

**UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE INDUSTRIEL ( PRODUCTIQUE )**

**PAR  
BRÉ MOUSSA DIOMANDÉ**

**DÉTERMINATION D'UNE SÉQUENCE OPTIMALE D'ASSEMBLAGE  
PAR  
LE REGROUPEMENT DES OPÉRATIONS**

**AVRIL 1995**

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## RÉSUMÉ

Pour un produit donné, il existe souvent un nombre impressionnant de séquences d'assemblage possibles. Puisque le choix de la séquence parmi toutes les possibilités peut influencer de façon remarquable le coût d'assemblage, le but de cette étude est d'optimiser les opérations d'assemblage en se basant sur le critère de regroupement des opérations.

Pour atteindre ce but, tout d'abord, la modélisation par une représentation géométrique des frontières des composants constituant un produit s'impose. Cette modélisation des composants produit un fichier nommé le fichier B-Rep (Boundary Representation). Ce fichier est à la base de cette étude. Puisque toutes les informations géométriques (surfaces, faces et points) d'un composant peuvent être connues à partir de ce fichier, la génération du modèle relationnel qui permet de visualiser les contacts entre les composants est effectuée. Ce modèle relationnel donne aussi des informations sur l'état de ces contacts, le type de contacts et l'identification des composants concernés par ces contacts. L'ensemble de ces informations forment entre ces composants un ensemble désigné relation.

Ensuite la formalisation du critère de regroupement des opérations est effectuée en identifiant 3 définitions et leurs paramètres d'évaluation:

- la direction d'assemblage est évaluée par la matrice des demi-degrés de liberté[73]. Cette matrice des demi-degrés de liberté décrit l'état des contacts entre 2 composants.
- la forme de l'outil est évaluée par la morphologie des composants.
- les tolérances sont évaluées par la fonction d'assemblage qui décrit le degré de difficulté d'exécution des opérations.

Puis la méthode générative [59] basée sur la théorie des graphes a été empruntée pour obtenir toutes les séquences à partir des relations générées par la création du modèle relationnel. A chaque fois que 2 opérations consécutives générées par cette méthode ont la même valeur de paramètre d'évaluation relié à chacune des 3 définitions, ces opérations sont regroupées dans un ensemble approprié à chacune des 3 définitions. La séquence optimale du point de vue de la direction d'assemblage par exemple sera la séquence ayant la plus grande valeur dans l'ensemble relié à la direction d'assemblage parmi toutes les séquences possibles. On procède de la même manière pour déterminer la séquence optimale en fonction des 2 autres définitions.

Les résultats de cette étude seront utilisés dans le projet de montages d'usinage modulaires pour pièces prismatiques du Laboratoire de Productique de l'U.Q.T.R. La séquence optimale pour assembler les modules, qui assureront l'isostatisme de la pièce à usiner, se générera ainsi de façon automatique.

## **REMERCIEMENTS**

Je remercie M. Luc Laperrière pour ses conseils, ses critiques, ses encouragements , son support, sa disponibilité qui ont été utiles dans la réalisation de ce projet à titre de directeur. Je tiens aussi à remercier M. Demagna Koffi pour ses conseils en tant que co-directeur dans la supervision de ce projet et M. Jean-Christophe Cuillère pour ses explications concernant l'utilisation du logiciel ICEM/DDN.

Mes remerciements s'adressent aussi, à mes collègues du Laboratoire de Productique Eric Buczkowski et Louis Moglo pour leurs conseils, ainsi qu'à tous mes amis.

Enfin mes remerciements vont aux fonds FODAR, FIR et CRSNG dont le support financier a été fortement apprécié.

## TABLE DES MATIÈRES

RÉSUMÉ.....	i
REMERCIEMENTS.....	ii
TABLE DES MATIÈRES.....	iii
LISTE DES FIGURES.....	vii
 <b>CHAPITRES</b>	
1 INTRODUCTION.....	1
1.1. Système d'assemblage.....	1
1.2. Élaboration des gammes d'assemblage.....	3
1.3. Élaboration automatique des séquences d'assemblage assisté par ordinateur.....	3
2 ÉTAT DE L'ART.....	6
2.1. Résumé de quelques articles.....	6
2.1.1. Article 1: Détermination des sous-assemblages d'un produit à partir des séquences temporelles d'assemblage. ....	7
2.1.2. Article 2: Simplified Generation of All Mechanical Assembly Sequences.....	8
2.1.3. Article 3: An integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products.....	9
2.1.4. Article 4: Planning of Products Assembly and Disassembly.....	11
2.1.5. Article 5: Automated Sequencing and Subassembly Detection in Assembly Planning.....	14
2.1.6. Article 6: A Pragmatic Approach to Computer Aided Assembly Planning.....	17
2.2 Analyse de l'état de l'art.....	19
2.2.1. Modélisation.....	19
2.2.2. Génération des séquences d'assemblage.....	20
2.2.2.1. Génération des séquences à l'aide des contraintes de priorité.....	20

	2.2.2.2. Méthode de désassemblage.....	21
	2.2.2.3. Autres méthodes.....	21
	2.2.3. Représentation des séquences d'assemblage.....	22
	2.3. Proposition du sujet de recherche.....	28
	2.3.1. Graphe des liaisons.....	28
	2.3.2. Détermination de la séquence optimale d'assemblage.....	29
3	GÉNÉRATION AUTOMATIQUE DU GRAPHE DES LIAISONS D'UN PRODUIT DONNÉ.....	30
	3.1. Modélisation solide des composants du produit.....	30
	3.2. Génération du graphe des liaisons.....	30
	3.2.1. Création du fichier B_Rep.....	30
	3.2.2. Détection des contacts entre les composants.....	32
	3.2.2.1. Formulation mathématique des conditions de contact.....	32
	3.2.2.2. Mise en oeuvre algorithmique des conditions de contact.....	41
	3.2.2.3. Limitation de la détection des contacts.....	43
	3.2.3. Génération des relations.....	44
	3.2.3.1. Définitions.....	44
	3.2.3.2. Mise en oeuvre informatique de la génération des relations.....	47
	3.3. Matrice des demi-degrés de liberté.....	47
	3.3.1. Représentation des états des contacts.....	47
	3.3.2. Mise en oeuvre informatique de la génération des matrices de demi-degrés de liberté.....	48
	3.4. Affichage du graphe des liaisons sur un écran d'ordinateur.....	50
4	DÉFINITIONS DU REGROUPEMENT DES OPÉRATIONS ET PARAMÈTRES D'ÉVALUATION.....	53
	4.1. Regroupement des opérations en fonction de la direction d'assemblage.....	53
	4.1.1. Direction d'assemblage.....	53

	4.1.2. Paramètre d'évaluation de la direction d'assemblage: la matrice des demi-degrés de liberté.....	54
	4.1.3. Utilisation de la matrice des demi-degrés de liberté.....	54
	4.2. Regroupement des opérations en fonction du changement des outils.....	63
	4.2.1. Forme de l'outil.....	63
	4.2.2. Paramètre d'évaluation de la forme de l'outil: morphologie des composants.....	63
	4.2.3. L'utilisation de la morphologie des composants.....	64
	4.3. Regroupement des opérations en fonction des tolérances.....	68
	4.3.1. Paramètre d'évaluation: accumulation des tolérances.....	68
	4.3.2. Utilisation de l'accumulation des tolérances.....	69
	4.3.3. Calcul des fonctions d'assemblage des 6 séquences.....	71
5	DÉTERMINATION DE LA SÉQUENCE OPTIMALE PAR LE REGROUPEMENT DES OPÉRATIONS.....	75
	5.1. La génération des séquences d'assemblage.....	75
	5.2. Sélection d'une séquence optimale basée sur le regroupement des opérations.....	77
	5.2.1. Mémorisation de la similitude de la direction d'assemblage, de la forme de saisie et de la fonction d'assemblage de 2 opérations consécutives d'un noeud.....	77
	5.2.2 Proposition d'une séquence optimale.....	78
6	RÉSULTATS.....	80
	6.1. Séquence optimale en fonction de la direction d'assemblage.....	80
	6.2. Séquence optimale en fonction de la forme des composants impliqués dans les opérations d'assemblage.....	87
	6.3. Séquence optimale en fonction des tolérances.....	89
7	CONCLUSION ET DISCUSSION.....	96
	7.1. CONCLUSION.....	96
	7.2. DISCUSSION.....	98
8	BIBLIOGRAPHIE.....	100

9 ANNEXES.....112

9.1 Fichier B\_Rep.....112

## LISTE DES FIGURES

Figure 1.1: système d'assemblage.....	1
Figure 1.2: place de l'assemblage dans l'exécution d'un produit.....	2
Figure 2.1: schéma et graphe fonctionnel d'une pompe à huile.....	7
Figure 2.2: séquences opérationnelles de la pompe à huile où les liaisons sont représentées par des chiffres.....	8
Figure 2.3: cycle d'assemblage.....	10
Figure 2.4: exemple de rupture des liaisons.....	11
Figure 2.5: exemple de génération de séquences.....	12
Figure 2.6: regroupements des noeuds identiques.....	13
Figure 2.7: réduction de la taille des séquences.....	13
Figure 2.8: exemple de représentation des liaisons.....	15
Figure 2.9: détection des sous-assemblages.....	16
Figure 2.10: approximation d'un composant à un parallélépipède.....	17
Figure 2.11: méthode des comparaisons.....	18
Figure 2.12: insertion et attachement.....	18
Figure 2.13: modèle relationnel d'un produit à 3 composants.....	20
Figure 2.14: séquences d'état d'un produit quelconque.....	23
Figure 2.15: arbre d'assemblage.....	23
Figure 2.16: arbre des sous-assemblages.....	24
Figure 2.17: arbre d'états.....	24
Figure 2.18: graphe dirigé.....	24
Figure 2.19: arbre AND/OR.....	25
Figure 2.20: graphe AND/OR.....	25
Figure 2.21: graphe de priorité d'un produit p.....	26
Figure 2.22: exemple de représentation des séquences.....	27
Figure 3.1: produit de 4 composants.....	31
Figure 3.2: algorithme de reconstitution du fichier b_rep.....	31
Figure 3.3: 2 composants en contact.....	32
Figure 3.4: modélisation mathématique d'un contact.....	32
Figure 3.5: contact plan-plan.....	33
Figure 3.6: modélisation mathématique d'un contact plan-plan.....	33
Figure 3.7: contact plan-cylindre.....	34
Figure 3.8: modélisation mathématique d'un contact plan-cylindre.....	35
Figure 3.9: contact plan-ellipse.....	35
Figure 3.10: modélisation mathématique d'un contact plan-ellipse.....	36
Figure 3.11: contact ellipse-ellipse.....	37

Figure 3.12: contact cylindre-cylindre.....	37
Figure 3.13: modélisation mathématique d'un contact cylindre-cylindre.....	38
Figure 3.14: contact sphère-sphère.....	39
Figure 3.15: contact sphère-cylindre.....	39
Figure 3.16: contact sphère-ellipse.....	40
Figure 3.17: contact sphère-plan.....	40
Figure 3.18: superposition de 2 surfaces.....	41
Figure 3.19: algorithme de comparaison des surfaces.....	42
Figure 3.20: procédure de comparaison des surfaces.....	43
Figure 3.21: algorithme de détection d'un contact plan-plan.....	43
Figure 3.22: exemple de composants en contact.....	44
Figure 3.23: bloc 1 et 4 ayant une relation de blocage.....	45
Figure 3.24: bloc 1 et 4 ayant une relation libre.....	45
Figure 3.25: graphe des liaisons.....	46
Figure 3.26: description de la relation entre le bloc 3 et 4 selon le format proposé[59].....	46
Figure 3.27: représentation des contacts par la matrice des demi-degrés de liberté.....	47
Figure 3.28: matrice des demi-degrés de liberté du contact du point de vue du composant 1 du contact entre les composants 1 et 2.....	48
Figure 3.29: produit de 2 composants.....	49
Figure 3.30: procédure de modification des éléments de la matrice des demi-degrés de liberté de 2 composants.....	50
Figure 3.31: classe composant.....	51
Figure 3.32: procédure calculant les coordonnées des composants.....	52
Figure 3.33: procédure traçant les entités graphiques.....	52
Figure 4.1: direction d'assemblage de 2 composants.....	54
Figure 4.2: algorithme transformant les éléments de la matrice en des axes du système de coordonnées.....	55
Figure 4.3: exemple d'un produit à 10 composants.....	57
Figure 4.4: séquences d'assemblage respectant l'interférence géométrique.....	59
Figure 4.5: liste des directions d'assemblage de la figure 4.4.....	60
Figure 4.6: les régions de saisie et d'insertion d'un composant.....	63
Figure 4.7: les régions fixe et réceptrice d'un composant.....	64
Figure 4.8: composant touchant la ligne de rupture.....	64
Figure 4.9: algorithme déterminant la forme désignée d'un sous-assemblage.....	65
Figure 4.10: liste de la forme des composants des opérations de la figure 4.4.....	66
Figure 4.11: produit de 4 blocs.....	70
Figure 4.12: tolérances des composants.....	71
Figure 5.1: arbre sous-tendant minimum du graphe des liaisons des 4 blocs.....	75
Figure 5.2: génération de toutes les possibilités de rupture des liaisons du graphe.....	76
Figure 5.3: mémorisation de la direction d'assemblage, de la forme de saisie et de la fonction d'assemblage.....	78
Figure 5.4: valeurs des variables dir_x, dir_y et dir_z des séquences.....	79
Figure 5.5: coûts associés aux variables dir_x, dir_y et dir_z des séquences.....	79
Figure 6.1: montage d'usinage.....	81

Figure 6.2: vue explosée du montage d'usinage.....	82
Figure 6.3: vue indiquant les composants du montage d'usinage.....	83
Figure 6.4: " screen dump" modèle généré automatiquement à partir d'un dessin solide relationnel du montage d'usinage.....	84
Figure 6.5: séquence arbitraire.....	85
Figure 6.6: séquence optimale basée sur le regroupement de la direction d'assemblage.....	86
Figure 6.7: séquence non optimale.....	87
Figure 6.8: séquence optimale basée sur la forme de saisie.....	88
Figure 6.9: vue explosée de la lampe de poche.....	89
Figure 6.10: vue indentifiant les composants.....	91
Figure 6.11: " screen dump" modèle généré automatiquement à partir d'un dessin solide relationnel de la lampe de poche.....	92
Figure 6.12: tolérances sur les relations.....	93
Figure 6.13: séquence optimale du point de vue des tolérances.....	94
Figure 9.1: entités géométriques d'un cube.....	113
Figure 9.2: exemple de fichier B_Rep.....	114
Figure 9.3: sens des "coedge " et "edge".....	115
Figure 9.4: exemple de sens des faces.....	115

Figure 6.2: vue explosée du montage d'usinage.....	82
Figure 6.3: vue indiquant les composants du montage d'usinage.....	83
Figure 6.4: " screen dump" modèle généré automatiquement à partir d'un dessin solide relationnel du montage d'usinage.....	84
Figure 6.5: séquence arbitraire.....	85
Figure 6.6: séquence optimale basée sur le regroupement de la direction d'assemblage.....	86
Figure 6.7: séquence non optimale.....	87
Figure 6.8: séquence optimale basée sur la forme de saisie.....	88
Figure 6.9: vue explosée de la lampe de poche.....	89
Figure 6.10: vue indentifiant les composants.....	91
Figure 6.11: " screen dump" modèle généré automatiquement à partir d'un dessin solide relationnel de la lampe de poche.....	92
Figure 6.12: tolérances sur les relations.....	93
Figure 6.13: séquence optimale du point de vue des tolérances.....	94
Figure 9.1: entités géométriques d'un cube.....	113
Figure 9.2: exemple de fichier B_Rep.....	114
Figure 9.3: sens des "coedge " et "edge".....	115
Figure 9.4: exemple de sens des faces.....	115

## CHAPITRE 1

### INTRODUCTION

#### 1.1. Système d'assemblage

Un système d'assemblage est un système qui permet d'obtenir un produit fini à partir de pièces élémentaires comme l'indique la figure 1.1 [73].

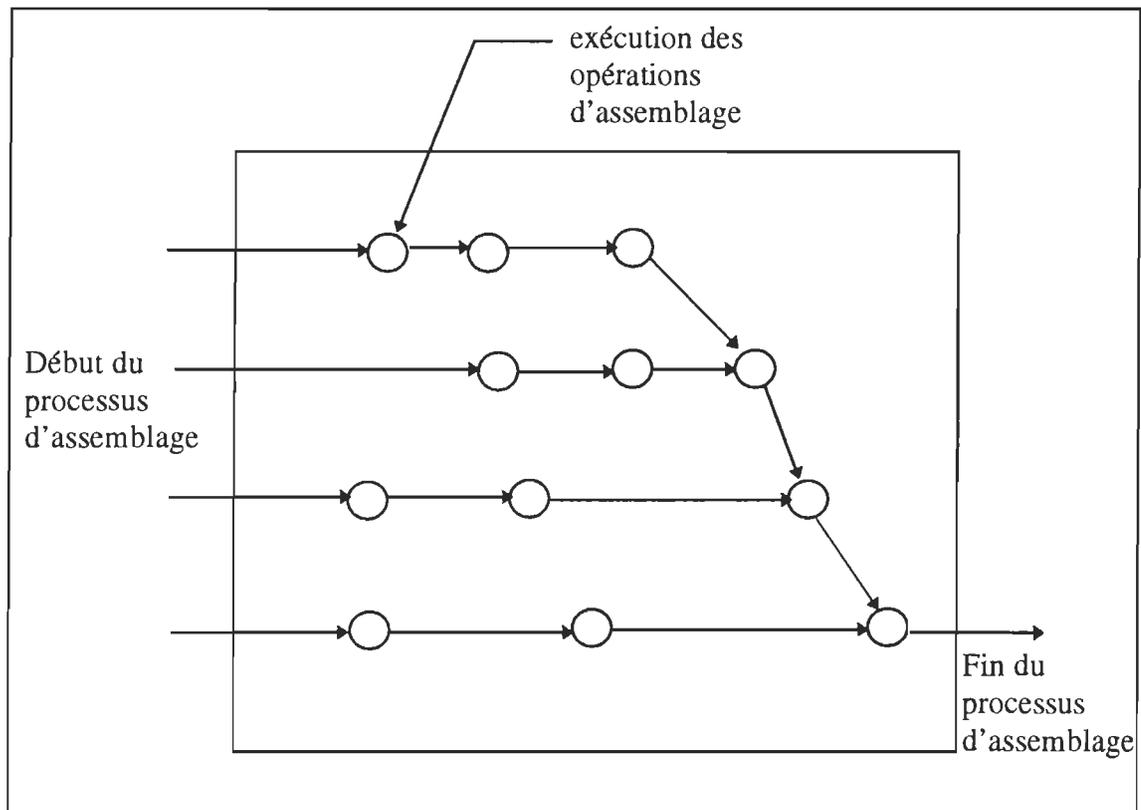


Figure 1.1: système d'assemblage

Le système d'assemblage implique généralement :

- 1- L'identification des composants du produit et les sous-assemblages;
- 2- La détermination des opérations et leur ordre d'exécution possible;
- 3- La sélection des machines, des outils et des fixations;

- 4- La durée de l'assemblage;
- 5- Le calcul du coût d'assemblage.

L'analyse de la méthode d'assemblage au niveau du système d'assemblage conduit à un nombre élevé de possibilités (gammes) d'assemblage des pièces mécaniques. Le choix d'une de ces possibilités influence significativement le coût d'assemblage d'un produit et le temps d'assemblage. Le produit à assembler faisant partie d'une chaîne de fabrication, l'exécution rapide de la phase d'assemblage permettra de rendre plus efficace la fabrication d'un produit.

Pour y arriver, il faut simplifier et automatiser le processus d'assemblage ainsi que la génération des séquences d'assemblage. Cette automatisation sera possible si les bureaux d'études et ceux des méthodes sont munis d'outils informatiques efficaces permettant d'optimiser l'étape de la préparation à l'assemblage qui est la génération de la gamme d'assemblage. La figure 1.2 nous montre la place de l'assemblage dans la phase d'élaboration d'un produit en entreprise.

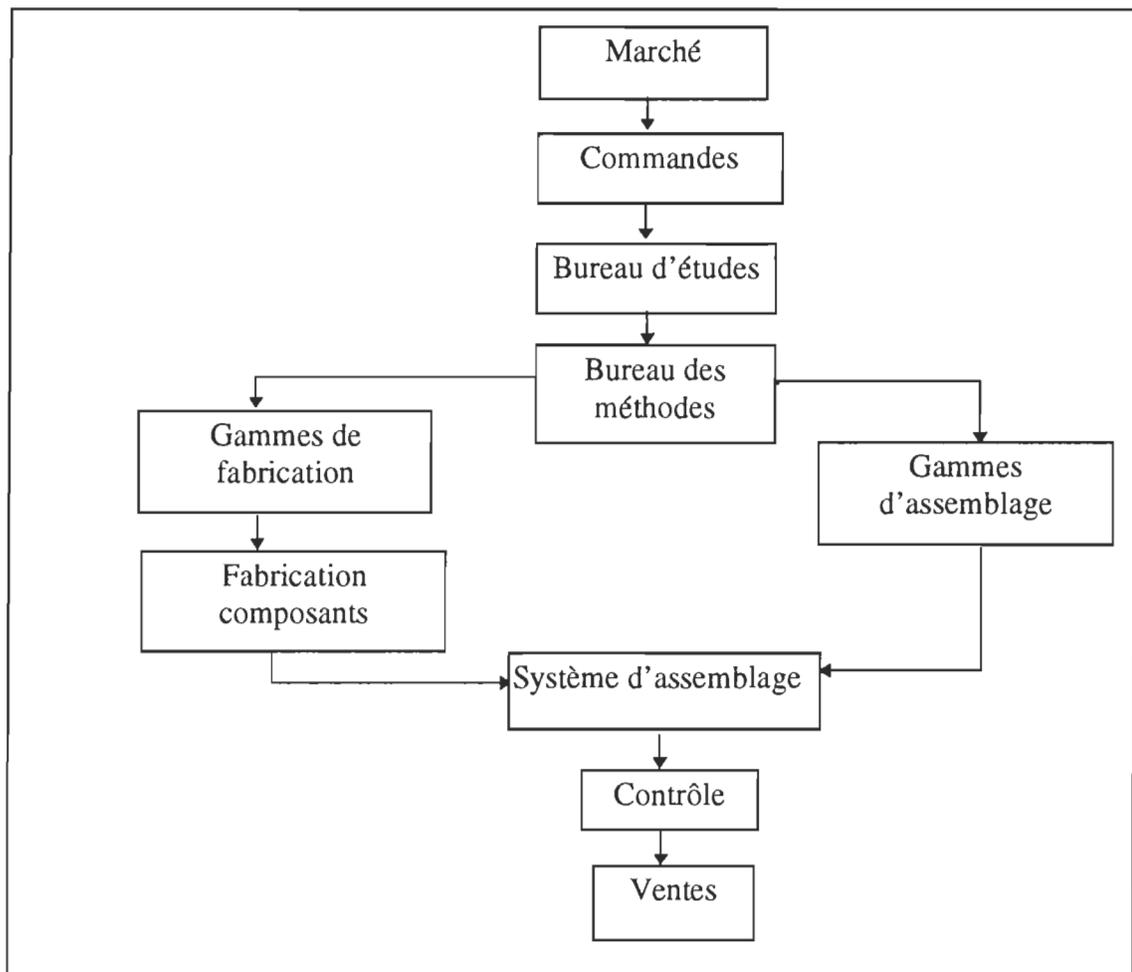


Figure 1.2: place de l'assemblage dans l'élaboration d'un produit

Pour pallier à ce phénomène, le domaine de l'assemblage a fait l'objet d'une étude marquée depuis quelques années. Cette étude s'est beaucoup attardée sur l'élaboration des gammes d'assemblage afin d'éviter au concepteur d'employer la méthode intuitive qui, à long terme, donne des résultats ne correspondant pas aux conditions techniques et économiques imposées.

De plus, la complexité combinatoire du nombre des gammes d'un produit, la réduction des coûts d'assemblage, l'augmentation de la productivité, la complexité des produits amènent les entreprises qui se disent compétitives à automatiser leurs systèmes d'assemblage.

## **1.2. Élaboration des gammes d'assemblage**

Les produits finis techniques réalisés industriellement sont essentiellement constitués de plusieurs pièces ou composants élémentaires qui sont le plus souvent fabriqués en des temps différents et en des lieux séparés.

Une **opération d'assemblage** consiste à établir un contact physique entre deux groupes constitués chacun de pièces élémentaires. Ces groupes sont généralement appelés **sous-assemblages** et conduisent à 3 situations :

- 1- Les 2 sous-assemblages peuvent être constitués chacun d'une pièce élémentaire;
- 2- Les 2 sous-assemblages peuvent être constitués chacun de plusieurs pièces élémentaires;
- 3- L'un de ces sous-assemblages peut être constitué d'une pièce élémentaire tandis que l'autre comporte plusieurs pièces élémentaires.

L'obtention d'une séquence d'assemblage consiste à générer une suite successive d'opérations d'assemblage à partir des pièces élémentaires pour aboutir à un produit final. Les opérations d'une séquence doivent satisfaire à des contraintes géométriques, de stabilité et technologiques. La réalisation effective de ces opérations met en jeu plusieurs éléments qui sont les sous-assemblages à assembler, les outils d'assemblage et les fixations. Ces différents éléments constituent une source de complication majeure qu'il faut vaincre pour réussir l'opération d'assemblage.

## **1.3. Élaboration automatique des séquences d'assemblage assistée par ordinateur**

L'apport de l'ordinateur dans l'élaboration automatique des séquences d'assemblage constitue un lien entre le concepteur et la production du produit assemblé. A l'aide donc de l'ordinateur, toutes les informations géométriques des composants sont analysées pour identifier les composants en contact et produire ainsi des relations. La génération des séquences se fait par la suite en analysant les relations entre les composants.

Le résultat de cette analyse est un éventail de séquences réalisables. La génération de ces séquences se fait en général suivant 2 approches:

1- L'approche des variantes qui est basée sur la technologie de groupe. Au fur et à mesure que les séquences des pièces sont générées, le produit, ses composants et les séquences d'assemblage sont stockés dans une base de données. Si de nouvelles séquences doivent être générées pour un nouveau produit, la base de données est consultée afin de vérifier si un produit semblable a déjà été traité. Si tel est le cas, les séquences contenues dans la base de données sont proposées, sinon de nouvelles séquences sont générées pour ce nouveau produit et toutes les informations obtenues de cette génération de séquences sont stockées dans la base de données.

2- L'approche générative qui consiste à recommencer le processus de génération de séquences d'assemblage à chaque fois que les séquences d'assemblage d'un nouveau produit doivent être générées. Cette approche ne tient pas compte des séquences déjà générées car elle ne constitue pas une base de données avec les séquences d'assemblage des produits traités.

L'approche des variantes a pour avantage d'éviter la répétition des séquences déjà générées dans le passé car les produits assemblés sont très fréquemment de même type avec quelques différences. Cette approche permet donc de classer les produits en fonction des composants (morphologie), d'ensemble de séquences et de groupes de produits. Avec cette méthode, une base de connaissances peut être formée.

Comme inconvénient, cette approche nécessite une grande mémoire pour former une base de connaissances car le nombre de séquences d'assemblage pour un produit avec plus de composants augmente d'une manière exponentielle.

L'approche générative a pour avantage d'éviter le besoin d'une grande mémoire pour des produits avec plus de composants. Cette manière de recommencer le processus de génération de séquences d'assemblage en ne tenant pas compte des produits déjà traités dans le passé permet de générer des séquences sur des modèles mathématiques ou des théories sans intervention humaine. L'utilisation de ces modèles ou de ces théories entraînent une exactitude des solutions fournies.

Dans ce projet, l'approche utilisée est l'approche générative puisque le logiciel permettant d'obtenir toutes les séquences est basé sur cette approche. La séquence optimale à obtenir dans cette étude sera probablement fournie à un robot ou un manipulateur pour l'exécuter.

Le prochain chapitre de ce mémoire qui est l'état d'art dans le domaine de l'assemblage est présenté. A l'issue de l'analyse de cet état d'art, le sujet de recherche sera dégagé qui constituera une contribution au domaine de l'assemblage.

Le chapitre 3 expose une méthode de génération automatique du model relationnel d'un produit à assembler. Cette méthode est un moyen efficace de visualisation rapide des liens entre les différents composants d'un produit et d'obtention des informations telles que les relations qui seront utilisées par le logiciel qui génère les séquences d'assemblage.

Le chapitre 4 présente les définitions utilisées pour effectuer les regroupements des opérations ainsi que leur paramètre d'évaluation. C'est en fonction de ces définitions que les regroupements seront effectués pour sélectionner la séquence optimale d'assemblage.

Le chapitre 5 décrit la méthode employée pour sélectionner une séquence optimale d'assemblage. Cette sélection consiste à évaluer les opérations au fur à mesure qu'elles sont générées en fonction des définitions et à faire immédiatement le regroupement si les conditions de regroupements sont réunies.

Le chapitre 6 présente les résultats obtenus. Deux exemples ont été choisis pour illustrer le choix de la séquence optimale d'assemblage en fonction des 3 définitions. Un premier exemple qui est un montage d'usinage est utilisé pour les 2 premières définitions à savoir la direction d'assemblage et le changement d'outil. Le dernier exemple, qui est une lampe de poche, est utilisé pour la troisième définition: les tolérances.

Une conclusion et une discussion sont faites dans le chapitre 7. Le chapitre 8 présente la bibliographie utilisée dans cette étude et enfin le chapitre 9 contient les annexes.

## CHAPITRE 2

### ÉTAT DE L'ART

Depuis quelques années, la plupart des chercheurs dans le domaine de l'assemblage ont emprunté le même cheminement pour proposer un certain nombre de séquences d'assemblage. Ce cheminement se résume en 4 grandes étapes.

a) La modélisation du produit à assembler qui consiste à faire une analyse de la représentation géométrique des différents composants du produit à assembler et par la suite produire un schéma décrivant les liens physiques entre ces composants. Ces liens physiques intrinsèques au produit sont des contacts qui peuvent être du type physique, fonctionnel ou encore du type blocage.

b) La génération et représentation des séquences d'assemblage: à cette étape de l'assemblage, le graphe des liaisons (schéma des relations entre les composants) est utilisé pour produire les séquences d'une manière exhaustive qui forment un ensemble de chemin d'assemblage des composants. Ces séquences ainsi obtenues, sont représentées suivant plusieurs méthodes mises en oeuvre par différents auteurs. L'ordonnancement des opérations à l'intérieur des séquences se fait par le biais des relations de priorité, des heuristiques ou en utilisant la théorie des graphes.

c) L'élimination de certaines séquences parmi les séquences générées: les séquences dont les opérations ne satisfont pas soit les contraintes de géométrie, soit de stabilité, soit matériel ou soit technologique sont éliminées. Cette élimination réduisant ainsi le nombre de séquences d'assemblage.

d) L'évaluation des gammes restantes: des définitions ou des fonctions sont formulées afin d'évaluer les séquences restantes. Cette évaluation permettra d'éliminer un certain nombre de séquences voire obtenir une seule séquence optimale.

#### 2.1. Résumé de quelques articles

L'étude détaillée de quelques articles pertinents sur le sujet de l'assemblage permettra de faire une analyse pour ressortir un sujet de recherche.

### 2.1.1. Article 1: Détermination des sous-assemblages d'un produit à partir des séquences temporelles d'assemblage [9]

La détermination des séquences opératoires d'assemblage, assistée par ordinateur, élaborée préalablement par Bourjault et Henrioud nécessite une taille mémoire et des combinaisons élevées pour un produit de plus de dix composants. Pour pallier à cet handicap, ils proposent dans cet article une méthode axée sur la mise en évidence des sous-assemblages.

En effet, l'étape initiale consiste à représenter à partir du schéma des constituants du produit toutes les liaisons mécaniques ou fonctionnelles à caractère binaire (contacts physiques) entre ces différents constituants en un schéma appelé graphe des liaisons fonctionnelles non orienté et connexe (il n'y a pas de composants isolés). La représentation du graphe des liaisons fonctionnelles est montrée à la figure 2.1.

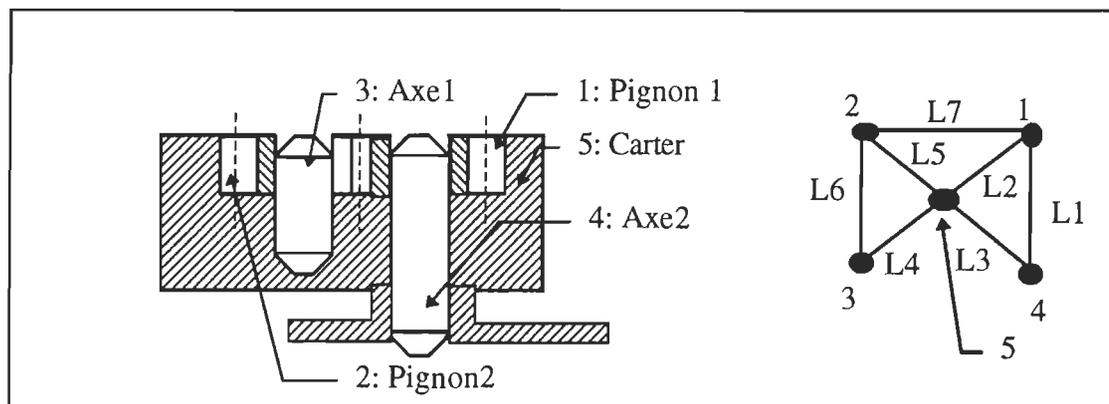


Figure 2.1: schéma et graphe fonctionnel d'une pompe à huile.

A partir de ce graphe ainsi obtenu, les conditions de faisabilité représentées par une fonction logique (C) sont déterminées pour chaque liaison. Cette fonction est constituée de la somme des produits des variables binaires (f) qui définissent l'état de la liaison à un instant quelconque du processus d'assemblage. Cette fonction présentée par la formule 2.1.

$$C_i = \sum_{k=1}^{q_i} F = \sum_{k=1}^{q_i} \left( \prod_{j=1}^n f_j^k \right) \quad (2.1)$$

L'exploitation des fonctions ainsi obtenues conduit à l'obtention des séquences opératoires (suite d'actions fonctionnelles à réaliser en série et/ou en parallèle) qui respectent les conditions de faisabilité. La figure 2.2 montre un exemple de séquences opératoires. Dans cette figure les chiffres séparés par une virgule indiquent que les liaisons correspondant à ces chiffres sont établies en même temps.

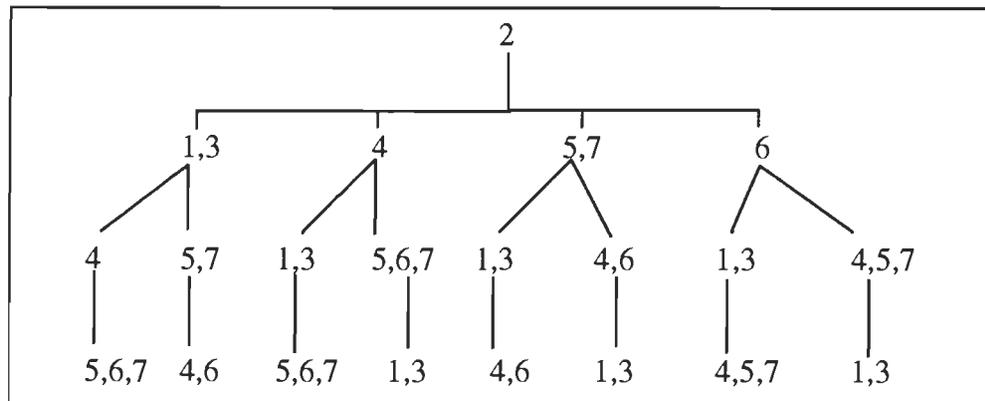


Figure 2.2: séquences opérationnelles de la pompe à huile où les liaisons sont représentées par des chiffres.

Enfin, à partir de ces séquences opératoires, qui sont une donnée dans cet article, la stratégie de la détermination a posteriori des sous-assemblages est appliquée en fonction des critères qui stipulent que 2 sous-assemblages n'aient pas d'actions fonctionnelles communes, et que les composants mis en jeu soient différents et que l'assemblage de ces 2 sous-assemblages permettent de tendre vers le produit fini.

Cette méthodologie proposée par cet article a des limites: description incomplète du processus d'assemblage, regroupement de la liaison mécanique et physique en un seul terme liaison fonctionnelle car l'action fonctionnelle n'est pas entièrement intrinsèque au produit.

### **2.1.2. Article 2: Simplified Generation of All Mechanical Assembly Sequences [19]**

Dans cet article de De Fazio et Whitney, la méthode de Bourjault a été améliorée. Cette amélioration rend facile la technique de détermination des séquences d'assemblage pour des produits avec un nombre de composants élevés. La différence majeure entre la méthode de Bourjault et celle de l'auteur se situe au niveau des questions à poser à l'utilisateur.

La méthode de Bourjault présentée consiste à caractériser l'assemblage par un réseau à partir des informations contenues dans la liste des composants et le schéma de définition du produit. Ce réseau comporte des noeuds (composants) et des arêtes (liaisons) reliant 2 noeuds.

