objectifs.

Une seule solution rajoutée peut entraîner la suppression d'une grande partie de ce front. De ce fait, le front après la découverte de cette solution peut devenir :

- plus efficace puisqu'on a supprimé un nombre de solutions dominées,
- moins riche puisque cet ajout a supprimé toute une partie de l'espace de recherche qui aurait pu être interprétée par les concepteurs.

Gestion de la liste Pareto

Après avoir fixé le nombre de phases d'intensification, nous avons essayé plusieurs versions de l'algorithme avec différents modes de gestion de la liste de Pareto, comme nous l'avons décrit au Chapitre 4.

Dans la première version, à chaque fois qu'une solution est rajoutée, elle est comparée à toutes les solutions déjà contenues dans le front. Les solutions dominées sont automatiquement supprimées.

Dans la deuxième version, lorsqu'une solution est ajoutée au front de Pareto, celui-ci n'est pas parcouru directement afin de supprimer les solutions dominées par la dernière. Le front est mis à jour à la fin de chaque phase d'intensification, et ce pour pouvoir explorer les voisinages de toutes les solutions obtenues lors de la phase précédente.

Comme attendu, la deuxième version de l'algorithme met plus de temps (entre 5 et 10 fois plus) pour retrouver l'ensemble Pareto final. Ceci est dû à la taille du front qui augmente au cours d'une phase, ce qui implique plus de voisinages à explorer (20 explorations de

voisinage supplémentaires par solution du front, puisque nous avons 10 variables et deux mouvements possibles par variable). Mais aussi, la mise à jour du front est beaucoup plus longue à la fin de la phase, car les solutions du front doivent être comparées deux à deux. En effet, pour un même nombre n de solutions dans le front, comparer la dernière aux autres nécessiterait $(n-1)$ opérations de comparaison ; alors que comparer toutes les solutions deux à deux nécessiterait $n^2/2$ opérations.

Quant au positionnement de la solution à rajouter, les deux possibilités énoncées dans le Chapitre 4 ont été testées. Dans la première possibilité (V1) le voisinage de la solution trouvée est immédiatement exploré et dans la deuxième (V2) la solution est rajoutée à la fin du front afin d'évaluer son voisinage ultérieurement.

Nous avons comparé les fronts obtenus avec les deux versions de l'algorithme V1 et V2. Nous avons constaté que les solutions trouvées par V1 dominent 0.56% des solutions trouvées de V2 (9 sur 1605) ; et que les solutions trouvées par V2 dominent 33.95% des solutions trouvées par V1 (2009 sur 5917). Quant au nombre de solutions non dominées, il est de 3908 pour V1 et 1596 pour V2 pour des temps de calcul respectifs de 432s pour V1 et 1783s pour V2.

Partant de ces résultats, nous opterons pour la deuxième possibilité (V2) qui permet d'explorer d'une façon identique, le plus possible de zones de l'espace de solutions ; et par suite de ne pas consacrer la majeure partie du temps imparti à la fouille d'un nombre réduit de zones.

Simulation finale

Dans la version retenue, la phase de diversification (ou de recherche exhaustive) dure 480s et dégage 776 solutions non-dominées. Ensuite, les pas sont progressivement réduits et les voisinages des solutions non dominées sont explorés. Cette phase d'intensification dure 1814s et le front de Pareto final contient 7801 solutions. Le temps de recherche alloué (3600s) n'est pas atteint sachant que le temps total d'exécution de l'algorithme est 2295s.

L'utilisation des cartes de Kohonen comme outil de visualisation du front de Pareto nous permet de construire la carte des Figure 45 et Figure 46.

Une carte de Kohonen est un ensemble de nœuds interconnectés, chaque nœud est un représentant d'un ensemble de solutions ayant des caractéristiques proches (au sens de la distance euclidienne). Un nœud est placé dans un espace multidimensionnel, les dimensions, dans notre cas, correspondent à l'ensemble des variables et des objectifs de conception. La

carte respecte les propriétés de densité et de voisinage de la base de donnée dans laquelle elle est déployée.

Ayant une topologie 2D, la carte peut être visualisée en deux dimensions afin d'être visualisée. Les Figure 45 et Figure 46 montrent pour chaque dimension (variable ou objectif), la valeur de chaque nœud.

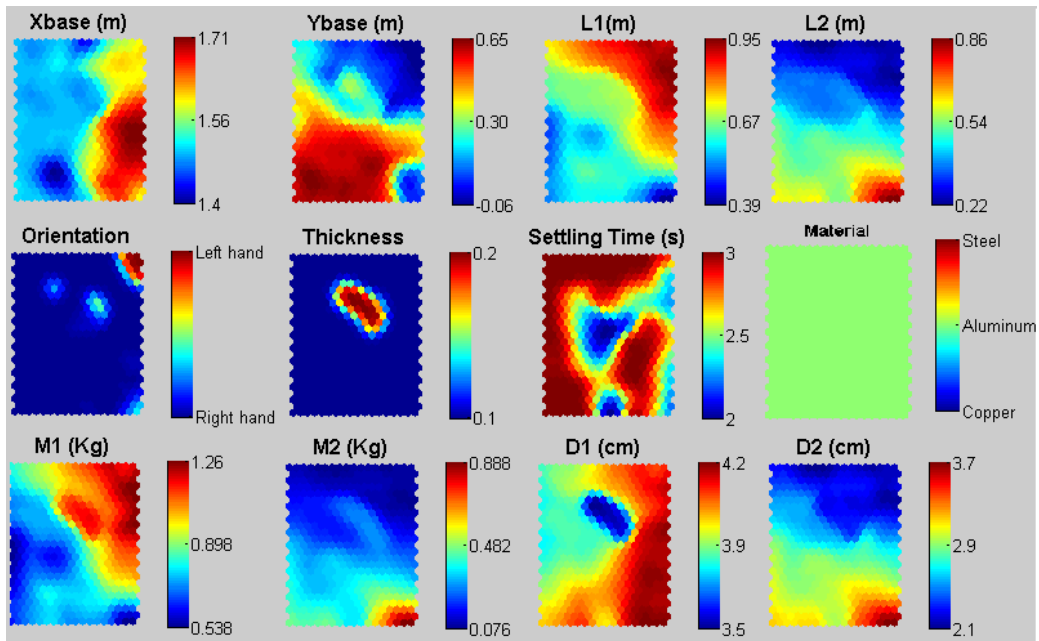


Figure 45 : Variables de conception du robot à deux degrés de liberté

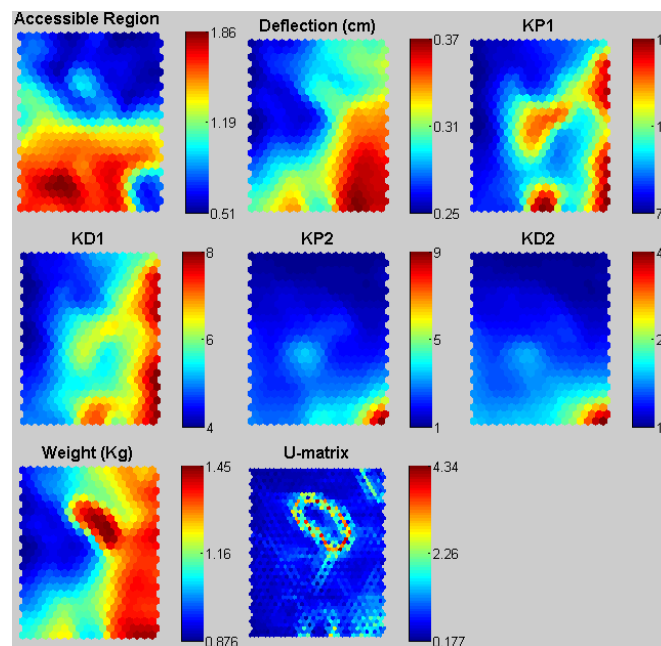


Figure 46 : Objectifs de conception du robot à deux degrés de liberté

Les cartes peuvent fournir aux concepteurs un nombre important d'informations utiles. Nous citons à titre d'exemple :

- L'aluminium est retenu pour toutes les solutions,
- Des intervalles plus petits pour les variables de conception peuvent être choisis (en comparaison avec les intervalles initiaux),
- La somme de la longueur des bras devrait être d'environ 1,2 m,
- Utilisation de bras minces ($T_h = 0.1$ ou $T_h = 0.2$),
- Une orientation des bras à droite est plus appropriée dans la plupart du temps³,
- Pour chaque axe du robot, les gains proportionnels et dérivatifs (KP et KD) ont des valeurs proportionnelles (ce qui peut être vérifié par le calcul).
- La dernière carte, U-matrix, indique la distance euclidienne entre les nœuds voisins. Cela reflète la disposition topographique des solutions dans l'espace des données. Dans cet exemple, nous pouvons voir un cluster qui indique qu'un ensemble de solutions est relativement loin des autres. Le même cluster peut être retrouvé dans les cartes de l'épaisseur et du poids. Ceci indique que l'utilisation d'une valeur de 0,2 pour l'épaisseur (rapport entre épaisseur et diamètre du bras) augmente le poids robot mais les solutions restent acceptables et intéressantes...

Chaque nœud de la carte représente un ensemble de solutions proches du front de Pareto. La Figure 47 montre un exemple de solution à travers les valeurs de certaines variables et objectifs de conception.

³ L'exemple traité présentant une symétrie verticale (due à la forme du petit M), les solutions où la base du robot se situe sur l'axe de symétrie vertical sont indépendantes de l'orientation des bras. Nous avons choisi de ne retenir que la première possibilité rencontrée par la recherche.

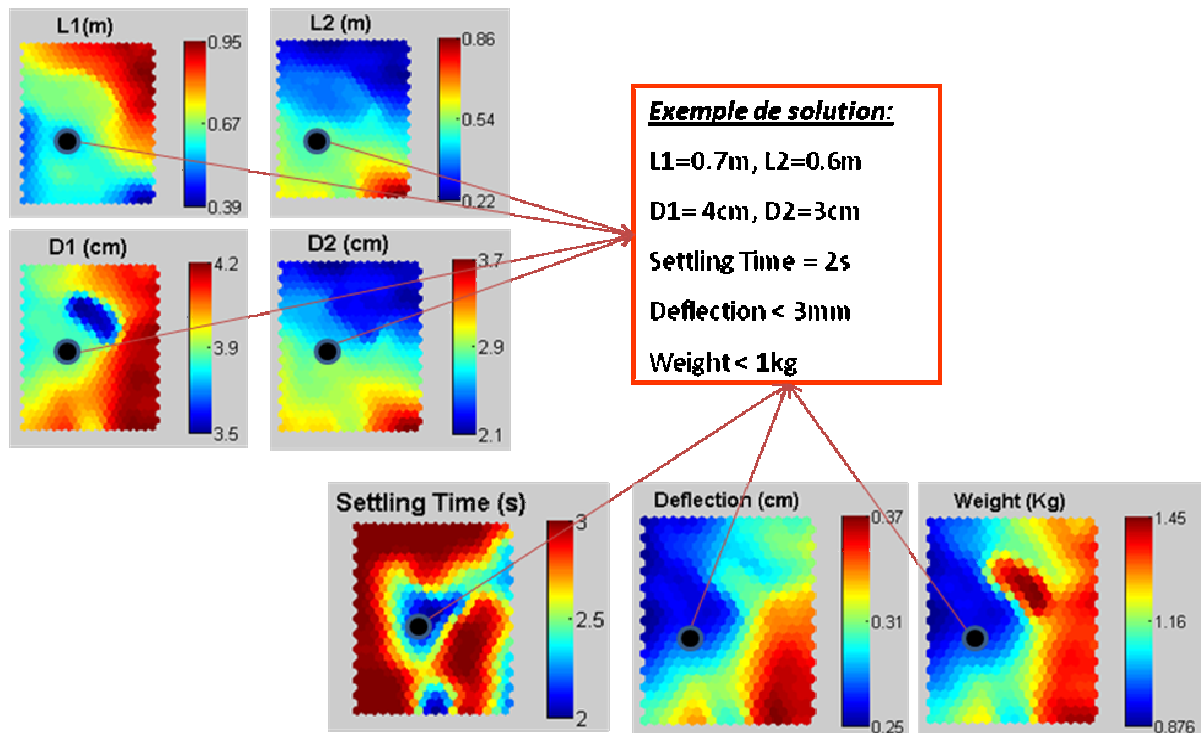


Figure 47 : Exemple d'une solution de la carte de Kohonen représentant un ensemble de solutions semblables du front de Pareto

Pour vérifier la performance de notre algorithme (temps d'exécution et qualité des fronts obtenus), nous avons lancé un algorithme exhaustif qui visite toutes les solutions possibles et retient l'ensemble de solutions non dominées. Pour gagner en temps d'exécution, nous avons utilisé des pas de 2 cm pour les quatre paramètres continus (un pas de 1 cm est utilisé pour les simulations habituelles), et nous avons choisi l'aluminium seulement comme matériau (retenu pour toutes les solutions avec l'algorithme HS). Cette manipulation divise le temps de calcul par environ 48 fois. La recherche exhaustive dure plus de 108 heures.

Afin de pouvoir comparer les fronts obtenus, nous les avons projetés suivant les valeurs de quelques objectifs. Les Figure et Figure 48 montrent les fronts obtenus avec les algorithmes HS (Hybrid Search), ES (recherche exhaustive), ainsi que le front obtenu après la phase de diversification de HS : SNE. Ces exemples représentent des projections sur les axes « poids total du robot » (Weight), « flèche » (Deflection), et « Gain Proportionnel de l'axe 2 » (KP₂).

