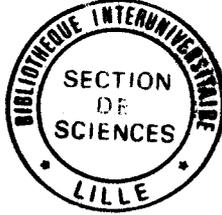


N° d'ordre: 734

50376  
1991  
208



65800

50376  
1991  
208

**THESE**

présentée à

**L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE  
FLANDRES ARTOIS**

pour l'obtention du titre de

**DOCTEUR**

**En Productique : Automatique et Informatique Industrielle**

par

**Abderrahmane OKAT**

**Maître ès Sciences Construction Mécanique à l'E.U.D.I.L.**

**CONTRIBUTION A LA CONDUITE D'UNE CELLULE FLEXIBLE  
DE DECOUPE AU LASER**

Soutenue le 7 Novembre 1991 devant la commission d'examen:

**JURY:**

<b>Président:</b>	<b>Pierre VIDAL</b>	<b>Professeur à l'U.S.T.L.F.A. de Lille 1</b>
<b>Rapporteur:</b>	<b>Pierre VIDAL</b>	
<b>Rapporteur:</b>	<b>René SOENEN</b>	<b>Professeur à l'Université de Valenciennes</b>
<b>Directeur de Recherche:</b>	<b>Marcel STAROSWIECKI</b>	<b>Professeur à l'U.S.T.L.F.A. de Lille 1</b>
<b>Examineur:</b>	<b>Salah MAOUCHE</b>	<b>Maître de Conférence à l'U.S.T.L.F.A. de Lille 1</b>
<b>Examineur:</b>	<b>Alain SCHMITT</b>	<b>Responsable de Recherche à l'Ecole des Mines de DOUAI</b>
<b>Invité:</b>	<b>Didier LANSIAUX</b>	<b>Directeur de l'Ecole des Mines de DOUAI</b>

Le travail exposé dans ce mémoire a été réalisé au Laboratoire d'Automatique de l'Université de Lille de Monsieur le Professeur d'Université Pierre VIDAL et sous la direction de Monsieur le Professeur Marcel STAROSWIECKI. Avant d'en entreprendre l'exposé je tiens à exprimer ma reconnaissance à toutes personnes qui m'ont aidé dans ce travail.

J'exprime ma gratitude à Monsieur Pierre VIDAL, Professeur d'Université et Directeur du Centre d'Automatique de l'Université des Sciences et Techniques de Lille Flandres Artois, pour l'accueil qu'il m'a réservé au sein de son laboratoire et de m'avoir permis d'entreprendre mes travaux sur la plateforme productique de son laboratoire. Je le remercie d'avoir accepté la présidence de mon jury de thèse. D'autre part, je lui suis très reconnaissant d'avoir apporté des jugements, scientifiques et techniques, pour une amélioration de cette thèse en tant que rapporteur.

J'adresse mes remerciements à Monsieur Marcel STAROSWIECKI, Professeur à l'U.S.T.L.F.A. , qui m'a confié cette étude.

Que Monsieur René SOENEN, Professeur à l'Université de Valenciennes, accepte mes plus vifs remerciements pour l'intérêt qu'il a bien voulu porter au jugement de ce travail, pour ses précieux conseils et pour sa présence parmi les membres du jury en tant que rapporteur.

Que Monsieur Salah MAOUCHE, Maître de Conférence à l'U.S.T.L.F.A. soit remercié très sincèrement pour ses conseils éclairés et son parfait encadrement.

Je remercie également Monsieur SCHMITT, Docteur en Mécanique et Responsable d'études en Productique au Centre de Recherche DORIGNIES pour l'intérêt qu'il a témoigné à ces travaux en tant que membre du jury.

L'ensemble de mes travaux a pu être mené à bien grâce à la coopération et l'aide de tout le personnel du Laboratoire d'Automatique de Lille Flandres Artois. C'est pourquoi je tiens à adresser à Mesdames M. C. ELMEKKI, A. PIGNON, ainsi que Messieurs B. CEURSTEMONT, M. POLLART, G. VEINANT mes plus vifs remerciements.

*A mes parents sans qui je ne serais pas là.  
A ma femme pour sa patience et son soutien moral.  
A mes enfants pour mes absences.  
A mes soeurs et frères pour leur compréhension.*

<b>INTRODUCTION GENERALE</b> .....	8
<b><u>CHAPITRE I: ANALYSE D'UNE ENTREPRISE DE CONFECTION</u></b>	
<b>INTRODUCTION</b> .....	10
<b>I.1 NECESSITE D'UNE METHODE D'ANALYSE</b> .....	10
<b>I.2 S.A.D.T. UNE METHODE DE COMMUNICATION</b> .....	11
I.2.1 Présentation .....	11
I.2.2 Le modèle S.A.D.T. ....	13
I.2.3 Syntaxe des diagrammes .....	13
<b>I.3 ORGANISATION D'UNE ENTREPRISE DE CONFECTION</b> .....	13
I.3.1 Fonction "ETUDIER" .....	15
I.3.1.1 <i>La conception</i> .....	15
I.3.1.2 <i>L'industrialisation</i> .....	17
I.3.1.3 <i>Préparation de la fabrication</i> .....	18
I.3.2 Fonction "ORGANISER LA PRODUCTION".....	19
I.3.2.1 <i>Fonction "ORDONNANCER"</i> .....	20
I.3.2.2 <i>Fonction "GERER LES STOCKS"</i> .....	20
I.3.3 Fonction "FABRICATION" .....	21
<b>I.4 LE CONCEPT D'INFORMATION</b> .....	21
I.4.1 Structure informationnelle .....	22
I.4.1.1 <i>Schéma global</i> .....	22
I.4.1.2 <i>Structure horizontale</i> .....	23
I.4.1.3 <i>Structure verticale</i> .....	23
<b>I.5 ORGANISATION DES FABRICATIONS : LA COUPE ET LE PIQUAGE</b> .....	25
I.5.1 Fonction "LANCEMENT AU PIQUAGE".....	26
I.5.2 Fonction "LANCEMENT DE LA COUPE" .....	26
I.5.3 Lancement et suivi de fabrication .....	27
I.5.3.1 <i>Prélancement-programme de fabrication</i> .....	27
I.5.3.2 <i>Lancement</i> .....	28
<b>I.6 SUPPORT DE L'ETUDE</b> .....	30
<b>I.7 PRESENTATION DU SITE EXPERIMENTAL</b> .....	31
<b>I.8 CONCLUSION</b> .....	32

## CHAPITRE II: PRESENTATION DE LA DECOUPE

<b>INTRODUCTION.....</b>	<b>34</b>
<b>II.1 MAITRISE DE LA CONSOMMATION DE LA MATIERE PREMIERE .....</b>	<b>35</b>
<b>II.2 OPTIMISATION DE LA DECOUPE (APPROCHE CLASSIQUE).....</b>	<b>35</b>
<b>II.3 LA DECOUPE.....</b>	<b>37</b>
II.3.1 Découpe mono-dimensionnelle .....	37
II.3.1.1 Formulation .....	37
II.3.1.2 Modélisation du problème mono-dimensionnel .....	40
II.3.2 Découpe bi-dimensionnelle .....	41
II.3.2.1 Formulation .....	42
II.3.2.2 Modélisation .....	43
<b>II.4 STRUCTURE DE DONNEE .....</b>	<b>44</b>
II.4.1. Introduction.....	44
II.4.2 Structuration de données.....	45
II.4.2.1 Architecture.....	45
II.4.2.2 Données communes au placement et à la découpe.....	45
II.4.2.3 Modèle géométrique .....	46
II.4.2.4 Représentation de stockage .....	49
II.4.2.5 Modèle de traitement .....	50
<b>II.5 DECOUPE DE PIECES DE VETEMENTS .....</b>	<b>51</b>
II.5.1 La cellule de découpe .....	51
II.5.2 Modularité de la cellule.....	52
II.5.3 Aspects techniques de la cellule.....	53
<b>II.6 CONCLUSION.....</b>	<b>54</b>

## CHAPITRE III: GESTION DE L'AVANCE DU TISSU

<b>INTRODUCTION</b> .....	56
<b>III.1 GESTION DE L'AVANCE DU TISSU</b> .....	57
III.1.1 Position du problème.....	57
III.1.2 Données du problème .....	57
<b>III.2 CRITERE ET CONTRAINTES</b> .....	58
III.2.1 Contraintes.....	58
III.2.2 Critère.....	58
<b>III.3 FORMALISATION DU PROBLEME Pb1</b> .....	60
III.3.1 Approche du problème .....	60
<i>III.3.1.1 Approche classique</i> .....	60
<i>III.3.1.2 Une approche du problème</i> .....	63
<b>III.4 FORMALISATION PAR LA PROGRAMMATION DYNAMIQUE</b> .....	65
III.4.1 Rappels .....	65
III.4.2 Caractéristiques du problème .....	67
III.4.3 Représentations.....	68
<i>III.4.3.1 représentation par les "boîtes"</i> .....	68
<i>III.4.3.2 Représentation par graphe séquentiel</i> .....	69
III.4.4 Formalisation.....	71
III.4.5 Equation de récurrence.....	73
III.4.6 limite de la programmation dynamique .....	74
<b>III.5 FORMALISATION PAR LA PROGRAMMATION LINEAIRE</b> .....	74
III.5.1 Notion de réseau .....	75
III.5.2 Notion sur la programmation linéaire .....	77
III.5.3 Formalisation.....	78
III.5.4 Exemple d'application.....	79
<b>III.6 CONCLUSION</b> .....	80

## CHAPITRE IV: GENERATION DE TRAJECTOIRES

<b>INTRODUCTION.....</b>	<b>81</b>
<b>IV.1 POSITION DU PROBLEME.....</b>	<b>81</b>
IV.1.1 Données du problème.....	81
IV.1.2 Critère d'optimisation.....	83
<b>IV.2 APPROCHE DU PROBLEME.....</b>	<b>83</b>
IV.2.1 Caractéristique d'une tâche.....	83
IV.2.2 Caractéristique d'un trajet.....	85
IV.2.3 Notion de coût.....	87
IV.2.3.1 Coûts associés au déplacement et à la découpe.....	87
IV.2.3.2 Coûts associés à un trajet.....	88
IV.2.4 Notion de variables décisionnelles.....	88
<b>IV.3 REPRESENTATION GRAPHIQUE.....</b>	<b>89</b>
IV.3.1 Définition.....	89
IV.3.2 Représentation graphique.....	90
IV.3.3 Représentation matricielle.....	91
<b>IV.4 PREMIERE FORMALISATION DU PROBLEME Pb2: cas général.....</b>	<b>92</b>
IV.4.1 Problème du voyageur de commerce asymétrique.....	92
IV.4.2 Formulation du problème Pb2.....	93
IV.4.3 Conclusion.....	95
<b>IV.5 MODIFICATION DE LA FORMULATION: cas du processus de découpe.....</b>	<b>95</b>
IV.5.1 Problème du "voyageur de commerce symétrique".....	96
IV.5.2 Formulation simplifiée de Pb2.....	97
<b>IV.6 CONCLUSION.....</b>	<b>99</b>

## CHAPITRE V: RESOLUTION

<b>INTRODUCTION</b> .....	100
<b>V.1 POSITION DU PROBLEME</b> .....	101
V.1.1 Donnée du problème (Pb).....	101
V.1.2 Approche classique .....	102
<b>V.2 UNE APPROCHE DU PROBLEME</b> .....	104
V.2.1. Différentes étapes de l'approche pour la résolution de (Pb).....	104
V.2.2 Transformation de la formulation du problème de base (Pb).....	105
V.2.2.1 <i>Rappel</i> .....	105
V.2.2.2 <i>Transformation de la formulation du problème</i> .....	107
V.2.3 Réindexation et contraction.....	109
V.2.3.1 <i>Réindexation</i> .....	109
V.2.3.2 <i>Contraction et passage en maximisation</i> .....	111
<b>V.3 RESOLUTION</b> .....	113
V.3.1 Rappels sur la théorie de la dualité en optimisation combinatoire .....	113
V.3.1.1 <i>Relaxation Lagrangienne et théorie de la dualité</i> .....	114
V.3.1.2 <i>Particularité de (B)</i> .....	116
V.3.2 Approche classique .....	117
V.3.2.1 <i>Majorant et minorant initiaux</i> .....	118
V.3.2.2 <i>Enumération implicite</i> .....	119
<b>V.4 CONCLUSION</b> .....	121

**CHAPITRE VI: APPLICATION NUMERIQUE ET RESULTAT**

<b>INTRODUCTION.....</b>	<b>122</b>
<b>VI.1 PRESENTATION DE LA CELLULE .....</b>	<b>121</b>
V.1.1 Description .....	121
V.1.2 Composition .....	124
<b>VI.2 MODULES ET FONCTIONNALITES.....</b>	<b>129</b>
V.2.1 Modules initiaux de programmation .....	129
V.2.2 Programmation Découpe .....	130
V.2.2.1 Structuration des données.....	131
V.2.2.2 Structure Informationnelle.....	133
<b>VI.3 APPLICATION NUMERIQUE.....</b>	<b>134</b>
<b>VI.4 CONCLUSION.....</b>	<b>142</b>

**CONCLUSION GENERALE****REFERENCES BIBLIOGRAPHIQUES****ANNEXES**

Les notations utilisées tout au long de cette thèse sont les suivantes:

étant donné un ensemble  $K$ :

- $[K]$  : enveloppe convexe de  $K$   
 $K/U$  : complémentaire d'un sous-ensemble  $U$  de  $K$

étant donnée une matrice de format  $(I*J)$  où  $I$  et  $J$  sont des sous-ensembles finis:

- $a_{ij}$  : élément  $(i,j) \in I*J$  de  $A$   
 $A_i$  : colonne  $i \in I$   
 $A^j$  : colonne  $j \in J$   
 $y_j$  : élément  $j \in J$  de  $y$

étant donné un problème d'optimisation  $(B)$

- $v^*(B)$  : valeur optimale de  $(B)$   
 $\bar{v}(B)$  : (resp.  $\underline{v}(B)$ ) majorant (resp. minorant) de  $v^*(B)$   
 $F(B)$  : domaine de  $(B)$   
 $(B \mid x \in X)$  : lorsque  $(B)$  est résolu avec les conditions supplémentaires  $x \in X$

lorsque  $B$  est un problème à  $n$  variables bivalentes:

- $cx^*$  : solution optimale de  $(B)$   
 $V = \{x \in R^n \mid x_j = 0 \text{ ou } 1, j = 1, \dots, n\}$   
 $(\bar{B})$  = problème  $(B)$  résolu dans lequel  $[V]$  remplace  $V$   
 $c\bar{x}$  = solution optimale de  $(\bar{B})$

étant donné un plan de placement  $P$  et la surface de travail  $S$

- $L_p$  : longueur associée à  $P$   
 $W_p$  : largeur associée à  $P$   
 $L_o$  : longueur associée à  $S$   
 $W_o$  : largeur associée à  $S$   
 $x_{ij}$  : variable de décision  
 $c_{ij}$  : coût associé à la décision  $x_{ij}$   
 $n$  : nombre de pas total sur  $P$   
 $\alpha$  : nombre total de pas sur  $S$

L'homme crée et adapte sans cesse de nouveaux outils destinés à accroître son efficacité dans le traitement des diverses tâches qu'il doit effectuer. La mise au point des techniques informatiques, particulièrement adaptées aux tâches répétitives et fastidieuses, lui permet d'accorder plus de temps aux activités d'analyse et de conception. La volonté d'une intégration de plus en plus poussée de ces techniques est donc naturelle et souhaitable. Après s'être imposé dans les fonctions économiques et financières, l'ordinateur est de plus en plus utilisé dans les autres services de l'entreprise donnant ainsi naissance à de nouvelles techniques telles que:

- la conception assistée par ordinateur C.A.O
- la fabrication assistée par ordinateur F.A.O
- la gestion de production assistée par ordinateur G.P.A.O

Les recherches menées dans ces domaines ont permis la réalisation de divers systèmes complémentaires, qui ne sont encore que trop rarement connectables. Les tentatives d'intégration ressemblent plus, à ce niveau, à des extensions ponctuelles. En effet, les logiciels existants n'apportent pas toutes les fonctionnalités et les interactions nécessaires à une réelle intégration. L'intégration des outils d'aide à la décision ne peut être envisagée que par la connaissance du processus à automatiser. Seule une analyse complète et détaillée de la fonction technique et de son environnement permet d'identifier précisément:

- les tâches devant être réalisées (actions).
- les informations échangées entre les tâches et avec l'environnement (données).
- les interactions nécessaires entre les tâches et l'environnement (séquencements et réactions).

Le niveau d'intégration retenu dépend essentiellement, d'une part, des problèmes techniques rencontrés (faisabilité algorithmique) et, d'autre part, de la politique générale de direction de l'entreprise (politique et capacité d'investissement, rentabilité de l'automatisation de certaines tâches, aspects socio-économiques...). Si ce niveau d'intégration fixe les limites fonctionnelles du système devant être mis en place, il ne doit en aucune façon fixer les limites de l'étude du processus en voie d'automatisation afin d'en préserver les possibilités d'extension futures. En fait, nous devons avoir une perception globale de l'activité industrielle de façon à intégrer, au système proposé, les fonctionnalités et les interactions nécessaires aux différentes phases de l'élaboration et de la réalisation des produits.

Les secteurs industriels de découpe et d'emboutissage de la cartonnerie, de la verrerie, de la confection et plus généralement tous les secteurs utilisant comme premier processus, la découpe d'un matériau plan, sont directement concernés par ce concept d'intégration. En effet, par leurs spécificités, ils nécessitent, en plus, des échanges de données associés aux produits pour différentes étapes de la conception à la fabrication, une intégration fonctionnelle des fonctions de conception, de préparation, de gestion et de fabrication.

Le but de notre étude est double: mettre en évidence les méthodes et les moyens nécessaires d'une part à l'intégration des fonctions de conception et de préparation de la fabrication dans le cadre des industries de la découpe et d'autre part, à l'optimisation du processus de production.

Dans une première partie nous présentons, l'analyse fonctionnelle d'une entreprise de confection. Nous prenons en compte les spécificités dues au processus de réalisation par découpe. Le modèle utilisé se base sur la méthode de communication S.A.D.T..

La paramétrisation (ou structuration des données) est une phase préliminaire au traitement des formes par découpe. Cette phase est présentée au Chap.II.

Ensuite, le problème de l'optimisation de la découpe est abordé de la manière suivante:

- dans le Chap.III, nous proposons une formulation du problème pour la gestion du processus d'avance du tissu. Une approche par la programmation dynamique est présentée, ainsi qu'une formalisation par un programme linéaire en variables bivalentes.

- au Chap. IV, nous abordons le problème de la génération de trajectoire de l'outil. Sa formalisation est basée, quant à elle, sur le problème du "voyageur de commerce symétrique et asymétrique". Ceci nous conduit à un programme linéaire en variables bivalentes.

Enfin, les techniques de résolution des problèmes seront abordés au Chap.V, où nous décrirons les méthodes exactes et heuristiques. Ce dernier chapitre est donc consacré à la résolution, par un algorithme approché, du problème de la découpe de formes planes complexes pour l'industrie de la confection.

## **INTRODUCTION**

La vie d'une entreprise est régie par un ensemble de fonctions. L'intervention sur l'une ou l'autre de ces fonctions ne peut être envisagée sans connaissance approfondie non seulement de la fonction considérée mais aussi des fonctions connexes. En effet, si l'élément modifié ne s'intègre pas parfaitement dans le processus, l'intervention aboutit obligatoirement à un échec.

Nous analysons dans ce premier chapitre, l'organisation fonctionnelle et informationnelle d'une entreprise. Cette connaissance de l'entreprise nous permettra d'envisager les techniques et les modes d'implantation nécessaires à l'automatisation des fonctions retenues.

### **I.1 NECESSITE D'UNE METHODE D'ANALYSE**

La complexité d'un processus et la rigueur nécessaire à la conduite de l'étude induisent l'utilisation d'une méthode d'analyse fonctionnelle. En effet une démarche à la fois structurée et très modulaire permet de gagner en clarté et en compréhension du modèle. La structure hiérarchique à niveau de décomposition quelconque offre les avantages d'être modulaire et de permettre d'isoler certains traitements, introduisant ainsi une fiabilité accrue de l'analyse et donc de l'implémentation. De plus, elle permet la répartition des tâches lors de la réalisation, en assurant la cohérence des données de traitement et à un niveau supérieur, la compatibilité des informations manipulées avec le reste du système général. L'outil utilisé pour notre analyse est la méthode d'analyse fonctionnelle et d'étude S.A.D.T. (Structured Analysis and Design Technique [1]). Cette méthode d'analyse est basée sur les concepts suivants:

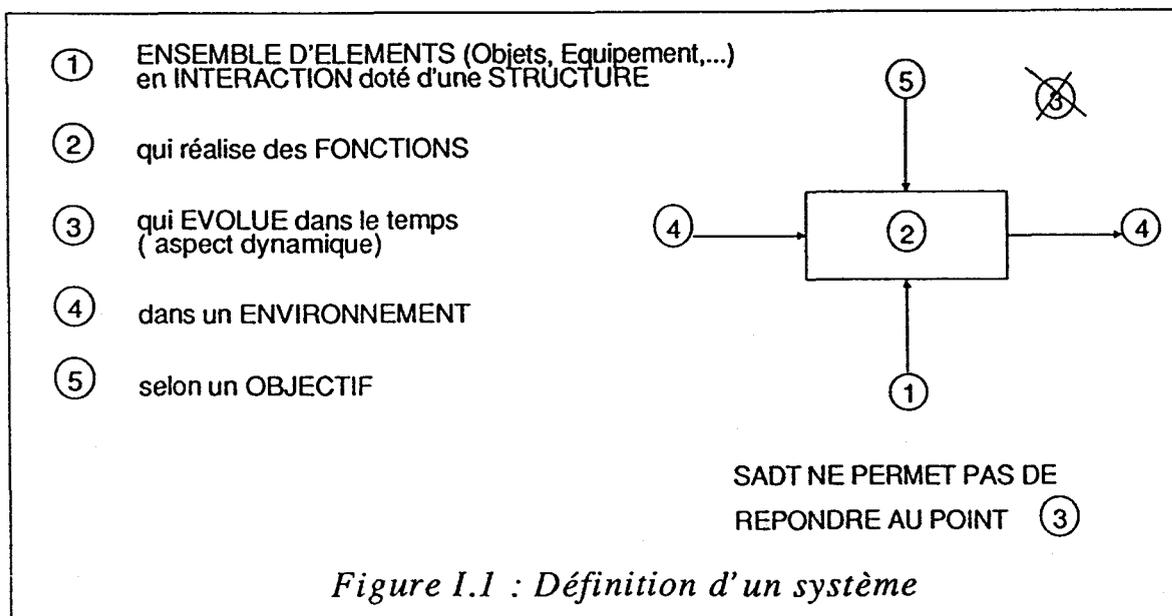
- le problème est abordé en construisant une modélisation qui exprime la connaissance approfondie du processus. Celui-ci peut être représenté par plusieurs modèles S.A.D.T., selon des points de vues différents.

- l'analyse est menée de façon descendante, hiérarchique, modulaire et structurée.
- une distinction est faite entre, d'une part, l'analyse fonctionnelle et, d'autre part, le modèle de mise en oeuvre.
- la modélisation s'effectue non seulement sur les objets mais aussi sur les activités.
- S.A.D.T. favorise un travail d'équipe discipliné et coordonné.
- les choix importants faits pendant la conception doivent obligatoirement être consignés par écrit, introduisant ainsi une forme d'historique de création.

**I.2 S.A.D.T. UNE METHODE DE COMMUNICATION**

**I.2.1 Présentation**

S.A.D.T. est une méthode générale qui cherche à favoriser la communication dans l'entreprise. Elle est présentée comme une méthode de spécification fonctionnelle, conduisant à la réalisation d'un ou plusieurs modèles. Le modèle S.A.D.T. permet de formaliser l'état du système étudié comme le montre la figure I.1, et, d'en dégager les points critiques, notamment la complexité du système.



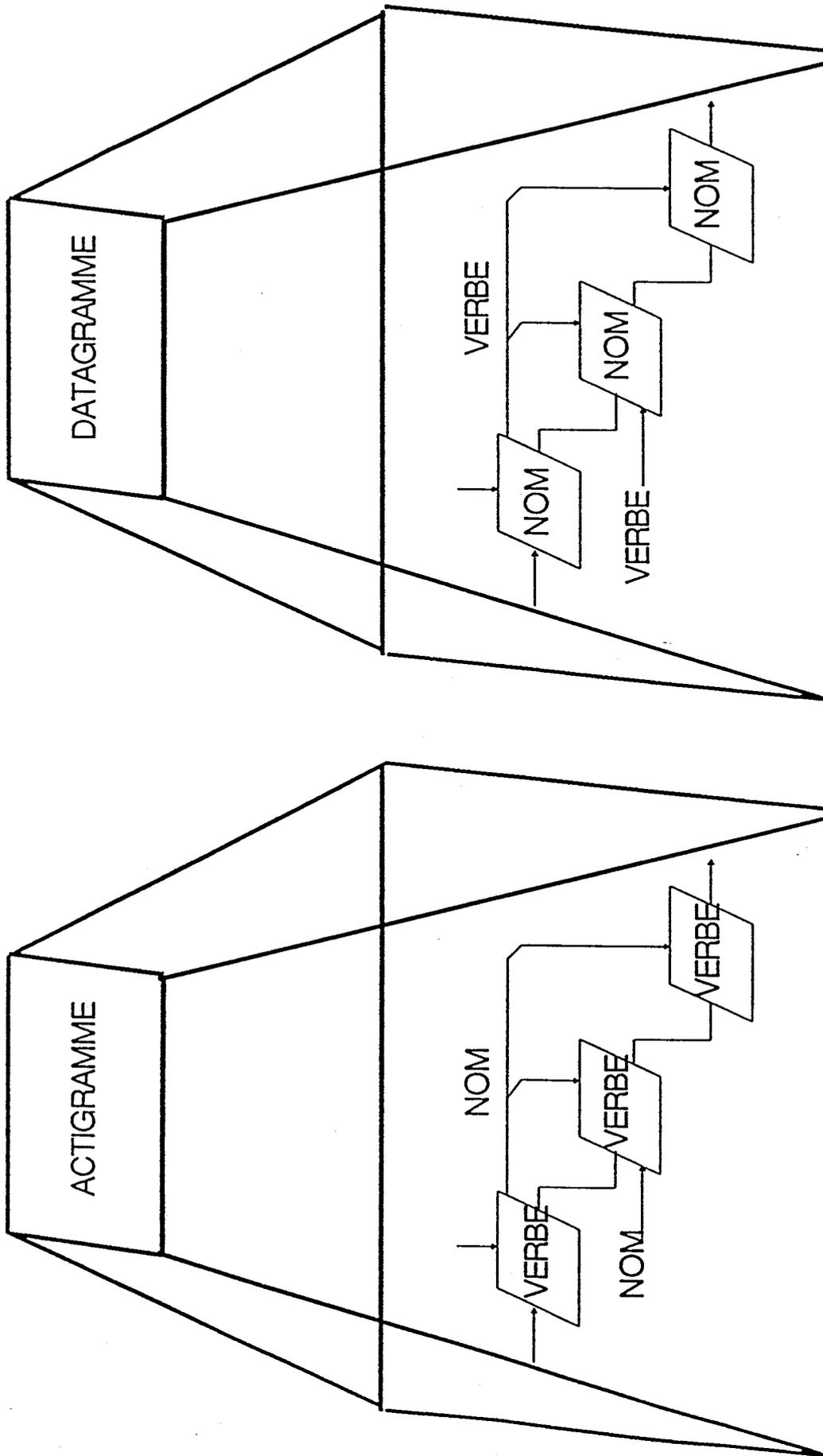


Figure I.2 : Actigrammes ( décomposition des activités) et datagrammes ( décomposition des données)

- l'analyse est menée de façon descendante, hiérarchique, modulaire et structurée.
- une distinction est faite entre, d'une part, l'analyse fonctionnelle et, d'autre part, le modèle de mise en oeuvre.
- la modélisation s'effectue non seulement sur les objets mais aussi sur les activités.
- S.A.D.T. favorise un travail d'équipe discipliné et coordonné.
- les choix importants faits pendant la conception doivent obligatoirement être consignés par écrit, introduisant ainsi une forme d'historique de création.

**I.2 S.A.D.T. UNE METHODE DE COMMUNICATION**

**I.2.1 Présentation**

S.A.D.T. est une méthode générale qui cherche à favoriser la communication dans l'entreprise. Elle est présentée comme une méthode de spécification fonctionnelle, conduisant à la réalisation d'un ou plusieurs modèles. Le modèle S.A.D.T. permet de formaliser l'état du système étudié comme le montre la figure I.1, et, d'en dégager les points critiques, notamment la complexité du système.

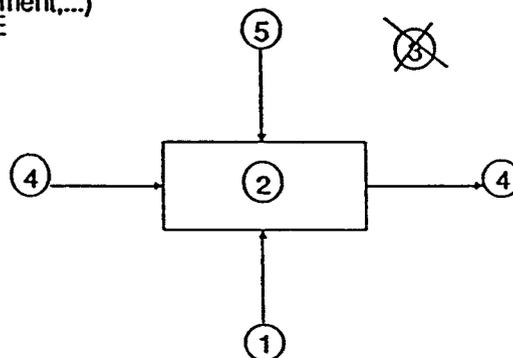
① ENSEMBLE D'ELEMENTS (Objets, Equipement,...) en INTERACTION doté d'une STRUCTURE

② qui réalise des FONCTIONS

③ qui EVOLUE dans le temps ('aspect dynamique)

④ dans un ENVIRONNEMENT

⑤ selon un OBJECTIF



SADT NE PERMET PAS DE REpondre AU POINT ③

Figure I.1: Définition d'un système

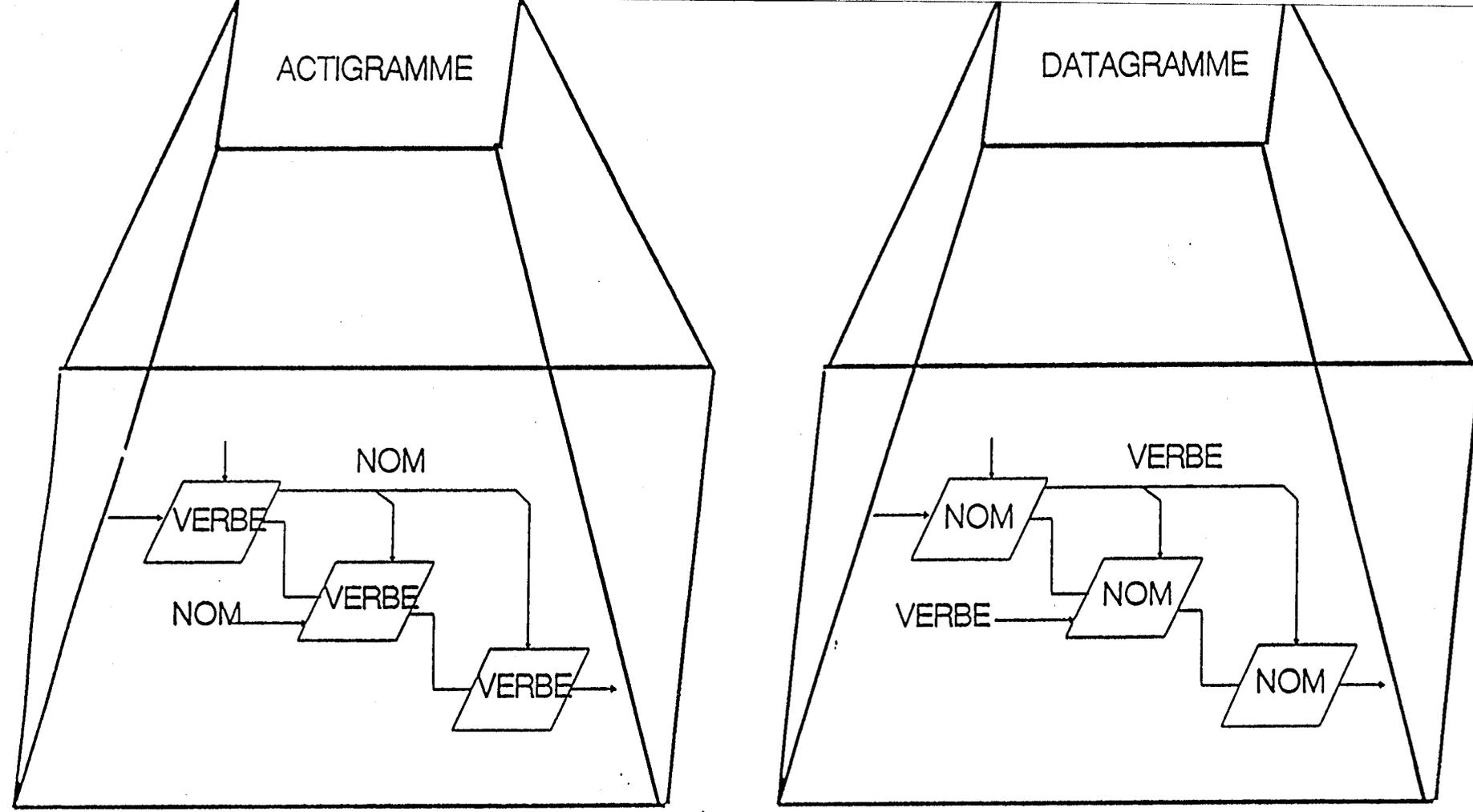


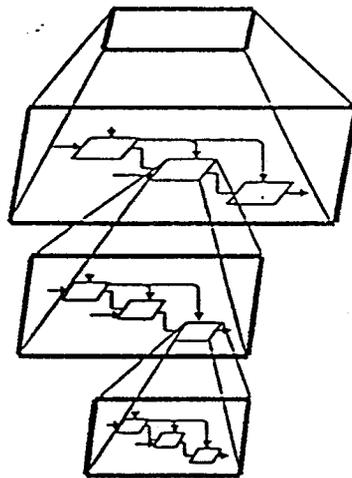
Figure 1.2 : Actigrammes ( décomposition des activités)  
et datagrammes ( décomposition des données)

3882356

Cette méthode permet donc de "Modéliser la réalité". Les aspects "données" et "activités" d'un projet sont toujours examinés ensemble comme le montre la figure I.2. Ainsi, S.A.D.T. demande la création de deux diagrammes:

- celui des ACTIVITES, représentées par des boîtes manipulant des données véhiculées sur des flèches,
- celui des DONNEES, également représentées par des boîtes où les flèches montrent, cette fois, les ACTIVITES qui créent et utilisent les données.

Les liaisons entre les modules S.A.D.T. sont assurées comme le montre la figure I.3.



*Figure I.3 : Niveau hiérarchique de décomposition*

### I.2.2 Le modèle S.A.D.T.

Un modèle S.A.D.T. permet de représenter la structure informationnelle de l'entreprise.

### I.2.3 Syntaxe des diagrammes

Le diagramme d'activité identifié par un verbe d'action:

- crée, génère une donnée de sortie,
- transforme, modifie, change d'état une donnée d'entrée,
- sollicite la donnée d'entrée,

à partir de directives de contrôle, en s'appuyant sur les potentialités des mécanismes ( figure I.4).

Nous venons de présenter les principale notions nécessaire à l'étude suivante. En ce qui concerne le formalisme du modèle SADT, le lecteur pourra se référer à [2, 3] pour plus d'informations

## I.3 ORGANISATION D'UNE ENTREPRISE DE CONFECTION

La vie d'une entreprise industrielle est régie par un ensemble de fonctions [4] techniques, commerciales, financières, administratives... Ces différentes fonctions réalisent des tâches spécifiques qui nécessitent une interaction informationnelle non seulement avec les tâches du même service mais aussi avec les autres services de l'entreprise, voire des entreprises partenaires (clients, sous-traitants,...). La principale fonction d'une entreprise industrielle est de concevoir, de réaliser et de vendre ses produits (fonctions techniques et commerciales). Il est donc normal que son organisation supporte, avec le maximum d'efficacité, les tâches et les échanges d'informations nécessaires aux deux étapes initiales de la vie d'un produit ( figure I.5):

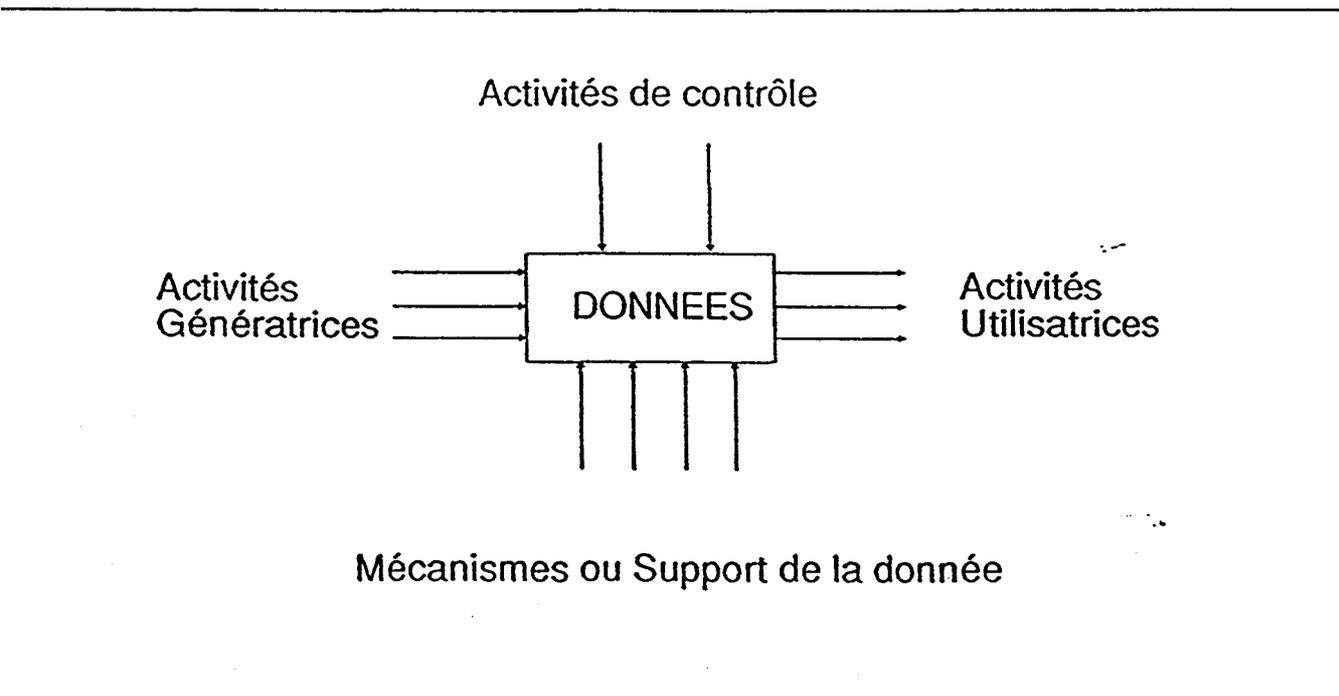
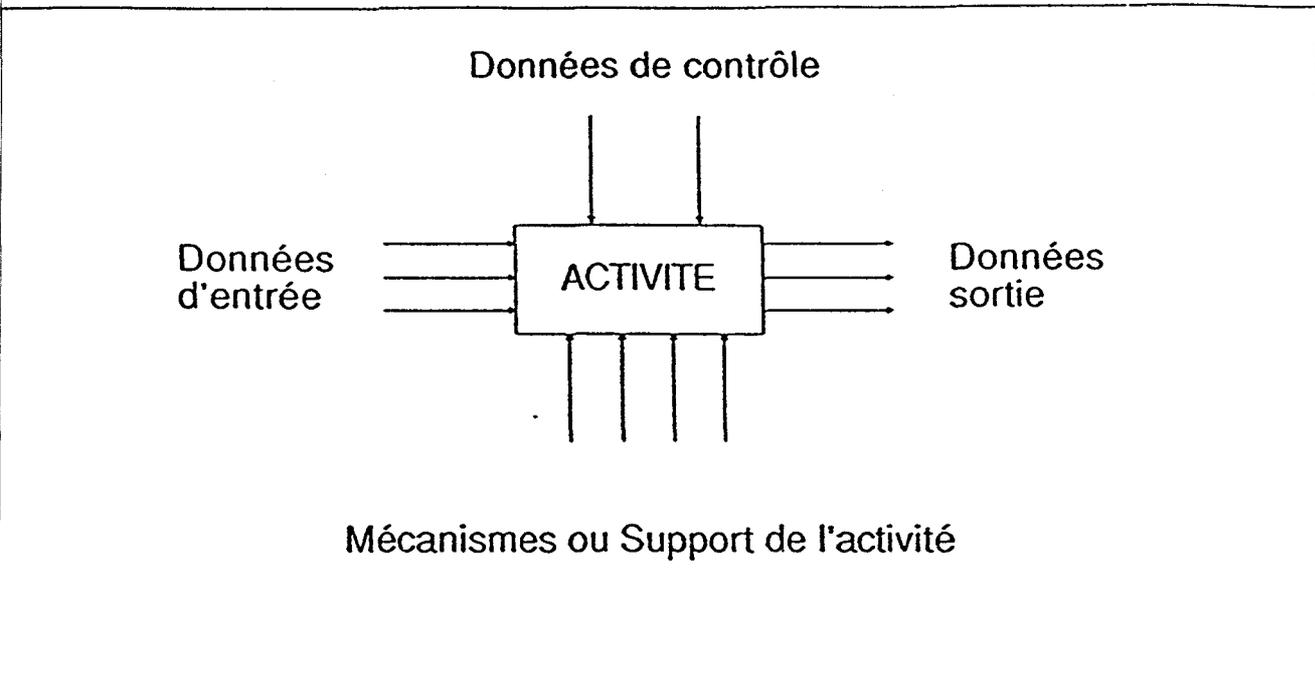
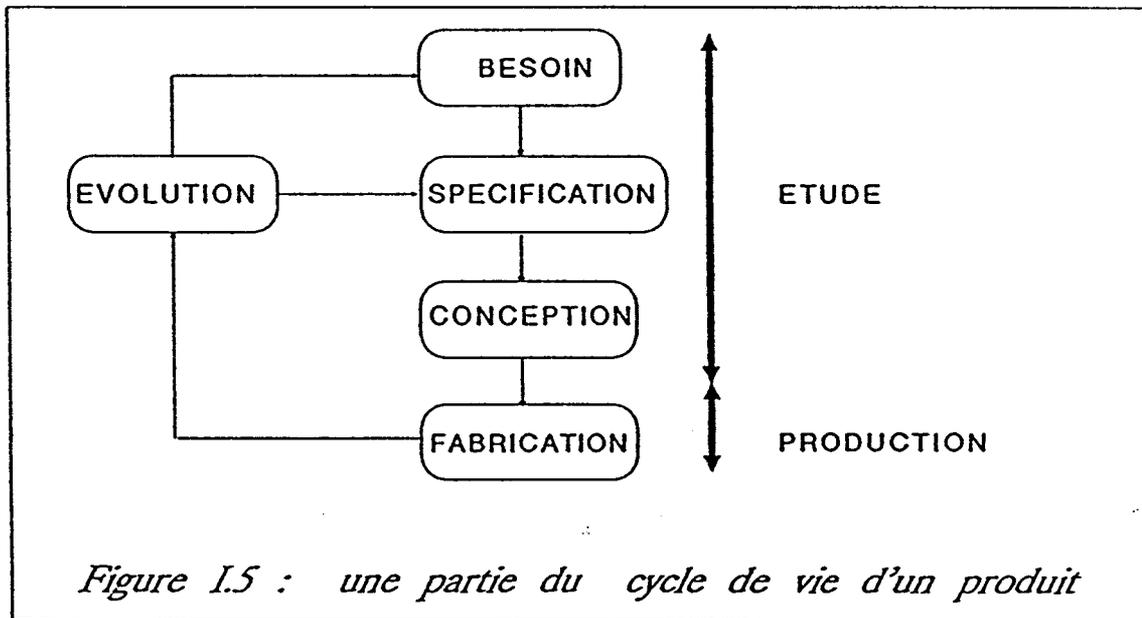


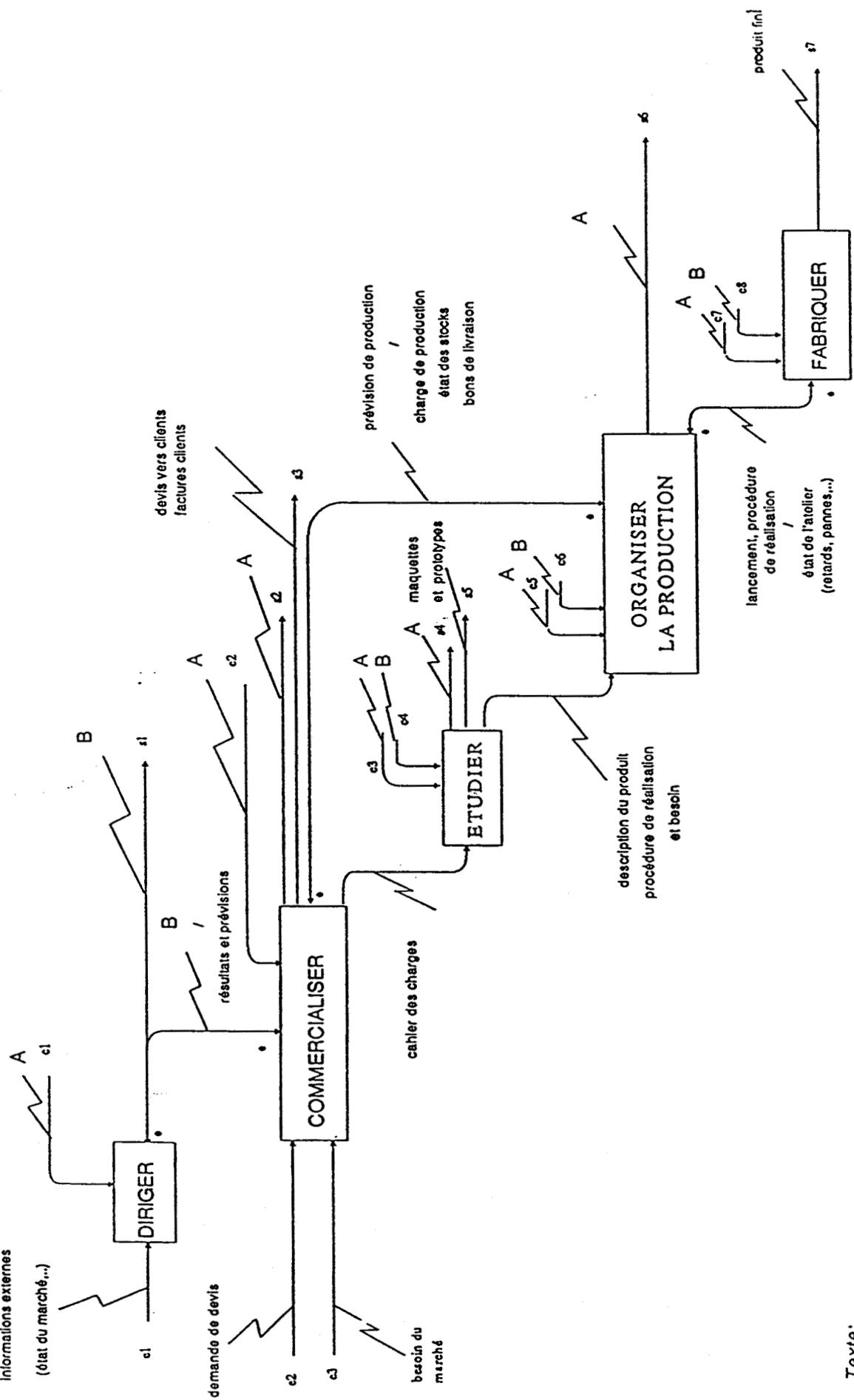
Figure I.4 : Symbole graphique d'un diagramme de données et d'activité

- la phase d'étude,
- la phase de production.



Nous présentons donc l'organisation fonctionnelle et informationnelle d'une entreprise en prenant en compte les caractéristiques spécifiques au secteur de la découpe dans la confection.

Notre modélisation S.A.D.T. de l'entreprise, présentée figure I.6, est essentiellement basée sur la fonction technique. Nous y avons cependant intégré les fonctions commerciales et de direction afin de pouvoir définir la nature des flux d'information qu'elles échangent avec la fonction technique. D'autre part, cette approche préserve l'intégralité des cycles d'élaboration et de réalisation des produits.



**Texte:**

- A : savoir faire de l'entreprise
- B : Politique générale de gestion de l'entreprise

Figure 1.6 : organigramme général de l'entreprise

### **I.3.1 Fonction "ETUDIER"**

Examinons tout d'abord, dans un contexte conventionnel [5], les différentes étapes aboutissant à la fabrication du vêtement. Le diagramme relatif à cette activité est représentée figure I.7. Nous y retrouvons les trois modules suivant:

- la conception,
- l'industrialisation,
- la préparation de la fabrication.

On note que chacune de ces étapes correspond à une certaine spécialisation des opérations au sein du bureau d'étude. Le diagramme de la fig. I.7 montre le cheminement du produit à l'intérieur de la cellule de conception ainsi que le nom des corps de métier concernés dont nous expliquons le rôle dans les paragraphes suivant.