Pour générer une séquence d'opérations avec cette méthode, une série de questions sont posées à l'utilisateur qui répond par un oui ou un non. Les réponses incertaines constituent des états interdits.

Quant à la méthode de l'auteur de l'article, elle comporte 2 caractéristiques principales: le nombre réduit de questions à poser à l'utilisateur et la validation directe des relations de priorité en introduisant un petit module qui étudie les conditions d'établissement des liaisons. Pour des questions incertaines, l'utilisateur répond en prose. Sa réponse est transformée par la suite en une forme symbolique logique. La comparaison de ces 2 méthodes faite dans cet article met en évidence des points communs et des différences.

Ainsi comme points communs, les deux méthodes utilisent les relations de priorité pour déterminer les séquences d'assemblage et posent des questions à l'utilisateur pour acquérir de la connaissance et ainsi produire ces relations de priorité.

La liste partielle des états interdits pour n'importe quelle action obtenue avec la méthode de Bourjault et les relations de priorité déterminées avec la méthode de l'auteur sont identiques sur le plan de la forme et du contenu des informations.

Comme différence, la méthode de Bourjault pose plus de questions que la nouvelle méthode, la réponse aux questions doit être oui ou non tandis que celle de la méthode améliorée est soit rien ou soit des relations de priorité.

Le nombre de questions à poser et le contenu des questions sont connus à l'avance dans Bourjault et non dans la méthode améliorée.

Dans Bourjault il existe une étape intermédiaire avant de déterminer une relation de priorité tandis que dans la méthode améliorée cela se fait directement sans étape intermédiaire. Plus de 18 liaisons et 11 composants dans un produit rendent complexe l'utilisation de la méthode de Bourjault tandis qu'avec la méthode améliorée, le traitement se fait facilement et prend peu de temps.

Les séquences d'opérations obtenues comportent les relations de priorité basées sur la géométrie et les dimensions du produit. L'article propose d'autres contraintes qui consistent à faire de l'assemblage connexe de certains sous-assemblages ou composants, ordonnancement de la séquence d'assemblage des composants du sous-groupe, l'assemblage de groupe de composants sans tenir compte de l'ordre dans le groupe, l'assemblage doit satisfaire un ou plusieurs états spécifiques des sous-assemblages (stabilité, temps et coût d'assemblage, probabilité de bris, équipements, etc), pour réduire les choix possibles.

### **2.1.3. Article 3: An integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products [2]**

Cet article de Baldwin et al exposent le cycle de la détermination des gammes d'assemblage qu'on retrouve dans la plupart des travaux sur l'assemblage. La figure 2.3 présente ce cycle d'assemblage.

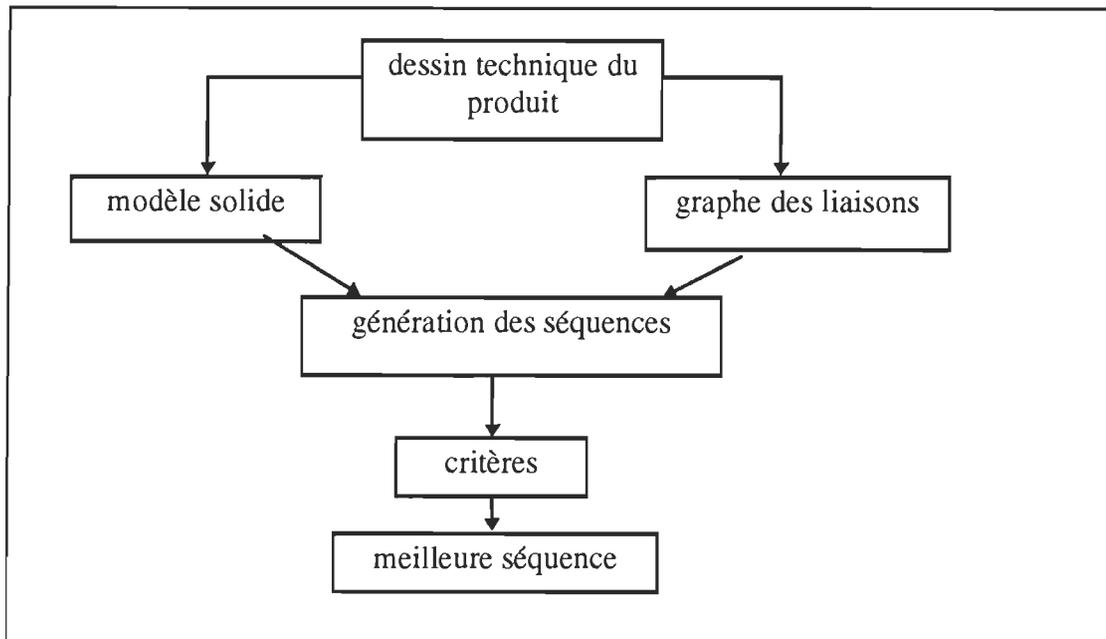


Figure 2.3: cycle d'assemblage

En effet, il commence par le design du produit (dessin, liste des composants), représentation du modèle du produit en terme de connection des composants et graphe de liaisons, génération de la séquence en posant des questions à l'utilisateur pour obtenir des relations de priorité. Le programme informatique de Lui présenté dans cet article est basé sur ce cycle.

Ce qui conduit à l'obtention de toutes les séquences d'assemblage possibles. L'application des contraintes et des critères réduit le nombre de gammes réalisables. Un rebouclage au début du design du produit est possible si d'autres détails ont été considérés au cours de l'assemblage.

La génération de toutes les séquences d'assemblage est réalisée à l'aide de la méthode de rupture des liaisons liant les composants développée par Homem de Mello et Sanderson [41] pour trouver et représenter toutes les contraintes géométriques et mécaniques comme relations de priorité. La figure 2.4 présente cette méthode de rupture des liaisons à partir d'un graphe des liaisons d'un produit quelconque ayant 4 composants.

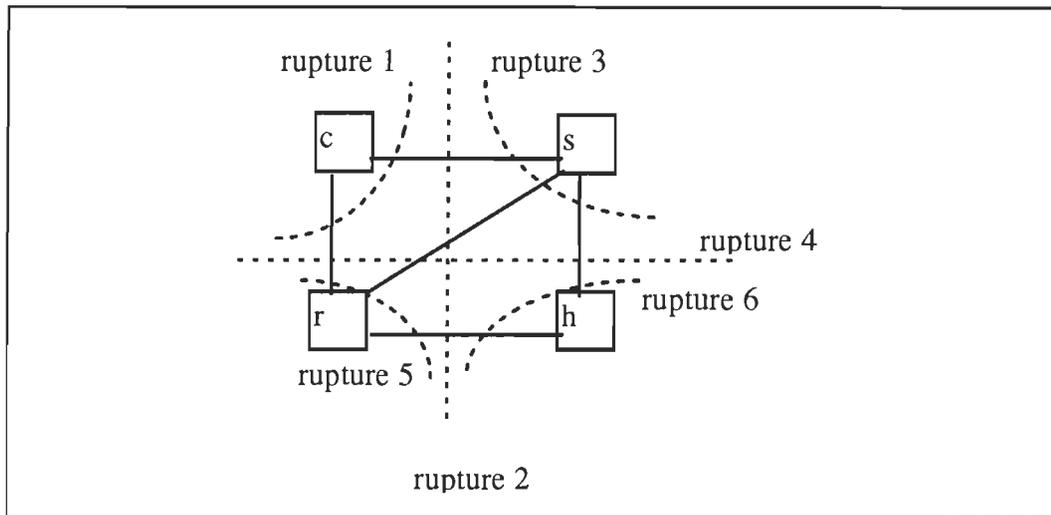


Figure 2.4: exemple de rupture des liaisons

L'algorithme de Lui permet dans un premier temps l'introduction des informations contenues dans le graphe des liaisons pour ensuite générer tous les sous-assemblages possibles et vérifier toutes les connections de ces sous-assemblages ainsi formés. Puis l'algorithme détermine les relations de priorité par des questions-réponses sur le désassemblage des sous-assemblages obtenus auparavant. Ensuite il décèle les états impossibles d'assemblage. Et il aboutit ainsi à l'obtention de toutes les séquences d'assemblage possibles.

Après l'obtention de ces séquences, l'utilisateur dispose d'un environnement de simulation à deux options: l'une pour sélectionner quelques séquences meilleures du point de vue de l'utilisateur et qui sont ensuite évaluées rigoureusement pour détecter des problèmes d'assemblage. L'autre pour évaluer et montrer individuellement toutes les séquences en produisant un coût fictif alloué basé sur des critères de refixation et de réorientation.

Cet environnement est supporté par une représentation graphique en temps réel des différentes étapes quelle que soit l'option choisie.

#### **2.1.4. Article 4: Planning of Products Assembly and Disassembly [58]**

Cet article Laperrière et ELMaraghy exposent une approche qui consiste à générer une liste de toutes les séquences possibles par la méthode de la rupture des liaisons du graphe des liaisons. La figure 2.5 montre un exemple de cette méthode pour un produit de 4 composants.

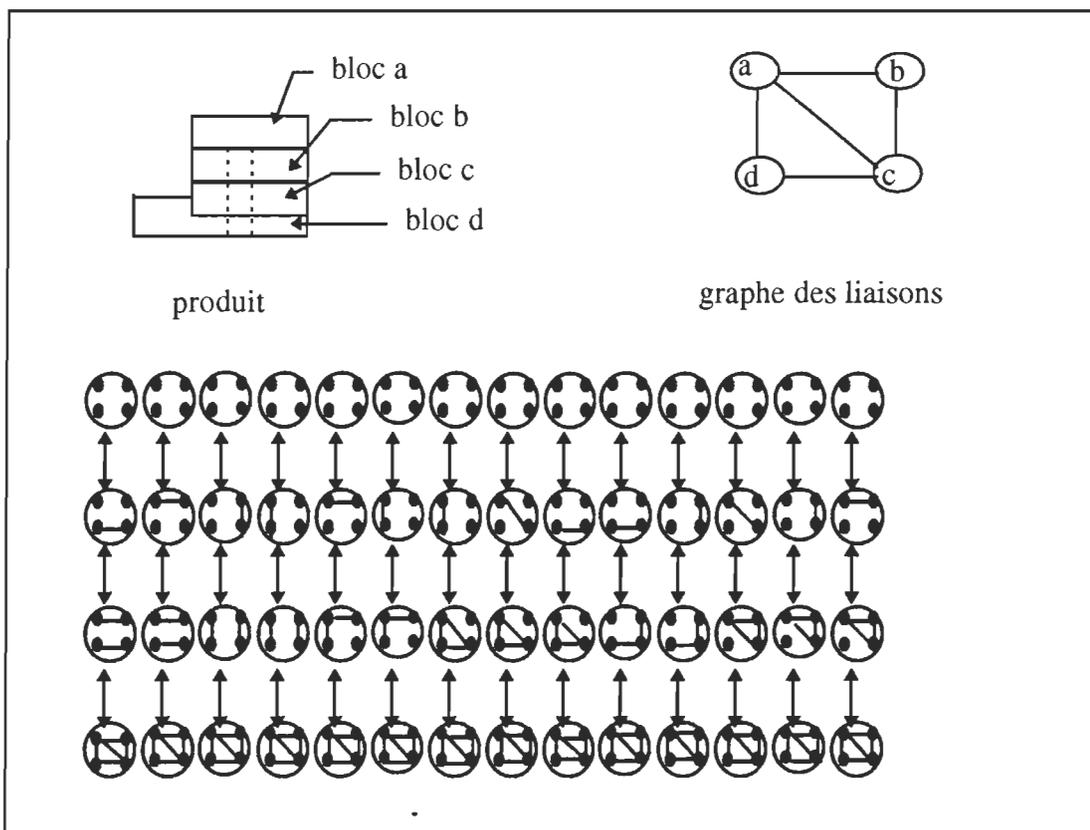


Figure 2.5: exemple de génération de séquences

Cette liste est ensuite réduite par le regroupement des noeuds identiques en un seul comme le montre la figure 2.6.

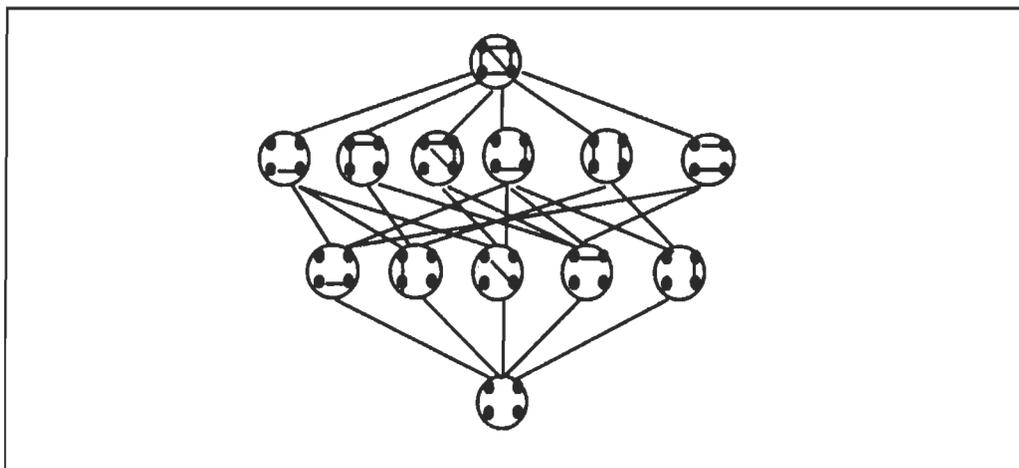


Figure 2.6: regroupement des noeuds identiques

Dans le but de réduire encore davantage la taille de la liste des séquences, les contraintes géométriques et d'accessibilité sont utilisées (figure 2.7).

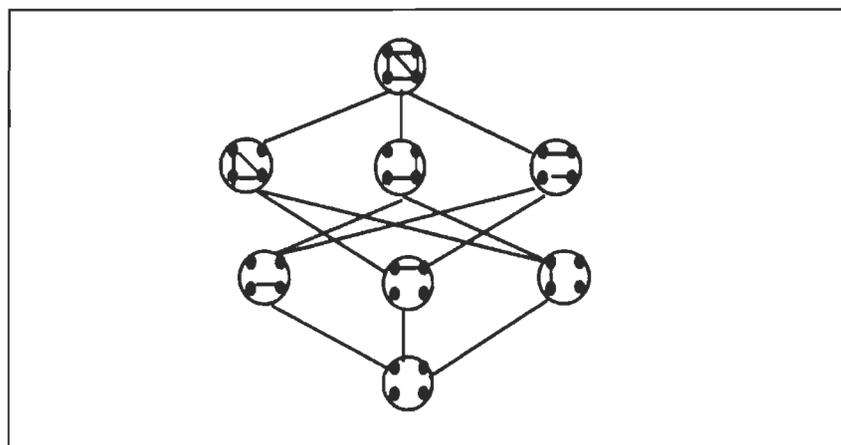


Figure 2.7: réduction de la taille des séquences

En effet, la méthode basée sur les contraintes géométriques consiste à l'analyse des degrés de liberté de 2 composants en fonction des liens de contact, attachement ou blocage dans une opération de désassemblage.

L'évaluation des critères identifiés à savoir la réorientation réduisant le temps d'assemblage, le parallélisme, réduisant le temps total d'assemblage, la stabilité réduisant le temps de cycle, le regroupement des tâches similaires qui réduit le temps de préparation du matériel d'assemblage et le nombre de changement d'outils sont utilisés dans la recherche de la séquence optimale d'assemblage.

La considération de la réorientation dans GAPP ( Generative Assembly Process Planner) est basée sur 2 hypothèses: la mise en contact d'un sous-assemblage mobile à d'un autre fixe et le sous-assemblage mobile ne doit pas être assemblé par le dessous.

GAPP compte le nombre de sous-assemblages fixes et mobiles et décide la réorientation par comparaison de ces nombres. En cas d'égalité, GAPP se base sur la direction de désassemblage. GAPP détermine aussi le coût associé à la réorientation en évaluant le nombre de réorientation dans les séquences.

Le parallélisme est déterminé dans cet article en comptant le nombre de composants dans deux sous-assemblages résultant d'une opération de désassemblage. La stabilité est détectée ici par le biais des degrés de liberté qui sont au nombre de six mais GAPP ne considère que les 3 translations. Plus le nombre de degrés de liberté d'un sous-assemblage est moindre plus le sous-assemblage est stable. Le coût associé à la stabilité est calculé en évaluant les degrés de liberté de chaque composant au moyen d'une fonction linéaire.

Le regroupement est réalisé en associant l'opération de rupture des liaisons à une opération type. Ce regroupement est évalué par une fonction constante.

L'utilisation de l'algorithme A\* dans cet article permet de trouver le chemin optimal en utilisant les coûts associés aux critères cités plus haut dans une fonction linéaire.

Dans un processus automatique d'assemblage, la méthode exposée dans cet article permet de déceler les erreurs d'assemblage et proposer une nouvelle séquence d'assemblage.

#### **2.1.5. Article 5: Automated Sequencing and Subassembly Detection in Assembly Planning [23]**

Dans cet article de Dini et Santochi, une représentation des liaisons existant entre les composants d'un produit est faite sous forme matricielle d'où une matrice d'interférence, de contact et la dernière de connexion suivant une direction donnée. Les éléments de ces matrices sont des valeurs binaires déterminés en fonction de la relation de contact entre les composants du produit comme l'indique la figure 2.8.

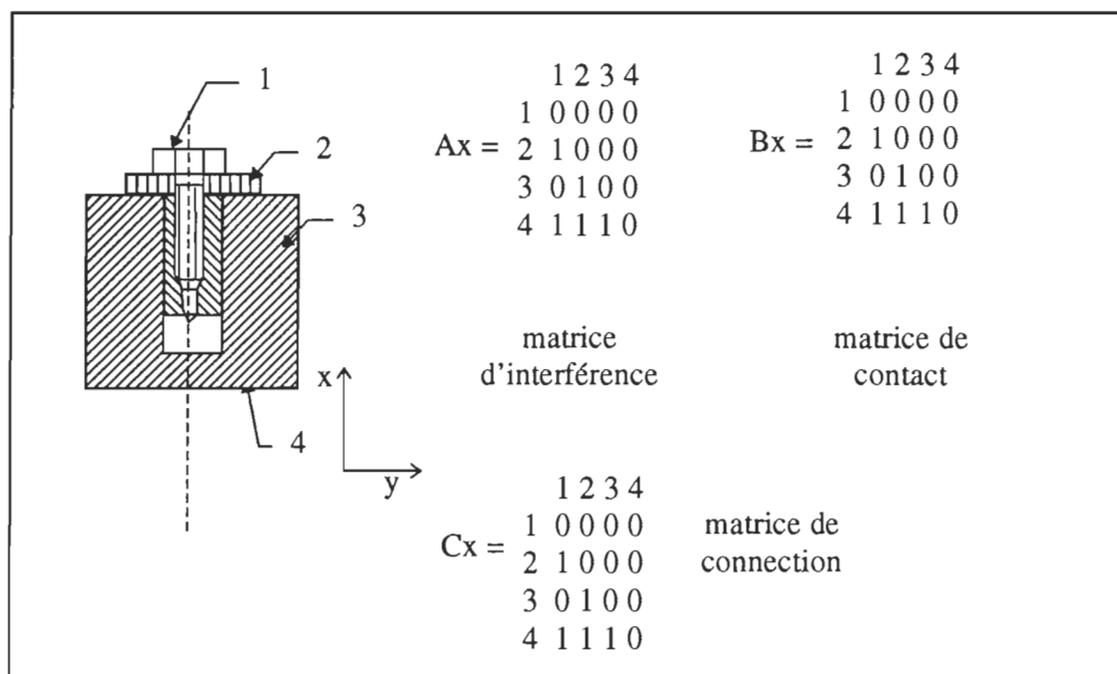


Figure 2.8: exemple de représentation des liaisons

A partir de ces matrices obtenues, les sous-assemblages ( figure 2.9) sont détectés en fonction de 3 conditions: chaque sous-assemblage doit produire une matrice contractée formée à partir de la sous-matrice extraite de la matrice d'interférence, de contact ou de connection, les composants du sous-assemblage doivent constituer un ensemble stable et le sous-assemblage ne doit pas empêcher l'assemblage des autres composants restants.

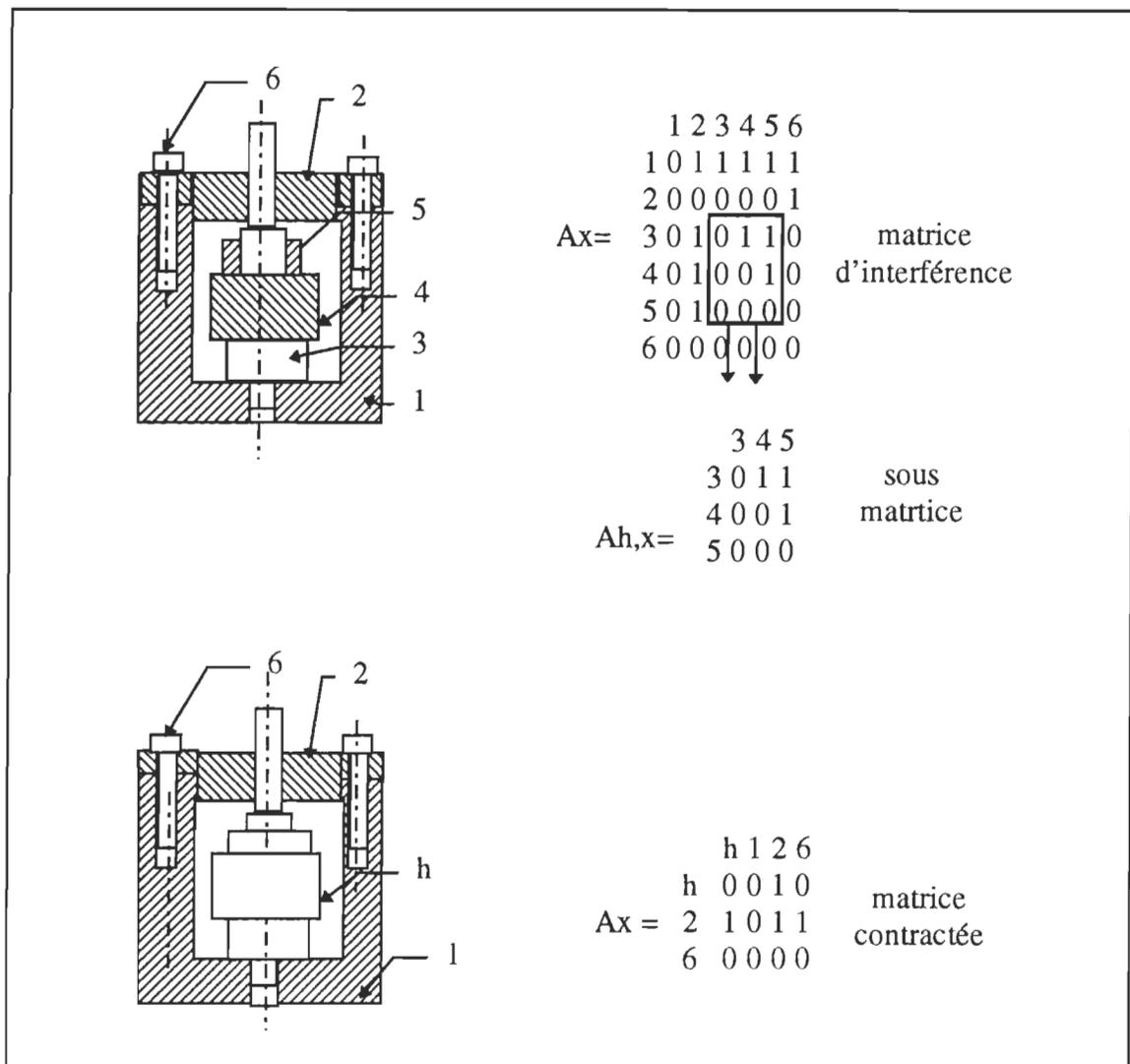


Figure 2.9: Détection des sous-assemblages

Pendant la génération des sous-assemblages, une vérification de la condition de désassemblage, exigeant que les composants d'un sous-assemblage ne doivent pas produire d'interférence avec les autres composants restants et que la valeur des éléments de la matrice de connection soit nulle, est effectuée.

Cette méthode ne tient compte que de 3 directions au lieu de 6 directions. Néanmoins elle a pour avantage de permettre une génération facile des séquences et des sous-assemblages déjà préassemblés.

### 2.1.6. Article 6: A Pragmatic Approach to Computer Aided Assembly Planning [20]

Dans cet article de Delchambre, dans un contexte général de l'assemblage, il existe 2 notions:

- off-line qui comprend la conception de l'assemblage (génération des séquences d'assemblage à l'aide des informations géométriques, non-géométriques et du plan d'exécution affectant les opérations d'assemblage aux différentes stations de travail.
- on-line qui contrôle l'exécution des opérations d'assemblage.

Concernant la première notion, le processus d'assemblage d'un produit mis en oeuvre dans cet article débute premièrement par la modélisation du produit final qui donne des informations géométriques (la forme, les dimensions de chaque composant et les coordonnées de ces composants).

La méthode de l'approximation des composants à des formes géométriques usuelles tel que le parallélépipède est utilisée pour faire cette modélisation (figure 2.10). Dans cette figure, le parallélépipède est représenté par les coordonnées extrêmes de ses sommets.

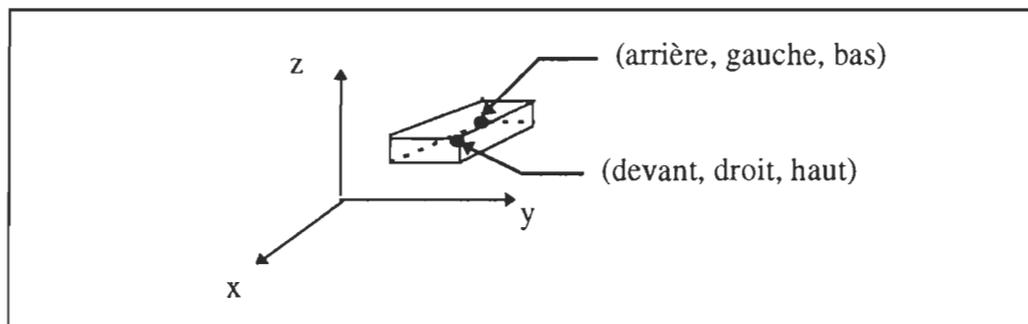


Figure 2.10: Approximation d'un composant à un parallélépipède

Deuxièmement les informations non-géométriques sont fournies par l'utilisateur. Par exemple pour la vis, l'utilisateur doit donner le sens de vissage.

Troisièmement le module géométrique fournit la liste des composants qui provoquent l'interférence. La méthode de comparaison des positions (figure 2.11) permet de traiter ce problème.

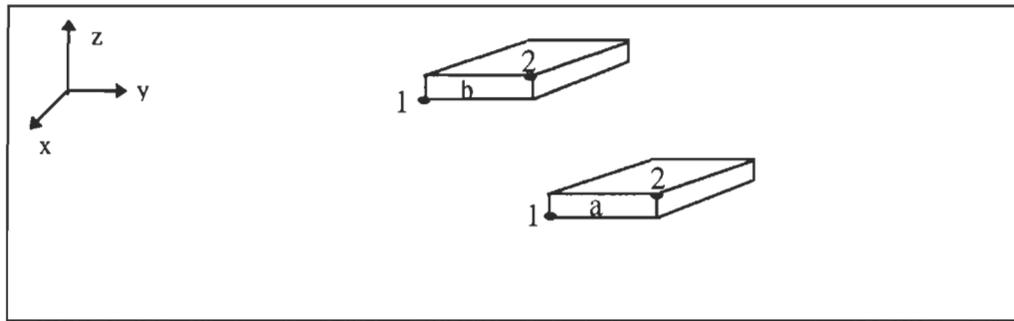


Figure 2.11 : méthode des comparaisons

a est sous b si :

- $bx1 < ax2$
- $by1 < ay2$
- $bx2 > ax1$
- $by2 > ay1$
- $by2 \geq az2$

Quatrièmement la détermination des informations relationnelles. Dans cet article, les liaisons sont repérées et séparées en sous-groupe: contact (composant ayant plus d'un degré de liberté et instable dans toutes les directions), insertion (composants avec un degré de liberté et instable dans une seule direction) et l'attachement (pas de mouvement). La figure 2.12 montre un exemple.

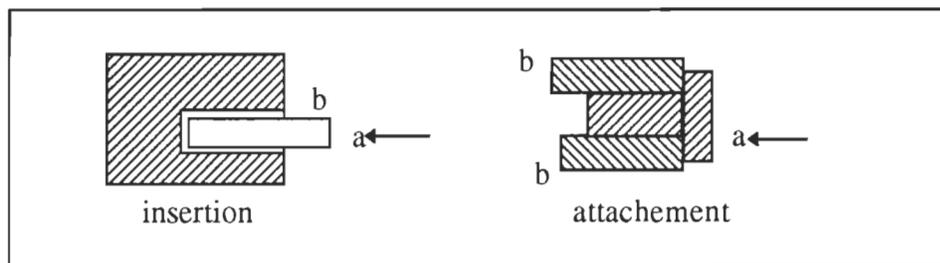


Figure 2.12: insertion et attachement

Les contacts physiques et le type de liaison sont déterminés automatiquement par la méthode de superposition et de proximité.

Afin de générer les séquences d'assemblage, les contraintes qui sont au nombre de quatre sont les contraintes de géométrie qui produisent les relations de priorité et les liaisons non réalisables physiquement.

Les contraintes mécaniques modifient les relations de priorité obtenues auparavant.

Les contraintes de stabilité sont déterminées à partir des liaisons de vissage et des liaisons instables (les insertions et les fixations).

Les contraintes technologiques sont obtenues à l'aide des informations fournies par l'utilisateur sur les équipements.

À l'issue de toutes ces étapes, le processus produit les séquences d'assemblage ayant comme informations disponibles les relations de priorité obtenues par l'utilisation des contraintes géométriques, mécaniques, de stabilité, technologiques.

## **2.2. Analyse de l'état de l'art**

### **2.2.1. Modélisation du produit**

Pour modéliser un produit donné, Bourjault et Henrioud [9] utilisent une représentation des relations entre les composants suivant le graphe de contact.

Le graphe de contact étant défini comme un graphe dans lequel les liaisons mécaniques entre les différents composants sont identifiées, Bourjault et Henrioud définissent le graphe de connexion à partir de ce principe.

Le graphe des liaisons comporte des nœuds qui représentent les composants et les arêtes reliant chaque nœud qui désignent un contact.

La modélisation du produit par un diagramme de liaison est aussi utilisée dans certains articles. Le diagramme de liaison est un diagramme où chaque composant est désigné par un point, chaque arête représente la liaison entre chaque paire de composants. Dans les articles [19], [8], [103] cette représentation est utilisée.

Dans l'article de Homen-de-Mello [44], la méthode du modèle relationnel est utilisée. Ce modèle est la combinaison d'un graphe et des fonctions attribuées. Dans ce graphe, les rectangles désignent les composants, les cercles les contacts entre les composants et les triangles les attachements. Des lignes relient ces différents éléments pour ainsi former le graphe. Ce graphe a pour avantage de considérer, les éléments d'attachement: les vis ou les boulons, comme des composants du produit ou non. La figure 2.13 montre un exemple de graphe relationnel.

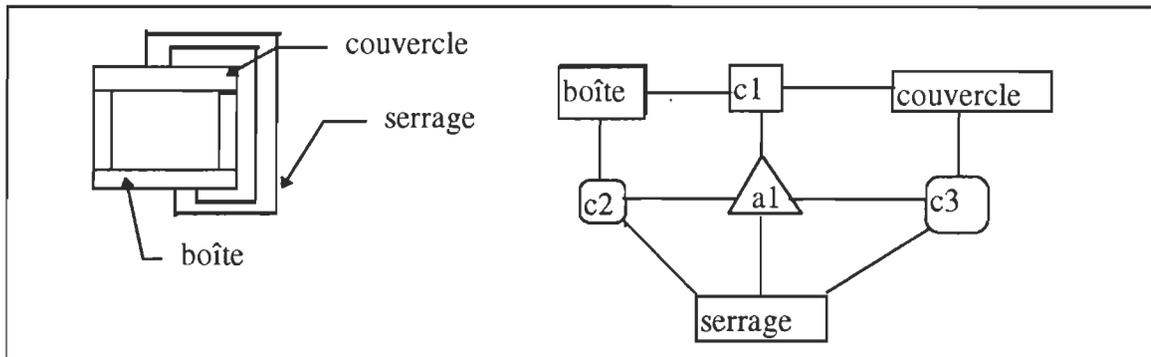


Figure 2.13: modèle relationnel d'un produit à 3 composants

## **2.2.2. Génération des séquences d'assemblage**

### **2.2.2.1. Génération des séquences à l'aide des contraintes de priorité**

Pour générer les séquences d'assemblage, plusieurs approches sont à retenir. La première approche consiste à générer les séquences au moyen des contraintes de priorité. Les contraintes de priorité sont obtenues en prenant en compte l'assemblage ou le désassemblage du produit.

Bourjault [9] utilise la génération des séquences basée sur l'idée d'assemblage. Il pose des questions à l'utilisateur pour déterminer les contraintes de priorité. Questions auxquelles l'utilisateur doit répondre par un oui ou un non. Le nombre maximal de questions à poser est de  $2^L$  avec  $L$  pour le nombre de liaisons. Cette approche n'est applicable qu'aux produits ayant au plus 15 composants.

Whitney et De Fazio [19] ont amélioré la méthode de Bourjault en réduisant le nombre de questions à poser à l'utilisateur à  $2*L$  et cette méthode est applicable à des produits ayant un nombre élevé de composants.

Bourjault et Henrioud [23] utilisent une méthode exploitant le graphe de liaisons fonctionnelles qui permet de mettre en évidence les contraintes de réalisation, exprimées sous forme de relations n-aires  $R$  et  $S$ . Ils montrent que la réalisation d'une action fonctionnelle peut être représentée par un réseau de Pétri élémentaire.

Chen [103] utilise aussi la méthode des questions-réponses pour générer les séquences. En plus de poser des questions, il emploie la méthode de l'acquisition de la connaissance de l'ordre de priorité dérivée du problème du commis voyageur.

Delchambre [20] génère les séquences d'assemblage représentées par un graphe où les noeuds représentent les opérations d'assemblage et les arêtes reliant chaque noeud les contraintes de priorité en fonction des opérations. Les données pour générer ce graphe sont les ordres de priorité sur le plan de la géométrie, mécanique, de la stabilité et des contraintes technologiques déterminées auparavant.

#### **2.2.2.2. Méthode de désassemblage**

La génération des séquences à partir des composants élémentaires, donne moins de possibilités lorsque l'opération d'assemblage commence avec le produit final pour se terminer avec des composants élémentaires.

En tenant compte de cette idée, Homem de Mello et Sanderson [42], examinent toutes les possibilités de rompre un assemblage en 2 sous-assemblages. Par l'utilisation récursive de la méthode de la rupture des liaisons du graphe de connection et la vérification de la faisabilité physique de la décomposition, ils parviennent à générer toutes les séquences d'assemblage.

Cette vérification est effectuée par le modèle relationnel. La décomposition utilisée est une décomposition a priori.

Baldwin et al [2], le programme informatique développé génère les séquences d'assemblage par la décomposition en sous-assemblages des composants du produit avec la méthode de la rupture des liaisons.

Lorsque le nombre de composants est élevé, le temps de détermination de toutes les séquences possibles devient extrêmement long. C'est pourquoi certains auteurs se sont intéressés à l'idée de regrouper les composants du produit en sous-assemblage pour réduire le temps d'exécution.

Ainsi, Bourjault et Henrioud [9], déterminent les sous-assemblages par la stratégie a posteriori. Donc à partir des séquences d'assemblage, le critère de l'exécution indépendante de deux sous-assemblages est appliqué pour obtenir les sous-assemblages valides.

Dini et Santochi [23] ont représenté, suivant un modèle mathématique, le produit. Ce modèle comprend 3 matrices: interférence, contact et connection. A partir de cette modélisation, ils détectent les sous-assemblages suivant 3 conditions: l'existence d'une matrice contractée issue des 3 matrices, le sous-assemblage doit former une structure stable et que le sous-assemblage n'empêche pas l'assemblage des autres composants. Cette méthode a pour avantage de prendre en considération les sous-assemblages déjà préassemblés.

#### **2.2.2.3. Autres méthodes**

Ces méthodes concernent les articles qui n'utilisent pas la démarche conventionnelle mais plutôt des heuristiques, des systèmes de bases de données, un raisonnement géométrique pour générer les gammes.

En effet, Delchambre [20] génère les séquences d'assemblage suivant l'approche des contraintes de priorité en fonction des composants. Cet article détermine tout d'abord les contraintes qui sont des informations indispensables à la génération des séquences.

Ces contraintes sont au nombre de quatre: géométriques, mécaniques, de stabilité et technologiques. L'utilisation des heuristiques et le concours de l'usager permettent de les définir. Le logiciel mis en place suivant cette approche, fournit des séquences pouvant comprendre des parallélismes.