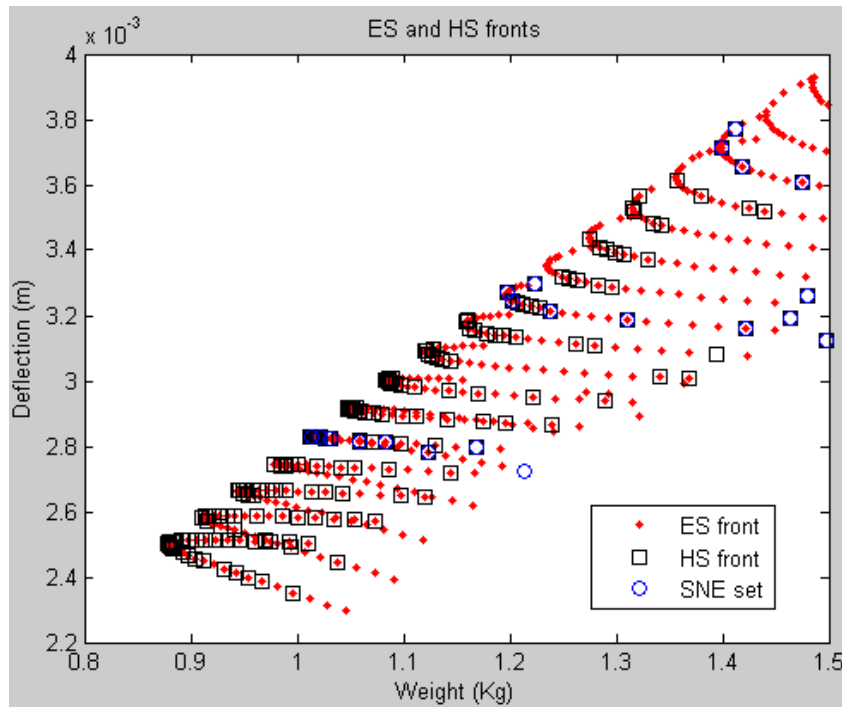


Figure 48 : Projection sur les axes « poids » et « flèche » des fronts obtenus avec les algorithmes HS (Hybrid Search), ES (recherche exhaustive), ainsi que le front obtenu après la phase de diversification de HS : SNE

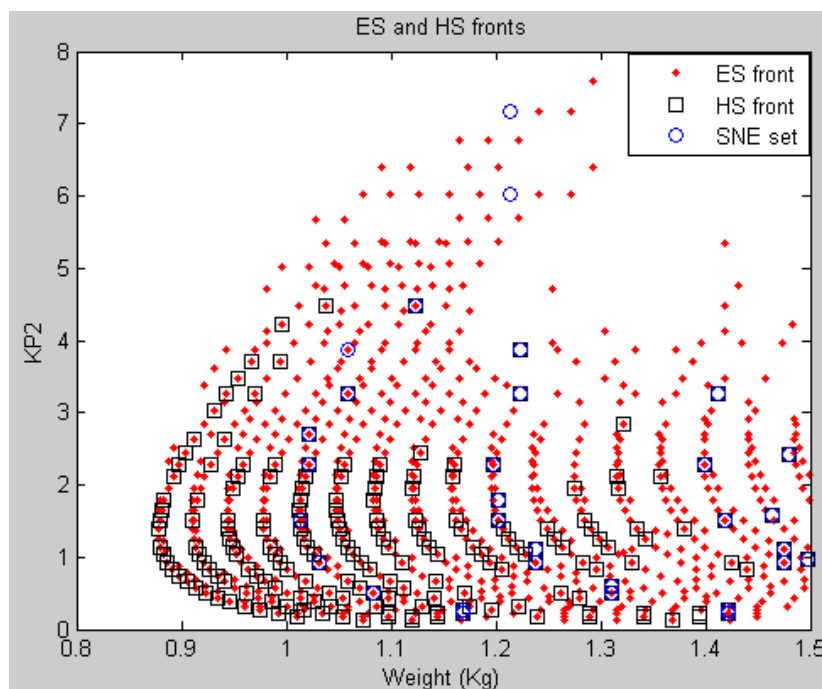


Figure 48 : Projection sur les axes « poids » et « gain proportionnel de l'axe 2 » des fronts obtenus avec les algorithmes HS (Hybrid Search), ES (recherche exhaustive), ainsi que le front obtenu après la phase de diversification de HS : SNE

L'ensemble des objectifs choisis étant à minimiser, nous remarquons que l'algorithme HS dégage les solutions optimisant chacun des objectifs. Nous remarquons également que certaines solutions retrouvées par la phase de diversification font partie du front final. Certaines d'entre elles sont des solutions optimales, d'autres ne le sont pas...

Dans l'exemple étudié, la recherche tabou est plus rapide que la recherche exhaustive, mais constitue un front de Pareto moins représentatif de l'espace des solutions. En fait, l'algorithme HS concentre le processus de recherche dans certaines régions de l'espace global.

L'algorithme HS explore une plus grande partie de l'espace de recherche et arrive à trouver des solutions de bonne qualité quand elles sont comparées à celles trouvées par la recherche exacte.

3. Bilan

Algorithme	Temps de calcul (s)	Nombre de solutions	Qualité du front	Caractéristiques de l'algorithme
Tabou	250	1600	Très bonnes solutions mais mauvaise couverture de l'espace des variables et des solutions	<ul style="list-style-type: none"> - Difficile à paramétrer - Nécessite une pondération des objectifs - Mauvaise exploration de l'espace de recherche + Rapidité d'exécution + Solutions individuelles de très bonne qualité
Hybride	2295	7801	Bonne qualité des solutions et très bonne couverture de l'espace des solutions par le front obtenu	<ul style="list-style-type: none"> + Facile à paramétrer + Ne nécessite pas de pondération des objectifs + Bonne exploration rapide de l'espace de recherche + Plus rapide que la recherche exhaustive - Moins rapide que le tabou + Solutions individuelles de bonne qualité
Exhaustif	> 400000	5209	Optimale	Optimale

Tableau 6 : Bilan des trois méthodes de recherche

Afin de compléter l'apport des outils de discrimination, nous allons les détailler sur le deuxième exemple de conception de robot désempileur pour textile.

II. Application des outils de discrimination en conception d'un robot déempileur pour l'industrie textile

Après avoir donné des résultats intéressants sur un exemple simple de choix de moteur/réducteur en fonction de la trajectoire [Mouelhi'07], nous avons appliqué les méthodes de discrimination sur un exemple plus concret [Mouelhi'08] : un robot déempileur de tissus composé de trois axes couplés.

1. Modèle du robot

La conception d'un robot pour l'industrie de l'habillement entre dans le contexte d'une automatisation de ses processus. Un des problèmes importants de cette automatisation est la saisie des pièces textiles. En effet, la géométrie de ces pièces et leur matériau varient très souvent, à cause des effets de mode, par exemple...

La Figure 49 illustre une partie du système imaginé, conçu et réalisé dans cadre d'un projet Européen [Rakotoarisoa'93].

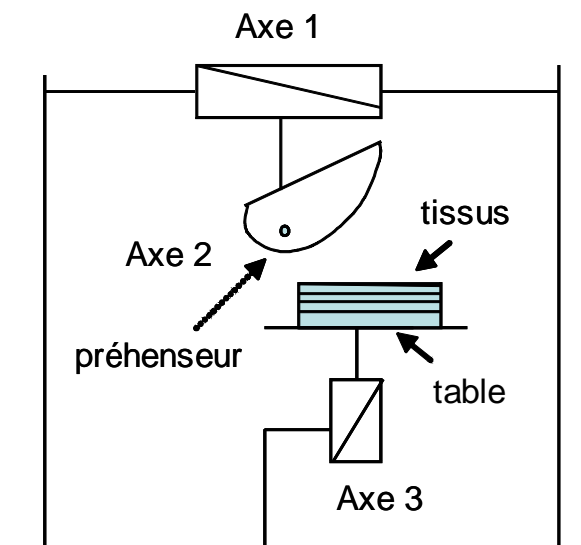


Figure 49 : Robot déempileur pour l'industrie textile [Rakotoarisoa'93]

La phase critique de ce système est la séparation du tissu supérieur de la pile. Les autres phases sont celles de la localisation du tissu sur le préhenseur et de la dépose avant assemblage.

Le système de la Figure 49 est composé de trois axes. Le premier est un axe de translation horizontal du chariot transportant le préhenseur. Le deuxième est son axe de rotation. Le

troisième est l'axe de translation verticale de la table de préhension. Les mouvements des trois axes sont coordonnés afin d'assurer un contact entre le préhenseur et le tissu quelque soit la hauteur de la pile.

La trajectoire du préhenseur lors de la phase de préhension est représentée sur la Figure . La forme spirale du préhenseur a été choisie pour assurer un roulement sans glissement au point M de contact entre le préhenseur et le tissu.

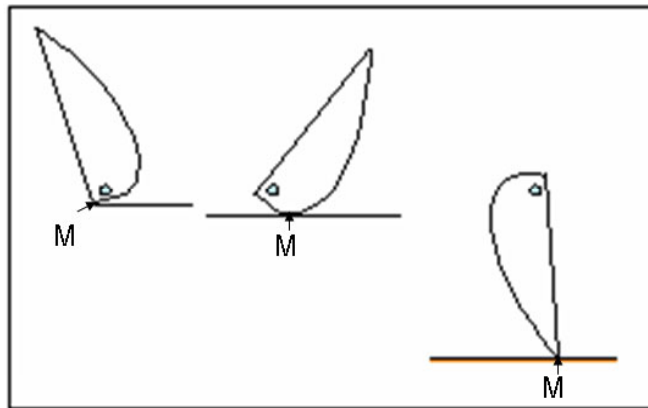


Figure 51 : Mouvement du préhenseur lors de la phase de séparation

Problématique

On donne :

- Les exigences du client à satisfaire : la longueur des pièces à saisir et la cadence et les caractéristiques du mouvement à effectuer.
- Des paramètres de conception influant sur le choix des moteurs : les caractéristiques du préhenseur (dimensions et masses des axes).
- Les contraintes cinématiques (vitesses et accélérations maximales...)

On cherche à minimiser :

- Le coût de la motorisation,
- La consommation énergétique,
- Le poids du système de préhension,
- ...

Notre objectif est de mettre en évidence les domaines d'utilisation (en fonction des spécifications techniques) des moteurs/réducteurs qui sont adaptés à la tâche et qui respectent les contraintes du problème.

Dans ce qui suit, nous donnons un exemple de trajectoire (cinématique) obtenu avec les valeurs des exigences techniques suivantes :

- Longueur du tissu (L) = 1 m,
- Cadence (Cad) = 3 s, et
- Accélération maximale du point M (A_{\max}) = $1,5 \text{ m/s}^2$.

Les courbes de la Figure 50 représentent une trajectoire cinématique (position, vitesse et accélération du point M) en fonction du temps.

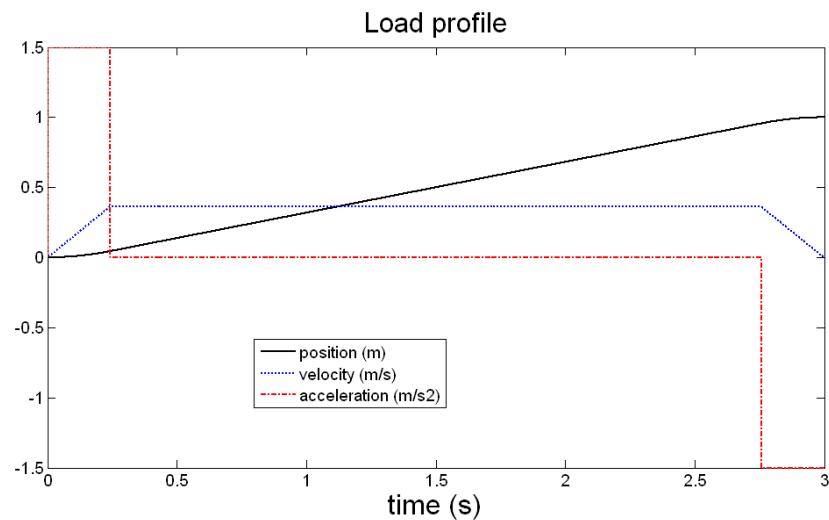


Figure 50 : Trajectoire cinématique du point de contact

Les Figure 51 à 55 représentent les trajectoires cinématiques des trois axes. Ces courbes sont établies à partir de la trajectoire cinématique du point M et des caractéristiques géométriques du robot.

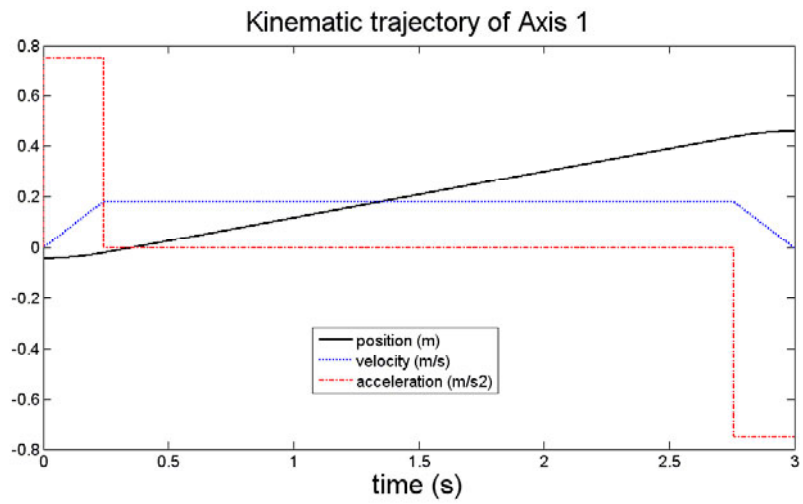


Figure 51 : Trajectoire cinématique du 1er axe

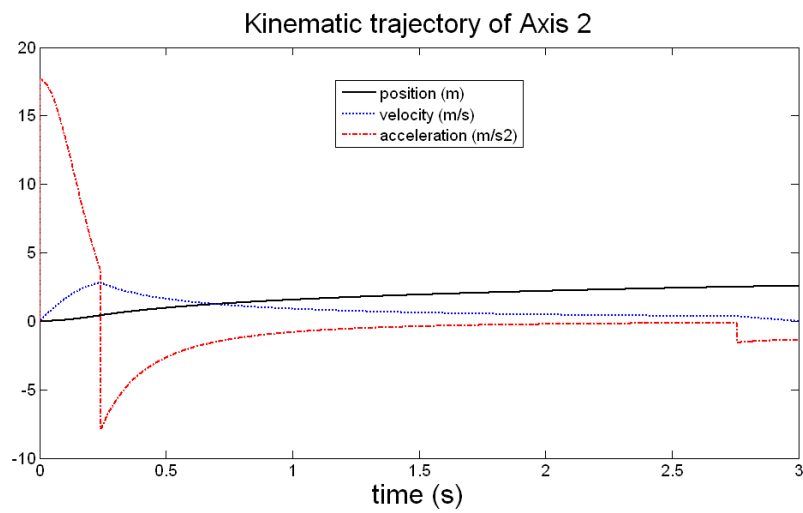


Figure 52 : Trajectoire cinématique du 2ème axe

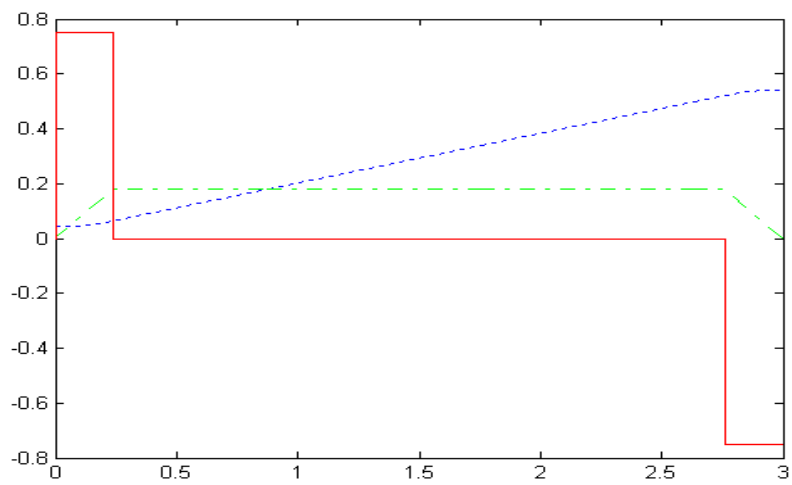


Figure 53 : Trajectoire cinématique du 3ème axe

2. Simulations

Les simulations effectuées comportent deux étapes. Lors de la première, qui est une étape de sélection, nous faisons varier les variables de conception du problème dans des intervalles spécifiés par les exigences techniques. Ensuite, et pour chaque configuration de ces variables, nous déterminons, grâce à une méthode de sélection, les meilleurs composants à utiliser (c.a.d. nous retenons la meilleure combinaison de moteur/réducteur). A la fin de cette première étape, nous avons construit une base de données constituée par un ensemble de solutions optimales⁴. Chaque solution, élément de la base, est appelé prototype, et est représentée par un vecteur constitué des variables de conception et de la valeur du critère optimisé. On mémorise aussi le couple moteur/réducteur solution.