#### **I.3.1.1 La conception**

Le document de travail servant à la cellule de conception est le croquis du styliste représentant l'expression artistique d'une idée. Son interprétation, pour obtenir le premier prototype du vêtement, constitue la première partie de la tâche du modéliste:

- choix des futurs panneaux du vêtement,
- choix des montages principaux,
- ...

La seconde partie de sa tâche est constituée par la création proprement dite de chaque panneau. Ce processus demande une grande expérience, une certaine intuition et peut être mené de plusieurs façons, selon la complexité du vêtement à créer.

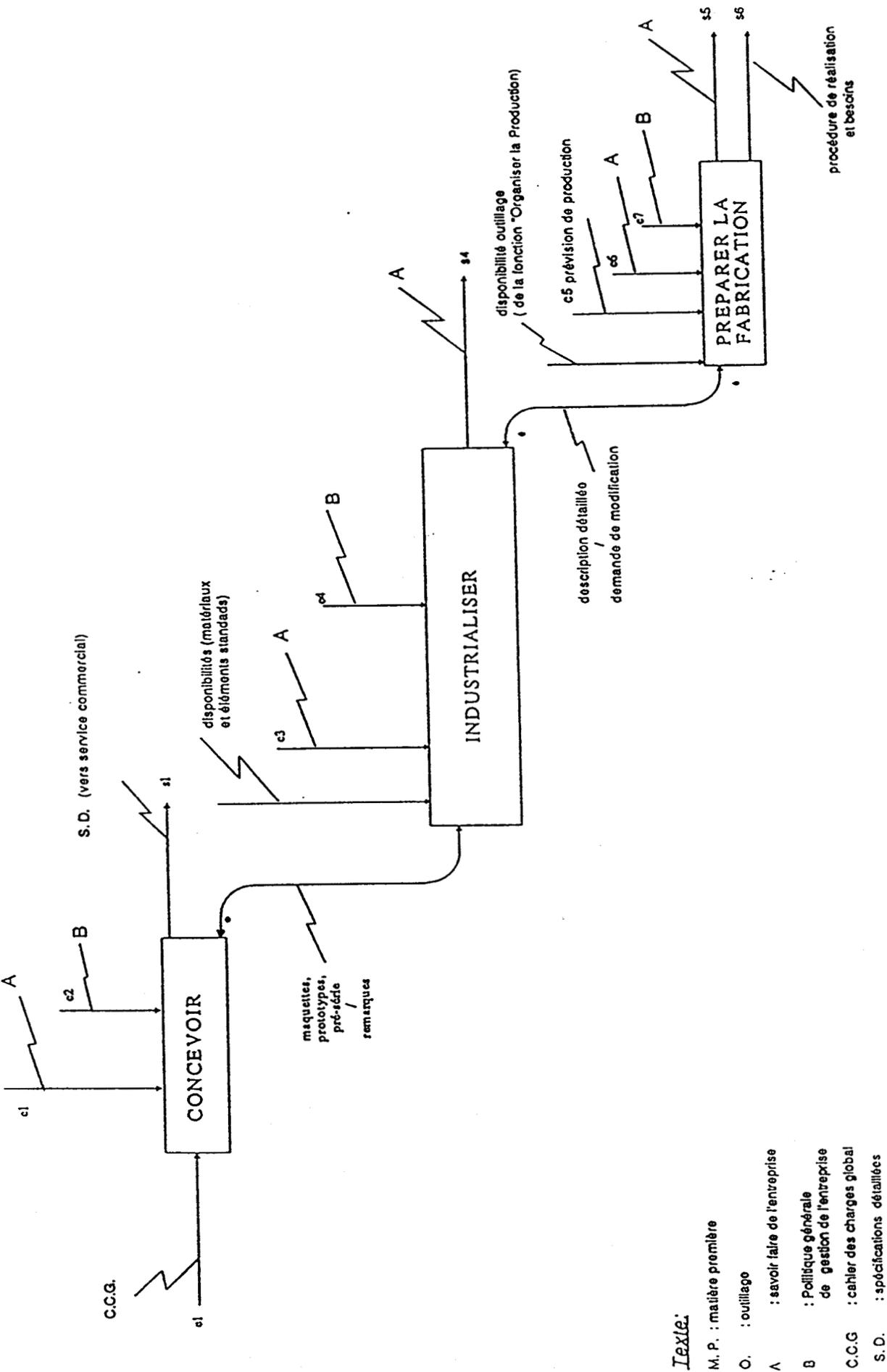


Figure 1.7 : Diagramme de la fonction "ETUDIER"

Si le vêtement est complexe (veste par exemple), on procède par approximations successives complétées d'essais sur mannequins. Il s'agit alors de conception bi-dimensionnelle, avec modifications par essayage tri-dimensionnel.

Dans certains cas, comme pour la conception de certains empiècements, la forme de la pièce est "dessinée" sur un panneau déjà existant, au cours de l'essayage. Après remise à plat du panneau, il ne reste plus qu'à découper la pièce conçue. Il s'agit par conséquent, dans ce procédé, de conception tri-dimensionnelle directe.

Si le vêtement est plus simple, comme c'est le cas pour certaines jupes, il est jugé plus rapide et plus commode d'utiliser un jeu de gabarits bien connus qui sont modifiés sur la table de travail pour s'adapter au modèle à concevoir. Ce type de conception est plus systématique que les précédents et s'appuie sur un ensemble de règles empiriques donnant la plupart du temps satisfaction.

L'essayage ultérieur de ce type de vêtement s'avère souvent positif et n'a donc valeur que de vérification.

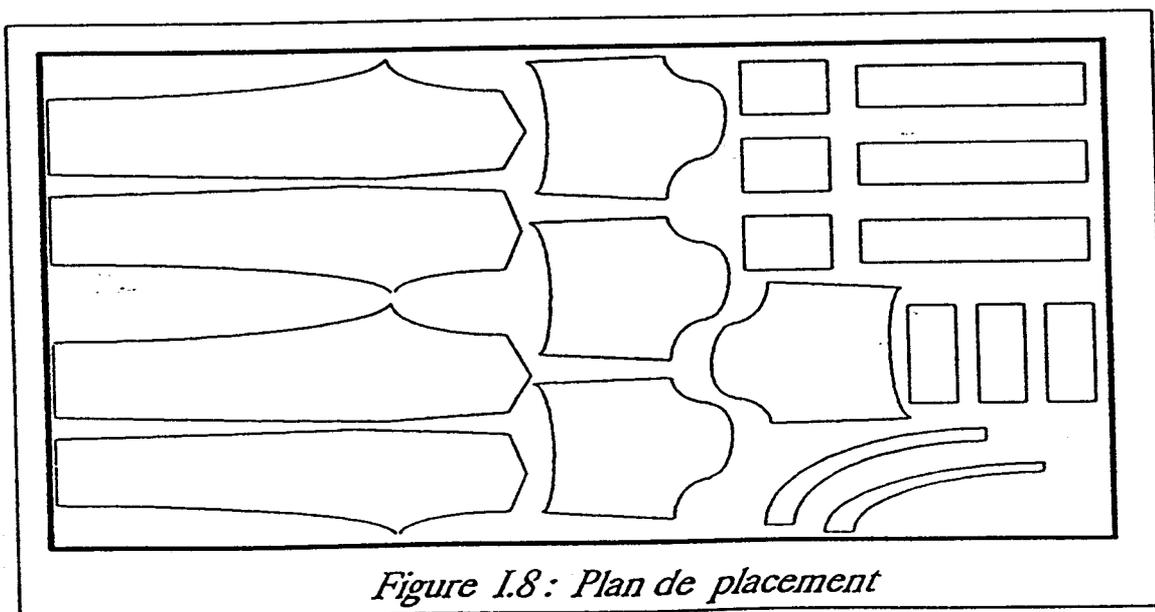


Figure 1.8: Plan de placement

La figure I.8 montre l'ensemble des panneaux issus de cette étape. Le patron obtenu est appelé "patron de toile" par référence à la toile de qualité et de texture particulières utilisée par le modéliste.

Nous retiendrons de cette description:

- l'aspect empirique et intuitif du travail du modéliste,
- la nécessité pour ce même modéliste de travailler en trois dimensions et donc de façon très concrète sur la forme à habiller, malgré les excellentes facultés d'abstraction et de synthèse de l'opérateur.

### 1.3.1.2 L'industrialisation

La "toile" étant créée, on procède ensuite à l'industrialisation du prototype, artisanal, mais néanmoins conçu pour faciliter cette industrialisation. On distingue dans cette phase trois étapes:

- choix des détails de montage. Il s'agit du troisième et dernier niveau de détail en ce qui concerne le montage, les deux précédents étant pris en compte respectivement par le styliste et le modéliste,
- rajout des "valeur-coutures" nécessaires à l'assemblage. Cette opération systématique est liée à la précédente. A chaque type de montage correspond une valeur couture standardisée parfaitement déterminée,
- détermination des lois de gradation. La gradation est l'opération qui permet de déduire le patron correspondant à une taille retenue pour le vêtement du patron connu en taille de base. Les patronnières déterminent, à l'aide d'un ensemble de règles connues empiriquement, un certain nombre de lois assimilables par le système de gradation automatisé (exemple: SOPHIA) déjà présent dans certaines entreprises et qui se charge de générer le patron correspondant.

Cette opération est intéressante car elle permet de n'effectuer les opérations d'industrialisation qu'une seule fois pour toutes les tailles du vêtement. Néanmoins, si un doute existe quant à la validité des lois de gradations, on réalise manuellement un prototype qui est essayé sur mannequin.

Si nous rapprochons le travail du modéliste de celui de la patronnière, on remarque que l'approche de cette dernière est beaucoup plus bi-dimensionnelle (une grande partie de ce travail s'affiche "à plat"), bien que la dualité 2D-3D du vêtement existe toujours à ce niveau. On remarque également l'aspect plus systématique de certains travaux, tel le rajout des valeurs-coutures ou les opérations de gradation.

### I.3.1.3 Préparation de la fabrication

Une fois le produit entièrement défini et validé, l'activité "ETUDIÉ" prend en charge la préparation de la fabrication. Cette phase précise les procédures et les ressources (humaines et matérielles) nécessaires à la réalisation du produit.

La fonction "METHODE" en confection est classique. Elle regroupe les activités de préparation des gammes de fabrication et l'évaluation des coûts de production. Ces activités se répartissent en deux niveaux:

- une première évaluation qui suit immédiatement la conception du produit et permet de vérifier si celui-ci est conforme à la politique commerciale de l'entreprise. Cette évaluation porte sur les coûts matière, machine, main-d'oeuvre et frais généraux supportés par le produit,
- le second niveau d'évaluation affine les modes opératoires sur lesquels sont basées les évaluations précédentes et ré-évalue les coûts correspondants. Ce travail de préparation nécessite une bonne connaissance des moyens matériels et humains de l'entreprise.

L'évaluation s'appuie sur l'utilisation d'une bibliothèque de méthodes standards mise en place et maintenue par les analystes eux-mêmes. Elle comprend essentiellement:

- un descriptif du montage (schémas),
- une décomposition du montage en opérations élémentaires,
- un relevé des temps technologiques et de manipulation pour chaque opération de montage.

Cette bibliothèque est fondamentale pour l'élaboration de la fonction Méthode, elle permet tout à la fois de définir les gammes et d'en évaluer les coûts machine et main-d'oeuvre. Le calcul des coûts matière est ensuite effectué. Il prend en compte:

- les caractéristiques géométriques des patronages,
- les caractéristiques des modes d'assemblages,
- le plan d'imbrication (ou placement) statique, qui fixera la position des pièces pour le module de découpe.

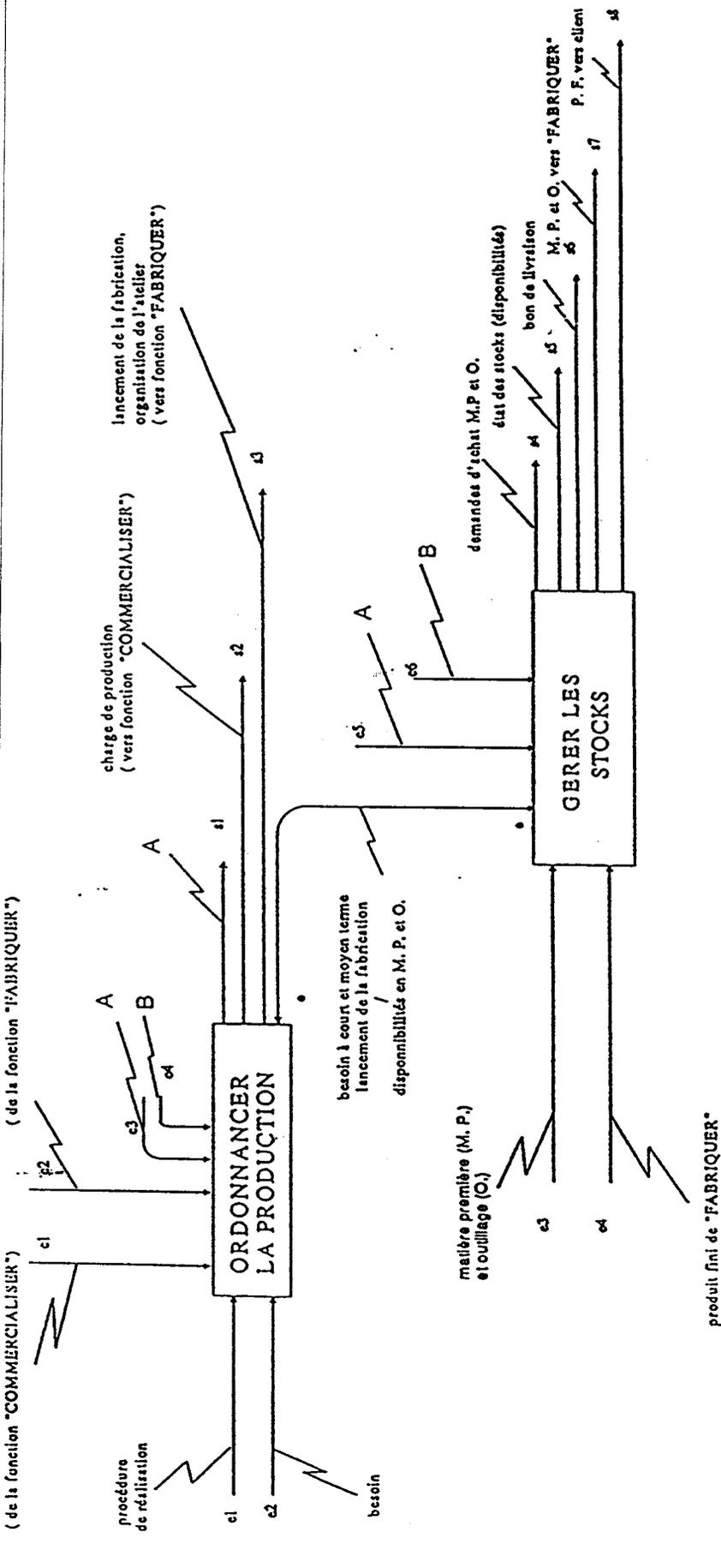
Une grande partie de l'activité du bureau des Méthodes est systématisée par la notion de montage standard, ce qui permet d'envisager une automatisation.

### I.3.2 Fonction "ORGANISER LA PRODUCTION"

Cette activité a été décomposée en deux tâches traitant:

- l'organisation "TEMPORELLE",
- l'organisation "MATERIELLE".

Le diagramme S.A.D.T. correspondant est présenté figure I.9.



**Texte:**

M. P. : matière première

O. : outillage

A : savoir faire de l'entreprise

B : Politique générale de gestion de l'entreprise

Figure 1.9 : diagramme de la fonction "ORGANISER LA PRODUCTION"

### I.3.2.1 Fonction "ORDONNANCER"

Le rôle essentiel de cette fonction est de générer l'organisation temporelle de la fabrication. Cette organisation doit permettre le respect des délais de fabrication fournis aux clients. Elle prend en compte, d'une part, les potentiels de production (matériel et humain) ainsi que leurs contraintes d'utilisations (réglage, entretien, horaire de travail,...) et, d'autre part, les disponibilités de matière première et d'outillage. Elle doit, de plus, pouvoir réagir à diverses perturbations (pannes, retard d'approvisionnement) tout en fournissant une répartition rationnelle du travail.

Partant des besoins (phase d'étude), des délais de fabrication souhaités et en fonction de la charge de production ainsi que des temps de réalisation, ce service fournit à la tâche de gestion des stocks les besoins affectés de délais de disponibilité (date de lancement prévue). Celle-ci l'informe, en retour, des disponibilités en matières premières et en outillage. L'ordonnanceur a donc les informations nécessaires pour déclencher efficacement la réalisation de l'outillage et le lancement de la production.

### I.3.2.2 Fonction "GERER LES STOCKS"

Cette tâche a pour fonction:

- de générer les demandes d'achats de matière première et d'outillage sous-traité,
- de les recevoir et d'en vérifier les qualités et quantités,
- de les fournir à la tâche de fabrication,
- de stocker et d'expédier les produits finis.

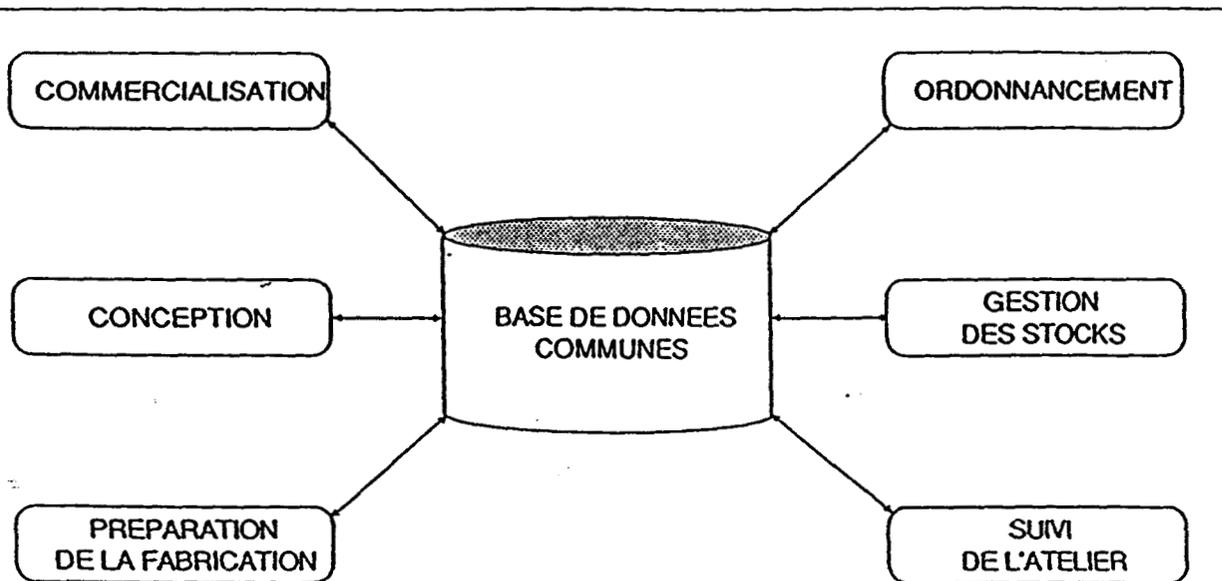


Figure 1.10 : organisation d'un système intégré  
autour d'une base de données commune

Une intervention sur un élément quelconque du processus peut donc être immédiatement prise en compte par les autres activités du système [6].

#### **1.4.1 Structure informationnelle [3]**

##### **1.4.1.1 Schéma global**

La complexité des informations à manipuler en conception ( par opposition aux applications de gestion) est telle que toute tentative de structuration des données doit être progressive et simplificatrice ( fig. 1.11) [7].

Nous définissons donc une structure verticale (ou structure d'agrégation) correspondant à une approche par niveau de détail de plus en plus fin.

De même, nous introduisons une structure horizontale correspondant à la décomposition du système en sous systèmes (domaines d'activité).

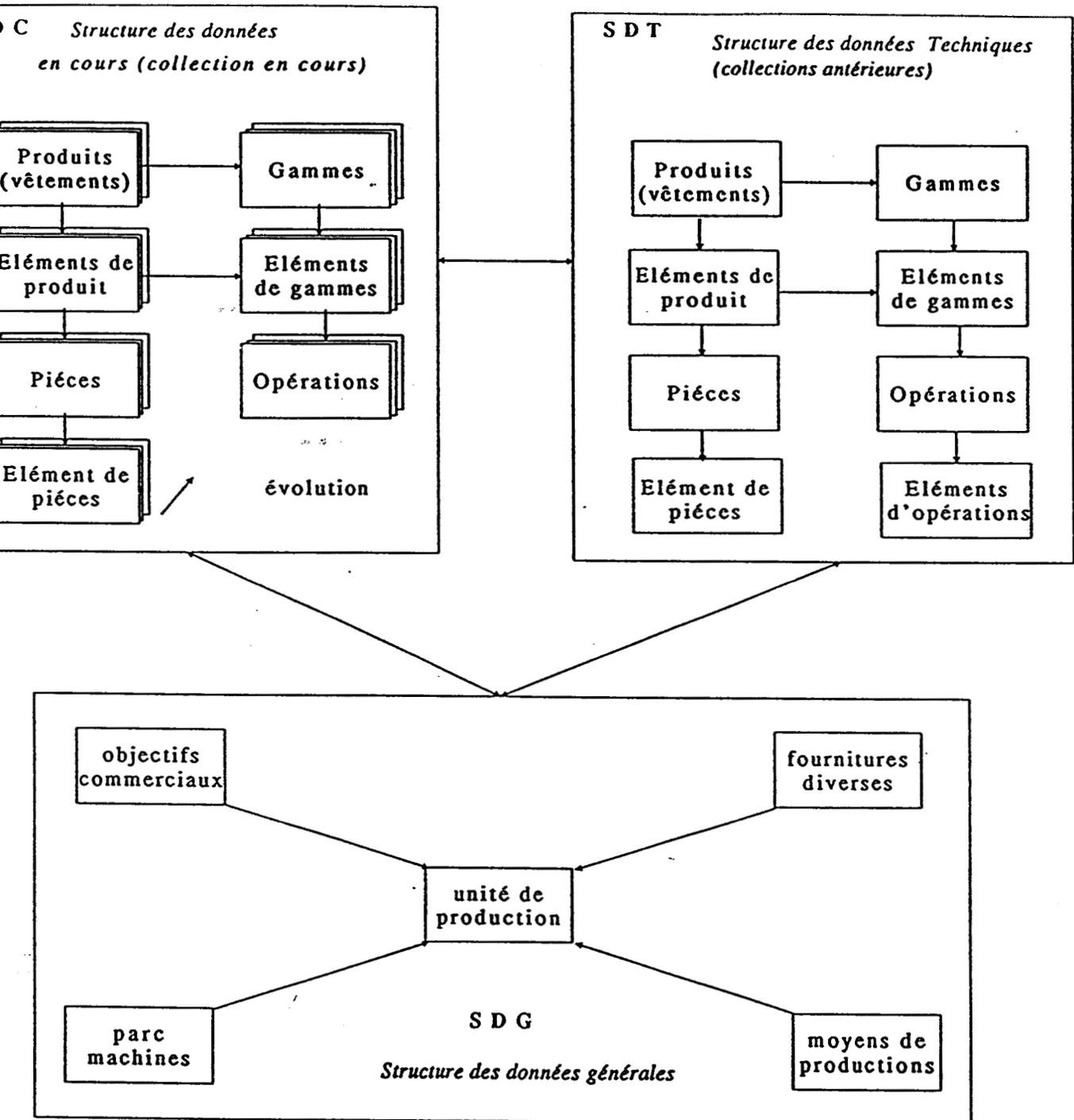


Figure I.11 : Décomposition fonctionnelle du système informationnel

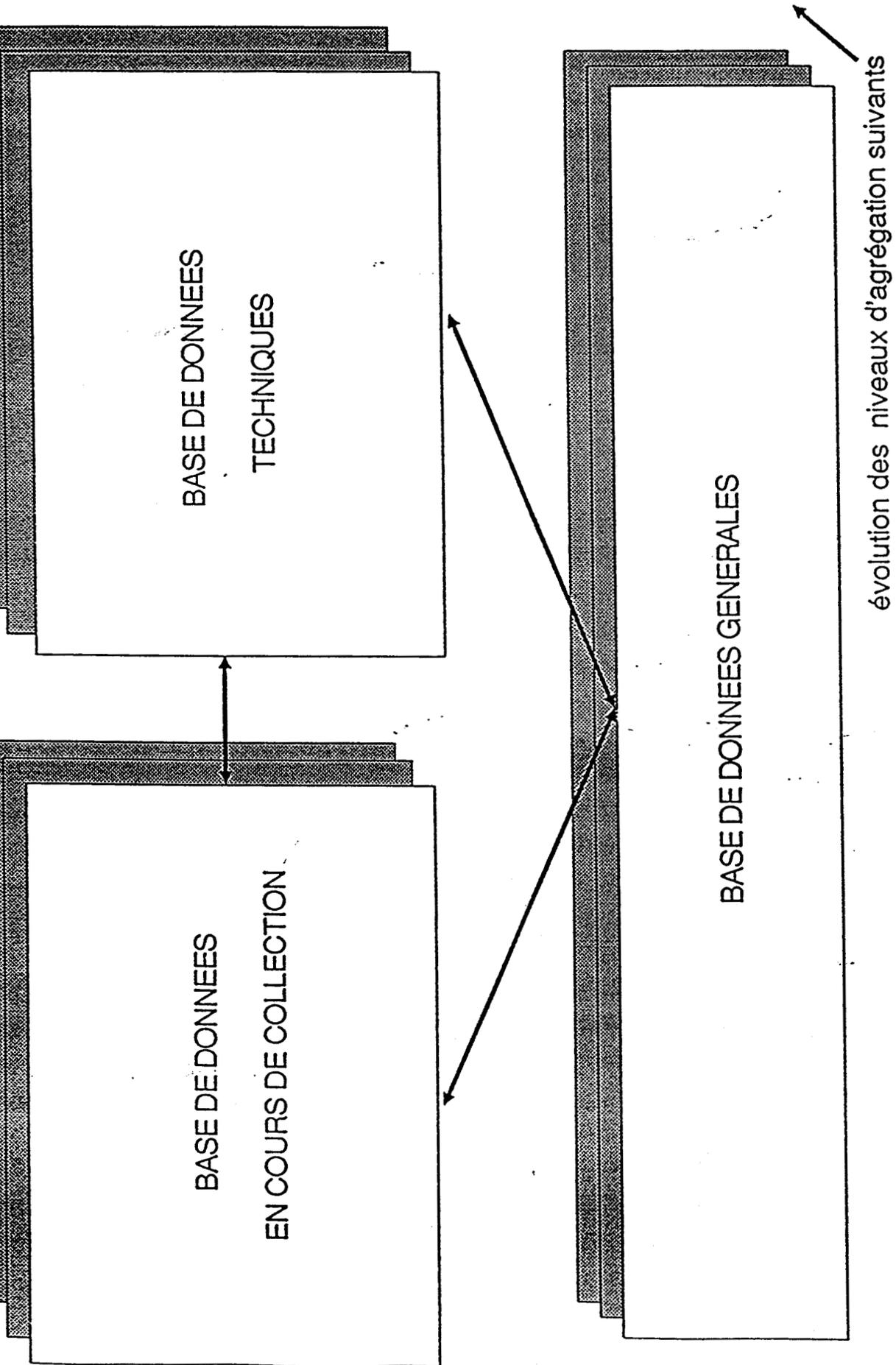
### 1.4.1.2 Structure horizontale

L'aspect de la structure horizontale des données est liée au niveau de décomposition verticale auquel on s'intéresse. L'examen du niveau le plus haut nous permet de distinguer les trois types de structures suivantes:

- une structure correspondant aux données caractérisant la collection en cours de conception (SDC). Elle évolue très rapidement durant le processus de conception. C'est dans cette structure que se retrouvent les aspects les plus dynamiques de la base de données.
- une structure correspondant aux données antérieures constituant l'acquis technique de l'entreprise (SDT). Il s'agit de toutes les informations concernant les anciennes collections. Elle évolue périodiquement, par remise à jour à la fin de chaque collection. Son rôle est de mémoriser de façon permanente la SDC. Elle ne nécessite pas un accès rapide mais elle ne doit nécessiter qu'un minimum d'espace mémoire.
- une structure correspondant aux données générales concernant l'entreprise SDG. Il s'agit là des informations caractérisant les moyens de production, la politique commerciale et technique à long terme de l'entreprise.

### 1.4.1.3 Structure verticale

La structure verticale des données représente une approche progressive des objets à manipuler. Les schémas des figures L.12a, L.12b, L.12c montrent les 3 premiers niveaux de cette structure. Celle-ci se présente sous la forme de niveaux d'agrégation décroissants. Nous n'avons représenté qu'une partie correspondant au niveau d'agrégation le plus élevé ( 5) jusqu'au niveau 3. De plus, pour le schéma relatif au niveau 3, nous n'avons représenté qu'une partie des interactions pour des raisons de clarté. Cette structure verticale représente une approche détaillée par niveaux ou structure d'agrégation



évolution des niveaux d'agrégation suivants

Figure 1.12a : Schéma partiel de la structure conceptuelle des données  
Niveau d'agrégation 5

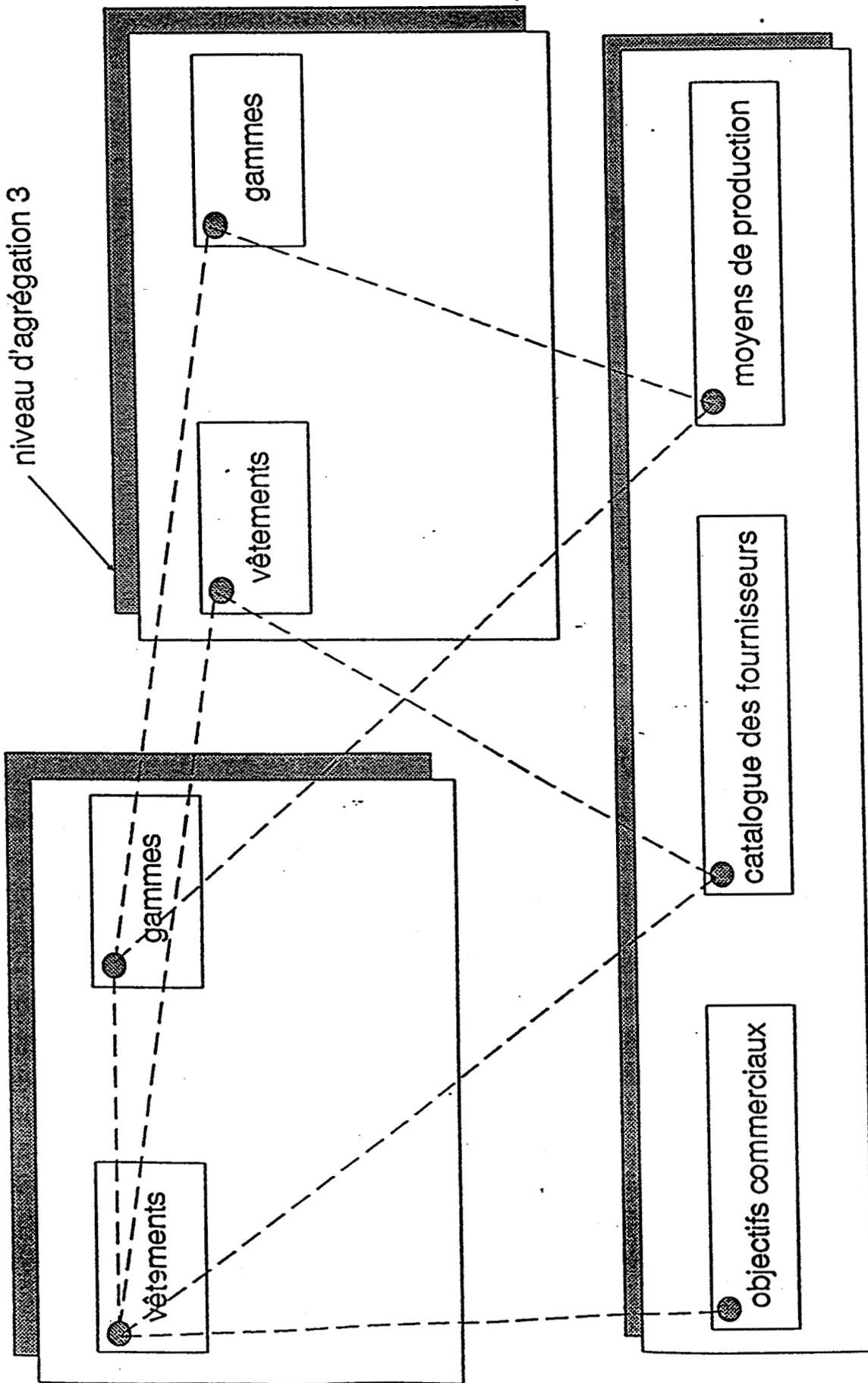


Figure 1.12b : Schéma partiel de la structure conceptuelle des données  
Niveau d'agrégation 4

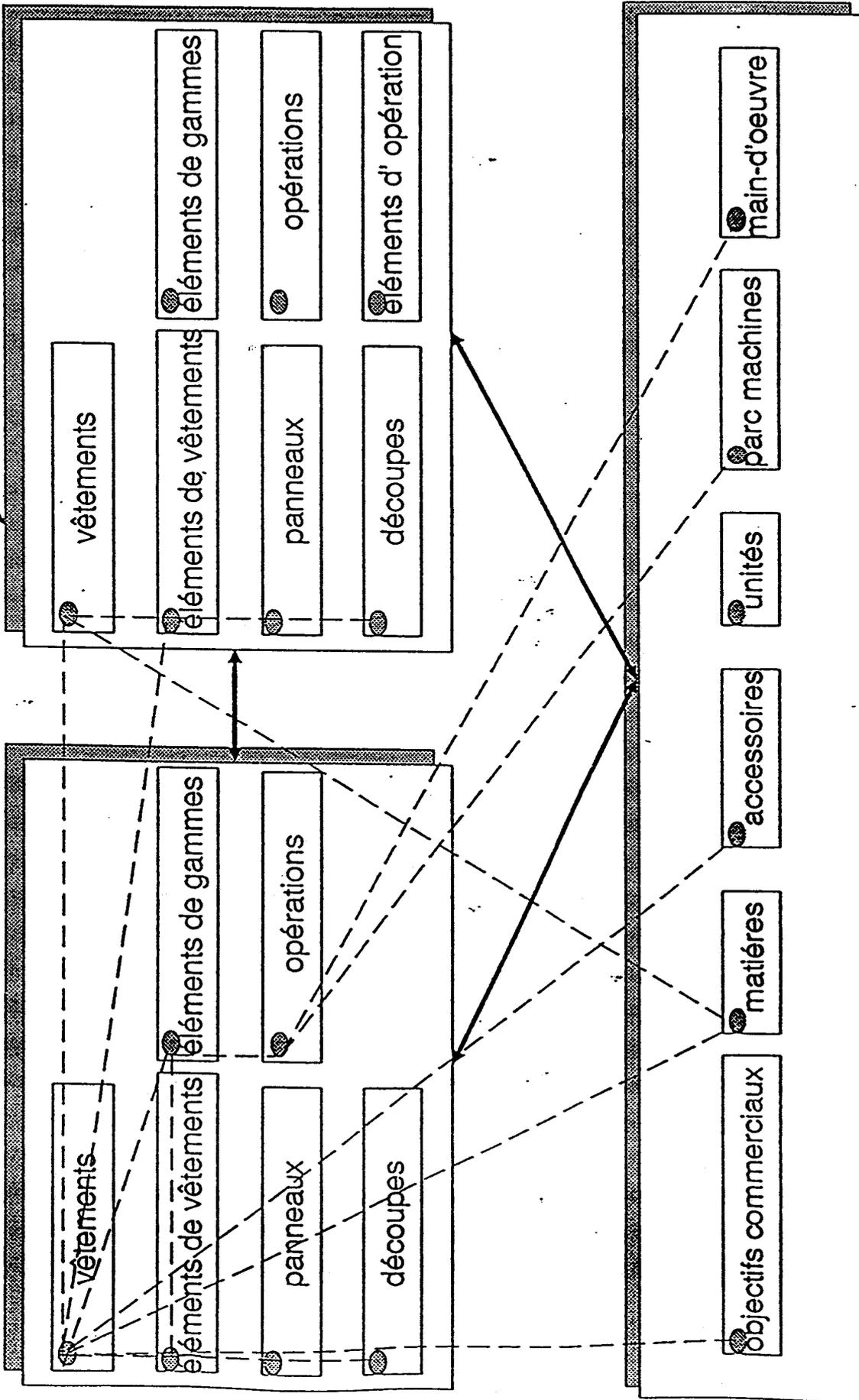


Figure 1.12c : Schéma partiel de la structure conceptuelle des données  
Niveau d'agrégation 3

La conception de vêtements est de type modulaire et par conséquent la gamme de fabrication l'est aussi. La succession des différentes étapes est la suivante:

- un vêtement est conçu par éléments de vêtement,
- un élément de vêtement, par panneaux,
- un panneau, par découpes,
- une gamme, par éléments de gammes
- un élément de gamme, par des opérations,
- une opération, par des éléments d'opérations.

C'est ainsi que le même sous-ensemble (élément de vêtement, panneau ou découpe) peut entrer dans la composition de plusieurs modèles de vêtements. Ce phénomène est représenté dans la structure verticale par une sous-structure relationnelle. Celle-ci se compose de liens dont les attributs expriment chaque particularité de réalisation rattachée au vêtement (localisation, matière, coloris,...).

Un autre aspect est constitué par l'utilisation très large des études techniques antérieures. En effet, il est reconnu que la conception d'un produit ne réserve qu'une faible part à la création, une grande partie se réduisant à une recopie.

La conception des vêtements n'échappe pas à cette règle, bien que la part de création y soit plus importante que pour beaucoup d'autres produits, du fait des exigences exprimées par le phénomène de la mode et des saisons (nécessité de deux collections par an). De multiples vêtements se réduisent ainsi à une simple recopie de produits existant archivés dans la base de données "techniques". Ce phénomène est représenté par une structure de lien entre la base de données "collection en cours" et la base de données "techniques".

Par ailleurs, il est tout aussi important d'élaborer plusieurs solutions pour chacune des étapes de conception du produit, et de les conserver jusqu'à la production des éléments d'évaluation qui permettront au concepteur de faire un choix.

Il peut être intéressant de conserver ultérieurement certaines de ces solutions, même lorsqu'elles n'ont pas été retenues. Les solutions dont le développement a atteint le stade industriel servent en outre, à constituer une bibliothèque de montage standard, en regroupant les opérations par familles d'éléments de produit.

C'est cet aspect du processus de conception que nous représentons dans la base de données par une arborescence d'objets, où l'existence d'attributs supplémentaires permet d'exprimer l'évolution du produit en cours d'élaboration.

D'autre part, l'élaboration des spécifications détaillées relatives au vêtement en cours de conception fait intervenir divers éléments nécessitant un approvisionnement extérieur: matière première ( tissus) et accessoires divers ( fermetures éclair). Les informations caractérisant l'ensemble de ces fournitures sont introduites au niveau du catalogue fournisseur.

De même, la détermination des temps opératoires ( temps technologiques et de manipulation) nécessite la prise en compte des caractéristiques de l'outil de production, ( aménagement du poste de travail, performances de l'outil de production, polyvalence de la main-d'oeuvre).

La structure de liens entre la base de données "collection en cours" et la base de données "générale" exprime la prise en compte de tous ces paramètres.

L'exploitation du catalogue des fournisseurs et des moyens caractérisant l'outil de production interviennent directement dans la définition du produit et de la gamme de fabrication. La structure de lien entre la base de données "techniques" et la base de données "générale" est conservée lors de son archivage dans la base de données "techniques".

La figure I.12 schématise la structure conceptuelle des données sur 5 niveaux d'agrégation.

## I.5 ORGANISATION DES FABRICATIONS : LA COUPE ET LE PIQUAGE

Après avoir défini la structure générale de l'entreprise, nous abordons la phase technique de fabrication [8]. En confection, on rencontre deux secteurs clés de la fabrication:

- la coupe,
- le piquage.

Les caractéristiques intrinsèques de ces deux secteurs sont différentes.

A) la coupe rassemble un ensemble de vêtements et la matière occupe une place relativement peu importante. C'est un regroupement de MODELES et de TAILLES. On travaille plusieurs vêtements à la fois. La nature du tissu n'a qu'une importance secondaire.

B) le piquage consiste en l'assemblage d'éléments relatifs à un seul vêtement. C'est un regroupement de MODELES et de TISSUS. On ne travaille qu'un vêtement à la fois. La taille n'a qu'une importance secondaire (assemblage des manches etc...)

Ces deux secteurs conduisent à une étude séparée de la fonction "LANCEMENT".

#### I.5.1 Fonction "LANCEMENT AU PIQUAGE"

Elle dépend du "LANCÈMENT DE LA COUPE", celui-ci assurant l'alimentation et la création d'un tampon ou réserve de stockage, le magasinage étant une variable interne à chaque entreprise (un jour de marge, deux jours,..).

#### I.5.2 Fonction "LANCEMENT DE LA COUPE"

Plusieurs facteurs entrent en jeu pour le lancement, on y retrouve essentiellement:

- A) la fonction "COMMERCIALISER":
- commandes saisonnières,
  - délais de livraison,

**B) la fonction "ORGANISATION DE LA PRODUCTION":**

- disponibilité des matières premières.
- disponibilité main d'oeuvre,
- disponibilité en matériel,
- engagement des unités de productions

**I.5.3 Lancement et suivi de fabrication**

Nous allons donner, dans ce paragraphe, les deux phases essentielles que l'on retrouve lors du lancement d'un produit (figure I.13).

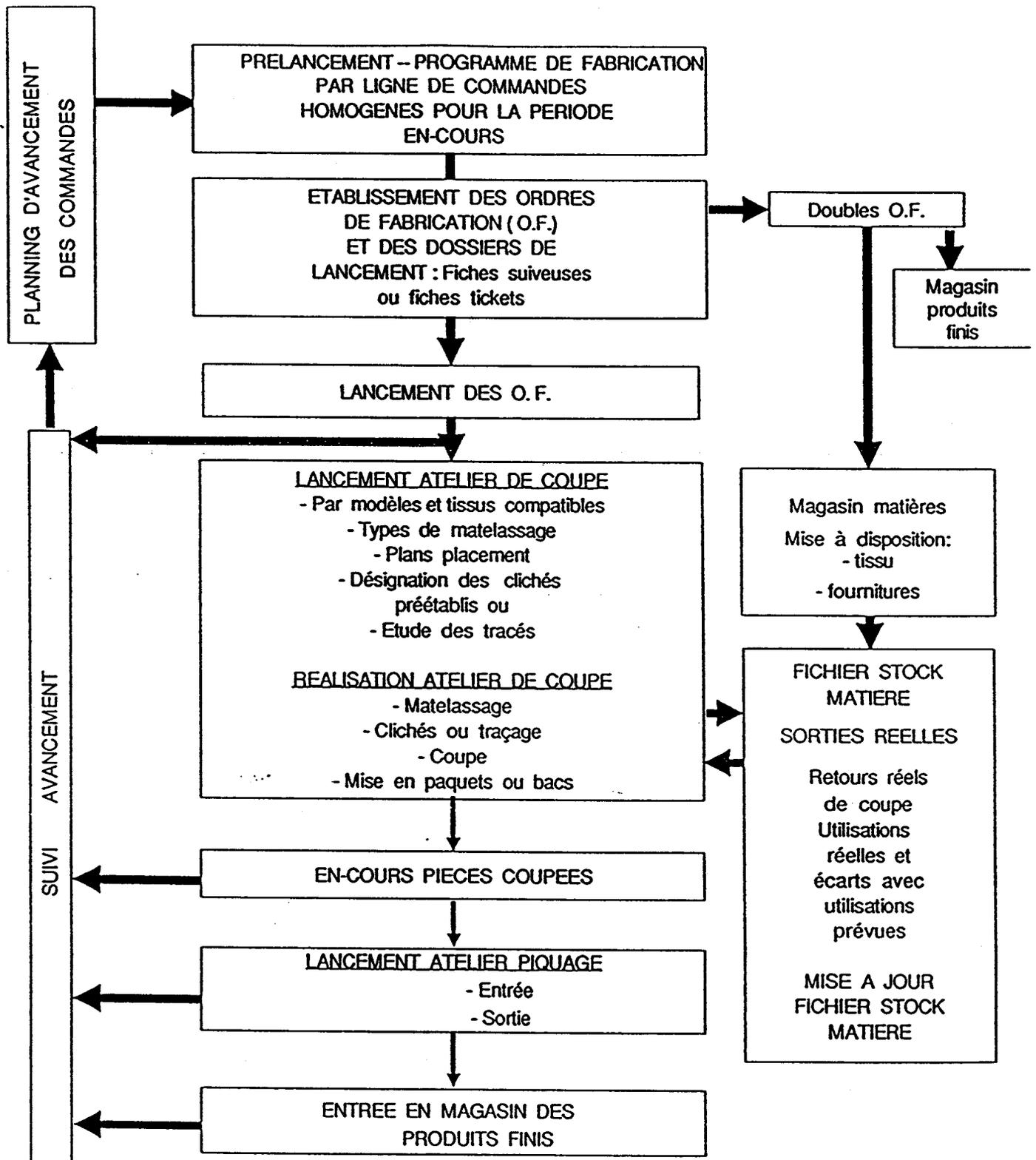
**I.5.3.1 Prélancement-programme de fabrication**

La fabrication est programmée d'après les délais émis par la fonction "ORDONNANCER", qui gère le carnet des commandes.

La programmation, c'est la mise en place d'une charge de travail fonction de la capacité matérielle du parc machine (capacité de production).

- la charge de travail est caractérisée par les commandes de fabrication, ordonnées par leur date d'exécution et par les besoins en ressources.
- la capacité de production est représentée par le volume de pièces qu'il est possible de produire pour le secteur piquage.

Le prélancement se fait par quinzaine et se présente sous la forme d'un tableau [8].



➔ : flèches informationnelles  
 ↓ : flèches transfert matière

Figure I.13 : lancement et suivi de fabrication

Exemple:

O. F.	Modèles ou n°	Matières	Coloris ou n°	Quantité pièces par tailles						Total	O. C.
				38	40	42	44	46	48		
1	Enok	Tergal	Noir	4	4	14	14	14	12	48	1
2	Enok	Tergal	Bleu	2	4	9	14	19	0	48	
3	Enok	Tergal	Blanc	3	3	8	12	14	12	52	
4	Julius	Laine	Brun	10	10	20	20	33	25	118	2
5	Julius	Laine	Gris	15	18	35	50	70	50	238	
6	Adrien	Laine	Brun	8	8	12	15	25	30	98	3
7	Adrien	Laine	Vert	10	10	12	15	20	30	97	
8	Marco	Lin	Beige	5	5	8	12	15	20	65	4
9	Marco	Lin	Blanc	5	5	6	12	16	22	66	

Pour ce pré-lancement de deuxième quinzaine du mois, à chaque ligne d' O. C. ( Ordres de Coupe) correspondent plusieurs O. F. ( Ordres de Fabrication).

### 1.5.3.2 Lancement

Au sein de cette activité on regroupe les documents pour la mise en fabrication des séries dans l'atelier. On y retrouve essentiellement:

- les ordres de coupe,
- les fiches de matelas,
- le placement.

#### A) ordre de coupe:

Il se présente sous la forme d'un tableau, dressé de différentes façons suivant les entreprises, qui comprend les renseignements suivants:

- type de vêtement ( robe, pantalon, veste, etc...),
- nom du modèle ou son numéro ( argus, ninon, etc..),
- numéro de référence de la matière à couper ainsi que son coloris,
- les tailles à traiter,
- les quantités par taille etc...

D'autres informations propres à chaque entreprise peuvent y figurer ( n° de fiche, mois de livraison, etc...).

### **B) la fiche matelas:**

Elle se présente sous la forme d'un tableau. Elle donne les renseignements suivants:

- n° d'identification du tableau d'ordre de coupe,
- matière
- le ou les modèles,
- la longueur du ou des tracés,
- nombre de plis par matière,
- emploi du tissu prévu,
- le schéma du type de matelas ( hauteur constante ou en escalier, etc..) sa longueur et le nombre de plis total.

### **C) le placement:**

Il correspond à la mise en place des pièces à découper sur un plan, de manière à minimiser la perte de matière première. Il se présente sous la forme d'un cliché en papier ou carton sur lequel est reproduit le positionnement des éléments du patron.

En conclusion, nous remarquerons le rapport qui peut exister entre les deux secteurs présentés. Le piquage est en étroite corrélation avec la coupe, mais s'autorise une marge de manoeuvre en fonction du stock présent dans le magasin.

Cet atelier se compose de plusieurs cellules de manière à prendre en charge tout le processus de confection c'est à dire:

- alimentation et inspection pour la détection de défauts de la matière première,
- découpe des pièces de vêtement selon un plan de placement en tenant compte de l'inspection,
- préhension et stockage des pièces,
- assemblage automatique à l'aide de deux robots,
- couture et évacuation.