Parlant de parallélisme, Hoummady et Ghosh[47] se sont intéressés spécialement à ce sujet. En effet, cet article définit 2 types de simultanéité: simultanéité optionnelle et obligatoire. Il existe 3 types de simultanéité obligatoire: systématique, conditionnelle et automatique.

La détermination de ces simultanéités se fait en fonction des états interdits dans le processus d'assemblage et matérialisés par les relations R et S sous forme de réseau Pétri.

Wolter [103] emploie trois étapes pour résoudre le problème de la conception de l'assemblage: proposition des trajectoires d'insertion, construction des contraintes géométriques qui consiste à trouver quels composants provoquent une interférence au moment du montage pour chaque trajectoire proposée de chaque composant, générer le plan en sélectionnant une trajectoire pour chaque composant et ordonner toutes les trajectoires.

Un logiciel a été développé suivant cette méthode de résolution du problème et donne des résultats qui sont conditionnés par certains paramètres (résultats meilleurs si nombre de contraintes élevé).

Dans l'article de Campagne [11], la méthode des variantes est utilisée pour élaborer automatiquement les gammes d'assemblage. Cette méthode est basée sur la définition des séquences d'assemblage types associées à des familles de pièces et sur l'identification successive des contraintes logiques et physiques de réalisation spécifiques à chaque produit à réaliser.

Cette méthode élimine les tâches répétitives et permet de formaliser, de structurer et de mémoriser les connaissances en une base de données.

Giraud et Sidore [25], présentent un algorithme basé sur la méthode de rétraction des mouvements planaires. Cette méthode de rétraction génère des trajectoires pour l'assemblage des objets polyèdres.

### **2.2.3. Représentation des séquences d'assemblage**

La représentation des séquences d'assemblage étant une liste ordonnée des tâches ou des opérations à effectuer, plusieurs méthodes ont été développées afin de permettre une bonne visualisation des séquences.

Wolter [104] s'est attardé sur ce problème. Il analyse et compare les différentes méthodes de représentation des séquences d'assemblage et il définit un plan d'assemblage en 3 termes:

- une séquence d'état vue comme une séquence d'opérations d'assemblage donnant plus de détails sur la disposition des sous-assemblages entre eux. La figure 2.14 montre un exemple d'une représentation des séquences sous forme d'état.

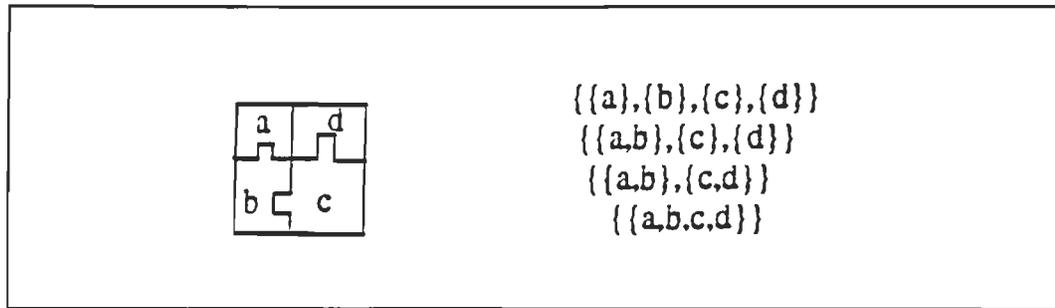


Figure 2.14: séquences d'état d'un produit quelconque

- l'arbre d'assemblage partiel défini comme étant une décomposition récursive de l'assemblage en 2 sous-assemblages (figure 2.15). C'est une méthode qui donne moins de détails.

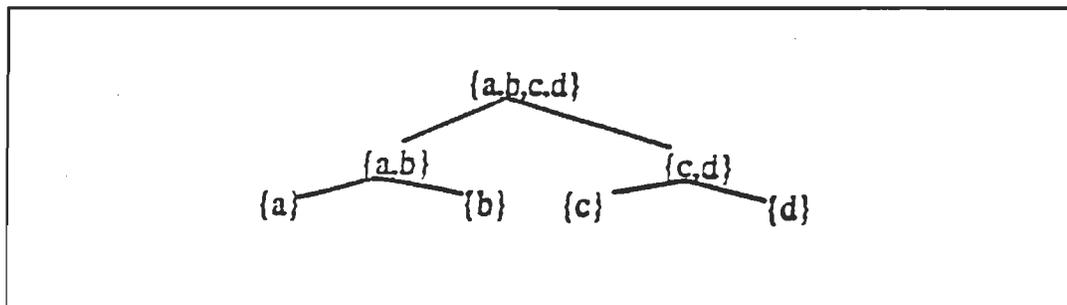


Figure 2.15: arbre d'assemblage

- l'arbre des sous-assemblages qui est une séquence d'opérations consistant à introduire les composants ou les sous-assemblages dans leur fixation. C'est une méthode qui donne plus d'informations sur la structure de l'assemblage et l'ordre des sous-assemblages comme l'indique la figure 2.16.

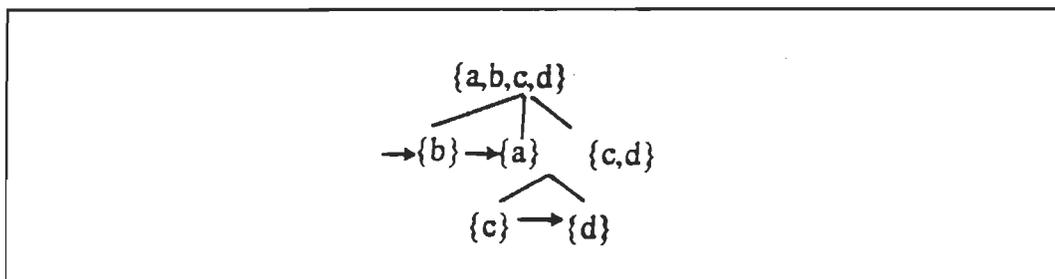


Figure 2.16: arbre des sous-assemblages

A partir des séquences d'états, Bourjault [9] a développé une représentation des séquences sous forme d'arbre où il existe une relation de père et fils entre les états de l'arbre (figure 2.17).

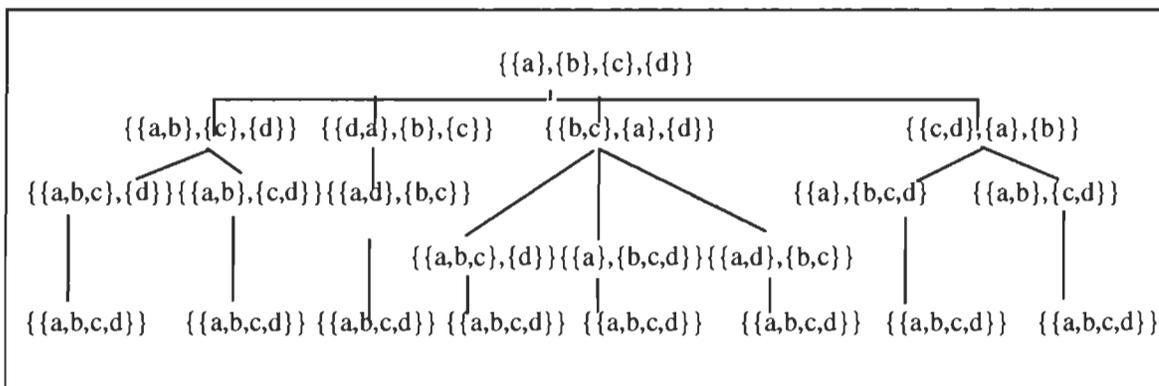


Figure 2.17: Arbre d'états

Quant à De Fazio et Whitney [19], ils ont amélioré l'arbre de Bourjault en une représentation de graphe dirigé où dans chaque état du graphe tous les composants du produit sont cités. Cet arbre est plus compact que celui de Bourjault. La figure 2.18 montre un exemple de cet arbre.

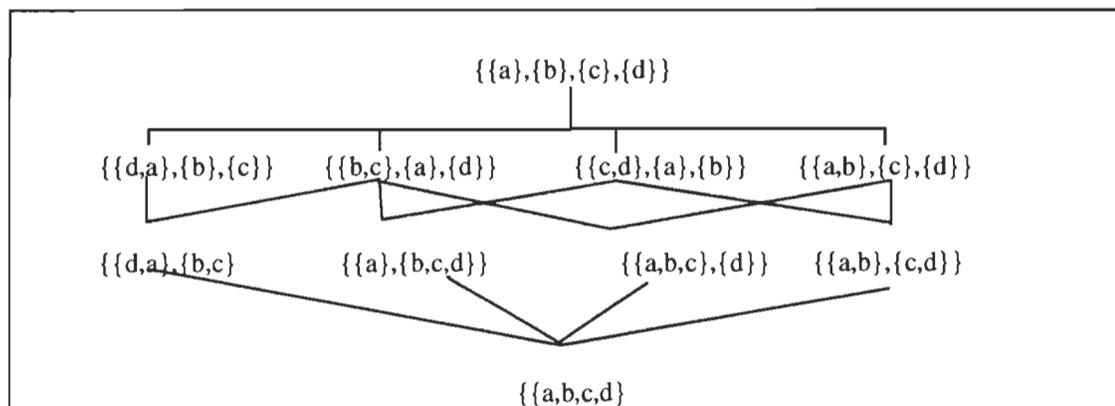


Figure 2.18: Graphe dirigé

A partir de la représentation en arbre d'assemblage partiel, l'arbre AND/OR a été développé dans lequel dans chaque état de l'arbre seuls les composants concernés dans l'établissement de la liaison sont cités (figure 2.19).

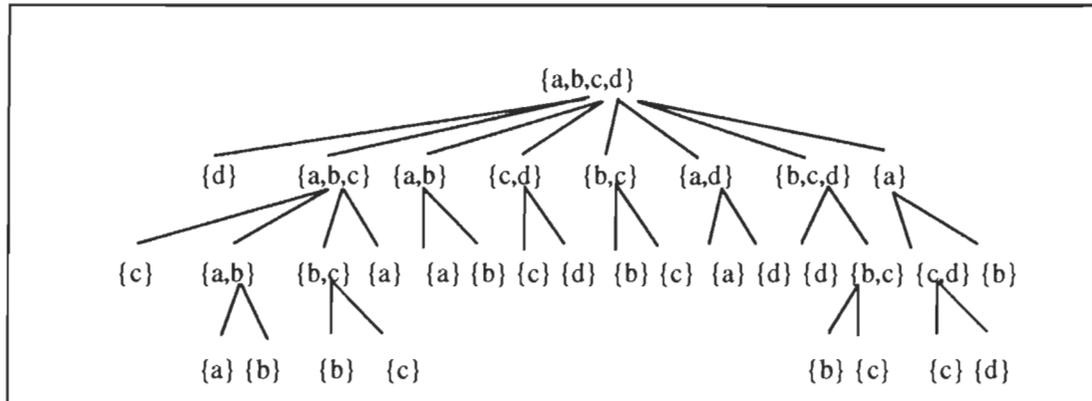


Figure 2.19: Arbre AND/OR

Homem et Sanderson [41] ont développé l'arbre AND/OR en introduisant la notion d'indépendance historique. Dans ce graphe seuls les composants concernés par la liaison sont mentionnés (figure 2.20).

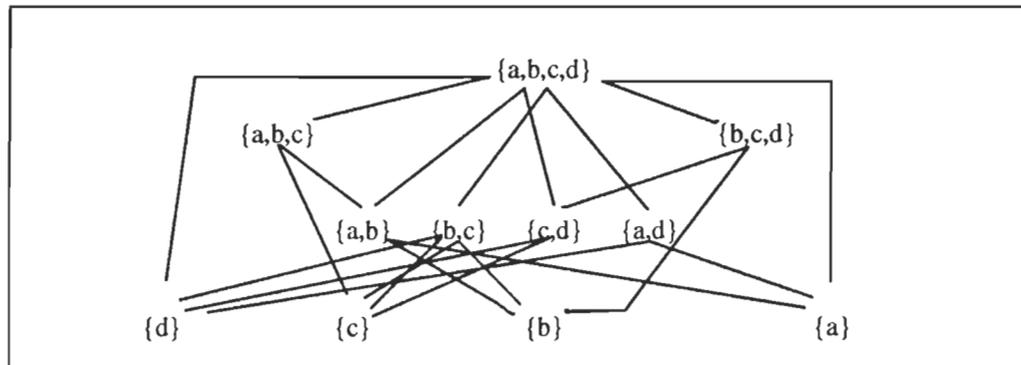


Figure 2.20: graphe AND/OR

Dans cet article, retenons les travaux de Whitney et De Fazio qui ont amélioré l'arbre de Bourjault en une représentation en graphe dirigé. Dans ce graphe, tous les composants sont nommés dans chaque état du processus d'assemblage.

La représentation la plus compacte est le graphe AND/OR mais il a plus de noeuds que le graphe dirigé. En plus le graphe AND/OR montre une possibilité d'exécution des opérations parallèles. Dans ce graphe, seuls les composants concernés par la liaison sont cités.

Dans l'article de Henrioud et Bourjault [32], le graphe de priorité qui est une autre représentation des gammes ne permet pas de rendre compacte la représentation. C'est un graphe où les opérations sont désignées par des noeuds et les relations de priorité par des

flèches. Ce graphe est obtenu par les liaisons à établir avant et après pour une liaison considérée (figure 2.21).

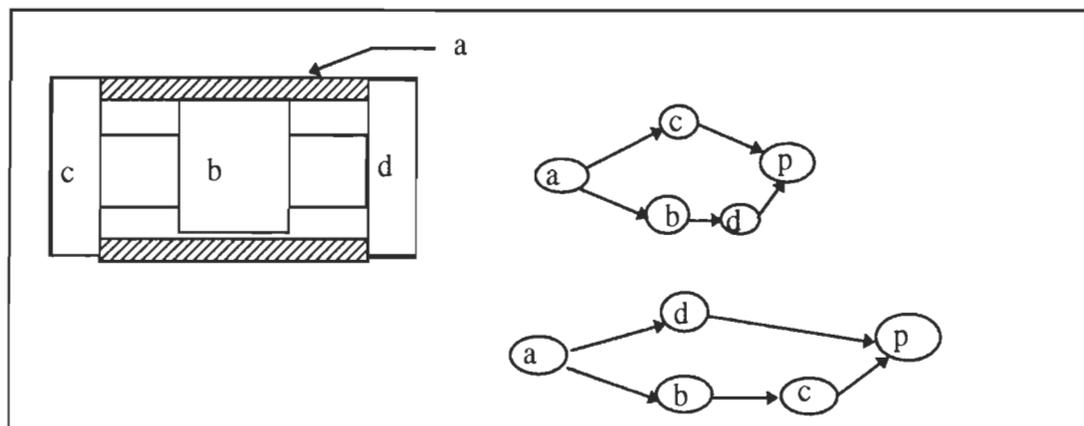


Figure 2.21: graphe de priorité d'un produit p

Dans l'article de Bourjault et Henrioud [8] [19], une représentation aussi compacte est utilisée où les boîtes du graphe représentent les états du processus d'assemblage, les lignes de connection entre les boîtes représentent les états de transition (figure 2.22).

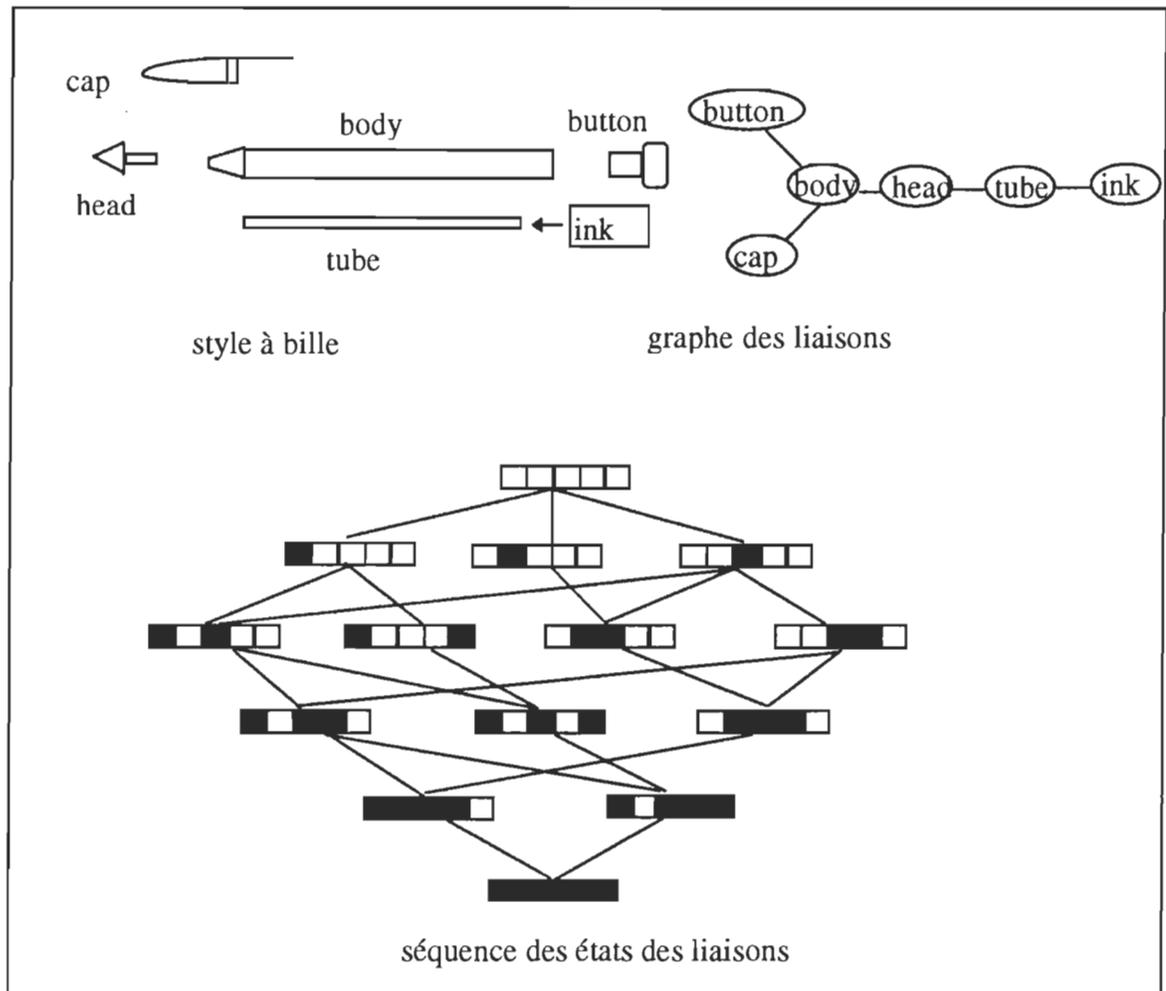


Figure 2.22: exemple de représentation des séquences

Le nombre de séquences obtenu à cette étape de la méthodologie, est énorme et il faut réduire ce nombre pour ne retenir qu'une ou quelques séquences bonnes et optimales.

C'est pourquoi ces séquences sont testées à l'aide des contraintes pour éliminer les séquences indésirables.

Dans les articles de Bonneville et al [6], Henrioud et Bourjault [32] les contraintes utilisées sont du type:

- géométrique (vérifier l'impossibilité d'union de 2 composants, un composant et un sous-assemblage ou 2 sous-assemblages). L'approche utilisée [6] est le déplacement des composant tandis que Boujault et Henrioud utilisent des règles.
- de stabilité: les composants doivent être stables pendant le processus d'assemblage.
- matériel: existence d'une trajectoire pour les outils d'assemblage.

Boneschanscher [4] s'est intéressé spécialement à la stabilité. Dans cet article la stabilité de chaque composant est calculée à l'aide d'un algorithme basé sur les méthodes de détermination des réactions du modèle adopté.

Cet algorithme formalise le modèle adopté en des équations statiques en utilisant des formules mathématiques empruntées à des auteurs tels que Erdmann et Minkowski. Ensuite la programmation linéaire et la programmation simplexe appliquées à ces équations permettent de déterminer la stabilité des composants.

Quant à Hoffman [36], il utilise un engin BRAEN pour générer les séquences d'opérations pour assembler les produits. Les composants sont représentés comme des objets B-Rep avec des surfaces exprimées au moyen de modèle de Bezier.

Cet article fait la distinction entre 2 classes de composants: environnement et produit. Les produits sont assemblés dans un environnement dans lequel se trouve des composants environnements. L'entrée de l'environnement reçoit les composants produits à désassembler. Après l'application de BRAEN, les composants sont stockés à la sortie de l'environnement.

Krishnan et Sanderson[56], emploient un raisonnement sur la faisabilité des mouvements du composant qui sont sujets à des contraintes de géométrie par rapport aux autres composants, sous-assemblages, fixations et autres équipements. Le problème d'assemblage dans cet article est considéré comme une hiérarchie de tests de faisabilité. L'introduction de l'algèbre des cônes polyédriques qui est un outil, permet de combiner les contraintes géométriques et les différentes relations entre les composants.

Popplestone et al [82] exposent une méthode qui consiste à utiliser la symétrie du modèle géométrique des composants. L'outil mathématique utilisé dans cet article pour obtenir cette symétrie est la théorie de groupe. En effet, à un niveau élevé de l'assemblage des composants, du point de vue robotique, il faut déterminer comment les corps sont mis en contact et comment les relations spatiales sont établies.

### 2.3. Proposition du sujet de recherche

#### 2.3.1. Graphe des liaisons

Dans la plupart des travaux sur l'assemblage, les graphes de liaisons sont obtenus, à partir du schéma du produit à assembler, d'une manière manuelle.

En effet, les auteurs transforment le schéma du produit à assembler en graphe de liaisons en identifiant toutes les liaisons entre les différents composants impliqués. Cette méthode manuelle de génération du graphe prend énormément de temps car il faut examiner minutieusement le schéma pour faire une bonne description des relations entre les composants.

Il est donc nécessaire de faire afficher le graphe des liaisons d'une manière automatique. Conscients de cette nécessité, nous nous proposons de **déterminer automatiquement les graphes de liaisons.**

Pour résoudre ce problème, nous appliquerons une autre approche qui consiste à utiliser les informations contenues dans le fichier B-Rep (Boundary representation). Ce fichier s'obtient automatiquement après la modélisation solide des composants. Dans ce fichier, les contours, les arêtes, les sommets et la géométrie des surfaces de chaque face, la courbe des surfaces des composants sont décrits par la combinaison des chiffres, des signes de dollar et des dièses (figure 9.2 du chapitre 9). L'interprétation et la transformation des informations de ce fichier se fera par un programme informatique.

### **2.3.2. Détermination de la séquence optimale d'assemblage**

A la lumière des différents travaux lus, beaucoup d'efforts ont été mis par des auteurs afin de générer des séquences d'assemblage réalisables après l'utilisation des contraintes (interférence, stabilité, technologique). Parmi les séquences d'assemblage générées, seule une séquence doit être sélectionnée pour obtenir le produit final.

La sélection de cette séquence d'assemblage doit être optimale en terme de coût et de temps d'assemblage. Cette optimisation motive de nos jours les entreprises à automatiser leur système d'assemblage si elles veulent être compétitives.

Dans cette optique, nous nous proposons de générer une séquence optimale pour produire un outil qui permettra à ces entreprises d'atteindre cette compétitivité.

Pour y arriver, nous proposons de mettre en place un modèle de regroupement des opérations. Ce regroupement se fera en fonction de la direction d'assemblage, de la similarité des outils utilisés pour manipuler les composants à l'intérieur des opérations et des tolérances impliquées dans ces opérations (degré de difficulté d'exécution des opérations dans une séquence).

Deux opérations successives peuvent être regroupées en une seule opération si par exemple la direction d'assemblage de chaque opération est identique. La séquence ayant le plus d'opérations regroupées sera considérée optimale du point de vue de la direction d'assemblage. Ce regroupement se traduit donc dans le cas de la direction d'assemblage par l'exécution des opérations dans la direction dans laquelle plus de regroupements ont été effectués.

La même méthode de regroupement sera utilisée pour les 2 autres définitions. Dans le cas de la similarité des outils, la séquence optimale du point de vue des outils sera une séquence dans laquelle le même outil sera utilisé pour exécuter presque toutes les opérations. En ce qui concerne les tolérances, la séquence optimale en fonction des tolérances, sera la séquence dont le degré de difficulté d'exécution des opérations sera moindre.

En effet, **la détermination d'une séquence optimale basée sur le regroupement des opérations** sera appliquée après l'utilisation des contraintes (géométriques, stabilité, etc..) à l'ensemble des séquences d'assemblage.

## CHAPITRE 3

### GÉNÉRATION AUTOMATIQUE DU GRAPHE DES LIAISONS D'UN PRODUIT DONNÉ

Ce chapitre présente une méthode de génération et d'affichage automatique du graphe des liaisons d'un produit à partir d'un fichier qui contient la représentation géométrique des frontières de tous les composants du produit.

#### 3.1. Modélisation solide des composants du produit

Chaque composant du produit est modélisé en utilisant un modelleur solide commercial (ICEM/DDN) de Control Data. Ce modelleur utilise une représentation d'arrangement hybride. Un modèle solide peut être créé par une des d'entités suivantes:

- une primitive peut être un bloc de forme parallélépipédique, un cône, un cylindre, un solide de révolution, une projection ou un bloc triangulaire.
- un objet s'obtient par l'association de primitives. Cette association est réalisée par des opérations booléennes (union, différence, intersection).

Le modelleur sauvegarde toutes les informations issues de l'analyse de la représentation géométrique des frontières des primitives et des objets dans un fichier appelé fichier B\_Rep. Un exemple de ce fichier est présenté à la figure 9.2 du chapitre 9.

Pour mémoriser le nom des composants, chaque composant est modélisé individuellement tout en conservant la position spatiale qu'elle occupe lorsque le produit est assemblé.

A l'aide de ce fichier B\_Rep, un programme informatique génère le graphe des liaisons.

#### 3.2. Génération du graphe des liaisons

##### 3.2.1. Création du fichier B\_Rep

Avant de générer le graphe des liaisons, la mémorisation des noms des composants s'avère nécessaire. Cette mémorisation des noms est utilisé lors de l'affichage du graphe des liaisons sur un écran d'ordinateur.

Dans un fichier B\_Rep, le nom donné au fichier apparaît à la première ligne. Si tous les composants du produit sont modélisés dans un même fichier, un seul nom sera attribué à l'ensemble des composants.

un fichier unique . Pour illustrer cette méthode, examinons le produit constitué de 4 blocs de la figure 3.1.

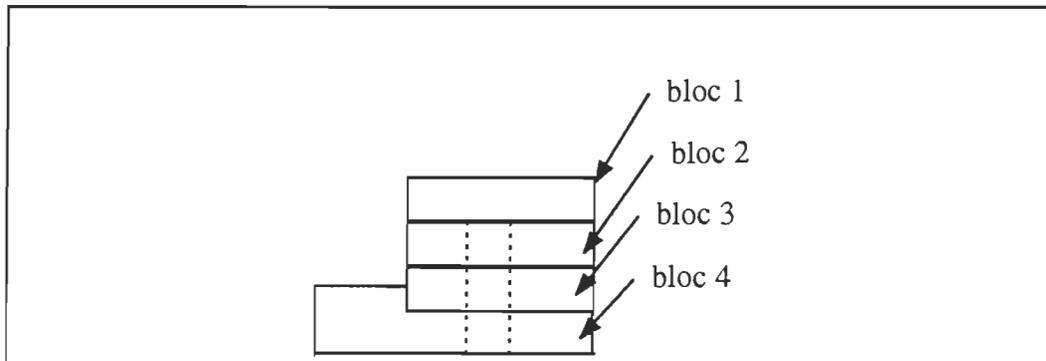


Figure 3.1: produit de 4 composants

La modélisation des 4 blocs donne les fichiers B\_Rep suivants:

composant	nom de fichier B_Rep
bloc 1	bloc1_12tmzk3.brp
bloc 2	bloc2_12tmzed.brp
bloc 3	bloc3_12tmz5z.brp
bloc 4	bloc4_12tmvt1.brp

Ces 4 noms de fichier se trouvent dans un fichier nommé fich\_bloc. L'algorithme make\_file\_b\_rep reconstituant le fichier B\_Rep de tous les composants est présenté à la figure 3.2.

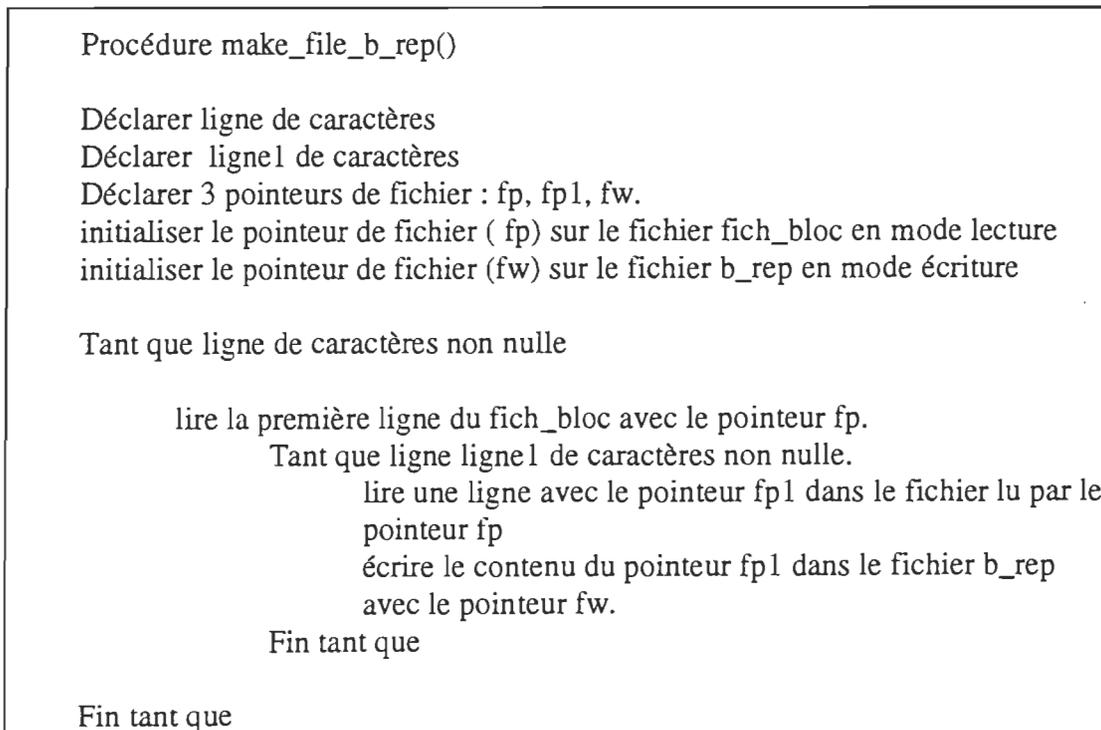


Figure 3.2: algorithme de reconstitution du fichier B\_Rep

Cet algorithme lit donc le contenu de chaque fichier B\_Rep à partir du fichier fich\_bloc des composants et écrit le contenu lu dans un fichier unique. Le résultat obtenu avec cet algorithme est un fichier unique contenant tous les fichiers B\_Rep des composants. Dans l'exemple des 4 blocs, le fichier final se nomme B\_Rep et contient tous les fichiers B\_Rep des composants.

Ce fichier final étant obtenu, l'étape suivante est la détection des contacts entre les composants.

### 3.2.2. Détection des contacts entre les composants

La méthode utilisée pour détecter les contacts entre les pièces, consiste à comparer chaque surface de chaque composant avec les autres surfaces des autres composants. L'existence d'au moins un contact entre les surfaces de 2 composants, entraîne un contact entre ces 2 composants en question.

#### 3.2.2.1. Formulation mathématique des conditions de contact entre 2 composants

La figure 3.3 présente une situation de contact entre 2 composants.

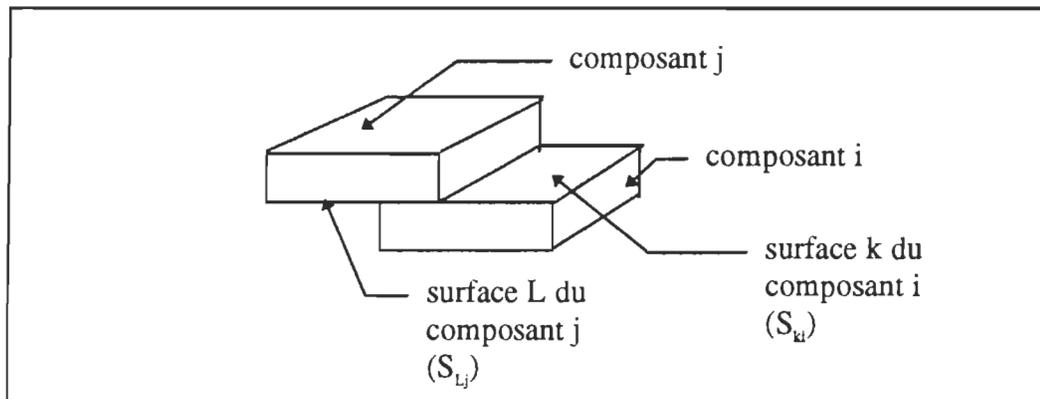


Figure 3.3: 2 composants en contact

$S_{Lj}$  et  $S_{ki}$  sur la figure 3.4 sont en contact si et seulement si:

- $n_i \cdot n_j < 0$   $n_i$  et  $n_j$  sont les vecteurs normaux des surfaces  $k$  et  $L$
- distance  $(S_{ki}, S_{Lj}) = 0$

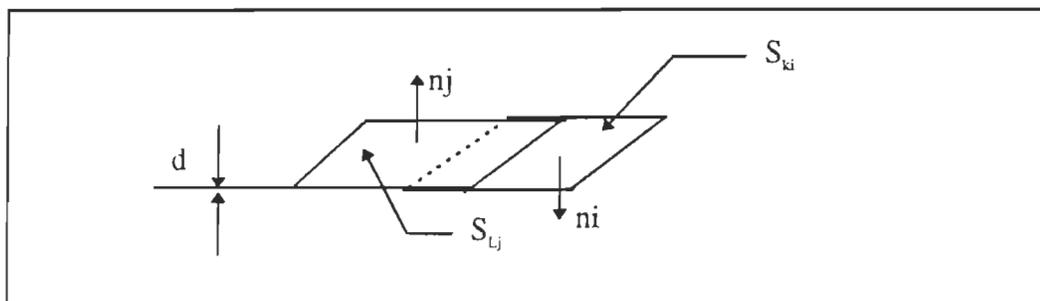


Figure 3.4: modélisation mathématique d'un contact

Pour les contacts usuels les conditions de contact se présentent de la façon suivante :

a) un contact plan entre 2 surfaces de 2 composants montré sur la figure 3.5.

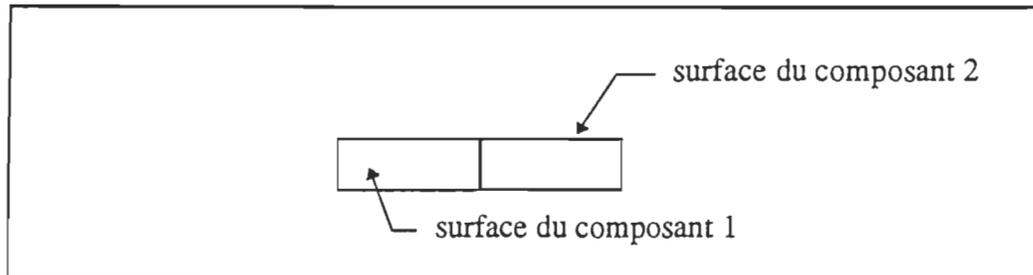


Figure 3.5: contact plan-plan

Ces 2 surfaces sont en contact si et seulement si le produit scalaire de leur vecteur normal est inférieur à zéro et que la distance entre le point appartenant à chaque surface est nulle :

$$n1 * n2 < 0 \quad (3.1)$$

$$d = 0 \quad (3.2)$$

La formulation mathématique à partir de la figure 3.6 suivant l'axe x se fait de la façon suivante.