La seconde étape consiste à un apprentissage mené sur l'ensemble des données de cette base. Un premier type d'apprentissage construit des cartes de Kohonen et permet de mettre en évidence le domaine de validité (ou d'utilisation) des différents composants (moteurs et/ou réducteurs) en fonction des valeurs des variables de conception.

La méthode IGNG (définie au Chapitre 4), permet de proposer des composants à utiliser, pour des valeurs de spécifications non rencontrées lors du processus d'apprentissage, ou même pour des spécifications incomplètes. Les résultats obtenus avec les deux méthodes d'apprentissage sont comparées sur une base de test.

Dans la section suivante, nous présentons les conditions des simulations effectuées, ainsi que les résultats obtenus.

A. Constitution de la base d'apprentissage

Plusieurs méthodes de sélection de moteurs et de réducteurs existent [Cusimano'07], [Roos'06], et [Dequidt'99]. Le principe de ces méthodes est de choisir le meilleur ensemble moteur/réducteur pouvant déplacer une charge donnée selon la trajectoire spécifiée. Nous avons opté pour la méthode de Roos [Roos'06] fournissant rapidement un pré-dimensionnement des moteurs/réducteurs. Notre méthode d'analyse proposée est indépendante de la méthode de sélection choisie.

⁴ Optimales selon la méthode de sélection utilisée.

Application de la méthode de sélection

L'application de la méthode de sélection de Roos sur le cas du robot déempileur pour textile consiste, pour chaque combinaison des spécifications techniques de départ, à :

- Calculer la trajectoire cinématique du point M.
- En déduire la trajectoire cinématique des axes (modèle donné en Annexe C).
- Déterminer les moteurs les plus performants et sans réducteur pouvant effectuer la tâche sur chacun des axes et satisfaisant les exigences de vitesse, de couple moyen et de couple maximum à fournir au cours du mouvement (pour estimé le couple moteur à fournir, on injecte les solutions cinématiques dans le modèle dynamique du robot donné en Annexe C).
- Supprimer, pour chaque axe, les moteurs plus puissants que ceux déterminés à l'étape précédente.
- Déterminer la liste des réducteurs compatibles avec des moteurs moins puissants de la gamme pour chaque axe.
- Calculer les critères de sélection parmi les suivants : puissance maximale, puissance moyenne, couple moyen, couple de pointe, couple résiduel, consommation d'énergie, poids...
- Garder les combinaisons moteur/réducteur qui optimisent les critères sélectionnés.

Dans cette étude, nous **avons choisi de minimiser le pic de puissance (ou la puissance maximale)** car ce critère détermine la puissance et le coût de l'électronique.

Les variables de conception sont : Accélération maximale du point M (A_{max}), Cadence (Cad) et Masse du préhenseur (m). Nous avons discrétisé ces variables de la manière suivante :

Variable de conception	Min	Max	Pas de discrétisation
Accélération maximale du point M (m/s^2)	1	5	0.2
Cadence (s)	2	10	1
Masse du préhenseur (Kg)	5	25	2

Tableau 7 Discrétisation des variables de conception

En utilisant cette discrétisation, nous avons obtenu près de 8000 prototypes (combinaisons de valeurs des spécifications techniques).

Les moteurs et réducteurs sont choisis sur catalogue, parmi :

- La famille de moteurs brushless KOLLMORGEN de type AKM4 (12 moteurs équipés de résolveurs) pouvant être associés à des réducteurs universels à 2 trains de type PLE80 (12 réducteurs avec des facteurs de réduction allant de 1/3 jusqu'à 1/320) sur les axes 1 et 3.
- La famille de moteurs brushless KOLLMORGEN de type AKM5 (13 moteurs équipés de résolveurs) pouvant être associés à des réducteurs universels à 2 trains de type PLE120 (12 réducteurs avec des facteurs de réduction allant de 1/3 jusqu'à 1/320) sur l'axe 2.

Nous présentons dans les Tables 9 et 10 la liste des moteurs et réducteurs du catalogue envisagés sur l'axe 1.

Pour chaque configuration des variables de conception on applique la méthode de Roos et on retient le moteur/réducteur qui minimise le pic de puissance.

Nom	Label	T_m (Nm)	T_{peak} (Nm)	J_m (kg/cm ²)	I_{peak} (A)	V_{max}	P (Kg)	Res (Ω)	K_r (Nm/A)
"AKM41C"	A	1.88	6.12	0.81	5.80	1200	2.44	21.7	1.34
"AKM41E"	B	1.82	6.28	0.81	11.4	3000	2.44	5.70	0.71
"AKM41H"	C	1.62	6.36	0.81	22.4	6000	2.44	1.51	0.37
"AKM42E"	D	3.12	11.3	1.50	11.0	1800	3.39	7.22	1.26
"AKM42G"	E	2.90	11.5	1.50	19.2	3500	3.39	2.38	0.74
"AKM42J"	F	2.38	11.6	1.50	33.7	6000	3.39	0.80	0.43
"AKM43E"	G	4.24	15.9	2.10	11.0	1500	4.35	8.04	1.72

Tableau 8 : Liste des moteurs retenus sur le 1er axe

avec, T_m : couple moyen, T_{peak} : couple maximum, J_m : moment d'inertie moteur, I_{peak} Courant maxi, V_{max} : vitesse maximum, P : poids du moteur, Res : résistance inducteur, K_r : constante de couple, Label : repérage du moteur.

Sachant que les réducteurs compatibles sont ceux de la famille "PLE80", la liste des réducteurs retenus sur l'axe 1 est :

Nom	Label	T_r (Nm)	T_{max} (Nm)	N_{gear}	J_g (kg/cm ²)	P (Kg)	Ren (%)
"PLE80-3"	K	85	136	1/3	0.77	2.1	96
"PLE80-8"	L	50	80	1/8	0.39	2.1	96
"PLE80-15"	M	110	176	1/15	0.71	2.6	94

Tableau 9 : Liste des réducteurs retenus sur le 1er axe

avec, T_r : couple moyen transmis en sortie, T_{max} : couple maximum, N_{gear} : rapport de réduction, J_g : moment d'inertie du réducteur, P : poids, Ren : rendement, Label : repérage du réducteur.

B. Exploitation des résultats

Les domaines des variables de conception étant définis, nous pouvons visualiser, dans ce cas particulier, dans un espace à 3 dimensions les points représentant les moteurs utilisés sur le axe 1 du préhenseur en fonction des variables de conception (A_{max} , Cad, et m). Nous avons ainsi détecté (voir Figure 54) un ensemble de regroupements représentant le domaine de validité de chaque moteur sur cet axe.

Dans la Figure 54, les intervalles des variables explorés sont : A_{max} entre 1 et 5 m/s², Cad (temps de préhension) entre 2 et 10 s, et m entre 5 et 30 Kg.

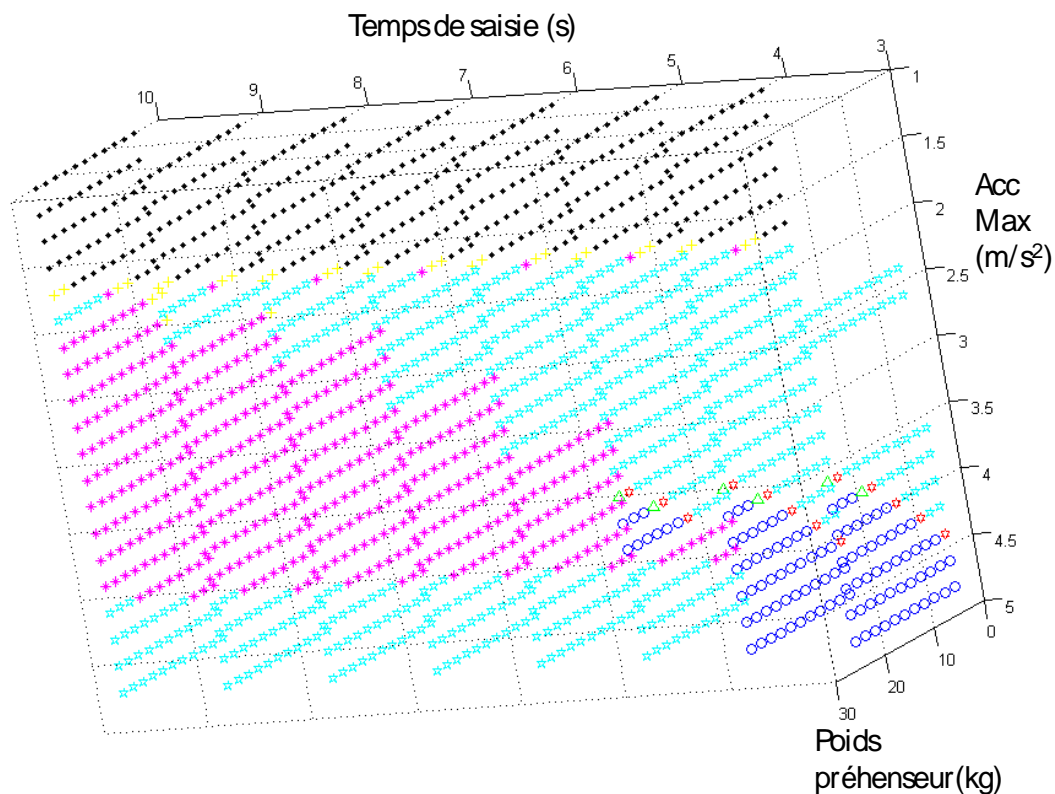


Figure 54 : Clusters des moteurs en 3D

Les points correspondent aux types de moteurs retenus (du plus petit au plus grand) : 'A' en Noir, 'B' en Jaune, 'C' en Magenta, 'D' en cyan, 'E' en rouge, 'F' en vert, et 'G' en bleu.

Notons que cette visualisation n'est pas possible dès que le nombre de variables de conception dépasse 3, d'où l'intérêt des cartes de Kohonen présentées ci-dessous.

C. Discrimination avec des méthodes d'apprentissage

Soit une base de données de prototypes décrits chacun par un vecteur dont les composantes sont l'accélération maximale du point M, la cadence, la masse du préhenseur et le pic de puissance. Le processus d'apprentissage construit une carte qui sert au choix du moteur et au choix du réducteur. La carte est composée de représentants de l'espace de données placés dans un espace de dimension 2 et étiquetés suivant leurs types (types des moteurs ou types des réducteurs). Le représentant à l'étiquette du moteur (ou du réducteur) le plus souvent présent parmi les prototypes qui ont comme plus proche voisin ce représentant. Donc, de la carte issue de l'apprentissage on déduit une carte portant les étiquettes du moteur et une seconde portant les étiquettes du réducteur.

Avant apprentissage, la base de données est normalisée en centrant les données de chaque dimension autour du 0 (origine du repère) et en assurant un écart type unitaire. On normalise pour les raisons suivantes :

- Assurer une plus grande homogénéité des représentations : La normalisation permet une étendue de mesure identique pour toutes les caractéristiques étudiées. En effet, dans notre exemple, l'accélération ($m.s^{-2}$), la cadence (s), et la masse (Kg) ont, avant normalisation, des ordres de grandeur différents.
- Donner une influence identique à toutes les dimensions : Dans les algorithmes d'apprentissage, des distances euclidiennes entre prototypes et représentants sont calculées. Pour que chaque dimension ait la même influence, il est nécessaire d'avoir une même échelle (un même ordre de grandeur) pour ces différentes dimensions.

a) Apprentissage avec les cartes de Kohonen

Un apprentissage mené sur la base de données en utilisant la méthode des cartes de Kohonen, fournit à la fois, un bon outil de visualisation, et un outil d'interrogation. La simulation a été effectuée sous l'environnement Matlab et avec la boîte à outils SOM Toolbox [Vesanto'99]. Nous présentons, dans ce qui suit, les illustrations les plus intéressantes pour notre utilisation.

Outil de visualisation

Nous avons obtenu deux cartes représentant le domaine de validité de chaque composant (moteur et réducteur associé). Ces cartes sont composées de près de 600 représentants (soit 7.5% du nombre de prototypes dans la base de départ).

Figure 55, nous pouvons observer quelques ‘clusters’ de moteurs du même type. Les lettres (de 'A' à 'G') correspondent à la classe du prototype (i.e. la référence du composant dans la Tableau 8) et les '0' correspondent aux cas sans solution.

Figure , nous pouvons observer quelles sont les valeurs des variables de conception de chaque nœud (représentant des prototypes) de la carte : les prototypes pour lesquels ce représentant est le plus proche ont des valeurs voisines de variables de conception.

