



HAL
open science

Optimisation de la maintenance par la conception robuste de la modularité: Application au torréfacteur de soja

Aslain Brisco Ngnassi Djami

► To cite this version:

Aslain Brisco Ngnassi Djami. Optimisation de la maintenance par la conception robuste de la modularité: Application au torréfacteur de soja. Génie mécanique [physics.class-ph]. Université de Ngaoundéré (Cameroun), 2022. Français. NNT: . tel-03517219

HAL Id: tel-03517219

<https://theses.hal.science/tel-03517219v1>

Submitted on 7 Jan 2022

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.

UNIVERSITE DE NGAOUNDERE
ECOLE NATIONALE SUPERIEURE DES
SCIENCES AGRO – INDUSTRIELLES

THE UNIVERSITY OF NGAOUNDERE
NATIONAL SCHOOL OF AGRO-
INDUSTRIAL SCIENCES



DEPARTEMENT DE GENIE MECANIQUE
DEPARTMENT OF MECHANICAL ENGINEERING

Unité de Formation Doctorale - Physique Appliquée et Ingénierie
Optimisation de la maintenance par la conception robuste de la
modularité : Application au torrificateur de soja

THESE

Présentée et soutenue publiquement le 20 novembre 2021 en vue de l'obtention
du diplôme de **DOCTORAT/Ph.D.**

Spécialité : Ingénierie des Equipements et Productique

Par :

NGNASSI DJAMI Aslain Brisco

Master en Sciences et Technologies – Génie Mécanique et Productique

Matricule : 12M081EN

Jury :

Noms et Prénoms	Grade	Membre	Affiliation
M. KUITCHE Alexis	Professeur	Président	Université de Ngaoundéré
M. DOKA YAMIGNO Serge	Professeur	Directeur	Université de Ngaoundéré
M. NZIE Wolfgang	Maitre de Conférences	Directeur	Université de Ngaoundéré
M. NTAMACK Guy Edgar	Maitre de Conférences	Rapporteur	Université de Ngaoundéré
M. KENMEUGNE Bienvenu	Maitre de Conférences	Rapporteur	Université de Yaoundé I
M. NTENGA Richard	Maitre de Conférences	Examineur	Université de Ngaoundéré

Année 2021

DEDICACE

Je dédie ce modeste travail à la famille NGNASSI

REMERCIEMENTS

Cette étude a été menée au sein l'Unité de Formation Doctorale de Physique Appliquée et Ingénierie (UFD-PAI) de l'Ecole Nationale Supérieure des Sciences Agro-Industrielles (ENSAI) de l'Université de Ngaoundéré, Cameroun.

Je rends grâce au Seigneur DIEU pour m'avoir accordé la santé, la force et l'intelligence nécessaires pour l'accomplissement de ce travail. Que son nom soit loué !

Mes remerciements s'adressent également à plusieurs personnes qui ont rendu ce travail de recherche possible. Je pense :

À Monsieur DOKA YAMIGNO Serge, Professeur Titulaire à l'Université de Ngaoundéré et Monsieur NZIE Wolfgang, Maître de Conférences à l'Université de Ngaoundéré, tous deux directeurs de cette thèse. J'ai profité tout au long de mes années de thèse de leur vivacité intellectuelle, de leur très grande rigueur scientifique ;

Au Professeur BEDA Tibi, Responsable du laboratoire de Mécanique, Matériaux et Photonique (La₂MP), où mes travaux de recherche ont été réalisés ;

Au Professeur KUITCHE Alexis, Responsable de l'unité de formation doctorale PAI-ENSAI, pour m'avoir accueilli au sein de son école doctorale ;

Aux différents membres du jury, pour l'attention qu'ils ont portée à ce travail. Je vous en suis très reconnaissant et vous prie de recevoir ici l'expression de ma profonde déférence.

Il me serait difficile d'oublier de remercier :

Le Professeur NSO Emmanuel Jong, Directeur de l'ENSAI qui a permis que cette thèse soit soutenue dans son établissement ;

Le Docteur SAMON Jean Bosco, pour ses encouragements, son aide et ses précieux conseils dans cette recherche.

Seront également remerciés :

Mes collègues du Laboratoire La₂MP pour tout le soutien et les moments agréables passés ensemble ;

Quelques étudiants de l'IUT et de l'ENSAI, pour avoir également contribué au rendu de ce travail.

Je profite de cette occasion pour remercier particulièrement mon papa, Monsieur NGNASSI Jean Claude et ma maman, Madame NKWETCHEU Odette pour tous les sacrifices déployés jusqu'à ce jour pour que je puisse réaliser ce travail.

TABLE DES MATIERES

DEDICACE.....	i
REMERCIEMENTS.....	ii
TABLE DES MATIERES	iii
LISTE DES FIGURES.....	ix
LISTE DES TABLEAUX	xii
LISTE DES ABREVIATIONS	xiv
LISTE DES SYMBOLES ET LEURS UNITÉS DE MESURE.....	xv
RESUME.....	xvi
ABSTRACT	xvii
INTRODUCTION GENERALE	1
CHAPITRE I : ETAT DE L'ART ET PROBLEMATIQUE.....	4
I.1 OPTIMISATION DE LA MAINTENANCE	5
I.1.1 La maintenance et ses attributs	5
I.1.2 La fiabilité.....	5
I.1.2.1 Définition.....	5
I.1.2.2 Indicateurs de la fiabilité.....	6
I.1.2.3 Maintenance dans une approche fondée sur la fiabilité	7
I.1.3 La Disponibilité	10
I.1.3.1 Définition.....	10
I.1.3.2 Indicateurs (ou quantification) de la disponibilité.....	10
I.1.4 La maintenabilité	12
I.1.4.1 Définition.....	12
I.1.4.2 Indicateurs de la maintenabilité	13
I.1.4.3 Critères de maintenabilité	14
I.1.4.3.1 Critères contextuels.....	14

I.1.4.3.2 Critères intrinsèques	15
I.2 RECENSION DES TRAVAUX SUR LA MODULARITE	23
I.2.1 Définition	23
I.2.2 principes de construction d'un système modulaire	24
I.2.3 Fondements de l'architecture modulaire d'un système	26
I.2.3.1 L'approche système	27
I.2.3.1.1 Concepts de système, d'entité et d'interaction	27
I.2.3.1.2 Concept de vues	28
I.2.3.1.3 Concept d'architecture, de module et d'élément intégrateur	31
I.2.3.2 Conception modulaire des produits	33
I.2.3.2.1 Modularité et familles de produits	33
I.2.3.2.2 Avantages et inconvénients de la conception modulaire	35
I.2.3.2.3 Méthodes de modularisation du produit	36
I.2.3.2.4 Outils matriciels de représentation d'une architecture DSM	41
I.2.3.2.4.1 Classification des DSM	41
I.2.3.2.4.2 Synthèse sur les types de DSM	45
I.2.3.2.4.3 Les DSM numériques	46
I.2.3.2.5 Positionnement du problème de la modularité	50
I.3 ANALYSE BIBLIOGRAPHIQUE SUR LES METHODES DE CONCEPTION ROBUSTE	50
I.3.1 Les origines de la conception robuste	51
I.3.2 La conception robuste selon Taguchi	51
I.3.2.1 La fonction perte	51
I.3.2.2 Le ratio signal / bruit	52
I.3.3 Formulation du problème de conception robuste local	53
I.3.3.1 Les variables de conception	53
I.3.3.2 Les paramètres de conception environnementaux	53

I.3.3.3 Les fonctions performances.....	53
I.3.3.4 Exemples de formulation d'un problème de conception robuste	54
I.3.3.4.1 Exemple 1 : système de refroidissement d'une salle	54
I.3.3.4.2 Exemple 2 : mécanisme articulé	54
I.3.4 Définition d'une méthodologie de conception robust	55
I.3.4.1 Recensement des incertitudes	55
I.3.4.1.1 Différents types d'incertitudes	55
I.3.4.1.2 Modélisation des incertitudes.....	57
I.3.4.2 Méthodologie de propagation des incertitudes.....	57
I.3.4.2.1 Méthode de Monte-Carlo sur le code de calculs.....	58
I.3.4.2.2 Méthode de Monte-Carlo sur un modèle local	60
I.3.4.2.3 Méthode FOSM	61
I.3.4.2.4 Méthode URQ.....	62
I.3.4.3 Méthodes de conception robuste pour la résolution des problèmes d'optimisation	64
I.3.4.3.1 Méthodes trouvant un compromis entre les objectifs.....	65
I.3.4.3.2 Méthodes établissant le front de Pareto	66
I.3.4.4 Etude comparative des méthodes établissant le front de Pareto et des méthodes recherchant un compromis entre les objectifs	73
I.4 PROBLEMATIQUE DE LA THESE	74
I.5 METHODOLOGIE	75
CHAPITRE II : CHOIX DE LA MEILLEURE SOLUTION DE CONCEPTION D'UN SYSTEME AU MOYEN D'UN INDICE-MULTI PARAMETRES	76
II.1 INTRODUCTION	76
II.2. METHODOLOGIE D'EVALUATION DE LA COMPLEXITE DE DESASSEMBLAGE.....	77
II.2.1 Choix des paramètres de désassemblage	78
II.2.1.1 Paramètres susceptibles de représenter les contraintes de désassemblage	78

II.2.1.2 Paramètres représentant le désassemblage non destructif	82
II.2.1.2.1 Types de contacts	83
II.2.1.2.2 Types de combinaisons	85
II.2.1.2.3 Outils de désassemblage	85
II.2.1.2.4 Directions de désassemblage	86
II.2.1.2.5 Qualification des opérateurs	86
II.2.1.2.6 Équipements requis	87
II.2.1.3 Pondération des paramètres de désassemblage	87
II.2.2 Identification des différents composants du système	88
II.2.3 Graphes des liaisons	89
II.2.4 Possibilités de solutions	90
II.2.5 Proposition d'un indicateur d'évaluation de la solution de conception	90
II.3 RESULTATS	95
II.3.1. Identification des composants du système	95
II.3.1.1 FAST du torréfacteur de soja	95
II.3.1.2 Vue éclatée du torréfacteur de soja	96
II.3.2. Modélisation du produit par le graphe de liaisons	98
II.3.3 Evaluation de la complexité de désassemblage	100
II.4 CONCLUSION	102
CHAPITRE III : PROPOSITION D'UN OUTIL QUI S'APPUIE SUR	
L'ALGORITHME DE CLUSTERING	103
III.1 INTRODUCTION	103
III.2 METHODOLOGIE DE L'ALGORITHME DE CLUSTERING EXISTANT ..	103
III.2.1 Modélisation du produit	103
III.2.2 Première étape de l'algorithme de clustering existant et choix d'une bonne topologie	104
III.2.2.1 Première étape de l'algorithme de clustering existant	104
III.2.2.2 Choix d'une bonne topologie des modules	105

III.2.3 Deuxième étape de l’algorithme de clustering existant	109
III.2.4 Troisième étape de l’algorithme de clustering existant	112
III.2.4.1 La fonction d’enchère	112
III.2.4.2 La fonction Coût	113
III.2.4.3 Le recuit simulé	114
III.3 LIMITES DE L’ALGORITHME DE CLUSTERING EXISTANT	115
III.4 NOUVEL ALGORITHME DE CLUSTERING	117
III.4.1 Matrice structurelle de conception modifiée	119
III.4.2 Coefficients correcteurs du coût de couplage des modules	119
III.4.3 La fonction Coût	121
III.4.4 Identification des éléments intégrateurs	123
III.5 CONCLUSION	123
CHAPITRE IV : VALIDATION DE L’ALGORITHME PROPOSE	125
IV.1 INTRODUCTION	125
IV.2 RESULTATS DE L’ALGORITHME DE CLUSTERING EXISTANT	125
IV.2.1 Modélisation du produit	125
IV.2.2 Première étape de l’algorithme de clustering existant et topologie des modules	127
IV.2.2.1 Première étape de l’algorithme de clustering existant	127
IV.2.2.2 Choix d’une bonne topologie des modules	132
IV.2.2.2.1 Matrice de précedence des composants	132
IV.2.2.2.2 Changement de direction de désassemblage des modules	135
IV.2.2.2.3 Matrice de précedence des modules et Gamme de désassemblage	138
IV.2.2.2.4 Topologie des modules du torréfacteur de soja	143
IV.2.3 Deuxième étape de l’algorithme de clustering existant	144
IV.2.3.1 Matrice structurelle de conception	144
IV.2.3.2 Quantification de la modularité	148

IV.2.4 Troisième étape de l’algorithme de clustering existant	153
IV.3 RESULTATS DE L’ALGORITHME PROPOSE ET DISCUSSION	158
IV.3.1 Matrice structurelle de conception	158
IV.3.2 Coefficients correcteurs	161
IV.3.3 Coût de couplage	166
IV.3.4 Identification des éléments intégrateurs du torrificateur de soja	170
IV.3.5 Etude comparative du coût de couplage en fonction du nombre de modules .	175
CONCLUSION	176
CONCLUSION GENERALE ET PERSPECTIVES.....	178
REFERENCES BIBLIORAPHIQUES	180
ANNEXES.....	196
<i>ANNEXE 1 : Schémas côté de la cuvette</i>	<i>196</i>
<i>ANNEXE 2 : Schémas côté de l’axe du cylindre.....</i>	<i>197</i>
<i>ANNEXE 3 : Schémas côté du cylindre principal.....</i>	<i>198</i>
<i>ANNEXE 4 : Schémas côté de la trémie d’entrée</i>	<i>199</i>
<i>ANNEXE 5 : Schémas côté de la trémie de sortie</i>	<i>200</i>
<i>ANNEXE 6 : Torrificateur de soja</i>	<i>201</i>
<i>ANNEXE 7 : Torrificateur de soja reconfiguré.....</i>	<i>202</i>
<i>ANNEXE 8 : Quelques publications</i>	<i>203</i>

LISTE DES FIGURES

Figure 1: Mise en place d'une boucle d'amélioration et d'un retour d'expérience (Cotaina et al., 2000).....	7
Figure 2: Méthodes de planification de démontage (Shana et Hung, 2015)	18
Figure 3 : Visualisation de la modularité	24
Figure 4 : Allocation des fonctions aux constituants (Meinadier, 2002)	32
Figure 5 : Heuristiques pour l'architecture des fonctions	37
Figure 6 : Matrice Indication Module	40
Figure 7: Classification des DSM	41
Figure 8 : Exemple de DSM temporelle.....	43
Figure 9 : Deux résultats de partitionnement d'une DSM.....	44
Figure 10 : Exemple de numérisation des DSM temporelles (Eppinger et al., 1994).....	47
Figure 11 : Exemple de DSM Produit (Pimmler et al., 1994).....	49
Figure 12 : Fonctions perte « double échelons » et « quadratique »	52
Figure 13 : Système bielle-manivelle	55
Figure 14: Schéma de la "méthodologie de propagation d'incertitudes" (de Rocquigny et al., 2008).....	58
Figure 15 : Propagation des incertitudes par simulations de Monte-Carlo sur le code de calculs.....	59
Figure 16 : Propagation des incertitudes par simulations de Monte-Carlo sur un modèle ajusté localement	61
Figure 17 : Propagation des incertitudes par la méthode FOSM	62
Figure 18 : Illustration de la méthode d'intersection de limites normales	67
Figure 19 : Illustration de la méthode à objectifs pondérés	68
Figure 20 : Illustration de la méthode génétique de tri basée sur la non-dominance	70
Figure 21 : Illustration du Clustering en dimension 2.....	71
Figure 22 : Synthèse des méthodes de conception robuste pour la résolution d'un problème d'optimisation.....	72
Figure 23 : Description de la méthodologie	78
Figure 24: Structure d'un diagramme Pieuvre	88
Figure 25: Structure d'un diagramme FAST.....	89
Figure 26: Graphe des liaisons hybride (Li et al., 2002).....	90
Figure 27: Diagramme pieuvre du torréfacteur de soja en phase d'utilisation	95

Figure 28: Le FAST relative à la fonction torr�fier.	96
Figure 29: Vue �clat�e de la solution.	97
Figure 30: Grphe des liaisons de du torr�facteur de soja.....	99
Figure 31: Exemple de matrice de pr�c�dence (Sa�id, 2018).....	107
Figure 32: Matrice de pr�c�dence modifi�e des composants (Sa�id, 2018)	108
Figure 33: Organigramme du nouvel algorithme de clustering	118
Figure 34: Regroupement en quatre modules du torr�facteur de soja.....	130
Figure 35: Regroupement en trois modules du torr�facteur de soja.....	131
Figure 36: Regroupement en deux modules du torr�facteur de soja	132
Figure 37: Matrice de pr�c�dence des composants du torr�facteur de soja � quatre modules.	133
Figure 38: Matrice de pr�c�dence des composants du torr�facteur de soja � trois modules..	134
Figure 39: Matrice de pr�c�dence des composants du torr�facteur de soja � deux modules.	135
Figure 40: Matrice de pr�c�dence modifi�e du torr�facteur de soja � quatre modules.....	136
Figure 41: Matrice de pr�c�dence modifi�e du torr�facteur de soja � trois modules.....	137
Figure 42: Matrice de pr�c�dence modifi�e du torr�facteur de soja � deux modules.	138
Figure 43: Matrice de pr�c�dence des quatre modules du torr�facteur.....	139
Figure 44: Matrice de pr�c�dence des trois modules du torr�facteur.....	139
Figure 45: Matrice de pr�c�dence des deux modules du torr�facteur	139
Figure 46: Gamme de d�sassemblage compl�te du torr�facteur de soja � quatre modules. ..	140
Figure 47: Gamme de d�sassemblage compl�te du torr�facteur de soja � trois modules.	141
Figure 48: Gamme de d�sassemblage compl�te du torr�facteur de soja � deux modules.....	142
Figure 49: Topologie des modules du torr�facteur de soja (pour quatre, trois et deux modules)	143
Figure 50: DSM du torr�facteur de soja � quatre modules.....	144
Figure 51: DSM du torr�facteur de soja � trois modules	146
Figure 52: DSM du torr�facteur de soja � deux modules.....	147
Figure 53: Evolution du MSI des quatre modules avec l'algorithme existant	151
Figure 54: Evolution du MSI des trois modules avec l'algorithme existant	152
Figure 55: Evolution du MSI des deux modules avec l'algorithme existant	153
Figure 56: Evolution du co�t des quatre modules avec l'algorithme existant	156
Figure 57: Evolution du co�t des trois modules avec l'algorithme existant	157
Figure 58: Evolution du co�t des deux modules avec l'algorithme existant.....	158
Figure 59: DSM num�rique du torr�facteur de soja � quatre modules.....	159

Figure 60: DSM numérique du torréfacteur de soja à trois modules	160
Figure 61: DSM numérique du torréfacteur de soja à deux modules.....	161
Figure 62: Etude comparative du MSI corrigé du torréfacteur à quatre modules et de celui obtenu avec l’algorithme existant	163
Figure 63: Etude comparative du MSI corrigé du torréfacteur à trois modules et de celui obtenu avec l’algorithme existant	165
Figure 64: Evolution du coût de couplage corrigé en fonction de la taille des quatre modules	169
Figure 65: Architecture du torréfacteur de soja avec les éléments intégrateurs.....	172
Figure 66: Correction du MSI avec et sans les éléments intégrateurs.....	173

LISTE DES TABLEAUX

Tableau 1 : Critères contextuels (Menye, 2009)	14
Tableau 2 : Synthèse des principes de constructions modulaires.....	26
Tableau 3 : Les critères de modularité dans la méthode MFD	38
Tableau 4 : Taxonomie des interactions du Produit (Pimmler et Eppinger, 1994).....	44
Tableau 5 : Taxonomie des interactions entre acteurs (Sosa et al., 2004)	45
Tableau 6 : Récapitulatif des caractéristiques des DSM.....	45
Tableau 7 : Métrique pour les DSM Produit (Pimmler et Eppinger, 1994).....	49
Tableau 8 : Les paramètres influençant le désassemblage et les références correspondantes	81
Tableau 9 : Zones de contacts, liaisons associées et efforts transmissibles.	83
Tableau 10 :Types de contacts.	84
Tableau 11 :Type de combinaisons.....	85
Tableau 12 :Outils de désassemblage.	85
Tableau 13 :Nombre de directions de désassemblage.....	86
Tableau 14 :Qualifications des opérateurs	86
Tableau 15 :Équipements requis.	87
Tableau 16 :Pondérations des paramètres de désassemblage.	88
Tableau 17 :Seuils de satisfaction de l'indicateur de fixations faciles.....	90
Tableau 18 :Exemple des coefficients associés aux nombres de directions d'accès	93
Tableau 19 :Exemple des coefficients associés aux outils de désassemblage	93
Tableau 20 :Exemple des coefficients associés aux types de contacts	93
Tableau 21 :Exemple des coefficients associés à aux types de combinaisons.....	94
Tableau 22 :Exemple des coefficients associés à la qualification des opérateurs.....	94
Tableau 23 :Exemple des coefficients associés aux équipements requis.....	94
Tableau 24 :Fonctions de services.	95
Tableau 25 :Légende des composants.....	97
Tableau 26 :Evaluation des liaisons entre les composants avec le nouveau modèle.	100
Tableau 27 :Bilan de l'évaluation avec le nouveau modèle.....	101
Tableau 28 :Evaluation des liaisons entre les composants avec l'ancien modèle.....	101
Tableau 29 :Bilan de l'évaluation avec l'ancien modèle	102
Tableau 30 :Matrice de précedence pour les modules (Saïd, 2018)	108
Tableau 31 :Modifications de la matrice structurelle de conception	119
Tableau 32 :Matrice de relations du torréfacteur de soja.....	126

Tableau 33: Première transformation de la matrice de relations.....	128
Tableau 34: Deuxième transformation de la matrice de relations.....	128
Tableau 35: Troisième transformation de la matrice de relations	129
Tableau 36: Codification des composants du torréfacteur à quatre modules.....	145
Tableau 37: Nouvelle Codification des composants du torréfacteur à trois ou à deux modules	148
Tableau 38: Degré de modularité des quatre modules du torréfacteur avec l’algorithme existant	150
Tableau 39: Degré de modularité des trois modules du torréfacteur avec l’algorithme existant	151
Tableau 40: Degré de modularité des deux modules du torréfacteur avec l’algorithme existant	152
Tableau 41: Coût de couplage des quatre modules du torréfacteur avec l’algorithme existant	155
Tableau 42: Coût de couplage des trois modules du torréfacteur avec l’algorithme existant	157
Tableau 43: Coût de couplage des deux modules du torréfacteur avec l’algorithme existant	157
Tableau 44: Coefficients correcteurs du coût de couplage du torréfacteur à quatre modules	163
Tableau 45: Coefficients correcteurs du coût de couplage du torréfacteur à trois modules ..	164
Tableau 46: Coefficients correcteurs du coût de couplage du torréfacteur à deux modules..	165
Tableau 47: Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les quatre modules du torréfacteur	168
Tableau 48: Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les trois modules du torréfacteur	170
Tableau 49: Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les deux modules du torréfacteur	170
Tableau 50: Coefficients correcteur du torréfacteur de soja avec prise en compte des éléments intégrateurs	172
Tableau 51: Coûts de couplage corrigé du torréfacteur de soja avec prise en compte des éléments intégrateurs.....	174
Tableau 52: Synthèse des coûts générés par l’algorithme de clustering proposé avec et sans éléments intégrateurs.....	174
Tableau 53: Comparaison des configurations obtenues avant et après implémentation de l’algorithme de clustering amélioré.....	175
Tableau 54: Variation du coût de couplage du torréfacteur avec le nombre de modules	176

LISTE DES ABREVIATIONS

- Add**: Arbre de Défaillance
- AFNOR** : Association Française de Normalisation
- AMDE** : Analyse des Modes de Défaillance et leurs Effets
- AMDEC** : Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité
- BD** : Bloc Diagramm
- CA** : Component Accessibility
- CAO** : Conception Assistée par Ordinateur
- DSM** : Design Structure Matrix
- FAST**: Function Analysis System Technique
- FOSM**: First Order Second Moment
- IC**: Indice de Couplage
- IEC**: International Electrotechnique Commission
- IEEE**: Institute of Electrical and Electronics Engineers
- If_f**: Indice de Fixations Faciles
- MFD**: Modular Function Deployment
- MIM** : Module Indication Matrix
- MMT** : Temps Moyen des Actions Préventives ou Correctives
- MSI**: Module Strength Indicator
- MTBF**: Mean Time Between Failure
- MTBM** : Mean Time Between Maintenance
- MTI** : Moyenne des Temps d'Indisponibilité
- MTL** : Moyenne des Temps Logistique
- MTTR** : Moyenne des Temps Techniques de Réparation
- NASA** : National Aeronautics and Space Administration
- NF** : Norme Française
- OMF** : Optimisation de la Maintenance par la Fiabilité
- QFD**: Quality Function Deployment
- RCM**: Reliability Centered Maintenance
- RT** : « Ready Time » ou Temps Moyens d'Attente
- SMI**: Singular Value Modularity Index
- TPM**: Total Productive Maintenance
- URQ** : Univariate Reduced Quadrature

LISTE DES SYMBOLES ET LEURS UNITÉS DE MESURE

A_i : facteur d'accessibilité du composant C_i ,

A_{ji} : facteur d'accessibilité du connecteur de type j fixant le composant C_i ,

d_i : distance de déplacement du composant C_i pour le démonter de l'ensemble,

DC : coût de démontage,

F_i : facteur de force du composant C_i ,

$f(t)$: densité de probabilité de temps avant défaillance ,

$g(t)$: densité de réparation en fonction du temps (t),

K_0 : facteur constant tenant compte de l'espace, l'outillage, etc.,

K_s : facteur constant tenant compte de la sécurité du composant,

K_t : temps standard pour déplacer un composant de poids (masse) inférieure à K_w à une distance d_i ,

K_w : poids (masse) standard limite que peut soulever un opérateur source (lifting source),

LR : coût horaire,

$M(t)$: maintenabilité instantané,

$N_{i,j}$: nombre de connecteurs de type j fixant le composant C_i ,

O_i : facteur d'orientation du composant C_i ,

O_{ji} : facteur d'orientation du connecteur de type j fixant le composant C_i ,

$R(t)$: fiabilité instantané,

T_j : temps standard de démontage d'un connecteur de type j fixant C_i ,

W_i : poids (masse) du composant C_i ,

$\lambda(t)$: taux de défaillance(pannes /heure),

$\mu(t)$: taux de réparation (réparations /heure).

RESUME

Pour un équipement donné, la connaissance de son architecture modulaire présente l'avantage de faciliter sa maintenance. En effet, lors d'un problème relevant de la maintenance, on n'agit pas sur tout l'équipement, sauf sur le module défaillant et on gagne aussi en temps pour détecter le défaut observé. Cependant, la définition d'une bonne architecture est conditionnée par un choix judicieux des paramètres de conceptions, plus spécifiquement des paramètres de désassemblage du fait que lors la phase initiale de conception, on intègre de plus en plus les contraintes de désassemblage pour garantir une bonne désassemblabilité des équipements pour des fins de maintenance. Suite aux difficultés majeures de maintenance des systèmes de topologies complexes, la conception modulaire est donc un atout pour les entreprises. D'autre part, la conception modulaire bien qu'elle présente des avantages de standardisation et de reconfigurabilité est énormément onéreuse, ce qui nécessite plus de ressources, plus de temps et plus de travail des concepteurs. Cela engendre des conséquences sur la maintenance des équipements, car plus les modules de l'équipement sont complexes ou plus denses, plus la maintenance est couteuse. Il est donc question dans cette thèse d'améliorer la maintenance par la conception de la modularité. L'atteinte de cet objectif passe par une méthodologie qui consiste d'abord à évaluer la meilleure solution de conception pour apprécier la pertinence des contraintes de conceptions choisies, ensuite développer un outil qui s'appuie sur l'algorithme de clustering existant , en ayant au préalable présenté les limites de l'algorithme de clustering existant et enfin valider l'outil proposé en observant une réduction en coûts de couplage des modules pour garantir une bonne maintenance et déduire une reconfiguration de l'équipement.

Mots-clés : Conception robuste, Optimisation, Maintenance, Modularité

ABSTRACT

For a given piece of equipment, knowing its modular architecture has the advantage of facilitating its maintenance. Indeed, during a maintenance problem, we do not act on all the equipment, except on the faulty module and we also gain time to detect the observed fault. However, the definition of a good architecture is conditioned by a judicious choice of the design parameters, more specifically of the disassembly parameters because during the initial design phase, the constraints of disassembly are increasingly integrated to guarantee a good disassembly of equipment for maintenance purposes. Following the major difficulties in maintaining complex topology systems, the modular design is therefore an asset for companies. On the other hand, the modular design although it has advantages of standardization and reconfigurability is enormously expensive, which requires more resources, more time and more work of the designers. This has consequences for the maintenance of the equipment, because the more complex or denser the modules of the equipment, the more expensive the maintenance is. This thesis therefore aims to improve maintenance by designing modularity. Achieving this objective requires a methodology which consists first of evaluating the best design solution to assess the relevance of the design constraints chosen, then developing a tool which is based on the existing clustering algorithm, by having first presented the limits of the existing clustering algorithm and finally validate the proposed tool by observing a reduction in module coupling costs to guarantee good maintenance and deduce reconfiguration of the equipment.

Keywords: Robust design, Optimization, Maintenance, Modularity

INTRODUCTION GENERALE

La préoccupation majeure des industries manufacturières, il y a quelques années encore, résidait essentiellement dans l'optimisation du triptyque coût/qualité/délai. Aujourd'hui, pour rester compétitives, les entreprises doivent s'adapter aux changements qui s'opèrent dans le marché où elles évoluent. Ce marché est caractérisé par une demande de produits (systèmes) de plus en plus personnalisés, de qualité sans cesse croissante, disponibles en un minimum de temps, aux moindres coûts et respectueux de l'environnement. On comprend, dès cet instant, que toute cause d'arrêt du système de production due par exemple à un problème de maintenance est mal perçue et peut être à l'origine de tensions entre acteurs de production et chefs d'entreprises. C'est dans ce contexte que s'est développée ces dernières années une approche par la modularité pour certaines structures industrielles. Grâce aux stratégies de la conception modulaire, les entreprises proposent des produits issus de l'assemblage d'éléments standards et spécifiques, permettant ainsi d'obtenir des produits personnalisés. Ces stratégies constituent un bon compromis entre la pratique de la fabrication sur mesure et la standardisation de produits et processus (Jose et Tollenaere, 2005). La fabrication sur mesure vise à la satisfaction stricte de chaque besoin, mais conduit à une augmentation des coûts de production. Quant à la standardisation, qui vise à proposer un produit unique permettant de satisfaire un ensemble de besoins. Elle permet de diminuer le nombre de références à gérer et d'augmenter les quantités de chaque composant avec les conséquences de réduction de la complexité dans le système de fabrication. Elle présente l'inconvénient de proposer des produits surdimensionnés. En effet, aux composants communs utilisés dans la fabrication des produits divers, sont ajoutés des fonctions pas toujours utiles.

Un outil d'aide à la conception de modules industriels (pour les faisceaux électriques automobiles) a été proposé dans (Agard et Tollenaere, 2002). Il permet d'assurer une diversité fonctionnelle totale des produits finis, en s'appuyant à la fois sur la description du produit et les processus de production. Un compromis est alors recherché entre la taille des modules à réaliser et les coûts de production pour l'assemblage final des produits sous une contrainte de temps d'assemblage final liée à une livraison synchrone.

Optimiser la maintenance des systèmes de production reviendrait donc à optimiser la modularité des composants de ses derniers. En effet, vu la topologie des systèmes de plus en plus complexes, et vu ce changement de technologie qui est sans cesse croissant au fil du temps, il est opportun pour des concepteurs de mettre sur pieds des systèmes facilement modulables et

à moindre coût de couplage des modules, de façon à ce qu'à la survenance d'un dysfonctionnement relevant d'un problème de maintenance, on ne soit plus contraint d'agir sur tout le système si ce n'est sur le module défectueux (à moindre temps de désassemblage ou d'assemblage des composants du module concerné), et ceci afin de ne pas altérer la fonction principale du système de production.

L'optimisation de la modularité est fortement corrélée à l'amélioration de la maintenance des produits au cours de leur utilisation. En effet, plus les sous-ensembles (modules) des produits (systèmes) sont de grandes tailles, donc plus complexes, plus ils nécessitent de contraintes et de ressources lors d'une opération de maintenance. Ce qui rend ainsi difficile la maintenance. Dans ce contexte, les chercheurs sont amenés à développer des méthodes de conception robustes permettant d'augmenter le degré de modularité à l'intérieur des modules et de réduire le coût de couplage des modules. Ainsi, ce projet de recherche intitulé "Optimisation de la maintenance par la conception robuste de la modularité : Applications " évoque la **problématique** de difficultés de maintenance à cause d'une mauvaise modularité.

Dans cet optique, l'**objectif principal** de ce travail de recherche est d' « *Améliorer la maintenance par la conception de la modularité* ».

Ainsi, cet objectif à poursuivre se décline en trois **objectifs spécifiques** :

- ✓ Evaluer la solution de conception la plus respectueuse des contraintes de désassemblage ;
- ✓ Proposer un outil qui améliore au mieux la modularité ;
- ✓ Valider l'outil proposé par une application.

Les **hypothèses formulées** sont données comme suit :

- **H1** : Les éléments appartenant à un module interagissent faiblement avec les éléments appartenant à d'autres modules ;
- **H2** : Les interactions d'un module sont une fonction décroissante de la taille des modules ;
- **H3** : La prise en compte des éléments intégrateurs améliore la modularité.

La démarche à suivre pour atteindre les objectifs est donnée comme suit :

- ✚ Faire un état de l'art sur les méthodes de conception robustes ;

- ✚ Identifier et évaluer les paramètres de désassemblage qui conduisent à une solution de conception satisfaisante ;
- ✚ Présenter les fonctionnalités de la méthode de conception robuste sur laquelle on s'appuie, et en dégager les limites.

Sommairement, le présent mémoire retraçant nos recherches est composé de quatre chapitres articulés comme suit :

Un *état de l'art*, subdivisé en trois parties, dont la première traite de l'optimisation de la maintenance pour nous conduire finalement à la présentation des critères de maintenabilité contextuels et intrinsèques. Dans la seconde partie, nous nous intéressons à la branche "critères de maintenabilité intrinsèques" pour ensuite nous focaliser sur la modularité, afin d'exposer les travaux scientifiques qui traitent de la modularité. La dernière partie quand-à-elle, fait une analyse bibliographique sur les méthodes de conception robuste pour la résolution des problèmes d'optimisation, afin de choisir judicieusement celle à utiliser et à améliorer dans le cadre de ce travail de recherche. Nous dégagons ensuite la problématique issue de la limite des travaux de la littérature.

Une *démarche d'évaluation de la meilleure solution de conception d'un produit*, au moyen d'un indice-multi paramètres qui tient compte des paramètres de désassemblages et une application.

Une *présentation des fonctionnalités de l'algorithme de clustering*, permettant la définition d'une architecture modulaire des produits, d'évaluer le degré de modularité et de calculer les coûts de couplage des modules dès la phase préliminaire du processus de conception et le *développement d'un outil qui s'appuie sur l'algorithme de clustering existant*, en ayant au préalable présenté les faiblesses de l'algorithme existant, déployé dans la littérature.

Une *validation de l'outil proposé*, en ajustant le degré de modularité et le coût de couplage des modules et en générant une architecture et reconfiguration optimale du produit.

CHAPITRE I : ETAT DE L'ART ET PROBLEMATIQUE

La performance d'un système industriel est fortement corrélée à sa fiabilité, sa disponibilité et donc à sa maintenabilité, qui sont les principaux paramètres constitutifs de la maintenance. Dans la mesure où les activités de maintenance font partie du cycle de vie d'un produit, il est impératif que ces activités, plus visibles en exploitation, soient prises en compte dès la conception, jusqu'à leur démantèlement (Lyonnet et *al.*, 1998 ; Tollenaere, 1998). Les problèmes liés à la maintenance constituent un frein important à la disponibilité des équipements (Samon, 2018). Selon Starkey (1989), il est courant que les équipements de valeur soient abandonnés pour les problèmes mineurs. La conception de certains matériels et la politique des constructeurs ne facilitent pas toujours la commande de certaines pièces détachées, d'où la nécessité d'avoir recours aux pièces d'occasion.

La maintenance a donc pour objectif principal la maîtrise du potentiel de production de l'entreprise, en assurant la sécurité des personnes et des biens. Elle ne doit donc pas seulement intervenir lors de l'exploitation des équipements, par des actions correctives et préventives ; elle a ainsi un rôle important à jouer (Monchy, 1987) : en amont de l'exploitation, lors de la phase de conception et en aval, lors de la phase de démantèlement. En effet, les caractéristiques de fiabilité et de maintenabilité, conditionnant la disponibilité de l'équipement, sont définies en conception. Il est important d'utiliser les connaissances de la maintenance lors de la conception d'un nouvel équipement et lors du démantèlement. La prise en compte des problèmes rencontrés par la maintenance, comme la modularité complexe, est une aide à l'amélioration de la maintenabilité d'un équipement lors de la conception. Ainsi, afin d'éviter la formation des modules de trop grandes tailles, il faudrait corriger les coûts de couplage des modules au moyen de certains coefficients correcteurs. La maintenance doit donc permettre d'atteindre les objectifs de productivité, de rentabilité et de croissance. Pour y parvenir, elle doit notamment maîtriser ses équipements de production dans les dimensions technique, économique et sociale (Boucly et *al.*, 1987).

Dans la suite de ce chapitre, nous parlerons d'abord de l'optimisation de la maintenance pour enfin chuter vers la présentation des critères de maintenabilité contextuels et intrinsèques. Ensuite, dans la branche Critères de maintenabilité intrinsèques qui se construis mieux en conception, nous nous intéresserons à la modularité pour positionner le problème de modularité. Enfin, nous ferons un état de l'art sur les méthodes de conception roustes, afin de dégager la problématique de notre travail de recherche.

I.1 OPTIMISATION DE LA MAINTENANCE

I.1.1 La maintenance et ses attributs

Selon l'AFNOR par la norme NF X 60-010, la **maintenance** se définit comme étant l'ensemble des actions permettant de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé. Bien maintenir, c'est assurer l'ensemble de ces opérations au coût optimal.

D'une manière générale, la maintenance a pour objet de garantir l'état de bon fonctionnement de l'outil de production et d'assurer ainsi sa disponibilité. Il va de soi que la maintenance sera d'autant plus efficace qu'elle sera capable d'anticiper les défaillances du système.

Jusqu'à un passé récent, la maintenance industrielle n'était perçue que comme une obligation, un poids pour les entreprises et exclusivement une source de dépenses. Les opérations de maintenance ont certes, un coût, mais lorsqu'elles sont intelligemment organisées et gérées, elles contribuent aussi à la création de valeur. Les améliorer permet donc à l'entreprise d'être plus performante et mieux armée face à la concurrence. L'optimisation de la maintenance passe notamment par :

- La mise en œuvre d'une stratégie adaptée aux spécificités de l'entreprise ;
- L'amélioration des procédures et canaux d'échanges entre les acteurs : responsables, opérateurs, techniciens de maintenance, etc.

Afin de déterminer la politique de maintenance adéquate, il est au préalable nécessaire de connaître les trois principaux attributs de l'équipement qui sont la fiabilité, la disponibilité et la maintenabilité.

I.1.2 La fiabilité

I.1.2.1 Définition

La **fiabilité** est « l'aptitude d'une entité à accomplir une fonction requise ou à satisfaire les besoins des utilisateurs, dans des conditions données, pendant une durée donnée (AFNOR, 1988) ». Le terme est un néologisme introduit dans les années 60 pour traduire le terme anglo-saxon « reliability » et si on accepte de la considérer comme une science, la fiabilité est la science des défaillances (Pagès et Gondran, 1980).

La **défaillance** se définit comme étant « l'altération ou la cessation d'un bien à accomplir la fonction requise » (AFNOR NF X 60-011). Suite à une défaillance, le bien peut

être indisponible ou sa performance peut tomber en dessous d'un seuil défini : il est alors incapable d'assurer les fonctions requises, ce qui est évalué au regard de critères simples, d'une part de défaillance potentielle dont le seuil associé est choisi de telle manière que si la dégradation ne l'atteint pas, le risque de défaillance avant une autre visite est acceptable et qu'il n'est pas nécessaire d'intervenir ; et d'autre part de dégradation qui se définit comme un état du produit ou du système présentant une perte de performance de l'une des fonctions assurées par le produit (AFNOR, 1988). Si les performances sont en dessous du seuil d'arrêt défini dans les spécifications fonctionnelles, il n'y a plus dégradation mais défaillance (Nzié, 2006).

I.1.2.2 Indicateurs de la fiabilité

Ce paragraphe est un recueil des principaux indicateurs utilisés pour estimer le niveau de fiabilité d'un système. Nous pouvons trouver plus de détails dans les ouvrages (Military Handbook – 217F, 1991 ; Ireson et *al.*, 1995 ; IEC 61703, 2001 ; Nzié, 2006). Le choix d'un indicateur dépend de plusieurs facteurs tels que la nature du système (réparable ou non réparable) et l'utilisation prévue. Quelques-uns sont présentés ci-dessous.

➤ *Le taux de défaillance ou taux de panne*

Connaissant le taux de défaillance $\lambda(t)$, on évalue la fiabilité par la relation ci-dessous où $f(t)$ est la densité de probabilité de défaillance :

$$R(t) = P(T > t) = \int_t^{\infty} f(x)dx = e^{-\int_0^t \lambda(x)dx} \quad (1)$$

➤ *Le ratio (nombre d'équipements $N(t)$ encore en état / nombre d'équipements N_0 testés)*

Ce ratio permet d'évaluer la fiabilité par la relation :

$$R(t) = \frac{N(t)}{N_0} \quad (2)$$

➤ *La MTBF (Mean Time Between Failure)*

A partir de la fiabilité, on calcule la MTBF par la relation (3).

$$MTBF = \int_0^{\infty} tf(t)dt = \int_0^{\infty} R(t)dt \quad (3)$$

On évalue également la MTBF comme le rapport entre la durée cumulée de bon fonctionnement et le nombre de défaillances observées.

➤ *La durée de vie utile*

Gittard (2009) présente quelques méthodes permettant d'estimer la fiabilité d'un système. Parmi ces méthodes, nous avons les arbres de défaillance (**AdD**), le bloc-diagramme de fiabilité

(BD), l'Analyse des Modes de Défaillance et leurs Effets (AMDE) et l'Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité (AMDEC).

Plusieurs secteurs d'activités, par exemple les secteurs pétrolier et nucléaire, et de nombreuses entreprises comme EDF (Électricité de France) disposent des bases de données de fiabilités des équipements qu'ils utilisent (Military Handbook – 217F, 1991 ; Smith, 2005).

I.1.2.3 Maintenance dans une approche fondée sur la fiabilité

Plusieurs auteurs dans la littérature se sont intéressés à cet aspect. C'est le cas par exemple de la NASA (2000) qui propose une approche efficace dans le cadre d'une activité de maintenance fondée sur la fiabilité. L'EC 60300-3-11(1999) qui constate que l'analyse des politiques de maintenance dans l'industrie du transport aérien à la fin des années 60 et au début des années 70 a conduit au développement du concept RCM (Reliability Centered Maintenance) qui a fait l'objet de directives. En France, on retrouve cette démarche chez Electricité de France avec la mise en œuvre d'une démarche dite « **Optimisation de la Maintenance par la Fiabilité (OMF)** » (Zwingelstein, 1996 ; Despujols, 2004). Cette approche est un concept continu recueillant des données de performance sur les systèmes (Figure 1).

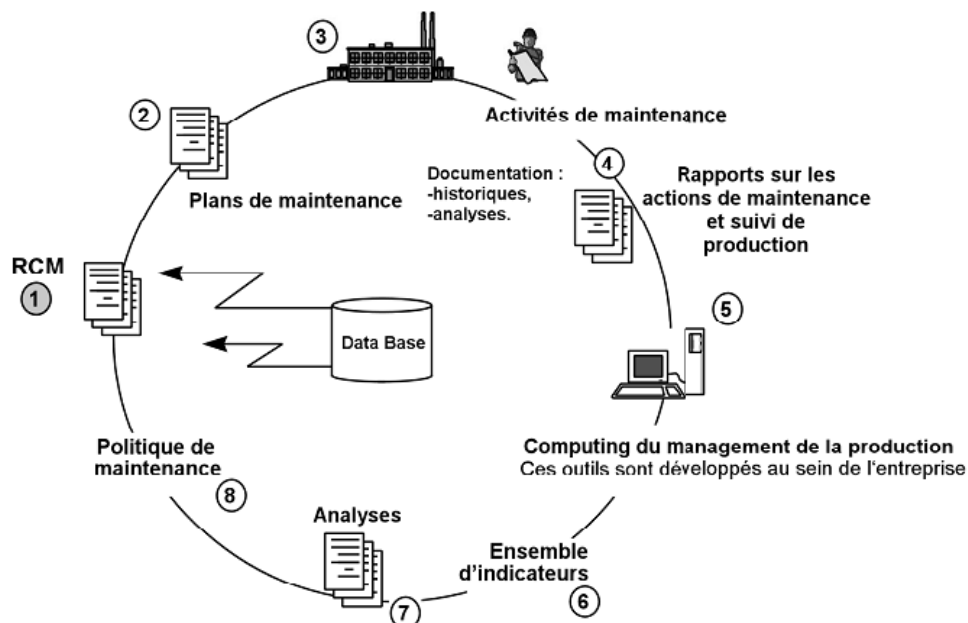


Figure 1: Mise en place d'une boucle d'amélioration et d'un retour d'expérience (Cotaina et al., 2000)

Yousfi (2014) définit l'OMF comme une méthode d'aide à la décision pour élaborer le programme de maintenance préventive d'une installation, en respectant des contraintes et en optimisant des critères. Les critères sont généralement : la disponibilité, la qualité du produit ou

du service rendu et les coûts d'exploitation. Parmi les contraintes, on trouve notamment la sécurité des personnes et la préservation de l'environnement...

L'OMF permet de déterminer : où des actions préventives sont nécessaires (sur quels matériels) ; quelles sont les actions à effectuer ; quand (avec quelle fréquence) doit-on les réaliser. Pour cela, elle propose trois grandes phases (Yousfi, 2014) :

- ✚ Une **phase d'analyse du retour d'expérience**, qui consiste à rechercher ce qui s'est passé sur les matériels (en termes de fiabilité, de disponibilité et de coûts), et ce qui a été fait comme actes de maintenance ou comme modifications.
- ✚ Une **phase d'évaluation des risques**, qui consiste à envisager les événements graves qui pourraient se passer (le possible), et qui met en œuvre les techniques d'analyse du fonctionnement et du dysfonctionnement. Ce travail, qui se rapproche de celui du concepteur, est ensuite complété par une prise en compte de l'historique de maintenance plus proche de la vision de l'exploitant. Le possible et l'observé permettent d'envisager le probable. C'est pour prévenir les défaillances qui présentent une gravité et qui ont des chances de se produire qu'il faudra envisager d'effectuer des tâches de maintenance préventive.
- ✚ Une **phase d'optimisation de la maintenance**, qui détermine les tâches à effectuer et leur fréquence en envisageant éventuellement des améliorations ou des modifications.

D'autres auteurs dans la littérature se sont intéressés à l'allocation de la fiabilité pour l'optimisation de la maintenance. L'**allocation** de fiabilité se définit comme un processus qui permet de traduire les exigences de fiabilité du système en exigences de fiabilité des composants (Military Handbook – 338B, 1998). La littérature présente plusieurs méthodes d'allocations de la fiabilité (Kececioglu, 1991 ; Ebeling, 1997 ; Military Handbook – 338B, 1998). Quelques-unes sont présentées ci-dessous :

- ***Allocation équitable***

Cette méthode attribue la même importance aux composants et leur attribue une même exigence de fiabilité. Pour un système série de N composants dont les taux de pannes sont constants, l'allocation équitable conduit à :

$$R_i^* = \sqrt[N]{R_S^*} \quad (4)$$

Ou bien

$$\lambda_i^* = \frac{\lambda_S^*}{N} \quad (5)$$

Où R_S^* et R_i^* (ou λ_S^* et λ_i^*) sont respectivement l'exigence de fiabilité (ou de taux de panne) du système et l'exigence alloué au composant i .

- **Techniques des efforts égaux**

L'idée ici est d'évaluer l'exigence de fiabilité allouée au système, connaissant la fiabilité de chaque composant i du système en effectuant un produit des fiabilités des N composants.

$$R_S^* = \prod_{i=1}^N R_i \quad (6)$$

- **Méthode ARINC (Aeronautical Radio, Incorporated)**

Cette méthode a été proposée par (Alven, 1964). Elle suppose que les composants sont indépendants, en série et ont un taux de panne constant.

Pour un taux de défaillance prévisionnel constant ($\lambda_i = cste$),

✓ Le facteur de pondération est donné par :

$$w_i = \frac{\lambda_i}{\sum_{i=1}^N \lambda_i} \quad (7)$$

✓ Le taux de défaillance alloué est donné par :

$$\lambda_i^* = w_i \cdot \lambda_S^* \quad (8)$$

✓ La fiabilité allouée est donnée par :

$$R_i^* = [R_S^*]^{w_i} \quad (9)$$

λ_i : taux de panne initial du composant i . λ_i peut être obtenu par exemple à partir des bases de données.

Chaque composant se voit allouer une exigence de taux de panne proportionnel à son taux de panne initial.

- **Méthode de faisabilité technique**

Pour un système de N sous-systèmes i donné et en considérant chaque sous-système, l'idée ici est de pouvoir évaluer son facteur de poids (w_i), son facteur de complexité (C_i) et son taux de défaillance alloué (λ_i). L'évaluation de ces paramètres nécessite la connaissance de quatre paramètres variant de 0 à 10 qui sont :

- de complexité, noté (W_1) ;
- d'état de l'art (W_2) ;
- de temps d'utilisation (W_3) ;
- de sévérité d'environnement (W_4) .

✓ Le facteur de poids du sous-système est donné par :

$$w_i = W_{1i} \times W_{2i} \times W_{3i} \times W_{4i} \quad (10)$$

✓ Le facteur de complexité du sous-système est donné par :

$$C_i = \frac{w_i}{\sum_{i=1}^N w_i} \quad (11)$$

✓ Son taux de défaillance alloué est :

$$\lambda_i = \lambda_S \cdot C_i \quad (12)$$

λ_S représente le taux de défaillance du système considéré.

D'autres méthodes peuvent être consultées dans les travaux de (Kececioglu, 1991 ; Boyd, 1992 ; Military Herdbook – 338B, 1998) ; Subba et *al.*, 1999) : la Méthode AGREE (Advisory Group on Reliability of Electronic Equipment) ; l'algorithme de minimisation des efforts ou la méthode Karmioli ; la méthode de Boyd ou de Combinaison d'allocation équitable avec la méthode ARINC et l'allocation optimale de la fiabilité.

I.1.3 La Disponibilité

I.1.3.1 Définition

La norme IEC 50-191 (1990) définit la **disponibilité** comme étant la « probabilité pour que l'équipement ou le système utilisé dans les conditions prévues soit en état d'accomplir une fonction requise à un instant donné ».

Une autre définition similaire est proposée par Ait-Kadi (2007) qui définit la **disponibilité instantanée** comme « la probabilité que le système soit en état d'opération à un instant t et ce, indépendamment des états précédents ».

Si les durées de vie et les durées de réparation suivent une distribution exponentielle avec respectivement un taux de panne λ et un taux de réparation μ , tous constants, la disponibilité instantanée est donnée par l'expression (13).

$$A(t) = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} e^{-(\mu + \lambda)t} \quad (13)$$

I.1.3.2 Indicateurs (ou quantification) de la disponibilité

La disponibilité peut se mesurer :

- sur un intervalle de temps $[0, T]$ donné (disponibilité moyenne), $A(T)$:

$$A(T) = \frac{1}{T} \int_0^T A(t) dt \quad (14)$$

La disponibilité moyenne peut être généralisée sur un intervalle de temps $[t_1, t_2]$, représentant par exemple une mission. On parle alors de disponibilité de mission, $A_{t_2-t_1}$.

$$A_{t_2-t_1} = \frac{1}{T} \int_{t_1}^{t_2} A(t) dt \quad (15)$$

- à un instant donné (disponibilité instantanée), $A(t)$;
- à la limite, si elle existe, de la disponibilité instantanée lorsque $t \rightarrow \infty$ (disponibilité asymptotique), A_∞ :

$$A_\infty = \lim_{t \rightarrow \infty} A(T) \quad (16)$$

Dans la littérature, il existe trois types de disponibilité en régime qui dépendent de ce qu'on inclut dans les définitions du temps de bon fonctionnement et du temps d'arrêt (Ireson, 1995 ; Ebeling, 1997 ; Dhillon, 2002). Ces disponibilités sont présentées ci-dessous.

➤ **Disponibilité stationnaire ou asymptotique A_∞**

Elle représente la disponibilité du système dans un environnement idéal où les ressources requises (compétences humaines, outillage, pièces de rechange, documentation, etc.) sont disponibles. Le calcul de A_∞ découle de la relation (13), lorsque $t \rightarrow \infty$. Cette expression prend en compte uniquement les temps d'arrêt dus à la maintenance corrective. Les arrêts dus à la maintenance préventive, les temps logistiques et les temps administratifs ne sont pas considérés. On a alors :

$$A_\infty = \frac{\mu}{\mu + \lambda} = \frac{MTBF}{MTBF + MTTR} \quad (17)$$

➤ **Disponibilité inhérente A_{inh}**

La disponibilité inhérente prend en compte les temps d'arrêt dus à l'exécution des tâches de maintenance corrective et préventive. Elle est calculée selon l'expression (18). Les temps administratifs et logistiques ne sont pas pris en compte.

$$A_{inh} = \frac{MTBM}{MTBM + \bar{M}} \quad (18)$$

$MTBM$ et \bar{M} représentent respectivement la moyenne des temps de fonctionnement entre les actions de maintenance préventive et corrective et la moyenne combinée des temps actifs de maintenance corrective et préventive.

➤ **Disponibilité opérationnelle A_{op}**

La disponibilité opérationnelle propre est donnée par l'expression :

$$A_{op} = \frac{MTBF}{MTBF + MTI} = \frac{MTBF + RT}{MTBF + MTTR + MTL} = \frac{MTBM}{MTBM + MMT} \quad (19)$$

Avec :

MTBF : Moyenne des Temps de Bon Fonctionnement

MTI : Moyenne des Temps d'Indisponibilité

RT : « Ready Time » ou Temps Moyens d'Attente, le système étant prêt à fonctionner

MTL : Moyenne des Temps Logistique

MTTR : Moyenne des Temps Techniques de réparation

MTBM : Temps Moyen Entre Actions de Maintenance Préventive ou Corrective

MMT : Temps Moyen des Actions Préventives ou Correctives

Elle est aussi donnée par la relation :

$$A_{op} = \frac{T_o - T_A}{T_o} = \frac{TM_c \cdot N}{TM_c \cdot N + T_A} \quad (20)$$

Avec :

T_o : Temps d'ouverture de la ligne ou temps requis

TM_c.N : Temps moyen de cycle x Nombre de pièces produites

T_A : temps cumulés des arrêts propres

I.1.4 La maintenabilité

I.1.4.1 Définition

Plusieurs auteurs dans la littérature définissent la maintenabilité. C'est le cas de (IEC 60050-191, 1990 ; Military HandBook – 470A ,1997 ; ARMP-7, 2003).

Pour une entité donnée, utilisée dans des conditions données d'utilisation, la **maintenabilité** est la probabilité pour qu'une opération donnée de maintenance active puisse être effectuée pendant un intervalle de temps donné, lorsque la maintenance est assurée dans des conditions données et avec l'utilisation de procédures et de moyens prescrits (IEC 60050-191, 1990 ; ARMP-7, 2003).

Une autre définition complémentaire fournie par le Military HandBook – 470A (1997) est la suivante : le **maintenabilité** est la relative facilité et économie de temps et de ressources avec laquelle un système peut être maintenu ou remis dans un état de fonctionnement spécifique, lorsque la maintenance est exécutée par un personnel ayant les compétences requises et utilisant les procédures et les ressources matérielles prescrites, à chaque niveau de maintenance spécifié.

Cette dernière définition indique que le **maintenabilité** est une caractéristique ou aptitude d'un système et que sa mesure se base essentiellement sur l'appréciation du déroulement des activités de maintenance. La maintenance désigne l'ensemble des activités

destinées à maintenir ou à rétablir un système dans un état spécifié ou dans des conditions données de sûreté de fonctionnement, pour accomplir une fonction requise (Ait-Kadi et *al.*, 2004).

Comme la durée de vie pour la fiabilité, la durée de réparation lors de la maintenance d'un système est une variable aléatoire dont la distribution cumulative est donnée par l'expression (21).

I.1.4.2 Indicateurs de la maintenabilité

➤ *Maintenabilité instantané*

Elle est définie par :

$$M(t) = \int_0^t g(t)dt = 1 - e^{-\int_0^t \mu(t)dt} \quad (21)$$

Où $g(t)$ est la fonction densité de probabilité du temps de réparation et $\mu(t)$ le taux instantané de réparation.

➤ *Moyenne des temps techniques de réparation*

La moyenne des temps techniques de réparation MTTR, qui est l'une des principales mesures de la maintenabilité, est calculée à partir de l'expression (22)

$$MTTR = \int_0^{\infty} t.g(t)dt = \int_0^{\infty} (1 - M(t))dt \quad (22)$$

➤ *Taux instantané de réparation*

On appelle **taux de réparation** d'un système réparable au temps t la probabilité que l'entité soit réparée entre les instants t et $t + dt$, sachant qu'elle n'était pas réparée sur l'intervalle $[0, t]$. Elle se note :

$$\mu(t) = P(\text{entité réparée sur } [t, t + \Delta t] / \text{elle n'était pas réparée sur } [0, t]) \quad (23)$$

Le taux de réparation peut s'assimiler à la proportion d'entités réparées sur $[t, t + dt]$ rapportées aux entités non réparées à l'instant t . En prenant un raisonnement similaire à celui utilisé pour le calcul de l'intensité de défaillance $\lambda(t)$ et à l'aide du théorème des probabilités conditionnelles, on déduit après des calculs simples que :

$$\mu(t)dt = \frac{g(t)}{1 - M(t)} \quad (24)$$

➤ *Densité de probabilité du temps de réparation*

En combinant les relations (21) et (24), on en déduit que :

$$g(t) = \mu(t).e^{-\int_0^t \mu(t)dt} \quad (25)$$

I.1.4.3 Critères de maintenabilité

Lors de la conception d'un système, plusieurs critères doivent être pris en compte pour que sa maintenance soit facile, rapide et économique en ressources. Ces critères sont généralement classés en deux groupes : les critères contextuels qui sont relatifs au contexte d'utilisation et les critères intrinsèques qui sont inhérent au système (Menye, 2009).

I.1.4.3.1 Critères contextuels

Certains éléments caractéristiques du contexte d'utilisation d'un système peuvent avoir une influence majeure sur sa maintenabilité. Menye (2009) classe ses éléments en quatre groupes : les ressources humaines, les ressources matérielles, l'organisation mise en place et les conditions environnementales. Le tableau 1 fait une synthèse des principaux critères contextuels et les facteurs qui les caractérisent.

Tableau 1 : Critères contextuels (Menye, 2009)

	Critères contextuels	Facteurs
Ressources humaines	Equipe de maintenance	<ul style="list-style-type: none"> • Effectif • Compétences • Disponibilités • Coût (salaire, embauche, licenciement ,...)
Ressources matérielles	Equipement support	<ul style="list-style-type: none"> • Quantité • Disponibilité • Coût (achat, réparation, exploitation,...)
	Documentation technique	<ul style="list-style-type: none"> • Disponibilité • Mis à jours
	Pièces de rechange	<ul style="list-style-type: none"> • Disponibilité • Délai d'approvisionnement • Coût (achat, transport, stockage, pénurie,...)
Environnement organisationnel	Procédures administratives	<ul style="list-style-type: none"> • Nombre de niveaux de décision • Quantité de documents à remplir • Etc.
	Organisation et gestion de la maintenance	<ul style="list-style-type: none"> • Eloignement des ateliers de réparation • Politiques de maintenance • Sous-traitance • Etc.

Environnement physique	Conditions environnementales	<ul style="list-style-type: none"> • Température • Humidité • Précipitation (pluie, neige, etc.) • Composition de l'air • Radiation • Champs électromagnétique • Vibrations • Chocs • Situation géographique • Saison • Luminosité • Etc.
------------------------	------------------------------	---

Pour adapter le système à son contexte d'exploitation, le concepteur doit connaître l'effet de chaque facteur d'influence de contexte et définir le système de telle sorte qu'il soit résistant à cet effet. Par exemple, utiliser des matériaux résistants au climat du lieu d'utilisation, utiliser les technologies dont la maintenance peut être exécutée sur place, intégrer les composants dont les pièces de rechange sont disponibles sur place, dimensionner les composants pour qu'ils soient plus résistants aux chocs, etc. Le concepteur peut par exemple se baser sur certains travaux pour atteindre ce but (Military Standard – 810F, 2000 ; NSWC-07, 2007).