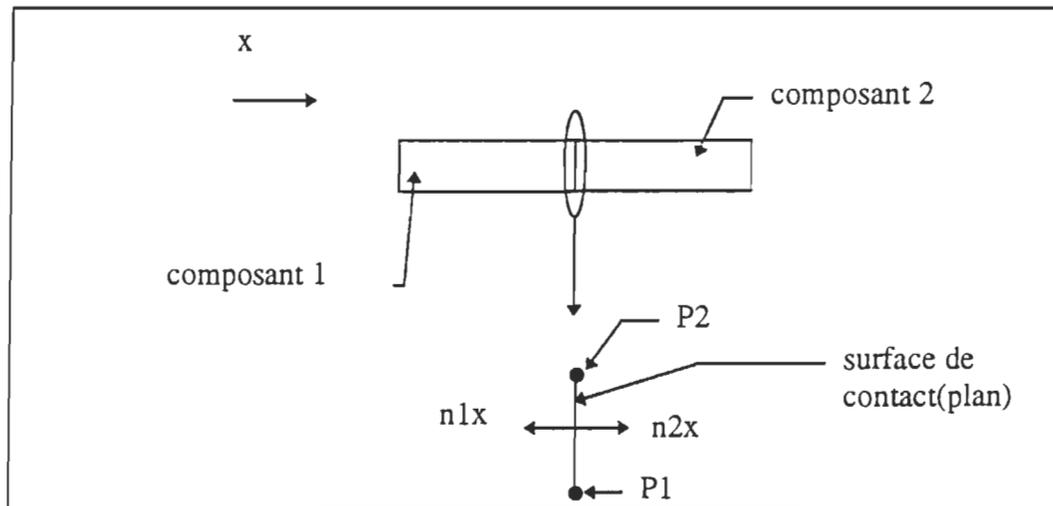


Figure 3.6: modélisation mathématique d'un contact plan-plan

$$\text{Le produit scalaire } n1x \times n2x < 0 \quad (3.3)$$

La distance  $d$  est la valeur absolue de la projection du point  $P2$  sur la surface 1 qui s'évalue par le rapport du produit scalaire de la distance entre les 2 points  $P1$ ,  $P2$ , la normale de la surface 1 et la norme de cette normale :

$$d = \frac{|P_1P_2 \cdot n_1|}{|n_1|} \quad (3.4)$$

avec  $P_1P_2 = (x_2 - x_1) \cdot i + (y_2 - y_1) \cdot j + (z_2 - z_1) \cdot k$   
 et  $n_1 = n_{1x} \cdot i + n_{1y} \cdot j + n_{1z} \cdot k$

$$d = \frac{(x_2 - x_1) \cdot n_{1x} \cdot i + (y_2 - y_1) \cdot n_{1y} \cdot j + (z_2 - z_1) \cdot n_{1z} \cdot k}{\sqrt{n_{1x}^2 + n_{1y}^2 + n_{1z}^2}} \quad (3.5)$$

On procède de la même manière pour les autres axes en remplaçant les indices par le nom de l'axe.

**b) un contact plan-cylindre entre 2 surfaces de 2 composants montré sur la figure 3.7**

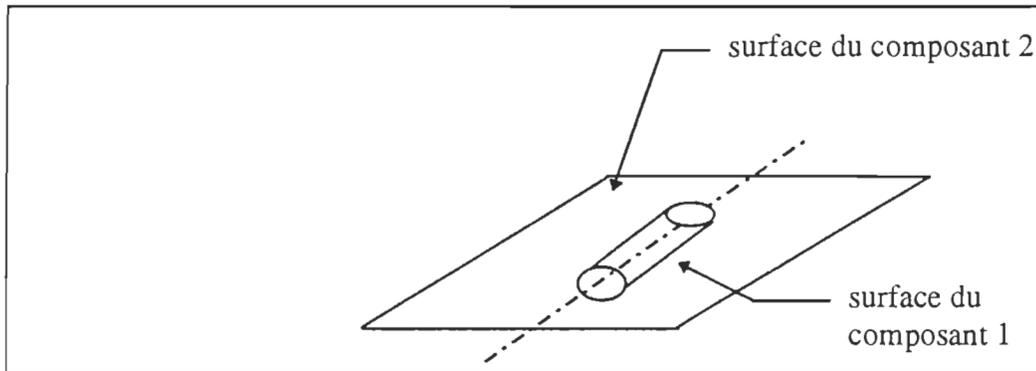


Figure 3.7: contact plan-cylindre

Ces 2 surfaces sont en contact si et seulement si le produit scalaire de leur vecteur normal est égal à zéro et que la distance entre le point appartenant à chaque surface est nulle :

$$n_1 \cdot n_2 = 0 \quad (3.6)$$

$$d = \text{rayon de la surface 1} \quad (3.7)$$

La formulation mathématique suivant l'axe x à partir de la figure 3.8 se présente de la façon suivante:

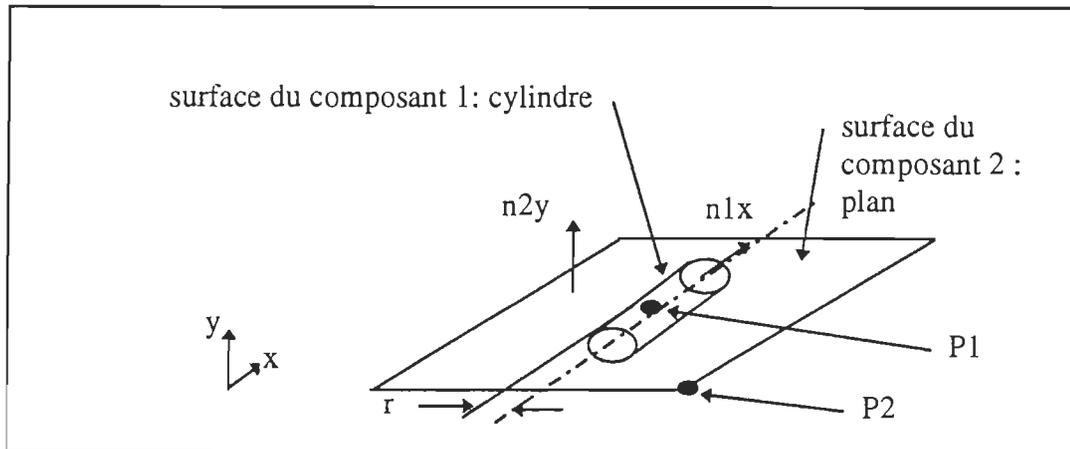


Figure 3.8: modélisation mathématique d'un contact plan-cylindre

$$\text{Le produit scalaire } n1x * n2y = 0 \quad (3.8)$$

La distance  $d$  est la valeur absolue de la projection du point  $P2$  sur la surface 1 qui s'évalue par le rapport du produit scalaire de la distance entre les 2 points  $P1, P2$ , la normale et la norme de cette normale :

$$d = \frac{|P1P2 \cdot n1|}{|n1|} \quad (3.9)$$

avec  $P1P2 = (x2 - x1) * i + (y2 - y1) * j + (z2 - z1) * k$   
 et  $n1x * i + n1y * j + n1z * k$

$$d = \frac{(x2 - x1) * n1x * i + (y2 - y1) * n1y * j + (z2 - z1) * n1z * k}{\sqrt{n1x^2 + n1y^2 + n1z^2}} \quad (3.10)$$

On procède de la même manière pour les autres axes en remplaçant les indices par le nom de l'axe.

c) un contact plan-ellipse entre 2 surfaces de 2 composants montré sur la figure 3.9.

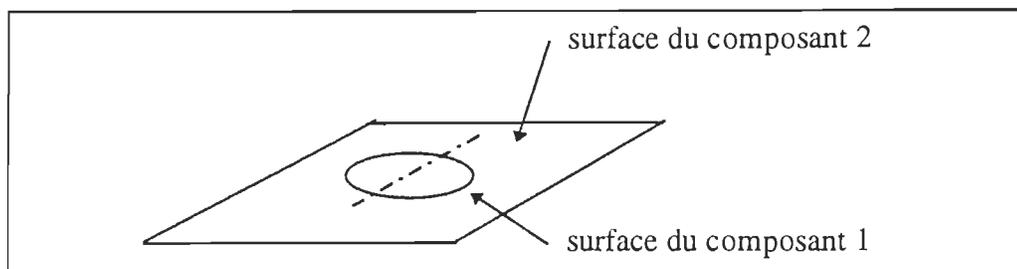


Figure 3.9: contact plan-ellipse

Ces 2 surfaces sont en contact si et seulement si le produit scalaire de leur vecteur normal est égal à zéro et que la distance entre le point appartenant à chaque surface est nulle :

$$n1 \cdot n2 < 0 \quad (3.11)$$

$$d = 0 \quad (3.12)$$

La formulation mathématique suivant l'axe y montré sur la figure 3.10 se présente de la façon suivante:

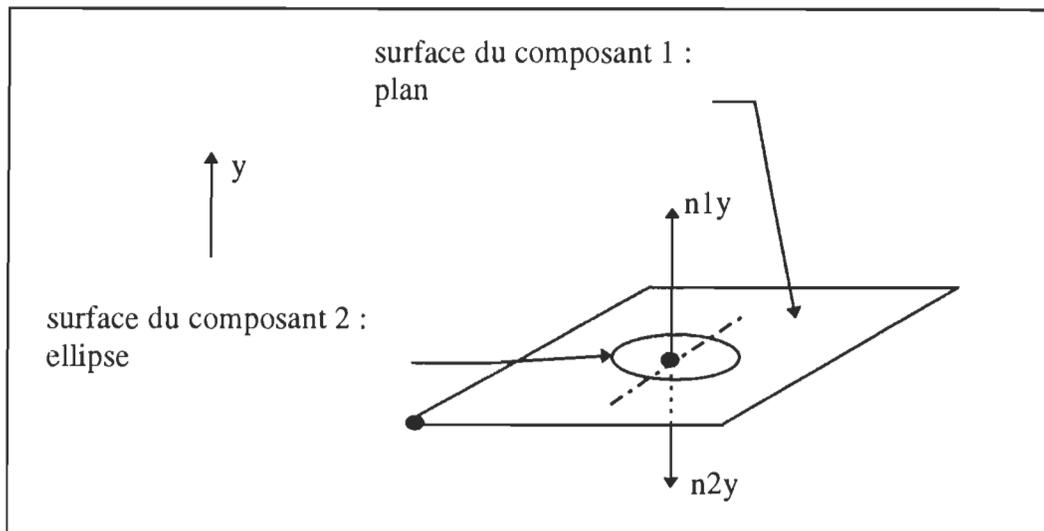


Figure 3.10: modélisation mathématique d'un contact plan-ellipse

$$\text{Le produit scalaire } n1 \cdot n2 = 0 \quad (3.13)$$

La distance  $d$  est la valeur absolue de la projection du point  $P2$  sur la surface 1 qui s'évalue par le rapport du produit scalaire de la distance entre les 2 points  $P1, P2$ , la normale et la norme de cette normale :

$$d = \frac{|P1P2 \cdot n1|}{|n1|} \quad (3.14)$$

$$\text{avec } P1P2 = (x2 - x1) \cdot i + (y2 - y1) \cdot j + (z2 - z1) \cdot k$$

$$\text{et } n1 = n1x \cdot i + n1y \cdot j + n1z \cdot k$$

$$d = \frac{(x2 - x1) \cdot n1x \cdot i + (y2 - y1) \cdot n1y \cdot j + (z2 - z1) \cdot n1z \cdot k}{\sqrt{n1x^2 + n1y^2 + n1z^2}} \quad (3.15)$$

On procède de la même manière pour les autres axes en remplaçant les indices par le nom de l'axe.

d) un contact ellipse-ellipse comme l'indique la figure 3.11.

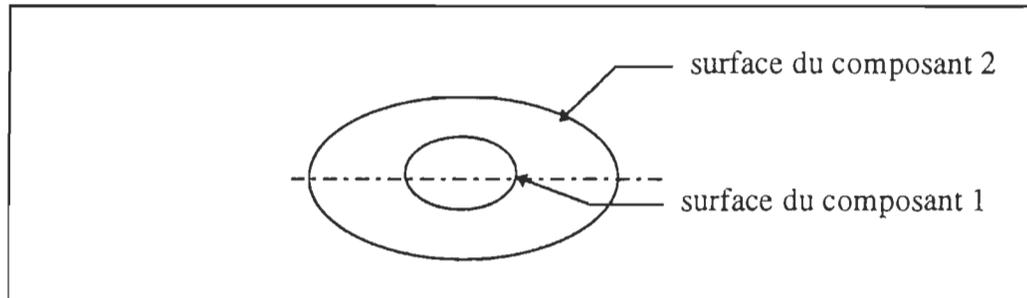


Figure 3.11: contact ellipse-ellipse

le même principe sera appliqué

e) un contact cylindre-cylindre de 2 surfaces de 2 composants montré sur la figure 3.12.

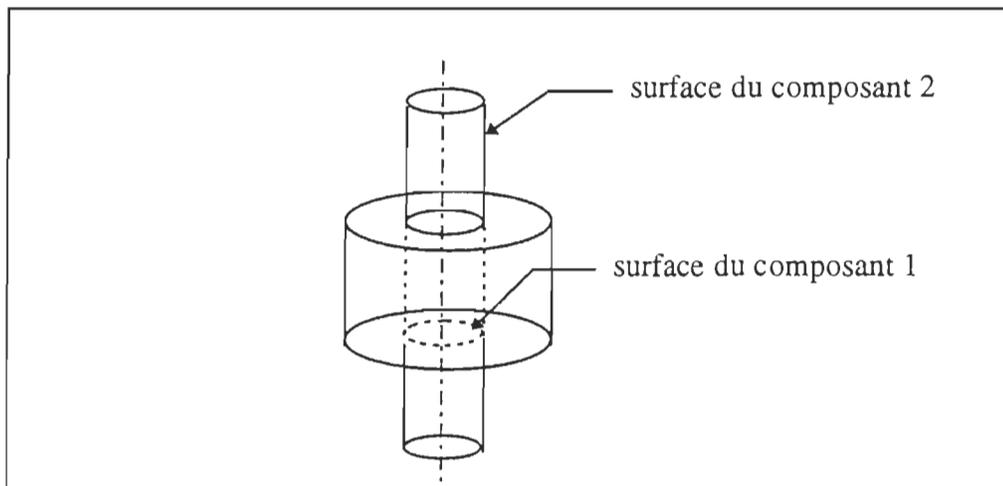


Figure 3.12: contact cylindre-cylindre

Ces 2 surfaces sont en contact si et seulement si ces 2 conditions sont remplies selon la figure 3.13.

- l'axe des 2 composants se confond
- la distance centre à centre des composants respecte une certaine distance

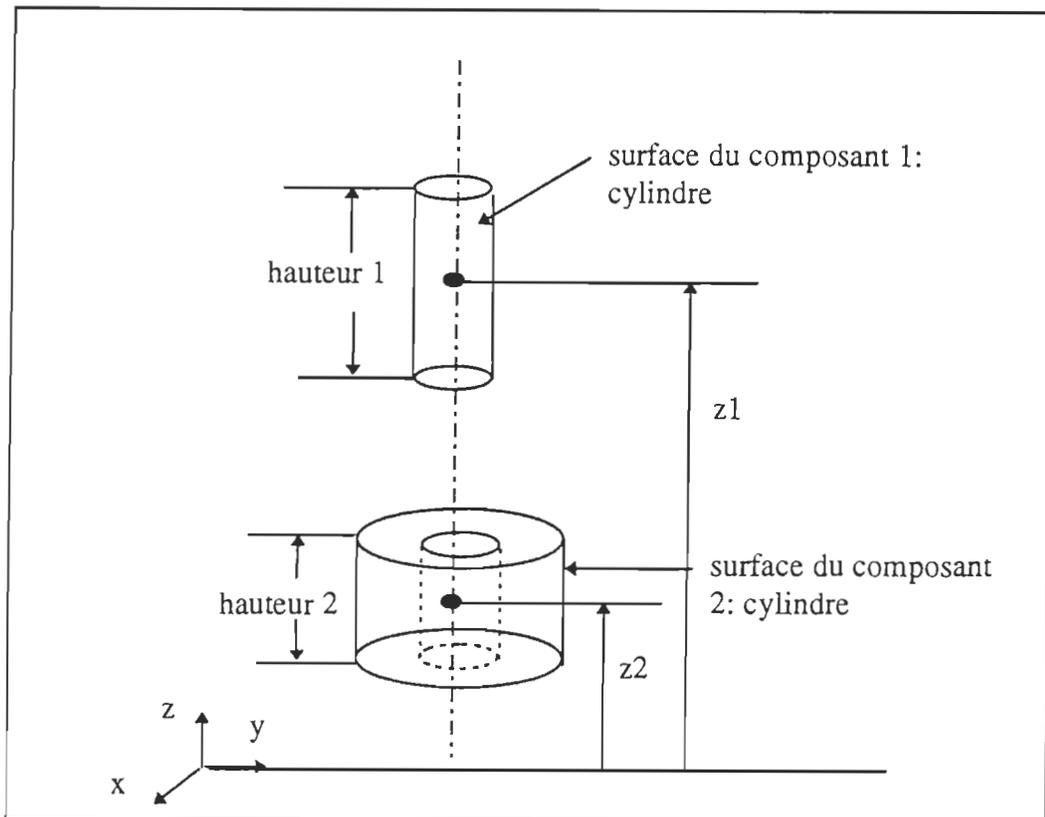


Figure 3.13: modélisation mathématique d'un contact cylindre-cylindre

La formulation mathématique de la condition selon laquelle les 2 axes des 2 composants doivent être confondus.

- Si les 2 hauteurs se trouvent suivant l'axe x alors  
 $y_1 = y_2$  et  $z_1 = z_2$
- Si les 2 hauteurs se trouvent suivant l'axe y alors  
 $x_1 = x_2$  et  $z_1 = z_2$
- Si les 2 hauteurs se trouvent suivant l'axe z alors  
 $y_1 = y_2$  et  $x_1 = x_2$

La formulation mathématique de la deuxième condition à partir de la figure 3.12:

- Si les 2 hauteurs se trouvent suivant l'axe x alors  
 si  $x_1 < x_2 \Rightarrow x_1 + \text{hauteur}_1/2 \geq x_2 - \text{hauteur}_2/2$   
 si  $x_1 > x_2 \Rightarrow x_1 - \text{hauteur}_1/2 \leq x_2 + \text{hauteur}_2/2$
- Si les 2 hauteurs se trouvent suivant l'axe y alors

si  $y_1 < y_2 \Rightarrow y_1 + \text{hauteur}1/2 \geq y_2 - \text{hauteur}2/2$   
 si  $y_1 > y_2 \Rightarrow y_1 - \text{hauteur}1/2 \leq y_2 + \text{hauteur}2/2$

- Si les 2 hauteurs se trouvent suivant l'axe z alors  
 si  $z_1 < z_2 \Rightarrow z_1 + \text{hauteur}1/2 \geq z_2 - \text{hauteur}2/2$   
 si  $z_1 > z_2 \Rightarrow z_1 - \text{hauteur}1/2 \leq z_2 + \text{hauteur}2/2$ .

**f) contact sphère-sphère de 2 surfaces sur la figure 3.14.**

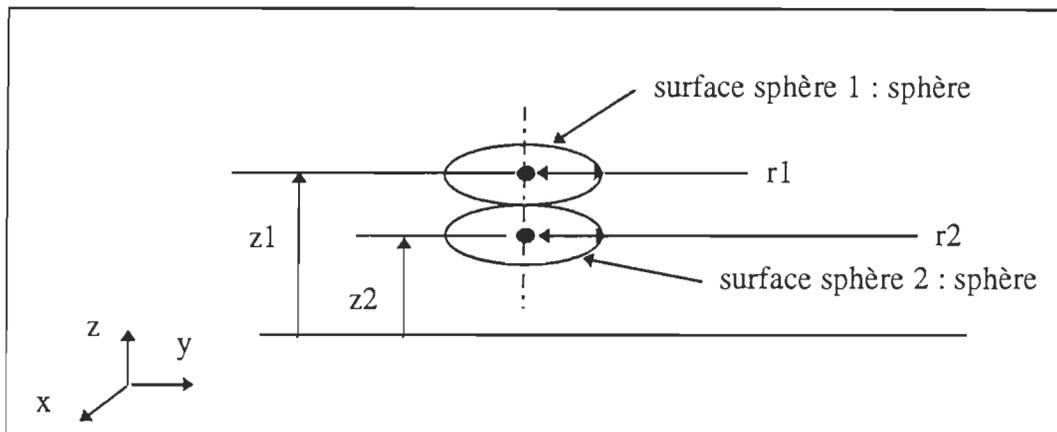


Figure 3.14: contact sphère-sphère

Ces 2 surfaces sont en contact si et seulement si:  
 $x_1 - r_1 = x_2 + r_2$  ou  $y_1 - r_1 = y_2 + r_2$  ou  $z_1 - r_1 = z_2 + r_2$

**g) contact sphère-cylindre sur la figure 3.15.**

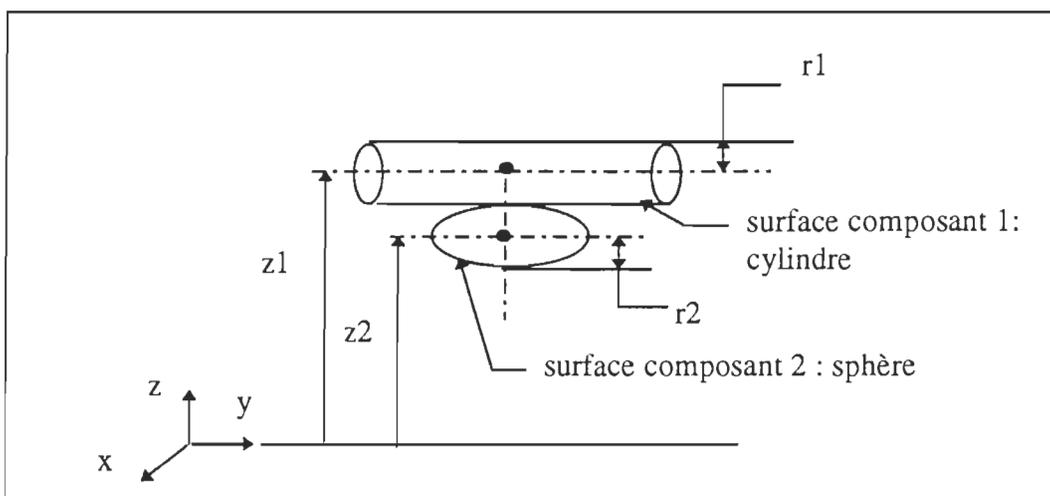


Figure 3.15 :contact sphère-cylindre

Ces 2 surfaces sont en contact si et seulement si:

$$x_1 - r_1 = x_2 + r_2 \text{ ou } y_1 - r_1 = y_2 + r_2 \text{ ou } z_1 - r_1 = z_2 + r_2.$$

**h) un contact sphère-ellipse montré sur la figure 3.16.**

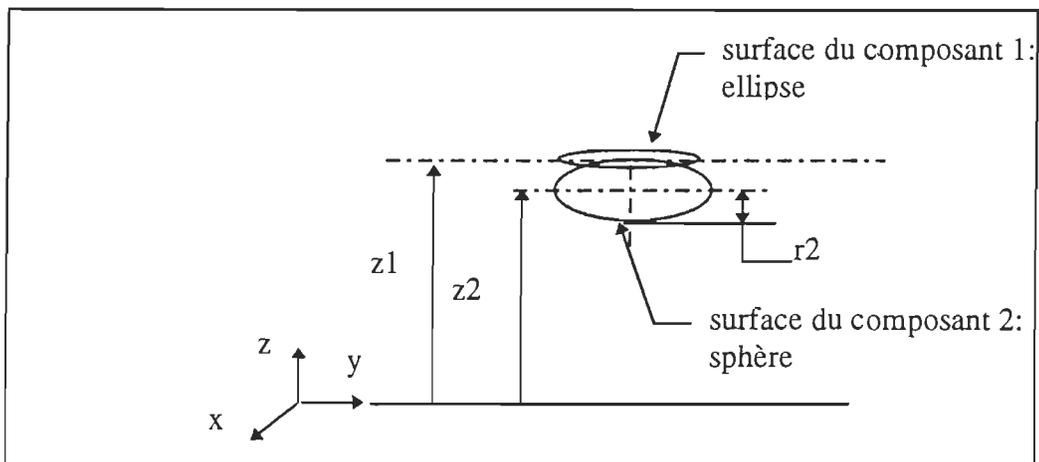


Figure 3.16. :contact sphère-ellipse

Ces 2 surfaces sont en contact si et seulement si:  
 $x_2 + r_2 = x_1$  ou  $y_2 + r_2 = y_1$  ou  $z_2 + r_2 = z_1$ .

**i) un contact sphère-plan montré sur la figure 3.17.**

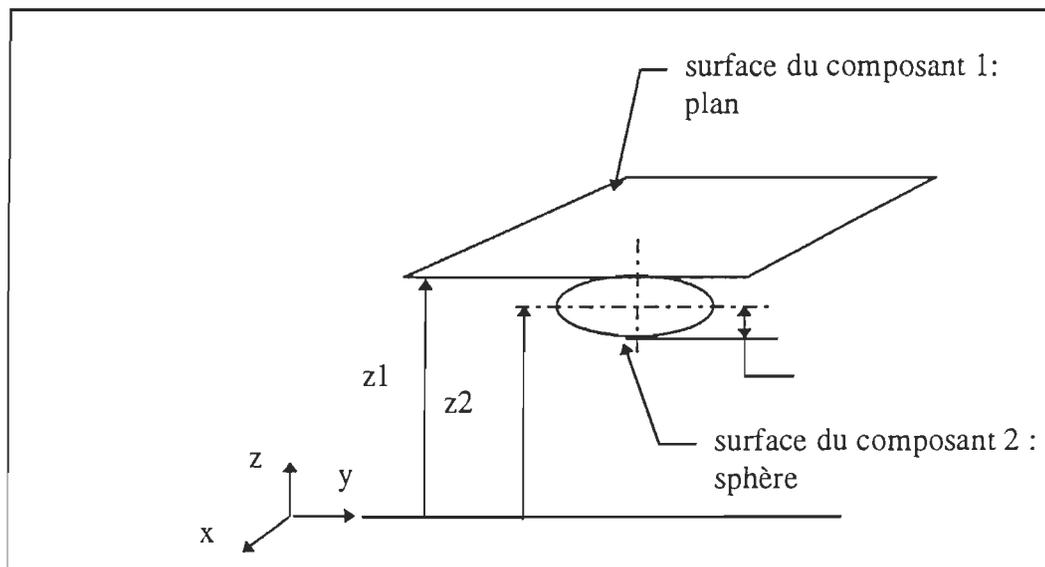


Figure 3.17:contact sphère-plan

Ces 2 surfaces sont en contact si et seulement si:  
 $x_2 + r_2 = x_1$  ou  $y_2 + r_2 = y_1$  ou  $z_2 + r_2 = z_1$

Les 2 conditions mentionnées auparavant ne sont pas suffisantes pour détecter les contacts entre les composants. Une troisième condition doit être utilisée pour rendre robuste cette détection des contacts. Cette dernière condition consiste à vérifier si les 2 surfaces des 2 composants, dont le produit scalaire de leur vecteur normal est négatif et la distance entre ces 2 surfaces est nulle, se superposent. Cette condition de superposition est réalisée si au moins une partie des 2 surfaces est confondue. L'exemple de la figure 3.18 montre un exemple de superposition pour un contact plan-plan.

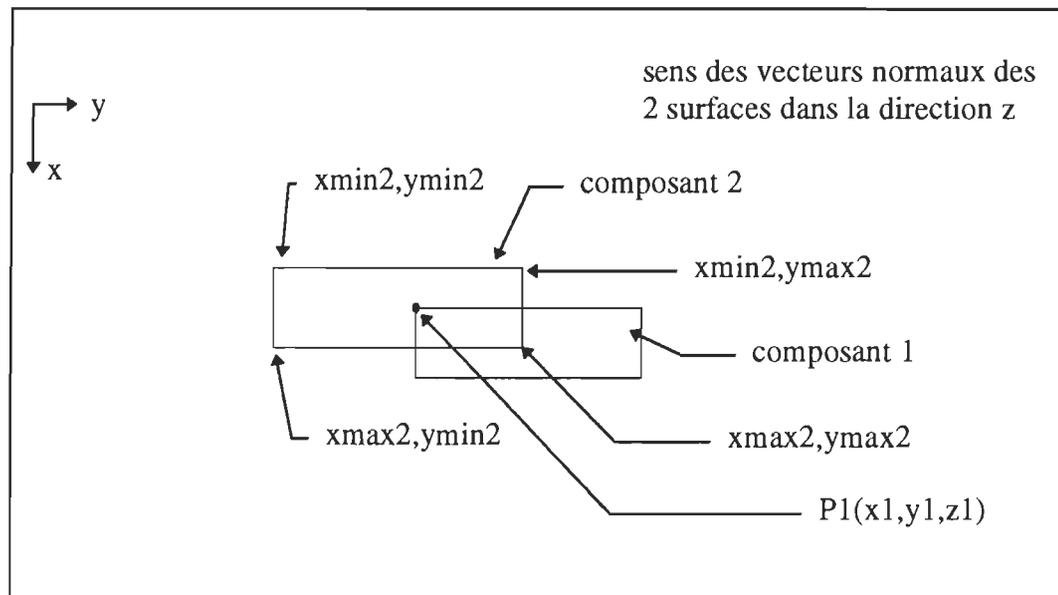


Figure 3.18: superposition de 2 surfaces

Les 2 surfaces de la figure 3.18 se superposent si et seulement si :

$$x_{\min 2} \leq P1x \leq x_{\max 2} \text{ et } y_{\min 2} \leq P1y \leq y_{\max 2} \dots\dots\dots (3.16)$$

Pour les autres types de contacts, la même méthode de superposition sera appliquée.

### **3.2.2.2. Mise en oeuvre algorithmique des conditions de contact**

La lecture séquentielle du fichier B-Rep final par un algorithme permet de détecter ces contacts. Cet algorithme est présenté à la figure 3.19.

```
Procédure comparaison_surface()  
  
pointeur pt1, pt2  
initialiser pt1 en le faisant pointer sur le premier composant de la liste des  
composants  
initialiser pt2 en le faisant pointer sur le deuxième composant de la liste des  
composants.  
Tant que le pointeur pt1 est non nul  
    Tant que le pointeur pt2 est non nul  
        prendre le composant sur lequel pointe pt2  
        Pour i =1 allant à nombre de surfaces du composant  
            appeler la procédure contact(composant 1, composant 2)  
        Fin pour  
    pointer pt2 sur le prochain composant  
    Fin tant que  
pointer pt1 sur le prochain composant  
Fin tant que
```

Figure 3.19: algorithme de comparaison des surfaces

Cette procédure permet de consulter tous les composants de la liste des composants. Lorsqu'un composant est retrouvé, la procédure compare une à une les surfaces du composant trouvé avec les autres surfaces des autres composants. Cette comparaison est faite à l'aide de la procédure présentée à la figure 3.20.

```

Procédure contact(composant1 (c1), composant2 (c2), surface du composant 1
(s1), surface du composant 2 (s2))
calcul de la distance entre les 2 surfaces s1 et s2
si les 2 surfaces sont des surfaces plan ou si l'une des surfaces est une surface
ellipse
et l'autre une surface plan
    appeler la procédure plan-plan(c1,c2,s1,s2)
sinon si les 2 surfaces sont des surfaces d'une ellipse
    appeler la procédure ellipse-ellipse(c1,c2,s1,s2)
sinon si les 2 surfaces sont des surfaces d'un cylindre
    appeler la procédure cylindre-cylindre(c1,c2,s1,s2)
sinon si l'une des 2 surfaces est une surface plan et l'autre une surface
cylindre appeler la procédure plan-cylindre(c1,c2,s1,s2)
sinon si l'une des 2 surfaces est une surface sphère et l'autre une surface
ellipse appeler la procédure sph-ellipse(c1,c2,s1,s2)
sinon si l'une des 2 surfaces est une surface sphère et l'autre une surface
cylindre
    appeler la procédure sph-cylindre(c1,c2,s1,s2)
sinon si l'une des 2 surfaces est une surface sphère et l'autre une surface plan
    appeler la procédure plan-sphère(c1,c2,s1,s2)
sinon si les 2 surfaces sont des surfaces d'une sphère
    appeler la procédure sp-sp(c1,c2,s1,s2)

```

Figure 3.20: procédure de comparaison de surfaces

Cette procédure appelle donc en fonction des types surfaces la procédure correspondante pour vérifier s'il existe un contact entre ces 2 surfaces.

Pour illustrer la vérification de l'existence d'un contact entre 2 surfaces, examinons la procédure plan-plan(c1,c2,s1,s2) de la figure 3.21.

```

Procédure plan-plan(c1,c2,s1,s2)

```

```

    vérifier si les 2 surfaces n'appartiennent pas à un trou
    si le produit scalaire des vecteurs normaux des 2 surfaces est inférieur
    à zéro et que la distance entre ces 2 surfaces est nulle
        appeler la procédure contact(c1,c2,s1,s2)

```

Figure 3.21: algorithme de détection d'un contact plan-plan

Lorsqu'un contact est effectif entre 2 composants, la procédure contact permet de stocker, dans la matrice des contacts de l'objet classe des composants, les numéros des composants. Par exemple si le composant 1 et le composant 2 ont une surface en contact, le composant 1 aura comme valeur de la deuxième ligne de sa matrice des

contacts 2 tandis que le composant 2 aura comme valeur de la première ligne de sa matrice des contacts 1.

### **3.2.2.3. Limitations de la méthode de détection des contacts**

Les informations dont nous disposons sur la signification des champs d'une surface ne nous permettent pas d'interpréter la modélisation d'un composant par l'option "torus" tore , surfaces B-Splines, wedge (morceau d'un parallélépipède). Donc on ne peut avoir le fichier B\_Rep des composants ayant ces formes.

Les composants du produit doivent être assimilés à des formes usuelles (primitives) autorisées par le modeleur ICEM/DDN. L'angle entre les surfaces des composants doit être de 90 degrés.

### **3.2.3. Génération des relations**

#### **3.2.3.1. Définitions**

Deux composants ont une relation de contact s'il existe un contact physique entre ces 2 composants lorsque le produit dont ils font partie est assemblé. La relation de contact est une relation de symétrie. La figure 3.22 montre un exemple de composants en contact.

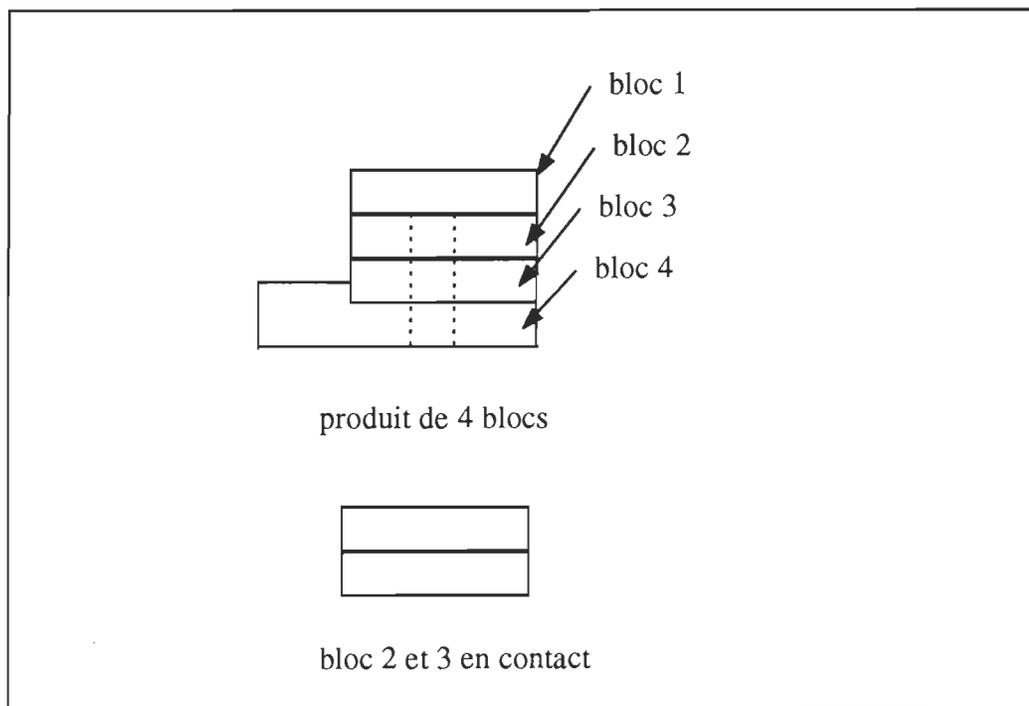


Figure 3.22: exemple de composants en contact

Deux composants ont une relation de non-contact appelé blocage s'il n'existe pas un contact physique entre ces 2 composants mais la réduction de la distance entre

ces composants provoquent une collision entre eux. La relation de blocage est aussi une relation symétrique. La figure 3.23 montre un exemple de relation de type blocage.

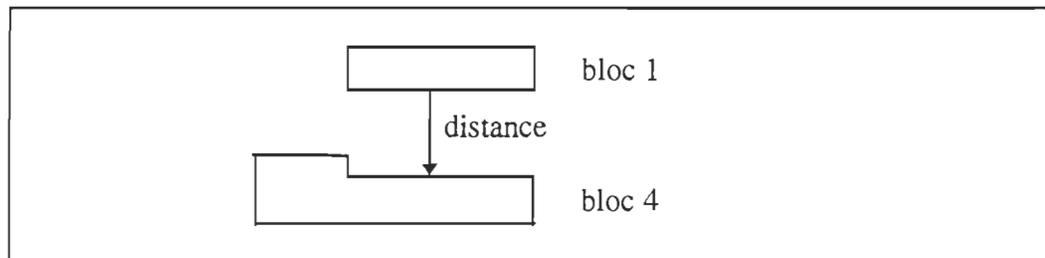


Figure 3.23: bloc 1 et 4 ayant une relation de blocage

Deux composants ont une relation libre s'il n'existe ni une relation de contact ni une relation de blocage entre eux. La figure 3.24 montre un exemple de relation libre.