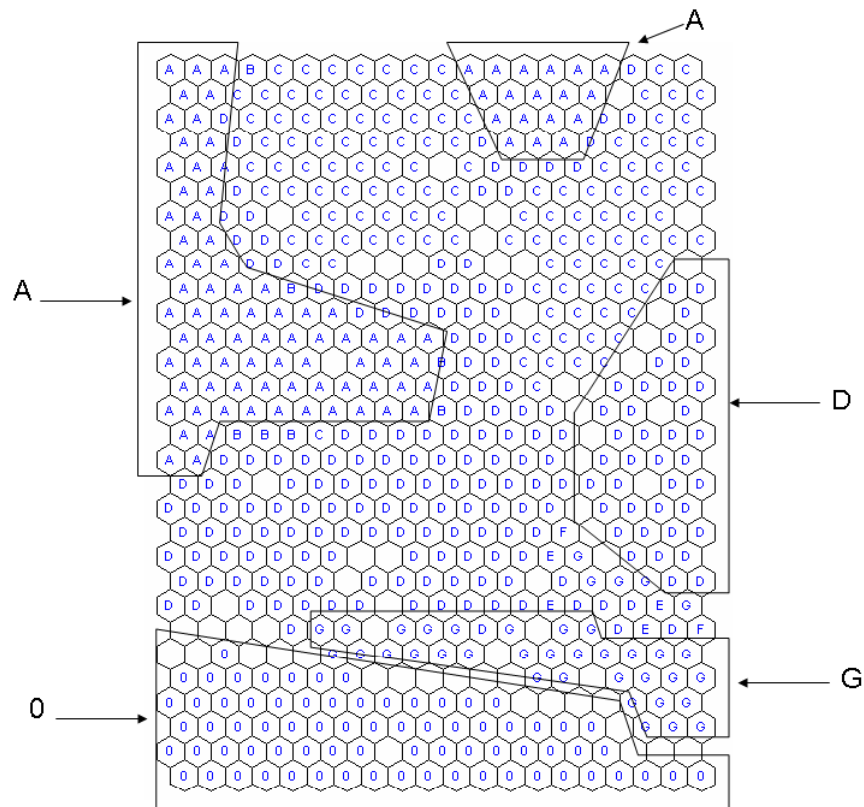


Figure 55 : Domaines de validité (clusters) des moteurs

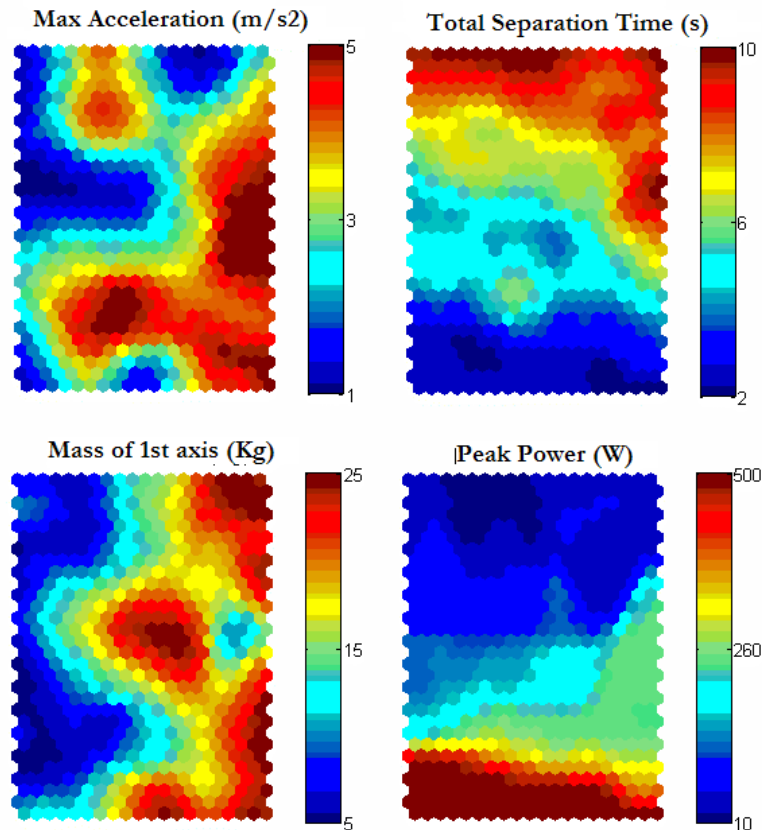


Figure 56 : Valeurs des variables de conception pour chaque nœud de la carte

En analysant les différentes composantes des Figure 55 et Figure , l'utilisateur peut obtenir certaines informations intéressantes, comme :

- Pas de solution lorsque la cadence est inférieure à 2,5 s.
- Le moteur "AKM41C" ('A') est retenu lorsque l'accélération maximale du point M est inférieure ou égale à 1,5 m.s⁻².
- Le moteur "AKM42E" ('D') est retenu lorsque l'accélération maximale du point M est entre 3 et 5 m.s⁻² et la cadence entre 3 et 8 s.

Plusieurs autres types de cartes peuvent être trouvées dans [Mouelhi'07], [Mouelhi'08], et [Vesanto'99].

En plus d'un outil de visualisation, les cartes de Kohonen offrent aux concepteurs un outil d'interrogation. Cet outil sert à déterminer le composant (par exemple le moteur sur l'axe 1) le mieux adapté à des spécifications techniques incomplètes ou imprécises.

Interrogation

En des termes plus techniques, le réseau de neurones, formant la carte de Kohonen, peut être considéré comme une mémoire auto-associative. Cette mémoire a la propriété de généraliser ou bien de recomposer une information manquante, même dans le cas de données imprécises.

En effet, une fois la carte générée, les concepteurs peuvent déterminer la position sur la carte d'une configuration (combinaison de variables de conceptions). Cela se fait par la recherche du représentant le plus proche (en distance euclidienne) de cette configuration : le BMU pour « Best Matching Unit ».

Dans ce qui suit, nous illustrons trois exemples d'utilisation des cartes de Kohonen comme mémoire auto-associative.

- Si, par exemple, le BMU d'une solution se trouve à la frontière d'un cluster, alors le concepteur peut conclure que la relaxation d'une (ou de quelques) contrainte(s) modifie le choix du moteur ou du réducteur.
- Il est également possible de déterminer le type d'une solution non utilisée lors du processus d'apprentissage. Pour vérifier cette propriété, la base de données a été divisée aléatoirement en une base d'apprentissage (80% de la base initiale) et une base de tests (l'ensemble restant de prototypes). Les
- Tableau 10 et 12 illustrent les taux d'erreurs obtenus lors de ces interrogations : une erreur est caractérisée par le fait que le moteur optimum n'est pas trouvé parmi les étiquettes du premier au « Nb BMU » plus proches représentant de la configuration présentée. Pour chaque interrogation, la base est décomposée aléatoirement en une base de test et une base d'apprentissage, le processus complet de décomposition, apprentissage et interrogations a été lancé 100 fois. Chaque valeur des tableaux 11 et 12 représente une moyenne des taux d'erreur obtenus après 100 lancements différents.

Nb BMU	Taux d'erreur moyen
1	4.95%
2	2.60%
3	1.84%

Tableau 10 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode de Kohonen avec des prototypes de la base de tests

- En cas d'absence d'une (ou plusieurs) valeur(s) des composantes d'entrée, il peut s'avérer utile de rechercher le représentant le plus proche en faisant abstraction de la donnée manquante (dans le cas présent, la valeur du pic de puissance). Ceci peut servir, par exemple, à prédire la classe du moteur ou bien du réducteur que l'on peut choisir.

Nb BMU	Taux d'erreur moyen
1	12.63%
2	5.90%
3	3.43%

Tableau 11 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode Kohonen avec des spécifications techniques incomplètes (sans la valeur du pic de puissance) de la base de tests

Les taux moyens d'erreurs lors des interrogations des cartes de Kohonen ont été jugés assez élevés (plus de 5%). Ceci s'explique par le fait que la topologie des cartes et le voisinage sont prédéfinis et fixes. Afin de réduire ces taux d'erreurs, mais aussi pour avoir la possibilité de faire évoluer les cartes dans le cas où de nouvelles données (composants ou spécifications techniques) sont introduites, nous nous sommes intéressés aux méthodes incrémentales de discrimination.

b) Apprentissage incrémental avec la méthode IGNG

Critère d'arrêt :

L'algorithme IGNG s'arrête lorsque la moyenne des déplacements des représentants devient stable et assez faible. Ceci est expliqué par la stabilité de la morphologie du réseau des représentants et l'inutilité d'itérations complémentaires.

Après plusieurs simulations, nous avons retenu les valeurs suivantes pour les paramètres de l'algorithme décrit en Annexe A:

$\text{Sigma} = 0.55$, $\text{Epsilon}_b = 0.5$, $\text{Epsilon}_m = 0.235$, et $\text{age}_{\text{max}} = 4$.

En utilisant cette configuration de paramètres, nous obtenons une carte formée par près de 600 représentants : moins de 7.5% du nombre de prototypes d'entrée (près de 8000), ce qui représente le nombre de représentants obtenus avec la méthode des cartes de Kohonen. En effet, ceci permet de comparer les résultats des deux apprentissages. Nous avons également utilisé la même procédure de décomposition de la base (80% - 20%).

La Figure 56 représente la moyenne de déplacement des représentants en fonction du nombre d'itérations. Dans cette application, le nombre d'itérations nécessaire est de 50.

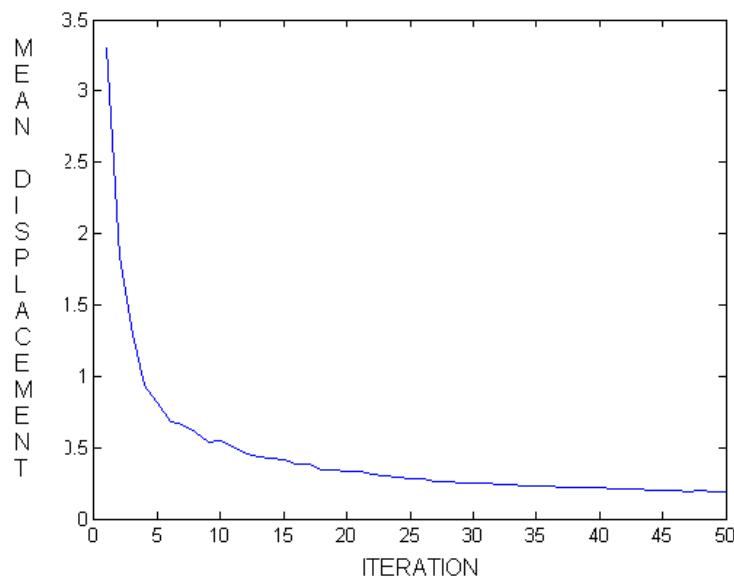


Figure 56 : Critère d'arrêt de l'algorithme IGNG

Les deux types d'interrogations menés sur les cartes de Kohonen ont été effectués sur les réseaux (ou cartes) résultant de cet apprentissage. Dans le premier type, nous avons cherché le moteur le plus adéquat à utiliser pour une combinaison des spécifications d'entrée non utilisée lors du processus d'apprentissage. Dans le deuxième type, nous avons essayé de déterminer le moteur adéquat avec des spécifications techniques incomplètes (sans le pic de puissance).

Les résultats des interrogations, 'quel moteur pour cette configuration des variables de conception de la base de test' effectuées sont représentés dans les Tableaux 13 et 14, sachant que les taux d'erreurs représentent des moyennes obtenues sur 100 simulations différentes, et que Nb BMU est le nombre de plus proches représentants considérés de la configuration présentée.

Nb BMU	Taux d'erreurs moyens
1	1.33%
2	0.06%
3	0%

Tableau 12 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode IGNG avec des prototypes de la base de tests

Nb BMU	Taux d'erreurs moyens
1	2.41%
2	0.57%
3	0.51%

Tableau 13 : Taux d'erreur moyens. Interrogation des cartes obtenues avec la méthode IGNG avec des spécifications techniques incomplètes (sans la valeur du pic de puissance) de la base de tests

Matrice de confusion

La matrice de confusion suivante détaille les résultats des interrogations effectuées sur la carte des moteurs obtenue avec la méthode IGNG lorsqu'une seule solution (type de moteur) est proposée. Elle permet de déterminer les types d'erreurs lors des interrogations.

Les lignes représentent les labels des moteurs qui devraient être proposés et les colonnes représentent les labels des moteurs qui ont été proposés. La case $\{i,j\}$ de cette matrice contient le nombre de fois où la meilleure solution est $i^{\text{ème}}$ moteur, la solution proposée, $j^{\text{ème}}$ moteur.

	'A'	'B'	'C'	'D'	'E'	'F'	'G'	'O'
'A'	255	5						
'B'	16	40						
'C'			333					
'D'	8			531				
'E'					12			
'F'						12		
'G'							124	
'O'								240

Tableau 14 : Matrice de confusion : Interrogations avec des prototypes de la base de test des cartes obtenues avec la méthode IGNG

Près de la moitié des erreurs peuvent être automatiquement évitées. En effet, lors d'une interrogation, le représentant le plus proche d'une donnée (en distance euclidienne) est déterminé. Si, lors de l'apprentissage, on détermine la distance maximale représentant la zone d'influence d'un représentant, plusieurs réponses (trop éloignées du représentant en question) peuvent être écartées. Les tests ont montré que dans les cas d'une proposition erronée 51,85% des fois, la distance entre la donnée et le représentant dépassait la zone d'influence de ce même représentant.

Aspect incrémental

L'une des raisons essentielles qui nous ont guidées vers l'utilisation de la méthode Incremental Growing Neural Gas est son aspect incrémental : Lors de l'ajout de nouvelles données ou de nouveaux composants, il suffit de lancer un apprentissage (complémentaire) sur la carte déjà construite avec ces nouveaux prototypes.

Cet aspect a été vérifié sur l'application du robot désempileur pour textile. En effet, dans un premier temps, la base de connaissances a été divisée en deux parties (20% - 80%). La plus grande base (80%) a servi pour l'apprentissage et l'autre pour les tests. Dans un deuxième temps, la base, qui a servi à l'apprentissage, a été divisée en deux parties (10% - 70%). La plus grande (70%) a servi à l'apprentissage initial, la plus petite (10%) pour un apprentissage incrémental (complémentaire). La même base (20%) a été utilisée pour les interrogations (test) dans les deux cas.

Les résultats de cette simulation ont montré que les différents taux d'erreur obtenus dans les deux cas sont identiques à 0,1% d'erreur près.

c) IGNG vs Kohonen

La méthode des cartes de Kohonen fournit aux concepteurs un bon outil de visualisation de l'ensemble de solutions. Elle fournit également un outil d'interrogation permettant de généraliser et de reconstituer des données manquantes, même dans le cas de connaissances imprécises.

A cause d'une topologie prédéfinie du réseau et d'une relation de voisinage fixe, l'outil d'interrogation obtenu par la méthode des cartes de Kohonen n'a pas été suffisamment performante lors des tests effectués.

Grâce aux propriétés de la méthode IGNG, les taux d'erreurs obtenus lors des interrogations sont nettement meilleurs. De plus, cette méthode permet un apprentissage sur de nouvelles données (spécifications techniques ou composants) de façon incrémentale (sans avoir à relancer l'apprentissage sur toute la base de connaissances).

Malheureusement, avec cette méthode, nous n'avons pas la possibilité de visualisation. En effet, avec notre connaissance limitée dans la théorie des graphes, nous n'avons pas pu montrer qu'il est possible d'avoir une idée précise sur la topologie (planéité, nombre et degrés des connexions...) des cartes obtenues. Cette piste a alors été abandonnée.

Dans cette section, nous avons proposé des outils aidant les concepteurs dans le choix de composants lors de l'activité de conception organique (Embodiment Design) : Une activité importante dans le processus de conception de produits complexes. Ces outils sont basés sur des méthodes de discrimination : cartes de Kohonen et Incremental Growing Neural Gas.

Testée sur un exemple de choix de moteurs/réducteurs pour un robot désempileur pour textile, cette étude propose aux concepteurs :

- Un bon outil de visualisation de l'ensemble de solutions, à partir des cartes de Kohonen. Il permet, pour une configuration de spécifications techniques, de déterminer la position relative d'un composant, par rapport à d'autres spécifications techniques, et optimisant la valeur d'un critère (dans ce cas, le pic de puissance)
- Un outil d'interrogation permettant de généraliser et de reconstituer des données manquantes, même dans le cas de connaissances imprécises. Grâce aux propriétés de la méthode IGNG, il permet un apprentissage sur de nouvelles données (spécifications

techniques ou composants) de façon incrémentale (sans avoir à relancer l'apprentissage sur toute la base de connaissances).

