

50376
1994
63

50376
1994
63

THESE

N° d'ordre 1239

Présentée à

L'université des Sciences et Technologies de LILLE

pour l'obtention du titre de

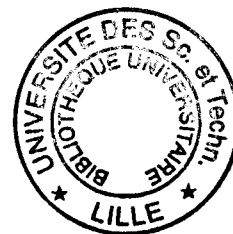
DOCTEUR

en Productique: Automatique et Informatique Industrielle

par

Gilles ROUSSEL

Maître ès Sciences EEA



***Optimisation du placement de formes irrégulières sur
matières planes.***

Application à l'industrie de la confection

Soutenue le: 10 Janvier 1994

devant la commission d'examen:

Jury:

<u>Président:</u>	Pierre VIDAL	Professeur à L'USTL Lille
<u>Rapporteurs:</u>	Alain JUTARD	Professeur à L'INSA de Lyon
	René SOENEN	Professeur à L'université de Valenciennes
<u>Examineurs:</u>	Fausto GASPERINI	Directeur Général des Recherches du programme BRITE/EURAM C.E.E. Bruxelles
	Jack Gérard POSTAIRE	Professeur à L'USTL Lille
	Jean Marc TOULOTTE	Professeur à L'USTL Lille
<u>Directeurs:</u>	Salah MAOUCHE	Maître de conférence à L'USTL Lille
	Pierre VIDAL	Professeur à L'USTL Lille

Avant propos

L'étude qui est relatée dans ce rapport a été menée au sein du C.A.L. -Centre d'Automatique de Lille-. Il est le résultat de trois années de travail, au sein d'une équipe enthousiaste et solidaire. Il est donc juste de remercier ici les personnes, sans qui le présent travail n'aurait ni commencé, ni terminé. Il m'est donc agréable:

- De présenter mes plus vifs remerciements à **Monsieur le Professeur Pierre VIDAL**, directeur du C.A.L., initiateur et codirecteur de cette étude. Sa clairvoyance et ses conseils avertis m'ont permis de suivre un objectif bien défini. Son contact aisé a largement contribué à entretenir une très bonne qualité des relations de travail.

- De féliciter **Monsieur Salah MAOUCHE**, maître de Conférence à l'USTL Lille I, codirecteur de cette thèse pour sa collaboration fidèle. Il fut, tout au long de cette étude, mon interlocuteur privilégié dont les conseils éclairés m'ont permis d'optimiser la trajectoire de ma recherche. Ses qualités humaines ont grandement facilité une relation de travail infaillible et durable. Plus qu'un collaborateur, il est un ami. Je lui dois donc toute ma reconnaissance.

Il n'est point de thèse sans le jugement impartial des rapporteurs extérieurs. Leur travail d'appréciation, de contrôle, et de correction mérite une reconnaissance particulière. J'exprime donc toute ma reconnaissance à

Monsieur le Professeur **Alain JUTARD**, pour sa décision spontanée de participer à ce jury.

et à Monsieur le Professeur **René SOENEN**, dont la connaissance et l'expérience réputées dans le domaine du placement et de la confection constituent un apport supplémentaire à ce travail.

J'apprécie beaucoup l'honneur que me font ces automaticiens de renom en jugeant ce travail.

Je remercie aussi très sincèrement:

- **Monsieur Fausto GASPERINI**, Directeur Général des Recherches du programme Européen BRITE/EURAM pour sa participation. Son implication au sein de la communauté Européenne confère à ce jury une ouverture internationale et industrielle

-Monsieur le Professeur **Jack Gérard POSTAIRE**, pour avoir spontanément accepté de faire parti du jury. Je suis très honoré pour l'intérêt qu'il a porté à mon travail et je lui exprime ma très grande considération ainsi que mon estime.

-Monsieur le Professeur **Jean Marc TOULOTTE** pour sa collaboration indirecte ou celle des membres de son équipe de recherche, en m'aidant à me familiariser avec le langage de simulation orienté objet Smalltalk. Je lui exprime bien sûr ma reconnaissance et mon estime.

Leur présence dans ce jury ainsi que leur jugement m'honorent.

Enfin, je ne pouvais terminer cette liste de remerciements, sans citer la collaboration de **Mme Annick Pignon** et **Mme Marie Claude El Mekki** , ainsi que celle de **Mr Bernard Ceurstemont** dont la disponibilité et l'humour permanent amenaient un support technique et moral très appréciable. Je salue également **toutes celles et tous ceux** qui à tous moments m'ont témoigné leur sympathie et leur soutien.

Sommaire

Intoduction Générale	1
-----------------------------------	----------

Chapitre I

Présentation d'une unité de confection et du problème de l'optimisation de la découpe	4
I Objectifs d'optimisation dans le contexte économique de l'industrie Textile.....	4
II Présentation d'une unite de confection.....	5
2.1 La création.....	6
2.1.1 Le styliste :	6
2.1.2 Le modéliste :	6
2.1.3 La patronnière :	6
2.1.4 La mécanicienne - modèle :	7
2.1.5 Evolution de la création	7
2.2 L'industrialisation.....	8
2.2.1 La patronnière gradeuse :	8
2.2.2 Le placeur - traceur	10
2.2.3 Les placements assistés par ordinateur	10
a) Le placement interactif	10
b) Le placement automatique	11
2.3 La fabrication	12
2.3.1 Le traceur	14
2.3.2 Le matelasseur.....	14
2.3.3 Le coupeur.....	17
2.3.4 Le triage	18
III L'enjeu de l'optimisation de la découpe	20
IV Présentation d'une cellule de découpe flexible	21
4.1 Le chargement de la pièce de tissu.....	22
4.2 Le module de visitage	23
4.3 Le module de placement	23
4.4 Le module de découpe	24
4.5 Le module de déchargement et de stockage.....	26
4.6 Coopération des différents processus de l'unité de découpe	26
V Conclusion	31

Chapitre II

Formulation du problème du placement et état de l'art.....	32
I Introduction	32
II Formulation du problème du placement de pièces de vêtements.....	32
2.1 Définitions.....	33
2.1.1 Le placement	33
2.1.2 Définitions relatives à la matière.....	33
a) La laize: W_m	34
b) Repère matière : (O_m, X_m, Y_m)	35

c) Chaîne du tissu	36
d) Trame du tissu	36
e) Axe du droit fil	36
f) Les types de tissu	36
2.1.3 Définitions relatives aux pièces	37
2.2 Les contraintes	39
2.2.1 Les contraintes géométriques générales	39
2.2.1.1 Contrainte de disjonction	39
2.2.1.2 Contrainte d'inclusion à la matière	40
2.2.1.3 Contrainte de début de rouleau	40
2.2.1.4 Contrainte du droit fil	40
2.2.1.5 Contrainte de prise en compte des défauts	42
2.2.2 Contraintes géométriques spécifiques	43
2.2.2.1 La contrainte "dossé - dédossé"	43
2.2.2.2 La contrainte de sens du tissu	44
2.2.2.3 La contrainte de chaîne	44
2.2.2.4 La contrainte de raccord (ou de positionnement relatif)	44
2.2.2.5 La contrainte de positionnement absolue par rapport au motif (ou motif placé)	46
2.2.3 La contrainte temporelle	47
2.3 Les fonctions de but	48
2.3.1 Classification des fonctions objectif	49
2.3.2 Définitions des grandeurs d'évaluation	50
2.3.2.1 grandeurs relatives au placement	51
2.3.2.2 Grandeurs relatives à une association de deux formes	52
2.3.3 Synthèse	53
2.4 Placement en bande	60
2.4.1 Finitude d'une bande	60
2.4.2 Consistance d'une bande	62
III Etat de l'art sur le placement de formes irrégulières	63
3.1 Classification du problème de découpe	63
3.1.1 Objectifs d'optimisation	64
3.1.2 Contraintes de découpe	65
3.1.3 La matière première	67
3.1.4 La commande	68
3.1.5 Les contraintes de placement	69
3.1.6 Les contraintes temporelles de cycle en ligne	69
3.2 Synthèses bibliographiques	69
3.3 Le Placement-découpe des formes irrégulières	70
3.3.1 Placement interactif	71
3.3.2 Placement automatique	72
3.3.2.1 Les méthodes à deux ou n étapes	72
3.3.2.2 Les algorithmes déterministes	74
3.3.2.3 Les algorithmes non déterministes	76
3.3.2.4 Les systèmes experts	78
3.3.3 Placement semi-automatique	79
IV Conclusion	79

Chapitre III

Représentation numérique des formes irrégulières	81
I Introduction	81
II Représentation d'un contour dans l'espace discret.....	82
2.1 Support d'image discrète.....	82
2.2 Objet de référence: la frontière discrète	83
2.2.1 Intersection des contours.....	83
2.2.2 Interpolation des points de la frontière discrète	83
III Description des formes	84
3.1 Les critères d'analyse	84
3.2 Les codes de description de contours.....	86
3.2.1 Les codes exacts.....	86
3.2.2 Les approximations de contour.....	86
3.2.3 Détermination du contour apparent.....	87
IV Description du contour par les " Peignes de formes "	88
4.1 Rectangle circonscrit.....	88
4.2 Les sous-contours.....	89
4.3 Sous-contours apparents stricts.....	91
4.4 Définition des peignes de contour.....	93
4.5 appellation.....	94
4.6 Justification de l'emploi des peignes de contour	95
V Propriétés du codage par "peigne de contour"	95
5.1 Réversibilité	95
5.2 Redondance	96
5.3 Taux de compression	97
5.4 Invariances	98
5.4.1 Non-invariance par rotation d'angle quelconque.....	98
5.4.2 Invariance par translation.....	98
5.4.3 Non-invariance par homothétie.....	98
5.5 Déformation du contour	98
VI Peignes réduits.....	99
6.1 Définition	99
6.2 Approximation polygonale	101
6.2.1 Introduction.....	101
6.2.2 Les algorithmes existants	102
6.2.3 Principe	102
VII Caractérisation des peignes	103
7.1 Longueur d'un sous-contour apparent.....	103
7.2 Surface d'un peigne réduit.....	104
7.3 Taux de la déficience rectangulaire.....	104
7.4 Complexité d'un peigne.....	105
VIII Opérateurs de formes unitaires.....	107
8.1 Rotation de $\pi/2$	107
8.2 Symétries par rapport à $Cr_k X$ et $Cr_k Y$	108
8.2.1 Symétrie par rapport à $Cr_k X$	108
8.2.2 Symétrie par rapport à $Cr_k Y$	109

8.3 Peigne et forme duale.....	109
8.3.1 Définition	110
IX Classe des formes irrégulières : Forme_Modèle	111
9.1 Les variables d'instances	111
9.2 Les variables de classe	111
9.3 Messages et Méthodes	112
X Conclusion	112

Chapitre IV

Imbrication des formes irrégulières	113
I Introduction	113
II Placement de rectangles paraxiaux non disjonctifs.....	114
2.1 Objectif.....	114
2.2 Représentation d'un rectangle paraxial par une matrice.....	115
2.3 Propriétés des matrices de rectangle	115
2.3.1 dimensions d'un rectangle:.....	116
2.3.2 surface d'un rectangle.....	116
2.3.3 Signe d'un rectangle	116
2.3.4 Rectangles particuliers	116
2.3.5 Extraction des vecteurs particuliers	117
2.4 Positionnement relatif de deux rectangles.....	117
2.5 Opérations sur les rectangles.....	118
2.5.1 Réunion de deux rectangles	118
2.5.2 Intersection de deux rectangles	119
2.5.3 Signe des surfaces des rectangles réunion et intersection....	120
a) Réunion	120
b) intersection.....	120
2.6 Changement de repère.....	121
III Concaténation géométrique de deux formes.....	121
3.1 Définition:	121
3.2 Convention:.....	122
3.3 Obtention :.....	122
IV Evaluation de la complémentarité	129
4.1 Optimisation de l'Imbrication	129
4.2 Coordonnées relatives entre deux peignes	130
4.3 Changement de repère.....	131
4.4 Contrainte de disjonction	132
4.5 Simplification de la vérification de la contrainte	134
4.6 Principe de vérification de la contrainte de disjonction :.....	135
4.7 Indices de complémentarité.....	137
4.7.1 Distance quadratique inter-peigne.....	137
4.7.2 Surface du rectangle d'intersection.....	139
4.7.3 Couloir de pénétration.....	140
4.7.4 Surface du rectangle de la concaténation	140
4.7.5 Surface modifiée du rectangle circonscrit.....	141
V Optimisation locale	142

5.1 Choix du critère.....	142
5.2 Réduction du domaine de recherche	142
5.3 Réduction de l'intervalle de recherche dans le cas de matériau uni ...	143
5.3.1 Bornes limites de l'intervalle de recherche.....	143
5.3.2 Mise en correspondance des creux et des bosses	146
5.4 Recherche sous contrainte de motif	149
VI Complément à la classe Forme_Modèle.....	152
VII Influence de l'approximation polygonale	152
7.1 Objectif.....	152
7.2 Description de l'essai.....	153
7.2.1 Choix des formes	153
7.2.2 Algorithme d'essai.....	153
7.3 Résultats.....	153
7.3.1 Présentation.....	153
7.3.2 Analyse.....	154
VII Conclusion.....	156

Chapitre V

Méthodes d'optimisation combinatoire pour la recherche du placement.....	157
I Introduction	157
II Recuit simulé et représentation du problème: 1er essai	158
2.1 Introduction.....	158
2.2 Représentation.....	159
2.3 Générateur de configuration aléatoire.....	159
2.4 Optimisation.....	160
2.5 Vers une autre représentation.....	160
III Représentation en graphe.....	161
3.1 L'espace des configurations	161
3.2 Représentation implicite	164
IV Variation de la fonction de coût et topologie de l'espace de configuration	167
V Méthode stochastique: le recuit simulé.....	170
5.1 Amélioration itérative aléatoire.....	170
5.2 Le recuit simulé.....	171
5.2.1 Analogie avec l'optimisation combinatoire.....	171
5.2.2 Algorithme de Metropolis.....	171
5.2.3 Comportement du recuit simulé sur le graphe	173
5.3 Paramètres de contrôle du recuit simulé	174
5.3.1 Placement initial.....	175
5.3.2 Température initiale T_0	175
5.3.3 Changement d'état.....	175
5.3.4 Loi de décroissance du paramètre de contrôle T	177
5.3.5 Equilibre statistique.....	177
5.3.6 Critère d'arrêt.....	178
5.4 Résultats	179
5.4.1 Constitution de l'encours de placement type.....	179

5.4.2 Etude d'un exemple	180
5.4.3 Ajustement des paramètres	182
a) Ajustement de la vitesse de décroissance en T	182
b) Ajustement du paramètre T_0	184
5.5 Conclusions sur la méthode du recuit simulé.....	185
VI Parcours de graphe: l'algorithme ε -admissible	186
6.1 Introduction	186
6.2 Choix de l'algorithme	186
6.3 Politique de recherche en graphe	187
6.3.1 Recherches	187
6.3.2 L'algorithme A	187
6.3.3 L'algorithme A^*	188
6.4 L'algorithme A_ε	189
6.4.1 Principe	189
6.4.2 Algorithme	190
6.4.3 Heuristique d'estimation de coût h	192
6.4.4 Heuristique de calcul de la longueur de chemin	195
6.4.5 Tactique de recherche persévérante	196
6.4.6 Limitation des noeuds développés	196
6.4.7 Fonction de choix f_c	197
6.5 Résultats avec A_ε	197
6.5.1 Qualité des placements.....	198
6.5.2 Vitesse d'obtention	198
6.5.3 Efficacité	199
VII Comparaison des deux algorithmes d'optimisation	200
7.1 Analyse des résultats.....	200
7.2 Analyse des trajectoires de parcours de graphe	201
7.2.1 Le recuit simulé.....	201
7.2.2 L'algorithme A_ε	203
VII Conclusion.....	205
Conclusion Générale	206
Bibliographie.....	209
Bibliographie personnelle	219
Annexe A: Topologie de l'espace discret	220
Annexe B: Codes de description de contour	222
B.1 Les codes exacts	222
B.1.1 Codes de FREEMAN [FRE 61]	222
B.1.2 Codage par transition [HUA 75]	223
B.1.3 Signature polaire.....	223
B.2 Les représentations approchées	225
B.2.1 Nombre de forme [BRI 79]	225
B.2.2 Strip trees [BAL 81]	226
B.2.3 Squelette de formes [MON 69]	227
B.2.4 Descripteur de Fourier [ZAH 72].....	228

B.2.5 Approximation polynomiale.....	228
Annexe C: Algorithme d'approximation polygonale pour l'obtention des peignes de contour réduit.....	229
Annexe D: Formules de changement de repère.....	230
Annexe E: Algorithme de suppression des points cachés et de reconstruction de la frontière de jonction	231
Annexe F: Algorithme de calcul de la profondeur d'imbrication.....	223
Annexe G: Algorithme de recherche du décalage optimal (tissu uni).....	234
Annexe H: algorithme de recherche du décalage optimal (tissu à motifs).....	235
Annexe I: Elagage des noeuds pendants non ϵ-admissibles.....	237
1.1 Proposition	
1.2 Démonstration	
Annexe J: Couples de formes utilisées dans l'algorithme d'évaluation de l'influence de l'approximation polygonale.....	238

RESUME

Dans cette thèse, nous présentons une méthodologie de résolution du problème du placement de formes quelconques en vue de minimiser la chute de matière dans le processus de découpe en bande de pièces de vêtements. Les contraintes technologiques de la cellule de découpe et l'objectif de production flexible vis à vis du carnet de commande nous a conduit à élaborer un algorithme de placement par bande en temps réel. Dans un premier temps, une méthode originale de description des formes quelconques a été développée. Elle consiste à échantillonner les sous-contours visibles depuis les quatre côtés du rectangle circonscrit à la forme et à générer quatre *peignes de formes*. Ensuite, un algorithme d'approximation polygonal supprime les *dents* redondantes à la description, donnant naissance au *peigne de forme réduit*. Partant de ce principe, un ensemble d'opérateurs a été construit pour évaluer les caractéristiques d'une forme, pour effectuer une transformation géométrique (rotation, symétrie...), évaluer un coefficient de complémentarité et enfin, pour effectuer une concaténation de deux formes positionnées de façon quasi optimale relativement à un critère. Ainsi le placement est vu comme un problème d'optimisation combinatoire dans un arbre, où chaque configuration est représentée comme un noeud de l'espace de représentation. La recherche d'une solution est alors ramenée à un parcours d'arbre. La détermination d'un chemin sous optimal dans l'arbre est réalisée à l'aide de deux algorithmes d'optimisation que nous avons comparé. Le premier, que nous avons adapté au parcours d'arbre, est appelé recuit simulé. Le second, est l'algorithme ϵ -admissible: A_ϵ . C'est ce dernier qui a été retenu pour son efficacité et la faculté de réglage du compromis complexité/ admissibilité.

Introduction générale

L'avenir des métiers de l'habillement est lié, au moins en partie, à l'évolution des moyens dont ils disposent pour produire, ce que nous porterons toujours : les vêtements. Qu'on le veuille ou non, la bataille pour conserver une place prépondérante dans les marchés économiques de cette branche de l'industrie passe par une compétitivité dont le facteur d'accroissement principal est la productivité. Cette constatation nous amène inlassablement à améliorer du même coup notre appareil de production.

L'automatisation des chaînes de fabrication contribue pour une large part à l'objectif de production de masse. Mais il arrive parfois que cette volonté acharnée du "tout automatique" se heurte à des difficultés pratiques. C'est souvent le cas lorsque l'on croit, peut être naïvement que la machine peut remplacer l'homme à tous les niveaux de son intervention, y compris là où il excelle: la création. Cette ambition de supplanter l'Homo Sapiens nourrit les motivations de la recherche, mais la distance qui la sépare de son aboutissement nous permettra de préserver notre souveraineté encore longtemps. Pointer vers l'horizon ne signifie pas l'atteindre.

Parmi les obstacles difficiles à franchir, l'imitation de l'abstraction visuelle semble en être un. Citons, en ce qui nous concerne, l'exemple du placement des patrons de coupe : ceux qui sont encore utilisés au niveau des productions unitaires. Le placeur humain sait, par expérience mais aussi par son pouvoir d'abstraction, quelles pièces il doit apposer à telles autres afin que le placement soit de très bonne qualité, tant au niveau local qu'il vient de modifier qu'au niveau du résultat final. La seule perception visuelle des formes à placer prises individuellement lui suffit pour appréhender leurs rôles au sein d'un placement qui n'existe pas encore. Quelques essais infructueux lui suffisent souvent avant de proposer une solution qui soit celle qu'il retient. Celle-ci est d'ailleurs d'un rendement généralement très bon. Mais voilà, victime de sa lenteur naturelle à essayer plusieurs combinaisons d'un problème qui en compte un nombre souvent très large, il est mis en compétition avec une machine dont précisément la force est de compter: l'ordinateur.

La réalité économique engendra donc la rivalité homme/machine tout en acceptant que les méthodes que la machine utilise ne constituent qu'une approche lointaine des mécanismes humains de réflexion. Ici l'abstraction est remplacée par l'itération d'un processus constructif dont le résultat instantané est évalué a posteriori (le placement final), ou a priori par estimation depuis une situation intermédiaire. Le but étant de ne retenir que les solutions qui nous donnent satisfaction au sens d'une règle que l'on s'est définie au départ: un critère. Cette forme de mécanisme marque le ton de notre travail qui s'appuiera sur ce que l'on appelle plus formellement: l'optimisation combinatoire.

La présente étude n'est qu'une tentative d'imitation de ce que le placeur humain sait faire par instinct d'une part, par expérience d'autre part: l'optimisation du placement. Mais celle-ci se veut résolument insolente vis à vis de lui, car son but est de faire aussi bien, mais plus vite. Pour cela, la méthode présentée suggère de nouveaux points de vue pour espérer approcher de plus près une théorie encore mal éclairée.

Dans le *chapitre I*, nous tenterons de présenter à travers une visite accélérée d'un atelier de confection, les bases de travail sur lesquelles repose le métier de l'habillement ainsi que les obstacles économiques ou techniques qu'il rencontre. Nous aborderons de plus près les habitudes de travail de l'atelier de coupe, notre centre d'intérêt, sans oublier les ateliers voisins dont les interactions imposent la prise en compte de certains paramètres comme le temps par exemple. Nous énoncerons ensuite les hypothèses de travail, et en particulier notre support d'application, dont les développements théoriques firent l'objet de travaux précédents. Elle est l'illustration d'un cas particulier d'une classe de processus de transformation automatisés : la cellule flexible de découpe. Cette structure répond bien au flux de commandes aléatoires dans le temps, pour produire des petites séries, de complément ou de luxe. Sa vocation n'est pas ici de supplanter la technique très rentable du matelassage, mais de pouvoir fournir "juste à temps" les pièces découpées à l'atelier d'assemblage. Nous terminerons sur la description des interactions temporelles synchronisant les différents processus environnants qui sont en relation avec la cellule.

Dans le *chapitre II* nous avons cherché à définir ou à rappeler le formalisme précis décrivant notre problème de placement. Outre les contraintes générales relatives à tous les placements de formes, nous y avons ajouté celles spécifiant le placement sur textiles. Les particularités liées au tissu uni ou à motifs sont ici prises en compte. Les défauts de la matière sont mentionnés, mais le traitement de ces derniers au niveau de l'algorithme n'a pas été intégré. La définition du cahier des charges ne pouvait se concevoir sans préciser les objectifs d'optimisation. Une analyse et un classement des critères d'optimisation relevés au grès de notre étude bibliographique nous aideront à réaliser un choix de la fonction de but. Celle qui est retenue, la perte en bande, est en adéquation avec la méthode de travail pour laquelle nous opterons: la réalisation de bandes. Elle présente l'avantage, de pouvoir l'utiliser dans une règle de consistance de bande sans fixer arbitrairement la longueur de celle ci. Mais pour garantir une limite dimensionnelle à nos placements, nous imposons quand même une règle souple de terminaison de bandes. Enfin, ce chapitre se terminera en proposant une classification, ainsi que par un point assez large sur l'état de l'art concernant le problème du placement. Une telle analyse nous est apparue utile pour aiguïser nos connaissances et cerner les manquements de la théorie ... et de la pratique.

Le *chapitre III* se veut novateur dans le domaine de la description des formes irrégulières. Il propose un concept original de codage, les *peignes de contours*, particulièrement adapté au problème, au regard de la contrainte la plus "agréable" du problème : le droit fil. S'apparentant à la déficience convexe, ce codage utilise la spécificité de la contrainte de paraxialité propre au textile et reprend un principe similaire adapté au cas du rectangle circonscrit. On verra que cette façon de

connaître les formes en divisant leurs contours selon les quatre côtés du rectangle permet de ne considérer leur description qu'au voisinage de la zone d'intérêt : le côté de mise en imbrication. D'autre part, soucieux de minimiser la complexité des traitements que nous aurons à faire subir aux peignes de contours, on supprimera les informations redondantes du code à l'aide d'un algorithme d'approximation polygonale développé à cet effet. Cette transformation supplémentaire donne naissance aux *peignes réduits*. Les propriétés du code résultant du principe de description seront passées en revue. Elles renseignent sur les possibilités ou les limitations liées à l'emploi de cette technique. Enfin, quelques opérations relatives à la modification d'une seule forme seront élaborées en vue de leur utilisation dans la construction du placement. Remarquons qu'à partir de ce chapitre, nous percevons les formes comme des objets auxquels on attache une structure de données contenant toutes les informations qui leur sont propres. Une classe d'objets définie par les peignes réduits de contours sera appelée: la classe `Forme_Modèle`.

Le *chapitre IV* présente le noyau de calculs du problème du placement. Il présente les outils nécessaires à la construction géométrique de l'assemblage. Après avoir formalisé analytiquement les opérations d'intersection et de réunion de rectangles, nous répéterons la démarche en considérant celle des formes irrégulières. Nous construirons tout d'abord l'opérateur de concaténation de deux formes qui, lorsqu'il est utilisé itérativement, permet d'associer les éléments de l'amalgame. Viendra ensuite l'opérateur permettant d'évaluer la qualité d'une imbrication locale entre deux formes et donc, par voie de conséquence de déterminer la position optimale vis-à-vis d'un critère surfacique que nous définirons. Notons que les opérations décrites dans ce chapitre prennent intrinsèquement en compte toutes les contraintes à ce niveau du traitement. Nous nous efforcerons tout au long de ce chapitre d'essayer de diminuer la complexité de nos opérateurs autant que possible, sachant que ceux-ci représentent le noyau de calcul de l'algorithme de placement. Enfin, nous préciserons les modifications qu'il faut apporter afin de prendre en compte au besoin les motifs du tissu.

Enfin le *chapitre V* représente l'aboutissement de l'étude. Il réutilise les opérateurs locaux à l'intérieur des algorithmes d'optimisation combinatoire que nous essaierons. L'approche que nous avons retenue tout au long de cette étude nous a amené à dissocier complètement la vue locale de la vue globale. Au niveau global d'optimisation, la position exacte des formes n'est pas connue, seuls l'ordre des formes et leurs configurations dans le placement sont maîtrisables. Le passage du niveau local au niveau global s'effectue par l'intermédiaire de paramètres dont nous chercherons les valeurs optimales vis-à-vis du critère de chute minimale. La représentation du problème combinatoire nécessite quelques attentions, car la dynamique du processus d'optimisation en dépend. Nous aborderons ensuite la mise en oeuvre de deux méthodes d'optimisation concurrentes: le recuit simulé que nous avons adapté au parcours d'arbre, ainsi que l'algorithme A ϵ emprunté à la recherche en graphe. Leur principe étant général, il est nécessaire d'y amener quelques aménagements pour que la résolution se fasse dans de bonnes conditions. Une comparaison permettra d'établir un choix qui n'était pas a priori facile.

CHAPITRE I

PRESENTATION D'UNE UNITE DE CONFECTION ET DU PROBLEME DE L'OPTIMISATION DE LA DECOUPE

I Objectifs d'optimisation dans le contexte économique de l'industrie textile

Dans l'industrie de l'habillement, le secteur de la coupe a longtemps résisté aux efforts d'organisation et d'automatisation. L'assimilation du placeur - coupeur à un artisan, voire à un artiste puisqu'il arriva souvent qu'il fût également le styliste dans les petites entreprises, a longtemps survécu. Bien que des machines informatisées sophistiquées ainsi que des logiciels performants aient pénétré le sanctuaire, on constate que nombreux sont les ateliers où l'artisanat cohabite plus ou moins bien avec des matériels de pointe. Et, même si des méthodes et des contrôles sont mis en place pour diminuer les coûts de production, on doit constater que le poids de l'atelier de coupe sur les coûts de l'entreprise n'est toujours pas maîtrisé.

Il n'est pas rare de constater que des surcoûts et même des dysfonctionnements, ayant pourtant une incidence non négligeable, continuent d'être générés à différents stades de la chaîne de production:

- l'encours inévitable entre la coupe et le piquage est rarement pris en compte
- les temps de coupe sont estimés globalement
- les pertes de tissu ne sont pas contrôlées rigoureusement.

"Il est temps de prendre en compte les problèmes spécifiques à la coupe, surtout dans le contexte nouveau de la diversification" [CON 91]. Diversification qu'il faut naturellement prendre au sens de la variété des problèmes rencontrés dans ces industries en pleine mutation technologique. Les notions tayloristes sont abandonnées au profit des chaînes de taille semi-industrielle introduisant ainsi les nouvelles données que sont la flexibilité et le "juste à temps". Ces données renforcent l'importance des processus d'optimisation au niveau du

lancement de la fabrication. La recherche d'une part de la diminution des coûts de main d'oeuvre et de matière première, d'autre part d'une qualité accrue et constante, ainsi que la gestion des encours entre les diverses unités fonctionnelles de fabrication, sont les maîtres mots d'une ère nouvelle qui se profile pour les entreprises en quête de rentabilité.

Avant de prolonger cette réflexion économique - technologique, il semble opportun d'introduire quelques notions introductives sur l'organisation d'une unité de production dans l'industrie de l'habillement. Sans doute seront elles utiles pour cerner au moyen de quelques exemples chiffrés l'importance de l'enjeu suite à la modification de quelques aspects de la fabrication, et pour justifier ainsi l'étude que l'on s'est fixée.

2 Présentation d'une unite de confection

L'objectif général de cette étude portant sur l'optimisation de la coupe des pièces de vêtement, il semble nécessaire de situer cette fonction au sein de l'unité de confection. Pour prouver la légitimité des données utilisées, des contraintes imposées ou des objectifs fixés dans la problématique de notre travail, il apparaît utile d'insérer la fonction de coupe entre les fonctions amont et avale qui lui sont directement liées.

La confection d'un vêtement est parcellisée en différents postes et phases de travail. Citons par ordre chronologique de leur intervention [DUB 88] [LAU 85]:

LA CREATION

- L'élaboration du croquis par le styliste
- La réalisation du patron de toile par le modéliste
- L'établissement du patron de base par la patronnière - modéliste
- Le prototypage par la mécanicienne - modèle

L'INDUSTRIALISATION

- L'acceptation en phase d'industrialisation
- La gradation
- Le placement

LA FABRICATION

- Le traçage

- Le matelassage
- La coupe
- Le triage des pièces découpées
- Le convoyage vers l'unité d'assemblage
- Les traitements préalables éventuels
- L'assemblage
- La finition

Précisons que, si certaines étapes apparaissent dans ce schéma traditionnel, elles peuvent être supprimées par des changements tactiques de la fabrication ou absorbées par le processus d'intégration propre à tout système de CFAO ou GPAO. C'est pour cela que dans la description suivante, nous identifierons les acteurs et non les fonctions, puisque que leur existence est implicitement liée à l'archétype d'une usine classique.

2.1 LA CREATION

Cette étape constitue la genèse du prototype que l'on soumettra à l'avis des décideurs de l'entreprise avant de lancer la phase d'industrialisation. Dans cette étape, les acteurs sont:

2.1.1 Le styliste :

Il conçoit les nouveaux modèles en produisant un catalogue de tendances pour la collection.

2.1.2 Le modéliste :

A partir des esquisses du styliste, le modéliste (souvent une femme) concrétise les idées du styliste. Il réalise un patron de "toile" en modelant la toile sur un mannequin, avant de tracer le patron-modèle, ou alors il effectue directement une coupe "à plat" (patron obtenu directement en deux dimensions). Ces opérations sont mises en oeuvre en tenant compte des réactions du tissu pour obtenir le meilleur aplomb possible.

2.1.3 La patronnière :

Elle réalise le patron de base en taille-standard en tenant compte des normes morphologiques de la population visée.

2.1.4 La mécanicienne - modèle :

Elle confectionne le vêtement prototype complet dans sa matière définitive et à l'unité, en tenant compte des indications fournies par le modéliste. Des retouches éventuelles sont ensuite apportées au niveau de :

- la position des pinces
- la forme des emmanchures
- l'ampleur du vêtement
- l'aplomb

Le patron-modèle définitif est alors obtenu en même temps que le prototype est présenté à la presse, au salon, à la clientèle en Janvier pour la collection d'été et en fin Juillet pour la collection d'hiver. Le cycle traditionnel de création peut être long (environ 6 mois), fastidieux et donc coûteux, et bien des modèles seront rejetés.