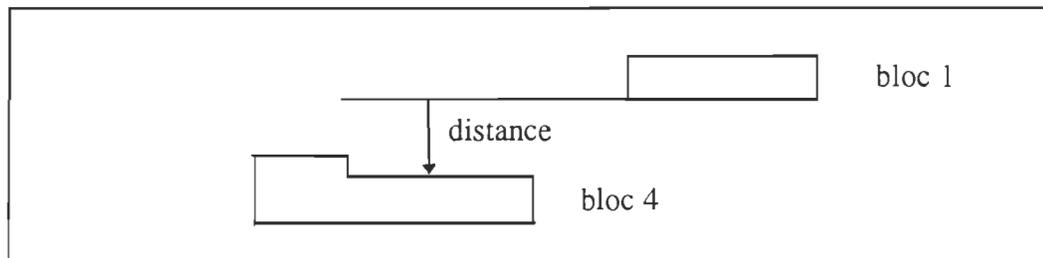


Figure 3.24: bloc 1 et 4 ayant une relation libre

Deux composants ont une relation entre eux s'ils ont une relation de contact de blocage ou libre. L'ensemble des relations est stocké dans une liste. À partir de ces définitions, le graphe des liaisons est modélisé. L'exemple de la figure 3.25 montre le graphe des liaisons de 4 blocs.

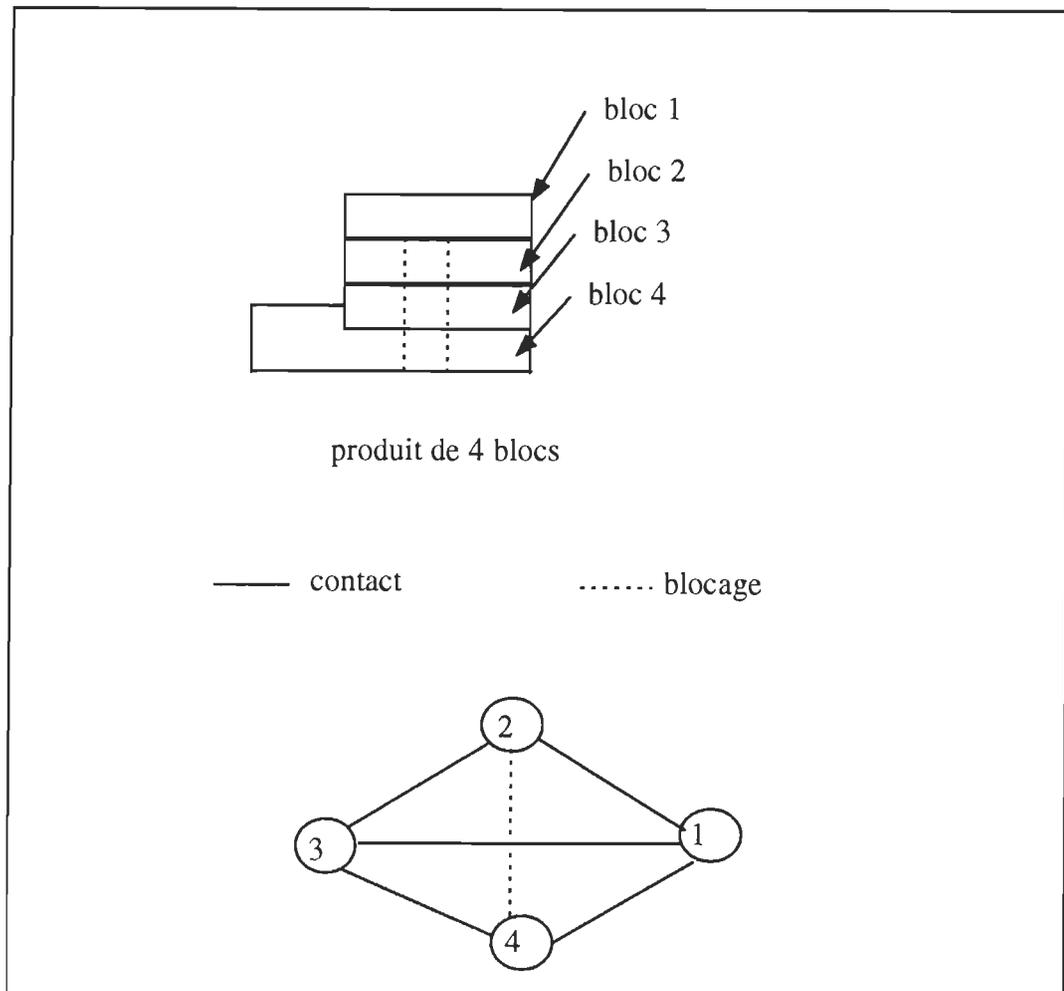


Figure 3.25: graphe des liaisons

Une relation est décrite [59] suivant le format suivant:

Relation  $i$  : variable (prenant la valeur "oui" ou la valeur "non"), nom du composant mobile, le nom du composant fixe, la direction de désassemblage du composant mobile, la direction de désassemblage du composant fixe, le type de relation, le type d'opération nécessaire à l'établissement de la relation. La variable but indique si les composants d'une relation sont complètement séparés après l'opération de désassemblage ou pas. Une variable contenant une valeur "non" indique qu'il y a séparation complète des composants tandis qu'une valeur "oui" indique qu'il n'y a pas de séparation complète. Par exemple la relation entre les composants 3 et 4 sera décrite de la façon suivante:

Relation (3-4) = non, bloc 3, bloc 4, z- x+ x- y+ y-, z+ x+ x- y+ y-, contact,plan.

Figure 3.26: description de la relation entre le bloc 3 et 4 selon le format proposé dans [59]

### 3.2.3.2. Mise en oeuvre informatique de la génération des relations

Après vérification de l'existence d'un contact entre au moins une surface de 2 composants, si la réponse est positive, un objet relation (r) de la classe relation est créé [59] suivant ce format :

```
allocation d'une zone mémoire à l'objet relation
création de l'objet relation r = (variable , nom du composant 1, nom du
composant 2, direction d'assemblage du composant 1, direction d'assemblage
du composant 2, type de relation, type d'opération)
ajouter cette relation créée à la liste des relations
```

A la fin du processus de génération, la liste des relations contiendra toutes les relations de contact et de blocage existant entre les composants d'un produit.

## 3.3. Matrice des demi-degrés de liberté

### 3.3.1. Représentation des états des contacts

La spécification de l'état des contacts des composants avec leur demi-degrés de liberté après une opération d'assemblage peut être représentée par une matrice [73]. Cette matrice des demi-degrés de liberté appelé pseudo-matrice contient 12 éléments: 3 lignes et 4 colonnes. Les lignes correspondent aux axes d'un repère orthonormé. Les colonnes représentent les translations et les rotations comme l'indique la figure 3.27. Dans cette figure, par exemple tx représente le mouvement de translation suivant l'axe x et rx le mouvement de rotation suivant l'axe x.

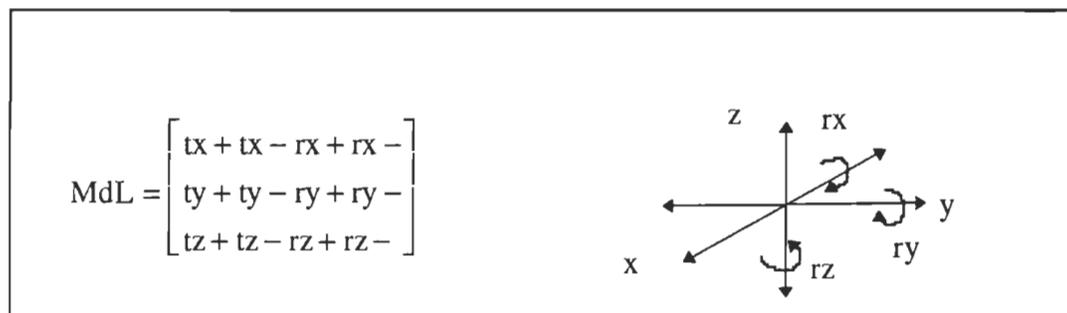


Figure 3.27: représentation des contacts par la matrice des demi-degrés de liberté

Les éléments de cette matrice sont évalués par des valeurs 1 ou 0. Un élément ayant une valeur 1 signifie que l'élément décrit un mouvement libre tandis qu'une valeur 0 signifie que l'élément décrit un mouvement bloqué. L'exemple suivant de la figure 3.28 présente la matrice des demi-degrés de liberté de 2 composants (une tige et un bloc) en contact.

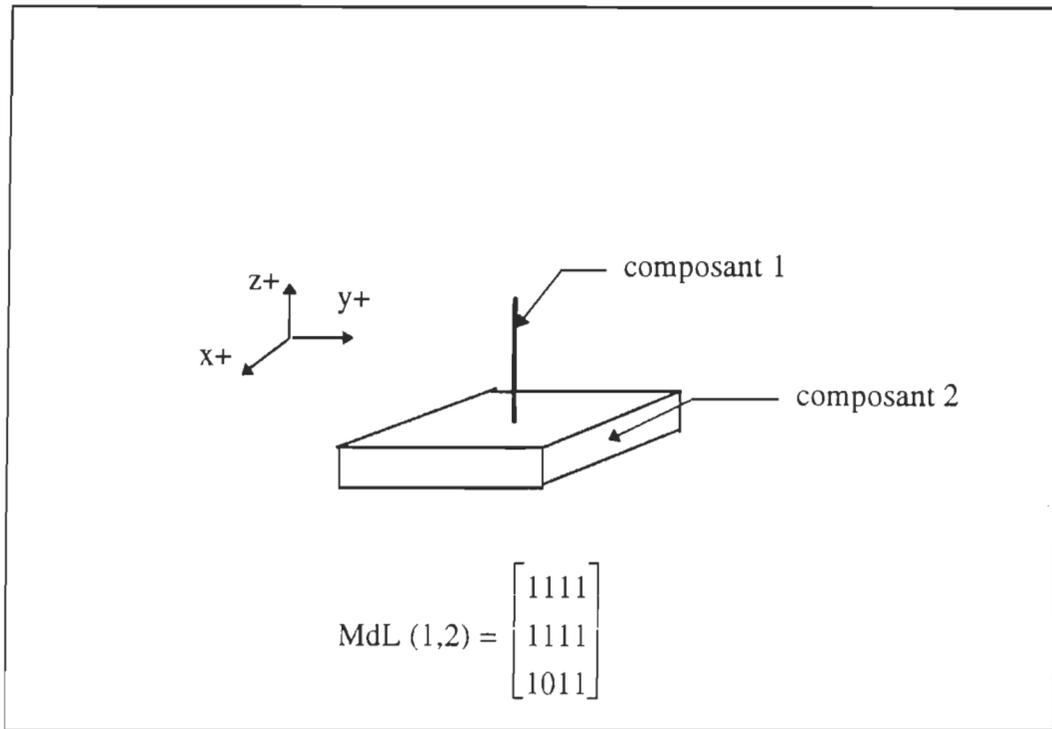


Figure 3.28: matrice des demi-degrés de liberté du contact du point de vue du composant 1 du contact entre les composants 1 et 2

### 3.3 .2. Mise en oeuvre informatique de la génération des matrices des demi-degrés de liberté

Si un contact est trouvé par le programme suivant un axe quelconque, la matrice des demi-degrés de liberté (MdL) est automatiquement compilée de la façon suivante:

Les valeurs de la matrice des demi-degrés de liberté, qui sont initialement des 1, sont remplacées par des zéros selon que le contact entraîne un blocage de la rotation et/ou de la translation suivant l'axe correspondant.

La comparaison de toutes les surfaces contenues dans l'enregistrement du composant i avec celles du composant j conduit à modifier la matrice des degrés de liberté s'il existe des contacts entre ces 2 composants. Cette matrice reste intacte s'il s'agit d'une relation libre.

Pour toutes les surfaces du composant i la matrice des demi-degrés de liberté s'écrit de la façon suivante:

$$\text{MdL (i,j)} = \begin{bmatrix} \text{cpc}_x & \text{cpe}_x & \text{cpp}_x & \text{cps}_x & \text{cee}_x & \text{ces}_x & \text{css}_x & \text{ccc}_x \\ \text{cpc}_y & \text{cpe}_y & \text{cpp}_y & \text{cps}_y & \text{cee}_y & \text{ces}_y & \text{css}_y & \text{ccc}_y \end{bmatrix}$$

$$[cpc_z \ cpe_z \ cpp_z \ cps_z \ cee_z \ ces_z \ css_z \ ccc_z]$$

où

$\cap^c$  est l'opérateur d'intersection

cpc est un contact plan-cylindre

cpe est un contact plan ellipse

cpp est un contact plan-plan

cps est un contact plan-sphère

cee est un contact ellipse-ellipse

ces est un contact ellipse-sphère

css est un contact sphère-sphère

ccc est un contact cylindre-cylindre

Les indices x, y et z représentent les axes d'un système de coordonnées.

Les règles d'intersection des contacts de la matrice des demi-degrés de liberté d'un composant dans le cas où un composant a plusieurs contacts avec un autre composant sont:

$$1 \cap^c 1 = 1$$

$$0 \cap^c 1 = 0$$

$$1 \cap^c 0 = 0$$

$$0 \cap^c 0 = 0$$

La matrice des demi-degrés de liberté  $MdL(j,i)$  est l'opposée en terme des mouvements de translation de celle de  $MdL(i,j)$ .

La figure 3.29 montre un exemple de compilation des valeurs de la matrice des demi-degrés de liberté en utilisant les règles d'intersection :

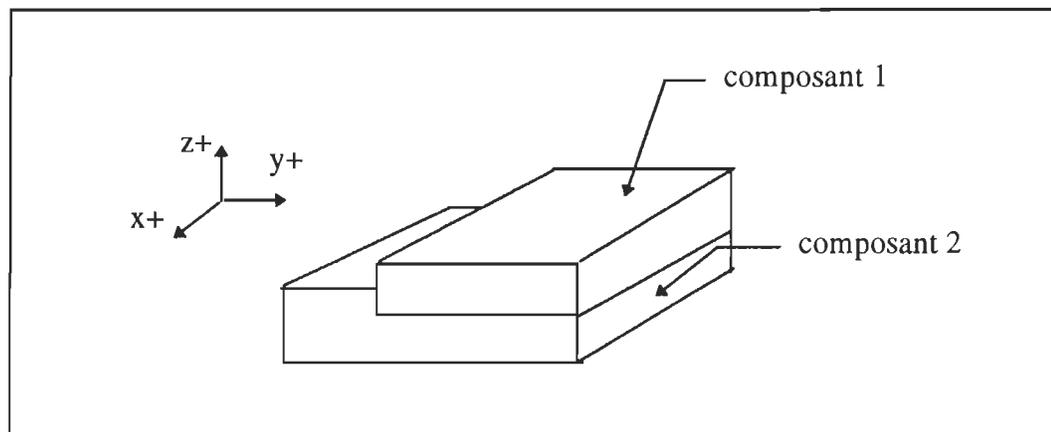


Figure 3.29: produit de 2 composants

La matrice  $MdL$  du contact suivant l'axe z des composants 1-2 vue du côté du composant 1:

$$MdL(1,2) = \begin{bmatrix} 1100 \\ 1100 \\ 1011 \end{bmatrix}$$

La matrice MdL du contact suivant l'axe y des composants 1-2 :

$$\text{MdL}(1,2) = \begin{bmatrix} 1100 \\ 1011 \\ 1100 \end{bmatrix}$$

L'application de la règle de modification donne la matrice MdL définitive des contacts entre ces 2 composants du côté du composant 1 :

$$\text{MdL}(1,2) = \begin{bmatrix} 1100 \\ 1100 \\ 1011 \end{bmatrix} \cap \begin{bmatrix} 1100 \\ 1011 \\ 1100 \end{bmatrix} = \begin{bmatrix} 1100 \\ 1000 \\ 1000 \end{bmatrix}$$

La matrice MdL vue du côté du composant 2 est l'inverse du point de vue des translations de la matrice MdL vue du côté du composant 1 :

$$\text{MdL}(2,1) = \begin{bmatrix} 1100 \\ 0100 \\ 0100 \end{bmatrix}$$

L'algorithme permettant de modifier la matrice des demi-degrés de liberté est présenté à la figure 3.30.

Procédure `modif_matrice_dl(composant i , composant j)`  
 mettre à zéro la position de la matrice des demi-degrés des composants i et j correspondant au mouvement de translation et à l'axe suivant lequel le mouvement s'effectue.  
 mettre à zéro les positions de la matrice des demi-degrés de liberté des composants i et j correspondant au mouvement de rotation de la matrice des demi-degrés de liberté et à l'axe suivant lequel le mouvement s'effectue.

Figure 3.30: procédure de modification des éléments de la matrice des demi-degrés de liberté de 2 composants en contact

### **3.4. Affichage du graphe des liaisons sur un écran d'ordinateur**

La philosophie de l'élaboration de la génération du graphe des liaisons d'un produit est basée sur la programmation objet dont le langage c++ s'y prête bien. Donc à ce stage du processus de génération, un objet de la classe composant est décrit suivant le modèle [59] proposé à la figure 3.31.

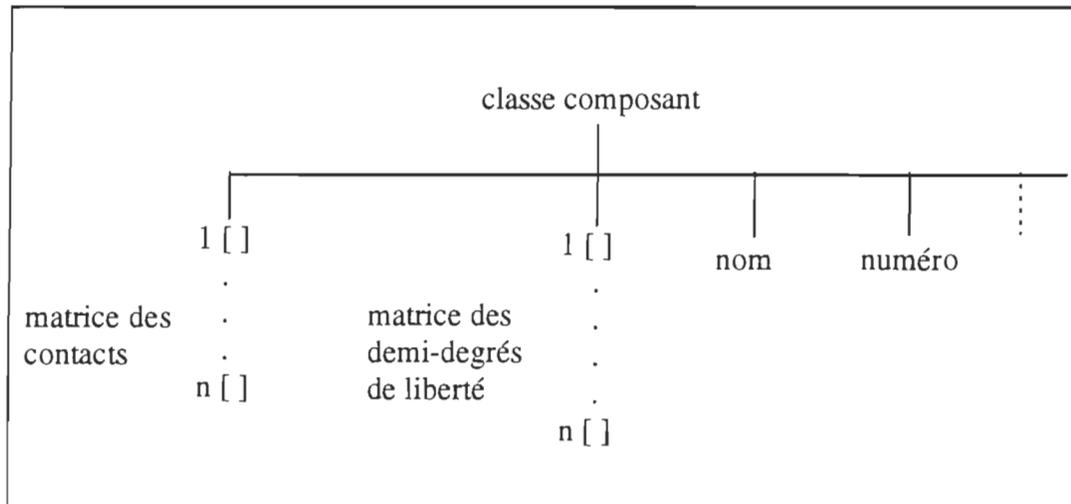


Figure 3.31: classe composant

La figure 3.31 indique que toutes les informations reliées à un composant sont regroupées en une structure appelée classe. La matrice des contacts mentionnée dans cette figure contient le numéro des composants qui sont en contact avec le composant de cette classe. Par exemple les contacts entre le composant 1 et les autres composants de la figure 3.28 seront traduits par l'enregistrement des numéros 2, 3, et 4 dans la matrice des contacts du composant 1. Le numéro 1 se retrouvera aussi dans la matrice des contacts du composant 2, 3, et 4. Disposant donc de toutes les informations sur les composants pour afficher le graphe, l'utilisation de l'environnement "Graphics Library" GL de IRIS 4.5.H de Silicon Graphics permet de tracer les points et les lignes pour afficher le graphe.

Avec l'aide des matrices des contacts, du nom des composants, le graphe est construit suivant ces étapes:

- calcul des coordonnées de l'emplacement des entités graphiques (cercles, nom du composant et lignes). Ce calcul est réalisé au moyen des coordonnées polaires suivant un angle de 360 degrés. La formule utilisée est :

$$x = \text{distance entre les composants et le centre du cercle (rayon)} \times \cos(\text{numéro composant} \times (2 \times \text{PI}/\text{nombre de composants})) \quad (3.16)$$

$$y = \text{distance entre les composants et le centre du cercle (rayon)} \times \sin(\text{numéro composant} \times (2 \times \text{PI}/\text{nombre de composants})) \quad (3.17)$$

Les coordonnées de l'emplacement de chaque composant sont donc calculées par les formules ci-dessus et stockées dans une liste. L'algorithme permettant de faire les calculs des coordonnées est présenté à la figure 3.32.

```

Procédure calcul_coordonnées()

Tant que la liste non vide
    prendre un composant
    créer l'objet coordonnées en utilisant les formules ci-dessus
    ajouter l'objet créer à la liste
Fin tant que

```

Figure 3.32: procédure calculant les coordonnées des composants

- créer une fenêtre avec un nom de fenêtre.

- tracer le graphe en construisant un cercle et en y mettant le nom du composant à l'emplacement calculé pour chaque composant plus haut. Puis relier ces cercles par des lignes. L'algorithme permettant de construire le graphe est présenté à la figure 3.33.

```

Procédure trace_graphe()

Tant que la liste des composants non vide
    prendre un composant
    pour i allant de 1 à nombre maximal de contacts
        construire un premier cercle avec l'emplacement
        correspondant à la valeur du numéro du composant pris
        écrire le nom du composant dans le cercle construit
        construire une ligne joignant l'emplacement correspondant au
        numéro du composant lu et de la valeur de i
        construire un deuxième cercle avec l'emplacement
        correspondant à la valeur de i
    fin pour
Fin tant que

```

Figure 3.33: procédure traçant les entités graphiques

Après l'affichage du graphique, la prochaine étape est de générer les séquences d'assemblage. Le logiciel permettant de le faire est GAPP (Generative Assembly Process Planner) [59]. Ce logiciel exige comme données d'entrée la liste des relations. La réussite du processus de génération par GAPP dépend de l'exactitude des relations obtenues aux différentes étapes précédentes. L'exemple 6.3 du chapitre 6 montre un exemple d'affichage des liens de contacts, de blocage et libre entre les composants d'un produit.

## CHAPITRE 4

### DÉFINITIONS DU REGROUPEMENT DES OPÉRATIONS ET DES PARAMÈTRES D'ÉVALUATION

Pour développer un modèle qui produira une séquence optimale, il faut donner plusieurs définitions basées sur le terme de regroupement. Ensuite, trouver les paramètres reliés à ces définitions sur lesquelles un raisonnement sera effectué. Ce chapitre développe les différentes définitions ainsi que leur paramètre d'évaluation.

#### 4.1. Regroupement des opérations en fonction de la direction d'assemblage

##### 4.1.1. Direction d'assemblage

La direction d'assemblage d'une opération est une direction suivant laquelle 2 sous-assemblages sont mis en contact. Cette direction peut être une des 6 directions (x+,x-,y+,y-,z+,z-) d'un repère orthonormé de l'espace. La figure 4.1 montre un exemple de direction d'assemblage de 2 composants.

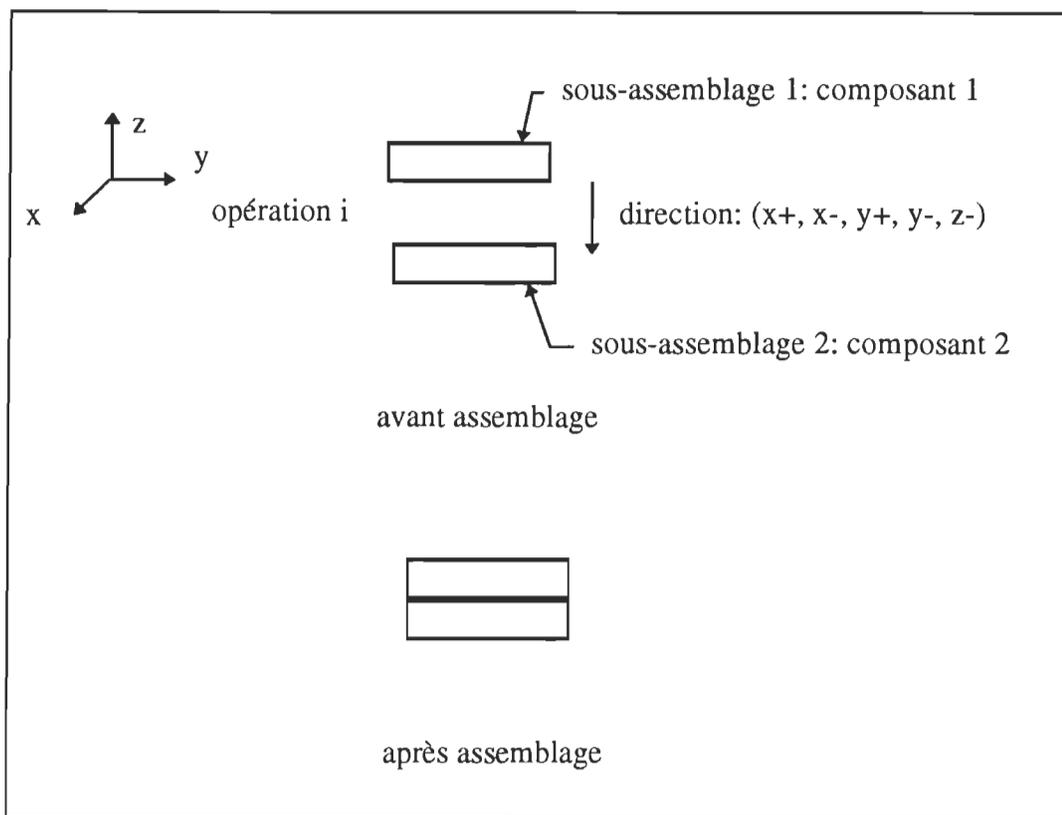


Figure 4.1: direction d'assemblage de 2 composants

#### **4.1.2. Paramètre d'évaluation de la direction d'assemblage: la matrice des demi-degrés de liberté**

Les éléments de la matrice des demi-degrés de liberté sont obtenus automatiquement par la détection des contacts (§3.2.2) entre les composants. L'existence d'un contact entre 2 composants conduit à la modification des éléments de la matrice des demi-degrés de liberté. Ces éléments ont une valeur de 1 dès le départ.

Après modification des valeurs des matrices suite à l'existence des contacts entre les composants, les éléments de la matrice ayant des valeurs de 0 traduisent un blocage dans la direction d'assemblage concernée tandis qu'une valeur de 1 traduit un mouvement libre dans la direction concernée.

#### **4.1.3. Utilisation de la matrice des demi-degrés de liberté**

Une relation existant entre 2 composants en contact donne plusieurs informations, soit le type de relation, le type de contact, le type d'opération, le nom des composants en contact, la direction d'assemblage de chaque composant et le degré de rupture des liaisons.

La génération de ces liaisons a été développée au §3.2.3. L'information concernant la direction d'assemblage des composants impliqués dans une relation est obtenue par l'extraction des valeurs des translations suivant les 3 directions (x, y, z) de la matrice des demi-degrés de liberté. Les valeurs des éléments des matrices qui sont des valeurs 1 ou 0 sont transformées en x+, x-, y+, y-, z+, z-. L'algorithme de la figure 4.2 permet de faire cette transformation.

```

Procédure make_relation (c1, c2, type relation, type opération , entier)

    Pour i allant de 1 au nombre maximal de ligne de la matrice
        si la première colonne de la première ligne a la valeur 0 et la
           deuxième colonne de cette ligne a la valeur 1
            mettre x- dans dir(1) (variable qui contient la
               direction d'assemblage du composant 1) et x+ dans
               dir(2) (variable qui contient la direction d'assemblage
               du composant 2)

        sinon si la première colonne de la deuxième ligne a la valeur 0 et
           la deuxième colonne de cette ligne a la valeur 1
            mettre y- dans dir(1) (variable qui contient la
               direction d'assemblage de composant 1) et y+ dans
               dir(2) (variable qui contient la direction d'assemblage
               du composant 2)

        sinon si la colonne de la troisième ligne a la valeur 0 et la
           deuxième colonne de cette ligne a la valeur 1
            mettre z- dans dir(1) (variable qui contient la
               direction d'assemblage de composant 1) et z+ dans
               dir(2) (variable qui contient la direction d'assemblage
               du composant 2)

        sinon si les colonnes des lignes ne contiennent que des valeurs 1,
           mettre dans les variables les axes correspondants.

    fin pour

    créer la relation et l'ajouter à la liste
  
```

Figure 4.2 : algorithme transformant les éléments de la matrice en des axes du système de coordonnées

Disposant des directions d'assemblage des composants impliqués dans les relations et pour une opération donnée, il suffit de consulter la variable réservée aux directions

d'assemblage pour connaître la direction d'assemblage à adopter pour unir les 2 sous-assemblages à l'intérieur de cette opération. L'exemple de la figure 4.1 donne le résultat suivant:

$$\begin{aligned}\text{dir}(1) &= \{ x+, x-, y+, y-, z- \} \\ \text{dir}(2) &= \{ x+, x-, y+, y-, z+ \}\end{aligned}$$

L'évaluation de la direction d'assemblage d'une opération se fait en utilisant la méthode de génération des séquences a priori. Au fur et à mesure que l'engin [59] assurant la génération des séquences d'assemblage génère des opérations en utilisant la méthode de désassemblage (rupture des liaisons), leurs directions d'assemblage sont évaluées. Le mécanisme de génération des séquences de l'engin est détaillé au chapitre 5. La méthode de génération des séquences a priori est aussi utilisée pour les autres définitions. Il existe des ensembles qui sont reliés aux 3 directions d'un axe ( x, y, z).

A chaque fois que 2 opérations consécutives ont la même direction d'assemblage, la valeur de l'ensemble correspondant à la direction d'assemblage de ces 2 opérations est augmentée de la valeur 1. Considérons l'exemple de la figure 4.3 pour illustrer cette méthode de regroupement.

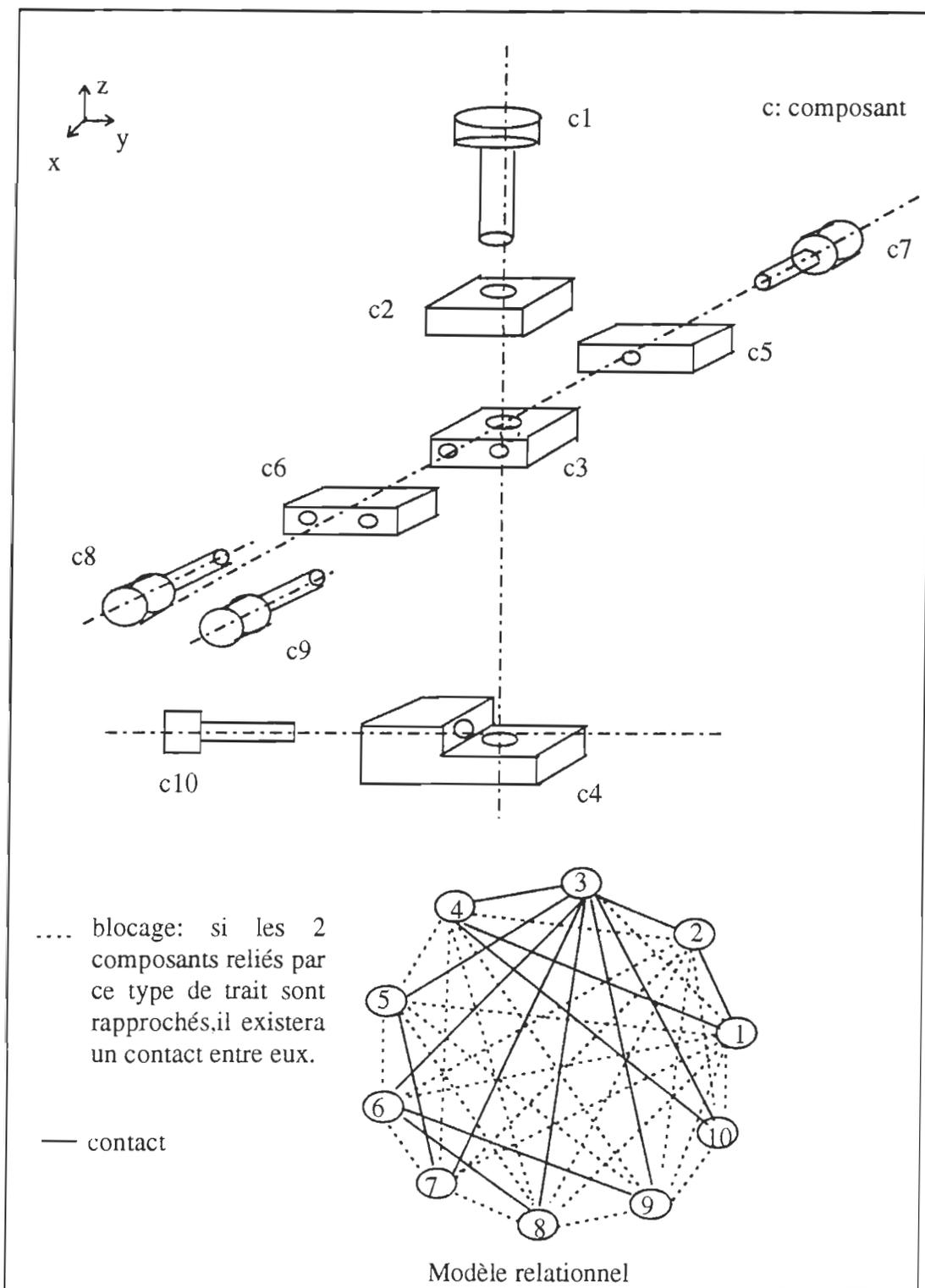


Figure 4.3 : exemple d'un produit à 10 composants

Dans la figure 4.3, le modèle relationnel met en évidence les relations de contact entre les composants. Ces relations de contact sont matérialisées par des traits pleins. Par exemple une relation de contact existe entre les composants 5 et 7. Un trait plein relie alors ces 2 composants dans le modèle relationnel.

La relation de blocage entre 2 composants est matérialisée dans le modèle relationnel par un trait pointillé. Deux composants reliés par un trait pointillé dans le modèle relationnel signifie que si ces 2 composants sont rapprochés, ils se toucheront. Trois types de contacts existent entre 2 composants : soit une relation de contact, soit une relation de blocage ou une relation libre ( pas de contact ni de blocage) qui est aussi représentée dans le modèle relationnel par un trait pointillé. L'application de ces types de relation entre les composants d'un produit conduit à l'obtention d'un graphe complet. Dans l'exemple de la figure 4.3, 45 relations seront générées en utilisant la formule 4.1.

$$n(n-1)/2 \text{ avec } n \text{ égal au nombre de composants.} \quad (4.1)$$

Dans le modèle relationnel de la figure 4.3, les traits pointillés spécifient soit un contact de blocage ou un contact libre. L'examen de la disposition des composants permet de distinguer ces 2 types de relations.

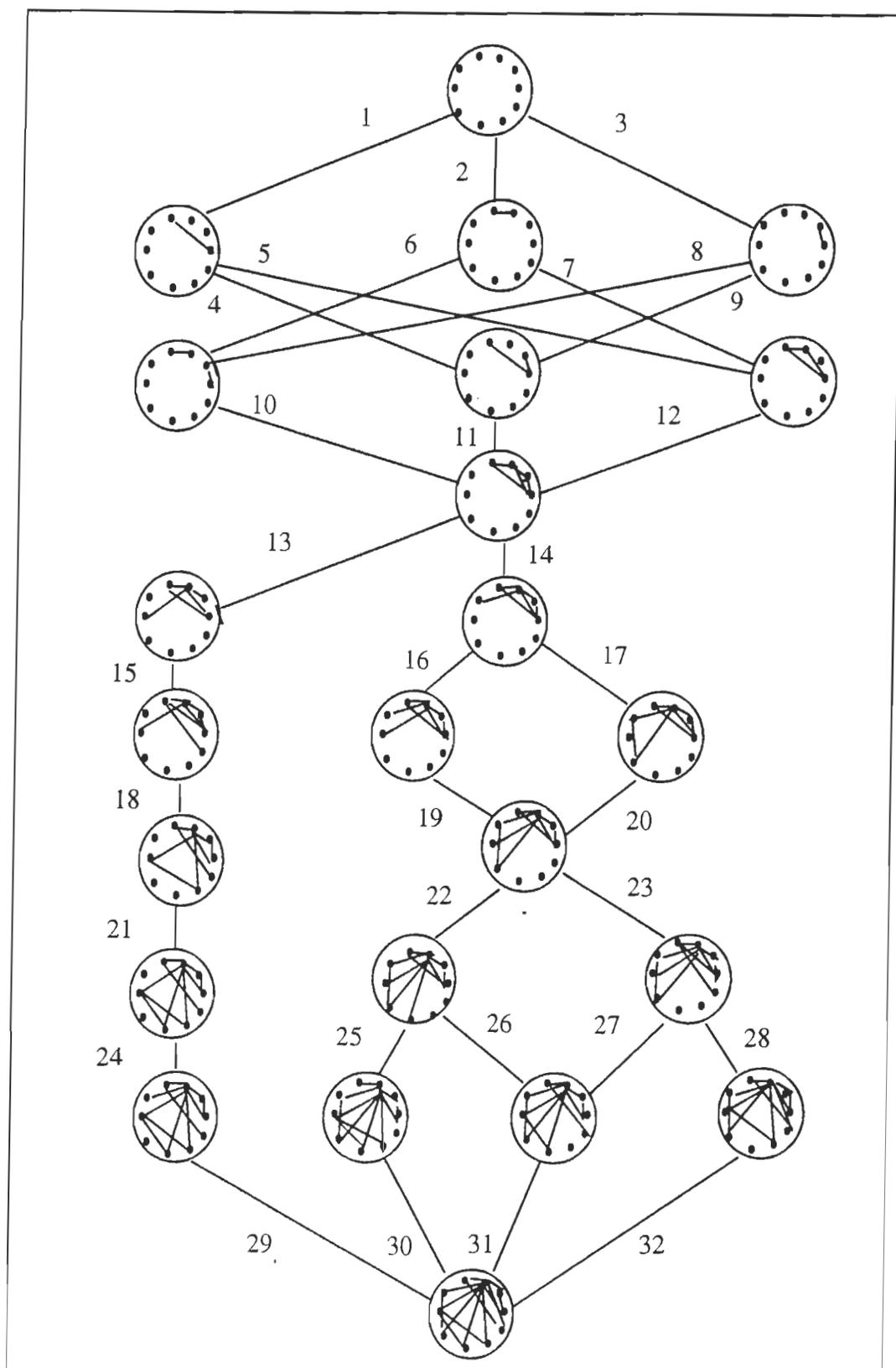


Figure 4.4: quelques séquences d'assemblage sans interférence géométrique

Pour ne pas surcharger les graphiques de la figure 4.4, les relations de blocage et libre n'ont pas été représentées. Les séquences de la figure 4.4 ne sont pas exhaustives. Celles représentées serviront à mettre en évidence l'utilisation de la définition 1.