Conclusion

Dans ce chapitre, nous avons appliqué les méthodes proposées dans le Chapitre 4 sur deux cas d'optimisation multiobjectif en conception de robots. L'outil proposé nous a permis de déterminer, en un temps fixé a priori, l'ensemble des solutions efficaces du problème de conception d'un robot planaire. Pour ce faire, nous avons appliqué les deux méthodes de recherche permettent de choisir les variables de conception de manière à optimiser les objectifs de conception, tout en respectant les contraintes.

Dans la deuxième partie du chapitre, nous avons appliqué les méthodes de discrimination sur l'exemple d'un robot désempileur pour l'industrie textile. Ainsi avons-nous fourni aux concepteurs un outil de visualisation du domaine de validité des composants utilisés (moteurs et réducteurs des trois axes). Cet outil est obtenu par l'application de la méthode des cartes de Kohonen.

Cet outil a également été utilisé pour la recherche d'une solution (le composant à choisir) en fonction des variables de conception du problème. La méthode incrémentale de discrimination (Incremental Growing Neural Gas) s'est avérée plus précise que celle des cartes de Kohonen. De plus, son aspect incrémental la rend plus flexible aux besoins variables.

Lors des tests effectués, le taux d'erreur moyen obtenu lors de la recherche de solutions avec des combinaisons de variables non utilisées lors de l'apprentissage est de 1%. Lorsque les combinaisons sont incomplètes, le taux d'erreur moyen est de près de 2%.

Conclusion Générale et Perspectives

Les travaux de recherche menés au cours de cette thèse portent sur l'optimisation en conception multiobjectif et multidisciplinaire. D'une part, nous avons étudié différents modèles de conception de produits complexes. En effet, dans la littérature, riche, plusieurs types de modèles, couvrant différents points de vue des acteurs de la conception, existent. Ces points de vue doivent s'accorder pour générer des solutions compatibles avec les règles et objectifs de chaque métier et respectant les objectifs globaux de conception définis par les besoins du client. C'est dans ce but que nous nous sommes intéressés à l'Ingénierie Système, cadre fédérateur regroupant des modèles décrivant un large ensemble de points de vue des acteurs de la conception.

L'hétérogénéité des composants du système à concevoir nécessite des connaissances de différents domaines de l'ingénierie. L'implication d'experts de métiers différents complexifie l'ensemble des processus de conception à cause de leurs propres méthodes et terminologies.

Cette complexité se répercute sur l'ensemble des processus de conception ainsi que sur leurs activités support. Plus particulièrement, en conception organique, l'activité support d'optimisation peut devenir relativement longue, fastidieuse et coûteuse.

Aussi, l'implication de plusieurs composantes interagissant entre elles et qui plus est issues de disciplines différentes, fait que l'optimisation globale du système à concevoir ne se limite plus à l'optimisation de ses composantes du fait du couplage entre les disciplines mais plutôt à une optimisation du système global nécessitant des compromis entre les disciplines. C'est pour cette raison que nous nous sommes intéressés aux approches de MDO (Multidisciplinary Design Optimization) proposant des stratégies pour lever ces couplages.

Une autre difficulté en conception multidisciplinaire est due au fait que des solutions retenues lors des phases amont du cycle de développement peuvent être remises en question lors des phases aval de ce cycle à cause, notamment, de choix technologiques induisant de nouvelles exigences sur le système. Cette remise en question est aussi nécessaire que coûteuse. Les différents processus d'optimisation déjà effectués doivent alors être relancés.

Dans cette optique, ce travail de recherche vise à assister le concepteur dans l'activité support qu'est l'optimisation. Le but est de définir une méthodologie outillée pour l'aide à

l'optimisation multiobjectif en phase de conception organique de systèmes multidisciplinaires dans les cas où les problèmes peuvent être posés dans un formalisme numérique. Cette aide comporte deux volets principaux. Le premier concerne la capacité de dégager rapidement un ensemble de solutions efficaces : le front de Pareto, donnant ainsi le choix aux concepteurs entre plusieurs alternatives. Le deuxième concerne un outil d'aide à l'interprétation d'un grand ensemble de données multidimensionnelles.

Concernant le premier volet, nous avons proposé deux algorithmes basés sur des méta-heuristiques et répondant à la problématique de l'affectation de valeurs aux variables en conception pluridisciplinaire et multiobjectif. Ces deux algorithmes d'optimisation permettent de fournir aux concepteurs, en un temps acceptable, un ensemble de solutions efficaces.

Les deux approches proposées ont été testées dans le cas d'une optimisation des variables de conception d'un robot à deux degrés de liberté. La définition de l'architecture d'un tel robot nécessite l'attribution de valeurs à 10 variables mixtes (discrètes et continues) et couplées. Un algorithme de recherche exacte, évaluant les 325 milliards solutions possibles et en retenant la (ou les) meilleure(s) nécessite un temps de calcul de plus de 111 heures.

Le premier algorithme, variante du tabou de Glover par sa gestion des variables mixtes et par sa détermination du front de Pareto, est applicable dans le cas où un ordre hiérarchique peut être défini pour les multiples objectifs du problème. Après le réglage de ses paramètres par essais erreur, l'exécution de l'algorithme a permis, en 250 secondes dans le cadre de notre étude, de dégager une solution de très bonne qualité et un front de Pareto constitué des solutions efficaces rencontrées lors de la recherche.

Le deuxième algorithme proposé est une hybridation entre recherche exacte et méta-heuristiques. Une première phase de recherche exacte, dite de diversification, permet de dégager les zones intéressantes de l'espace de recherche. Ces zones sont mieux fouillées lors de la seconde phase, dite d'intensification et basée sur des mécanismes d'exploration de voisinage et d'évitement de piégeage par les optimums locaux.

Ce deuxième algorithme a été expérimenté sur le même exemple. Son exécution a permis de dégager rapidement, en un temps de calcul inférieur à 300 secondes, un front de Pareto de bonne qualité et couvrant tout l'espace de recherche.

Le deuxième volet porte sur les outils de discrimination fournissant aux concepteurs une aide à l'interprétation des fronts de Pareto. Ces outils sont basés sur deux méthodes

d'apprentissage : les cartes de Kohonen et Incremental Growing Neural Gas (IGNG). L'apport de ces cartes a été validé lors d'une étude concernant les variables de conception d'un robot désempileur pour l'industrie textile. Un tel système composé de trois axes couplés, présente des difficultés lors des choix de ses composants (moteurs et réducteurs des trois axes). En effet, après la discrétisation des variables de conception, une méthode de sélection optimale de moteurs réducteurs a été appliquée pour chaque configuration de ces variables. Par la suite, et sur la base constituée par ces différentes solutions, les méthodes de discrimination sont lancées à travers des procédures d'apprentissage.

Les cartes de Kohonen permettent de représenter, en deux dimensions, un ensemble de données initialement placées dans un espace de dimension supérieure. Elles fournissent donc aux concepteurs un outil permettant de visualiser l'ensemble des solutions retenues (le front de Pareto) et d'avoir certaines informations pertinentes, comme par exemple, des clusters regroupant des solutions semblables. L'application de cette méthode sur l'étude du robot désempileur a permis de mettre en évidence les domaines de validité des moteurs et réducteurs. Ces clusters mettent en évidence les valeurs des variables de conception pour lesquelles les différents composants sont retenus. Ceci permet aux concepteurs de connaître leur marge de manœuvre concernant les choix des variables. Par exemple, dans le cas où la solution retenue se trouve en frontière d'un cluster, les concepteurs savent qu'une variation mineure sur les variables implique un changement dans le choix d'un (ou de plusieurs) composant(s). Par contre, si la solution se trouve au milieu d'un cluster, cela indique que les concepteurs disposent d'une marge assez importante sur une (ou plusieurs) variables.

Afin de fournir un visuel aux utilisateurs, la topologie, tout comme la structure de la carte, est imposée au début de la procédure d'apprentissage. Nous nous sommes également intéressés à quelques méthodes incrémentales de discrimination, et plus particulièrement à la méthode IGNG. La particularité de cette méthode est que le réseau (ou carte) est construit au fur et à mesure de la procédure d'apprentissage. Ce qui implique qu'à chaque fois que l'on désire rajouter de nouvelles données (valeurs de variables de conception ou composants) à notre base de connaissances, il n'est pas nécessaire comme avec les cartes de Kohonen de reprendre toute la procédure depuis le début.

L'application de cette méthode nous a permis de fournir aux concepteurs un outil de recherche de composants. Le réseau créé lors de l'apprentissage est alors utilisé comme mémoire associative permettant, à partir une combinaison de variables de conception, même imprécises et incomplètes, de déterminer les composants optimaux sans recourir à

l'application de la méthode de sélection, coûteuse en temps car elle nécessite d'évaluer quasiment toutes les combinaisons possibles...

Lors de ces travaux de recherches plusieurs pistes de travaux futurs ont été dégagées. Certaines, concernant des améliorations pouvant être apportées nos algorithmes de recherche, ont été proposées dans la conclusion du Chapitre 4. Ce travail amont a ainsi permis de mettre en évidence des domaines de recherche à approfondir pour traiter du problème difficile de la conception de systèmes pluridisciplinaires. Il conviendrait notamment de confronter notre méthode avec celles approchées et hybrides en optimisation multiobjectif.

D'autres pistes dégagées s'articulent autour du cadre d'utilisation de nos approches. En effet, l'optimisation étant un processus support à l'ingénierie s'appliquant à toutes les phases de conception, il serait intéressant de voir si nos travaux peuvent être transposés à une phase autre que la conception de l'architecture organique.

D'un point de vue plus technique, nos approches ont été illustrées sur un cas disposant d'un couplage faible résolu par la stratégie mono-niveau « All-At-Once ». Il serait intéressant d'étudier les stratégies multi-niveaux afin d'appliquer les approches proposées dans ce cadre.

Dans le cas d'une conception multidisciplinaire, il arrive, dans certains cas, que les équipes de concepteurs et/ou experts métiers soient distribués géographiquement. Afin de prendre en compte cet aspect dans nos travaux futurs, il pourrait être intéressant de s'intéresser à une implémentation multi-agents de la résolution de certains problèmes de conception. En effet, non seulement leur modularité pourrait pallier à cette problématique de distance, mais en plus, leurs capacités d'apprentissage et d'auto-organisation, entre autres, pourraient être utilisées dans une optique de paralléliser les calculs et de réduire les coûts de résolution.

Références bibliographiques

- [Agte'06] AGTE J. S. A tool for application of bi-level integrated system synthesis (BLISS) to multidisciplinary design of optimization problems, 2006.
- [Alexandrov'99] ALEXANDROV N.M., LEWIS R.M. Comparative Properties of Collaborative Optimization and Other Approaches to MDO. ICASE report No. 99-24. NASA/CR-1999-209354. Juillet, 1999.
- [Alexandrov'00a] ALEXANDROV N.M., LEWIS R.M., Analytical and Computational Aspect of Collaborative Optimization, Rapport Technique NASA Langley Research Center no NASA/TM-2000-21010, 2000.
- [Alexandrov'00b] ALEXANDROV N.M., LEWIS R. M. Algorithmic perspectives on problem formulations in MDO. Proc. Of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 6-8 Sept, 2000. Long Beach, CA. AIAA-2000-4719.
- [AFIS, 2007] Association Française d'Ingénierie Système. www.afis.fr
- [Ahmed'02] AHMED S., HANSEN C.T. A decision-making model for engineering designers, Shahin T.M.M. (ed.), Computer based design, EDC2002, pp. 217-227, Cambridge, 2002.
- [Andersson'00] ANDERSSON Johan. A survey of multiobjective optimization in engineering design. Technical Report: LiTH-IKP-R-1097. Department of Mechanical Engineering. Linköping University, Linköping, Sweden, 2000.
- [Balling'94] BALLING R. J., SOBIESKI J. Optimization of coupled systems : A critical overview of approaches. NASA Contractor Report 195019 ICASE Report No. 94-100. Décembre, 1994.
- [Battiti'94] BATTITI R., TECCHIOLLI G. The reactive tabu search. ORSA Journal on Computing, Vol. 6, pp. 128-140, 1994.
- [Barichard'03] BARICHARD V., Approches hybrides pour les problèmes multiobjectifs, Thèse, Ecole Doctorale d'Angers, France, 2003.

- [Berthiau'01] BERTHIAU Gérard, SIARRY Patrick. Etat de l'art des méthodes d'optimisation globale. *RAIRO Operations Research* 35329-365, 2001.
- [Blessing'96] BLESSING L. Comparison of design models proposed in prescriptive literature. Proc. of the COSTA3/A4 Int. Res. Workshop on The role of design in the shaping of technology, pp. 187-212, Lyon, 1996.
- [Braesch'95]. BRAESCH Christian, et al. La modélisation systémique en entreprise, Hermès. 1995.
- [Braun'96] BRAUN R.D., A.A. MOORE, KROO I.M. Use of the Collaborative Optimization Architecture for Launch Vehicle Design. In: 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1996.
- [Brown'06] BROWN Nichols F., OLDS John R. Evaluation of multidisciplinary Optimization Techniques Applied to a Reusable Launch Vehicle. *Journal of spacecraft and rockets*, Vol. 43, No. 6, November–December 2006.
- [Carrier'04] CARRIER G. Multidisciplinary optimization of a supersonic transport aircraft wing platform. Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004. Jyväskylä, 24-28 July 2004.
- [Cirrus'98] CIRRUS S. Discovery of Design Methodologies for the Integration of Multidisciplinary Design Problems. PhD Thesis. Worcester Polytechnic Institute, UK, 1998.
- [Clivillé'06] CLIVILLE V., BERRAH L., MAURIS G. Quantitative expression and aggregation of performance measurements based on the MACBETH multi-criteria method. *International Journal of Production Economics*, 2006.
- [Clément'09] CLEMENT Joel. Optimisation multidisciplinaire : étude théorique et appliquée. Thèse soutenue le 3 juin 2009. Université de Toulouse, France, 2009.
- [Coelho'09a] COELHO Rajan Filomeno, BREITKOPF Piotr. Optimisation multidisciplinaire en mécanique, Vol. 1. Ed. Hermes Lavoisier. ISBN 978-2-7462-2195-6.
- [Coelho'09b] COELHO Rajan Filomeno, BREITKOPF Piotr. Optimisation multidisciplinaire en mécanique, Vol. 2. Ed. Hermes Lavoisier. ISBN 978-2-7462-2195-6.
- [Coello'96] COELLO C.A. Multiobjective Engineering Design Optimization using Genetic Algorithms. PhD thesis, Department of Computer Science, Tulane University, May 1996.