2.1.5 Evolution de la création

Les nouvelles contraintes introduites par le marché (variété plus importante des modèles, raccourcissement du cycle de création) ont imposé l'introduction de l'ordinateur accompagné des outils de C.A.O.

A ce niveau, deux procédés s'affrontent :

- a) La D.A.O - STYLE associée à la M.I.B (modification de l'image de base). Ces logiciels proposent la possibilité au styliste de réaliser un dessin digitalisé. Ce dessin (introduit à l'aide d'une table à digitaliser ou à partir d'une esquisse scannérisée) est la matière de base de la création. Ensuite, le dessin est modifié par des opérations géométriques, colorisé à l'aide d'une palette de coloris et de motifs en mémoire, laquelle a été chargée par des échantillons de tissu digitalisés par caméra ou par scanneur. La base de données s'enrichie au fur et à mesure des collections et le travail de création n'intervient plus que pour 10% dans la réalisation des éléments graphiques. Les modèles graphiques obtenus permettent de constituer un catalogue de collections transmis aux services commerciaux.
- b) La D.A.O TECHNIQUE : Elle est particulièrement adaptée au bureau des méthodes. Son but est de réaliser le dessin technique (*Figure 1*) du produit avec tous les renseignements nécessaires, pour qu'il soit bien interprété lors de la production. Ces informations servent au calcul des mensurations et de la

gradation, à l'établissement du processus opératoire, etc. Elle est, pour l'instant, la seule interface entre le modéliste et le processus aval d'industrialisation.

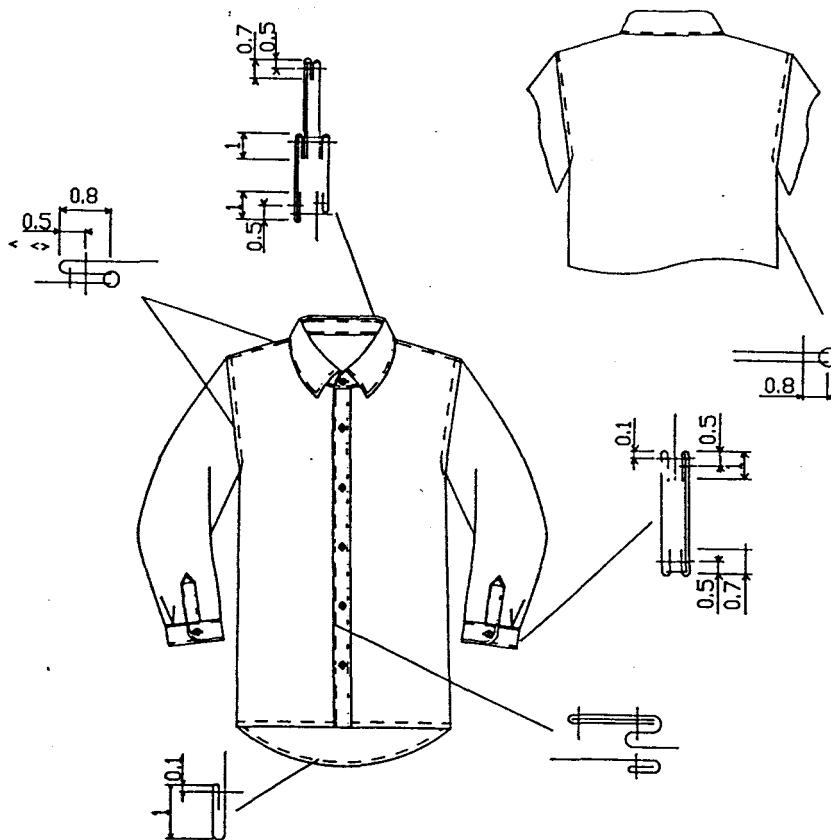


Figure 1 : exemple de dessin technique

Certes des "passerelles" logicielles sont à l'étude afin de concrétiser le processus d'intégration, par lequel l'intervention des acteurs intermédiaires sera de plus en plus rare et de plus en plus rapide. La prise en compte de la troisième dimension au niveau de la D.A.O. - STYLE sera à terme le passeport du dessin artistique pour entrer dans le monde du dessin technique, et permettra de passer directement du dessin sur écran au patron de base. conjointement aux efforts industriels effectués à ce niveau, des projets universitaires s'attellent à informatiser la phase de conception en y incluant, par exemple, des facultés d'apprentissage [CAO 92].

2.2 L'industrialisation

2.2.1 La patronnière gradeuse :

Sa fonction consiste à décliner les patrons des vêtements à fabriquer dans les différentes tailles (cf. Figure 2)

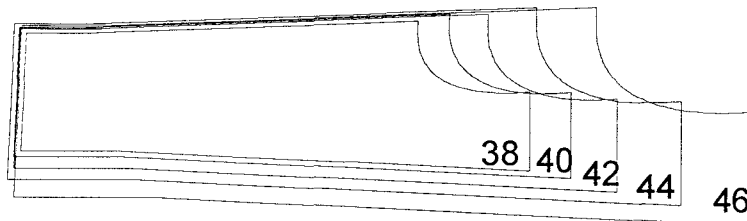


Figure 2 : Clé de tailles différentes obtenues par gradation

Il existe des règles de gradation, généralement empiriques car non-linéaires. Les différentes tailles portent sur trois mesures élémentaires (demi - tour de poitrine, demi - tour de ceinture, stature) qui sont exprimées en cm ou par une lettre. On définit à partir de ces données, pour les hommes adultes, sept catégories de conformations générales: athlétique, élancé, normal, fort, trapu, corpulent et ventru. Dans chacune de ces sept classes, on peut trouver une variété de cinq statures espacées chacune de 2 cm: petit, court, moyen, long et extra long : Ce qui fait 105 mesures pour 35 tailles. Les méthodes modernes de C.A.O. consistent à décrire la forme dans un repère en coordonnées cartésiennes ou polaires, et à appliquer aux points singuliers la formule de gradation. Cette fonction est très souvent intégrée au système de D.A.O - C.A.O. utilisé en création.

Lors de cette étape on inclut au niveau des patrons (cf. Figure 3):

- les **crans de montage** facilitant l'assemblage
- les "**valeurs - couture**" correspondant à une marge de contour permettant également un montage et un piquage plus aisé

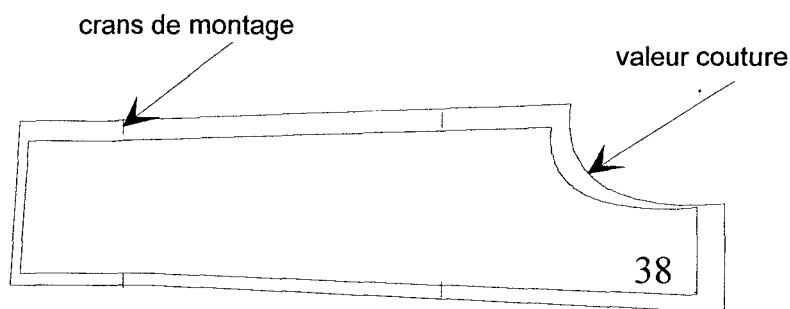


Figure 3: Exemple de patron

2.2.2 Le placeur - traceur

Son rôle consiste, par rapport à une laize de tissu, à étudier le placement des patrons, matérialisant les différentes parties d'un vêtement avec ou sans mélange de plusieurs tailles, pour minimiser la perte de matière. Pendant cette étude, les fonctions de coupe et de montage doivent être considérées pour que celles-ci se déroulent dans de bonnes conditions. Le placement doit en particulier respecter certaines contraintes (sens du droit fil, nécessité ou non d'un minimum d'espace entre chaque pièce pour permettre la coupe). Chaque placement aboutit à un taux plus ou moins élevé d'utilisation de la surface totale (75 à 90% environ), et demande un temps plus ou moins important (de quelques minutes à quelques heures). L'efficacité dépend du nombre de patrons, de la forme des gabarits mais aussi des compétences du placeur, l'expérience et l'intuition jouant un rôle prépondérant.

2.2.3 Les placements assistés par ordinateur

L'ordinateur a fait son apparition au niveau du placement depuis longtemps, car sa puissance de calcul en fait un outil de choix pour cette tâche particulièrement combinatoire et fastidieuse. L'évolution des systèmes automatisés de placement est progressive, et l'optimisation du placement fait encore actuellement l'objet de recherches au niveau industriel et universitaire. Voici les fonctionnalités des principaux systèmes actuels.

a) Le placement interactif (cf. Figure 4)

Dans la mémoire du calculateur, sont stockées les images des parties de vêtement pour chacune des tailles des différents modèles de la collection. Le placeur a des informations sur les approvisionnements disponibles. Si les pièces de tissu ont été systématiquement inspectées et mesurées par un analyseur de laize, les mesures précises sont alors introduites. Elles deviennent l'une des contraintes du placement. Le placeur connaît le programme de

d'un algorithme systématique. Certaines heuristiques ne prenant pas en compte les cas particuliers, on confie alors cette tâche à un placeur-expert, dont l'intelligence et l'expérience permettent de détecter rapidement les gains possibles, grâce à un niveau d'abstraction qui est propre à l'être humain.

b.2) Le système fournit un placement définitif en un temps plus ou moins long, et avec une efficacité plus ou moins bonne. Ensuite, le résultat est stocké en mémoire, pour être exploité ultérieurement lors du lancement de la fabrication. Au cours d'une journée, on multiplie le nombre de placements différents, afin d'enrichir la bibliothèque, et d'améliorer les chances de fournir un bon placement pour une commande donnée.

b.3) Le niveau le plus intégré en l'état actuel consiste à fournir un placement en temps réel pour une commande donnée. Le résultat est enregistré sur un fichier, qui est transmis instantanément à la table de découpe, et permet de commander directement l'outil de coupe. Ce type de fonctionnement convient essentiellement à l'organisation d'un système de production à cycle court, dans lequel on crée, on enregistre les commandes en permanence et on fabrique en séries très courtes. Tous les problèmes liés à cette structure ne sont pas résolus (qualité des placements, vitesse d'obtention, prise en compte des motifs et des défauts). Cependant, quelques systèmes sont déjà commercialisés et introduits dans certains ateliers de coupe, dont la configuration est en adéquation avec les performances proposées.

2.3 La fabrication

Les étapes suivantes marquent la phase de fabrication. Celle - ci doit être programmée en fonction de la capacité des organes de production, en particulier au niveau du piquage qui est plus lent. C'est le rôle de l'ordonnancement. Chaque quinzaine, un programme de fabrication (*cf. Figure 5*) est lancé fixant les charges de travail. Ce document rassemble les commandes homogènes en fonction du modèle, de la matière et du coloris.

Exemple :

O.F.	Modèles ou N°	Matière	Coloris ou N°	38	40	42	44	46	48	Total	O.C.
1	Enok	Tergal	Noir	20	20	20	20	20	20	120	1
2	Enok	Tergal	Bleu	10	10	10	10	10	10	60	1
3	Enok	Tergal	Blanc	30	30	30	30	30	30	180	1
4	Julius	Laine	Brun	10	10	20	20	33	25	118	2
5	Julius	Laine	Gris	15	18	35	50	70	50	238	2
6	Adrien	Laine	Brun	8	8	12	15	20	30	97	3
7	Adrien	Laine	Vert	10	10	12	15	20	30	97	3
8	Marco	Lin	Beige	5	5	8	12	15	20	65	4
9	Marco	Lin	Blanc	5	5	6	12	16	22	66	4

Figure 5 : Programme de fabrication

Pour ce pré-lancement de deuxième quinzaine du mois, à chaque ligne d'O.C. (Ordre de coupe) correspondent plusieurs O.F. (ordres de fabrication)

Chaque **ordre de fabrication** précise le nombre de modèles pour un coloris donné, et ce pour les différentes tailles.

Chaque **ordre de coupe** groupe les ordres de fabrication par modèles identiques pour homogénéiser les opérations d'assemblage et de finition.

Un ensemble de documents est constitué pour fixer les consignes de production:

- L'ordre de coupe
- les fiches de matelas décrivant les piles de plis de tissu à réaliser pour gagner du temps lors de la coupe
- Les placements associés à chaque pile de plis ou leurs références pour le traçage ou la découpe assistée
- Les fiches de consignes de montage

Ces documents sont communiqués aux différents maillons de la chaîne de fabrication.

2.3.1 Le traceur

Dans les configurations techniques les plus avancées, le traçage n'existe plus. En coupe automatique, l'outil de coupe suit un tracé non matérialisé, qui correspond au placement mémorisé dans le calculateur. Quand la coupe reste traditionnelle, le placement mémorisé est reproduit sur une feuille de papier par un traceur à grande vitesse. La feuille est ensuite posée sur le tissu à découper, et indique au coupeur les contours qu'il doit suivre. Mais avant de couper le tissu, on superpose plusieurs plis: c'est le matelassage.

2.3.2 Le matelasseur

L'opération de matelassage est effectuée dans un but de gain de temps. Ainsi, avec un seul passage de l'outil de coupe, on multiplie la coupe d'un vêtement en fonction des tailles et des coloris demandés par l'ordre de fabrication. Pour cela, le matelasseur dispose de documents précis : les fiches de matelas. Celles-ci lui indiquent:

- le numéro d'identification dans le tableau de coupe
- Le placement à utiliser
- La longueur du matelas (de 1 à 10 mètres avec une précision de 1 mm)
- Le nombre de plis par coloris ou par matière (typiquement 50 plis, hauteur <20 cm)
- Le schéma du matelas (hauteur constante ou en escalier) (*cf. Figure 6*)

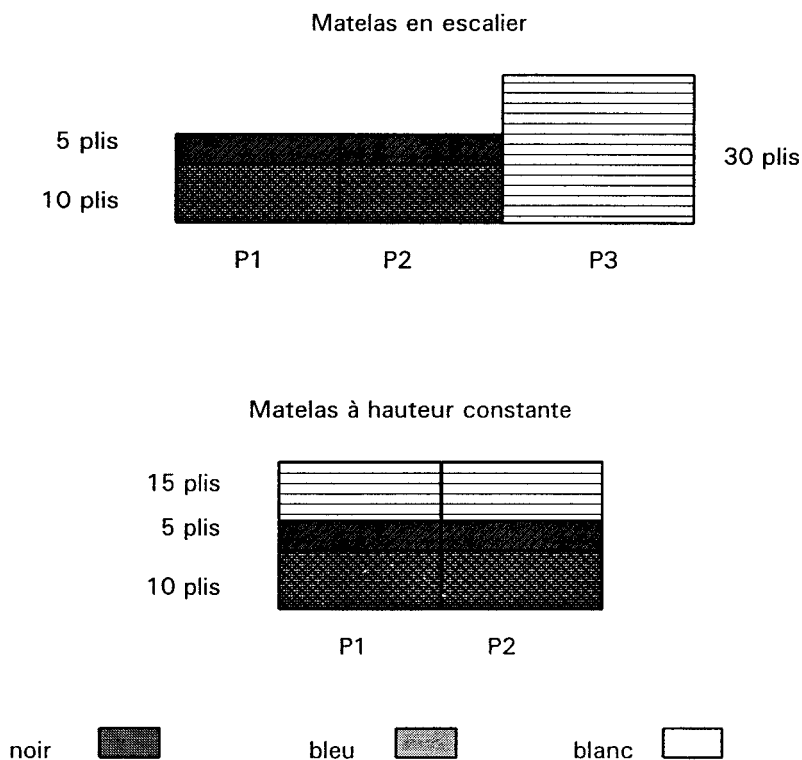


Figure 6 : exemple de matelas

La préparation du schéma de matelassage constitue également un problème d'optimisation. A partir du bordereau de coupe (cf. Figure 7) et de la bibliothèque de placements (cf. Figure 8) réalisés à l'aide de combinaisons de tailles différentes, l'objectif consiste d'une part à déterminer l'ensemble des plis à former, et les placements à choisir pour produire exactement le bordereau de commande (si l'entreprise ne travaille pas pour le stock) et d'autre part à minimiser la fonction de coût. Cette fonction tient compte évidemment du prix de la matière, mais peut aussi faire intervenir le coût des temps opératoires pour la réalisation des matelas, de la coupe et de l'éclatement des paquets. Une formulation de ce problème a été proposée par [ELO 92]. La fonction de but tient compte des différents paramètres cités précédemment, et est linéarisée de façon à ce que le problème puisse être résolu avec des méthodes d'optimisation linéaire.

O.F.	Modèles ou N°	Matière	Coloris ou N°	Tailles						Total	O.C.
				38	40	42	44	46	48		
1	Enok	Tergal	Noir	20	20	20	20	20	20	120	1
2	Enok	Tergal	Bleu	10	10	10	10	10	10	60	1
3	Enok	Tergal	Blanc	30	30	30	30	30	30	180	1

Figure 7 : Bordereau de coupe

Placements	38	40	42	44	46	48
P1	2	2	2	0	0	0
P2	0	0	0	2	2	2
P3	1	1	1	1	1	1

Figure 8 : Bibliothèque de placements

Les manoeuvres de réalisation des piles de plis s'effectuent rapidement avec un chariot matelasseur - plieur, permettant un alignement plus aisé et une vitesse d'étalement plus élevée. Il lui incombe également de bien superposer les tissus à motifs. Il dispose souvent d'une table à aspiration pour immobiliser le matelas.

A l'heure actuelle, le chariot automatique programmable effectue toutes ces opérations de façon quasi - autonome. Il permet d'augmenter la vitesse de matelassage (100 m/min), tout en réduisant le temps des opérations annexes. Ces machines sont généralement associées à un chargeur de pièces, qui peut comporter jusqu'à 28 rouleaux. L'opération de visitage se fait automatiquement à l'aide d'un système de traitement d'images. Elle permet en même temps de connaître la longueur exacte de l'étoffe, la laize, la tension optimale d'enroulement (important pour que la périodicité spatiale des motifs soit constante). Elle participe aussi à l'alignement des lisières (pour faciliter le déroulement).

2.3.3 Le coupeur

Il procède à la découpe du matelas par contournage des parties de vêtement tracées sur le patron de placement. Les couteaux mobiles à lame verticale circulaire ou à ruban dont il se sert, ont été améliorés au fil des années:

- Augmentation de la force de pénétration dans les matelas
- Elévation de la résistance de l'outil de coupe (affûtage intégré)
- Ajout d'un système d'aspiration du tissu vers la table, supprimant ainsi les semelles d'appui et donc l'effort latéral de guidage de l'outil.

Le progrès le plus important a été l'arrivée des nouvelles technologies de coupe (laser, jet d'eau). Ces techniques allient plusieurs avantages:

- Vitesse de coupe (de 5 à 20 m/min selon l'épaisseur du matelas)
- Précision (permettant de resserrer le placement)
- Netteté de la découpe
- Pas d'efforts latéraux de coupe
- Faible usure
- Découpe des formes complexes très facile
- Procédé facilement automatisable

En revanche, ils amènent à tenir compte de quelques inconvénients qui sont:

- Problèmes de sécurités liés à la réflexion du faisceau laser
- Réglages pointus de la puissance pour éviter les dégradations de la table de découpe et du matériau découpé (fusion et soudure de la fibre synthétique, érosion de la table de découpe).

Associés à un calculateur et fixés sur un organe de commande semblable à une table traçante, ces organes de coupe rendent la procédure de découpe quasi - autonome. L'outil de coupe est guidé par les données fournies par le fichier du placement sélectionné. Certaines de ces tables sont même équipées d'une caméra associée à l'outil de coupe permettant un recalage précis des points de raccord des éléments concernés, lorsqu'il s'agit d'un tissu à motifs [LOR 90]. Au moment du placement, une marge correspondant à un demi - motif est laissée autour

des formes à raccorder. Puis, lors de la découpe, le faisceau se positionne sur le point de raccord de la forme à découper. Un recalage interactif est effectué en déplaçant à distance l'outil de coupe de façon à assurer le raccord de la zone à découper avec le motif de la pièce venant en contact au moment de l'assemblage.

2.3.4 Le triage

Avant de parvenir au montage, les matelas découpés doivent subir plusieurs opérations. A ce stade, aucun système ne permet encore actuellement de trier les pièces des différents vêtements constituant le matelas, et de les envoyer automatiquement vers les installations de piquage. Ces opérations consistent :

- à éclater les matelas en **bûches** (morceaux de vêtements découpés en plusieurs plis)
- à constituer des **paquets** à partir des bûches, relatifs à un même vêtement
- à éclater les paquets en **couleurs de fil** différentes
- à insérer les **fiches de consigne**

Ensuite les pièces de vêtements sont transportées vers les ateliers de **piquage** et de **montage** au moyen d'un **convoyeur**. Là, elles peuvent subir un **traitement préalable** qui consiste à préparer les différentes pièces avant le montage (surfilage ou traitement thermique pour éviter l'effilochage des bords, plis, fronces, broderies, smocks, festons, ...). Elles sont ensuite assemblées par piquage avant de passer en **finition** (boutons, boutonnières, ourlets), **repassage** et **emballage**.

De gros efforts de recherche et de développement sont engagés pour accentuer l'automatisation de l'aval de la coupe, c'est-à-dire le triage, le convoyage, les opérations préalables à l'assemblage, le piquage et la finition. Cette automatisation est d'ailleurs rendue difficile en raison de la nature souple du tissu. De cette progression de l'intégration, dépendent la vitesse de production et la taille des encours de pièces coupées. Tant qu'il existera une différence de rythme de production entre la coupe et le reste de la chaîne, les stocks-tampons de pièces ne pourront être contrôlés que par la modularité de la chaîne de découpe et la flexibilité.

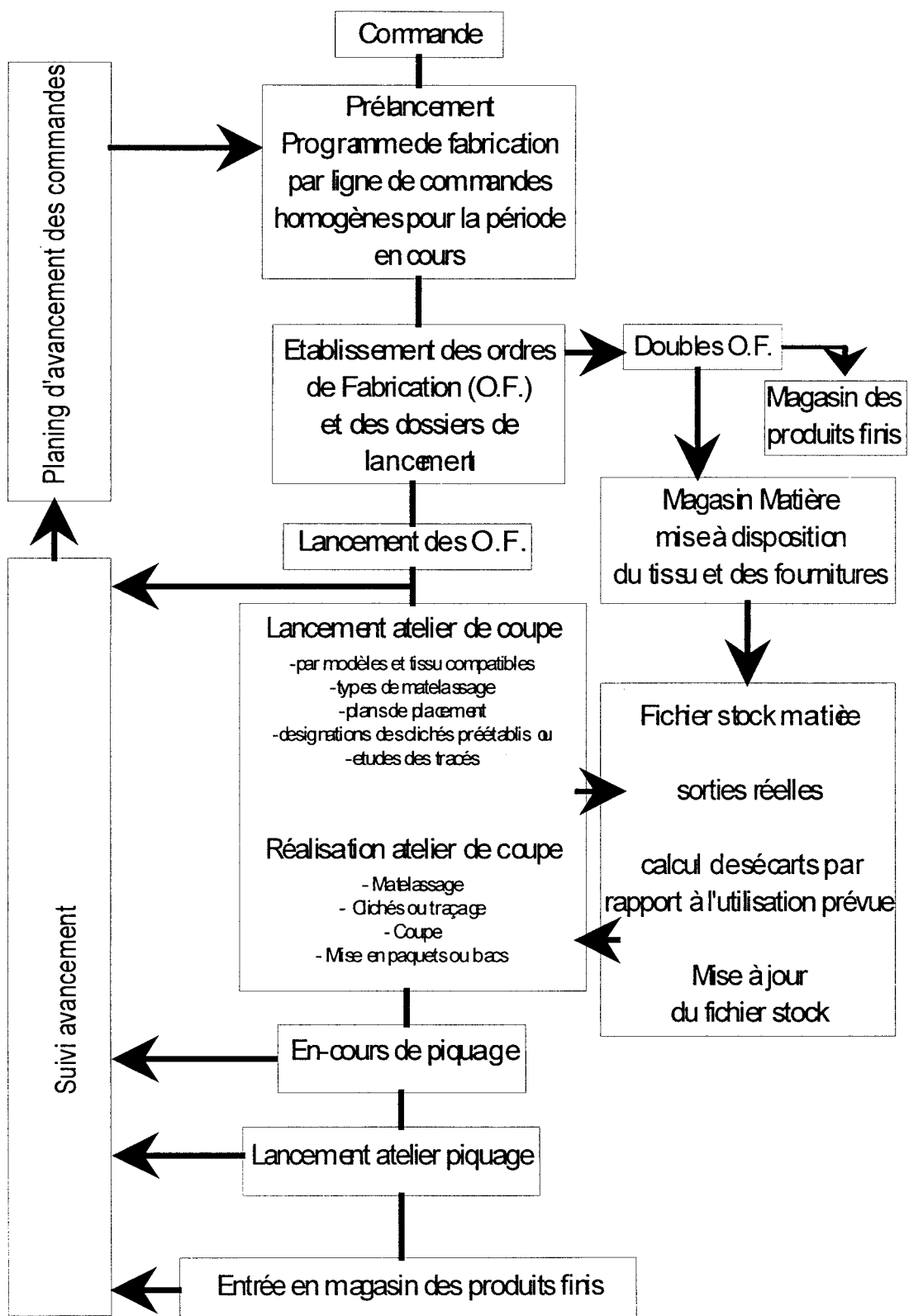


Figure 9 : Lancement et suivi de fabrication [LAU 85]

3 L'enjeu de l'optimisation de la découpe

Aujourd'hui, les problèmes spécifiques de la coupe relèvent de deux ordres :

- Economique : quand on prend en compte les coûts des encours, de la non - qualité, de l'emploi-matière et de la main d'oeuvre
- Organisationnel : quand on réfléchit au fonctionnement de l'atelier de coupe dans un contexte de "juste à temps", et que l'on considère les nouveaux critères d'optimisation exigeant un équilibre optimal entre le degré de flexibilité, le niveau de qualité, le volume de déchets acceptable et le coût main d'oeuvre.

L'insuffisance d'autofinancement des sociétés textiles implique un recours à des emprunts sur lesquels pèse des intérêts financiers non négligeables. ces intérêts sont quasiment proportionnels à la longueur du cycle à financer, c'est-à-dire essentiellement le cycle de fabrication. Celui - ci peut se décomposer par :

- Une attente de lancement de 15 jours à 1 mois (stockage des commandes pour regrouper les quantités)
- des encours de coupe pendant 2 à 5 jours
- des encours de piquage pendant 2 à 5 jours (stockage des pièces coupées afin d'obtenir des quantités qui permettent de réaliser un équilibrage convenable).

Dans le but d'optimiser les lancements, on est amené à traiter avec des délais différents des parties de commandes appartenant à la même commande. Ceci peut générer des livraisons morcelées ou un stock supplémentaire au niveau des produits finis, ce qui implique une attente variable selon les entreprises, et donc des coûts plus importants.

Autre coût, celui de la non - qualité, constaté au niveau de la coupe, par le fait que des anomalies matière y sont principalement traitées par élimination des métrages défectueux, ou par le remplacement des pièces coupées.

Dans beaucoup d'entreprises les pertes de tissu ne sont pas contrôlées rigoureusement. Elles représentent pourtant parfois 25% du coût des matières premières, et une réduction de 2% sur l'emploi tissu ¹ peut conduire à une augmentation de 10% du bénéfice de l'entreprise [CON 91]. D'autres estimations montrent que dans une industrie où 56% du

¹Emploi tissu: longueur de tissu nécessaire pour réaliser un vêtement.

coût d'un vêtement fini correspond au coût du tissu, alors une diminution de 2% des pertes de matière peut augmenter de 40% le bénéfice réalisé par vêtement vendu ! [BOB 89]

Enfin, les opérations de coupe, sont généralement ignorées par le service des méthodes de l'entreprise, ceci pour de multiples raisons : leur diversité, l'absence de cycles homogènes, le faible nombre de personnes concernées en comparaison avec le piquage. En conséquence, l'activité dans une salle de coupe dépasse rarement 60%.

Une simulation intéressante (*Figure 10*) permet de prendre en compte les économies réalisables. En prenant l'exemple d'une entreprise dont le budget de fonctionnement de l'atelier de coupe est de 100%, on peut évaluer le poids des différents coûts et les potentiels d'économies :

	<i>Poids actuels</i>	<i>Potentiel d'économie</i>	<i>Objectifs</i>
<i>Non - flexibilité</i>	14%	50%	7%
<i>Non - qualité</i>	17%	50%	8.5%
<i>Déchets matière</i>	55%	27%	14.9%
<i>Coût main d'oeuvre</i>	14%	40%	5.6%
<i>Total</i>	100%		36%

Figure 10

Ces quelques chiffres montrent aisément que l'automatisation permettrait d'atteindre le même résultat de production, avec un budget de fonctionnement de 36%.

Si la technique du matelassage permet de tendre vers un résultat semblable, lorsque les séries sont fortes, et que le délai de livraison est souple, en revanche, elle ne satisfait pas tous les critères économiques dès que les commandes deviennent ponctuelles, urgentes ou faibles.

Or naturellement, cette politique devient de plus en plus fréquente dans un contexte où les clients sont eux-même confrontés à la nécessité de travailler en flux tendu.

4 Présentation d'une cellule de découpe flexible

Une alternative à la technique du matelassage consisterait à utiliser un système de découpe en continu sur pli unique. Une telle structure a été imaginée dans le cadre d'un projet

portant sur un atelier expérimental de découpe de vêtement. L'étude des problèmes théoriques soulevés par cette structure est en cours au Centre d'Automatique de Lille 1 [MAO 85] [MAO 92]. En voici une description (*Figure 11*):

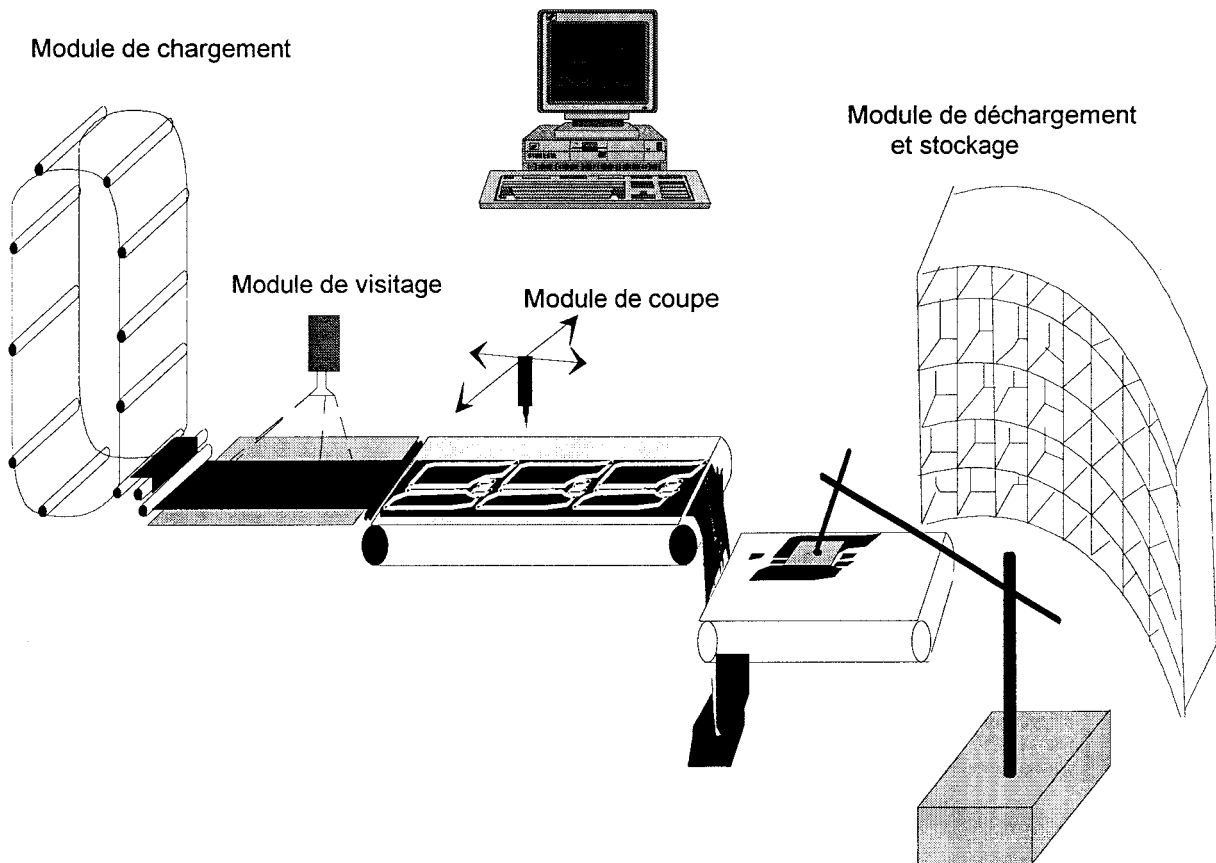


Figure 11 : Cellule flexible de découpe

Cette structure fait apparaître cinq modules:

4.1 Le chargement de la pièce de tissu

Celui-ci s'effectue à l'aide d'un carrousel, structure métallique sur laquelle est disposé un ensemble de rouleaux de tissu. Automatisé, cet appareillage permet de supporter les rouleaux de tissu, et rend possible le déchargement pour le débobinage ainsi que le rembobinage. Le choix de la pièce de tissu à engager sur la table de découpe est fait par un processus de gestion des ordres de fabrication.