I.1.4.3.2 Critères intrinsèques

Ireson et *al.* (1995) et Military Handbook – 470A (1997) présentent quelques critères intrinsèques qui confèrent à un système un bon niveau de maintenabilité.

➤ Accessibilité

Elle se caractérise par la facilité avec laquelle un élément peut être atteint lorsque les opérations de maintenance sont nécessaires. L'objectif visé, en la mettant en œuvre pendant la conception, est de réduire les durées d'intervention directe sur l'équipement. La spécification de l'accessibilité porte sur le temps d'accès à un composant, le nombre maximum d'unités à déposer autour d'un composant donné pour atteindre celui-ci, la masse totale des unités à déposer autour d'un constituant donné par rapport à la masse du composant lui-même, la référence aux normes sur l'ergonomie et la sécurité. Sa mise en œuvre doit prévoir des passerelles d'accès à hauteur adaptée, des dispositifs de déverrouillage lorsque pour des raisons de sécurité, le système physique est protégé par un code ou un verrou. Suh (1990), a développé une méthode quantitative de mesure de l'accessibilité d'un composant ou « Component Accessibility (CA) » en phase de conception. Il définit la CA comme l'inverse de l'information

nécessaire à un agent d'assemblage idéal à manipuler le composant suivant une direction en ligne droite sans collision vers une destination en présence d'autres composants assemblés. Son aide permet au concepteur de choisir un arrangement optimal des composants pour les opérations d'assemblage / démontage. L'information requise de la définition de la CA est relative à la probabilité P de succès. Le contenu de l'information est défini par :

$$I = \log_2(1/P) = \log_2(1/(\text{espace accessible actuelle}/\text{espace accessible idéale})) \quad (26)$$

Et l'accessibilité du composant est :

$$CA = 1/I \quad (27)$$

La connaissance sur l'accessibilité d'un composant permet au concepteur d'avoir une appréciation globale des options disponibles avant de choisir le déplacement d'un composant (dans le processus d'assemblage ou de démontage en maintenance). Lorsqu'on est dans le plan pour un corps qui est déplacé pour être assemblé, s'il n'y a pas d'obstacle, l'espace accessible est 360° (idéale), sinon son espace accessible (actuelle) est 360° diminué de l'angle interdit.

Lorsqu'on ne tient pas compte de la hauteur du corps.

$$CA = 1/\log_2(360^\circ / (360^\circ - \text{Angle interdit})) \quad (28)$$

En tenant compte de la hauteur du corps, on définit l'accessibilité par :

$$CA = \frac{1}{\log_2 \left(\frac{\frac{2\pi}{3}}{\frac{2\pi}{3} - \prod_{j=1}^n U_{k=1}^2 FV_{jk}} \right)} \quad (29)$$

n est le nombre d'obstacles et FV_{jk} est le volume interdit induit par le $k^{\text{ième}}$ triangle sphérique de l'espace obstacle j . Pour $j=1$:

$$FV_k = \frac{4}{3} \tan^{-1} \left(\tan\left(\frac{S_k}{2}\right) \tan\left(\frac{S_k - a_k}{2}\right) \tan\left(\frac{S_k - b_k}{2}\right) \tan\left(\frac{S_k - c_k}{2}\right) \right)^{1/2} \quad (30)$$

(Formule de trigonométrie sphérique)

$s_k = (a_k + b_k + c_k)/2$ et a_k , b_k et c_k sont des côtés du $k^{\text{ième}}$ triangle sphérique de l'espace obstacle.

Cette dernière méthode est intéressante, car elle permet de faire en conception des allocations de l'accessibilité d'un composant. Nous notons aussi que la formule est appliquée ici pour les obstacles de formes parallélépipédiques, donc n'est pas générale pour toutes les formes qu'on pourra rencontrer dans l'environnement de la maintenance.

➤ **Démontabilité**

Selon la norme NF X 60300, la démontabilité ou aptitude au démontage est la conception du bien autorisant l'accès à chacun de ses composants et assurant un démontage et un remontage facile de ses sous-ensembles. L'ordre de démontage et de remontage peut être ainsi déterminé et précisé dans la documentation de maintenance. La pose et la dépose comprennent l'ensemble des opérations nécessaires à la suppression et au rétablissement des liaisons entre l'élément à démonter et l'ensemble dont il fait partie, afin de mieux en disposer en vue d'assurer au mieux son remplacement ou sa maintenance. La démontabilité dépend fortement de l'accessibilité, de la modularité, et de la connectique (la connectique et la modularité étant présenté ci-dessous). Grâce à la connectique, on peut faire des allocations de démontabilité utiles pour le démontage / montage du produit suivant (Nzié, 2006) :

- Résistance d'attache (de liaison), exprimée par la force requise pour démonter ;
- Facilité de mise en action, traduite par la force requise pour assembler la liaison et les composants reliés (raccordés), le temps pour réaliser la connectique (raccordement, liaison) ;
- Facilité de desserrage, traduite par la force requise, le temps de desserrer les objets ;
- Facilité d'accès, traduite par le fait que l'espace disponible excède la taille des mains, les doigts, les outils ;
- Précision de la position relative des objets, se traduisant par la distance représentant la disparité maximum d'un objet par rapport à l'autre.

On spécifie la démontabilité par le nombre maximum et le coût maximum des outillages spéciaux nécessaires, le degré approprié de complexité des outillages et des réglages à effectuer au remontage, le nombre d'intervenants nécessaires, le nombre d'éléments à démonter pour accéder à l'élément échangeable et le nombre d'éléments qui ne peuvent pas être démontés sans destruction. Selon Shana et Hung (2015), l'étude du démontage part tout d'abord de la définition des méthodes de planification du démontage. A cet effet, ils distinguent plusieurs niveaux de démontage comme le précise la figure ci-contre.

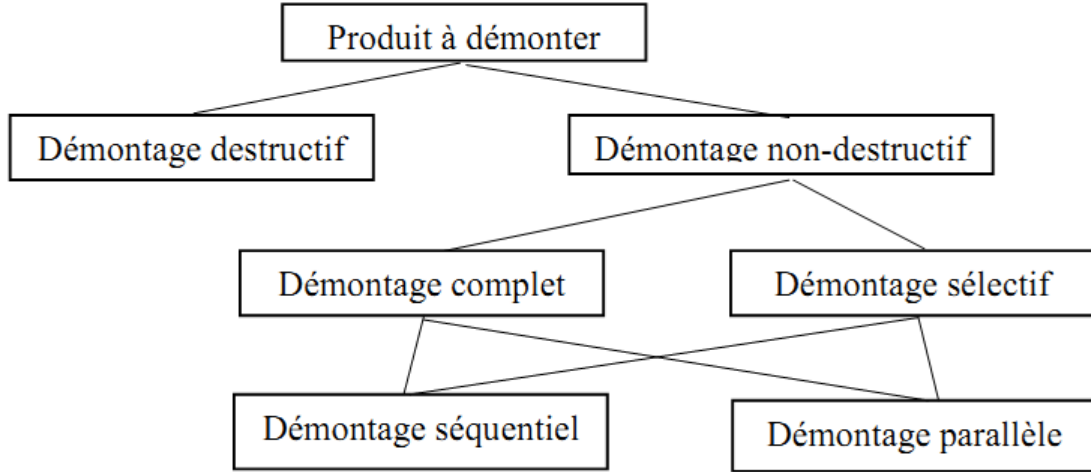


Figure 2: Méthodes de planification de démontage (Shana et Hung, 2015)

La méthodologie de démontage sélective ou « selective disassembly methodology » de Srinivasan permet d’analyser la démontabilité d’un modèle CAO. Elle se passe en trois étapes qui sont : l’identification des composants à démonter, la détermination d’une séquence de démontage optimale, la prise des décisions de conception du démontage à base de la séquence évaluée (Srinivasan et *al.*, 1999). Elle est implémentée dans un logiciel, A3D (Assembly and Disassembly in 3 dimensions) et permet de faire en conception des évaluations et allocations importantes de maintenance suivantes : Le temps pour démonter les connecteurs fixant les composants de la séquence S, pour C_i composant de S,

$$DT_F = \sum_i \left(\sum_j (N_{i,j} \times T_j + A_{ji} \times O_{ji}) \right) \quad (31) \text{ pour } C_i \text{ composant de S,}$$

Le temps pour démonter les composants de S (les connecteurs étant déjà démontés),

$$DT_C = \sum_i \left(A_i \times O_i \times (K_i \times d_i) \times \left[\frac{W_i \times F_i}{K_w} \right] \right) \quad (32) \text{ pour } C_i \text{ composant de S,}$$

Le coût de démontage :

$$DC = K_0 \times K_s \times LR \times (DT_F + DT_C) \quad (33)$$

K_0 : facteur constant tenant compte de l’espace, l’outillage, etc.,

K_s : facteur constant tenant compte de la sécurité du composant,

LR : coût horaire,

$N_{i,j}$: nombre de connecteurs de type j fixant le composant C_i ,

T_j : temps standard de démontage d’un connecteur de type j fixant C_i ,

A_{ji} : facteur d’accessibilité du connecteur de type j fixant le composant C_i ,

O_{ji} : facteur d’orientation du connecteur de type j fixant le composant C_i ,

A_i : facteur d'accessibilité du composant C_i ,

O_i : facteur d'orientation du composant C_i ,

K_w : poids (masse) standard limite que peut soulever un opérateur source (lifting source) ;

d_i : distance de déplacement du composant C_i pour le démonter de l'ensemble,

K_t : temps standard pour déplacer un composant de poids (masse) inférieure à K_w à une distance d_i ,

W_i : poids (masse) du composant C_i ,

F_i : facteur de force du composant C_i .

Cependant, cette méthode suppose que les mouvements relatifs des composants sont déterminés sans considération des outils, que les ensembles sont sans frottement et définis par une géométrie nominale (généralement en maintenance, les composants ont été soumis à l'usure, la corrosion, etc.), que les composants se démontent de l'ensemble suivant un simple mouvement rectiligne et qu'un composant n'est démontable qu'après le démontage d'un composant adjacent, que le démontage est séquentiel, monotonique (un composant est totalement démonté pendant son démontage). Or la simultanéité de démontage est courante en maintenance.

➤ **Connectique**

C'est un véritable critère de maintenabilité, en ce sens qu'elle permet le démontage rapide des éléments défaillants ainsi que leur rechange et aussi certains aspects de la testabilité. En la construisant bien, l'objectif pour l'acteur de maintenance est de faciliter le remplacement des éléments défaillants, de permettre certain test fonctionnel et de diagnostic, afin de réduire les temps de dépose – pose et les temps de diagnostic et de validation. Globalement le travail du dépanneur devient facile et le temps d'immobilisation et donc d'indisponibilité du système réduit. La connectique se spécifie par le nombre maximal de connexions / déconnexions des éléments associés, la force à développer pour exécuter la connexion / déconnexion, la qualité des liaisons.

Des travaux de Hsu (2002), la connectique ou assemblabilité d'un produit ou Product Assemblability (PA) se définit comme étant l'inverse du contenu de l'information de la totalité des composants de ce produit. Et le contenu de l'information du produit est la somme des contenus d'information des composants associés à chaque événement A_i , qui est : avoir une destination du composant en ligne droite lors de l'assemblage (ou montage). Alors, pour un produit à n composants, la séquence d'assemblage consiste en n étapes représentant l'ordre

d'assemblage. Si un événement dans une étape i est A_i , il y a A_1, A_2, \dots, A_n événements associés dans le processus global.

$$PA = \frac{1}{I_{produit}} = \frac{1}{\log_2 \left(\frac{1}{P(A_1 \cap A_2 \cap \dots \cap A_n)} \right)} \quad (34)$$

On note dans cette étude qu'un événement est associé à un seul composant à la fois. Or, n'a-t-on pas toujours vu que des écrous d'une bride sont montés à la fois ?

Il est clair que les résultats de l'étude sont incontestablement d'un apport bénéfique pour la reconception des produits. En effet, elle fournit une possibilité de comparaisons des alternatives en fonction des CA. Cependant, elle ne généralise pas la forme des composants (par exemple la forme concave des faces), la direction d'assemblage est toujours une ligne droite, ce qui n'est pas toujours le cas dans un environnement complexe de maintenance. Concevoir un composant est une chose. Une autre chose est la conception d'un système de plusieurs composants où l'intégrité structurelle et fonctionnelle est un amalgame complexe des composants. Quand on est en présence d'un tel système, quatre tâches de conception, dont l'acteur de maintenance est partie prenante, s'imposent pour la mise en œuvre de la connectique :

- Identifier par le moyen d'une décomposition structurelle les éléments qui composent le système ;
- Définir la nature des liaisons structurelles entre éléments constitutifs ;
- Si nécessaire, établir chaque fois les exigences fonctionnelles de ces liaisons ;
- Concevoir les liaisons.

Comme c'est le cas pour les composants technologiques structurels, pour le quatrième point (Concevoir les liaisons pour configurer la connectique), il incombe au concepteur maintenant de participer au choix de (s) : matériaux, configuration, géométrie, exigences de process (tels que les détails de fabrication, assemblage, charge). Cependant, nous ne devons pas oublier qu'un excès de connectique peut nuire à la fiabilité. En effet, elle est souvent un point de fragilité des systèmes par sa nature mécanique et technologique et par l'usure engendrée par la fréquence des démontages des éléments. Le choix d'une bonne connectique doit donc être motivé par des considérations de fiabilité.

➤ **Défectabilité**

Elle concerne la réduction des temps de localisation et de diagnostic. Un logiciel de recherche et de localisation des défauts, les outils d'aide au diagnostic, une supervision, mais aussi un simple voyant ou le repérage des câbles et des points de mesure sont autant d'éléments

de réduction des temps d'investigation. La réalisation d'une AMDEC amène le concepteur à évaluer le critère « détectabilité » et à proposer des solutions si nécessaires.

➤ **Aptitude de pose et dépose**

Elle concerne les modules qui nécessitent un échange standard en préventif ou en cas de défaillance. Elle concerne les liaisons à supprimer pour isoler le module de son ensemble. Prenons l'exemple d'un groupe moteur électrique / pompe centrifuge : la dépose se rapporte à l'électricité (consignation, accès au bornier, connectique), à l'hydraulique (vannes d'isolement, vidange, boulonnerie des brides), à la mécanique (boulonnerie de la fixation). Des solutions plus ou moins rapides existent pour faciliter chacune de ces opérations de maintenance. Quelques problèmes à optimiser pour améliorer l'aptitude à la pose / dépose :

- Réduction du nombre de liaisons ;
- Réduction du nombre d'outils à utiliser (standardisation des liaisons) ;
- Assurer un pré positionnement à la pose : repères, tétons de centrage, rails de guidage ;
- Absence de réglages, préférable à des réglages longs et délicats ;
- Facilité d'accès.

Notons que l'interchangeabilité d'un module se fait souvent en « temps réel » d'indisponibilité de l'équipement, contrairement à sa remise en état réalisée en temps différé. Son aptitude à la dépose est donc un facteur de disponibilité de l'équipement.

➤ **Standardisation**

Elle vise à la simplification par réduction aussi bien en matière de fabrication que de logistique et de maintenance. En maintenance, elle s'exerce à tous les niveaux techniques et commerciaux, en permettant la réduction des stocks aussi bien que la rapidité et la simplicité des interventions. Prenons quelques exemples :

- Les équipements : il est plus facile de maintenir 10 machines de même type que de types différents.
- Les technologies : il est plus facile de se tenir à un modèle d'automate et de former les techniciens à sa programmation que de multiplier les formations.
- Les modules : utiliser 20 moteurs électriques ou 20 pompes centrifuges identiques offre plus de facilité d'organisation de la maintenance que s'ils étaient tous différents.
- Les outillages : démonter tout un module avec une clé de 13 est plus simple que d'avoir à inventorier toute sa caisse à outils.

- Les composants élémentaires : quincaillerie, visserie, graisseurs, robinets, trappes de visites, interrupteurs sont autant d'éléments qu'il est facile de standardiser.
- Les lubrifiants et leur fournisseur : ne pas suivre les préconisations par références de marque, qui conduiraient à une profusion de fûts. Il existe des tableaux d'équivalence et il suffit de 5 types d'huile et de 2 types de graisse pour assurer la lubrification d'un site industriel.
- Les procédures : standardiser leur présentation facilite l'exploitation.
- Les fournisseurs : un équilibre est à trouver entre le monopole accordé à un fournisseur privilégié et la multiplication des sources.

Notons que la normalisation est un outil de standardisation, qui elle-même facilite l'interchangeabilité.

➤ **Modularité**

Pour la simplification de la fabrication d'un équipement et pour l'amélioration de sa maintenance, c'est ce critère de maintenabilité que nous allons prendre en compte dans le cadre de nos travaux.

D'après Nzie (2006), un **module** est une association de composants reliés physiquement, groupés dans le but de constituer un sous-ensemble aisément démontable et remplaçable. Souvent considéré comme un critère de conception, la modularité l'est d'avantage pour la maintenabilité, car il permet de concevoir des sous-ensembles fonctionnels démontables et interchangeables, utilisables indifféremment dans des fonctions distinctes ou même sur des matériels différents. La **modularité** du produit se spécifie par rapport : à la **masse** et l'**encombrement maximal** de ses **modules**, en fonction de l'aptitude à la manutention, à son analyse fonctionnelle, aux normes spécifiques (par exemple la taille des tiroirs), au nombre maximum de références utilisées.

La facilité de l'interchangeabilité (carte électronique par exemple) est un facteur favorisant le transfert de tâches vers les opérateurs, dans le cas de la TPM (Totale Productive Maintenance) (Bellaouar et Beleulmi, 2014). Le module de remplacement peut provenir :

- d'un stock interne (module neuf ou remis en stock après réparation) ;
- d'un stock externe (module ou composant tenu en stock chez le fournisseur) ;
- du bien lui-même (par échange de deux éléments constitutifs, redondance) ;
- d'un bien identique hors service ou déclassé (cannibalisation) ;
- d'un bien différent comportant le même module;

- d'un atelier de fabrication.

Dans le cas d'un composant, il doit être fabriqué dans le respect des spécifications et des tolérances normalisées (joint d'étanchéité, coussinet de palier, etc).

Cela offre une gamme large de solutions pour organiser une intervention. L'interchangeabilité suppose le respect des normes (ajustements, filetages, produits, lubrifiants, raccords, connexions, etc).

A la section suivante, nous allons faire une recension des travaux portant sur la modularité, afin de positionner le problème lié à la modularité.

I.2 RECENSION DES TRAVAUX SUR LA MODULARITE

I.2.1 Définition

Nous retiendrons comme définition que : « La **modularité** est un mode de structuration d'un système complexe qui préconise de concevoir des structures basées sur la minimisation de l'interdépendance entre des composants et la maximisation de l'interdépendance à l'intérieur de chaque composant. Les composants peuvent être mélangés pour obtenir une nouvelle configuration sans perte de la fonctionnalité du système ou de performance », d'après (Baldwin et Clark, 1997 ; Langlois, 2002).

La modularité est une façon de concevoir une structure dans le but d'en réduire sa complexité. La nature des interdépendances entre les composants et la façon dont ils s'interfacent ont des implications profondes sur la complexité structurelle. Manipuler cette complexité peut être fait par la réduction du nombre d'unités et en groupant ces unités dans des sous-systèmes. La réduction de la complexité passe par la réduction des interdépendances entre les composants à travers des sous-systèmes (Langlois, 2002). Les composants sont appelés des **modules**.

La définition d'un module dans le Larousse est « *élément juxtaposable, combinable à d'autres de même nature ou concourant à une même fonction* ».

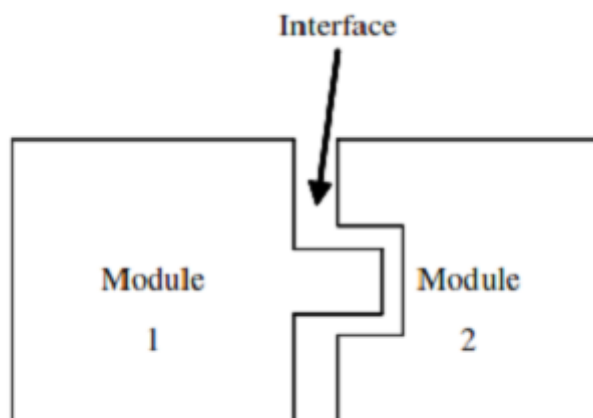


Figure 3 : Visualisation de la modularité

I.2.2 principes de construction d'un système modulaire

En gestion industrielle, la modularité est étudiée depuis les années 1960. Une première contribution est faite par Starr (1965) qui propose que la production modulaire soit un moyen d'augmenter la variété d'offres de produits pour satisfaire aux exigences du marché sans sacrifier l'efficacité dans la production. Starr (1965) indique que la conception des parties du produit est essentielle pour la production modulaire parce que « c'est l'essence du concept modulaire pour concevoir, développer et produire ces parties qui peuvent être combinées d'un grand nombre de façons ». Le concept de modularité est fondé sur trois principes de construction d'un système complexe : la quasi-décomposabilité des modules, l'attache lâche entre les modules et l'interfaçage par standardisation.

➤ ***La quasi-décomposabilité (Near decomposability)***

Selon Simon (1969), la construction du produit est imaginée pour potentiellement être considérée comme quasi-décomposable. La décomposition en modules est essentiellement une façon de gérer la complexité.

La quasi-décomposabilité signifie qu'il peut y avoir des niveaux différents d'analyse et par conséquent, des niveaux différents de modularité. De plus, le même module peut faire partie de systèmes différents. Enfin, il existe des unités minimales qui sont les plus petits composants, non décomposables. Donc, le degré de modularité dépend du type de système dans l'analyse, peut varier avec l'unité d'analyse, doit être mesuré le long d'un continuum d'une structure intégrale à un module élémentaire et peut changer au fil du temps, puisque la modularité est aussi un principe de reconfiguration. Chaque système est modulaire dans une certaine mesure : très peu de systèmes sont composés de parties qui interagissent et s'affectent si fermement qu'il

n'y a aucune occasion de reconfigurer les sous-systèmes. En outre, Alexander (1971) indique explicitement l'importance de cette notion pour l'adaptabilité de systèmes : « Aucun système adaptatif complexe ne réussira son adaptation dans un temps raisonnable à moins que l'adaptation ne puisse passer par le sous-système, chaque sous-système étant relativement indépendant des autres ».

➤ ***L'attache lâche (Loose coupling)***

Weick (1976) a proposé la notion d'attache lâche (loose coupling) comme une façon de capturer la nature de la coordination des modules qui ne sont pas identifiés par « des mots comme la connexion, le lien, ou l'interdépendance ». L'attache lâche des modules véhicule des connotations éphémères, dissolubles dans le temps et de la colle « qui maintient des organisations ensemble ». Comme la notion de Simon (1969) de quasi-décomposabilité des systèmes, le concept d'attache lâche embrasse l'idée que la plupart des systèmes ne sont ni entièrement détachés, ni accrochés entièrement, mais plutôt presque décomposables avec des liens lâches. Cela permet ainsi la compréhension d'organisations comme les systèmes qui sont ouverts et fermés en même temps. L'attache lâche signifie que les modes de coordinations présentés en infra (contractualisation, ajustement mutuel, standardisation des flux ou la planification des tâches) doivent exister mais que chaque module doit aussi avoir un degré d'autonomie afin de pouvoir être séparé et recollé ailleurs, ou modifié, ou reconçu et donc permettre au manager de faire de la reconfiguration continue.

➤ ***L'interfaçage***

Baldwin et Clark (1997) soutiennent que dans un système modulaire, chaque module communique et interagit avec les autres via les interfaces qui permettent le découplage des modules. La coordination passe par la définition de lois de structuration et de paramétrage. La standardisation est selon eux le mode de coordination le plus adapté à la construction modulaire. Les lois visibles de structuration (appelées aussi informations visibles) sont des processus de construction et d'assemblage des modules permettant de spécifier que chaque module est une partie d'un système entier et quelles seront les fonctions de chaque module ; le rôle des interfaces et la façon dont les modules interagissent entre eux ; les standards de collectes d'informations et de mesures de performance. Les lois invisibles de paramétrage (appelées aussi informations cachées) sont des lois internes à chaque module, n'affectant pas la modularité générale.

Ces principes de structuration d'un système permettent alors plus facilement de sous-traiter un module ou de récupérer une technologie via l'intégration d'un nouveau module.

Une synthèse de ces principes avec leurs définitions est présentée ci-après.

Tableau 2 : Synthèse des principes de constructions modulaires

Auteur	Principe	Définition
Simon (1969)	Systèmes quasi-décomposables	Décomposer les systèmes de façon hiérarchique est la première manière de réduire la complexité
Orton et Weick (1990)	Attache lâche (<i>Loose coupling</i>)	Plus on a un « <i>loose coupling</i> » entre les modules c'est-à-dire un interfaçage faible, plus les modules sont accessibles à partir de plusieurs plateformes et plus le système devient modularisable et customisable pour le client.
Baldwin et Clark (1997)	Interfaçage : lois visibles de structuration (appelées aussi informations visibles)	Des process de construction et d'assemblage des modules permettent de spécifier que chaque module est une partie d'un système entier et quelles seront leurs fonctions respectives ; le rôle des interfaces et la façon dont les modules interagissent entre eux ; les standards de collecte d'informations et de mesures de performance ;
	Interfaçage : lois invisibles de paramétrages (<i>Information hiding</i>)	Les lois internes à chaque module n'affectent pas la modularité générale. Les modules devraient être caractérisés par des lois de fonctionnements clés et celles-ci devraient être cachées aux autres modules par les interfaces qui révèlent, ainsi aussi peu que possible, chaque fonctionnement interne.

I.2.3 Fondements de l'architecture modulaire d'un système

Dans cette partie, nous présentons un état de l'art sur les méthodes de modularisation d'un produit. Pour mieux comprendre, nous revenons d'abord sur les principaux concepts de l'approche système. Cette approche est utile dans l'analyse et la modélisation des activités de conception, dans la recherche d'architecture d'un système et dans l'organisation d'un système de

conception. Ensuite nous présentons un état de l'art des travaux sur les DSM (*Design Structure Matrix*), afin de présenter les représentations équivalentes d'un système.

I.2.3.1 L'approche système

Von (1968) publie "The general system theory", dans lequel il définit les principes de ce que deviendra la théorie des systèmes. Il préconise d'étudier les interactions dynamiques entre les parties et non pas de se borner à un regard qui tend à isoler (vision réductionniste), à disséquer et à emprisonner les éléments dans une logique de causalité linéaire. Il s'agit d'être particulièrement attentif à la structure des interactions et à leur contrôle.

L'approche système va emprunter les concepts de boucles de rétroaction et de régulation à la cybernétique qui a pour but principal l'étude des régulations chez les organismes vivants et les systèmes construits par l'homme. Elle a ensuite évolué vers l'étude de la complexité, avec une attention particulière aux systèmes dynamiques.

En France, De Rosnay (1975), Le Moigne (1977), Morin (1977, 1992) et Simon (1997) ont contribué à développer une théorie, voire "une science des systèmes", qui se veut d'abord interdisciplinaire et qui vise à rendre compte de phénomènes complexes, en perpétuelle évolution, irréductibles à un modèle analytique de type causal et déterministe. Le macroscopie De Rosnay (1975) est un des principaux ouvrages français d'initiation à l'approche système. Il présente les concepts de causalité circulaire (et non plus seulement linéaire) et l'importance du rôle de la rétroaction (feed-back).

Le paragraphe suivant précise une autre définition d'un système et présente les principes utiles pour la modélisation de son architecture.

I.2.3.1.1 Concepts de système, d'entité et d'interaction

Selon De Rosnay (1975), un système est un "ensemble d'éléments en interaction dynamique, organisés en fonction d'un but". Selon Morin (1977), "on peut concevoir le système comme une unité globale organisée d'interrelations entre éléments, actions ou individus". L'organisation étant alors "l'agencement de relations entre composants ou individus qui produit une unité complexe ou système, dotée de qualités inconnues au niveau des composants ou des individus". Inversement, il précise que "les éléments doivent donc être définis à la fois dans et par leurs caractères originaux, dans et avec les interrelations auxquelles ils participent avec la perspective de l'organisation où ils s'agencent, dans et avec la perspective du tout où ils s'intègrent."

Le concept d'architecture sera utilisé par la suite dans un sens proche de celui que Morin (1977) donne au concept d'organisation. Une architecture sera alors une représentation de l'agencement des relations.

Nous proposons de définir le concept de système comme une représentation (cognitive ou explicite) construite par un sujet : un système est une partie de l'univers qu'un sujet se représente comme une entité, entretenant avec son environnement des interactions pertinentes, et constituée d'entités internes entretenant des interactions mutuelles et externes cohérentes.

Cette intelligibilité du système est requise lorsque nous avons l'intention de le comprendre ou d'agir sur lui, que ce soit pour le concevoir, le structurer, le piloter, l'améliorer, etc. Le système devient alors une entité "agentive " sur, à partir, ou dans laquelle le sujet doit agir (Searle ,1998).

Un système artificiel ou artefact (Simon, 1997 ; Micaelli et Forest, 2003) est conçu et agencé intentionnellement, pour répondre à des besoins (fonctions de service ou missions du système). L'artefact existe non de lui-même, au contraire d'une chose naturelle, mais parce qu'il a été conçu de sorte à assurer sa fonction (Searle ,1998).

Le système peut être lui-même intégré dans différents sur-systèmes, selon les points de vue considérés. Il peut adapter ses frontières et son organisation, afin de prendre en compte l'évolution de son environnement ou pour améliorer son fonctionnement (régulation, apprentissage et pilotage) (Probst et Ulrich, 1989).

I.2.3.1.2 Concept de vues

L'approche système consiste à considérer un système comme une entité dynamique qui entretient avec son environnement des interactions finalisées (approche téléologique, externe, par la pertinence). La modélisation du système s'effectue ensuite à partir des interactions mutuelles entre ses constituants (approche interne, par la cohérence), qui lui confèrent un comportement global cohérent (intégration, émergence).

a. Vue externe (ou contextuelle)

En vue externe, le système est représenté comme une boîte noire échangeant des flux avec son environnement, ainsi que par des scénarios d'échange (flux d'intrants et d'extrants) permettant de fournir des prestations de service à des entités de son environnement, avec lesquels il va interagir au cours de son cycle de vie. Le cycle de vie se décompose en différentes phases de vie, et chaque phase, en modes de fonctionnement. Les prestations attendues par les destinataires correspondent aux missions ou fonctions de service du système. Celles-ci sont

quantifiées selon les performances attendues et spécifiées par leurs critères d'appréciation. La méthode traditionnellement utilisée pour élaborer la vue externe est l'Analyse Fonctionnelle.

b. Vue fonctionnelle interne

En vue interne, le système doit satisfaire un ensemble de fonctions techniques dont l'agencement restitue les fonctions de service. On parle ici d'arborescence puis d'architecture fonctionnelle du système, en cohérence avec (Pahl et Beitz, 1996). L'arborescence fonctionnelle est obtenue en décomposant itérativement les fonctions de service en sous fonctions jusqu'à l'obtention de fonctions pour lesquelles on peut définir des solutions techniques (Meinadier, 2002).

L'élaboration de l'architecture fonctionnelle est utile lorsqu'on cherche à optimiser l'organisation des sous-fonctions (par exemple, regroupement ou factorisation de module, dissociation pour faciliter les évolutions, redondance pour assurer la robustesse lors des évolutions ou pour la sûreté de fonctionnement) (Stone et *al.*, 2000). L'enchaînement des fonctions de l'architecture fonctionnelle permet de réaliser les processus de fonctionnement internes au système, lui conférant ainsi les comportements spécifiés par les fonctions de service et leurs scénarios. Cette architecture fonctionnelle est en théorie indépendante des choix de conception et est donc réutilisable pour des projets semblables. En pratique, la décomposition fonctionnelle ne se fait pas sans préjuger des choix technologiques. La validation de l'architecture technique va nécessiter des itérations, pour modifier et/ou compléter les exigences et l'architecture fonctionnelle.

c. Vue organique (interne)

Dans la vue organique, le système est représenté comme un agencement de constituants (ou d'organes ou de modules) réalisant par leurs interactions les fonctions identifiées dans l'architecture fonctionnelle. Cet agencement est appelé l'"architecture organique" ou technique. Le point clé est la définition des interfaces qui relient les constituants.

d. Vue dynamique (interne)

Pour structurer un système, il est possible de le décomposer en vue externe, en vue fonctionnelle interne et en vue organique, comme nous l'avons présenté précédemment. Ces vues sont des représentations statiques (ou structurelles) du système. Il est possible aussi de recourir à des représentations dynamiques du système (vue dynamique) qui décrivent son évolution temporelle.

Le système peut être considéré comme un ensemble de processus de fonctionnement qui décrivent l'enchaînement des activités des constituants du système, afin de restituer les fonctions de service et leurs performances.

Les normes IEEE 1220 (1998) intègrent dans la définition de l'architecture fonctionnelle, des aspects dynamiques en tenant compte du séquençement de l'exécution des fonctions, des flux de données et de contrôle qui le conditionnent. Par la suite, nous parlerons d'architecture fonctionnelle pour désigner uniquement l'arrangement des fonctions et leurs interactions.

Même si la modélisation dynamique d'un système est primordiale pour certaines activités de conception comme la détermination des lois de commande ou l'analyse des modes de fonctionnement / dysfonctionnement, nous ne présenterons pas les travaux de recherche concernant les langages et outils de modélisation dynamique, car ils sont en dehors de notre champ de recherche. Une comparaison de langage et d'outils pourra être trouvée dans (Baron, 2005).

e. Complication et complexité

Nous n'entrerons pas dans un débat philosophique sur le sens de ces termes, mais il est intéressant de préciser leur différence.

La complication est une propriété structurelle. Un système compliqué comprend un nombre élevé de composants et de relations (ils ont une cardinalité élevée). D'où l'intérêt, dès qu'un système est identifié comme tel, de le considérer comme "semi-décomposable" (Simon, 1997). Pour mieux le comprendre ou le maîtriser, une technique performante consiste alors à le décomposer par strate, à le structurer en modules selon Ulrich et Eppinger (2000), et à garantir l'ajustement des modules d'une même strate en recourant éventuellement à un module particulier, dit intégratif (qui crée la cohésion du système).

La complexité est une propriété à la fois structurelle et dynamique. Pour Micaëlli et Forest (2003), un système est complexe s'il doit être appréhendé à l'aide de différentes vues (ou modèles), appariées par des relations non triviales (aspect structurel), et si cette modélisation ne permet pas de rendre compte de l'ensemble des états possibles du système du fait de sa dynamique (aspect dynamique).

Reconnaître un système comme complexe conduit à raisonner en termes de vues.

Il s'agit d'abord d'identifier et d'apparier plusieurs vues (ou modèles), une seule, isolée, ne pouvant suffire (Suh, 1990 ; Simon, 1997). Par exemple pour un artefact, vue fonctionnelle et vue organique ou pour un acteur, vue cognitive et vue collective.

I.2.3.1.3 Concept d'architecture, de module et d'élément intégrateur

Le terme architecture recouvre différents sens selon les auteurs qui le manipulent. Deux définitions complémentaires peuvent être pertinentes.

La première considère que l'architecture d'un produit réside essentiellement dans les choix de passage de l'espace fonctionnel à l'espace organique, chacun de ces espaces ayant aussi une architecture propre. La seconde considère que l'architecture est une vue interne à un ensemble d'éléments inter-reliés.

Les travaux de deux auteurs, Ulrich et Eppinger, sont fréquemment cités par les chercheurs travaillant sur l'architecture de produit. Ulrich et Tung (1991) et Ulrich (1995) définissent l'architecture du produit comme :

- l'arrangement des fonctions ;
- l'allocation des fonctions aux composants ;
- la spécification des interfaces entre les composants.

En France, le point de vue de Meinadier (2002) sur l'architecture du produit rejoint celui d'Ulrich. On parle alors d'architecture fonctionnelle et d'architecture organique. Au cours des décompositions conduisant à ces architectures, les exigences sont allouées (réparties) sur les éléments de la décomposition et les interfaces sont définies. La Figure 4 présente une allocation des fonctions aux constituants.

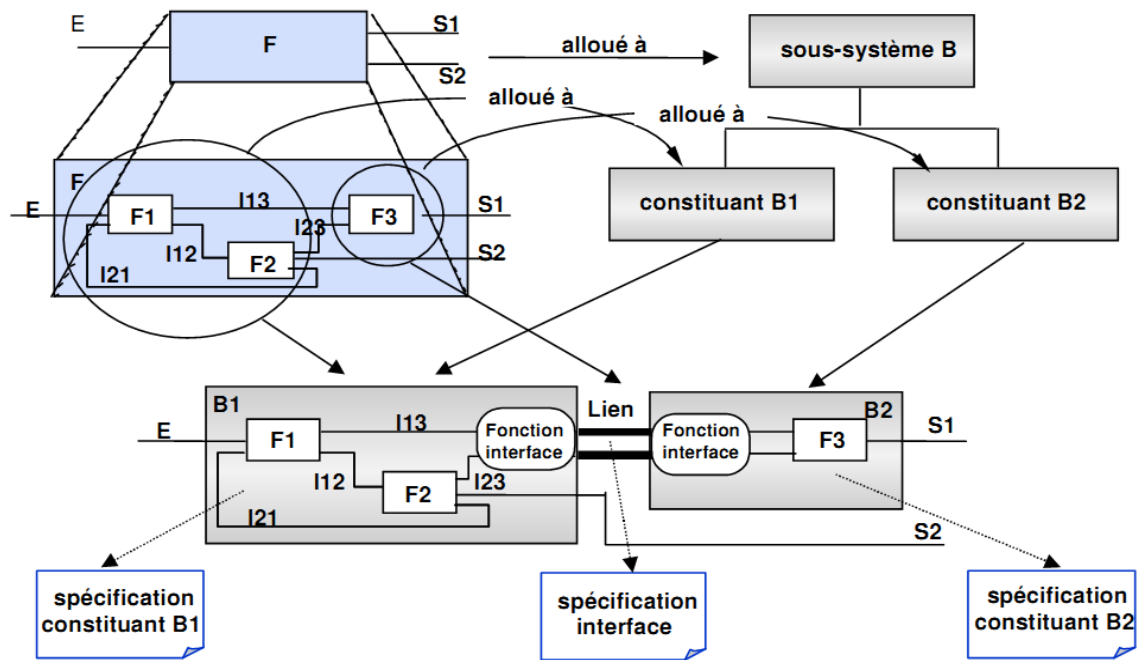


Figure 4 : Allocation des fonctions aux constituants (Meinadier, 2002)

D'autres définitions du concept d'architecture se rejoignent sur le fait que l'architecture d'un système est la description des éléments qui le composent et des interactions qui les lient (Pimmler et Eppinger, 1994 ; Van Wie et *al.*, 2001 ; Crawley et *al.*, 2004 ; Chen et Liu, 2005 ; Holttä-Otto, 2005). Dans le cadre de la conception concurrente, cette définition est enrichie par la prise en compte de l'ensemble du cycle de vie du produit : "l'architecture du produit est la structure des composants, leurs interactions et les principes qui gouvernent leur conception et évolution tout au long du cycle de vie" (Dahmus et *al.*, 2001).

Le choix de la définition d'une architecture est déterminant dans la façon d'identifier un module.

La typologie d'Ulrich (1995) met en évidence deux types d'architectures du produit :

- Une architecture modulaire, dans laquelle l'espace des fonctions peut être décomposé en sous-ensembles de telle manière que chaque sous-ensemble impacte distinctement un ou un ensemble de composants. Cela rejoint le concept de conception découplée, défini par (Suh, 1990).
- Une architecture intégrale, dans laquelle toutes les fonctions et tous les composants sont fortement liés. Cette définition rejoint le concept de conception couplée, défini par Suh.

Sosa et *al.* (2003) proposent d'introduire un autre type d'architecture, qui est plus proche des cas réels : une architecture hybride composée à la fois de modules (groupements d'éléments) et d'éléments intégrateurs, un élément pouvant être une fonction, un constituant...

Nous appuyant sur les travaux précédents ainsi que sur ceux de Baldwin et Clark (2000), nous définissons ainsi les concepts de module et d'élément intégrateur :

- un module est une entité constituée d'un ensemble d'éléments fortement liés entre eux et faiblement liés à d'autres éléments externes au module.
- un élément intégrateur est un élément appartenant à un module qui interagit fortement en nombre ou en intensité avec des éléments appartenant à d'autres modules.

L'intérêt de cette définition est sa cohérence avec une représentation matricielle des interactions entre les éléments d'un domaine et l'utilisation d'un algorithme pour identifier une architecture « optimale » de ce domaine.

I.2.3.2 Conception modulaire des produits

La **conception modulaire** est une des méthodes de conception de produits permettant la diversification de l'offre (Saïd, 2018 ; Gibaru Olivier, 2020). Un intérêt considérable s'est porté dans les années récentes sur la conception modulaire dans les secteurs (tels que l'informatique, l'automobile, l'aéronautique, etc) où la réactivité constitue un facteur-clé de succès. L'objectif de cette orientation stratégique est de réduire simultanément la complexité, les temps et les coûts de fabrication.

Dans cette partie, nous présentons d'abord la modularité et les familles de produits. Ensuite nous présentons les avantages et les inconvénients de la conception modulaire. Nous réalisons ensuite une synthèse bibliographique sur les méthodes de modularisation d'un produit. Enfin, nous présentons succinctement les outils matriciels de représentation d'une architecture (DSM et matrices d'incidence) et positionnons le problème lié à la modularité.

I.2.3.2.1 Modularité et familles de produits

Plusieurs politiques de conception de produits ont été proposés pour offrir une grande variété de produits finis à savoir :

- la différenciation retardée ;
- la conception modulaire ;
- la standardisation.

La conception modulaire consiste à assembler un produit à partir de modules fonctionnels. Un module est constitué d'un ensemble d'éléments fortement liés entre eux et faiblement à d'autres éléments externes au module (Baldwin et Clark, 2000).

Une famille de produits est un ensemble de produits ayant une ou plusieurs similarités dans leurs caractéristiques (fonctions, composants, processus d'assemblage, technologie) qui satisfont les exigences particulières des clients. Chaque produit individuel de la famille de produits est appelé variante du produit.

Pahl et Beitz (1996) proposent la classification suivante des modules :

- *Module basique* : il implémente une ou plusieurs fonctions fondamentales du produit ou de la famille de produits.
- *Module auxiliaire* : il implémente une ou plusieurs fonctions auxiliaires qui sont utilisées en association avec les fonctions fondamentales pour créer de la variété dans la famille de produits.
- *Module adaptatif* : il regroupe les fonctions qui permettent d'adapter un composant (ou un système) à un autre produit ou (à un autre système). Les modules adaptatifs permettent de gérer les contraintes imprévisibles.
- *Non module* : il implémente des fonctions spécifiques dictées par les besoins des clients. Ces fonctions ne sont pas partagées avec d'autres produits de la famille.

Dans sa thèse, Mtopi (2006) utilise une autre typologie plus adaptée aux familles de produits.

Il utilise les concepts suivants :

- ✚ le « produit de base », rassemble tous les modules implémentant les fonctions principales de la famille et est considéré comme un « module composé »,
- ✚ un module indécomposable est dit « primitif »,
- ✚ un module commun, correspond à une ou plusieurs fonctions présentes dans tous les produits de la famille,
- ✚ un module distinctif, introduit la variété dans la famille.

La modularité a aussi été associée à la notion de similitude dans le sens où un module regroupe des éléments ayant des caractéristiques similaires (par exemple, des composants ayant la même géométrie, des fonctions liées à la même discipline scientifique comme la thermique, l'automatique) (Huang et Kusiak, 1998).

I.2.3.2.2 Avantages et inconvénients de la conception modulaire

Dans les mémoires de thèse de Mtopi et de Harmel, nous avons dressé un inventaire des avantages et inconvénients de la conception modulaire (Huang, 2000 ; Baldwin et Clark, 2000 ; Oosterman, 2001 ; Holtta et Salonen, 2003). Retenons ici les quatre points essentiels :

- Aider à maîtriser la complexité en séparant le système en modules inter-reliés, avec une définition appropriée des interfaces entre modules. Cette approche permet d'envisager des opérations de maintenance ou des évolutions technologiques de modules au cours du cycle de vie du système, sans affecter l'ensemble du système (Newcomb et *al.*, 1998 ; Dahmus et *al.*, 2001).
- Permettre des économies d'échelle : lorsqu'on augmente les quantités de production, on réduit les coûts par un meilleur amortissement des coûts fixes. Or, la conception modulaire permet la réutilisation de certains modules ou composants (standardisation) et donc l'augmentation des quantités (Sanchez, 1999). Cela permet aussi de générer de l'économie de variété.
- Réduire les délais de développement ("Time to market") par une optimisation des processus de conception (Browning (1998, 1999)). Si les interfaces entre les composants sont bien définies, il est possible de paralléliser les activités de conception, et donc de réduire les délais de développement.
- Faciliter la mise en place d'équipes de conception, adaptées à l'architecture du produit. Les échanges entre équipes et leur coordination sont prévus dans le cadre du management. Ils sont réduits par une définition appropriée des interfaces (Baldwin et Clark, 1997 ; Ulrich et Eppinger, 2000 ; Blackenfelt, 2001 ; Sosa et *al.*, 2003). Se spécialisant sur la conception d'un module, l'équipe développe une compétence collective qui peut devenir stratégique si elle est source d'innovation et d'avantage compétitif (Sanchez et *al.*, 1996). Elle favorise aussi l'externalisation du développement de modules.

Mais la conception modulaire a aussi quelques inconvénients (Oosterman, 2001) :

- Une utilisation intensive de la modularisation rend les produits tous semblables.
- La modularisation fait augmenter les risques de voir les produits copiés par les entreprises concurrentes.
- La conception initiale des modules est très difficile en raison de la complexité des contraintes sur les interfaces.

- Plusieurs auteurs affirment qu'une architecture modulaire est un frein pour la conception de produits performants à cause du compromis que réalise une architecture modulaire parmi tous les produits.
- La conception modulaire peut engendrer des surcoûts, si par soucis de conserver les mêmes interfaces, on remplace un composant par un autre plus cher.

Cette synthèse montre que les entreprises s'engageant dans une stratégie de conception modulaire de ses produits peuvent en tirer des bénéfices importants, mais doivent être conscientes des limites et des risques de la modularisation.

I.2.3.2.3 Méthodes de modularisation du produit

Les outils d'aide à l'identification des modules dans un produit sont peu nombreux dans la littérature de l'ingénierie de la conception. Parmi elles, trois méthodes sont fréquemment référencées :

- Méthode heuristique pour l'architecture des fonctions,
- Méthode MFD (Modular Function Deployment),
- Méthode de "clustering" d'une matrice structurelle de conception (en anglais DSM pour Design Structure Matrix).

Une étude comparative de ces trois méthodes a été réalisée par Holtta et Salonen (2003). Pour notre part, dans cette partie, nous ne présenterons que les deux premières méthodes. La troisième étant celle que nous allons utiliser dans ce travail, elle sera traitée en détail dans la suite du travail.

a) Méthode des heuristiques

Développé par Stone (1998) et Stone et *al.* (2000), elle est basée sur **trois** heuristiques qui permettent d'identifier les modules fonctionnels d'un produit. Un ensemble de fonctions de ce produit interagit par des flux d'informations (flux dominants, flux ramifiés, flux convertis) pour former des modules physiques (Figure 5). Les principaux critères de modularisation sont fonctionnels et indépendants de l'architecture physique du produit. Les trois heuristiques sont :

- **Heuristique des flux dominants** : elle examine les flux à travers une arborescence fonctionnelle et suit les flux jusqu'à ce qu'ils sortent du système ou subissent une transformation. Elle conclut que s'il existe un ensemble de fonctions à travers lesquelles un flux passe jusqu'à la sortie ou la transformation, alors ces fonctions forment un module.

- **Heuristique des flux ramifiés** : elle examine les flux divergents ou convergents issus de chaînes de fonctions parallèles. Chacune de ces chaînes peut devenir un module dont l'interface correspond au point de branchement de la chaîne.
- **Heuristique du module de conversion-transmission** : elle examine les flux convertis. Ces flux sont convertis au niveau de certaines fonctions. On considère alors ces fonctions comme étant des modules à elles seules. Si dans une même chaîne de fonctions, il y a plusieurs fonctions de transmission ou de conversions, alors ces fonctions délimitent un module de conversion ou de transmission.

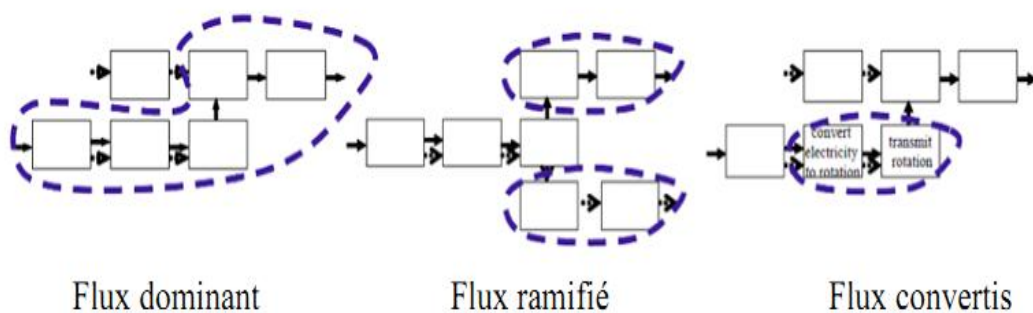


Figure 5 : Heuristiques pour l'architecture des fonctions

Zamirowski et Otto (1999) proposent trois heuristiques additionnelles pour identifier les fonctions partagées dans un produit unique ou dans une famille de produit et les fonctions uniques propres à un produit donné. Ainsi, les fonctions qui partagent des flux similaires ou qui apparaissent plusieurs fois peuvent être regroupées pour former un unique module. Les fonctions uniques sont celles spécifiques à un seul produit, elles peuvent être regroupées pour former un module.

Nous remarquerons que les principaux critères de modularisation sont fonctionnels et indépendants de l'architecture physique du produit.

Cette méthode est présentée comme étant idéale pour la conception de famille de produits d'après Holtta et Salonen (2003), mais elle présente le défaut d'être dépendante de l'interprétation et de la décision humaine. Elle ne peut pas être automatisée et de ce fait ne peut pas être appliquée sur des produits complexes (Otto et Wood, 2001). Pour des présentations plus approfondies de cette méthode et surtout de la démarche de mise en œuvre, nous conseillons la lecture des travaux de (Dahmus et *al.* 2000 ; Otto et Wood, 2001 ; Holtta et Salonen, 2003).

b) Méthode MFD (Modular Function Deployment)

C'est une méthode de modularisation plus orientée management qu'ingénierie. Elle a été développée par (Erixon, 1996 ; Ericsson et Erixon, 1999) et est dédiée à la modularisation d'un seul produit. Dans cette méthode, des critères de modularité autres que la fonctionnalité sont considérés. Il y a **douze critères** de modularité dans la méthode MFD (Tableau 3).

Un ou quelques critères de modularité sont choisis en fonction de la stratégie de développement de produit modulaire de l'entreprise. L'entreprise applique ensuite la démarche de MFD qui comprend **cinq** étapes :

1. Clarifier la spécification du produit en utilisant une matrice QFD avec les pilotes de modularité qui remplacent les besoins des clients.
2. Analyser les fonctions et sélectionner les solutions techniques.
3. Identifier les modules possibles en utilisant la matrice MIM (Module Indication Matrix). Celle-ci sert à examiner les corrélations entre les critères de sélection de modules (normalement, ce sont les critères en relation avec les activités du cycle de vie du produit, par exemple facilité de conception, recyclage, etc.) et les composants sélectionnés dans l'étape antérieure. Un questionnaire est fourni pour soutenir cette activité. Les composants (verticalement) sont évalués par rapport aux critères (horizontaux) selon le questionnaire fourni (Figure 6). Les composants avec une meilleure note sont appelés « porteurs ». Le reste des composants qui ont une bonne note dans les mêmes critères que les composants porteurs sont assemblés avec ces derniers pour former des modules.
4. Évaluer les concepts en testant les interfaces entre les modules.
5. Améliorer chaque module.

L'identification des modules selon la méthode MFD se fait en classant les fonctions selon un score croissant. Les fonctions participant à un même critère de modularité sont candidates pour former un module.

Tableau 3 : Les critères de modularité dans la méthode MFD

Développement du produit	Reconduction	Une partie du produit peut être réutilisée et reconduite dans plusieurs produits.
	Évolution technologique	Les pièces sont susceptibles de subir des changements en raison des demandes de clients ou des innovations technologiques. Il est important de pouvoir adapter

		les interfaces de sorte que la nouvelle technologie puisse être introduite sans remettre en cause l'ensemble du produit.
	Modifications planifiées	L'entreprise a l'intention de développer et de changer certaines parties du produit.
Variabilité	Spécifications variables	Pour prendre en compte la personnalisation des produits efficacement, un concepteur devrait tâcher d'assigner toutes les variations à un petit nombre de composants.
	Style	Les modules de style contiennent les composants qui peuvent être altérées pour créer des variations du produit.
Production	Unité en commun	Les composants et fonctions en commun entre plusieurs produits permettent le partage d'une unité de production.
	Processus et/ou organisation	Des pièces exigeant le même processus de production sont groupées ensemble.
Qualité	Essais séparés	La possibilité de tester séparément chaque module avant la livraison à l'assemblage final peut contribuer à l'amélioration.
Achats	Disponibilité des fournisseurs	Modules standard d'achat disponibles chez les fournisseurs externes.
Après-vente	Service et maintenance	Les composants nécessitant une maintenance peuvent être groupés ensemble pour former un module qui facilite les opérations de remplacement et de réparation.
	Amélioration	Il faut donner aux clients la possibilité de faire évoluer leur produit.
	Recyclage	Le nombre de matériaux dans chaque module devrait être limité. Les matériaux facilement recyclables

		peuvent former un module séparé.
--	--	----------------------------------

	Fonction 1	Fonction 2	Fonction 3	Fonction 4	Fonction 5	Fonction 6	Fonction 7	Fonction 8	Fonction 9
Reconduction	3			1					
Évolution technologique		3	9		3			9	
Modifications planifiées		1	3			3		9	
Spécifications variables	1				3				
Style		3		9		9		1	
Unité en commun	9		3				9		
Processus et/ou organisation	3	1			3				
Essais séparés	3		3					1	9
Disponibilité des fournisseurs									9
Service et maintenance		9			9				
Amélioration			1		3			1	
Recyclage				3					3

Figure 6 : Matrice Indication Module

La méthode MFD présente l'avantage de laisser le choix aux ingénieurs de définir les stratégies de modularisation en jouant sur les pilotes. Ce même avantage est l'un des inconvénients de la méthode qui reste dépendante des ingénieurs dans le choix des modules et de leurs tailles, ceci a pour conséquence de limiter la possibilité d'automatiser la méthode et d'avoir la même architecture d'un utilisateur à un autre (faible reproductibilité).

Les deux méthodes de modularisation présentées ci-dessus montrent certes plusieurs avantages, mais aussi certaines limites :

- Elles dépendent des choix des ingénieurs à certains moments de la démarche de modularisation.
- Elles ne peuvent pas être automatisées et sont difficilement exploitables sur des exemples complexes où une assistance par un traitement informatique serait souhaitable.
- Elles sont essentiellement adaptées à l'espace fonctionnel.

Ces limites font que, nous opterons pour une autre méthode de modularisation, à savoir la méthode de clustering des matrices DSM. Celle-ci permet à la fois de modéliser les

architectures par les DSM et de générer des architectures modulaires « satisfaisantes », à travers l'utilisation d'un algorithme de *clustering*.

I.2.3.2.4 Outils matriciels de représentation d'une architecture DSM

DSM est l'acronyme de « **Design Structure Matrix** » ce qui est traduit par **matrice structurelle de conception**. Dans la suite de ces travaux, nous allons utiliser le terme DSM pour faire référence à une matrice structurelle.

Une DSM est un outil de modélisation, utilisé dans les projets de conception et dérivé des matrices binaires avec l'introduction de nouvelles capacités et possibilités. Nous présentons dans ce qui suit les fondements de cet outil matriciel et ses apports dans la modélisation des domaines d'un projet.

I.2.3.2.4.1 Classification des DSM

Il existe deux classes de DSM, en lien avec la classification utilisée pour les matrices binaires (Ghassen, 2008). On retrouve ainsi la classe des matrices symétriques sous le nom de DSM statiques et la classe des matrices asymétriques sous le nom de classe des DSM temporelles (figure 7). Les DSM statiques tirent leur nom du fait qu'elles sont utilisées pour matérialiser les interactions entre les éléments d'un système, indépendamment du temps. Les deux sous-classes les plus utilisées des DSM statiques sont les DSM Produit utilisées pour représenter l'architecture des produits et les DSM Organisation (ou Acteurs) pour modéliser l'organisation des acteurs dans un projet ou dans l'entreprise. Les DSM temporelles quant à elles permettent de représenter les systèmes dont les éléments sont liés par des relations de précedence, qui peuvent être des relations de précedence de type temporel (DSM Processus) ou des relations de hiérarchie ou de contrainte (DSM Paramètres).

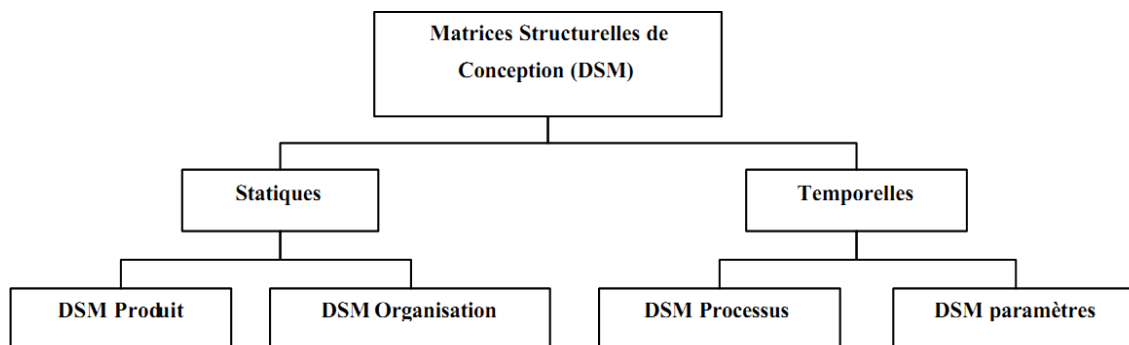


Figure 7: Classification des DSM

a) Les DSM temporelles

La terminologie exacte utilisée pour faire référence à la modélisation matricielle des systèmes hiérarchisés est « Dependency Structure Matrix », ce qui permet d'obtenir le même acronyme DSM.

Les deux sous-classes les plus connues des DSM temporelles sont les DSM processus et les DSM paramètres.

Les DSM Processus. Elles permettent de modéliser l'enchaînement des tâches de tous les processus de l'entreprise. Dans l'entreprise, les tâches déterminent principalement l'allocation des ressources et servent de base pour estimer la durée et le coût du projet. C'est pourquoi les deux grands domaines d'application des DSM temporelles sont :

- L'optimisation des processus du point de vue coût et délais : nous citerons les travaux de Yassine et *al.* (2001) sur les probabilités de reconception, ceux de Carrascosa et *al.* (1998) sur l'estimation des délais de conception et ceux de Browning et d'Eppinger (2002) sur le lien entre le séquençement des tâches et les coûts de développement.
- L'optimisation des processus pour la conception des équipes de conception : nous citerons les travaux de (Eppinger et *al.*, 1994 ; Browning ,1999 ; Braha, 2002 ; Chen et Lin, 2003).

Les DSM Paramètres. Elles ont été initialement utilisées pour modéliser les relations de précedence et les contraintes entre les paramètres et variables en conception. Elles s'appliquent plus généralement à tous les systèmes dont les éléments ne sont pas homogènes. Elles permettent d'optimiser les boucles de conception entre groupes de paramètres (Browning, 2001).

L'objectif général lorsqu'on analyse des DSM temporelles est de réorganiser les éléments en ligne et en colonne de manière à réduire les retours en arrière lorsqu'il s'agit de tâches, ou d'identifier une hiérarchie entre les éléments. Sur la figure 8, si on considère que les activités s'enchaînent de A vers K, alors on remarque qu'il y a trois retours en arrière (les croix au-dessus de la diagonale).

Cette réorganisation est assurée par des algorithmes de séquençement et de partitionnement.