- [Cramer'93] CRAMER Evin J., DENNIS Jr J. E., FRANK Paul D., LEWIS Robert Michael, SHUBIN Gregory R. Problem Formulation for Multidisciplinary Optimization. Proc. of the AIAA Symposium on Multidisciplinary Design Optimization, CRPC-TR93334, September 1993.
- [Cusimano'07] CUSIMANO G. Optimization of the choice of the system electric drive device/transmission for mechatronic applications. Mechanism and Machine Theory 42, 2007.
- [Dequidt'99] DEQUIDT A., VALDES E., CASTELAIN J.M. Nombres adimensionnels pour la conception intégrée des mécanismes commandés. Actes du 14e Congrès Français de Mécanique, 1999.
- [De Rosnay'75]. DE ROSNAY Joël, Le microscope, Le Seuil, 1975.
- [Durand'79] DURAND Daniel. La systémique. Presses universitaires de France. 126p. ISBN : 2 13 052345 5.
- [Duranté'04] DURANTE N., DUFOUR A., PAIN V. Multidisciplinary Analysis and Optimisation Approach for the Design of Expendable Launchers. Proc of the 10th AIAA/ISSMO Multidisciplinary Analysis and optimization Conference, New York, 2004.
- [Ennaji'04] ENNAJI A., PRUDENT Y. Clustering incrémental pour un apprentissage distribué : vers un système évolutif et robuste. CAP'2004, 2004.
- [Ennaji'05] ENNAJI A., PRUDENT Y. An Incremental Growing Neural Gas learns topologies. IJCNN'05, Montréal, Canada, 2005.
- [Faisandier'05a] FAISANDIER A. Ingénierie des systèmes complexes : Compléments et mise en œuvre. MAP système, Edition 1, 2005.
- [Faisandier'05b]. FAISANDIER Alain. Revisiting the notion of system - Organizations and Enterprises as systems. INCOSE 2005, 15th Annual International Symposium, Systems Engineering: Bridging Industry, Government, and Academia, Rochester, New York, USA 10-14 July 2005.
- [Flager'07] FLAGER Forest, HAYMAKER John. A Comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace industries. Report of the Department of Civil and Environmental Engineering, Terman Engineering Center, Stanford, US. 2007.

- [Fonseca'93] FONSECA C.M., FLEMING P.J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416–423, San Mateo, California, 1993 .
- [Gandibleux'02] GANDIBLEUX X., DEGOUTIN F., DELORME X., A first feedback on set packing problems with two objectives. Workshop on Multiple Objective Metaheuristics (MOMH), Carré des Sciences, Paris, France, November 4-5 2002.
- [Garey'79] GAREY M.R. et JOHNSON D.S. Computers and Intractability : A Guide to the Theory of NP Completeness. W.H. Freeman and Company, New York, 1979.
- [Gero'90] GERO, J.S. Design prototypes: A knowledge representation scheme for design. Artificial Intelligence Magazine, Vol. 11, 4, 26-36, 1990.
- [Gero'02] GERO, J.S., KANNENGISSER U. The situated function-behaviour-structure framework. Gero J.S. Eds., Artificial Intelligence in Design'02, Kluwer, 2002.
- [Glover'89] GLOVER F., LAGUNA M. Tabu Search. Eds. John Wiley and Sons, 1989.
- [Glover'95] GLOVER F., KELLY P., LAGUNA M. Genetic algorithms and tabu search : hybrids for optimization. Computers Operational Research, Vol. 22, pp. 111-134, 1995.
- [Habet'04] HABET D. Coopération entre recherches locale et exhaustive pour l'optimisation combinatoire. Thèse de doctorat, Ecole Doctorale en Sciences et Santé, 2004.
- [Hajela'90] HAJELA P. Genetic search - an approach to the nonconvex optimization problem. AIAA Journal. Vol. 28(7), pp.1205–1210, 1990.
- [Hulme'00] HULME K.F., BLOEBAUM C.L. A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE. Structural Multidisciplinary Optimization 19, pp17-35. Springer-Verlag 2000.
- [Irisarry'09] IRISARRI François-Xavier. Stratégies de calcul pour l'optimisation multiobjectif des structures composites. Thèse de l'Université de Toulouse III, France. 2009, 274p.
- [Jones'02] JONES D.F., MIRRAZAVI S.K., TAMIZ M.. Multi-objective meta-heuristics: An overview of current state-of-the-art. European Journal of Operational Research 137. p. 1-9. 2002.
- [Kirkpatrick'83] KIRKPATRICK S., GELATT C.D., VECCHI P.M. Optimisation by Simulated Annealing. Science, Vol. 220 : 671-680, 1983.

- [Kodiyalam'98] KODIYALAM Srinivas. Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Phase I. NASA/CR-1998-208716.
- [Kohonen'01] KOHONEN T. Self-organizing maps. Springer, Berlin, Germany, 2001.
- [Kopacek'09] KOPACEK P., LUTZ P., PASHKEVICH A., Robotics and Factory of the Future, New Trends and Challenge in Mechatronics', from INCOM 206. Mechatronics, Vol. 19, Issue 4, p. 435-436, June 2009.
- [Laporte'92] LAPORTE G. The Vehicle Routing Problem : an overview of exact and approximate algorithms. European Journal of Operational Research, 59(3). pp.345:348, 1992.
- [Larsen'00] LARSEN J. Parallelization of the vehicle routing problem with time windows. PhD Thesis, Institute of Mathematical Modelling, Technical University of Denmark, 2000.
- [Le Coq'92] LE COQ Marc. Approche intégrative en conception de projet. Thèse de doctorat, ENSAM, décembre 1992.
- [Le Moigne'90] LE MOIGNE Jean-Louis (1990), La modélisation des systèmes complexes, Paris, Bordas, Dunot.
- [Lonchamp'04] LONCHAMPT P. Co-évolution et processus de conception intégrée de produits : Modèle et support de l'activité de conception. Thèse, Institut National Polytechnique de Grenoble, France, 2004.
- [Matthews'02] MATTHEUWS P.C., BLESSING L.T.M, WALLACE K.M. The introduction of a design heuristics extraction metho, Advanced Engineering Informatics, vol. 16, 2002.
- [Meinardier'98]. MEINARDIER Jean-pierre, Ingénierie et intégration des systèmes, Hermes. 1998.
- [Merkel'01] MERKEL D., HUI S., DITTENBACH M., RAUBER A. Adaptative Hierarchical Incremental Grid Growing : An architecture for high-dimensional data visualisation. 2001.
- [Micaelli'02] MICAELLI J.P. Institutionnalisme, évolutionnisme: le défi de la conception. Actes des Journées d'étude Institutionnalismes et Évolutionnismes-Confrontations autour de perspectives empiriques, Lyon, 2002.
- [Morin'77] MORIN Edgar, La Méthode - la nature de la nature, Le Seuil. 1977.
- [Mortensen'99] MORTENSEN N.H., ANDREASEN M. M. Contribution to a Theory of Detailed Design. Proc. Of the 10th Symposium « Fertigungsgerechtes Konstruieren », 1999.

[Mouelhi'07] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. A self-organizing map for visualizing the validity domain of mechatronic components. Research and Education in Mechatronics REM'2007, Tallinn, Estonie, 2007.

[Mouelhi'08] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. A Decision Help Tool based on Incremental Growing Neural Gas for components selection. Mechatronics 2008, Le Grand Bornand, France, 2008.

[Mouelhi'09a] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. A hybrid search algorithm in a multi-agent system environment for multicriteria optimization of products design. Proceedings of the International Joint Conference on Neural Networks (IJCNN09), pp. 2160-2167, juin 2009, Atlanta, USA, ISBN : 978-1-4244-3553-1.

[Mouelhi'09b] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. Multi-criteria Optimization and Evaluation in Complex Products Design. 39th International Conference on Computers & Industrial Engineering (CIE 39), juillet 2009, Troyes, France.

[Mouelhi'09c] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. A Hybrid Search Algorithm for Multicriteria Optimization and Evaluation in Mechatronic Products Design. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009), juillet 2009, Singapour.

[Mouelhi'09d] MOUELHI Ouael, COUTURIER Pierre, REDARCE Tanneguy. An Artificial Intelligence Approach for the Multicriteria Optimization in Mechatronic Product Design. IEEE International Conference on Mechatronics and Automation (ICMA 2009), aout 2009, Changchun, Jilin, Chine.

[Norme Française'90] NORMES FRANCAISES :

NF X50-151, décembre 1991, Analyse de la valeur, analyse fonctionnelle - Expression fonctionnelle du besoin et cahier des charges fonctionnel.

NF X50-152, août 1990, Analyse de la valeur-Caractéristiques fondamentales.

NF X50-100, décembre 1996, Analyse fonctionnelle – Caractéristiques fondamentales.

[Oja'01] OJA M., KASKI S., KOHONEN T. Bibliography of Self-Organizing Map (SOM) Papers : 1998-2001 Addendum. Helsinki University of Technology, Neural Networks Research Centre, Finlande, 2001.

- [Osyczka'85] OSYCZKA A. Multicriteria optimization for engineering design. Eds. John S. Gero. Design Optimization (Academic Press). 1985. pp. 193–227.
- [Pahl'96] G. Pahl, W. Beitz. Engineering Design - a systematic approach. Springer - Verlag, Londres, 1996.
- [Parte'04] PARTE Y., MASMOUDI M. Disciplinary Interaction Variable Elimination (DIVE) Approach for MDO.2004
- [Penalva'97] PENALVA Jean Michel. La modélisation par les systèmes en situations complexes. Thèse de l'Université de Paris XI - Paris Sud. 1997.
- [Pugh'91] PUGH Stuart. Total design – Integrated methods for successful product engineering. Pearson Education Limited, Addison-Wesley Publishers Ltd., 1991.
- [Purcell'94] PURCELL T., GERO J., EDWARDS H. and MATKA E. Design fixation and intelligent design aids. Gero J.S. and Sudweeks F. (ed.), Artificial Intelligence in Design, Kluwer, 1994.
- [Ribert'99] RIBERT A., ENNAJI A., LECOURTIER Y. An Incremental Hierarchical Clustering. Vision Interface '99, Trois-Rivières, Canada, 1999.
- [Rakotoarisoa'93] RAKOTOARISOA R.D. Contribution à la planification de trajectoires pour la robotique. Application à un robot désempileur de textile. Thèse, Institut National des Sciences Appliquées de Lyon, France, 1993.
- [Regli'00] REGLI W.C., HU X., ATWOOD M., SUN W. A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. Engineering with Computers, 16, 209-235, 2000.
- [Roos'06] ROOS F., JOHANSSON H., WIKANDER J. Optimal selection of motor and gearhead in mechatronic application. Mechatronics, VOL. 16, 2006.
- [Roy'05] ROY B., Paradigms and challenges, Multiple Criteria Decision Analysis - State of the Art Surveys, Springer, pages 3-24, 2005.
- [Roy'93] ROY B., BOUYSSOU D., Aide Multicritère à la Décision : Méthodes et Cas. Economica, Paris, 1993.
- [Scaravetti'04] SCARAVETTI D. Formalisation préalable d'un problème de conception, pour l'aide à la décision en conception préliminaire. Thèse de l'Ecole Nationale Supérieure d'Arts et Métiers, France, 2004.

- [Shubin'95] SHUBIN Gregory R. Application of Alternative Multidisciplinary Optimization Formulations to a Model Problem for Static Aeroelasticity. Journal of computational physics. Vol. 118. pp. 73-85. 1995.
- [Simon'73] SIMON H. The structure of ill-structured problems, Artificial intelligence, Vol. 4, pp. 181-20, 1973.
- [Sobieski'88] SOBIESKI J. Optimization by Decomposition : A Step from Hierarchic to Non-Hierarchic Systems. NASA Technical Report CP-3031, 1988.
- [Sobieski'93] SOBIESKI J. Multidisciplinary Design Optimization: Attempt at Definiton. Proceedings of Industry-University Workshop on Multidisciplinary Aircraft Design, pp 23-48, 1993.
- [Sobieski'96] SOBIESKI J. Multidisciplinary Aerospace Design Optimization : Survey of Recent Developments, AIAA, 1996.
- [Sobieski'98] SOBIESKI J., AGTE S., SANDUSKY R.. Bi-Level Integrated System Synthesis (BLISS). NASA / TM - 1998 - 208715, Aug. 1998.
- [Steuer'02] STEUER R., QI Y., Computational Investigations Evidencing Multiple Oblectives in Portfolio Optimization, in Multiobjective Programming and Goal-Programming, Theory and applications, Springer, pp. 35-43, 2002.
- [Suh'90] SUH N.P. The Principles of Design, Oxford University Press, 1990.
- [Ullman'02] ULLMAN D.G., The mechanical design process, Mc Graw Hill, 3ème éd., 2002.
- [Ulrich'00] ULRICH K.T. and EPPINGER S.D., Product design and development, Second edition, McGraw Hill International editions, 2000.
- [Vesanto'99] VESANTO J., HIMBERG J., ALHONIEMI E., PARHANKANGAS J. Self-organizing map in Matlab : the SOM toolbox. Proceedings of the Matlab DSP Conference. Finlande, 1999.
- [Xie'09] XIE H., GAUTIER M., LUTZ P., REGNIER S., CHAILLET N., Recent advances in the study of Micro/nano Robotics in France, IARP Workshop on Service Robotics and Nanorobotics., Chine, 2009 [hal-00432750-version 1].
- [Yannou'98] YANNOU B., « Analyse fonctionnelle et Analyse de la Valeur », Tollenaere M. (Dir.), Conception de produits mécaniques, méthodes, modèles et outils, Hermes, 1998.

[Yi'07] YI S. I., SHIN J.K., PARK G.J. Comparison of MDO methods with mathematical examples. Structural multidisciplinary optimization 35. pp. 391-402. Springer-Verlag, 2008.

[Zennir'04] ZENNIR Y. « Apprentissage par renforcement et systèmes distribués : Application à l'apprentissage de la marche d'un robot hexapode ». Thèse, Institut National des Sciences Appliquées de Lyon, France, 2004.

Annexe A : Méthodes de discrimination utilisées

1 Cartes de Kohonen

La méthode de construction des cartes auto-organisatrices de Kohonen est une méthode de quantification vectorielle qui place des représentants dans un espace de données en conservant les propriétés de voisinage et de densité de la distribution de données à représenter. Elle permet de représenter dans un espace de dimension réduite des données de dimension supérieure à 3 qui ne peuvent être visualisées facilement.

Cette méthode utilise des réseaux de neurones dont le calcul des paramètres est basé sur le mécanisme d'apprentissage non supervisé suivant :

- Chaque neurone est un représentant de d paramètres (d étant la dimension des données à représenter). Les neurones sont connectés entre eux par une relation de voisinage déterminant la topologie de la carte. Initialement les paramètres des neurones sont initialisés aléatoirement.

Figure 57 montre le voisinage de niveaux 0, 1 et 2 du point central de la carte qui a une topologie 2D avec une structure de voisinage hexagonale ou rectangulaire.

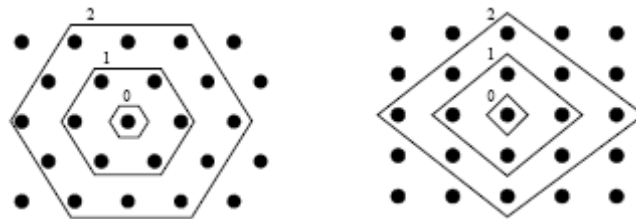


Figure 57: Voisinage 2D du neurone central (Hexagonal à gauche ou rectangulaire à droite)

- Deux modes d'apprentissage sont possibles. Dans le premier mode séquentiel et itératif, les données X sont présentées une à une, et l'algorithme, comme le montre la Figure 58, met à jour la position du représentant le plus proche de X , appelé BMU (Best Matching Unit), ainsi que ces voisins.