4.2 Le module de visitage

Associé à un système de traitement d'image, il est capable de réaliser l'acquisition de la laize, de reconnaître le motif [TAN 87], [DOD 91], d'en déduire les caractéristiques associées (périodicité, forme), de déterminer la position des motifs placés ainsi que la forme et la position des défauts. Le module est également capable de modéliser la forme du début de rouleau, et de comptabiliser la longueur de la matière débitée. La largeur maximale admise est de 1 m 50. La précision souhaitée des informations géométriques est de 1 mm. Les motifs détectés sont référencés dans une bibliothèque, dont l'accès permet d'obtenir des informations complémentaires.

4.3 Le module de placement

C'est l'objet de la présente étude. Utilisant les informations issues d'une bibliothèque de formes dont les références sont fournies par l'ordre de coupe, l'algorithme doit être en mesure de fournir le placement d'un sous-ensemble de ces formes, lesquelles seront découpées dans la séquence suivante. Il cherche à optimiser une fonction de but visant à minimiser les pertes de matière, tout en respectant un temps de calcul acceptable par rapport au rythme des opérations connexes. Il doit à terme prendre en compte les raccords des pièces qui seront assemblées, lorsque le tissu employé est à motifs périodiques ou placés. Les défauts devraient également être contournés, afin d'éliminer les pertes liées à la non - qualité des pièces débitées.

C'est à ce niveau du programme que le dialogue homme/machine intervient, puisque les données relatives au carnet de commande sont introduites selon le schéma suivant :

- style demandé
- modèle à produire
- quantité dans chaque taille
- référence du tissu
- priorité de lancement (normal, urgent, immédiat,..)

Les patrons des vêtements font partie de la bibliothèque de données, dont les différentes tailles sont générées par gradation automatique.

Le lancement de la fabrication s'effectue à partir d'un algorithme de décision. Cet algorithme classe les commandes par tissus identiques (pour éviter les changements fréquents de rouleaux de tissu), par modèles semblables (pour homogénéiser l'assemblage) et par ordre

de priorité. On conçoit pour des raisons évidentes de gestion du temps, que les ordres de coupe propres à une matière soient effectués jusqu'à leur terme. Ceci pour éviter les pertes de temps importantes liées à la manoeuvre de changement de rouleau. Cependant, le caractère immédiat d'une commande perturbe volontairement le cycle en cours en se substituant à l'ordre de fabrication courant.

4.4 Le module de découpe

Il pilote l'outil de coupe (laser, jet d'eau par exemple) par contournage. Les informations qu'il utilise, sont issues du module de placement. Ce module a accès à toutes les données que le module de placement utilise ou enregistre, en particulier les coordonnées de tous les points $P_i (x_i, y_i)$ du contour de chaque forme placée.

L'outil peut se déplacer à l'intérieur d'une surface de travail de dimension $L_o \times W_o$. Le plan d'imbrication constitué d'un sous ensemble des pièces du carnet de commandes a une dimension $L_p \times W_p$ tel que (*Figure 12*) :

$$L_p \geq L_o$$

$$W_p \leq W_o$$

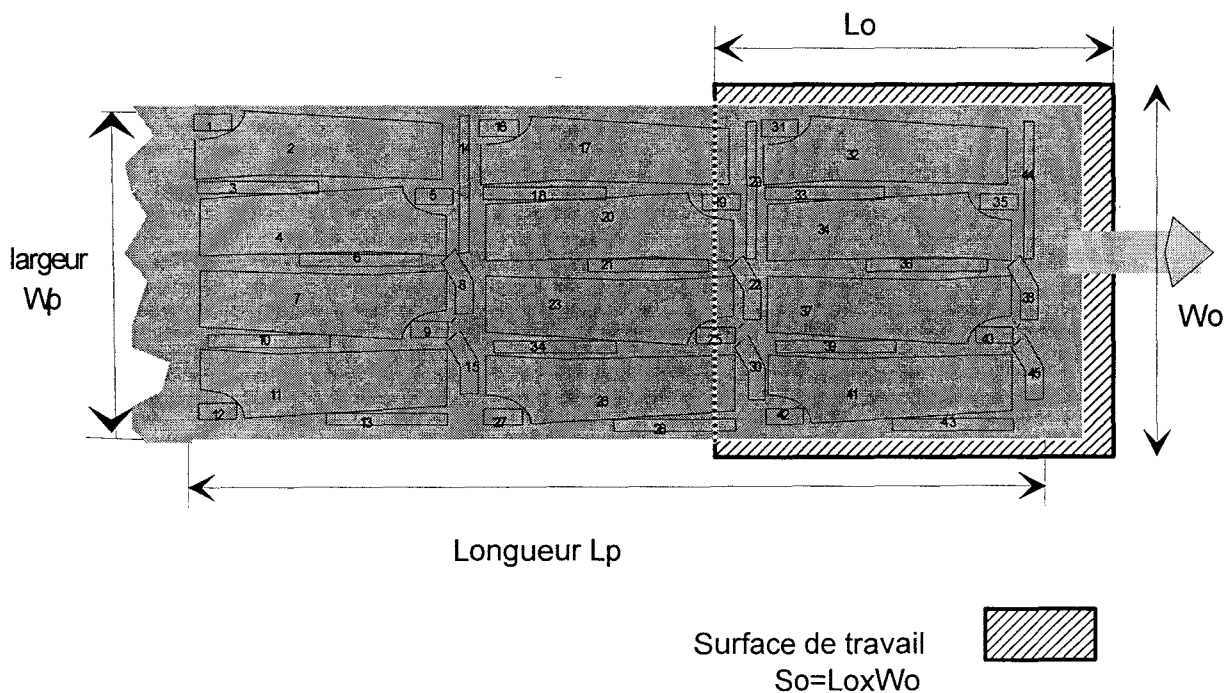


Figure 12: Plan de placement et surface de travail

Le plan de placement P est donc discrétisé en bande de longueur " T_e " fixée, et le tissu avance d'un nombre variable de pas entiers de longueur " T_e ", de sorte que le processus de découpe soit optimisé en temps (*Figure 13*). En effet, le choix d'avancer par pas successifs engendre un morcellement des pièces et une décomposition en contours ouverts de celles - ci (*Figure 14*). Chaque portion de contour doit être découpée et nécessite une commande dynamique de l'outil, telle que l'on ait une phase d'accélération depuis la vitesse nulle, une phase à vitesse constante et une phase de décélération jusqu'à l'arrêt de l'outil. Relativement à la distance parcourue, ce cycle de déplacement nécessite un temps de parcours plus important que si la découpe s'effectuait à vitesse constante, comme cela est possible pour un contour fermé. La gestion de l'avance consiste donc à décider du nombre de pas d'avance du tissu, de façon à ce que les trajectoires à vitesse constante soient les plus longues possible pour que le temps de production soit minimum. Naturellement, ce problème est associé à la gestion de la trajectoire de l'outil de coupe. Celui - ci est guidé par une commande d'axes sur un plan X,Y et doit suivre une trajectoire composée de contours fermés, de portions de contours ouverts ou de portions rectilignes sans découpe. Le problème revient alors à la recherche d'un chemin optimal, dit "problème du voyageur de commerce" [OKA 91].

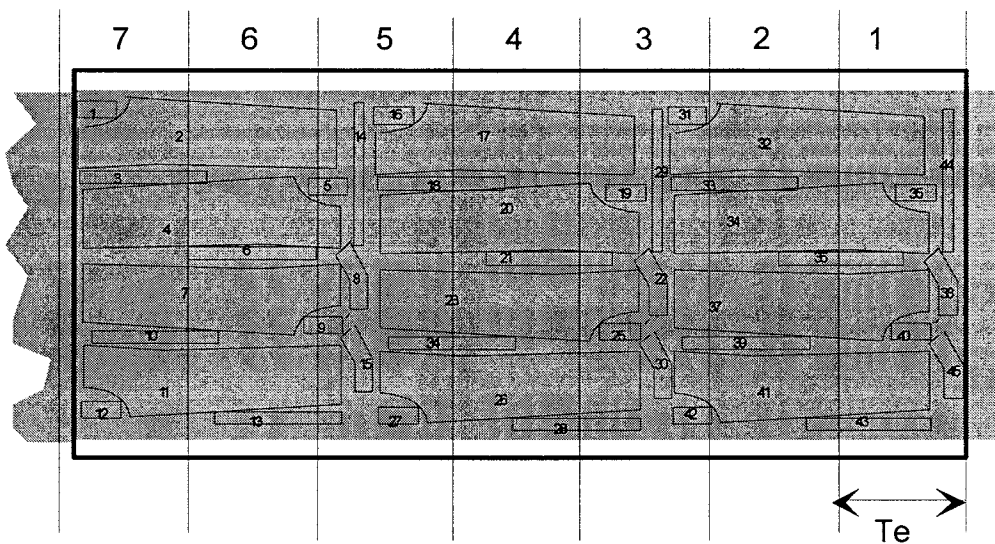


Figure 13 : Discretisation du plan de placement

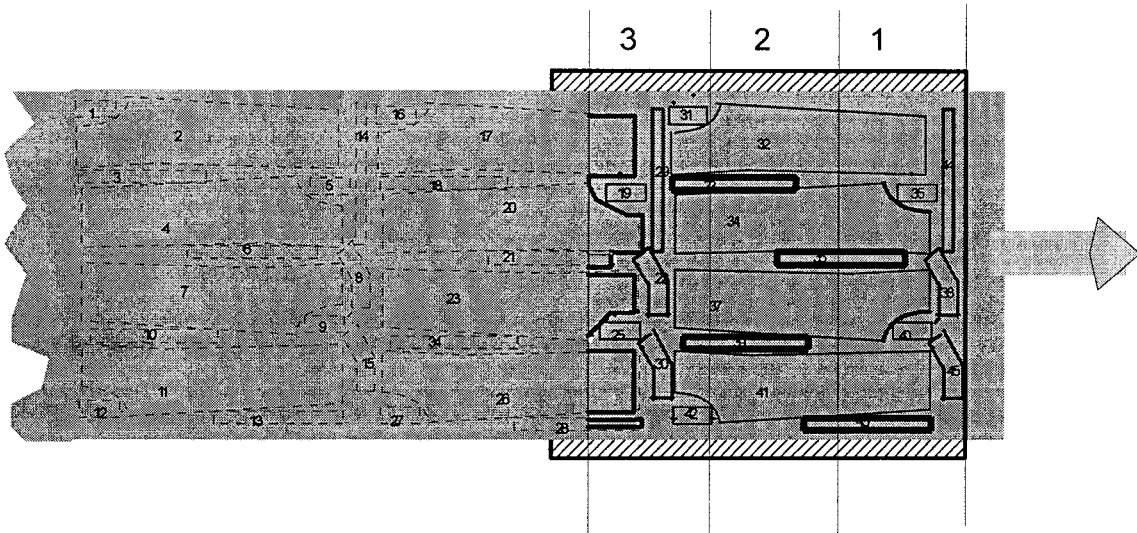


Figure 14 : Exemple de découpe pour une décision de 3 pas

4.5 Le module de déchargement et de stockage

Après la coupe, les objets complètement découpés doivent être évacués du tapis roulant sur lequel ils reposent, puis triés et stockés dans le magasin d'encours à l'assemblage. Un robot manipulateur doit intervenir de façon autonome pour cette tâche de déchargement. Son travail est rendu difficile par la nature souple du tissu, et la technique utilisée pour le préhenseur est capitale vis-à-vis de l'efficacité de la manipulation. Le travail ne peut s'effectuer dans de bonnes conditions que si le tapis roulant se trouve à l'arrêt. Les données nécessaires à ce module sont : la position par rapport au nouveau repère lié au robot, la forme des pièces découpées, ainsi que leurs références. Elles sont fournies au fur et à mesure de la coupe complète d'une pièce.

4.6 Coopération des différents processus de l'unité de découpe

Les différentes tâches participant au cycle de production de la cellule ne sont pas classiquement séquentielles mais peuvent évoluer en parallèle. Chacune d'elles peut être vue comme un processus indépendant. Chaque processus travaille avec son propre rythme, et doit coopérer avec les processus connexes avec qui il partage des ressources communes. La Figure 15 illustre la coopération entre les différents processus.

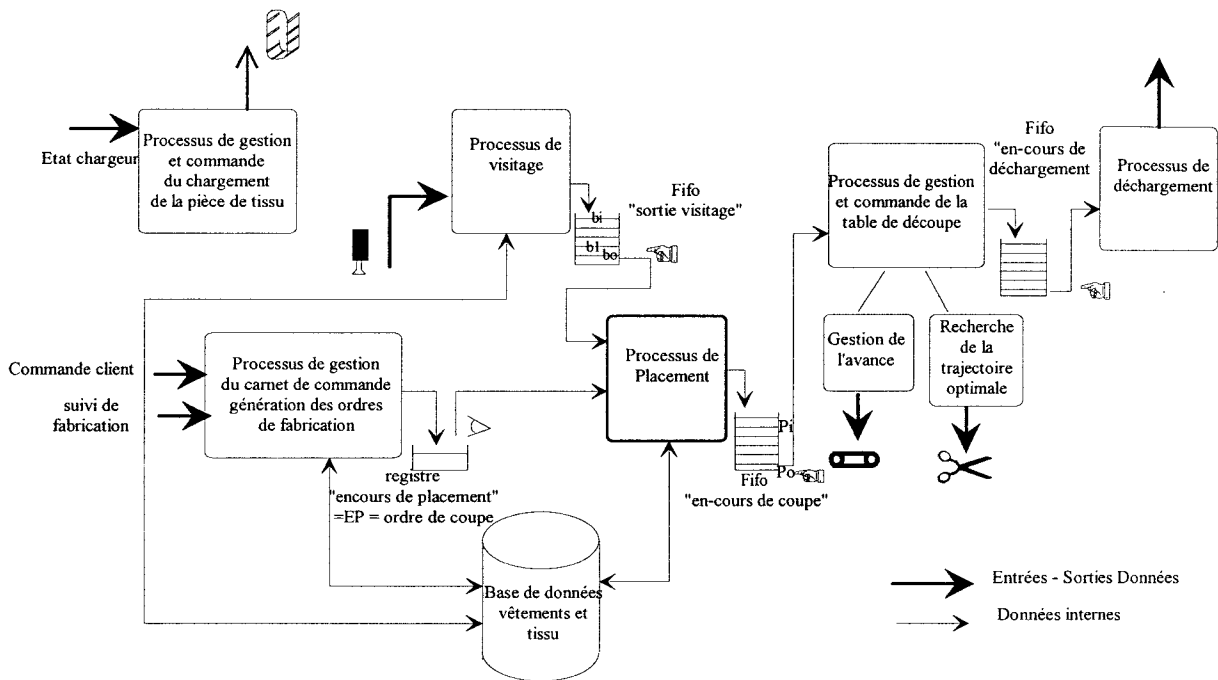


Figure 15

Le diagramme fait apparaître:

- Les rectangles symbolisant les processus (chargement de la pièce de tissu, visite, placement, découpe, déchargement des pièces de vêtements)
- Les liaisons orientées correspondant au chemin suivi par les données transmises entre les processus
- les tampons de données nécessaires en raison des différences de rythme entre chaque processus
- Le cylindre symbolisant la base de données générale du système.

Contrairement à la technique du matelassage où chaque opération s'effectue séquentiellement, ici les temps technologiques propres à chaque tâche se chevauchent. Le système travaille en temps masqué. Bien sûr, les tâches évoluent à des vitesses différentes, et ceci impose que les données en attente soient stockées transitoirement dans des structures de type FIFO (First in - First out).

Le registre "encours de placement" ou EP a son accès qui est partagé entre le processus de gestion du carnet de commande et le processus de placement. Son contenu est dynamique vis à vis des deux processus :

- Les ordres de coupe sont ajoutés en fonction de la nature de la commande, du délai de livraison et du suivi de fabrication
- Les pièces placées dans la dernière bande sont soustraites du registre
- Le contenu peut être remplacé par un ordre de fabrication à traiter immédiatement

Un encours de placement sera caractérisé par une liste d'effectifs $A(a_0, a_1, \dots, a_N)$ ou chaque élément a_k représente l'effectif qu'il faut produire de la forme F_k

L'utilisation du registre de l'encours de placement permet ici de prendre en compte instantanément une évolution de l'ordre de fabrication.

Concernant le processus de placement, le schéma suivant met en évidence, au moyen d'un GRAFCET, le cycle de fonctionnement de ce module (*cf. Figure 16*). Il précise le séquençement des opérations d'échange des données avec les processus voisins.

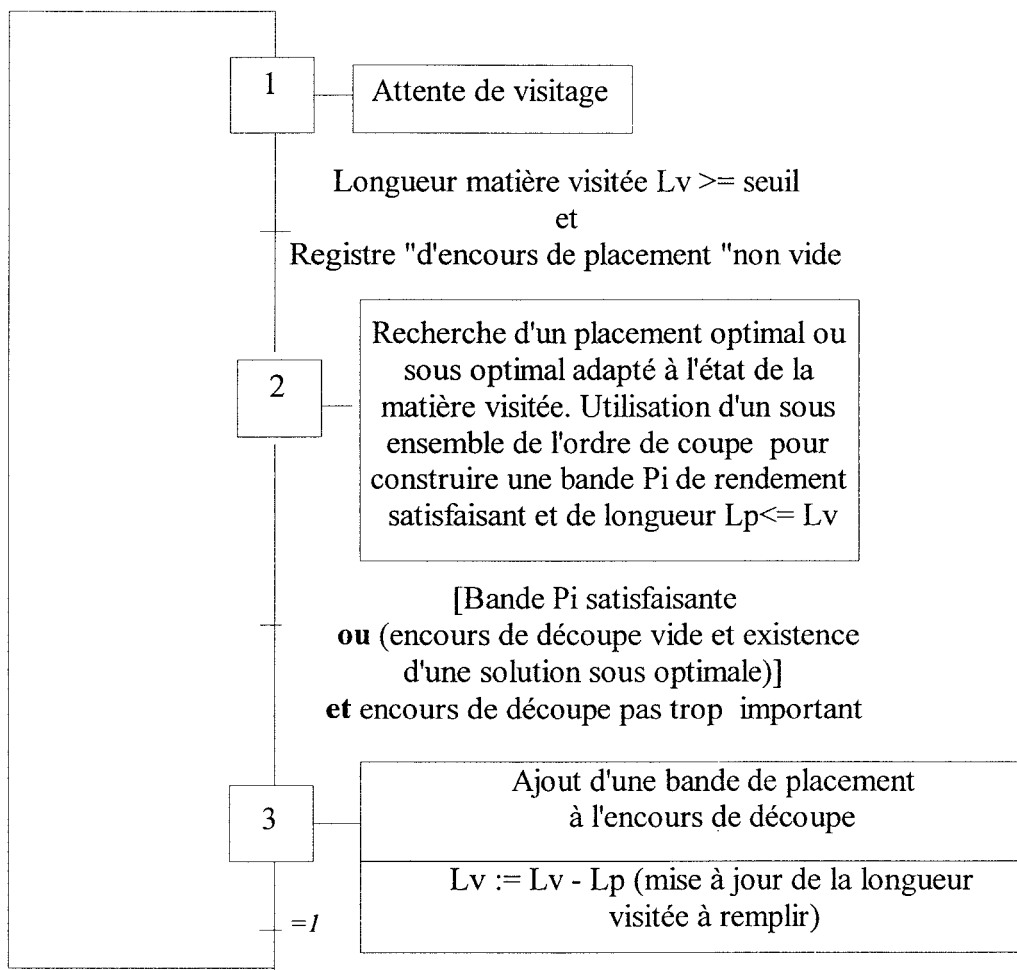


Figure: 16 Grafcet associé au processus de placement

Le cycle associé au placement comporte trois étapes essentielles:

Étape 1: Il s'agit d'une synchronisation avec le module de visitage, si ce dernier est momentanément retardé par rapport au processus de placement, et que la liste "Sortie visitage" est vide ou insuffisamment remplie.

Transition 1 : si la liste des bandes de matière visitée est suffisamment remplie, et que le registre d'encours de placement contient des formes à placer, alors cette transition est franchissable.

Étape 2 : l'algorithme de placement choisit les formes à placer dans le registre "encours de placement" pour générer une bande P_i de placement, satisfaisant un seuil de rendement matière. Cette bande a une longueur nécessairement inférieure à la longueur de matière visitée.

Transition 2 : le franchissement n'est possible que si:

Le placement dans la bande courante a atteint un seuil de rendement convenable

ou

que Le système de placement possède une solution de qualité inférieure au seuil de rendement mais qui ne peut être améliorée en raison de la nécessité de fournir une solution au module de découpe en attente d'activité (encours de coupe vide).

Dans tous les cas, il ne sera pas possible de sortir la solution du processus de placement si la pile "encours de coupe" est saturée. Ce temps pourra être utilement consacré à améliorer le rendement de la bande courante.

Etape 3 : L'activation de cette étape consiste à ajouter une bande placée à la pile "encours de coupe" et à réactualiser la longueur de matière visitée.

5 CONCLUSION

Dans ce chapitre, on a essayé de présenter l'environnement de notre étude. Pour cela, nous avons entrouvert les portes d'un atelier de confection pour y percevoir les différentes fonctions de production participant à l'élaboration d'un vêtement. Parmi ces fonctions, il y en a une qui nous intéresse particulièrement : le placement. Les nouvelles contraintes de production imposent aux industries de la confection d'améliorer les performances de cette opération afin de réduire le coût de matière première. Mais dans un contexte où les industries doivent fabriquer à flux tendu, les méthodes de fabrication sont amenées à s'adapter. Dans ce cadre là, l'unité flexible qui a été présentée s'avère une alternative possible à la classique méthode du matelassage. Le haut niveau d'intégration atteint avec cette structure oblige à prendre en compte les données échangées avec les processus amont et aval du placement. La connaissance précise des données véhiculées entre les processus voisins et des contraintes imposées par eux, va contribuer à expliciter, dans le chapitre deux, la formulation détaillée de notre problématique.

CHAPITRE II FORMULATION DU PROBLEME DU PLACEMENT ET ETAT DE L'ART

L'éducation, c'est l'homme qui
progresses de la certitude ignorante à
l'incertitude réfléchie.
Don Clarks' Scrapbook.

I Introduction

Le premier chapitre a permis de présenter les motivations économiques, ainsi que les choix technologiques justifiant cette étude. Une attention particulière est ici portée pour établir une formulation du problème du placement de formes quelconques en y intégrant les spécificités propres au contexte textile. Une analyse bibliographique est ensuite proposée. Celle-ci permettra de positionner l'étude par rapport à la très grande variété des formulations rencontrées. La résolution du problème du placement - découpe a en effet déjà soulevé bien des convoitises scientifiques, mais l'étendue des possibilités rend le sujet intarissable.

II Formulation du problème du placement de pièces de vêtements

Dans cette partie, il convient tout d'abord de définir les entités utilisées, c'est-à-dire la matière ainsi que les pièces à produire. Ensuite, seront présentées les contraintes auxquelles ces entités sont soumises. Pour terminer, nous envisagerons un éventail des fonctions de but qu'il est possible d'utiliser pour tendre vers l'objectif de réduction des coûts.

2.1 Définitions

2.1.1 Le placement

Le placement (ou invariablement plan de coupe, patron, activité, amalgame) est en réalité un plan donnant la position de chaque pièce à produire. Le placement doit être construit de la "meilleure manière", c'est-à-dire de telle façon que toutes les contraintes soient satisfaites, tout en tenant compte d'une fonction économique calculant le coût de production qu'il faut minimiser. Ici, la fonction économique est simplement le coût de la matière. Le témoignage des placeurs manuels ou les expériences réalisées avec les algorithmes précédents indiquent que, en général, plus il y a de vêtements à placer, plus le rendement matière augmente. Cependant, celui-ci tend vers une limite maximale propre à chaque type d'article

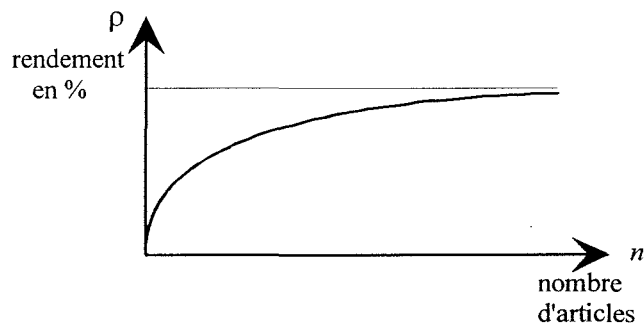


Figure 2.1

Les articles inclus peuvent être totalement différents (veste + gilet + pantalon), ou de même morphologie (par exemple: uniquement des pantalons), de tailles différentes ou de toute autre combinaison. Le nombre de pièces sera donc variable et parfois important. Ceci étant, on considérera dans la suite, une succession de sous - placement P_i , chacun réalisant une partie de l'ordre de coupe. La réunion de tous les sous - placements constitue le placement P satisfaisant la réalisation de l'ordre de coupe complet:

$$\cup P_i = P \quad (1)$$

2.1.2 Définitions relatives à la matière

La matière "M" est plane, souple et fibreuse. Elle est déroulée dans une bobine de longueur supérieure à la dimension du placement de toutes les pièces de vêtement à produire. Par hypothèse, on considérera que la matière est bornée à gauche et non-bornée à droite, la limite gauche étant simplement le début du rouleau.

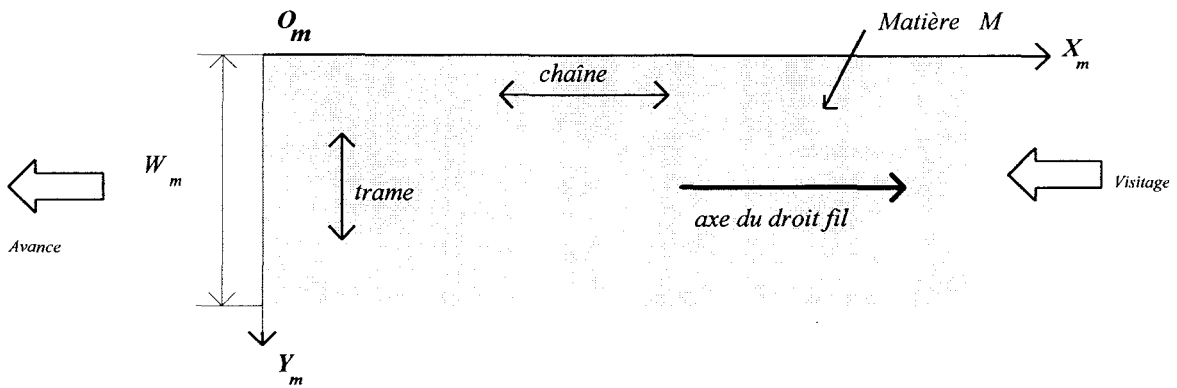


Figure 2.2

On définit:

a) La laize: W_m

Sa valeur nominale vaut en général : $90 \leq W_m (cm) \leq 150$. En pratique, pour une laize fixée, le visitage de la pièce de tissu fait apparaître une variation de la laize. Par mesure de simplification, et pour chaque bande de tissu visitée, la largeur W_{B_j} de référence sera la plus petite laize de la bande B_j .

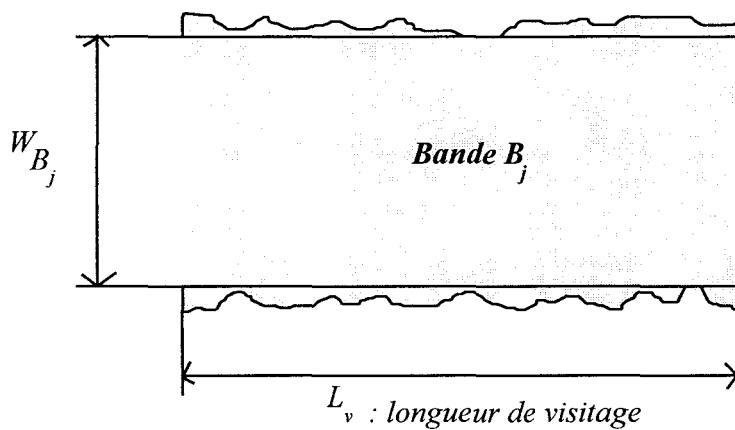


Figure 2.3

Par ailleurs, les sous - placements produits chevaucheront assez fréquemment deux ou plusieurs bandes élémentaires de visitage de longueur L_v . Dans ce cas, on conviendra que la largeur autorisée pour ce placement P_i , sera la plus petite largeur parmi celles des bandes destinées à l'accueillir :

$$w_m = \underset{B_j / B_j \cap P_i \neq \emptyset}{MIN} w_{B_j} \quad (2)$$

Comme on ne connaît pas a priori la longueur du placement, on estime celle-ci (\hat{L}_i) en prenant un rendement matière pessimiste ρ ¹ choisi arbitrairement tel que ρ est égal à 60% sur matière unie, 45% sur matière à motifs :

$$\hat{L}_i = \frac{\rho \times \text{surface des formes à placer}}{W_M} \quad (3)$$

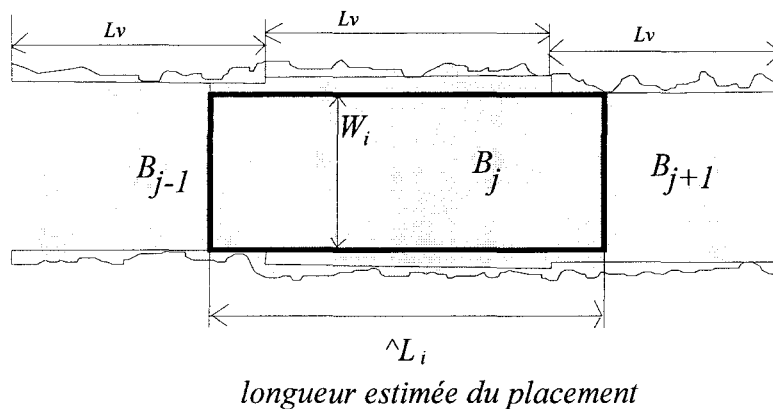


Figure 2.4

b) Repère matière : (O_m, X_m, Y_m)

L'axe $O_m X_m$ est orienté vers les modules amonts de visitage et d'approvisionnement matière. Lorsque le placement sera achevé, la matière progressera sur nos figures vers la gauche. L'évolution du placement s'effectue donc vers la droite.

L'axe $O_m Y_m$ représente fictivement la limite gauche du placement. Il arrive que le début de la matière ou que la découpe du placement précédent, présente un profil non rectiligne, laissant un espace pouvant être récupéré. Dans ce cas, l'axe $O_m Y_m$ demeurera l'axe de référence pour le placement comme pour la modélisation du contour du front du tissu. Il est à préciser que le front est alors identifié par la forme F_0 , laquelle sera soumise à une contrainte spécifique définie ultérieurement.

¹ définition du rendement matière $\rho = \frac{\text{surface utile}}{W_m \times L_i}$

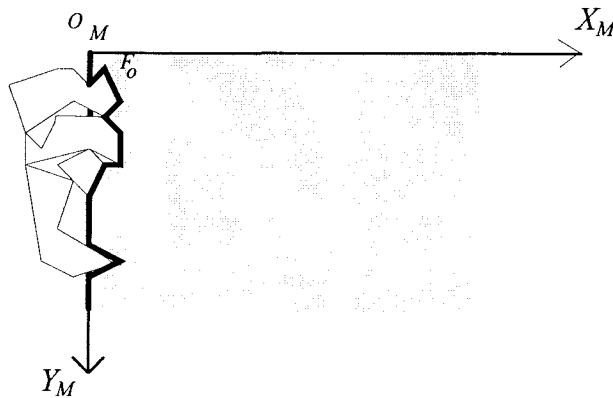


Figure 2.5

c) Chaîne du tissu

C'est la direction de la fibre du tissu.

d) Trame du tissu

C'est la direction perpendiculaire à la chaîne

e) Axe du droit fil

C'est un axe fictif parallèle à la direction de la chaîne du tissu et orienté par le sens de l'évolution du placement.

f) Les types de tissu

Dans le cadre du problème que nous étudierons, nous distinguerons essentiellement deux types de tissu:

- Les tissus unis ou quasi unis : les tissus quasi unis sont imprimés par des motifs dont la petite taille de la périodicité permet de ne pas la considérer. Les tissus à côtes tels que les velours entrent dans cette catégorie.
- Les tissus à motifs : ils peuvent être à rayures, écossais ou à motifs libres. Par projection sur le repère matière, il est toujours possible de définir une périodicité du motif selon la chaîne (T_x) ou selon la trame (T_y) du tissu.