Op. 1: dir(1) = {z+} dir(2) = {z-}	Op. 2: dir(1) = {z+} dir(2) = {z-}
Op. 3: dir(1) = {x+,x-,y+,y-,z+} dir(2) = {x+,x-,y+,y-,z-}	Op. 4: dir(1) = {x+,x-,y+,y-,z+} dir(2) = {x+,x-,y+,y-,z-}
Op. 5: dir(1) = {z+} dir(2) = {z-}	Op. 6: dir(1) = {x+,x-,y+,y-,z+} dir(2) = {x+,x-,y+,y-,z-}
Op. 7: dir(1) = {z+} dir(2) = {z-}	Op. 8: dir(1) = {z+} dir(2) = {z-}
Op. 9: dir(1) = {x+,x-,y+,y-,z+} dir(2) = {x+,x-,y+,y-,z-}	Op. 10: dir(1) = {z+} dir(2) = {z-}
Op. 11: dir(1) = {z+} dir(2) = {z-}	Op. 12: dir(1) = {z+} dir(2) = {z-}
Op. 13: dir(1) = {x+} dir(2) = {x-}	Op. 14: dir(1) = {x+} dir(2) = {x-}
Op. 15: dir(1) = {z+} dir(2) = {z-}	Op. 16: dir(1) = {z+} dir(2) = {z-}
Op. 17: dir(1) = {x+} dir(2) = {x-}	Op. 18: dir(1) = {x+} dir(2) = {x-}
Op. 19: dir(1) = {y+} dir(2) = {y-}	Op. 20: dir(1) = {x+} dir(2) = {x-}
Op. 21: dir(1) = {x+} dir(2) = {x-}	Op. 22: dir(1) = {x+} dir(2) = {x-}
Op. 23: dir(1) = {x+} dir(2) = {x-}	Op. 24: dir(1) = {x+} dir(2) = {x-}
Op. 25: dir(1) = {x+} dir(2) = {x-}	Op. 26: dir(1) = {x+} dir(2) = {x-}
Op. 27: dir(1) = {x+} dir(2) = {x-}	Op. 28: dir(1) = {x+} dir(2) = {x-}
Op. 29: dir(1) = {x+} dir(2) = {x-}	Op. 30: dir(1) = {x+} dir(2) = {x-}
Op. 31: dir(1) = {x+} dir(2) = {x-}	Op. 32: dir(1) = {x+} dir(2) = {x-}

Figure 4.5 : liste des directions d'assemblage de la figure 4.4

Pour illustrer l'utilisation de la définition 1, prenons 9 séquences de la figure 4.4 et examinons la direction d'assemblage des opérations impliquées dans ces séquences. Dans cette figure, op signifie opération.

**Séquence 1: opérations: 29,24,21,18,15,13, 10, 6, 2**

Les opérations 29, 24, 21,18, 15, 13, sont consécutives et ont la même direction d'assemblage ( figure 4.5). L'ensemble relié à l'axe x est augmenté de la valeur 5. Les opérations consécutives 10, 6, 2 ont la même direction d'assemblage. L'ensemble relié à la direction z est augmenté de la valeur 2 car il existe 2 opérations consécutives ( 10-6, 6-2). Pour cette séquence, l'ensemble relié à l'axe x a la valeur 5, celui relié à l'axe y a la valeur 0 et celui relié à l'axe z a la valeur 2.

**Séquence 2: opérations: 30, 25, 22, 19, 16, 14, 10, 6, 2**

Les opérations 30, 25 et 22 sont consécutives et ont la même direction d'assemblage ( figure 4.5). Ainsi que celle de 16 et 14. L'ensemble relié à l'axe x est augmenté de la valeur 3. Les opérations consécutives 10, 6, 2 ont la même direction d'assemblage. L'ensemble relié à la direction z est augmenté de la valeur 2. Pour cette séquence, l'ensemble relié à l'axe x a la valeur 3, celui relié à l'axe y a la valeur 0 et celui relié à l'axe z a la valeur 2.

**Séquence 3: opérations: 30, 25, 22, 20, 17, 14, 10, 6, 2**

Les opérations 30, 25, 22, 20, 17 et 14 sont consécutives et ont la même direction d'assemblage ( figure 4.5). L'ensemble relié à l'axe x est augmenté de la valeur 5. Les opérations consécutives 10, 6, 2 ont la même direction d'assemblage. L'ensemble relié à la direction z est augmenté de la valeur 2. Pour cette séquence, l'ensemble relié à l'axe x a la valeur 5, celui relié à l'axe y a la valeur 0 et celui relié à l'axe z a la valeur 2.

**Séquence 4: opérations: 31, 26, 22, 19, 16, 14, 10, 6, 2**

Les opérations 31, 26 et 22 sont consécutives et ont la même direction d'assemblage ( figure 4.5). Ainsi que 16 et 14. L'ensemble relié à l'axe x est augmenté de la valeur 3. Les opérations consécutives 10, 6, 2 ont la même direction d'assemblage. L'ensemble relié à la direction z est augmenté de la valeur 2. Pour cette séquence, l'ensemble relié à l'axe x a la valeur 3, celui relié à l'axe y a la valeur 0 et celui relié à l'axe z a la valeur 2.

**Séquence 5: opérations: 31, 26, 22, 20, 17, 14, 10, 6, 2**

Les opérations 31, 26, 22, 20, 17 et 14 sont consécutives et ont la même direction d'assemblage ( figure 4.5). L'ensemble relié à l'axe x est augmenté de la valeur 5. Les

une seule direction. Elle aura pour effet de simplifier les fixations, de nécessiter peu d'habileté par exemple pour le robot, d'éviter des extra-opérations et des réorientations.

## **4.2. Regroupement des opérations en fonction du changement des outils**

### **4.2.1. Forme de l'outil**

L'exécution d'une opération d'assemblage, qui consiste à unir 2 sous-assemblages à l'intérieur de cette opération, nécessite un outil pour manipuler ces sous-assemblages. L'outil de saisie doit avoir la même forme que la morphologie des sous-assemblages. Les formes usuelles des composants sont du type cylindrique, prismatique, sphérique, conique ou complexe. Ces formes usuelles des composants détermineront donc les formes des outils de manipulation. L'exemple de la figure 4.1 nécessitera un outil de forme prismatique pour manipuler les composants impliqués dans l'opération étant donné que la forme de ces composants est un prisme régulier.

### **4 2 2. Paramètre d'évaluation de la forme de l'outil: morphologie des composants**

Introduisons la notion de composant de base et la notion de composant déplacé. Un composant est dit de base s'il reçoit un autre composant. Un composant est dit déplacé s'il est amené en contact avec un autre composant.

Le composant déplacé comporte 2 régions: une région de saisie et une autre d'insertion. La figure 4.6 met en évidence les différentes régions d'un composant.

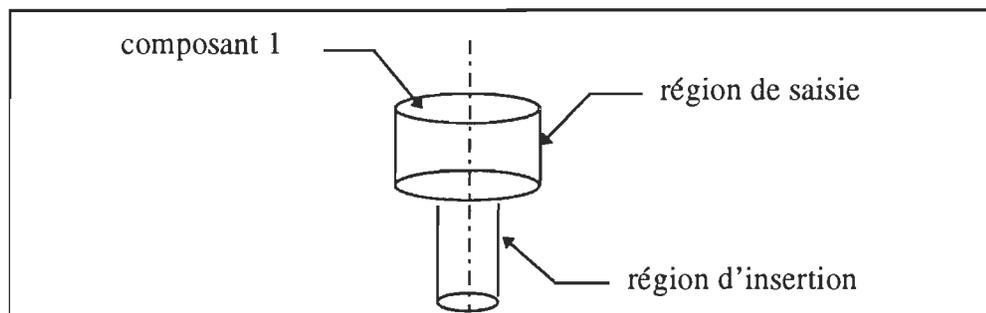


Figure 4.6: les régions de saisie et d'insertion d'un composant

Le composant de base comporte aussi 2 régions, la région réceptrice et la région fixe. La figure 4.7 montre un exemple de ces régions.

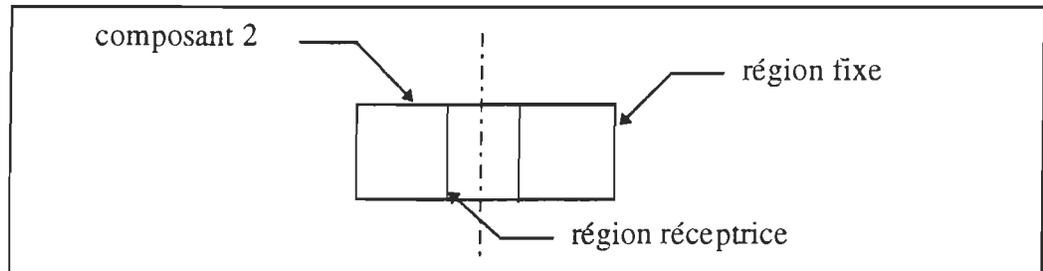


Figure 4.7: région fixe et réceptrice d'un composant

Les différentes régions des composants déplacés et ceux de base ont différents types de formes.

Ainsi, la région saisie et la région insérée du composant déplacé peuvent être soit de la forme cylindrique, sphérique, prismatique, triangulaire ou complexe.

La région réceptrice et ou la région fixe du composant de base peuvent être aussi soit de la forme cylindrique, sphérique, prismatique, triangulaire ou complexe. Dans ce projet, seule la région saisie sera prise en considération car c'est elle qui est en contact direct avec l'outil.

#### 4.2.3. L'utilisation de la morphologie des composants

Dans une opération, il existe un composant d'un sous-assemblage qui touche la ligne de rupture des liaisons. La figure 4.8 montre un exemple de composants qui touche la ligne de rupture

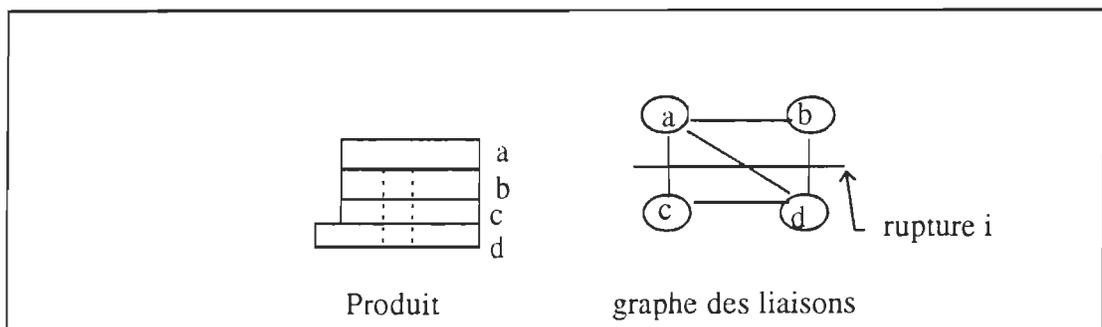


Figure 4.8: composant touchant la ligne de rupture

Il existe une branche dans le domaine d'assemblage, "grasping", qui étudie tous les aspects reliés à la surface de saisie d'un composant en tenant compte d'un certain nombre

de critères (surface usinée, surface non usinée, surface accessible). Dans ce projet, nous nous limiterons à la détermination simple, en ne tenant pas compte de ces critères, de la forme de saisie (région de saisie) d'un sous-assemblage formé de plusieurs composants ou d'un seul composant. Cette forme de saisie sera nommée forme désignée.

Dans la figure 4.8 les composants a et b touchent la ligne de rupture. Donc pour une opération donnée, une forme appelée désignée de chaque sous-assemblage est égale à la forme du composant le plus éloigné du composant qui touche la ligne de rupture en faisant l'hypothèse que tous les sous-assemblages sont stables s'ils sont déplacés. Cette méthode constitue donc la méthode de détermination simple mentionnée plus haut. Dans le cas de la figure 4.8, la forme désignée pour le sous-assemblage constitué des composants a et b sera la forme du composant a étant donné qu'il est le plus éloigné. L'algorithme permettant de déterminer cette forme commune est présenté à la figure 4.9.

```

procédure find_common_shape()
    Tant que la liste des relations d'un sous-assemblage est non nulle
        Prendre une relation de la liste
        Trouver le nom des composants de cette relation
            Forme de saisie = forme du composant le plus éloigné du
                                composant touchant la ligne de rupture
    Fin tant que
  
```

Figure 4.9: algorithme déterminant la forme désignée d'un sous-assemblage

Si la forme de l'outil nécessaire (forme désignée) à l'exécution de 2 opérations consécutives est identique, cette forme est mémorisée. Le parcours de toutes les opérations de toutes les séquences permet de retenir les formes utilisées pour exécuter les opérations à l'intérieur des séquences.

Pour illustrer cette méthode, reprenons l'exemple de la figure 4.3 et 4.4 en considérant toujours 9 séquences. La liste des formes des composants à l'intérieur des opérations est présentée à la figure 4.10.

Opération 1: rs(1) = {C} rs(2) = {P}	Opération 2: rs(1) = {P} rs(2) = {P}
Opération 3: rs(1) = {C} rs(2) = {P}	Opération 4: rs(1) = {C} rs(2) = {P}
Opération 5: rs(1) = {C} rs(2) = {P}	Opération 6: rs(1) = {C} rs(2) = {P}
Opération 7: rs(1) = {C} rs(2) = {P}	Opération 8: rs(1) = {P} rs(2) = {P}
Opération 9: rs(1) = {C} rs(2) = {P}	Opération 10: rs(1) = {C} rs(2) = {P}
Opération 11: rs(1) = {C} rs(2) = {P}	Opération 12: rs(1) = {C} rs(2) = {P}
Opération 13: rs(1) = {P} rs(2) = {P}	Opération 14: rs(1) = {P} rs(2) = {P}
Opération 15: rs(1) = {P} rs(2) = {P}	Opération 16: rs(1) = {P} rs(2) = {P}
Opération 17: rs(1) = {C} rs(2) = {P}	Opération 18: rs(1) = {C} rs(2) = {P}
Opération 19: rs(1) = {C} rs(2) = {P}	Opération 20: rs(1) = {P} rs(2) = {P}
Opération 21: rs(1) = {C} rs(2) = {P}	Opération 22: rs(1) = {C} rs(2) = {P}
Opération 23: rs(1) = {P} rs(2) = {P}	Opération 24: rs(1) = {C} rs(2) = {P}
Opération 25: rs(1) = {C} rs(2) = {C}	Opération 26: rs(1) = {P} rs(2) = {P}
Opération 27: rs(1) = {C} rs(2) = {C}	Opération 28: rs(1) = {C} rs(2) = {C}
Opération 29: rs(1) = {C} rs(2) = {C}	Opération 30: rs(1) = {P} rs(2) = {P}
Opération 31: rs(1) = {C} rs(2) = {C}	Opération 32: rs(1) = {C} rs(2) = {C}

Figure 4.10 : liste de la forme des composants des opérations de la figure 4.4

La variable rs(1) indique la forme désignée du sous-assemblage 1 (sous-assemblage déplacé) et rs(2) indique la forme désignée du sous-assemblage 2 (sous-assemblage fixe). La lettre P indique qu'il s'agit d'une forme prismatique et C pour une forme cylindrique.

**Séquence 1: opérations: 29, 24, 21, 18, 15, 13, 10, 6, 2**

Les opérations 29, 24, 21 et 18 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle de 15 et 13, 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 4. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 4.

**Séquence 2: opérations: 30, 25, 22, 19, 16, 14, 10, 6, 2**

Les opérations 25, 22 et 19 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle de 16 et 14, 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 4. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 4.

**Séquence 3: opérations: 30, 25, 22, 20, 17, 14, 10, 6, 2**

Les opérations 25, 22 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle de 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 2. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 2.

**Séquence 4: opérations: 31, 26, 22, 19, 16, 14, 10, 6, 2**

Les opérations 22, 19 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle 16 et 14, 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 3. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 3.

**Séquence 5: opérations: 31, 26, 22, 20, 17, 14, 10, 6, 2**

Les opérations 10 et 6 sont consécutives et ont la même forme désignée (figure 4.11). L'ensemble relié à la forme de l'outil est augmenté de la valeur 1. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 1.

**Séquence 6: opérations: 31, 27, 23, 19, 16, 14, 10, 6, 2**

Les opérations 31 et 27 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 2. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 2.

**Séquence 7: opérations: 31, 27, 23, 20, 17, 14, 10, 6, 2**

Les opérations 31 et 27, 23 et 20 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle de 10 et 6. L'ensemble relié à la forme de l'outil est

augmenté de la valeur 3. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 3.

#### **Séquence 8: opérations: 32, 28, 23, 19, 16, 14, 10, 6, 2**

Les opérations 32, 28 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle 16 et 14, et 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 3. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 3.

#### **Séquence 9: opérations: 32, 28, 23, 20, 17, 14, 10, 6, 2**

Les opérations 32 et 28, 23 et 20 sont consécutives et ont la même forme désignée (figure 4.11). Ainsi que celle de 10 et 6. L'ensemble relié à la forme de l'outil est augmenté de la valeur 3. Pour cette séquence, l'ensemble relié à la forme de l'outil a une valeur de 3.

L'examen de ces formes permet d'effectuer des regroupements en fonction de la similarité de 2 opérations consécutives. Le choix de la séquence optimale sera discuté dans le chapitre 5. L'objectif de ce regroupement est d'avoir un nombre minimum de changement d'outil indépendamment de la direction d'assemblage car il aura un gain énorme en temps pendant le processus d'assemblage.

### **4.3. Regroupement des opérations en fonction des tolérances**

Dans la littérature sur le problème d'assemblage, les différentes méthodes proposées par les auteurs ne tiennent pas compte des tolérances des différentes pièces impliquées dans l'assemblage. Cette définition permet de faire des regroupements d'opérations basés sur la **fonction d'assemblage (fa)**. Deux opérations ayant la même fonction d'assemblage sont regroupées en un seul ensemble. Cette fonction d'assemblage s'obtient en faisant l'intersection des **contraintes d'assemblage** des différentes liaisons impliquées dans une opération.

#### **4.3.1. Paramètre d'évaluation: accumulation des tolérances**

La philosophie du dimensionnement et le choix des tolérances est une fonction de design qui n'affecte pas seulement la fonctionnalité du produit mais aussi la fabrication et l'assemblage. Le dimensionnement et les tolérances associées affectent l'assemblage de 2 façons:

- assurent que l'assemblage est facile, moyen ou difficile pour **une technique ou une technologie** comparativement à d'autres.
- assurent que l'exécution des opérations d'assemblage est facile par certaines séquences, moyenne ou difficile par d'autres.

Les **contraintes d'assemblage** usuelles dans un processus d'assemblage sont les suivantes:

- des contraintes de contact, d'attachement, de dimension d'assemblage et d'insertion.
- des contraintes d'assemblage basées sur la tolérance de localisation (position, concentricité..), d'orientation (perpendicularité, angularité, parallélisme..).

#### **4.3.2. Utilisation de l'accumulation des tolérances**

La fonction d'assemblage est déterminée par la sommation des intervalles de tolérance: d'ajustement (**aj**) entre 2 composants cylindriques, linéaire (**tl**) dans le cas des composants non cylindriques, de position (**tp**), d'orientation (**to**), de l'outil (**tou**) et du type de relation (**tr**) quelle que soit la direction d'assemblage. La formulation mathématique de cette fonction est la suivante :

$$fa = (tl1 + tl2+..+ tln) + ( aj1+ aj2+..+ajn) + ( tp1 + tp2 +..+ tpn) + (to1 + to2+..+ ton) + (tou1+ tou2 +..+ toun)] + (tr1 + tr2 +..+trn).....(4.2)$$

Les valeurs de ces intervalles de tolérance sont connues car elles sont inscrites sur le schéma des composants. Seul aj est déterminé automatiquement [3] en fonction du diamètre de l'alésage.

Dans un processus de génération des séquences d'assemblage, au fur et à mesure qu'un état d'assemblage est généré à partir d'un état parent, la fonction d'assemblage associée à cet état enfant est calculée.

Il existe un ensemble relié à la fonction d'assemblage. Si les fonctions d'assemblage de 2 opérations consécutives sont identiques, la valeur de cet ensemble est augmentée de 1.

On procède ainsi pour toutes les opérations de toutes les séquences possibles. La séquence dont la valeur de l'ensemble relié à la fonction d'assemblage est élevée est considérée optimale.

Pour illustrer cette approche, examinons l'assemblage des 4 blocs de la figure 4.11.

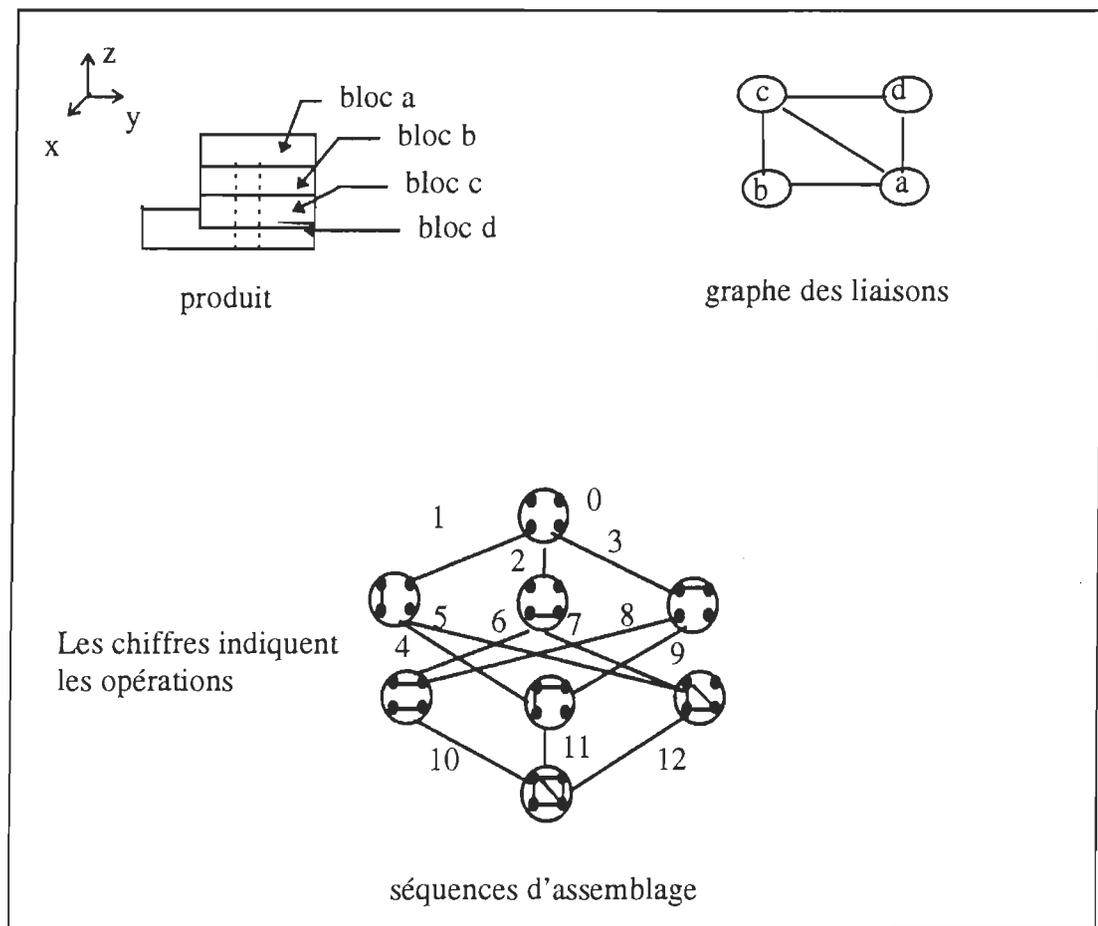


Figure 4.11: produit de 4 blocs

La figure 4.12 indique les tolérances des composants.

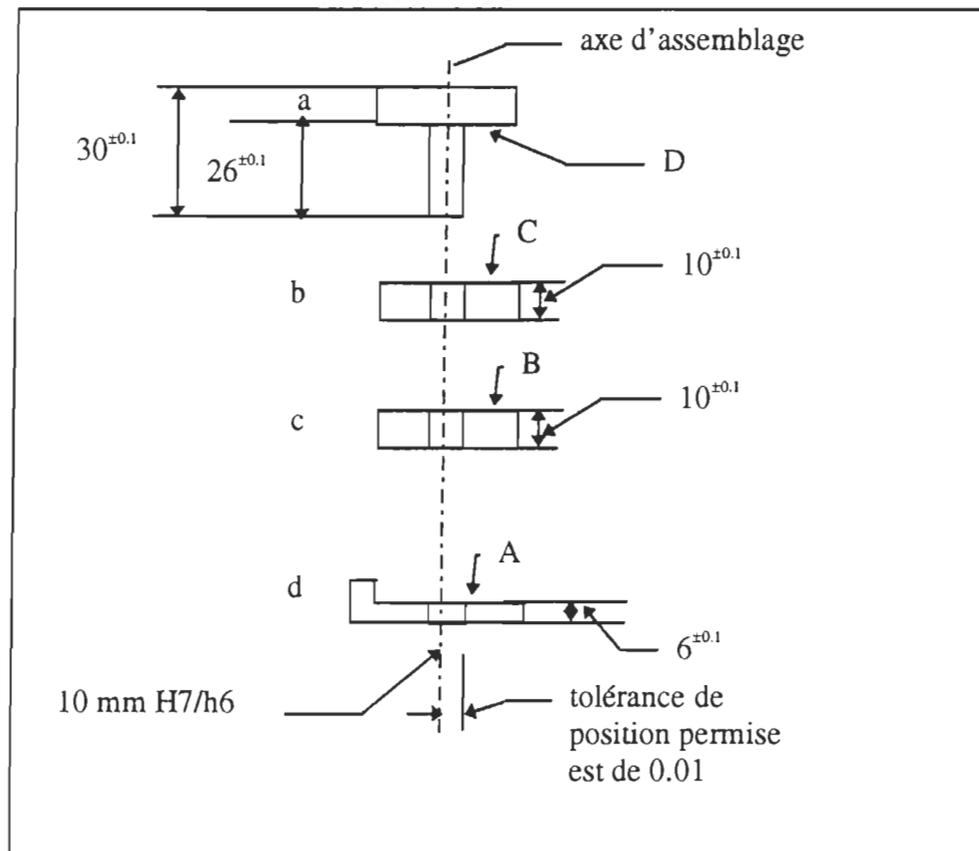


Figure 4.12: tolérances des composants

Dans la figure 4.12, la tolérance de position par rapport à l'axe d'assemblage permise est la valeur en dehors de laquelle une erreur de position peut survenir lors de l'assemblage des composants. En effet si le robot ne se positionne pas par rapport à l'axe d'assemblage en respectant la tolérance de position permise, il risque de ne pas réussir l'opération d'assemblage.

#### Attribution des poids (de 0 à 1) aux différents types de relations

Une relation de type contact est facile (0.25). La relation de type attachement est considérée moyenne (0.5) et celle d'insertion est difficile (0.75).

#### 4.3.3. Calcul des fonctions d'assemblage des 6 séquences

##### Séquence 1: opérations: 11, 9, 3

fa opération 11 = une contrainte de position + une contrainte d'insertion avec  $tr = 0.75$

fa opération 11 =  $0.2 + 0.2 + 0.01 + 0.024 + 0.75$

fa opération 11 = 1.184

Intervalle de tolérance de H7 = 0.024 avec un diamètre de 10 mm [3]

fa opération 9 = une contrainte de position + une contrainte de contact plan  
avec  $tr = 0.25$

fa opération 9 =  $0.2 + 0.2 + 0.01 + 0.024 + 0.25$

fa opération 9 = 0.684

fa opération 3 = une contrainte de position + une contrainte de contact (0.25)

fa opération 3 =  $0.2 + 0.2 + 0.01 + 0.25$

fa opération 3 = 0.66

Pas de regroupement car les valeurs de fa des 3 opérations sont différentes.

### **Séquence 2: opérations: 11, 4, 1**

fa opération 11 = 1.184

fa opération 4 = une contrainte de position + une contrainte de contact plan avec  
 $tr = 0.25$

fa opération 4 =  $0.2 + 0.2 + 0.01 + 0.25$

fa opération 4 = 0.66

fa opération 1 = une contrainte de position + une contrainte de contact plan  
avec  $tr = 0.25$

fa opération 1 =  $0.2 + 0.2 + 0.01 + 0.25$

fa opération 1 = 0.66

Les opérations 4 et 1 ont la même fonction d'assemblage. L'ensemble relié à la fonction d'assemblage est augmenté de la valeur 1. Pour cette opération, l'ensemble relié à la fonction d'assemblage a une valeur de 1.

### **Séquence 3: opérations : 10, 8, 3**

fa opération 10 = une contrainte de position + une contrainte d'insertion  
avec  $tr = 0.75$

fa opération 10 =  $0.2 + 0.2 + 0.01 + 0.024 + 0.75$

fa opération 10 = 1.184

fa opération 8 = une contrainte de position + une contrainte d'insertion avec  
 $tr = 0.75$

$$\text{fa opération 8} = 0.2 + 0.2 + 0.01 + 0.024 + 0.75$$

$$\text{fa opération 8} = 1.184$$

$$\text{fa opération 3} = 0.66$$

Les opérations 10 et 8 ont la même fonction d'assemblage. L'ensemble relié à la fonction d'assemblage est augmenté de la valeur 1. Pour cette opération, l'ensemble relié à la fonction d'assemblage a une valeur de 1.

#### **Séquence 4: opérations : 10, 6, 2**

$$\text{fa opération 10} = 1.184$$

fa opération 6 = une contrainte de position + une contrainte d'insertion avec  $tr = 0.75$

$$\text{fa opération 6} = 0.2 + 0.2 + 0.01 + 0.024 + 0.75$$

$$\text{fa opération 6} = 1.184$$

fa opération 2 = une contrainte de position + une contrainte de contact avec  $tr = 0.25$

$$\text{fa opération 2} = 0.2 + 0.2 + 0.01 + 0.024 + 0.25$$

$$\text{fa opération 2} = 0.66$$

Les opérations 10 et 6 ont la même fonction d'assemblage. L'ensemble relié à la fonction d'assemblage est augmenté de la valeur 1. Pour cette opération, l'ensemble relié à la fonction d'assemblage a une valeur de 1.

#### **Séquence 5: opérations: 12, 7, 2**

fa opération 12 = une contrainte de position + une contrainte d'insertion avec  $tr = 0.75$

$$\text{fa opération 12} = 0.2 + 0.2 + 0.01 + 0.024 + 0.75$$

$$\text{fa opération 12} = 1.184$$

fa opération 7 = une contrainte de position + une contrainte de contact avec  $tr = 0.25$

$$\text{fa opération 7} = 0.2 + 0.2 + 0.01 + 0.024 + 0.25$$

$$\text{fa opération 7} = 0.66$$

fa opération 2 = une contrainte de position + une contrainte d'insertion avec  $tr = 0.75$

fa opération 2 =  $0.2 + 0.2 + 0.01 + 0.024 + 0.75$   
fa opération 2 = 1.184

Pas de regroupement car les valeurs de fa des 3 opérations sont différentes.

**Séquence 6: opérations: 12, 5, 2**

fa opération 2 = 1.184  
fa opération 4 = 0.66  
fa opération 11 = 1.184

Pas de regroupement car les valeurs de fa des 3 opérations sont différentes.

Le choix de la séquence optimale sera discuté dans le chapitre 5. L'objectif de ce regroupement est d'effectuer le maximum d'opérations avec un degré de difficulté d'exécution moindre. Ce qui a pour effet de faciliter la tâche au robot qui exécutera cette séquence optimale.

## CHAPITRE 5

### DÉTERMINATION DE LA SÉQUENCE OPTIMALE PAR LE REGROUPEMENT DES OPÉRATIONS

#### 5.1. La génération des séquences d'assemblage

Pour sélectionner une séquence d'assemblage optimale, il faut disposer d'un certain nombre de séquences. Pour générer toutes ces séquences d'assemblage, nous utiliserons les travaux [59] dont les résultats ont débouché sur l'implantation d'un engin. Cet engin utilise l'approche générative pour générer les séquences d'assemblage et dont l'acronyme est GAPP (Generative Assembly Process Planner). La méthode de génération des séquences est basée sur la rupture des liaisons (figure 2.5) et sur le graphe dirigé des états d'assemblage. L'engin utilise 4 étapes pour produire les séquences:

1- Création de la liste des relations entre les composants du produit. GAPP reçoit comme données d'entrée les relations créées au § 3.2.3 avec le format requis.

2- La sélection d'un arbre sous-tendant minimum du graphe des liaisons. Cet arbre est sélectionné par l'algorithme de Kruskal à partir du graphe des liaisons. La figure 5.1 montre un exemple d'arbre sous-tendant minimum du graphe des liaisons des 4 blocs.

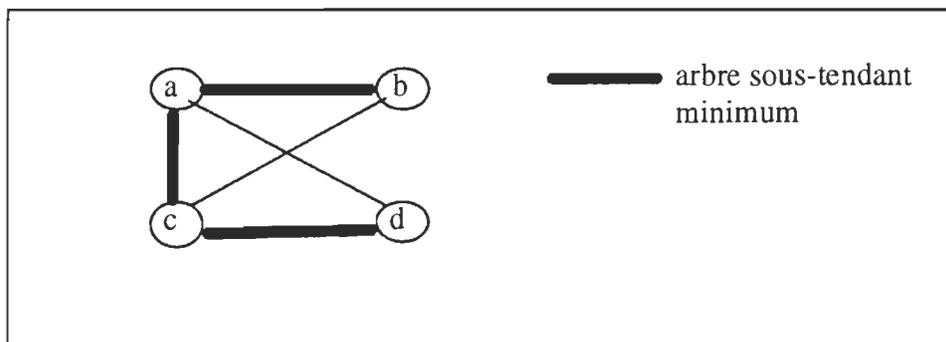


Figure 5.1: arbre sous-tendant minimum du graphe des liaisons des 4 blocs

3- Génération de toutes possibilités de rupture des liaisons à partir du graphe des liaisons (cutsets).