Dans cette mise à jour, les représentants sont rapprochés de X proportionnellement à la distance qui les sépare de X avec un facteur de proportionnalité décroissant avec le niveau de voisinage et décroissant dans le temps.

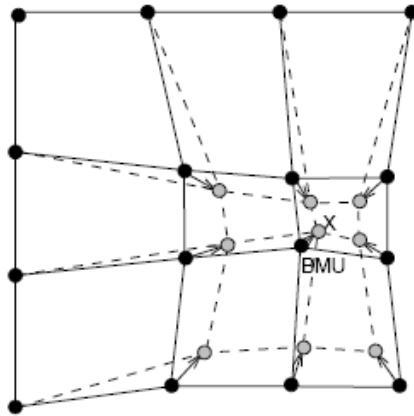


Figure 58: Mise à jour du neurone gagnant (BMU) et de ses voisins directs

Quant au deuxième algorithme appelé batch et aussi itératif, l'ensemble de données est présenté en même temps. Les déplacements des représentants sont les moyennes des déplacements calculés pour chaque présentation des données de la base. Le deuxième algorithme est souvent plus rapide que le premier.

L'implémentation de cette méthode, sous l'environnement Matlab a donné naissance à un outil permettant de créer, de visualiser et d'analyser des données : SOM toolbox (boîte à outils SOM) téléchargeable avec une licence GNU sur <http://www.cis.hut.fi/projects/somtoolbox>.

2 Incremental Growing Neural Gas (IGNG)

Cette méthode est un modèle constructif et incrémental de carte auto-organisatrice, ne présentant aucune contrainte sur la topologie du réseau. La mise à jour continue se fait par apprentissage de type Hebbien compétitif à chaque présentation d'un prototype de la base de données. Les règles définissant cet apprentissage sont les suivantes :

- Si dans l'hypersphère de rayon σ (Figure 60), se trouvent les deux plus proches voisins de X, on renforce la connexion existante (en lui donnant un âge nul), sinon on crée un ou deux nouveaux représentants et une liaison entre représentants les plus proches de X (Figure 61).
- L'âge des connexions est incrémenté à chaque présentation de donnée X. Une connexion atteignant un âge maximum est supprimée.
- Les neurones isolés (reliés à aucun autre), sont supprimés

- Le représentant le plus proche et ses voisins directement connectés sont rapprochés de la donnée X , le plus proche représentant d'avantage que ses voisins.

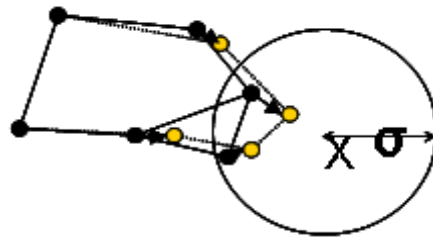


Figure 60 : Mise à jour des neurones gagnant : 1er cas

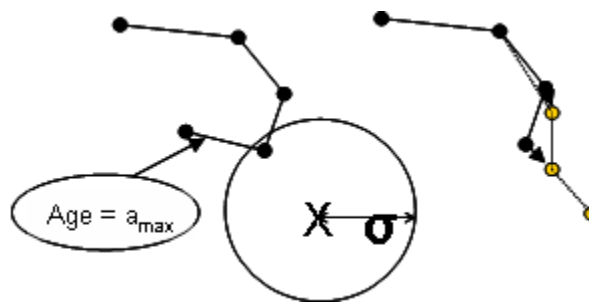


Figure 59 : Mise à jour des neurones gagnant : 2ème cas

Avant de présenter l'algorithme d'apprentissage de la méthode IGNG, nous définissons quelques variables :

- Epsilon_b : pas d'adaptation (facteur de rapprochement) du représentant le plus proche,
- Epsilon_m : pas d'adaptation du second représentant le plus proche,
- Sigma : rayon de l'hyper-sphère représentant le domaine d'influence de chaque neurone,
- age_{\max} : L'âge maximum d'une connexion.

Algorithm 1: Incremental Growing Neural Gas Algorithm

```
begin
  repeat
     $\delta \leftarrow$  choose a prototype randomly from database;
    Winner  $\leftarrow$  nearest node ( $\delta$ );
    if (distance (Winner,  $\delta$ )  $\leq \sigma$ ) then
      | update position (Winner,  $\delta$ ,  $\epsilon_b$ ,  $\epsilon_m$ );
    else
      | create new node( $\delta$ );
    Second  $\leftarrow$  second nearest node ( $\delta$ );
    if (distance (Second,  $\delta$ )  $\geq \sigma$ ) then
      | update position (Second,  $\delta$ ,  $\epsilon_b$ ,  $\epsilon_m$ );
    else
      | create new node ( $\delta$ );
    Create connection (Winner, Second);
    Increment connections age (Winner, neighbours);
    Connection age (Winner, Second)  $\leftarrow$  0;
    for each connection do
      | if age  $\geq$  max age then
        | | delete it;
    until Stop Criterion;
    for each prototype  $\delta$  do
      | Winner  $\leftarrow$  nearest node ( $\delta$ );
      | Winner type vote (prototype type);
    for each node do
      | Type  $\leftarrow$  most voted type
    Eliminate nodes that are a BMU of any prototype;
    Interrogate;
end;
```

Figure 60 : Algorithm Incremental Growing Neural Gas

Annexe B : Modèle du robot à deux degrés de liberté

Cette annexe est consacrée aux détails de l'architecture du robot à deux degrés de liberté. Nous commençons par rappeler l'architecture, le cahier des charges et les variables de conception. Nous définissons par la suite les domaines de conception ainsi que les équations permettant de calculer les objectifs en fonction des variables.

1 Architecture

Comme la montre la Figure 63, l'architecture générale retenue, le robot planaire à deux degrés de liberté est composé de deux segments ou axes. Le premier axe est fixé au plan de travail par sa base d'un côté, et relié de l'autre côté au deuxième axe qui porte sur son extrémité la charge à déplacer.

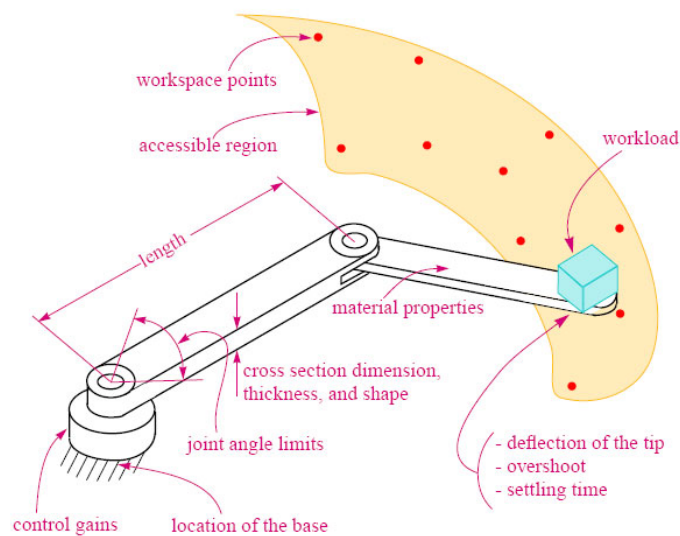


Figure 61 : Architecture du robot à deux degrés de liberté [Cirrus'98]

Dans le cahier des charges, la répartition des points, devant être atteints par l'extrémité du robot et formant l'espace de travail, décrit un petit M (« small M ») comme le montre la Figure 62.

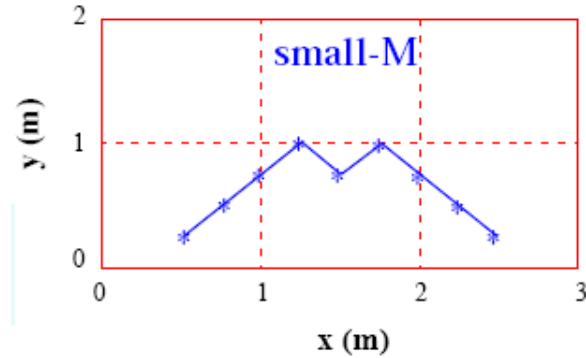


Figure 62 : Ensemble des points à atteindre par l'extrémité du robot formant un petit M

2 Variables de conception

Afin de définir l'architecture décrite ci-dessus, les variables de conception sont regroupées par rapport aux points de vue disciplinaires : cinématique, structure des matériaux et dynamique.

- Les variables géométriques : la localisation de la base du robot dans le plan de travail (X_{base} et Y_{base}), les longueurs des bras (L_1 et L_2), et le type des angles (configuration gauche ou droite du « coude du robot »: Or).
- Les variables structurelles : le type de matériel (mat), les sections (D_1 et D_2), le rapport entre l'épaisseur et le diamètre des bras (Thickness : Th). Ces variables et les longueurs des bras et doivent être choisis de manière à préserver la solidité du robot et de réduire son poids.
- Les variables dynamiques : le temps de réponse (Settling Time : ST) et le dépassement maximum du système (Overshoot : ov).

3 Domaines des variables

Les domaines des variables cinématiques sont définis dans le Tableau 15, Où :

- X_i, Y_i sont les coordonnées des points à atteindre,
- $L = \max (X_i) - \min (X_i)$
- $W = \max (Y_i) - \min (Y_i)$

Variable	Valeur minimale	Valeur maximale
X_{base} (m)	$\text{Min}(X_i) - L/2$	$\text{Max}(X_i) + L/2$
Y_{base} (m)	$\text{Min}(Y_i) - W/2$	$\text{Max}(Y_i) + W/2$
L_1 (m)	0.01	$\text{Max}\{W,L\}$
L_2 (m)	0.01	$\text{Max}\{W,L\}$
Orientation	À droite	À gauche

Tableau 15 : Domaines des variables cinématiques

La Figure 63 montre, pour l'exemple choisi, les limites de la localisation de la base du robot.

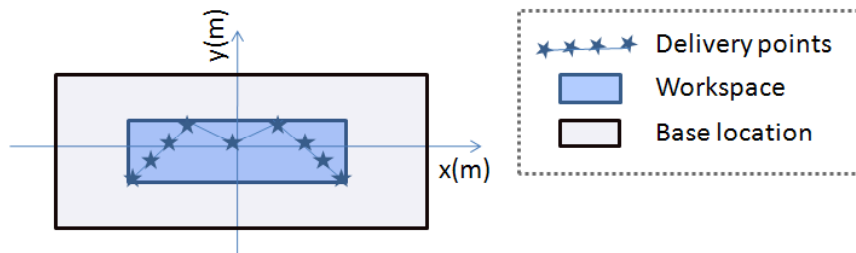


Figure 63 : Points à atteindre, workspace et localisation de la base

La Figure 64 montre un exemple de solutions avec les deux orientations possibles.

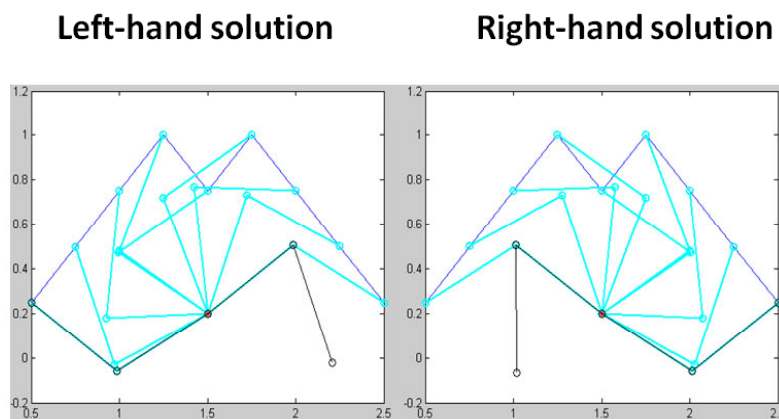


Figure 64 : Exemple de solutions

Concernant les variables structurelles Th et mat , les valeurs possibles sont :

- $0.1 \leq Th \leq 0.5$, avec $Th = t/d$ (Figure 65)

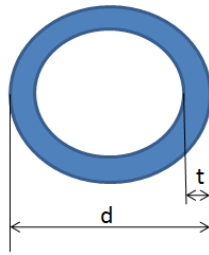


Figure 65: Epaisseur des bras

- Le matériau est choisi dans le Tableau 16 :

Matériau	Limite d'élasticité (MPa)	Module d'Young (GPa)	Densité (Kg/m ³)
Aluminium	200	70	2700
Cuivre	330	130	8920
Acier	450	210	8000

Tableau 16 : Matériaux utilisés et leurs caractéristiques

Quant aux variables dynamiques, nous avons fixé la valeur du premier dépassement sur chaque axe à 5%, et le temps de réponse (ST) est choisi entre 1 et 3 secondes.

4 Objectifs de conception

L'objectif des équipes de concepteurs est de choisir la meilleure combinaison de variables de conception afin d'optimiser les objectifs suivants :

- Maximiser la région accessible (domaine géométrique) : l'ensemble des points dans l'espace accessible par la pointe du robot est appelé la région accessible (Accessible Region : AR).

$$AR = (\max(\theta_1) - \min(\theta_1)) * L_1 * L_2 * (\cos_{\max}(\theta_2) - \cos_{\min}(\theta_2)),$$

Avec θ_1 et θ_2 les angles formés par les bras et l'horizontale.

- Minimiser la flèche (domaine structure) : la flèche des bras (Deflection : Def) est un indicateur de la rigidité du robot.

$Def = Db1 + Db2 + Db3$, avec

$$D_{b1} = ((m_2 + m_c) * g * L_2^3) / (3 * E_2 * I_2)$$

$$D_{b2} = ((m_1 + m_2 + m_c) * g * L_1^3) / (3 * E_1 * I_1) + ((m_2 + m_c) * g * L_2 * L_1^2) / (2 * E_1 * I_1)$$

$$D_{b3} = \{((m_1 + m_2 + m_c) * g * L_1^2) / (2 * E_1 * I_1) + ((m_2 + m_c) * g * L_2 * L_1) / (E_1 * I_1)\} * L_2$$

Où, E et I sont le module d'Young et l'inertie des axes. m_1 , m_2 et m_c sont les masses des axes et de la charge à transporter, g attraction terrestre.

- Minimiser les gains de commande (domaine dynamique) : la loi de commande doit garantir les performances du robot, en ce qui concerne la stabilité et la précision. Les gains de commande reflètent l'effort requis par le système de contrôle pour fournir l'accélération et la vitesse souhaitées en respectant les performances de précision et de dépassement transitoire. De grands gains de commande impliquent des actionneurs puissants qui augmentent le coût global du système. Les gains de commande sont définis pour chaque axe sous la forme de gains proportionnel (KP_1 et KP_2) et de gains dérivatif (KD_1 et KD_2).

Pour chaque axe on définit le schéma d'asservissement de la Figure 68 et les coefficients qui imposent un comportement dynamique correspondant à un système du second ordre.