ACTIVITES		A	B	C	D	E	F	G	H	I	J	K
Recevoir spécifications	A	A										
Générer concepts	B	X	B									
Concevoir les modèles	C	X	X	C								
Produire les modèles	D			X	D							
Développer programme de tests	E	X	X	X		E						
Tester les modèles	F			X	X	X	F					
Concevoir les plans	G	X	X	X			X	G	X	X		
Concevoir les moules	H	X	X				X	X	H	X		
Concevoir les outils d'assemblage	I							X	X	I		
Fabriquer les moules	J								X		J	
Initialiser la production	K											K

Figure 8 : Exemple de DSM temporelle

Le partitionnement est le processus de réorganisation des lignes et des colonnes d'une DSM temporelle de sorte qu'on réduise les interactions représentant les retours en arrière. Cela revient à rendre la DSM le plus possible triangulaire. Cependant cette action n'est pas toujours synonyme de réduction de la durée globale du processus modélisé. En effet, l'exemple en figure 9 montre un processus composé de 7 tâches dont les relations de précédence sont modélisées par une DSM. La figure 9 (a) montre le résultat du partitionnement qui a pour but de réduire les boucles en arrière, on remarque qu'on a uniquement deux sauts en arrière de E vers A et de F vers C. On suppose que le processus en question est fini lorsqu'on passe au moins une fois par toutes les tâches et que toutes les tâches ont des durées équivalentes, alors on obtient l'enchaînement suivant (A, E, C, F, B, A, C, A, D, B, E, D, F, E, G, F) avec 16 tâches. On remarque alors, que plus les interactions sont éloignées de la diagonale, plus la boucle est grande.

Pour remédier à ces problèmes d'allongement des processus, une alternative est de réarranger les lignes et les colonnes des DSM temporelles de manière à ce que les interactions soient les plus proches de la diagonale. Ainsi, on augmente le nombre des boucles arrière, mais ces boucles seront plus courtes. La figure 9 (b) montre qu'avec la deuxième technique de séquençement, on obtient 4 boucles mais un processus plus court (A, E, E, D, D, B, C, F, E, A, F, C, G, F) avec 14 tâches.

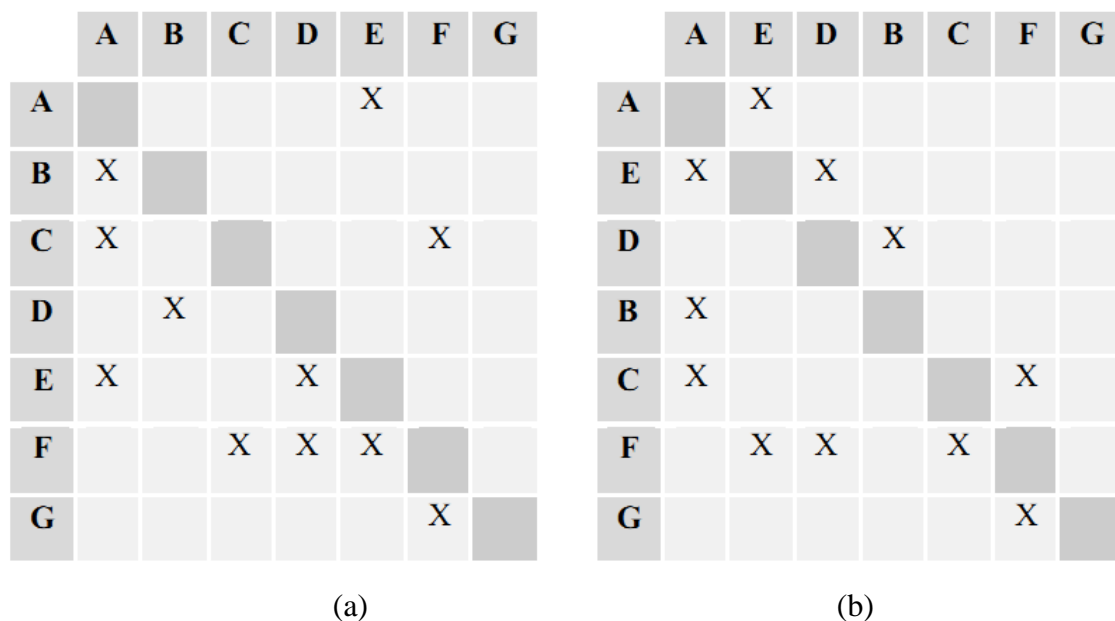


Figure 9 : Deux résultats de partitionnement d'une DSM

b) Les DSM statiques

Les DSM statiques par opposition aux DSM temporelles permettent de modéliser d'une manière non hiérarchique les interactions entre les constituants d'un système, ce système pouvant être un produit ou une organisation.

Historiquement, les DSM statiques binaires ont été choisies d'après Kusiak et *al.* (1993) pour modéliser l'architecture des produits et modéliser les interactions qui lient leurs composants. Ces DSM sont binaires et permettent d'identifier l'existence ou non d'un type d'interaction étudiée. Les DSM statiques binaires sont encore utilisées malgré l'apparition des DSM numériques (Bohm et *al.*, 2004).

Une DSM Produit modélise les interactions entre les constituants d'un produit. Pimmler et Eppinger (1994) proposent une taxonomie (tableau 4) permettant de différencier les différents types d'interactions. Cette taxonomie permet ainsi de construire des DSM produits où les cellules, autres que les diagonales, contiennent des vecteurs de dimension 4 contenant des valeurs binaires.

Tableau 4 : Taxonomie des interactions du Produit (Pimmler et Eppinger, 1994)

Type de DSM	Type d'interaction	Explication
Produit	Spatiale	Lié à la propagation de contraintes physiques
	Energie	Lié aux flux d'énergie

	Information	Lié au flux d'information
	Matériel	Lié au flux de matière

Meinadier (2002) justifie l'utilisation des matrices DSM auxquelles il donne le nom de matrices de couplage et d'intégration, car elles permettent en effet de représenter et visualiser facilement les interactions (de différentes natures) entre les constituants. Elles sont utiles d'abord lors de la définition de l'architecture organique (regroupement de modules, optimisation des couplages) et ensuite lors de l'intégration pour vérifier que les zones de couplages et les interfaces entre constituants sont bien réalisées et vérifiées.

Avec une DSM « Organisation », on modélise les interactions entre les acteurs dans une équipe de conception. Parmi les travaux les plus récents utilisant les DSM Organisation binaires, on citera ceux de (Sosa et *al.*, 2004). Dans ses travaux, on propose la taxonomie des interactions présentée dans le tableau 5.

Tableau 5 : Taxonomie des interactions entre acteurs (Sosa et *al.*, 2004)

Type de DSM	Type d'interaction	Explication
Acteurs	Qualité	Lié aux type de moyens de communication (direct, mail, document, etc.)
	Fréquence	Lié à la fréquence des interactions
	Direction	Lié à des interactions
	<< Timing >>	Lié à la référence des interactions par rapport au projet

I.2.3.2.4.2 Synthèse sur les types de DSM

Le tableau 6 résume les principaux types de DSM utilisées dans la modélisation de projet.

Tableau 6 : Récapitulatif des caractéristiques des DSM

Type de DSM	Représentation	Application	Méthode d'analyse
Produit	Interaction entre composants	Conception et architecture des systèmes	Clustering
Acteurs	Caractérisation des interfaces entre acteurs	Conception d'organisation, gestion des interfaces entre acteurs, coopération et collaboration, flux d'information	Clustering

Activités	Relations Input/Output entre activités	Structuration des processus, planification des projets	partitionnement
Paramètres	Précédences et hiérarchies entre paramètres	Conception et architectures des systèmes à composants non homogènes	partitionnement

Avant la numérisation proprement dite des DSM, on assistait à une forme intermédiaire entre les DSM binaires et celles numériques. Il s'agit des DSM utilisant une métrique discrète, la plupart du temps basée sur une sémantique linguistique du genre Faible, Moyen et Fort. Ainsi, dans les travaux de Sosa (2005), on retrouve une métrique utilisant les classes Fort, Faible et Nul.

I.2.3.2.4.3 Les DSM numériques

Dans les DSM binaires, un seul attribut est utilisé pour caractériser les interactions à l'intérieur d'un système à savoir l'existence ou l'absence d'une interaction. Une quantification des interactions modélisées dans les DSM permet de relativiser le poids des interactions et ainsi d'augmenter la qualité informative des DSM.

a) DSM temporelles numériques

Concernant les DSM temporelles, nous pouvons retenir les travaux d'Eppinger et *al.* (1994) qui proposent de rendre numérique une DSM processus (Figure 10) en portant sur la diagonale la durée de chaque tâche et en transformant la représentation binaire des interactions en évaluation numérique du degré de dépendance entre les tâches (valeur comprise entre 0 et 1).

Le traitement proposé par les auteurs est de simuler le partitionnement qui réduit la durée totale du processus modélisé.

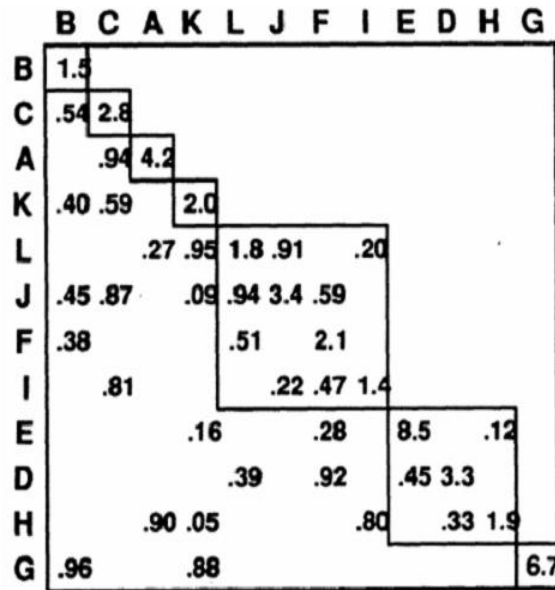


Figure 10 : Exemple de numérisation des DSM temporelles (Eppinger et al., 1994)

La question qu'on peut se poser lorsqu'on veut faire évoluer le caractère informatif des DSM binaires est : quels attributs rendre numériques ?

Dans la littérature, on trouve plusieurs méthodes de numérisation des DSM temporels. Dans ce qui suit, nous allons passer en revue les plus connues :

- Steward (1981), qui suggère d'utiliser des valeurs numériques pour caractériser l'ordre dans lequel les tâches de retour en arrière sont traitées. La tâche avec la plus haute valeur est celle qui doit être analysée en premier.
- Yassine et al. (2001), qui proposent de quantifier le degré d'importance des interactions. Une échelle linguistique est utilisée pour caractériser les interactions : 1 = importante, 2 = moyenne, 3 = faible dépendance. Dans ce cas, on obtient trois classes de dépendance ordonnées selon leur degré d'importance.
- On peut caractériser la force du lien de dépendance entre les tâches. Deux mesures sont usuellement utilisées, soit entre 0 et 1 (Eppinger et al., 1994), soit entre 1 et 10 (Yassine et al., 2000). L'algorithme de partitionnement utilisé a pour but alors de minimiser une somme globale des dépendances dans la DSM.
- On peut caractériser le volume des flux d'information. L'algorithme de partitionnement a les mêmes objectifs que celui du cas précédent (Guivarch, 2003).
- On peut caractériser la variabilité de l'information échangée. Cette mesure de variabilité reflète l'incertitude sur les échanges d'information entre tâches. Cette mesure peut être

construite statistiquement ou évaluée subjectivement par les acteurs du projet (Yassine, 2004).

- On peut caractériser la probabilité de répétition, ainsi une valeur numérique renseigne sur la probabilité de causer des répétitions (Smith et *al.*, 1997 ; Browning, 1998 ; Cho et Eppinger, 2001).
- Carrascosa et *al.* (1998) proposent de construire et de caractériser les couplages entre tâches par des probabilités, puis de provoquer des changements, ce qui représente une vision plus large que le modèle présenté dans le point précédent.

Avant de définir les DSM statiques numériques, il est important d'ajouter que dans certains travaux, nous trouvons une modélisation du domaine des activités et des tâches par des DSM statiques numériques (Chen et Lin, 2003). L'objectif est alors d'identifier les regroupements entre tâches pour les affecter par la suite à des acteurs collectifs (des équipes).

b) DSM statiques numériques

Les travaux d'enrichissement de l'outil DSM se sont intéressés tout d'abord aux DSM temporelles, ce qui a ouvert la porte à la numérisation des DSM statiques et plus spécifiquement à la numérisation des DSM Organisation. En effet, les premiers travaux sur la numérisation des DSM statiques ont été réalisés par McCord et Eppinger (1993) et ont porté sur la construction d'une DSM acteurs en partant des tâches que les acteurs réalisent. Ils proposent alors de quantifier les couplages entre les acteurs en utilisant une DSM temporelle numérique qui évalue l'intensité des dépendances entre les tâches.

La formalisation de la numérisation des DSM organisation a été réalisée quand les premières taxonomies des types d'interactions entre acteurs ont été proposées, ces taxonomies servaient de base pour la proposition de métriques. Morelli et *al.* (1995) ont utilisé la « qualité » comme métrique pour qualifier les moyens de communications entre les acteurs. Les travaux les plus récents quant à eux utilisent beaucoup plus la fréquence des interactions comme métrique (Sosa, 2004).

Quant aux DSM produit, certains travaux pionniers dans la numérisation proposent une métrique discrète qui s'applique à tous les types d'interactions identifiées (Pimmler et Eppinger, 1994). Cette métrique représentée dans le tableau 7 classe les interactions en 5 types.

La DSM représentée dans la figure 11 montre l'une des premières applications des DSM Produit numériques. Le produit en question est un moteur thermique.

Tableau 7 : Métrique pour les DSM Produit (Pimmler et Eppinger, 1994)

Type de DSM	Type	Métrique	
Produit	Spatiale	Nécessaire :(+2)	L'interaction est nécessaire pour le fonctionnement du produit
	Energie	Désiré :(+1)	L'interaction est bénéfique mais non absolument nécessaire
	Information	Indifférent : (-1)	L'interaction n'affecte pas le fonctionnement du produit
	Matériel	Non-désirée :(-1) Nuisible : (-2)	L'interaction affecte le fonctionnement du produit L'interaction empêche le bon fonctionnement du produit

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator A		2 0 0 2			2 -2 0 0											
Engine Fan B	2 0 0 2				2 0 0 2								1 0 0 0			
Heater Core C				1 0 0 0			2 0 0 0	-1 0 0 0								0 0 0 2
Heater Hoses D			1 0 0 0						-1 0 0 0							
Condenser E	2 -2 0 0	2 0 0 2				0 2 0 2		-2 2 0 2								
Compressor F					0 2 0 2			0 2 0 2	1 0 0 2	0 0 2 0	0 0 2 0		1 0 0 0			
Evaporator Case G			2 0 0 0					2 0 0 0						2 0 0 0	2 0 0 0	2 0 0 2
Evaporator Core H			-1 0 0 0		-2 2 0 2	0 2 0 2	2 0 0 0		1 0 0 2							0 0 0 2
Accumulator I				-1 0 0 0		1 0 0 2		1 0 0 2		1 0 0 0						
Refrigeration Controls J						0 0 2 0			1 0 0 0		0 0 2 0		1 0 0 0			
Air Controls K						0 0 2 0				0 0 2 0		0 0 2 0	1 0 0 0	0 0 2 0	0 0 2 0	
Sensors L											0 0 2 0		1 0 0 0			
Command Distribution M		1 0 0 0				1 0 0 0				1 0 0 0	1 0 0 0	1 0 0 0		1 0 0 0	1 0 0 0	1 0 0 0
Actuators N							2 0 0 0						0 0 2 0	1 0 0 0		
Blower Controller O							2 0 0 0					0 0 2 0	1 0 0 0			2 0 0 2
Blower Motor P			0 0 0 2				2 0 0 2	0 0 0 2					1 0 0 0		2 0 0 2	

Figure 11 : Exemple de DSM Produit (Pimmler et al., 1994)

Une autre métrique très répandue dans les DSM produit numériques, est celle qui utilise des valeurs numériques réelles la plupart du temps positives (bornées ou pas) (Rodriguez-Torro, 2004 ; Clarkson et al., 2004 ; Larses, 2005).

I.2.3.2.5 Positionnement du problème de la modularité

Par la modularité, il peut être possible de développer un certain nombre d'avantages de conception (Sanchez et Mahoney, 1996 ; Baldwin et Clark, 1997 ; Ethiraj et Levinthal, 2004). La construction modulaire permet de procéder à des changements rapides par la division et la substitution de modules (Baldwin et Clark, 1997). De plus, réutiliser le même module dans plusieurs structures permet des économies d'échelle et des économies de substitution (Garud et Kumaraswamy, 1995).

La modularité de produit permet d'avoir des stratégies d'offre de configuration de produit comme la personnalisation de masse (*mass-customization*) et des stratégies d'assortiment extensif de produits aux fonctions proches (Hsuan Mikkola et Skjøtt-Larsen, 2004). Enfin, réduire l'interdépendance entre des modules peut diminuer la spécificité des ressources selon Baldwin et Clark (2000) et augmenter les occasions d'externalisation (Schilling et Steensma, 2001). Les constructions modulaires sont plus robustes aux changements dans l'environnement (Pil et Cohen, 2006).

Cependant, la modularité n'est pas un choix simple car son **principal problème** est qu'elle engendre des surcoûts dans l'assemblage ou le désassemblage des modules. On note également le problème de surcoût pour réaliser une conception modulaire par rapport à une conception intégrée (Langlois, 2002). Il serait donc opportun de mesurer le degré de modularité pour faciliter l'assemblage et le désassemblage des composants de chaque module constitutif de notre système et de proposer un moyen d'améliorer la fonction coût de couplage des modules pour éviter la formation de modules de trop grandes tailles.

Dans la section suivante, nous ferons un état de l'art sur les méthodes de conception robuste afin de choisir une méthode que nous allons améliorer pour résoudre le problème posé.

I.3 ANALYSE BIBLIOGRAPHIQUE SUR LES METHODES DE CONCEPTION ROBUSTE

La prise en compte des variations est essentielle dans la phase de développement d'un produit. En effet, toute conception est soumise à des variations qui peuvent être dues à des sources diverses, incluant les défauts de fabrication, les incertitudes sur les propriétés des matériaux, l'environnement, etc. L'ignorance de ces variations peut se traduire par des conceptions non robustes, onéreuses et défailtantes.

Selon Taguchi (1993), « au lieu d'éliminer ou de réduire les causes de la variabilité des performances d'un produit, il est préférable d'ajuster sa conception afin de le rendre insensible

aux causes des variations ». La conception robuste vise ainsi à optimiser les paramètres de conception d'un produit et de son procédé de fabrication afin de réduire la sensibilité de ses performances aux bruits internes et externes.

I.3.1 Les origines de la conception robuste

Le concept de la robustesse a été introduit pour la première fois par l'ingénieur japonais et statisticien Taguchi (1978). Il a proposé dans les années cinquante de transformer le contrôle de qualité en ligne en contrôle de qualité hors ligne. Cette étape est à l'origine du concept de qualité robuste en ingénierie. Il fallut cependant attendre le début des années quatre-vingt-dix pour trouver les premières contributions dans la littérature qui intègrent ce concept dans le cadre de la conception.

En effet, la conception robuste est au cœur de l'amélioration de la productivité en ingénierie. De nombreuses compagnies internationales ont réalisé des économies considérables en utilisant des méthodes de conception robuste, et ce dans différents secteurs d'activité : mécanique, automobile, télécommunications, électronique, informatique, etc (Phadke, 1989 ; Taguchi, 1993 ; Wu et Wu, 2000).

Les entreprises ont parallèlement investi lourdement dans l'approche « Six- Sigma » à la fin des années quatre-vingt et au début des années quatre-vingt – dix (Thornton, 2001). Cette approche permet de réduire les coûts en détectant les problèmes rencontrés durant les phases de conception et de fabrication et en s'affranchissant des causes de défaillance immédiatement. Ces efforts ont cependant eu un impact important sur le coût des produits et ainsi sur les revenus des entreprises. En effet, l'approche « Six- Sigma » a été utilisée au maximum de son potentiel et a montré ses limites. De nouvelles méthodes, telles que les méthodes de conception robustes, se sont donc avérées nécessaires pour améliorer la productivité des entreprises.

I.3.2 La conception robuste selon Taguchi

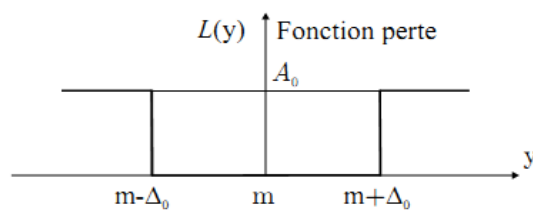
La philosophie de Taguchi repose sur deux concepts : la fonction perte et le ratio signal/bruit, (Taguchi, 1993 ; Wu et Wu, 2000).

I.3.2.1 La fonction perte

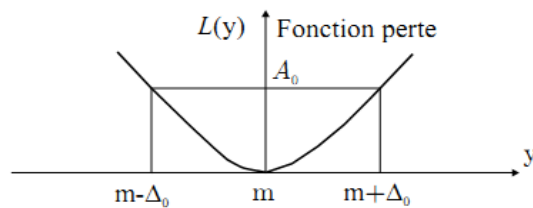
La fraction de produits en dehors des limites spécifiées est couramment utilisée comme une mesure de la qualité. Bien que ce soit une bonne mesure de la perte due au rebut, ce n'est pas un bon indicateur de la satisfaction du client. C'est la raison pour laquelle Taguchi a défini

la **fonction perte** comme une mesure de la perte de la qualité subie par un client, due à une mauvaise conception du produit acheté. Le respect des tolérances ne garantit pas nécessairement une bonne qualité du produit. Phadke (1989) présente ainsi une fonction perte quadratique à la place de la fonction perte double échelons. En effet, la qualité d'un produit diminue progressivement avec la déviation des variables par rapport à leur valeur nominale. Les allures de ces fonctions sont représentées par la figure 12 où y est la cible de la variable de conception, $m \pm \Delta_0$ est la plage de variations de y tolérée et A_0 est la perte due à la défaillance d'un produit. Ainsi, l'expression de la fonction perte est la suivante :

$$L = k(y - m)^2 \quad (35)$$



(a) Fonction double échelons



(b) Fonction perte quadratique

$$L(y) = k(y - m)^2$$

$$k = A_0 / \Delta_0$$

Figure 12 : Fonctions perte « double échelons » et « quadratique »

En outre, si la fonction y a pour moyenne m et pour variance σ^2 , la perte de qualité moyenne par produit est donnée par l'équation suivante (μ étant la cible nominale visée) :

$$Q = k [(\mu - m)^2 + \sigma^2] \quad (36)$$

I.3.2.2 Le ratio signal / bruit

La robustesse d'une conception varie avec les valeurs des variables de conception. Le ratio **signal/bruit** est une mesure de la sensibilité de la conception aux changements des conditions environnementales et peut être utilisé pour calculer les valeurs optimales des variables de conception.

D'après l'équation (35), la perte de qualité moyenne dépend de la déviation de la moyenne de la variable de conception y par rapport à sa cible et de sa variance. Rendre la conception robuste consiste ainsi à résoudre le problème d'optimisation visant à minimiser la variance des spécifications tout en maintenant leur moyenne sur leur cible. Selon Phadke (1989), il est plus

facile d'ajuster la moyenne sur la cible que de minimiser la variance. En conséquence, il est plus judicieux pour le concepteur de minimiser la variance des performances d'une conception dans un premier temps et d'ajuster ensuite la moyenne des variables de conception. Le problème d'optimisation peut ainsi se décomposer en deux étapes :

1. Maximiser le ratio signal/bruit η défini par :

$$\eta = 10 \log(\mu^2 / \sigma^2) \quad (37)$$

2. Ajuster la moyenne sur la cible en utilisant une ou deux variable(s) de conception n'ayant pas (ou ayant peu) d'influence(s) sur la variance des spécifications.

I.3.3 Formulation du problème de conception robuste local

Dans tout problème de conception robuste, trois ensembles sont dissociés : l'ensemble des variables de conception, l'ensemble des paramètres de conception environnementaux et l'ensemble des fonctions performances (Taguchi, 1993). Al- Widyan et Angeles (2005) introduisent par ailleurs les notions de conception globalement robuste et de conception localement robuste. Ici, nous nous focalisons sur la deuxième notion.

I.3.3.1 Les variables de conception

Les variables de conception (**VC**) d'un système mécanique sont généralement ses dimensions (longueurs des barres, orientations des axes des liaisons, etc). Les valeurs nominales de ces variables sont contrôlables et calculées par le concepteur afin d'optimiser les performances du mécanisme. Leur valeur réelle est cependant incertaine à cause des défauts de fabrication et de l'usure des pièces par exemple. Ces variables sont regroupées dans le vecteur \mathbf{x} de dimension l :

$$\mathbf{x} = [x_1 x_2 \dots x_l]^T \quad (38)$$

I.3.3.2 Les paramètres de conception environnementaux

Les paramètres de conception environnementaux (**PCE**) décrivent l'environnement du système. Ils ne peuvent pas être maîtrisés par le concepteur puisque leurs variations sont aléatoires. La température, la pression ambiante, le niveau d'humidité et l'utilisateur du bien conçu sont des exemples de paramètres de conception environnementaux. Ces paramètres sont regroupés dans le vecteur \mathbf{p} de dimension m :

$$\mathbf{p} = [p_1 p_2 \dots p_m]^T \quad (39)$$

I.3.3.3 Les fonctions performances

Les fonctions performances (**FP**) d'un système mécanique peuvent être multiples. Les n fonctions performances d'un système mécanique sont regroupées dans le vecteur f de dimension n et la $i^{\text{ème}}$ fonction performance est notée f_i :

$$f = [f_1 f_2 \dots f_n]^T \quad (40)$$

Les fonctions performances des systèmes mécaniques étudiés dépendent généralement des variables et des paramètres de conception :

$$f = f(x, p) \quad (41)$$

I.3.3.4 Exemples de formulation d'un problème de conception robuste

La formulation du problème de conception robuste d'un système de refroidissement d'une salle et celle d'un mécanisme articulé illustre la différence entre les ensembles définis précédemment.

I.3.3.4.1 Exemple 1 : système de refroidissement d'une salle

Pour la conception d'un système de refroidissement d'une salle, le réglage du thermostat est le signal et la température résultante de la salle est la fonction performance. Le système a ici une seule fonction performance : la température de la salle. La température extérieure, l'ouverture et la fermeture des fenêtres, le nombre de personnes présentes dans la salle sont des paramètres de conception du système, puisqu'ils ne peuvent pas être contrôlés par le concepteur. A l'inverse, le nombre de personnes déclarées, la taille de l'unité de conditionnement d'air, l'épaisseur et le type d'isolation sont des variables de conception puisque leur valeur nominale peut être contrôlée par le concepteur.

Afin de rendre le système robuste, le concepteur doit ainsi calculer les valeurs nominales des variables de conception afin de minimiser la variation de la température de la salle (la fonction performance) et de la maintenir au voisinage de la température fixée par le réglage du thermostat.

I.3.3.4.2 Exemple 2 : mécanisme articulé

Le système bielle- manivelle représenté par la figure **13** est composé d'une bielle de longueur l_r , d'une manivelle de longueur l_c et d'un piston. e est l'excentricité entre l'axe de rotation de la manivelle et la direction de la translation du piston.

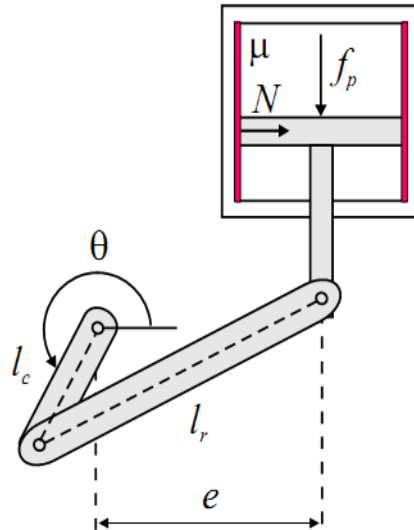


Figure 13 : Système bielle-manivelle

L'inégalité $l_r \geq l_c + e$ est une condition nécessaire au fonctionnement du mécanisme. f_p et μ sont respectivement la force exercée sur le piston et le coefficient de frottement entre le piston et le cylindre. Ils sont supposés incontrôlables par le concepteur. N_{moy} est la moyenne de la force radiale exercée par le cylindre sur le piston. Rendre la conception du mécanisme robuste consiste ici à calculer les valeurs nominales de l_r , l_c , et e afin de minimiser N_{moy} et sa sensibilité aux variations des variables et paramètres de conception. En définitive, $\mathbf{x} = [l_c \ l_r \ e]^T$, $\mathbf{p} = [f_p \ \mu]^T$ et $f = [N_{moy}]$.

I.3.4 Définition d'une méthodologie de conception robuste

I.3.4.1 Recensement des incertitudes

Nous disposons d'une boîte blanche dont les entrées $X = (X_1, \dots, X_n)$ regroupent les entrées du système U et les paramètres environnementaux V . Les sorties sont représentées par $Y = (Y_1, \dots, Y_k)$ et les contraintes par $T = (T_1, \dots, T_m)$. Le code de calcul f tel que $Y = f(X)$ est la fonction objectif. Les fonctions g_i telles que $T_i = g_i(X)$, $i = 1, \dots, m$ sont les fonctions contraintes. Ces fonctions contraintes permettent de définir l'espace de conception par les relations $T_i \geq 0$, $i = 1, \dots, m$.

I.3.4.1.1 Différents types d'incertitudes

Tout système physique, et donc notre boîte blanche, est sujet à différentes sources d'incertitudes qui sont en général classées en deux catégories (Beyer and Sendhoff, 2007) :

- *incertitudes objectives ou aléatoires* : incertitudes irréductibles dues à des phénomènes physiques non contrôlables comme la température, la pression mais aussi les conditions de fabrication de pièces ou les propriétés des matériaux ;
- *incertitudes de modèle ou épistémiques* : incertitudes dues à un manque de connaissance du concepteur vis-à-vis du comportement de son système. Ces incertitudes peuvent être en général réduites par un effort de modélisation, d'investissement dans la connaissance.

Le vecteur X défini précédemment est entaché d'incertitudes aléatoires. Les incertitudes épistémiques touchent quant à elles les fonctions f et g_i , $i = 1, \dots, m$.

Un classement plus précis des différentes variables incertaines en fonction de la possibilité ou non d'action sur ces variables dans le processus de conception a été proposé dans (Beyer and Sendhoff, 2007) :

- *incertitudes de type I* : incertitudes liées à l'environnement et aux conditions d'utilisation, ce sont donc des incertitudes aléatoires. Les variables que nous avons notées V précédemment présentent ce type d'incertitudes. Elles sont indépendantes de la conception du système physique. Elles peuvent être connues soit par des valeurs caractéristiques notées $v_j^{(c)}$ pour chaque variable V_j , soit par une densité de probabilité complète obtenue à partir de mesures ;
- *incertitudes de type II* : incertitudes de fabrication des pièces. Les variables géométriques, que nous avons notées U , ne sont connues du concepteur que par leur valeur nominale qu'il peut ajuster dans l'optique d'une conception robuste. Notons u une réalisation de U : à cause des imperfections potentielles du processus de fabrication, les variables U n'ont jamais la valeur nominale souhaitée u^* et une dispersion Δu apparaît autour de u^* ($u = u^* + \Delta u$) ;
- *incertitudes sur les fonctions du système* : incertitudes liées à l'évaluation des performances du système, ce sont des incertitudes épistémiques qui touchent ici la fonction f .

On peut parfois les regrouper avec les incertitudes de type I ;

- *incertitudes de faisabilité* : incertitudes sur les fonctions contraintes définissant l'espace de conception. Ce sont encore des incertitudes épistémiques mais qui touchent cette fois les fonctions g_i .

La prise en compte de cette incertitude est une condition indispensable pour un résultat optimal et robuste des études menées sur un système. Une fois que les sources d'incertitudes ont été identifiées, il convient de les modéliser et de les quantifier.

I.3.4.1.2 Modélisation des incertitudes

La prise en compte des incertitudes peut se faire de différentes manières :

- *approche au pire des cas (ou déterministe)* : chaque variable incertaine est connue par ses valeurs haute $u_i^{(+)}$ et basse $u_i^{(-)}$ pour les variables de décision $u_i, i = 1, \dots, d_1$ et une valeur caractéristique $u_j^{(c)}$ pour les variables $V_j, j = 1, \dots, d_2$;
- *approche probabiliste* : chaque variable incertaine (U et V) est définie par sa densité de probabilité $\tilde{f}_i(u_i), i = 1, \dots, d_1$ et $\tilde{f}_j(u_j), i = 1, \dots, d_2$;
- *approche possibiliste* : une mesure floue est associée à chaque variable incertaine (Dubois et al., 2008). Il peut s'agir d'une modélisation déterministe avec une loi sous-jacente inconnue mais dont on connaît une statistique (la moyenne par exemple).

La méthode déterministe est simple mais limitée, tandis que la méthode possibiliste est plus générale mais aussi plus complexe.

A chaque variable incertaine $X_i, i = 1, \dots, d$, est associée une densité de probabilité $\tilde{f}_i(x_i)$. Les variables du système n'étant pas entachées d'incertitudes, sont fixées à leur valeur nominale. Elles seront donc considérées comme des paramètres et non plus comme des variables.

Les incertitudes sur les fonctions du système et les incertitudes de faisabilité sont beaucoup plus difficiles à modéliser. On les retrouve souvent dans les codes numériques censés reproduire le comportement physique d'un système. Il existe donc un biais entre les simulations et la réalité. Cependant, l'accès aux résultats dans la réalité n'est pas toujours possible, ce qui rend l'évaluation de ce biais inaccessible en pratique. Avec l'amélioration des systèmes informatiques et des capacités de calculs, nous considérons que les modèles utilisés conduisent à des erreurs suffisamment faibles pour que leurs incertitudes soient négligées.

La modélisation des incertitudes de types I et II par des lois de probabilité conduit à une variabilité des sorties qui présentent donc également des lois de probabilité. L'obtention d'informations sur ces lois en sortie se fait par la propagation d'incertitudes.

I.3.4.2 Méthodologie de propagation des incertitudes

Partant du modèle physique connu (code de calcul), l'étude d'une variable d'intérêt Y à partir de données stochastiques (vecteur X) permet de déterminer une quantité d'intérêt (moments de la loi de Y par exemple) et donc de réaliser la propagation des incertitudes (cf. Figure 14) au travers du modèle. Le cheminement inverse consiste à considérer un grand

nombre d'évaluations de la quantité d'intérêt, afin de hiérarchiser les incertitudes. Cela revient à calculer les sensibilités, au sens probabiliste, c'est-à-dire l'influence des paramètres d'entrée sur la variabilité de la réponse.

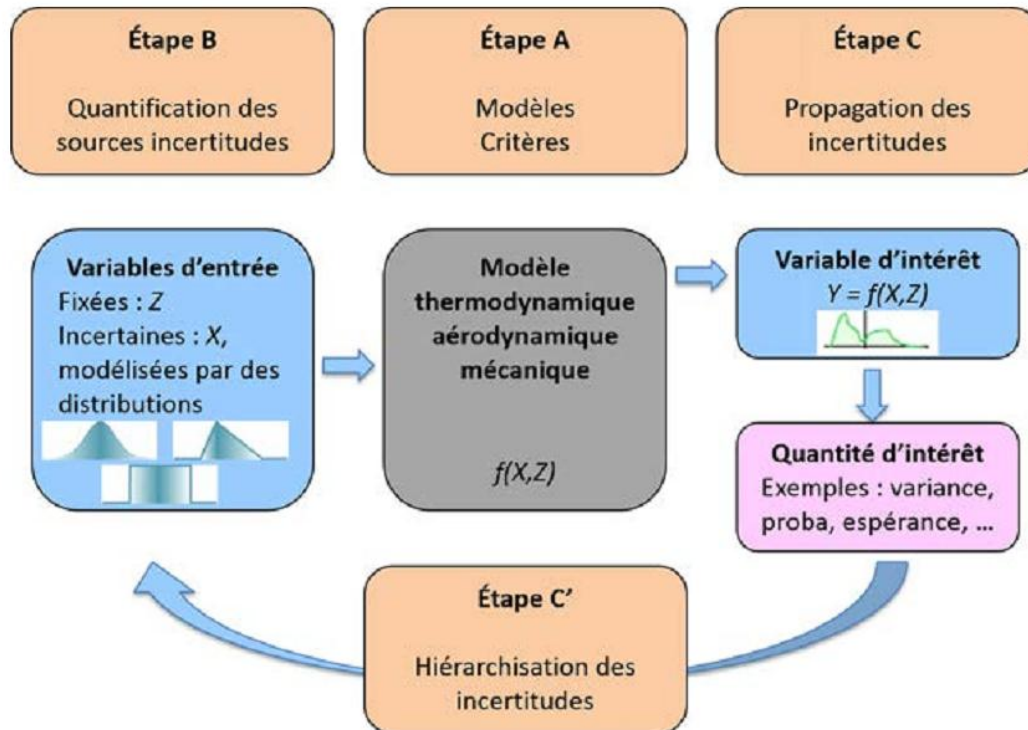


Figure 14: Schéma de la "méthodologie de propagation d'incertitudes" (de Rocquigny et *al.*, 2008)

Le principal problème de la modélisation probabiliste est que les résultats obtenus en sortie dépendent du choix des différentes lois sur les entrées. Ce phénomène a notamment été étudié par (Lehman et *al.*, 2004). Il faut donc suffisamment d'informations sur une incertitude pour choisir la loi qui lui sera attribuée.

Les quantités d'intérêt obtenues en sortie peuvent être obtenues de différentes façons : Monte-Carlo sur le code de calculs, Monte-Carlo sur des modèles locaux, la méthode FOSM (First Order Second Moment) et la méthode URQ (Univariate Reduced Quadrature). Ces quatre méthodes font l'objet des sections suivantes.

I.3.4.2.1 Méthode de Monte-Carlo sur le code de calculs

La méthode la plus évidente est la méthode de Monte-Carlo. Elle consiste à effectuer des tirages de Monte-Carlo à chaque appel du code. Ces tirages sont effectués suivant la loi de chaque variable entachée d'incertitudes et sont répartis autour du point évalué afin de représenter l'incertitude de la variable. La robustesse de la sortie, à travers la (ou les) quantité(s)

d'intérêt choisie(s), est donc calculée en chaque point évalué. Les quantités d'intérêt, moments de loi de la sortie Y , sont estimées localement par les moments empiriques de Y sur les N tirages de Monte-Carlo. La moyenne empirique est donnée par la relation (42) et l'écart-type empirique par la relation (43).

$$m_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (42)$$

$$\sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - m_y)^2} \quad (43)$$

Du point de vue de l'écart-type, on propage ainsi l'incertitude de la variable X_i sur la sortie Y selon le schéma de la Figure 15.

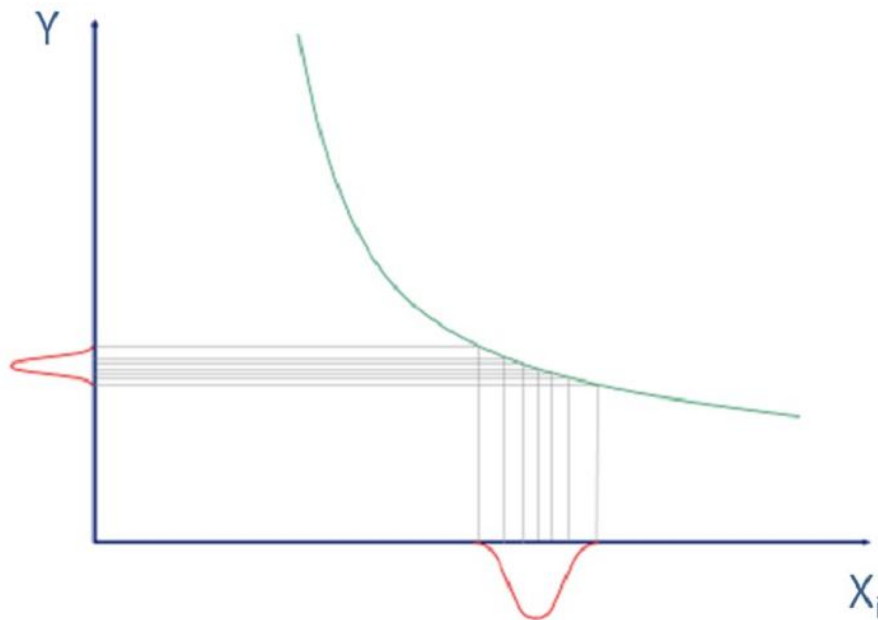


Figure 15 : Propagation des incertitudes par simulations de Monte-Carlo sur le code de calculs

Chaque simulation selon la loi des entrées fournira une valeur de la sortie. Toutes ces valeurs permettent d'estimer localement les moments de la loi de cette sortie.

Il est clair que plus le nombre de tirages est important, plus l'estimation de la quantité d'intérêt est précise. Cette méthode peut donc rapidement devenir coûteuse, surtout si le code de calculs lui-même est coûteux.

I.3.4.2.2 Méthode de Monte-Carlo sur un modèle local

Cette méthode permet d'estimer les quantités d'intérêt de manière moins coûteuse que les simulations de Monte-Carlo sur le code de calcul. En effet, les simulations de Monte-Carlo effectuées autour du point en cours ne sont plus évaluées via le code de calculs, mais par une approximation locale de la fonction étudiée. Une méthode bien connue consiste à considérer une approximation locale linéaire.

L'approximation linéaire de f en un point a consiste à négliger le reste dans le développement limité d'ordre 1 de f au voisinage de a . On écrit alors, pour x dans un voisinage de a :

$$f(x) = f(a) + Df(a)(x - a) = f(a) + \sum_{i=1}^d \frac{\partial f}{\partial x_i}(a)(x_i - a_i) \quad (44)$$

où $Df(a)$ est la différentielle de f en a , qui peut être vu comme le coefficient directeur de l'approximation linéaire. Ceci revient à confondre localement f avec sa tangente. Comme on ne connaît pas forcément l'expression de f , on va se servir de la dérivation numérique pour estimer $Df(a)$.

On choisit $h \in \mathbb{R}^d$ et $(u, v) \in (\mathbb{R}^d)^2$ tels que pour tout $i \in [1, d]$, $|u_i - v_i| = h_i$ et $|u_i - a_i| =$

$|v_i - a_i| = \frac{h_i}{2}$. Sur chaque dimension, h_i doit être faible de telle sorte que u et v soient proches.

Ainsi, pour $i \in [1, d]$, on a :

$$\frac{\partial f}{\partial x_i}(a) = \frac{f(a_1, \dots, a_{i-1}, u_i, a_{i+1}, \dots, a_d) - f(a_1, \dots, a_{i-1}, v_i, a_{i+1}, \dots, a_d)}{h_i} \quad (45)$$

Sur chaque dimension, ceci revient à remplacer la tangente par la corde. L'estimation de $Df(a)$ nécessite donc d'évaluer $2d$ points.

Les moments de Y sont calculés comme précédemment sauf que les y_i ne sont pas des évaluations du code mais des valeurs approchées par le modèle linéaire. Le schéma de la Figure 15 devient donc celui de la Figure 16.

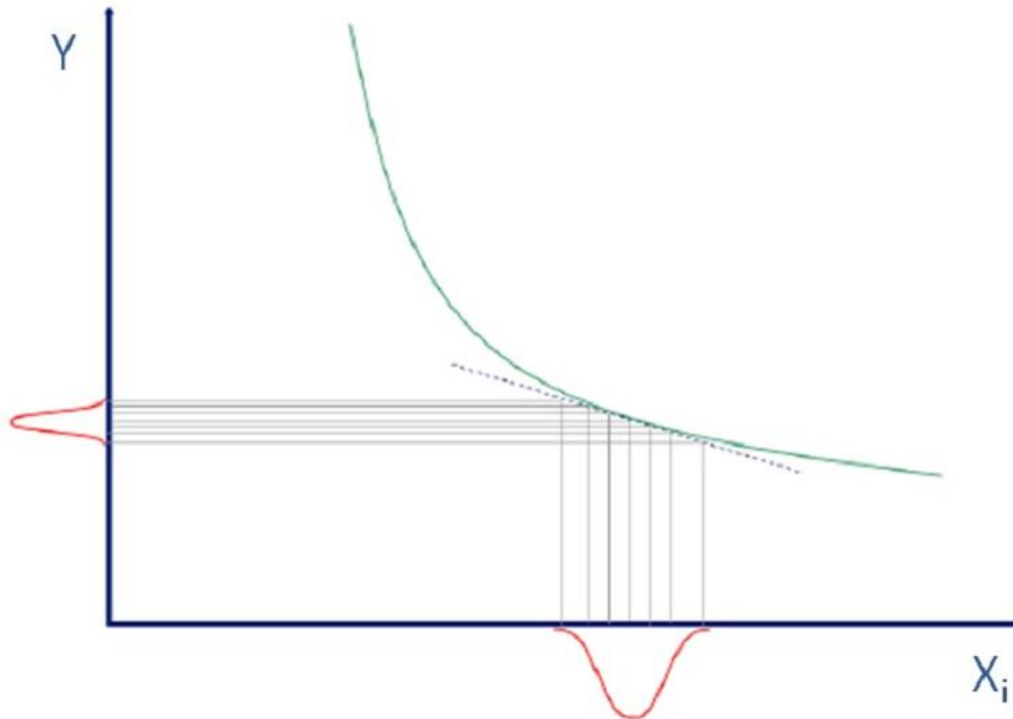


Figure 16 : Propagation des incertitudes par simulations de Monte-Carlo sur un modèle ajusté localement

Le principal inconvénient de cette méthode est que la qualité de l'estimation des moments n'est pas bonne, si la vraie fonction est discontinue ou présente des non-linéarités importantes.

I.3.4.2.3 Méthode FOSM

La méthode FOSM (First Order Second Moment) est une méthode probabiliste permettant de déterminer les moments stochastiques d'une fonction à partir de variables d'entrée aléatoires. Ainsi, la moyenne de Y sera approchée par la relation (46) et l'écart-type, par la relation (47).

$$m_y \approx f(\mu) \quad (46)$$

$$\sigma_y \approx \sqrt{\sum_{i=1}^d \sum_{j=1}^d \frac{\partial f(\mu)}{\partial x_i} \frac{\partial f(\mu)}{\partial x_j} Cov(X_i, X_j)} \quad (47)$$

où d est la dimension du vecteur d'entrée X et $\mu = (\mu_1, \dots, \mu_d)$, le vecteur moyenne de X . Ces approximations sont obtenues en remplaçant la fonction f par son développement de Taylor à l'ordre 1.

Notons que si les variables X_i sont indépendantes, c'est-à-dire $\text{Cov}(X_i, X_j) = 0, i \neq j$, alors :

$$\sigma_y \approx \sqrt{\sum_{i=1}^d \left(\frac{\partial f(\mu)}{\partial x_i} \right)^2 \sigma_{x_i}^2} \quad (48)$$

Dans le cas d'une seule variable d'entrée ($d = 1$), on a alors $\sigma_y = \frac{\partial f(\mu)}{\partial x_i} \sigma_x$.

La propagation des incertitudes ressemble à celle de la méthode de Monte-Carlo sur modèle local, puisqu'il faut ici aussi estimer les dérivées partielles de f , sauf que nous n'avons plus besoin de simulations de la loi de X pour estimer les moments de la loi de Y (cf. Figure 17).

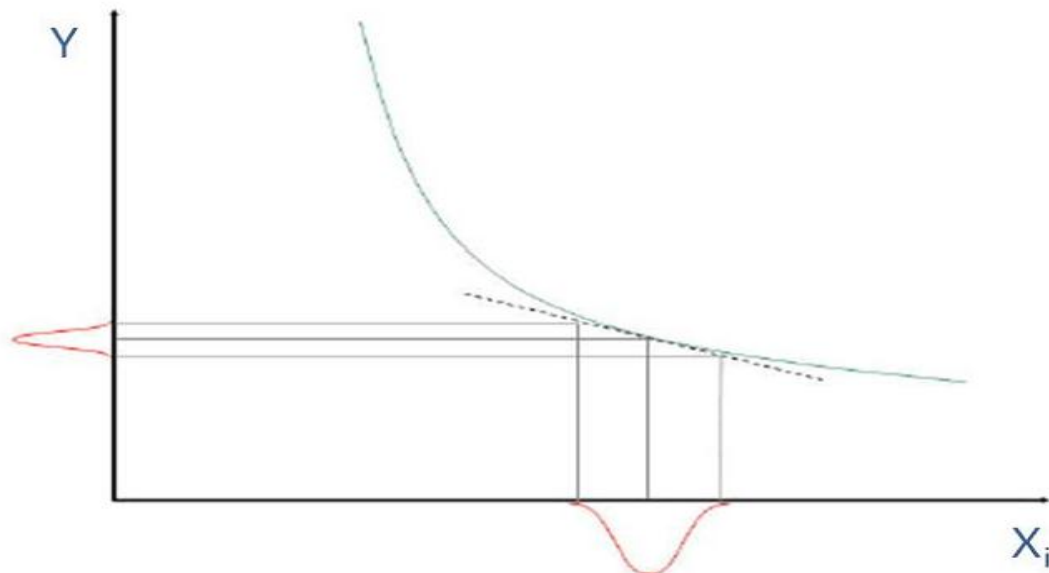


Figure 17 : Propagation des incertitudes par la méthode FOSM

Cette méthode est plus rapide que la méthode de Monte-Carlo sur code de calculs, mais est comparable à la méthode de Monte-Carlo sur modèle local. Comme cette dernière, la méthode FOSM dépend de la forme de la fonction étudiée et peut être en difficulté, dans des cas de discontinuités ou de fortes non linéarités. Le lecteur intéressé par cette méthode pourra se référer à (Haldar and Mahadevan, 2000).

I.3.4.2.4 Méthode URQ

Cette méthode est basée sur la quadrature de Gauss qui permet d'approcher la valeur numérique d'une intégrale, à partir de points pris sur le domaine d'intégration. Ces points sont appelés des nœuds. Dans la méthode URQ (Univariate Reduced Quadrature), $2d + 1$ nœuds sont considérés. Le nombre de variables incertaines d dans le système comprend les d_1 entrées

et les d_2 paramètres environnementaux, s'il y en a. Elles sont considérées comme indépendantes.

Soit $X = (X_1, \dots, X_d)$ le vecteur des variables aléatoires du système. On note μ_X la moyenne de X et $\sigma_X = (\sigma_{X_1}, \dots, \sigma_{X_d})$ son écart-type. Les $2d + 1$ points $x = (x^0, x^+, x^-)$ qui seront évalués sont définis de la manière suivante :

$$\begin{aligned} x^0 &= \mu_X \\ x_i^+ &= \mu_X + h_i^+ \sigma_{X_i} e_i, \\ x_i^- &= \mu_X + h_i^- \sigma_{X_i} e_i, \end{aligned} \quad (49)$$

où e_i est le $i^{\text{ème}}$ vecteur de la matrice identité de taille d , h_i^+ et h_i^- sont des paramètres d'échelle dépendant des moments centrés réduits d'ordres 3 et 4 (respectivement le coefficient d'asymétrie et le kurtosis notés γ_X et Γ_X de la loi de X). Ces $2d + 1$ points permettent, par la quadrature de Gauss, d'estimer l'espérance et la variance de la sortie $Y = f(X)$ en utilisant les formules suivantes :

$$\mu_Y = \omega_0 f(\mu_X) + \sum_{i=1}^d \omega_i \left(\frac{f(x_i^+)}{h_i^+} - \frac{f(x_i^-)}{h_i^-} \right) \quad (50)$$

$$\sigma_Y^2 = \sum_{i=1}^d \left(w_i^+ \left(\frac{f(x_i^+) - f(\mu_X)}{h_i^+} \right)^2 + w_i^- \left(\frac{f(x_i^-) - f(\mu_X)}{h_i^-} \right)^2 + w_i^\pm \left(\frac{(f(x_i^+) - f(\mu_X))(f(x_i^-) - f(\mu_X))}{h_i^+ h_i^-} \right) \right) \quad (51)$$

Par une parallèle entre la quadrature de Gauss et le développement de Taylor de f à l'ordre 4, on obtient les valeurs de h_i^+ et h_i^- :

$$h_i^\pm = \frac{\gamma X_i}{2} \pm \sqrt{\Gamma_{X_i} - \frac{3\gamma_{X_i}^2}{4}} \quad (52)$$

Les différents poids présents dans les relations (50) et (51) sont alors :

$$\begin{aligned}
w_0 &= 1 + \sum_{i=1}^d \frac{1}{h_i^+ h_i^-}, \\
w_i &= \frac{1}{h_i^+ - h_i^-}, \\
w_i^+ &= \frac{(h_i^+)^2 - h_i^+ h_i^- - 1}{(h_i^+ - h_i^-)^2}, \\
w_i^- &= \frac{(h_i^-)^2 - h_i^+ h_i^- - 1}{(h_i^+ - h_i^-)^2}, \\
w_i^\pm &= \frac{2}{(h_i^+ - h_i^-)^2}.
\end{aligned} \tag{53}$$

Comparée à la méthode de Monte-Carlo, la méthode URQ permet de réaliser la propagation d'incertitudes plus rapidement et avec moins d'appels au code de calculs. De plus, la précision de l'estimation n'est que légèrement moins bonne que celle fournie par les simulations de Monte- Carlo. Le lecteur intéressé par cette méthode de propagation pourra se référer à (Padulo and Guenov, 2011).

Nous avons défini plusieurs méthodes de propagation des incertitudes à travers un code de calculs. Définissons maintenant le problème de conception robuste, afin de déterminer ensuite les méthodes de résolution que nous devons utiliser pour résoudre un tel problème.

I.3.4.3 Méthodes de conception robuste pour la résolution des problèmes d'optimisation

Le principe d'une optimisation multi-objectifs est différent du principe d'une approche mono-objectif. Le but principal d'une optimisation mono-objectif, est de trouver la solution optimale globale qui donne la meilleure valeur (plus petite ou plus grande) de la fonction mono-objectif. Dans un problème d'optimisation multi-objectifs, on dispose de $k \geq 2$ fonctions objectifs, chaque fonction objective pouvant avoir une solution optimale différente. Le but d'un problème multi-objectifs est de trouver de « bons compromis », plutôt qu'une seule solution. Certaines méthodes permettent de trouver une solution réalisant un compromis entre les différents objectifs. D'autres permettent d'en trouver plusieurs et de construire le front de Pareto. Dans toute la suite, nous noterons f_1 et f_2 les fonctions objectifs correspondant à chacune des deux optimisations que nous supposons être des minimisations. La plupart des méthodes décrites ici le sont également dans (Collette and Siarry, 2002).

I.3.4.3.1 Méthodes trouvant un compromis entre les objectifs

Parmi les méthodes effectuant un compromis entre les objectifs de l'optimisation, nous exposerons rapidement quatre méthodes :

- la méthode hiérarchique,
- la méthode par compromis,
- la fonction de distance,
- la méthode du critère global.

Ces méthodes sont étudiées par exemple dans (de Oliveira and Saramago, 2010).

➤ Méthode hiérarchique

La méthode hiérarchique consiste à organiser les objectifs par ordre d'importance. Chaque fonction objectif est minimisée individuellement, sous une contrainte qui empêche le minimum de la nouvelle fonction d'être supérieur à une fraction choisie du minimum de la fonction objectif précédente.

Dans le cas d'une optimisation bi-objectifs, 1 est le rang de l'objectif le plus important, 2 celui du moins important. Deux optimisations simples à un objectif sont effectuées successivement.

La première permet de trouver le point $x^{(1)} = (x_1^{(1)}, \dots, x_d^{(1)})$ tel que $f_1(x_1^{(1)}) = \min_x f_1(x)$.

La seconde optimisation consiste à trouver $x^{(2)} = (x_1^{(2)}, \dots, x_d^{(2)})$ tel que :

$$f_2(x^{(2)}) = \min_x f_2(x),$$

$$\text{sous la contrainte } f_2(x^{(2)}) \leq \left(1 + \frac{\varepsilon}{100}\right) f_1(x^{(1)}),$$

où ε est un réel positif inférieur à 100. Le choix de ε se fait par l'utilisateur. Plus sa valeur est petite, plus la condition est difficile à satisfaire. Le lecteur intéressé par cette méthode pourra se référer à (Osyczka, 1981).

➤ Méthode par compromis

Cette méthode consiste à convertir le problème d'optimisation multi-objectifs, en un problème d'optimisation simple à un seul objectif. Pour cela, un des deux objectifs est choisi comme étant l'objectif principal, tandis que l'autre devient une contrainte du problème. On cherche alors à résoudre :

$$\min_x f_1(x), \tag{54}$$

$$\text{sous la contrainte } f_2(x) \leq \omega \min_x f_2(x) \quad (55)$$

La quantité $w > 1$ est un poids permettant d'assouplir la contrainte qui serait trop forte avec $w = 1$. Le minimum $\min_x f_2(x)$ est le résultat de l'optimisation simple de f_2 . Les fonctions objectives sont interchangeable, le choix se fait par l'utilisateur, tout comme pour la valeur de w . Le lecteur intéressé par cette méthode pourra se référer à (Nakayama and Sawaragi, 1984).

➤ **Fonction de distance**

Cette méthode est utilisée lorsque les optimisations consistent à atteindre des valeurs cibles a_1 et a_2 . La méthode consiste à minimiser l'écart entre les observations de chaque fonction et leurs cibles respectives. Pour cela, on utilise une fonction de distance :

$$E_p = \left(|f_1(x) - a_1|^p + |f_2(x) - a_2|^p \right)^{1/p}, \quad 1 \leq p \leq \infty \quad (56)$$

où p est l'ordre de la méthode. Lorsque $p = 1$, on parle de programmation par buts. Pour $p = 2$, il s'agit de la norme euclidienne.

➤ **Méthode du critère global**

Cette méthode consiste à trouver la combinaison des entrées assurant le minimum d'un critère global. Ce critère consiste à établir la proximité de chaque fonction objectif f_i , à un minimum a_i^* , selon un écart relatif. Le but est de trouver :

$$\min_x \left[\left(\frac{a_1^* - f_1(x)}{a_1^*} \right)^2 + \left(\frac{a_2^* - f_2(x)}{a_2^*} \right)^2 \right] \quad (57)$$

Les minima a_i^* sont les valeurs optimales de chaque fonction objectif f_i du problème d'optimisation simple à un objectif. Le lecteur intéressé par cette méthode pourra se référer à (Rao, 2009).

Ces quatre méthodes permettent d'obtenir une unique solution au problème d'optimisation multi-objectifs. Les méthodes suivantes consistent à obtenir plusieurs points solutions afin d'établir un front de Pareto.

I.3.4.3.2 Méthodes établissant le front de Pareto

Le front de Pareto est la limite séparant la région des points faisables (satisfaisant toutes les contraintes) des points non-faisables. Dans un cas bi-objectifs, ce front est une ligne continue que l'on représente par rapport aux deux objectifs. Le but des méthodes d'optimisation générant le front de Pareto, est de trouver un certain nombre de points sur la ligne de Pareto, en donnant

différents poids aux différents objectifs. Par exemple, pour obtenir trois points dans un cas bi-objectifs, on pourra considérer les trois jeux de poids $(1,0)$; $(0.5, 0.5)$; $(0,1)$. Six méthodes de ce type sont exposées.

➤ **Méthode d'intersection de limites normales**

Cette méthode, appelée *normal-boundary intersection method* en anglais, utilise une paramétrisation géométrique intuitive, afin d'établir une répartition équilibrée de points sur le front de Pareto. Ceci permet d'avoir une représentation précise de toute la ligne. Les optima simples sous contraintes de chaque fonction objectif sont déterminés f_1^* et f_2^* . Ces deux solutions correspondent respectivement aux poids $(1,0)$ et $(0,1)$. Elles constituent les extrémités du front de Pareto. La méthode consiste ensuite à trouver les solutions Pareto-optimales, sur des lignes perpendiculaires au segment $[f_1^*, f_2^*]$. Les points d'intersections des perpendiculaires à $[f_1^*, f_2^*]$ sont équi-répartis sur le segment. La méthode est représentée à la Figure 18. Les points bleus ainsi générés sont bien répartis sur le front de Pareto. La zone délimitée par une courbe noire est la région acceptable. Cette méthode a été étudiée dans (Das and Dennis, 1998) puis dans (Shukla, 2007).

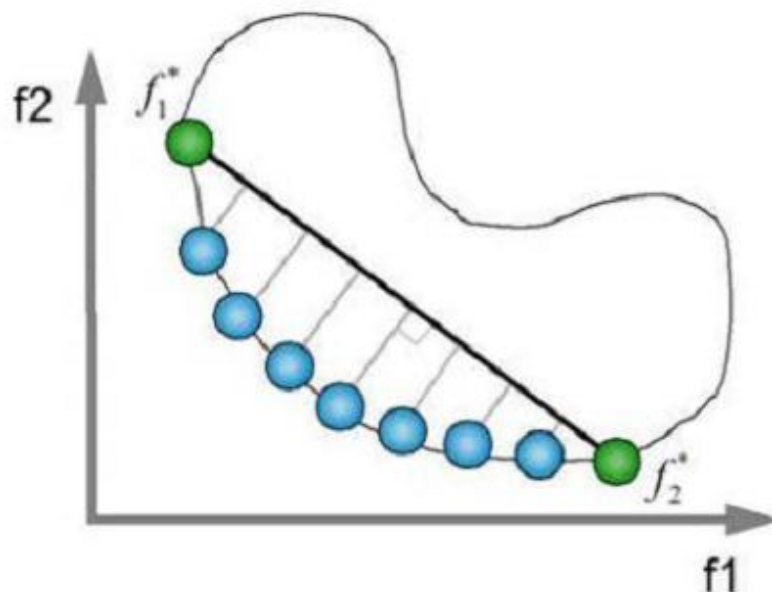


Figure 18 : Illustration de la méthode d'intersection de limites normales

➤ **Méthode à objectifs pondérés**

La méthode consiste à transformer le problème d'optimisation multi-objectifs en un problème d'optimisation simple à un seul objectif.

$$\min_x \omega_1 f_1(x) + \omega_2 f_2(x) ; \omega_i \geq 0 \text{ et } \omega_1 + \omega_2 = 1 \quad (58)$$

Cette optimisation simple est effectuée pour plusieurs choix de poids, le nombre est choisi par l'utilisateur. A chaque jeu de poids correspond une solution de Pareto, comme le montre la Figure 19. Ici, cinq jeux de poids ont été considérés, ce qui nous donne cinq points du front de Pareto. Le lecteur intéressé par cette méthode pourra se référer aux travaux de (Kim and Weck, 2005 ; Marler and Arora, 2010).