A partir de l'arbre sous-tendant minimum sélectionné, GAPP génère le système fondamental relié à cet arbre. Ensuite toutes les combinaisons de rupture des liaisons possibles sont déterminées à partir de ce système fondamental. GAPP utilise un mot

de 32 bits pour faire cette combinaison. L'exemple tiré de [59] qui illustre bien cette génération est présenté à la figure 5.2

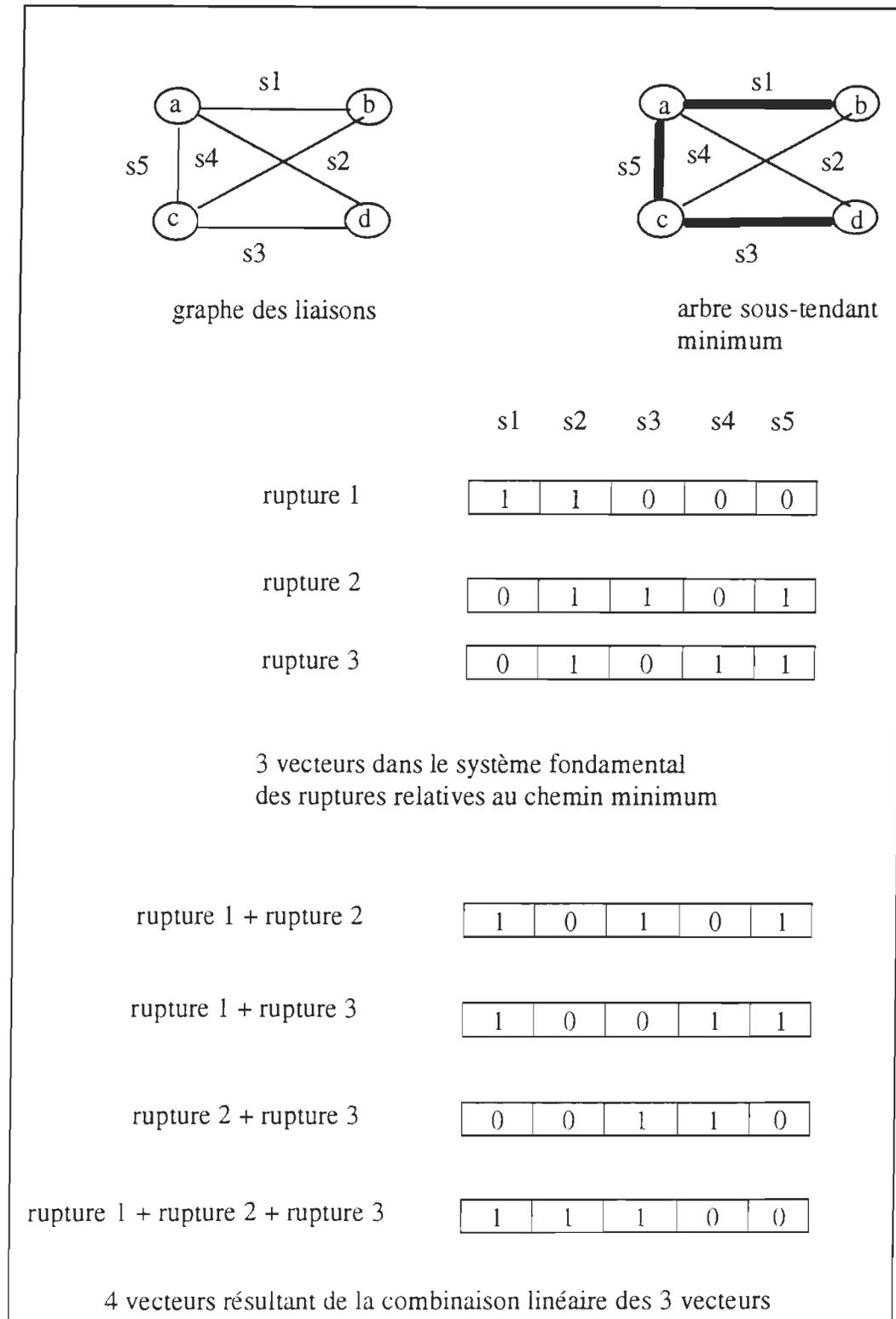


Figure 5.2: génération de toutes les possibilités de rupture des liaisons du graphe

Un problème de dépassement de capacité se produit lorsque le nombre de relations du produit dépasse 32. Pour pallier à ce problème, nous avons utilisé la position des relations impliquées dans la rupture des liaisons au lieu des positions transformées en base 2 comme le fait GAPP.

#### 4- Génération des séquences

Disposant de toutes les combinaisons de rupture possibles, le processus de génération du graphe dirigé des états d'assemblage est assimilé à un problème de la théorie des graphes. Ainsi la notion de noeuds (états) est utilisée. Deux listes existent: la liste des noeuds non traités (open list) et la liste des noeuds traités (close list). Le processus s'arrête lorsque la liste des noeuds non traités est vide[59].

Un noeud comprend un certain nombre d'ensembles de possibilités de rupture des liaisons. Si un élément de cet ensemble respecte la contrainte de géométrie (existence d'une direction d'assemblage), un enfant de ce noeud parent est créé et est introduit dans la liste des noeuds non traités. Si toutes les possibilités de rupture de ce noeud sont épuisées, le noeud est introduit dans la liste des noeuds traités. Les séquences sont retracées par le lien de parent-enfant entre les noeuds.

### **5.2 . Sélection d'une séquence optimale d'assemblage par le regroupement des opérations**

Lorsqu'un noeud (état du processus d'assemblage) est sélectionné dans la liste des noeuds non traités, les possibilités de rupture (opérations) reliées à ce noeud sont évaluées par GAPP pour vérifier si elles ne provoquent pas d'interférence géométrique.

Si ces possibilités ne provoquent pas d'interférence géométrique (il existe une direction d'assemblage suivant laquelle les possibilités de rupture peuvent être effectuées) un noeud enfant est créé. La procédure dans GAPP qui réalise cette vérification est "check\_predicates". C'est à l'intérieur de cette procédure que les directions d'assemblage, la forme de l'outil (forme de saisie des composants) et les tolérances seront déterminées.

#### **5.2.1. Mémorisation de la similitude de la direction d'assemblage, de la forme de saisie et de la fonction d'assemblage de 2 opérations consécutives d'un noeud**

A cette étape du processus de génération des séquences d'assemblage, la réalisation pratique des § 4.1.3, 4.2.4 et 4.3.3 est effectuée. A chaque fois que GAPP créé un nouveau noeud, ce noeud est appelé noeud enfant tandis que l'ancien noeud est appelé noeud parent. Pour mémoriser l'état des ensembles décrits aux § 4.1.3, 4.2.4 et 4.3.3, des variables ont été définies. La variable reliée à la direction d'assemblage en x est dir\_x, celle en y est dir\_y et celle en z est dir\_z. La variable reliée à la forme de saisie des composants est cfo et celle reliée à la tolérance est cfa.

Donc l'augmentation des valeurs de ces ensembles mentionnée dans les § 4.1.3, 4.2.4 et 4.3.3 sera donc matérialisée par les variables énoncées ci-dessus. Le noeud

parent et enfant de chaque opération provoquant une augmentation de la valeur de ces variables est mémorisé par un pointeur si la direction d'assemblage d'une opération du noeud parent et du noeud enfant sont identiques. Un mécanisme existe dans GAPP qui permet de diriger le pointeur sur les noeuds parents et les noeuds enfants en fonction du coût des noeuds. La valeur des variables décrites plus haut constitue donc le coût des noeuds. Cette manière de diriger le pointeur sur les noeuds ayant un plus faible coût permet de garder en mémoire le chemin optimal en fonction des coûts des noeuds, donc la séquence d'assemblage optimale. L'algorithme permettant de modifier la valeur des ensembles est présenté à la figure 5.3.

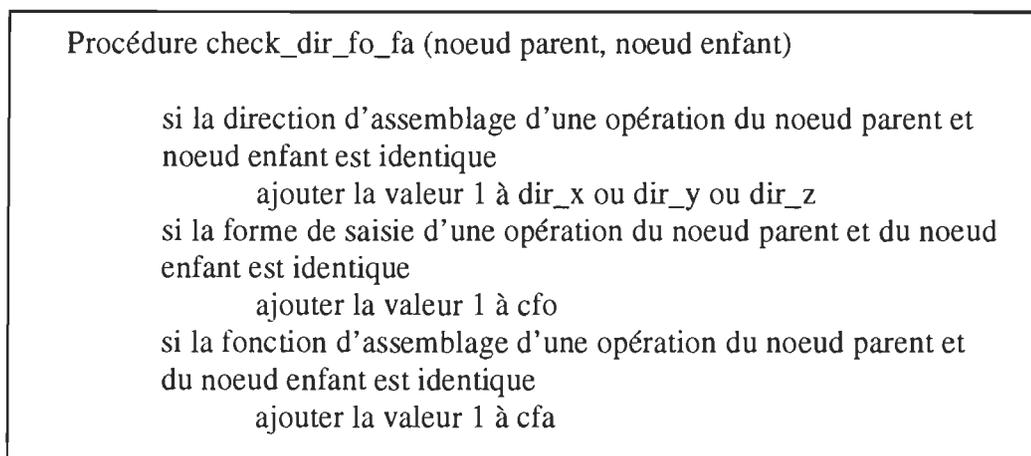


Figure 5.3: mémorisation de la direction d'assemblage, de la forme de saisie et de la fonction d'assemblage

### **5.2.2. Proposition d'une séquence optimale d'assemblage**

La détermination de la séquence optimale en général se fait en examinant la valeur des variables associées aux directions d'assemblage, à la forme de saisie des composants et à la tolérance des opérations des noeuds décrites plus haut.

La séquence d'assemblage optimale du point de vue de la direction d'assemblage sera la séquence dont la valeur de la somme des variables dir\_x, dir\_y et dir\_z de ses opérations est plus élevée par rapport à celle des opérations des autres séquences. Le coût associé à un noeud est inversement proportionnel à la valeur des variables reliées aux définitions 1, 2 et 3.

La séquence d'assemblage optimale du point de vue de la forme de saisie des composants sera la séquence dont la valeur cfa de ses opérations est la plus élevée par rapport à celle des opérations des autres séquences d'assemblage.

La séquence d'assemblage optimale du point de vue de la tolérance sera la séquence dont la valeur de variable cfo est la plus élevée par rapport à celle des opérations des autres séquences d'assemblage.

Pour illustrer ce choix, reprenons les exemples des § 4.1.3, 4.2.4 et 4.3.3. Les séquences optimales du point de vue de la direction d'assemblage parmi les 9 séquences considérées sont présentées à la figure 5.4.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
dir_x	5	3	5	3	5	3	5	3	5
dir_y	0	0	0	0	0	0	0	0	0
dir_z	2	2	2	2	2	2	2	2	2
total	7	5	7	5	7	5	7	5	7

Figure 5.4 : valeurs des variables dir\_x, dir\_y et dir\_z des séquences

Dans cette figure, s1 indique la séquence numéro 1. La figure 5.5 présente la transformation de ces valeurs dir\_x, dir\_y et dir\_z en coût.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
dir_x	1/5	1/3	1/5	1/3	1/5	1/3	1/5	1/3	1/5
dir_y	0	0	0	0	0	0	0	0	0
dir_z	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2
total	7/10	5/6	7/10	5/6	7/10	5/6	7/10	5/6	7/10

Figure 5.5 : coûts associés aux variables dir\_x, dir\_y et dir\_z des séquences

L'examen de la figure des coûts permet de dire que 5 séquences sont optimales du point de vue de la direction d'assemblage. (séquence 1, séquence 3, séquence 5, séquence 7, séquence 9). Ces séquences ont un coût faible par rapport aux autres séquences. On procède ainsi pour les autres définitions.

En cas d'égalité de coût entre les séquences, la séquence qui aura le coût le plus faible parmi les variables dir\_x, dir\_y et dir\_z sera considérée optimale. Dans l'exemple ci-dessus cette règle n'est pas applicable car les 5 séquences optimales ont toutes des coûts des variables dir\_x, dir\_y et dir\_z identiques.

## CHAPITRE 6

### RÉSULTATS

Ce chapitre présente les résultats obtenus dans ce projet. Les séquences optimales d'assemblage du point de vue des 3 définitions exposées au chapitre 4 sont présentées.

#### **6.1. Séquence optimale en fonction de la direction d'assemblage**

Le produit utilisé pour mettre en évidence l'obtention d'une séquence optimale en fonction de la direction d'assemblage est l'isostatisme d'une pièce prismatique pour une opération d'usinage en fraisage surfacique. Les formes de ces composants ont été modifiées lors de la modélisation solide. La figure 6.1 montre le produit à l'état assemblé. La figure 6.2 montre une vue explosée de ce produit. La figure 6.3 montre une vue qui désigne les différents composants du produit. Voici la liste des composants. Les noms entre parathèse apparaîtront dans le modèle relationnel.

Support du montage	( sup )
Pièce à usiner	( usin )
Appui 1	( appui1 )
Appui 2	( appui2 )
Appui 3	( appui3 )
Appui 4	( appui4 )
Appui 5	( appui5 )
Appui 6	( appui6 )
Serrage	( serrage )
Support serrage	( supp1 )
Vis de serrage	( visser )

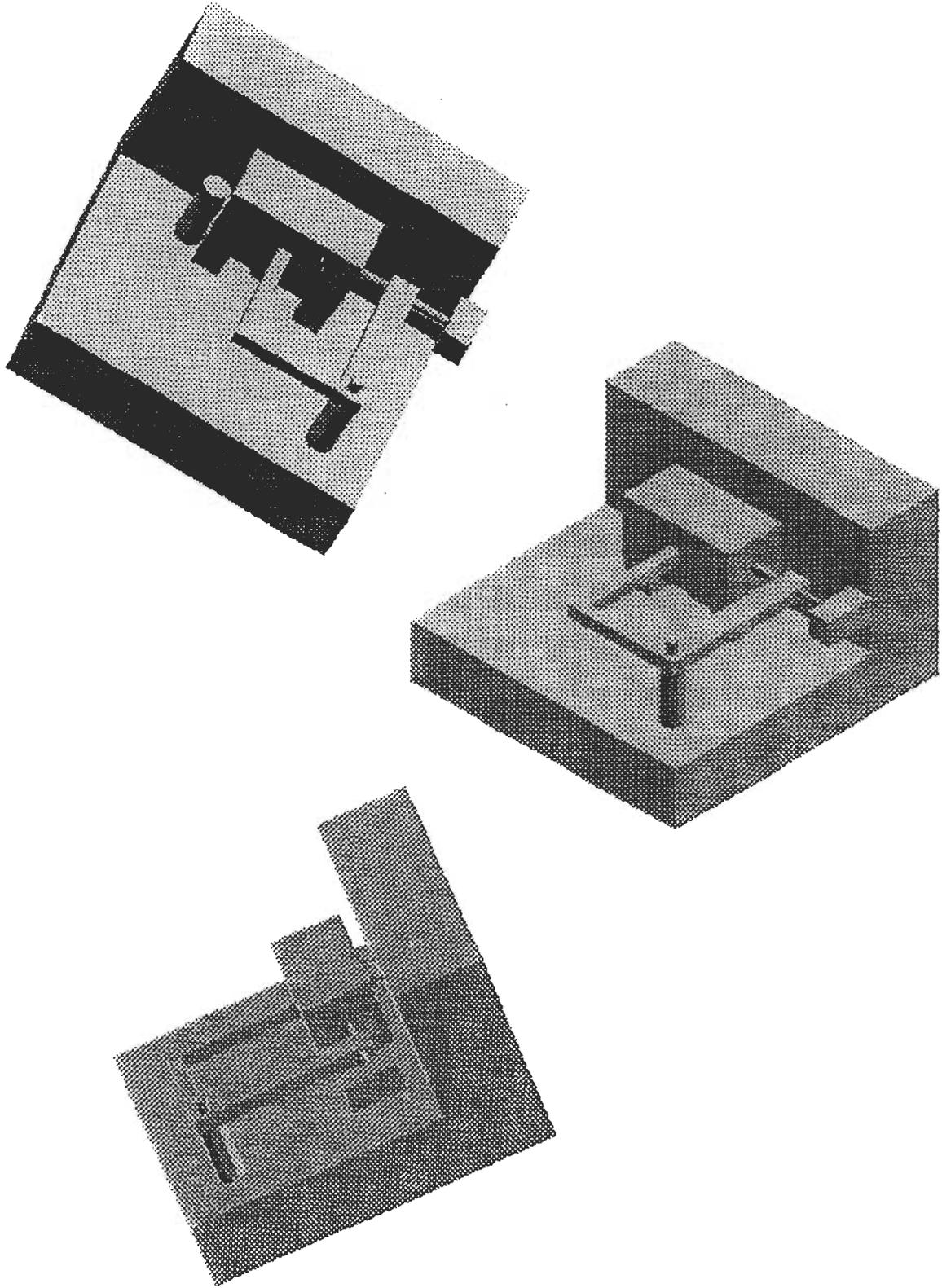


Figure 6.1: montage d'usinage

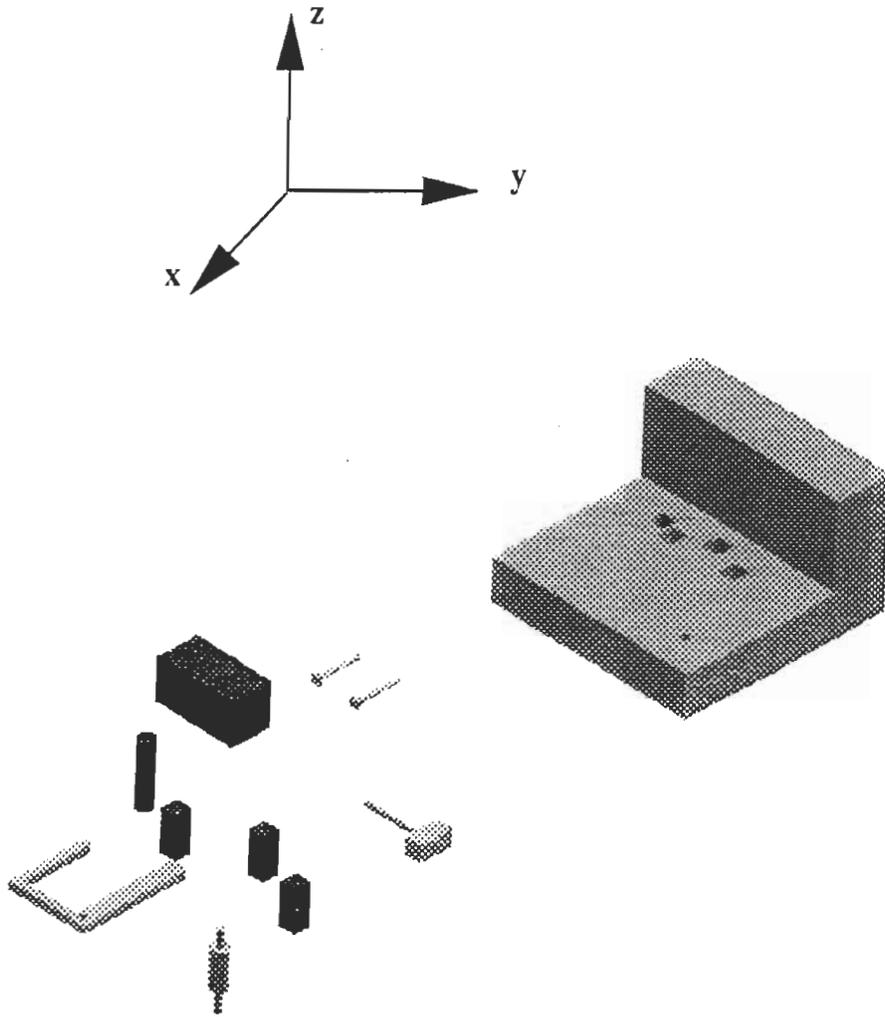


Figure 6.2: vue explosée du montage d'usinage

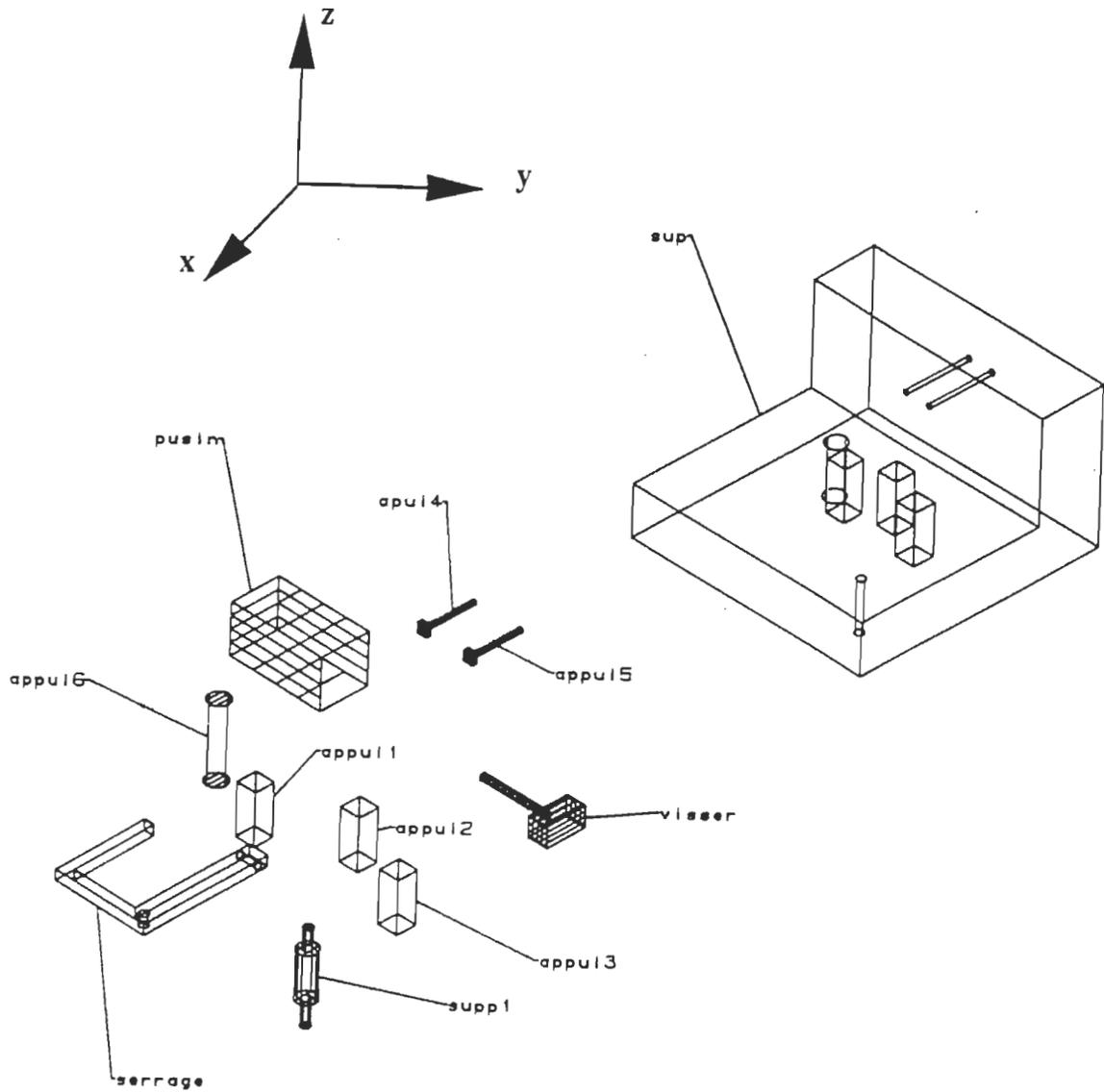


Figure 6.3: vue indiquant les composants du montage d'usinage

Le traitement de ce produit par l'engin GAPP, donne dans un premier temps le modèle relationnel indiqué à la figure 6.4.

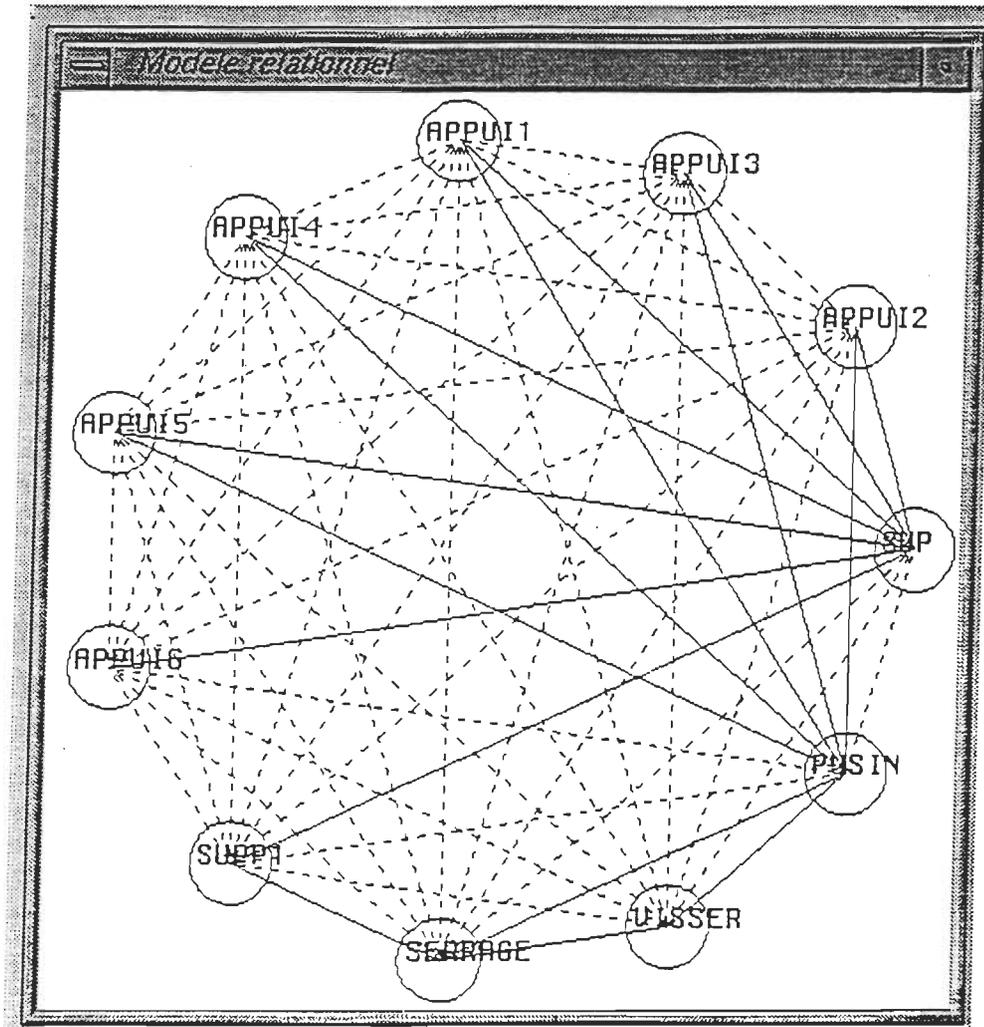


Figure 6.4: "screen dump" du modèle relationnel généré automatiquement à partir d'un dessin solide du montage d'usinage

Par la suite, une séquence optimale est générée. La figure 6.5 montre une séquence arbitraire du point de vue de la direction d'assemblage du montage d'usinage. Dans cette figure chaque ligne est une opération composée du type de contact, du sous-assemblage 1 (déplacé), du sous-assemblage 2 (fixe) et de la direction d'assemblage. Deux types de contacts sont mentionnés sur cette figure : insertion (fit) et contact (against). Les opérations sont numérotées. Le sens de la direction d'assemblage est indiqué par le signe + ou le signe -. Ce format de présentation sera identique pour les autres séquences. Dans le cas du regroupement basé sur la forme de saisie, la direction d'assemblage sera remplacée par cette forme de saisie et par la valeur de la fonction d'assemblage dans le cas des tolérances.

1	fit (APPUI1) (SUP) z-
2	fit (SERRAGE) (VISSER) y+
3	fit ( SERRAGE VISSER) (SUPP1) z-
4	fit (APPUI4) (APPUI1 SUP) x-
5	fit(APPUI2) (APPUI1 SUP APPUI4) z-
6	against (PUSIN) (SERRAGE VISSER SUPP1) y+
7	fit (APPUI3) (APPUI2 APPUI1 SUP APPUI4) z-
8	fit (APPUI5) (APPUI3 APPUI2 APPUI1 SUP APPUI4) x-
9	fit (APPUI6) (APPUI5 APPUI3 APPUI2 APPUI1 SUP APPUI4) z-
10	against (PUSIN SERRAGE VISSER SUPP1) (APPUI6 APPUI5 APPUI3 APPUI2 APPUI1 SUP APPUI4) z-

Figure 6.5: séquence arbitraire

Pour cette séquence arbitraire, le nombre de regroupements dans les directions x et y est égal à zéro tandis que celui de la direction z est égal à 1. Le nombre de regroupements total est donc égal à 1. La figure 6.6 montre une séquence optimale générée par GAPP. Pour cette séquence optimale le nombre de regroupements suivant la direction :

x = 0  
y = 1; les opérations 1 et 2.  
z = 3; les opérations 3, 4 et 5. Les opérations 7 et 8.  
total =4

Le nombre de regroupements total pour cette séquence est de 4 comparé à celui de la séquence arbitraire. Ces regroupements montrent donc que quelques opérations ont été faites dans la même direction.

```

1  against ( PUSIN ) ( VISSER ) from z+x+x-y-z-
2  fit ( SERRAGE ) ( PUSIN VISSER ) from y+
3  fit ( SUP ) ( SUPPI ) from z+
4  against ( APPUI1 ) ( PUSIN SERRAGE VISSER ) from z+x+x-y-
5  fit ( SUP SUPPI ) ( APPUI6 ) from z+
6  fit ( APPUI5 ) ( SUP APPUI6 SUPPI ) from x+
7  fit ( APPUI3 ) ( SUP APPUI5 APPUI6 SUPPI ) from z-z+
8  fit ( APPUI2 ) ( SUP APPUI3 APPUI5 APPUI6 SUPPI ) from z+
   RE-ORIENT 180 DEGREES
9  fit ( APPUI4 ) ( SUP APPUI2 APPUI3 APPUI5 APPUI6 SUPPI ) from x+
10 fit ( SUP APPUI2 APPUI3 APPUI4 APPUI5 APPUI6 SUPPI ) ( APPUI1 PUSIN SERR
    AGE VISSER ) from z+
   RE-ORIENT 180 DEGREES

```

Figure 6.6: séquence optimale basée sur le regroupement de la direction d'assemblage

## 6.2. Séquence optimale en fonction de la forme des composants impliqués dans les opérations d'assemblage

Le produit montage d'usinage sera aussi utilisé pour mettre en évidence le choix d'une séquence d'assemblage optimale basé sur la forme de saisie des composants. Le modèle relationnel de ce produit est le même qu'au paragraphe précédent. La figure 6.7 montre la séquence optimale obtenue après exécution de l'engin GAPP.

1	fit (APPUI1) (SUP) prismatic
2	fit (SERRAGE) (VISSER) prismatic
3	fit (SUPP1) ( SERRAGE VISSER) cylindrical
4	fit (APPUI4) (APPUI1 SUP) prismatic
5	fit(APPUI2) (APPUI1 SUP APPUI4) prismatic
6	against (PUSIN) (SERRAGE VISSER SUPP1) prismatic
7	fit (APPUI3) (APPUI2 APPUI1 SUP APPUI4) prismatic
8	fit (APPUI5) (APPUI3 APPUI2 APPUI1 SUP APPUI4) prismatic
9	fit (APPUI6) (APPUI5 APPUI3 APPUI2 APPUI1 SUP APPUI4) cylindrical
10	against (PUSIN SERRAGE VISSER SUPP1) (APPUI6 APPUI5 APPUI3 APPUI2 APPUI1 SUP APPUI4) prismatic

Figure 6.7: séquence non optimale

Dans cette figure quelques regroupements (5) sont possibles (opérations 1 et 2, 4, 5, 6, 7 et 8). En utilisant GAPP, on obtient une séquence plus optimale comme le montre la figure 6.8. Pour cette séquence optimale (figure 6.8), le nombre de regroupements est de 9 ( opérations 1 à 9). L'effet de ces 9 regroupements est qu'il faut un seul changement d'outil dans l'exécution de cette séquence d'assemblage ( opérations 9 et 10).

```

1  against ( PUSIN ) ( VISSER ) prismatic
2  fit ( SERRAGE ) ( PUSIN VISSER ) prismatic
3  against ( APPUI1 ) ( PUSIN SERRAGE VISSER ) prismatic
4  fit ( SUP ) ( SUPPI ) prismatic
5  fit ( SUP SUPPI ) ( APPUI6 ) prismatic
6  fit ( APPUI5 ) ( SUP APPUI6 SUPPI ) prismatic
7  fit ( APPUI4 ) ( SUP APPUI5 APPUI6 SUPPI ) prismatic
8  fit ( APPUI3 ) ( SUP APPUI4 APPUI5 APPUI6 SUPPI ) prismatic
9  fit ( APPUI2 ) ( SUP APPUI3 APPUI4 APPUI5 APPUI6 SUPPI ) prismatic
   RE-ORIENT 180 DEGREES
10 fit ( SUP APPUI2 APPUI3 APPUI4 APPUI5 APPUI6 SUPPI ) ( APPUI1
    PUSIN SERRAGE VISSER ) cylindrical
    RE-ORIENT 180 DEGREES

```

Figure 6.8: séquence optimale basée sur la forme de saisie

### 6.3. Séquence optimale en fonction des tolérances

Le produit utilisé est la lampe de poche. Cette lampe comporte 9 composants. La figure 6.9 met en évidence la vue explosée de ce produit.



Figure 6.9: vue explosée de la lampe de poche

voici la liste des composants de ce produit.

Couvercle	( cap )
Ressort	( spring )
Pile 2	( baterry 2)
Pile 1	( baterry 1)
Corps	( body )
Réflecteur	( Ref )
Ampoule	( bulb )
Protecteur	( lens )
Tête	( head )

La figure 6.10 montre une vue indentifiant les différents composants par leur nom.

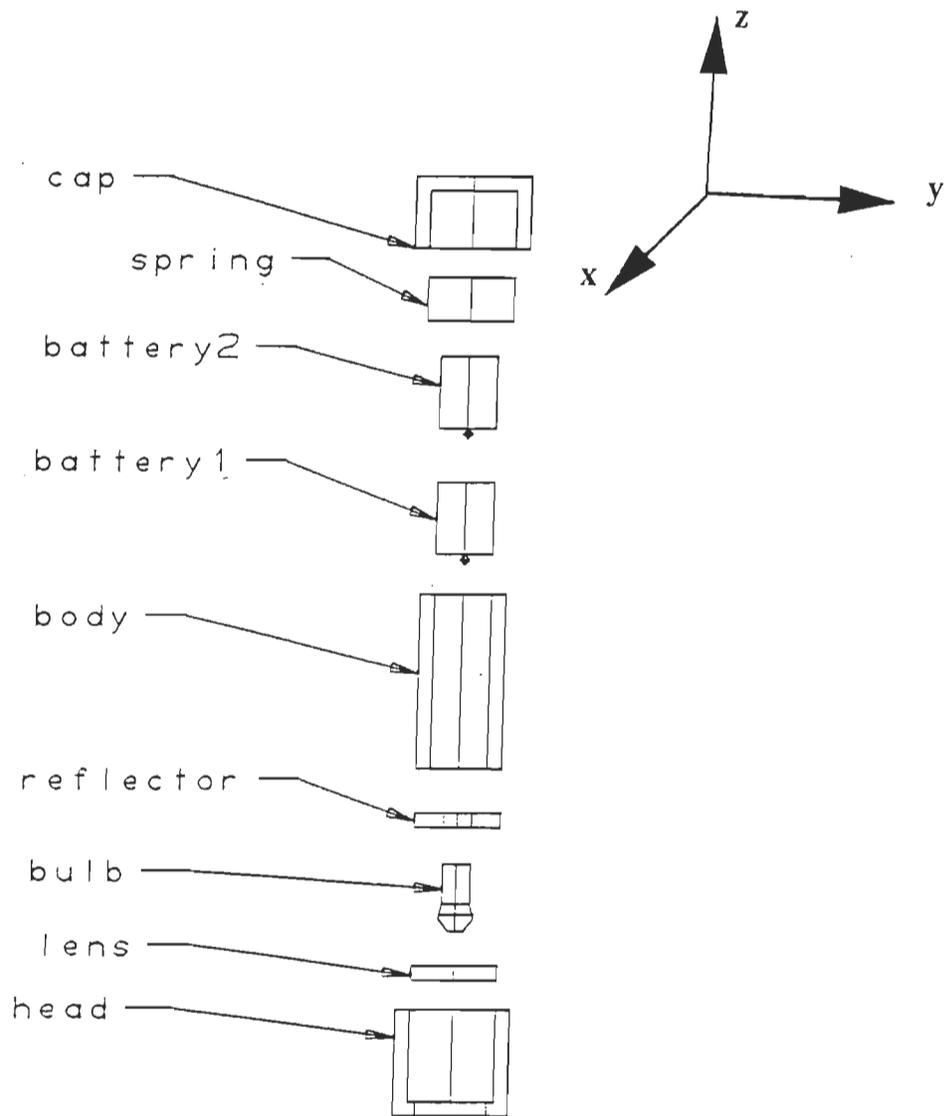


Figure 6.10: vue identifiant les composants

L'exécution de l'engin GAPP a permis d'obtenir le séquence d'assemblage optimale du point de vue des tolérances. La figure 6.11 montre le modèle relationnel de ce produit.