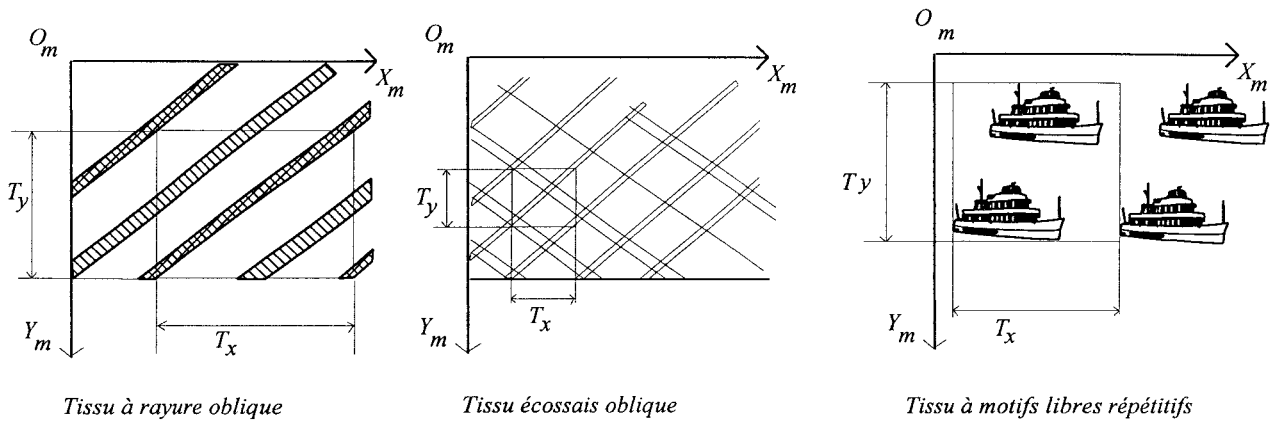


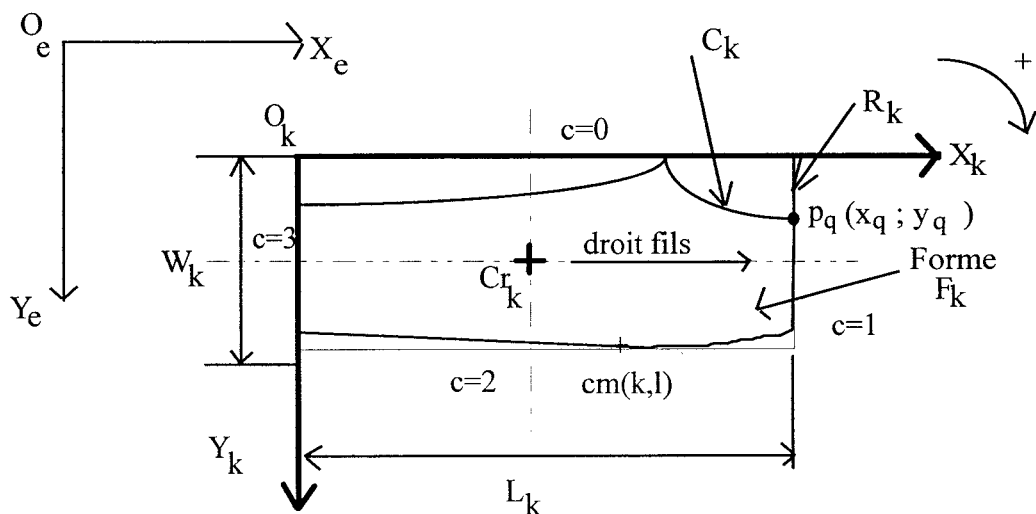
Figure 2.6

Dans tous les cas, les motifs répétitifs peuvent être ramenés à un rectangle de dimensions (T_x, T_y) , dont les côtés sont parallèles aux axes du repère-matière. On supposera par la suite que

$$0 \leq T_y \leq W_m \text{ et } 0 \leq T_x \quad (4)$$

Les cas où $T_x=0$ ou $T_y=0$ correspondent théoriquement au cas d'un tissu sans motif dans la direction considérée.

2.1.3 Définitions relatives aux pièces



Figures 2.7

Le contour C_k de chaque forme F_k est défini initialement par une liste ordonnée de points de coordonnées entières $\pi_q(x_q, y_q)$ dans le repère (O_e, X_e, Y_e) tels que :

$$\left| x_{q+1} - x_q \right| \leq 1 \quad (5)$$

$$\left| y_{q+1} - y_q \right| \leq 1 \quad (6)$$

Le repère (O_e, X_e, Y_e) est un repère intermédiaire dans lequel le contour est édité. Les formes définies dans ce repère ont leur direction de droit fil² parallèle à l'axe $O_e X_e$. Leur sens est pratiquement lié au sens du poil.

Le rectangle exinscrit R_k est choisi de telle sorte que deux de ses côtés soient parallèles à l'axe $O_e X_e$. Ses dimensions sont W_k, L_k , telles que :

$$W_k = \left| \bar{y} - \underline{y} \right| \quad (7) \quad \text{et}$$

$$\text{où } \bar{y} = \underset{\pi_q \in C_k}{\text{Max}} (y_q) \quad \text{et} \quad \underline{y} = \underset{\pi_q \in C_k}{\text{Min}} (y_q)$$

$$L_k = \left| \bar{x} - \underline{x} \right| \quad (8)$$

$$\text{où } \bar{x} = \underset{\pi_q \in C_k}{\text{Max}} (x_q) \quad \text{et} \quad \underline{x} = \underset{\pi_q \in C_k}{\text{Min}} (x_q)$$

Le repère (O_k, x_k, y_k) est appelé "repère propre". Il servira de référence à toutes les données géométriques portant sur la forme F_k . L'origine O_k est située conventionnellement sur le coin supérieur gauche du rectangle. Les axes $O_k X_k$ et $O_k Y_k$ sont respectivement parallèles aux axes $O_e X_e$ et $O_e Y_e$.

Le cran de montage $cm(k,l)$ correspond au point de raccord avec une autre pièce F_l . Son existence est justifiée lorsque la matière réceptrice est à motifs. Dans un but esthétique, les crans de montage permettent d'assurer la continuité des motifs entre les différentes pièces en contact. La pièce F_l possède, elle aussi, un cran de montage correspondant $cm(l,k)$. La position des crans de montage est fixée lors de la conception. Ses coordonnées sont : $(X_{cm(k,l)}, Y_{cm(k,l)})$.

Les quatre côtés du rectangle R_k sont étiquetés par c , $c \in \{0,1,2,3\}$. Ils sont numérotés dans le sens des aiguilles d'une montre. Le côté $c=0$ correspond au premier côté rencontré en partant de l'origine O_k .

Cr_k sera le centre du rectangle exinscrit ainsi que le centre de rotation. Ses coordonnées par rapport au repère (O_k, X_k, Y_k) sont :

$$x_{Cr} = \frac{L_k}{2} \quad ; \quad y_{Cr} = \frac{W_k}{2} \quad (8)$$

²le droit fil est précisé lors de la phase de conception

2.2 Les contraintes

2.2.1 Les contraintes géométriques générales

Elles concernent invariablement toutes les parties de vêtement.

2.2.1.1 Contrainte de disjonction

Le recouvrement des pièces est interdit. Cela traduit dans le placement P que l'intersection des formes prises deux à deux est vide. D'où la règle:

$$\forall F_k \text{ et } F_l \in P \text{ avec } k \neq l \quad (9) \\ INT(F_k \cap F_l) = \emptyset$$

La fonction $INT(F)$ décrit la région intérieure au contour de F à l'exclusion du contour lui-même. Son emploi indique ici que des portions de contour de deux formes peuvent être communes, et plus généralement que deux formes voisines peuvent être en contact en au moins un point.

Remarque N°1 : Pour des raisons technologiques (épaisseur du trait de coupe, tolérance de précision), on serait tenté de contraindre plus fortement la règle en imposant un espacement entre chaque forme. En réalité, cette précaution légitime peut être prise en compte plus simplement en accroissant la marge de contour au niveau de chaque forme ou plus simplement au niveau de la valeur-couture; c'est ce que nous supposons

Remarque N°2 : Si cette contrainte est simple dans sa formulation, en revanche elle peut être responsable d'une grande complication dans la résolution du problème du placement. Lorsque les formes traitées sont géométriquement simples (rectangles, cercles), son expression mathématique peut se limiter souvent à un ensemble réduit d'équations et d'inéquations linéaires. La résolution peut ainsi utiliser les méthodes classiques d'optimisation linéaire avec contraintes. La réutilisation de ces méthodes dans le cas des formes quelconques introduirait, a priori une explosion du nombre de ces inéquations s'il fallait exprimer la même contrainte pour chaque point de chaque forme.

Remarque N°3 : Aucune autre spécification mathématique de cette contrainte n'est exprimée ici en raison de la dépendance avec le modèle de traitement utilisé pour représenter les formes irrégulières.

2.2.1.2 Contrainte d'inclusion à la matière

Elle traduit l'inclusion de la pièce dans le tissu.

$$\forall F_K \in P, \quad 0 \leq y_{O_k} \leq W_M - W_k \quad \text{et} \quad x_{O_k} \geq 0 \quad (10)$$

La matière n'étant pas bornée à droite, aucune contrainte sur la valeur maximale de l'abscisse n'est proposée. Nous verrons cependant les limites arbitraires que nous nous fixerons pour travailler en placement par bande. Cependant, une condition sur l'abscisse minimal est ici nécessaire, mais non suffisante si l'on tient compte du profil F_O du début de rouleau.

2.2.1.3 Contrainte de début de rouleau

C'est une contrainte permettant la récupération des espaces libres de début de rouleau ou de la fin de placement d'une bande précédente. Elle se traduit par l'existence d'une région éventuellement non vide F_O respectant (9), à laquelle s'ajoute une contrainte de positionnement absolu dans (O_M, X_M, Y_M) telle que :

$$x_{O_0} = 0 \quad ; \quad y_{O_0} = 0 \quad (11)$$

2.2.1.4 Contrainte du droit fil

L'orientation des pièces repose essentiellement sur des considérations de style, sur des contraintes mécaniques du vêtement fini ou sur des critères technologiques de découpe. Le droit fil est une contrainte absolue imposant à chacune des pièces de vêtement d'avoir son axe de droit fil parallèle ou perpendiculaire à la fibre du tissu.

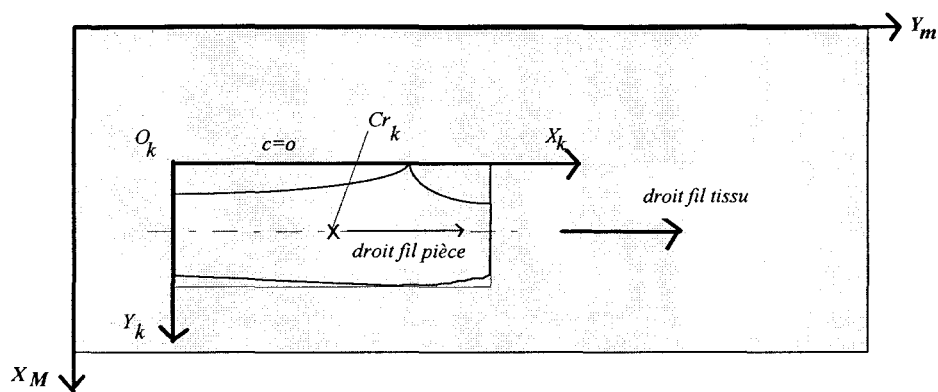


Figure 2.8

La contrainte du droit fil limite à 8 le nombre de positions possibles pour une pièce. Quatre orientations (quatre angles de rotations de centre Cr_k) ainsi que deux positions symétriques par rapport à l'axe $Cr_k X$ sont admissibles. Il n'est pas utile ici de considérer la symétrie par rapport à $Cr_k Y$; celle-ci n'apporte pas de positions supplémentaires à la combinaison des autres transformations.

Les quatre orientations

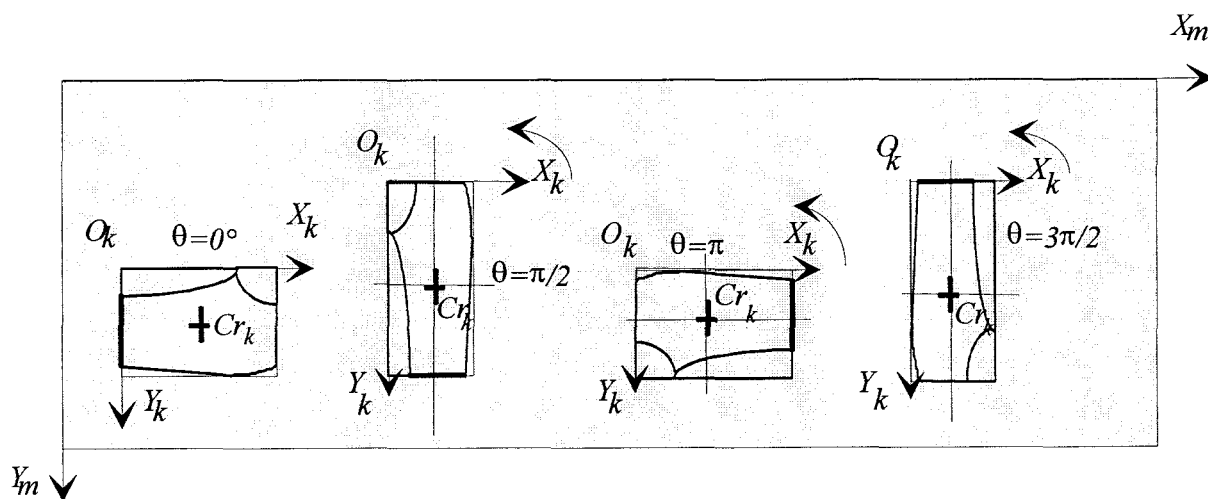


Figure 2.9

Si θ est l'angle formé entre l'axe du droit fil pièce et l'axe du droit fil tissu, alors la règle peut s'écrire:

$$\theta = \lambda \times \frac{\pi}{2}, \quad \lambda \in \text{orientations_admises}(k) \tag{12}$$

$$\text{orientations_admises}(k) \subset \text{orientations_admissibles} = \{0,1,2,3\}$$

Nous appellerons "*orientations_admissibles*", l'ensemble constant $\{0,1,2,3\}$. L'ensemble "*orientations_admises*", inclus dans "*orientations_admissibles*", contiendra les valeurs permises à λ . Toutes les orientations sont autorisées ici par défaut.

La symétrie ou le recto - verso

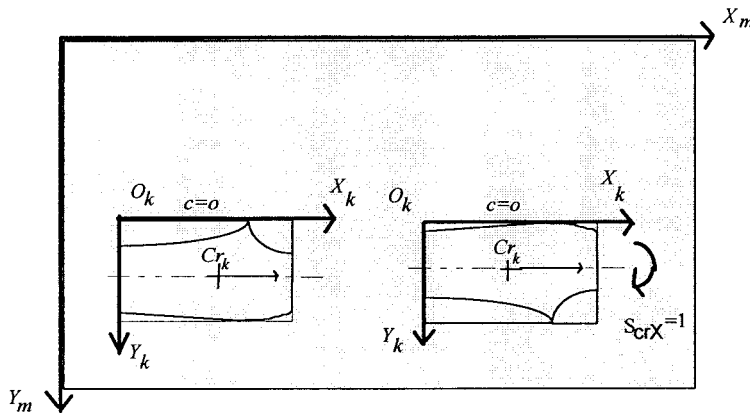


Figure 2.10

Afin de contrôler cette contrainte, il est possible de paramétrer la variable booléenne S_{CrX} , dont l'état "1" autorise l'action de retournement par rapport à $C_k X_k$.

par défaut : $S_{CrX} = 1$ (13)

Remarque :

Lorsque le placement était réalisé manuellement, il arrivait au placeur de biaiser volontairement le respect du droit fil. Une tolérance de $\pm 1\%$ (pivotement de 1 cm pour 1m de longueur) permettait de "tricher" pour économiser une quantité substantielle de tissu. Cette tolérance n'est pas prise en compte ici. Le système est certes plus contraint, mais dans un problème où la dimension de l'espace des configurations est synonyme de temps d'exploration, les risques d'égarement dans la recherche d'une bonne solution s'en trouvent diminués.

2.2.1.5 Contrainte de prise en compte des défauts

Les défauts locaux du tissu considérés dans notre étude peuvent être :

- des taches
- des accrocs

- des ruptures de fibres
- des anomalies de tissage
- des défauts de nuance

Ils peuvent être de surface plus ou moins importante et de forme variable. Leur détection ainsi que leur repérage intervient au niveau du visitage . Le défaut devient alors une région qu'il faut éviter.

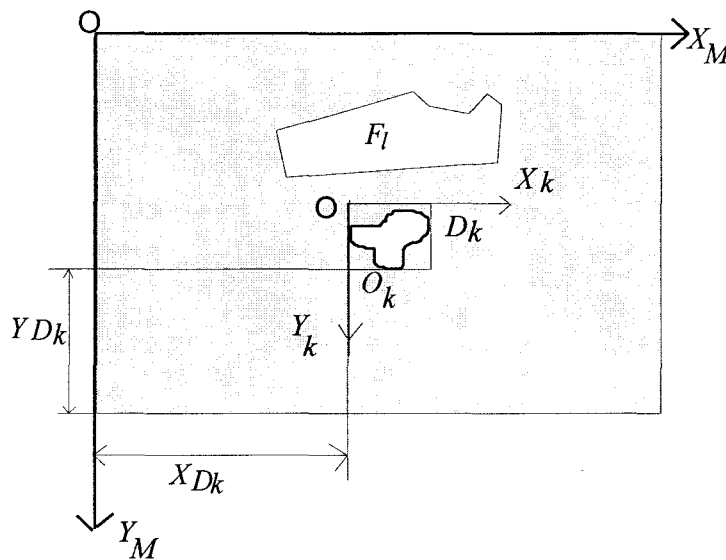


Figure 2.11

La contrainte s'exprime par:

$$\forall D_k \in M, \quad \forall F_l \in P$$

$$INT(D_k \cap F_l) = \emptyset \quad (14)$$

Elle traduit ici le fait que l'on caractérise le défaut comme une forme dont la position absolue est imposée et qui doit satisfaire la même contrainte de disjonction que les autres pièces à produire.

2.2.2 Contraintes géométriques spécifiques

Ces contraintes sont indépendantes et cumulables. Elles s'appliquent individuellement à chaque pièce de vêtement. Elles sont souvent induites par la nature du tissu ou introduites pour des considérations esthétiques.

2.2.2.1 La contrainte "dossé - dédossé"

Il arrive souvent que les tissus aient un aspect différent au verso. Dans ce cas, il n'est pas possible de réaliser le retournement des pièces non symétriques par rapport à l'axe de symétrie $Cr_k X_k$. L'autorisation de la symétrie est invalidée:

$$ScrX = 0 \quad (15)$$

2.2.2.2 La contrainte de sens du tissu

Le sens est une propriété fréquente dès lors que l'on travaille avec les tissus à motifs ou les velours. Il est alors impossible de disposer les parties visibles du vêtement à "rebrousse poil" ou à "contre sens" du motif. Par conséquent on interdit aux formes ne possédant pas de symétries intrinsèques, toutes les rotations de centre Cr_k . L'ensemble des orientations admises devient donc:

$$orientations_admises(k) = \{0\} \quad (16)$$

Pour ces formes, la seule orientation permise se limite à la position où les axes orientés de droit fil de la pièce et du tissu sont confondus.

2.2.2.3 La contrainte de chaîne

Pour les tissus unis, il n'est pas rare que l'on puisse faire faire un demi-tour aux pièces, pourvu que l'on respecte l'alignement de la chaîne. Dans ce cas, la contrainte peut s'écrire:

$$orientations_admises(k) = \{0,2\} \quad (17)$$

2.2.2.4 La contrainte de raccord (ou de positionnement relatif)

Cette contrainte est nécessaire pour assurer la continuité des motifs au niveau des raccords des différentes parties visibles d'un vêtement. Au moment du placement, et en supposant que les périodicités T_x et T_y des motifs sont constantes, les pièces en raccord ont leurs crans de montage espacés d'un nombre entier de fois la période spatiale du motif, et ce dans les deux directions du repère lié à la matière.

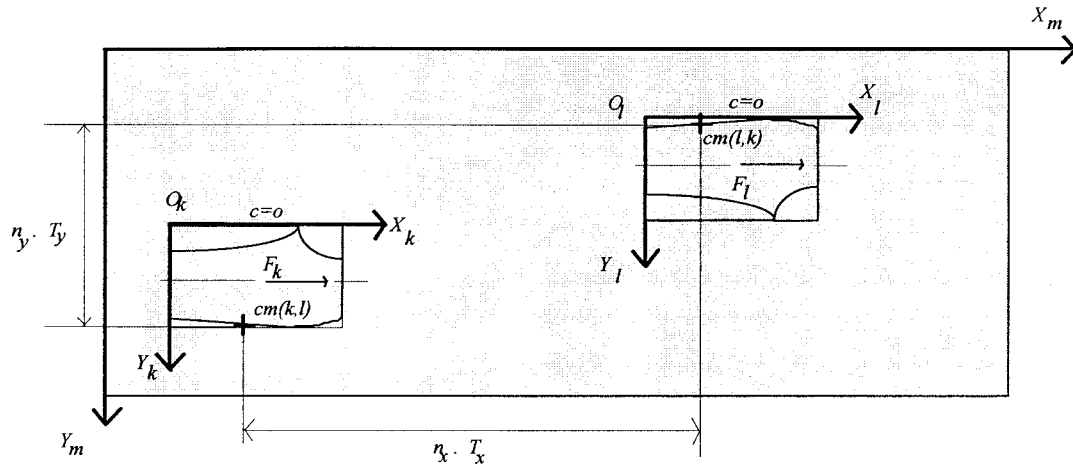


Figure 2.12

[Mor 73] propose d'écrire cette contrainte sous la forme:

$$\begin{aligned}
 \text{si } T_x > 0 \quad \text{alors} \quad & |x_{cm(k,l)} - x_{cm(l,k)}| = n_x \times T_x \\
 \text{si } T_y > 0 \quad \text{alors} \quad & |y_{cm(k,l)} - y_{cm(l,k)}| = n_y \times T_y \quad (18) \\
 & \text{avec } n_x, n_y \in \mathbb{N}^2
 \end{aligned}$$

Remarque :

L'hypothèse de la périodicité constante n'est pas toujours vérifiée, en particulier pour les tissus extensibles dont la tension au déroulement est difficile à régler. De même, lorsqu'il s'agit d'un tissu coloré, la continuité de la coloration n'est pas toujours assurée. Dans ce cas, la règle précédente n'est plus automatiquement validée aux points de raccord. Cependant, pour des distances limitées entre les points de raccord de deux pièces connexes d'un vêtement, on pourra considérer la périodicité constante ou la coloration inchangée. Il apparaît donc nécessaire de limiter l'éloignement des points de raccord par les bornes E_x, E_y telles que:

$$\begin{aligned}
 n_x \times T_x \leq E_x \quad \text{et} \quad n_y \times T_y \leq E_y \\
 \text{soit:} \\
 n_x \leq \text{Ent}\left(\frac{E_x}{T_x}\right) \quad \text{et} \quad n_y \leq \text{Ent}\left(\frac{E_y}{T_y}\right) \quad (19) \\
 n_x, n_y \in \mathbb{N}^2
 \end{aligned}$$

E_x et E_y sont des paramètres propres à la matière.

2.2.2.5 La contrainte de positionnement absolue par rapport au motif (ou motif placé)

C'est une contrainte plus forte que la contrainte de positionnement relatif. Elle ne concerne souvent que quelques pièces du vêtement. Son but consiste à positionner un motif à un point précis de la pièce, par exemple un bateau juste sur la poche de chemise, c'est le motif placé. En raisonnant par la réciproque, on conviendra qu'il s'agit de placer la forme par rapport au motif, lequel est répétitif.

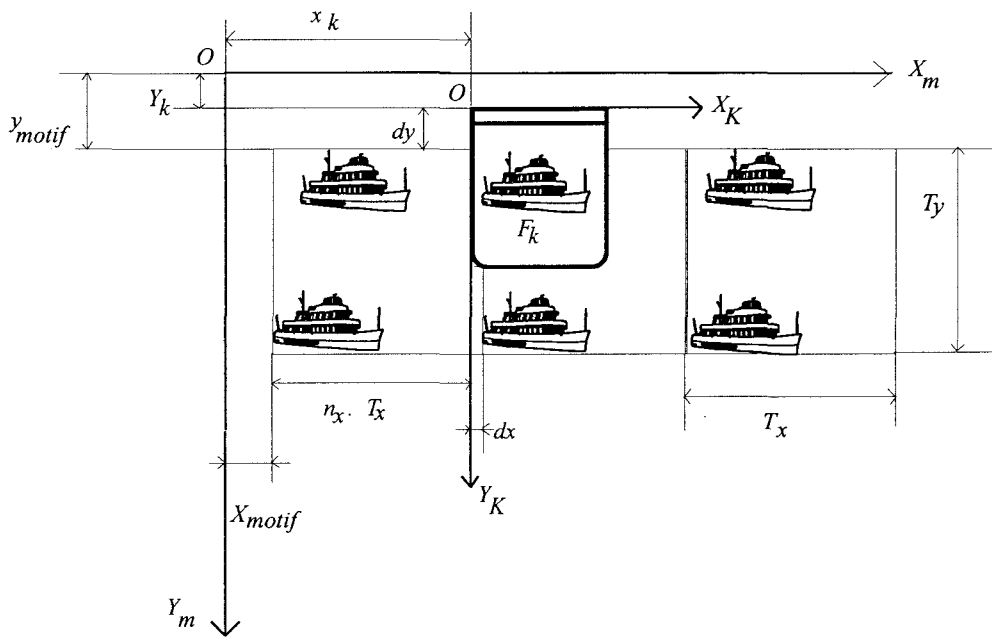


Figure 2.13

Supposant connaître les coordonnées précises du motif (x_{motif}, y_{motif}) se situant au plus près de l'origine du repère matière (calculé à partir du visitage et de l'avance), on peut alors exprimer cette contrainte par:

$$\begin{aligned} \text{si } T_x > 0 \quad \text{alors} \quad x_{motif} + n_x \times T_x &= x_k + dx \\ \text{si } T_y > 0 \quad \text{alors} \quad y_{motif} + n_y \times T_y &= y_k + dy \end{aligned} \quad (20)$$

dx et dy représentent les coordonnées de l'origine de la forme par rapport au repère motif telles que:

$$\begin{aligned} |dx| &\leq T_x \\ |dy| &\leq T_y \end{aligned} \quad (21)$$

signifiant que le positionnement relatif *forme/motif* ne peut excéder la période du motif.

2.2.3 La contrainte temporelle

La contrainte temporelle concerne le processus global de placement. On a vu, dans le premier chapitre, que le module de placement interagit avec d'autres modules qui évoluent en parallèle et avec des temps de réponse qui leur sont propres. Un mécanisme de synchronisation est mis en place pour garantir la bonne circulation des flux de données. Mais la différence des rythmes de travail engendre des attentes improductives. La cadence élevée qu'il est souhaité d'atteindre pour l'ensemble de la cellule et le voeu de tendre vers un fonctionnement permettant une production en juste à temps, fait apparaître la nécessité de tronquer parfois la durée d'exécution des tâches les plus longues. Naturellement, cette entorse à la "liberté d'évolution" ne peut se concevoir que si l'action en question ne pénalise en rien la qualité de fabrication du produit. Tout au plus, il peut être permis d'augmenter légèrement le coût. Peut-on diminuer le temps imparti à la recherche du placement optimal ? En réalité, l'efficacité d'un placement se mesure en terme de coût-matière, laquelle peut être économisée par un algorithme d'optimisation, au prix d'un temps de calcul directement appréciable. Une compression du traitement est alors synonyme de solution non-optimale, c'est-à-dire de perte. Mais la réalité industrielle oblige souvent à ce genre de compromis.

On a pu comprendre à la lecture du GRAFCET du chapitre I, que la réceptivité de la transition N°2 désactivait le processus de placement:

Transition N°2:

[bande Pi satisfaisante ou ("en-cours de découpe" vide et présence d'une solution)] et "en-cours de découpe" pas trop important

Cette réceptivité contient les termes logiques de la contrainte temporelle qui régissent le placement. Chaque terme est un événement surgissant à un instant qui balise l'axe des temps.

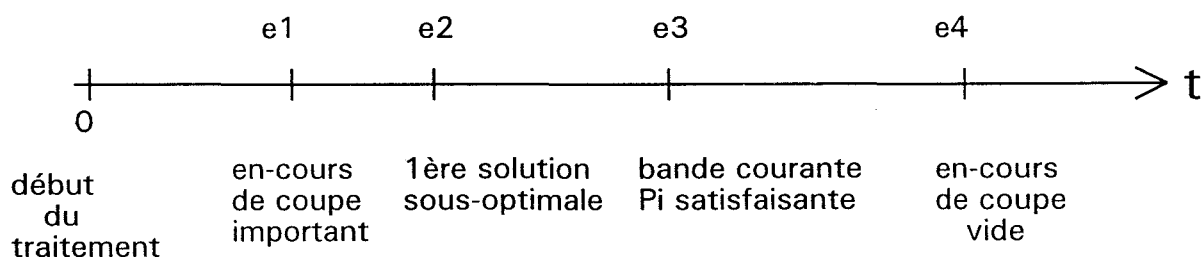


Figure 2.14

La chronologie proposée correspond à une situation normale vis-à-vis de la cadence. Cette situation est celle où l'événement $e3$ survient après $e1$, c'est-à-dire qu'une nouvelle bande satisfaisante est disponible après que l'en-cours de coupe ait diminué. Naturellement, ce même événement $e3$ précède $e4$, venant remplir l'en-cours de coupe avant qu'il ne soit vide. La position de l'instant $e3$ est liée au choix d'un critère d'arrêt de la recherche de l'optimum. Ce critère apparaît donc ici comme une contrainte à l'algorithme de placement.

Lorsqu'il s'agit de faire un compromis entre l'efficacité du placement et le temps d'obtention de celui-ci, [RED 88] suggère de fixer une borne inférieure V_{min} , à la "vitesse de décroissance relative de l'optimum temporaire de la fonction de but".

Soit g_n et g_{n+1} les valeurs des deux minima consécutifs de la fonction de but g aux instants respectifs e_n et e_{n+1} tels que:

$$\begin{aligned} g_n &\leq g_{n-1} \\ e_{n-1} &\leq e_n \\ \text{alors} \end{aligned}$$

$$\frac{g_{n-1} - g_n}{g_{n-1}} \times \frac{1}{e_n - e_{n-1}} \geq V_{min}$$

où $\frac{g_{n-1} - g_n}{g_{n-1}}$ est la décroissance relative de g

(22)

Cette contrainte signifie qu'il est inutile de prolonger la recherche d'une solution meilleure, compte tenu qu'il faudrait attendre un laps de temps supérieur à $(e_n - e_{n-1})$ pour espérer atteindre une nouvelle décroissance de $\frac{g_{n-1} - g_n}{g_{n-1}}\%$ de la fonction de but.

Evidemment, le choix de V_{min} conditionne l'arrêt de la recherche et donc l'instant e_n de la solution retenue, c'est-à-dire la position de l'événement $e3$ sur l'axe des temps (cf. Figure 2.14). La valeur de V_{min} influe donc sur l'équilibre du régime permanent de l'en-cours de coupe.

2.3 Les fonctions de but

Appelées aussi fonctions objectif, elles formalisent le problème du point de vue de l'objectif à atteindre. Elles quantifient chaque configuration du système, permettant de comparer deux états, d'orienter la recherche des solutions ou de décider de la recevabilité de la situation finale.

2.3.1 Classification des fonctions de but

Malheureusement, il n'y a pas unicité de la fonction de but pour un problème donné. Il y a par exemple, dans le cas du placement de formes, au moins deux façons de considérer l'évaluation de la qualité d'une solution : la surface des pertes ou la longueur de la matière utilisée. Les deux indices sont minimaux si la solution est optimale, pourtant l'utilisation de l'un ou de l'autre ne permet pas d'obtenir la même convergence dans un même algorithme. Dans le meilleur des cas, si la résolution est exacte et donne une solution optimale, le choix de la fonction objectif n'influe que sur la vitesse de convergence vers l'optimal. Mais bien souvent, son impact va même jusqu'à modifier la qualité du placement lorsqu'il est élaboré par un algorithme heuristique ne fournissant qu'une solution sous-optimale. Pour notre exemple, on conçoit facilement que choisir itérativement la forme générant un placement intermédiaire de longueur minimale, n'engendre pas la même solution que si l'on sélectionne à chaque ajout de forme celle provoquant la perte de matière la plus faible, même si l'objectif est a priori semblable. Ceci milite en faveur de la distinction qu'il convient de faire entre les fonctions de but destinées aux algorithmes exacts et celles que l'on emploiera plus spécifiquement pour les heuristiques.

Pensant à juste titre qu'il est illusoire d'atteindre la solution optimale dans les conditions industrielles les plus courantes, [DEL 89] a jugé prometteur en terme de temps, d'analyser les fonctions de but du placement selon trois rôles différents qu'il assimile à trois heuristiques : l'heuristique d'imbrication, d'évaluation, ou de construction. La première permet de sélectionner les formes aux imbrications prometteuses, la deuxième évalue l'association de deux formes, la troisième guide l'évolution de l'amalgame. Il montre ici l'intérêt de considérer la construction d'un placement en se plaçant à des niveaux de résolution différents.

Il est vrai que l'idée de percevoir le problème à des distances focales multiples est proche de l'attitude du placeur humain. Celui-ci traite globalement la géométrie de l'amalgame et affine ensuite les associations en accommodant sa rétine sur une zone locale. C'est une façon de traiter le problème séquentiellement. On voit ici l'importance que l'on peut accorder à tel ou tel indice, suivant l'ampleur de la région que l'on observe.