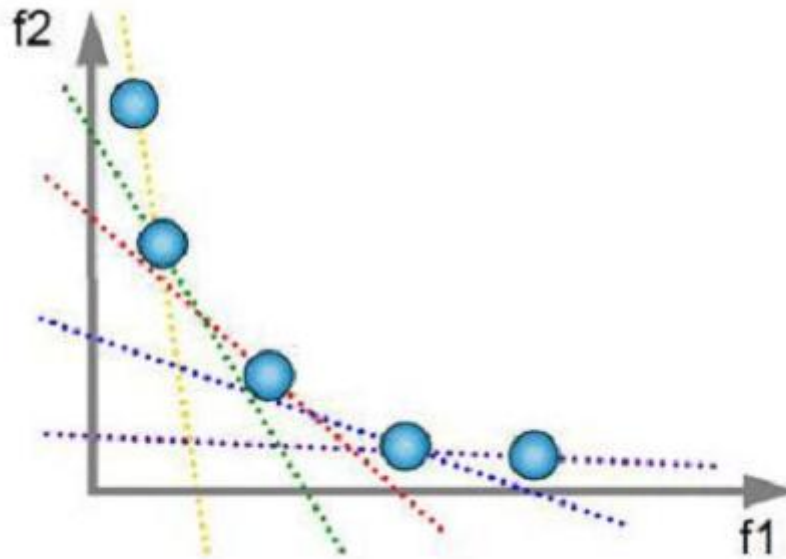


Figure 19 : Illustration de la méthode à objectifs pondérés

➤ **Méthode de Tchebycheff pondérée**

On considère les valeurs minimales de f_1 et f_2 obtenues par optimisation simple, notées f_1^* et f_2^* . La méthode de Tchebycheff pondérée consiste à résoudre le problème :

$$\min_x \max \left(w_1 |f_1(x) - f_1^*|, w_2 |f_2(x) - f_2^*| \right) ; w_1 + w_2 = 1 \quad (59)$$

Les poids prennent différentes valeurs, afin de trouver différents points sur le front de Pareto. Cette méthode a été étudiée dans les travaux de (Steuer and Choo, 1983 ; Miettinen, 2001).

➤ **Méthode de l'optimum min-max**

Proche de la méthode précédente, la méthode de l'optimum min-max consiste à résoudre le problème :

$$\min_x \max \left(w_1 \frac{f_1(x) - f_1^*}{f_1^*}, w_2 \frac{f_2(x) - f_2^*}{f_2^*} \right), \quad (60)$$

où f_1^* et f_2^* ont déterminés comme précédemment et $w_1 + w_2 = 1$. Différents jeux de poids fournissent différents points sur le front de Pareto. Le lecteur intéressé pourra se référer à (Zhang and Gao, 2006).

➤ Méthode génétique de tri basée sur la non-dominance

Cette méthode, appelée *non-dominated sorting genetic method*, s'appuie sur les méthodes génétiques. Dans un premier temps, définissons la dominance dans le cas de deux minimisations :

Définition (Dominance). Une solution A domine une solution B si et seulement si :

$$\forall i \in \{1, 2\}, f_i(A) \leq f_i(B) \text{ et } \exists j \in \{1, 2\}, f_j(A) < f_j(B) .$$

Si la solution A domine la solution B , on dit que B est dominée par A ou bien A est non dominée par B ou entre les deux solutions, A est la solution non dominée.

Si la solution A domine la solution B , on dit que B est dominée par A ou bien A est non dominée par B ou entre les deux solutions, A est la solution non dominée.

L'algorithme associé à la méthode étudiée ici, noté NSGA-II et proposé par Deb et al. (2002), a les caractéristiques suivantes :

- il utilise une approche élitiste pour sauvegarder les meilleures solutions trouvées lors des itérations précédentes,
- il utilise une procédure rapide de tri basée sur la non-dominance,
- il utilise un opérateur de comparaison basé sur une distance de surpeuplement,
- il n'a aucun paramètre à régler.

Notons que la distance de surpeuplement est une mesure de proximité d'un point avec ses voisins. Une grande distance moyenne de surpeuplement sur un échantillon signifie qu'il y a une grande diversité dans la population étudiée. A chaque étape de l'algorithme, on dispose d'une population (R_t) de $2N$ individus. Elle est séparée en une population de parents (P_t) de taille N et une population d'enfants (Q_t) de taille N . La population de taille $2N$ est ensuite triée selon un critère de non-dominance pour identifier différents fronts F_1, F_2, \dots . Le premier front est l'ensemble des individus non-dominés, le deuxième comprend les individus dominés uniquement par ceux du premier front, et ainsi de suite. Les meilleurs individus vont donc se retrouver dans le ou les premiers fronts. Une nouvelle population parente (P_{t+1}) est formée en considérant les meilleurs individus appartenant aux premiers fronts. Tous les individus de chaque front successif sont sélectionnés tant que le nombre d'individus ne dépasse pas N . Si le nombre d'individus présents dans (P_{t+1}) est inférieur à N , on va sélectionner les individus du front suivant, qui n'a pas été inclus dans (P_{t+1}), et appliquer une procédure de surpeuplement pour atteindre les N individus. Une fois que les individus appartenant à la population (P_{t+1})

sont identifiés, une nouvelle population enfant (Q_{t+1}) est créée par sélection, croisement et mutation.

La sélection est basée sur un opérateur de comparaison. Cet opérateur permet d'ordonner deux solutions selon leur rang et la valeur de la distance de surpeuplement. Entre deux solutions de rangs différents, on préfère la solution avec le plus petit rang (ou le plus petit front). Pour deux solutions qui appartiennent au même front, on préfère la solution qui est localisée dans la région où la densité de solutions est moindre, soit l'individu possédant la plus grande valeur de distance de surpeuplement.

Le fonctionnement de l'algorithme est représenté à la Figure 20.

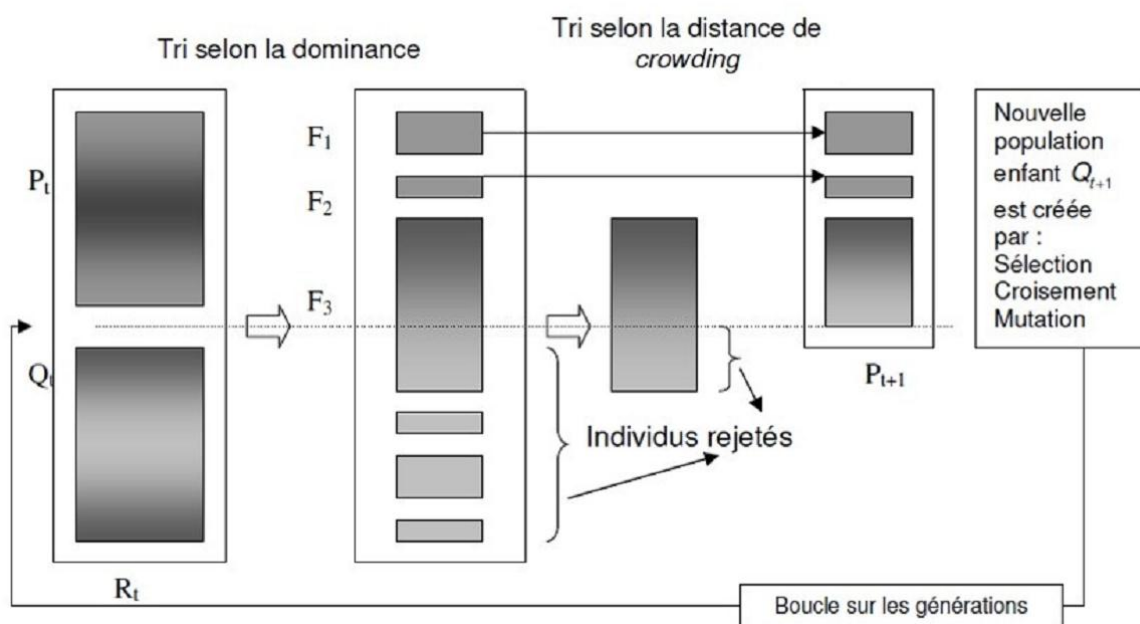


Figure 20 : Illustration de la méthode génétique de tri basée sur la non-dominance

➤ Algorithme de clustering

Le « clustering » fait référence au classement d'objets dans différents groupes, plus précisément au partitionnement des données en paquets homogènes qu'on appelle « cluster ».

Lorsque le nombre d'individus de l'archive externe est grand, les performances de l'algorithme d'optimisation choisie peuvent se dégrader significativement. En effet, le nombre d'individus influe sur le calcul de la valeur d'adaptation. Celui-ci devient moins fiable, et peut tromper la recherche en la focalisant, par exemple, sur des zones déjà explorées. Pour pallier à ce défaut, une solution consiste à utiliser des techniques de *clustering* (réduisant ainsi l'ensemble de Pareto sans supprimer ses caractéristiques). Ces techniques ont été étudiées intensivement dans le contexte des analyses de cluster et ont été appliquées avec succès pour

déterminer des partitions d'une collection relativement hétérogène d'éléments. La technique de clustering utilisée par SPEA (*Strength Pareto Evolutionary Algorithm*) est assez intuitive. Au début de la procédure, chaque individu constitue son propre groupe, puis on fusionne deux à deux les groupes les plus proches en terme de distance. Cette étape est itérée jusqu'à l'obtention du nombre désiré de groupes. Une fois les groupes identifiés, il ne reste plus qu'à choisir un représentant par groupe. Ce représentant peut être déterminé de plusieurs façons, par exemple en prenant le barycentre du groupe. C'est ce représentant qui sera gardé, les autres éléments étant tout simplement supprimés. Les étapes de la méthode de clustering sont illustrées à la figure 21.

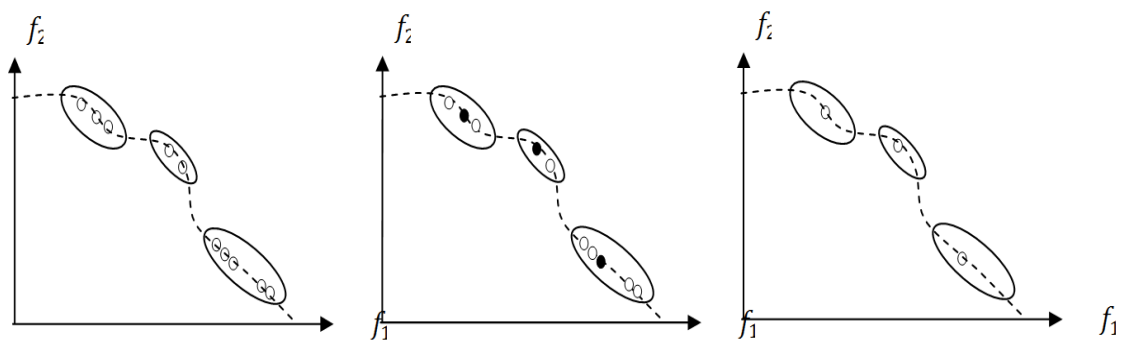


Figure 21 : Illustration du Clustering en dimension 2.

La synthèse des méthodes de conception robuste pour la résolution d'un problème d'optimisation est donnée par la figure 22.

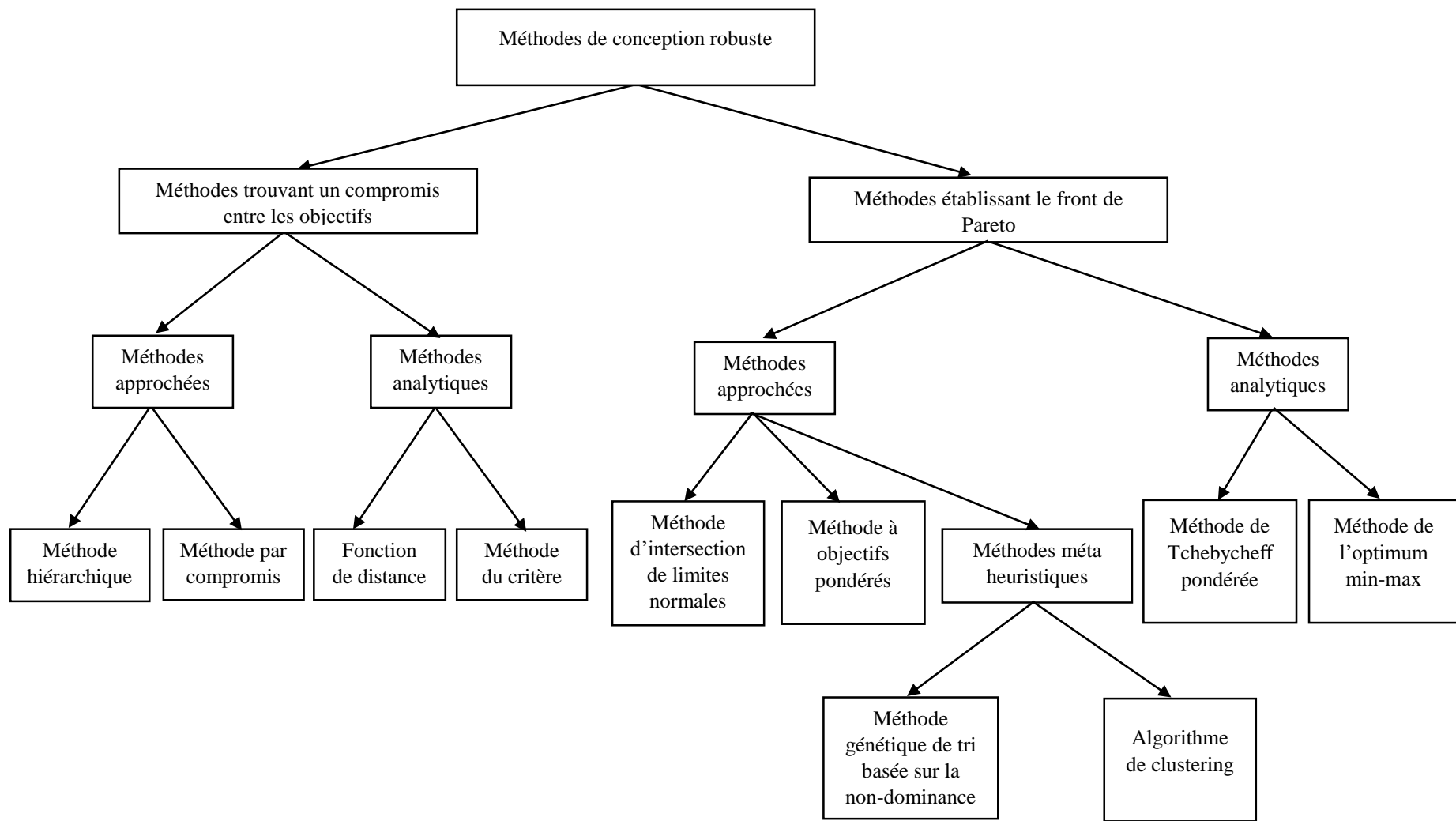


Figure 22 : Synthèse des méthodes de conception robuste pour la résolution d'un problème d'optimisation

I.3.4.4 Etude comparative des méthodes établissant le front de Pareto et des méthodes recherchant un compromis entre les objectifs

Dans la synthèse des méthodes de conception robuste de résolution des problèmes d'optimisation, nous avons d'une part les méthodes établissant un compromis entre les objectifs et les méthodes établissant le front de Pareto.

Les méthodes recherchant un compromis entre les objectifs sont les méthodes d'optimisation multi-objectif. Les problèmes multi-objectifs ont la particularité d'être beaucoup plus difficiles à traiter que leur équivalent mono-objectif. La difficulté réside dans l'absence d'une relation d'ordre total entre les solutions. Une solution peut être meilleure qu'une autre sur certains objectifs et moins bonne sur les autres. Donc il n'existe généralement pas une solution unique qui procure simultanément la solution optimale pour l'ensemble des objectifs. Voilà pourquoi le concept de solution optimale devient moins pertinent en optimisation multi-objectif. Les méthodes recherchant un compromis entre les objectifs, présentent donc une limite. Dans ce cas, la solution optimale n'est plus une solution unique mais, un ensemble de solutions compromis entre les différents objectifs à optimiser. Il est vital pour identifier ces meilleurs compromis, de définir une relation d'ordre entre ces éléments. La plus célèbre et la plus utilisée, est la **relation de dominance** au sens de **Pareto**. L'ensemble des meilleurs compromis est appelé le **front de Pareto**, la surface de compromis ou l'ensemble des solutions efficaces. Cet ensemble de solutions constitue un équilibre, dans le sens qu'aucune amélioration ne peut être faite sur un objectif sans dégradation d'au moins un autre objectif. La solution de Pareto consiste à obtenir le front de Pareto.

Parmi les méthodes établissant le front de Pareto, les méthodes méta-heuristiques sont les plus répandues dans la résolution des problèmes de partitionnement (qui nous intéresse dans le cadre de notre thèse). Il se trouve que la taille des données influe dans la recherche de la solution optimale, lorsqu'on fait recours aux algorithmes génétiques. Cette méthode devient donc moins fiable. Nous optons alors pour la méthode des algorithmes de clustering (k-means ou k-clusters ou k-modules) pour la résolution des problèmes de partitionnement. En effet, avec cette méthode, il est plus facile d'identifier des groupes de données inconnues à partir d'un ensemble de données complexes. Les résultats sont présentés de manière rapide.

I.4 PROBLEMATIQUE DE LA THESE

Dans ce qui suit, nous donnons un aperçu sur le nombre et l'importance des travaux par sous thèmes de recherche :

- *L'élaboration de l'architecture du produit, relative à l'agencement hiérarchisé ou non d'éléments* : (Pimmler et Eppinger, 1994 ; Ulrich, 1995 ; Erixon, 1996 ; Huang, 2000 ; Stone et al., 2000 ; Ulrich et Eppinger, 2000 ; Dahmus et al., 2001 ; Whitfield et al., 2002 ; Yu et al., 2003 ; Larses et Blackenfelt, 2003 ; Holttä et Salonen, 2003 ; Sharman et Yassine, 2004 ; Hölttä, 2005 ; Jiao et al., 2007). Ces travaux nombreux traitent de la modélisation et de l'identification de l'architecture du produit. Quelques-uns s'ouvrent à la modélisation du projet de conception pour lier l'architecture du produit aux autres caractéristiques du projet (Oosterman, 2001).
- *La définition des interfaces et des flux d'échanges dans le produit* : (Blackenfelt et Sellgren, 2000 ; Van et al., 2001 ; Chen et Liu, 2005). Ces travaux s'intéressent aux outils et aux méthodes de la conception détaillée du produit, ainsi qu'aux échanges entre composants, permettant ainsi de faire le lien avec la vue dynamique du produit.
- *La propagation d'une évolution dans une architecture* : (Clarkson et al., 2004 ; Keller et al., 2005). Ces travaux traitent de la propagation de modifications à travers une architecture ou aux conséquences d'une évolution technologique sur l'architecture globale du produit.
- *La conception modulaire d'une famille de produit et l'intégration des contraintes de désassemblage ou d'assemblage en conception* : (Baldwin et Clark, 2000 ; Agard et Tollenaere, 2002 ; Addouche, 2003 ; Blaise, 2008 ; Frédéric, 2010 ; Saïd (2013, 2014, 2018)). Ces travaux ne proposent pas une méthode d'amélioration du degré de modularité et du coût de couplage des modules. La non proposition d'une telle méthode peut occasionner un frein de la conception modulaire et par conséquent, susciter un problème de maintenance (désassemblage difficile et coût de couplage plus élevé), car plus le degré de modularité à l'intérieur des modules est élevé, plus l'assemblage ou le désassemblage des composants à l'intérieur du module est facile et plus le coût de couplage des modules est élevé, plus les modules sont de trop grandes tailles et le système devient plus complexe. C'est fort de cette analyse que la problématique de cette thèse est :

Comment minimiser les coûts de couplage des modules pour une maintenance plus aisée par la proposition d'une méthode de conception robuste ?

I.5 METHODOLOGIE

La méthodologie suivie dans le cadre de cette thèse pour obtenir des réponses à la problématique se subdivise en trois étapes, l'application est faite sur un torréfacteur de soja :

- ❖ Définir un indice d'évaluation de la meilleure solution de conception ;
- ❖ Modifier les fonctionnalités de l'algorithme de clustering existant ;
- ❖ Définir une meilleure architecture et reconfiguration du produit.

CHAPITRE II : CHOIX DE LA MEILLEURE SOLUTION DE CONCEPTION D'UN SYSTEME AU MOYEN D'UN INDICE-MULTI PARAMETRES

II.1 INTRODUCTION

Avec la prise de conscience des enjeux de développement durable et du renforcement du cadre législatif relatif à la gestion des déchets et des volumes de produits en fin de cycle de vie, l'urgence de mettre en place des modalités de développement à la fois performantes sur le plan économique, responsables sur le plan social et respectueuses de l'environnement devient essentielle pour les concepteurs. Une des stratégies qui se présentent à ces derniers, est de concevoir les produits (équipements) dans une vision de réutilisation des composants, de recyclage des matières et de remanufacturing (Haoues et *al.*, 2004 ; Millet et *al.*, 2007). Cette stratégie implique le désassemblage des produits. Dans ce contexte, les concepteurs sont amenés à intégrer de plus en plus tôt, les contraintes liées au désassemblage du produit dans le processus de conception afin de garantir sa désassemblabilité (démontabilité). Mais comment définir et évaluer la désassemblabilité d'un produit ? Quelles sont les étapes à suivre afin d'intégrer la désassemblabilité dans les phases préliminaires de conception ? Comment aider le concepteur à faire le choix d'une solution de conception qui satisfait des contraintes de désassemblabilité ?

Afin de répondre à ses questions, la section **II.2** présente une méthodologie d'évaluation de la complexité de désassemblage, suivie d'une application sur le torréfacteur de soja.

Motivations de l'étude

Le torréfacteur de soja est un équipement qui permet de soumettre à l'action du feu le soja, donc de le faire griller. En agro-alimentaire, la torréfaction est très importante ; en ceci qu'elle permet de réduire ainsi le taux de mycotoxines, d'augmenter le taux de protéines non dégradables et d'accroître le taux d'indigestion maximal. Il est donc important de choisir un torréfacteur fiable afin d'améliorer le rendement de torréfaction.

Problématique de conception du torréfacteur

Plusieurs travaux de la littérature ont porté sur l'étude des torréfacteurs, à l'instar de ceux de (Zeki, 1993 ; Ahouansou et Singbo, 2010 ; Alves De Macedo, 2017 ; Carré, 2020).

Ahouansou et Singbo (2010) ont évalué les performances techniques de deux torréfacteurs d'amande de karité : le torréfacteur traditionnel à la marmite et le torréfacteur manuel. Il se trouve que le torréfacteur traditionnel soumet les transformatrices à la chaleur et aux courbatures. La capacité horaire de production étant d'environ 9kg/h pour un temps de travail de 2,65 h. Le torréfacteur manuel se trouve donc être meilleur que le torréfacteur traditionnel, avec une capacité horaire d'environ 24 kg/h et un temps de travail d'environ 1h. Cependant il reste limité à l'utilisation d'une seule source de chaleur qui est le feu à bois, plutôt polluant et nuisible à l'environnement.

Carré (2020) va plus loin, en présentant plusieurs technologies de transformation du soja à la ferme. Ces technologies qui sont des torréfacteurs automatiques, bien qu'ayant plusieurs avantages (contrôle de température simple, pouvant fonctionner sans présence humaine, temps de séjour contrôlé, robustes, ...), ont également des inconvénients (coût de maintenance élevé, problème d'encombrement, énergie couteuse, coût d'investissement relativement élevé, localisation des défauts d'usures de vis parfois difficile en fonction de la direction de désassemblage complexe, démontabilité difficile,...).

Pour une conception plus aisée des systèmes en général pour des fins de rendements (des torréfacteurs en particulier), il serait judicieux d'intégrer les contraintes de désassemblage dès la phase initiale de conception. D'où la problématique de conception du torréfacteur.

II.2. METHODOLOGIE D'EVALUATION DE LA COMPLEXITE DE DESASSEMBLAGE

La méthodologie adoptée est décrite par l'ordinogramme suivant :

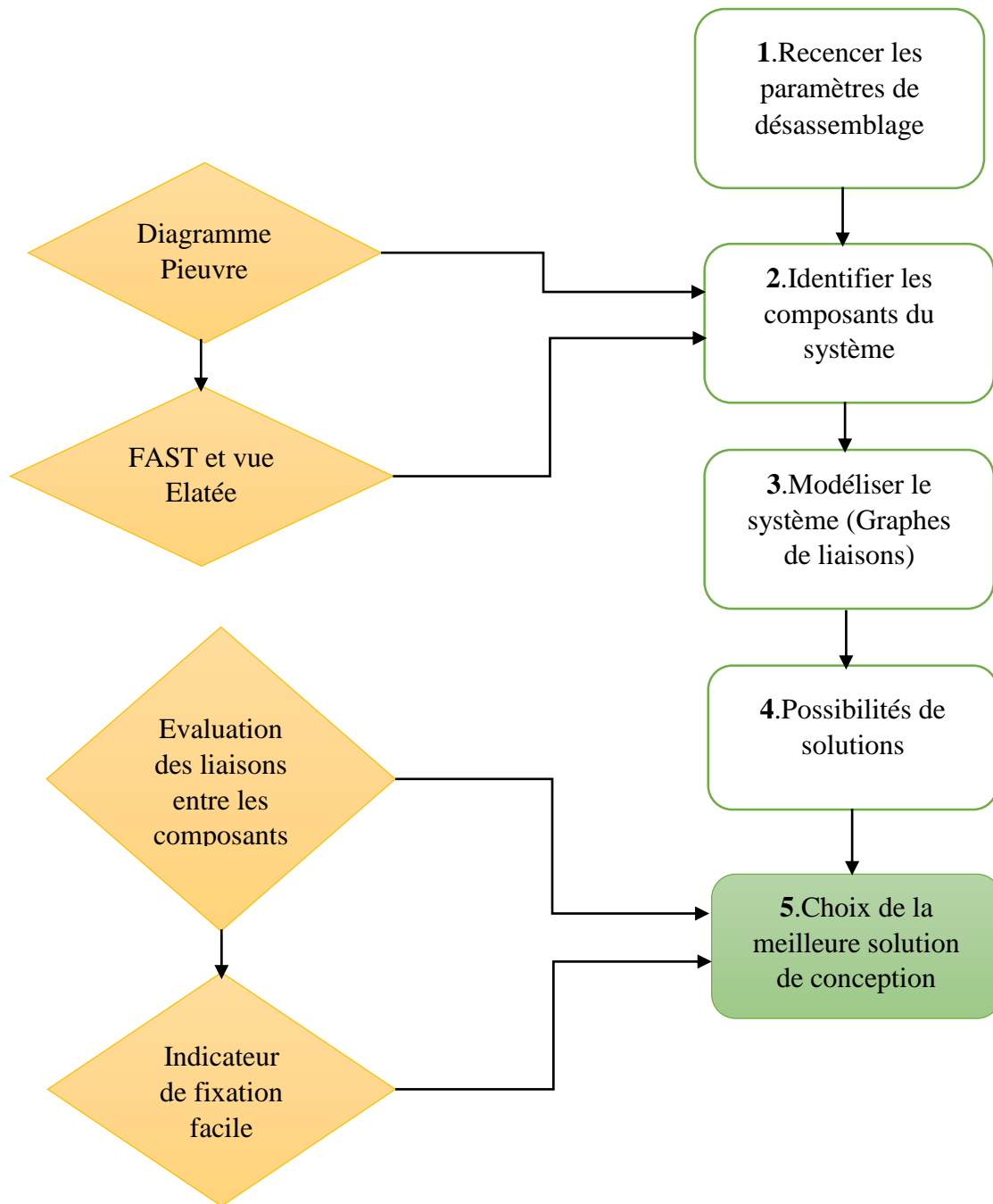


Figure 23 : Description de la méthodologie

II.2.1 Choix des paramètres de désassemblage

II.2.1.1 Paramètres susceptibles de représenter les contraintes de désassemblage

Une fois la fin de vie du produit atteinte, différentes fins de vie sont possibles allant de la mise en décharge jusqu'à la remise en état et la réutilisation du produit. Il est bien souvent nécessaire de désassembler le produit (système) afin de séparer les différents éléments qui ont encore de la valeur ou qui peuvent être réutilisés (Jeandin, 2015). Suivant les choix qui ont été faits, le produit peut être désassemblé complètement ou partiellement afin de ne récupérer que

les pièces qui présentent le plus d'intérêt. Différentes techniques de désassemblage sont également possibles. Dans le meilleur des cas, le produit peut être désassemblé sans endommager les pièces en retirant les attachements. Mais bien souvent cela est trop fastidieux pour être mis en place. Il est alors possible de se tourner vers des techniques de désassemblage dites semi-destructives. Cela correspond au fait de détériorer une ou plusieurs pièces d'un assemblage afin de récupérer les pièces ou les matériaux les plus intéressants. Le gain de temps est très important sans pour autant engendrer une perte importante de valeur. Nous ne considérerons pas les techniques consistant à broyer le produit pour ensuite trier les différents matériaux car les attachements ont bien moins d'impact dans cette situation. En effet, le problème réside essentiellement dans le fait de parvenir à trier les différents matériaux et non à séparer les différentes pièces. Grâce aux travaux de Jeandin (2015) nous allons dresser une liste des paramètres permettant de représenter au mieux toutes les facettes du désassemblage. La liste est la suivante :

- **Temps de désassemblage** : Un des paramètres les plus importants car il est directement relié au coût de désassemblage et donc bien évidemment à la rentabilité des opérations de désassemblage. Un assemblage trop long à désassembler sera nettement moins intéressant et le désassemblage pourrait ne pas être effectué si ce-dernier est vraiment trop important.
- **Outils** : Suivant l'attachement choisi pour réaliser un assemblage, l'outil pour opérer le désassemblage sera spécifique et plus ou moins perfectionné. Le choix d'un désassemblage non-destructif ou semi-destructif influence également l'outil utilisé. Certains attachements nécessitent des outils spéciaux qui ont un coût nettement plus élevé qu'un simple tournevis et peuvent demander des formations pour être utilisés. D'autres problèmes rencontrés lors du désassemblage concernent la diversité des attachements et le possible endommagement de ceux-ci. Pour solutionner ces problèmes, des études sont faites sur des outils flexibles (Duflou et *al.*, 2008) facilitant le désassemblage de tous types d'attachements ou encore des outils spécifiques permettant de briser la tête des vis afin de libérer facilement les pièces même si l'attachement est endommagé (Pak et Sodhi, 2002).
- **Couple** : Plus la force nécessaire pour défaire un attachement est importante et plus les risques pour l'opérateur seront importants ou plus le système devra être robuste dans le cas d'un désassemblage automatisé.

- **Direction d'accès** : Un attachement peut être facilement démontable mais s'il n'est pas ou difficilement accessible, le temps de désassemblage va fortement être augmenté. Contrairement à l'assemblage durant lequel les pièces sont assemblées les unes après les autres, lors du désassemblage il n'est pas rare de s'intéresser à une pièce spécifique et les attachements servant à sa fixation ne sont pas nécessairement accessibles et il sera alors difficile de positionner l'outil pour opérer le désassemblage.
- **Complexité du mouvement de désassemblage** : Tout comme pour les opérations d'assemblage, le geste à effectuer pour défaire un attachement influence la facilité du désassemblage, le type de machine à utiliser.
- **Facilité de revente des pièces** : Ce paramètre regroupe tous les problèmes qui peuvent être rencontrés suite au démontage d'une pièce ou d'un sous-assemblage. Dans certains cas comme l'aéronautique, les pièces doivent être rectifiées avant de pouvoir être réutilisées dans un nouveau produit. Des réparations ou des opérations d'usinage peuvent être nécessaires. Les attachements assurant un désassemblage sans dommages sur les pièces seront généralement plus intéressants à utiliser.
- **Possibilité d'automatiser le désassemblage** : L'automatisation de l'assemblage des produits a permis des gains importants de temps, des diminutions de coûts. Il peut en être de même en désassemblage même si d'autres problèmes se posent. En effet, la diversité des pièces à désassembler impose l'utilisation de lignes flexibles capables de s'adapter à tous les designs. Cependant si nous prenons pour exemple les appareils photos jetables dans Duflou et *al.* (2008), la ligne de désassemblage qui a été mise au point permet des gains importants et met bien en avant le fait que le désassemblage est loin d'être seulement une contrainte supplémentaire.
- **Possibilité de réaliser un désassemblage non-destructif** : Le choix de tel ou tel attachement lors de la phase de conception va directement influencer la possibilité d'opérer ou non un désassemblage non-destructif. Ceci est très important dans le cas où la pièce ou le sous-assemblage doit être réutilisé.
- **Risque de dégradation** : Lors du désassemblage il est possible que les pièces soient endommagées. Si ces pièces doivent être réutilisées il sera préférable d'opter pour des attachements assurant un démontage avec le moins de risques d'endommagement possibles.

- **Qualification des ouvriers** : La qualification des ouvriers comprend non seulement les éventuelles formations nécessaires pour utiliser un outil ou une machine mais également le temps nécessaire pour prendre connaissance des informations relatives au produit qui va être désassemblé.
- **Équipements requis** : Suivant le produit désassemblé, les outils utilisés, différents équipements peuvent être requis. Dans la majeure partie des cas des gants suffisent, cependant si le produit contient des matières toxiques ou si des opérations de sciage sont effectuées d'autres protections sont nécessaires pour garantir la sécurité des opérateurs.
- **Type de fixation des pièces** : Pour effectuer les opérations de désassemblage il est parfois nécessaire de fixer les pièces. Ce paramètre est utilisé pour représenter les différentes situations possibles.
- **Identification des attachements** : Un des points qui pose le plus de problème lors du désassemblage est la capacité à identifier l'attachement utilisé pour réaliser une liaison. En effet, si nous prenons pour exemple les attachements solidaires, ils sont généralement masqués et difficiles à repérer, le temps de désassemblage va forcément en pâtir et les pièces risquent d'être endommagées à cause d'une mauvaise identification et donc d'un démontage inadéquat. Certains produits possèdent des marquages afin de préciser la position des attachements et ainsi faciliter le travail des opérateurs lors du désassemblage.
- **Résistance des matériaux** : Ce paramètre est particulièrement important pour le désassemblage semi-destructif. En effet, plus le matériau est résistant et plus le nombre de problèmes rencontrés sera conséquent, temps plus importants, efforts élevés, outils spécifiques, etc.

Tableau 8 : Les paramètres influençant le désassemblage et les références correspondantes

Paramètres	Sources
Temps de désassemblage	Rose, Ishii et Masui, 1998 ; Kroll et Carver, 1999 ; Das, Yedlarajah et Narendra, 2000 ; Güngör, 2006 ; Ghazilla, Taha, Yusoff, Rashid et Sakundarini, 2014
Outils	Chen, Navin-Chandra et Prinz, 1993; Kroll et Carver, 1999 ; Sonnenberge, 2001; Güngör, 2006

Force requise	Kroll et Carver, 1999 ; Das, Yedlarajiah et Narendra, 2000; Sonnenberge, 2001
Direction d'accès	Kroll et Carver, 1999; Rose, Ishii et Masui, 1998 ; Das, Yedlarajiah et Narendra, 2000
Complexité du mouvement de désassemblage	Sonnenberge, 2001;Güngör, 2006
Facilité de revente des pièces	Chen, Navin-Chandra et Prinz, 1993 ; Güngör, 2006 ; Ghazilla, Taha, Yusoff, Rashid et Sakundarini, 2014;
Possibilité d'automatiser le désassemblage	Ghazilla, Taha, Yusoff, Rashid et Sakundarini, 2014; Güngör, 2006
Possibilité de réaliser un désassemblage non-destructif	Sonnenberge, 2001;Güngör, 2006
Risque de dégradation	Sonnenberge, 2001 ; Güngör, 2006 ; Ghazilla, Taha, Yusoff, Rashid et Sakundarini, 2014
Qualification des ouvriers	Das, Yedlarajiah et Narendra, 2000;
Equipement requis	Das, Yedlarajiah et Narendra, 2000;
Type de fixation des pièces	Das, Yedlarajiah et Narendra, 2000;
Identification des attachements	Chen, Navin-Chandra et Prinz, 1993;
Résistance des matériaux	Chen, Navin-Chandra et Prinz, 1993; Sonnenberge, 2001

II.2.1.2 Paramètres représentant le désassemblage non destructif

Dans la partie précédente nous avons déterminé 14 paramètres susceptibles de représenter l'ensemble des problèmes rencontrés lors de cette dernière phase de la vie des produits. Ce qui est trop important pour être implémenté directement.

Nous allons dresser une liste plus réduite des paramètres permettant de représenter au mieux toutes les facettes du désassemblage non destructif d'un produit et pouvant être utilisés dans sa phase de conception préliminaire. La liste est la suivante :

- types des contacts ;
- types des combinaisons ;
- outils de désassemblage ;
- nombre de directions de désassemblage ;
- qualifications des opérateurs ;

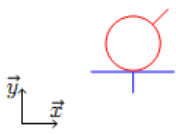
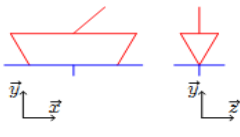
- équipements requis ;

Une échelle de valeurs a été créée pour chacun des paramètres permettant de calculer la complexité de désassemblage. Les différentes valeurs permettent de représenter les principales possibilités et de les classer de la plus intéressante à la plus contraignante. Nous allons voir en détail ces échelles et décrire chacune des valeurs. Pour choisir ces échelles de valeurs, nous nous sommes principalement appuyés sur trois articles (Kroll et Hanft, 1998 ; Kroll et Carver, 1999 ; Das et *al.*, 2000).

II.2.1.2.1 Types de contacts

Les contacts représentent la relation physique entre deux composants. Deux composants C_1 , C_2 sont en contact si : $\exists S M_{1i} \in C_1, S M_{2i} \in C_2$, telle que $S M_{2i}$ coïncide avec $S M_{1i}$. Cette relation est créée par le contact d'une surface associée au composant C_1 sur une surface associée au composant C_2 dites surfaces fonctionnelles. Ces surfaces sont de nature différente (plane, cylindrique, sphérique, conique,...). Par conséquent, la zone de contact dépend des surfaces en contact. La zone de contact peut être : surfacique, linéique, ponctuelle ou toute combinaison entre ces éléments selon la position relative de ces surfaces et leurs caractéristiques. La forme des pièces en contact varie. Pour reconnaître la liaison, il vaut mieux se fier à la nature de la zone de contact. Le tableau 9 présente les zones de contacts, les liaisons associées et les efforts transmissibles.

Tableau 9 : Zones de contacts, liaisons associées et efforts transmissibles.

Zones de contacts	Liaisons	Représentation plane	Efforts transmissibles
Point	Liaison sphère-plan (liaison ponctuelle)		$A \begin{Bmatrix} 0 & 0 \\ Y_{12} & 0 \\ 0 & 0 \end{Bmatrix}_{\mathcal{R}}$
Segment de droite	Liaison rectiligne		$A \begin{Bmatrix} 0 & 0 \\ Y_{12} & 0 \\ 0 & N_{12} \end{Bmatrix}_{\mathcal{R}}$
Cercle ou arc de cercle	Liaison annulaire (ligne en anneau)		

			$A \begin{Bmatrix} 0 & 0 \\ Y_{12} & 0 \\ Z_{12} & 0 \end{Bmatrix}_{\mathcal{R}}$
Surface plane	Liaison appui plan		$A \begin{Bmatrix} 0 & L_{12} \\ Y_{12} & 0 \\ 0 & N_{12} \end{Bmatrix}_{\mathcal{R}}$
Surface cylindrique	Liaison pivot glissant		$A \begin{Bmatrix} 0 & 0 \\ Y_{12} & M_{12} \\ Z_{12} & N_{12} \end{Bmatrix}_{\mathcal{R}}$
Surface conique	Liaison pivot		$A \begin{Bmatrix} X_{12} & 0 \\ Y_{12} & M_{12} \\ Z_{12} & N_{12} \end{Bmatrix}_{\mathcal{R}}$
Surface hélicoïdale	Liaison hélicoïdale		$A \begin{Bmatrix} X_{12} & -hX_{12} \\ Y_{12} & M_{12} \\ Z_{12} & N_{12} \end{Bmatrix}_{\mathcal{R}}$ avec $h = \pm \frac{p}{2\pi}$
Surface sphérique	Liaison rotule		$A \begin{Bmatrix} X_{12} & 0 \\ Y_{12} & 0 \\ Z_{12} & 0 \end{Bmatrix}_{\mathcal{R}}$

L'existence d'au moins un contact entre les surfaces de deux composants, entraîne un contact entre les deux composants en question. Le Tableau 10 représente l'échelle de valeur créée pour les types de contacts.

Tableau 10:Types de contacts.

Types des contacts	Échelle de valeur
Aucun contact	1
Ponctuel	2
Linéaire	3
Surfacique	4

Multi point contact	5
Multi surface contact	6

II.2.1.2.2 Types de combinaisons

Les combinaisons représentent les éléments qui mettent une énergie de cohésion destinée à pérenniser certains contacts. Ces énergies de cohésion se matérialisent par des éléments solides ou liquides mais aussi par des actions chimiques, mécaniques. Le Tableau 11 représente l'échelle de valeur créée pour les types de combinaisons.

Tableau 11:Type de combinaisons

Types de combinaisons	Échelle de valeur
Mettre	1
Insertion, vissage, rivetage	2
Tourner	3
Combiner	4
Collage, soudage	5

II.2.1.2.3 Outils de désassemblage

Les informations concernant les outils utilisés pour rompre une solidarisation entre deux composants C_1 et C_2 considèrent les difficultés pour les outils opératoires lors du désassemblage. Bien évidemment, cela dépend du type de solidarisation et de sa complexité.

Une valeur importante de type d'outil correspondante à une solidarisation signifie qu'il est coûteux ou plus compliqué de désassembler deux composants reliés par cette solidarisation. L'idéal étant de pouvoir désassembler sans outil. Desai et Mital (2003) proposent deux échelles pour les outils servant à exercer une force ou un moment. Le Tableau 12 représente l'échelle de valeur créée pour les outils de désassemblage.

Tableau 12:Outils de désassemblage.

Outils de désassemblage	Échelle de valeur
Aucun	1
Outil à air comprimé	2
Outillage mécanique	3
Fourni par le fabricant	4

Spécifique	5
Improvisé	6

II.2.1.2.4 Directions de désassemblage

Un composant peut-être facilement désassemblé, mais s'il n'est pas ou difficilement accessible, le temps de désassemblage va fortement augmenter. Contrairement à l'assemblage durant lequel les composants sont assemblés les uns après les autres, lors du désassemblage il n'est pas rare de s'intéresser à un composant spécifique. Il sera alors difficile de positionner l'outil pour opérer le désassemblage. Le Tableau 13 représente l'échelle de valeur créée pour les directions de désassemblage.

Tableau 13:Nombre de directions de désassemblage

Nombre des directions de désassemblage	Échelle de valeur
Par-dessus	1
Sur les côtés	2
Plus de 15cm de profondeur	3
Par-dessous	4
Axes combinés	5
Sans visibilité	6

II.2.1.2.5 Qualification des opérateurs

La qualification des opérateurs comprend non seulement les éventuelles formations nécessaires pour utiliser un outil ou une machine, mais également le temps nécessaire pour prendre connaissance des informations relatives au produit destiné au désassemblage. Le Tableau 14 représente l'échelle de valeur créée pour la qualification des opérateurs.

Tableau 14:Qualifications des opérateurs

Qualifications des opérateurs	Échelle de valeur
Aucune	1
10 à 20 secondes	2
Plus de 30 secondes	3
Discussion	4
Formation	5

II.2.1.2.6 Équipements requis

Suivant le produit désassemblé, les outils utilisés... différents équipements peuvent être requis. Dans la majeure partie des cas, des gants suffisent, cependant si le produit contient des matières toxiques ou si des opérations de sciage sont effectuées d'autres protections sont nécessaires pour garantir la sécurité des opérateurs. Le Tableau 15 représente l'échelle de valeur créée pour les équipements requis.

Tableau 15:Équipements requis.

Équipements requis	Échelle de valeur
Aucun	1
Gants	2
Masques	3
Protection anti-feu	4
Filtration de l'air	5
Combinaison intégrale	6

II.2.1.3 Pondération des paramètres de désassemblage

Nous avons étudié différentes méthodes traitant du désassemblage afin de déterminer une pondération pour les paramètres cités précédemment et de voir lesquels sont les plus influents (Das et *al.*, 2000). Pondérer les paramètres correspond à donner plus ou moins d'importance à un paramètre. La somme des pondérations est égale à 100. Les différentes alternatives seront donc classées sur une échelle de 0 à 100 et plus le score sera élevé, plus le désassemblage sera considéré comme complexe. Les pondérations appliquées aux paramètres sont présentées dans le Tableau 16.

La pondération la plus importante a été donnée aux nombres de directions et aux outils de désassemblage, car ils jouent un rôle important dans la complexité de désassemblage car des outils complexes et certaines directions de désassemblage peuvent être très contraignants et limiter l'intérêt d'un désassemblage. La qualification des opérateurs et les équipements requis ont reçu des pondérations moins importantes, car ces deux paramètres ont moins des conséquences sur la difficulté de désassemblage et les coûts associés en comparaison des autres paramètres.

Tableau 16: Pondérations des paramètres de désassemblage.

Paramètres	Pondération
Nombre de directions de désassemblage	30
Outils de désassemblage	25
Types des contacts	20
Types des combinaisons	15
Qualifications des opérateurs	5
Équipements requis	5

II.2.2 Identification des différents composants du système

L'identification des milieux extérieurs et la formulation du besoin du système se fait par le biais du diagramme Pieuvre ou graphe des intersections. Ce diagramme est constitué du produit, au centre, et, autour, des éléments de son environnement (milieu extérieur). On fait apparaître les relations (fonctions) entre le produit et le milieu extérieur. La structure d'un tel diagramme est donné par la figure 24.

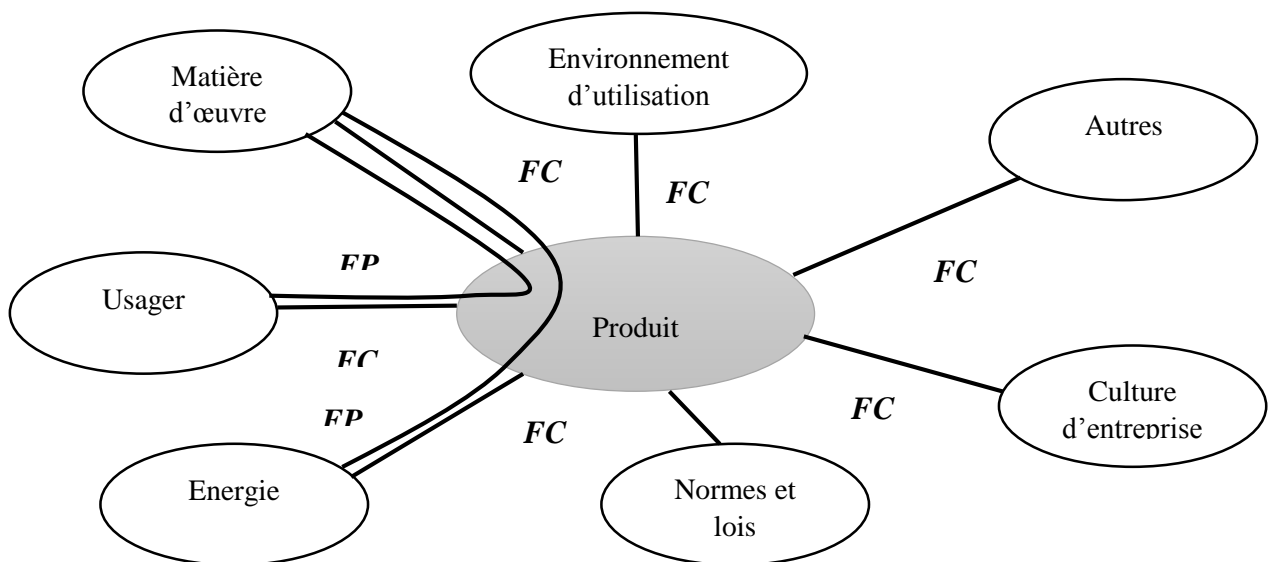


Figure 24: Structure d'un diagramme Pieuvre

Le FAST (Function Analysis System Technique) sera donc élaboré, qui à partir d'une fonction de service permettra d'aboutir aux fonctions techniques et aux solutions constructives associées. Il détaille en plusieurs niveaux la réalisation d'une ou plusieurs fonctions de service. La structure du diagramme FAST est donnée par la figure 25.

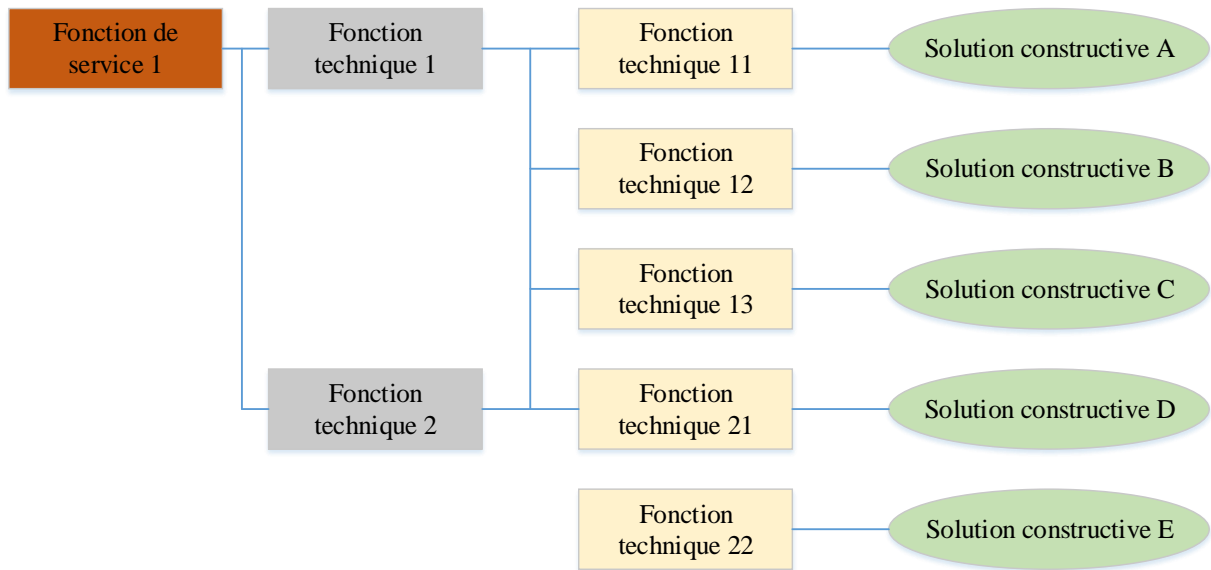


Figure 25: Structure d'un diagramme FAST

A l'issue de ce FAST, une vue éclatée de la solution sera générée automatiquement et avec l'appui d'une légende qui permettra d'avoir une vision globale du système en termes de composants.

II.2.3 Graphes des liaisons

Le Graphe des liaisons (*Component Mating Graph, CMG*) représente le produit (système) au niveau des composants. Les nœuds et les arcs de ce graphe représentent, respectivement, les composants et les liaisons entre les composants (Zussman et *al.*, 1994 ; Murayama et *al.*, 2001). Les liaisons peuvent être orientés ou non orientés. Les liaisons non orientées désignent l'existence d'une relation géométrique (contrainte de contact) entre deux composants tels que contact plan ou cylindrique mais aussi un contact causé par les différents types d'attachement tels vissage (screwing), ajustement par rupture (snap-fit), ajustement par pression (press fit). Les liaisons orientées, quant à elles désignent une information de précedence (contrainte de non contact) entre deux composants tels que les informations concernant l'accessibilité à une tâche de désassemblage (Li et *al.*, 2002). Cette approche de modélisation est illustrée sur la Figure 26.

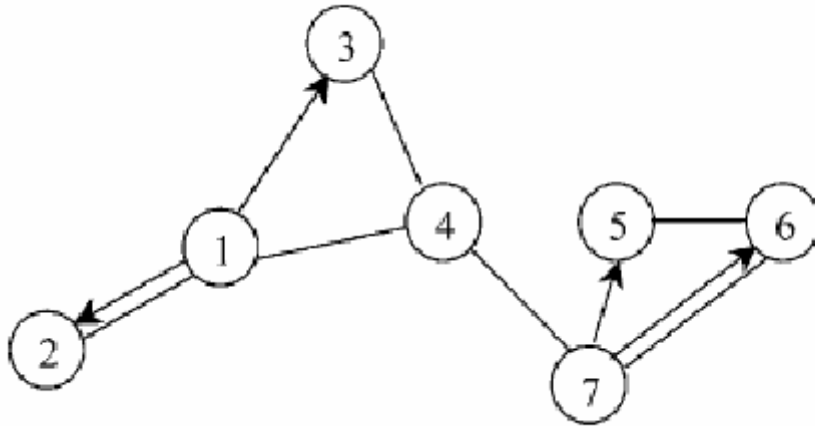


Figure 26: Graphe des liaisons hybride (Li et al., 2002).

II.2.4 Possibilités de solutions

Pour une conception donnée qui tient compte des paramètres de désassemblages, tels que présentés plus haut, on peut envisager plusieurs solutions de conception. Parmi ces solutions de conceptions, peuvent figurer trois types de solutions, celles qui sont : mauvaises, pas satisfaisantes et satisfaisantes. La solution de conception est appréciée au moyen d'un indicateur (par exemple l'indice de fixations faciles) qui est un nombre réel compris entre 0 et 100%. Plus il est proche de 100%, plus la contrainte de séparabilité est satisfaite, plus il est proche de 0% moins cette contrainte est satisfaite. Le tableau 17 donne les seuils de satisfaction de l'indicateur de fixation facile.

Tableau 17: Seuils de satisfaction de l'indicateur de fixations faciles

0	25%	50%	100%
Pas satisfaisante		Satisfaisante	Bonne

II.2.5 Proposition d'un indicateur d'évaluation de la solution de conception

En faisant la synthèse bibliographique sur les critères de conception pour le désassemblage, il en ressort **six principaux critères** (indicateurs de fixations identiques, indicateur du nombre de composants aux matériaux identiques, indicateur du poids de composants aux matériaux identiques, indicateur du nombre de fixation ayant la même direction d'accès et indicateur de fixations faciles) (Dowie and Simon, 1994 ; Dowie, 1995).

De ces indicateurs, l'*indicateur de fixations faciles* est le plus pertinent car c'est le seul qui tient compte de la complexité du mouvement de désassemblage. En effet, le geste à effectuer

pour défaire un attachement influence sur la facilité du désassemblage, ce qui n'est donc pas pris en compte par les autres indicateurs. Les paramètres de désassemblage peuvent donc être qualifiés selon une échelle prédéterminée. Afin de qualifier les paramètres de désassemblage, une pondération est proposée qui augmente avec la difficulté à rompre la fixation considérée au moyen des coefficients qui définissent le niveau de sophistication de ces paramètres (Dowie and Simon, 1994 ; Dowie, 1995 ; Said, 2013).

Cependant, l'indicateur de fixations faciles proposé par les auteurs tient uniquement compte de trois paramètres de désassemblage : le type d'outils, le nombre de directions de désassemblage ou d'accès et le type de combinaisons ou de fixations. Bien souvent, le désassemblage du produit est trop fastidieux pour être mis en place. En effet, très souvent celui qui effectue l'opération de désassemblage ou d'assemblage n'a pas d'équipements de protection individuelle (par exemple les gants) pour être habilité à retirer les attachements du produit ,ce qui en un moment donné peut conduire à mettre plus de temps que prévu .D'autre part, l'absence de qualification de l'opérateur à intervenir sur le produit, peut conduire à un désassemblage destructif (dans la mesure où la pièce est endommagée pendant le désassemblage),ce qui pourrait conduire au recyclage (engendrant ainsi des coûts supplémentaires) en vue d'une récupération future de la pièce endommagée. La contrainte de temps est donc un paramètre très important car elle est directement reliée au coût de désassemblage et donc bien évidemment à la rentabilité des opérations de désassemblage. Un assemblage trop long à désassembler sera nettement moins intéressant et le désassemblage pourrait ne pas être effectué si ce-dernier est vraiment trop important (Jeandin, 2015). On comprend dès cet instant que pour un désassemblage efficace, il est plus pertinent de considérer plus de trois paramètres, donc dans le meilleur des cas les six paramètres de désassemblage non destructifs. Le nouvel indicateur que nous proposons qui tient compte des six paramètres de désassemblage cités plus haut est donné par la formule suivante :

$$If_f = \frac{\sum_{i=1}^n (A_{NDD} CNDD_i + A_{OD} COD_i + A_{TC} CTC_i + A_{TCO} CTCO_i + A_{QO} CQO_i + A_{ER} CER_i)}{n} \quad (61)$$

Avec :

CNDD : complexité de directions de désassemblage,

COD : complexité de l'outil de désassemblage,

CTC : complexité du type de contact,

CTCO : complexité du type de combinaison,

C_{QO} : complexité de la qualification des opérateurs,

CER : complexité des équipements requis,

n : nombre de liaisons concernées,

I_{f_f} : Indicateur de fixation facile (en %).

La complexité du paramètre considéré dans la liaison i est donnée par la formule :

$$C_i = a_i I_i p_i \quad (62)$$

Avec :

a_i : coefficient du paramètre représentant le désassemblage considéré,

I_i : valeur assignée au paramètre de désassemblage considéré (Tableaux 9 à 14),

p_i : pondération du paramètre de désassemblage considéré (Tableau 15).

Il est à noter que la pondération de chaque paramètre de désassemblage est déjà connue, et la valeur assignée à chaque paramètre de désassemblage sera extraite du tableau des liaisons entre les composants du système ou produit d'étude. S'agissant du coefficient du paramètre représentant le désassemblage, les tableaux qui suivent illustreront le calcul.

Les termes $A_{NDD}, A_{OD}, A_{TC}, A_{TCO}, A_{QO}$ et A_{ER} sont respectivement les poids des différents facteurs de l'indicateur I_{f_f} . Ces derniers sont calculés comme suit :

$$A_k = \frac{2(m+1-k)}{m(m+1)} \quad (63)$$

avec :

m : nombre de paramètres de désassemblages considérés (soit **six** pour notre formule),

k : nombre correspondant à l'ordre de priorité des facteurs.

D'après ce qui précède, la pondération la plus importante est donnée aux nombres des directions et aux outils de désassemblage, et la moins importantes est donnée à la qualification des opérateurs et aux équipements requis. En tenant compte de cet ordre de priorité, les valeurs attribuées de k sont données ainsi qu'il suit :

$k = 1$ pour le nombre de directions de désassemblage,

$k = 2$ pour les outils de désassemblage,

$k = 3$ pour les types de contacts,

$k = 4$ pour les types de combinaisons,

$k = 5$ pour la qualification des opérateurs,

$k = 6$ pour les équipements requis.