Notre cellule intervient dans cet atelier, au niveau des opérations de découpe des plans de placement. Notre étude concerne donc une phase de la fonction technique de "FABRICATION".

## I.7 CONCLUSION

L'étude de l'organisation fonctionnelle et informationnelle d'une entreprise de confection nous a permis de mettre en évidence les fonctionnalités nécessaires à l'automatisation des tâches de conception et de préparation de la fabrication ainsi que la nature de leurs relations. L'étude est menée dans le cas d'un fonctionnement normal de l'entreprise. Il est cependant important d'envisager les situations à caractère plus exceptionnel.

La prise en compte de ces aspects met en évidence des liaisons informationnelles n'apparaissant pas en mode de fonctionnement courant. Ces "nouvelles" liaisons remettent en cause la perception générale du séquençement des diverses activités.

La réflexion engagée sur le concept d'information (paragraphe I.4) nous a amené à ne plus envisager l'automatisation des fonctions de l'entreprise comme l'implantation

d'outils logiciels indépendants mais comme la mise en place d'une structure commune permettant, à chaque outil logiciel, d'accéder à toutes les informations relatives à l'état du processus. De plus, nous avons mis en évidence un accroissement de la fiabilité du système par l'élimination de certains risques d'incohérences.

Nous avons ensuite présenté un cas particulier des fonctions d'étude et de préparation de la fabrication. L'étude de ces fonctions fait clairement ressortir les besoins algorithmiques relatifs:

- à l'utilisation des techniques de structuration (phase d'étude des vêtements, des pièces),
- à l'utilisation des techniques d'optimisation de placement et de découpe (phase de préparation de la fabrication).

Avant de présenter l'optimisation de la découpe (chap. III et IV), nous consacrerons le prochain chapitre à la présentation de la découpe et à la structuration des données.

## II.1 MAITRISE DE LA CONSOMMATION DE LA MATIERE PREMIERE

Ce problème est généralement lié à :

A) La complexité de la forme géométrique des pièces.

B) Au processus de découpe :

- Tronçonnage (bois,...)
- Cisailage (Tôles, ...)
- Découpe par contournage avec divers outils (laser, chalumeau, jet d'eau, ...)

C) La diversité des caractéristiques des matières premières utilisées.

- La forme : rectangulaire pour la tôle, bande de longueur constante pour le tissu et quelconque pour le cuir.
- Propriétés mécaniques rendant la matière anisotrope.

D) La variabilité du carnet de commande.

Ensemble de pièces à produire dans des délais et quantités variables.

## II.2 OPTIMISATION DE LA DECOUPE (APPROCHE CLASSIQUE)

On rencontre le problème de l'optimisation de la production de pièces manufacturées, dès qu'il y a nécessité de couper un ensemble de pièces de formes quelconques. Dans l'industrie de la confection, du cuir, de la cartonnerie, ce problème se présente sous la forme d'une minimisation du temps de production pour le processus de découpe.

Cette complexité est relative :

A) Au mode de fabrication (II.1.B)

B) A la surface disponible sur la machine-outil

- Zone d'évolution de l'outil inférieure à la surface de la matière.
- Mode de déplacement de la matière.

Les outils de F.A.O., C.A.O., C.F.A.O. mis à la disposition des différents modules, constituant la cellule de découpe, doivent être des outils communs et généraux. En effet les informations relatives à la fabrication d'un modèle de vêtement doivent être accessibles notamment pour :

- La gestion des données ( caractéristiques du carnet de commande).
- La gestion du dialogue homme/machine.
- La visualisation graphique des informations de placement.
- La manipulation interactive des pièces à placer.

Le module de découpe nécessite ces informations afin de permettre une aide à la décision pour la gestion de l'avance du tissu et pour la détermination du trajet optimum de l'outil.

Le travail que nous présentons a pour objet la réalisation de cet outil. C'est-à-dire un logiciel d'aide à la décision pour une gestion améliorée de la production de pièces de vêtements. Ceci nous conduit à la conception d'une application, pour une cellule de découpe par laser dans l'industrie de la confection, prenant en compte les contraintes temps réel et technologiques liées au mode de fabrication. Cette action nous conduit :

- à analyser, dans ce chapitre, les différents aspects du processus de découpe (découpe à une ou deux dimensions) d'une matière première (verre, carton, ...)

en pièces élémentaires en définissant les modèles mathématiques qui leur sont associés.

- à présenter les outils informatiques pour l'optimisation de la découpe.
- à établir une procédure algorithmique pour la réalisation d'un module d'aide à la décision pour l'avance du tissu et une procédure de calcul pour la minimisation du temps de découpe après avance.

## II.3 LA DECOUPE

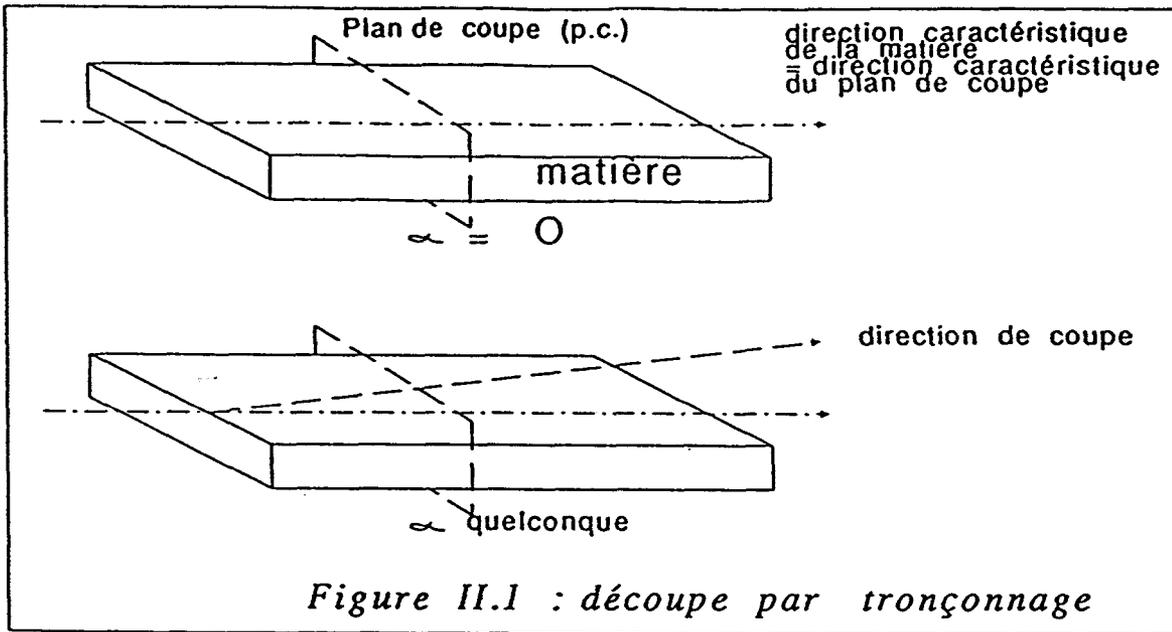
### II.3.1 Découpe mono-dimensionnelle

Le problème de la découpe mono-dimensionnelle se rencontre fréquemment dans différents secteurs industriels (chaudronnerie, industrie du bois, ...). De nombreux travaux ont conduit à l'élaboration d'algorithmes d'optimisation performants notamment :

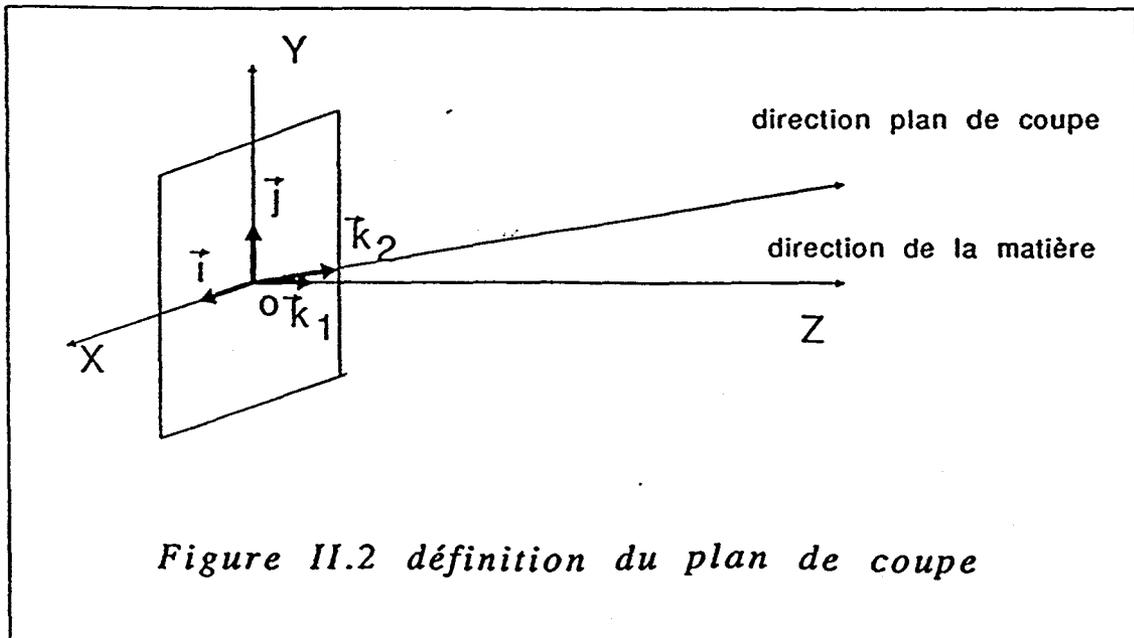
- Découpe de tubes à la sortie des laminoirs [10,11]
- Tronçonnage de planches en scierie [12]

#### II.3.1.1 Formulation

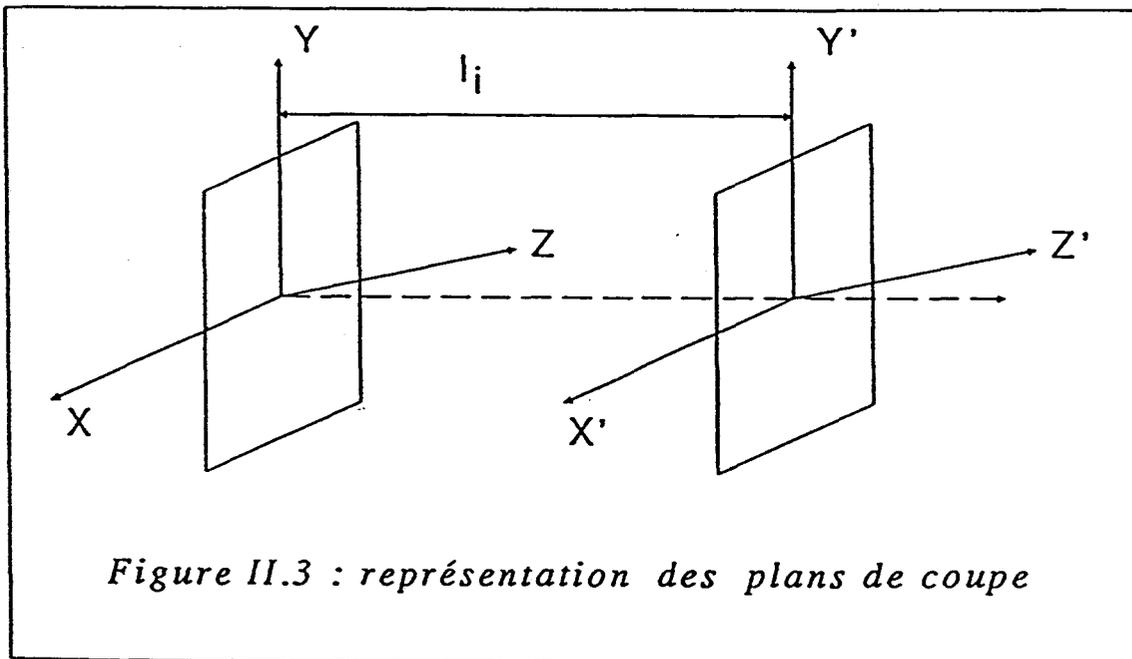
La commande de l'outil est indépendante du mode de découpe (sciage, cisailage, chalumeau, ...) et de la forme de la matière (plaque, fil, profilé, ...). Le déplacement de l'outil est toujours mono-directionnel. Le problème consiste à tronçonner suivant un plan (plan de tronçonnage) faisant avec une direction caractéristique de la matière un angle  $\alpha$  recherché (figure II.1).



A chaque plan de coupe ou de tronçonnage est associé un repère cartésien xoy et une direction caractéristique de la matière (figure II.2).



Le tronçonnage consiste à découper suivant deux plans définis par  $\alpha$  ( en fonction de la direction de la matière). Il est alors défini par la distance " $l_i$ " entre ces deux plans (figure II.3).



A chaque plan de coupe ou de tronçonnage est associé un repère cartésien  $xoy$  et une direction caractéristique de la matière (figure II.2).

On associe à chaque objet  $O_i$  à couper, le triplet  $(l_i, \alpha, n_i)$ , où:

- $l_i$ : caractérise la distance entre deux plans de coupes,
- $\alpha$ : angle d'inclinaison du plan de coupe (outil de coupe), par rapport à la direction caractéristique de la matière,
- $n_i$ : désigne le nombre de reproductions à réaliser pour la pièce  $i$ .

La matière dans laquelle on réalise la découpe des objets  $O_i$ , est fixée et nous admettons qu'elle est délimitée par deux plans de coupe extrêmes. Elle est alors caractérisée par la distance  $L$  entre ces deux plans. Par ailleurs, nous disposons en stock de plusieurs blocs matière susceptibles d'être utilisés pour la réalisation des  $O_i$  objets. A chaque bloc correspond un indice  $k$  et une longueur  $L_k$ .

Le problème consiste à exécuter dans un même bloc matière,  $n_{ik}$  objets de longueurs  $l_i$  différentes, tout en respectant le carnet de commande correspondant à  $n_i$  objets à obtenir sur les  $k$  blocs matière..

Toutefois, on suppose généralement qu'ils sont tous caractérisés par le même angle  $\alpha$ .

### II.3.1.2 Modélisation du problème mono-dimensionnel:

L'objectif consiste alors à découper, dans un ensemble de blocs matière  $k \in K$   $n_i$  objets  $O_i$  de longueur  $l_i$  avec  $i \in I$ . Pour que le processus de découpe d'une matière  $k$  soit possible, la contrainte suivante doit être réalisée :

$$\sum_{i \in I} n_{ik} \cdot l_i \leq L_k \quad \forall k \in K \quad (\text{II.1})$$

$$n_{ik} \in \mathbb{N}^* \quad \forall i, \forall k$$

Cette condition exprime que la somme des longueurs des objets coupés est inférieure ou égale à la longueur de la matière  $k$ . Après découpe de la matière  $k$ , la chute  $C_k$  s'exprime :

$$C_k = L_k - \sum_{i \in I} n_{ik} \cdot l_i \quad (\text{II.2})$$

Pour réduire la consommation de matière globale, nous devons alors rechercher, pour l'ensemble des blocs de matières utilisés, la quantité d'objets à réaliser telle que :

On tronçonne le bloc de matière par le plan de coupe :

- AB pour obtenir  $O_1$  et  $\bigcup_{i=2}^5 O_i$

- CD pour obtenir  $O_2$  et  $\bigcup_{i=3}^5 O_i$

- EF pour obtenir  $O_3$  et  $\bigcup_{i=4}^5 O_i$

- GH pour obtenir  $O_4$  et  $O_5$

### II.3.2.2 Modélisation

Le problème à résoudre se ramène à un problème plan dit de Gilmore Gomory . La matière  $k$  à découper se présente sous forme d'un parallélépipède  $R_k (L_k, W_k)$  de longueur  $L_k$  et de largeur  $W_k$  pour  $k \in K$  d'épaisseur  $E_k$

Les objets  $O_i$  (pièces) à obtenir sont des parallélépipèdes  $r_i (l_i, w_i)$  de surface  $l_i \cdot w_i$  pour  $i \in I$  de hauteur  $E_k$ . L'expression de la matière non utilisée lors de la fabrication de  $n_{ik}$  objets  $O_i$  dans une matière  $R_k$  est égale à :

$$L_k \cdot W_k - \sum_{i \in I} [ n_{ik} \cdot (l_i \cdot w_i) ] \quad (\text{II.6})$$

avec  $n_{ik} \in \mathbb{N}^*$

Cette réalisation peut nécessiter l'utilisation de plusieurs blocs matières, l'expression de chutes est alors:

$$\sum_{k \in K} L_k \cdot W_k - \sum_{k \in K} \sum_{i \in I} n_{ik} \cdot (l_i \cdot w_i) \quad (\text{II.7})$$

avec:

-  $n_{ik} \in \mathbb{N}^*$  pour  $i \in I$  et  $k \in K$

-  $n_{ik} \leq n_i$  pour  $i \in I$  et  $k \in K$  (II.8)

On caractérise l'ensemble "J" des configurations possibles, par un ensemble de pièces placées en série sur la longueur  $L_k$  ou la largeur  $W_k$ .

Le problème pour une découpe guillotine se formule alors:

$$(I.10) \quad \left[ \begin{array}{l} \min_K \quad [ \sum_{k \in K} L_k \cdot W_k - \sum_{k \in K} \sum_{i \in I} n_{ik} \cdot (l_i \cdot w_i) ] \\ \text{Sous les contraintes :} \\ - \sum_{k \in K} n_{ik} = n_i \quad \text{pour } i \in I \\ - n_{ik} \leq n_i \quad \text{pour } i \in I, k \in K \\ - \sum_{i \in \alpha_j} l_i + \sum_{i \in \beta_j} w_i \leq L_k \quad \text{pour } j \in J, k \in K \\ - \sum_{i \in \alpha_j} w_i + \sum_{i \in \beta_j} l_i \leq W_k \quad \text{pour } j \in J, k \in K \\ - n_{ik} \in N^* \quad \text{pour } i \in I, k \in K \end{array} \right.$$

avec:

-  $\alpha_j$ : ensemble des pièces pour lesquelles  $l_i // L_k$  pour  $i \in I$  et appartenant à la  $j^{\text{ème}}$  configuration ( $j \in J$ ) et  $k \in K$ ,

-  $\beta_j$ : ensemble des pièces pour lesquelles  $w_i // W_k$  pour  $i \in I$ , appartenant à la  $j^{\text{ème}}$  configuration ( $j \in J$ ) et  $k \in K$ ,

On constate que ce problème d'optimisation se formule sous la forme d'un problème bilinéaire en nombres entiers [16,17].

#### II.4 STRUCTURE DE DONNEE

Les deux paragraphes précédents étaient consacrés à la présentation de deux modes de découpe ainsi qu'à leur formulation mathématique. Dans ce paragraphe, nous présentons les outils informatiques issus du module de placement et généralisés au

niveau de la cellule de découpe. Ces outils seront nécessaires au module d'optimisation du processus de découpe. Nous nous proposons d'en donner un aperçu et d'en retirer la partie nécessaire au problème d'optimisation.

## II.4.2 Structuration de données

### III.4.2.1 Architecture

Comme tout système de CAO, le module de placement nécessite l'utilisation d'un gestionnaire de données. Il faut distinguer deux organisations :

A) L'une permettant le stockage :

- des objets (pièces) à réaliser.
- des matières susceptibles d'être utilisées.
- des plans d'imbrications retenus ultérieurement.
- des données technologiques de réalisation et d'approvisionnement.

B) L'autre assurant le support des données pendant le processus d'imbrication.

Chacune d'elles doit gérer des informations de type :

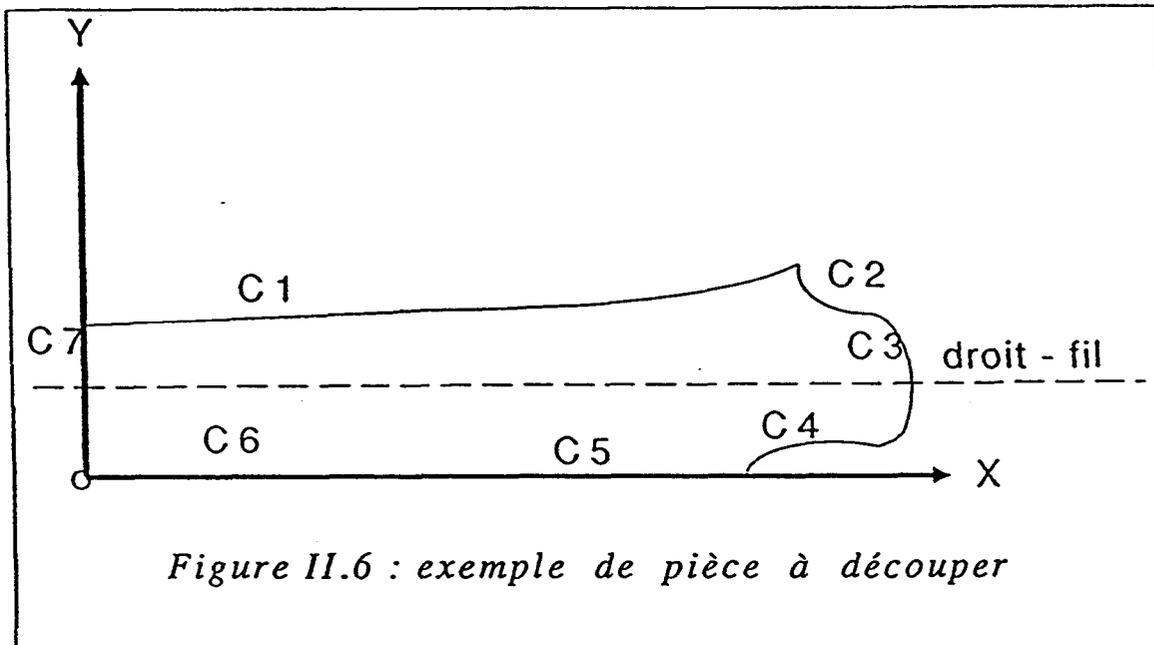
- géométrique : longueur de pièces, de matière, ...
- graphique pour la visualisation des pièces et des plans d'imbrications.
- technologique : droit-fil, couleur, motif, ...

Dans l'exposé suivant, nous ne donnerons que les caractéristiques de données communes au placement et à la découpe.

### II.4.2.2 Données communes au placement et à la découpe

Un objet est une entité qui supporte des informations de types technologiques, géométriques et graphiques (figure II.6).

La principale base de donnée nécessaire à la découpe est celle qui contient la position des pièces dans le référentiel d'un plan d'imbrication, ainsi que le contour. Nous utiliserons donc le modèle géométrique issu du module de placement.



#### Remarque:

L'acquisition de chaque objet est obtenue par binarisation. Ensuite, par une technique d'interpolation polynomiale on détermine le polynôme caractéristique de tous les contours élémentaires composant l'objet. Enfin, on construit la forme initiale par une autre interpolation, nous permettant d'acquérir tous les points caractéristiques ainsi que la position des contours par rapport au référentiel. Ces différentes étapes sont présentées sous forme de modèle géométrique et de traitement.

#### II.4.2.3 Modèle géométrique

Chaque objet possède un référentiel relatif. Toute ses coordonnées sont placées sur ce référentiel. Le contour de l'objet est décomposé en éléments simples, on définit ainsi une succession de contours élémentaires  $C_i$  caractérisés par un polynôme paramétrique  $p_i(u)$  de Bézier qui a pour avantages [18]:

- de modéliser la forme de l'objet réel avec un nombre réduit de points (polygone).
- d'assurer la continuité aux points de raccordements entre éléments fonctionnels composant le contour de l'objet.

La représentation de l'objet étudié, associée à la structure de stockages, se fait de telle manière que les contours délimitant la matière soient orientés de façon que cette matière (au domaine plein) se trouve sur la gauche d'un observateur se déplaçant le long de ces contours, dans le sens trigonométrique.

#### Définition de l'intérieur d'un domaine[9]

La frontière extérieure  $C_e$  est parcourue dans le sens trigonométrique, les autres ( $C_{i1}$ ,  $C_{i2}$ ) dans le sens inverse (figure II.7).

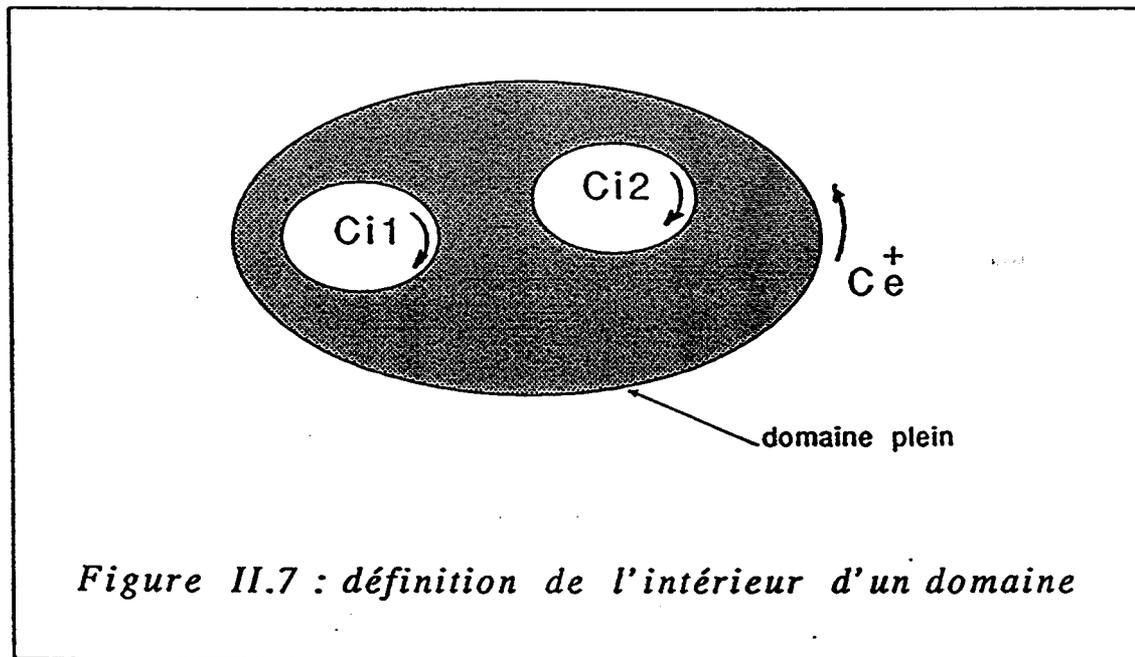


Figure II.7 : définition de l'intérieur d'un domaine

#### Rappel:

L'interpolation polynomiale ( ou polynômes de Bezier) a pour objectif d'établir un modèle mathématique pour représenter le contour associé à un objet dans l'espace bi-dimensionnel. Elle est utilisée avec succès dans plusieurs secteurs de l'industrie ( aéronautique, automobile, etc..).

Les courbes de Bézier constituent la représentation géométrique d'un polynôme paramétrique dont l'expression mathématique est donnée par:

$$(A1) \quad \left[ \begin{array}{l} p(u) = \sum_{i=0}^n P_i \cdot B_{i,n}(u) \\ \text{avec } B_{i,n}(u) = \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \\ \text{où:} \\ \quad - \binom{n}{i} = n! / i! (n-i)! \\ \quad - u \in [0, 1] \\ \quad - P_i \text{ représentent les points de contrôle,} \\ \quad \quad i \in \{1, \dots, n\} \end{array} \right.$$

L'expression (A1) peut donc s'écrire:

$$(A2) \quad \left[ \begin{array}{l} p(u) = \sum_{i=0}^n P_i \cdot \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \\ \text{ou : } p(u) = \sum_{i=0}^n k_i \cdot u^i \cdot (1-u)^{n-i} \\ \text{où :} \\ \quad - k_i = P_i * \binom{n}{i} = P_i \cdot n! / (n-i)! i! \\ \quad - u \in [0, 1] \text{ et } i \in \{1, \dots, n\} \end{array} \right.$$

### Exemple:

Nous traiterons ici, un exemple pour examiner le comportement de la courbe résultante, d'un objet géométrique figure II.6. Soit le contour C1 correspondant au cas d'une courbe (n=2):

$$p(u) = \sum_{i=0}^2 k_i \cdot u^i \cdot (1-u)^{2-i}$$

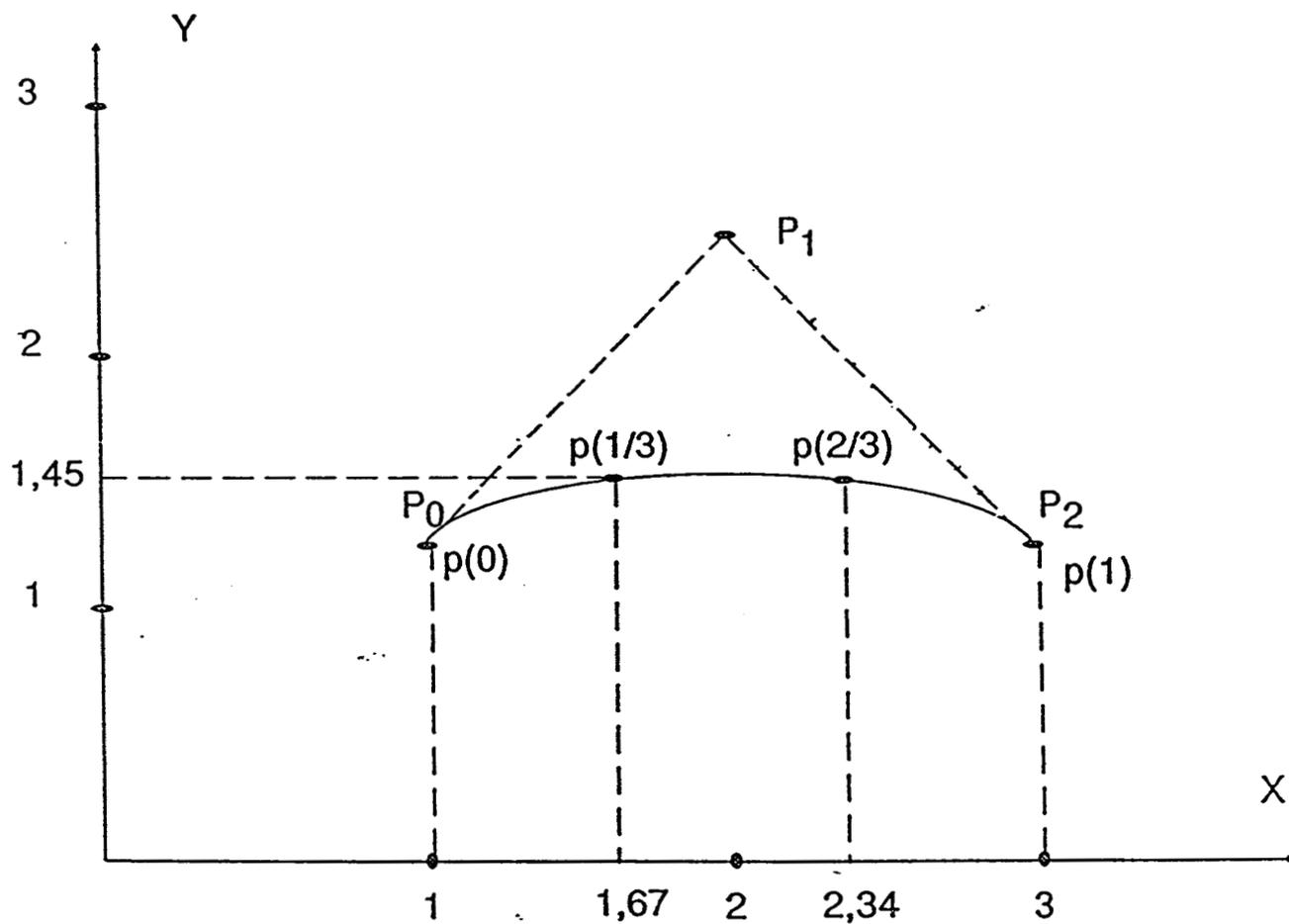


Figure II.8 : Exemple de construction de courbe de Bezier ( $n=2$ )

en développant on trouve:

$$p(u) = k_0 \cdot (1-u)^2 + k_1 \cdot u \cdot (1-u) + k_2 \cdot u^2$$

ou, sous une autre forme:

$$p(u) = (k_0 - k_1 + k_2) \cdot u^2 + (k_1 - k_2) \cdot u + k_0$$

$$p(u) = (P_0 - 2 \cdot P_1 + P_2) \cdot u^2 + 2 \cdot (P_1 - P_2) \cdot u + P_0$$

Les points remarquables de la courbe sont:

$$p(0) = P_0$$

$$p(1/3) = 1/9 \cdot (4 \cdot P_0 + 4 \cdot P_1 + P_2)$$

$$p(2/3) = 1/9 \cdot (P_0 + P_1 + 4 \cdot P_2)$$

$$p(1) = P_2$$

En connaissant les coordonnées de chaque point de contrôle  $P_0$ ,  $P_1$ ,  $P_2$ , nous approximations la partie polygonale par la courbe correspondante Figure II.8. La courbe en trait plein approxime grossièrement la ligne polygonale en trait pointillés.

#### Remarque:

Plus on calcule de points intermédiaires et meilleure sera l'approximation. Les paramètres à mémoriser sont de deux types:

- degré (n) du polynôme,
- coefficients du polynôme.

#### II.4.2.4 Représentation de stockage

L'ensemble des informations décrivant l'objet peut être représenté sous forme d'une structure arborescente. L'objet étudié est constitué de n éléments fonctionnels caractérisés par des polynômes  $p_i(u)$  ( $i = 1, \dots, n$ ) qui définissent le modèle géométrique associé à l'objet stocké.

Au niveau de ce modèle, nous avons fait apparaître la valeur du paramètre  $u$  (0,1,2,...,m) qui détermine le polynôme  $p_i(u)$ .

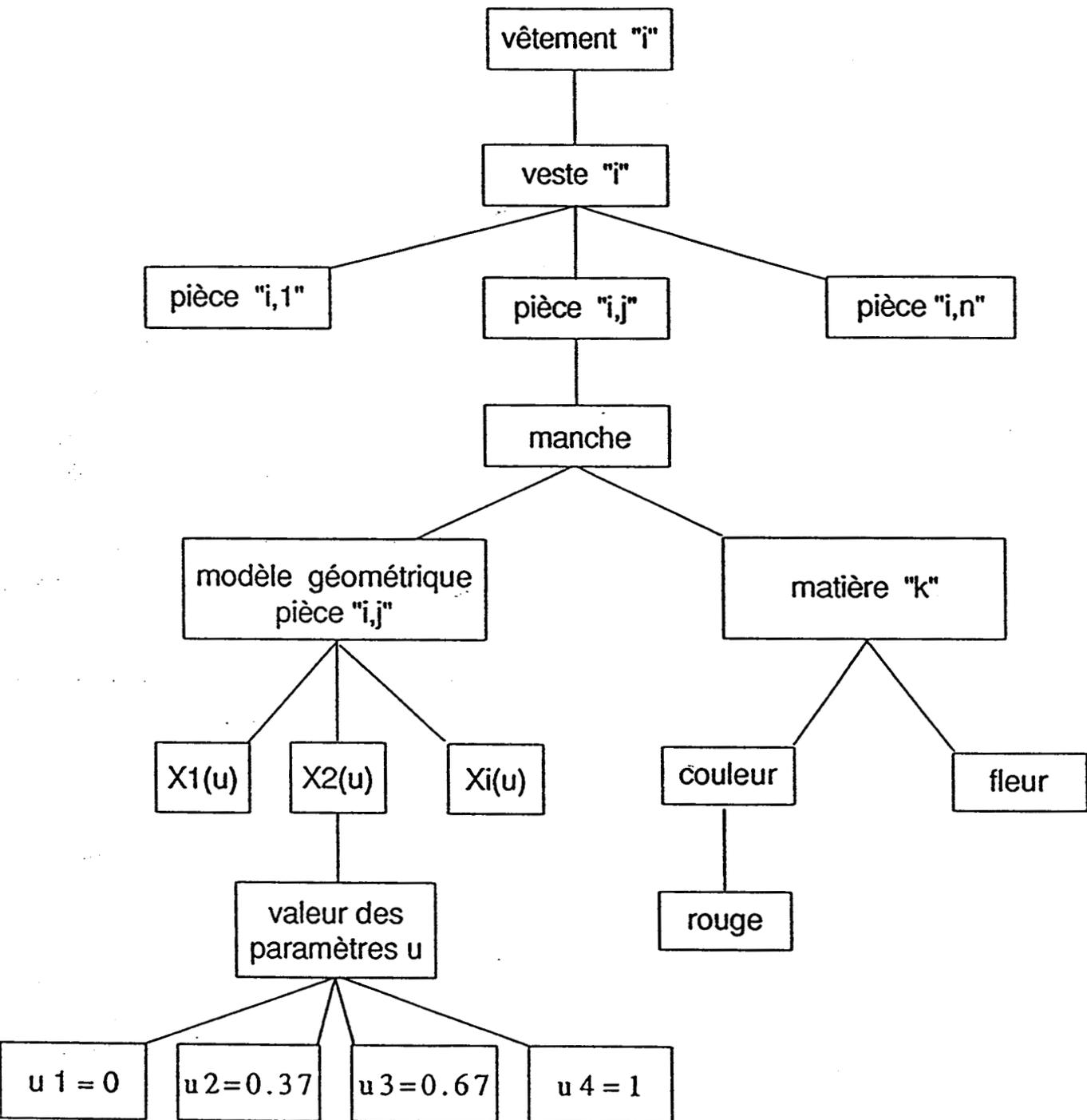


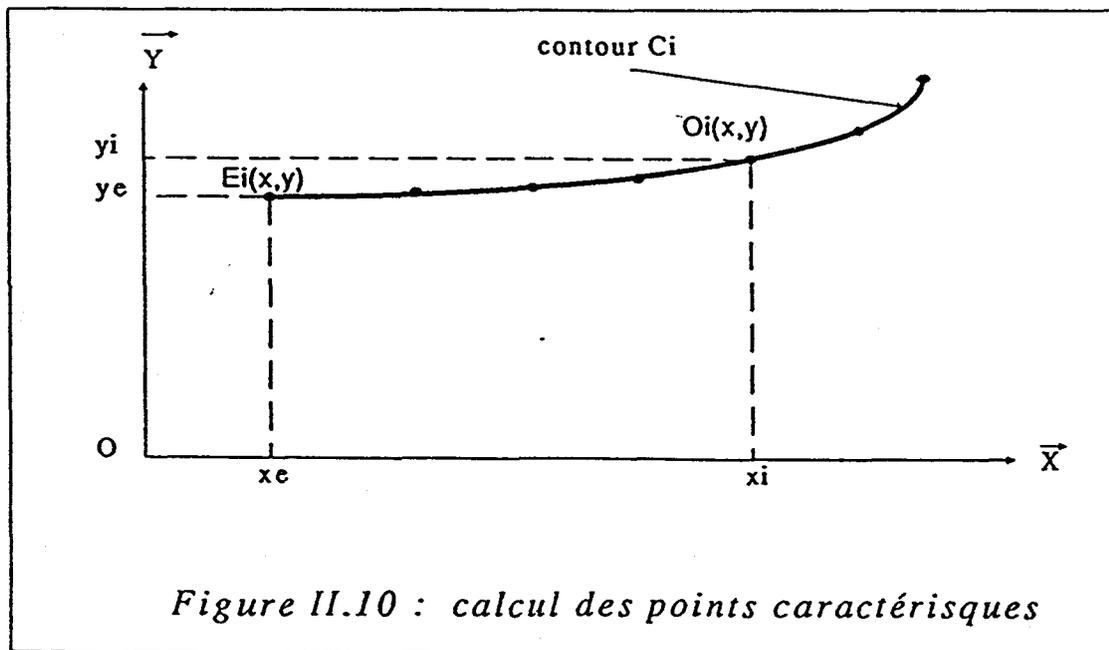
Figure II 9 : modèle géométrique

**Exemple:**

La figure II.9 donne le type de représentation associé à notre exemple précédent pour lequel  $n=6$ . Pour cette représentation, nous avons fait apparaître la valeur du paramètre  $u$   $(0,1,2/3,1/3)$  qui détermine le polynôme  $p_3(u)$ .

**II.4.2.5 Modèle de traitement**

Le modèle de traitement, associé à la géométrie des pièces, est obtenu après interpolation linéaire du polynôme de Bézier. On détermine, ainsi, la succession de points relative au contour binarisé et la position de celui-ci (figure II.10).



On en déduit alors la structure arborescente (figure II.11)

Ayant présenté la structure des informations caractérisant les objets à manipuler, nous nous intéressons plus particulièrement au processus de découpe de ces objets.

## **II.5 DECOUPE DE PIÈCES DE VÊTEMENTS**

Au paragraphe précédent, nous avons présenté les données nécessaires à l'étude de la découpe. Ces bases de données seront employées afin de conserver un mode informationnel commun à tous les modules composant la cellule de découpe. Dans ce paragraphe nous allons présenter les différents aspects techniques relatifs au mode de découpe (laser).

Nous présenterons le problème de l'optimisation de la découpe sous la forme de deux problèmes. L'un concernera la recherche d'un placement optimal des pièces de vêtements sur chaque patron, l'autre correspond à la gestion de l'avance du tissu par une aide à la décision.

### **II.5.1 La cellule de découpe**

La découpe de pièces de tissu (uni ou à motifs) occasionne au bureau des méthodes, un travail d'organisation important pour faire face aux nombreuses questions que soulève la mise en oeuvre des ateliers de coupe. Il faudra en effet résoudre simultanément les problèmes suivants:

- choix des technologies de coupes,
- placement optimal des patrons et minimisation des pertes,
- gestion optimale de la production,
- mise en oeuvre du matériel (principal ou périphérique),
- gestion du personnel,
- etc...

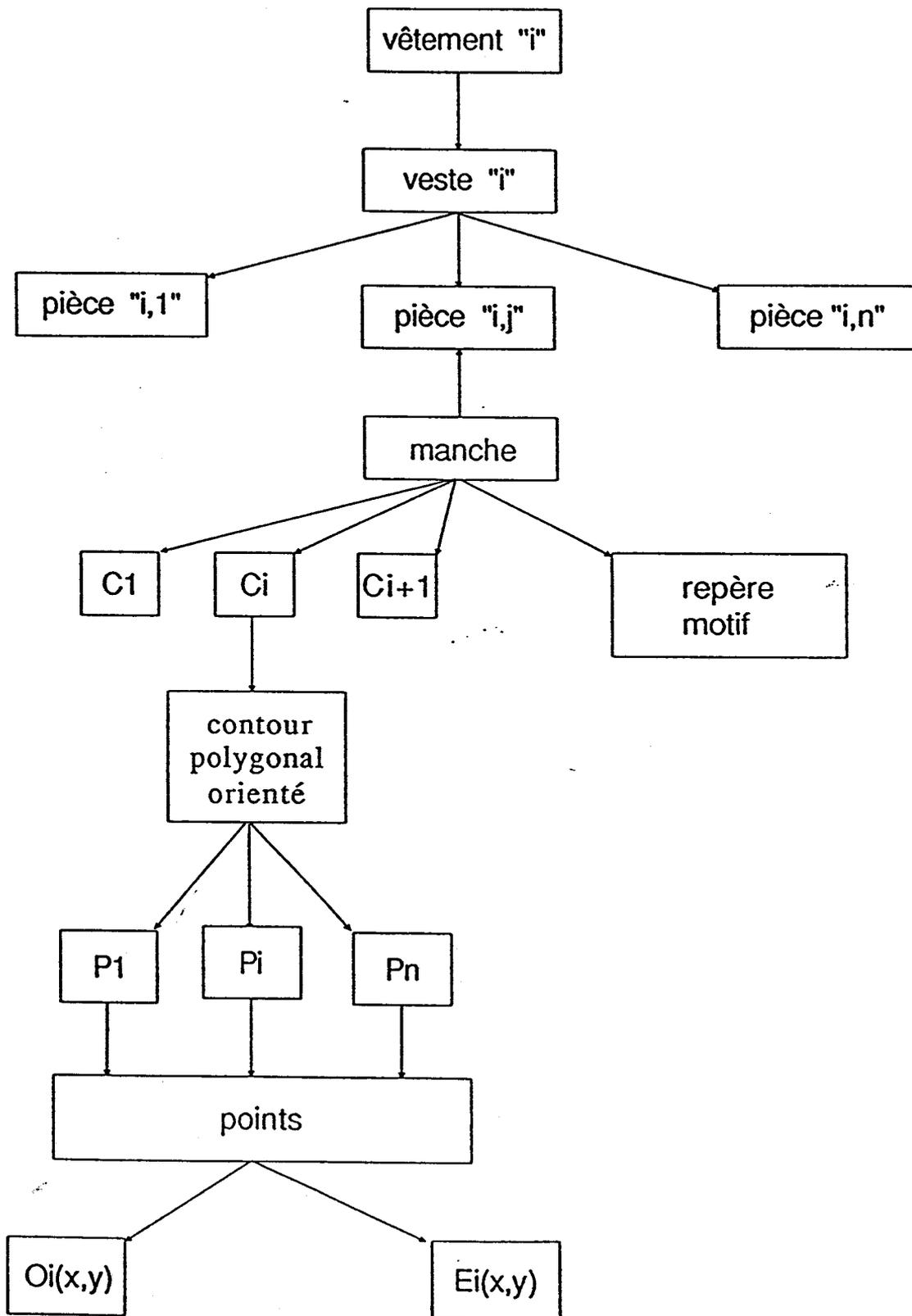


Figure II.11 : modèle de trait

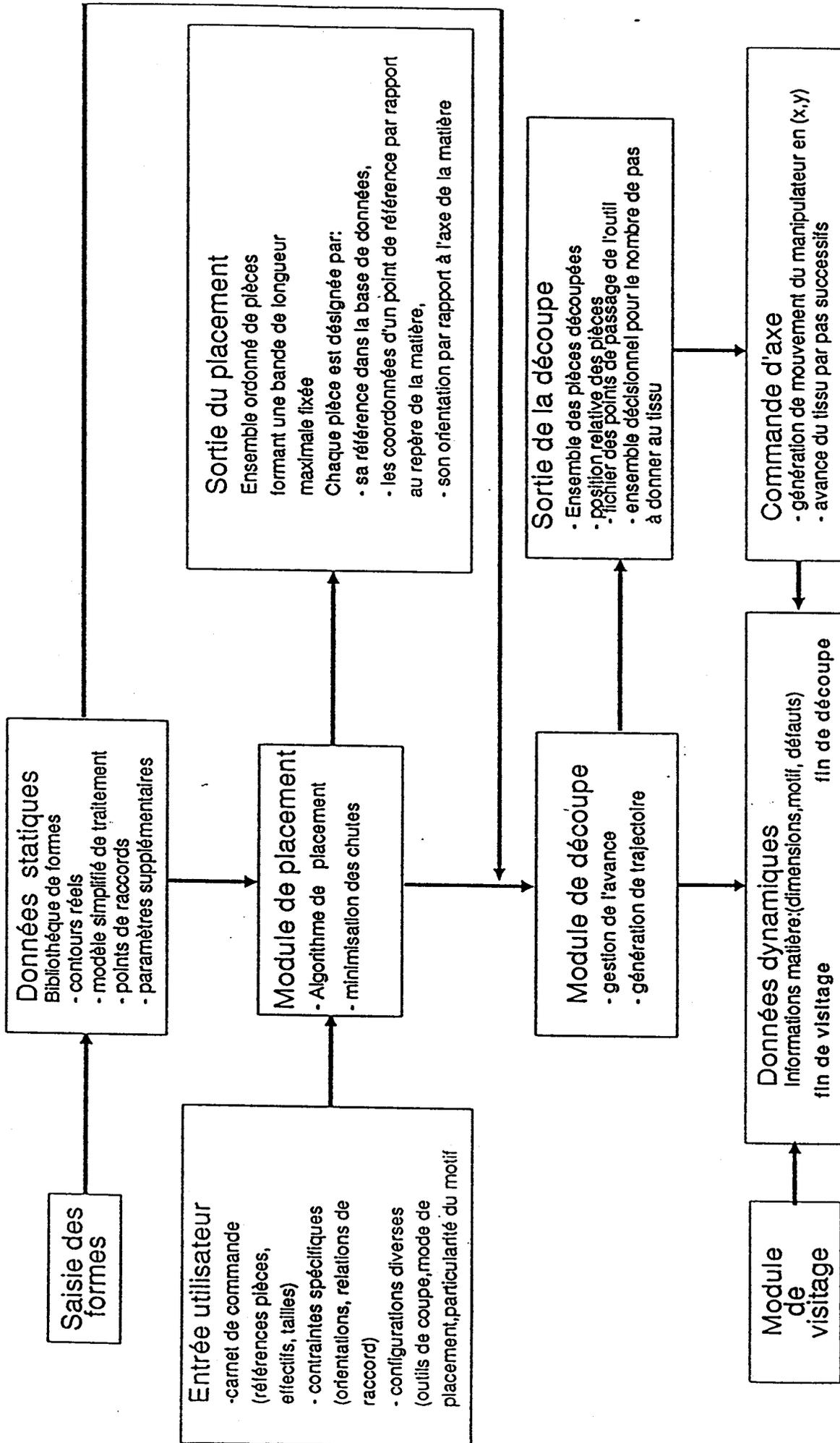


Figure II.12 : organigramme de traitement des pièces dans la cellule flexible de coupe

Dans le chapitre suivant, nous nous attacherons à présenter une formulation du problème de la gestion du processus de découpe au laser. Partant du plan d'imbrication des pièces de vêtement, nous montrerons que ce problème nécessite, de par sa complexité, une approche totalement différente du problème d'ordonnancement de tâches sur machine unique.