Une autre façon de diviser la difficulté de l'évaluation d'un placement consiste à supprimer des dimensions spatiales. On se pose alors la question suivante : Doit-on exprimer la mesure en terme de surfaces ou de longueur? Doit-on considérer qu'un placement-plan se doit d'être quantifié en terme de chute surfacique ou de surface occupée; ou peut-on se dire que compte tenu de la contrainte de largeur, la minimisation de la longueur entraîne fatalement le tassement des formes qui occuperont donc moins de surface ? Dans le premier cas, la fonction comptabilise un espace à deux dimensions, dans l'autre elle représente une distance unidimensionnelle.

Dans ce qui précède, la contrainte de largeur participe implicitement à la construction du placement. Il arrive que la contrainte intervienne explicitement dans la fonction-objectif. On a constaté au niveau de la contrainte temporelle, qu'elle pouvait s'exprimer en fonction du but. Pourquoi la fonction de but n'inclurait-elle pas la contrainte ? Cette idée a déjà germé chez bien des auteurs montrant des exemples d'algorithmes d'optimisation avec contraintes. Dans ces cas, l'expression de la contrainte dans la fonction-objectif apparaît comme une pénalisation qu'il faut réduire au minimum. Elle est souvent pondérée par un coefficient qui affecte une importance relative par rapport à d'autres termes qui expriment l'objectif à atteindre.

Enfin l'idée d'exprimer plusieurs grandeurs dans une même fonction semble faire école dans plusieurs heuristiques. C'est aussi le cas, par exemple, lorsqu'il s'agit de relativiser une grandeur ou une variation de grandeur par rapport à une autre. On parle souvent de rapport, de rendement ou de grandeur relative. Ce genre de fonction guide souvent l'évolution d'un placement par comparaison objective, c'est-à-dire indépendamment de l'amplitude de l'une de ces grandeurs. L'exemple le plus courant est le rendement surfacique : $\frac{\text{surface utile}}{\text{surface matière consommée}}$ faisant évoluer le placement vers une forme naturelle rectangulaire. L'inconvénient de ce genre de fonction est qu'elle représente une grandeur qualifiée d'intensive. Les lois mathématiques (comme l'addition) ne sont plus forcément internes dans l'ensemble de définition de ces fonctions (ici $[0,1]$). Par opposition, on dira des fonctions de coût absolu (comme la surface d'une chute) qu'elles sont extensives. On peut donc les additionner dans \mathfrak{R} en respectant la dimension de la fonction.

2.3.2 Définitions des grandeurs d'évaluation

Il est temps de proposer maintenant une présentation des grandeurs susceptibles d'apparaître comme membres de la fonction objectif. A la lumière de l'analyse précédente, il deviendrait extrêmement arbitraire de réaliser une classification des différentes grandeurs en fonction d'un des critères que l'on a énoncés précédemment (exact / heuristique, global / local, surface / distance, contrainte implicite / explicite, grandeurs extensives/ intensives). A un découpage aléatoire qui ferait perdre le bénéfice de cette analyse, on préférera une énumération ordonnée selon un simple souci de clarté graphique. On complétera cet inventaire non exhaustif par une synthèse rassemblant les informations apparaissant dans la littérature ainsi que des interprétations personnelles.

2.3.2.1 grandeurs relatives au placement

Elles sont présentées ci-dessous.

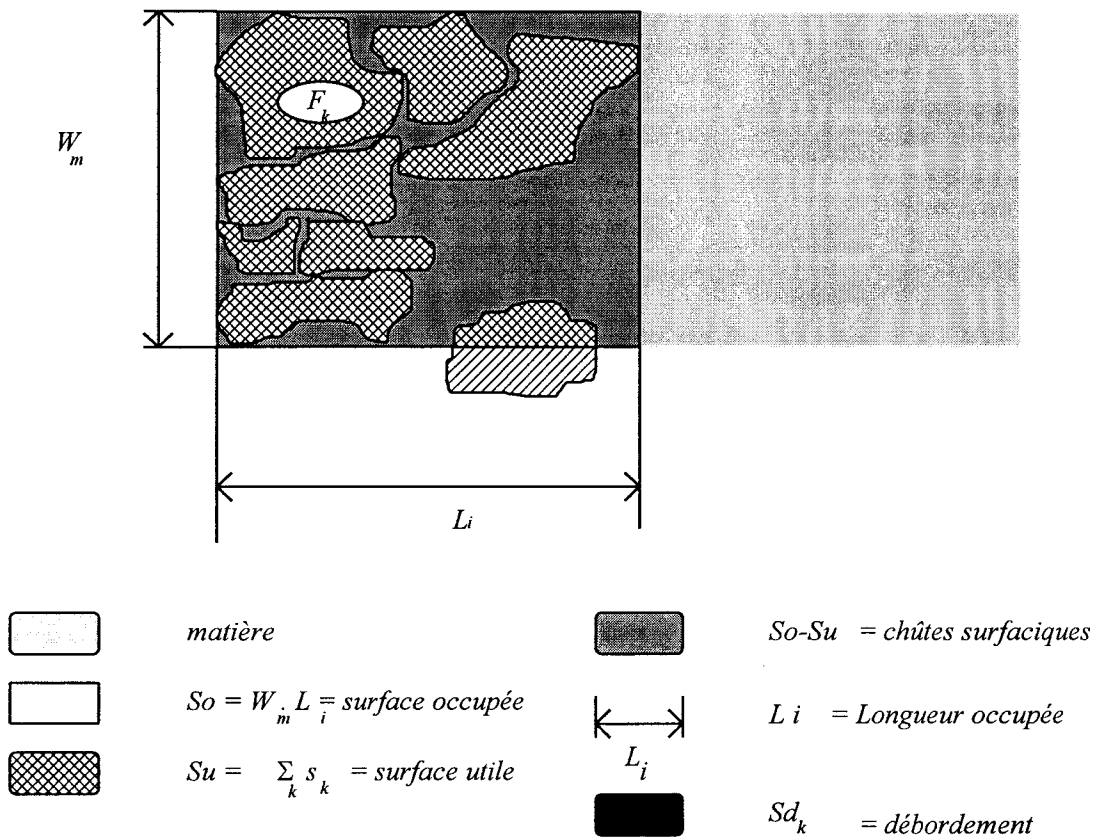


Figure 2.15

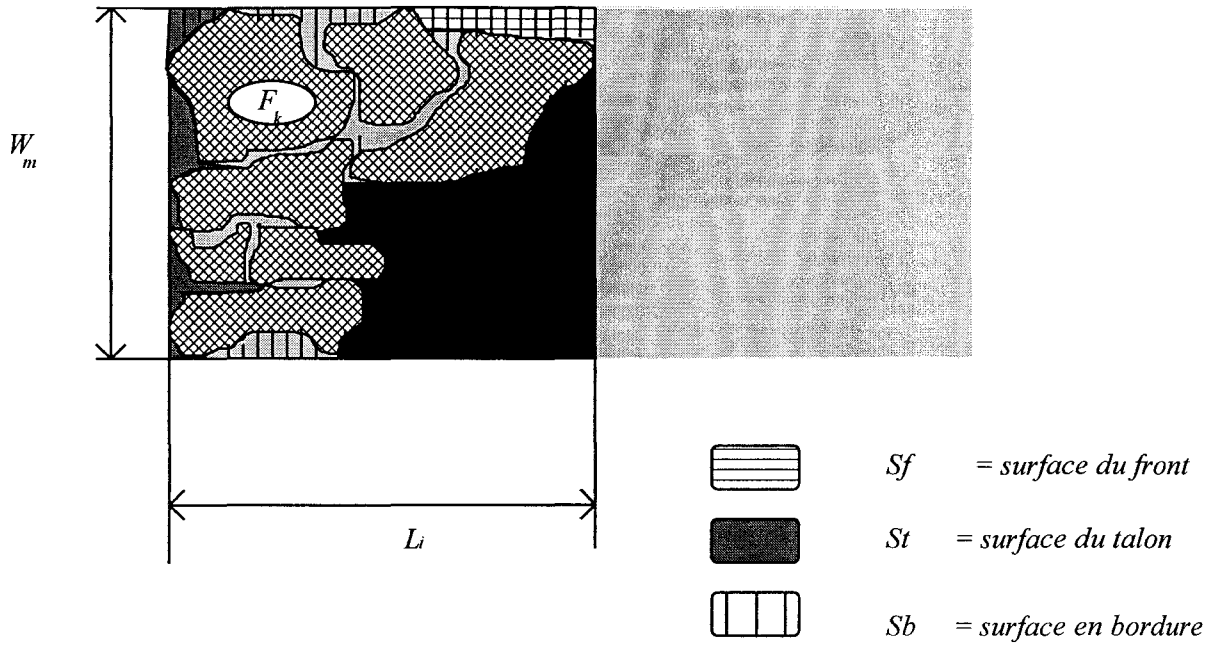


Figure 2.16

2.3.2.2 Grandeurs relatives à une association de deux formes

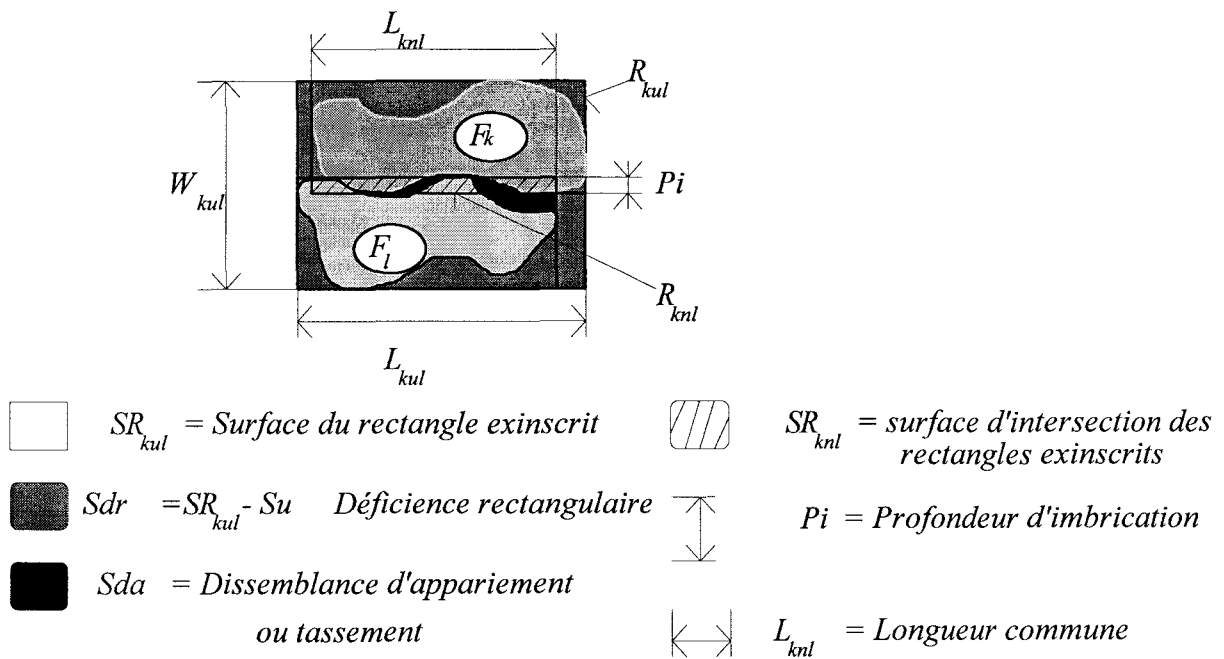


Figure 2.17

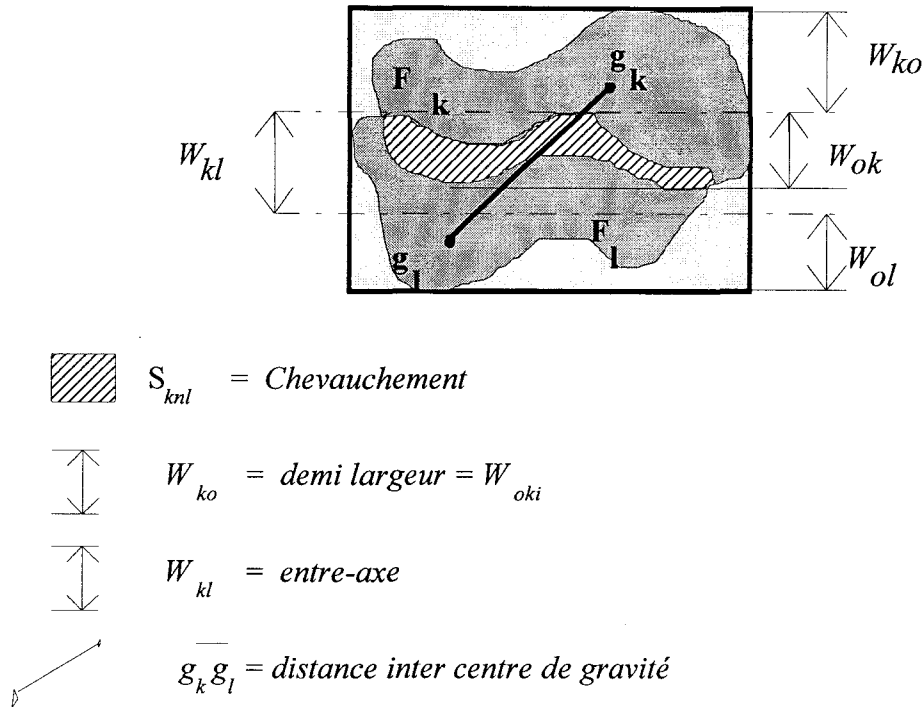


Figure 2.18

2.3.3 Synthèse

On a rassemblé dans les deux tableaux suivants, l'essentiel des fonctions objectif recensées au gré des travaux étudiés et qui traitent du problème du placement. Elles utilisent les grandeurs que nous avons définie précédemment. Le premier tableau rassemble les grandeurs simples, participant seules ou avec d'autres grandeurs simples à l'algorithme de placement. Le deuxième tableau énumère les fonctions composées conjugant plusieurs grandeurs caractéristiques.

Pour chacune des fonctions énoncées, nous préciserons pour en faciliter l'analyse, les observations personnelles suivantes.

- la fonction d'optimisation (min / max)
- la méthode qu'il est possible d'utiliser compte tenu de l'objectif de minimisation du coût matière (exact / heuristique)
- Le champ d'utilisation de la grandeur (global / local)
- la dimension traitée (surface / distance)
- la relation avec une contrainte (sans / implicite / explicite)
- la nature de la fonction (extensive / intensive)
- la référence bibliographique éventuelle où elle est citée pour un but semblable

-
- l'effet local dans le cas d'une sélection itérative des formes par ce critère. (voir algorithme de Freeman au §3)
 - l'effet global en supposant utiliser un algorithme exact.

Tous les critères présentés dans cette synthèse convergent vers un même objectif de minimisation des chutes. En réalité, le choix doit s'effectuer afin que la fonction objectif soit en adéquation avec l'espace des configurations ainsi qu'avec l'algorithme d'optimisation utilisé. Ces choix seront effectués au cours des discussions ultérieures

<i>Grandeurs simples</i>	<i>critère (min/ max)</i>	<i>exact / heuristique</i>	<i>global / local</i>	<i>surface/ distance</i>	<i>contrainte (sans/ implicite/ explicite)</i>	<i>extensive/ intensive</i>	<i>ref. bibliographi que</i>	<i>effet local</i>	<i>effet global</i>
<i>So</i> (surface occupée)	min	E et H	g et l	S	I (laize)	Ex		place les plus petites formes en premier	occupation optimale de la matière
<i>Su</i> (surface utile)	max	E	g et l	S	I (rectangle matière)	Ex	[HAI 70], [DEL 89]	place les plus grandes pièces en premier	occupation optimale de la matière
<i>So-Su</i> (pertes)	min	E et H	g et l	S	sans	Ex	[ALB 80]	place les plus grandes pièces en premier	occupation optimale de la matière
<i>Li</i> (longueur occupée)	min	E et H	g et l	Di	I (laize)	Ex	[ALB 80]	place les pièces les plus courtes en premier	occupation optimale de la matière
<i>Sdk</i> (débordement)	min	H	g et l	S	E	Ex	[PRE 83]	permet simplement de satisfaire la contrainte d'inclusion	
<i>Sf</i> (front)	min	H	g et l	S	I (laize)	Ex		aligne les formes à droite	rend le front rectiligne
<i>St</i> (talon)	min	H	g et l	S	I (laize)	Ex	[ALB 80]	aligne les formes à gauche	rend le talon rectiligne
<i>Sb</i> (bordure)	min	H	g et l	S	I (laize)	Ex	[DEL 89] [MOR 73]	associe les formes plates	extérieur plat

<i>Grandeurs simples</i>	<i>critère (min/max)</i>	<i>exact / heuristique</i>	<i>global / local</i>	<i>surface/ distance</i>	<i>contrainte (sans/ implicite/ explicite)</i>	<i>extensive/ intensive</i>	<i>ref. bibliographique</i>	<i>effet local</i>	<i>effet global</i>
SR_{kul} (rectangle exinscrit)	min	H	g et l	S	I (laize)	Ex	[ADA 76], [ADA 71], [ALB 77]	favorise les bonnes imbrications	directions privilégiées
Sdr (défiance rectangulaire)	min	H	g et l	S	I (laize)	Ex		favorise la construction de module rectangulaire (remplissage des concavités, assemblages homogènes, faible dépassement)	contour extérieur rectangulaire
Sda (dissemblance d'appariement)	min	H	l	S	I (disjonction)	Ex	[DEL 89]	favorise l'appariement des petites pièces avec les grandes	
SR_{knl} (intersection des rectangles exinscrits)	min	H	l	S	I (formes k et l non rectangulaires)	Ex	[DEL 89]	favorise l'imbrication, indéterminisme si absence de recouvrement	
W_{knl} profondeur d'imbrication	max	H	l	D	I (pas de décalage entre k et l)	Ex		favorise l'imbrication. Risques de décalage des deux formes	

<i>Grandeurs simples</i>	<i>critère (min/ max)</i>	<i>exact / heuristique</i>	<i>global / local</i>	<i>surface/ distance</i>	<i>contrainte (sans/ implicite/ explicite)</i>	<i>extensive/ intensive</i>	<i>ref. bibliographi que</i>	<i>effet local</i>	<i>effet global</i>
S_{kl} surface de chevauchement	min	H	l	S	E	Ex		favorise la validation de la contrainte de disjonction	
W_{kl} entre axe	min	H	l	D	I (disjonction)	Ex	[DIV 73]	favorise l'imbrication risque de décalage	
$d(g_k g_l)$ (dist. inter-centres de gravités)	min	H	g et l	D	I (disjonction)	Ex	[DEL 89], [DAG 90]	favorise le choix des petites pièces	évolution circulaire

<i>Grandeurs composées</i>	<i>critère (min/ max)</i>	<i>exact / heuris- tique</i>	<i>global / local</i>	<i>surface/ distance</i>	<i>contrainte (sans/ implicite/ explicite)</i>	<i>extensive/ intensive</i>	<i>ref. bibliogra- phique</i>	<i>effet local</i>	<i>effet global</i>
<i>Su/So (rendement matière)</i>	max	H	G et L	_	I	In		- Si la pièce déborde du rect. exinsc. courant, alors favorise la sélection des grandes pièces. - Sinon favorise la sélection des petites pièces	forme générale rectangulaire ou formation de bande
$(So^n + S_k) / So^{n+1}$ <i>Accroissement relatif de la surface occupée</i>	max	H	G et L	_	I	In	[DEL 89]	favorise les bonnes imbrications dans une direction privilégiée.	évolution vers une construction par bande
<i>Su/SR_{kul}</i> <i>(remplissage rectangle exinscrit)</i>	max	H	L et G	_	I	In	[DEL 89]	favorise les bonnes imbrications indépendamment de la taille des formes	évolution vers une forme générale rectangulaire

<p> $a. \sum R_{k \cup l} +$ $b. \sum_{k \in P} \sum_{l \in P} S_k S_l \cdot G_k G_l +$ $c. \sum_{k, l \in P} S_{k \cap l} \quad a, b, c \in \mathbb{R}^3 +$ </p> <p><i>Pondération (surf. rect. circ., moment de surf, surf. d'inters.)</i></p>	min	H	G	-	E	Ex (énergie)	[DAG 90]		minimise le chevauchement des formes, la surface du rectangle exinscrit et la distance totale entre les centres de gravité
<p>soit m le nombre de formes à placer</p> <p> $\sum_{k=1}^m \sum_{l=k+1}^m S_{k \cap l} + \sum_{k=1}^m S d_k$ $+ a. \sum_{k=1}^m S R_k \cdot y_k$ </p> <p><i>pondération (surf d'inters., surf. tot. moment de surf)</i></p>	min	H	G	-	E	Ex (énergie)	[PRE 83]		minimise le chevauchement deux à deux, minimise le débordement de chaque forme, favorise le placement des grandes formes à gauche
<p> $(S_o - S_u)_i +$ $a\% \cdot \sum_{k \notin P} S_k$ </p> <p><i>pondération (pertes surf. à l'itérat i, estim. pertes = fract. surf. non plac.)</i></p>	min	E avec recherche H	G	-	I	Ex	[ALB 80]		minimise en estimant les chutes futures

2.4 Placement en bande

Les contraintes temporelles et les choix technologiques (surface de travail de l'outil) exposés dans le chapitre I concernant notre cellule de découpe, nous engageant à diviser l'ordre de coupe. On ne réalisera pas un seul placement P optimal, mais une succession de sous placement P_i (eux même sous optimaux) que l'on appellera par convention: **bande**. Chaque bande doit être optimisée avec les considérations de finitude et de consistance.

2.4.1 Finitude d'une bande

a) *La limite gauche d'une bande correspond :*

- au début de la matière s'il s'agit de la première bande P_1

ou

- à la fin de la bande précédente, s'il s'agit des bandes P_i suivantes.

b) *La limite droite d'une bande est fixée :*

-a priori par un interval de tolérance

-a posteriori à l'instant de fin de recherche décidé par la contrainte temporelle.

En fait, une bande étant censée pouvoir contenir les plus longues formes, la longueur sera donc au moins égale au maximum des longueurs des formes non placées. D'autre part, on peut autoriser un décalage longitudinal limité entre les différentes formes si cela permet de faire varier avantageusement la fonction de but. En effet, on remarque expérimentalement que un petit déport entre deux formes permet parfois une imbrication très rentable (cf. Figure 2.19).

Ceci peut s'écrire par la double inéquation suivante:

$$Li_{min} = \underset{\substack{F_k \in EP \\ \alpha_k \neq 0}}{MAX} L_k$$

$$\forall P_i \quad Li_{min} \leq L_i \leq (1 + \alpha) Li_{min} \quad (23)$$

En outre, la tolérance du déport caractérisé ici par le paramètre α est choisi relativement petit. En effet, une valeur élevée peut conduire à obtenir un front de placement très irrégulier, éloignant ainsi la bande de la forme générale rectangulaire. D'autre part, une tolérance trop large diminue l'effet du placement en bande réduite et augmente

potentiellement le nombre de formes incluses dans la bande et donc aussi la complexité de la recherche.

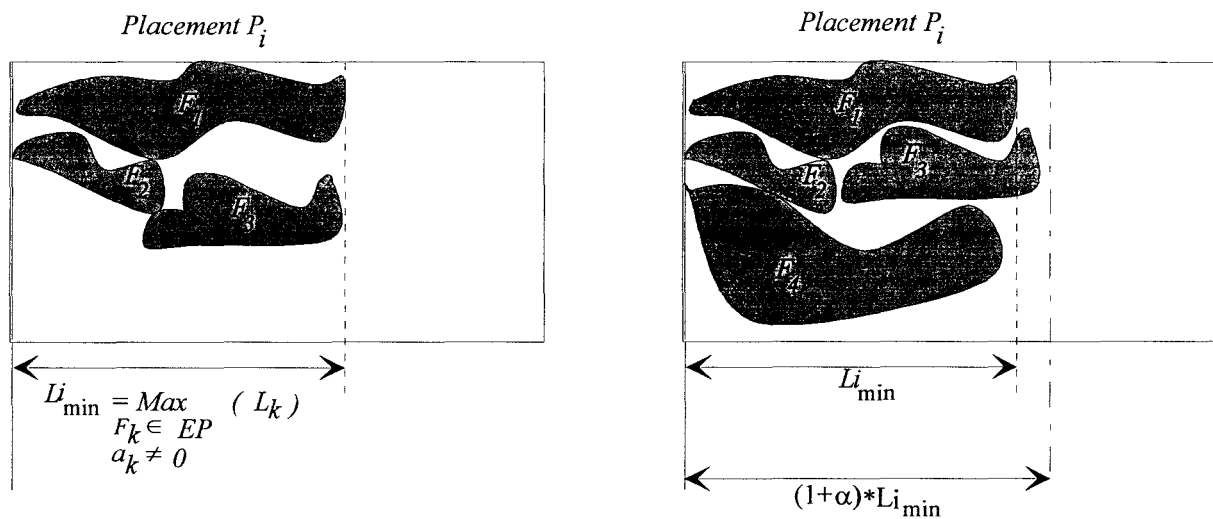


Figure 2.19 : finitude d'une bande

Cependant, si la complexité du front de bande P_i est importante, on prendra comme contour gauche de la bande suivante P_{i+1} , le profil imposé par le bande P_i :

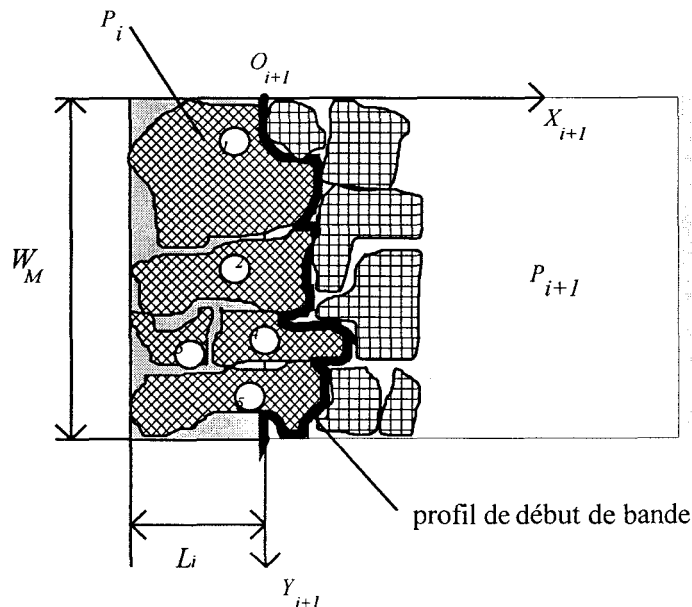


Figure 2.20 : profil de début de bande

2.4.2 Consistance d'une bande

Intuitivement, la qualité d'une bande dépend de l'étendue de l'espace des configurations qu'il faut visiter pendant un temps déterminé. Or, si la taille du carnet de commande est grande, l'espace des configurations s'élargie en conséquence, et donc l'espoir d'avoir une faible perte en un temps donné s'amenuise. Pour éviter cette situation inadéquate par rapport à l'objectif de juste à temps, on préfère réduire le nombre de formes à placer sur une même bande: c'est la contrainte de *consistance*. Il est vrai que les restrictions apportées par la contrainte de finitude constitue une première limitation du nombre de formes à introduire. Mais avec cette seule condition, l'algorithme tendra systématiquement à remplir l'espace pour atteindre la limite de longueur maximale $(1+\alpha)L_{min}^i$. Cette limite n'est donc plus "floue", mais stricte, et apparaît alors comme une contrainte forte pouvant exclure l'obtention d'une solution voisine plus rentable mais plus courte.

D'autres contraintes réduisant l'espace de recherche sont possibles:

- 1) $card(P_i) \leq N_{max}$. On limite à N_{max} le nombre de formes à introduire dans une bande.
- 2) $S_u \leq S_{u_{max}}$. On limite à $S_{u_{max}}$ la surface utile d'une bande

L'inconvénient majeur de ces deux critères est qu'ils contraignent le nombre de formes indépendamment de la qualité du placement. On peut par exemple imaginer le cas où un placement pourrait être amélioré en ajoutant une forme supplémentaire, mais où les conditions de consistance 1) ou 2) interdirait cette adjonction. A ces lois arbitraires nous préférons une règle plus souple visant à adapter le nombre de formes à la qualité maximale de la bande.

La solution retenue ici est de tenir compte des propriétés de la fonction de coût choisie pour contraindre sans biais la consistance d'une bande.

soit la fonction de coût correspondant à la chute dans la bande :

$$g(P_i) = S_o - S_u = W_M \cdot L_i - \sum_{F_k \in P_i} S_k \quad \text{avec:}$$

$g(P_i)$ = pertes associée au placement P_i

W_M = largeur matière

L_i = longueur du placement P_i

S_k = surface de la forme F_k

Si l'on considère que le placement est construit progressivement en ajoutant successivement sur la matière les formes du carnet de commande, toutes les formes étant en contact l'une contre l'autre, alors la fonction $g(P_i)$ subit des variations constituées de phases décroissantes et croissantes. Une décroissance reflète une diminution des pertes, c'est à dire aussi que la forme ajoutée n'allonge pas sensiblement la bande. Dès que g croît, cela signifie que l'on a probablement augmenté la longueur du placement et que le bilan des pertes est défavorable. Ainsi, en veillant à ce que la chute soit monotone décroissante, on guide ainsi la construction d'une bande rectangulaire sans aucune contrainte arbitraire de surface utile ou de nombre de formes. La minimisation de cette fonction nous fournira alors un sous placement de consistance optimale vis à vis des pertes.

III Etat de l'art sur le placement de formes irrégulières

Avant de présenter, dans les prochains chapitres le résultat de nos investigations, il semble opportun de situer l'étude par rapport à l'immensité du champ de possibilités qui s'ouvre au simple nom : "problème de découpe". Sans trop nous égarer sur des considérations théoriques qui n'apportent pas d'aides directes au problème, on proposera dans un premier temps une classique approche de la découpe telle qu'elle est présentée dans la plupart des survols bibliographiques. Une brève analyse permettra d'écarter tout un pan de la classification exposée et quelques références d'études bibliographiques très complètes seront suggérées pour aiguiller un lecteur avide de réflexions sur ces aspects de la découpe. En revanche, un point sur l'état de l'art dans le domaine de la découpe par contournage revêt ici d'un intérêt certain. Même si la littérature est moins prolixe dans cette classe de problèmes sur cet aspect, nous nous étendrons volontairement afin de dégager les fondements même d'une théorie encore en développement.

3.1 Classification du problème de découpe

Rappelons la définition universellement connue en recherche opérationnelle du "problème de découpe" :

Comment, à partir d'un stock de matière première, éventuellement sous forme de plusieurs entités de contour et de dimension différents, produire des objets éventuellement de forme et de dimension différentes, suivant une demande connue ou non à l'avance, avec le minimum de matière?

Cette définition qui se veut générale, montre bien ici la variété des spécificités que l'on peut lui adjoindre. Ces spécificités sont, en réalité, rencontrées dans la très large classe d'industries de transformation de la matière. Celle-ci peut être découpée physiquement (barres ou tôles de métal, bois, papier, tissu, cuir,...) ou virtuellement (remplissage de conteneur ou de four, répartition de tâches entre plusieurs processeurs, allocation de mémoire entre plusieurs

programmes, placement de composants sur un circuit imprimé, conception des masques de circuits intégrés ou reconstitution de puzzle).

Pour s'aider à classer les problèmes de découpe, [DMI 91] a scindé la problématique en quatre facteurs que sont : matières premières, commandes à produire, objectifs d'optimisation et contraintes de placement, faisant abstraction de la différence entre les contraintes propres au placement avec celles inhérentes à la découpe. Or, il semble, à la lumière de notre exemple, que cette abstraction soit parfois trop simplificatrice. Les contraintes énoncées dans ce même chapitre sont seulement relatives au placement indépendamment de la façon de découper les formes. Par contre, la nature irrégulière des formes impose implicitement que la découpe s'effectue par contournage et non par cisailage par exemple. On découvre alors que la contrainte de découpe dénonce intrinsèquement la géométrie de l'objet à découper. Cette précision étant apportée, voici l'analyse qui en résulte.