Les poids des différents facteurs sont donc les suivants :

$$A_{NDD} = 0,28 \quad ; \quad A_{OD} = 0,23 \quad ; \quad A_{TC} = 0,19$$

$$A_{TCO} = 0,14 \quad ; \quad A_{QO} = 0,09 \quad ; \quad A_{ER} = 0,04.$$

En suivant la démarche proposée dans (Dowie and Simon, 1994 ; Dowie, 1995), les coefficients associés à chaque paramètre de désassemblage sont donnés par les tableaux suivants (la pondération proposée augmente avec la difficulté de rompre la fixation considérée) :

Tableau 18:Exemple des coefficients associés aux nombres de directions d'accès

Nombre des directions de désassemblage	Coefficients
Par-dessus	1/6
Sur les côtés	1/5
Plus de 15cm de profondeur	1/4
Par-dessous	1/3
Axes combinés	1/2
Sans visibilité	1

Tableau 19:Exemple des coefficients associés aux outils de désassemblage

Outils de désassemblages	Coefficients
Aucun	1/6
Outil à air comprimé	1/5
Outillage mécanique	1/4
Fourni par le fabricant	1/3
Spécifique	1/2
Improvisé	1

Tableau 20:Exemple des coefficients associés aux types de contacts

Types de contacts	Coefficients
Aucun	1/6
Ponctuel	1/5
Linéaire	1/4
Surfacique	1/3

Multi point contact	1/2
Multi surface contact	1

Tableau 21:Exemple des coefficients associés à aux types de combinaisons

Types de combinaisons	Coefficients
Pose	1/5
Insertion, vissage, rivetage	1/4
Tourner	1/3
Combiner	1/2
Collage, soudage	1

Tableau 22:Exemple des coefficients associés à la qualification des opérateurs

Qualifications des opérateurs	Coefficients
Aucune	1/5
10 à 20 secondes	1/4
Plus de 30 secondes	1/3
Discussion	1/2
Formation	1

Tableau 23:Exemple des coefficients associés aux équipements requis

Équipements requis	Coefficients
Aucune	1/6
Gants	1/5
Masques	1/4
Protection anti-feu	1/3
Filtration de l'air	1/2
Combinaison intégrale	1

II.3 RESULTATS

Cette partie illustre la mise en application de la démarche d'évaluation de la complexité de désassemblage intégrant les contraintes de désassemblage, proposée dans la partie précédente, sur un produit : le *torréfacteur de soja*. Le cas d'application actuel fera également l'objet des chapitres suivants.

Avec le flux important de matières premières (soja) à griller ou à torréfier, il se trouve que c'est d'avantage plus pénible de le faire traditionnellement. C'est ainsi qu'en automatisant le procédé de torréfaction, on utilise des torréfacteurs qui présentent l'avantage de réduire considérablement le temps de torréfaction et de perte d'énergie. La création d'un diagramme pieuvre (Figure 27) en phase d'utilisation du torréfacteur, permet d'identifier les milieux extérieurs au produit et de formuler les fonctions de services (Tableau 24) qu'il doit satisfaire.

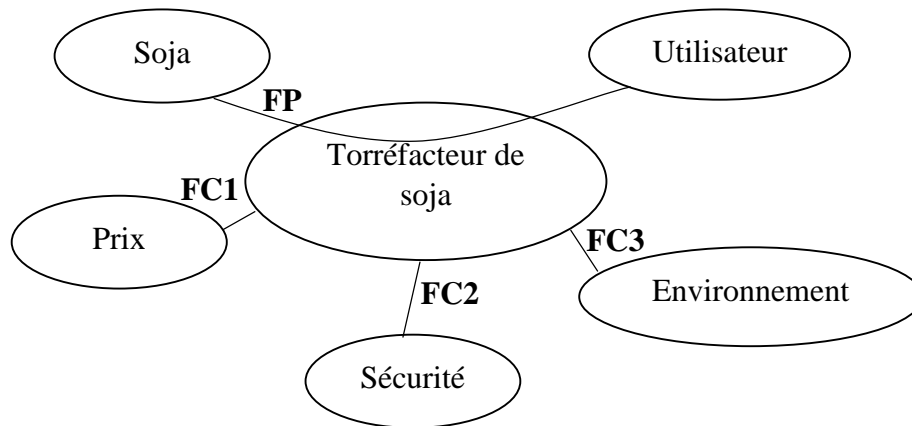


Figure 27:Diagramme pieuvre du torréfacteur de soja en phase d'utilisation

Tableau 24:Fonctions de services.

FP	Permettre à l'utilisateur de faire torréfier le soja
FC1	Etre facilement accessible financièrement(coût minimal)
FC2	Ne pas représenter de danger pour l'utilisateur
FC3	Doit s'adapter à l'environnement

Rappelons que *FP* est la fonction principale et que *FC* est la fonction complémentaire.

II.3.1. Identification des composants du système

II.3.1.1 FAST du torréfacteur de soja

Le FAST du torréfacteur de soja est donné par la figure 28.

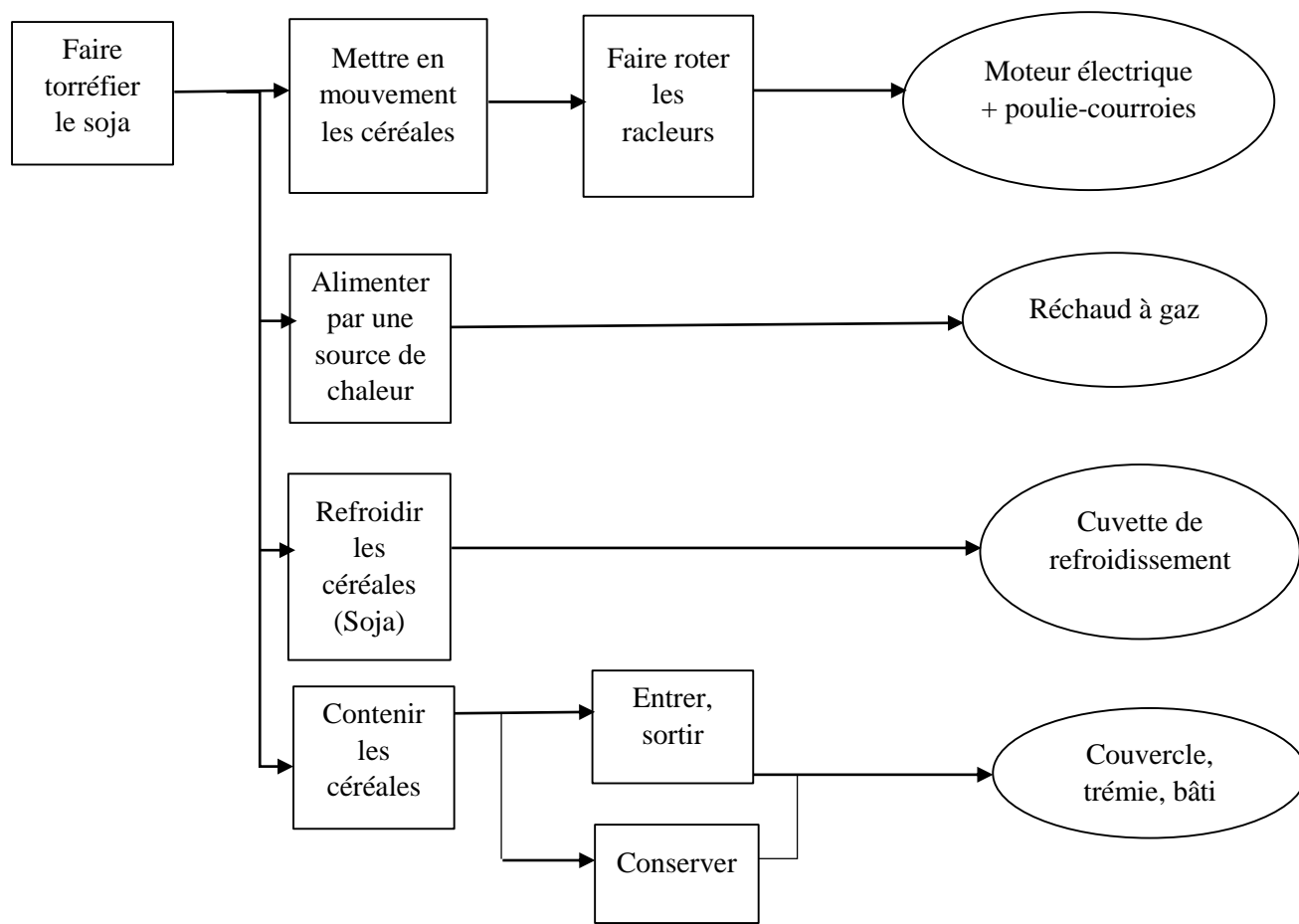


Figure 28:Le FAST relative à la fonction torréfier.

Suite à l'analyse fonctionnelle, basée sur le FAST nous avons identifié les différentes fonctions techniques permettant de réaliser la fonction de service : « *permettre à l'utilisateur de faire torréfier le soja* ». Nous avons retenu les fonctions techniques :

- Mettre,
- Alimenter,
- Refroidir,
- Contenir.

II.3.1.2 Vue éclatée du torréfacteur de soja

Les fonctions techniques et sous-fonctions techniques nous ont permis de détailler les solutions technologiques. Le torréfacteur de soja est constitué de 19 composants tels que listés dans le Tableau 25. La Figure 29 représente une illustration du produit décrivant une solution en vue éclatée.

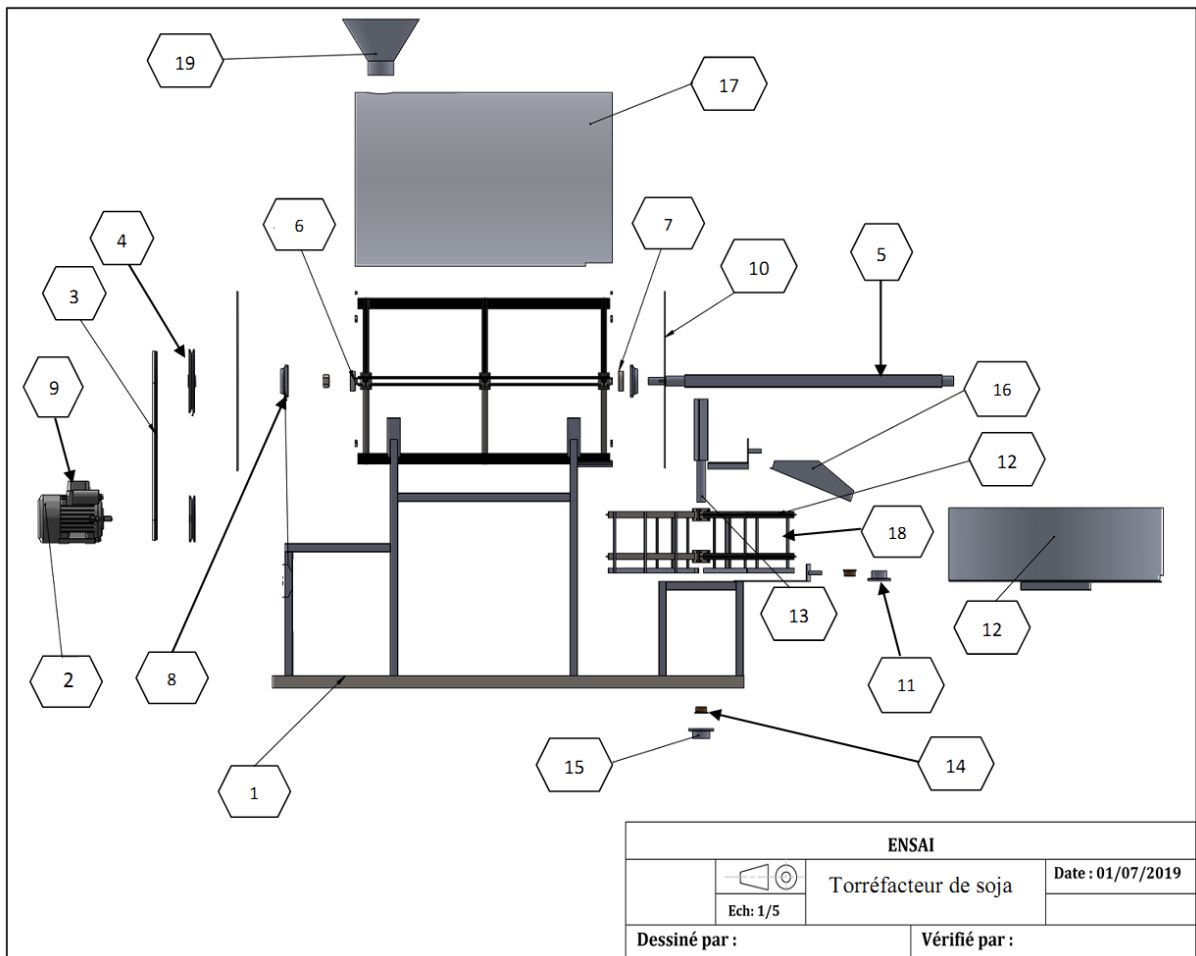


Figure 29:Vue éclatée de la solution.

Tableau 25:Légende des composants.

Repère des composants	Noms des composants
1	Bâti
2	Moteur
3	Courroie
4	Disque 2(poulie)
5	Axe principal
6	Roulement axe principal 1
7	Roulement axe principal 2
8	Support roulement côté moteur
9	Disque moteur
10	Capot
11	support roulement côté cuvette

12	Cuvette
13	Axe secondaire
14	Roulement axe secondaire
15	support roulement axe secondaire
16	Trémie de sortie
17	Cylindre principal
18	Racleur
19	Trémie d'entrée

II.3.2. Modélisation du produit par le graphe de liaisons

La structure du produit permet de construire le graphe des liaisons présenté à la figure 30.

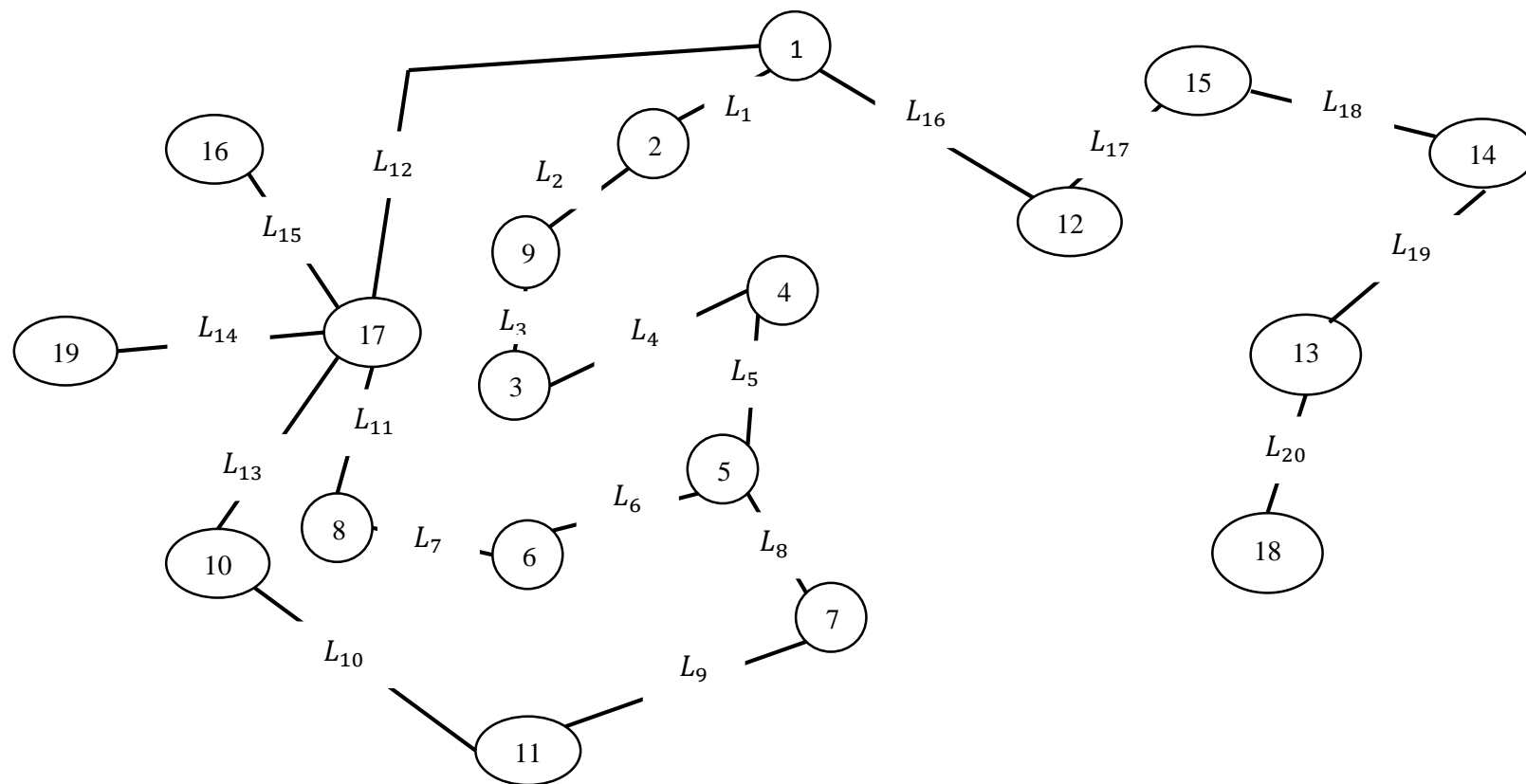


Figure 30: Graphe des liaisons de du torrificateur de soja.

II.3.3 Evaluation de la complexité de désassemblage

Comme mentionnée dans la méthodologie, un indice I_{ff} (équation 61) est choisi comme indicateur en conception afin d'évaluer la complexité de désassemblage d'une solution générée par le concepteur.

La complexité de désassemblage du torréfacteur de soja est évaluée à partir de six paramètres proposés pour deux solutions dont les caractéristiques sont les suivantes :

- **Solution 1** : le torréfacteur est constitué de deux grands ensembles : l'ensemble supérieure (*bâti-cylindre principal*) et l'ensemble inférieure (*bâti-cuvette*).

L'ensemble supérieure se subdivise en trois parties. La première partie est composée des pièces 1, 17, 19, 10, 16, 11 et 8. La deuxième partie est composée des pièces 2, 3 et 9. La troisième partie est composée des pièces 4, 5, 6 et 7. L'assemblage entre ces trois parties est assuré par système vis-écrou.

L'ensemble inférieure se subdivise en deux parties. La première partie est composée des pièces 1, 14 et 15. La deuxième partie est composée des pièces 12, 13 et 18. L'assemblage entre ces deux parties est assuré par système vis-écrou.

- **Solution 2** : cette solution est similaire à la précédente, la seule différence est que les pièces 17 et 16 sont soudées.

Le tableau 26 fait une évaluation des liaisons entre les composants du produit et le tableau 27 ressort les valeurs des indices de fixations faciles pour chacune des solutions proposées, lorsqu'on applique le modèle développé dans ce travail qui considère les six paramètres de désassemblage non destructifs pour évaluer la solution de conception.

Tableau 26: Evaluation des liaisons entre les composants avec le nouveau modèle.

Liaisons	TC	OD	NDD	TCO	QO	ER	Valeur
1--2	6	3	5	4	3	2	23
2--9	4	1	4	3	1	1	14
3--9	4	3	5	2	2	2	18
3--4	4	1	4	3	1	1	14
4--5	4	3	5	4	3	2	21
5--6	4	3	5	2	3	2	19
5--7	4	3	5	2	3	2	19
6--8	6	1	5	2	2	2	18
7--11	6	1	5	2	2	2	18

11--10	4	3	5	4	3	2	21
17--10	6	3	5	2	2	2	20
17--8	4	3	5	4	3	2	21
17--1	6	3	5	4	3	2	23
17--16	6	3	5	4	3	2	23
1--12	6	3	5	4	3	2	23
15--14	4	3	4	2	2	2	17
12--15	4	3	5	4	3	2	21
14--13	4	3	5	4	3	2	21
18--13	4	3	5	4	3	2	21
19--17	4	3	5	4	3	2	21

Tableau 27: Bilan de l'évaluation avec le nouveau modèle

<i>Torréfacteur de soja</i>	<i>Solution 1</i>	<i>Solution 2</i>
I_{f_f} (en %)	$37 \pm 0,9$	$39 \pm 0,5$

D'après le tableau 27 les indices de fixations faciles obtenus pour les deux solutions proposées sont de l'ordre de 37 % pour la solution 1 et de 39 % pour la solution 2. Les deux solutions appartiennent à l'intervalle [25%, 50%] qui traduisent une solution satisfaisante. Mais la solution 2 présente l'indice de fixations faciles le plus proche de 50%, elle est alors jugée comme la solution de conception la plus respectueuse des contraintes de désassemblage en fin de vie que la solution 1.

En appliquant le modèle développé par Dowie and Simon (1994) et Dowie and D. a (1995) qui tient compte de trois paramètres désassemblages pour évaluer la solution de conception, le tableau 28 fait une évaluation des liaisons entre les composants du produit et le tableau 29 ressort les valeurs des indices de fixations faciles pour chacune des solutions proposées.

Tableau 28: Evaluation des liaisons entre les composants avec l'ancien modèle.

Liaisons	OD	NDD	TCO	Valeur
1--2	3	5	4	12
2--9	1	4	3	8
3--9	3	5	2	10
3--4	1	4	3	8
4--5	3	5	4	12
5--6	3	5	2	10

5--7	3	5	2	10
6--8	1	5	2	18
7--11	1	5	2	18
11--10	3	5	4	21
17--10	3	5	2	20
17--8	3	5	4	21
17--1	3	5	4	23
17--16	3	5	4	23
1--12	3	5	4	23
15--14	3	4	2	17
12--15	3	5	4	21
14--13	3	5	4	21
18--13	3	5	4	21
19--17	3	5	4	21

Tableau 29: Bilan de l'évaluation avec l'ancien modèle

<i>Torréfacteur de soja</i>	<i>Solution 1</i>	<i>Solution 2</i>
<i>If_f (en %)</i>	<i>24,05 ± 0,41</i>	<i>24,73 ± 0,64</i>

D'après le tableau 29, les indices de fixations faciles obtenus pour les deux solutions proposées sont de l'ordre de 24,05 % pour la solution 1 et de 24,73 % pour la solution 2. Les deux solutions appartiennent à l'intervalle [0%, 25%] qui traduisent une solution pas satisfaisante. Donc le modèle développé dans ce travail est meilleur que celui proposé par Dowie and Simon (1994) et Dowie and D. a (1995) car il a pu générer deux solutions de conception satisfaisantes.

II.4 CONCLUSION

Ce chapitre a été consacré au choix de la meilleure solution de conception au moyen d'un critère concernant l'intégration du désassemblage en fin de vie durant la phase de conception préliminaire. Une évaluation des solutions de conception d'un torréfacteur de soja a été présentée à l'issue de laquelle le choix de la solution de conception la plus respectueuse des contraintes de désassemblage a été fait grâce au modèle développé dans ce travail. Ces paramètres de conception issus de ce modèle seront fort utiles pour déployer l'algorithme de clustering au torréfacteur de soja.

CHAPITRE III : PROPOSITION D'UN OUTIL QUI S'APPUIE SUR L'ALGORITHME DE CLUSTERING

III.1 INTRODUCTION

Dans ce chapitre, nous proposons un outil qui s'appuie sur l'algorithme de clustering existant. Il s'agit d'un nouvel algorithme de clustering qui utilise le formalisme de représentation matricielle pour mettre en œuvre l'identification des architectures des domaines du produit (système ou équipement). Le principe retenu est d'utiliser les DSM (Design Structure Matrix) en entrée pour proposer des architectures optimales des produits. Notre contribution concerne l'amélioration d'un algorithme de clustering existant qui a servi de référence à plusieurs travaux antérieurs tels que ceux de (Idicula ,1995 ; Fernandez ,1998 ; Thebeau, 2001). Nous présentons d'abord les limites de l'algorithme existant, ensuite nous proposons plusieurs améliorations de cet algorithme, à partir de notre analyse bibliographique sur la conception modulaire et les indices de modularité.

III.2 METHODOLOGIE DE L'ALGORITHME DE CLUSTERING EXISTANT

L'algorithme de clustering existant repose sur plusieurs étapes, décrites ci-dessous.

III.2.1 Modélisation du produit

Les composants du produit (système) et leurs relations sont représentés par un graphe de liaisons, $G = (C, R)$ où chaque sommet représente un composant, et chaque couple de deux composants appelé « arc » représente les informations liées aux relations.

Soit $C = (C_1, C_2, C_3, \dots, C_n)$ l'ensemble de tous les composants du produit de cardinalité n .

Soit $R = (R_1, R_2, R_3, \dots, R_m)$ l'ensemble de toutes les relations entre les composants du produit de cardinalité m .

Pour chaque couple de composants du graphe, nous lui attribuons une valeur qui est la somme des paramètres représentant le désassemblage. Ainsi le résultat sera présenté sous la forme d'une matrice de relation. On définit la matrice de relation par un élément $c_{i,j}$ tel que :

$$c_{i,j} = \begin{cases} TC + TCO + OD + NDD + QO + ER, & \text{si } c_i \text{ et } c_j \text{ ont une relation} \\ 0, & \text{sinon} \end{cases} \quad (64)$$

avec :

- TC : Types des Contacts ;

- TCO : Types des Combinaisons ;
- OD : Outils de désassemblage ;
- NDD : Nombre de Direction de Désassemblage ;
- QO : Qualifications des opérateurs ;
- ER : Équipements requis ;
- k : indice de la liaison entre les composants c_i et c_j .

La valeur dans une cellule de la matrice est non nulle si les composants respectifs sont connectés et égale à zéro si ce n'est pas le cas. Pour des raisons de symétrie de la matrice, on a : $c_{i,j} = c_{j,i}$. D'après Lambert et *al.* (2005), le nombre maximal de connexions K dans un produit avec N composants est donné par l'expression suivante :

$$K = N(N - 1) \quad (65)$$

Si toutes les connexions dans un produit sont établies, c'est-à-dire que tous les éléments significatifs sont différents de 0, la matrice représente un produit fortement connecté, c'est-à-dire un produit dans lequel tous les composants sont mutuellement connectés.

III.2.2 Première étape de l'algorithme de clustering existant et choix d'une bonne topologie

III.2.2.1 Première étape de l'algorithme de clustering existant

Le **clustering** est un processus qui partitionne un ensemble de données en sous-classes (clusters ou grappes) ayant du sens.

Cette étape permet de réorganiser consécutivement les lignes et les colonnes de la matrice modélisant les relations entre les composants d'un produit, selon un indice de similarité jusqu'à l'obtention des blocs diagonaux. Ainsi tous les composants qui sont en interaction minimal sont regroupés ensemble pour former des modules.

Optimiser le regroupement, consiste donc à maximiser la similarité intra-classe et à minimiser la similarité inter-classes.

Pour trouver le nombre des modules, nous utilisons la formule proposée par (Ericsson et Erixon, 1999) :

$$0,5\sqrt{NCP} \leq NM \leq \sqrt{NCP} \quad (66)$$

avec :

- NCP : Nombre de Composants du Produit ;
- NM : Nombre de Modules du Produit.

Ce regroupement se fait en plusieurs étapes :

1. Générer un nombre de modules possibles, d'après l'équation 66.
2. Additionner les entrées de chaque colonne de la matrice des relations.
3. Choisir des composants avec les plus fortes intensités de liaisons et assigner les à un module.
4. Choisir une colonne parmi celles qui correspondent aux composants de l'étape précédente et tracer une ligne verticale.
5. Pour chaque entrée (valeur) différente de 0 coupées par la ligne verticale, tracer une ligne horizontale.
6. Assigner les composants au module auquel appartient le composant de la colonne choisie de sorte à maximiser les relations des composants du même module et à minimiser les relations entre les composants de modules différents.
7. Transformer la matrice en supprimant les composants assignés à un module.
8. Répéter les étapes 6 à 7 jusqu'à ce que chaque composant soit assigné à un module.

III.2.2.2 Choix d'une bonne topologie des modules

Le choix d'une bonne topologie des modules est fortement corrélé aux séquences ou gammes de désassemblage du produit. En effet, avec le processus de désassemblage, on peut identifier des composants qui peuvent être récupéré en parallèle (en même temps). On constate dès cet instant que lors d'une opération de maintenance faisant appel aux désassemblage, on peut se retrouver à démonter en même temps plusieurs composants. Ces composants présentent donc un mode de fonctionnement de type série (le dysfonctionnement de l'un des composants affecte systématiquement les autres). Les autres composants ne faisant pas partir de ce mode de fonctionnement (type parallèle) peuvent donc être désassemblé indépendamment l'un de l'autre. C'est ainsi qu'à l'issue des gammes de désassemblage, on définit une topologie des modules.

Pour obtenir la séquence de désassemblage optimale (conduisant à un désassemblage complet) , nous nous inspirons des méthodes développées par (Bourjault, 1984 ; Subramani et

Dewhurst, 1991 ; Lee et Kumana, 1992 ; Srinivasan et Gadh , 1998 ; Jabbour et Mascle, 1998 ; Kuo et *al.*, 2000 ; Srinivasan et Gadh, 2000 ; Gungor et Gupta, 2001 ; Moore et *al.*, 2001 ; Kongar et Gupta, 2001 ; Failli et Dini, 2001 ; Srinivasan et Gadh, 2002 ; Torres et *al.*, 2003 ; Wang et *al.* , 2003 ; Lambert et Gupta, 2005 ; Kongar et Gupta, 2006 ; Giudice et Fargione, 2007 ; Tseng et *al.*, 2009 ; Tripathi et *al.*, 2009 ; Zhang et Zhang, 2010 ; Xue et *al.*, 2010 ; Saïd, 2018).

La séquence de désassemblage générée par l’algorithme que nous proposons met en évidence l’ordre de désassemblage des modules du produit, puis les composants d’un module. Pour la génération de la séquence de désassemblage, trois données d’entrée sont nécessaires :

- la matrice de précédence entre les composants ;
- les directions de désassemblage des modules ;
- la matrice de précédence entre les modules .

La matrice de précédence entre les composants est définie à partir du graphe des liaisons et de l’architecture modulaire du produit. Un élément p_{ij} de la matrice est tel que :

$$p_{ij} = \begin{cases} 1, \text{ si le composant } i \text{ est désassemblé avant le composant } j \\ 0, \text{ si le composant } i \text{ n'est pas désassemblé avant le composant } j \\ x, \text{ si aucune contrainte de précédence existe entre les composants } i \text{ et } j \\ \text{cette notation peut prendre la valeur } 0 \text{ ou } 1 \end{cases} \quad (67)$$

Les entrées de la diagonale principale de la matrice de précédence sont égales à 0.

Pour illustrer au mieux les trois données d’entrées nécessaires dont nous aurons besoins, nous allons nous appuyer principalement sur les travaux de Saïd (2018).

La figure **31** donne une illustration de la *matrice de précédence* d’une lampe de bureau subdivisée en cinq modules (SE1, SE2, SE3, SE4 et SE5).

		SE 1		SE 2			SE 3				SE 4				SE 5
		3	6	1	4	2	7	8	9	10	14	11	12	13	5
SE	3	X	0	0	0	0	0	0	0	0	0	0	0	0	0
1	6	1	x	1	1	0	0	0	0	0	0	0	0	0	0
SE	1	1	1	x	1	1	0	0	0	0	0	0	0	0	0
2	4	0	0	0	x	0	0	0	0	0	0	0	0	0	0
	2	0	1	1	0	x	1	1	0	0	0	0	0	0	0
SE	7	0	0	0	0	1	x	0	0	1	0	0	0	0	0
3	8	0	0	0	1	1	1	x	0	1	0	1	0	0	0
	9	0	0	0	0	0	0	0	x	0	0	0	0	0	0
	10	0	0	0	0	1	1	1	1	x	1	0	0	0	0
SE	14	0	0	0	0	0	0	0	0	0	x	0	1	1	1
4	11	0	0	0	0	0	0	1	0	0	1	x	1	1	1
	12	0	0	0	0	0	0	0	0	0	0	1	x	0	1
	13	0	0	0	0	0	0	0	0	0	1	1	0	x	1
SE	5	0	0	0	0	0	0	0	0	0	0	0	0	0	x
5															

Figure 31:Exemple de matrice de précédence (Saïd, 2018)

Du fait que des composants appartiennent au même module, la valeur 0 est attribuée aux entrées de chaque module de valeur égal à 1. On change ainsi la *direction de désassemblage des modules* et la nouvelle matrice de précédence intégrant ce changement de direction est donnée à la figure 32.

		SE 1		SE 2			SE 3				SE 4				SE 5
		3	6	1	4	2	7	8	9	10	14	11	12	13	5
SE 1	3	X	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	x	1	1	0	0	0	0	0	0	0	0	0	0
SE 2	1	1	1	x	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	x	0	0	0	0	0	0	0	0	0	0
	2	0	1	0	0	x	1	1	0	0	0	0	0	0	0
SE 3	7	0	0	0	0	1	x	0	0	0	0	0	0	0	0
	8	0	0	0	1	1	0	x	0	0	0	1	0	0	0
	9	0	0	0	0	0	0	0	x	0	0	0	0	0	0
	10	0	0	0	0	1	0	0	0	x	1	0	0	0	0
SE 4	14	0	0	0	0	0	0	0	0	0	x	0	0	0	1
	11	0	0	0	0	0	0	1	0	0	0	x	0	0	1
	12	0	0	0	0	0	0	0	0	0	0	0	x	0	1
	13	0	0	0	0	0	0	0	0	0	0	0	0	x	1
SE 5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Figure 32: Matrice de précedence modifiée des composants (Saïd, 2018)

En considérant les modules et les relations existant entre eux, le tableau 30 donne une illustration d'une *matrice de précedence des modules*. Dans les modules et relations existant entre eux, s'il existe au moins un composant i désassemblé avant le composant j , la valeur assignée est 1. Dans le cas contraire, on assigne la valeur 0 (même s'il s'agit d'un désassemblage où aucune contrainte de précedence existe).

Tableau 30: Matrice de précedence pour les modules (Saïd, 2018)

	SE1	SE2	SE3	SE4	SE5	Niveau
SE1	0	1	0	0	0	1
SE2	1	0	1	0	0	2
SE3	0	1	0	1	1	3
SE4	0	0	1	0	1	2
SE5	0	0	0	0	0	0

À partir de cette matrice de précedence des modules, nous pouvons déduire, dans sa dernière ligne et colonne, les niveaux d'accessibilité suivant les deux sens de la direction de

désassemblage. Ces niveaux d'accessibilité sont caractérisés par le nombre de composants qui précèdent les composants à désassembler.

Finalement, la gamme de désassemblage résultante peut être représentée selon différentes façons (graphe des liaisons, graphe des adjacences,...). Nous proposons de la représenter sous forme d'arbre de désassemblage pour une vision plus complète des modules et des composants à désassembler et tenant compte de l'ordre de désassemblage.

III.2.3 Deuxième étape de l'algorithme de clustering existant

Dans cette étape, nous ferons une synthèse des métriques de mesure de la modularité et nous justifierons notre choix.

Quand on parle de modularité, on est en droit de se poser la question : à quel point un produit est-il modulaire ? Afin de quantifier la modularité, plusieurs mesures et métriques ont été développées. Les mesures développées servent alors à comparer différentes architectures d'un même produit.

Les mesures que nous présentons dans ce paragraphe sont classées selon une complexité croissante.

Blackenfelt (2000) utilise deux métriques pour mesurer la modularité d'un produit : le MI (Module Independence) (Newcomb et al., 1998) et l'ARP (Average Ratio of Potential) dont les expressions sont :

$$MI = \sum_{m=1}^n \frac{in_m}{tot} \quad (68)$$

$$ARP = \sum_{m=1}^n \frac{1}{n} \frac{in_m}{in_{pot}} \quad (69)$$

Avec :

- n : le nombre de modules ;
- m : l'indice d'un module ;
- in_m : la somme des interactions à l'intérieur du module m ;
- in_{pot} : la somme des interactions à l'intérieur d'un module pondéré par un facteur à fixer.

La différence principale entre ces deux métriques est que l'ARP est maximal pour une architecture composée de plusieurs petits modules.

Les métriques utilisées par Blackenfelt ne prennent pas en compte les interactions externes aux modules, de ce fait la pertinence des métriques est très limitée.

Une autre mesure de la modularité est le SMI (Holttä et *al.*, 2005) pour « Singular Value Modularity Index », la mise en place de cet indicateur nécessite une étape intermédiaire qui est la diagonalisation de la DSM du produit. On a alors :

$$DSM = U \sum_{DSM} V^T \quad (70)$$

Avec :

$$\sum_{DSM} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_N \end{pmatrix} \quad (71)$$

Où les σ_i sont les valeurs propres classés dans un ordre décroissant.

U et V^T représentant respectivement les vecteurs propres (normaux) et les vecteurs propres orthogonaux de la DSM .

N est le nombre de composants (nombre de lignes et de colonnes dans le DSM) du système.

On définit alors SMI par :

$$SMI \left(\sum_{DSM} \right) = 1 - \frac{1}{N\sigma_1} \sum_{i=1}^{N-1} \sigma_i (\sigma_i - \sigma_{i+1}) \quad (72)$$

La valeur de SMI est encadrée par 0 et 1, un SMI proche de 1 indique un fort degré de modularité, un SMI proche de 0 définit une architecture intégrale.

Se basant sur les valeurs propres de la DSM, cette métrique ne permet pas de faire le lien entre la taille des DSM, la taille des modules, leurs densités et la densité des interactions externes. L'interprétation de cette métrique nécessite de se référer à la DSM initiale.

Guo et Gershenson (2004) ont étudié huit métriques de modularisation et proposent une mesure de la modularité basée sur les intensités des interactions entre éléments dans un système. Cette mesure (métrique) se compose de deux parties : une partie positive qui donne un poids

aux interactions internes aux modules et une partie négative qui pénalise les interactions externes aux modules (Expression 73).

$$\frac{\sum_{k=1}^{M_m} \frac{\sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} R_{ij}}{(m_k - n_k + 1)^2} - \sum_{k=1}^{M_m} \frac{\sum_{i=n_k}^{m_k} \left(\sum_{j=1}^{n_k-1} R_{ij} + \sum_{j=m_k+1}^{N_c} R_{ij} \right)}{(m_k - n_k + 1)(N_c - m_k + n_k - 1)}}{M_m} \quad (73)$$

Avec :

- n_k : index du premier élément du $k^{\text{ème}}$ module ;
- m_k : index du dernier élément du $k^{\text{ème}}$ module ;
- M_m : nombre total des modules dans le produit ;
- N_c : nombre total de composants dans le produit ;
- R_{ij} : valeur de l'interaction entre les éléments i et j .

Cette mesure peut être utilisée avec toutes les DSM composants, elle présente l'avantage de prendre en compte les interactions extérieures aux modules. Cependant, Wietfield et *al.* (2002) utilisent une métrique proche de celle de Guo avec des pondérations plus pertinentes (le nombre d'interactions disponibles). Cette mesure de la modularité est l'indicateur MSI pour Module Strength Indicator. Le MSI s'applique à un module et se compose de deux parties (Equations 74 et 75).

$$MSI_i = \frac{\sum_{i=n_1}^{n_2} \sum_{j=n_1}^{n_2} DSM(i, j)}{(n_2 - n_1 + 1)^2 - (n_2 - n_1 + 1)} \quad (74)$$

$$MSI_e = \frac{\sum_{i=1}^{n_1-1} \sum_{j=n_1}^{n_2} (DSM(i, j) + DSM(j, i))}{2 \times ((n_1 - 1) \times (n_2 - n_1 + 1))} + \frac{\sum_{i=n_2+1}^N \sum_{j=n_1}^{n_2} (DSM(i, j) + DSM(j, i))}{2 \times ((N - n_2) \times (n_2 - n_1 + 1))} \quad (75)$$

$$MSI = MSI_i - MSI_e \quad (76)$$

Avec :

- $DSM(i, j)$: Valeur de l'interaction entre l'élément i et j dans la DSM ;
- n_1 : Index du premier élément du module ;
- n_2 : Index du dernier élément du module ;
- N : Le nombre total d'éléments ;

- MSI_i et MSI_e représentent respectivement la densité des interactions à l'intérieur et à l'extérieur d'un module.

L'indicateur MSI permet alors de mesurer le degré de modularité de chaque module en comparant les interactions à l'intérieur et à l'extérieur du module. Cet indicateur permet ainsi à l'architecte système d'évaluer la modularité de chaque module et de simuler plusieurs configurations. Ce dernier indicateur semble donc le plus complet pour évaluer la modularité d'une architecture (Ghassen, 2007). Une dernière remarque concernant cette métrique : les auteurs proposent les indices n_1 et n_2 et affirment que le nombre d'éléments est $n_2 - n_1 + 1$, pour que cette affirmation soit vraie il faut que n_1 soit l'indice du dernier élément dans le module précédent (Ghassen, 2007 ; Eric, 2008).

III.2.4 Troisième étape de l'algorithme de clustering existant

L'idée de cette étape est d'évaluer une fonction objective permettant d'identifier pour un produit donné les modules qui exigent le plus d'attention chez le concepteur. Nous ferons la synthèse des fonctions objectives qu'on peut évaluer dans cette étape afin de choisir notre fonction objective à évaluer.

L'algorithme proposé par Idicula (1995) pour cette étape est un algorithme de clustering hiérarchique du type Bottom-Up qui groupe « les tâches de conception dans des modules qui sont faiblement couplés entre eux, alors que chaque module est composé de tâches fortement couplées » (Ghassen, 2007).

Il y a trois points importants qui font la spécificité de cet algorithme :

- la fonction d'enchère qui permet à un module de tester le coût de l'incorporation d'un élément ;
- la « fonction objectif » qui exprime le coût de coordination de l'architecture testée et qui garantit l'obtention de l'architecture la plus adaptée en minimisant ce coût ;
- le recuit simulé qui permet à l'algorithme de sortir du chemin d'un optimum local pour trouver une architecture qui minimise le coût de coordination .

III.2.4.1 La fonction d'enchère

L'algorithme choisit aléatoirement et donc équiprobablement un élément. Une fois que l'élément est choisi, l'algorithme calcule une enchère de chaque module. Une telle enchère est une mesure de la façon dont les membres « forts » d'un module interagissent avec l'élément choisi. L'équation 77 montre la forme exacte de la fonction d'enchère.

$$Bid(cluster_k, element_i) = \sum_{j=1}^{size} \frac{\left((DSM(i, j) + DSM(j, i))^{pow_dep} \times CL_MAT(k, j) \right)}{cl_size(k)^{pow_bid}} \quad (77)$$

Avec :

- $Bid(cluster_k, element_i)$: l'enchère du module k envers l'élément i ;
- $DSM(i, j)$: valeur de l'interaction entre i et j dans la matrice ;
- $size$: taille de la DSM, c'est-à-dire le nombre d'éléments composant le domaine ;
- $cl_size(k)$: nombre d'éléments contenus dans le module k ;
- pow_dep : contrôle l'importance donnée aux interactions fortes relativement aux interactions faibles (une grande valeur augmente cette différence) ;
- pow_bid : contrôle la valeur de l'enchère relativement à la taille des modules (une grande valeur décourage la formation de grands modules et encourage la formation de modules de la même taille) ;
- $CL_MAT(k, j)$: variable binaire qui prend 1 quand l'élément j est dans le module k.

III.2.4.2 La fonction Coût

L'algorithme est construit autour d'une « fonction objectif » appelée le coût total de couplage. Cette fonction a pour but de faire correspondre notre vision de l'architecture d'un système à la formulation mathématique qu'elle utilise. Ainsi, le but est de transcrire les observations suivantes dans une formulation mathématique :

- Lorsqu'une interaction lie deux éléments appartenant à deux modules différents alors le coût de spécification/définition de cette interaction dépend de la taille des deux modules qu'elle rapproche. Cette remarque est à l'origine de l'évolution du coût de couplage proposé par la suite par (Ghassen, 2007 ; Eric, 2008) (voir équation 80).
- Pour obtenir l'expression finale du coût total de couplage : à l'image de la proposition d'Idicula (1995), Ghassen (2007) et par Eric (2008) ont construit deux fonctions complémentaires du coût de couplage : l'une pour caractériser les interactions internes à un module et l'autre pour caractériser les interactions externes à ce module.

Pour chaque interaction dans la matrice DSM, l'algorithme d'Idicula calcule un coût de couplage. Ensuite, la somme de tous les coûts de couplage donne le coût total de couplage.

Les équations 78 et 79 montrent les coûts de couplage pour une interaction.

Si les deux éléments i et j appartiennent au même module k (M_k), alors :

$$\text{Coût Couplage}(i, j) = (DSM(i, j) + DSM(j, i)) \times (\text{taille}(M_k))^2 \quad (78)$$

Sinon, si aucun module ne contient les éléments i et j , alors il existe deux modules M_i et M_j qui contiennent respectivement les éléments i et j . Le coût de l'interaction entre les éléments i et j s'écrit alors :

$$\text{Coût Couplage}(i, j) = (DSM(i, j) + DSM(j, i)) \times (\text{taille}(M_i) + \text{taille}(M_j))^2 \quad (79)$$

L'équation 80 résume l'expression du coût total de couplage.

$$\text{Coût Total Couplage} = \sum_i \sum_j \text{Coût Couplage}(i, j) \quad (80)$$

Les trois formulations précédentes des coûts de couplage peuvent être réécrites en adoptant comme référence les modules. On peut formuler alors une fonction de coût de couplage, soit interne, soit externe à un module k , M_k (Equations 81 et 82).

$$\text{Coût Couplage Interne}(M_k) = \sum_{i \in M_k} \sum_{j \in M_k, j \neq i} (DSM(i, j) + DSM(j, i)) \times (\text{taille}(M_k))^2 \quad (81)$$

$$\text{Coût Couplage Externe}(M_k) = \sum_{i \in M_k} \sum_{j \notin M_k, j \in M_n} \left[(DSM(i, j) + DSM(j, i)) \times (\text{taille}(DSM) + \text{taille}(M_k) + \text{taille}(M_n))^2 \right] \quad (82)$$

Alors le Coût Total de Couplage peut s'écrire sous la forme présenté dans l'équation 83.

$$\text{Coût Total Couplage} = \sum_{M_k}^{\text{Modules}} (\text{Coût Couplage Interne}(M_k) + \text{Coût Couplage Externe}(M_k)) \quad (83)$$

III.2.4.3 Le recuit simulé

Dans deux étapes de l'algorithme, la décision portant sur le choix de l'étape suivante n'est pas déterminée par les résultats obtenus mais par un processus aléatoire. Ce type de processus est connu sous le nom de « Simulated Annealing » qui en français est traduit en « Recuit Simulé ».

Le Recuit Simulé est un méta-algorithme probabiliste générique, il permet de réaliser une bonne approximation de l'optimum global d'une « fonction objectif » dans un grand espace de définition. Il a été développé parallèlement par Kirkpatrick et *al.* (1983), et par Cerny (1985).

La dénomination de cet algorithme est inspirée de la métallurgie, où on répète cycliquement la chauffe et le refroidissement d'un métal pour augmenter la taille de ses cristaux et réduire ses défauts.

Le principe de l'algorithme est le suivant : l'algorithme est initialisé avec une solution prise au hasard. À chaque itération, il modifie légèrement la solution pour essayer d'améliorer la « fonction objectif ». Des sauts aléatoires sont introduits pour permettre de sortir d'optima locaux, et donc accepter des solutions même si elles n'améliorent pas la « fonction objectif ».

Des conditions d'arrêt de l'algorithme sont définies pour limiter la durée d'exploration de l'espace de solutions.

Dans l'ultime évolution de l'algorithme d'Idicula, proposée par Thebeau (2001), le recuit simulé permet à l'algorithme de réaliser aléatoirement certaines opérations, ce qui a pour intérêt d'augmenter le nombre de solutions explorées par l'algorithme. Cependant, Ghassen (2007) a remarqué une erreur liée à l'utilisation du recuit simulé.

Cette erreur se situe dans la boucle qui permet d'accepter un coût de couplage supérieur, l'algorithme de référence enregistre alors la meilleure architecture référencée en cours obtenue jusqu'au meilleur coût actuel et qui peut être obtenue dans la boucle de stabilité en cours, mais cette architecture n'est pas sauvegardée dans une variable qui contient la meilleure architecture obtenue. Il s'ensuit à la sortie de cette boucle que l'algorithme peut accepter un coût qui améliore la variable sans vérifier si le meilleur coût actuel améliore le précédent. Par conséquent, l'algorithme peut ne pas prendre en compte la meilleure architecture parmi celles qu'il teste.

A l'issue de cette analyse de l'algorithme d'Idicula, nous optons pour une dénomination plus générique de la « fonction objectif » à savoir : « *coût total de couplage* ». En effet, cette fonction objective est indépendante du domaine étudié.

III.3 LIMITES DE L'ALGORITHME DE CLUSTERING EXISTANT

Il convient d'abord de définir les qualités attendues d'un bon algorithme de clustering. Cette question nous renvoie à une question encore délicate aujourd'hui dans la littérature s'intéressant à la conception modulaire : qu'est-ce qu'une architecture optimale ?

D'une façon générale, l'identification d'une architecture est un problème de décision multicritère. Il s'agit de tenir compte de la complexité de la conception liée à cette architecture. Le principe de système semi-décomposable formulé par Simon (1997) et les axiomes de la

conception axiomatique vont dans ce sens et tendent à privilégier la création de sous-systèmes qui soient le plus possible découplés. Il s'agit de limiter la diversité technique en standardisant les composants et faire des économies d'échelle en production. Il s'agit de s'assurer que le système sera facilement maintenable. En résumé, toutes les parties prenantes du cycle de vie du système vont spécifier des exigences sur son architecture.

La méthode de modularisation MFD essaie de prendre en compte plusieurs critères pour proposer un module mais cette méthode reste manuelle et subjective. Les algorithmes de clustering cherchent généralement à optimiser une "fonction objectif" qui ne prend en compte que les couplages d'un domaine du système (par exemple, fonctionnel ou organique) (Thebaud, 2001 ; Whitfield et *al.*, 2002 ; Yu et *al.*, 2003).

En ce qui concerne la suite de ce travail, nous définirons une *architecture optimale* comme étant la solution minimisant la "fonction objectif" d'un algorithme de clustering.

Éric (2008) caractérise les qualités attendues d'un algorithme de la façon suivante :

- S1 : Converger de façon répétée (reproductibilité) et rapide (temps de calcul) vers la solution optimale, du point de vue de la "fonction objectif" choisie ;
- S2 : Favoriser des modules denses (densité des modules) ;
- S3 : Pénaliser les modules de grande taille (nombre d'éléments dans un module) car ils peuvent être complexes à concevoir et nécessiter de nombreuses interfaces (intégration difficile) ;
- S4 : Être facile à programmer ou à modifier ;
- S5 : Nécessiter peu de paramètres à fixer ;
- S6 : Être compréhensible par son utilisateur. Les principes du calcul doivent être faciles à expliquer. Le résultat doit pouvoir s'interpréter facilement.

Les algorithmes proposés par Yu et *al.* (2003 ; 2007) ont l'avantage d'une meilleure convergence vers l'optimum que l'algorithme de Thebaud mais un temps de calcul qui peut être plus long (les auteurs ne fournissent pas ces données comparatives). Les programmes sources ne sont pas disponibles et difficilement compréhensibles par un bon ingénieur de conception.

L'algorithme développé par Thebaud satisfait les spécifications S1, S4 et S6.

Globalement l'algorithme de clustering existant ne satisfait pas les spécifications S2, S3, S5 et ne prend pas en compte la notion d'éléments intégrateurs afin de réduire la complexité des modules.

Nous allons proposer des modifications pour améliorer cet algorithme sur ces critères.

III.4 NOUVEL ALGORITHME DE CLUSTERING

Dans ce qui suit, nous allons présenter le nouvel algorithme de clustering et son fonctionnement.

L'algorithme utilisé se base sur l'évaluation des couplages entre éléments d'un même domaine pour générer d'une manière systématique des groupements en modules de ces mêmes éléments en optimisant une « fonction objectif ». Il ne gère que les valeurs positives des couplages.

L'algorithme d'Idicula permet de produire d'une manière stochastique des modules à partir d'une DSM. Cependant, nous avons constaté que cet algorithme est lourd à utiliser et est fonction des paramètres de contrôle qui peuvent être difficile à manipuler pour permettre de trouver une combinaison optimale.

Dans notre démarche d'amélioration de l'algorithme de référence, nous avons intégré des notions fondamentales de l'architecture modulaire afin d'orienter le fonctionnement de l'algorithme.

Les modifications que nous apportons concernent quatre points spécifiques :

- La DSM ;
- Le MSI ;
- La « Fonction objectif » ;
- L'identification des éléments intégrateurs.

L'organigramme du nouvel algorithme de clustering que nous proposons est donné à la figure 33.

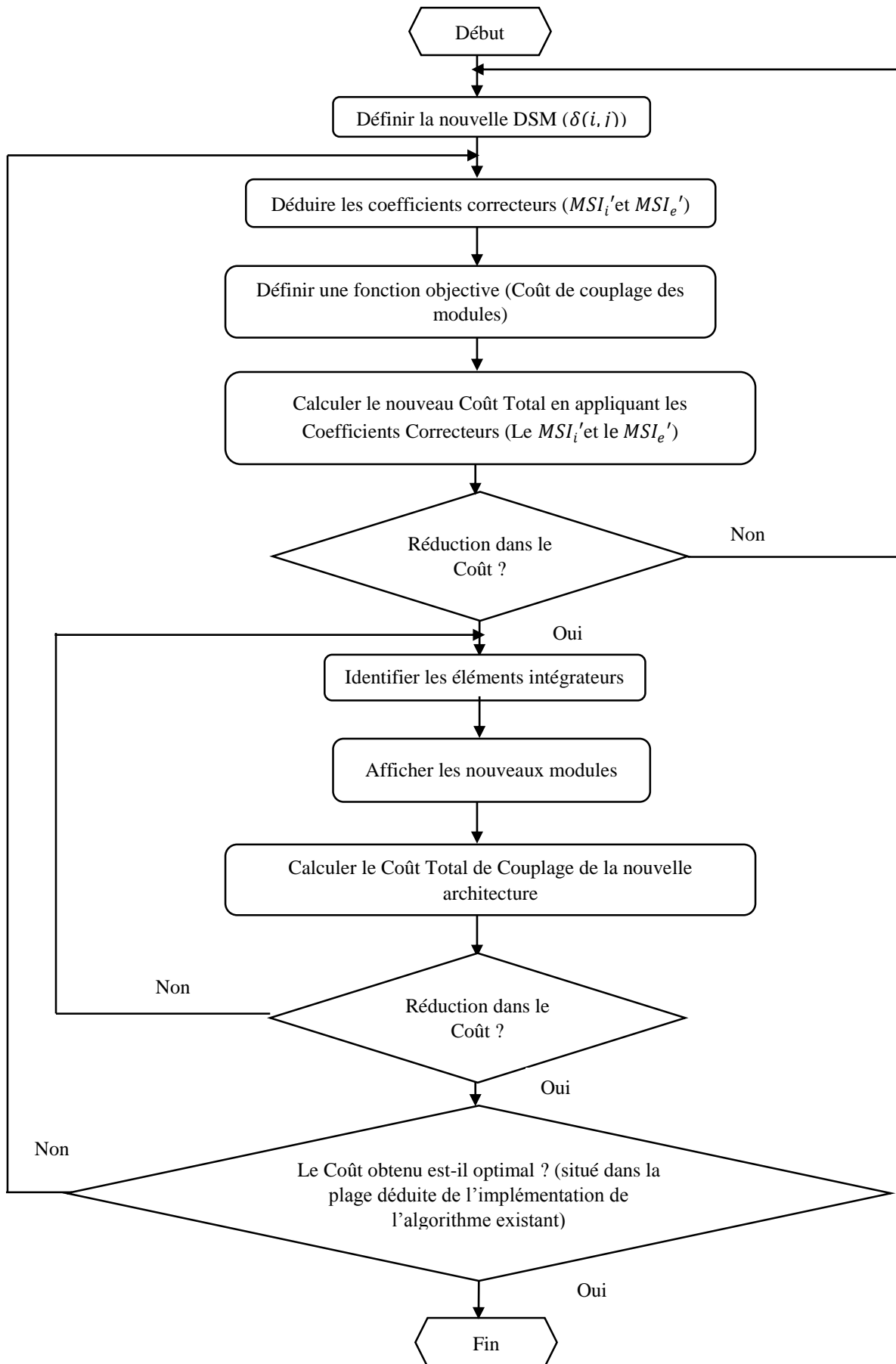


Figure 33: Organigramme du nouvel algorithme de clustering

III.4.1 Matrice structurelle de conception modifiée

La première modification de l'algorithme existant concerne la matrice structurelle de conception. En se servant des modifications apportées par Ghassen (2007) et Eric (2008), les transformations des éléments de la DSM de base que nous proposons sont données dans le tableau 31.

Tableau 31: Modifications de la matrice structurelle de conception

DSM(i, j)	Valeur de couplage entre les éléments C_i et C_j . Notons que pour $i = j$, DSM(i, j) = 0 .
$\delta(i, j)$	Prend la valeur 1 quand $DSM(i, j) \neq 0$ et la valeur 0 quand $DSM(i, j) = 0$.

La matrice structurelle de conception de l'algorithme de base tient compte de l'intensité des interactions entre les éléments (**DSM (i, j)**), la nouvelle matrice structurelle de conception que nous proposons tient compte uniquement de l'existence ou non des interactions. La transformation ainsi opérée est donnée comme suit :

$$\begin{pmatrix} 0 & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & 0 \end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix} 0 & \dots & x \\ \vdots & \ddots & \vdots \\ x & \dots & 0 \end{pmatrix}$$

Dans la nouvelle matrice structurelle de conception d'ordre n , x représente une valeur binaire qui peut prendre les valeurs **0** ou **1**, en fonction de la valeur de couplage entre les éléments C_i et C_j que génère l'algorithme de base ($1 \leq i \leq n$ et $1 \leq j \leq n$).

III.4.2 Coefficients correcteurs du coût de couplage des modules

La deuxième modification concerne le Module Strength Indicator (MSI). En appliquant l'algorithme de base développée par Idicula (1995), on observe une formation de modules de grandes tailles (dont moins dense à l'intérieur), plus contraignants pour la maintenance. Nous proposons à cet effet un procédé pour éviter la formation de modules de "trop" grande taille (former des modules plus denses à l'intérieur). Pour cela, nous corrigeons les coûts de couplage du précédent algorithme par deux coefficients correcteurs, que nous appelons **MSIi'** et **MSIe'**.

Pour le calcul de ces coefficients, on met l'accent sur deux caractéristiques d'une architecture modulaire :

- Un module est le groupement d'éléments interagissant fortement entre eux ;

- Les éléments appartenant à un module interagissent faiblement avec les éléments appartenant à d'autres modules.

A la lumière de ces deux remarques, nous nous sommes inspiré de la métrique de modularité proposée par Whitfield et *al.* (2002).

La partie MSI_i' (pour densité des interactions à l'intérieur) prend en compte les interactions internes à un module et donne de ce fait une mesure de la cohésion et de la robustesse (Strength) interne du module.

La partie MSI_e' (pour densité des interactions à l'extérieur) prend en compte les interactions externes qui lient les éléments appartenant au module aux autres éléments externes au module, ainsi MSI_e' mesure la force des interactions externe au module.

Il apparaît à travers les deux définitions que MSI_i' et MSI_e' sont deux indicateurs antagonistes.

C'est pourquoi l'indicateur **MSI'** (Module Strength Indicator corrigé du module considéré) s'écrit comme la différence des deux.

Dans notre travail, nous n'allons pas utiliser directement l'indicateur MSI' , mais une forme dérivée des deux indicateurs MSI_i' et MSI_e' . Alors que ces deux indicateurs originaux tiennent compte de l'intensité des interactions, les indicateurs que nous proposons tiennent compte uniquement de l'existence ou non des interactions. On obtient alors les indicateurs MSI_i' et MSI_e' présentés en équations 84 et 85.

$$MSI_i'(M_k) = \frac{\sum_{i=n_1}^{n_2} \sum_{j=n_1, j \neq i}^{n_2} \delta(i, j)}{(n_2 - n_1 + 1)^2 - (n_2 - n_1 + 1)} \quad (84)$$

$$MSI_e'(M_k) = \frac{\sum_{i=1}^{n_1-1} \sum_{j=n_1}^{n_2} (\delta(i, j) + \delta(j, i))}{2 \times ((n_1 - 1) \times (n_2 - n_1 + 1))} + \frac{\sum_{i=n_2+1}^{taille(DSM)} \sum_{j=n_1}^{n_2} (\delta(i, j) + \delta(j, i))}{2 \times ((taille(DSM) - n_2) \times (n_2 - n_1 + 1))} \quad (85)$$

Avec :

- $Taille(DSM)$: fonction qui renvoie la taille d'un module ou de la DSM, i.e. nombre d'éléments dans une sous-matrice de la DSM ;
- n_1 : index du premier élément du module M_k ;
- n_2 : index du dernier élément du module M_k ($n_2 \geq n_1$). Si $n_2 = n_1$, le module ne contient en fait qu'un seul élément.

Notons que le dénominateur du MSI_i' correspond au nombre total d'interactions possibles du module considéré, en dehors de la diagonale et que nous considérons uniquement des DSM symétriques.

L'indicateur MSI a été développé pour des DSM temporelles et plus précisément pour des *DSM paramètres* tel que celles développées dans l'algorithme de clustering de base qui s'inspirent des paramètres de conception. Nous proposons alors un indicateur MSI' développé pour des *DSM numériques* qui prennent leurs valeurs entre 0 et 1. On obtient alors des indicateurs MSI_i' et MSI_e' ayant leurs valeurs entre 0 et 1. Du fait que les indicateurs MSI_i' et MSI_e' prennent en compte des interactions valant 1 ou 0, MSI_i' et MSI_e' ont une valeur bornée par 0 et 1.