## INTRODUCTION

Nous avons vu ( chap II ), que le placement et la découpe sont présentés comme un problème d'optimisation unique, de par la technologie de coupe employée (guillotine, cisailage ). Ils sont formulés par un programme linéaire et résolus par la méthode des coupes de GOMORY .

Cependant, l'application directe à notre cellule, des méthodes "classiques" d'optimisation nous semble difficile à réaliser. En effet, les nouvelles techniques de coupe ( laser, jet d'eau ) commandées par ordinateur, nous ont permis de dissocier ces deux problèmes.

C'est pourquoi, dans ce chapitre, nous proposons une approche du problème ainsi qu'une formulation. En ce qui concerne la formulation elle sera abordée de deux façons, à savoir:

- une formalisation basée sur la programmation dynamique, en utilisant une représentation par graphe séquentiel, nous conduisant à un programme mathématique en nombres entiers,
- enfin, une formalisation basée sur un graphe de type réseau, ce qui nous permettra d'obtenir un programme linéaire en variable bivalentes.

D'autre part, au cours de notre formulation, nous ferons intervenir une fonction de coût dont l'obtention fera l'objet du chapitre suivant. En effet, la détermination de cette valeur constitue en elle-même un autre problème d'optimisation.

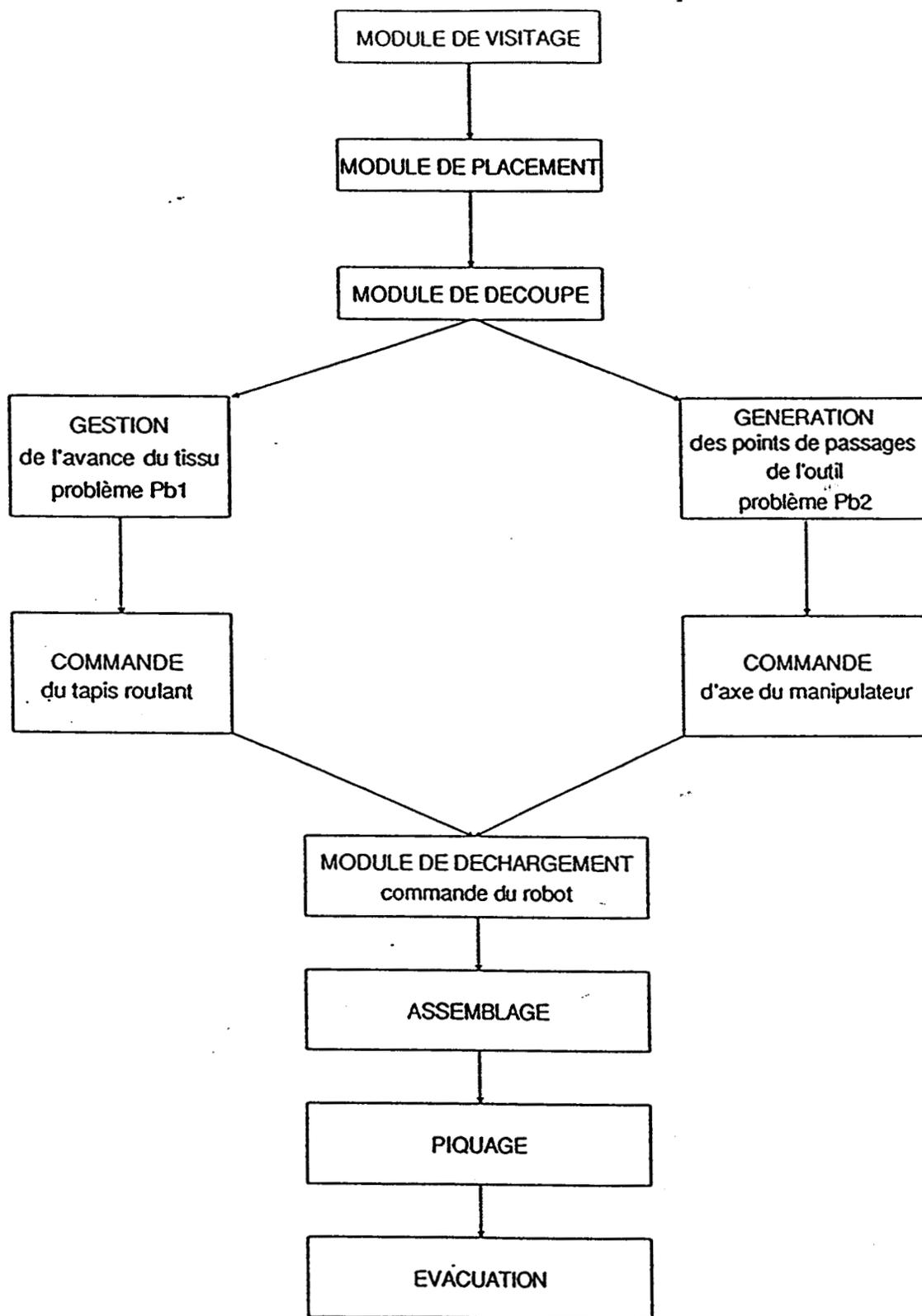


Figure III.1 : organigramme des différentes structures de la cellule

### III.1 GESTION DE L'AVANCE DU TISSU:

#### III.1.1 Position du problème

Nous rappelons que notre cellule se compose:

- d'un carrousel d'alimentation en bobines de tissu
- d'un tapis roulant sur lequel est déroulé le tissu à découper
- d'un manipulateur à 2 degrés de libertés
- d'un outil au laser.

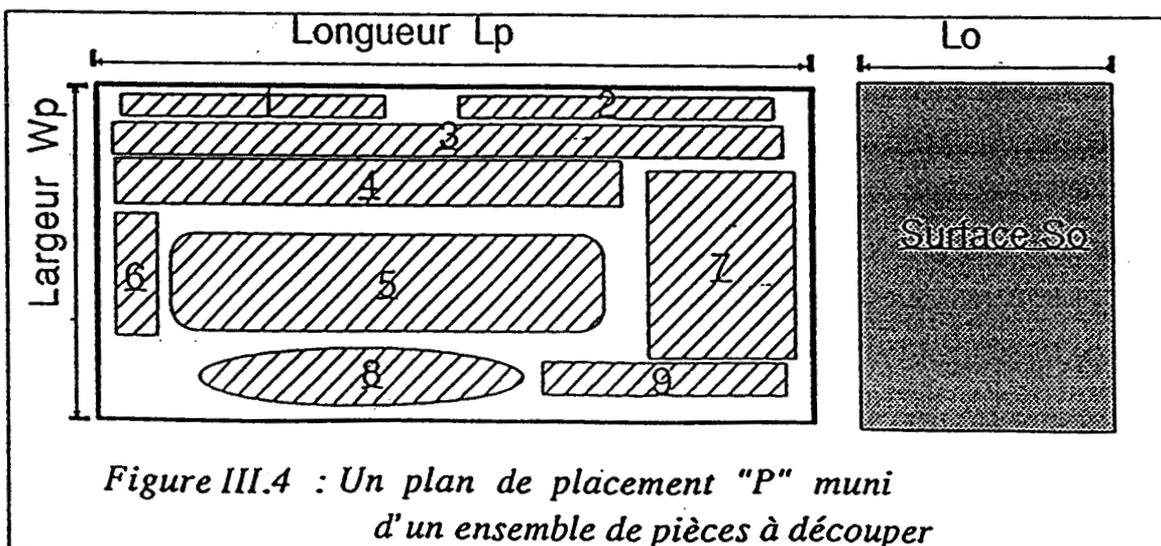
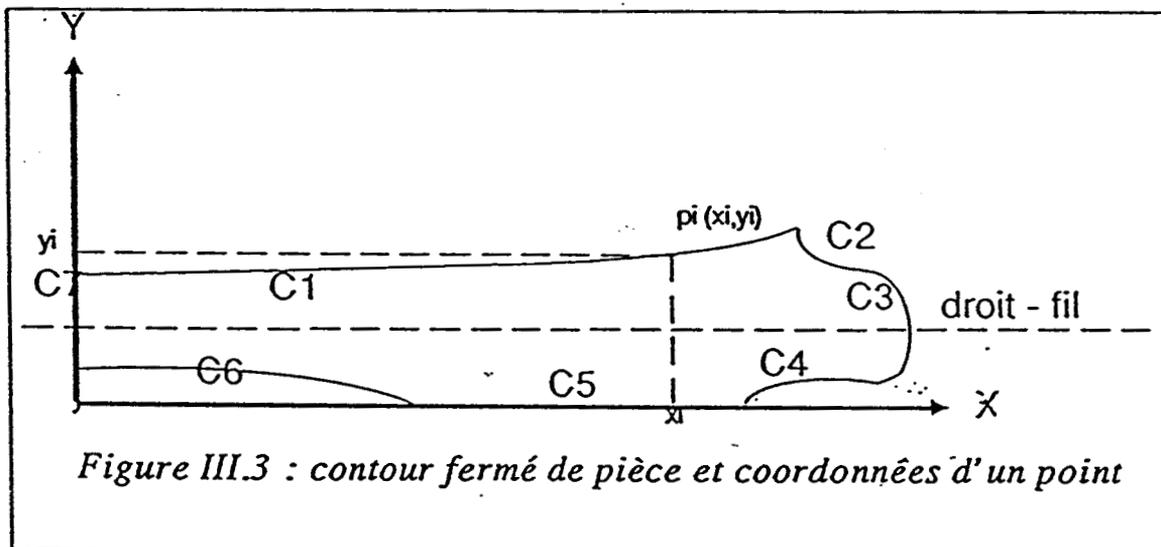
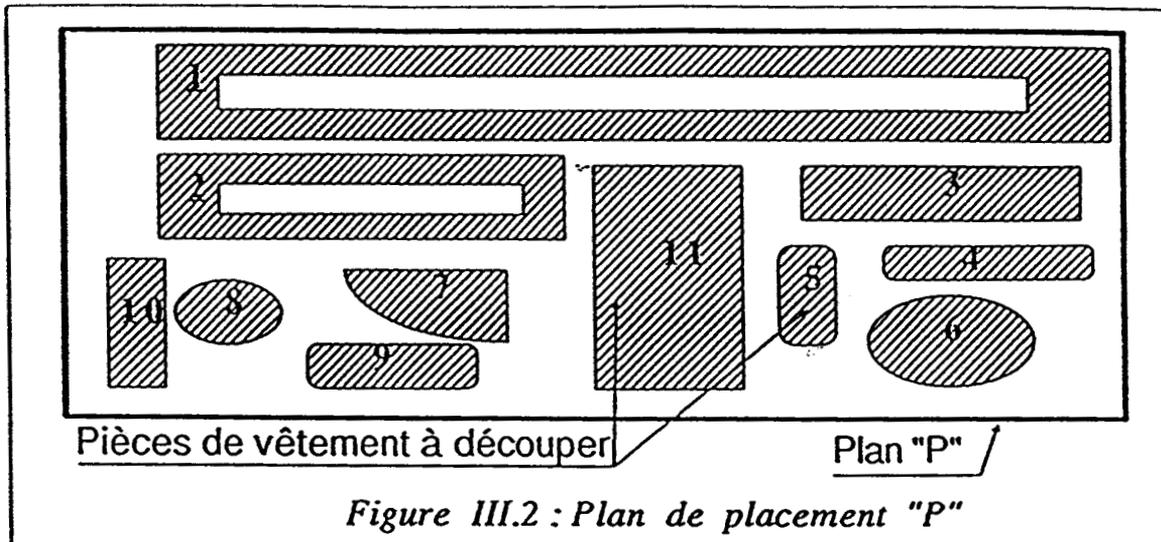
L'objectif de la cellule est de produire des pièces de vêtements ( ou objets ). Le module de placement se charge de disposer au mieux cet ensemble d'objets, de manière à minimiser les chutes de tissu. Ensuite, nous recevons ce plan de placement (ou d'imbrication), correspondant à notre base de données. Alors, se pose le problème de la découpe de cet ensemble. La figure III.1 situe le problème Pb1, qui nous intéresse, dans l'organigramme des différentes structures qui composent la cellule.

#### III.1.2 Données du problème

Les données de Pb1 sont les objets à exécuter. Ils sont référencés et positionnés sur le plan d'imbrication "P". La figure III.2 représente un plan "P", muni d'un ensemble de pièces. Les dimensions de "P" sont:

- sa longueur  $L_p$
- sa largeur  $W_p$ .

Nous avons vu que la base de données commune au placement et à la découpe était le polynôme caractéristique de chaque objet. Ces polynômes nous permettent de manipuler aisément les pièces, et nous donnent la succession des points  $p_i (x_i, y_i)$  qui caractérisent un contour fermé (figure III.3).



D'autre part, au cours de notre formulation du problème nous introduirons la notion de coût associé à un trajet. La détermination de ces coûts fera l'objet de l'étude du Chap.4. Ayant positionné notre problème, nous allons en donner les contraintes et critères.

## III.2 CRITERE ET CONTRAINTES

### III.2.1 Contraintes

Notre manipulateur possède une surface de travail  $S_0$  limitée. Nous disposons d'une longueur  $L_0$  et d'une largeur  $W_0$  avec  $S_0 = L_0 * W_0$ . De plus, la surface du plan "P" est telle que  $S_p = L_p * W_p$  ( figure III.4).

Notre processus de découpe induit une contrainte de type technologique que l'on formulera comme suit:

$$- W_p \leq W_0 \quad (1)$$

De plus, dans notre cas, nous disposons de plans de placement dont la longueur est telle que:

$$- L_p \geq L_0 \quad (2)$$

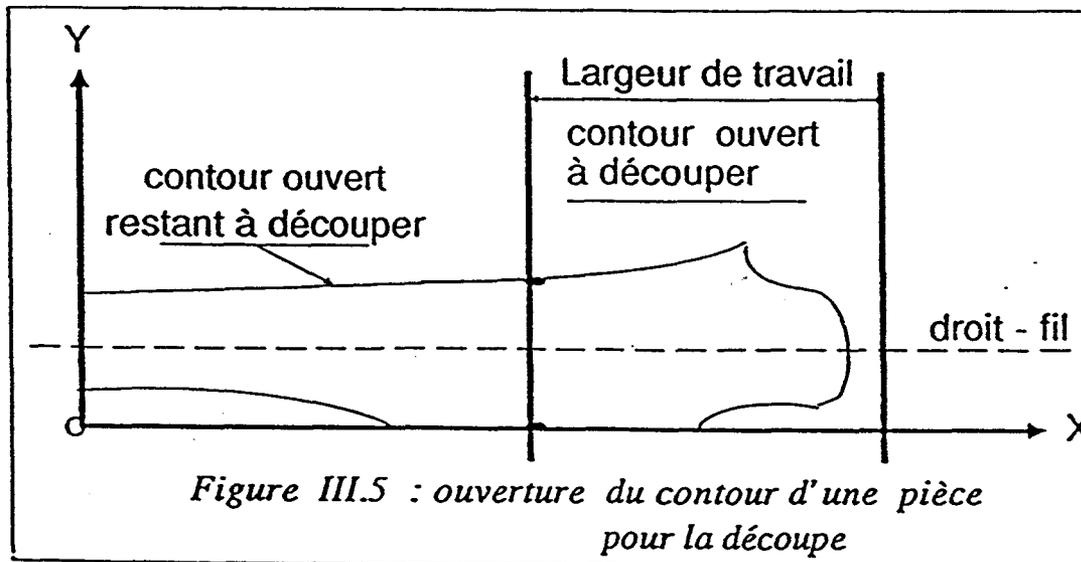
#### Note:

- (2) nous impose une exécution du plan de placement par étapes. Compte tenu de cette inégalité, nous devons générer une commande de moteur discontinue. En effet, pour chaque décision, nous devons commander l'avance du tissu par pas successifs.

### III.2.2 Critère

Le processus de production est donc contraint sur son mode d'avance. Ceci nous impose un morcellement de tout objet dont la longueur serait supérieure à celle de la surface de travail  $S_0$ .

Dès lors, un objet qui était caractérisé par son contour fermé, se décompose en un ensemble de contours ouverts (figure III.5).



Le fait, de morceler une pièce, engendre une commande cinématique de l'outil différente. Cette commande est la suivante, dans le cas d'un contour fermé:

- phase d'accélération,
- phase à vitesse constante,
- phase de décélération.

Si l'on applique ce type de diagramme pour un contour  $C_i$  quelconque, celui-ci suppose un temps "t" pour le mouvement de l'outil. La décomposition de  $C_i$ , en contours élémentaires ouverts (ou sous-contours), nécessite un diagramme identique pour chaque sous-contour, mais aussi des pertes de temps dues à l'avance du tissu, après la découpe de chacun d'entre eux. On voit que ce temps "t" est majorable d'une pénalité, variable, suivant les décisions que l'on prend lors de la décomposition de son contour initial, en sous-contours.

Notre étude consiste à rechercher un ensemble de décisions admissibles, tel que le temps de production d'un plan "P" soit minimum. Nous chercherons à optimiser ce temps de production, par un choix judicieux de la découpe du tissu.

### III.3 FORMALISATION DU PROBLEME Pb1

#### III.3.1 Approche du problème

Les problèmes de gestion de systèmes ou de processus de fabrication sont souvent traités sous la forme de problèmes d'ordonnancement. Cependant, chaque approche est guidée par le processus étudié. C'est pourquoi nous pensons qu'il est utile de présenter l'approche classique et d'en déduire notre approche du problème.

##### III.3.1.1 Approche classique

Les différentes données d'un problème d'ordonnancement sont les tâches, les contraintes, les ressources, et la fonction économique.

Les tâches peuvent être reliées par des contraintes "potentielles". Celles-ci englobent les contraintes de succession : par exemple, pour un réducteur de vitesse, avant de placer les roulements il faut monter la pignonerie, etc... Elles comprennent également les contraintes de localisation temporelle: la tâche i doit être achevée à telle date ou, au contraire, son exécution ne peut pas commencer avant telle date.

Ces tâches requièrent pour leur exécution certaines ressources: elles peuvent passer sur des machines, utiliser de la main d'oeuvre, etc...Ces ressources introduisent des contraintes de types "cumulatives" et "disjonctives". Enfin, il faut programmer les tâches de façon à optimiser un certain objectif, qui sera, suivant le cas, la minimisation de la durée totale (c'est le critère le plus fréquemment employé) ou le respect des dates de commandes ou encore la minimisation d'un coût. Décrivons, plus en détail, les différents concepts qui viennent d'être évoqués.

### i) les tâches

Dans un problème d'ordonnancement, les tâches sont supposées être connues à l'avance. Les données d'une tâche sont:

- sa durée
- sa date de début au plus tôt
- sa date de fin au plus tard

#### i.1) relation entre les tâches

Elles sont souvent liées entre elles par des relations d'antériorité, donc dépendantes. Dans le cas contraire, elles sont dites indépendantes.

#### i.2) mode d'exécution

Il est possible d'exécuter une tâche soit en une fois, on dira que son exécution est non préemptive, ou morcelée, on dit que la préemption est possible.

#### i.3) les problèmes d'ateliers

L'ordonnancement d'opérations pour un système de production peut être caractérisé selon la nature des contraintes de précédence entre les opérations. On rencontre trois types de problèmes d'ateliers:

- si les opérations élémentaires sont indépendantes, on parlera de problème d'"open-shop"
- si les opérations élémentaires d'un travail sont liées par un ordre total, il s'agira d'un problème de "job-shop"

- enfin, si les opérations élémentaires des travaux sont liées par le même ordre total, il s'agit d'un problème de "flow-shop"

Les problèmes d'ateliers sont des problèmes combinatoires.

### ii) les ressources

On distingue deux types de ressources nécessaires aux tâches:

- une ressource est renouvelable si elle est encore disponible pour d'autres tâches: par exemple, les machines, le personnel, les fichiers, etc..
- une ressource est consommable si son utilisation n'est plus possible pour les autres tâches: par exemple, la matière première, etc...

### iii) les critères

Les critères, ou fonctions économiques, les plus utilisés font intervenir:

- la durée totale
- les retards
- minimisation d'un coût de stockage d'encours
- nombre d'interruptions ou de préemptions ( ex: en multiprogrammation)

On s'aperçoit que les critères peuvent être extrêmement variés et complexes.

En conclusion, nous pouvons définir schématiquement les deux grandes classes de problèmes d'ordonnement:

- les problèmes de gestion d'ateliers (open shop, flow shop, job shop etc.), pour les systèmes.
  
- les problèmes d'ordonnement de tâches sur machines (one machine scheduling problem etc.), pour les processus. Notre étude s'intègre dans cette dernière partie.

Il existe une importante bibliographie sur le thème de l'ordonnement de tâches sur machine unique [19,20,21,22,23,24]. Cependant, notre problème n'entre pas dans le cadre des méthodes de résolution proposées. En effet, nous ne nous trouvons pas dans un cas d'ordonnement dynamique, dans le sens où nous ne sommes pas contraints par des dates de début et de fin sur les tâches de chaque plan de placement.

De plus, notre cellule comporte un module de détection de défauts sur le tissu. Le module de placement prend en compte tous les défauts en temps réel et dispose les pièces de vêtement en fonction des données reçues. Ceci implique un positionnement des pièces en temps réel, ce qui ne nous permet pas une connaissance des tâches de découpe à l'avance. C'est pourquoi, nous avons cherché à modéliser notre problème comme un cas d'ordonnement de tâches sur un processus de fabrication séquentiel.

### III.3.1.2 Une approche du problème

Après avoir présenté notre problème d'optimisation, nous nous proposons de le formuler en nous basant sur la programmation dynamique et la programmation linéaire. Avant d'en donner les caractéristiques, nous présenterons les différentes étapes qui nous ont conduit à ce type de programme. Nous rappelons que les données sont:

- le plan d'imbrication de longueur  $L_p$
  
- un ensemble de pièces à découper
  
- une surface de travail de longueur  $L_o$ .

### but de l'optimisation

Une pièce de vêtement est représentée par un contour fermé. Dans notre cas, le problème consiste à déterminer un ensemble de pas à donner pour l'avance du tissu. Il est évident que le choix d'avancer par pas successifs engendre un morcellement de la pièce, d'où une décomposition en contours ouverts de celle-ci.

Soit une pièce présente sur "P" et la décision d'avancer d'une certaine longueur. Le contour ouvert doit être découpé et nécessite une commande d'outil telle que l'on ait une phase d'accélération, une phase à vitesse constante puis une phase de décélération. Cette commande nécessite un temps de parcours.

Le choix du morcellement de "P" influe sur les temps de découpes, de par les phases d'accélération et décélération que l'on engendre après une avance. Notre but est donc de définir le meilleur choix de décomposition de "P" tel que le temps de production soit minimum.

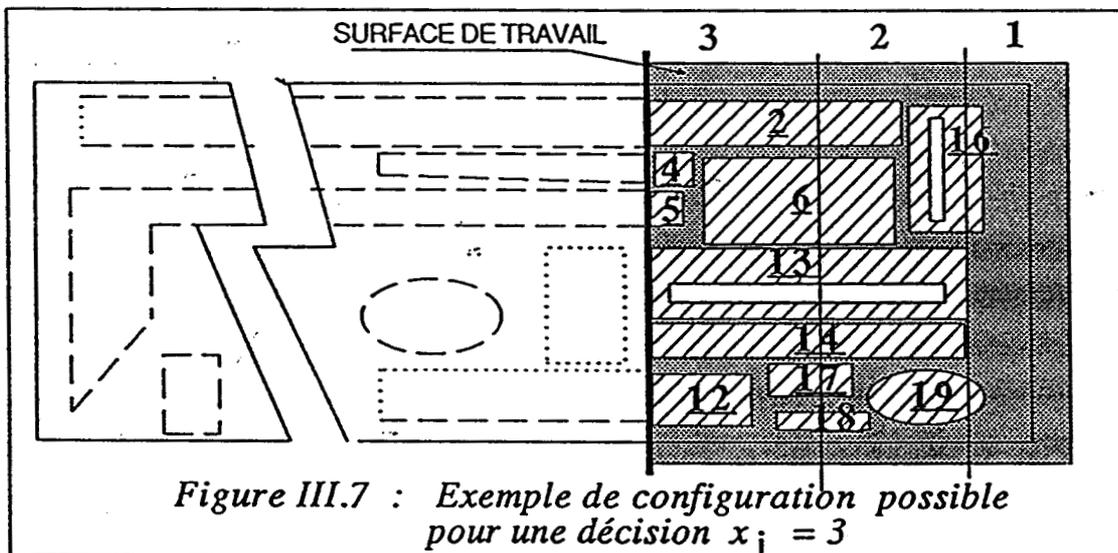
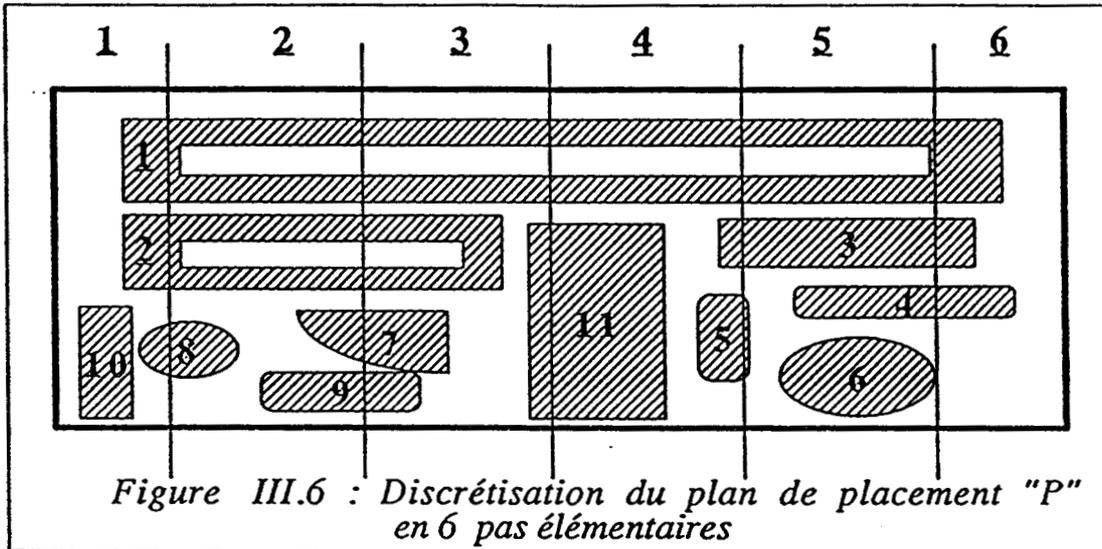
### approche du problème

Afin de mieux aborder ce problème de choix sur l'avance, nous avons pris comme base la discrétisation du plan "P" en "n" pas élémentaires. Tout les pas seront de longueur "l" constante (figure III.6):

$$- L_p = n * l$$

$$- L_o = \alpha * l$$

Notre problème consiste à déterminer un ensemble  $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$  tel que la production de pièces de vêtement soit optimisée ( Exemple figure III.7).



### III.4 FORMALISATION PAR LA PROGRAMMATION DYNAMIQUE

La programmation dynamique offre la possibilité de représenter l'évolution d'un processus séquentiel, et de le résoudre de façon optimale. Nous allons présenter les deux principales représentations.

#### III.4.1 Rappels

Avant d'aborder cette mise en forme d'un processus séquentiel, il nous faut donner les notions essentielles de la programmation dynamique pour l'étude d'un processus séquentiel.

##### i) définitions [25,26]

\* Etape:

Le processus considéré existe en un nombre fini d'étapes, numérotées 1,2,...,n.

\* Etat du processus:

A chaque étape le processus est décrit par une variable appelée "ETAT"; l'état du processus à l'étape k est noté  $e_k$ , élément de l'ensemble des états possibles du processus à l'étape k.

\* Variable de décision

Le décideur peut agir sur le processus au moyen d'une variable de contrôle  $x$ . La valeur qu'il va donner à l'étape k à  $x$  est notée  $x_k$ . Ce choix affecte l'état du processus à l'étape suivante.

\* Changement d'état

Le processus peut changer d'état à la fin de chaque période: le nouvel état est une fonction de l'état précédent, de la variable de décision, de l'étape k:

$$- e_k = f(e_{k-1}, x_k, k).$$

\* L'objectif

A chaque étape on associe un coût, suivant la valeur de la décision prise à cette étape. On recherche une trajectoire dont la somme des coûts est minimum.

ii.1) notations

A la k<sup>ième</sup> étape ( $k = 1, \dots, u$ ) on associe deux ensembles:

- $X_k$  : l'ensemble des décisions que l'on peut prendre à cette étape.
- $E_k$  : l'ensemble des états dans lesquels le système peut se trouver à la fin de la k<sup>ième</sup> étape.

Une décision  $x_k \in X_k$  alors que le processus se trouve dans l'état  $e_{k-1} \in E_{k-1}$  a pour effet:

- \* de faire évoluer le processus dans l'état  $e_k \in E_k$  à la fin de la k<sup>ième</sup> étape.
- \* d'engendrer un coût  $c_k(e_{k-1}, x_k)$ .

ii.2) définition

On appelle politique de l'état  $e_{i-1} \in E_{i-1}$  à l'état  $e_j \in E_j$  une séquence de décisions  $P(e_{i-1}, e_j) = (x_i, x_{i+1}, \dots, x_{j-1}, x_j)$ , telle que le coût associé à la politique  $P(e_{i-1}, e_j)$  est:

$$- C(P(e_{i-1}, e_j)) = \sum_{k=i}^j c_k(e_{k-1}, x_k)$$

### III.4.2 Caractéristiques du problème

Notre processus peut être défini comme étant un processus séquentiel [28], il peut être intégralement défini par les trois caractéristiques : état, étape, décision. Notre modèle présente des décisions qui se succèdent dans le temps, les décisions d'une étape donnée étant liées à l'état dans lequel se trouve le processus, après avoir évolué sous l'effet des décisions antérieures. C'est là un modèle séquentiel.

Rappelons les données de notre problème:

- $n$  : le nombre de pas total à effectuer
- $\alpha$  : le nombre maximum de pas imposé par le processus de découpe.

On pose:

- $x_k$  = nombre de pas que l'on peut donner au module "commande d'axe" du tapis,
- $e_k$  = quantité totale de pas effectués à la fin de la  $k^{\text{ième}}$  étape.

La valeur de la variable  $e_k$  décrit parfaitement "l'état du système" à la fin de la  $k^{\text{ième}}$  étape. Pour cette raison on dit que  $e_k$  est une variable d'état. La valeur de  $x_k$  caractérise la "décision" prise à la  $k^{\text{ième}}$  étape:  $x_k$  sera appelée variable de décision.

Ces deux variables devront satisfaire les contraintes suivantes:

$$- 0 \leq x_k \leq \alpha \quad \text{pour } k = 1, \dots, u \quad (1)$$

$$- \sum_{i=1, \dots, u} x_i = n \quad (2)$$

$$- 0 \leq u \leq n \quad (3)$$

$$- e_k = e_{k-1} + x_k \quad \text{pour } k = 1, \dots, u \quad (4)$$

$$- 0 \leq e_k \leq n \quad \text{pour } k = 1, \dots, u \quad (5)$$

avec:

$$- u, n, x_k, e_k, \alpha \in \mathbb{N}$$

et pour lesquelles:

- (1) Une décision  $x_k$ , à la  $k^{\text{ième}}$  étape, a pour effet de permettre l'avance du tissu sur un certain nombre de pas. Ce nombre ne peut excéder  $\alpha$ .
- (2) L'ensemble des décisions  $X = \{x_1, x_2, \dots, x_u\}$  doit permettre la réalisation du plan "P".
- (3) Le nombre total de pas ne peut excéder "n".
- (4) Chaque décision a pour effet de faire évoluer le processus d'un état à un autre. A la fin de la  $k^{\text{ième}}$  étape, on affecte à l'état  $e_k$ , la quantité de pas déjà effectués pour l'état  $e_{k-1}$ , augmenté de la quantité de pas lors de la prise de décision  $x_k$ .

On désire définir une trajectoire de coûts minimum entre un état initial  $e_0$  et un état final  $e_u$ .

Nous associerons un coût  $c_k(e_{k-1}, x_k)$  à chaque décision  $x_k$ . La détermination de ce coût fera l'objet du chapitre suivant.

### III.4.3 Représentations

#### III.4.3.1 représentation par les "boîtes"

Une première représentation de l'évolution d'un processus séquentiel peut se faire à l'aide du graphe suivant ( figure III.8). Cette représentation se caractérise comme suit:

- le passage, de la sortie d'une boîte  $b_{k-1}$  à l'entrée d'une boîte  $b_k$ , est représenté par la variable d'état  $e_{k-1}$ .

- la variable de décision  $x_k \in X_k$  permet de passer de l'état  $e_{k-1}$  à l'état  $e_k$ ,
- la prise de décision engendre un coût  $c_k(e_{k-1}, x_k)$ .

### Exemple:

Si l'on transpose cette représentation à notre problème Pb1 on tire le schéma suivant ( figure III.9) avec:

- $e_{k-1}$  : la quantité de pas déjà exécutés.
- $x_k$  : à cette  $k^{\text{ième}}$  étape, la décision  $x_k$  doit être telle que:
- $0 \leq x_k \leq 5$

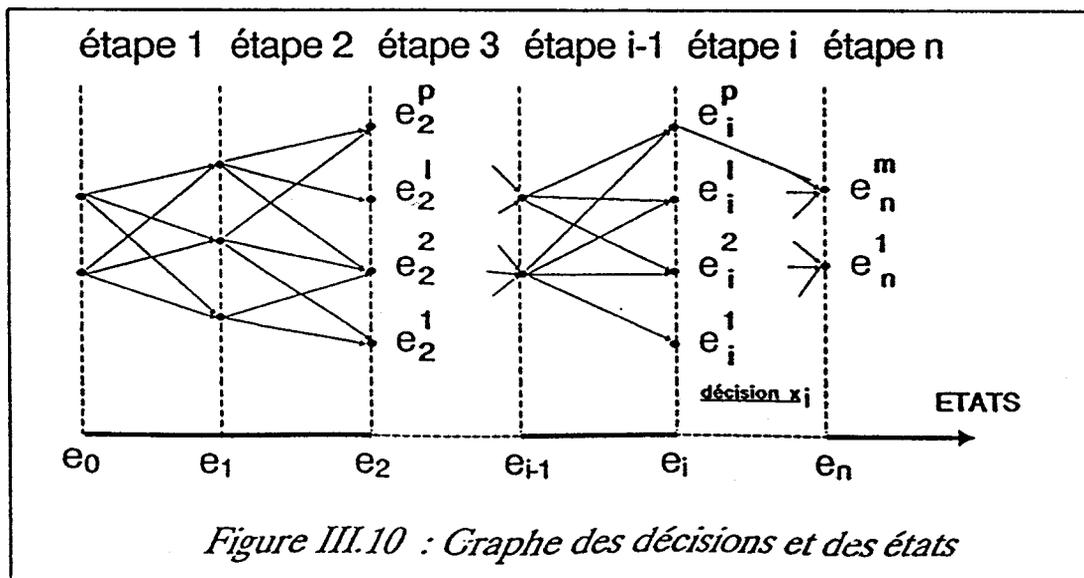
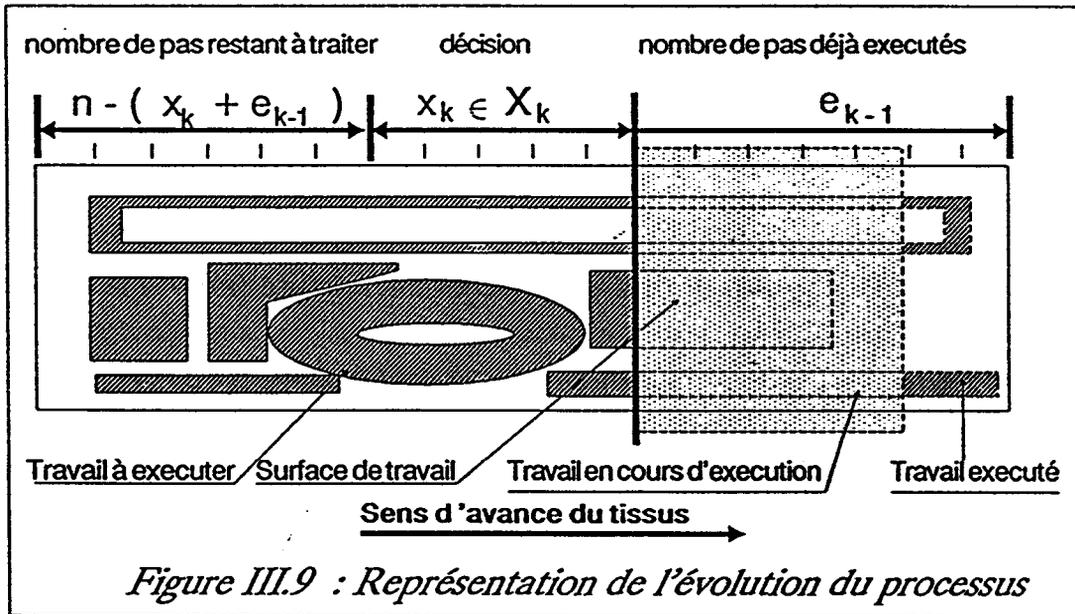
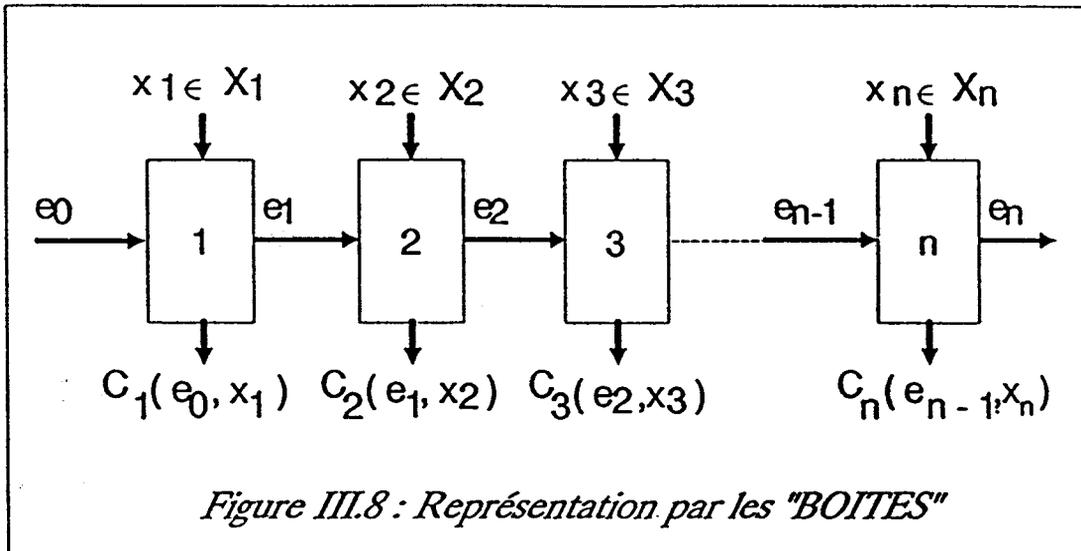
### III.4.3.2 Représentation par graphe séquentiel

#### i) Notion de graphe séquentiel [27]

#### définition

Un graphe séquentiel est un graphe  $G = [E, U]$  qui satisfait les trois conditions suivantes:

- L'ensemble  $E$  des sommets est réparti en sous-ensembles  $e_0, e_1, \dots, e_n$  ( $n$  pouvant être fini ou infini) disjoints, totalement et strictement ordonnés:  $e_0$  précède  $e_1$ ,  $e_1$  précède  $e_2, \dots, e_{n-1}$  précède  $e_n$ . Notons que  $e_0$  et  $e_n$  constituent des cas particuliers;  $e_0$  n'a pas de précédent et est encore appelé entrée du graphe,  $e_n$  n'a pas de successeur et est appelé sortie du graphe;
- Un arc quelconque du graphe séquentiel ne peut joindre que deux sommets appartenant à deux sous-ensembles  $e_k, e_{k+1}$ .



- Un chemin  $(e_i^j, e_{i+1}^l)$  correspond à une décision  $x_i^{j,l}$  d'emprunter l'arc  $(e_i^j, e_{i+1}^l)$ .

Un tel graphe peut être représenté de la manière suivante (fig. III.10). Les sommets d'un même sous-ensemble  $e_n$  sont disposés sur une même droite. Chaque arc joint deux sommets appartenant à deux droites consécutives. Il n'existe aucun arc joignant deux points situés sur une même droite.

Ainsi, nous pouvons définir un graphe séquentiel complet  $G = [E, U]$ , comme étant un graphe séquentiel tel que tout sommet est un sommet d'au moins un chemin reliant un sommet de la première étape à un sommet de la dernière étape. La notion de graphe séquentiel est particulièrement intéressante pour l'étude de processus comportant des décisions successives.

Notre problème comporte aussi des décisions successives, ce qui nous conduit à utiliser cette représentation. Examinons, sur un exemple, le graphe associé à notre problème pour lequel les données sont  $n$  et  $\alpha$ .

### ii) Exemple d'application au problème

Soit le graphe  $G = [E, U]$  défini comme suit:

- l'ensemble  $E$  des sommets est caractérisé par l'ensemble des états. Chaque état représente une quantité de pas déjà effectués pour le processus d'avance.
- l'ensemble des arcs correspond aux chemins envisageables à partir d'un état donné, sachant que l'on ne peut évoluer qu'au maximum d'un nombre  $\alpha$  de pas.
- une contrainte sur toute décision sera telle que:

$$* 0 \leq x_k \leq \alpha \quad \text{pour } k = 1, \dots, u$$

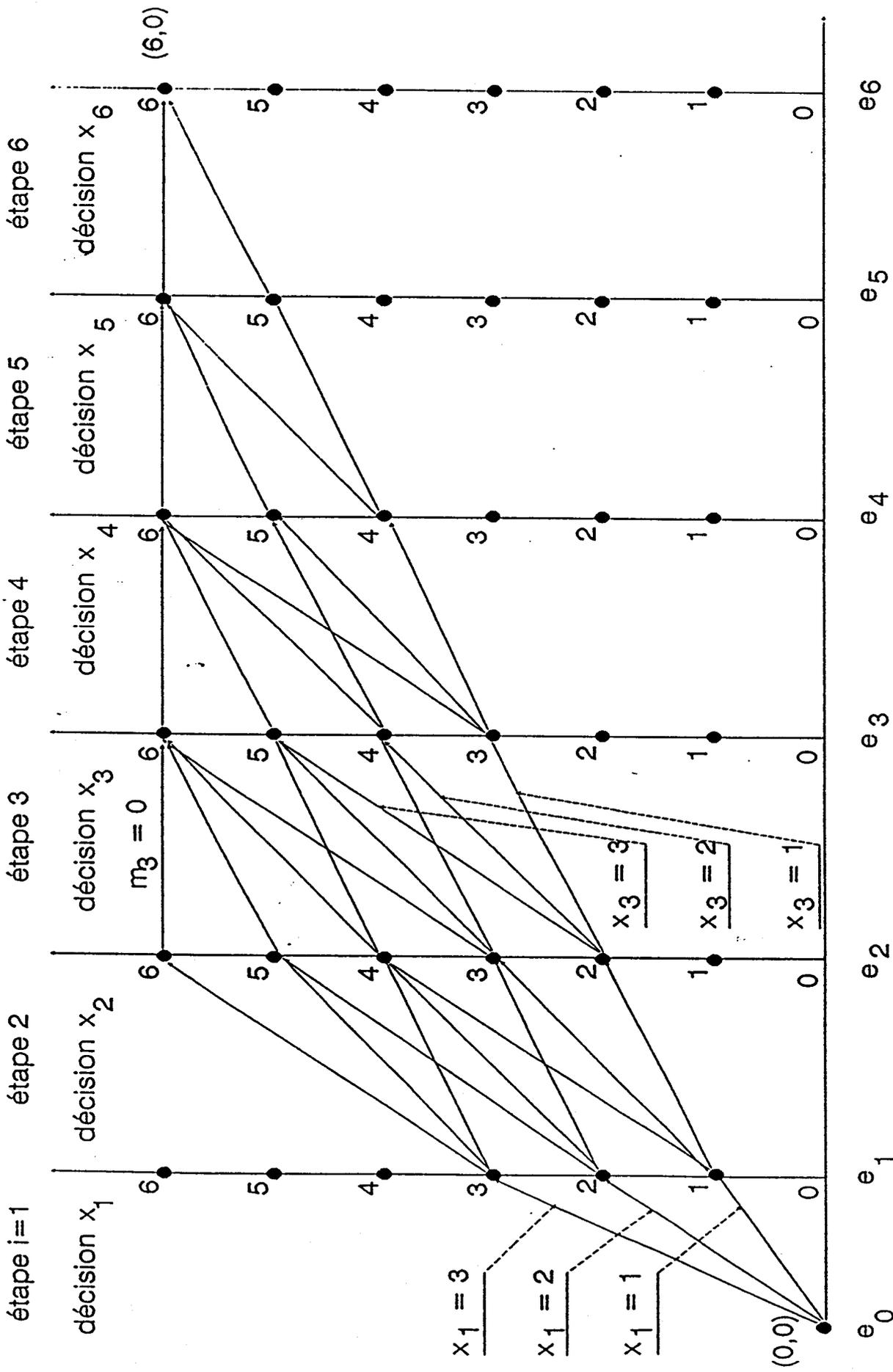


Figure III.11 : Graphe des décisions et des états appliqué au processus de découpe

-  $n = 6$  : c'est le nombre de pas obtenu sur le plan de placement, après discrétisation de celui-ci,

-  $\alpha = 3$  : le nombre de pas correspondant à la longueur de la surface de travail de l'outil (voir fig. III.7).

Notre problème est alors équivalent sur le graphe  $G$  à la recherche d'un chemin de longueur minimale entre le sommet  $(0,0)$  et le sommet  $(6,6)$  (fig. III.11).

Plus généralement, [26] montre que tout problème de programmation dynamique, à nombre fini d'états, peut toujours être considéré comme un problème de plus court (plus long) chemin dans un graphe séquentiel.

#### III.4.4 Formalisation

Nous venons de donner deux représentations de l'évolution de notre processus. Nous allons présenter une formulation pour ce problème. Notre but est de minimiser la somme des coûts en partant de l'état  $e_0$  à l'état  $e_n$ . A chaque étape  $k$  est associée une fonction de coût  $c_k(e_{k-1}, x_k)$  relativement à la prise de décision  $x_k$ . On peut donc retirer de cette étude le tableau suivant à la  $k^{\text{ième}}$  étape:

tableau 1:

<u>Etape k</u>	
Une variable de décision	$x_k$
Une variable d'état	$e_k$
Une fonction de transfert	$e_k = e_{k-1} + x_k$
Une fonction de coût	$c_k(e_{k-1}, x_k)$

d'où le programme mathématique associé au problème Pb1:

(PD)	minimiser	$z = \sum_{k=1}^u c_k (e_{k-1}, x_k)$	
	sous les		
	contraintes	$e_j = e_{j-1} + x_j$	pour $j = 1, \dots, n$ (1)
		$0 \leq e_j \leq n$	pour $j = 0, \dots, n$ (2)
		$0 \leq x_j \leq \alpha$	pour $j = 0, \dots, n$ (3)
		$e_0 = 0$	(4)
	$e_n = n$	(5)	
	$x_j, e_j \in \mathbb{IN}$ et $c_k \in \mathbb{IR}^+$		

où:

- $z$  est la fonction objectif ou le critère à minimiser,
- $z$  est une fonction séparable monotone.
- (1): chaque état dépend de l'état qui le précède et de la décision prise à cette étape.
- (2) et (3): limite des valeurs possibles, respectivement sur les états et les variables décisionnelles,
- (4) et (5): caractérisent respectivement l'état initial et l'état final.

### III.4.5 Equation de récurrence

Une fois le tableau 1 défini, c'est à dire que l'évolution du processus a été ramenée à une évolution séquentielle, il est possible de formuler ce problème à l'aide d'une équation de récurrence. Celle-ci est obtenue à partir du théorème d'optimalité pour toute fonction décomposable, monotone et non décroissante.

#### Le principe d'optimalité (principe de Bellman) [27]:

Soit  $P^*(e_0, e_n)$  une politique optimale conduisant de l'état  $e_0 \in E_0$  à l'état  $e_n \in E_n$ . Quelle que soit la décision prise à l'étape  $k$  qui amène le système de l'état  $e_{k-1} \in E_{k-1}$  à l'état  $e_k \in E_k$ , la portion de la politique optimale  $P^*(e_0, e_k)$  entre  $e_0$  et  $e_{k-1}$  est optimale.