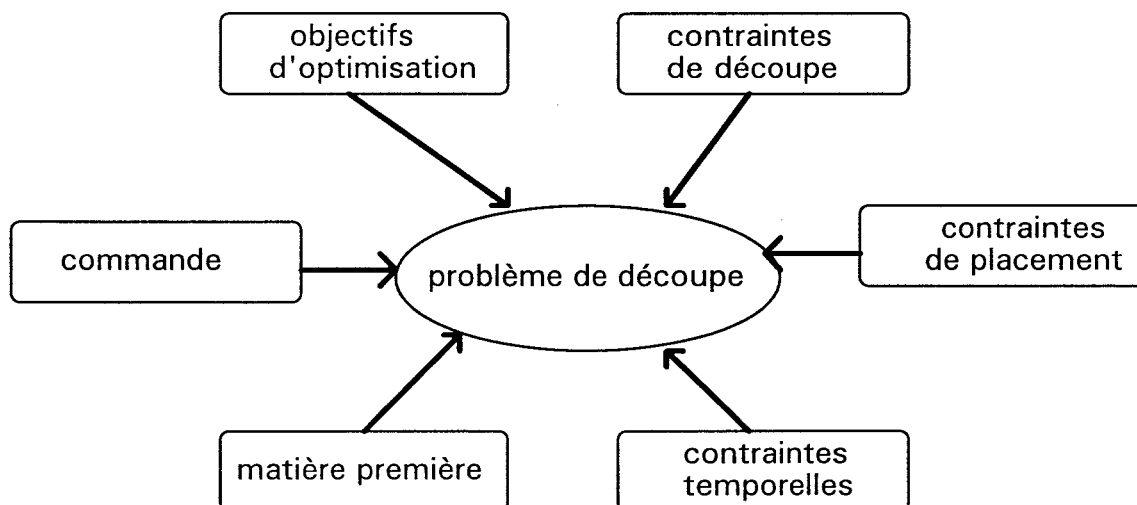


figure 2.21

Explicitons chacun de ces acteurs.

3.1.1 Objectifs d'optimisation

Avant de faire un distinguo au niveau des objectifs d'optimisation, il convient d'apporter une précision sur le vocabulaire. On parle de problème de découpe ou du problème du placement. La littérature fait en général bien la différence et considère le placement (PbP) comme un problème auxiliaire au problème de découpe (PbD). En fait il s'agit plus d'une inclusion de (PbP) dans (PbD). Le problème du placement se fixe le même objectif mais sur une seule entité matière. Il est donc toujours présent quelque soit le problème.

[DMI 91] dans son premier chapitre, a clairement fait apparaître 3 scénarios possibles d'optimisation.

- 1) Toute la matière doit être utilisée, entraînant souvent la surproduction de certaines pièces. Les chutes sont minimales, mais non récupérables car inférieures à la surface de la plus petite pièce.
- 2) Le nombre d'entités matière utilisées est minimal. La surproduction est interdite. Aucune hypothèse sur la chute n'est posée.
- 3) Un compromis est cherché entre le nombre d'entité matière et la chute. La surproduction est autorisée, mais souvent limitée.

On pourra ajouter:

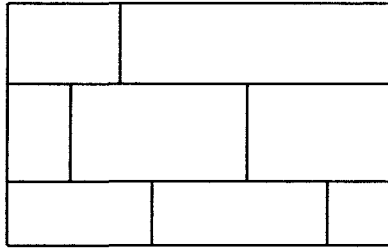
- 4) Le Placement pur. Une seule entité est réalisée. La chute doit être minimale. La surproduction est éventuellement autorisée.

Lorsque le problème ne se pose que pour une seule entité matière, c'est le scénario (4) qui est vécu. Mais s'il s'agit du scénario (2), il se peut que le placement sur une des entités matière ne soit pas optimal localement, mais de bonne qualité sur le plan de l'occupation matière globale. Dans ce cas, le placement est vu comme une activité qui n'est pas forcément optimale pour l'entité en question. Notre objectif est ici plutôt opposé. On cherche à construire des bandes de bonne qualité, sans considérer le résultat global en fin de production. Mais la perte globale étant en général essentiellement liée à la chute de fin de bande, la possibilité de récupérer celle-ci compense ici les effets négatifs du principe de bande.

3.1.2 Contraintes de découpe

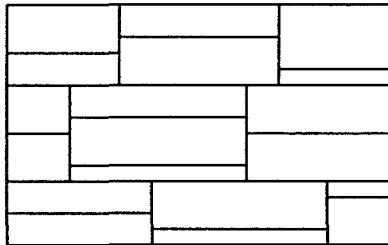
Ces contraintes apparaissent avec la seconde dimension spatiale. On distingue par ordre croissant de relaxation de la contrainte:

Les différents modes de découpe



La découpe guillotine à deux stades

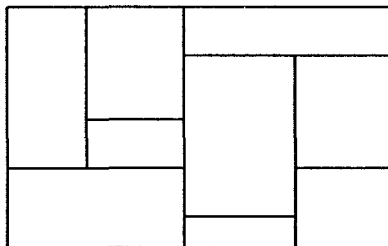
Elle s'effectue par cisailage de la plaque d'un bout à l'autre, et ce successivement dans les deux directions (longueur, largeur).



La découpe guillotine à n stades (ici

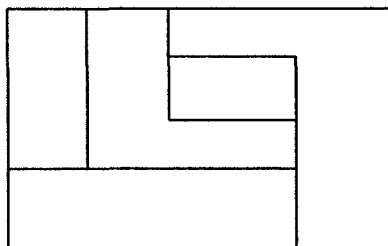
3)

On effectue la découpe guillotine n fois en alternant la direction de coupe.



La découpe guillotine quelconque

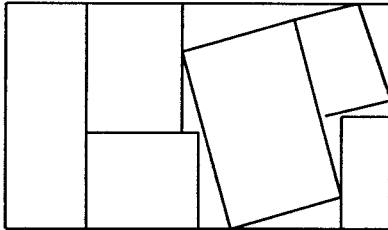
Ce procédé impose seulement que la coupe soit débouchante en au moins 1 coté de la plaque ou d'une sous-plaque. Ceci est possible par tronçonnage.



La découpe orthogonale

Seules les coupes parallèles au bord de la plaque sont autorisées. L'outil de coupe doit être ponctuel.

Rectangulaires



La découpe non - orthogonale ou par contournage

La découpe peut s'effectuer dans toutes les directions et n'est plus contrainte (laser, jet d'eau, oxycoupage,...). Les formes à découper sont:

ou Irrégulières

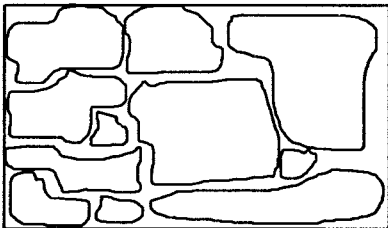


Figure 2.22

3.1.3 La matière première

A ce niveau, aucune considération mécanique sur la matière n'est prise en compte ici. Seules les dimensions significatives, c'est à dire les directions d'évolution du placement interviennent dans cette énumération. On distingue:

- Les matières à une dimension. C'est le cas de la découpe des barres d'acier de longueurs bornées ou, selon un autre point de vue, le cas l'ordonnement de tâches de durées variables.
- Les matières à deux dimensions. Les plaques à découper ont deux dimensions fixes (longueur et largeur bornées). C'est le cas des industries du bois et de l'acier.
- Les matières à trois dimensions: les objets à découper sont des solides qui ont trois dimensions significatives, par exemple la découpe de blocs isolants.

Lorsqu'une dimension n'est pas bornée, on supprime alors une 1/2 dimension significative. Ainsi le problème de découpe sur matière unique de longueur semi-infinie

(bornée à gauche, infinie à droite) est donc un problème à une dimension et demie. C'est le cas de notre étude.

Remarquons que jusqu'ici, la matière a toujours été considérée rectangulaire par hypothèse. Or cette classification serait vraiment incomplète s'il n'était nullement fait mention du cas où la matière est de forme quelconque, comme cela est rencontré dans l'industrie du cuir.

Notons enfin que le matériau peut être connu ou inconnu a priori dans sa forme et ses dimensions. Il est connu généralement dans les situations où ses propriétés sont immuables ou lorsqu'il a déjà été transformé (tôle, verre). Il est inconnu lorsque ses caractéristiques sont imprévisibles ou acquises en ligne (tissu, cuir, ..). Dans la présente étude, on considèrera que le matériau est globalement inconnu sur l'ensemble du placement, mais qui est complètement défini localement, c'est à dire au niveau des sous placements P_i .

3.1.4 La commande

On distingue les cas où la commande est :

- connue a priori. C'est le cas le plus courant.
- non connue a priori. Ceci se produit lorsqu'il est nécessaire de débiter la production d'une commande alors que celle-ci n'est pas entièrement connue. Dans ces conditions, on peut considérer la commande comme aléatoire avec des degrés d'urgence plus ou moins importants, comme le suppose notre étude.

On peut également faire la distinction entre :

- faible demande. Le nombre de pièces différentes et les effectifs sont petits. On peut alors considérer qu'il est possible de connaître les caractéristiques du lot de matière à débiter. La faible demande invite à choisir un algorithme de résolution exacte.
- forte demande. Le carnet de commande est composé d'une grande variété de formes, chacune avec un effectif important. Il est constant pour un ordre de coupe donné. L'algorithme d'optimisation doit alors souvent se contenter d'une solution sous optimale. Dans le cas où elle a des caractéristiques imprévisibles, la matière est probablement inconnue sur la majeure partie du lot à découper.

Peu d'études portent sur une classification des problèmes de découpe sous l'angle de la connaissance de la matière, ainsi que de la taille du carnet de commande. [DMI] semble être le premier à l'avoir examiné.

3.1.5 Les contraintes de placement

C'est à ce niveau que l'on fait apparaître les contraintes portant sur les formes à découper. On cite généralement :

- Les contraintes absolues par rapport à la matière (dimensions, défauts, obstacles, orientations ...)
- Les contraintes relatives entre les formes (disjonction, raccords, liaisons entre postes de travail, liaisons électriques entre circuits intégrés,..)

3.1.6 Les contraintes temporelles de cycle en ligne

Si la contrainte géométrique réduit l'espace des configurations admissibles et donc simplifie la résolution, en revanche la présence d'une contrainte temporelle ajoute une difficulté d'ordre algorithmique à la résolution.

Deux comportements sont possibles vis à vis de la prise en compte de la contrainte temporelle:

- L'algorithme cherche une solution sous optimale de qualité moyenne correcte en un temps quasiment constant par rapport à un contexte habituel.
- L'algorithme cherche une solution évolutive et il peut être interrompu dès qu'une sollicitation extérieure apparaît. Chaque solution intermédiaire peut être améliorée ou constituer la solution finale. Peu d'études font intervenir explicitement le temps dans les algorithmes. [DUN 84] et [BUR 85] ont exploité cette idée dans le domaine du placement-routage.

3.2 Synthèses bibliographiques

Les articles sur les problèmes de découpe sont légion si l'on prospecte au moyen des mots-clés : découpe (cutting stock), placement (layout), ordonnancement (assortment), bi-partitionnement (trim loss), rangement de boîtes (bin packing), pb du sac à dos (knapsack) etc...; au point de rencontrer quelques difficultés lorsqu'il s'agit de résoudre un problème classique. La profusion nuit parfois à l'éclaircissement.

Il n'est pas du rôle de ce mémoire de tenter de passer en revue toutes les études traitant de ce domaine. Il faudrait trop en dire, avec la crainte de ne pas en dire assez. Pour se frayer un chemin dans cette "jungle" bibliographique, on ne saurait que trop conseiller quelques bonnes analyses bibliographiques, faisant le point sur un art déjà très documenté.

La plus récente synthèse semble être le premier chapitre de la thèse de [DMI 91] qui propose une classification originale des problèmes de découpe au travers des combinaisons de volume de commande et de connaissance du matériau les plus rencontrées. Il propose lui

même une résolution du problème de découpe de panneaux de bois massif dans un contexte où la matière est de forme et de dimensions variables pour une forte demande connue à l'avance. Son approche est basée sur un algorithme à deux niveaux. Un niveau global de régulation dépendant de la commande, et un niveau local de placement dépendant de la matière présentée. Le niveau global guide le niveau local en lui indiquant les règles de placement à utiliser.

On retrouve, dans sa bibliographie, quelques références proposées par [SAV 87] qui s'est consacré à un survol des formulations et des méthodes de résolution. Cette étude est particulièrement attrayante par sa clarté et sa richesse en informations. Il propose en outre une classification plus classique, tenant compte des dimensions significatives et des contraintes de découpe. Pour un survol exhaustif des problèmes pratiques de découpe (en particulier les Pb 1D), il est conseillé de lire [HIN 80]. On citera aussi avec beaucoup d'intérêt les survols partiels effectués par [GOL 76], [MAD 80], [ISR 82], [PRE 83], [ZIS 84], [DAG 88]:

Golden présente trois formulations du problème unidimensionnel en s'attardant sur les algorithmes du type "rangement de boîtes" (bin packing) et finit en proposant une approche de résolution sur la méthode du sous gradient. Madsen présente une liste exhaustive de références sur le problème de découpe antérieur à 1980. Israni et Sanders présentent surtout le problème bidimensionnel ainsi que les principales méthodes de résolution qui lui sont associées. Ils terminent en proposant une méthode de résolution simple pouvant être facilement implanté sur microcalculateur.

Preempti a intégré un état de l'art très complet dans sa thèse sur les méthodes stochastiques dans les problèmes de placement. Il propose une bibliographie abondante, où une clé est attribuée à chaque référence. Cette clé est relative au type de problème, à la résolution et à l'industrie d'application. Zissimopoulos et Dagli font eux aussi de très bonnes synthèses sur le problème bidimensionnel. Enfin, un excellent tour d'horizon des problèmes de rangement est fait par Coffman, Garrey et Johnson [GAR 81] [Coff XX]. Les problèmes de sacs-à-dos (knapsack) sont aussi inventoriés par Martello et Toth [MAR XX].

3.3 Le Placement-découpe des formes irrégulières.

La résolution du problème de la découpe et en particulier du placement des formes irrégulières souffre d'une plus grande discrétion que celle du problème de la découpe guillotine ou orthogonale. Les raisons en sont multiples.

La raison essentielle est sans doute liée à l'état de la connaissance dans le domaine de la représentation des formes irrégulières et des méthodes d'optimisation. A la complexité combinatoire du problème, s'ajoute la complexité spatiale des formes mises en jeu. Deux données suffisent pour connaître les dimensions d'un rectangle. En revanche, la représentation

d'une forme quelconque, ne serait-ce par son contour, nécessite une méthode de description et une structuration des données en adéquation avec les outils mathématiques d'optimisation. L'obtention d'un placement est donc liée à la modélisation des formes dont la préoccupation est en partie avec l'avènement du traitement de l'image. A cela s'ajoute le fait que les outils d'optimisation linéaire à la disposition des mathématiciens sont peu adaptés dès qu'il s'agit de traiter des objets qui sont par définition non-linéaires. Sauf en réalisant une approximation avec le rectangle exinscrit (lequel se révèle être très approximatif lorsque les formes sont très découpées), le transfert des compétences en placement de rectangle, est donc a priori difficile. Quant aux méthodes d'optimisation combinatoires, elles sont finalement apparues relativement tard, car souvent associées aux progrès effectués en Intelligence Artificielle.

La deuxième raison est très certainement liée aux limites technologiques imposées, jusqu'à nos jours, par les calculateurs en terme de mémoire et de rapidité de calcul. Décrire 100 objets aux contours quelconques occupe plus de mémoire que les 200 nombres suffisants pour représenter 100 rectangles. De plus, les méthodes d'optimisation sont en général gourmandes en mémoire dès qu'elles génèrent des solutions intermédiaires multiples comme la programmation dynamique.

Enfin, il ne faut pas ignorer que la découpe est en réalité un problème industriel. Des solutions ont probablement été élaborées dans quelques laboratoires de recherche et de développement, mais soigneusement dissimulées par leur exploitant qui veulent légitimement conserver le privilège de l'exclusivité-- concurrence oblige --.

La spécificité du problème de découpe des formes irrégulières conduit donc à écarter une étude approfondie des méthodes de résolution du placement et découpe de rectangles mais invite en revanche à expliciter les méthodes qui sont apparues dans la bibliographie et portant sur notre centre d'intérêt: la découpe par contournage des formes irrégulière (DFI).

En général, trois familles sont proposées pour classer les méthodes de placement-découpe. On distingue les approches interactives, semi-automatiques et automatiques.

3.3.1 Placement interactif

Dans cette approche, un opérateur interagit avec le terminal d'ordinateur. Un dialogue s'instaure, permettant la description du carnet de commande, de la matière et des contraintes spécifiques à certaines formes. Un écran de visualisation est ensuite activé, présentant une fenêtre des types de pièces à placer et une fenêtre de placement. Un menu est accessible permettant d'actionner des fonctions de sélection de formes ou de placement (translation, rotation, symétrie, concaténation). Pour chaque opération géométrique, il connaît la répercussion sur le rendement du placement. Une assistance à l'opérateur est souvent associée à ce genre de logiciel. Elle évite les opérations interdites comme le chevauchement des

formes, le respect des raccords dans le cas des placements textiles à motifs ou le débordement de la matière. C'est essentiellement dans l'industrie du vêtement [CAT 76] [DRA 86] ou dans le placement de circuit intégré [VIR 76] que l'on rencontre le plus souvent ce principe. Les fonctions proposées sont souvent issues des techniques de D.A.O. et les formes sont généralement représentées par leur contour polygonal ou par une succession d'éléments de Bezier. Les transformations géométriques sont appliquées aux sommets des polygones ou aux points de contrôles des courbes de Bezier. Chaque modification du placement introduit une séquence de vérification. La modification est affichée si elle est acceptée. Un calcul est aussitôt effectué.

Il est également intéressant de citer [AVN 88.a], [AVN 88.b] et [AME 89] pour leurs travaux portant sur le placement ou le déplacement (en rotation et en translation sans chevauchement) de polygones dans un polygone. Cette étude est particulièrement adaptée pour la planification de trajectoire de robot, mais on pourrait adapter la théorie pour le placement interactif des formes quelconques dans une matière de forme quelconque. La complexité minimale de l'algorithme (de l'ordre de $m^3n^3 \log(m.n)$, m et n étant le nombre de sommets des deux polygones) imposerait cependant quelques simplifications pour considérer le problème à un stade automatique. Les possibilités offertes par cette théorie, pourrait en tous les cas, convenir au placement sur matière de forme quelconque et contenant des défauts de surface à éviter.

3.3.2 Placement automatique

Différentes écoles apparaissent au niveau des algorithmes utilisés pour fournir automatiquement une solution de placement. On distingue en effet les méthodes "two stage", les algorithmes déterministes appliquant séquentiellement des lois heuristiques de placement, les algorithmes non déterministes mais utilisant des heuristiques de recherche et enfin les systèmes experts.

3.3.2.1 Les méthodes à deux ou n étapes

Les premiers résultats avec cette technique sont dus à M.J. Haims. Ils apparaissent dans sa thèse [HAI 66] et ont été publiés dans [HAI 70]. Le problème est de découper au mieux, des formes quelconques dans un ensemble de rectangles de dimensions données avec surproduction possible. La méthode repose sur le principe suivant:

Chaque forme est initialement exinscrite dans un rectangle de surface minimal, appelé module. Ces premiers modules sont ensuite disposés de façon optimale sur la matière au moyen d'un algorithme de programmation dynamique. Par améliorations successives

(multistage solution), les formes sont ensuite agglomérées par groupe de deux ou trois (maximum huit) de telle façon que les nouveaux modules constitués aient une surface minimale. Les modules sont de nouveau disposés sur la matière en optimisant une fonction objectif. Il s'agit ici de la somme des coefficients économiques associés à chaque forme. Chaque coefficient peut représenter la surface, le prix de vente, le coût de découpe ou dépendre de l'emplacement sur la matière, des défauts ou de son orientation. Seules les fonctions linéaires sont considérées dans cette approche.

Ainsi, l'algorithme de programmation dynamique convertit le problème à multivariables en un problème à étapes multiples. Les itérations sont utilisées en fait pour déterminer si les modules d'ordre plus élevés (contenant plus de formes complexes) améliorent la solution.

La même philosophie est retrouvée chez M. Adamowicz dont les résultats ont été publiés dans [ADA 71]. L'algorithme repose également sur la construction de modules rectangulaires suivie d'un placement de rectangles. Mais la comparaison s'arrête là, car les méthodes employées sont différentes. Premièrement, les modules sont créés préalablement au placement de rectangles et sont définitifs. D'autre part la construction de ces modules est basée sur la recherche du meilleur rendement, en essayant différentes options de groupement:

étape 0 : choisir une forme.

étape 1 : si forme rectangulaire alors module terminé.

étape 2 : si forme quelconque alors déterminer rectangle exinscrit.

étape 3 :

Si rectangle exinscrit satisfaisant (rendement correct) alors module terminé.
Aller en 0.

Sinon grouper la forme avec une forme du même type, orientée à 180° et déterminer le rectangle exinscrit.

étape 4 :

Si groupement satisfaisant alors module terminé, aller en 0.

Sinon aller en 5.

étape 5 :

reprendre la forme de l'*étape 2* et grouper avec quatre petites formes.

Si groupement satisfaisant alors module terminé, aller en 0

Sinon aller en 6

étape 6 :

Reprendre le groupement de l'étape 3 et le compacter avec quatre petites formes.

Si groupement satisfaisant alors module terminé, aller en 0.

Sinon groupement obtenu en 3 considéré satisfaisant, module terminé, aller en 1.

Chaque opération de groupement s'effectue par la technique du No Fit Polygon développée dans [ADA 69], générant le polygone décrit par un point d'une forme B lorsque celle-ci fait le tour d'une forme A sans la chevaucher. La meilleure position relative est donnée par le rectangle exinscrit minimal.

Deuxièmement, la technique du placement de rectangles reprend la méthode développée par les mêmes auteurs dans d'autres articles [ADA 70 a], [ADA 70 b] et [ADA 70 c]. Elle consiste à construire des bandes homogènes de longueur inférieure ou égale à la matière, constituées avec des modules identiques. Un algorithme de programmation dynamique permet ensuite de choisir les bandes de façon à optimiser une fonction-objectif de remplissage de la matière.

Cet algorithme s'adapte particulièrement bien aux carnets de commande à forte demande, sans surproduction, composé d'une quantité importante de formes du même type. La présence de petites formes améliore grandement le rendement.

Une version modifiée de l'algorithme est apparue dans [ADA 76.a] simplifiant la recherche des modules. Dans [ADA 76.b], la modification de l'algorithme de placement de rectangle permet de récupérer les chutes en bordure de la matière.

3.3.2.2 Les algorithmes déterministes

L'aspect commun des études suivantes porte simplement sur le fait que chacune d'elles est basée sur un ensemble de lois heuristiques fondées sur des techniques de placement séquentielles et déterministes. L'énumération suivante est classée par ordre chronologique:

[ART 66] décrit une solution heuristique au problème du placement textile (Marker making), sans formalisme mathématique d'optimisation. R.C. ART introduit la notion de rectangle exinscrit dont deux des côtés sont parallèles au droit fil. Les contours sont simplifiés par un polygone convexe. {article non disponible}

[GUR 69] expose une formulation du problème à partir de la théorie des graphes. O. GÜREL créa pour la circonstance les "graphes circulaires". {article non disponible}

[MOR 69] {article non disponible}

[DIV 73] présente une méthode originale de placement à partir d'un classement hiérarchique déterminé à l'aide de certaines notions topologiques. A partir de la définition d'un indice linéaire de voisinage calculé pour tous les couples de formes, une heuristique consiste alors à déterminer un multicycle hamiltonien minimal visant à donner l'ordre de la suite des index des formes à placer.

[MOR 73] propose une heuristique de remplissage de la matière en commençant par les extrémités gauches et droites et continue par l'intérieur du coupon. Après avoir simplifié leur contour, les formes à placer sont caractérisées par un ensemble de 18 paramètres tels que les "extensions" (c'est à dire la largeur et la longueur du rectangle exinscrit), les huit points de tangences avec les côtés du rectangle, les huit surfaces comprises entre la forme et le rectangle exinscrit. Ces paramètres servent dans un premier temps à classer les formes en trois familles (grandes, moyennes, petites) et ensuite permettent de sélectionner les formes au moment du placement. L'heuristique de placement est inspirée très fortement des pratiques manuelles empiriques des placeurs en textile, et qui sont expliquées dans [SIL 64]. L'idée consiste donc à construire des "colonnes" compactes, en commençant par les colonnes frontières (droite et gauche) du coupon. Seules les grandes et moyennes pièces sont placées dans cet algorithme. Pour construire les colonnes, on commence par choisir deux formes qui n'occasionnent pas trop de pertes surfaciques en bordure, afin de positionner l'une en haut et l'autre en bas. Ensuite, on remplit la surface intermédiaire afin de réaliser une colonne relativement homogène et présentant la perte minimale. Le remplissage vertical est achevé lorsque le jeu final n'est plus assez important pour accueillir une autre forme. Localement, deux indices permettent de déterminer les positions relatives entre chaque formes en contacts. Il s'agit ici en fait de maximiser "l'emboîtement" vertical, ainsi que la longueur de la zone horizontale commune. Lorsqu'une colonne est constituée, on la place en contact avec la frontière gauche ou droite selon la qualité de l'emboîtement horizontal. Une nouvelle frontière est ainsi créée, et la procédure est réitérée jusqu'à épuisement des formes. A la fin, la frontière droite est mise en contact avec la frontière gauche.

[TAN 73] utilise à la base une technique de détermination de l'enveloppe de la réunion des formes lorsque celles-ci occupent une position donnée. Les positions essayées, ainsi que les opérations effectuées (translation, rotation, symétries), sont définies par des règles communes aux experts-placeurs. Globalement, l'algorithme de placement détermine successivement la forme à ajouter qui minimise un critère de surface. La forme initiale étant celle de plus grande surface.

[FRE 74] explicite un algorithme fondé sur une optimisation locale itérative. FREEMAN fait l'hypothèse qu'en associant une à une les formes restantes à placer en minimisant systématiquement une fonction surfacique, alors l'amalgame résultant est une

solution proche de l'optimum global. Le problème se réduit alors à une optimisation locale (en translatant les deux formes l'une contre l'autre), dont la complexité est liée à la finesse de la discrétisation lors de la description du contour. Cette méthode est évidemment temporellement sensible à la richesse du domaine de commande, ainsi qu'au nombre important de formes. En revanche, elle est facilement généralisable au cas des matières de formes quelconques. La qualité des solutions dépend beaucoup du critère utilisé, bien que les objectifs soient les mêmes (cf 2.2)

Dans [SOE 77], l'auteur développe un algorithme également basé sur une bifocalisation du problème. L'approche locale consiste à essayer tous les couples de pièces pour calculer l'occupation surfacique du rectangle exinscrit de chaque module potentiel. Ceci est effectué par comparaison matricielle, en translatant une des deux formes, afin de déterminer le positionnement relatif optimal. Toutes les combinaisons de couples sont essayées. Ensuite, l'idée consiste à créer des bandes en groupant les modules dans le sens de la largeur. A chaque itération on choisit deux modules différents d'imbrication maximale. Si l'association des deux modules a une largeur inférieure à la matière, celle-ci est acceptée. Sinon, elle est rejetée et chaque module participera à l'élaboration d'une autre bande. De plus, R. SOENEN présente plusieurs modèles précis de formulation de placement, permettant selon la nature du processus de découpe, d'établir une commande acceptable sur le plan des contraintes liées à la matière ainsi qu'au niveau des caractéristiques de la demande.

[CHO 79] montre une méthode de construction de bandes homogènes. Chaque bande est constituée d'un type unique de pièces, lesquelles peuvent être orientées modulo 180° ou symétriquement par rapport à leur position d'origine.

[MAO 85] établit un indice de voisinage surfacique entre chaque couple de formes. Il évalue cet indice en calculant la surface d'imbrication surfacique favorisant le remplissage des creux par les petites formes. Ces indices sont calculés en hors ligne pour quelques positions relatives essentielles. Un classement hiérarchique est alors effectué par ordre d'indice décroissant. Des bandes sont ensuite construites en juxtaposant les formes dans la direction transversale, de telle façon que chaque indice correspondant aux formes connexes soit maximal, c'est à dire en tête du classement hiérarchique. Une précaution est cependant prise afin d'associer en priorité les formes de longueur homogène.

[DEL 89] présente une approche de la résolution à différents niveaux de perception: local et global. Il propose donc de résoudre les problèmes de l'association des formes, de l'évaluation de la qualité et de la construction des placements au moyen de trois heuristiques utilisant des critères multiples. Les quatre méthodes d'association sont basées sur l'utilisation des paramètres descriptifs des contours. Ces heuristiques constituent alors une base de règles qu'il est possible d'activer par un algorithme de décision propre au contexte industriel.

3.3.2.3 Les algorithmes non déterministes

Ces algorithmes élaborent des configurations pouvant être remises en cause après chaque évaluation du placement final ou intermédiaire. Cette possibilité de "retour en arrière" n'est possible que si l'espace des configurations potentielles est convenablement codé. La recherche se traduit alors souvent par une exploration du graphe d'états, ou éventuellement par une programmation linéaire moyennant une linéarisation des objets manipulés. Si l'espace des configurations est considéré dans son ensemble, c'est à dire que tous les états sont normalement accessibles, l'exploration est, cependant, très souvent "canalisée", afin d'éviter le piège de l'explosion combinatoire. Généralement, la recherche est contrôlée par une heuristique, limitant les explorations peu prometteuses. Dans cette classe, on trouvera donc les méthodes de programmation linéaire, les parcours de graphe et les méthodes stochastiques.

[ADA 69] et [ADA 71.b] exposent une méthode qui améliore itérativement une solution à l'aide d'une programmation linéaire. Le principe est encore basé sur la construction de modules, mais leur organisation peut évoluer au cours de la recherche. Cependant le programme expérimental n'était pas complètement opérationnel et les performances en temps, ainsi que la qualité des solutions, ne sont pas connues.

A. ALBANO et G. SAPUPPO suggèrent dans [ALB 80] une approche où le problème du placement est transformé en une recherche dans un graphe d'états. Chaque état intermédiaire correspond à un placement inachevé et chaque feuille de "l'arbre" représente un placement terminé, c'est à dire contenant toutes les formes à placer. L'arbre des solutions est exploré selon la technique du "meilleur d'abord", guidé par une heuristique d'estimation du coût. Celle-ci est composée de deux termes:

- La chute au noeud courant,
- La chute supplémentaire estimée pour atteindre l'état final.

Afin de limiter encore les combinaisons peu prometteuses, un ensemble de filtres éliminent certains chemins. Un premier filtre n'autorise que les orientations de pièces favorisant le remplissage à gauche. Un deuxième filtre ne conserve qu'une partie paramétrable des états précédents et ne laisse que ceux favorisant l'introduction des plus grandes pièces. Les états retenus sont développés. Etant tous de même niveau N, on ne retiendra que ceux dont les successeurs aux niveaux inférieurs (dont la profondeur est paramétrable) permettent d'atteindre un coût estimé minimal. Une procédure de simplification de contour est encore appliquée, afin de supprimer les configurations peu envisageables.

[KUN 81] propose de résoudre un problème de "sac à dos" unidimensionnel pour sélectionner les modules à placer sur la matière. Même si ce sont des modules rectangulaires

qui sont d'abord formés, c'est le contour réel de ces modules qui est utilisé lors de la phase de placement. Ces modules sont mémorisés pour construire une bibliothèque qui servira ensuite pour déterminer la composition d'un placement selon un programme linéaire. Le but est ici de réaliser des placement répondant au principe du matelassage.

[LE BOR 82] est le rapport d'un projet de fin d'études portant sur la réalisation d'un logiciel de découpe de tôle: RIMBAUD "logiciel de Recherche d'IMBrication AUtomatique pour la Découpe". Il traite des formes irrégulières qu'il faut placer sur des tôles de dimensions fixées. Selon la bibliographie de [PRE 83], il associe une recherche par la technique du Branch and Bound, une heuristique et une optimisation du type sac à dos. Depuis sa conception, le logiciel a été commercialisé.

[PRE 83] est la première étude incluant les méthodes stochastiques pour la DFI. Basé sur le principe du "recuit simulé" (voir chapitre 5), le placement est caractérisé par une énergie dont les termes expriment la quantité de matière utilisée, le chevauchement et les débordements. Au bout d'un certain nombre de changement de configurations du système, l'énergie devient faible et sa surface occupée proche de l'optimum. Mais les contraintes sont rarement entièrement vérifiées (chevauchement et débordement résiduels) et il faudrait faire fonctionner l'algorithme très longtemps pour espérer atteindre une solution admissible. Au lieu de cela, Prempti préconise de limiter la recherche et d'améliorer la solution obtenue à l'aide d'une heuristique à actions locales. L'auteur indique également comment améliorer les placements en introduisant des aléas à différents niveaux d'un algorithme déterministe.

[DAG 90] aborde la même approche que Prempti, mais a développé l'algorithme selon une architecture de type parallèle, afin de générer simultanément plusieurs perturbations différentes à une même configuration. La fonction-énergie utilisée est une composition de la surface de recouvrement, de la surface du rectangle exinscrit au placement, et de la somme des distances entre tous les centres de gravité des formes. Aucun résultat n'est cependant présenté.

[KUN XX] est une présentation de quelques placements automatisés de pièces de formes irrégulières. La notion de module est traitée ainsi que l'approche "stratège-tacticien". De plus, une courte description des méthodes aléatoires est donnée, consistant à faire plusieurs essais en perturbant aléatoirement quelques caractéristiques.

3.3.2.4 Les systèmes experts

Ces systèmes fonctionnent à l'aide d'une base de règles définies par le concepteur, lequel modélise sa connaissance ou celle d'un expert placeur. Même si les perspectives sont attrayantes, les réalisations sont cependant assez rares, car la difficulté est double : déterminer les bonnes règles et également limiter la recherche, sensible à une combinatoire toujours présente. Un ensemble d'heuristiques est donc toujours nécessaire.