Avant d'expliquer l'utilisation de ces indicateurs dans la fonction de coût, nous allons étudier le comportement de ces deux indicateurs.

L'indicateur MSI_i' est maximal et vaut 1, lorsque le module est totalement dense, c'est-à-dire tous les éléments qui le composent interagissent. L'indicateur est nul, lorsque les éléments qui composent le module n'interagissent pas du tout entre eux (ce qui remet en question l'utilisation du terme module pour qualifier ce groupement d'éléments).

L'indicateur MSI_e' est nul lorsque le module concerné n'interagit avec aucun élément externe (il est complètement découplé). Il est maximal lorsque chacun des éléments qui le composent interagit avec tous les éléments externes à ce module. Cette dernière configuration est également contraire à l'application du terme module à un groupement d'éléments.

Ainsi, un groupement d'éléments tend à devenir un module parfait quand son MSI_i' tend vers 1 et son MSI_e' tend vers 0.

III.4.3 La fonction Coût

La troisième modification concerne la "fonction objectif" qui est l'élément déterminant de l'algorithme de clustering. Notons que sa formulation correspond à une modélisation de ce que l'expert qui utilise l'algorithme peut considérer comme une "architecture optimale".

Si nous reprenons la « fonction objectif » choisie plus haut (équation 83), nous remarquons que nous pénalisons plus fortement les interactions externes que les interactions internes. Cette configuration peut aboutir à la création de modules relativement grands, même si la taille du module est pénalisée. Ainsi, pour ajuster et corriger la « fonction objectif » nous allons nous inspirer des travaux de (Ghassen, 2007 ; Eric ,2008). Ces travaux ne donnent pas

des contraintes sur les coefficients correcteurs et n'intègrent pas ces contraintes dans la formulation de la fonction objectif. Nous formulons donc une fonction objectif définie par l'équation 86, qui tient compte des contraintes sur les coefficients correcteurs du coût de couplage des modules.

$$\begin{aligned}
 \text{minimiser} \quad & \text{Coût Total Couplage} = \sum_i \sum_j \text{Coût Total Couplage}(i, j) \\
 \text{tel que :} \quad & MSI'_i < MSI_i \\
 & MSI'_e < MSI_e
 \end{aligned} \tag{86}$$

Les corrections que nous apportons avec cette nouvelle fonction objectif concernent le coût de couplage interne et le coût de couplage externe, pour un module considéré.

Ainsi, les corrections apportées sont données par les équations 87 et 88.

$$\text{Coût Couplage Interne Corrigé}(M_k) = \text{Coût Couplage Interne}(M_k) / MSI'_i(M_k) \tag{87}$$

$$\text{Coût Couplage Externe Corrigé}(M_k) = \text{Coût Couplage Externe}(M_k) \times MSI'_e(M_k) \tag{88}$$

Ces deux corrections sont interprétées de la manière suivante :

- en divisant le coût de couplage interne par MSI'_i , nous favorisons la création de modules plutôt denses que grands ;
- en multipliant le coût de couplage externe par MSI'_e , nous favorisons la minimisation des interactions externes aux modules.

Notons la double logique de la nouvelle "fonction objectif" :

- plus MSI'_i augmente, plus le module est dense, plus le coût baisse ;
- plus MSI'_e augmente, plus il y a d'interactions externes au module, plus le coût augmente.

Le Coût Total Corrigé s'écrit alors :

$$\text{Coût Total Couplage Corrigé} = \sum_{M_k}^{\text{Modules}} \left(\text{Coût Couplage Interne Corrigé}(M_k) + \text{Coût Couplage Externe Corrigé}(M_k) \right) \tag{89}$$

Afin de montrer la pertinence de cette dernière « fonction objectif », nous allons comparer dans la partie résultats de ce chapitre notre algorithme et l'algorithme référence.

III.4.4 Identification des éléments intégrateurs

Une dernière modification concerne le traitement spécifique que nous réservons aux éléments intégrateurs. Ghassen (2007) définit un élément intégrateur comme un élément qui n'appartient à aucun module et qui interagit avec des éléments appartenant à plusieurs modules. Cette définition pose cependant un problème dans la mesure où l'architecture de base ne comporte pas d'éléments n'appartenant à aucun module. En effet, tout module contient au moins un composant, donc un composant unique pourrait définir son propre module. Dans l'optique de construire des modules de petites tailles, on peut considérer un élément appartenant déjà à un module comme un élément intégrateur dans la mesure où il interagit fortement avec un autre module. Nous définissons donc un élément intégrateur comme un élément appartenant à un module et qui interagit avec des éléments appartenant à un ou à plusieurs modules.

Les algorithmes de clustering ont pour objectif de créer des modules et donc tentent d'insérer tout élément à l'intérieur d'un module. Les fonctions de coût (celle de Thebaud (2001) ou celle que nous proposons) ne peuvent pas garantir que les éléments intégrateurs seront placés à l'extérieur des autres modules, ce qui au final, n'est pas conforme à la vision de l'architecte et perturbe la convergence de l'algorithme.

C'est pourquoi nous proposons de calculer un indice de couplage (*IC*) pour identifier "automatiquement" les éléments intégrateurs, définis comme ceux qui interagissent fortement en nombre avec les éléments appartenant à d'autres modules (Equation 90). Nous offrons aussi la possibilité à l'utilisateur de classer manuellement un élément qu'il considère comme étant intégrateur.

$$IC(\text{élément}_j) = \frac{\sum_{i=1, i \neq j}^{\text{taille}(DSM)} \delta(i, j)}{\text{taille}(DSM) - 1} \quad (90)$$

L'IC mesure le taux de couplage entre un élément et le reste du système. Pour tenir compte du jugement des architectes, Ghassen (2007) fixe l'IC à 60%. Au-delà de ce seuil, les éléments sont considérés comme étant intégrateurs. Eric (2008) par contre, offre la latitude à l'utilisateur de définir le seuil de l'IC.

III.5 CONCLUSION

Cette partie a été consacrée à la proposition d'une méthode d'algorithme de clustering pour une bonne modularisation des systèmes. Il se trouve que l'algorithme de clustering déployé dans la littérature présente beaucoup de limites, dans la mesure où il ne considère que les DSM

paramètres (dont on aurait six paramètres à varier, rendant plus lourd l'implémentation de l'algorithme), il génère des modules complexes (de grandes tailles et engendrant des coûts de couplages élevés) et ne prend pas en compte les éléments intégrateurs, permettant la création des modules plus denses. Il est donc important de réduire ces coûts de couplage pour une conception et une maintenance plus aisées. Des corrections de l'algorithme existant ont été apportées en proposant un nouvel algorithme de clustering. Nous validerons cet algorithme par une application sur le torréfacteur de soja au Chapitre 4.

CHAPITRE IV : VALIDATION DE L'ALGORITHME PROPOSE

IV.1 INTRODUCTION

Dans ce chapitre, nous implémentons d'abord l'algorithme de clustering existant, ensuite nous implémentons le nouvel algorithme proposé et comparons les résultats avec ceux générés par le précédent algorithme.

Dans l'implémentation de l'algorithme de clustering existant, la modélisation du produit est fondamentale pour l'identification d'une architecture modulaire. En effet, cette modélisation conduit à aboutir à une matrice de relations globale entre tous les composants d'un produit. Grâce à cette matrice de relations qui est fortement corrélée à la première étape de l'algorithme de clustering existant qui consiste à déployer une chronologie de regroupement de tous les composants en modules, on aboutit ainsi à une architecture modulaire. L'architecture modulaire étant ainsi obtenue et une bonne topologie des modules étant définie, au moyen d'une des métriques proposées dans la littérature qui fait l'objet de la deuxième étape de l'algorithme de clustering existant, la modularité sera mesurée afin d'identifier le module qui assure une plus forte cohésion interne et dont la maintenance est plus facile. A l'issue de cette mesure de la modularité, la troisième étape de l'algorithme de clustering existant va consister à évaluer les coûts de couplage entre les modules du produit afin d'identifier les modules qui suscitent le plus d'attention chez les concepteurs. Enfin, les résultats de l'implémentation du nouvel algorithme, comparés à ceux de la littérature permettront de le valider.

IV.2 RESULTATS DE L'ALGORITHME DE CLUSTERING EXISTANT

Cette partie illustre la mise en application des fonctionnalités de l'algorithme existant présentée au chapitre 3 qui consiste à définir une bonne topologie modulaire, à évaluer le degré de modularité (le MSI) et à évaluer le coût de couplage des modules, sur le même produit du chapitre 2 : le *torréfacteur de soja* (la solution de conception issue du chapitre 2 est donnée en Annexe 10).

IV.2.1 Modélisation du produit

En exécutant la formule donnée par l'équation (64) qui consiste à générer toutes les cellules de la matrice de relations des composants du système et fonction des paramètres de désassemblage sélectionnés au chapitre 2, on obtient la matrice de relation donnée par le tableau 32.

Tableau 32:Matrice de relations du torréfacteur de soja

Composants	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	23										23					23		
2	23	0							14										
3			0	14					18										
4			14	0	21														
5				21	0	19	19												
6					19	0		18											
7					19		0				18								
8						18		0									21		
9		14	18						0										
10										0	21						20		
11							18			21	0								
12	23											0			21				
13													0	21				21	
14													21	0	17				
15												21		17	0				
16																0	23		
17	23							21		20						23	0		21
18													21					0	
19																	21		0
TOTAL	69	37	32	35	59	37	37	39	32	41	39	44	42	38	38	23	108	21	21

En observant cette matrice, on constate que le nombre de connexions est **343** ,qui est bien en accord avec la formule établie par

Lambert et *al.* (2005) qui stipule que le nombre de connections maximales pour notre cas est : $19(19 - 1) = 343$ connections possibles. On est donc à la limite du nombre de connexions des composants de notre système.

IV.2.2 Première étape de l'algorithme de clustering existant et topologie des modules

IV.2.2.1 Première étape de l'algorithme de clustering existant

L'application de la formule de Ericsson et Erixon (1999) permettant de calculer le nombre de modules possibles nous a permis de constater que le nombre de modules que l'on peut former est **2, 3** ou **4** (voir équation 91).

$$0,5\sqrt{19} \leq NM \leq \sqrt{19} \quad (91)$$

a) Cas du torréfacteur à quatre modules

Les composants du torréfacteur de soja sont regroupés en modules, en appliquant l'algorithme de regroupement des composants en modules proposé plus haut. L'addition des entrées de chaque colonne de la matrice des relations (obtenue précédemment), a permis de constater que les composants 1, 5, 12 et 17 ont des intensités de liaisons les plus élevées. Nous les assignons à un module chacun. Ensuite nous assignons le reste des composants à un module, de telle façon à maximiser les interactions entre les composants d'un même module et en minimisant les interactions entre les composants de modules différents. Ainsi, au cluster(module) contenant le composant 1 est assigné le composant 2 ; au cluster contenant le composant 5 est assigné le composant 4 et au cluster contenant le composant 17 est assigné le composant 16. En transformant la matrice des relations en supprimant les composants assignés à un module, on obtient la nouvelle matrice donnée par le tableau **33**.

Tableau 33:Première transformation de la matrice de relations

Composants	1	3	5	6	7	8	9	10	11	12	13	14	15	17	18	19
1	0									23				23		
3		0					18									
5			0	19	19											
6			19	0		18										
7			19		0				18							
8				18		0								21		
9		18					0									
10								0	21					20		
11					18			21	0							
12	23									0			21			
13											0	21			21	
14											21	0	17			
15										21		17	0			
17	23					21		20						0		21
18											21				0	
19														21		0

A l'issue de cette première transformation de la matrice de relations, on constate que les composants 6 et 7 sont assignés au cluster contenant initialement le composant 5. En transformant la nouvelle matrice (précédemment obtenue), on obtient la matrice du tableau 34.

Tableau 34:Deuxième transformation de la matrice de relations

Composants	1	3	5	8	9	10	11	12	13	14	15	17	18	19
1	0							23				23		
3		0			18									
5			0											
8				0								21		
9		18			0									
10						0	21					20		
11						21	0							
12	23							0			21			
13									0	21			21	
14									21	0	17			
15								21		17	0			
17	23			21		20						0		21
18									21				0	
19												21		0

A l'issue de cette deuxième transformation de la matrice de relations, on constate que les composants 8 et 19 sont assignés au cluster contenant initialement le composant 17. En transformant la matrice du tableau 34, on obtient la matrice du tableau 35.

Tableau 35: Troisième transformation de la matrice de relations

Composants	1	3	5	9	10	11	12	13	14	15	17	18
1	0						23				23	
3		0		18								
5			0									
9		18		0								
10					0	21					20	
11					21	0						
12	23						0			21		
13								0	21			21
14								21	0	17		
15							21		17	0		
17	23				20						0	
18								21				0

A l'issue de cette troisième transformation de la matrice de relations, on constate que les composants 10 est assigné au cluster contenant initialement le composant 17. Ensuite en appliquant la même démarche, nous assignons le reste des composants à un module, de telle façon à maximiser les interactions entre les composants d'un même module et en minimisant les interactions entre les composants de modules différents.

On obtient alors la décomposition modulaire du torrificateur de soja à la figure 34.

Donc le module M_1 est formé des composants 8 ;10 ;16 ;17 et 19. Le module M_2 est formé des composants 11 ;12 ;13 ;14 ;15 et 18. Le module M_3 est formé des composants 3 ;4 ;5 ;6 et 7. Le module M_4 est formé des composants 1 ;2 et 9.

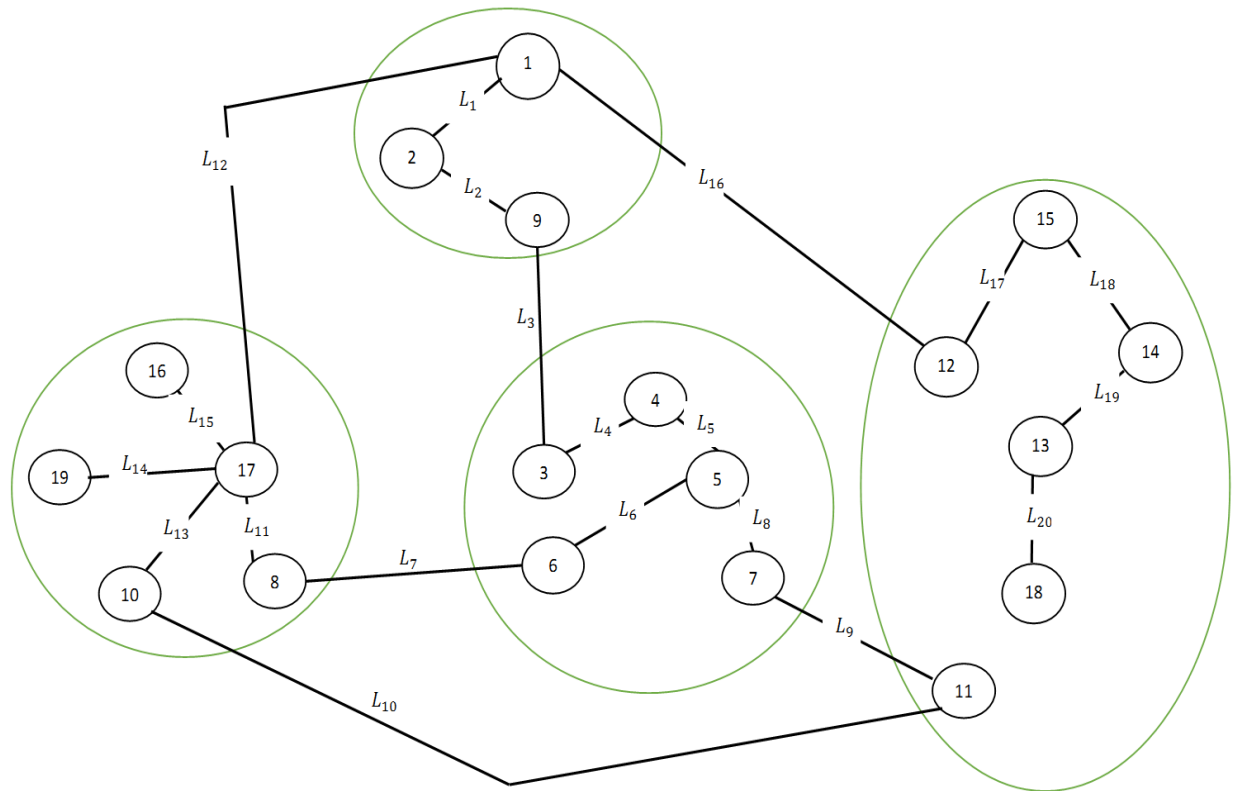


Figure 34: Regroupement en quatre modules du torrificateur de soja

b) Cas du torrificateur à trois modules

En appliquant l’algorithme de regroupement des composants en modules, l’addition des entrées de chaque colonne de la matrice des relations, permet de constater que les composants 1, 5 et 17 ont des intensités de liaisons les plus élevées. Nous les assignons à un module chacun.

En transformant la matrice des relations (en suivant le même raisonnement pour le torrificateur à quatre modules) en supprimant les composants assignés à un module, de telle façon à maximiser les interactions entre les composants d’un même module et en minimisant les interactions entre les composants de modules différents, on obtient la décomposition modulaire donnée par la figure 35.

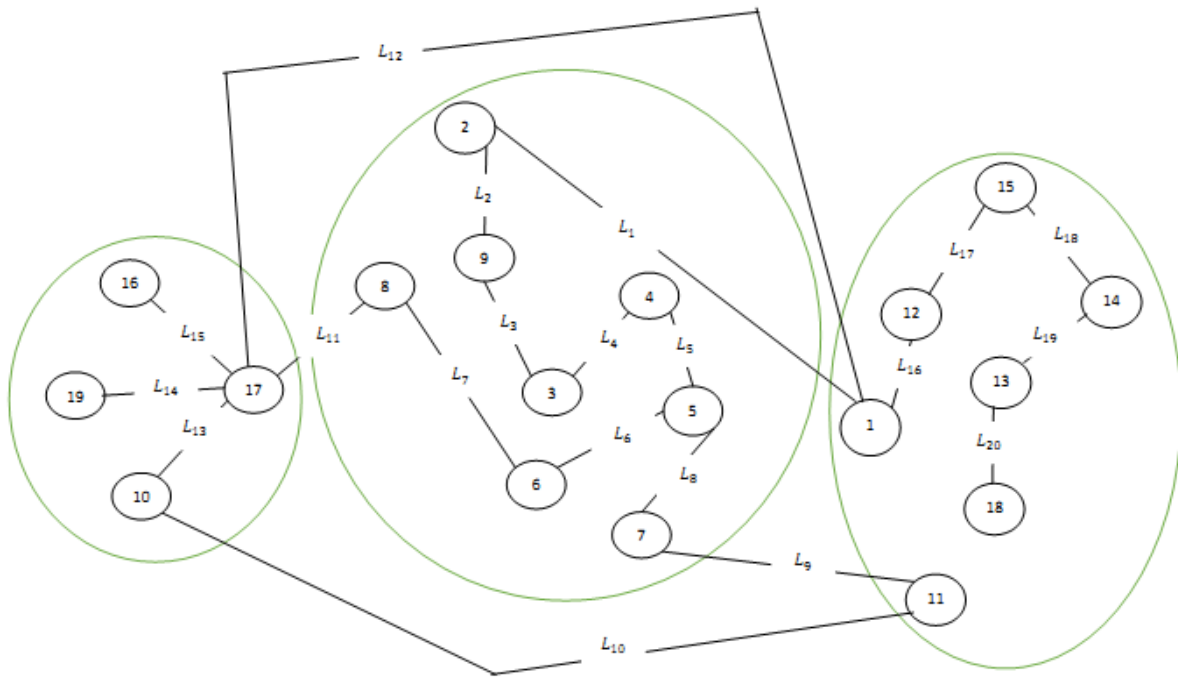


Figure 35: Regroupement en trois modules du torrificateur de soja

c) Cas du torrificateur à deux modules

En appliquant l'algorithme de regroupement des composants en modules, l'addition des entrées de chaque colonne de la matrice des relations, permet de constater que les composants 1 et 17 ont des intensités de liaisons les plus élevées. Nous les assignons à un module chacun.

En transformant la matrice des relations (en suivant le même raisonnement pour le torrificateur à quatre modules) en supprimant les composants assignés à un module, de telle façon à maximiser les interactions entre les composants d'un même module et en minimisant les interactions entre les composants de modules différents, on obtient la décomposition modulaire donnée par la figure 36.

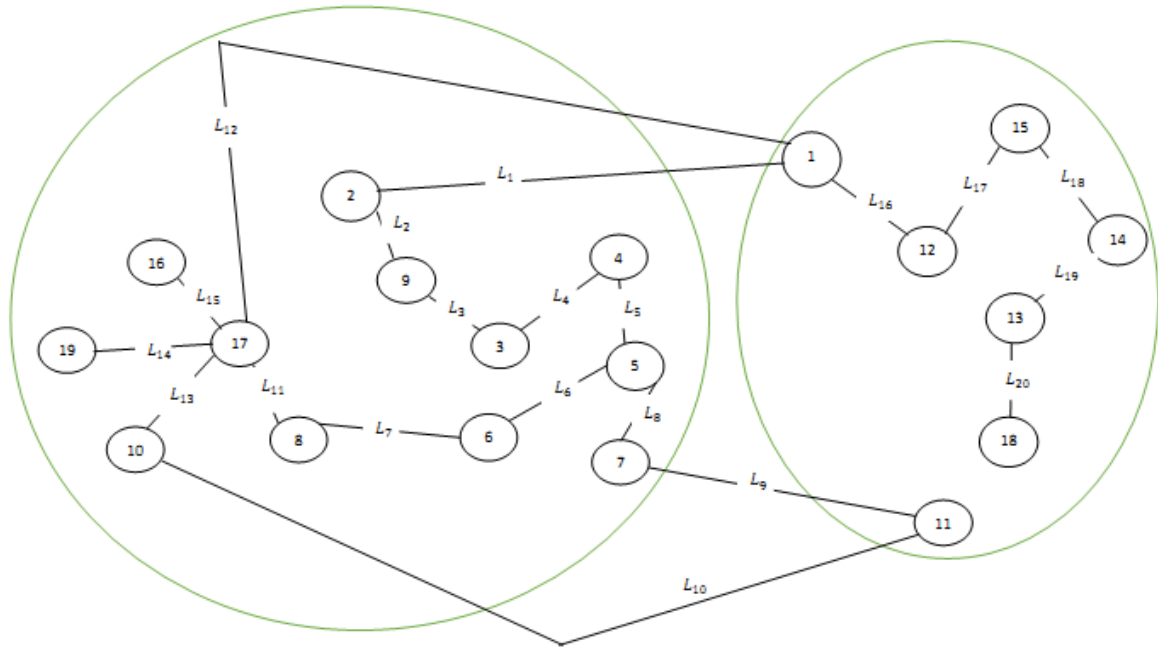


Figure 36:Regroupement en deux modules du torrificateur de soja

IV.2.2.2 Choix d'une bonne topologie des modules

Nous allons générer la séquence (gamme) de désassemblage des modules afin de déduire les couplages possibles entre les différents modules du torrificateur de soja.

IV.2.2.2.1 Matrice de précéance des composants

Nous illustrons via la matrice de précéance pour le torrificateur de soja, en fonction de son nombre de modules. Afin de construire cette matrice, l'ensemble des composants qui doivent être désassemblés avant chaque composant du torrificateur de soja est déterminé. L'ensemble est déterminé à partir du graphe des liaisons de la structure pour permettre son désassemblage.

L'application de la relation 67 définie dans le chapitre précédent, permet d'avoir respectivement la matrice de précéance du torrificateur de soja à quatre modules, à trois modules et à deux modules, par les figures 37, 38 et 39.

Les éléments (entrées) de la diagonale principale de la matrice de précéance sont égaux à 0.

		M_1					M_2						M_3					M_4		
		8	10	16	17	19	11	12	13	14	15	18	3	4	5	6	7	1	2	9
M_1	8	0	0	x	x	x	x	x	x	x	x	x	0	x	x	x	x	0	x	x
	10	1	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	16	x	x	0	1	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	17	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	19	x	x	x	1	0	x	x	0	x	x	x	x	x	x	x	x	x	x	x
M_2	11	x	1	1	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	12	x	x	x	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x
	13	x	x	x	x	1	x	x	0	x	x	x	x	x	x	x	x	1	x	x
	14	x	x	x	x	x	0	x	x	0	x	x	x	x	x	x	x	x	x	x
	15	x	x	x	x	x	x	1	x	x	0	0	x	x	x	x	x	1	x	x
	18	x	x	x	x	x	x	x	x	x	1	0	x	x	x	x	x	x	x	x
M_3	3	1	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	x	x
	4	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x
	5	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	x	x	x	x
	6	x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	x	x	x	x
	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	x
M_4	1	0	x	x	x	x	x	x	0	x	0	x	x	x	x	x	0	0	x	0
	2	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	0	x
	9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	0

Figure 37: Matrice de précédence des composants du torréfacteur de soja à quatre modules.

		M_1								M_2			M_3							
		2	3	4	5	6	7	8	9	10	16	17	19	1	11	12	13	14	15	18
M_1	2	0	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	3	x	0	x	x	x	0	1	x	x	x	x	x	x	x	x	x	x	x	x
	4	0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	5	x	x	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	6	x	1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	7	x	x	x	x	x	0	x	x	x	x	x	x	1	x	x	x	x	x	x
	8	x	0	x	x	x	x	0	x	0	x	x	x	0	x	x	x	x	x	x
	9	x	x	x	x	x	0	x	0	x	x	x	x	1	x	x	x	x	x	x
	10	x	x	x	x	x	x	1	x	0	x	x	x	0	0	x	x	x	x	x
M_2	16	x	x	x	x	x	x	x	x	0	1	x	x	0	x	x	x	x	x	
	17	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	
	19	x	x	x	x	x	x	x	x	x	1	0	x	x	x	0	x	x	x	
M_3	1	x	x	x	x	x	0	0	0	x	x	x	x	0	x	x	0	x	0	x
	11	x	x	x	x	x	x	x	x	1	1	x	x	x	0	x	x	x	x	x
	12	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	
	13	x	x	x	x	x	x	x	x	x	x	1	1	x	x	0	x	x	x	
	14	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	
	15	x	x	x	x	x	x	x	x	x	x	x	1	x	1	x	x	0	0	
	18	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	

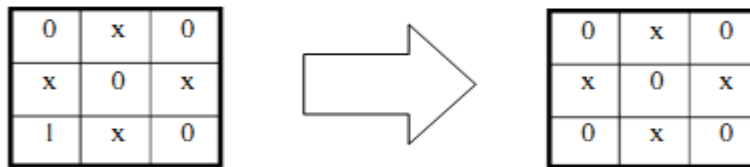
Figure 38:Matrice de précédence des composants du torréfacteur de soja à trois modules.

		M_1											M_2							
		2	3	4	5	6	7	8	9	10	16	17	19	1	11	12	13	14	15	18
M_1	2	0	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	3	x	0	x	x	x	0	1	x	x	x	x	x	x	x	x	x	x	x	x
	4	0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	5	x	x	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	6	x	1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	7	x	x	x	x	x	0	x	x	x	x	x	x	1	x	x	x	x	x	x
	8	x	0	x	x	x	x	0	x	0	x	x	x	0	x	x	x	x	x	x
	9	x	x	x	x	x	0	x	0	x	x	x	x	1	x	x	x	x	x	x
	10	x	x	x	x	x	x	1	x	0	x	x	x	0	0	x	x	x	x	x
	16	x	x	x	x	x	x	x	x	x	0	1	x	x	0	x	x	x	x	x
	17	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x
	19	x	x	x	x	x	x	x	x	x	x	1	0	x	x	x	0	x	x	x
M_2	1	x	x	x	x	x	0	0	0	x	x	x	x	0	x	x	0	x	0	x
	11	x	x	x	x	x	x	x	x	1	1	x	x	x	0	x	x	x	x	x
	12	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0
	13	x	x	x	x	x	x	x	x	x	x	x	1	1	x	x	0	x	x	x
	14	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x
	15	x	x	x	x	x	x	x	x	x	x	x	x	1	x	1	x	x	0	0
	18	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0

Figure 39:Matrice de précédence des composants du torrificateur de soja à deux modules.

IV.2.2.2.2 Changement de direction de désassemblage des modules

On exploite la matrice de précédence des composants pour en tirer les niveaux de désassemblage du torrificateur à quatre modules, à trois modules et à deux modules (respectivement aux figures 40, 41 et 42) afin de construire l'arbre de désassemblage. Du fait que des composants appartiennent au même module, on affecte la valeur 0 au lieu de 1. Par exemple pour le module 4 (dans le cas du torrificateur à quatre modules), on obtient :



		M_1					M_2						M_3					M_4		
		8	10	16	17	19	11	12	13	14	15	18	3	4	5	6	7	1	2	9
M_1	8	0	0	x	x	x	x	x	x	x	x	x	0	x	x	x	x	0	x	x
	10	0	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	16	x	x	0	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	17	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	19	x	x	x	0	0	x	x	0	x	x	x	x	x	x	x	x	x	x	x
M_2	11	x	1	1	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	12	x	x	x	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x
	13	x	x	x	x	1	x	x	0	x	x	x	x	x	x	x	x	1	x	x
	14	x	x	x	x	x	0	x	x	0	x	x	x	x	x	x	x	x	x	x
	15	x	x	x	x	x	x	0	x	x	0	0	x	x	x	x	x	1	x	x
	18	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x
M_3	3	1	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	x	x
	4	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x
	5	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x
	6	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x
	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	x
M_4	1	0	x	x	x	x	x	x	0	x	0	x	x	x	x	x	0	0	x	0
	2	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	0	x
	9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	0

Figure 40: Matrice de précedence modifiée du torréfacteur de soja à quatre modules.

		M_1								M_2			M_3							
		2	3	4	5	6	7	8	9	10	16	17	19	1	11	12	13	14	15	18
M_1	2	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	3	x	0	x	x	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x
	4	0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	5	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	6	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	7	x	x	x	x	x	0	x	x	x	x	x	x	1	x	x	x	x	x	x
	8	x	0	x	x	x	x	0	x	0	x	x	x	0	x	x	x	x	x	x
	9	x	x	x	x	x	0	x	0	x	x	x	x	1	x	x	x	x	x	x
	10	x	x	x	x	x	x	1	x	0	x	x	x	0	0	x	x	x	x	x
M_2	16	x	x	x	x	x	x	x	x	0	0	x	x	0	x	x	x	x	x	
	17	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	
	19	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x	x	x	
M_3	1	x	x	x	x	x	0	0	0	x	x	x	x	0	x	x	0	x	0	x
	11	x	x	x	x	x	x	x	x	1	1	x	x	x	0	x	x	x	x	x
	12	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	
	13	x	x	x	x	x	x	x	x	x	x	1	0	x	x	0	x	x	x	
	14	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	
	15	x	x	x	x	x	x	x	x	x	x	x	0	x	0	x	x	0	0	
	18	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

Figure 41: Matrice de précedence modifiée du torréfacteur de soja à trois modules.

		M_1											M_2							
		2	3	4	5	6	7	8	9	10	16	17	19	1	11	12	13	14	15	18
M_1	2	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	3	x	0	x	x	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x
	4	0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	5	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	6	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	7	x	x	x	x	x	0	x	x	x	x	x	x	1	x	x	x	x	x	x
	8	x	0	x	x	x	x	0	x	0	x	x	x	0	x	x	x	x	x	x
	9	x	x	x	x	x	0	x	0	x	x	x	x	1	x	x	x	x	x	x
	10	x	x	x	x	x	x	0	x	0	x	x	x	0	0	x	x	x	x	x
	16	x	x	x	x	x	x	x	x	x	0	0	x	x	0	x	x	x	x	x
	17	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x
19	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x	x	x	
M_2	1	x	x	x	x	x	0	0	0	x	x	x	x	0	x	x	0	x	0	x
	11	x	x	x	x	x	x	x	x	1	1	x	x	x	0	x	x	x	x	x
	12	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0
	13	x	x	x	x	x	x	x	x	x	x	x	1	0	x	x	0	x	x	x
	14	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x
	15	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	x	x	0	0
	18	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Figure 42: Matrice de précédence modifiée du torrificateur de soja à deux modules.

IV.2.2.2.3 Matrice de précédence des modules et Gamme de désassemblage

En considérant les modules et les relations existant entre eux, nous obtenons la matrice simplifiée des figures 43, 44 et 45, respectivement pour le torrificateur à quatre modules, à trois modules et à deux modules.

$$\begin{array}{c}
 M_1 \quad M_2 \quad M_3 \quad M_4 \quad \text{Niveau} \\
 \\
 M_{ij} = \begin{array}{l}
 M_1 \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} 0 \\
 M_2 \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} 2 \\
 M_3 \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} 2 \\
 M_4 \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} 1
 \end{array}
 \end{array}$$

Figure 43:Matrice de précédence des quatre modules du torréfacteur

$$\begin{array}{c}
 M_1 \quad M_2 \quad M_3 \quad \text{Niveau} \\
 \\
 M_{ij} = \begin{array}{l}
 M_1 \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} 1 \\
 M_2 \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} 1 \\
 M_3 \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} 1
 \end{array}
 \end{array}$$

Figure 44:Matrice de précédence des trois modules du torréfacteur

$$\begin{array}{c}
 M_1 \quad M_2 \quad \text{Niveau} \\
 \\
 M_{ij} = \begin{array}{l}
 M_1 \begin{bmatrix} 0 & 1 \end{bmatrix} 1 \\
 M_2 \begin{bmatrix} 1 & 0 \end{bmatrix} 1
 \end{array}
 \end{array}$$

Figure 45:Matrice de précédence des deux modules du torréfacteur

La combinaison des résultats de l'analyse hiérarchique des modules constituant le torréfacteur de soja et de la matrice de précédence des composants qui les constituent permet de générer la gamme de désassemblage (Figures 46, 47 et 48).

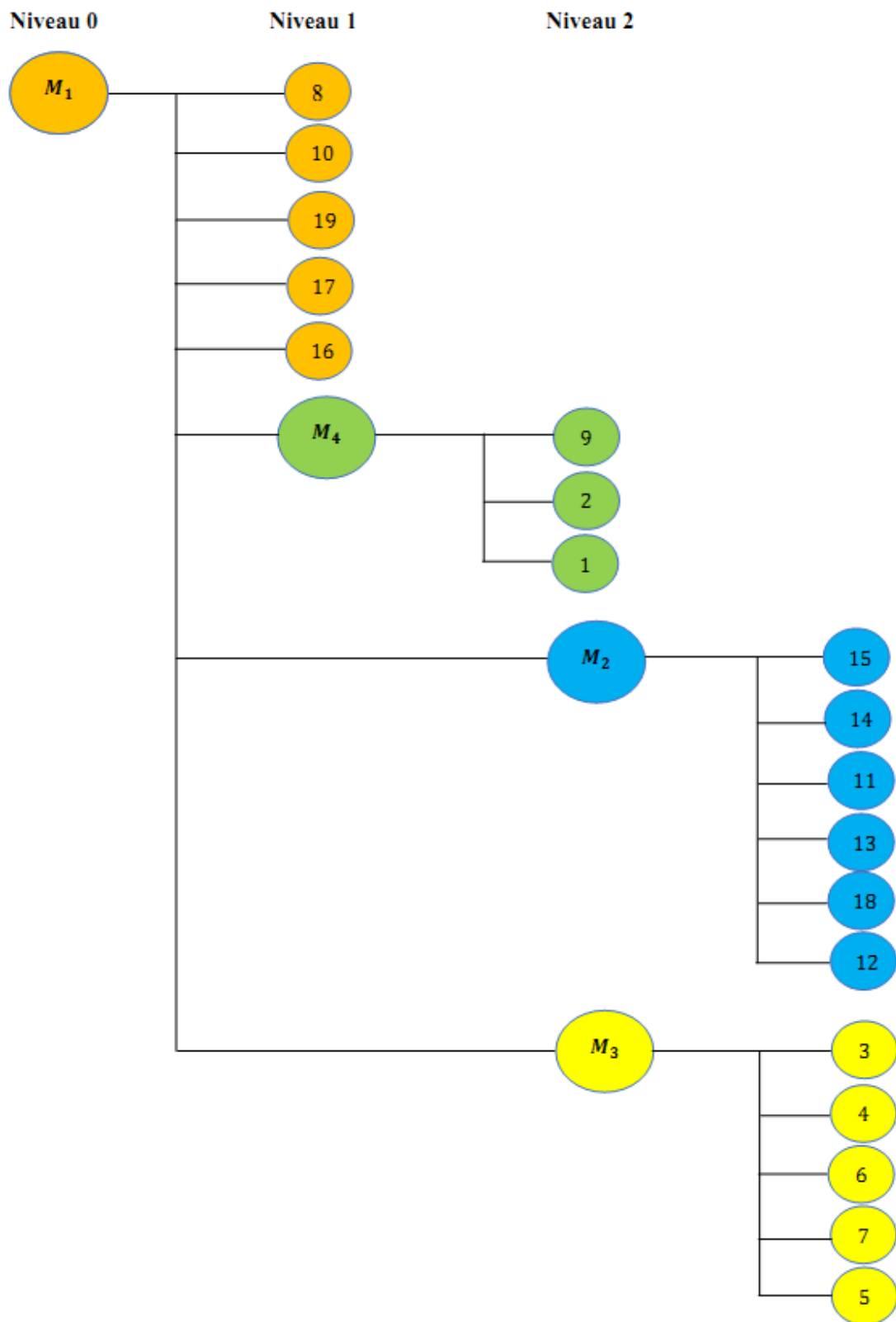


Figure 46: Gamme de désassemblage complète du torrificateur de soja à quatre modules.

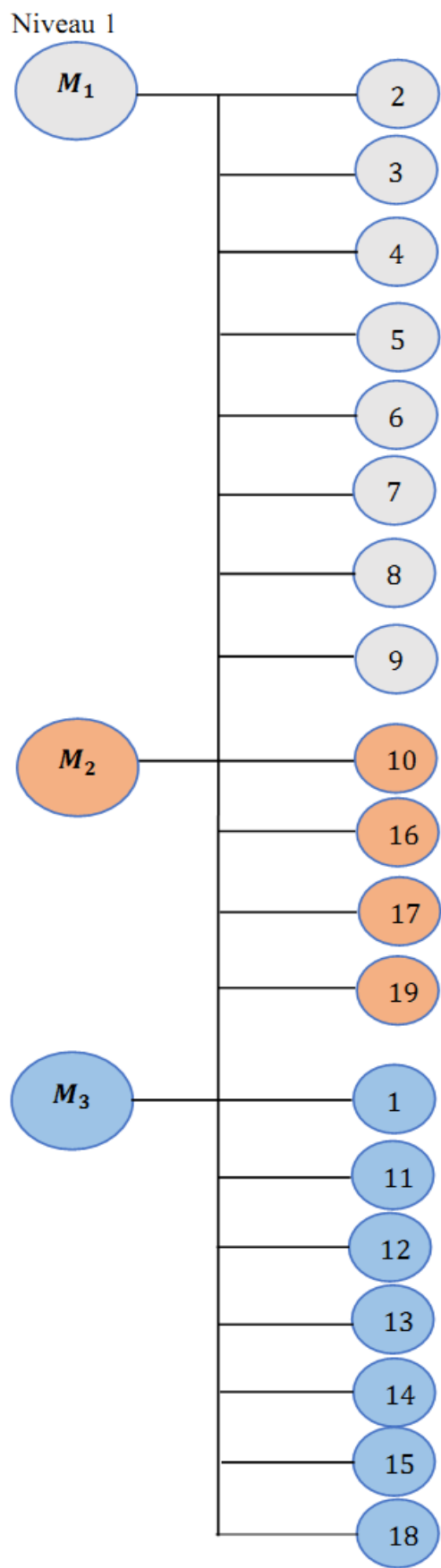


Figure 47: Gamme de désassemblage complète du torrificateur de soja à trois modules.

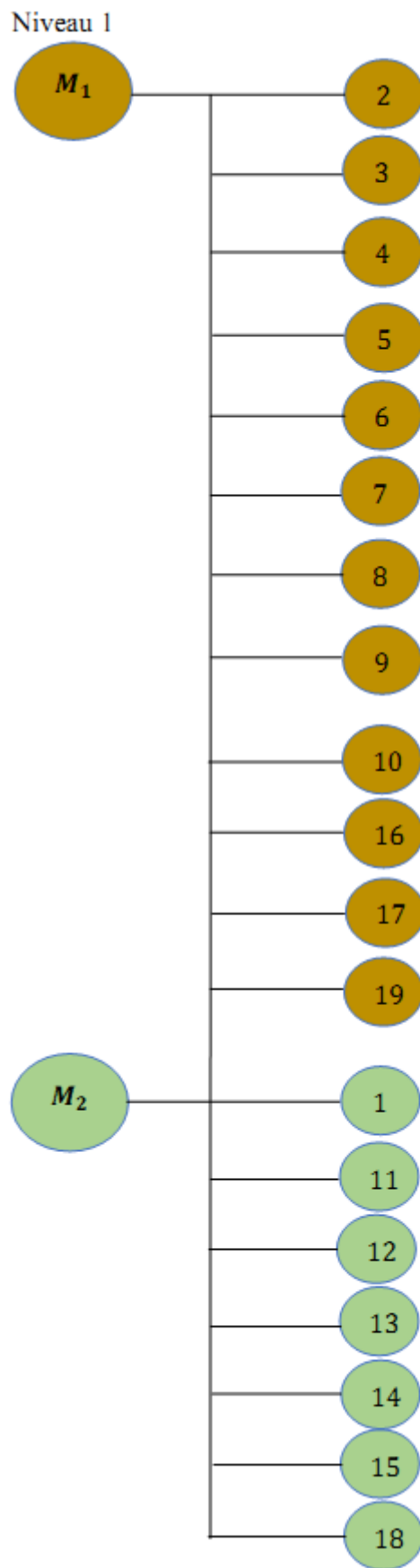


Figure 48: Gamme de désassemblage complète du torréfacteur de soja à deux modules.

L'arbre du torréfacteur à quatre modules présente les étapes à suivre pour le désassemblage du produit. La décomposition du produit en modules permet de procéder au désassemblage des composants ou bien des modules en parallèle, comme montré pour le niveau **1** les composants 8, 10, 19, 17, 16 et le module 4 sont récupérés en parallèle. Au niveau **2**, les composants 9, 2 et 1 et les modules 2 et 3 sont récupérés en parallèle. Par contre, lorsque le torréfacteur se subdivise en trois ou en deux modules, on constate un seul niveau de désassemblage, ce qui suppose que lors du désassemblage, on démonte à la fois tous les modules.

IV.2.2.2.4 Topologie des modules du torréfacteur de soja

D'après les gammes de désassemblage obtenues précédemment, il en ressort trois niveaux de démontage couplés en parallèles pour le torréfacteur à quatre modules (car par niveaux on intervient sur une catégorie de modules), et un seul niveau de démontage pour le torréfacteur lorsqu'il a trois ou deux modules.

Pour le torréfacteur ayant quatre modules, au niveau **0**, on démonte le module 1. Au niveau **1**, on démonte le module 4. Au niveau **2**, on démonte à la fois les modules 2 et 3, les deux modules ont donc un mode de fonctionnement de type série. C'est fort de ce constat que la topologie des modules du torréfacteur pour les différents cas de figures est donnée par la figure **49**.

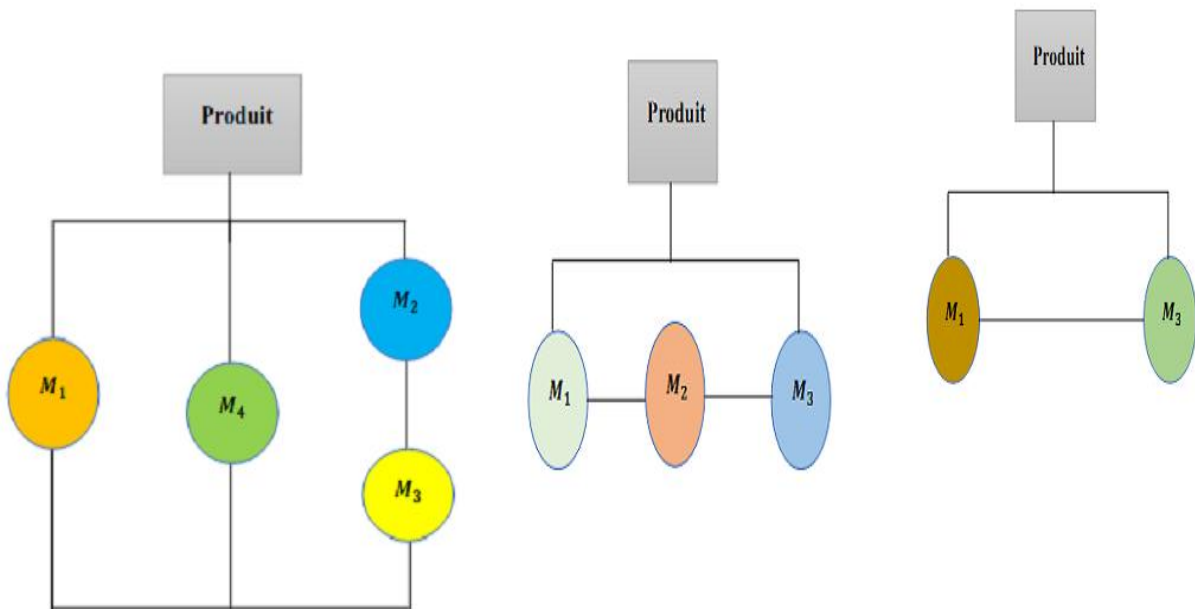


Figure 49: Topologie des modules du torréfacteur de soja (pour quatre, trois et deux modules)

IV.2.3 Deuxième étape de l'algorithme de clustering existant

Nous allons quantifier la modularité de notre torréfacteur de soja au moyen du MSI (Module Strength Indicator).

Cet indicateur de modularité étant fonction de la DSM (Design Structure Matrix), nous allons d'abord dresser la matrice structurale de conception qui découle directement de la matrice de relations et de la décomposition modulaire des composants du torréfacteur de soja.

IV.2.3.1 Matrice structurale de conception

D'après le tableau 32 (matrice de relations) et grâce au regroupement des composants en modules, les DSM du torréfacteur de soja à quatre modules, à trois modules et à deux modules sont respectivement données par les figures 50, 51 et 52.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	
C_1	0			21											18					
C_2		0		20		21														
C_3			0	23												18				
C_4	21	20	23	0	21													23		
C_5				21	0	19														
C_6		21			19	0										18				
C_7							0			21								23		
C_8								0	21		21									
C_9								21	0	17										
C_{10}							21		17	0										
C_{11}								21			0									
C_{12}												0	14						18	
C_{13}												14	0	21						
C_{14}													21	0	19	19				
C_{15}	18													19	0					
C_{16}			18			18								19		0				
C_{17}				23			23											0	23	
C_{18}																		23	0	14
C_{19}												18							14	0

Figure 50:DSM du torréfacteur de soja à quatre modules

Cette codification proposée des composants du torréfacteur de soja à quatre modules par les C_i ($1 \leq i \leq 19$), c'est pour mieux appliquer la formule de mesure de la modularité.

La codification des composants est donnée dans le tableau 36.

Tableau 36:Codification des composants du torrificateur à quatre modules

Nom	Désignation
Support roulement côté moteur	C ₁
Capot	C ₂
Trémie de sortie	C ₃
Cylindre principal	C ₄
Trémie d'entrée	C ₅
Support roulement côté cuvette	C ₆
Cuvette	C ₇
Axe secondaire	C ₈
Roulement axe secondaire	C ₉
Support roulement axe secondaire	C ₁₀
Racleur	C ₁₁
Courroie	C ₁₂
Disque 2	C ₁₃
Axe principal	C ₁₄
Roulement axe principal 1	C ₁₅
Roulement axe principal 2	C ₁₆
Bâti	C ₁₇
Moteur	C ₁₈
Disque moteur	C ₁₉

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}
E_1	0							14					23						
E_2		0	14					18											
E_3		14	0	21															
E_4			21	0	19	19													
E_5				19	0		18												
E_6				19		0								18					
E_7					18		0				21								
E_8	14	18						0											
E_9									0	20				21					
E_{10}										0	23								
E_{11}							21		20	23	0	21							
E_{12}											21	0							
E_{13}	23										23		0		23				
E_{14}						18			21					0					
E_{15}													23		0			21	
E_{16}																0	21		21
E_{17}																21	0	17	
E_{18}															21		17	0	
E_{19}																21			0

Figure 51:DSM du torréfacteur de soja à trois modules

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}
E_1	0							14					23						
E_2		0	14					18											
E_3		14	0	21															
E_4			21	0	19	19													
E_5				19	0		18												
E_6				19		0								18					
E_7					18		0				21								
E_8	14	18						0											
E_9									0		20			21					
E_{10}										0	23								
E_{11}							21		20	23	0	21							
E_{12}											21	0							
E_{13}	23										23		0		23				
E_{14}						18			21					0					
E_{15}													23		0			21	
E_{16}																0	21		21
E_{17}															21	0	17		
E_{18}															21		17	0	
E_{19}																21			0

Figure 52:DSM du torrificateur de soja à deux modules

Nous avons défini une nouvelle codification des composants du torrificateur de soja à trois ou à deux modules par les E_i ($1 \leq i \leq 19$), afin de mieux appliquer la formule de mesure de la modularité.

La codification des composants est donnée dans le tableau **37**.

Tableau 37:Nouvelle Codification des composants du torréfacteur à trois ou à deux modules

Désignation	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}
Nom	Moteur	Courroie	Disque 2	Axe principal	Roulement axe principal 1	Roulement axe principal 2	Support roulement côté moteur	Disque moteur	Capot	Trémie de sortie	Cylindre principal	Trémie de d'entrée	Bâti	Support roulement côté cuvette	Cuvette	Axe secondaire	Roulement axe secondaire	Support roulement axe secondaire	Racleur

IV.2.3.2 Quantification de la modularité

Notre produit pouvant être formé de quatre modules, de trois modules ou de deux modules, nous allons dans les différentes configurations pour chaque module, mesurer le degré de modularité en comparant les interactions à l'intérieur et à l'extérieur du module.

Rappelons tout de même que les mesures des interactions à l'intérieur et à l'extérieur du module sont données par les relations 74 et 75.

Pour des raisons de symétrie de notre DSM, la relation (75) devient donc :

$$MSI_e = \frac{\sum_{i=1}^{n_1-1} \sum_{j=n_1}^{n_2} 2DSM(i, j)}{2 \times ((n_1 - 1) \times (n_2 - n_1 + 1))} + \frac{\sum_{i=n_2+1}^N \sum_{j=n_1}^{n_2} 2DSM(i, j)}{2 \times ((N - n_2) \times (n_2 - n_1 + 1))} \quad (92)$$

La DSM du module considéré est donc donnée par la relation (93)

$$MSI = MSI_i - MSI_e \quad (93)$$

Pour notre DSM, on a : $N = 19$ composants. Nous allons balayer les différents nombres de modules possibles de notre torréfacteur.

a) Cas du torréfacteur à quatre modules

➤ pour le module 1 (avec $n_1 = 1$ et $n_2 = 5$), on a :

$$MSI_i(M_1) = \frac{\sum_{i=1}^5 \sum_{j=1}^5 DSM(i, j)}{(5-1+1)^2 - (5-1+1)} = \frac{170}{20} = 8,5 ;$$

$$MSI_e(M_1) = \frac{\sum_{i=1}^0 \sum_{j=1}^5 2DSM(i, j)}{2 \times ((1-1) \times (5-1+1))} + \frac{\sum_{i=6}^{19} \sum_{j=1}^5 2DSM(i, j)}{2 \times ((19-5) \times (5-1+1))} = 0 + \frac{162}{140} = 1,1571 ;$$

$$MSI(M_1) = MSI_i(M_1) - MSI_e(M_1) = 7,3429 .$$

➤ pour le module **2** (avec $\mathbf{n}_1 = \mathbf{6}$ et $\mathbf{n}_2 = \mathbf{11}$), on a :

$$MSI_i(M_2) = \frac{\sum_{i=6}^{11} \sum_{j=6}^{11} DSM(i, j)}{(11-6+1)^2 - (11-6+1)} = \frac{160}{30} = 5,3333 ;$$

$$MSI_e(M_2) = \frac{\sum_{i=1}^5 \sum_{j=6}^{11} 2DSM(i, j)}{2 \times ((6-1) \times (11-6+1))} + \frac{\sum_{i=12}^{19} \sum_{j=6}^{11} 2DSM(i, j)}{2 \times ((19-11) \times (11-6+1))} = \frac{80}{60} + \frac{82}{96} = 2,1875 ;$$

$$MSI(M_2) = MSI_i(M_2) - MSI_e(M_2) = 3,1458 .$$

➤ pour le module **3** (avec $\mathbf{n}_1 = \mathbf{12}$ et $\mathbf{n}_2 = \mathbf{16}$), on a :

$$MSI_i(M_3) = \frac{\sum_{i=12}^{16} \sum_{j=12}^{16} DSM(i, j)}{(16-12+1)^2 - (16-12+1)} = \frac{146}{20} = 7,3 ;$$

$$MSI_e(M_3) = \frac{\sum_{i=1}^{11} \sum_{j=12}^{16} 2DSM(i, j)}{2 \times ((12-1) \times (16-12+1))} + \frac{\sum_{i=17}^{19} \sum_{j=12}^{16} 2DSM(i, j)}{2 \times ((19-16) \times (16-12+1))} = \frac{108}{110} + \frac{36}{30} = 2,1818 ;$$

$$MSI(M_3) = MSI_i(M_3) - MSI_e(M_3) = 5,1182 .$$

➤ pour le module **4** (avec $\mathbf{n}_1 = \mathbf{17}$ et $\mathbf{n}_2 = \mathbf{19}$), on a :

$$MSI_i(M_4) = \frac{\sum_{i=17}^{19} \sum_{j=17}^{19} DSM(i, j)}{(19-17+1)^2 - (19-17+1)} = \frac{74}{6} = 12,3333 ;$$

$$MSI_e(M_4) = \frac{\sum_{i=1}^{16} \sum_{j=17}^{19} 2DSM(i, j)}{2 \times ((17-1) \times (19-17+1))} + \frac{\sum_{i=20}^{19} \sum_{j=17}^{19} 2DSM(i, j)}{2 \times ((19-19) \times (19-17+1))} = \frac{128}{96} + 0 = 1,3333 ;$$

$$MSI(M_4) = MSI_i(M_4) - MSI_e(M_4) = 11.$$

Le tableau 38 fait la synthèse des MSI évalués pour chaque module du torréfacteur à quatre modules.

Tableau 38:Degré de modularité des quatre modules du torréfacteur avec l'algorithme existant

Modules		M_1	M_2	M_3	M_4
MSI	MSI_i	8,5	5,3333	7,3	12,3333
	MSI_e	1,1571	2,1875	2,1818	1,3333
	$MSI = MSI_i - MSI_e$	7,3429	3,1458	5,1182	11

Au vu de ces résultats, on constate que le degré de modularité augmente lorsque le nombre d'interactions internes au module croît et que le nombre d'interactions externes décroît. Le module 4 présente l'interaction la plus élevée, il assure donc une bonne cohésion interne du fait que les interactions à l'intérieure du module sont d'autant plus importantes que les interactions externes au module qui décroissent. D'après Hwai-En et al. (2008), plus l'intensité de liaison est forte à l'intérieure d'un module, signifie qu'elle est faible à l'extérieure du module, ce qui indique d'un point de vue maintenance qu'il est facile d'assembler et de désassembler des composants dans un module. Cette observation se traduit par un fort degré de connexion étroit entre les composants d'un module. De ce fait, le module 4 est le module le plus facile à maintenir, et cette observation faite par Hwai-En et al. (2008) est donc pertinente car on constate bien que le module 4 est moins complexe avec une taille plus faible que celles des autres modules. Cependant, le module 2 présente l'interaction la plus faible, il assure une faible cohésion interne et donc plus difficile à maintenir. On voit bien qu'il est plus complexe avec une plus grande taille, soit de 6 composants. Donc les interactions d'un module sont une fonction décroissante de la taille du module (figure 53). D'autre part, on constate bien que les modules 1 et 3 ont la même taille. En se fiant juste à ce paramètre, il serait difficile de savoir lors d'un problème de maintenance le module qu'on peut facilement maintenir. Grâce à la connaissance de la densité de chaque module (MSI), on constate que le module 1 est plus dense que le module 3, par conséquent il est donc plus facile d'intervenir sur le module 1 que sur le module 3.

La figure 53 traduit bien l'évolution du MSI des modules.

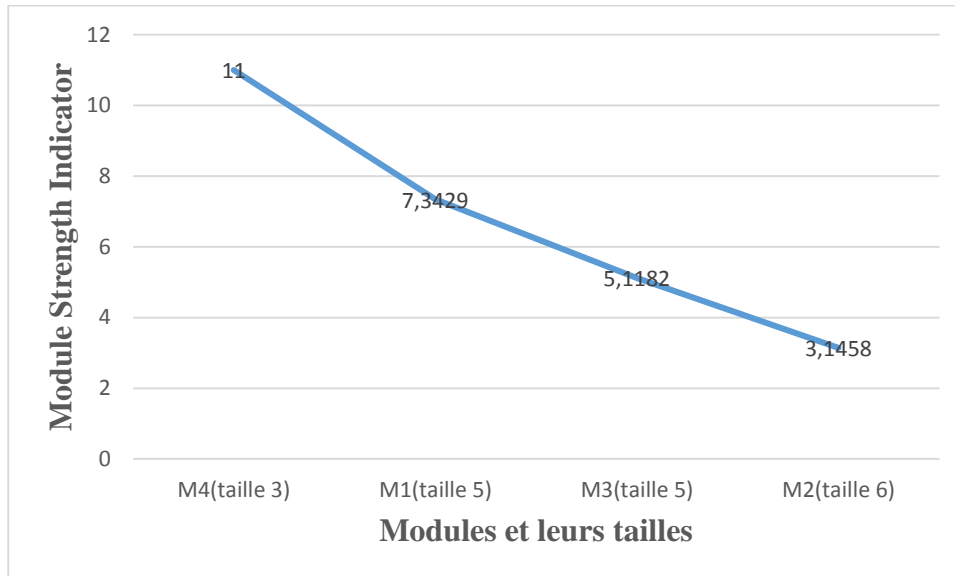


Figure 53: Evolution du MSI des quatre modules avec l'algorithme existant

b) Cas du torréfacteur à trois modules

En appliquant le modèle d'évaluation de la métrique de la modularité proposé par Wietfield et al. (2002) sur les trois modules du torréfacteur, on obtient les résultats consignés dans le tableau 39.

Tableau 39: Degré de modularité des trois modules du torréfacteur avec l'algorithme existant

Modules		M_1	M_2	M_3
MSI	MSI_i	4,39	10,6	4,9
	MSI_e	0,7	0,75	0,5
	$MSI = MSI_i - MSI_e$	3,69	9,85	4,4

On constate que pour le torréfacteur à trois modules, le module M_2 est le plus facile à maintenir. Par contre, le module M_1 est le plus difficile à maintenir, du fait d'un MSI plus faible.

L'évolution du MSI en fonction de la taille des modules est donnée par la figure 54.

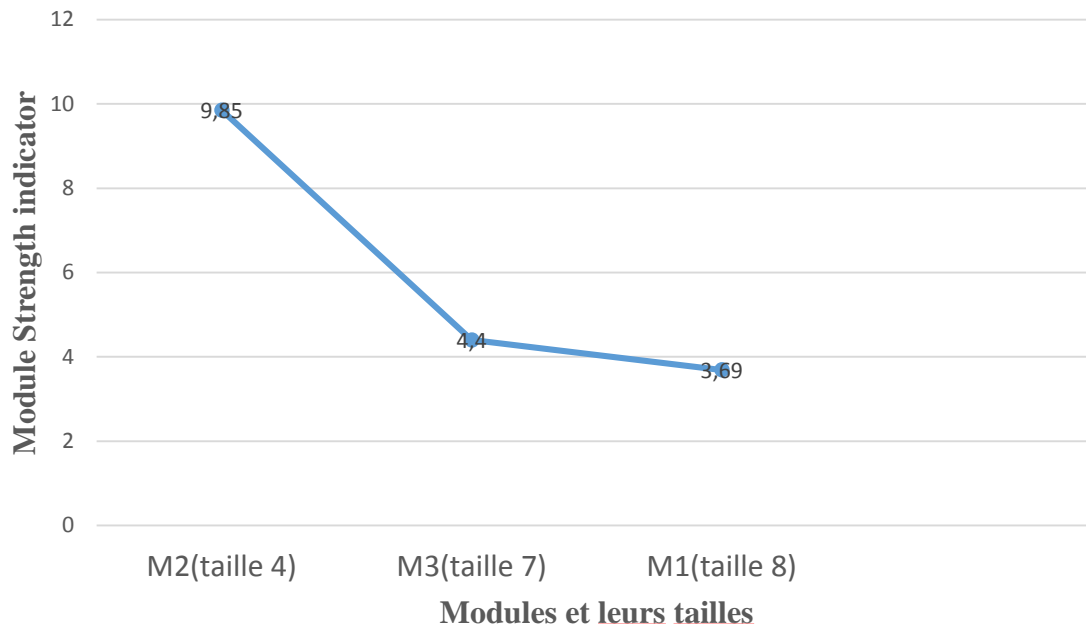


Figure 54: Evolution du MSI des trois modules avec l’algorithme existant

Au regard de la figure 54, on fait également le même constat selon lequel les interactions d’un module sont une fonction décroissante de la taille du module.

c) Cas du torréfacteur à deux modules

En appliquant comme précédemment le modèle d’évaluation de métrique de la modularité proposé par Wietfield et al. (2002) sur les deux modules du torréfacteur, on obtient les résultats consignés dans le tableau 40.