On pose:

-  $J^*(e_k)$  = coût d'une politique optimale de  $e_0$  à  $e_k \in E_k$ ,

et on a la formule de récurrence "avant":

$$J_k^*(e_k) = \min_{x_k} \{ c(e_{k-1}, x_k) + J_{k-1}^*(e_{k-1}) \} \quad (5a)$$

on pose:

-  $x_k^*(e_k)$  = "meilleure décision à l'étape  $k$  qui permet d'arriver à  $e_k$ ".

$x_k^*(e_k)$  est une valeur de  $x_k$  pour laquelle le min est atteint dans (5a). Pour déterminer une politique optimale  $(x_1^*, x_2^*, \dots, x_n^*)$  on pose  $e_n^* = e_n$  et on résoud de proche en proche pour  $k = n, n-1, \dots, 2, 1$  les équations:

$$- x_k^* = x_k(e_k),$$

$$- e_{k-1}^* \text{ est défini par } e_k^* = f_k(e_{k-1}^*, x_k^*).$$

### III.4.6 limite de la programmation dynamique [28]

#### i) avantage

La programmation dynamique permet une optimisation de tout problème séquentiel. Comme nous venons de le présenter, dans notre cas, elle est intéressante dès lors où la taille du vecteur d'état reste peu élevée.

#### ii) inconvénient

Un inconvénient est celui où le vecteur d'état est de taille plus importante. En effet, la programmation dynamique reste une méthode énumérative (énumération non exhaustive ou implicite).

La complexité algorithmique (temporelle et spatiale) est donc fonction de la dimension de l'espace d'état et restreint l'utilisation de la méthode (le lecteur pourra se référer à la nombreuse bibliographie concernant les limites d'utilisation de la programmation dynamique [26,28]). C'est pourquoi, nous proposons une approche basée sur la programmation linéaire.

### III.5 FORMALISATION PAR LA PROGRAMMATION LINEAIRE

Avant de donner les quelques notions de programmation linéaire, nécessaires à notre formulation, nous allons donner une autre représentation de l'évolution de notre processus au moyen d'un réseau .

### III.5.1 Notion de réseau

#### i) définition [28, 29]

On considère un réseau  $R = [V, U, c]$  défini de la manière suivante:

- $V = \{v_1, v_2, \dots, v_p\}$  ensemble des sommets,
- $U = \{u_1, u_2, \dots, u_l\}$  ensemble des arcs,
- $c = \{c_1, c_2, \dots, c_l\}$  ensemble des coûts associés à chaque arc  $u \in U$ .

#### ii) représentation

La représentation graphique, que nous allons donner, se base sur la définition i). Les caractéristiques de notre réseau sont les suivantes:

- à l'ensemble  $V$  on associera l'ensemble des états possibles dans lequel peut se retrouver le processus. Ces états seront caractérisés par un paramètre  $j$  ( $j = 0, 1, 2, \dots, n$ ).
- un arc  $u \in U$  si il existe un arc  $(i, j)$  (ou une décision) permettant de passer d'un sommet  $i$  à un sommet  $j$ .
- $c$  : associe à chaque arc  $u$  le coût  $c(u)$

De plus, à chaque arc  $u \in U$  nous associerons la variable de décision définie comme suit:

$$x_{ij} = \begin{cases} 1 & \text{si l'arc } u = (i, j) \text{ est sélectionné,} \\ 0 & \text{sinon,} \end{cases}$$

ainsi qu'un coût  $c_{ij} = c(u)$  relativement à cette décision.

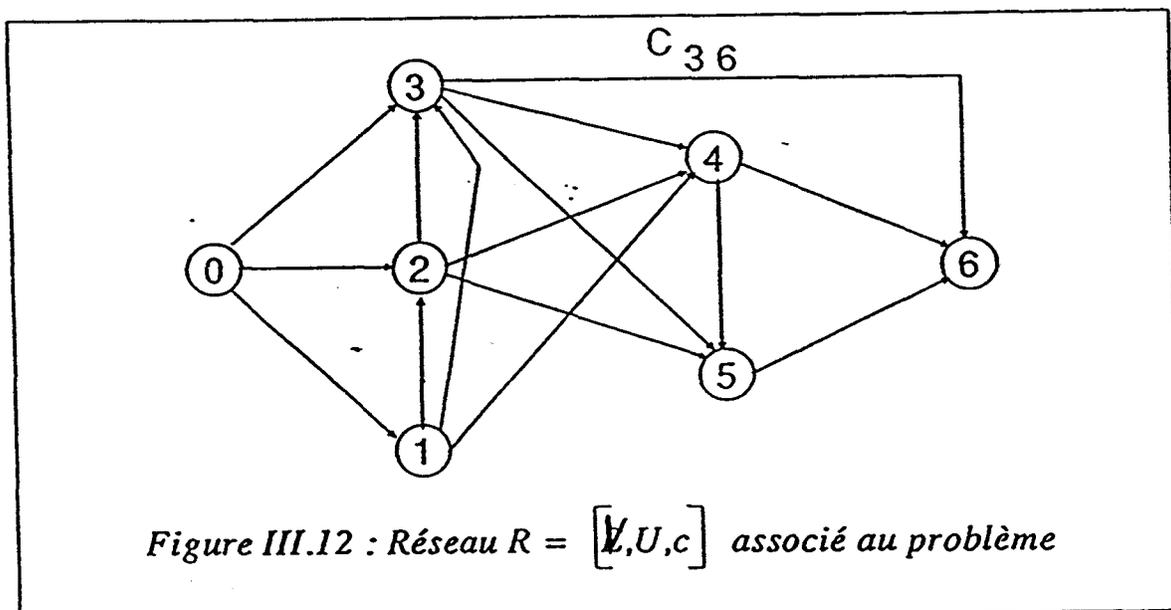
**Note:**

Comme nous l'avons précisé en introduction de ce chapitre, nous ne calculerons pas ici la valeur de ce coût. En effet, il fera l'objet du chapitre suivant car il nécessite, à lui seul, une étude d'optimisation de trajectoire.

**iii) Exemple**

On considère l'application à notre problème de ce type de réseau pour lequel les données sont:

- $n = 6$  : le nombre de pas du plan de placement "P", après discrétisation de celui-ci,
- $\alpha = 3$  : le nombre de pas correspondant à la surface de travail de l'outil.



On tire le Réseau R associé à ce problème (figure III.12). Ce réseau possède donc 6 sommets numérotés de 0 à 6 et dont les caractéristiques sont les suivantes:

- un sommet source "0",
- un sommet puits "6" correspondant au sixième pas,
- et enfin, 5 sommets relatifs aux 5 états possibles dans lesquels on peut faire évoluer le processus.

Remarque:

Du sommet source "0" on désire atteindre le sommet puits "n". Pour ce faire, on dispose d'arcs orientés entre chaque sommet. Le passage d'un sommet "i" à un sommet "j" consiste à faire évoluer le processus d'un état à un autre et engendre un coût  $C_{ij}$ .

On cherchera à faire évoluer le processus d'un état initial "0" à un état final "n", tel que la somme des coûts soit minimum (ex. : 0,1,4,6 est une solution réalisable).

### III.5.2 Notion sur la programmation linéaire [21]

Avant de présenter la formulation de notre problème tel un programme linéaire en variables entières, nous allons fournir les quelques définitions relatives à cette branche de la programmation mathématique.

i) définition

Soit une  $m \times n$ -matrice A, un m-vecteur colonne b et un n-vecteur ligne c, et les programmes mathématiques suivants:

$$(PC) \left[ \begin{array}{l} \min z = cx \quad x \geq 0 \\ \text{sous les contraintes } Ax \leq b \end{array} \right.$$

Un "programme linéaire" est un problème dans lequel les variables sont des réels qui doivent satisfaire un ensemble d'équations et/ou d'inéquations linéaires (dites "contraintes") et la valeur d'une fonction linéaire de ces variables (appelée "fonction objectif"), doit être minimum (ou maximum)

### III.5.3 Formalisation

Notre formulation se base sur le réseau R associé au problème Pb1. On tire le PLNE1 suivant:

PLNE1	$\text{minimiser } z = \sum_{(i,j) \in \text{réseau}} c_{ij} x_{ij}$	
	$\text{sous les contraintes } \sum_{j=1}^{\alpha} x_{0j} = 1$	(1)
	$\sum_{j=n-\alpha}^{n-1} x_{jn} = 1$	(2)
	$\sum_{i < k} x_{ik} - \sum_{j > k} x_{kj} = 0 \quad \text{pour } k = 1, 2, \dots, n-1$	(3)
	$x_{ij} = \{0, 1\}$	(4)
	$c_{ij} \in \mathbb{R}^+$	<p style="margin: 0;">pour <math>i = 0, \dots, n</math> pour <math>j = 0, \dots, n</math></p>

ce programme est construit de la manière suivante:

- à chaque arc  $u = (i,j) \in U$  on affecte une variable décisionnelle binaire  $x_{ij}$  telle que:

$$x_{ij} = \begin{cases} 1 & \text{si l'arc } u = (i,j) \text{ est sélectionné,} \\ 0 & \text{sinon,} \end{cases}$$

- à chaque sommet  $v_k \in V$  on associe une des  $n$  positions dans laquelle peut se retrouver le tissu ( ceci correspond à l'indice de chaque pas sur le plan de placement pour  $k = \{1, \dots, n\}$ ).

Soit  $x_{ij}$  la variable relative à la décision de passer d'un sommet  $i$  à un sommet  $j$ , les contraintes du programme sont telles que:

- (1) du sommet source "0" il ne peut y avoir qu'une destination possible,
- (2) du sommet puits "n" il ne peut y avoir qu'une source d'arrivée possible,
- (3) pour tout autre sommet, il ne peut y avoir qu'un arc incident et un adjacent,
- (4) variable binaire associée à toute décision.
- $c_{ij}$  : le coût associé à la décision  $x_{ij}$  lors du passage du sommet  $i$  au sommet  $j$ .

### III.5.4 Exemple d'application

Reprenons le réseau R du problème et écrivons le PLNE1 de celui-ci, on tire:

$$\text{minimiser } z1 = c_{01}x_{01} + c_{02}x_{02} + c_{03}x_{03} + c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{23}x_{23}$$

$$+ c_{24}x_{24} + c_{25}x_{25} + c_{34}x_{34} + c_{35}x_{35} + c_{36}x_{36} + c_{45}x_{45} + c_{46}x_{46} + c_{56}x_{56}$$

sous les contraintes :

$$- x_{01} + x_{02} + x_{03} = 1 \quad (1)$$

$$- x_{01} - (x_{12} + x_{13} + x_{14}) = 0 \quad (2)$$

$$- x_{02} + x_{12} - (x_{23} + x_{24} + x_{25}) = 0 \quad (3)$$

$$- x_{03} + x_{13} + x_{23} - (x_{34} + x_{35} + x_{36}) = 0 \quad (4)$$

$$- x_{14} + x_{24} + x_{34} - (x_{45} + x_{46}) = 0 \quad (5)$$

$$- x_{25} + x_{35} + x_{45} - (x_{56}) = 0 \quad (6)$$

$$- x_{36} + x_{46} + x_{56} = 1 \quad (7)$$

où:

- (1) et (2) sont les contraintes respectives aux sommets 0 et 6,
- (2), (3), (4) et (5) celles relatives aux sommets 1,2, 3, 4, 5.

### III.6 CONCLUSION

Dans ce chapitre, nous venons de donner une approche du problème de la gestion de l'avance du tissu. Elle se présente sous la forme d'un partitionnement de la surface du plan de placement en "n" surfaces élémentaires. Ensuite, une première formulation est présentée. Elle se base sur la notion de processus séquentiel et aboutit à un programme mathématique. Cependant, vu les inconvénients relatifs à la résolution par la programmation dynamique, nous avons formulé ce problème par un programme linéaire (PLNE1) en variables bivalentes.

Toutefois, une inconnue n'a pas encore été définie, il s'agit du coût affecté à chaque décision. Dans le chapitre suivant, nous allons aborder ce problème et en fournir une approche ainsi que sa formulation.

## INTRODUCTION

La transformation du tissu se fait par découpe au laser. Elle aboutit à un produit fini manufacturé à savoir l'ensemble des pièces de vêtements. Nous avons présenté au chap. III une approche du problème de gestion de l'avance du tissu. Dans la présentation qui a été faite, nous avons introduit une variable de coût relative à la décision  $x_i$ , d'avancer d'un certain nombre de pas, telle que  $1 \leq x_i \leq \alpha$ . La détermination de cette variable fait l'objet de ce chapitre.

Ce problème est la conséquence directe du problème Pb1 (chap. III). En effet, chaque prise de décision engendre un coût qui interviendra au niveau de la politique optimale recherchée. Dans ce chapitre, nous allons donner une approche du problème ainsi qu'une formulation générale par un programme linéaire.

La formulation que nous proposons est basée, essentiellement, sur le problème du voyageur de commerce.

### IV.1 POSITION DU PROBLEME

Un plan de placement possède un ensemble de pièces de vêtement que l'on devra couper, à l'aide d'un outil au laser, par un suivi de contour. La longueur de la surface de travail est inférieure à celle du plan de placement. Ceci nous conduit à une décomposition de chacune des pièces de vêtements en sous-contours élémentaires, si, pour une décision donnée, le contour de toute pièce ne peut être exécuté d'un seul tenant.

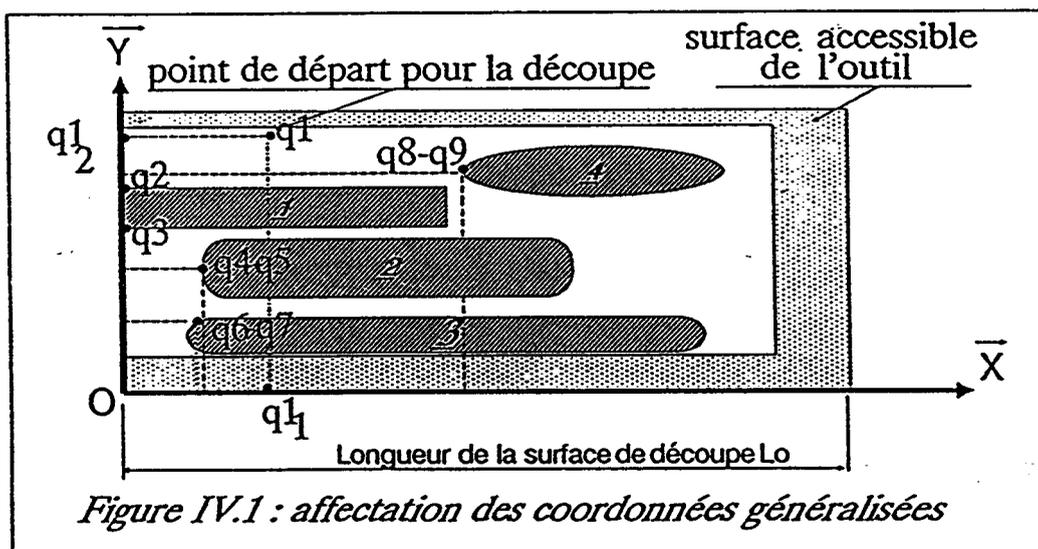
#### IV.1.1 Données du problème

Soit la décision  $x_i$  d'avancer d'un nombre de pas. Cette décision a pour effet de faire évoluer le processus tel que l'on ait sous la surface accessible de l'outil, un ensemble de sous-contours ou de contours à découper.

Dans le cas général, une tâche correspond à l'exécution, par le manipulateur supportant l'outil, d'un ensemble de mouvements élémentaires ayant pour but d'établir l'assemblage, le piquage, etc... Pour le processus de fabrication qui nous intéresse, la tâche représente le suivi, par l'outil, d'un contour (fermé ou ouvert) de la pièce de vêtement.

Les données de notre problème sont donc les suivantes:

- \* un ensemble  $T = \{t_1, \dots, t_k\}$  de tâches à exécuter, référencées par rapport à un référentiel relatif (celui de la zone de découpe),
- \* un point de départ, correspondant à la position de l'outil, après une découpe (décision précédente).



#### Exemple:

Sur la figure IV.1 on a représenté un ensemble  $T = \{t_1, t_2, t_3, t_4\}$  de quatre tâches, pour lequel  $t_i$  ( $i = 1, \dots, 4$ ) représente une tâche de découpe de la pièce n°i. De plus:

- Les tâches  $t_2, t_3$  et  $t_4$  représentent des contours fermés.
- par contre, pour une telle décision, la pièce n°1 fait l'objet d'une exécution en deux étapes. La première étape consiste à traiter la tâche  $t_1$ , caractérisée par un sous-contour élémentaire ouvert. La deuxième étape consiste à traiter la dernière portion de contour non exécuté.

Notre générateur doit fournir, au module de commande moteur du manipulateur, un trajet de coût minimum, dans lequel  $t_1, t_2, t_3, t_4, \dots, t_n$  sont parcourus.

#### IV.1.2 Critère d'optimisation

Le critère que nous chercherons à optimiser sera la minimisation du temps de parcours de l'outil, par une recherche d'un trajet de coût minimum. On se propose donc de générer une trajectoire de l'outil telle que l'ensemble T soit réalisé.

### IV.2 APPROCHE DU PROBLEME:

#### IV.2.1 Caractéristique d'une tâche [30,31]

Soient  $x_i$  une décision et un ensemble  $T = \{ t_1, t_2, \dots, t_k \}$  de tâches à exécuter. Nous caractériserons une tâche par  $m$  points;  $q^1, q^2, \dots, q^m$  (appelés configuration) où  $q^j$  est un vecteur prenant valeur dans l'espace à  $n$  dimension du manipulateur:

$$- q^j = [ q^j_1, q^j_2, \dots, q^j_n ]^T \quad (IV.1)$$

où  $q^j_i$  est l'état de la  $i^{\text{ème}}$  coordonnée généralisée dans la  $j^{\text{ème}}$  configuration.

Notre manipulateur de découpe ne possède que deux degrés de libertés. L'évolution de l'outil se fait sur un plan. On ne travaille donc que sur  $R^2$  ( $n = 2$ ).

Chaque point  $m$  sera caractérisé par son vecteur:

$$- q^m = [q^m_1, q^m_2]^T \quad (IV.2)$$

Ainsi, une tâche de découpe  $t_j$  sera caractérisée par un couple de points  $(q^i, q^j)$  tel que:

$$- j = i + 1 \quad (IV.3)$$

**Remarque:**

Deux cas de positionnements de tâches peuvent être rencontrés, à savoir:

- la tâche correspond à un contour fermé et se trouve donc à l'intérieur de la surface de travail (ex. : tâche  $t_4$ ). Nous lui affecterons le couple de points  $(q^i, q^j)$  à l'extrémité gauche de la tâche. Cette décision relève de la facilité d'implémentation algorithmique. De plus, nous affectons deux points au lieu d'un, car cela nous évitera des confusions sur le début ou la fin d'une tâche lors de la formulation.

- la tâche correspond à un sous-contour de la pièce de vêtement ( ex. pièce n°1). Alors, aux endroits où l'axe  $oy$  coupe la pièce considérée, nous lui affecterons indifféremment un point en ordonnée sur  $oy$ , pour chaque intersection. Prenons l'exemple de la pièce n°1, elle coupe l'axe  $oy$  en deux points, donnant lieu à deux sous-contour, dont la tâche  $t_1$ . Ainsi, sur l'axe  $oy$ , nous affecterons le couple de points  $(q^2, q^3)$  pour  $t_1$ .

### IV.2.2 Caractéristique d'un trajet

Soit:

$$\Gamma = \{ O_1, O_2, \dots, O_s, \dots, O_p \}$$

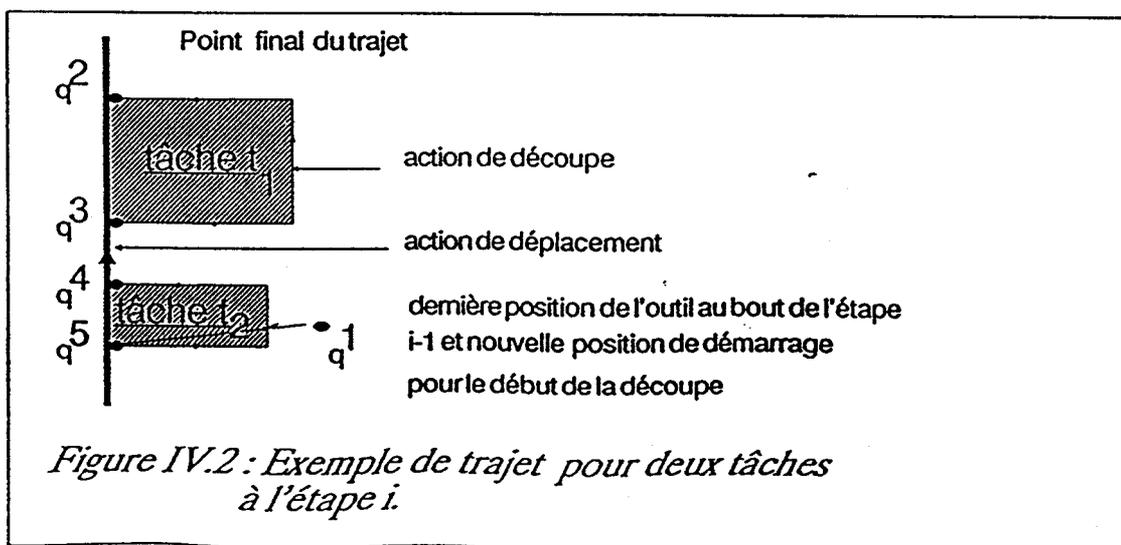
l'ensemble de tous les trajets possibles pour l'exécution d'un ensemble T. Un trajet  $O_s$  ( $s = 1, \dots, p$ ) consistera à rechercher, au travers d'un ensemble de tâches  $T = \{ t_1, t_2, \dots, t_k \}$ , une séquence de points, telle que toutes les tâches soient traitées une et une seule fois.

Pour une décision donnée  $x_i$ , on commencera le trajet à partir d'un point initial. Il correspond à la position du dernier point, pour la décision précédente  $x_{i-1}$ . Une fois le trajet déterminé, le point final de celui-ci deviendra le nouveau point initial, lors de la recherche d'un autre trajet, pour une nouvelle prise de décision  $x_{i+1}$ .

#### Exemple:

Soient:

- $T = \{ t_1, t_2 \}$  deux tâches à exécuter,
- $q^1$  la dernière position de l'outil
- $n$  le nombre de tâches (ici  $n = 2$ ) figure IV.2.



Un trajet  $O_5$  sera caractérisé par une séquence de points, permettant l'exécution des tâches  $t_1$  et  $t_2$ . De cet exemple, on tire l'ensemble  $\Gamma$  des trajets réalisables pour ces deux tâches  $(t_1, t_2)$ , tel que:

$$\Gamma = \{ O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8 \},$$

et pour lequel:

$$O_1 = (q^1, q^2, q^3, q^4, q^5),$$

$$O_2 = (q^1, q^2, q^3, q^5, q^4),$$

$$O_3 = (q^1, q^3, q^2, q^4, q^5),$$

$$O_4 = ..$$

$$O_5 = ..$$

$$O_8 = (q^1, q^5, q^4, q^3, q^2).$$

#### Remarque:

Il existe  $n!$  façons d'ordonnancer  $n$  tâches sur une machine. Cependant, dans notre cas, on peut exécuter chacune de ces tâches de deux façons différentes. En effet, on peut traiter une tâche en la commençant par le point  $q^i$  et en sortir par le point  $q^j$  ou vice versa.

Pour un ordonnancement réalisable parmi les  $n!$ , et en tenant compte de la remarque précédente; il existe  $2^n$  solutions réalisables pour cet ordonnancement.

D'où, pour les  $n!$  ordonnancements, le nombre total de solutions réalisables  $\sigma = \text{card}(\Gamma)$ :

$$\sigma = 2^n \cdot n!$$

Au travers de l'exemple à deux tâches, on voit que la cardinalité de  $\Gamma$  est relativement élevée ( pour 2 tâches,  $\sigma = 8$ ).

De par la valeur importante du nombre de solutions réalisables, nous constatons que le problème qui nous intéresse, nécessite une approche particulière. En effet, il ne nous est pas possible de calculer toutes les solutions du problème par une simple sommation des coûts sur chaque trajet. Après avoir caractérisé le terme de tâche et de trajet, le problème qui se pose est le suivant:

- Quel est, parmi l'ensemble  $\Gamma$ , l'élément qui nous permette de parcourir toutes les tâches en un temps minimum ?.

Notre problème fait partie de la classe des problèmes hautement combinatoires, vu le nombre de solutions réalisables. Pour résoudre un tel problème, nous avons à notre disposition des techniques dont la recherche opérationnelle. A partir d'un problème combinatoire donné, c'est à dire dans lequel il existe un grand nombre de solutions à visiter afin de déterminer l'optimalité ( min ou max), la recherche opérationnelle permet de résoudre de tels problèmes, grâce à des techniques d'études telles:

- La programmation dynamique,
- La programmation linéaire,
- La programmation non linéaire,
- La théorie des graphes etc..

Nous allons présenter la formulation de notre problème, pour lequel nous avons introduit deux variables, l'une correspondant au coût  $c_{ij}$  et l'autre à la variable décisionnelle  $x_{ij}$ .

### IV.2.3 Notion de coût

#### IV.2.3.1 Coûts associés au déplacement et à la découpe

Soient:

- \* deux tâches  $t_k, t_l$  caractérisées respectivement par leur couplet de points  $(i, j)$   $(r, s)$  de coordonnées  $(q^i, q^j)$   $(q^r, q^s)$ ,
- \* un point de départ  $q^1$  et un trajet  $O_m \in \Gamma$  tel que  $O_m = (q^1, q^i, q^j, q^r, q^s)$ .

i) action de découpe

On associera à chaque parcours, un coût  $c_{ij}$  correspondant au temps nécessaire pour passer de  $q^i$  à  $q^j$  ( ce parcours, sera appelé arc obligé), et il en sera de même pour l'arc obligé  $q^r$  à  $q^s$ . Cette phase peut être considérée comme une action de découpe. On affecte ainsi un coût à chaque tâche.

ii) action de déplacement

Pour passer d'une tâche à une autre, il existe plusieurs déplacements possibles. Ainsi, pour passer de la phase fin de découpe pour la tâche  $t_k$  à la phase début de découpe pour la tâche  $t_l$ , on peut emprunter le parcours  $(q^j, q^r)$ . On lui affecte un coût de passage  $c_{jr}$ . Cette phase peut être considérée comme une action de déplacement. On affecte ainsi un coût à chaque déplacement.

IV.2.3.1 Coûts associés à un trajet

On peut déduire, à partir des coordonnées de chaque tâches, une matrice de coût  $C = [c_{ij}]$ . Un trajet  $O_m$  représente l'ordonnancement des tâches à exécuter, ou une succession de points de passage de l'outil. Dans un trajet, en partant d'un point initial, on retrouve une action de découpe (caractérisée par deux points), puis une action de déplacement (passage d'une tâche à une autre), puis une action de découpe, ..... etc.. Le coût  $C_m$ , associé au trajet  $O_m$ , est déterminé par la somme des coûts liés aux différentes actions.

Exemple:

Si l'on reprend les deux tâches  $t_k$  et  $t_l$  et un trajet quelconque  $O_m = (q^1, q^i, q^j, q^r, q^s)$ , on peut lui associer un coût  $C_m$  tel que:

$$C_m = c_{1i} + c_{ij} + c_{jr} + c_{rs}$$

Note:

L'évaluation du coût peut être la distance Euclidienne, le temps estimé en fonction de la distance, etc... Afin de conserver un aspect général du problème, on considère tout d'abord que la matrice C est de type asymétrique ( $c_{ij} \neq c_{ji}$ ), et que le coût  $c_{ij}$  sera obtenu par un algorithme adéquat.

IV.2.4 Notion de variables décisionnelles

Le problème consiste à rechercher un trajet de coût minimum dans l'ensemble  $\Gamma$ . Nous affecterons une variable décisionnelle  $x_{ij}$  telle que:

$$x_{ij} = \begin{cases} 1 & \text{si le parcours de } q^i \text{ à } q^j \text{ ou l'arc } u = (i, j) \text{ est} \\ & \text{sélectionné} \\ 0 & \text{sinon.} \end{cases}$$

Ainsi, à chaque action est associé un coût  $c_{ij}$  et une variable décisionnelle, binaire,  $x_{ij}$ . Nous pouvons, dès à présent, donner le critère et les contraintes de notre problème. Afin de mieux aborder la formulation, nous présentons une approche graphique du problème.

IV.3 REPRESENTATION GRAPHIQUEIV.3.1 Définition [28]:i) définition:

Un graphe  $G = [X, U]$  est dit simple s'il ne possède pas deux arcs ayant même extrémité initiale et même extrémité terminale. Un arc  $u = (xy)$  d'un graphe peut être défini par le couple  $(x, y)$  de ses extrémités.

ii) définition:

Un graphe simple  $G = [X, U]$  est dit complet si  $(xy) \in U \Rightarrow (yx) \in U$ .

IV.3.2 Représentation graphique

L'ensemble des arcs se décompose de la manière suivante:

- une tâche représente une action de découpe que l'on représentera graphiquement par un arc en trait épais ( ou arc obligé),
- une action de déplacement, permettant le passage d'une tâche à une autre, sera représentée par un arc en trait fin.

On tire le graphe  $G = [X, U]$ , muni d'une valuation  $c(u)$  sur chacun de ses arcs, pour lequel:

- l'ensemble  $X$  des sommets est caractérisé par les indices des points d'entrée ou de sortie de chaque tâche ex.:  $X = \{1, i, j, \dots, w, x\}$ ,
- l'ensemble  $U = U_e \cup U_f$  se décompose en deux sous-ensembles tels que:
  - \*  $U_e = \{(i,j), \dots, (w,x)\}$  est le sous-ensemble des arcs épais ou arcs obligés,
  - \*  $U_f = \{(1,i), (1,j), \dots, (t,w)\}$  est le sous-ensemble des arcs fins,
- $c_{ij}$  est le coût ou la longueur associé à l'arc  $u = (i,j)$ .

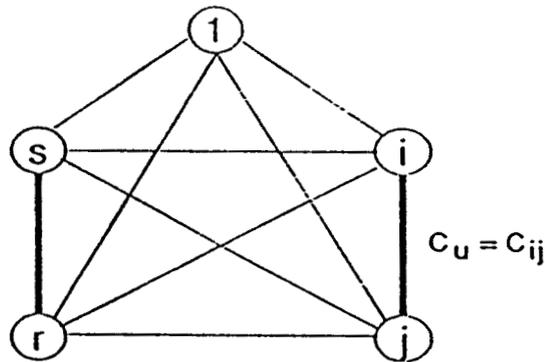


Figure IV.3: Graphe  $G = [X, U]$  associé à deux tâches

### Exemple:

Soit l'exemple à deux tâches  $t_k$  et  $t_p$ , pour lequel le graphe  $G = [X, U]$  est construit de la manière suivante (fig. IV.3):

- l'ensemble  $X = \{1, i, j, r, s\}$ ,
- $U_e = \{(i,j), (r,s)\}$
- $U_f = \{(1,i), (1,j), (1,r), (1,s), (i,r), (i,s), (j,r), (j,s)\}$ .

### Remarque:

Dans le cadre d'une étude générale, nous prendrons le coût  $c_{ij} \neq c_{ji}$ . Ceci nous conduira à une formulation générale du problème. Dans un deuxième temps, nous prendrons le coût  $c_{ij} = c_{ji}$ , ce qui correspond à l'application au processus réel.

#### IV.3.3 Représentation matricielle

De tout complet  $G$ , il est possible d'extraire une matrice d'incidence ( dans le cas orienté), mais aussi une matrice de coût. Dans notre cas, nous associerons au graphe  $G$ , la matrice de coûts asymétrique  $C = [c_{ij}]$  des coûts relatifs à chaque parcours.

#### IV.4 PREMIERE FORMALISATION DU PROBLEME Pbl: cas général

##### IV.4.1 Problème du voyageur de commerce asymétrique

Notre approche du problème se base sur celle du voyageur de commerce dont l'énoncé est le suivant:

- Un voyageur de commerce a "w" villes à visiter dont la distance de parcours est connue. Comment ce "voyageur" va-t-il organiser sa tournée de manière à visiter chacun de ses clients une et une seule fois, tout en parcourant: un minimum de kilomètres. Autrement dit comment déterminer un circuit Hamiltonien de valeur minimale dans un graphe valué.

Malgré des algorithmes de plus en plus efficaces et des ordinateurs de plus en plus rapides, ce problème, classé parmi les problèmes NP complets, reste difficile à résoudre exactement (pour un nombre de villes supérieur à la centaine). Dans la littérature [32,33], On formule ce problème par un programme linéaire en variables entières, dont la formulation est:

(PLNE1)

$$\text{minimiser } z_1 = \sum_{i=1}^w \sum_{j=1}^w c_{ij} x_{ij}$$

$$\sum_{j=1}^w x_{ij} = 1 \quad \text{pour } i = 1, \dots, w \quad (1)$$

$$\sum_{i=1}^w x_{ij} = 1 \quad \text{pour } j = 1, \dots, w \quad (2)$$

$$x_{ij} = 1 \text{ si l'arc } (i,j) \text{ de coût } c_{ij} \\ \text{appartient au circuit optimal} \\ = 0 \text{ sinon}$$

##### i) critère

Le critère à optimiser est la minimisation de la longueur totale.

## ii) contraintes

- (1) exprime que parmi l'ensemble des "i" arcs incident au sommet "j" ( ou ville), il ne peut en emprunter qu'un seul.
- (2) exprime que parmi l'ensemble des "j" arcs incidents à l'arc "i", il ne peut en emprunter qu'un seul.

### IV.4.2 Formulation du problème Pb2

Notre formalisation du problème reste identique, car nous recherchons un plus court chemin au travers d'un ensemble de tâches à exécuter. Cependant il nous a fallu introduire une contrainte supplémentaire. Le critère et les contraintes, spécifiques à notre problème, sont les suivants:

#### i) critère

Le but de notre générateur est de fournir, au module de commande d'axes, parmi l'ensemble  $\Gamma$  des trajets, celui qui présente un coût minimum. Le critère, que nous avons retenu, est la minimisation de la longueur totale sur un trajet. Cette longueur correspond à la somme des longueurs sur chacune des actions qui compose le trajet ( action de déplacement, de découpe, de déplacement, etc...). [ algorithme développé dans [31] ].

Nous conservons la formulation (1), pour laquelle le nombre de villes à parcourir "w" est en rapport avec le nombre de tâches n.

De plus, pour une tâche, on dispose de deux points (i,j) de coordonnées  $(q^i, q^j)$ . Pour n tâches, on en déduit les  $2.n$  points de passages sur le plan de placement. Une trajectoire de l'outil consiste à déterminer un trajet regroupant les  $2.n + 1$  points de passage ( en incluant le point initial de coordonnée  $q^1$ ).

La formulation de notre critère est donc la suivante:

Soit  $m$  le nombre de points de coordonnées  $q^i$  d'un ensemble de  $n$  de tâches à exécuter:

$$\text{minimiser } z_2 = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij}$$

avec  $m = 2 \cdot n + 1$ .

### ii) contraintes

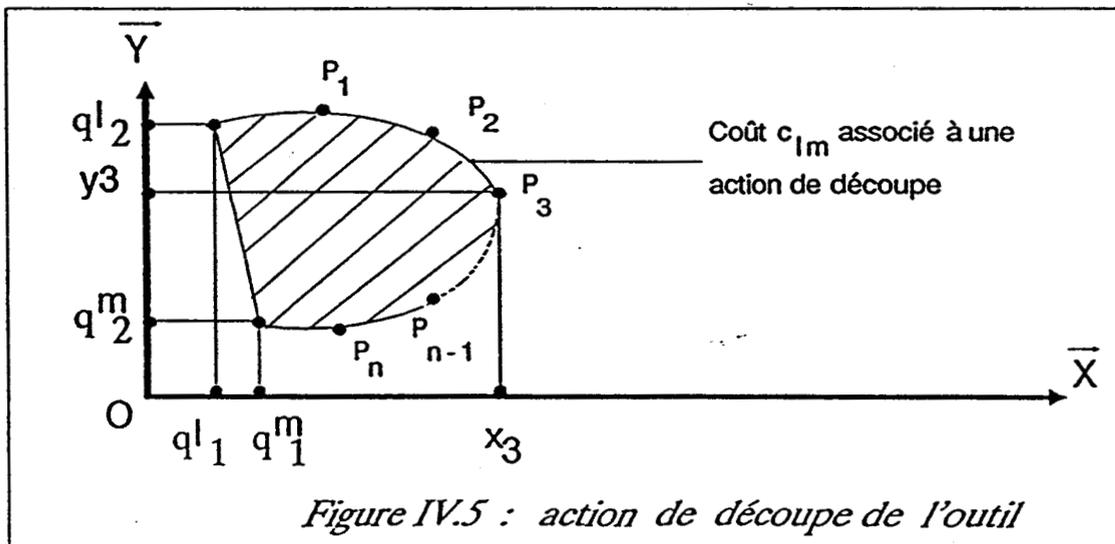
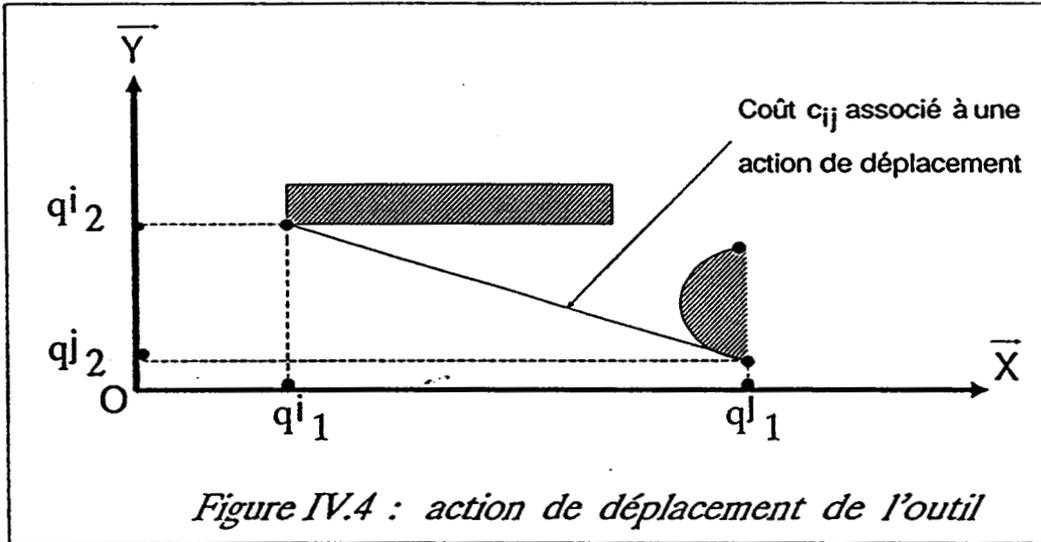
- Les deux premières contraintes du problème concernent l'unicité des décisions, c'est à dire qu'en un point "i" de coordonnée  $q^i$ , il ne peut y avoir qu'un parcours d'arrivée et un de départ, d'où les contraintes (4) et (5), pour lesquelles la donnée  $w$  est remplacée par  $m$ .

- La troisième contrainte traduit l'existence d'un arc obligé. Etant donné, que nous avons caractérisé une tâche  $t_j$  par ses coordonnées  $(q^i, q^j)$  telle que  $j = i + 1$ , On peut donc formuler cette troisième contrainte par:

$$x_{ij} + x_{ji} = 1 \quad \text{pour } i = 2, 4, \dots, 2 \cdot n \\ \text{et } j = i + 1.$$

D'où le programme linéaire PLNE2 associé au problème Pb2 dans le cadre général:

(PLNE2)	$\text{minimiser } z_2 = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij}$	
	$\sum_{j=1}^m x_{ij} = 1 \quad \text{pour } i = 1, \dots, m \quad (4)$	
	$\sum_{i=1}^m x_{ij} = 1 \quad \text{pour } j = 1, \dots, m \quad (5)$	
	$x_{ij} + x_{ji} = 1 \quad \text{pour } i = 2, 4, \dots, 2 \cdot n \\ \text{et } j = i + 1. \quad (6)$	



#### IV.4.3 Conclusion

Ce problème correspond à un programme linéaire en variables bivalentes. Il est important de remarquer que le nombre de contraintes est relativement important ( $m^2$ ). Ceci nous conduira à un encombrement mémoire en rapport avec la taille de la matrice des contraintes. La formulation suivante aura pour but de réduire cette matrice.

#### IV.5 MODIFICATION DE LA FORMULATION: cas du processus de découpe

Dans le paragraphe précédent, nous avons présenté une étude générale pour la recherche d'un trajet de coût minimum, pour laquelle la matrice de coût était de type asymétrique.

La découpe se fait par déplacement de l'outil, ce qui induit la détermination préalable de toutes les longueurs de chacun de ces déplacements. C'est pourquoi, afin de nous rapprocher au mieux des conditions réelles de fonctionnement du processus, il nous semble indispensable de simplifier le problème, en prenant comme hypothèse, que la matrice C est de type symétrique. En effet, une simplification évidente, nous permettant de réduire les temps de calcul, consiste à prendre la distance euclidienne entre chaque couple de points  $(q^i, q^j)$ . La détermination de ce coût se fait comme suit:

a) pour une action de déplacement ( fig. IV.4):

Pour se déplacer d'une tâche à une autre, on passe d'un point  $q^i$  de coordonnées  $(q^i_1, q^i_2)$  à un point  $q^j$  de coordonnées  $(q^j_1, q^j_2)$ . Le coût associé à cette action de déplacement est:

$$c_{ij} = \sqrt{(q^j_2 - q^i_2)^2 + (q^j_1 - q^i_1)^2}$$

b) pour une action de découpe ( fig. IV.5):

Une tâche se compose d'une succession de points  $P_k$  de coordonnées  $(x_k, y_k)$

et de deux points  $(q^l, q^m)$  de coordonnées respectives  $(q^l_1, q^l_2), (q^m_1, q^m_2)$  avec  $m = l+1$ . Le coût associé à cette action de découpe est:

$$c_{lm} = \sqrt{(q^l_2 - y_1)^2 + (q^l_1 - x_1)^2} \\ + \sqrt{(q^m_2 - y_n)^2 + (q^m_1 - x_n)^2} \\ + \sum_{k=1}^{n-1} \sqrt{(y_{k+1} - y_k)^2 + (x_{k+1} - x_k)^2}$$

Avec  $c_{lm} = c_{ml}$ .

#### IV.5.1 Problème du "voyageur de commerce symétrique"

Tout comme pour le cas général, nous nous sommes basés sur la formulation du problème du voyageur de commerce, non plus asymétrique, mais symétrique. A ce problème est associé un graphe  $G = [X, U]$  pour lequel:

- X représente l'ensemble des sommets ( ou villes)
- U l'ensemble des arêtes entre chaque sommet.
- à chaque arc  $u = (i, j)$  est affecté un poids ou une longueur correspondant au coût  $c_{ij}$  d'un parcours tel que  $c_{ij} = c_{ji}$ .

La formulation, par un programme linéaire en variables bivalentes, du voyageur de commerce symétrique est la suivante [34,35]:

$$(PLNE3) \quad \left[ \begin{array}{l} \min z_3 = \sum_{i=1}^n \sum_{j>i}^n c_{ij} x_{ij} \\ \sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = 2 \quad i = 2, \dots, n \quad (7) \\ \sum_{j=2}^n x_{1j} = 2 \quad (8) \\ \sum_{1 \leq i < j \leq n} x_{ij} = n \quad (9) \end{array} \right.$$

i) critère

Son expression générale est réduite, car toutes les variables décisionnelles symétriques sont supprimées (ex. :  $x_{32}$  est supprimé car on utilise  $x_{23}$  etc...).

ii) contraintes

- (7) exprime que parmi les arêtes incidentes au sommet  $i$  on ne peut en prendre que 2,
- (8) exprime que parmi l'ensemble des " $j$ " arêtes incidentes au sommet " $i$ " on ne peut en prendre que "2",
- (9) exprime que pour tout circuit, il nous faut  $m$  variables décisionnelles.

IV.4.4 Formulation simplifiée de Pb2

Notre formulation diffère de celle présentée précédemment. En effet, dans le cadre général de la formulation du "voyageur de commerce symétrique", celui-ci peut emprunter n'importe quel trajet à partir de la ville où il se trouve. Dans notre cas, le choix des phases de déplacements engendre l'exécution d'une tâche (arc obligé). On découpe celle-ci en la démarant par un point initial de coordonnée  $q^i$  et on en sort par un point final de coordonnée  $q^j$ , ou vice versa.

Nous imposerons donc à toutes les variables décisionnelles, correspondant aux tâches ou actions de découpe, d'être à 1. Car le choix de déplacement fixe implicitement l'action de découpe, ce qui se traduit par:

$$x_{ij} \begin{cases} = 1 \text{ si } (i,j) \text{ représente le couple associé à une tâche} \\ \in \{0,1\} \text{ sinon} \end{cases}$$

Chaque tâche étant caractérisée par un couple de points  $(q^i, q^j)$  tel que  $j = i + 1$ , on peut donc introduire la nouvelle contrainte:

$$x_{ij} = 1 \quad \begin{array}{l} \text{pour } i = 2, 4, \dots, 2*n \\ \text{et } j = i + 1. \end{array}$$

D'où, la formulation simplifiée du problème Pb1, pour une matrice de coûts symétriques, que l'on formalisera sous le programme linéaire suivant (PLNE4):

(PLNE4)	$\min z_4 = \sum_{i=1}^m \sum_{j>i}^m c_{ij} x_{ij}$	
	$\sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = 1 \quad i = 2, \dots, m \quad (10)$	
	$\sum_{j=2}^m x_{1j} = 2 \quad (11)$	
	$\sum_{1 \leq i < j \leq m} x_{ij} = h \quad (12)$	
	$x_{ij} = 1 \quad \begin{array}{l} i = 2, 4, \dots, 2 \cdot n \\ \text{et } j = i + 1. \end{array} \quad (13)$	
	$x_{ij} \in \{0, 1\}$	

Où:

- (10): parmi les arêtes incidentes au sommet "i" il ne peut y en avoir qu'une,
- (12): pour former un circuit ou une solution réalisable, il nous faut m variables décisionnelles:

$$* m = 2 \cdot n + 1,$$

Or la contrainte (13) supprime les n variables relatives aux actions de découpe. D'où, le nombre réduit de variables sur un trajet:

$$* h = m - n = n + 1.$$

## IV.5 CONCLUSION

L'étude de la gestion de l'avance du tissu, nécessite la donnée d'un coût correspondant à chaque prise de décision. Dans ce chapitre, nous avons abordé sa détermination par la recherche d'un trajet de coût minimum. En effet, pour chaque décision  $x_i$ , on présente dans la zone de travail accessible de l'outil, un ensemble de contours ou sous-contours à découper.

La formulation de ce problème se base sur celle du "voyageur de commerce asymétrique ou symétrique". La recherche d'un trajet, au travers d'un ensemble de points, est donc formulable par un programme linéaire en variables bivalentes.

En conclusion de ce chapitre, nous constatons que, quelle que soit la formulation proposée, le problème étudié présente une combinatoire élevée, et nous conduit à une formulation complexe.

Cependant, dans le chapitre suivant, nous montrerons qu'il est possible de réduire la taille de la matrice des contraintes. Des techniques, propres à la Programmation Mathématique, nous conduisent à élaborer une formulation plus concise et plus simple d'emploi.

**CHAPITRE V**  
**RÉSOLUTION**



## INTRODUCTION

Au chapitre III et IV nous avons montré qu'il était possible de formaliser nos problèmes comme des programmes linéaires en variables bivalentes.

Dans ce chapitre, nous présentons l'approche classique des méthodes de résolution pour ce type de programmes. Nous reformulons le programme linéaire afin de le ramener à un problème classique de recherche opérationnelle, à savoir le problème du "KNAPSACK ou SAC-A-DOS" en égalité.

Pour ce faire, nous donnons la méthode de passage d'un programme multicontraintes à un programme à contrainte unique. Nous formulons le problème du Knapsack et de la contraction d'un système diophantien en une contrainte unique. Nous modifions notre problème de minimisation en un problème de maximisation, après une réindexation des coûts et variables.

Enfin, après une synthèse des méthodes de résolutions du Knapsack, nous présentons un algorithme efficace pour la résolution de ce nouveau problème. Cet algorithme se base sur une méthode de type "énumération implicite". On recherche une solution du problème par une procédure de séparation et d'évaluation progressive ( P.S.E.P.).

En un noeud de l'arborescence, l'évaluation par défaut obtenue pour résoudre le sous-problème considéré se base sur la Relaxation Lagrangienne. A cette étape, suivant la valeur de l'évaluation, on décide de poursuivre ou non la procédure de descente dans l'arborescence associée au problème. C'est la notion de séparation qui est alors mise en évidence. De plus, lors de la détermination d'une solution réalisable meilleure que celle déterminée initialement, on peut arrêter la procédure de calcul et obtenir ainsi une solution du problème.