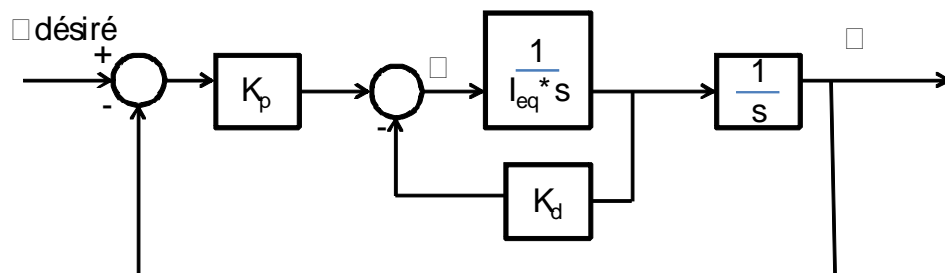


Figure 66: Asservissement de l'axe

En considérant un modèle dynamique linéaire pour chacun des axes, on peut alors calculer les coefficients de commande.

$\theta(s) = \frac{\tau(s)}{I_{eq}(s)}$ modèle linéaire de chaque axe où τ est le couple appliqué et s la variable de Laplace.

Les gains KP et KD assurant une réponse de type second ordre sont calculés selon :

$$KP = \omega_n * I_{eq}$$

$$KD = 2 * \xi * \Omega_{\omega_n} * I_{eq},$$

avec :

ω_n est la fréquence naturelle désirée (rd/s)

ξ est le coefficient d'amortissement désiré

I_{eq} est le moment d'inertie :

$$I_{eq1} = (m_2 + m_c) * L_2^2 + 2 * (m_2 + m_c) * L_1 * L_2 * \cos(\theta_2) + (m_1 + m_2 + m_c) * L_2^2$$

$$I_{eq2} = (m_2 + m_c) * L_2^2$$

- Minimiser le poids total du robot (Weight : W). Le poids impacte la puissance des moteurs et des composants électroniques de puissance qui entrent dans le prix du robot.

$$W = m_1 + m_2$$

5 Contraintes de conception

Les choix des concepteurs sont limités car ils doivent respecter les contraintes suivantes :

- Les longueurs des bras et la position de la base du robot sont limitées.
- La contrainte de flexion sur chaque segment doit être inférieure à la limite d'élasticité.
- Des valeurs d'acceptation sont fixées pour les objectifs du problème :

AR doit être inférieur à 2

La flèche doit être inférieure à 1 cm

KP et KD doivent être inférieurs à 10

Annexe C : Modèle du robot désempileur

1 Description

La conception d'un robot pour l'industrie de l'habillement entre dans le contexte d'une automatisation de ses processus. Un des problèmes importants de cette automatisation est la saisie des pièces textile. En effet, la diversité de ces pièces, leurs tailles et leurs matériaux varient très souvent, à cause des effets de mode, par exemple...

Parmi les nombreux systèmes de préhension existants, nous nous sommes intéressés au modèle proposé par Rakotoarisoa [Rakotoarisoa'93]. La Figure 67 illustre une partie du prototype imaginé, conçu et fabriqué.

La partie du prototype concerne la phase de préhension (séparation du tissu). Les autres phases sont celles de vision et de dépose. Elles permettent, après la phase de séparation, de placer le préhenseur (Gripper Device) en bonne position et de déposer le tissu.

Le prototype est composé de trois axes. Le premier est un axe de translation horizontale du chariot transportant le préhenseur. Le deuxième est son axe de rotation. Le troisième est l'axe de translation verticale de la table de préhension. Son mouvement est coordonné avec les autres axes afin d'assurer un contact entre la le préhenseur et le tissu quelque soit la hauteur de la pile.

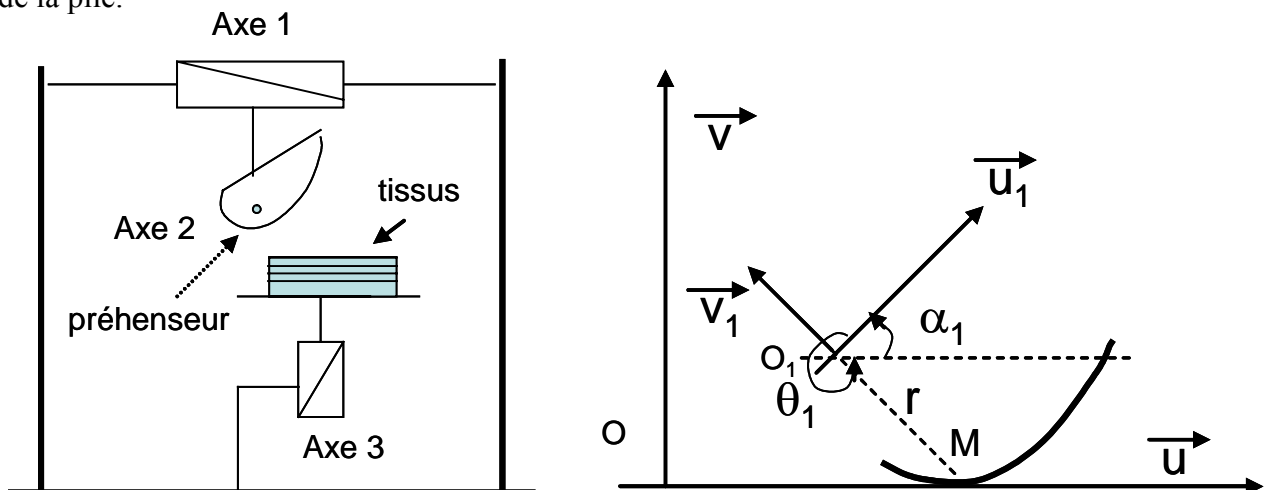


Figure 68: Description du robot en phase de préhension

- R repère (O, u, v) lié à la table de préhension
- O origine du repère lié à la table de préhension
- R₁ repère mobile (O, u₁, v₁) lié au préhenseur
- O₁ origine du repère mobile lié au préhenseur
- M point de contact du préhenseur sur la pile de tissu
- r rayon de courbure au point de contact
- α₁ angle de rotation du repère R₁ par rapport au repère R
- θ₁ angle de rotation du repère u₁ et O₁M
- X₁, Y₁ coordonnées du point O₁ dans R
- X, Y composantes du vecteur OM dans le repère R

On pose $\lambda(t) = \mathbf{X}(t)$ position du point M dans le repère R

Pour assurer un roulement sans glissement du préhenseur (condition obtenue si pour r et θ₁ fixés la vitesse du point M est nulle) on peut choisir pour forme du préhenseur une spirale exponentielle.

On a alors : $r(\theta_1) = R_0 \cdot \exp(\varphi \cdot \theta_1)$ avec R₀ rayon de base de la spirale avec $\theta_1 + \alpha_1 = \text{cste} = \text{Arctg}(-1/\varphi) + k\pi$

2 Modèle cinématique

Soient q_i les coordonnées généralisées des 3 premiers axes :

$$q_1 = X_1$$

$$q_2 = \theta_1$$

$$q_3 = Y_1$$

Le modèle cinématique s'écrit pour chacun des axes

Axe 1 : déplacement linéaire

$$q_1(t) = \sin^2(\theta_1 + \alpha_1) * \lambda(t) - R * \cos(\theta_1 + \alpha_1)$$

$$\frac{dq_1(t)}{dt} = \sin^2(\theta_1 + \alpha_1) * \dot{\lambda}(t)$$

$$\frac{d^2q_1(t)}{dt^2} = \sin^2(\theta_1 + \alpha_1) * \ddot{\lambda}(t)$$

Axe 2 : déplacement angulaire

$$q_2(t) = \frac{1}{\varphi} \cdot \left[\text{Ln} \left(\frac{\cos(\theta_1 + \alpha_1)}{R} * \lambda(t) + 1 \right) \right]$$

$$\frac{dq_2(t)}{dt} = \frac{-\text{tg}(\theta_1 + \alpha_1)}{\left[\lambda(t) + \frac{R}{\cos(\theta_1 + \alpha_1)} \right]} * \dot{\lambda}(t)$$

$$\frac{d^2q_2(t)}{dt^2} = \frac{\text{tg}(\theta_1 + \alpha_1) * \dot{\lambda}^2(t)}{\left[\lambda(t) + \frac{R}{\cos(\theta_1 + \alpha_1)} \right]^2} - \frac{\text{tg}(\theta_1 + \alpha_1)}{\left[\lambda(t) + \frac{R}{\cos(\theta_1 + \alpha_1)} \right]} * \ddot{\lambda}(t)$$

Axe 3 : déplacement linéaire

$$q_3(t) = -\sin(\theta_1 + \alpha_1) [\cos(\theta_1 + \alpha_1) * \lambda(t) + R]$$

$$\frac{dq_3(t)}{dt} = -\sin(\theta_1 + \alpha_1) * \cos(\theta_1 + \alpha_1) * \dot{\lambda}(t)$$

$$\frac{d^2q_3}{dt^2} = -\sin(\theta_1 + \alpha_1) * \cos(\theta_1 + \alpha_1) * \ddot{\lambda}(t)$$

3 Modèle dynamique

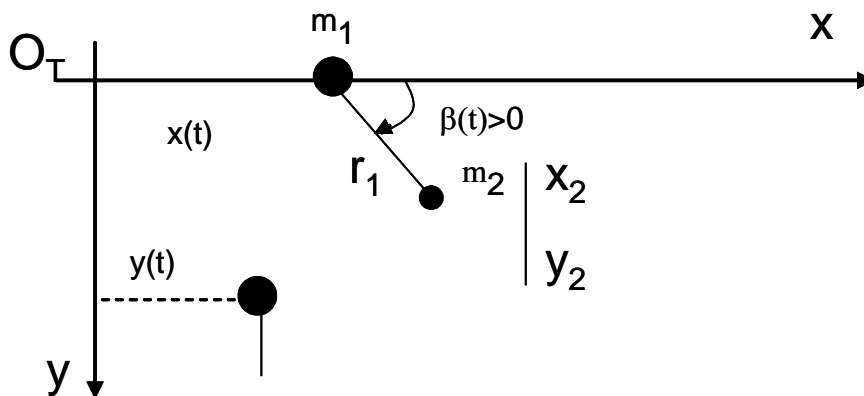


Figure 69: Description du robot pour le modèle dynamique

Hypothèse : Les corps 1,2, 3 sont représentés par leur masse m.

Soit :

r_1 : distance entre l'axe de rotation et le centre de gravité du corps 2.

J : inertie du préhenseur par rapport à l'axe de rotation

g : accélération de l'apesanteur

Γ : le vecteur des efforts généralisés (forces et couple),

on a :

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ C_2 \\ \Gamma_3 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 + I_{a1} & -m_2 * r_1 * \sin\beta & 0 \\ -m_2 * r_1 * \sin\beta & J + I_{a2} & 0 \\ 0 & 0 & m_3 + I_{a3} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\beta} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 & -m_2 * r_1 * \cos\beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}^2 \\ \dot{\beta}^2 \\ \dot{y}^2 \end{bmatrix} + \begin{bmatrix} 0 \\ -m_2 * g * r_1 * \cos\beta \\ -m_3 * g \end{bmatrix} + \begin{bmatrix} f_1 * \dot{x} \\ f_2 * \dot{\beta} \\ f_3 * \dot{y} \end{bmatrix}$$

Les termes I_{ai} sont introduits pour prendre en compte les inerties des actionneurs.

f_i : coefficient de frottement ramené à l'arbre moteur

4 Modélisation des ensembles variateur-actionneur

Pour chacun des axes $i = 1, 2, 3$:

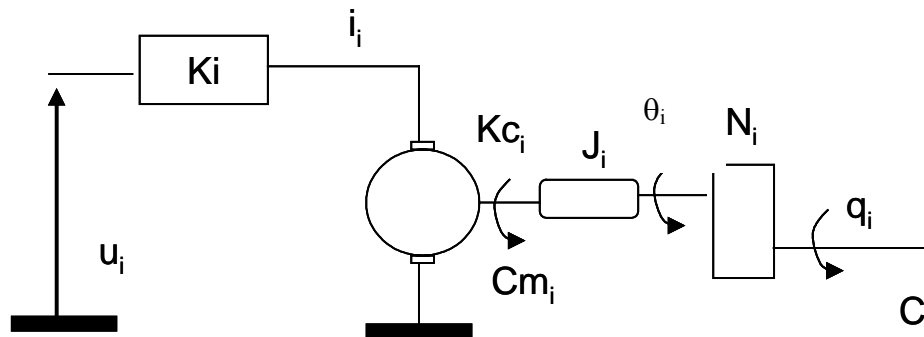


Figure 70: Modélisation de l'ensemble variateur-actionneur

u : tension de commande de l'actionneur

K : gain variateur

i : courant de l'actionneur

C_m : couple de commande de l'actionneur

K_c : constante de couple

J : inertie propre de l'actionneur augmentée de l'inertie des organes de transmission

θ : position angulaire de l'actionneur

N : rapport de réduction

C : couple transmis

q : coordonnées généralisées

$I_{ai} = (J_{Mi} + J_{Ri}) * N_i^2$ où J_{Mi} et J_{Ri} sont respectivement les inerties du rotor du moteur i et du réducteur i .

On a $C * dq/dt = C_m * d\theta/dt$

Et $C = C_m * N$

5 Modélisation des liaisons

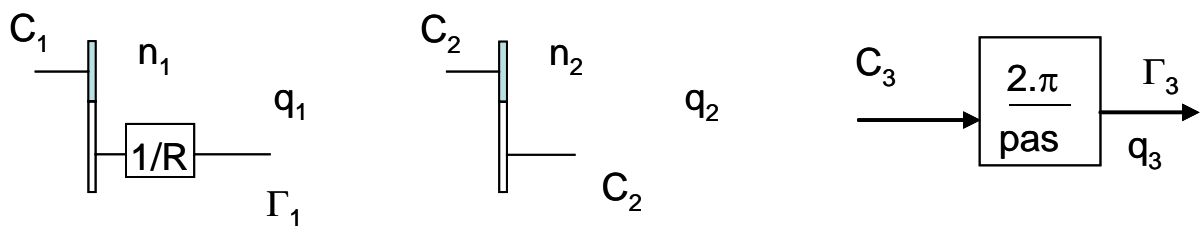


Figure 72: Modélisation des liaisons

Première liaison

Seconde liaison

Troisième liaison

Poulie+courroie

Vis à bille

$$N_1 = n_1/R$$

$$N_2 = n_2$$

$$N_3 = 2\pi/\text{pas}$$

6 Données

$5\text{kg} < m_1 < 25\text{kg}$ dépend de la structure et du choix de matériau du chariot de l'axe 1

$$m_2 = 9 \text{ kg}$$

$$m_3 = 34 \text{ kg}$$

pas de la vis à bille : 0.02m/tour

rayon de la poulie 0,05 m

$$f_1 = 0.17 \text{ N.m.s}$$

$$f_2 = 0.8 \text{ N.m.s}$$

$$f_3 = 3.14 \text{ N.m.s}$$

$$\varphi = 1$$

$$R_0 = 0,06 \text{ m}$$

$$r_1 = 0,1 \text{ m}$$