[PER 78] est une étude portant essentiellement sur l'optimisation de l'espace. Le système réalisé par L. M. Pereira intègre les contraintes du type adjacence, visibilité et distance. Les contraintes sont attachées à chaque objet et ne sont donc pas communes à l'ensemble des pièces. Ce qui limite assez fortement les possibilités du système.

[CLE 87] élimine les limitations précédentes, mais ne présente qu'une résolution avec des formes rectangulaires. Dans cette approche, l'ensemble des contraintes définit un espace de solutions qu'il faut explorer. Les prédicats utilisés permettent une modélisation simple des objets à placer ainsi que l'obtention d'une expression unifiée des contraintes. Ensuite les formes sont placées les unes après les autres selon des critères non géométriques, tout en respectant la contrainte précédente. L'ordonnement des formes est guidé par un tri multi-critères. Les résultats sont obtenus à partir d'un prototype écrit en Prolog. Mais les auteurs préconisent l'emploi d'un générateur de contraintes spécifiques à l'application, afin de réduire l'espace et accélérer la recherche, en limitant le nombre de "retours arrière".

3.3.3 Placement semi-automatique

Dans l'approche semi-automatique, on vise à dépasser les limites du placement automatique, en particulier lorsque celui-ci est élaboré à l'aide d'une procédure déterministe. Dans ces systèmes, une tentative de solution est d'abord proposée par un algorithme automatique. Des améliorations sont ensuite possibles à l'aide des fonctionnalités identiques à celles disponibles sur les systèmes interactifs. [KUN 81], [DRA 86] et [ALB 77] semblent être les seules publications dédiées à ce genre de techniques. Dans [ALB 77], la solution automatique est obtenue à l'aide de l'algorithme défini dans [ADA 71.a]. Les fonctions graphiques accessibles sont celles qui sont décrites dans [ADA 76].

IV Conclusion

Ce chapitre a eu pour objet de présenter le cahier des charges du système de placement automatique de patrons de vêtements, tel qu'on le conçoit selon les motivations exposées au chapitre 1. On s'est imposé un formalisme précis pour lever toute ambiguïté d'interprétations et des choix ont été effectués concernant la méthode générale de travail et en particulier le placement en bande.

Mais au-delà des spécificités ajoutées par le contexte textile, de nombreuses questions plus fondamentales restent posées concernant l'algorithme d'optimisation à utiliser pour résoudre ce problème. Quelle méthode choisir pour minimiser un critère économique, et par ailleurs, quel est le critère qui correspond au mieux à la méthode sélectionnée? Pour tenter de répondre à ces questions, une synthèse des fonctions-objectif rencontrées dans la littérature est d'abord proposée. Ensuite il semblait intéressant d'aiguiser nos connaissances sur l'art du placement automatique, en faisant un inventaire des méthodes déjà expérimentées.

Même si les méthodes utilisées présentent parfois des performances peu adaptées à notre nouvelle réalité industrielle, les commentaires associées à l'explication de ces expériences se veulent volontairement peu critiques. En effet, les algorithmes sont généralement mis au point pour une application particulière où ils conviennent souvent très bien. De plus, il est difficile d'établir une comparaison objective sans disposer d'un exemple commun et enfin les performances présentées sont généralement peu significatives lorsqu'elles sont exprimées en terme de temps de calcul et non de complexité. L'évolution des calculateurs a effacé tous les repères temporels fiables.

Tout au long de ce chapitre, il est fait allusion à la difficulté supplémentaire apportée par la nature irrégulière des formes. Une brève analyse fait vite apparaître que peu de méthodes différentes sont utilisées pour représenter les formes quelconques. Au point de se demander si cette absence de théorie ne fixe pas des limites contraignantes aux algorithmes plus élaborés que l'on peut imaginer. C'est dans l'esprit d'élargir le champ de perspectives, qu'on se propose dans le chapitre III, d'étudier une nouvelle méthode de modélisations des formes et dans le chapitre IV de proposer les opérations géométriques associés.

CHAPITRE III: REPRESENTATION NUMERIQUE DES FORMES IRREGULIERES

On doit utiliser les modèles, mais ne pas y
croire

Henri Teil

I Introduction

Dans le problème du placement automatique des formes irrégulières, on est amené à se poser la question essentielle de la représentation des objets à placer. Comment passer d'un espace où un placeur humain manipule des gabarits en carton à une représentation numérique, seul aliment comestible pour un calculateur? Derrière cette problématique jaillit un faisceau de questions auxquelles nous proposons de répondre dans ce chapitre.

La première d'entre elles concerne la saisie de la forme. Dans ce cadre, nous nous limiterons à la perception du contour, information suffisante à la connaissance de l'objet à placer. Dans la suite de ce chapitre, nous considérons que le contour est connu, car déjà introduit dans la mémoire du calculateur au moyen d'un éditeur de contour. Selon l'approche intégrée présentée dans le chapitre 1, le contour peut également être extrait de la base de données associée au système de C.A.O.. Cet aspect du problème ne correspondant pas au coeur de nos préoccupations, nous passerons volontairement sous silence la description d'un tel dispositif de tracé de contour, bien connu en l'occurrence. C'est donc la définition de la forme, évoquée au § 2.13 du chapitre II qui sera le point de départ de l'étude.

En revanche, nous ne pouvons ignorer la façon de modéliser l'information reçue, qui est le contour. La première étape de notre exposé consistera donc à éclaircir le point de vue concernant la discrétisation du contour. Comment passer d'un espace continu tel qu'on le voit dans la réalité ou que l'on croit le voir sur un moniteur d'ordinateur, à la représentation dans l'espace discret ? Nous donnerons en guise de réponse, les hypothèses classiques de travail en ce qui concerne la discrétisation des contours et nous expliciterons le modèle de visualisation utilisé.

La question suivante consiste à choisir le modèle de traitement qui assure une bonne adéquation entre l'accès aux données, les opérateurs à appliquer et les objectifs du traitement. On parlera alors de codage. A ce titre, on présentera les propriétés générales d'un code qui sont les critères sur lesquels on peut se baser pour établir un choix. Aussi passerons-nous en revue les grandes familles de codes avant de présenter une méthode originale pour décrire le contour des formes que nous traiterons. Outre quelques définitions et le principe d'obtention donné en annexe, nous en donnerons les avantages et les inconvénients.

Nous continuerons en proposant les approximations possibles de notre représentation en vue de réduire encore l'information moyennant une dégradation acceptable de la qualité de la description.

Enfin, nous terminerons en présentant les opérateurs géométriques agissant sur une forme, c'est à dire en particulier la rotation, la symétrie et la notion de forme duale.

II Représentation d'un contour dans l'espace discret

Les transformations des objets d'un espace continu vers un espace discret nécessitent d'émettre un certain nombre d'hypothèses sur lesquelles on s'appuiera tout au long de ce chapitre.

2.1 Support d'image discrète.

L'image est discrétisée spatialement relativement à la définition géométrique du pixel, caractérisé ici par le pavage carré. Chaque pixel se voit associé un couple (i,j) indiquant les coordonnées du pixel dans l'image, relativement au maillage associé. Le maillage carré a été retenu car il correspond au codage utilisé par la plupart des capteurs et moniteurs vidéos.

Une forme F_k peut être représentée, soit réellement sur le plan image, soit virtuellement sur le plan mémoire. Dans les deux cas, on peut associer une représentation matricielle Σ de dimension $N \times M$, où N et M sont respectivement le nombre de lignes et de colonnes de la matrice-image. **(Figure 3.1)**

L'intensité lumineuse de chaque pixel peut s'exprimer sous la forme d'une fonction de niveau de gris à deux variables discrètes :

$$I = F(i, j) \quad , \quad \text{où } 0 \leq i \leq M, 0 \leq j \leq N \quad \text{et} \quad I \in [0, I_{\max}]$$

Notre objectif se limitant à la description du contour, l'image binaire suffit pour délimiter les formes. La quantification du niveau de gris de chaque pixel est segmentée et I prendra ses modalités dans l'ensemble binaire $\{0,1\}$.

$$I \in \{0,1\}$$

La matrice de support d'image discrète ne contiendra donc que des 0 et des 1.

Nous invitons le lecteur à se reporter à l'annexe A pour un rappel des informations concernant la topologie de l'espace discret [BON 90], [CHA 91].

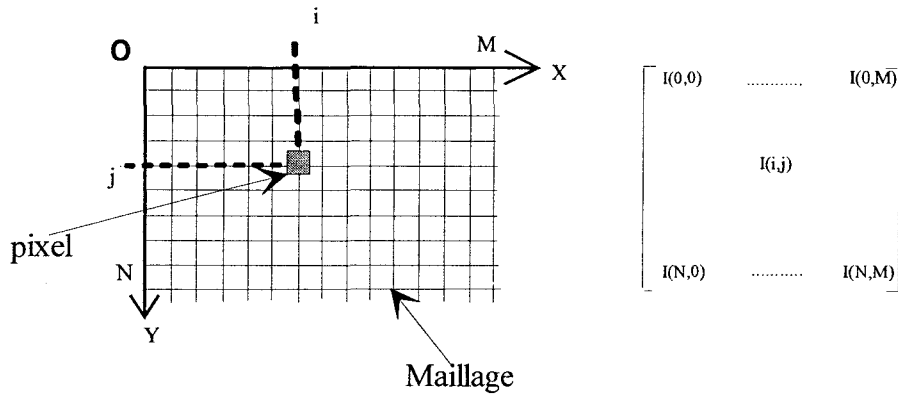


Figure 3.1

2.2 Objet de référence: la frontière discrète

Nous définirons le contour d'une forme F_k comme étant une courbe 8-connexe appartenant à la forme, séparant deux composantes 4-connexes, l'intérieur et le fond. Chaque élément de la courbe discrète est ordonné dans une liste C_k telle que:

$$C_k = ((x_0, y_0), (x_1, y_1), \dots, (x_q, y_q), \dots, (x_n, y_n))$$

Où $\forall 0 \leq q \leq n$, (x_q, y_q) est le couple de coordonnées du point π_q dans le repère lié à Σ . C'est cette liste de points codant la *frontière discrète* qui constituera notre référence de travail et constituera en même temps le modèle de représentation visuel attaché à l'objet-forme.

2.2.1 Intersection des contours

Nous supposons ici que les points constituant la frontière discrète appartiennent à la région de la forme que l'on a à décrire. Si bien que lorsque deux formes sont en contact, elles peuvent mettre en commun un ou plusieurs points de leur frontière respective. L'intersection est donc constituée des points de contact, mais les régions intérieures aux contours restent disjointes.

2.2.2 Interpolation des points de la frontière discrète

Cependant, selon la méthode d'édition des formes, il se peut que le contour obtenu soit du type polygonal, réduit aux seuls sommets des segments qui le composent. Dans ce cas, il conviendra

de réaliser une interpolation discrète de tous les points intermédiaires afin d'obtenir un contour 8-connexe. Un tel algorithme est présenté dans [DOR 84]. Il présente l'avantage, lors de la discrétisation, de satisfaire la contrainte d'unilatéralité des pixels par rapport au segment analogique, cette propriété étant nécessaire pour assurer la disjonction des formes lors du placement. Les points créés par l'algorithme d'interpolation doivent se situer sur le contour analogique (ou à l'extérieur), afin que lorsque deux formes sont en contact, l'intersection des régions des formes soit d'intérieur vide.

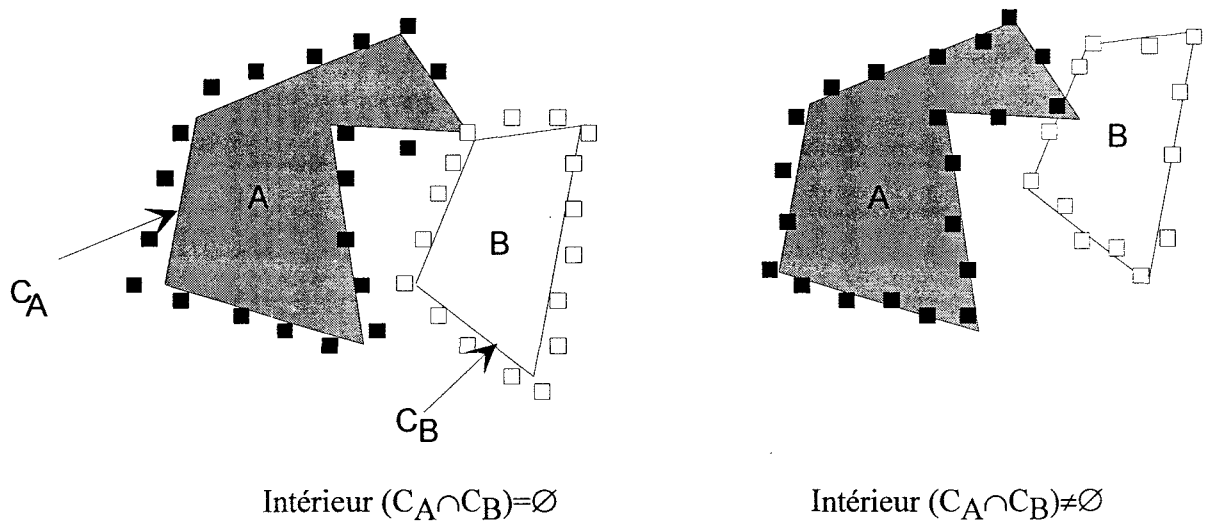


Figure 3.2: Usage du principe d'unilatéralité de la discrétisation pour le respect de la contrainte de disjonction

III Description des formes

Si le modèle de visualisation établi précédemment constitue un codage de base pour une utilisation à des fins d'affichage, en revanche il ne présente pas une structuration adéquate pour effectuer le traitement géométrique des contours. En conséquence, on est amené à transformer la représentation du contour. En théorie de l'information, on parle de codage.

3.1 Les critères d'analyse

La mise en forme d'un codage suscite l'analyse des critères suivants [FRE 74] :

- **La réversibilité de la transformation** : Elle qualifie le pouvoir de conservation de l'information en général, ou celle que l'on désire utiliser en particulier. Elle renseigne sur la nature du code, à savoir s'il s'agit d'un code exact ou si de l'information est perdue. Notons que le code source est un codage exact de la frontière d'une forme. Pour notre application, on peut raisonnablement accepter une approximation du

contour dans le code cible, pourvu que l'information déformée ou perdue n'altère pas la satisfaction des contraintes énoncées au chapitre II.

On pourra en revanche tolérer une légère dégradation de la qualité du placement caractérisé par un rendement éventuellement plus faible. Dans ce but, on peut déjà énoncer la règle suivante: *Les contraintes du problème du placement, telles que la disjonction des formes, la contrainte d'inclusion à la matière, la contrainte de raccord et la contrainte de prise en compte des défauts, ne seront pas perturbées si le contour résultant de la transformation tend à se rapprocher du polygone convexe du contour initial.* Cette règle introduit dès maintenant l'importance que l'on pourra accorder à la notion de contour convexe. Corrélativement à l'utilisation éventuelle d'une approximation du contour, le stockage en mémoire du modèle de visualisation est alors nécessaire. La donnée du contour exact de chaque forme à découper est obligatoire en sortie d'algorithme. Le couple modèle de visualisation et modèle de traitement apparaît alors comme une entité indissociable.

- *Le taux de compression* : Il exprime le rapport entre la quantité d'information de la représentation cible et la représentation initiale. Sa valeur sera représentative de la réduction d'occupation mémoire. Mais si une bonne compression de l'information optimise le stockage, elle est parfois responsable d'une augmentation du traitement pour extraire l'information utile. Une forte compression est donc à considérer avec prudence quand la représentation cible sert de modèle de traitement. De plus, l'évolution des calculateurs reculant sans cesse les limites des espaces mémoires utilisables, la compression des données devient pour ce genre d'application un gage de lourdeur de traitement. L'antagonisme classique entre l'espace occupé et le temps de traitement sera une fois de plus mis en apparence dans ce chapitre.

- *L'adéquation aux opérateurs de traitement* : Le code généré doit être adapté aux traitements mathématiques et géométriques de l'application visée. Dans le cadre du placement de formes, les opérations de commande des formes sont nombreuses. On citera en l'occurrence la rotation, la symétrie, la réunion, la recherche de complémentarité locale, la caractérisation des portions de contour. Dans notre cas, il convient de tirer parti de la contrainte du droit fil et donc de la réduction de l'ensemble des orientations pour simplifier le traitement ou en accroître les possibilités. Mais cette spécificité se traduira ici par une perte de généralité, puisque la description proposée n'est pas invariante par rotation.

3.2 Les codes de description de contours

Un petit tour d'horizon des codes de description des contours permettra de montrer l'intérêt de tel ou tel principe, et de faire apparaître éventuellement les inconvénients qui nous ont poussé à proposer une autre méthode. L'annexe B reprend plus en détail les différents types de codes rencontrés le plus couramment.

Nous y distinguons les codes exacts et les approximations du contour.

3.2.1 Les codes exacts

Parmi les codes exacts, on ne peut ignorer le *Code de FREEMAN* [FRE 61] ainsi que ses dérivés. De nombreux algorithmes ont été développés pour réaliser les traitements géométriques classiques, rotation d'un angle quelconque, union de deux contours, intersection, homothétie, convexification, mise en correspondance etc. Mais l'opération d'évaluation de complémentarité, noyau de l'algorithme de placement, semble engendrer une complexité de calcul trop importante pour constituer une procédure à utilisation fréquente.

Une autre représentation, appelée *Codage par transition* [HUA 75], permet d'obtenir aussi une bonne compression de l'information. Mais cette technique n'offre qu'un nombre limité d'opérateurs géométriques.

Signalons enfin dans cette catégorie, la *signature polaire* qui, en dépit d'une grande richesse d'utilisation possible, présente néanmoins de sérieux inconvénients, d'une part au niveau de la représentation des contours quelconques, et d'autre part au niveau de son adaptation au contour discret.

3.2.2 Les approximations de contour

Les approximations de contour sont également nombreuses. Les *Nombres de formes* [BRI 79] sont intéressants pour mesurer la similarité de deux formes, mais moins efficace dès qu'il s'agit de réaliser la concaténation. Les *Strip trees* [BAL 81] sont une représentation hiérarchique du contour d'une forme, dont la résolution est contrôlable. Cette méthode semble être une alternative possible pour coder les formes quelconques dans le cadre du placement constructif.

Les *Squelettes de formes* [MON 69] permettent d'extraire assez facilement les parties essentielles d'une forme quelconque (creux, bosses, etc.), mais ils se heurtent à une absence de primitive de traitement géométrique.

N'oublions pas les *Descripteurs* de FOURRIER [ZAH 72] ou les approximations polynomiales, qui nécessitent malheureusement une lourdeur de calcul inadéquate avec le principe de construction itératif.

Enfin, insistons un peu plus sur la méthode de LIU et THONNAT [LIU 91] qui utilise le *Contour apparent* pour déterminer la *déficiencce convexe* en vue de caractériser les contours lobés ou épineux des foraminifères planctoniques en micropaléontologie.

3.2.3 Détermination du contour apparent

Ce principe de codage d'une courbe discrète C (pas obligatoirement connexe) consiste à constituer initialement quatre listes: $C_{min}^x, C_{max}^x, C_{min}^y, C_{max}^y$. Ces listes contiennent les points sélectionnés par les relations:

$$\left. \begin{aligned} C_{min}^x &= \left\{ p \mid x_p = \underset{p' \in C, y_{p'} = y_p}{MIN} x_{p'} \right\} \\ C_{max}^x &= \left\{ p \mid x_p = \underset{p' \in C, y_{p'} = y_p}{MAX} x_{p'} \right\} \end{aligned} \right\} \text{obtenues par balayage sur } y$$

$$\left. \begin{aligned} C_{min}^y &= \left\{ p \mid y_p = \underset{p' \in C, x_{p'} = x_p}{MIN} y_{p'} \right\} \\ C_{max}^y &= \left\{ p \mid y_p = \underset{p' \in C, x_{p'} = x_p}{MAX} y_{p'} \right\} \end{aligned} \right\} \text{obtenues par balayage sur } x$$

On calcule ensuite:

$$T = C_{min}^x \cup C_{max}^x \cup C_{min}^y \cup C_{max}^y$$

On s'arrange ensuite pour reconstituer un contour polygonal 8-connexe, en raccordant au mieux les ruptures du contour T . On appelle alors $A(c)$ le contour apparent obtenu en parcourant P dans le sens horaire à partir du point extrême gauche. On réitère ensuite la même procédure sans être obligé de connecter le contour obtenu pour le rendre connexe. Ce contour est alors appelé contour apparent strict $\tilde{A}(c)$. On détermine ensuite la déficiencce convexe, qui est le complémentaire du contour apparent strict par rapport à son enveloppe convexe. On obtient un ensemble de segments constituant la courbe de distance (**Figure 3.3.d**)

Cette méthode méritait d'être introduite car nous avons trouvé là un parallèle intéressant à la méthode que nous avons élaboré et que nous présentons dans la suite de ce chapitre.

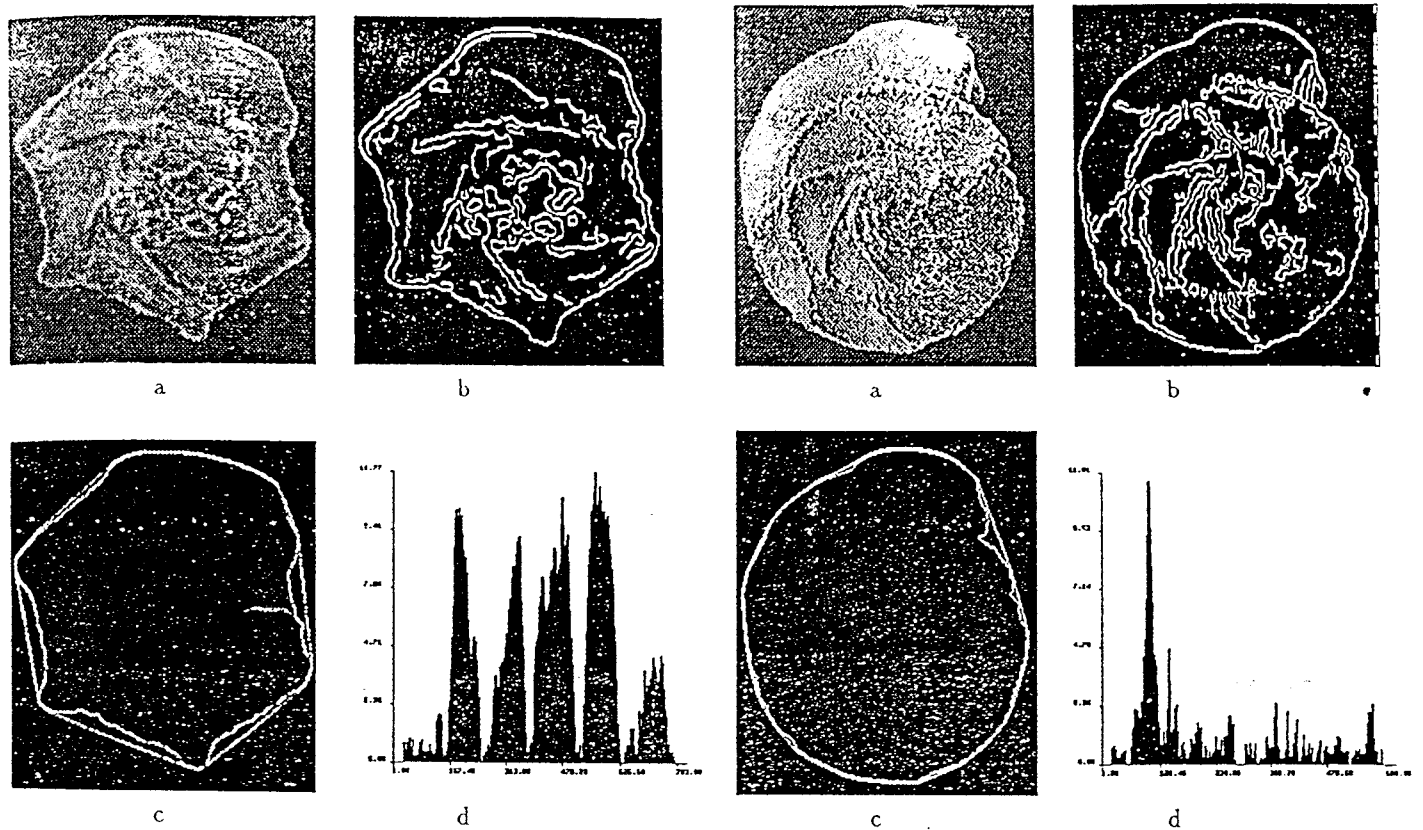


Figure 3.3: (a) Un foraminifère de l'espèce *Globotruncanita calcarata*. (b) Résultat du détecteur de Deriche suivi par un seuillage par hystérésis. (c) Contour apparent et enveloppe convexe. (d) Courbe des distances de la déficience convexe.

La figure 3.3 donne une illustration de la méthode de codage appliquées aux formes irrégulières. La courbe des déficiences convexes permet, en extrayant les modes de la distribution, de caractériser les creux et les bosses des formes traitées.

IV Description du contour par les " Peignes de formes "

4.1 Rectangle circonscrit

La recherche du rectangle circonscrit n'est envisageable qu'en fonction du choix d'une direction privilégiée de l'axe d'inertie du rectangle. En effet, il existe une infinité de rectangles circonscrits pour une forme donnée. Ce choix s'accompagne donc d'une relation pouvant être :

- l'axe principal du rectangle parallèle à un axe arbitraire de la forme (droit fil par exemple, cf. figure 3.4). C'est le cas de notre problème. L'axe du droit fil étant choisi parallèle à l'axe OX, le rectangle circonscrit est dit "*paraxial au repère*".
- axe principal du rectangle parallèle à l'axe principal d'inertie de la forme

- rectangle circonscrit orienté de telle façon que l'élongation soit maximale [FRE75].

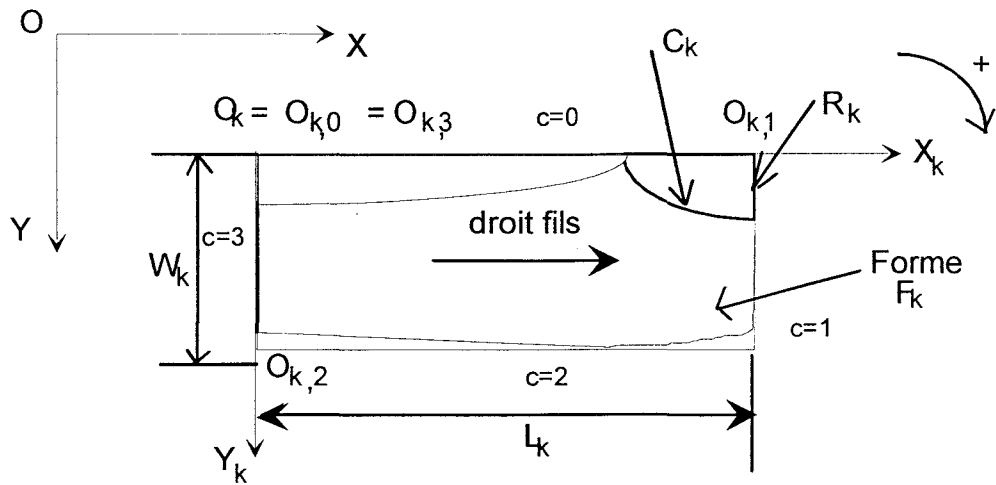


Figure 3.4.a

Dans la figure 3.4.a, on a numéroté les côtés dans le sens horaire, à partir du côté supérieur. On définit ensuite les origines associées à chaque côté comme cela est indiqué sur cette figure. Cette disposition permet d'avoir les axes $O_c X_c$ parallèles à OX si c est pair, et à OY si c est impair. Les sens sont identiques aux axes OX , respectivement OY . (cf. figure 3.4.b)

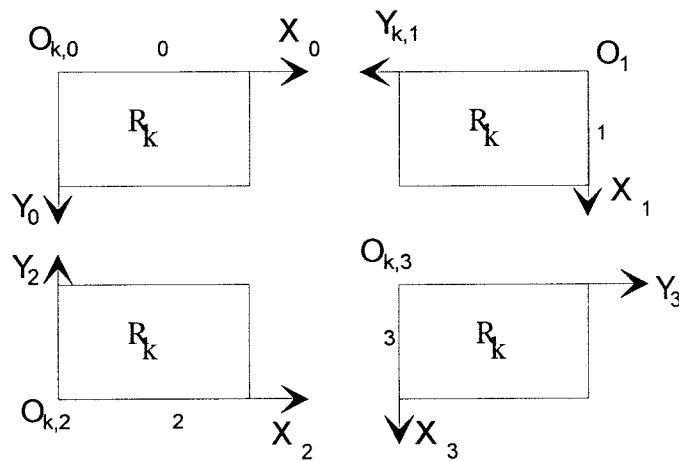


Figure 3.4.b

4.2 Les sous-contours

On constate que le rectangle circonscrit délimite quatre portions de contour associées aux quatre côtés du rectangle en parcourant le contour dans le sens positif. Chaque élément de contour correspond à la courbe comprise entre le dernier point de tangence avec le côté précédent et le premier point de tangence avec le côté suivant.

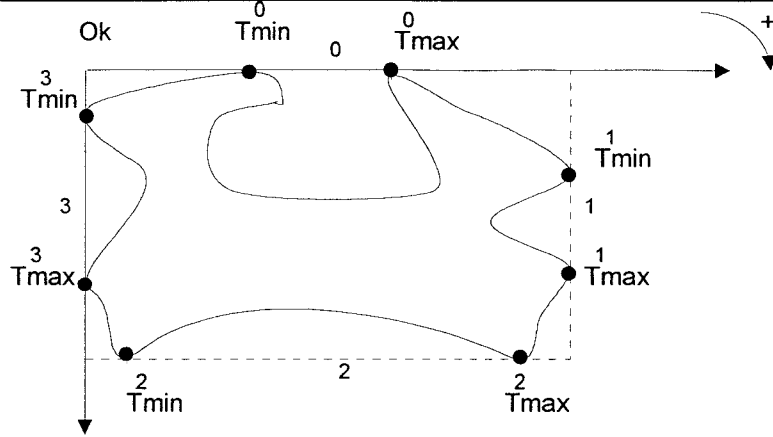


Figure 3.5

Les quatre portions de contour sont donc:

$$\begin{aligned}
 C_{k,0} &= (T_{min}^3, \dots, \pi_q, \dots, T_{min}^1) \\
 C_{k,1} &= (T_{max}^0, \dots, \pi_q, \dots, T_{max}^2) \\
 C_{k,2} &= (T_{max}^1, \dots, \pi_q, \dots, T_{max}^3) \\
 C_{k,3} &= (T_{min}^2, \dots, \pi_q, \dots, T_{min}^0) \\
 &\text{avec } \pi_q = (x_q, y_q).
 \end{aligned}$$

Les T_{min}^c et T_{max}^c sont les points de tangence définis par:

$$\begin{aligned}
 T_{min}^0 &= \left(\min_q x_q, 0 \right), & T_{max}^0 &= \left(\max_q x_q, 0 \right) \\
 T_{min}^1 &= \left(Lk, \min_q y_q \right), & T_{max}^1 &= \left(Lk, \max_q y_q \right) \\
 T_{min}^2 &= \left(\min_q x_q, Wk \right), & T_{max}^2 &= \left(\max_q x_q, Wk \right) \\
 T_{min}^3 &= \left(0, \min_q y_q \right), & T_{max}^3 &= \left(0, \max_q y_q \right)
 \end{aligned}$$

dans le repère $O_k X_k Y_k$

Les sous-contours $C_{k,c}$, $c = [0.. 3]$, sont des chemins 8-connexes.

Remarque Si un côté ne possède qu'un seul point de tangence, alors $T_{min}^c = T_{max}^c$.

4.3 Sous-contours apparents stricts

En reprenant la notion de contour apparent strict défini dans [LIU 91], on appellera "sous-contours apparents stricts" les listes:

$$0 \leq y_r \leq Wk \left\{ \begin{array}{l} C_{min}^x = \left\{ \pi_q \mid x_q = \min_{\pi_r \in C_k, y_r = y_q} x_r \right\} \\ C_{max}^x = \left\{ \pi_q \mid x_q = \max_{\pi_r \in C_k, y_r = y_q} x_r \right\} \end{array} \right\}$$

$$0 \leq x_r \leq Wk \left\{ \begin{array}{l} C_{min}^y = \left\{ \pi_q \mid y_q = \min_{\pi_r \in C_k, x_r = x_q} y_r \right\} \\ C_{max}^y = \left\{ \pi_q \mid y_q = \max_{\pi_r \in C_k, x_r = x_q} y_r \right\} \end{array} \right\}$$

Alors les listes C_{min}^y , C_{max}^x , C_{max}^y , C_{min}^x sont les sous-contours apparents stricts des portions de contour respectives $C_{k,0}$, $C_{k,1}$, $C_{k,2}$, $C_{k,3}$ (cf. figure 3.6).

On notera par exemple: $C_{min}^y = \tilde{A}(C_{k,0})$

On remarque, que les sous-contours apparents stricts ne sont pas toujours des chemins connexes. Dans la figure 3.6, l'alternance d'un point visible et d'un point caché engendre une rupture de connexité entre les points visibles.