Tableau 40: Degré de modularité des deux modules du torréfacteur avec l’algorithme existant

Modules		M_1	M_2
MSI	MSI_i	3,15	4,9
	MSI_e	0,7	1
	$MSI = MSI_i - MSI_e$	2,45	3,9

Au vu de ces résultats, on constate que le module M_2 est le plus facile à maintenir. L’évolution du MSI en fonction de la taille des modules est donnée par la figure 55.

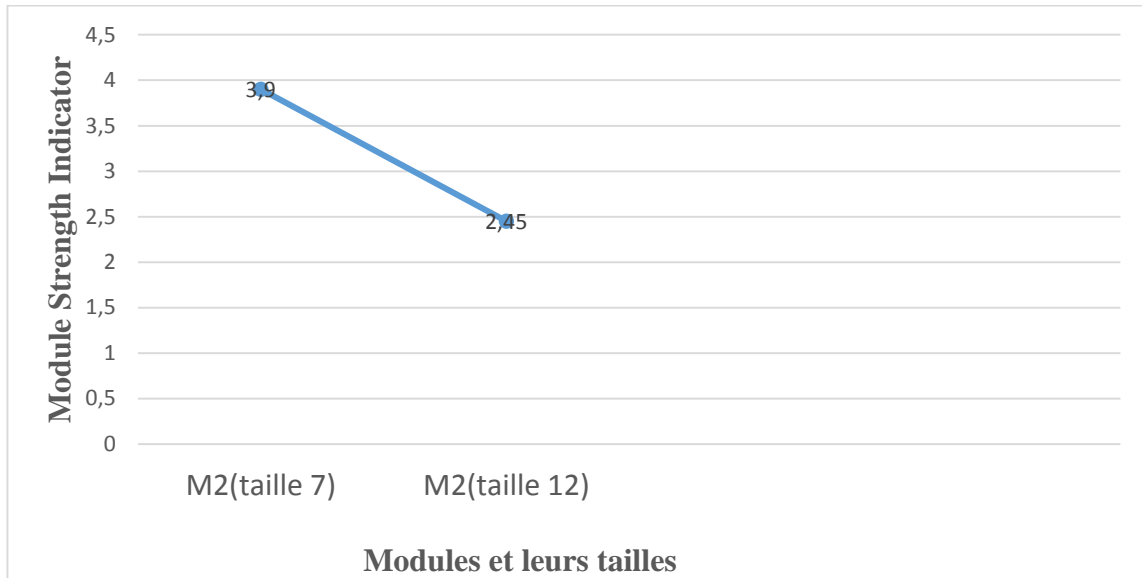


Figure 55: Evolution du MSI des deux modules avec l’algorithme existant

La conclusion qu’on fait à ce niveau est également similaire aux précédentes. Donc globalement que le torréfacteur aye quatre, trois ou deux modules, le MSI décroît lorsque la taille du module augmente.

IV.2.4 Troisième étape de l’algorithme de clustering existant

Nous allons ici évaluer les coûts de couplage de chaque module du torréfacteur de soja, en considérant les différents cas de figures et déduire le coût de couplage global.

Rappelons tout de même que les coûts de couplage interne et externes d’un module sont donnés par les équations 81 et 82.

Pour des raisons de symétrie de notre DSM, les relations 81 et 82 deviennent respectivement :

$$\text{Coût Couplage Interne}(M_k) = \sum_{i \in M_k} \sum_{j \in M_k, j \neq i} (2DSM(i, j)) \times (\text{taille}(M_k))^2 \quad (94)$$

$$\text{Coût Couplage Externe}(M_k) = \sum_{i \in M_k} \sum_{j \notin M_k, j \in M_n} \left[(2DSM(i, j)) \times (\text{taille}(DSM) + \text{taille}(M_k) + \text{taille}(M_n))^2 \right] \quad (95)$$

Le coût de couplage global des modules est donné par la relation 96.

$$\text{Coût Total Couplage} = \sum_{M_k}^{\text{Modules}} (\text{Coût Couplage Interne}(M_k) + \text{Coût Couplage Externe}(M_k)) \quad (96)$$

a) Cas du torréfacteur à quatre modules

➤ Pour le module **1**, on a :

$$\begin{aligned} \text{Coût Couplage Interne}(M_1) &= \sum_{i \in M_1} \sum_{j \in M_1, j \neq i} (2DSM(i, j)) \times (\text{taille}(M_1))^2 \\ &= 2(21+20+23+21) \times (5)^2 \\ &= 4250 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage Externe}(M_1) &= \sum_{i \in M_1} \sum_{j \notin M_1, j \in M_n} \left[(2DSM(i, j)) \times (\text{taille}(DSM) + \text{taille}(M_1) + \text{taille}(M_n))^2 \right] \\ &= 2 \times 21(5 + 6 + 19)^2 + 2 \times 19(5 + 6 + 19)^2 + 2 \times 18(5 + 5 + 19)^2 + 2 \times 18(5 + 5 + 19)^2 + 2 \times 23(5 + 3 + 19)^2 \\ &= 135810 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage}(M_1) &= \text{Coût Couplage Interne}(M_1) + \text{Coût Couplage Externe}(M_1) \\ &= 140060. \end{aligned}$$

➤ Pour le module **2**, on a :

$$\begin{aligned} \text{Coût Couplage Interne}(M_2) &= \sum_{i \in M_2} \sum_{j \in M_2, j \neq i} (2DSM(i, j)) \times (\text{taille}(M_2))^2 \\ &= 2(21+21+17+21) \times (6)^2 \\ &= 5760 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage Externe}(M_2) &= \sum_{i \in M_2} \sum_{j \notin M_2, j \in M_n} \left[(2DSM(i, j)) \times (\text{taille}(DSM) + \text{taille}(M_2) + \text{taille}(M_n))^2 \right] \\ &= 2 \times 18(6 + 5 + 19)^2 + 2 \times 23(6 + 3 + 19)^2 \\ &= 68464 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage}(M_2) &= \text{Coût Couplage Interne}(M_2) + \text{Coût Couplage Externe}(M_2) \\ &= 74224. \end{aligned}$$

➤ Pour le module **3**, on a :

$$\begin{aligned} \text{Coût Couplage Interne}(M_3) &= \sum_{i \in M_3} \sum_{j \in M_3, j \neq i} (2DSM(i, j)) \times (\text{taille}(M_3))^2 \\ &= 2(14+21+19+19) \times (5)^2 \\ &= 3650 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage Externe}(M_3) &= \sum_{i \in M_3} \sum_{j \notin M_3, j \in M_n} \left[(2DSM(i, j)) \times (taille(DSM) + taille(M_3) + taille(M_n))^2 \right] \\ &= 2 \times 18(5 + 3 + 19)^2 \\ &= 26244 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage}(M_3) &= \text{Coût Couplage Interne}(M_3) + \text{Coût Couplage Externe}(M_3) \\ &= 29894. \end{aligned}$$

➤ Pour le module **4**, on a :

$$\begin{aligned} \text{Coût Couplage Interne}(M_4) &= \sum_{i \in M_4} \sum_{j \in M_4, j \neq i} (2DSM(i, j)) \times (taille(M_4))^2 \\ &= 2(23+14) \times (3)^2 \\ &= 666 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage Externe}(M_4) &= \sum_{i \in M_4} \sum_{j \notin M_4, j \in M_n} \left[(2DSM(i, j)) \times (taille(DSM) + taille(M_4) + taille(M_n))^2 \right] \\ &= 0 ; \end{aligned}$$

$$\begin{aligned} \text{Coût Couplage}(M_4) &= \text{Coût Couplage Interne}(M_4) + \text{Coût Couplage Externe}(M_4) \\ &= 666. \end{aligned}$$

Le coût de couplage global des quatre modules du torréfacteur de soja s'obtient donc comme suit :

$$\begin{aligned} \text{Coût Total Couplage} &= \sum_{i=1}^4 \text{Coût Couplage}(M_i) \\ &= 244844. \end{aligned}$$

La synthèse des coûts de couplage du torréfacteur à quatre modules est donné par le tableau **41**.

Tableau 41: Coût de couplage des quatre modules du torréfacteur avec l'algorithme existant

Modules		M_1	M_2	M_3	M_4
Coût de Couplage	Interne	4250	5760	3650	666
	Externe	135810	68464	26244	0
	Interne + Externe	140060	74224	29894	666

Au vu de ces résultats, on constate que le module **1** a le coût de couplage le plus fort, donc ce module exige plus d'attention, plus de ressources, ou plus de travail des concepteurs. Par

contre, le module **4** présente un coût de couplage le plus faible, donc ce module exige moins d'attention, moins de ressources, ou moins de travail des concepteurs.

Partant de ce tableau, il se trouve que le coût de couplage interne des modules varie de 666 à 5760 et le coût de couplage externe des modules varie de 0 à 135810. En additionnant les bornes inférieures et supérieures des coûts, nous déduisons le seuil de satisfaction du coût de couplage donné par l'intervalle]666; 141570[.

La figure **56** montre l'évolution du coût des quatre modules.

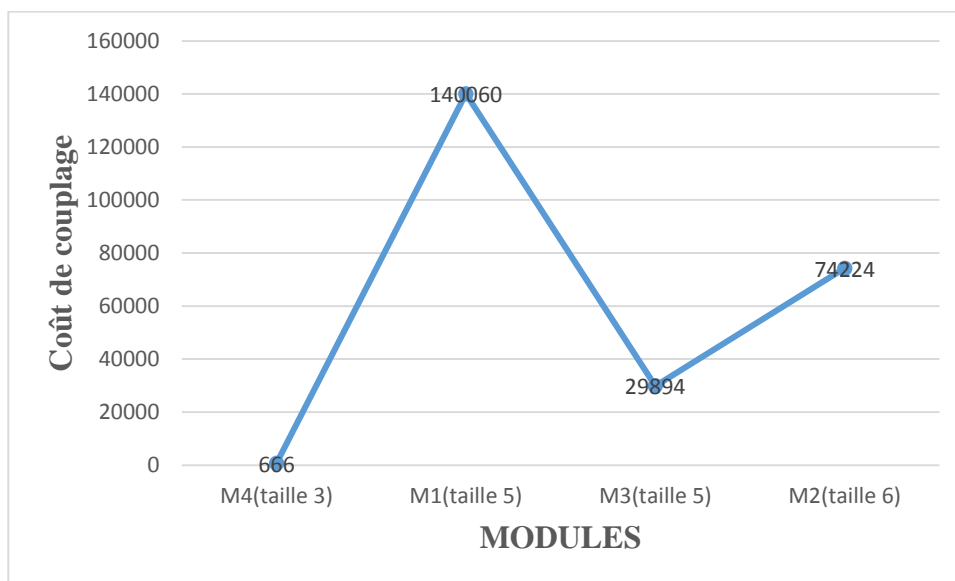


Figure 56: Evolution du coût des quatre modules avec l'algorithme existant

Au regard de ce graphe, le coût de couplage n'est quasiment pas une fonction croissante de la taille des modules. En effet, le module **2** qui est de plus grande taille a un coût de couplage plus faible que le module **1** de taille inférieure à celle du module **2**. Cela se traduit par le fait que les interactions du modules **1** sont plus fortes que celles du module **2**. Par contre, le coût de couplage est une fonction croissante de la taille des modules **4**, **3** et **2** respectivement. On constate dès cet instant que le coût de couplage est à la fois fonction de la taille et des valeurs des interactions des modules.

b) Cas du torréfacteur à trois modules

En appliquant l'algorithme existant sur les trois modules du torréfacteur, le coût de couplage obtenu est donné dans le tableau **42**.

Tableau 42:Coût de couplage des trois modules du torrificateur avec l’algorithme existant

Modules		M_1	M_2	M_3
Coût de Couplage	<i>Interne</i>	85705	54402	100245
	<i>Externe</i>	13105022	3280021	0
	<i>Interne + Externe</i>	13190727	3334423	100245

Le module M_3 est le moins contraignant pour les concepteurs, du fait d’un plus faible coût de couplage .

La figure 57 montre l’évolution du coût des trois modules.

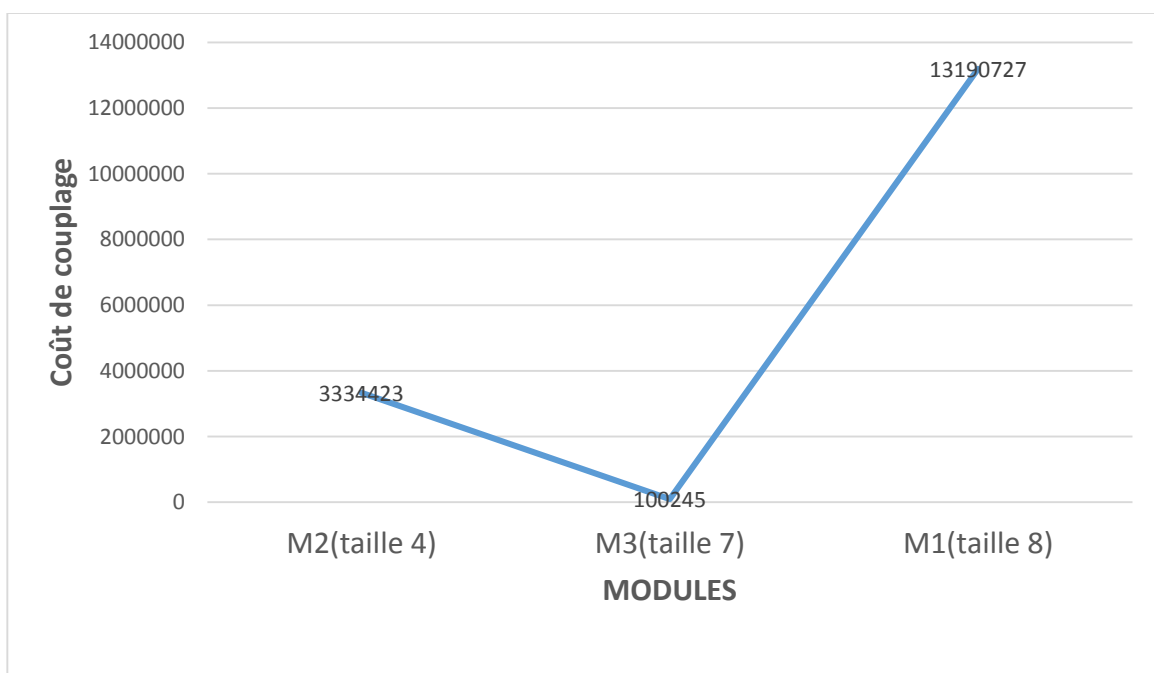


Figure 57:Evolution du coût des trois modules avec l’algorithme existant

On constate également que le coût de couplage ici n’est pas une fonction croissante de la taille des modules. Le coût global de couplage des trois modules du torrificateur de soja vaut : 16625395.

c) Cas du torrificateur à deux modules

En appliquant l’algorithme existant sur les deux modules du torrificateur, le coût de couplage obtenu est donné dans le tableau 43.

Tableau 43:Coût de couplage des deux modules du torrificateur avec l’algorithme existant

Modules		M_1	M_2
	<i>Interne</i>	119808	10094

Coût de Couplage	<i>Externe</i>	22445536	0
	<i>Interne + Externe</i>	22565344	10094

On constate que le module M_1 a le coût de couplage le plus élevé, il est donc plus contraignant pour les concepteurs. Par contre, le module M_2 est le module le moins contraignant pour la conception.

La figure 58 montre l'évolution du coût des deux modules.

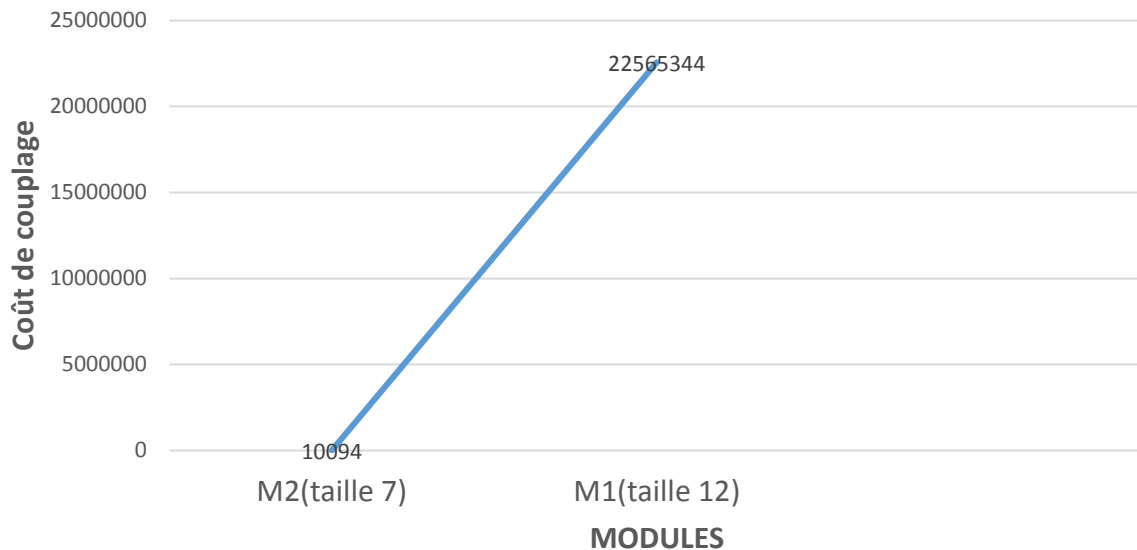


Figure 58: Evolution du coût des deux modules avec l'algorithme existant

D'après la figure 58, le coût de couplage est une fonction croissante de la taille des modules.

IV.3 RESULTATS DE L'ALGORITHME PROPOSE ET DISCUSSION

Cette partie illustre la mise en application du nouvel algorithme sur le *torréfacteur de soja*.

IV.3.1 Matrice structurelle de conception

En appliquant les modifications apportées à la DSM de référence (DSM produit), on obtient les DSM numérique pour notre produit à 19 composants, données aux figures 59, 60 et 61 respectivement pour le torréfacteur à quatre modules, à trois modules et à deux modules.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	
C_1	0			1											1					
C_2		0		1		1														
C_3			0	1												1				
C_4	1	1	1	0	1												1			
C_5				1	0	1														
C_6		1			1	0										1				
C_7							0			1							1			
C_8								0	1		1									
C_9								1	0	1										
C_{10}							1		1	0										
C_{11}								1			0									
C_{12}												0	1						1	
C_{13}												1	0	1						
C_{14}													1	0	1	1				
C_{15}	1													1	0					
C_{16}			1			1								1		0				
C_{17}				1			1											0	1	
C_{18}																		1	0	1
C_{19}												1							1	0

Figure 59:DSM numérique du torréfacteur de soja à quatre modules

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}
E_1	0							14					23						
E_2		0	14					18											
E_3		14	0	21															
E_4			21	0	19	19													
E_5				19	0		18												
E_6				19		0								18					
E_7					18		0				21								
E_8	14	18						0											
E_9									0	20				21					
E_{10}										0	23								
E_{11}							21		20	23	0	21							
E_{12}											21	0							
E_{13}	23										23		0		23				
E_{14}						18			21					0					
E_{15}													23		0			21	
E_{16}																0	21		21
E_{17}																21	0	17	
E_{18}															21		17	0	
E_{19}																21			0

Figure 60:DSM numérique du torréfacteur de soja à trois modules

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	E_{17}	E_{18}	E_{19}
E_1	0							1					1						
E_2		0	1					1											
E_3			1	0	1														
E_4				1	0	1	1												
E_5					1	0		1											
E_6						1		0						1					
E_7							1	0			1								
E_8	1	1						0											
E_9									0	1				1					
E_{10}										0	1								
E_{11}							1	1	1	0	1								
E_{12}											1	0							
E_{13}	1										1		0		1				
E_{14}						1		1						0					
E_{15}													1		0			1	
E_{16}																0	1		1
E_{17}																1	0	1	
E_{18}															1		1	0	
E_{19}																1			0

Figure 61:DSM numérique du torréfacteur de soja à deux modules

Les valeurs des interactions de ces matrices seront fort utiles dans l'évaluation des coefficients correcteurs des coûts de couplage des modules du torréfacteur de soja.

IV.3.2 Coefficients correcteurs

Les coefficients correcteurs du coût de couplage sont donnés par les relation 84 et 85. Pour des raisons de symétrie de la DSM, la relation 85 devient :

$$MSI'_e(M_k) = \frac{\sum_{i=1}^{n_1-1} \sum_{j=n_1}^{n_2} 2\delta(i, j)}{2 \times ((n_1 - 1) \times (n_2 - n_1 + 1))} + \frac{\sum_{i=n_2+1}^{taille(DSM)} \sum_{j=n_1}^{n_2} 2\delta(i, j)}{2 \times ((taille(DSM) - n_2) \times (n_2 - n_1 + 1))} \quad (97)$$

a) Cas du torréfacteur à quatre modules

Pour notre DSM numérique à **19** composants les coefficients correcteurs pour les différents modules sont évalués ainsi qu'il suit :

➤ pour le module **1** (avec $n_1 = 1$ et $n_2 = 5$), on a :

$$MSI'_i(M_1) = \frac{\sum_{i=1}^5 \sum_{j=1}^5 \delta(i, j)}{(5-1+1)^2 - (5-1+1)} = \frac{8}{20} = 0,4 ;$$

$$MSI'_e(M_1) = \frac{\sum_{i=1}^0 \sum_{j=1}^5 2\delta(i, j)}{2 \times ((1-1) \times (5-1+1))} + \frac{\sum_{i=6}^{19} \sum_{j=1}^5 2\delta(i, j)}{2 \times ((19-5) \times (5-1+1))} = 0 + \frac{8}{140} = 0,0571 ;$$

$$MSI'(M_1) = MSI'_i(M_1) - MSI'_e(M_1) = 0,3429 .$$

➤ pour le module **2** (avec $n_1 = 6$ et $n_2 = 11$), on a :

$$MSI'_i(M_2) = \frac{\sum_{i=6}^{11} \sum_{j=6}^{11} \delta(i, j)}{(11-6+1)^2 - (11-6+1)} = \frac{8}{30} = 0,2666 ;$$

$$MSI'_e(M_2) = \frac{\sum_{i=1}^5 \sum_{j=6}^{11} 2\delta(i, j)}{2 \times ((6-1) \times (11-6+1))} + \frac{\sum_{i=12}^{19} \sum_{j=6}^{11} 2\delta(i, j)}{2 \times ((19-11) \times (11-6+1))} = \frac{4}{60} + \frac{4}{96} = 0,1083 ;$$

$$MSI'(M_2) = MSI'_i(M_2) - MSI'_e(M_2) = 0,1583 .$$

➤ pour le module **3** (avec $n_1 = 12$ et $n_2 = 16$), on a :

$$MSI'_i(M_3) = \frac{\sum_{i=12}^{16} \sum_{j=12}^{16} \delta(i, j)}{(16-12+1)^2 - (16-12+1)} = \frac{8}{20} = 0,4 ;$$

$$MSI'_e(M_3) = \frac{\sum_{i=1}^{11} \sum_{j=12}^{16} 2\delta(i, j)}{2 \times ((12-1) \times (16-12+1))} + \frac{\sum_{i=17}^{19} \sum_{j=12}^{16} 2\delta(i, j)}{2 \times ((19-16) \times (16-12+1))} = \frac{6}{110} + \frac{2}{30} = 0,1212 ;$$

$$MSI'(M_3) = MSI'_i(M_3) - MSI'_e(M_3) = 0,2788 .$$

➤ pour le module **4** (avec $n_1 = 17$ et $n_2 = 19$), on a :

$$MSI'_i(M_4) = \frac{\sum_{i=17}^{19} \sum_{j=17}^{19} 2\delta(i, j)}{(19-17+1)^2 - (19-17+1)} = \frac{4}{6} = 0,6666 ;$$

$$MSI'_e(M_4) = \frac{\sum_{i=1}^{16} \sum_{j=17}^{19} 2\delta(i, j)}{2 \times ((17-1) \times (19-17+1))} + \frac{\sum_{i=20}^{19} \sum_{j=17}^{19} 2\delta(i, j)}{2 \times ((19-19) \times (19-17+1))} = \frac{6}{96} + 0 = 0,0625 ;$$

$$MSI'(M_4) = MSI'_i(M_4) - MSI'_e(M_4) = 0,6041 .$$

La synthèse des coefficients correcteurs du coût de couplage du torréfacteur à quatre modules est donné par le tableau 44.

Tableau 44: Coefficients correcteurs du coût de couplage du torréfacteur à quatre modules

Modules		M_1	M_2	M_3	M_4
MSI'	MSI'_i	0,4	0,2666	0,4	0,6666
	MSI'_e	0,0571	0,1083	0,1212	0,0625
	$MSI' = MSI'_i - MSI'_e$	0,3429	0,1583	0,2788	0,6041

La figure 62 suivante fait une étude comparative du MSI corrigé pour le torréfacteur à quatre modules et de celui obtenu avec l'algorithme existant.

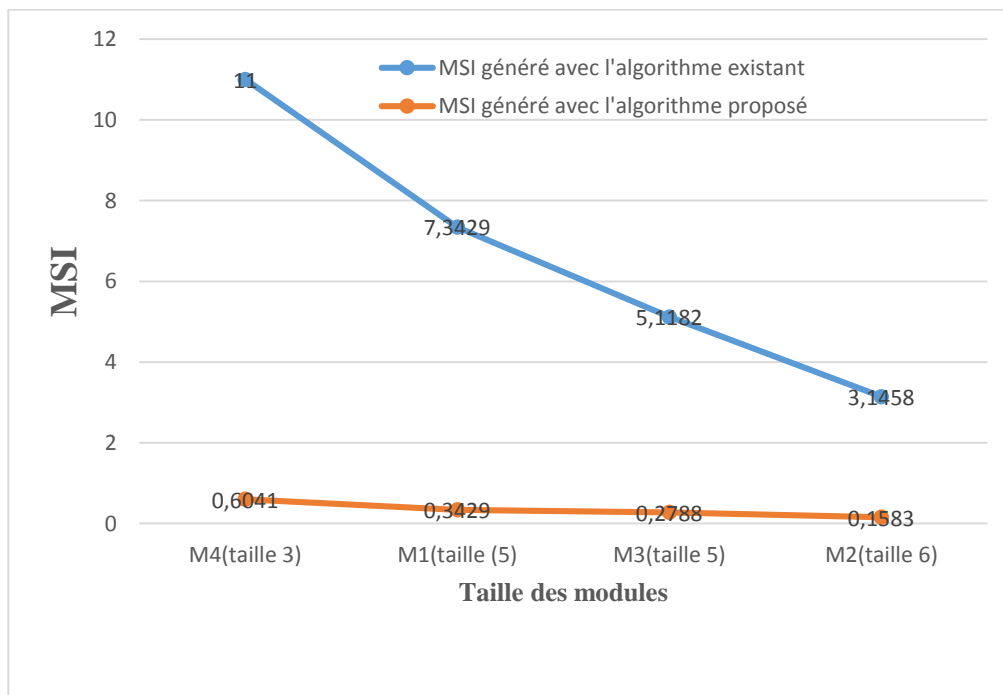


Figure 62: Etude comparative du MSI corrigé du torréfacteur à quatre modules et de celui obtenu avec l'algorithme existant

Au vu ces résultats, on fait une interprétation similaire sur le degré de modularité que celle donnée par l'application de l'algorithme existant sur le torréfacteur de soja. Le module **4** est celui qui pénalise le plus les interactions extérieures, donc il est le plus facile à maintenir. Par contre, le module **2** est celui qui présente la force de cohésion la plus faible (MSI faible), donc il est plus difficile à maintenir. Cette interprétation est donc tout à fait logique car plus le module est de grande taille, plus il est difficile à maintenir. Globalement cette étape de notre algorithme a favorisé à la création de modules denses, et plus spécifiquement du module **4** qui est le plus dense avec un coefficient de correction plus proche de 1.

b) Cas du torréfacteur à trois modules

En appliquant l'algorithme proposé, on obtient la synthèse des coefficients correcteurs du coût de couplage du torréfacteur à trois modules donné par le tableau **45**.

Tableau 45: Coefficients correcteurs du coût de couplage du torréfacteur à trois modules

Modules		M_1	M_2	M_3
MSI'	MSI_i'	0,25	0,5	0,23
	MSI_e'	0,03	0,017	0,04
	$MSI' = MSI_i' - MSI_e'$	0,22	0,48	0,19

Avec le nouvel algorithme, nous faisons le même constat que celui de la littérature qui stipule que le module M_2 du torréfacteur est le plus facile à maintenir. Par contre, le module le moins facile à maintenir ici est le module M_3 , donc dans la quantification de la modularité, en considérant le torréfacteur subdivisé en trois modules, notre algorithme n'est plus en accord avec celui de la littérature.

La figure **63** suivante fait une étude comparative du MSI corrigé du torréfacteur à trois modules et de celui obtenu avec l'algorithme existant.

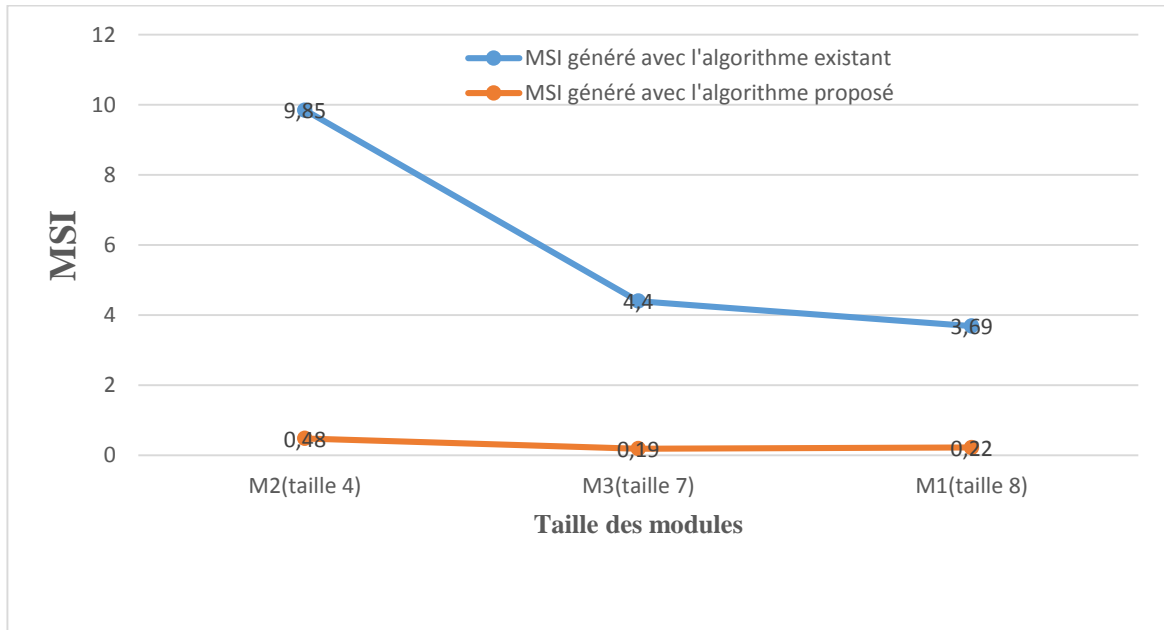


Figure 63: Etude comparative du MSI corrigé du torréfacteur à trois modules et de celui obtenu avec l'algorithme existant

Au regard de la figure 63, on constate bien que le MSI corrigé n'est pas en accord avec celui de la littérature. En effet, le MSI corrigé décroît lorsqu'on passe du module M_2 au module M_3 et augmente légèrement lorsqu'on passe du module M_3 au module M_1 .

c) Cas du torréfacteur à deux modules

En appliquant l'algorithme proposé, on obtient la synthèse des coefficients correcteurs du coût de couplage du torréfacteur à deux modules donné par le tableau 46.

Tableau 46: Coefficients correcteurs du coût de couplage du torréfacteur à deux modules

Modules		M_1	M_2
MSI'	MSI_i'	0,16	0,23
	MSI_e'	0,034	0,047
	$MSI'_e = MSI_i' - MSI_e'$	0,126	0,183

D'après les résultats obtenus, on fait toujours la même observation selon laquelle le module M_2 est le plus facile à maintenir. Nos résultats sont donc en accord avec ceux obtenus dans la littérature. Par contre, il faut noter que notre algorithme a favorisé la création des modules plus denses, avec un MSI se situant dans l'intervalle [0; 1].

IV.3.3 Coût de couplage

En tenant compte des coefficients correcteurs évalués précédemment (MSI_i' et MSI_e') et ayant pris connaissance des coûts de couplage générés par l'algorithme de base, nous allons évaluer le coût de couplage corrigé de chaque module constituant le torréfacteur de soja et déduire le nouveau coût de couplage global des modules du torréfacteur de soja.

Nous évaluons les coûts correcteurs définis par les relations 87 et 88 où nous donnons les valeurs arrondies à l'entier.

a) Cas du torréfacteur à quatre modules

➤ Pour le module **1**, on a :

$$\text{Coût Couplage Interne Corrigé}(M_1) = \text{Coût Couplage Interne}(M_1) / MSI_i'(M_1)$$

$$= \frac{4250}{0,4}$$

$$= 10625 ;$$

$$\text{Coût Couplage Externe Corrigé}(M_1) = \text{Coût Couplage Externe}(M_1) \times MSI_e'(M_1)$$

$$= 135810 \times 0,0571$$

$$= 7755 ;$$

$$\text{Coût Couplage Corrigé}(M_1) = \text{Coût Couplage Interne Corrigé}(M_1) + \text{Coût Couplage Externe Corrigé}(M_1)$$

$$= 18380.$$

➤ Pour le module **2**, on a :

$$\text{Coût Couplage Interne Corrigé}(M_2) = \text{Coût Couplage Interne}(M_2) / MSI_i'(M_2)$$

$$= \frac{5760}{0,2666}$$

$$= 21605 ;$$

$$\text{Coût Couplage Externe Corrigé}(M_2) = \text{Coût Couplage Externe}(M_2) \times MSI_e'(M_2)$$

$$= 68464 \times 0,1083$$

$$= 7415 ;$$

$$\text{Coût Couplage Corrigé}(M_2) = \text{Coût Couplage Interne Corrigé}(M_2) + \text{Coût Couplage Externe Corrigé}(M_2)$$

$$= 29020.$$

➤ Pour le module **3**, on a :

$$\text{Coût Couplage Interne Corrigé}(M_3) = \text{Coût Couplage Interne}(M_3) / MSI'_i(M_3)$$

$$= \frac{3650}{0,4}$$

$$= 9125 ;$$

$$\text{Coût Couplage Externe Corrigé}(M_3) = \text{Coût Couplage Externe}(M_3) \times MSI'_e(M_3)$$

$$= 26244 \times 0,1212$$

$$= 3181 ;$$

$$\text{Coût Couplage Corrigé}(M_3) = \text{Coût Couplage Interne Corrigé}(M_3) + \text{Coût Couplage Externe Corrigé}(M_3)$$

$$= 12306.$$

➤ Pour le module **4**, on a :

$$\text{Coût Couplage Interne Corrigé}(M_4) = \text{Coût Couplage Interne}(M_4) / MSI'_i(M_4)$$

$$= \frac{666}{0,6666}$$

$$= 999 ;$$

$$\text{Coût Couplage Externe Corrigé}(M_4) = \text{Coût Couplage Externe}(M_4) \times MSI'_e(M_4)$$

$$= 0 \times 0,0625$$

$$= 0 ;$$

$$\text{Coût Couplage Corrigé}(M_4) = \text{Coût Couplage Interne Corrigé}(M_4) + \text{Coût Couplage Externe Corrigé}(M_4)$$

$$= 999.$$

Le coût de couplage total corrigé des quatre modules du torréfacteur de soja est donné par la relation 98.

$$\text{Coût Total Couplage Corrigé} = \sum_{i=1}^4 \text{Coût Couplage Corrigé}(M_i)$$

(98)

$$= 60705.$$

Le tableau 47 récapitule les coûts de couplage obtenu avec l’algorithme existant et le nouvel algorithme de clustering proposé sur les quatre modules du torréfacteur et ressort la réduction obtenue en coût.

Tableau 47:Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les quatre modules du torréfacteur

Modules du torréfacteur	Coût de couplage Algorithme existant	Coût de couplage Nouvel algorithme	Réduction réalisée
Module 1	140060	18380	80 %
Module 2	74224	29020	60 %
Module 3	29894	12306	58 %
Module 4	666	999	

Ce tableau montre qu’avec le nouvel algorithme de clustering proposé, on réalise une réduction en coûts de couplage conséquent (de plus de 50 %) avec chacun des modules 1, 2 et 3. Par contre, le module 4 qui est le module le moins contraignant pour la conception et par conséquent pour la maintenance a subi une augmentation de coût, soit de 50 %. Cela se traduit par le simple fait que ce dernier module de l’architecture de base n’a pas de coût de couplage externe.

Globalement l’algorithme proposé a réduit de 75 % le coût de couplage global des modules du torréfacteur de soja à quatre modules (en passant de 244844 à 60705). Donc l’architecture proposée, matérialisée par la DSM numérique est meilleure du point de vu maintenance.

La figure 64 présente l’évolution du coût de couplage généré par le nouvel algorithme en fonction de la taille des quatre modules du torréfacteur de soja.

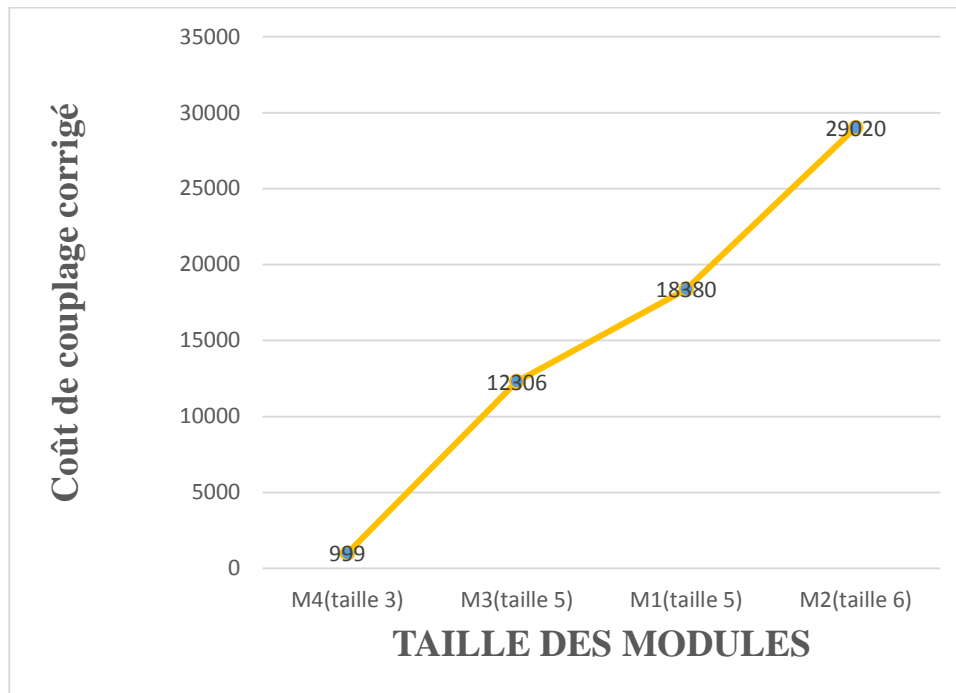


Figure 64: Evolution du coût de couplage corrigé en fonction de la taille des quatre modules

Avec l’algorithme existant, on a constaté que le module **1** a le coût de couplage le plus élevé et nettement supérieur à celui du module **2**. Pourtant du point de vue conception, le module **2** est le plus contraignant pour les concepteurs du fait de sa complexité avec une plus grande taille (soit de 6 composants), ce qui paraît donc poser un problème. L’algorithme que nous avons proposé montre à la figure **64** que le coût de couplage est une fonction croissante de la taille des modules, ce qui est d’ailleurs normal car la complexité de conception augmente avec la taille du produit à concevoir. D’un autre côté, les modules **1** et **3** sont de même taille (soit de 5 composants), il est donc difficile de percevoir l’ors d’un problème de maintenance de ces modules celui qui mettra le plus de temps d’intervention où qui suscitera le plus de coûts alloués pour la maintenance. Du coup avec l’algorithme que nous proposons, c’est fort de ce constat que des deux modules, le module **3** est le moins contraignant et nécessitant moins de ressources pour la maintenance.

b) Cas du torréfacteur à trois modules

Le tableau **48** récapitule les coûts de couplage obtenu avec l’algorithme existant et le nouvel algorithme de clustering proposé sur les trois modules du torréfacteur et ressort la réduction obtenue en coût.

Tableau 48:Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les trois modules du torréfacteur

Modules du torréfacteur	Coût de couplage Algorithme existant	Coût de couplage Nouvel algorithme	Réduction réalisée
Module 1	13190727	835970	93 %
Module 2	3334423	164564	95 %
Module 3	100245	435847	

L’algorithme proposé appliqué aux trois modules du torréfacteur a favorisé une réduction globale en coût de couplage de 91 %. Donc la nouvelle architecture est meilleure que celle générée par l’algorithme existant.

Les résultats obtenus avec l’algorithme proposé ne sont pas en accord avec ceux obtenus par l’algorithme existant, du fait qu’ici le coût de couplage augment avec la taille des modules.

c) Cas du torréfacteur à deux modules

Le tableau 49 récapitule les coûts de couplage obtenu avec l’algorithme existant et le nouvel algorithme de clustering proposé sur les deux modules du torréfacteur et ressort la réduction obtenue en coût.

Tableau 49:Synthèse des coûts générés par l’algorithme existant et le nouvel algorithme sur les deux modules du torréfacteur

Modules du torréfacteur	Coût de couplage Algorithme existant	Coût de couplage Nouvel algorithme	Réduction réalisée
Module 1	22565344	1511948	93,3 %
Module 2	10094	43887	

Avec l’algorithme proposé, on réalise une réduction en coût de couplage de 93,1%. L’architecture obtenue est donc meilleure que celle de la littérature. Nos résultats restent en accord avec ceux de la littérature, car le module M_2 demeure le module le moins contraignant pour la conception et pour la maintenance (car il est le plus facile à désassembler où à assembler).

IV.3.4 Identification des éléments intégrateurs du torréfacteur de soja

a) Cas du torréfacteur à quatre modules

En se référant à l’algorithme existant, on constate que les modules 1 et 2 nécessitent plus de travail des concepteurs, car ils ont un coût de couplage plus élevé. Cela se traduit évidemment par le fait qu’ils sont de plus grandes tailles. Dans l’optique de faciliter la maintenance de ces

modules, nous allons prendre en compte les éléments intégrateurs qui permettront de réduire la taille.

En considérant la DSM numérique du torréfacteur de soja décrite à la figure **59** , on remarque que des modules **1** et **2**, il ya deux éléments intégrateurs :le composant C_4 et le composant C_6 .

Le premier élément intégrateur (C_4) a 5 interactions, dont 4 avec le premier module et 1 avec le dernier module. Le deuxième élément intégrateur (C_6) quant à lui à 3 interactions ,dont 2 avec le module **1** et 1 avec le module **3**.Nous calculons donc l'indice de couplage des deux composants (défini à la relation 99) pour les identifier "automatiquement" comme éléments intégrateurs. Ainsi : avec le composant C_4 ,on a un IC de 71% et avec le composant C_6 ,on a un IC de 33%.Le composant C_4 est identifié automatiquement comme étant intégrateur, mais au regard du seuil fixé par Ghassen (2007), le composant C_6 ne saurait être considéré comme étant intégrateur. Etant donné que le composant C_6 ne réalise aucune interaction avec les composants du module auquel il appartient (module **2**), il devient donc tout à fait normal qu'il soit intégré dans un module où il interagit fortement avec les composants. Nous nous plaçons donc dans la même posture de Eric (2008) où nous laissons le choix à l'utilisateur de fixer un IC et nous classons de ce fait manuellement le composant C_6 comme étant un élément intégrateur.

En appliquant l'algorithme de clustering proposé avec les procédures spécifiques aux éléments intégrateurs, nous obtenons l'architecture présentée à la figure **65**.

	C_1	C_2	C_3	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_4
C_1	0													1					1
C_2		0			1														1
C_3			0												1				1
C_5				0	1														1
C_6		1		1	0										1				
C_7						0			1							1			
C_8							0	1		1									
C_9							1	0	1										
C_{10}						1		1	0										
C_{11}							1			0									
C_{12}											0	1						1	
C_{13}											1	0	1						
C_{14}												1	0	1	1				
C_{15}	1												1	0					
C_{16}			1		1								1		0				
C_{17}						1										0	1		1
C_{18}																1	0	1	
C_{19}										1							1	0	
C_4	1	1	1	1												1			0

Figure 65: Architecture du torréfacteur de soja avec les éléments intégrateurs

Cette architecture montre que l'algorithme a intégré l'élément C_4 dans le dernier module et a intégré l'élément C_6 dans le premier module avec lequel il interagissait prioritairement. Nous allons à présent comparer l'architecture obtenue avec les éléments intégrateurs et celle obtenue en appliquant l'algorithme de clustering sans les éléments intégrateurs. En appliquant les relations 87 et 88 qui donnent les coefficients correcteurs des modules du torréfacteur de soja, on obtient le tableau 50.

Tableau 50: Coefficients correcteur du torréfacteur de soja avec prise en compte des éléments intégrateurs

Modules du torréfacteur	MSI'_i	MSI'_e	MSI'
Module 1	0,2	0,1	0,1
Module 2	0,4	0,0222	0,3778
Module 3	0,4	0,1166	0,2834
Module 4	0,5	0,1	0,4

Le tableau 50 montre que le module 4 reste toujours le module qui pénalise le plus les interactions externes. Par contre, avec la prise en compte des intégrateurs, c'est le module 1 qui devient cette fois le plus difficile à maintenir.

La figure 66 montre l'évolution du MSI avec et sans la prise en compte des intégrateurs.

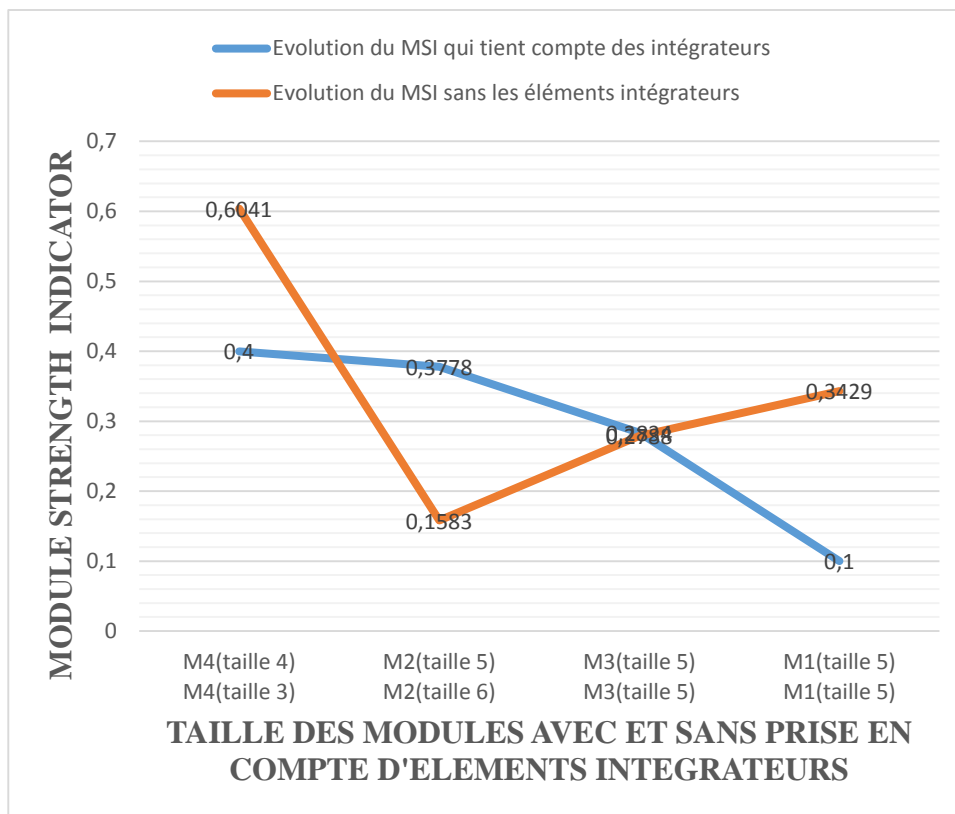


Figure 66: Correction du MSI avec et sans les éléments intégrateurs

La figure 66 montre que l'ajout d'un composant dans le module 4 suite aux éléments intégrateurs, conduit à la réduction du MSI de 33%, ce qui favorise la création d'un module moins dense. Par contre, l'économie d'échelle observée au niveau du module 2 (soit avec une réduction de 1 composant) conduit à une hausse considérable du MSI (soit de 138%), ce qui favorise plutôt la création d'un module plus dense. Le MSI du module 3 a légèrement fluctué, ceci se traduit par le fait que les éléments intégrateurs n'ont pas affecté la taille de ce module. Le module 1 bien qu'ayant conservé sa taille, a subi une réduction considérable de son MSI. Ça s'explique par le fait que ce module en cédant un élément intégrateur et ayant reçu un autre a considérablement augmenté ses interactions externes.

En appliquant les relations 87, 88 et 89, les résultats obtenus en coûts de couplage des modules du torréfacteur sont donnés par le tableau 51.

Tableau 51: Coûts de couplage corrigé du torréfacteur de soja avec prise en compte des éléments intégrateurs

Modules du torréfacteur	Coût de couplage interne	Coût de couplage externe	Coût de couplage corrigé
Module 1	11250	3581	14831
Module 2	14400	1520	15920
Module 3	9125	3060	12185
Module 4	1332	0	1332

Le tableau 51 montre que le module 4 reste le module le moins contraignant pour la conception et le module 2, le module qui suscite le plus de travail et de ressources des concepteurs.

Le tableau 52 récapitule les coûts de couplage obtenu avec l'algorithme de clustering proposé avec et sans éléments intégrateurs et ressort la réduction obtenue en coût.

Tableau 52: Synthèse des coûts générés par l'algorithme de clustering proposé avec et sans éléments intégrateurs

Modules du torréfacteur	Coût de couplage sans éléments intégrateurs	Coût de couplage avec éléments intégrateurs	Réduction réalisée
Module 1	18380	14831	19, 87 %
Module 2	29020	15920	45, 14 %
Module 3	12306	12185	0,98 %
Module 4	999	1332	

D'après le tableau ci-dessus, en considérant les éléments intégrateurs, on observe une réduction conséquente en coût de couplage des modules 1 et 2 et une légère réduction de coûts avec le module 3, ce qui se traduit par le fait que les éléments intégrateurs n'ont pas affecté la taille du module. D'un autre côté, le module 4 a subi une élévation de coûts parce qu'il, n'a pas de coût de couplage externe.

Objectivement, l'architecture obtenue est meilleure que la précédente du point de vue « mesure de la modularité » avec un coût de 44268, soit réalisant une réduction globale de 27%.

La reconfiguration du torréfacteur de soja grâce à la nouvelle architecture obtenue avec la prise en compte des éléments intégrateurs est donnée à l'Annexe 7.

La prise en compte des éléments intégrateurs a conduit à substituer la courroie par un motoréducteur et son accouplement (Annexe 7) et en allant dans *propriétés de masse* dans le logiciel de Conception SolidWorks (version 2016) utilisée pour la conception de notre équipement, nous avons obtenu les résultats consignés dans le tableau 53.

Tableau 53: Comparaison des configurations obtenues avant et après implémentation de l'algorithme de clustering amélioré

Torréfacteur de soja	Masse	Volume	Superficie
Ancienne configuration	156,73kg	48,8 dm^3	19,28 m^2
Nouvelle configuration	148,02kg	47,9 dm^3	17,79 m^2

Au regard de ce tableau, la nouvelle configuration du torréfacteur (annexe 12) a conduit à réduire respectivement la masse, le volume et la superficie. Le torréfacteur reconfiguré est donc moins encombrant. D'un autre côté, le réducteur utilisé est facile à installer et ne nécessite pas de calculs comme dans le système poulie-courroie. Le torréfacteur reconfiguré est donc plus facile à maintenir.

b) Cas du torréfacteur à trois modules

Partant de la figure 60, en appliquant le raisonnement déployé pour le cas du torréfacteur à 4 modules, on constate qu'aucun élément intégrateur n'est pris en compte dans l'application de l'algorithme proposé sur le torréfacteur à trois modules. Donc l'architecture obtenue ne saurait être reconfigurée.

c) Cas du torréfacteur à deux modules

Partant de la figure 61, en appliquant le raisonnement déployé pour le cas du torréfacteur à 4 modules, on constate également qu'aucun élément intégrateur n'est pris en compte dans l'application de l'algorithme proposé sur le torréfacteur à deux modules. Donc l'architecture obtenue ne saurait être reconfigurée.

IV.3.5 Etude comparative du coût de couplage en fonction du nombre de modules

Le tableau 54 donne la variation du coût de couplage du torréfacteur avec le nombre de modules dont il est constitué.

Tableau 54: Variation du coût de couplage du torréfacteur avec le nombre de modules

Nombre de Modules du torréfacteur	Coût de couplage Avec l'algorithme existant	Coût de couplage avec l'algorithme proposé
4 Modules	244844	44268
3 Modules	16625395	1436381
2 Modules	22575438	1555835

D'après le tableau 54, on constate qu'avec l'algorithme proposé, le coût global de couplage du torréfacteur augmente lorsque le nombre de modules diminue. L'algorithme proposé est donc en accord avec celui de la littérature, du fait qu'on fait le même constat précédent. Cela se traduit évidemment par le fait que plus on a de modules, moins les modules ont des composants, ce qui réduit drastiquement la complexité de l'équipement. D'où il est avantageux d'avoir un torréfacteur à quatre modules, que de l'avoir à trois ou à deux modules.

CONCLUSION

Cette partie a été consacrée à l'implémentation du nouvel algorithme de clustering proposé au chapitre 3, dans l'optique de réduire les coûts de couplages générés par l'algorithme de clustering existant et de définir une nouvelle architecture et reconfiguration du produit. Afin de valider la démarche proposée pour notre algorithme, nous avons fait une application sur le torréfacteur de soja qui a également servi d'étude de cas lorsque nous déployions l'algorithme existant. Nous avons expérimenté les différentes éventualités où le torréfacteur aurait quatre modules, trois modules ou deux modules. Nous avons constaté une non prise en compte des éléments intégrateurs, si le torréfacteur possède trois ou deux modules, dans ce cas il est donc impossible de reconfigurer le torréfacteur de soja. Pour le torréfacteur ayant quatre modules (le seul cas de figure permettant la prise en compte des éléments intégrateurs et permettant une reconfiguration du torréfacteur de soja), il en ressort que l'algorithme proposé sans tenir compte des éléments intégrateurs a donné une architecture qui a généré une réduction globale en coûts de 75 % sur l'algorithme existant. En prenant en compte les éléments intégrateurs, nous avons défini une autre architecture du torréfacteur de soja qui se trouve encore meilleure que celle donnée par le nouvel algorithme qui ne prend pas en compte les éléments intégrateurs. La nouvelle architecture a favorisé une économie d'échelle du module 2, en le réduisant d'un composant, le rendant ainsi moins complexe et plus facile à maintenir. Ce module a donc subi une réduction en coût de 45% et globalement on a réalisé une réduction de 27% en coûts des modules du torréfacteur de soja. L'algorithme proposé a permis de réduire les coûts de couplage

d'environ 82% et le torréfacteur reconfiguré obtenu est moins encombrant et plus facile à maintenir.

CONCLUSION GENERALE ET PERSPECTIVES

L'objectif général de ce travail de recherche était d'améliorer la maintenance par la conception de la modularité. L'outil proposé qui s'appuie sur l'algorithme de clustering existant a été validé par une application sur un torréfacteur de soja. L'atteinte de cet objectif général a nécessité une méthodologie.

La première étape de la méthodologie consiste à évaluer la solution de conception la plus respectueuse des contraintes de désassemblage. L'idée ici est d'apprécier la pertinence du choix des paramètres de désassemblage au moyen d'un critère de conception, l'indice de fixation faciles. Ce critère de conception fut choisi parmi tant d'autre du fait qu'il est le seul qui prend en compte la complexité du mouvement de désassemblage d'un produit. Néanmoins, ce critère développé dans la littérature présente des limites dans la mesure où il ne prend pas en compte les autres paramètres de désassemblage non-destructifs qui influences considérablement sur le mouvement de désassemblage. Un modèle (indice de fixations faciles amélioré) (intégrant tous les paramètres de désassemblage non-destructifs qui représentent au mieux toutes les facettes du désassemblage) a été proposé et comparativement à celui proposé dans la littérature, l'application sur un torréfacteur de soja montre que le nouvel indice de fixations faciles proposé génère des solutions de conception satisfaisantes. Les paramètres retenus par le modèle proposé ont permis l'exécution de la deuxième étape de la méthodologie.

A la seconde étape de la méthodologie, on ressort les fonctionnalités de l'algorithme de clustering existant, donc la finalité est de ressortir la topologie des modules au moyen de l'arbre de désassemblage complet, d'évaluer le degré de modularité des modules au moyen du MSI et d'évaluer le coût de couplage des modules du produit. Il ressort que la connaissance de la gamme de démontage et de la topologie d'un système facilite la maintenance en définissant les priorité d'intervention. Ensuite, nous présentons les limites de l'algorithme existant et enfin nous proposons un nouvel algorithme de clustering dans l'optique de réduire les coûts de couplages générés par l'algorithme de clustering existant, de générer une nouvelle architecture et de reconfigurer le produit. Plusieurs corrections de l'algorithme existant ont été faite telles que la définition d'une DSM numérique permettant de ressortir les coefficients correcteurs du coût de couplage, la formulation d'une fonction objective (le coût de couplage) permettant d'évaluer le coût de couplage de la nouvelle architecture proposée (matérialisée par la DSM numérique) et la définition des éléments intégrateurs afin de faire une économie d'échelle des modules de grandes tailles qui suscitent plus de contraintes et de ressources des concepteurs .

La troisième étape de la méthodologie consiste à valider l'algorithme de clustering proposé dans l'optique de réduire les coûts de couplage générés par l'algorithme de clustering

existant. En apportant plusieurs corrections de l'algorithme existant sans la prise en compte des éléments intégrateurs, l'algorithme proposé génère une architecture avec une réduction globale en coûts de 75% sur l'algorithme existant (cas du torréfacteur à quatre modules). D'autre part avec la prise en compte des éléments intégrateurs (la seule éventualité permettant la prise en compte des éléments intégrateurs étant le cas où le torréfacteur de soja peut se subdiviser en quatre modules), on obtient une nouvelle architecture qui favorise une économie d'échelle du module 2 (avec une réduction d'un composant), le rendant ainsi moins complexe et plus aisé à maintenir. On réalise une réduction de 27% avec la nouvelle architecture qui prend en compte les éléments intégrateurs sur celle qui ne prend pas en compte les éléments intégrateurs. Globalement l'algorithme proposé a permis de réduire les coûts de couplage d'environ 82% et le torréfacteur reconfiguré est moins encombrant et plus facile à maintenir, du fait de la facilité de son installation.

En guise de perspectives à ce travail de recherche, on pourrait :

- ✚ Enoncer des règles plus restrictives quant à la construction des DSM ;
- ✚ Appliquer l'algorithme sur des équipements plus complexes ;
- ✚ Etudier la robustesse des architectures simulées ;
- ✚ Développer une plateforme logicielle à partir des algorithmes et méthodes développés.

REFERENCES BIBLIORAPHIQUES

Addouche, S. (2003). Contribution à une démarche de conception optimisée des processus de désassemblage. Thèse de doctorat. Franche-Comté : Université de Franche-Comté, 204 p.

AFNOR. (1988). Recueil de normes françaises : Fiabilité, Maintenabilité et Disponibilité. Ed Afnor-Ute.

Agard, B et Tollenaere, M. (2002). Conception d'assemblages pour la customisation de masse. *Mécanique & Industrie*, 3 (2), 113 – 119.

Ahouansou, R et Singbo, A. (2010). Evaluation des performances techniques de deux torréfacteurs d'amande de karité (*Vitellaria paradoxa*). *Bulletin de la recherche Agronomique du Bénin*, 68, 20–28.

Ait-Kadi, D., Diallo, C and Chelbi, A. (2004). A Joint Spare-Provisioning and Preventive Maintenance Strategy for Availability Maximization under Limited Resources. *Proceedings of the 7th Probabilistic Safety Assessment and Management Conference*, Berlin, 5, 2840–2846.