## V.1 POSITION DU PROBLEME

### V.1.1 Donnée du problème

Dans les deux chapitres précédents, les formulations proposées sont des programmes linéaires en variables bivalentes. Les problèmes de la gestion du processus d'avance et de la génération de la trajectoire de l'outil sont formulés comme suit:

$$(Pb) \quad \left[ \begin{array}{l} \text{Optimiser} \quad C \cdot X \\ \text{sous les} \\ \text{contraintes} \quad AX = B \end{array} \right. \quad (1)$$

où:

- A est une matrice de format  $n \times m$
- C est un vecteur ligne de format  $1 \times m$
- B un vecteur colonne de format  $n \times 1$

telles que :  $c_j, a_{ij}, b_i \in \mathbb{R}$  et  $x_{ij} \in \{0,1\}$  pour  $i = \{1, \dots, n\}$  et  $j = \{1, \dots, m\}$

i) Note [28]:

Suivant la valeur des coefficients du vecteur colonne B et du type de contrainte ( inégalité ou égalité), on a les programmes linéaires suivant:

- |                         |             |                                |
|-------------------------|-------------|--------------------------------|
| - couplage généralisé : | $AX \leq B$ | $b_i = 1 \forall i$            |
| - partitionnement:      | $AX = B$    | $b_i = 1 \forall i$            |
| - recouvrement:         | $AX \geq B$ | $b_i = 1 \forall i$            |
| - affectation:          | $AX = B$    | $b_i \in \mathbb{N} \forall i$ |

ii) définition [35]:

Les problèmes du "Knapsack", sont des problèmes linéaires en nombres entiers particuliers s'écrivant sous la forme:

$$(Pk) \quad \left[ \begin{array}{l} \text{maximiser } z = c \cdot x \\ \text{sous la} \\ \text{contrainte } a \cdot x \leq b \end{array} \right. \quad (2)$$

où:

- a et c sont deux vecteurs ligne de format 1.n,
- b est une valeur entière donnée,
- x est le vecteur colonne inconnu  $x_j$  de format n.1,

tels que :  $c_j, a_j, x_j, b \in \mathbb{N}$  pour  $j = \{1, \dots, n\}$ .

V.1.2 Approche classique

La plupart des problèmes d'ordonnancement ou de découpe peuvent être modélisés par un programme linéaire en nombres entiers. Leur résolution se base généralement sur les méthodes suivantes:

- simplexe, simplexe révisé [36],
- coupes de Gomory [16],
- relaxation Lagrangienne [26],
- congruence décroissante [37].

Différentes techniques de résolution de (Pb) peuvent donc être employées; dont les quatres suivantes [38]:

- A) emploi des techniques basées sur la réduite de Smith qui permet de se ramener à une matrice triangulaire inférieure, infiniment plus maniable en pratique.
- B) utilisation de la méthode simpliciale (simplexe révisé). Dans cette méthode, à chaque itération, on cherche une colonne pivot à laquelle correspond un coût réduit. Si celui-ci est positif, la colonne correspondante est la colonne pivot, sinon on a trouvé la solution optimale.
- C) transformation en un problème du Knapsack équivalent par contraction de la matrice des contraintes en une contrainte unique, dont l'ensemble des solutions entières réalisables, est identique à celui du système initial.
- D) choix d'un sous-ensemble d'indices de colonnes  $I \subset \{1, \dots, n\}$  tel que  $A^I$  soit diagonalisable ( lorsque la matrice  $A$  le permet).

En théorie, la méthode (B) permet de considérer des problèmes de taille quelconque, mais en pratique, elle est très vite limitée par la grandeur des entiers obtenus ou par le temps de calcul.

C'est pourquoi, nous avons repris la méthode (C) utilisée lors de la résolution d'un problème d'affectation [38, 39]. Cette technique consiste à ramener le système de contrainte (1) à une contrainte unique. Les aspects suivants ont motivé notre choix:

- il est inutile de mémoriser la matrice "A", comme pour la méthode (B),
- [37] propose un algorithme de résolution très performant et d'implémentation facile,

- dans les deux cas de configurations où les paramètres suivants sont connus:

\* problème (Pb1): le nombre de sommets du réseau R,

\* problème (Pb2): le nombre de tâches à exécuter,

la contrainte unique n'est pas à recalculer en temps réel, mais est connue au préalable. En effet, nous calculons une seule fois la valeur des coefficients de cette contrainte, suivant les paramètres du problème considéré.

## V.2 UNE APPROCHE DU PROBLEME

### V.2.1. Différentes étapes de l'approche pour la résolution de (Pb)

L'approche que nous présentons se base sur la méthode de contraction de contraintes. Les différentes étapes de notre transformation sont (fig. V.1):

#### 1) problème initial:

Notre problème consiste à minimiser une fonction de coût linéaire  $C.X$  telle que l'ensemble des contraintes  $A.X = B$  soit vérifié.

#### 2) réindexation des coûts et variables:

Pour des raisons d'implémentation algorithmiques, on réindexe l'ensemble des coûts et des variables qui passent ainsi de 2 indices à 1.

#### 3) contraction des contraintes:

Notre approche réside dans la contraction du système d'équation  $A.X = B$  en une contrainte unique  $a.x = b$ . Pour ce faire, nous avons utilisé les techniques de contraction présentées en annexe 1. Ensuite en posant  $M = -C$ , on passe à un problème de maximisation.

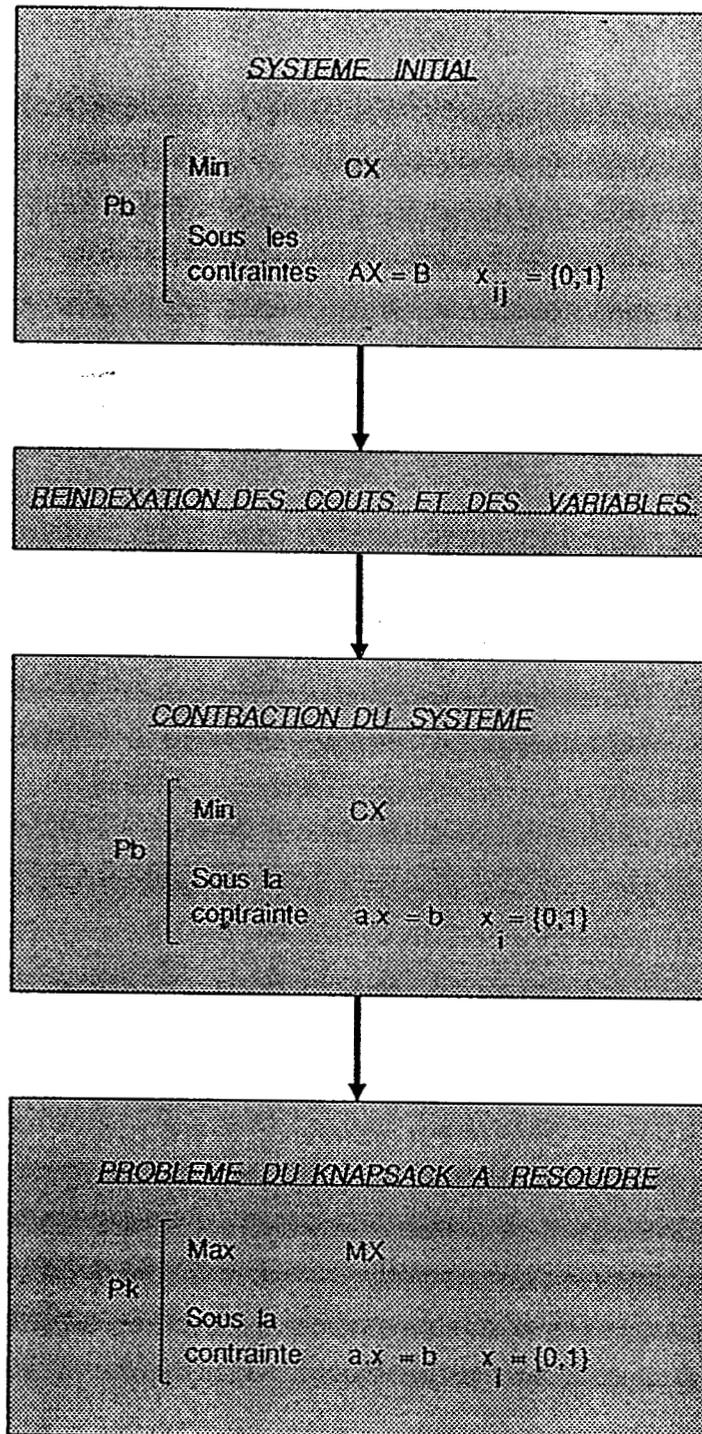


Figure V.1 : Reformulation du problème de base

4) résolution du nouveau problème:

Le problème ainsi obtenu se présente comme suit:

$$(B) \quad \left[ \begin{array}{ll} \text{Max} & z = M \cdot X \\ \text{Sous} & n \\ \text{la contrainte} & \sum_{i=1}^n a_i \cdot x_i = b \quad x_i \in \{0,1\} \quad \forall i \end{array} \right.$$

A cette étape, après avoir transformé notre problème en un problème du "knapsack en égalité", on se propose de le résoudre par énumération implicite.

V.2.2 Transformation de la formulation du problème de base (Pb)

On présente dans ce paragraphe, la méthode de passage du programme (Pb) à un problème du "Knapsack".

V.2.2.1 Rappel

Mathews [39] en 1897 avait démontré qu'il était possible de transformer un système d'équations à coefficients entiers tous strictement positifs en une équation unique équivalente ( possédant un ensemble de solutions entières réalisables identiques à celui du système).

L'élaboration de méthodes performantes de résolution du problème du Knapsack semble pouvoir accroître l'intérêt de résultats récents généralisant et améliorant de façon sensible ceux de Mathews (Padberg [40], Bradley [41], Glover et Woolsey [42], Glover [43]).

Il est primordial de noter que l'équation unique, équivalente au système, est obtenue à la suite d'une cascade de combinaisons linéaires d'équations prises deux à deux. Ces coefficients, qui sont donc fonction à la fois de la taille du système initial ( nombre de contraintes et de variables) et des valeurs des coefficients des contraintes initiales, peuvent prendre des valeurs extrêmement grandes. Même pour des systèmes de petite taille mais à coefficients élevés, la manipulation d'entiers aussi grands peut conduire à des difficultés sur le plan informatique. Ceci correspond au grand inconvénient de la méthode, comme le précise aussi [37].

Cependant, il est à noter que les coefficients de notre programme ne sont pas de taille élevée car l'ensemble des contraintes possède des coefficients bivalents. Nous sommes, dès lors, affranchis de cette contrainte d'encombrement mémoire, pour les coefficients de la contrainte issue de la contraction des contraintes initiales [38]. (pour fixer les idées, citons l'exemple de 22 contraintes d'un problème de 177 variables [39] possédant environ 7,3 % d'éléments égaux à 1 qui aboutit à des coefficients non nuls compris entre 0 et  $3,5 \cdot 10^{12}$ , en utilisant les résultats de Bradley).

Les résultats de Bradley, exposés et généralisés au cas d'inéquations, n'imposent aucune restriction de signe ou de non nullité pour les coefficients de l'équation résultante ( à l'inverse de Glover, ni pour ceux des équations initiales, qui peuvent donc être utilisées sans aucune modification préalable (à l'inverse de Glover et Woolsey).

Nous avons donc utilisé les résultats de [44] pour aboutir à un problème du Knapsack en égalité à variables bivalentes devant vérifier, outre la contrainte d'égalité, des conditions de type "partitionnement". La méthode de résolution fera l'objet des paragraphes suivants.

### V.2.2.2 Transformation de la formulation du problème

A partir des résultats dûs à Bradley mais aussi généralisés par [44] (sur la G-contraction), on obtient un problème équivalent au problème (Pb) en remplaçant le système de contraintes (1) par une seule contrainte.

### Formulation générale du problème de contraction

#### i) définition [44]

On appelle équation diophantienne, une équation telle que:

$$- f(x) = 0 \text{ avec } x_j \in \mathbb{R} \text{ et } f(x) \in \mathbb{Z}$$

( dans notre cas  $x_j \in \{0,1\}$  )

#### Exemple:

$$x_1 + x_2 + x_3 - 3 = 0 \quad \text{avec } x_1, x_2, x_3 \in \mathbb{R}$$

Etant donnés "n" un entier naturel (  $n \geq 2$  ) et  $f_i$  (  $i = 1, \dots, n$  ) des applications de l'ensemble X dans Z, où X est un ensemble arbitraire de  $\mathbb{R}^m$ , considérons le système suivant formé de "n" équations diophantiennes:

$f_1(x)$	=	0
$f_2(x)$	=	0
$f_3(x)$	=	0
⋮		
⋮		
⋮		
$f_n(x)$	=	0
$x$	∈	X

Notons  $S$  l'ensemble des solutions réalisables de ce système :

$$- S = \{ x \in X \mid f_i(x) = 0, i = 1, \dots, n \}$$

et posons:

$$- \bar{S} = \{ x \in X \mid \sum_{i=1}^n \delta_i f_i(x) = 0 \}$$

où  $\delta_1, \dots, \delta_n$  sont des entiers relatifs non nuls, appelés "multiplicateurs admissibles".

Le problème de contraction consiste à déterminer une combinaison linéaire de toutes les équations du système qui n'admet pour solutions réalisables que celles du système dans l'ensemble  $X$ . Il se formule donc de la manière suivante:

déterminer $(\delta_1, \dots, \delta_n) \in \mathbb{Z}_*^n$ tel que: $S = \bar{S}$
---

**Remarque:**

Il est évident que l'ensemble  $S$  est inclus dans l'ensemble  $\bar{S}$ ; le problème consiste donc à déterminer des entiers non nuls  $\delta_1, \dots, \delta_n$  tels que l'ensemble  $\bar{S}$  soit inclus dans l'ensemble  $S$ .

**Note:**

En annexe 1 sont exposés les résultats de Bradley qui permettent de contracter un système d'équations en une contrainte unique.

### V.2.3 Réindexation et contraction

#### V.2.3.1 Réindexation

Avant d'appliquer la contraction au problème (Pb), nous allons modifier sa présentation. En effet, en détaillant le problème (Pb) du paragraphe précédent, notre problème se présente comme suit:

$$(Pb) \quad \left[ \begin{array}{l} \text{minimiser } Z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \\ \text{sous les} \\ \text{contraintes} \quad \sum_{j=1}^m a_{ij} x_{ij} = b_i \quad \text{pour } i = 1, \dots, n \end{array} \right.$$

avec  $a_{ij}, x_{ij}, b_i \in \{0,1\}$  pour tout  $i, j$ .

Pour des raisons évidentes d'implémentation, nous réindexons les variables et les coûts de la manière suivante:

$$\begin{array}{l} -x_{11} = x_1 \quad c_{11} = c_1 \quad \text{et} \quad a_{11} = a_1 \\ -x_{12} = x_2 \quad c_{12} = c_2 \quad \text{et} \quad a_{12} = a_2 \\ - \\ - \\ -x_{nm} = x_k \quad c_{nm} = c_k \quad \text{et} \quad a_{nm} = a_k \end{array}$$

et en posant:

$$f_j = \sum_{i=1}^m a_{(j-1) \cdot m + i} \cdot x_{(j-1) \cdot m + i} - b_j = 0 \quad \text{pour } j = 1, \dots, n$$

on tire:

$$\left[ \begin{array}{l}
 \min \quad CX = \sum_{j=1}^n \sum_{i=1}^m c_{(j-1).m+i} \cdot x_{(j-1).m+i} \\
 \text{sous les} \\
 \text{contraintes} \\
 \sum_{i=1}^m a_{(j-1).m+i} \cdot x_{(j-1).m+i} - b_j = 0 \text{ pour } j = 1, \dots, n \quad (3) \\
 x \in (0,1)
 \end{array} \right.$$

avec  $i \in N, c_i \in R$ .

Soit:

$$\left[ \begin{array}{l}
 \min \quad CX \\
 \text{sous les} \\
 \text{contraintes} \quad f_j = 0 \text{ pour } j = 1, \dots, n \quad (3) \\
 x_j \in (0,1) \quad \forall j
 \end{array} \right.$$

Pour plus de clarté, le problème se présente comme suit après réindexation:

(Pb)	$  \left[ \begin{array}{l}  \min \quad CX = \sum_{i=1}^n c_i \cdot x_i \quad (4) \\  \text{sous les} \\  \text{contrainte } f_i(x) = 0 \text{ pour } i=1, \dots, n \quad (5) \\  x \in (0,1) \quad (6)  \end{array} \right.  $
------	--

V.2.3.2 Contraction et passage en maximisation

La contraction du système d'équations (5) nous conduit au problème équivalent:

(Pb)	[	Optimiser $CX = \sum_{i=1}^n c_i \cdot x_i$ sous la contrainte $\sum_{i=1}^n \delta_i f_i(x) = 0$ (7) $x \in \{0,1\}$	]
------	---	--	---

Exemple:

Dans cet exemple, nous donnons une application de la méthode de contraction appliquée à un système d'équations.

Soit le problème de minimisation:

(Pb)	[	$\min CX = [2, 3, 4, 9, 2, 3, 10, 5, 2] \cdot [x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33}]^T$ s.c $A \cdot X = B$ $\left[ \begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \cdot [x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33}]^T = [1, 1, 1]^T$ $x_{ij} = \{0,1\} \forall i, j$	]
------	---	--	---

1) réindexation des variables et des coûts:

On tire le système diophantien associé au système  $AX = B$ , pour lequel  $n = m = 3$ :

$$f_1 = x_{11} + x_{12} + x_{13} - 1 = 0$$

$$f_2 = 0.x_{21} + x_{22} + x_{23} - 1 = 0$$

$$f_3 = x_{31} + 0.x_{32} + x_{33} - 1 = 0$$

après réindexation:

$$f_1 = x_1 + x_2 + x_3 - 1 = 0$$

$$f_2 = 0.x_4 + x_5 + x_6 - 1 = 0$$

$$f_3 = x_7 + 0.x_8 + x_9 - 1 = 0$$

2) contraction du système

A partir des résultats [42], les multiplicateurs admissibles  $\delta_i$  ( $i=1, 2, 3$ ) sont obtenus en les prenant premiers entre eux. On tire l'équation unique :

$$a.x = b \iff x_1 + x_2 + x_3 + 0.x_4 + 2.x_5 + 2.x_6 + 3.x_7 + 0.x_8 + 3.x_9 = 6$$

3) passage en maximisation

on obtient le problème équivalent à (Pb):

$$(Pk) \left[ \begin{array}{l} \max \quad MX = [-2, -3, -4, -9, -2, -3, -10, -5, -2]. [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9]^T \\ \text{s.c} \quad a.x = b, \quad x_1 + x_2 + x_3 + 0.x_4 + 2.x_5 + 2.x_6 + 3.x_7 + 0.x_8 + 3.x_9 = 6 \\ \quad \quad \quad x_i = \{0, 1\} \quad \forall i \end{array} \right.$$

### V.3 RESOLUTION

Pour plus de clarté, nous considérerons que le problème posé se présente de la manière suivante:

$$(B) \quad \left[ \begin{array}{l} \text{maximiser} \quad \sum_{j=1}^n c_j x_j \\ \text{sous} \\ \text{la contrainte} \quad \sum_{j=1}^n a_j x_j = b \\ x_j \in \{0,1\} \end{array} \right.$$

Après avoir transformé notre problème en un problème du Knapsack en égalité, nous allons aborder les méthodes de résolution. Celles-ci sont généralement basées sur la théorie de la dualité. Le déroulement de notre approche est représenté figure V.2.

#### V.3.1 Rappels sur la théorie de la dualité en optimisation combinatoire

En annexe 2, on présente les principaux résultats sur la théorie de la dualité en optimisation combinatoire. Bien que l'utilisation des méthodes duales ait été proposée par [...], son utilisation n'est devenue courante que beaucoup plus récemment avec l'amélioration des techniques d'optimisation de fonctions non partout différentiables (relaxation Lagrangienne, algorithmes de sous-gradients). De nombreux problèmes combinatoires ont été résolus de façon exacte ou approchée par l'emploi de la théorie de la dualité et parmi eux l'on peut noter:

- . le problème du voyageur de commerce,
- . le problème du sac-à-dos,
- . le problème du sac-à-dos quadratique,

- . les problèmes de multiflots dans les réseaux,
- . les problèmes de localisation et classification,
- . le problème de l'emploi du temps,
- . le problème d'affectation généralisé,
- . le problème de plus court chemin avec contrainte,
- . les problèmes de partitionnement et recouvrement,
- . les problèmes d'intersection et de parité de matroïdes,
- . les problèmes de planification de la production,
- . les problèmes de planification des investissements,
- . le problème de l'arbre minimum avec contraintes sur les degrés,
- . le problème l'arbre minimum avec contraintes de capacité,
- . les problèmes en nombres entiers généraux.

On trouvera dans les références [45, 37] des exposés de synthèse sur le sujet, ainsi qu'une vaste bibliographie. On présente en annexe 2 les principaux résultats théoriques qui seront utilisés ultérieurement; pour davantage de détails on renvoie le lecteur à [46].

### V.3.1.1 Relaxation Lagrangienne et théorie de la dualité [47]

On rappelle la formulation du problème posé:

(B)	$\begin{array}{ll} \text{maximiser} & c \cdot x \\ \text{sous} & \\ \text{la contrainte} & a \cdot x = b \\ x \in & \{0,1\} \end{array}$
-----	--

Une relaxation de ce problème sera étudiée en le reformulant de la façon suivante:

$$(B) \quad \left[ \begin{array}{l} \text{maximiser} \quad c \cdot x \\ \text{sous} \\ \text{la contrainte} \quad g(x) = b - a \cdot x = 0 \\ x \in [0, 1] \end{array} \right.$$

On associe un multiplicateur de Lagrange  $\mu \geq 0$  à la contrainte de (B). Avec ce multiplicateur, on définit le Lagrangien  $L(x, \mu)$  par:

$$L(x, \mu) = c \cdot x + \mu \cdot g(x) = \mu \cdot b + (c - \mu \cdot a) \cdot x$$

et le problème  $P(\mu)$  par

$$P(\mu) \quad \left[ \begin{array}{l} w(\mu) = \max L(x, \mu) \\ \text{s.c } x \in [0, 1] \end{array} \right.$$

D'après le théorème de la dualité (annexe 2), on sait que l'on peut obtenir une borne supérieure pour  $c \cdot x^*$  (solution optimale de (B)), en résolvant le problème dual:

$$(D) \quad \left[ \begin{array}{l} w(\mu^*) = \min w(\mu) \\ \text{s.c } \mu \geq 0 \end{array} \right.$$

### Propriété

La fonction  $w(\mu)$  est une fonction linéaire par morceaux.

La solution du problème dual revient alors à minimiser une fonction concave linéaire par morceaux.

V.3.1.2 Particularité de (B)

Dans le cas particulier du problème du Knapsack, [38] montre que parmi l'ensemble des valeurs possibles de  $\mu$ , il en existe une meilleure que les autres. L'obtention de cette valeur se fait de la façon suivante:

On considère le problème du Knapsack:

$$(B) \quad \left[ \begin{array}{ll} \text{maximiser} & c \cdot x \\ \text{sous} & \\ \text{la contrainte} & a \cdot x = b \\ x_i \in \{0,1\} & i = 1, \dots, n \end{array} \right. \quad (1)$$

On détermine un sous-ensemble  $M$  de  $\{1, \dots, n\}$  d'indices tel que:

$$\left[ \begin{array}{l} \sum_{j \in M} a_j \leq b < \sum_{j \in MU(i)} a_j \\ \forall j \in M \quad c_j/a_j \geq c_p/a_p \quad \forall p \notin M; \quad c_i/a_i = \max \{c_p/a_p / p \notin M\} \end{array} \right.$$

Note:

Cette détermination se fait par un algorithme de complexité linéaire en moyenne. On effectue un relâchement sur la contrainte (1), ce qui se traduit comme suit:

$$- 0 \leq x_i \leq 1 \quad i = 1, \dots, n$$

Ce qui correspond au relâchement de la contrainte d'intégrité (1).

et  $x$  est défini comme suit:

$$x = \begin{cases} x_j = 1 & \forall j \in M & (1) \\ x_i = (b - \sum_{j \in M} a_j) / a_i \\ x_j = 0 & \forall j \in L = \{1, \dots, n\} \setminus \{M \cup \{i\}\} & (2) \end{cases}$$

avec:

- (1) variables de base,
- (2) variables hors base.

La valeur de  $\mu$  est obtenue en prenant  $\mu = c_i/a_i$  (voir [38] pour plus de détails).

### Application numérique

$$(B) \begin{cases} \max & 52x_1 + 17x_2 + 40x_3 + 793x_4 + 813x_5 + 736x_6 \\ \text{s. c.} & 45x_1 + 15x_2 + 40x_3 + 793x_4 + 820x_5 + 777x_6 = 837 \\ & x_j = 0 \text{ ou } 1 \quad j = 1, \dots, 6 \end{cases}$$

$M = \{1,2,3\}$ ,  $L = \{5,6\}$  et  $i = 4$  d'où  $u_i = 1$  et la valeur de la relaxation Lagrangienne est:

$$L(x, \mu) = c.x + \mu.(b - a.x) = \mu.b + (c - \mu.a).x = 846$$

avec  $\{x_j = 1, \forall j \in M; x_j = 0 \forall j \in L\}$  et  $\mu = u_i$ .

La solution optimale est  $cx^* = 805$ .

On remarquera que la solution obtenue est une solution majorante de (B).

### V.3.2 Approche classique

Dans le paragraphe précédent, nous avons montré l'utilité de la relaxation Lagrangienne. Nous proposons l'approche classique de résolution du knapsack.

Il existe une abondante bibliographie sur le thème de la résolution du Knapsack [48, 49, 50]. Les différentes étapes de résolution se présentent comme suit (fig. V.2):

- 1) détermination d'un majorant,
- 2) détermination d'un minorant,
- 3) exploration de l'arborescence associée à l'ensemble des solutions réalisables, par énumération implicite.

#### V.3.2.1 Majorant et minorant initiaux

Le but de ces procédures est de donner un encadrement de la valeur optimale du problème. on retrouvera dans [38] les différentes techniques algorithmique pour l'obtention de ces valeurs.

##### i) Majorant $\bar{v}(B)$ :

La détermination d'une valeur majorante initiale est obtenue en prenant la partie entière de la relaxation Lagrangienne.

##### ii) Minorant $\underline{v}(B)$ :

Nous ne pouvons utiliser les techniques de calcul classiques. En effet, dans notre cas elles peuvent conduire à des solutions non réalisables pour (B). C'est pourquoi, nous prendrons comme valeur de  $\underline{v}(B)$ , une solution réalisable du problème. Dans les deux cas suivant:

a) problème de la gestion de l'avance du tissu:

Une solution réalisable sera telle que l'on ait un chemin en partant d'un sommet source jusqu'au sommet puits. La valeur de ce chemin sera affectée à  $v(B)$ . De plus, il faut vérifier que l'on a tous les sommets une et une seule fois.

b) problème de la génération de trajectoire:

La valeur associée à  $v(B)$  est le coût d'un trajet.

On obtient, après cette première étape de calcul, un encadrement de la valeur optimale de (B):

$$- \underline{v}(B) \leq v^*(B) \leq \bar{v}(B)$$

V.3.2.2 Enumération implicite

Les méthodes que nous regroupons sous ce nom procèdent par parcours dans l'arbre des solutions réalisables en n'envisageant cependant qu'un nombre restreint d'entre elles. Le nom généralement donné à ces méthodes est "Séparation et évaluation Progressive" (S. E. P. ou "Branch and Bound") ou "L. I. F. O." (Last in First Out).

Ci dessous, nous montrons le principe général [38].

Algorithme S. E. P.

Cette méthode de séparation et évaluation progressive est basée sur un algorithme comportant 3 procédures:

- une procédure de séparation ("Branching") qui partitionne l'ensemble des solutions en deux sous-ensembles complémentaires.
- une procédure d'évaluation ("Bounding") de bornes inférieures pour le critère.
- une procédure de progression qui permet de choisir une branche de l'arbre dans laquelle l'exploration va se poursuivre.

Le succès des méthodes S.E.P. dans la résolution du Knapsack est surtout fonction de l'efficacité du calcul des bornes inférieures. En annexe 3, nous donnons le déroulement de l'algorithme lié à la méthode S.E.P. .

On montre en annexe 3 l'arborescence, associée à un ensemble de solutions, qui consiste à mettre à "1" ou à "0" un ensemble de variables. En un noeud de cette arborescence, on se trouve en présence d'un problème auxiliaire (PA). A l'aide de la relaxation Lagrangienne, on détermine un majorant  $\bar{v}(\text{PA})$  de ce nouveau problème. A partir des majorant et minorant initiaux, on décide de la descente ou de la remontée, en pratiquant le test suivant:

- Si  $\underline{v}(\text{B}) \leq \bar{v}(\text{PA}) \leq \bar{v}(\text{B})$  alors descente: mise à "1" d'une nouvelle variable:

sinon remonté: mise à "0" de la variable considérée et mise à "1" d'une nouvelle variable.

En une feuille de l'arborescence on obtient une solution réalisable c.x. On la compare au minorant initial et on tire:

- Si  $c.\bar{x} < \underline{v}(\text{B})$  alors :  $\underline{v}(\text{B}) := c.x.$

L'organigramme classique, associé à la méthode S.E.P., est représenté figure V.2

Parmi toutes les méthodes existantes, nous utiliserons celle proposée par [38], qui reste encore extrêmement performante pour la résolution du problème du Knapsack (voir [50]). En effet, la complexité algorithmique est la plus faible à l'heure actuelle [44].

En annexe 3, nous donnons le déroulement sur un exemple de l'algorithme.

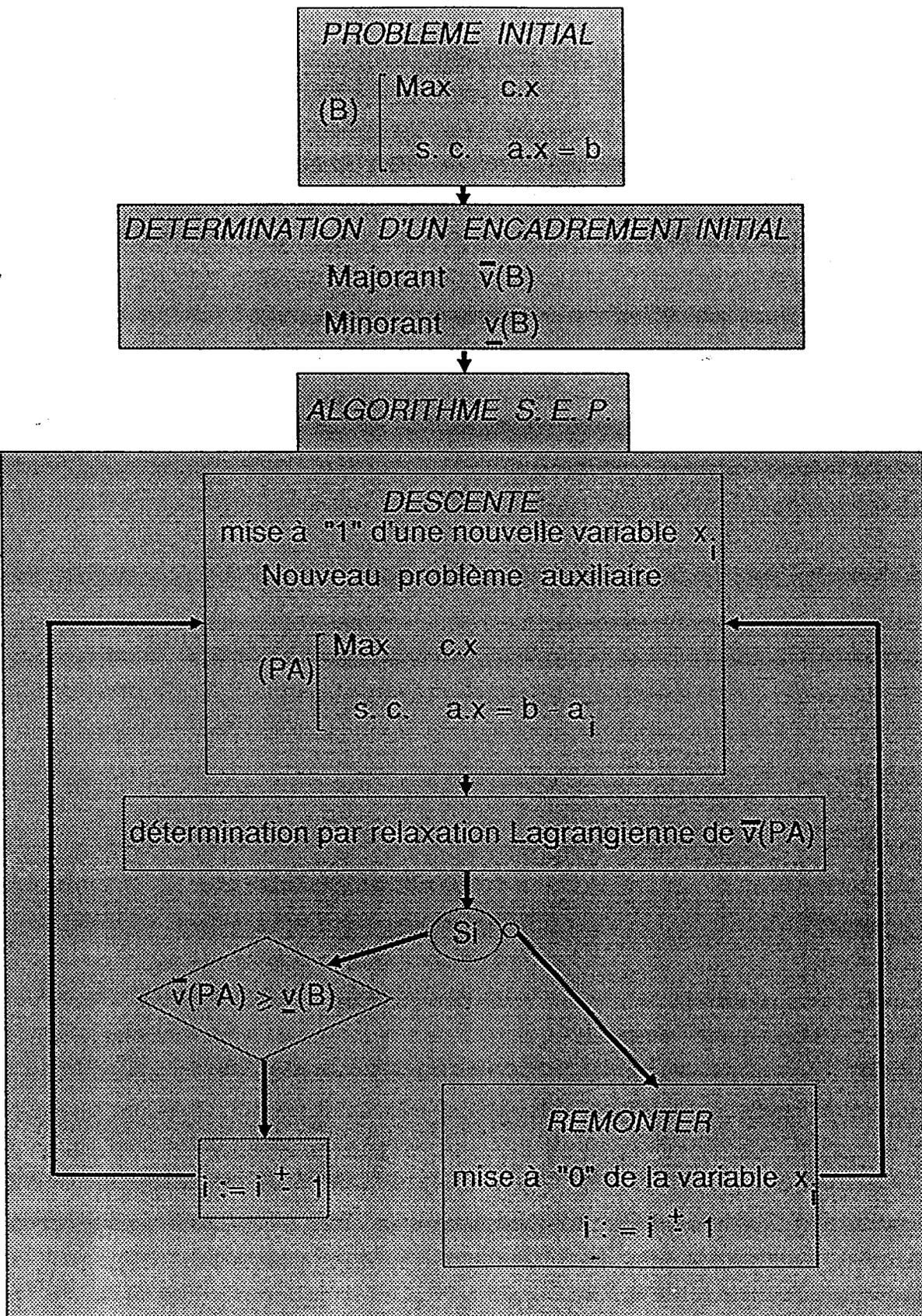


Figure V.2 : Organigramme de la procédure algorithmique

#### V.4 CONCLUSION

Au travers de ce chapitre, nous venons d'aborder dans un cadre général la méthode de modification d'un problème combinatoire multicontraintes, en un problème de type contrainte unique ou problème du Knapsack.

Nous y avons aussi présenté un algorithme de résolution basé sur l'énumération implicite de type P.S.E.P.

La résolution du problème du Knapsack demeure extrêmement complexe d'un point de vue implémentation. Cependant, l'algorithme proposé reste encore très performant et est de complexité temporelle relativement faible.

En annexe 4, nous proposons l'application au problème de génération de trajectoires.

Nous y donnons les principales modifications algorithmiques relativement à la méthode de résolution proposée aux paragraphes précédent.

## INTRODUCTION

Nous avons présenté aux chapitres III et IV la modélisation des problèmes d'optimisation de la gestion du processus d'avance du tissu et de la génération de trajectoires. Au chapitre V nous nous sommes attachés à résoudre ce type de problème.

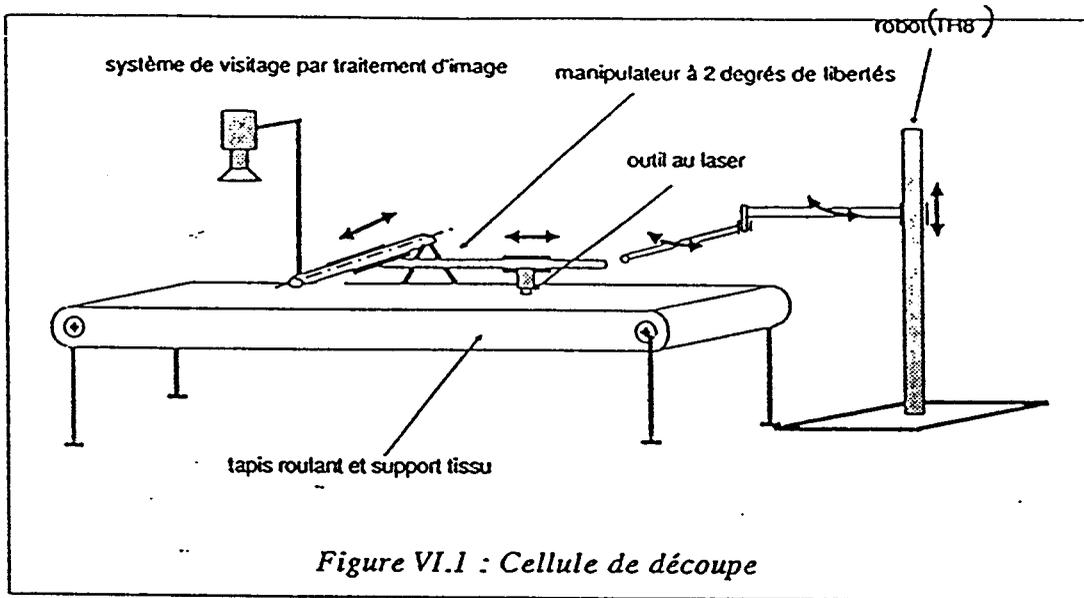
Dans ce chapitre, nous abordons l'aspect réalisation de ce mémoire. Après avoir présenté les différents aspects mécanique, automatique et informatique, nous donnons l'algorithme général de traitement.

### VI.1 PRESENTATION DE LA CELLULE

La cellule est conçue pour réaliser en temps réel les différentes opérations effectuées dans un atelier classique de coupe pour la confection d'un article d'habillement. Elle permet, d'une part, de réduire les temps de fabrication et d'autre part, de répondre en temps réel à toute modification du carnet de commande.

#### VI.1.1 Description

Il s'agit d'une unité de coupe sur feuille unique en mouvement continu. La cellule intègre les trois modules de base d'un atelier classique: le visitage, le placement et la coupe (fig. VI.1).



- le visage est chargé de l'acquisition des caractéristiques du tissu à découper.

Ces données, nécessaires au placement, sont:

- \* les variations de la laize,
- \* les caractéristiques des défauts,
- \* les paramètres géométriques des motifs définissant la structure des raccords.

- le placement intègre les informations issues du visage pour réaliser, en temps réel, l'imbrication des pièces à découper. Il concerne aussi bien les tissus unis que les tissus à motifs.

- la coupe est réalisée par suivi de contours. Le processeur de pilotage utilise les informations de positions et de formes issues du placement des pièces pour élaborer une optimisation:

- \* des trajectoires de l'outil de coupe,
- \* de la gestion de l'avance du tissu.

Notre travail se situe au niveau du troisième module: la coupe.

### VI.1.2 Composition

#### La table de découpe

La structure de la table est représentée par la figure VI.2.

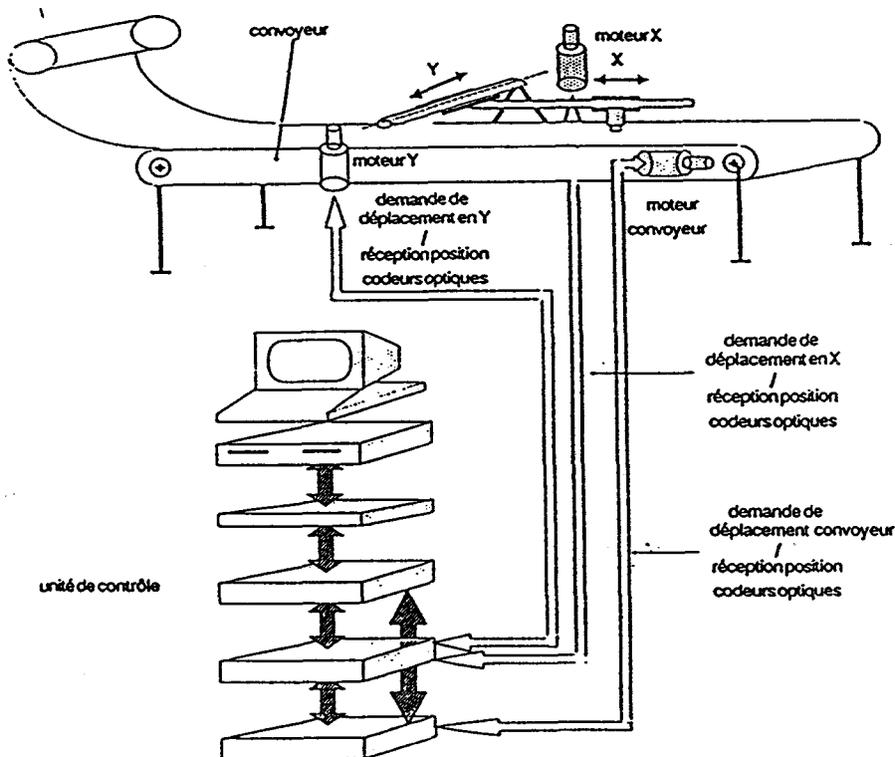


Figure VI.2: Architecture de la table et son unité de contrôle,

La coupe s'effectue par suivi de contour à l'aide d'un faisceau laser. La table de découpe se compose d'un manipulateur à 2 degrés de liberté dont les mouvements sont en X et Y (X suivant le défillement du tissu amplitude 60 cm et Y perpendiculaire au défillement amplitude 180 cm).

Le tissu est déposé sur le convoyeur. Il est ensuite amené sur la surface de travail de l'outil par déplacement du convoyeur. Le support du tissu est une bande de convoyage sous laquelle un système d'aspiration plaque le tissu afin de l'entraîner et de le maintenir.

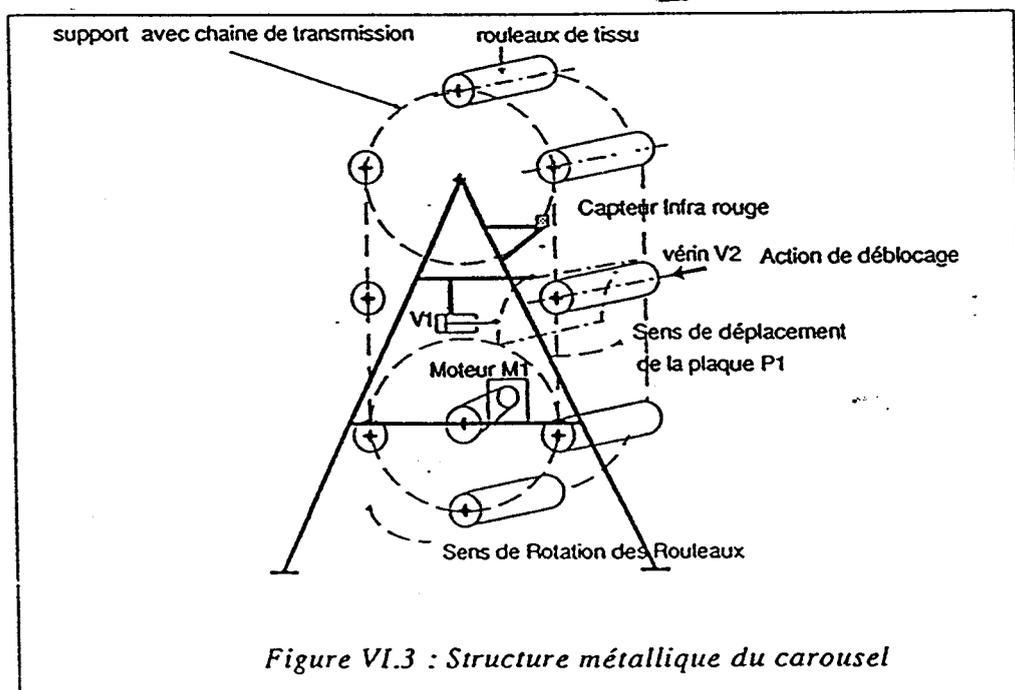
Pour assurer une alimentation continue de la table, nous avons étudié et construit:

- un système d'alimentation constitué d'un carrousel contrôlé par un automate programmable,
- un système de déchargement de la table constitué autour d'un robot.

L'interface homme/machine est réalisé par une unité de contrôle (fig. VI.2).

### Carrousel d'alimentation

La figure VI.3 représente la structure mécanique du système.



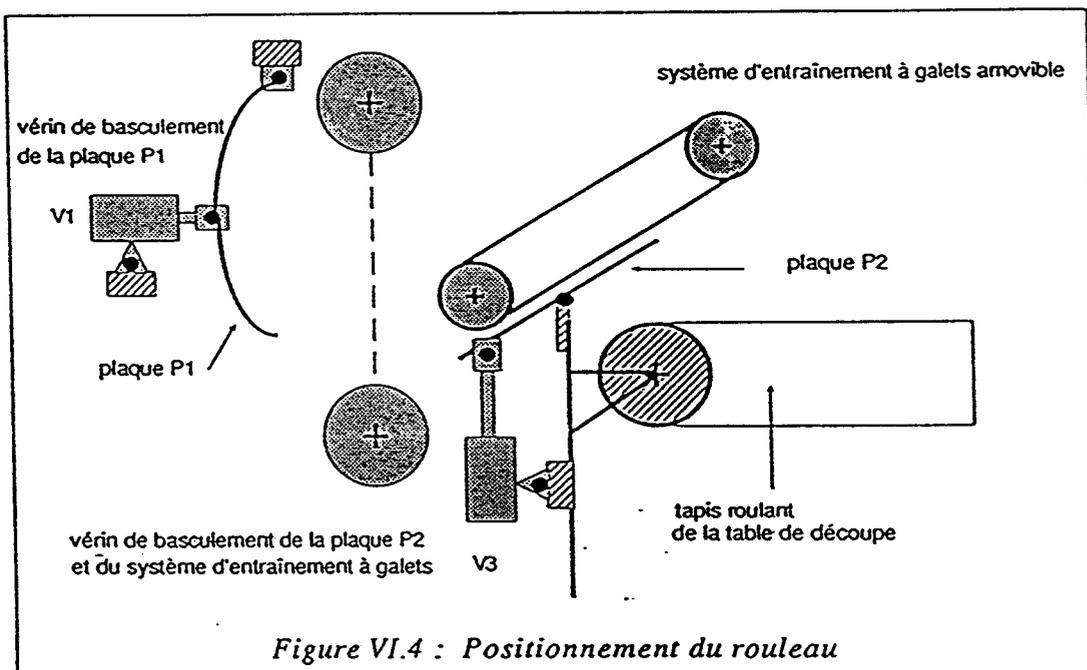
Le carrousel se compose d'une architecture métallique sur laquelle reposent 8 rouleaux de tissu. Le fonctionnement s'effectue comme suit:

- choix d'un rouleau:

les rouleaux sont numérotés et la sélection s'effectue par rotation jusqu'au numéro désiré. Le capteur sert de compteur.

- déroulement du tissu:

Chaque rouleau passe devant la plaque P1. Une fois le rouleau sélectionné, on arrête le défilement, et le rouleau R1 est prêt à être déchargé (Fig. VI.4).



La plaque P1 vient en buté contre la plaque P2. Grâce à la rotation du tapis roulant de la table de découpe on entraîne les galets qui, par frottement, transfèrent le tissu sur la table ( fig. VI.5).