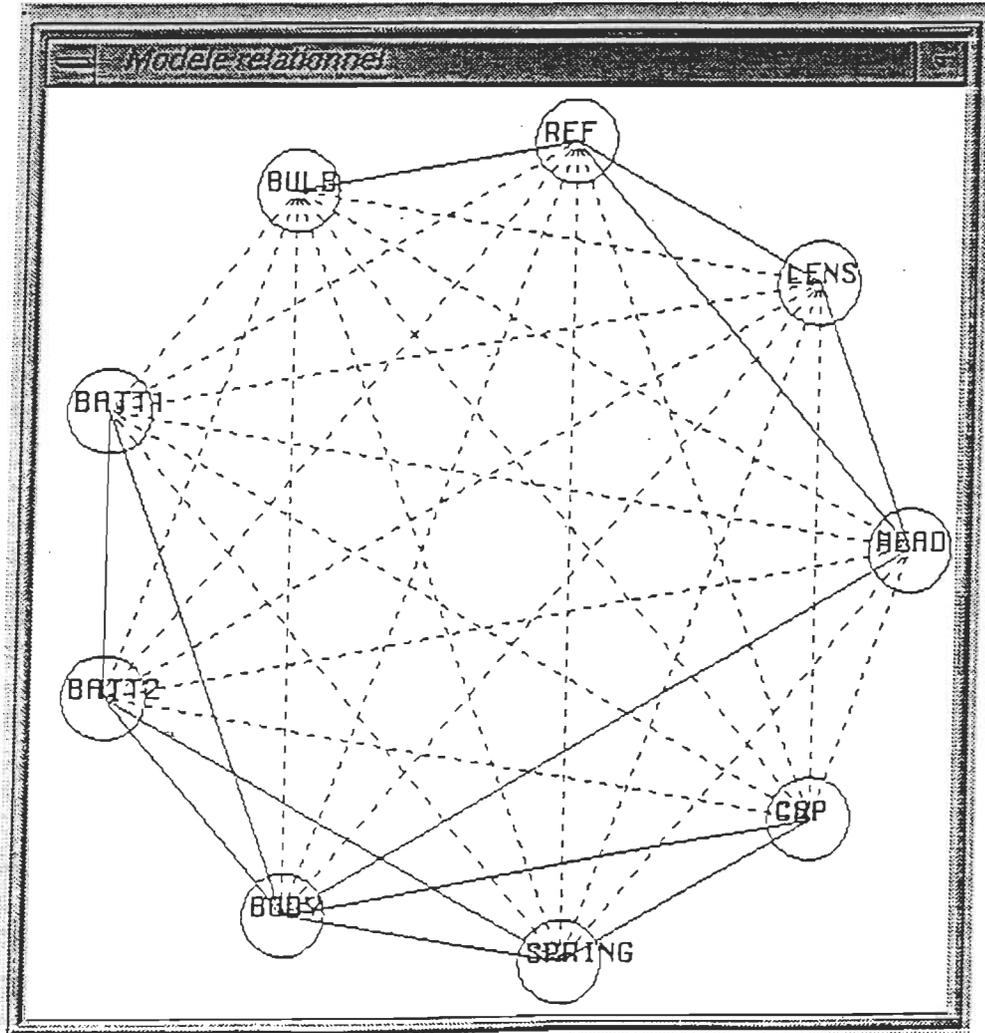


Figure 6.11: "screen dump" du modèle relationnel généré automatiquement à partir d'un dessin solide de la lampe de poche

Pour cet exemple, les informations nécessaires au calcul des fonctions d'assemblage (équation 4.1 ) des opérations sont présentées à la figure 6.12. Ces informations ont été obtenues d'une manière subjective.

Relations	tl	aj	tp	to	tou	tr
CAP SPRING	0.2	0	0.1	0	0	0.25
BATT2 SPRING	0.2	0	0.1	0	0	0.25
BATT2 CAP	0.2	0	0.1	0	0	0.25
BATT2 BODY	0.2	0	0.1	0	0	0.75
BATT1 BODY	0.2	0	0.1	0	0	0.75
BATT1 BULB	0.2	0	0.1	0	0	0.25
BODY HEAD	0.2	0	0.1	0	0	0.75
BODY CAP	0.2	0	0.1	0	0	0.75
HEAD REF	0.2	0	0.1	0	0	0.75
HEAD LENS	0.2	0	0.36	0	0	0.75
REF BULB	0.2	0	0.1	0	0	0.75

Figure 6.12: tolérances sur les relations

L'assemblage se faisant dans une seule direction, il n'est pas nécessaire de présenter une séquence quelconque. Dans cet exemple, 2 opérations consécutives ont la même fonction d'assemblage, si la différence de la valeur de ces 2 fonctions d'assemblage est inférieure ou égale à 0.5. La valeur de 0.5 est un choix. Cette différence peut prendre toutes les valeurs possibles. Cette différence doit être inscrite dans une des procédures de GAPP. La séquence optimale générée pour ce produit est présentée à la figure 6.13. Le nombre de regroupements est de 3 ( opérations 4, 5 et 6, 7 et 8). Ce regroupement suivant cette définition indique que dans la séquence optimale, 3 opérations ont le même degré de difficulté d'exécution moindre par rapport aux autres séquences possibles.

```
1 fit ( BODY ) ( SPRING ) 0.952381  
  RE-ORIENT 180 DEGREES  
2 fit ( CAP ) ( BODY SPRING ) 1.818182  
3 fit ( REF ) ( BULB ) 0.952381  
  RE-ORIENT 180 DEGREES  
4 fit ( BATT2 ) ( BODY CAP SPRING ) 0.454545  
5 fit ( BATT1 ) ( BODY BATT2 CAP SPRING ) 0.465116  
6 fit ( LENS ) ( HEAD ) 0.952381  
  RE-ORIENT 180 DEGREES  
7 against ( REF BULB ) ( BODY BATT1 BATT2 CAP SPRING ) 0.126582  
8 fit ( LENS HEAD ) ( REF BULB BATT1 BODY BATT2 CAP SPRING ) 0.082372  
  RE-ORIENT 180 DEGREES
```

Figure 6.13: séquence optimale du point de vue des tolérances

Les différentes séquences obtenues dans ce chapitre, mettent en évidence l'objectif défini au départ de cette étude qui était de trouver un moyen optimal de sélectionner une séquence parmi plusieurs autres séquences éligibles à partir de critères d'optimalité.

## CHAPITRE 7

### CONCLUSION ET DISCUSSION

#### 7.1. Conclusion

A travers ce rapport, nous avons constaté que le domaine de l'assemblage a fait l'objet de beaucoup de recherche. Grâce aux travaux de quelques auteurs pionniers dans le domaine de l'assemblage tel que Bourjault, Whitney et Homen de Mello etc., un énorme progrès a été fait dans ce domaine tant au niveau de la modélisation des composants, de la méthode de génération des séquences, de la représentation des séquences que sur l'évaluation et l'élimination des séquences d'assemblage.

Néanmoins, il reste beaucoup à faire pour proposer une séquence d'assemblage optimale parmi tant d'autres. C'est pourquoi nous avons jugé bon d'apporter notre contribution en orientant notre recherche vers la proposition d'une séquence optimale d'assemblage du point de vue de certains critères d'évaluation.

En effet, dans un premier temps, nous avons trouvé une méthode pour détecter les contacts entre les composants. Cette méthode consiste à comparer les surfaces des composants entre elles. Ces surfaces ont été obtenues par la modélisation des composants et a aboutit à l'obtention d'un fichier B-Rep. Ces fichiers obtenus, ont été mis dans un seul fichier. Cette manière de procéder a permis d'obtenir le nom des composants. Ces noms sont utiles lors de l'affichage du graphe des liaisons.

Cette détection permet de déterminer le type de contact. Un contact peut être du type contact physique, du type blocage ou libre. La modification de la matrice des demi-degrés de liberté en fonction du type de contact, permet de définir l'état de ces contacts.

L'intersection de ces matrices permet à l'engin qui génère les séquences d'assemblage de trouver une direction d'assemblage suivant laquelle une opération de désassemblage est possible.

L'obtention des contacts entre les composants permet aussi de générer automatiquement le modèle relationnel du produit. L'obtention automatique du modèle relationnel permet ainsi d'éliminer l'étape préliminaire à toute génération de séquences d'assemblage qui consiste à produire manuellement le graphe des liaisons des composants. Cette génération automatique des graphes donne de bons résultats car l'examen de ce graphe permet de visualiser les 3 types de relations ( contact, blocage et libre).

Dans un deuxième temps, pour choisir une séquence optimale parmi plusieurs autres, des définitions ont été établies ainsi que les paramètres permettant de les évaluer.

Ces définitions sont au nombre de 3:

- la direction d'assemblage évaluée par la matrice des demi-degrés de liberté
- la forme de saisie des composants évaluée par la forme du composant le plus éloigné impliqué dans le sous-assemblage
- les tolérances évaluées par la fonction d'assemblage

Disposant de ces définitions, l'engin générant les séquences a été utilisé. A chaque fois que l'engin génère une opération (enfant) d'assemblage, cette opération est évaluée en fonction des paramètres établis. Donc si l'opération qui vient d'être générée a les mêmes paramètres (direction, forme, tolérances) que l'opération (parent), les valeurs des compteurs de cette opération enfant relatifs aux paramètres sont augmentées de 1. Dans le cas contraire l'opération enfant hérite des valeurs des compteurs de l'opération parent. A la fin du processus de génération des séquences, la séquence optimale sera celle qui a la plus grande valeur des paramètres (direction, forme, tolérances).

Cette méthode est consistée à évaluer et à regrouper, en fonction des 3 définitions, au fur et à mesure que les opérations sont générées par GAPP.

La transformation des paramètres des opérations dans la séquence optimale en terme de coût conduit donc à un gain énorme en temps, argent et au niveau de la facilité d'exécution des opérations. Les résultats fournis par GAPP en prenant compte des définitions développées dans ce projet sont satisfaisants car un choix facile est effectué parmi plusieurs opérations.

## 7.2. Discussion

Le logiciel utilisé pour l'expansion des séquences d'assemblage utilisait un mot de 32 bits pour effectuer les combinaisons des sommets des graphes des liaisons. Si un produit a plus de 32 relations, GAPP fournissait des résultats non désirés. Pour pallier à ce problème, une matrice a été utilisée pour déterminer le nombre de combinaisons possibles lors de la rupture des liaisons d'un modèle relationnel.

Ce projet faisant partie d'un projet global (montages d'usinage modulaires) réalisé conjointement par l'U.Q.T.R et l'E.T.S, les résultats seront donc appliqués à la détermination de la séquence d'assemblage optimale des modules. La seule condition pour que les résultats de ce projet soient applicables est la transformation des caractéristiques géométriques des modules en fichier B\_Rep.

Pour bien faire fonctionner GAPP version graphique, une méthodologie doit être suivie.

- modéliser les composants sur un modeleur capable de produire le fichier B\_Rep. Il faut donc modéliser individuellement tous les composants du produit dans un fichier séparé. Faire attention à la disposition spatiale des composants dans le système d'axes pour éviter une détection des contacts erronés.

- reporter tous les noms des composants obtenus dans un seul et unique fichier selon l'ordre désiré. Le nom de ce fichier unique doit être inscrit dans la procédure Make\_B\_Rep\_File () à l'endroit où un pointeur de fichier est ouvert pour la lecture.

- compiler GAPP

- lors de l'exécution de GAPP version graphique, un choix doit être effectué avant que GAPP ne commence le processus de génération des séquences. Ce choix concerne les 3 définitions ( la direction d'assemblage, la forme de saisie des composants et les tolérances ). Après avoir indiqué son choix, GAPP exécute et produit une séquence optimale respectivement au critère choisi.

Dans ce projet, le travail le plus important a été la génération des relations et l'état de ces relations en terme de degrés de liberté par rapport aux autres composants. Le fichier B\_Rep est une représentation frontalière non unique. A cause de cette non unicité, les conditions établies peuvent dans certaines positions relatives de 2 composants donner des résultats non désirés. Il faut mentionner qu'il existe plusieurs positions relatives possibles entre 2 composants en situation de contact ou non.

La manière de déterminer la forme de saisie (définition 2) dans ce projet n'est pas totalement optimale. Une amélioration possible serait de tenir compte des critères tels que

pour le faire est le "grasping" . Donc un module sera développé et déterminera automatiquement d'une manière optimale la forme de saisie des composants.

Pour encore perfectionner ce logiciel en cours de développement, la stabilité doit être considérée. Les séquences générées par GAPP à ce stage de l'avancement du processus de génération des séquences, ne sont pas évaluées du point de vue de la stabilité. Cet aspect de la stabilité des sous-assemblages est en cours de développement.

La prochaine étape à ce projet consiste à développer un module de rétroaction (feedback). Ce module permettra de modifier le design du produit en fonction du coût élevé de certaines opérations dans la séquence optimale d'assemblage.

Ainsi GAPP pourra supporter la planification simultanée (ingénierie simultanée) de la génération des séquences. Il sera donc possible de modifier à tout moment la séquence optimale d'assemblage si des problèmes techniques pratiques arrivaient en cours de réalisation de la séquence optimale et qui augmenteraient les coûts des opérations à l'intérieur de la séquence optimale.

## CHAPITRE 8

### BIBLIOGRAPHIE

- [1] Abe Norihiro, Ohno Katsumi, Kitahashi Tadahiro, he Shoujie (1993), "Task Specification Using technical Illustrations", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 58-62.
- [2] Baldwin Daniel F., Abell Thomas E., Chang Maxi Lui Man, De Fazio Thomas L.,and Whitney Daniel E. (1991), "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products", *IEEEInternational Conference on Robotics and Automation*, Vol.7, No.1, pp. 78-94.
- [3] Bethume, J.(1989), "Modern Drafting", *Prentice-Hall inc*, New Jersey, p. 415.
- [4] Boneschanscher Nico, Drift Hans van der, Buckley Stephen J and Taylor Russell (1992) "Subassembly Stability", *Robotics*, pp. 780-785.
- [5] Barnett J.A (1990), "A System Engineering Approach to Automated Assembly Planning", *IEEE International Conference on Robotics and Automation*, pp. 387-389.
- [6] Bonneville François, Henrioud J.M and Bourjault Alain (1991), "Etude d'une contribution de la C.A.O à l'élaboration automatique des gammes d'assemblage", *Proceedings of the 10th international conference on CAD/CAM, Computer Graphics and Computer Aided Technologies*, p 9.
- [7] Bourjault A, and Chappe D (1991), "A Two-Stage Assembly Process Planning Tool for Robot-Based Flexible Assembly Systems", *International Journal Prod. Res.*, Vol. 29, No.2, pp. 247-266.
- [8] Bourjault A, Chappe D.and Henrioud J.M (1987), "Elaboration automatique des gammes d'assemblage à l'aide de réseaux de Petri", *APII.*, Vol. 21, No.4, pp. 323-342.

- [9] Bourjault A and Henrioud J.M (1987), "Détermination des sous-assemblages d'un produit à partir des séquences temporelles d'assemblage", *APII.*, Vol. 21, No.2, pp. 117-127.
- [10] Browne J., Tierney K., and Walsh M. (1991), "A Two-Stage Assembly Process Planning Tool for Robot-Based Flexible Assembly Systems", *International Journal Prod. Res.*, Vol. 29, No.2, pp. 247-266.
- [11] Campagne Jean-Pierre (1989), "Elaboration automatique de gammes d'assemblage", *Automatique-productique Informatique Industrielle (APII)*, Vol 23, No. 1, pp. 53-68.
- [12] Chen Chun Lung (1990), "A Graph Partitioning Approach for Robot Assembly Sequences Generation", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 1, pp. 71-76.
- [13] Chen C. L. Philip (1989), "Precedence knowledge Acquisition for generating Robot Assembly Sequences", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 2, pp.376-389.
- [14] Chen C. L. Philip (1990), "Planning Optimal Precedence Constraint Robot Assembly Sequences Problem With neural Computation", *Application of Artificial Intelligence VIII*, Vol. 1293, pp. 314-320.
- [15] Chen C. L. Philip (1991), "Automatic Assembly Sequences Generation by Pattern Matching", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 2, pp.376-389.
- [16] Dakin G., Liu Y., Popplestone R.J and Weiss R (1989), "Symmetry Inference in Planning Assembly", *IEEE International Conference on Robotics and Automation*, pp. 1865-1867.
- [17] De Fazio Thomas L, Abell E. Thomas, Amblard Guillaume P., and Whitney Daniel E. (1990), "Computer Aided Assembly Sequence Editing and Choice: Editing

Criteria, Bases, Rules, and Technique", *IEEE International Conference on Robotics and Automation*, pp. 416-421.

- [18] De Fazio Thomas L, Edsall A. C., Gustavson J. A., Hernandez P.M, Hutchins P.M, Leung H.W, Luby S. Metzinger R.W, Nevins J.L, Tung K.K and Whitney Daniel E, "A Prototype of Feature-Based Design for Assembly", 1990 ASME Design Technical Conference, Chicago, USA, pp. 9-16.
- [19] De Fazio, Thomas L, and Whitney Daniel E. (1987), "Simplified Generation of All Mechanical Assembly Sequences", *IEEE Journal of Robotics and Automation*, Vol, Ra-3, No. 6, pp. 640-657.
- [20] Delchambre A. (1990), "A pragmatic Approach to Computer Aided Assembly Planning", *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, , pp.1600-1605.
- [21] Delchambre Alain and Wafflard Alain (1991), "An Automatic, Systematic and User-Friendly Computer-Aided Planner for Robotized Assembly", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 592
- [22] Delchambre Alain and Gaspart Pierre (1992), "KBAP: An Industrial Prototype of Knowledge-Based Assembly Planner", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2404-2411.
- [23] Dini G. and Santochi M. (), "KBAP: Automated Sequencing and subassembly Detection in Assembly Planning, pp. 1-17.
- [24] Gasmi B., Castel C., Reboulet C., and Houdebert R. (1990), "SAFIR: Flexible Multi-Robots Assembly System", *IEEE International Conference on Robotics and Automation*, PP.582-587.
- [25] Giraud A. and Sidore D. (1992), "A Heuristic Motion Planner Using Contact for Assembly", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2165-2170.

- [26] Giusti F. and Santochi M. (1991), "Robotized Assembly of Modular Fixtures", *Annals of the CIRP*, Vol. 40, No.1, pp. 17-20.
- [27] Ghosh Kalyan and Hoummady Abdellah (1989), "Generation and Evaluation of Assembly Sequences in Computer-Automated Process Planning", *International Journal of Vehicle Design*, pp. 341-356.
- [28] Haynes Leonard S. and Morris Graham H. (1988), "A Formal Approach Specifying Assembly Operations", *International Journal Machine Tools Manufacturing*, Vol. 28, No.3, pp. 281-298.
- [29] Heemskerk J.M (1989), "The use of Heuristics in Assembly sequence Planning", *Annals of CIRP*, Vol. 38, No.1, pp. 37-40.
- [30] Heemskerk C.J.M and Reijers L.N (1990), "A concept for Computer-Aided Process Planning of Flexible Assembly", *Annals of CIRP*, Vol. 39, No.1, pp. 25-28.
- [31] Henrioud J. M. (1989), "Contribution à la conceptualisation de l'assemblage automatisé: Nouvelle approche en vue de la détermination des processus d'assemblage", *Doctorat d'État, Université de Besançon Franch-Comté, France*, p. 295.
- [32] Henrioud J. M. and Bourjault A. (1992), "Computer Aided Assembly Process Planning", *Journal of Engineering Manufacture*, Vol. 206, pp. 61-66.
- [33] Henrioud J.M and Bourjault H.(1990), "Détermination des arbres d'assemblages", *APII*, Vol.24, pp.564-563.
- [34] Hermandez J. A., Luby S. C., Hutchins P. M., Leung H. W., Gustavson R. E., De Fazio T. L., Whitney D. E., and Nevins J. L.(1991), "An Integrated System for Concurrent Design Engineering", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 205-210.

- [35] Hervé J.M. (1992), "Analyse Structurelle des Mécanismes par Groupe des Déplacements", *Proceedings of 1992 Mechanism and Machine Theory*, Vol. 13, pp. 437-450.
- [36] Hoffman Richard A. (1990), "Assembly Planning for B\_Rep Objects", *IEEE International Conference on Robotics and Automation*, pp. 314-320.
- [37] Homem-de-Mello L. S. (1990), "Two criteria for selection of assembly plans: maximizing the flexibility of sequencing the assembly tasks, and minimizing the assembly time through parallel execution of assembly tasks", *IEEE Transactions on Robotics and Automation*, pp. 1-18.
- [38] Homem-de-Mello L. S. and Desai R. S. (1990), "Assembly Planning for Large Truss Structures in Space", *IEEE International Conference on Robotics and Automation*, pp. 404-407.
- [39] Homem de mello L. S (1990), "AND/OR Graph representation of Assembly Plans", *IEEE International Conference on Robotics and Automation*, Vol.6, No.2, pp. 188-199.
- [40] Homem-de-Mello L. S. (1992), "Multi-Hierarchical representation of tetrahedral truss structures for Assembly Sequence Planning", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2397-2403.
- [41] Homem-de-Mello L. S. and Sanderson A. C. (1990), "Evaluation and Selection of Assembly Plans", *IEEE International Conference on Robotics and Automation*, pp. 1588-1593.
- [42] Homem-de-Mello L. S. and Sanderson Arthur C.(1991), "Representations of Mechanical Assembly sequences", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 211-226.
- [43] Homem-de-Mello L. S. and Sanderson Arthur C. (1991), "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences", *IEEE*

*International Conference on Robotics and Automation*, Vol. 7, No. 2, pp. 228-240.

- [44] Homem-de-Mello L. S. and Sanderson Arthur C. (1988), "Automatic Generation of Mechanical Assembly Sequences", Report n. CMU-RI-TR-88-19, The robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, p 43.
- [45] Homem-de-Mello L. S. and Sukhan Lee (1991), "LEGA: a computer-aided generator of assembly plans", *Computer-Aided Mechanical Assembly Planning*, Kluwer Academic Publishers, chapter 8.
- [46] Hoummady Abdellah and Ghosh Kalyan . (1989), "Generation and Evaluation of Assembly Sequences in Computer-Automated Process Planning", *International Journal of Computer Applications in Technology*, Vol. 2, No. 3, pp. 151-158.
- [47] Hoummady Abdellah and Ghosh Kalyan . (1990), "The Problem of Simultaneous Actions During Assembly", *International Journal of Computer Applications in Technology*, Vol. 33, No. 2, pp. 119-123.
- [48] Hsu Wynne, Lee C. S. G, and Su S. F. (1992), "Feedback Evaluation of Assembly Plans", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2419-2424.
- [49] Huang Kai-I(1993), "Development of an Assembly Planner Using decomposition Approach", *IEEE International Conference on Robotics and Automation*, Vol 2, pp. 63-68.
- [50] Huang Y.F and Lee C.S.G (1989), "Precedence Knowledge in Feature Mating Operation Assembly Planning", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 216-221.
- [51] Huang Y.F and Lee C.S.G (1990), "An Automatic Assembly Planning System", *IEEE International Conference on Robotics and Automation*, pp. 1594-1599.

- [52] Huang Y.F and Lee C. S. G. (1991), "A Framework of Knowledge-Based Assembly Planning", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 599-604.
  
- [53] Irizarry-Gaskins Vilma M. and Chang Tien-Chien (1990), "Knowledge Based Approach for Automatic Process Plan Generation in an Electronic Assembly Environment", *International Journal Prod. Res.*, Vol. 28, No. 9, pp. 1673-1693.
  
- [54] Jiang X.Y and Bunke H.(1992), "A simple and Efficient Algorithm for Determining the Symmetries of Polyhedra", *Graphical Models and Image Processing*, Vol. 54, No. 1, pp. 91-95.
  
- [55] Khosla Pradeep K. and Mattikali Raju(1989), "Determining the Assembly Sequence from A 3-D Model", *Journal of Mechanical Working Technology*, Vol. 20, pp. 153-162.
  
- [56] Krishnan S. S. and Sanderson Arthur C.(1990), "Reasoning About Geometric Constraints for Assembly Sequence Planning", *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, pp. 776-781.
  
- [57] Kroll Ehud.(1991), "Spatial Reasoning in Assembly Planning", *Proceedings of 1991 NSF Design and Manufacturing Systems Conference*, pp. 1089-1097.
  
- [58] Laperrière L. and ELMaraghy H. A. (1992), "Planning of Products Assembly and Disassembly", *Annals of CIRP*, Vol. 41, p.3.
  
- [59] Laperrière L.(1992), "Generative Assembly Process Planning", *Thèse de Doctorat*, McMaster University, Hamilton, Ontario, Canada.
  
- [60] Laperrière L. and ELMaraghy H. A. (1993), "Generation and Evaluation of Assembly Sequences in Concurrent Engineering", *Manufacturing Systems*, Vol. 22, No.2, pp. 123-131.

- [61] Lee Sukhan and Shin Gil Yeong (1990), "A Cooperative Planning System for Flexible Assembly", *IEEE International Conference on Robotics and Automation*, pp. 306-313.
- [62] Lee Sukhan (1992), "Backward Assembly Planning With Assembly Cost Analysis", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2382-2391.
- [63] Lee Sukhan and Shin yeong Gil (1990), "Assembly Planning Based on Gemetric Reasoning", *Computer & Graphics*, Vol. 14, No.2, pp. 237-250.
- [64] Lee Sukhan and Shin Yeong Gil (1990), "Assembly Planning Based On Subassembly Extraction", *IEEE International Conference on Robotics and Automation*, pp. 1606-1611.
- [65] Lee Sukhan and Wang Fu-Chung (1993), "A Physical Reasoning of Interconnection Forces for Efficient Assembly Planning", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol 2, pp. 307-313.
- [66] Lin Alan C. and Chang Tien-Chien (1989), "A Framework for Automated Mechanical Assembly Planning", *Journal of mechanical working technology*, Vol 20, pp. 237-248.
- [67] Lin Alan C. and Chang Tien-Chien (1993), "3D MAPS: Three-Dimensional Mechanical Assembly Planning System", *Journal of manufaturing System*, Vol 12, No. 6, pp. 437-456.
- [68] Liu Yanxi and Popplestone Robin (1991), "Symmetry groups in Analysis of Assembly kinematics", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 572-576.
- [69] Liu Yanxi and Popplestone Robin (1992), "From Charateristic Invariants to stiffness Matrices", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2375-2380.

- [70] Lozano-Perez, Tomas and Wilson, Randall H (1993), "Assembly Sequencing for Arbitrary Motions", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 527-532.
- [71] Maranzana Roland, Mascle Christian and Drolet Jocelyn (1992), "An Automatic Modular Fixturing Methodology for Machining Prismatic parts", *8th International Conference on CAD/CAM, Robotics and Factories of the Future*, Metz, France, pp. 1-11.
- [72] Mascle Christian and Figour J. (1990), "Methodological Approach of Sequences Determination Using the Disassembly Method", *IEEE International Conference on Robotics and Automation*, pp. 483-490.
- [73] Mascle C. (1990), "Approche méthodologique de détermination de gammes par le désassemblage", *Thèse de Doctorat*.
- [74] Mascle C (1990), "Subassemblies and Common Fitting Nodes in Determination of Assembly Sequences", *8th International Conference on CAD/CAM, Robotics and Factories of the Future*, Metz, France, pp. 12-20.
- [75] Mathur Rajive K. and Sanderson Arthur C. (1990), "A Hierarchical Planner for Space Truss Assembly", *Cooperative Intelligent Robotics in Space*, Vol. 1387, pp. 47-57.
- [76] Mattikalli Raju S. and Khosla Pradeep K. (1991), "Analysis of Restraints to Translational and Rotational Motion from the Geometry of Contact", *Design Manufacture/Integration ASME*, Vol. 39, pp. 65-70.
- [77] Mattikalli Raju S. and Khosla Pradeep K. (1992), "Motion Constraints from Contact Geometry: Representation and Analysis", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2178-2185.
- [78] Mattikalli Raju S, Khosla Pradeep K and Xu Yangsheng (), "Subassembly Identification and Motion Generation for Assembly: A geometric Approach",

*Proceedings of IEEE International Conference on Robotics and Automation*, pp. 399-402.

- [79] Ogata H. and Takahashi T.(1993), "A Geometric Approach to Task Understanding for Robotic Assembly Operations", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.1, pp. 58-64.
- [80] Okano, Akira (1993), "Computer-Aided Assembly Process Planning with Resource Assignment", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 301-306.
- [81] Park Jong Hun, Chung Myung Jin, Lim kye Young (1993), "Acquisition of Assembly Constraints Without Redundancies", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 69-74.
- [82] Popplestone Robin J., Liu Yanxxi and Weiss Rich (1990), "A Group Theoretic Approach to Assembly Planning", *AI magazine*, Vol.11, No. 1, pp. 82-97.
- [83] Pu Pearl (1992), "An Assembly Sequence Generation Algorithm Using Case-Based Search Techniques", pp. 55-63.
- [84] Rudiger Sack Jorg and Toussaint Godfried T.(1992), "Separability of pairs of polygons through single translations", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2425-2430.
- [85] Sanderson Arthur C., Homem de Mello L. S, and Zhang Hui (1990), "Assembly Sequence Planning", *IEEE International Conference on Robotics and Automation*, Vol.11, No. 1, pp. 62-81.
- [86] Santochi M. and Dini G.(1992), "Computer-Aided Planning of Assembly Operations: The Selection of Assembly Sequences", *Robotics and Computer-Integrated Manufacturing* , Vol.9, No. 6, pp. 439-446.

- [87] Shpitalni M., Elber G. and Lenz E. (1989), "Automatic Assembly of Three-Dimensional Structures via Connectivity Graphs", *Annals of CIRP*, Vol.38, No. 1, pp. 25-28.
- [88] Srikanth S. and Turner Joshua U.(1990), "Toward a Unified Representation of Mechanical Assemblies", *Engineering with Computers*, Vol. 6, pp. 103-112.
- [89] Su S.F and Lee C.S.G (1993), "Automatic Generation of Goal Regions for Assembly Tasks in the Presence of Uncertainty", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.1, pp. 77-82.
- [90] Sukhan, Lee and Wang, Fu-Chung (1993), "Physical Reasoning of Interconnection Forces for Efficient assembly Planning", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 307-313.
- [91] Sukhan, Lee and Chunsik Yi (1993), "Subassembly Stability and Reorientation", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 521-526.
- [92] Thomas, J.P. (1993), "Constructing Assembly Plans", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 515-520.
- [93] Thomas, J.P. and Baker, K D (1993), "Modelling of Assembly Partitions for a Distributed Environment", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 52-57.
- [94] Tsao, Jungfu and Wolter, Jan(1993), "Assembly Planning with Intermediate States", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 71-76.
- [95] Suzuki Tatsuya, Kanehara takahide, Inaba Akio, Okuma, Shigeru(1993), "On Algebraic and Graph Structural Properties of Assembly Petri Net", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.2, pp. 507-514.

- [96] Su, S F and Lee C. S. G (1993), "Automatic generation of Goal Regions for Assembly Tasks in the Presence of Uncertainty", *IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol.1, pp. 77-82.
- [97] Waarts J. J., Boneschanscher Nico, and Bronsvort Willem F (1992), "A Semi-Automatic Assembly Sequence Planner", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2431-2434.
- [98] Weule H. and Friedman Th., (1989), "Computer-Aided Product Analysis in Assembly Planning", *Annals of CIRP*, Vol. 38, No. 1, pp 1-4.
- [99] Whitney D.E., De Fazio T., Gustavson R.E., Graves S.C., Coopriker K., Holmes C.A., Klein C.J., Lui M. and Pappu S. (1988), "Computer-Aided Design of Flexible Assembly Systems", *IEEE International Conference on Robotics and Automation*, pp 100.
- [100] Wiendahl Hans-Peter and Zeugträger Karsten (1991), "Modeling and Simulation of Assembly Systems", *Annals of the CIRP*, Vol. 40, No.2, pp. 577-585.
- [101] Wilson Randall H. and Rit Jean-François (1990), "Maintaining Geometric Dependencies in an Assembly Planner", *IEEE International Conference on Robotics and Automation*, pp.890-895.
- [102] Wilson Randall H. and Schweikard Achim (1992), "Assembly Polyhedra With Single Translations", *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2392-2397.
- [103] Wolter Jan (1989), "On the Automatic generation of Assembly Plans", *Proceedings of 1989 IEEE International Conference on Robotics and Automation*, pp. 62-68.
- [104] Wolter Jan (1991), "A Combinatorial Analysis of Enumerative Data Structures for Assembly Planning", *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, pp. 611-618.

## CHAPITRE 9

### ANNEXES

#### 9.1. Fichier B-Rep

Un fichier B-Rep est un fichier contenant la représentation géométrique des frontières (boundary representation) d'un solide. Cette représentation géométrique des frontières est décrite par des entités géométriques : body, lump, shell, face, loop, coedge, edge, vertex, transform. Dans ce fichier, les champs contenus dans les lignes commençant par ces entités ont chacun une signification.

Les champs d'une ligne sont composés des entités :

- body (corps du solide) indique le numéro de ligne de son lump, de son wire.
- lump (bloc du solide) indique le numéro de ligne de son shell et de son body.
- shell ( enveloppe contenant ou délimitant un bloc) indique le numéro de ligne de la prochaine enveloppe, de la première face et du lump.
- face, élément surfacique constituant l'enveloppe définie par une surface géométrique et un contour, indique le numéro de ligne du loop, du shell, de la surface géométrique, de la face et du sens.
- loop, contour délimitant une face indique le numéro de ligne du premier coedge et de la face.
- coedge, élément linéique constituant un contour, indique le numéro de ligne du prochain coedge, du coedge précédent, du coedge partagé avec une autre face, du edge, du sens du coedge, du loop, et de la courbe géométrique.
- edge, élément linéique représentant une séparation entre 2 surfaces, indique le numéro de ligne du vertex de départ, du vertex de fin, du coedge, de la géométrie de la courbe et du sens du edge.
- vertex, élément ponctuel limitant les droites ou les courbes, indique le numéro de ligne du edge et du point.
- transform, matrice de changement de base, vecteur translation.

La figure 9.1 montre un exemple de ces entités géométriques. La figure 9.2 montre un exemple de fichier B-Rep.

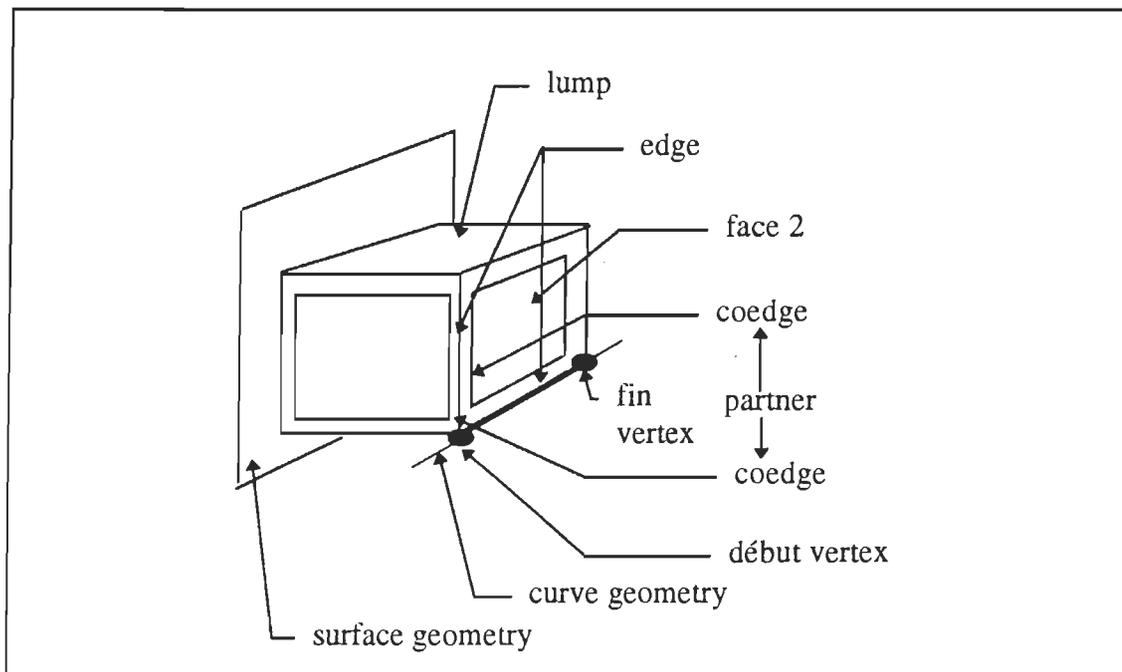


Figure 9.1: Entités géométriques d'un cube.

```

103 30 0 0
0 body $1 $2 $-1 $3 #
1 brep_non_volatile_id-cdc-attrib $-1 $-1 $-1 $0 7 #
2 lump $-1 $-1 $4 $0 #
3 transform $-1 1 0 0 0 0 1 0 -1 0 10 -3 -10 1 1 0 0 #
4 shell $-1 $-1 $-1 $5 $2 #
5 face $6 $7 $8 $4 $-1 $9 0 #
6 primitive_order-cdc-attrib $-1 $-1 $-1 $5 0 3 #
7 face $10 $11 $12 $4 $-1 $13 0 #
8 loop $-1 $14 $15 $5 #
9 cone-surface $-1 0 0 0 0 0 1 1 0 0 1 0 1 0 #
10 primitive_order-cdc-attrib $-1 $-1 $-1 $7 0 3 #
11 face $16 $-1 $17 $4 $-1 $18 0 #
12 loop $-1 $-1 $19 $7 #
13 plane-surface $-1 0 0 -3 0 0 -1 -1 0 0 0 #
14 loop $-1 $-1 $20 $5 #
15 coedge $-1 $15 $15 $19 $21 1 $8 $-1 #
16 primitive_order-cdc-attrib $-1 $-1 $-1 $11 0 3 #
17 loop $-1 $-1 $22 $11 #
18 plane-surface $-1 0 0 3 0 0 1 1 0 0 0 #
19 coedge $-1 $19 $19 $15 $21 0 $12 $-1 #
20 coedge $-1 $20 $20 $22 $23 1 $14 $-1 #
21 edge $-1 $24 $24 $19 $25 0 #
22 coedge $-1 $22 $22 $20 $23 0 $17 $-1 #
23 edge $-1 $26 $26 $22 $27 0 #
24 vertex $-1 $21 $28 #
25 ellipse-curve $-1 0 0 -3 0 0 -1 1 0 0 1 #
26 vertex $-1 $23 $29 #
27 ellipse-curve $-1 0 0 3 0 0 1 1 0 0 1 #
28 point $-1 1 0 -3 #
29 point $-1 1 0 3 #

```

A la ligne 7  
la flèche pointée  
sur 11 indique  
la prochaine face.  
Celle pointée sur  
13 indique le type  
de surface de la  
face de la ligne 7.

Figure 9.2: exemple de fichier B\_Rep