Ait-Kadi, D. (2007). Fiabilité des systèmes. Notes de cours, département de génie mécanique, Université Laval.

Alexander, C. (1971). De la synthèse de la forme : essai. Paris: Dunod.

Alven, V. (1964). Reliability engineering. Englewood Cliff, New Jersey: Prentice- Hall.

Alves De Macedo, L. (2017). Torréfaction de biomasse lignocellulosique : effet catalytique du potassium sur les espèces condensables. Thèse de Doctorat, Université de Lorraine ,168p.

Al-Widyan, K and Angeles, J. (2005). A Model- Based Formulation of Robust Design, *Journal of Mechanical Design* ,127 (3), 388 – 396.

ARMP-7. (2003). Reliability and Maintainability. NATO R & M Terminology, Ministry of Defence, UK, 1 (2).

Baldwin, C and Clark, K. (1997). Managing in an age of modularity. *Harvard Business Review*, 75, 84–93.

Baldwin, C and Clark, K. (2000). Design Rules: The Power of Modularity. *The Academy of Management Review*, 26 (1) :130–133.

Baron, C. (2005). L'évaluation dans la conception système : vers une aide à la conduite de projet. HDR, Institut National Polytechnique de Toulouse.

Beauville, A. (2020). Démarche de conception d'un système de production industriel reconfigurable, dans un contexte de fortes variations de marché en volume et en diversité. Thèse de doctorat, Paris, HESAM,187 p.

Bellaouar, A et Beleulmi, S. (2014). Fiabilité, Maintenabilité et Disponibilité. Support de cours. Université Constantine 1,106 pages.

Beyer, H-G and Sendho, B. (2007). Robust Optimization - A Comprehensive Survey. Computer Methods in Applied Mechanics and Engineering, 196 (33-34), 3190–3218.

Blackenfelt, M. (2000). Modularization by Relational Matrices - a Method for the Consideration of Strategic and Functional Aspects. In: Riithahuhta A., Pulkkinen A. (eds) Design for Configuration. Springer, Berlin, Heidelberg, 134-152.

Blackenfelt, M and Sellgren, U. (2000). Design of robust interfaces in modular products. Proceedings of the 2000 ASME Design Engineering Technical Conferences, Baltimore, Maryland, 1-9.

Blackenfelt, M. (2001). Managing complexity by product modularization. Doctoral Thesis, Department of Machine Design, Royal Institute of Technology, Stockholm.

Bohm, M. R and Stone, R. B. (2004). Representing Functionality to Support Reuse: Conceptual and Supporting Functions. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 411–419.

Boucly, F et Ogus, A. (1987). Le management de la maintenance, Afnor, Paris.

Bourjault, A. (1984). Contribution d'une approche méthodologique de l'assemblage automatisé : élaboration automatique des séquences opératoires. Thèse d'État. Université de Franche-Comté.

Boyd, J.A. (1992). Allocation of reliability requirements: a new approach. Annual Reliability and Maintainability Symposium 1992 Proceedings, 5– 6.

Braha, D. (2002). Partitioning Tasks to Product Development Teams. Second International Conference on Axiomatic Design, Cambridge , 1–11.

Browning, T.R. (1998). Use of dependency structure matrices for product development cycle time reduction. Proceedings of the 5.fth ISPE International Conference on Concurrent Engineering: Research and Applications, Tokyo, Japan,1–8.

Browning, T.R. (1999). Designing system development projects for organizational integration. Journal of Systems Engineering, 2(4), 217–225.

Browning T.R. (2001). Applying the design structure matrix to system decomposition and integration problems: A Review and New Directions. *IEEE Transactions on Engineering Management*, 48 (3) , 292–306.

Browning, T.R and Eppinger, S.D. (2002). Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. *IEEE Transactions on Engineering Management*, 49(4) , 443–458.

Carrascosa, M., Eppinger, S. D and Whitney, D. E. (1998). Using the design structure matrix to estimate product development time. *Design Automation Conference*, Atlanta, GA.

Carré, P. (2020). Transformation du soja à la ferme, technologies disponibles. Edition ITAB, 49 p.

Cerny, V (1985). Thermodynamical Approach to the Traveling Salesman Problem : An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications* , 45 , 41–51.

Chen, R, Navin-Chandra, D and Prinz, F.B. (1993). Product design for recyclability: A cost benefit model and its application. *Proceedings of the 1993 IEEE International Symposium on Electronics and the Environment*, Arlington, VA, USA ,178–183.

Chen, S.J and Lin, L. (2003). Decomposition of interdependent task group for concurrent engineering. *Computers & Industrial Engineering*, 44, 435 – 459.

Chen, K.M and Liu, R.J. (2005). Interface strategies in modular product innovation, *Technovation*, 25 (7), 771–782.

Cho, S.H and Eppinger, S.D. (2001). Product Development Process Modeling Using Advanced Simulation. *Design Engineering Technical Conferences Pittsburgh, Pennsylvania*,1–10.

Clarkson, P.J., Simons, C.S and Eckert, C.M. (2004). Predicting change propagation in complex design. *Journal of Mechanical Design*, 126 (5), 765–797.

Collette, Y. and Siarry, P. (2002). Optimisation multiobjectif. Editions Eyrolles.

Cotaina, N., Matos, F., Chabrol, J and Djeapragache, D. (2000). Study of Existing Reliability Centered Maintenance (RCM) Approaches Used in Different Industries. Technical Report Number: FIM/110.1/DATSI/00, 98 p.

Dahmus, J., Gonzalez-Zugasti, J and Otto, K. (2001). Modular Product Architecture. *Design Studies*, 22(5), 409–424.

- Das, I. and Dennis, J. E. (1998).** Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8(3), 631–657.
- Das, S.K., Yedlarajiah, P and Narendra, R. (2000).** An approach for estimating the end-of-life product disassembly effort and cost. *International Journal of Production Research*, 38 (3), 657–673.
- Deb, K., Pratap, A., Agarwal, S and Meyarivan, T. (2002).** A fast and elitist multiobjective genetic algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation* ,6 (2), 182–197.
- Desai, A and Mital, A. (2003).** Evaluation of disassemblability to enable design for disassembly in mass production. *International Journal of Industrial Ergonomics*, 32 (4), 265–281.
- de Oliveira, L. S and Saramago, S. F. P. (2010).** Multiobjective optimization techniques applied to engineering problems. *Journal of the Brazilian society of mechanical sciences and engineering* 32 (1), 94–105.
- de Rocquigny, E., Devictor, N and Tarantola, S. (2008).** *Uncertainty in Industrial Practice: A Guide to Quantitative Uncertainty Management.* John Wiley & Sons, Ltd. ISBN :978-0-470-99447-4.
- De Rosnay, J. (1975).** *Le microscope*, Editions du Seuil.
- Despujols, A. (2004).** *Optimisation de la maintenance par la fiabilité (OMF).* MT9310, Techniques de l'ingénieur.
- Dhillon, B.S. (2002).** *Engineering Maintenance : A Modern Approach.* CRC Press, ISBN 1-58716-142-7.
- Dowie, T. and Simon, M (1994).** *Guidelines for designing for disassembly and recycling.* Department of Mechanical Engineering, Design and Manufacture, the Manchester Metropolitan University.
- Dowie, T. (1995).** *A Disassembly Planning and Optimisation Methodology for Design.* PhD thesis. Manchester Metropolitan University, Manchester.
- Dubois, D., Lubiano, M. A., Prade, H., Gil, M. A., Grzegorzewski, P and Hryniewicz, O. (2008).** Soft methods for handling variability and imprecision. *Conferences proceedings Part of the Advances in Soft Computing Book Series*, ISBN:978-3-540-85026-7, 48.

- Duflou, J.R., Seliger, G., Kara, S., Umeda, Y., Ometto, A and Willems, B. (2008).** Efficiency of product disassembly: A case-based study. *CIRP Annals - Manufacturing Technology*, 57, 583–600.
- Ebeling, C. E. (1997).** An Introduction to Reliability and Maintainability Engineering. Waveland Press, Inc. ISBN 13:978-1-4786-3734-9, 658 p.
- EC 60300-3-11. (1999).** Dependability management - Part 3-11: Application guide -Reliability centred maintenance. International Electrotechnical Commission, 2, 1–13.
- Eppinger, S. D., Whitney, D. E., Smith, R. P and Gebala, D. A. (1994).** A Model-Based Method for Organizing Tasks in Product Development. *Research in Engineering Design*, 6 (1), 1–13.
- Ericsson, A and Erixon, G. (1999).** Controlling design variants: Modular product platforms. Society of Manufacturing Engineers, 145 p.
- Eric, B. (2008).** Contributions à l'instrumentation du métier d'architecte système : de l'architecture modulaire du produit à l'organisation du système de conception. Habilitation à Diriger des Recherches. Université de Franche-Comté, UFR Sciences et Techniques, Institut FEMTO-ST / Département AS2M, 206 p.
- Erixon, G. (1998).** Modular Function Deployment – A method for product modularization. PhD thesis, Royal Institute of Technology, Stockholm, Sweden.
- Ethiraj, S.K and Levinthal, D.A. (2004).** Modularity and innovation in complex systems. *Management science* 50 (2): 159 –173.
- Failli, F and Dini, G. (2001).** Optimization of disassembly sequences for recycling of end-of-life products by using a colony of ant-like agents. *Engineering of Intelligent Systems*, 2070, 632–639.
- Fernandez, C. (1998).** Integration analysis of product architecture to support effective team co-location. Thesis, Massachusetts Institute of Technology MIT, USA.
- Frédéric, D. (2010).** Conception intégrée et gestion d'informations techniques : application à l'ingénierie du produit et de sa séquence d'assemblage. Thèse de doctorat. Université de Technologie de Belfort-Montbéliard.
- Garud, R and Kumaraswamy, A. (1995).** Technological and organizational designs for realizing economies of substitution. *Strategic Management Journal* 16 (1) , 93 –109.

Ghassen, H. (2007). Vers une conception conjointe des architectures du produit et de l'organisation du projet dans le cadre de l'Ingénierie Système. Thèse de doctorat. Université de Franche-Comté, UFR Sciences et Techniques, Ecole Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques, 249 pages.

Ghazilla, R., Taha, Z., Yusoff, S., Rashid, S.H and Sakundarini, N. (2014). Development of decision support system for fastener selection in product recovery oriented design. The International Journal of Advanced Manufacturing Technology, 70 (5 - 8), 1403–1413.

Gittard, C. (2009). Application de l'approche système dans l'analyse de la fiabilité lors des phases amont de la conception de produits innovants. Mémoire d'ingénieur, Centrale Paris.

Giudice, F and Fargione, G. (2007). Disassembly planning of mechanical systems for service and recovery: A genetic algorithm based approach. Journal of Intelligent Manufacturing, 18 (3), 313 – 329.

Guivarch, A.D. (2003). Concurrent Process Mapping, Organizations, Project and Knowledge Management in Large-Scale Product Development Projects Using the Design Structure Matrix Method. Massachusetts Institute of technology, 149 – 152.

Gungor, A and Gupta, S. (2001). Disassembly sequence plan generation using a branch-and-bound algorithm. International Journal of Production Research, 39 (3) , 481 – 509.

Güngör, A. (2006). Evaluation of connection types in design for disassembly (DFD) using analytic network process. Computers and Industrial Engineering, 50, 35 – 54.

Guo, F and Gershenson, J. K. (2004). A comparison of modular product design methods on improvement and iteration. In Proc of ASME Design Engineering Technical Conferences, 261 – 269.

Haldar, A and Mahadevan, S. (2000). Probability, reliability, and statistical methods in engineering design. Wiley New York, ISBN:978-0-471-331119-3, 1.

Haoues, N., Froelich, D and Zwolinski, P. (2004). Disassembly for valorization in conceptual design. In Environmentally Conscious Manufacturing IV, 5583, 31–42.

Hölttä, K and Salonen, M. (2003). Comparing three different modularity methods. Proc. Of DETC'03, Design Engineering Technical Conferences, ASME 2003, Chicago, USA.

Hölttä-Otto, K. (2005). Modular product platform design. PhD, Helsinki University of Technology, Espoo, Finland, ISBN 951-22-7766-2.

Höltkä, K., Suh, E and Weck, O. (2005). Tradeoff Between Modularity and Performance for Engineering Systems and Products. International Conference On Engineering Design Iced 05 Melbourne, 1 – 14.

Hsuan Mikkola, J and Skjøtt-Larsen, T. (2004). Supply-chain integration: implications for mass customization, modularization and postponement strategies. Production Planning and Control, 15 (4): 352 – 361.

Hsu, H and Lin, G. (2002). Quantitative measurement of component accessibility and product assemblability for design for assembly application. Robotics and Computer Integrated Manufacturing, 18, 13 – 27.

Huang, C.C and Kusiak, A. (1998). Modularity in design of products and systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 28 (1) , 66 – 77.

Huang, C. (2000). Overview of modular product development. Proc. Natl. Sci. Council, ROC(A), 24 (3), 149 – 165.

Hwai-En, Tseng., Chien-Chen, Chang and Jia-Diann, Li. (2008). Modular design to support green life-cycle engineering. Science Direct, 34 (4Tapez une équation ici.), 2524 – 2537.

Idicula, J. (1995). Planning for Concurrent Engineering. Thesis draft, Nanyang Technology University.

IEC 60050-191. (1990). International Electrotechnical Vocabulary – Chapter 191: Dependability and quality of service. International Standard, International Electrotechnical Commission, IEC.

IEC 61703. (2001). Mathematical Expressions for Reliability, Availability, Maintainability and Maintenance Support Terms. International Standard, International Electrotechnical Commission, IEC, Geneva.

IEEE 1220. (1998). Standard for application and Management of the Systems Engineering Process. The Institute of Electrical and Electronics Engineers, 1 – 11.

Ireson, W., Coombs, C.F and Moss, R.Y. (1995). Handbook of Reliability Engineering and Management. McGraw-Hill Companies, ISBN 0-07-012750-6, 2.

Jabbour, T and Mascle, C. (1998). A Database for the representation of assembly features in mechanical products. International Journal of Computational Geometry and Applications, 8 (5-6), 483 – 507.

- Jeandin, T. (2015).** Méthode de choix des attachements pour la fin de vie des produits. Maîtrise des sciences appliquées, Université de Montréal, Ecole Polytechnique de Montréal, 122 p.
- Jiao, J., Simpson, T.W and Siddique, Z. (2007).** Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, Special issue on Product family design and platform-based product development, 18 (1), 5 – 29.
- Jose, A and Tollenaere, M. (2005).** Modular and platform methods for product family design: literature analysis. *Journal of intelligent Manufacturing*, 16, 371 – 390.
- Kececioglu, D. (1991).** *Reliability Engineering Handbook*. PTR Prentice Hall, 2.
- Keller, R., Eckert, C.M and Clarkson, P.J. (2005).** Visualizing change propagation. Proc. 15th Int. Conf. On Engineering Design, ICED'05, Melbourne, Australia.
- Kim, I. Y and Weck, O. L. D. (2005).** Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29 (2), 149 –158.
- Kirkpatrick, S., Gelatt, C.D and Vecchi, M.P. (1983).** Optimization by simulated annealing. *Science*, New series, 220 (4598) , 671–680
- Kongar, E and Gupta, S. (2001).** Genetic algorithm for disassembly process planning. In *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing*, 54–62.
- Kongar, E and Gupta, S. (2006).** Disassembly sequencing using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 30 (5 – 6), 497 – 506.
- Kroll, E and Carver, B.S. (1999).** Disassembly analysis through time estimation and other metrics. *Robotics and Computer Integrated Manufacturing*, 15 (3), 191 – 200.
- Kuo, T., Zhang, H. C and Huang, S. (2000).** Disassembly analysis for electromechanical products: a graph-based heuristic approach. *International Journal of Production Research*, 38 (5), 993 – 1007. Tapez une équation ici.
- Kusiak, A and Szczerbicki, E. (1993).** Transformation from Conceptual to Embodiment Design. *IEEE Transactions*, 25(4), 6 – 12.
- Lambert, A and Gupta, S. (2005).** Disassembly modelling for assembly, maintenance, reuse, and recycling. CRC Press, 1 – 19.

- Langlois, R.N. (2002).** Modularity in technology and organization. *Journal of Economic Behavior and Organization*, 49, 19 – 37.
- Larses, O and Blackenfelt, M. (2003).** Relational reasoning supported by quantitative methods for product Modularization. *Int. Conf. On Engineering Design, ICED 03 Stockholm*.
- Larses, O. (2005).** Applying quantitative methods for architecture design of embedded automotive systems. *Proc. INCOSE International Symposium*. Rochester, NY.
- Lee, Y. Q and Kumana, S. (1992).** Individual and group disassembly sequence generation through freedom and interference spaces. *Journal of Design and Manufacturing*, 2, 143–154.
- Lehman, J. S., Santner, T. J and Notz, I. (2004).** Designing computer experiments to determine robust control variables. *Statistica Sinica*, 14 (2), 571–590.
- Le Moigne, J.L. (1977).** La théorie du système général, Théorie de la modélisation. *Revue Française de Sociologie*, 19 (2), 361 p.
- Li, J. R., Khoo, L. P and Tor, S. B. (2002).** A Novel Representation Scheme for Disassembly Sequence Planning. *International Journal Advanced Manufacturing Technology*, 20 (8), 621–630.
- Lyonnet, P et Bouit, B. (1998).** Contraintes de maintenance dans la Conception de produits mécaniques, méthodes, modèles et outils. Paris: Hermès.
- Marler, R. T and Arora, J. S. (2010).** The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41 (6), 853 – 862.
- McCord, K. R and Eppinger, S. D. (1993).** Managing the Integration Problem in Concurrent Engineering, M.I.T. Sloan School of Management, Cambridge, MA, Working Paper no.3594.
- Meinadier, J.P. (2002).** Le métier d'intégration de systèmes. Hermès Sciences Publications.
- Micaëlli, J.P et Forest, J. (2003).** Artificialisme, Introduction à une théorie de la conception. Presses Polytechniques et Universitaires Romandes (PPUR), ISBN 2-88074 - 540.
- Miettinen, K. (2001).** Some methods for nonlinear multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, Springer, 1–20.
- Millet, D., Brissaud, D and Zwolinski, P. (2007).** Design for reuse, remanufacture and recycling; an alternative to design for landfill. *International Journal of Environmentally Conscious Design and Manufacture*, G. Bertolucci guest editors 13 (3–4).

- Military Handbook – 217F. (1991).** Reliability Prediction of Electronic Equipment. Notice 2, U. S. Department of Defense.
- Military Handbook – 470A. (1997).** Designing and Developing Maintainable products and Systems». U.S. Department of Defense, Washington, DC.
- Military Handbook – 338B. (1998).** Electronic Reliability Design Handbook. U.S. Department of Defense.
- Military Standard – 810F. (2000).** Environmental Engineering Considerations and Laboratory Tests. U.S. Department of Defense.
- Monchy, F. (1987).** La fonction maintenance : formation à la gestion de la maintenance industrielle. Collection technologies, Masson, Paris.
- Moore, K. E., Gungor, A and Gupta, S. (2001).** Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships. *European Journal of Operational Research*, 135 (2), 428 – 449.
- Morelli, M.D., Eppinger, S. D and Gulati, R.K. (1995).** Predicting Technical Communication in Product Development Organizations. *IEEE Transactions on Engineering Management*, 42 (3).
- Morin, E. (1977).** La Méthode, tome 1 : La Nature de la Nature. Paris : Seuil, 399 p.
- Morin, E. (1992).** Introduction à la pensée complexe. Paris : Seuil.
- Mtopi, F.B. (2006).** Contribution à une méthodologie de conception modulaire : Modélisation de la diversité dans les familles de produits. Thèse de l'Université de Franche- Comté.
- Murayama, T., Oba, F., Abe, S and Yamamichi, Y. (2001).** Disassembly Sequence Generation using Information Entropy and Heuristics for Component Replacement. *Proceedings of the IEEE International Symposium*, 208 – 213.
- Nakayama, H and Sewerage, Y. (1984).** Satisficing trade-off method for multiobjective programming. In *Interactive decision analysis*, Springer, 113–122.
- NASA. (2000).** Reliability Centered Maintenance Guide for Facilities and Collateral Equipment. Department of Defense, Washington, 472 p.
- Newcomb, P., Bras, B and Rosen, D. (1998).** Implications of modularity on product design for the life cycle. *Journal of Mechanical Design*, 120, 483 – 490.
- NSWC-07. (2007).** Handbook of Reliability Prediction Procedures for Mechanical Equipment. Naval Surface Warfare Center, US

- Nzie, W. (2006).** Intégration de la maintenance en conception : application à un équipement agroalimentaire. Thèse de doctorat. Université de Besançon.
- Oosterman, B. (2001).** Improving Product Development Projects by Matching Product Architecture and Organization. PhD thesis, Gröning University, The Netherlands, ISBN: 90-72591-99-2.
- Orton, J et Weick, K. (1990).** Loosely Coupled Systems: A Reconceptualization. *Academy of Management Review* 15 (2): 203–223.
- Osyczka, A. (1981).** An approach to multicriteria on optimization for structural design. Technical report, DTIC Document.
- Otto, K and Wood, K. (2001).** Product design: techniques in reverse engineering and new product development. Prentice Hall, Upper Saddle River, NJ.
- Padulo, M and Guenov, M. D. (2011).** Worst-case robust design optimization under distributional assumptions. *International Journal for Numerical Methods in Engineering*, 88 (8) ,797–816.
- Pagès, A et Gondran, M. (1980).** Fiabilité des systèmes. Edition Eyrolles, Paris 5ème.
- Pahl, G and Beitz, W. (1996).** Engineering Design: A Systematic Approach, Springer Verlag, London, 2nd Edition.
- Pak, K.G and Sodhi, R. (2002).** Destructive disassembly of bolts and screws by impact fracture. *Journal of Manufacturing Systems*, 21 (4), 316 – 324.
- Pascal, V. (2010).** Génération d'indicateurs de maintenance par une approche semi-paramétrique et par une approche markovienne. Thèse de doctorat. Université d'Orléans.
- Phadke, M. (1989).** Quality Engineering Using Robust Design. PTR Prentice- Hall Inc, Englewood Cliffs, New Jersey.
- Pil, F. K and Cohen, S.K. (2006).** Modularity: Implications for imitation, innovation, and sustained advantage. *Academy of Management Review*, 31 (4), 995 – 1011.
- Pimmler, T and Eppinger, S. (1994).** Integration Analysis of Product Decompositions. *Proc. ASME Design Theory and Method Conference (DTM'94)*, 68, 343 – 351.
- Probst, G et Ulrich, H. (1989).** Pensée globale et management. Paris : Les Ed. d'organisation.
- Rao, S. S. (2009).** Engineering optimization: theory and practice. John Wiley and Sons.

- Rodriguez-Torro, C., Jared, G and Swift, K. (2004).** Product Development complexity metrics: A framework for proactive-DFA implementation. International Design Conference-Design 2004, Dubrovnik, 483 – 490.
- Rose, C. M., Ishii, K and Masui, K. (1998).** How product characteristics determine end-of-life strategies. In Proceedings of the 1998 IEEE International Symposium on Electronics and the Environment, 4-6 May 1998, Oak Brook, IL, USA, 322 – 327.
- Said, C-W., Mitrouchev, P et Tollenaere, M. (2013).** Indicateurs d'évaluation de la désassemblabilité des produits. 21ème Congrès Français de Mécanique, Bordeaux, Réf S7B144EU,1 - 6.
- Said, C-W, Peter, M and Michel, T. (2014).** Disassembly sequencing for End-of-life products. Conference On Mechanical, Design Engineering & Advanced Manufacturing, Toulouse, France.
- Said, C-W. (2018).** Intégration des contraintes de désassemblage dans la conception modulaire des produits manufacturés. Contexte de développement durable. Thèse de doctorat. Université Grenoble Alpes.
- Samon, J.B. (2018).** Modélisation et évaluation de la démontabilité et de la remontabilité d'un équipement. Thèse de doctorat. ENSAI, Université de Ngaoundéré, Cameroun.
- Sanchez, R et Mahomey, J.T. (1996).** Modularity, flexibility and knowledge management inproduct and organization design. Strategic Management Journal, 17, 63 – 76.
- Sanchez, R. (1999).** Modular architectures in the marketing process. Journal of Marketing, 63, 92 – 112.
- Schilling, M and Steensma, H. (2001).** The use of modular organizational forms analysis. The Academy of Management Journal, 44 (6), 1149 – 1168.
- Searle, J. (1998).** La construction de la réalité sociale. Paris, Gallimard, 303 p.
- Shana, S and Hung, P. (2015).** A novel selective parallel disassembly planning method for green design. Journal of Engineering Design, 26 (10 - 12), 283 – 301.
- Sharman, D and Yassine, A. (2004).** Characterizing Complex Product Architectures. Systems Engineering Journal, 7 (1), 35 – 60.
- Shukla, P. K. (2007).** On the normal boundary intersection method for generation of efficient front. In Computational Science–ICCS 2007, Springer, 310 – 317.
- Simon, H. (1969).** La science des systèmes. EPI. Paris.

- Simon, H.A. (1997).** Sciences des systèmes, Sciences de l'artificiel. Version traduite en français par Le Moigne, J.L, Dunod.
- Smith, R.P and Eppinger, S.D. (1997).** A Predictive Model of Sequential Iteration in Engineering Design. *Management Science*, 43, 1104 – 1120.
- Sonnenberg, M. (2001).** Force and effort analysis of unfastening actions in disassembly processes. Thèse de doctorat, New Jersey Institute of Technology, New Jersey, USA.
- Sosa, M., Eppinger, S and Rowles, C. (2003).** Identifying modular and integrative systems and their impact on design team interactions. *Transactions of the ASME*, 125, 240 – 252.
- Sosa, M.E., Eppinger, S.D and Rowles, C.M. (2004).** The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, 50 (12), 1674 – 1689.
- Sosa, M.E. (2005).** A Network Approach to Define Modularity of Product Components. *Proceedings of ASME Design Theory and Methodology Conference*, Long Beach, CA.
- Srinivasan, H and Gadh, R. (1998).** A geometric algorithm for single selective disassembly using wave propagation abstraction. *Computer-Aided Design*, 30 (8), 603 – 613.
- Srinivasan, H., Figueroa, R and Gath, R. (1999).** Selective disassembly for virtual prototyping as applied to de-manufacturing. *Robotics and Computed Integrated Manufacturing*, 15, 231 – 245.
- Srinivasan, H and Gadh, R. (2000).** Efficient geometric disassembly of multiple components from an assembly using wave propagation. *Journal of Mechanical Design*, 122 (2), 179 – 184.
- Srinivasan, H and Gadh, R. (2002).** A non-interfering selective disassembly sequence for components with geometric constraints. *IIE Transactions*, 34, 349 – 361.
- Starkey, P. (1989).** Les conséquences pour la recherche sur la mécanisation et le développement de trente ans de vulgarisation des polyculteurs. Séminaire d'Economie Rurale, Montpellier (France) : CIRAD-MESRU, 12–18.
- Starr, M. (1965).** Modular production : a new concept. *Harvard Business Review*, 43 (6): 131 –142.
- Steuer, R. E and Choo, E.-U. (1983).** An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26 (3), 326 – 344.

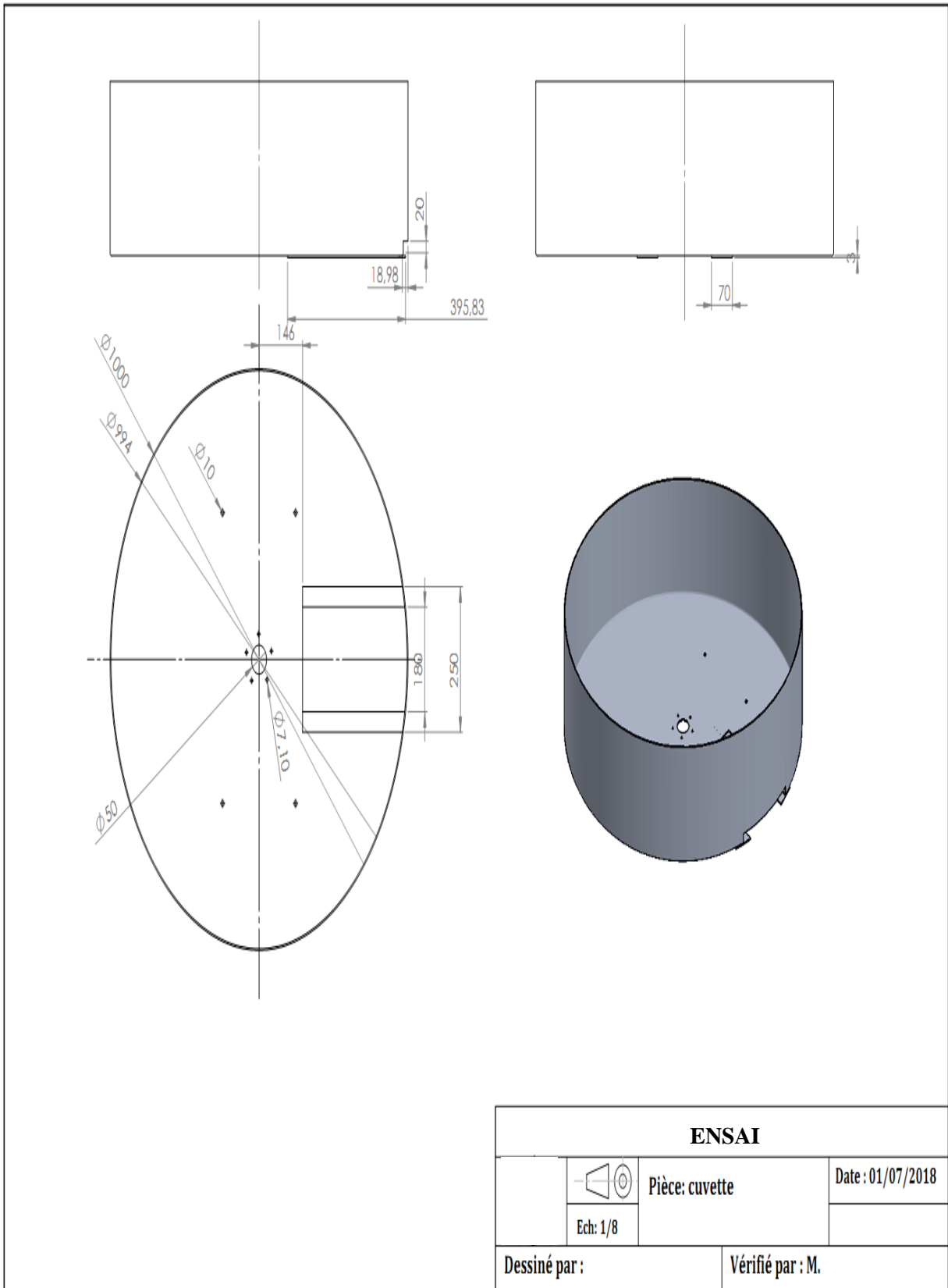
- Steward, D.V. (1981).** The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transactions on Engineering Management*, 28, 71–74.
- Stone, R, Wood, K and Crawford, R. (1998).** Heuristic Method to Identify Modules from a Functional Description of a Product. *Proceedings of DETC98 DETC98/DTM -5642 Atlanta, GA.*
- Stone, R., Wood, K and Crawford, R. (2000).** A Heuristic Method for Identifying Modules for Product Architectures. *Design Studies*, 21, 5 – 31.
- Subba Rao, V. M., Milind, D and Jayant, R. (1999).** Optimal Reliability allocation with discrete Cost-Reliability Data for components. *Operation Research*, 47 (6), 899 – 906.
- Subramani, A. K and Dewhurst, P. (1991).** Automatic Generation of Product Disassembly Sequences. *CIRP Annals- Manufacturing Technology*, 40 (1), 115 – 118.
- Suh, N. (1990).** *Principles of Design.* Oxford University Press, Cambridge, UK.
- Taguchi, G. (1978).** Off- Line and On-Line Quality Control Systems. *Proceedings of International Conferences on Quality Control, Tokyo, Japan.*
- Taguchi, G. (1993).** *On Robust Technology Development, Bringing Quality Engineering Upstream.* ASME Press, ISBN 0- 7918- 0028- 8, 164 p.
- Thebeau, R. E. (2001).** *Knowledge Management of System Interfaces and Interactions for Product Development Processes.* Master's Thesis. System Design and Management Program, Massachusetts Institute of Technology.
- Thornton, A. (2001).** Optimism vs. Pessimism: Design Decisions in the Face of Process Capability Uncertainty. *ASME Journal of Mechanical Design*, 123, 313–320.
- Tollenaere, M. (1998).** *Conception de produits mécanique.* Hermès, Paris.
- Torres, F., Puente, S. T and Aracil, R. (2003).** Disassembly Planning Based on Precedence Relations among Assemblies. *International Journal Advanced Manufacturing Technology*, 21 (5), 317 – 327.
- Tripathi, M., Agrawal, S., Pandey, M., Shankar, R and Tiwari, M. (2009).** Real world disassembly modeling and sequencing problem: Optimization by Algorithm of Self-Guided Ants (ASGA). *Robotics and Computer-Integrated Manufacturing*, 25 (3), 483 – 496.
- Tseng, Y., Kao, H.-T and Huang, F.-Y. (2009).** Integrated assembly and disassembly sequence planning using a GA approach. *International Journal of Production Research*, 48 (20), 1 – 23.

- Ulrich, K and Tung, K. (1991).** Fundamentals of product modularity. Proc. of ASME Winter Annual Meeting Symposium on Design and Manufacturing Integration, 73 -79.
- Ulrich, K. (1995).** The role of product architecture in the manufacturing firm. Research Policy, 24, 419 - 440.
- Ulrich, K and Eppinger, S. (2000).** Product Design and Development. 2nd Ed. McGraw-Hill, New York.
- Van Wie, M., Greer, J., Campbell, M., Stone, R and Wood, K. (2001).** Interfaces and product architecture. ASME Design Engineering Technical Conference Proc., DETC01/DTM-21689.
- Von, B. (1968).** General System Theory: Foundations, Development, Applications, New York, George Braziller. Traduction (1973) : Théorie générale des systèmes, Paris, Bordas (Dunod).
- Wang, J. F., Liu, J. H., Li, S. Q and Zhong, Y. F. (2003).** Intelligent selective disassembly using the ant colony algorithm. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 17 (4 - 5), 325 – 333.
- Weick, K. (1976).** Educational organizations as loosely coupled systems. Administrative Science Quarterly, 21 (1), 1 – 19.
- Whitfield, R.I., Smith, J.S and Duffy, A. (2002).** Identifying Component Modules. Proc. of Seventh International Conference on Artificial Intelligence in Design (AID'02), Cambridge, United Kingdom, 571 – 592.
- Wu, Y and Wu, A. (2000).** Taguchi Methods for Robust Design. ASME Press, New York.
- Xue, J., Qian, S and Zhang, Y. (2010).** Disassembly sequence planning based on ant colony optimization algorithm. IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, 1125 –1129.
- Yassine, A.A., Whitney, D. E., Lavine, J and Zambito, T. (2000).** Do-it-right first-time (DRFT) approach to DSM restructuring. In Proc. ASME 2000 Int. Design Engineering Technical Conf. (DTM), Baltimore, MD.
- Yassine, A., Whitney, D and Zambito, T. (2001).** Assessment of rework probabilities for simulating product development processes using the design structure matrix (DSM). Proceedings of DETC 01 ASME 2001 International Design Engineering Technical Conferences Computers and Information in Engineering Conference Pittsburgh, Pennsylvania.
- Yassine, A. (2004).** An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method. Italian Management Review, 9.

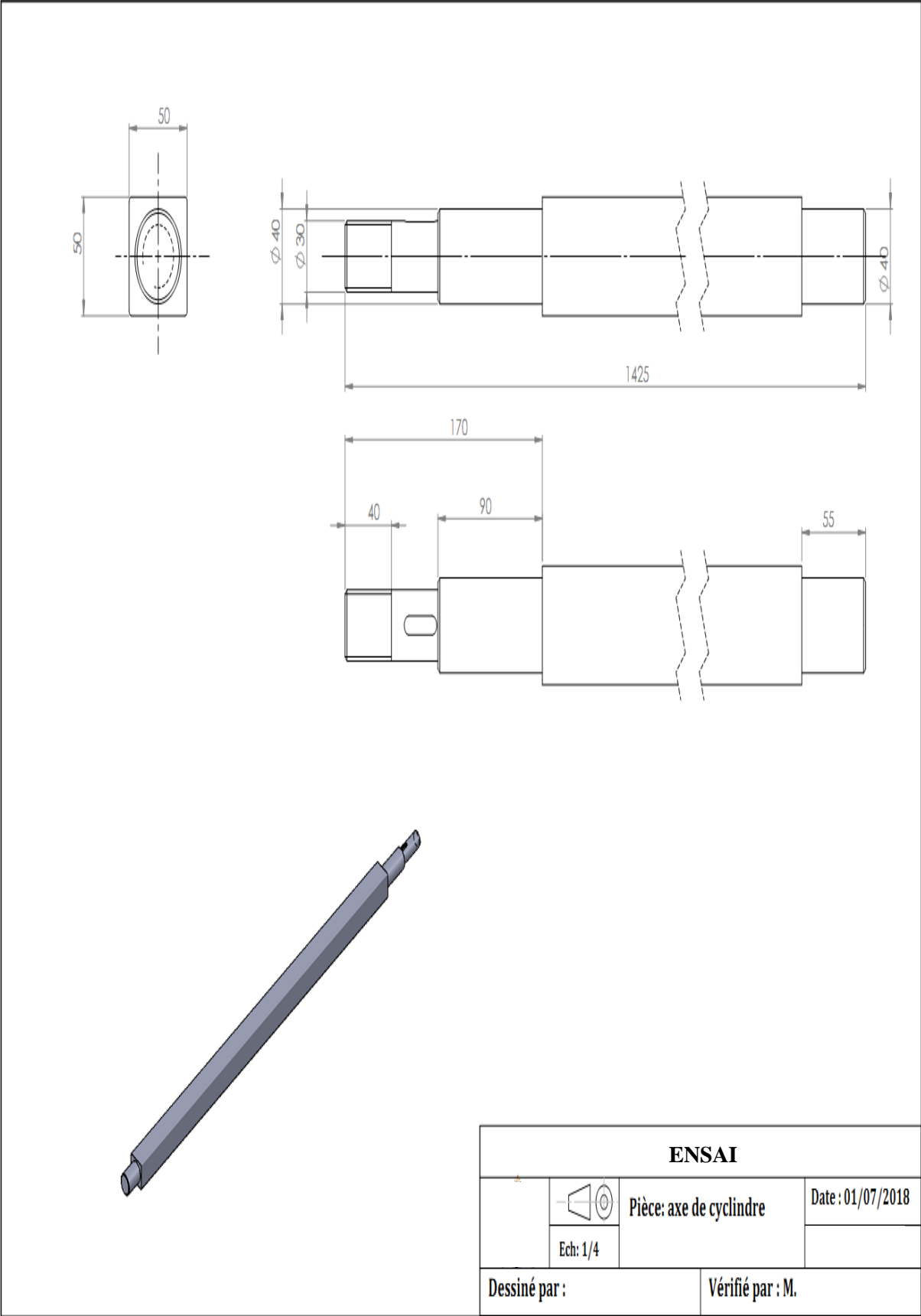
- Yousfi, N. (2014).** Optimisation de la Maintenance par la Fiabilité. Mémoire de Master en électromécanique.
- Yu, T., Yassine, A and Goldberg, D. (2003).** Genetic algorithm for developing modular product architectures. Proc. ASME 15th Int. Conf. Design Theory Methodology, Chicago.
- Yu, T., Yassine, A and Goldberg, A. (2007).** An information theoretic method for developing modular architectures using genetic algorithms. Research in Engineering Design, 18 (2), 91 –109.
- Zamirowski, E. J and Otto K. N. (1999).** Identifying Product Family Architecture Modularity Using Function and Variety Heuristics. In Proc of ASME Design Engineering Technical Conferences. Las Vegas, NV.
- Zeki, B. (1993).** Technologie de production de farines alimentaires et de produits protéiques issus du soja. Food & Agriculture Org, Institut de technologie d’Israël.
- Zhang, W. H and Gao, T. (2006).** A min–max method with adaptive weightings for uniformly spaced pareto optimum points. Computers and structures, 84 (28), 1760 –1769.
- Zhang, X and Zhang, S. Y. (2010).** Product cooperative disassembly sequence planning based on branch-and-bound algorithm. International Journal Advanced Manufacturing Technology, 51 (9-12), 1139 – 1147.
- Zussman, E., Kriwet, A and Seliger, G. (1994).** Disassembly oriented assessment methodology to support design for recycling. Annals of the CIRP, 43 (1), 9 – 14.
- Zwingelstein, G. (1996).** La Maintenance basée sur la fiabilité - Guide pratique d'application de la RCM. Editions HERMES.

ANNEXES

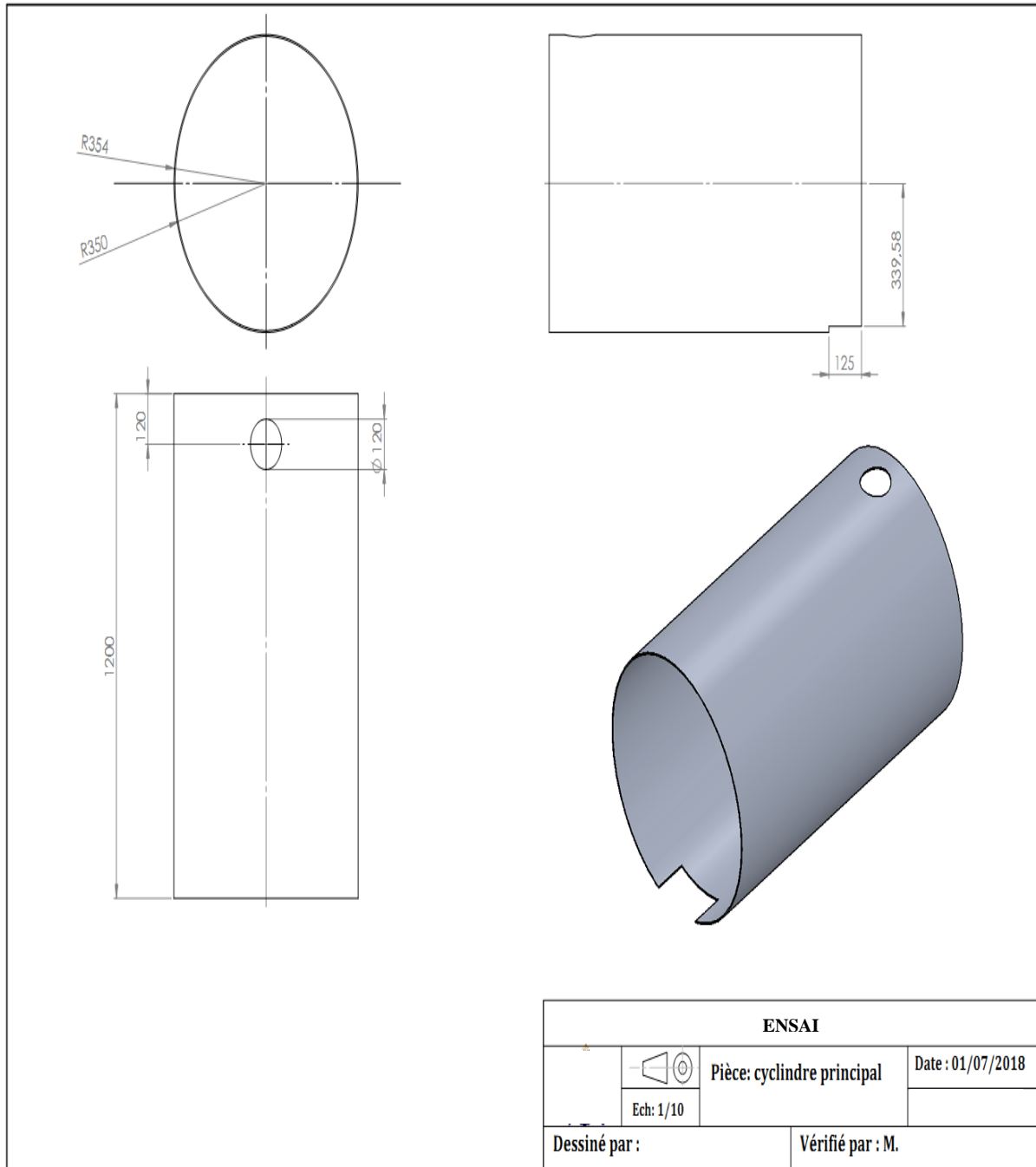
ANNEXE 1 : Schémas côté de la cuvette



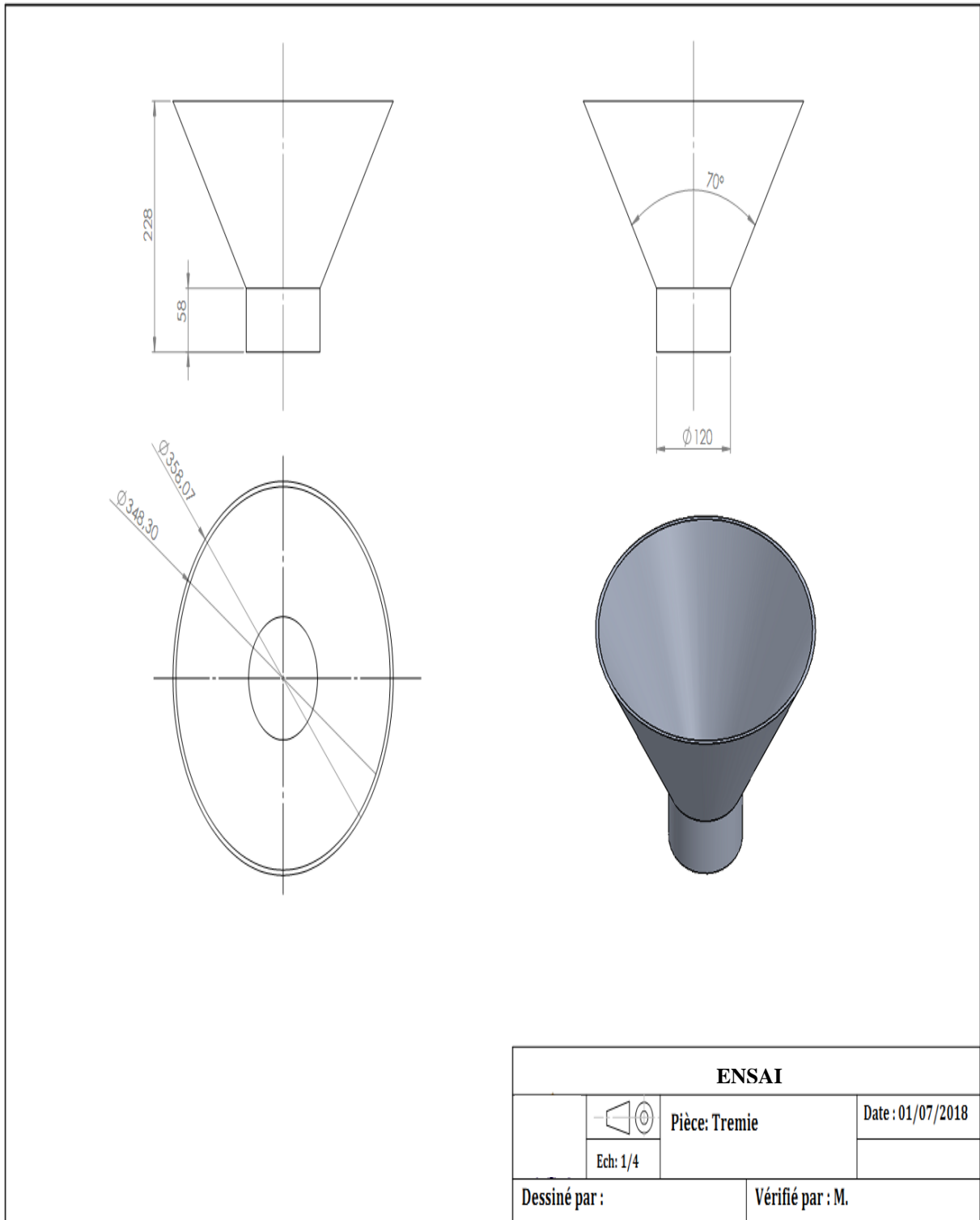
ANNEXE 2 : Schémas côté de l'axe du cylindre



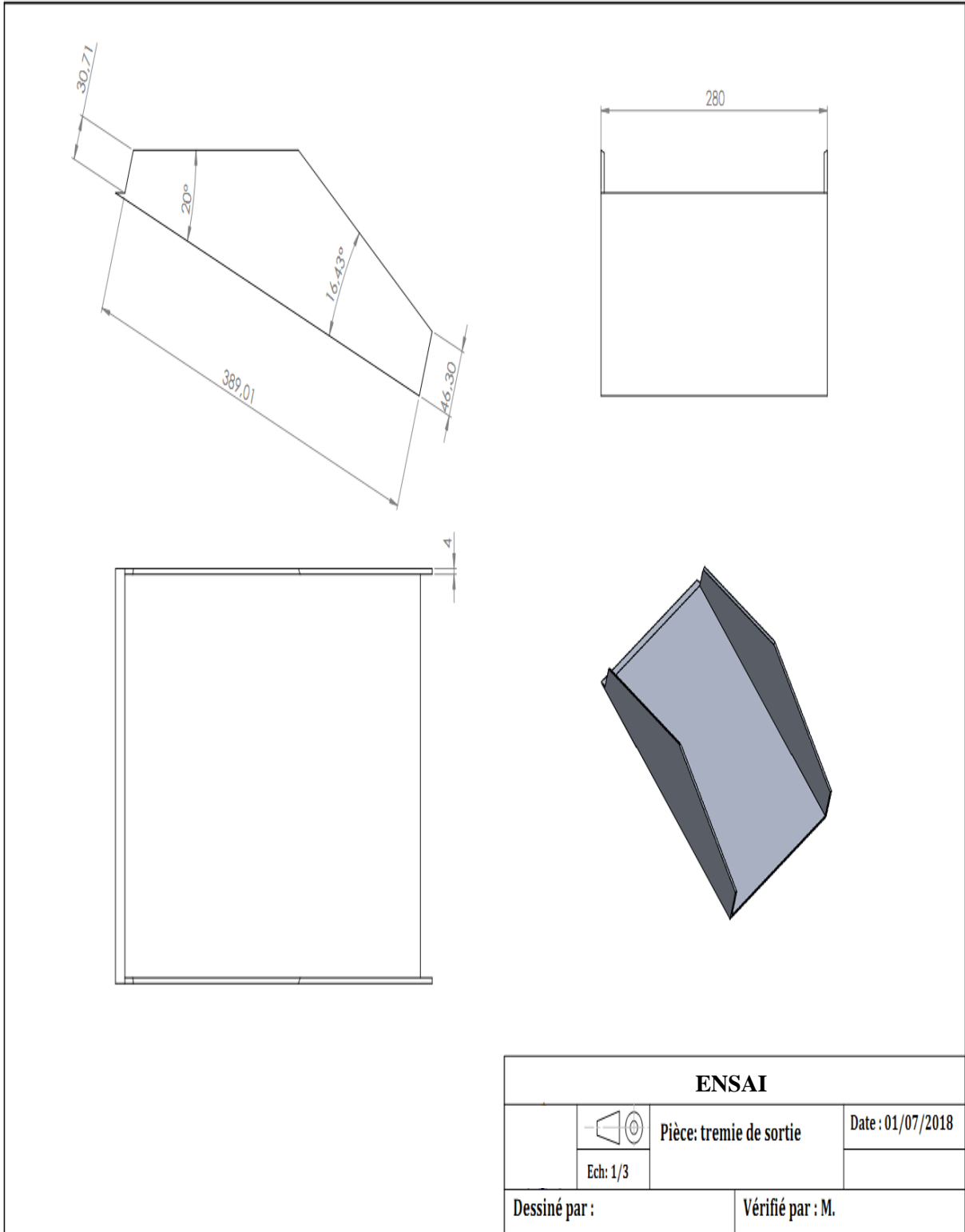
ANNEXE 3 : Schémas côté du cylindre principal



ANNEXE 4 : Schémas côté de la trémie d'entrée

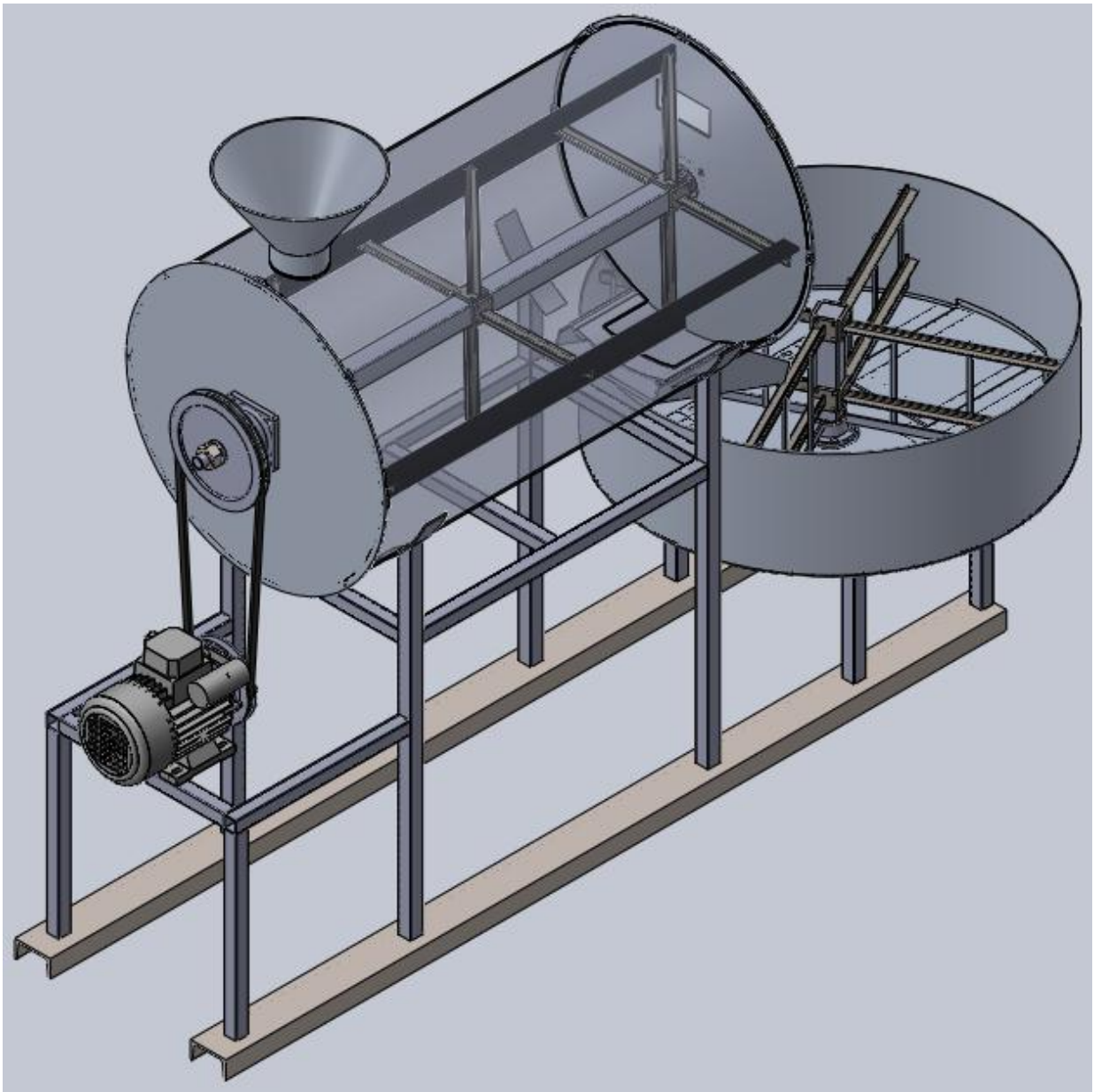


ANNEXE 5 : Schémas côté de la trémie de sortie

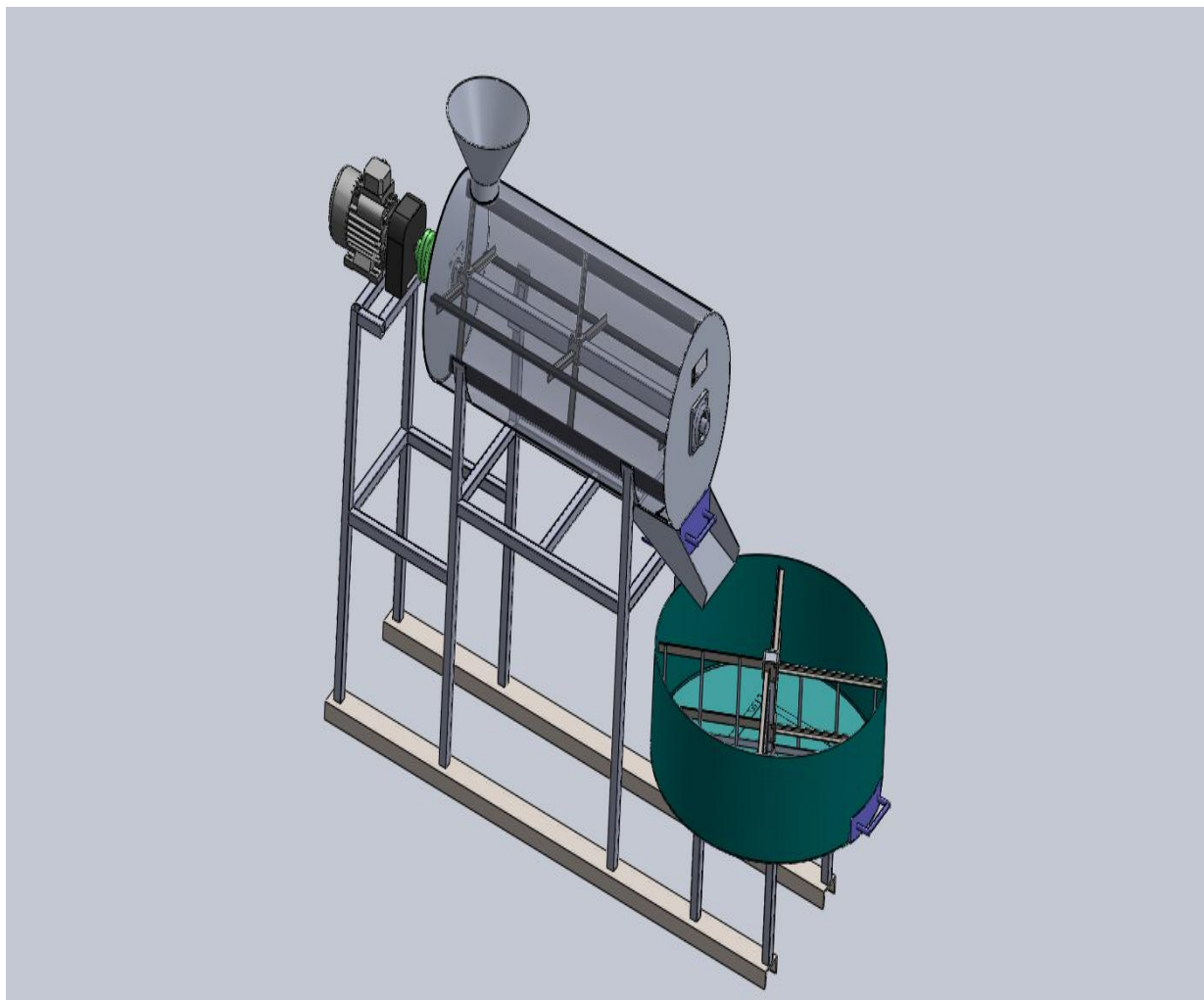


ENSAI			
	Pièce: tremie de sortie		Date : 01/07/2018
	Ech: 1/3		
Dessiné par :		Vérifié par : M.	

ANNEXE 6 : Torréfacteur de soja



ANNEXE 7 : Torréfacteur de soja reconfiguré



ANNEXE 8 : Quelques publications

Ngnassi Djami, A.B., Nzié, W. and Doka Yamigno, S. (2020). Disassembly Evaluation in Design of a System Using a Multi-Parameters Index. *Modern Mechanical Engineering*, 9, 65-80.

A.B. Ngnassi Djami, W. Nzié. and S. Doka Yamigno, “Maintenance Modularity Optimization using Clustering Algorithm: Application’’, *International Journal of Industrial Engineering (IJIE)*, ISSN:2349-9362, vol 7, n° 1, 2020, pp. 12-24.

Ngnassi Djami Aslain Brisco, Nzié Wolfgang and Doka Yamigno Serge, Mechanical System Topology Optimization for Better Maintenance, *Studies in Engineering and Technology*, ISSN: 2330 – 2038, EISSN 2330-2046, Vol. 8, No.1, 2020, pp.1-13.