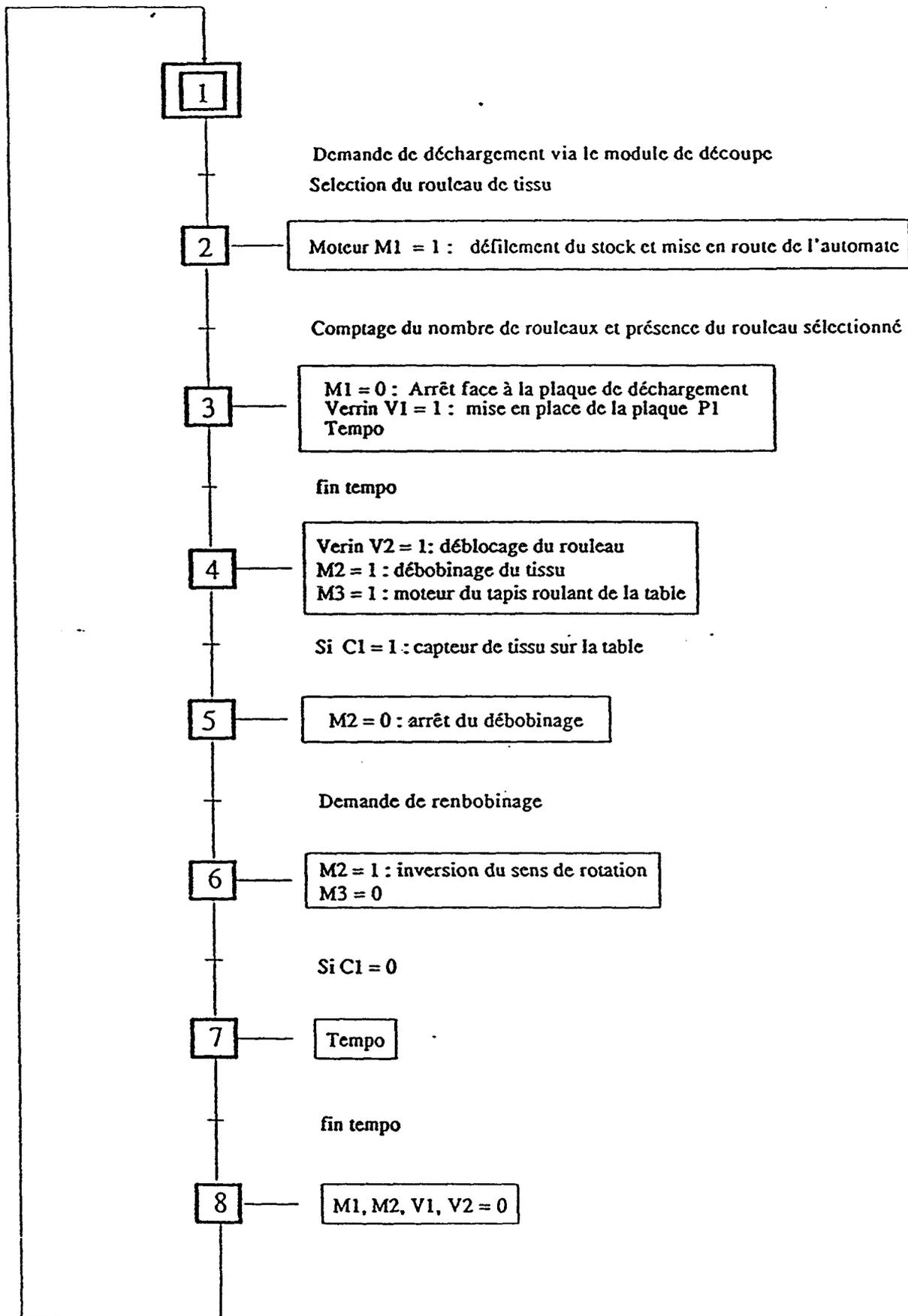


Figure VI.6 : Grafset associé au déchargement du tissu

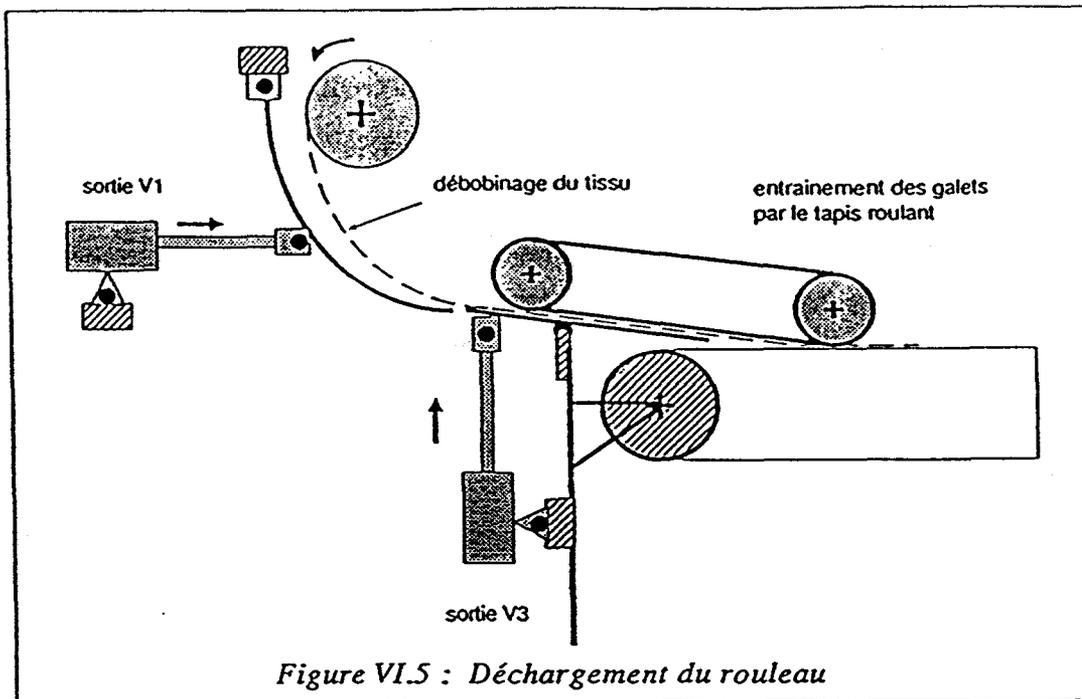


Figure VI.5 : Déchargement du rouleau

### Remarque:

Le rembobinage du tissu se fait par inversion du cycle précédent.

### Commande du carrousel

La commande est obtenue par un automate programmable. Le graficet, associé à cette commande, est représenté figure VI.6. L'ordre de mise en route du cycle de déchargement est transmis par le calculateur de découpe.

### Unité de contrôle de la table de découpe

L'unité de contrôle permet la réalisation de la découpe des pièces de vêtements. Les données sont issues du module de placement sous la forme d'un fichier de pièces à découper. L'unité dispose de la référence et de la position de ces pièces sur chaque plan de placement.

L'unité calcule les trajectoires optimales ainsi que la gestion de l'avance du tissu. Elle communique les données de déplacement, sous la forme de fichiers de points de coordonnées (x,y), à la carte d'axes des moteurs X, Y et convoyeur.

Ensuite, ce dernier module envoie les commandes analogiques aux variateurs (fig. VI.7).

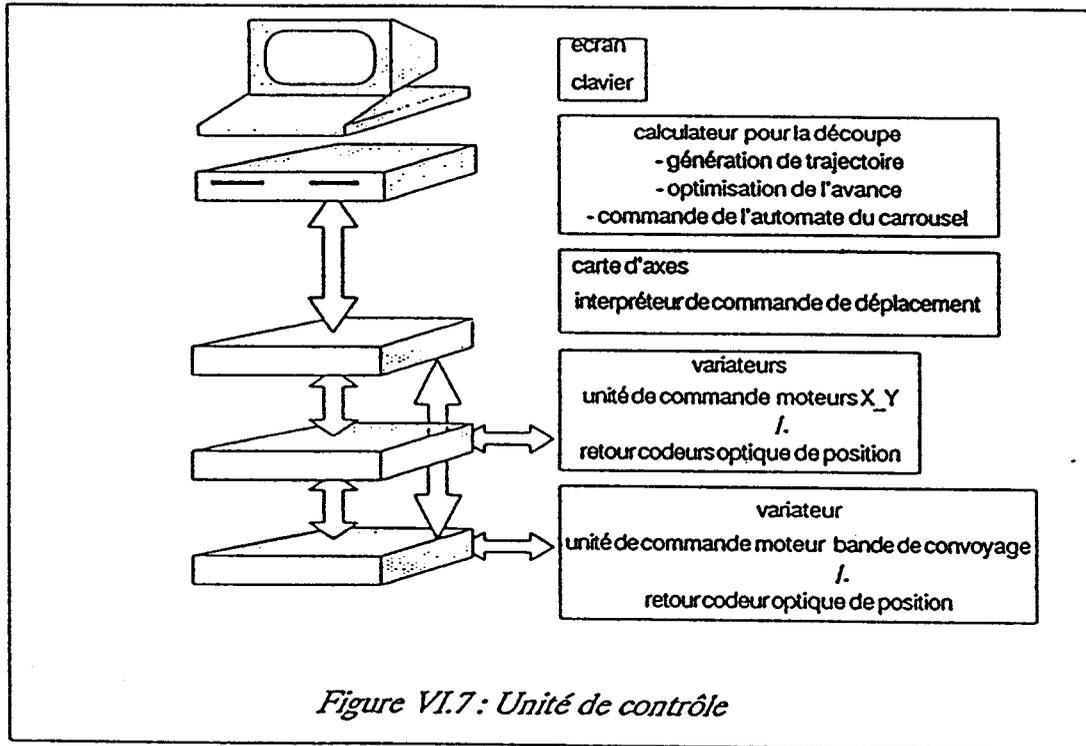


Figure VI.7 : Unité de contrôle

En conclusion, la structure informationnelle globale de la cellule est représentée sur la figure VI.8. C'est l'unité de coupe qui est au centre de notre étude.

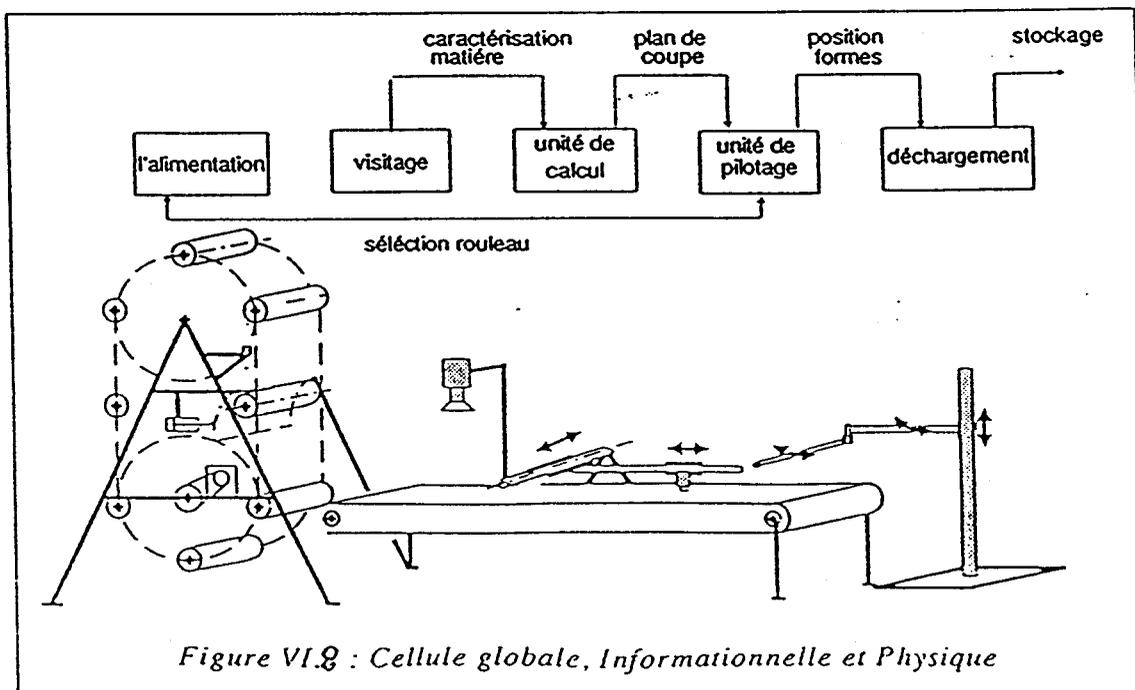


Figure VI.8 : Cellule globale, Informationnelle et Physique

## VI.2 MODULES ET FONCTIONNALITES

Après avoir défini les objectifs de la cellule et sa structure physique, il est possible de décrire d'une part les enchaînements de modules nécessaires à la programmation de la découpe, d'autre part, les fonctionnalités auxquelles peut faire appel chaque module.

### VI.2.1 Modules initiaux de programmation

La figure VI.9 montre l'enchaînement des différents modules. Un module est une unité de programmation cohérente utilisant des données en entrée et fournissant, après traitement, des données en sortie. Un certain nombre de données sont nécessaires à chaque étape:

- géométrie des formes à découper ( référence),
- position sur le plan de placement par rapport au repère absolu.

Chaque module constitue donc un traitement particulier sur les données qui aboutissent à la programmation de la découpe. Nous présentons succinctement les différents modules qui nous conduisent à la programmation découpe.

#### i) module configuration cellule:

Ce module va permettre la constitution de la cellule de découpe. Il est interactif par le mode maintenance. En effet, la cellule est constituée d'un environnement fixe de manipulateurs ( robot, positionneurs) qu'il faut placer dans l'espace. Les éléments de la cellule sont désignés par leur nom et l'opérateur peut choisir parmi ceux-ci.

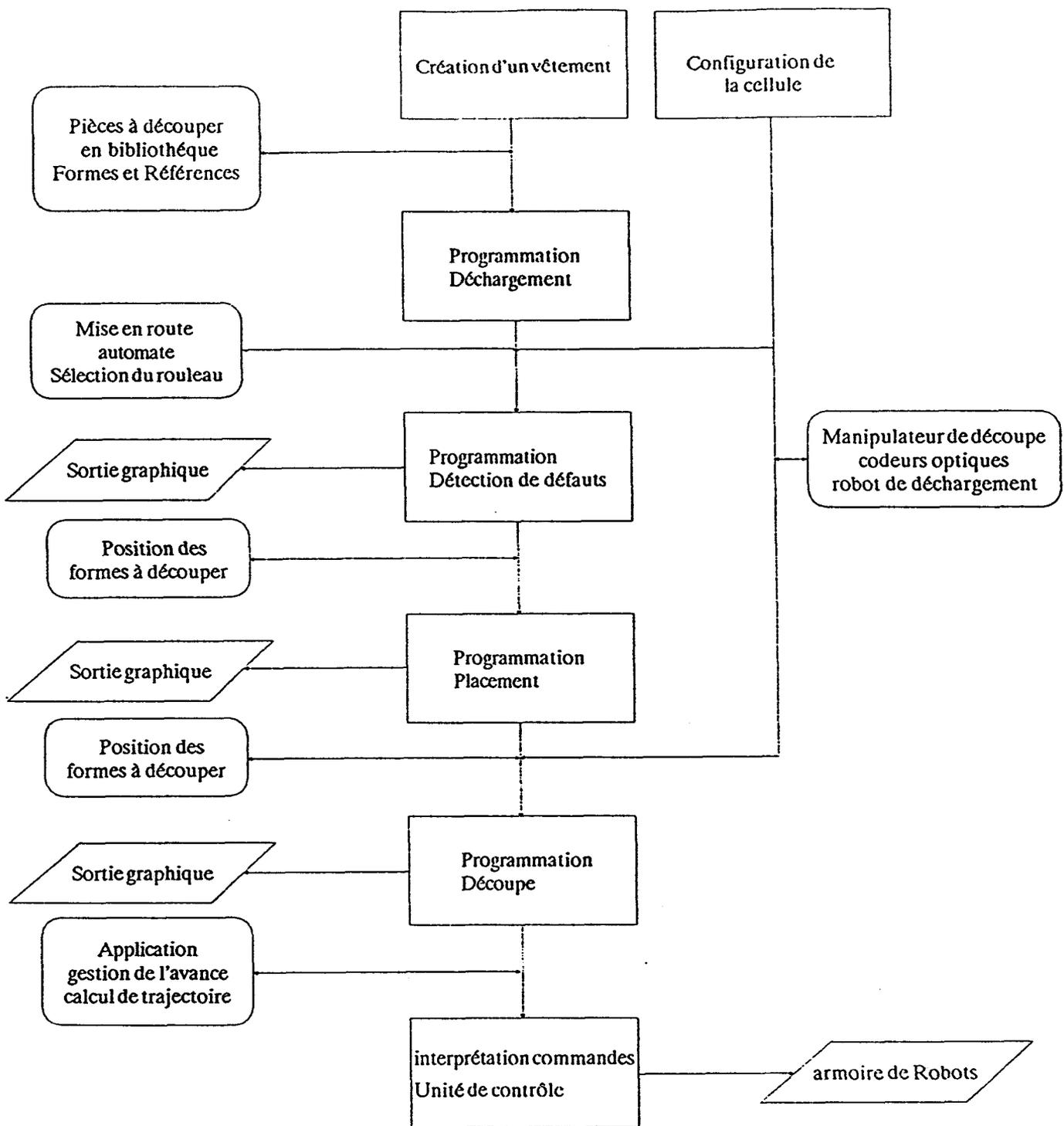


Figure VI.9: Structure générale informationnelle

**ii) création d'une pièce ou carnet de commande:**

Les données de bases sont les formes relatives à un ou des vêtements. La création d'un vêtement peut faire appel à des sous-ensembles déjà existants. Ce module fournit les références des pièces en bibliothèque et le type de tissu.

**iii) programmation du déchargement:**

A partir du carnet de commande, on lance la procédure de déchargement. L'automate met en phase le tissu avec la table et un capteur informe l'unité de contrôle de l'arrivée du tissu. Ce module déclenche, alors, le système de visitage.

**iv) programmation détection de défauts:**

A partir des informations issues d'une caméra ligne, on détecte en continu les défauts sur le tissu. Ce module de traitement délivre leur position.

**v) programmation placement:**

En sortie de ce module, on obtient:

- la référence des formes à découper sur le plan de placement,
- et la position de chaque repère propre.

**VI.2.2 Programmation découpe**

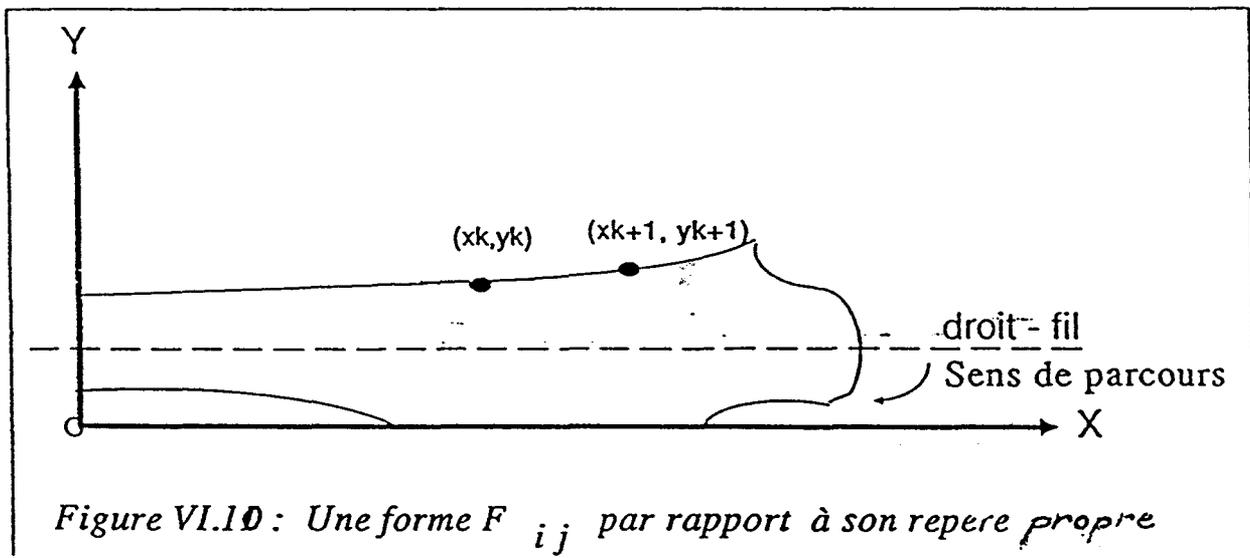
Dans ce paragraphe nous présentons l'aspect informatique qui est l'application des techniques présentées au chapitre III et IV. Le module d'optimisation de la coupe nécessite certaines données. Celle-ci sont issues du module de placement.

### VI.2.2.1 Structuration des données

En mémoire de masse de l'unité de contrôle, on dispose d'une bibliothèque de formes. Nous avons vu que les données issues du module de placement sont les références des formes à découper ainsi que la position de leur repère propre par rapport au repère absolu lié au plan de placement.

#### i) caractéristiques d'une forme:

une forme  $F_{ij}$  est représentée de la manière suivante (fig. VI.10):



- Où:
- $P = \{(x_k, y_k), (x_{k+1}, y_{k+1}), \dots, (x_i, y_i), (x_{k-1}, y_{k-1})\}$  est l'ensemble des points du contour dont les coordonnées sont liées au repère propre  $(xoy)$ ,
  - "o" est l'origine de référence du repère.

## ii) informations issues du module de placement

Le module de placement nous fixe la position des pièces à découper sur le plan. Nous associons à ce plan un repère absolu  $R_a(x_a, o_a, y_a)$ . Ce module envoie un fichier séquentiel dans lequel on retrouve:

- les références des formes à découper,
- la position de l'origine du point "o" par rapport au repère  $R_a$  de chaque forme.

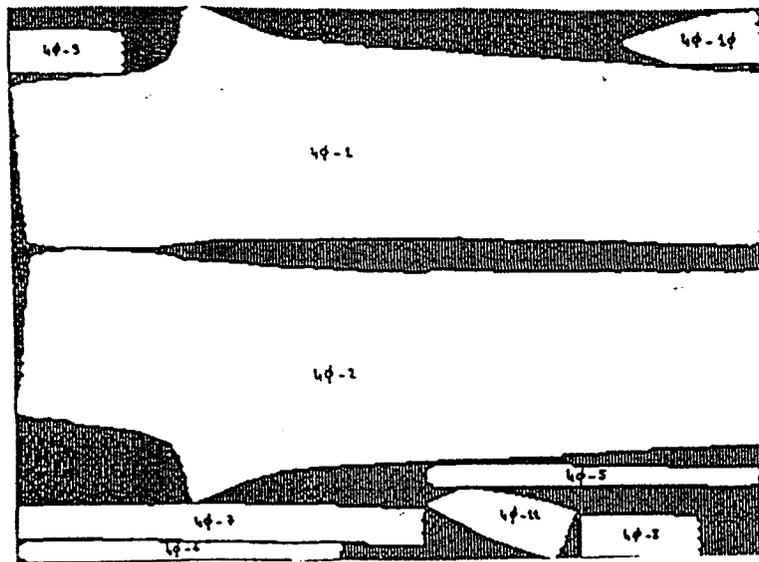


Fig. VI.11

La figure VI.11 représente 9 formes munies de leurs références et coordonnées sur un plan de placement.

Le fichier issu du module de placement est représenté par la figure VI.12. Les différents éléments sont:

- l'ensemble des plans de placements,
- les vêtements issus du carnet de commande,
- pièces ou formes à découper,
- position sur le repère absolu de chaque repère propre associé aux formes.

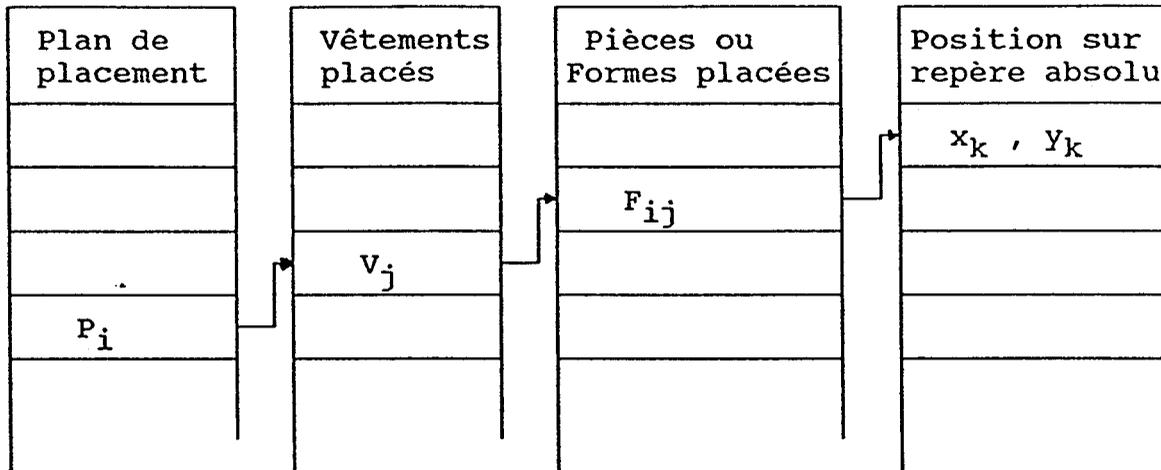


Figure VI.12 : Fichiers issus du placement

VI.2.2.2 Structure informationnelle

On peut désormais présenter la structure de l'enchaînement des différents modules et leur fonctionnalité qui caractérise la programmation découpe ( figure VI.13).

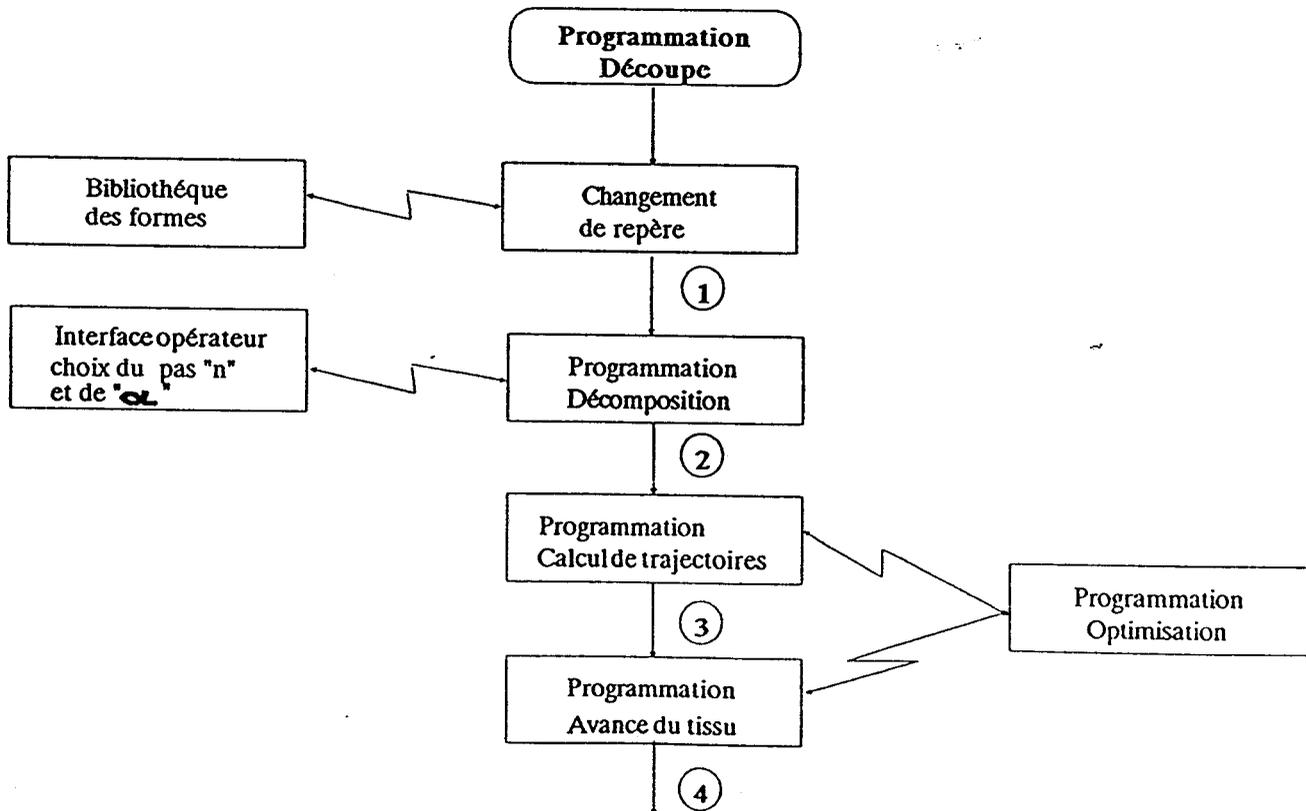


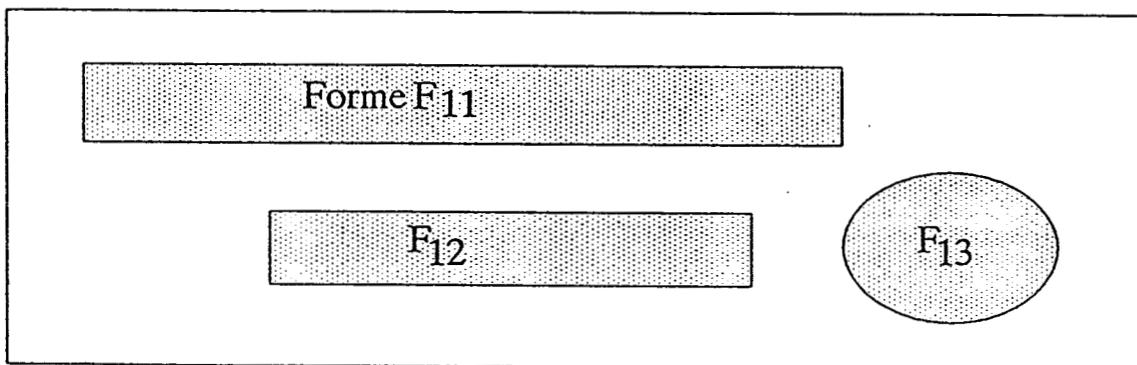
Figure VI.13 : Structure informationnelle de la programmation Découpe

### VI.3 APPLICATION NUMERIQUE

En référence aux différents chapitres et pour plus de clarté, nous expliciterons cette structure au moyen d'un exemple simple. Celui-ci est issu du module de placement dont les données de sortie sont les suivantes ( fig. VI.14):

- un plan de placement P1,
- un ensemble de 3 formes à découper  $F_{11}$ ,  $F_{12}$ ,  $F_{13}$ ,
- position de chaque repère propre par rapport au repère absolu  $R_a$ .

Plan de placement P1



*Figure VI.14: Exemple de plan de placement muni de 3 formes à découper*

On se propose d'optimiser la découpe de P1 par une application des algorithmes proposés au chapitres IV et V. Pour ce faire, les informations et données qui circulent entre chaque module seront décrites dans les paragraphes suivants. Nous reprenons donc la structure informationnelle de la programmation découpe sur cet exemple. A chaque étape correspond un numéros pour lequel un schéma est représenté.

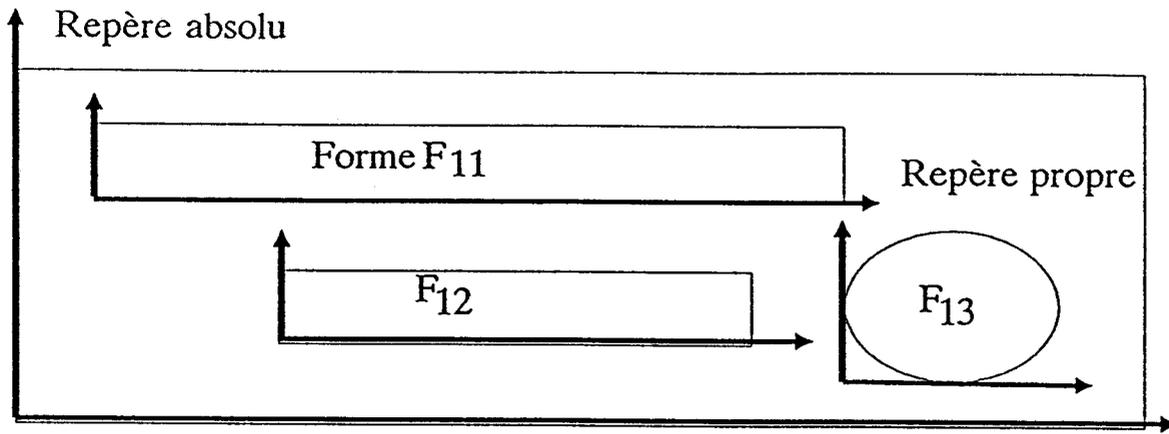


Figure VI.15: Programmation changement de repère

1ère étape

En sortie de programmation changement de repère, on obtient l'ensemble des formes sur le plan de placement ( fig. VI.15).

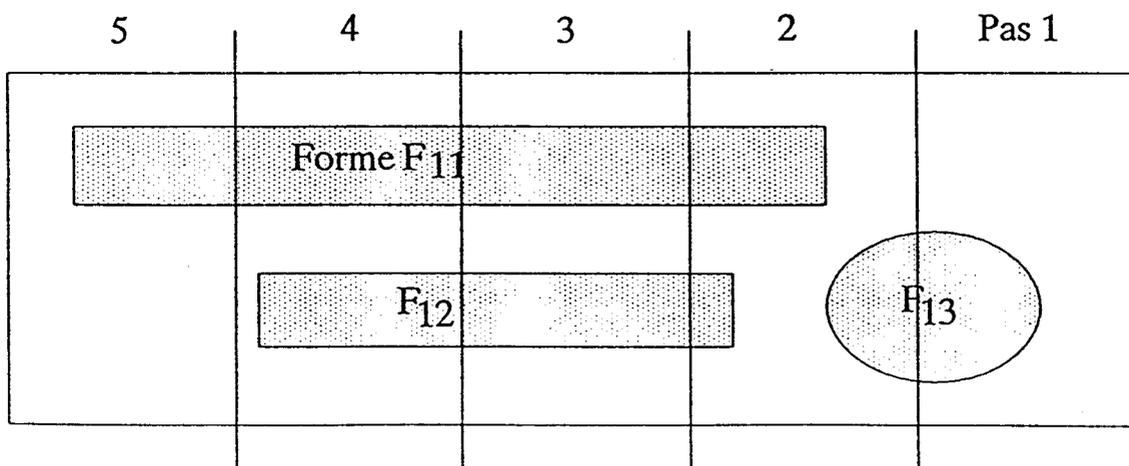


Figure VI.16 : Discrétisation en 5 pas

**2ième étape**

La discrétisation est obtenue à la fin de programmation décomposition ( fig. VI.16). Dans cet exemple  $\alpha = 2$  et  $n = 5$  ( Cf. chapitre III p. 64).

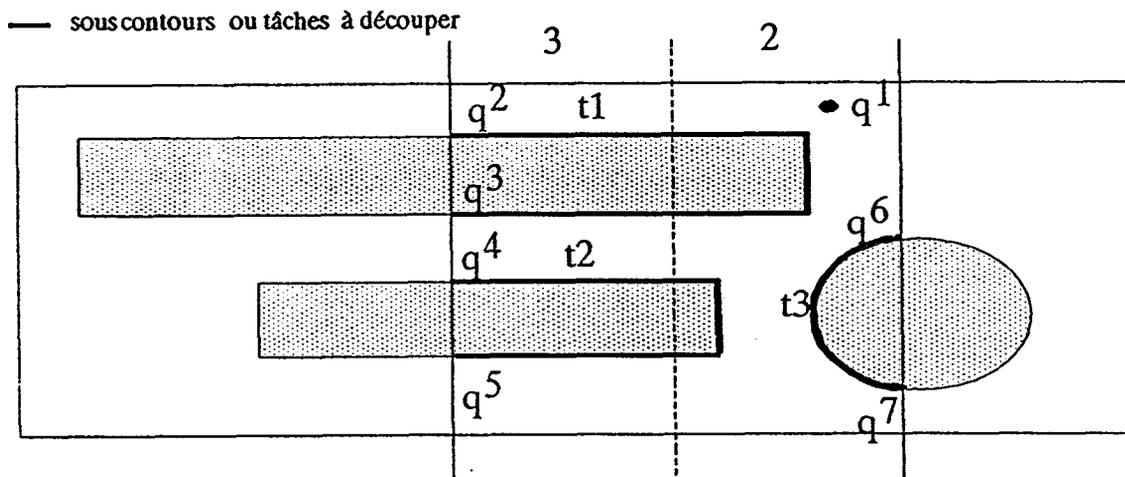


Figure VI.17: Exemple de sous-contours à découper entre les pas 2 et 3

**3ième étape:**

A cette étape, on calcule l'ensemble des distances euclidiennes ( fig. VI.17). Ensuite, on détermine une trajectoire de coût minimum sur chaque pas et combinaison de pas ( fonction de "n" et " $\alpha$ "). Le problème posé est mis sous la forme d'un programme linéaire ( Cf. chapitre IV p. 98). Sur chaque pas ou combinaison de pas se pose le problème de la minimisation du programme linéaire.

Dans l'exemple de la figure VI.17, on désire déterminer une trajectoire de coût minimum afin de découper les trois tâches ou sous-contours  $t_1, t_2, t_3$  présents sur les pas 2 et 3 ( $n_1 = 3$ ).

Le calcul de la matrice de coût symétrique par la programmation Calcul de trajectoire nous conduit à:

C =

$i \backslash j$	1	2	3	4	5	6	7
1	0	18	25	32	15	9	13
2	18	0	11	7	17	25	38
3	25	11	0	52	44	66	39
4	32	7	52	0	10	22	16
5	15	17	44	10	0	33	45
6	9	25	66	22	33	0	26
7	13	38	39	16	45	26	0

Pour laquelle, le coût  $C_{ij}$  correspond à la distance Euclidienne entre les points  $q^i$  et  $q^j$ .  
 Au chapitre IV, nous avons défini les données du problème comme suit:

- nombre de tâches:  $n_1 = 3$ ,
- nombre de variables décisionnelles:  $m = 2.n_1 + 1 = 7$ ,
- nombre de variables réduit sur un trajet:  $h = n_1 + 1 = 4$ .

D'où la formulation associée à ce problème:

$$\text{PL}_{23} \left\{ \begin{array}{l}
 \min \sum_{i=1}^7 \sum_{j>i} C_{ij} x_{ij} \\
 \sum_{j=2}^7 x_{1j} = 2 \quad (1) \\
 \sum_{j>i} x_{ij} = h = m - n_1 = 2*n_1 + 1 - n_1 = 4 \quad (2) \\
 \sum_{j>i} x_{ij} + \sum_{j<i} x_{ij} = 1 \quad \begin{array}{l} i = 2, 4, 6 \\ j = 3, 5, 7 \end{array} \quad (3) \\
 x_{ij} = 0 \text{ ou } 1
 \end{array} \right.$$

d'où le problème :  $\min z = 18x_{12} + 25x_{13} + 32x_{14} + 15x_{15} + 9x_{16} + 13x_{17} + 7x_{24} + 17x_{25} + 25x_{26} + 38x_{27} + 52x_{34} + 44x_{35} + 66x_{36} + 39x_{37} + 22x_{45} + 16x_{47} + 33x_{56} + 45x_{57}$

Sous les contraintes:

$$- (1) = x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} - 2 = 0$$

$$- (2) = x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{24} + x_{25} + x_{26} + x_{27} + x_{34} + x_{35} + x_{36} + x_{37} + x_{46} + x_{47} + x_{56} + x_{57} - 4 = 0$$

$$- (3) = \begin{aligned} &x_{12} + x_{24} + x_{25} + x_{26} + x_{27} - 1 = 0, \\ &x_{13} + x_{34} + x_{35} + x_{36} + x_{37} - 1 = 0, \\ &x_{14} + x_{24} + x_{34} + x_{46} + x_{47} - 1 = 0, \\ &x_{15} + x_{25} + x_{35} + x_{56} + x_{57} - 1 = 0, \\ &x_{16} + x_{26} + x_{36} + x_{46} + x_{56} - 1 = 0, \\ &x_{17} + x_{27} + x_{37} + x_{47} + x_{57} - 1 = 0, \end{aligned}$$

L'utilisation de la programmation Optimisation nous permet d'obtenir une séquence qui correspond au trajet 1-6-7-4-5-2-3-1 qui, par symétrie, est le trajet: 1-3-2-5-4-7-6-1 pour lequel les valeurs des coûts sur les tâches ne sont pas comptabilisées. La solution de ce programme linéaire est  $\min z = 67$  et correspond au trajet de la figure VI.18.

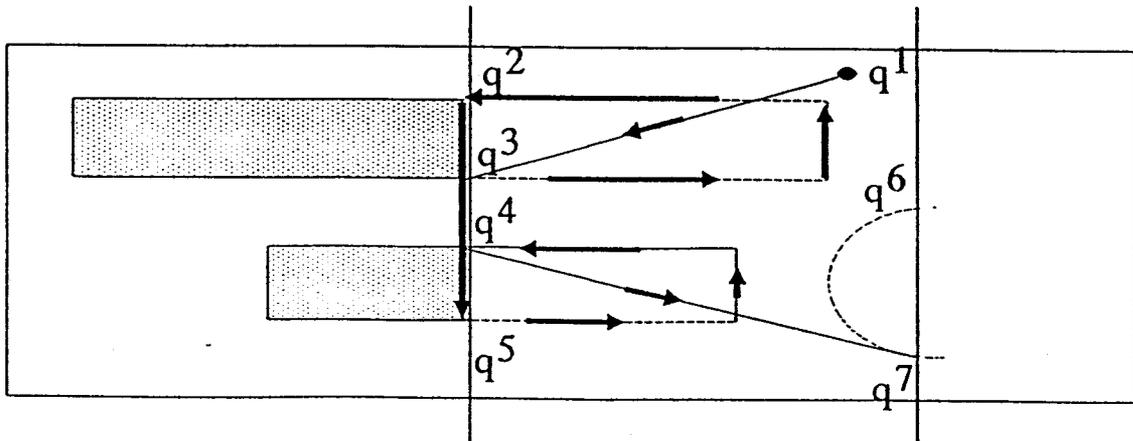


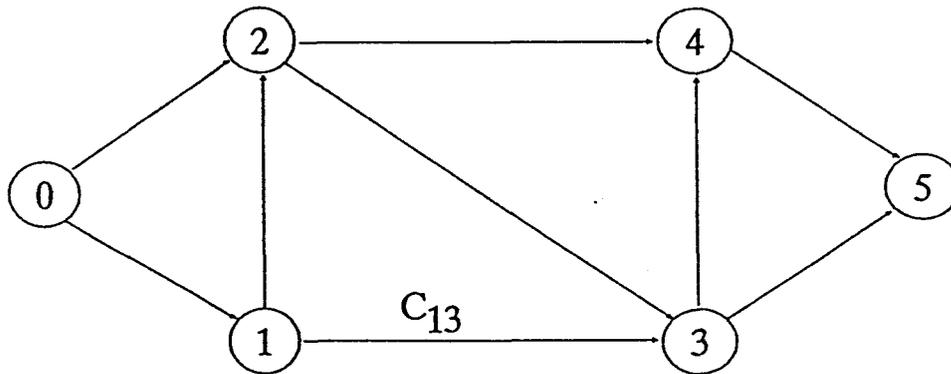
Figure VI.18 : Trajectoire optimale

Remarque:

En annexe 4 nous donnons le déroulement exacte de l'algorithme pour lequel la méthode P.S.E.P. est détaillée sur un exemple en annexe 3.

La procédure est répétée, pour chaque pas et chaque combinaison de pas. Ceci nous conduit à un ensemble de solutions relativement à chaque programme linéaire (Solution  $PL_1, PL_2, PL_3, \dots, PL_5, PL_{12}, PL_{23}, \dots, PL_{45}$ )

La mise sous forme d'un réseau de l'ensemble des solutions des problèmes  $PL_{ij}$  nous conduit à l'étape suivante ( Cf. chapitre III p. 76)



*Figure VI.19: Réseau associé à P1*

4ième étape:

A la fin de la troisième étape on obtient le réseau de la figure VI.19. A chaque arc  $u = (l,k)$  correspond le coût  $C_{lk}$  de passage d'un noeud "l" à un noeud "k" ( Cf. chapitre III p. 75).

De l'exemple, on tire le réseau de la figure VI.19 sur lequel on cherche à déterminer un chemin de coût minimum entre le noeud "0" et le noeud "5". On ramène se problème à un programme linéaire ( Cf. chapitre III P. 78).

La programmation Avance du tissu nous conduit au programme linéaire suivant:

$$\begin{array}{l}
 \text{minimiser} \quad z = \sum_{(i,j) \in \text{réseau}} c_{ij} x_{ij} \\
 \text{contraintes} \quad \sum_{j=1}^2 x_{0j} = 1 \quad (1) \\
 \sum_{j=3}^4 x_{j5} = 1 \quad (2) \\
 \sum_{i < k} x_{ik} - \sum_{j > k} x_{kj} = 0 \quad \text{pour } k = 1, 2, \dots, 4 \quad (3) \\
 \text{et } i, j \in \text{réseau} \\
 \text{et } u = (1, k) \in \text{réseau} \\
 x_{ij} = \{0, 1\} \quad \text{pour } i = 0, \dots, 5 \quad (4) \\
 \text{pour } j = 0, \dots, 5 \\
 c_{ij} \in \mathbb{R}^+
 \end{array}$$

PLNE

Ce qui correspond au développement suivant:

$$\text{minimiser } z = c_{01}x_{01} + c_{02}x_{02} + c_{12}x_{12} + c_{13}x_{13} + c_{23}x_{23} + c_{24}x_{24} + c_{34}x_{34} + c_{35}x_{35} + c_{45}x_{45}$$

sous les contraintes :

$$- x_{01} + x_{02} = 1 \quad (1)$$

$$- x_{35} + x_{45} = 1 \quad (2)$$

$$- x_{01} - (x_{12} + x_{13}) = 0 \quad (3)$$

$$- x_{02} + x_{12} - (x_{23} + x_{24}) = 0 \quad (3)$$

$$- x_{13} + x_{23} - (x_{34} + x_{35}) = 0 \quad (3)$$

$$- x_{24} + x_{34} - x_{45} = 0 \quad (3)$$

La matrice relative au réseau de la figure VI.19 est:

C =

	0	1	2	3	4	5
0	$\infty$	17	25	$\infty$	$\infty$	$\infty$
1	$\infty$	$\infty$	20	67	$\infty$	$\infty$
2	$\infty$	$\infty$	$\infty$	22	70	$\infty$
3	$\infty$	$\infty$	$\infty$	$\infty$	50	55
4	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	73

Pour lequel:

- $C_{lk} = \infty$  pour  $l = k$  et pour tout arc n'appartenant pas au réseau.

L'utilisation de la programmation Optimisation nous permet d'obtenir une séquence optimale qui correspond au chemin 0 2 3 5 ( fig. VI.20).

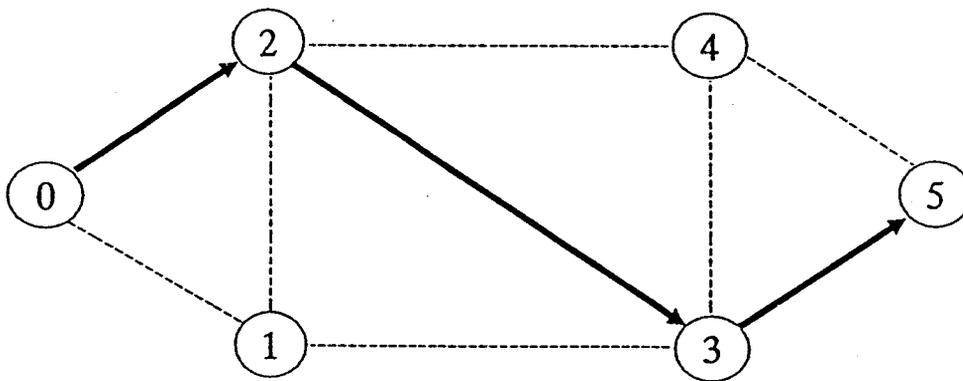
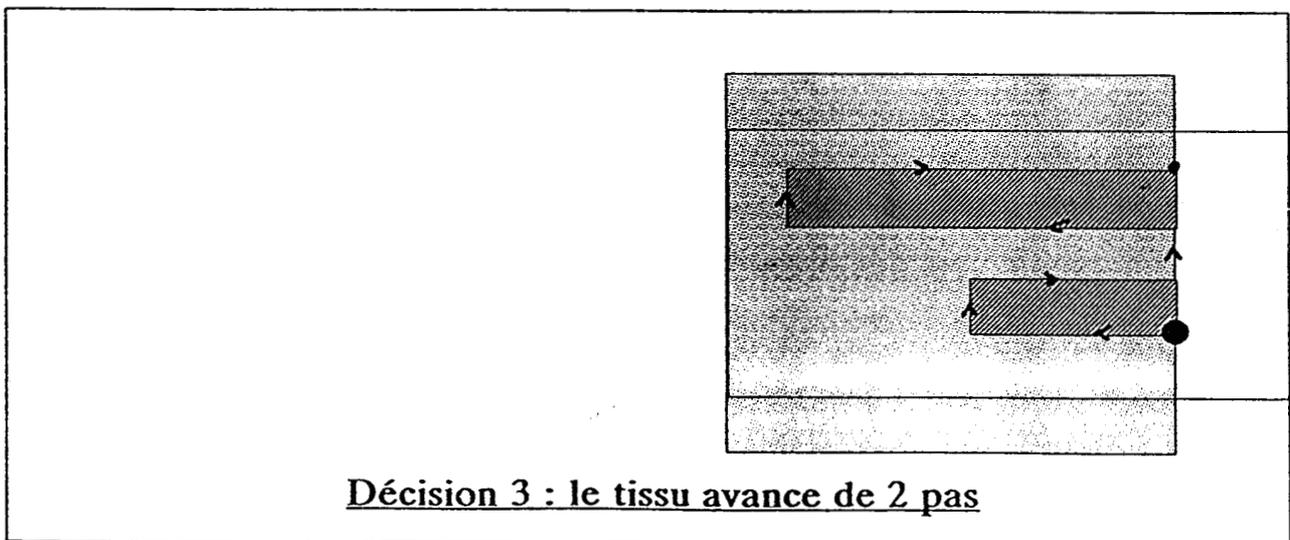
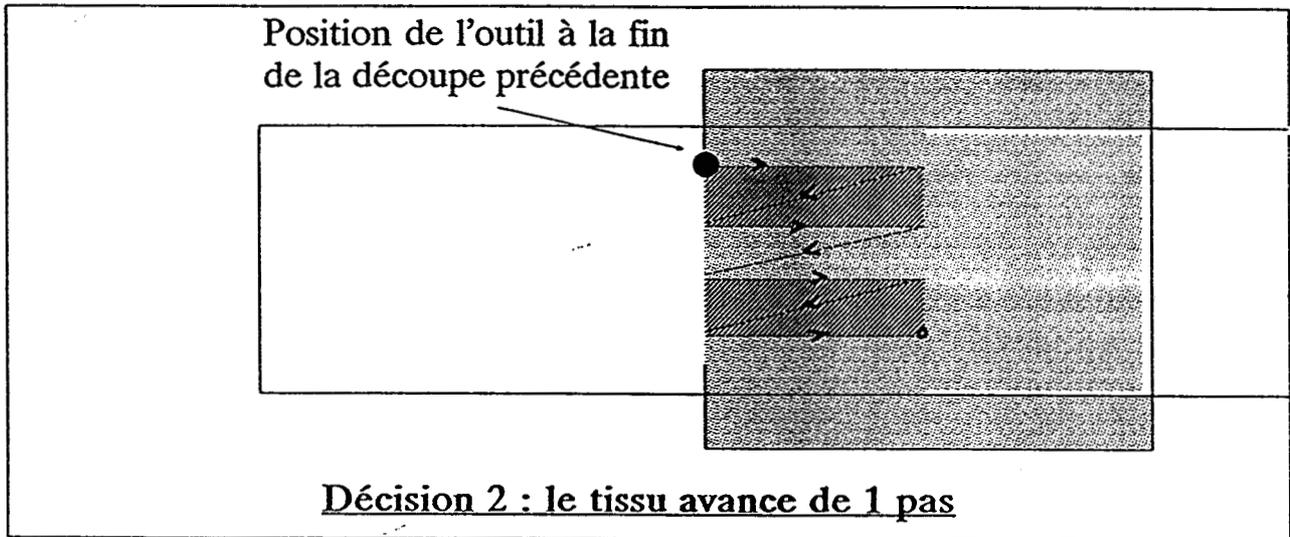
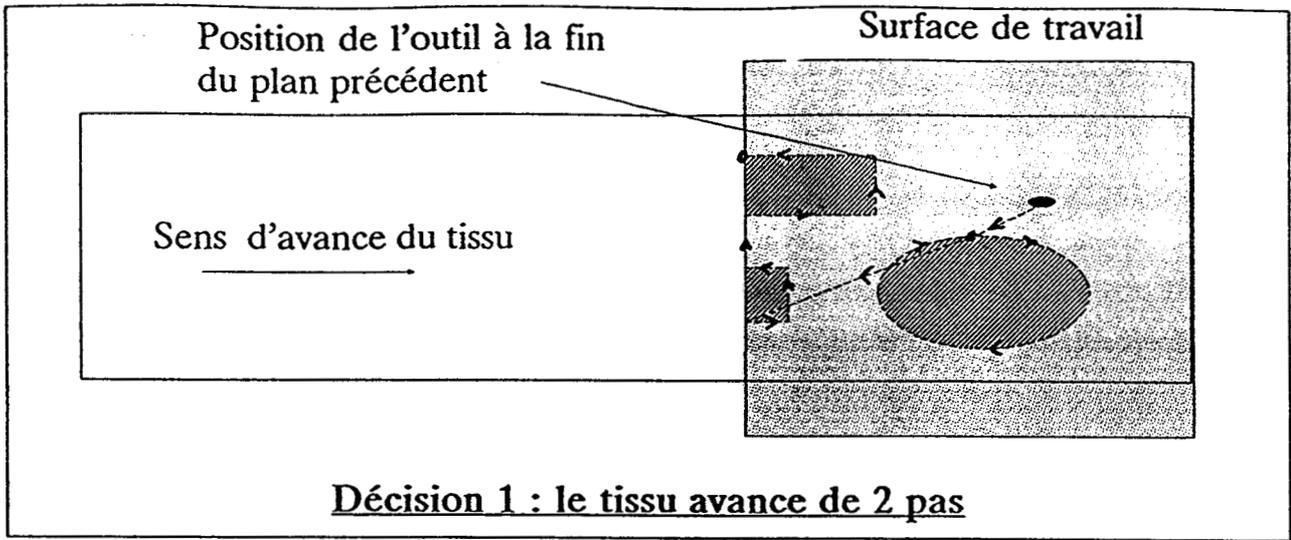


Figure VI.20 : Chemin optimale pour l'avance du tissu

Cette séquence consiste à donner au tissu une avance de 2 pas puis de 1 pas puis de 2 pas. Sur la figure VI.21 nous avons présenté les différentes étapes de l'évolution du tissu sous la surface de travail ainsi que les différentes trajectoires optimales de l'outil.



*Figure VI.21 : Evolution de l'avance du tissu*

## VI.4 CONCLUSION

L'étude de cette exemple montre l'intérêt que représente l'optimisation de la génération de trajectoires ainsi que celle de l'avance du tissu.

En effet, la solution évidente qui consiste à prendre, comme avance, la succession de pas suivante: 0 2 4 5 nous conduit à une valeur non optimale 168 qui se classe en 3<sup>ième</sup> position parmi les solutions. Cette solution est prise sans optimisation de l'avance c'est à dire que l'on décide d'avancer de 2 en 2 pas.

Ce travail de thèse concerne l'étude du problème de conduite d'une cellule de production flexible utilisant une table de découpe au laser. Il s'agit de produire des pièces de vêtements selon un carnet de commande. L'objectif est d'améliorer la conduite de ce processus de découpe.

Notre contribution se situe au niveau de l'automatisation de la cellule par le déchargement et le chargement du tissu sur la table de découpe, puis de la description mathématique par la mise au point de deux modèles permettant l'optimisation du processus.

Dans une première partie, on présente la structure informationnelle d'une entreprise de confection par un modèle S.A.D.T. . Ensuite, on rappelle le problème de la découpe mono et bi-dimensionnelle.

Après la présentation des caractéristiques principales de notre cellule, une modélisation par la programmation dynamique puis par un programme linéaire est proposée pour le problème de la gestion de l'avance du tissu. Une seconde modélisation par un programme linéaire est proposée pour le problème de génération de trajectoire.

Un programme de simulation a été écrit en PASCAL286 sous iRMX286. Le problème posé se caractérise particulièrement par la prise en compte d'une contrainte technologique. L'objectif est la minimisation du temps de production de chaque plan de placement.

Pour résoudre ces deux problèmes, nous avons employé les techniques de la Recherche Opérationnelle. Nous avons formulé nos problèmes comme des programmes linéaires en variables bivalentes. Nous les avons ramenés à des problèmes du knapsack en égalité par contraction du système de contraintes initial.

Les résultats que nous avons obtenus sur quelques exemples sont encourageants, puisque dans la plupart des cas on arrive à obtenir une solution proche de l'optimum.

Ce travail de thèse concerne l'étude du problème de conduite d'une cellule de production flexible utilisant une table de découpe au laser. Il s'agit de produire des pièces de vêtements selon un carnet de commande. L'objectif est d'améliorer la conduite de ce processus de découpe.

Notre contribution se situe au niveau de l'automatisation de la cellule par le déchargement et le chargement du tissu sur la table de découpe, puis de la description mathématique par la mise au point de deux modèles permettant l'optimisation du processus.

Dans une première partie, on présente la structure informationnelle d'une entreprise de confection par un modèle S.A.D.T. . Ensuite, on rappelle le problème de la découpe mono et bi-dimensionnelle.

Après la présentation des caractéristiques principales de notre cellule, une modélisation par la programmation dynamique puis par un programme linéaire est proposée pour le problème de la gestion de l'avance du tissu. Une seconde modélisation par un programme linéaire est proposée pour le problème de génération de trajectoire.

Un programme de simulation a été écrit en PASCAL286 sous iRMX286. Le problème posé se caractérise particulièrement par la prise en compte d'une contrainte technologique. L'objectif est la minimisation du temps de production de chaque plan de placement.

Pour résoudre ces deux problèmes, nous avons employé les techniques de la Recherche Opérationnelle. Nous avons formulé nos problèmes comme des programmes linéaires en variables bivalentes. Nous les avons ramenés à des problèmes du knapsack en égalité par contraction du système de contraintes initial.

Les résultats que nous avons obtenus sur quelques exemples sont encourageants, puisque dans la plupart des cas on arrive à obtenir une solution proche de l'optimum.

- [1] **S.A.D.T. : "Un langage pour communiquer", I. G. L. Technology, Collection Eyrolles, 1989.**
- [2] **Ros D. T. : "Structured analysis (S. A.): a language for communicating ideas", I. E. E. E. transactions on software engineering n°1, p. 16-34, 1977.**
- [3] **Borne F. : " Organisation des entreprises", Edition Fouvher, 1971.**
- [4] **Delaporte J. L. : "Intégration des fonctions de conception et de préparation de la fabrication pour les entreprises de découpe", Thèse de Doctorat d'Automatique, Université de VALENCIENNES et du HAINAUT CAMBRESIS, 1989.**
- [5] **Gibaud A. : "Aide à l'établissement des gammes de confection de vêtements et analyse des coûts de production", Rapport D. G. R. S. T., 1982**
- [6] **Maranzana R. : "Intégration des fonctions de conception et de fabrication autour d'une base de données relationnelle", Thèse de Doctorat de l'Université de VALENCIENNES et du HAINAUT CAMBRESIS, 1988.**
- [7] **Soenen R., Vallé B. : "Méthodologie de conception d'une base de données en gestion de production", 3ième journées Scientifique et Techniques de la Productique Automatisée, A. D. E. TOULOUSE, 3/5 juin 1981.**
- [8] **Lauriol A. : "L'organisation du travail et des fabrications dans les ateliers de coupes", Edition Vauclair, Modes et Techniques, 1985.**

- [9] **Soenen R.** : "Contribution à l'étude des systèmes de conduite en temps réel en vue de la commande d'une unité de fabrication", Thèse de Doctorat d'état, Université de VALENCIENNES et du HAINAUT CAMBRESIS, 1977.
- [10] **Costa M. C.** : "Formalisation et résolution des problèmes de découpe linéaire", R. A. I. R. O., n°1, février 1982, p. 65-82.
- [11] **Costa M. C.** : "Problème de découpe linéaire; formalisations et solutions économiques", Thèse de 3ième cycle, PARIS VI, 1980.
- [12] **Litton C. D.** : "A frequency approach to the one dimensional cutting problem", Oper. Res. Quart., vol. 28, n°4, 1977, p. 927-938.
- [13] **Albano A.** : "Optimal allocation of two dimensional irregular shapes using heuristic search methods", I. E. E. E. transactions on systems, Man and Cybernetics, vol. SMC-10, n°5, may 1980, p. 242-248.
- [14] **Laurent D. G.** : "A heuristic algorithm for optimal placement of rectangular object 's", Information Sciences, n°26, 1982, p. 127-139.
- [15] **Gilmore P. C.** : "Cutting stock, linear programming, knapsing, dynamic programming, some interconnections", Annals of discrete Math., n°4, tome 1, ED. Hammer, North Holland, 1979, p. 217-235.
- [16] **Gilmore P. C, Gomory R.E.** : "A linear approach to the cutting stock problem", Oper. Resear., 1961, vol. 9, p. 849-859.

- [17] Gilmore P. C, Gomory R.E. : "Multi stage cutting stock problems of two and more dimensions", Naval Research, 1964, p. 94-120.
- [18] Bezier P. : "Essai et définition numérique des courbes et des surfaces expérimentales", Thèse d'état, PARIS VI, 1977.
- [19] Erschler J., Fontan G., Merce C., Roubellat F. : "Applying new dominance concepts to job schedule optimization's", European Journal of Operational Research, vol. 11, 1982, p. 60-66.
- [20] Boctor F. F. : "Single machine lot scheduling; a comparison of some solution procedures", A. P. I. I. , 1985, p. 389-402.
- [21] Erschler J., Fontan G., Merce C. : "Un nouveau concept de dominance pour l'ordonnancement de travaux sur 1 machine", R. A. I. R. O., Oper. Resear., vol. 19, n°1, février 1985, p. 15-26.
- [22] Sedmann A. : "Common due date assignment to minimize total penalty for the one machine scheduling problem", Oper. Resear., 1982, p. 391-399.
- [23] Bianco L., Ricciardelli S. : "Scheduling of a single machine to minimize total weighted completion time subject to release dates", Naval Research Logistic Quaterly, vol. 29, n°1, March 1982.
- [24] Carlier J., Chretienne P. : "Problèmes d'ordonnancement; modélisation, complexité, algorithmes", Edition Masson, 1988.

- [25] Laurière J. L. : "Eléments de programmation dynamique", Edition Bordas, 1989.
- [26] Minoux : "Programmation mathématique", Edition Dunod, 1983, PARIS.
- [27] Chevalier A. : "La programmation dynamique", Edition Dunod décision, 1977.
- [28] Sakarovitch M. : "Optimisation combinatoire; méthodes mathématiques d'algorithmiques, programmation discrète", Edition Herman, 1985.
- [29] Hamdy A., Taha : "Operation research", Edition Mac Millan Publishing CO., INC, New-York, 1982.
- [30] Lopez P. : "Introduction à la robotique", Editest, tome 2, 1984.
- [31] Khalil W., Liégeois A. : "Génération de mouvements optimaux en robotique", R. A. I. R. O., vol. 18, n°1, 1984, p. 25-39.
- [32] Lawler L., Lenstra K. : "The travelling salesman problem", Wiley-Interscience Publication, 1985.
- [33] Cameri A., Maffioli F. : "Polynomial bounding for NP-Hard problems", Mathematical programming study, vol. 12, 1980, p. 115-119.
- [34] Grötschel M. : "On the symmetric travelling salesman problem", Mathematical programming study, vol. 12, 1980, p. 61-77.

- [35] **Roseaux** "Recherche opérationnelle", Edition Masson, tome 3, 1985.
- [36] **Bland R. G.** : "New finite pivoting rules for the simplexe method", *Math. of Oper. Res.*, 2, 1977, p. 103-107.
- [37] **Ribeiro C.** : "Algorithmes de recherche de plus courts chemins avec contraintes: étude théorique, implémentation et parallélisation", Thèse Doctorat de Mathématiques appliquées, PARIS VI, 1983.
- [38] **Fayard D., Plateau G.** : "Contribution à la résolution des programmes mathématiques en variables entières", Thèse d'Etat, Université des Sciences et Techniques de LILLE, 1979.
- [39] **Plateau G.** : "Résolution d'un problème d'affectation", E. D. F., série C, *Mathématique et Informatique*, 1973.
- [40] **Padberg M. W.** : "Equivalent knapsack type formulations bounded integer linear programs", *Management sciences research*, report n°227, 1970.
- [41] **Bradley G. H.** : "transformation of integer programs to knapsack problems", *Discrete mathematics*, vol n°1, n°1, 1971, p. 29-45.
- [42] **Glover F., Woolsey R. E.** : "Aggregation Diophantine equations", *Oper. Resear.*, 1972, p. 1-10.

- [43] Glover F. : "New results for reducing integer linear programs to knapsack problems", Management sciences, report n° 72-7, April 1972.
- [44] Guerch M. T. : "La contraction d'équations diophantines", Thèse 3<sup>ième</sup> cycle Mathématiques appliquées, Université des Sciences et Techniques de LILLE, 1983.
- [45] Desrosière J, Soumis F., Sauve M. : "Lagrangean methods for solving the minimal M-T.S.P. with time windows", Management sciences 34, 1005-1022, 1988.
- [46] Michlon P. : "Méthodes duals pour problèmes d'optimisations avec variables entières", Thèse de Doctorat Mathématiques appliquées, Université Blaise Pascal CLERMONT-FERRAND, 1989.
- [47] Jörnsten K, Nöserg M. : "A new lagreangean relaxation approach to the generalyzed assignment problem", E.J.O.R. 27, p. 313-323, 1986.
- [48] Yao C. : "On parallele computation for the knapsack problem", J.A.C.M. 29, 1982, p. 898-903.
- [49] Fréville A. : "Heuristique et réduction pour les problèmes mathématiques en variables 0-1 à contraintes d'inégalité", Thèse de 3<sup>ième</sup> cycle Mathématiques appliquées, Université des Sciences et Techniques de LILLE, 1983.
- [50] Plateau G., Roucairol C., Gachet S. : "Résolution du problème de multi-knapsack en parallèle: algorithme PR<sup>2</sup> 88", I.N.R.I.A., rapport de recherche n°961, programme 2, 1989.

- [51] Okat A., Maouche S., Staroswiecki M.: "Optimum trajectory for a laser cutting machine in the garment industry", International Conference on Industrial Robotic. Greendelwald, SUISSE, Fév. 20-27, 1987.
- [52] Okat A., Maouche S., Staroswiecki M.: " Optimum tasks and path generation for a Laser Cutting Machine", International Symposium Industrial, Vehicular and Space Applications of Microcomputeurs I.S.M.M. . New York, ETATS UNIS, Oct. 9-11, 1990.
- [53] Okat A., Maouche S.: "Optimal tasks for Nuclear Robots", International Conference Advanced Robot Technology and Research Association I.S.A.R.T. 91. Tokyo, JAPON, Nov. 10-11, 1991.
- [54] Okat A., Maouche S., Staroswiecki M.: "La gestion de production dans les ateliers flexibles: étude d'une cellule flexible de découpe au laser". Congrès de Génie Industriel, Ecole d'Ingénieurs de Tours, Mars 1991.
- [55] Okat A., Staroswiecki M.: "Aggregation constraints and knapsacking problem: a solution to a scheduling problem on a laser cutting machine", I.E.E.E. International Workshop on intelligent robots and systems'91, Osaka, JAPON, Nov. 2-5, 1991 ( à paraître).

Nous présentons le principal résultat utilisé pour la contraction de contraintes.

Le problème posé est le suivant:

$$(Pb) \left[ \begin{array}{l} \text{Max } c \cdot x \\ \text{Sous les} \\ \text{contraintes } A \cdot x = B \\ \text{avec } x_i, a_i \in \{0,1\}, c_i \in \mathbb{Z}, b_i \in \mathbb{N} \quad \forall i = 1, \dots, m \end{array} \right.$$

Le théorème [38] suivant a pour objectif de permettre la contraction d'un système de 2 équations en une seule vérifiant les contraintes de base.

Pour un système de  $n$  équations, il suffit d'itérer le procédé pour déterminer un système équivalent dans lequel les  $n$  équations sont contractées en une seule.

On suppose:

$$f_i(x) = b_i - \sum_{j=1}^m a_{ij} \cdot x_j \quad i = 1, 2$$

$$\text{et } x = \{ x \in \mathbb{N}^m \mid 0 \leq x_j \leq d_j; j = 1, \dots, m \} \text{ où } d_j \in \mathbb{N}_*^m$$

En outre, pour les résultats de [43], les coefficients des  $a_{ij}$  ( $i=1,2, j=1,\dots,m$ ) sont supposés positifs ou nuls

**Théorème:** [42]

Etant données  $(\delta_1, \delta_2)$  deux entiers naturels, premiers entre eux, vérifiant les conditions suivantes:

i)  $\delta_1$  ne divise pas  $b_2$  et  $\delta_2$  ne divise pas  $b_1$

ii)  $\delta_1 > b_2 - a_2$  et  $\delta_2 > b_1 - a_1$  où  $a_i = \min a_{ij}$   $i = 1, 2$  et  $a_{ij} \geq 0$ ,

alors  $(\delta_1, \delta_2)$  sont des multiplicateurs admissibles.

**Particularité de (Pb):**

Etant donné les valeurs des coefficients de la matrice A (0 ou 1), le théorème nous conduit à des valeurs de  $(\delta_1, \delta_2)$  telles que:

i)  $\delta_1$  ne divise pas  $b_2$  et  $\delta_2$  ne divise pas  $b_1$

ii)  $\delta_1 > b_2$  et  $\delta_2 > b_1$

car  $a_i = \min a_{ij} = 0 \quad i = 1, 2.$

**Note:**

Dans le cas où  $b_i = 1 \quad \forall i = 1, \dots, m$  alors les multiplicateurs peuvent être pris premiers entre eux et  $> 1$ .

Dans l'annexe 4 nous donnons le déroulement de la méthode de contraction appliquée au problème de génération de trajectoires avec coûts symétriques.

On présente maintenant les principaux résultats théoriques qui seront utilisés ultérieurement; pour davantage de détails on renvoie le lecteur à [37]. On considère le problème de programmation mathématique;

$$(P1) \quad \left[ \begin{array}{l} h(x^*) = \min_{x \in X} h(x) \\ \text{avec } g_i(x) \leq 0 \quad i = 1, 2, \dots, m \\ x \in X \end{array} \right.$$

où  $h(x)$  et  $g_i(x)$  sont des fonctions convexes ( $i=1,\dots,m$ ) de  $R^n$  dans  $R$ , et  $X$  est un ensemble fini et discret. On associe à chaque contrainte  $i$  un multiplicateur de Lagrange positif  $\mu_i$  et l'on définit le Lagrangien;

$$L(x, \mu) = h(x) + \mu \cdot g(x) = h(x) + \sum_{i=1}^m \mu_i \cdot g_i(x)$$

Soient  $\bar{x} \in X$  et  $\bar{\mu} \in R_+^m$ . On dit que le couple  $(\bar{x}, \bar{\mu})$  est un point-col de la fonction  $L(x, \mu)$  si

$$L(\bar{x}, \mu) \leq L(\bar{x}, \bar{\mu}) \leq L(x, \bar{\mu})$$

quels que soient  $\bar{x} \in X$  et  $\mu \in R_+^m$ . On démontre alors le théorème suivant, qui permet l'identification d'un point-col:

**Théorème 1:** [38]

Soient  $\bar{x} \in X$  et  $\bar{\mu} \in \mathbb{R}_+^m$ . Le couple  $(\bar{x}, \bar{\mu})$  est un point-col de la fonction  $L(x, \mu)$  si et seulement si:

$$(a) \quad L(\bar{x}, \bar{\mu}) = \min_{x \in X} L(x, \bar{\mu})$$

$$(b) \quad g_i(x) \leq 0 \quad (i=1, \dots, m)$$

$$(c) \quad \bar{\mu}_i \cdot g_i(x) = 0 \quad (i=1, \dots, m)$$

L'intérêt de la notion de point-col vient de ce qu'elle permet l'identification d'un optimum du problème (P1):

**Corollaire 1:**

Soit  $(\bar{x}, \bar{\mu})$  un point-col de la fonction  $L(x, \mu)$ . Alors,  $\bar{x}$  est un optimum global du problème (P1).

**Démonstration:**

Soit  $(\bar{x}, \bar{\mu})$  un point-col. Alors, d'après le théorème précédent,

$$L(\bar{x}, \bar{\mu}) = f(\bar{x})$$

D'après ce même théorème et la définition d'un point-col, quel que soit  $x \in X$ , on a:

$$L(\bar{x}, \bar{\mu}) \leq L(x, \bar{\mu}) = h(x) + \bar{\mu} \cdot g(x) \leq h(x)$$

d'où  $\bar{x}$  est l'optimum du problème (P1).

Lorsqu'il existe un point-col, la résolution du problème P1 peut se ramener à la détermination d'un point-col de la fonction  $L(x, \mu)$ . On définit la fonction duale  $w(\mu)$ , quel que soit  $\mu \in \mathbb{R}_+^m$ , par:

$$(P(\mu)) \quad w(\mu) = \min_{x \in X} L(x, \mu)$$

et le problème dual de (P1) par:

$$(D1) \quad w(\mu^*) = \max_{\substack{\mu \\ \mu \in \mathbb{R}_+^m}} w(\mu)$$

La recherche d'un point-col peut se ramener alors à la résolution du problème dual (D1), qui possède un certain nombre de propriétés importantes et fondamentales, même en l'absence de points-cols. Ces propriétés sont résumées dans le théorème suivant :

### Théorème 2:[38]

Soient (P1) et (D1) un couple de problèmes primal-dual. Alors, les propriétés suivantes sont vérifiées:

#### Propriété 1:

Quels que soient la solution réalisable  $x$  du problème (P1) et  $\mu \in \mathbb{R}_+^m$ ,  $w(\mu)$  est un minorant de  $h(x)$ .

#### Propriété 2:

La valeur maximale  $w(\mu^*)$  du dual est inférieure ou égale à la valeur minimale  $h(x^*)$  du primal.

**Propriété 3:**

La fonction  $w(\mu)$  est concave.

**Propriété 4:**

Si le Lagrangien  $L(x, \mu)$  a un point-col, alors  $w(\mu^*) = h(x^*)$ .

On voit alors que si le problème

$$(P(\mu)) \quad w(\mu) = \min_{x \in X} h(x) + \mu \cdot g(x)$$

peut être résolu "facilement", on peut trouver des bornes inférieures pour la valeur optimale du problème primal (P1) par la résolution du problème dual, en général plus facile puisqu'il s'agit de la maximisation d'une fonction concave. Dans le paragraphe suivant nous allons appliquer la théorie de la dualité au problème du Knapsack en variables bivalentes.

**Relaxation Lagrangienne appliquée au Knapsack:**

Nous rappelons que le problème d'optimisation est le suivant:

$$(B) \quad \left[ \begin{array}{l} \text{maximiser} \quad \sum_{j=1}^n f_j x_j \\ \text{sous} \\ \text{la contrainte} \quad \sum_{j=1}^n a_j x_j = b \\ x_j \in \{0,1\} \end{array} \right.$$

La théorie de la dualité étant présentée dans un cadre général, nous abordons son application au problème du "Knapsack". Pour le problème (B) on fait la correspondance avec le problème (P1) pour lequel le dual est:

$$- (P(\mu)) \quad w(\mu) = \min_{x \in X} h(x) + \mu \cdot g(x)$$

le dual de (B) est:

$$- (B(\mu)) \quad w(\mu) = \max_{x \in X} f(x) + \mu \cdot g'(x)$$

avec:

$$* g'(x) = b - a \cdot x$$

$$* x \in \{0,1\}$$

d'où le dual du problème (B):

$$- (B(\mu)) \quad w(\mu) = \max_{x \in X} f(x) + \mu \cdot (b - a \cdot x)$$

Les méthodes dites de relaxation Lagrangienne ont pour objectif de donner une évaluation très serrée dans les cas:

- d'une recherche d'heuristique,
- d'une évaluation par défaut pour une étude par énumération implicite.

Elles diminuent ainsi dans des proportions considérables le nombre de sommets de l'arborescence qu'il est nécessaire de visiter. L'expérience a montré que malgré le surcroît de calcul à chaque évaluation, cette approche conduisait à des procédures relativement efficaces (ainsi pour le voyageur de commerce a-t-on pu passer de cas à 40 villes à des cas à 100 voire 300 villes [...]).

Dans le cas du Knapsack , la formulation de la relaxation Lagrangienne est la suivante:

$$RB(\mu) = \max f x + \mu(b-ax) \text{ sous la contrainte } x \in \{0,1\}$$

Dans [38], on calcul une valeur optimale de  $\mu$  telle que  $RB(\mu)$  soit maximum. On la recherche de la manière suivante:

$$RB(u_i) = \max f x + u_i(b-ax) \text{ sous la contrainte } x \in \{0,1\}$$

où:  $u_i = f_i/a_i$  est calculé à l'optimum de (B) par relâchement des contraintes d'intégrité. C'est à dire que  $x_i$  doit vérifié:

$$- 0 \leq x_i \leq 1 \quad i=1,..,m$$

Cette relaxation sera utilisée lors de la recherche d'une solution optimale par énumération implicite.

Nous présentons l'algorithme de résolution général de (B) [..]. Les différentes étapes sont:

A) résolution de la relaxation de (B),

B) détermination de l'encadrement:

\* majorant  $v(B)$ ,

\* minorant  $v(B)$ ,

C) énumération implicite par la procédure S.E.P.

#### A) Résolution de (B)

(B) sera résolu (comme dans [38]) en déterminant un sous-ensemble  $M$  de  $\{1, \dots, n\}$  tel que:

$$\left[ \begin{array}{l} \sum_{j \in M} a_j \leq b < \sum_{j \in M \cup \{i\}} a_j \\ \forall j \in M \quad c_j/a_j \geq c_p/a_p \quad \forall p \in M ; \quad c_i/a_i = \max \{ c_p/a_p \in M \} \end{array} \right.$$

#### Note:

Cette détermination se fait par un algorithme de complexité linéaire en moyenne,  $x$  est défini comme suit:

$$\bar{x} = \left[ \begin{array}{l} \bar{x}_j = 1 \quad \forall j \in M \\ \bar{x}_i = (b - \sum_{j \in M} a_j)/a_i \\ \bar{x}_j = 0 \quad \forall j \in L = \{1, \dots, n\} \setminus \{M \cup \{i\}\} \end{array} \right. \quad (1)$$

avec:

- M : ensemble des indices des variables de base,
- L : ensemble des indices des variables hors base.
- i : indice de base.

La résolution de (B) nécessite le classement par ordre décroissants des rapports  $c_i/a_i$ :

$$- c_1/a_1 \geq c_2/a_2 \geq \dots \geq c_n/a_n.$$

### B) Encadrement de (B)

Dans l'approche de l'optimisation par énumération implicite, la connaissance d'un majorant et d'un minorant du problème de base est nécessaire. Nous donnerons ici la façon d'obtenir ces valeurs sans développer la théorie pour lequel le lecteur se référera à [38].

#### i) minorant $\underline{v}(B)$

La détermination de  $\underline{v}(B)$  permet de réduire l'arborescence des solutions réalisables. Celui-ci est obtenu par l'algorithme suivant :

1.  $\underline{v}(B) := 0$ ;  $j := 0$ ;
2. si  $j \leq n$  alors  $j := 0$  sinon stop.
3. si  $a_j \leq b$  alors  $\underline{v}(B) := \underline{v}(B) + c_j$ ;  
 $b := b - a_j$ ;  
 si  $b = 0$  alors stop.
- 4 aller en 2.

ii) Majorant  $\bar{v}(B)$

Le majorant utilisé se base sur la relaxation Lagrangienne du problème de base (B) ( une autre technique consiste à résoudre (B) en continu [38])

C) Enumération implicite

Cette méthode est liée à la technique énumérative dite L.L.F.O. qui consiste à gérer la liste d'indices PF des variables fixées à 0 ou 1 associée au déplacement dans l'arborescence:

- descente dans l'arborescence:

i) lorsqu'une variable  $x_k$  est fixée temporairement à 0 ou 1 alors PF devient  $PF \cup \{k\}$  et on souligne cet indice (permet un repérage dans PF en cours de descente).

$[X_1$  devient  $X_1 \cup \{k\}$  si  $x_k = 1$ ;  $X_0$  devient  $X_0 \cup \{k\}$  si  $x_k = 0]$   
 et  $X_2 = \{1, \dots, n\} \setminus \{x_1 \cup x_0\}$

ii) le choix d'attribuer la valeur 1 à une variable  $x_k$  se traduit par le remplacement:  $PF = PF \cup \{k\}$ .

$[X_1$  devient  $X_1 \cup \{k\}$  si  $x_k = 1$ ;  $X_0$  devient  $X_0 \cup \{k\}$  si  $x_k = 0]$   
 et  $X_2 = \{1, \dots, n\} \setminus \{x_1 \cup x_0\}$

- remontée dans l'arborescence:

lorsqu'il est démontré l'inutilité de résoudre un sous-problème, la remontée jusqu'au noeud associé au dernier choix se traduit par la recherche dans PF de l'indice non souligné situé le plus à droite pour:

- d'une part le souligner,

cherche à améliorer le minorant initial.

v) remontée dans l'arborescence : PF = {1,4,5,2} ( $X_1 = \{1\}$ ,  $X_0 = \{2,4,5\}$ )

$$(P) \equiv (B \mid x_1 = 1, x_2 = x_4 = x_5 = 0)$$

$$(P) \begin{cases} 52 + \max 40x_3 + 736x_6 \\ \text{s. c. } 40x_3 + 777x_6 \leq 792 \\ x_3, x_6 = 0 \text{ ou } 1 \end{cases}$$

$$\bar{v}(P) = 835 < \underline{v}(B); \text{ il est inutile de résoudre } (P)$$

vi) remontée dans l'arborescence : PF = {1} ( $X_1 = \emptyset$ ,  $X_0 = \{1\}$ )

$$(P) \equiv (B \mid x_1 = 0)$$

$$(P) \begin{cases} \max 17x_2 + 40x_3 + 793x_4 + 813x_5 + 736x_6 \\ \text{s. c. } 15x_2 + 40x_3 + 793x_4 + 820x_5 + 777x_6 \leq 837 \\ x_j = 0 \text{ ou } 1 \quad j = 2, \dots, 6 \end{cases}$$

$$\bar{v}(P) = 839 > \underline{v}(B)$$

A cette étape, on réitère la phase de descente. D'où l'arborescence associée au problème figure A.3.

$$RB(44) = 835$$

- d'autre part, supprimer tous les indices situés à sa droite.

Les particularité de la méthode sont les suivantes:

- 1) tri des variables suivant l'ordre décroissant des rapports  $c_j/a_j$ ,
- 2) la séparation s'opère sur la variable d'indice  $j^*$ , telle que:

$$c_{j^*}/a_{j^*} = \max \{ c_j/a_j \mid j \in X_2 \}$$

- 3) l'évaluation par excès en un noeud utilise la valeur de la relaxation Lagrangienne telle que:

$$\bar{v}(P) = u_i \cdot b + \max \{ (c - u_i \cdot a) \cdot x \mid x_j = 0 \ \forall j \in X_0; x_j = 1 \ \forall j \in X_1; x_j = 0 \text{ ou } 1 \ \forall j \in X_2 \}$$

Cette relaxations étant appliquée au sous-problème considéré.

### Application numérique

Dans cette application nous nous proposons de présenter une partie du déroulement de l'algorithme (pour plus de détails, le lecteur pourra se référer à [38])

On suppose à avoir à explorer les solutions du problème (B) suivant:

$$(B) \left[ \begin{array}{l} \max \quad 52x_1 + 17x_2 + 40x_3 + 793x_4 + 813x_5 + 736x_6 \\ \text{s. c.} \quad 45x_1 + 15x_2 + 40x_3 + 793x_4 + 820x_5 + 777x_6 \leq 837 \\ x_j = 0 \text{ ou } 1 \quad j = 1, \dots, 6 \end{array} \right.$$

en sachant que  $\bar{v}(B) = 839$  et  $v(B) = 833$

i) choix d'attribuer 1 à  $x_1$ :  $PF = \{1\}$  ( $X_1 = \{1\}$ ,  $X_0 = \emptyset$ )

$$(P) \equiv (B \mid x_1 = 1)$$

$$(P) \begin{cases} 52 + \max 17x_2 + 40x_3 + 793x_4 + 813x_5 + 736x_6 \\ \text{s. c. } + 15x_2 + 40x_3 + 793x_4 + 820x_5 + 777x_6 \leq 792 \\ x_j = 0 \text{ ou } 1 \quad j = 2, \dots, 6 \end{cases}$$

$a_4 = 793 > b(P)$  et  $a_5 = 820 > b(P) \implies x_4 = x_5 = 0$  ( $X_0 = \{4,5\}$ ). On les souligne pour les différencier par rapport à la variable  $x_1$ .

$$PF = \{1, \underline{4}, \underline{5}\}$$

$$\implies \bar{v}(P) = 837 > \underline{v}(B)$$

ii) choix d'attribuer la valeur 1 à  $x_2$ :  $PF = \{1, \underline{4}, \underline{5}, 2\}$  ( $X_1 = \{1,2\}$ ,  $X_0 = \{4,5\}$ )

$$(P) \equiv (B \mid x_1 = x_2 = 1, x_4 = x_5 = 0)$$

$$(P) \begin{cases} 69 + \max 40x_3 + 736x_6 \\ \text{s. c. } 40x_3 + 777x_6 \leq 777 \\ x_3, x_6 = 0 \text{ ou } 1 \end{cases}$$

$$\implies \bar{v}(P) = 837 > \underline{v}(B)$$

iii) choix d'attribuer la valeur 1 à  $x_3$ :  $PF = \{1, \underline{4}, \underline{5}, 2, 3\}$  ( $X_1 = \{1,2,3\}$ ,  $X_0 = \{4,5\}$ )

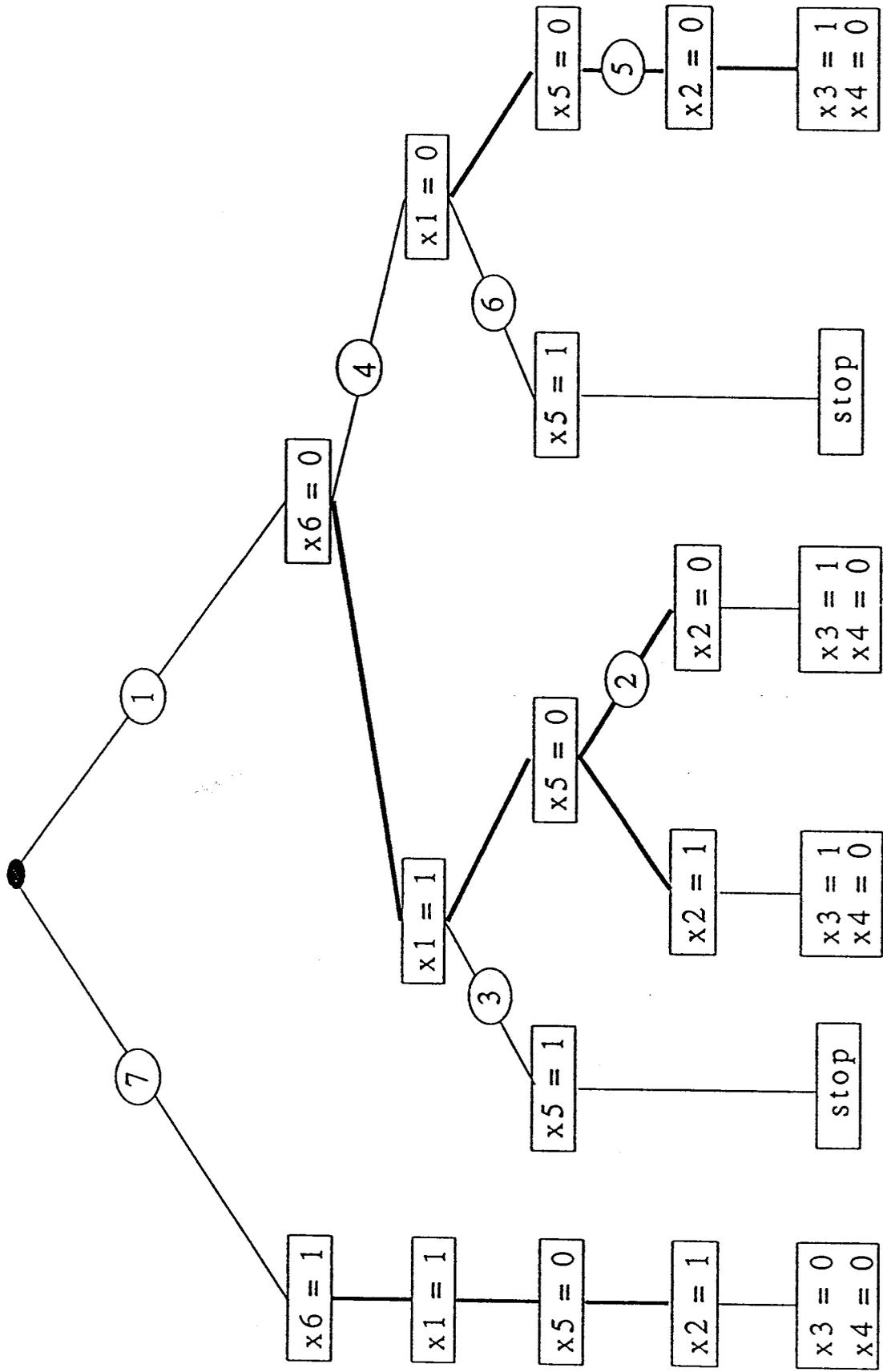
$\implies x_6 = 0$  et  $PF = \{1, \underline{4}, \underline{5}, 2, 3, \underline{6}\}$ ; la solution associée  $x$  est telle que  $cx = 109 < \underline{v}(B)$

iv) remontée dans l'arborescence:  $PF = \{1, \underline{4}, \underline{5}, 2, 3\}$

$$(X_1 = \{1,2\}, X_0 = \{3,4,5\})$$

$\implies x_6 = 1$  et  $PF = \{1, \underline{4}, \underline{5}, 2, 3, \underline{6}\}$ ; la solution associée  $x$  est telle que  $cx = 837 > \underline{v}(B)$

$\implies \underline{v}(B) = 837$ , ici on se trouve en une feuille de l'arborescence, et c'est ici que l'on



837

109

92

833

Figure A1 : Arborescence associée à l'exemple numérique.

Après avoir reformulé notre problème, nous donnons dans cette annexe l'algorithme détaillé qui reprend, à quelques modifications près, celui présenté en annexe 3. Les principales modifications interviennent au niveau de l'exploration des solutions et de la détermination du minorant. En effet, la particularité des problèmes traités (optimisation de l'avance du tissu et génération de trajectoire), demande quelques modifications (voir chap. V).

Le premier paragraphe est relatif à l'algorithme de résolution du problème de génération de trajectoires et à son déroulement par une application numérique.

#### APPLICATION AU PROBLEME Pb2: cas symétrique

##### Application numérique développée du chapitre VI

Soit le problème à trois tâches, pour lequel on détermine la matrice de coûts symétrique  $C (n_1 = 3)$ : (chap. IV et VI)

	1	2	3	4	5	6	7
1	0	18	25	32	15	9	13
2	18	0	11	7	17	25	38
C = 3	25	11	0	52	44	66	39
4	32	7	52	0	10	22	16
5	15	17	44	10	0	33	45
6	9	25	66	22	33	0	26
7	13	38	39	16	45	26	0

1) Formulation du problème

Au chapitre VI, nous avons défini les données du problème comme suit:

- nombre de tâches:  $n = 3$ ,
- nombre de variables décisionnelles:  $m = 2.n + 1 = 7$ ,
- nombre de variables réduit sur un trajet:  $h = n + 1 = 4$ .

D'où la formulation associée à ce problème:

$$\begin{array}{l}
 \min \sum_{i=1}^7 \sum_{j>i} c_{ij} x_{ij} \\
 \sum_{j=2}^7 x_{1j} = 2 \quad (1) \\
 \sum_{j>i} x_{ij} = h = m - n_1 = 2*n_1 + 1 - n_1 = 4 \quad (2) \\
 \sum_{j>i} x_{ij} + \sum_{j<i} x_{ij} = 1 \quad \begin{array}{l} i = 2,4,6 \\ j = 3,5,7 \end{array} \quad (3) \\
 x_{ij} = 0 \text{ ou } 1
 \end{array}$$

2) Modification du problème

En développant les contraintes (1), (2) et (3) on tire

- (1) =  $x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} - 2 = 0$
- (2) =  $x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{24} + x_{25} + x_{26} + x_{27} + x_{34} + x_{35} + x_{36} + x_{37} + x_{46} + x_{47} + x_{56} + x_{57} - 4 = 0$
- (3) =  $x_{12} + x_{24} + x_{25} + x_{26} + x_{27} - 1 = 0,$   
 $x_{13} + x_{34} + x_{35} + x_{36} + x_{37} - 1 = 0,$

$$x_{14} + x_{24} + x_{34} + x_{46} + x_{47} - 1 = 0,$$

$$x_{15} + x_{25} + x_{35} + x_{56} + x_{57} - 1 = 0,$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} - 1 = 0,$$

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} - 1 = 0,$$

Ce système peut être représenté de la manière suivante:

$$- (1) \iff f_1(x) = 0$$

$$- (2) \iff f_2(x) = 0$$

$$- (3) \iff f_3(x) = 0$$

$$\iff f_4(x) = 0$$

$$\iff f_5(x) = 0$$

$$\iff f_6(x) = 0$$

$$\iff f_7(x) = 0$$

$$\iff f_8(x) = 0$$

En utilisant la technique de contraction pour lequel les  $\delta_i$  sont premiers entre eux et doivent satisfaire le théorème de l'annexe 1. La contraction de ce système nous conduit aux multiplicateurs  $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7, \delta_8)$  tels que:

$$\delta_1 = 1, \delta_2 = 2, \delta_3 = 3, \delta_4 = 5, \delta_5 = 7, \delta_6 = 11, \delta_7 = 13, \delta_8 = 17$$

d'où l'équation unique:

$$6x_{12} + 8x_{13} + 10x_{14} + 14x_{15} + 16x_{16} + 20x_{17} + 12x_{24} + 16x_{25} + 16x_{26} + 22x_{27} + 14x_{34} + 18x_{35} + 20x_{36} + 24x_{37} + 22x_{46} + 26x_{47} + 26x_{56} + 30x_{57} = 66$$

#### Application:

Prenons un chemin quelconque et vérifions la validité de la contrainte unique. Si cette contrainte est vraie il faut qu'elle vérifie pour tout  $x_{ij} \in S$  solution réalisable:

$$\sum_{i=1}^5 \sum_{j=2}^7 a_{ij} \cdot x_{ij} = 66 \text{ pour tout } x_{ij} \in S$$

exemple: soit le chemin 1,2,3,4,5,6,7,1 qui donne:

$x_{ij} = 1$  si  $(ij) \in$  chemin, 0 sinon et on tire:

$$6x_{12} + 0x_{23} + 14x_{34} + 0x_{45} + 26x_{56} + 0x_{67} + 20x_{17} = 66$$

En développant à partir de la matrice de coût C on obtient le problème du Knapsack en maximisation suivant, en posant  $m_{ij} = -c_{ij}$ :

$$\text{maximiser } -18x_{12} - 25x_{13} - 32x_{14} - 15x_{15} - 9x_{16} - 13x_{17} - 7x_{24} - 17x_{25} - 25x_{26} - 38x_{27} - 52x_{34} - 44x_{35} - 66x_{36} - 39x_{37} - 22x_{46} - 16x_{47} - 33x_{56} - 45x_{57}$$

$$\text{sous la contrainte } 6x_{12} + 8x_{13} + 10x_{14} + 14x_{15} + 16x_{16} + 20x_{17} + 12x_{24} + 16x_{25} + 16x_{26} + 22x_{27} + 14x_{34} + 18x_{35} + 20x_{36} + 24x_{37} + 22x_{46} + 26x_{47} + 26x_{56} + 30x_{57} = 66$$

### 3) réindexation des coûts et variables:

Après réindexation on obtient le problème équivalent:

$$\text{maximiser } -18x_1 - 25x_2 - 32x_3 - 15x_4 - 9x_5 - 13x_6 - 7x_7 - 17x_8 - 25x_9 - 38x_{10} - 52x_{11} - 44x_{12} - 66x_{13} - 39x_{14} - 22x_{15} - 16x_{16} - 33x_{17} - 45x_{18}$$

$$\text{sous la contrainte } 6x_1 + 8x_2 + 10x_3 + 14x_4 + 16x_5 + 20x_6 + 12x_7 + 16x_8 + 16x_9 + 22x_{10} + 14x_{11} + 18x_{12} + 20x_{13} + 24x_{14} + 22x_{15} + 26x_{16} + 26x_{17} + 30x_{18} = 66$$

### 4) hiérarchisation et calcul des paramètres

#### 4.1 hiérarchisation

Après le calcul des  $m_j/a_j$ , on réindexe les variables suivant les rapports décroissants. Ce qui nous conduit à la présentation finale du problème (B):

$$\text{maximiser } -9x_1 - 7x_2 - 16x_3 - 13x_4 - 22x_5 - 17x_6 - 15x_7 - 33x_8 - 45x_9 - 25x_{10} - 39x_{11} - 38x_{12} - 44x_{13} - 18x_{14} - 25x_{15} - 32x_{16} - 66x_{17} - 52x_{18}$$

sous la contrainte  $16x_1 + 12x_2 + 26x_3 + 20x_4 + 22x_5 + 16x_6 + 14x_7 + 26x_8 + 30x_9 + 16x_{10} + 24x_{11} + 22x_{12} + 18x_{13} + 6x_{14} + 8x_{15} + 10x_{16} + 20x_{17} + 14x_{18} = 66$

#### 4.2 détermination de la variable de base $u_i$

Cette variable est obtenue en déterminant:

- l'ensemble M des variables de base,
- la variable de base  $x_i$ ,
- l'ensemble L des variables hors bases.

d'où:

$$\bar{x} = \begin{cases} \bar{x}_j = 1 & j = \{1, 2, 3\} \quad M = \{1, 2, 3\} \\ \bar{x}_i = -13/20 & i = 4 \\ \bar{x}_j = 0 & j = \{5, \dots, 18\} \quad L = \{5, \dots, 18\} \end{cases}$$

Le multiplicateur de Lagrange associé est  $u_i = m_i/a_i$ :

$$u_4 = m_4/a_4 = -13/20 = -0.65$$

#### 4.3 calcul d'un majorant: $\bar{v}(B)$

La détermination de  $\bar{v}(B)$  est obtenue par relaxation Lagrangienne associée à  $u_4$ .

$$\bar{v}(B) = RB(u_4) = -46$$

#### 4.4 calcul d'un minorant: $\underline{v}(B)$

A partir du classement des rapports décroissant des  $m_j/a_j$  on détermine un minorant de la manière suivante:

- faire la somme des coûts dans le sens des rapports décroissants des  $m_j/a_j$  tel que l'on ait un chemin réalisable.

Ce qui correspond à l'algorithme suivant:

1.  $j := 1, \underline{v}(B) := 0;$
2. Si  $m_j \in$  Séquence réalisable alors  $\underline{v}(B) := \underline{v}(B) + m_j;$   
Sinon  $j := j + 1;$  aller en 2
3. Si  $j := n$  alors Stop.

Dans l'exemple proposé on prendra la séquence de coûts:

$$\underline{v}(B) = m_1 + m_3 + m_6 + m_{15} = -67$$

Cette séquence correspond au trajet 1 6 7 4 5 2 3 1 qui, par symétrie, est le trajet: 1 3 2 5 4 7 6 1 pour lequel les valeurs des coûts sur les tâches ne sont pas comptabilisées

### 5) Enumération implicite

On utilise la méthode P.S.E.P. proposée en annexe 3 pour laquelle une modification s'impose. Lors de la descente ou de la remontée, on fixe à 1 ou 0 les variables appartenant à une séquence réalisable. En définitif, on parcourt l'arbre des solutions réalisables liées au problème de base.

A la fin de cette procédure on obtient la valeur optimale du problème:

$$v^*(B) = -67$$

### 6) Conclusion

La recherche d'une solution optimale est très vite atteinte de par la valeur du minorant.