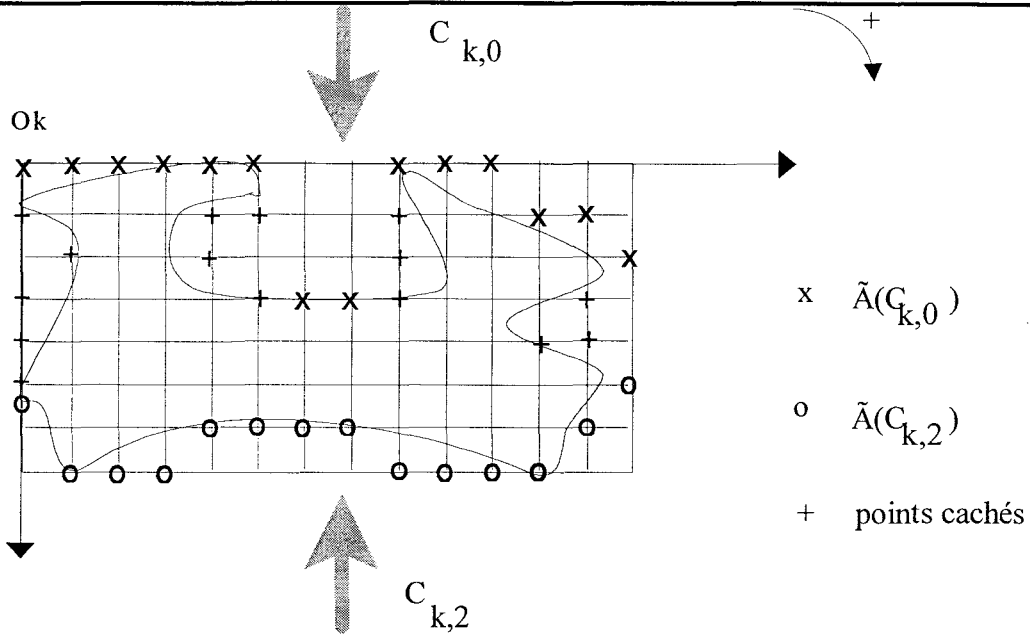


Figure 3.6.a

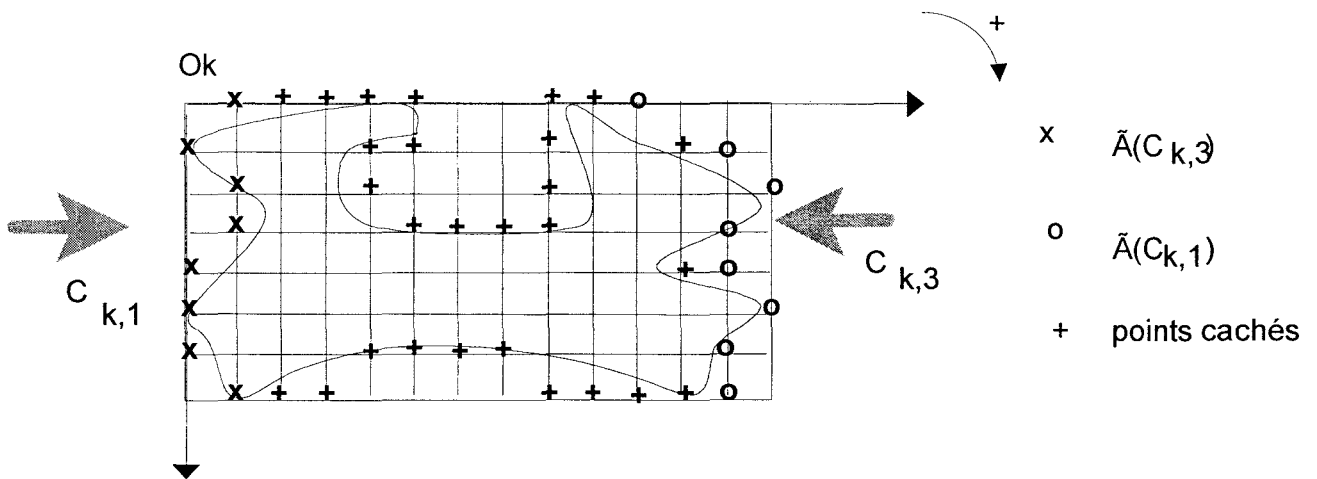


Figure 3.6.b

Remarque: La définition utilisée correspond ici à celle du sous-contour apparent simple exprimée dans [LIU 91]. Mais le contour C_k étant initialement simplement connexe, l'application de ces formules donne directement le contour apparent strict.

4.4 Définition des peignes de contour

Soit le contour 8-connexe C_k , nous appellerons "*peignes de contour*", les listes:

$$0 \leq x \leq Lk \quad D_{k,0} = \left\{ d_k(0,x) \mid d_k(0,x) = \underset{\pi_r \in C_k, x_r = x}{\text{MIN}} (y_r - y_{O_k}) \right\}$$

$$D_{k,2} = \left\{ d_k(2,x) \mid d_k(2,x) = \underset{\pi_r \in C_k, x_r = x}{\text{MIN}} (y_{O_k} + W_k - y_r) \right\}$$

$$0 \leq y \leq Wk \quad D_{k,1} = \left\{ d_k(1,y) \mid d_k(1,y) = \underset{\pi_r \in C_k, y_r = y}{\text{MIN}} (x_{O_k} + Lk - x_r) \right\}$$

$$D_{k,3} = \left\{ d_k(3,y) \mid d_k(3,y) = \underset{\pi_r \in C_k, y_r = y}{\text{MIN}} (x_r - x_{O_k}) \right\}$$

Chaque élément $d_k(c,z)$ d'un peigne $D_{k,c}$ est un nombre entier représentant la longueur d'un segment partant perpendiculairement au côté c du rectangle circonscrit et joignant un point du contour apparent strict. Ce segment est une "dent" du peigne.

L'ensemble de ces peignes couvre donc la surface représentant le complément du rectangle circonscrit R_k par rapport au contour apparent strict $\tilde{A}(C_k)$. Par analogie à la définition de la déficience convexe de [LIU 91], ce complément peut ici s'appeler la déficience rectangulaire. Chaque peigne couvre la déficience par rapport à la face tangente.

Remarque: Compte tenu de l'existence des points de tangence avec le rectangle circonscrit, chaque peigne contient au moins une dent de hauteur nulle.

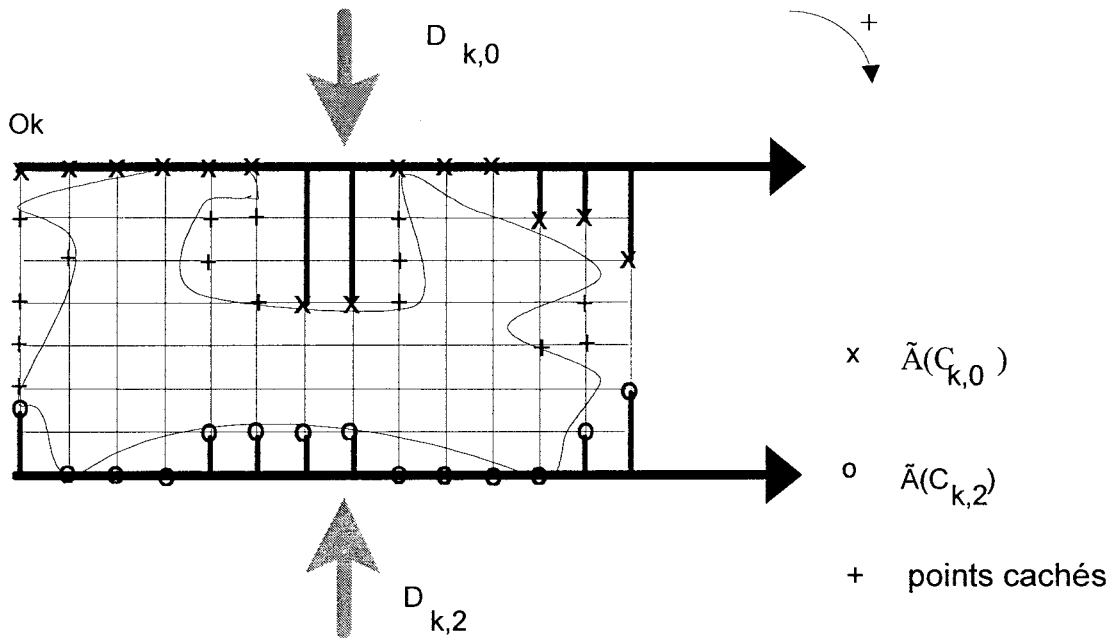


Figure 3.7.a

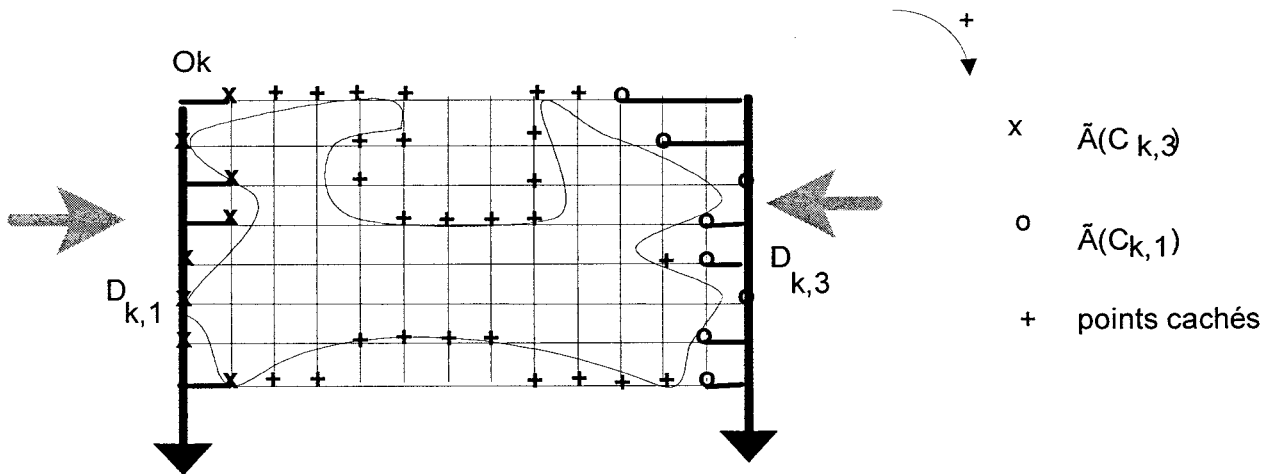


Figure 3.7.b

4.5 appellation

Considérant le rectangle circonscrit, chaque côté c est lié à un repère orthonormé O_c .

Soit $\tilde{C}_{k,c}(x) = \tilde{A}(C_{k,c}(x,y))$ la fonction continue représentant la portion de contour continue apparente stricte vue du côté c par rapport au repère (O_c, X_c, Y_c) . Soit $D_t(x)$ la distribution de Dirac appelée aussi *peigne de Dirac* de période T tel que (cf. Figure 3.8).

$$D_{k,c}(n) = \left[\tilde{C}_{k,c}(x) \cdot D_t(x) \right], \quad (\lfloor \psi \rfloor \text{ signifiant la partie entière de } \psi).$$

$D_{k,c}(n)$ étant la fonction échantillonnée discrétisant le sous-contour, alors la liste $D_{k,c} = (D_{k,2}(1), D_{k,2}(2), \dots, D_{k,c}(l_{k,c}))$ est appelée Peigne du contour C_k vu du côté c .

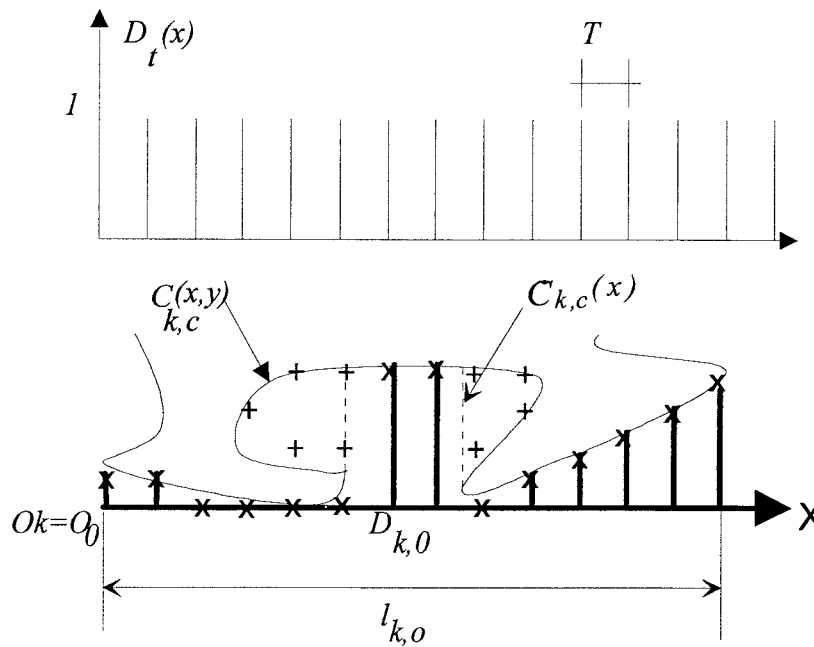


Figure 3.8

4.6 Justification de l'emploi des peignes de contour

La réalisation d'un placement est basée en fait sur la mise en imbrication itérative d'un placement intermédiaire et d'une nouvelle forme. Etant données les contraintes d'orientation régissant ce problème, toute association est réalisée par la mise en correspondance d'un côté du placement intermédiaire avec un des côtés d'une nouvelle forme. L'utilisation des peignes pour la description des formes permet, lors de la recherche de la position relative entre les deux objets, de limiter le traitement aux seuls points concernés par l'association, c'est à dire les points visibles de chacune des formes selon la direction choisie. Le problème de l'optimisation de l'imbrication de deux contours est donc réduit à un problème d'imbrication optimale de sous-contours beaucoup plus petits. C'est une démarche proche de la stratégie "diviser pour régner" en vue de réduire la complexité du traitement.

V Propriétés du codage par "peigne de contour"

5.1 Réversibilité

Le codage du contour par les peignes n'est pas exact si le contour apparent strict est différent du contour initial. Dans ce cas, certaines parties du contour, comme les concavités de

visibilité externe nulle¹ en observant perpendiculairement au côté c , ne sont pas codées et de l'information est perdue.

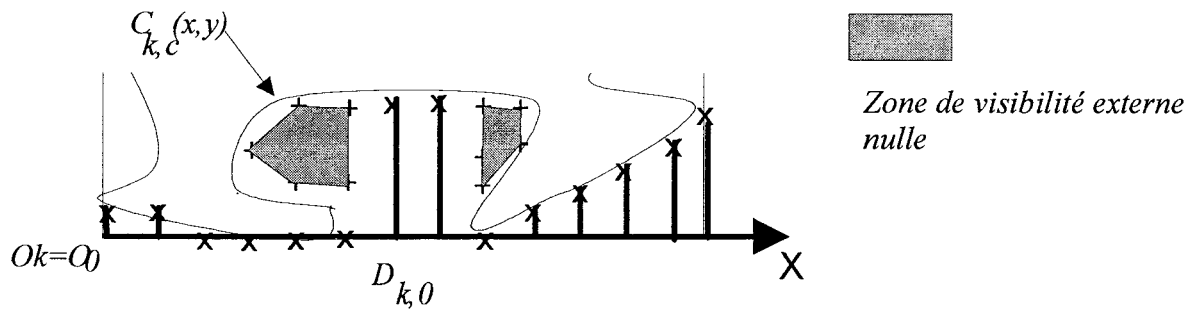


Figure 3.9

Si $\text{card}(C_k)$ et $\text{card}(\tilde{A}(C_k))$ renvoient le nombre de points appartenant à C_k respectivement à $\tilde{A}(C_k)$, alors:

$$\alpha = \frac{\text{card}(C_k)}{\text{card}(\tilde{A}(C_k))} \geq 1$$

où α représente la perte d'informations engendrée par le codage par peigne.

Compte-tenu de la forme générale des pièces rencontrées dans l'industrie du vêtement, le cas où $\alpha > 1$ semble rare. Le cas échéant, la perte d'information au niveau des concavités non-visibles ne sera pas préjudiciable à la qualité du placement.

5.2 Redondance

Comme on peut le constater sur la figure 3.10, certaines portions du contour sont codées par deux peignes successifs. La description est donc redondante.

¹**définition de la visibilité externe nulle en observant perpendiculairement au côté c :** Soit P_k une portion de contour délimitant une région F_k . Pour tout point π_q de P_k , soit $r(\pi_q)$ une demi droite issue du point π_q et partant dans la direction perpendiculairement au côté c , P_k est de visibilité externe nulle si et seulement si pour tout π_q de P_k , il existe aucune demi-droite telle que $P_k \cap r(\pi_q) = \{\pi_q\}$

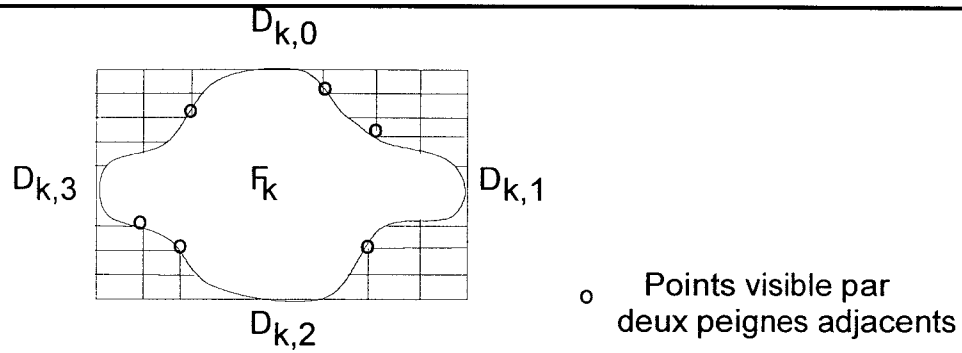


Figure 3.10

Nous désignerons par β , la redondance de description:

$$\text{soit } \beta = \frac{\sum_{c=0}^3 \text{card}(D_{k,c})}{\text{card}(\tilde{A}(C_k))} \geq 1$$

où $\text{card}(D_{k,c})$ renvoie le nombre de dents du peigne $D_{k,c}$, β indique la redondance de la description

Les valeurs particulières de β sont:

- Pour un rectangle, $\beta=1$ (pas de redondance)
- Pour un losange dont les axes sont parallèles aux axes principaux d'inertie du rectangle exinscrit, pour un cercle ou une ellipse où $\beta=2$, constitue le pire cas de redondance.

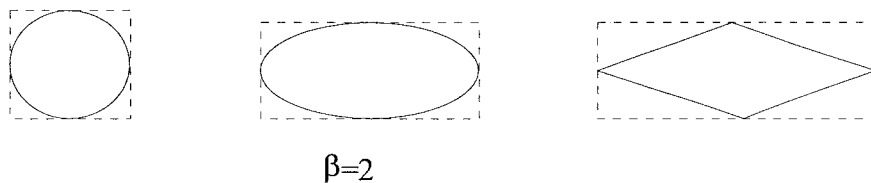


Figure 3.11

5.3 Taux de compression

On peut définir le taux de compression par la quantité T_1 :

$$T_1 = \frac{q(\text{description code source})}{q(\text{description code cible})}$$

En considérant que le code-source est caractérisé par les coordonnées de tous les sommets du contour 8-connexe C_k et en supposant une matrice image de 512*512 pixels, les quantités de bits d'informations nécessaires pour coder C_k et D_k valent:

$$q(\text{description code source}) = \text{card}(C_k) \times 2 \times 9 \text{ bits}$$

$$q(\text{description code cible}) = \sum_{c=0}^3 \text{card}(D_{k,c}) \times 9 \text{ bits}$$

on a donc:

$$T_1 = \frac{\text{card}(C_k)}{\sum_{c=0}^3 \text{card}(D_{k,c})} \times 2 \times \frac{9 \text{ bits}}{9 \text{ bits}} \quad \Rightarrow \quad T_1 = \frac{\alpha}{\beta} \times 2$$

Dans le pire cas de mauvaise compression: $\alpha=1$ (aucune perte d'information) et $\beta=2$ (redondance maximale), alors $T_1=1$

5.4 Invariances

5.4.1 Non-invariance par rotation d'angle quelconque

Cette représentation n'est pas invariante par rotation. Un changement d'orientation du contour dans le repère d'édition modifie le codage. Cependant, pour des rotations multiples de $\pi/2$, il est possible de retrouver un codage identique en réalisant une permutation circulaire de peignes.

5.4.2 Invariance par translation

La description n'est pas sensible à la translation du contour dans le repère d'édition, puisque les peignes sont exprimés à partir du rectangle exinscrit.

5.4.3 Non-invariance par homothétie

Cette remarque est évidente dans la mesure où une homothétie d'un contour engendre un nombre de points différents et donc les tailles et le nombre des dents sont aussi différents.

5.5 déformation du contour

Lorsqu'une forme présente des concavités, on a vu que de l'information pouvait être perdue au niveau des parties non visibles. Cela est rarement préjudiciable car les espaces non codés sont souvent irrécupérables au sens du placement. Mais lorsque le contour présente des sections

d'inflexion dont les tangentes sont parallèles aux côtés du rectangle exinscrit, alors le codage déforme légèrement cette portion de contour, en raison de la rupture de connexité engendrée lors du codage. Pour le cas des formes A et B illustrées dans la **figure 3.12**, le codage par les peignes engendre une distorsion du contour pouvant éventuellement interdire l'imbrication de deux formes rigoureusement complémentaires. Cette remarque compromet l'utilisation de ce code pour certains cas de puzzle, mais ne remet pas réellement en cause l'intérêt de celui-ci pour notre application.

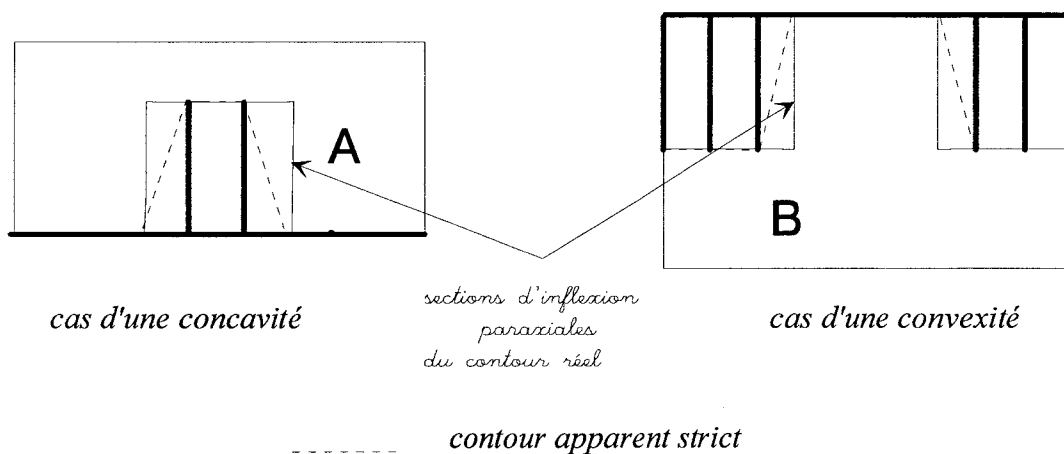


Figure 3.12



VI Peignes réduits

6.1 Définition

L'usage des peignes de forme s'est révélé gourmand en mémoire et pénalisant en complexité de calcul, au moins en ce qui concerne l'évaluation de l'imbrication entre deux formes ainsi que la concaténation. Si cette remarque est particulièrement justifiée lorsque les formes décrites sont composées de nombreux segments de droites, elle est par contre beaucoup moins motivée si le contour en question est essentiellement curviligne. L'idée était donc de trouver une méthode adaptative permettant de supprimer la description des points situés entre les sommets des segments. Dans les opérateurs qui seront définis dans le chap IV, les points intermédiaires pourront, en cas de nécessité être déduits par interpolation linéaire.

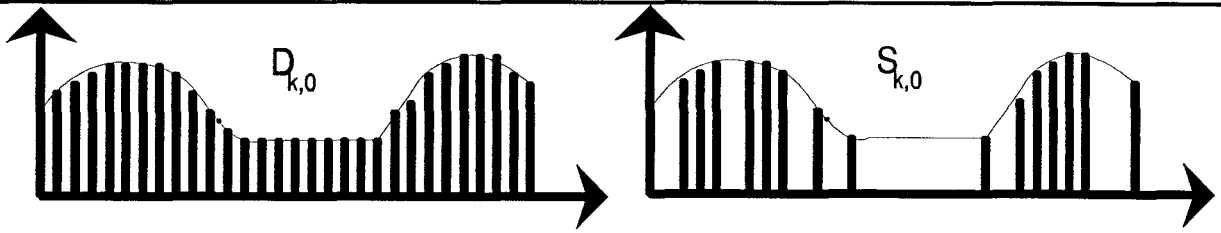


Figure 3.13

Par convention, on appellera $S_{k,c}$ la liste des sommets du sous-contour discret de la forme F_k vu du côté c . Les dents n'étant plus régulièrement espacées, la mémorisation de l'abscisse de chaque dent est rendue nécessaire. Il vient donc:

$$\forall c \in \{0, 1, 2, 3\}, S_{k,c} = \left\{ (x, d_{k,c}(x)) \mid 0 \leq x \leq l_{k,c}, d_{k,c}(x) \in \hat{P}(D_{k,c}) \cap D_{k,c} \right\}$$

où:

$\hat{P}(D_{k,c})$ est l'approximation polygonale du peigne du sous-contour $D_{k,c}$, et $l_{k,c}$ est la longueur du côté c .

On désignera par $S_{k,c}(i)$ le $i^{\text{ème}}$ sommet de la liste $S_{k,c}$.

L'utilisation de l'approximation polygonale permet donc de diminuer le nombre de points de contrôle dans la description de chaque sous-contours visibles. Il vient donc que:

$$\text{card}(S_{k,c}) \leq \text{card}(D_{k,c})$$

Remarque N°1: Les propriétés du codage par peigne et peigne réduit sont identiques à l'exception du taux de compression qui dépend dans le second cas, de la courbure de la forme, mais aussi de la précision souhaitée.

Remarque N°2 Nous désignerons par S_k l'association (et non la réunion) des quatre peignes réduits définissant le contour $\tilde{A}(C_k)$, c'est à dire:

$$S_k = (S_{k,0}, S_{k,1}, S_{k,2}, S_{k,3})$$

La notion d'approximation polygonale est motivée ici pour plusieurs raisons:

- Parce que d'une part, elle sous-entend l'utilisation d'algorithme permettant l'identification des segments de droite, souvent difficile à détecter en raison du bruit lié à la discrétisation.

- D'autre part, l'extraction de sommets caractéristiques en nombre réduit améliore, selon la nature du contour, le taux de compression du codage. Certes le codage n'est plus obligatoirement exact, mais la quantité d'information perdue peut être contrôlée par un critère de précision qui s'associe à la notion de bruit. Nous présentons donc la méthode d'approximation polygonale utilisée.

6.2 Approximation polygonale

6.2.1 Introduction

Les algorithmes de traitement mis en oeuvre dans le chapitre IV nous amèneront à constater que leur complexité est très souvent dépendante du nombre de sommets présents dans les peignes réduits. La réduction des calculs conduit à minimiser le nombre de ces sommets.

Cependant, il est impératif de prendre en considération les idées suivantes:

- 1) L'approximation doit être la plus fine possible par rapport à un critère de similarité et un paramètre de réglage pour ne pas altérer considérablement la qualité du placement.
- 2) L'algorithme ne doit pas entraîner de dérogations au principe de satisfaction de la contrainte de la disjonction des contours initiaux.
- 3) Le nombre de segments creux conditionneront le nombre de mises en correspondance de creux et bosses dans la recherche d'imbrication. Aussi, la présence du bruit dans les contours favorise les essais inefficaces. Il faut donc supprimer l'existence des concavités et convexités parasites introduites en général par les défauts d'aliassage².

On privilégiera la qualité du résultat obtenu à la vitesse d'obtention du polygone résultant, puisque ces calculs peuvent s'effectuer hors-ligne, c'est-à-dire préalablement à l'exécution de l'algorithme de placement.

De ce fait, l'algorithme d'approximations polygonales doit satisfaire aux exigences suivantes:

- 1) Contrôler la précision de l'approximation obtenue au moyen d'un paramètre défini par l'utilisateur. Ce paramètre permettra de calculer le seuil limite que la variation cumulée de la surface du peigne ne doit pas dépasser.
- 2) Les points de la frontière du contour initial doivent toujours répondre au principe d'unilatéralité par rapport aux segments du polygone résultant. Seuls les points de

²**Défauts d'aliassage:** Défauts introduits par une sous-définition des mémoires d'image pour représenter les segments obliques. Ils se manifestent par exemple sous forme de marches d'escalier le long des contours.

concavité à l'itération courante de l'algorithme pourront être supprimés de la liste des sommets du peigne à réduire lors du fusionnement des segments.

- 3) L'algorithme doit favoriser la suppression des points de concavité lié au bruit en fusionnant systématiquement en premier les segments des zones de concavité minimale.

6.2.2 Les algorithmes existants

Plusieurs algorithmes d'approximation polygonale existent dans la littérature. On relève l'approximation par découpage itératif [RAM 72] ou récursif [PAV 74], [BAR 88] utilisant la méthode de la corde.

On trouve ensuite la famille des algorithmes récurrents qui calculent pour chaque test de fusion une erreur cumulée (variation de surface [WAL 84], ou au sens des moindres carrés [LUX 85]).

Un autre algorithme plus récent utilise des méthodes structurales de reconnaissance de motifs dans les différences finies d'ordre 1 entre les points du contour [RAY 91]. On trouve aussi les approximations angulaires [BER 79]. Parmi les algorithmes énoncés précédemment, aucun d'entre eux ne satisfait à toutes les exigences demandées. En cela, le principe suivant répond au but fixé.

6.2.3 Principe

Il consiste par itérations successives, à supprimer les points de concavité dont la fusion³ des segments incidents et sortants engendre une variation de surface minimale. L'algorithme s'arrête lorsque la variation cumulée dépasse la limite maximale ΔS_{max} . Ce seuil est égal au produit $\varepsilon \times l_{k,c}$, où ε est un paramètre fixant la précision souhaitée par unité de surface et $l_{k,c}$ la longueur du côté c de F_k .

Cet algorithme utilise la définition du produit vectoriel afin de calculer la variation élémentaire de surface lors de la suppression d'un point.

³ action de supprimer les segments incident (ex [AB]) et sortant (ex [BC]) d'un point B d'un polygone et de les remplacer par le segment [AC]

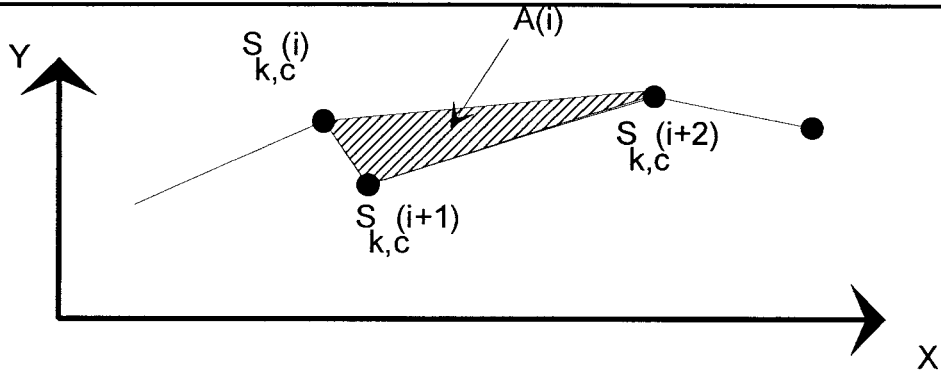


Figure 3.14

En effet l'aire algébrique $A(i)$ est telle que

$$\begin{pmatrix} 0 \\ 0 \\ 2 \times A(i) \end{pmatrix} = \frac{\quad}{S_{k,c}^{(i)}S_{k,c}^{(i+1)}} \rightarrow \wedge \frac{\quad}{S_{k,c}^{(i+1)}S_{k,c}^{(i+2)}} \rightarrow$$

Donc si lors de la fusion, le point $S_{k,c}^{(i+1)}$ est supprimé, la variation de surface engendrée dans cette approximation est $A(i)$.

D'autre part, la détermination de la concavité peut se faire au moyen du signe de $A(i)$:

Si $A(i) > 0$ alors $S_{k,c}^{(i+1)}$ est un point de concavité

Nous éliminerons donc uniquement les sommets dont la valeur de $A(i)$ est positive, en priorité ceux dont la valeur est minimale et ce tant que la variation de surface cumulée soit inférieure à ΔS_{max} . L'annexe C présente l'algorithme détaillé de cette procédure. La méthode algorithmique y est désignée par le message :

$S_{k,c}$ *approx_Poly*: ε

VII Caractérisation des peignes

Les facteurs présentés dans cette partie permettront d'évaluer les caractéristiques locales en effectuant des calculs au niveau d'un ou plusieurs peignes-réduits. Ils peuvent participer à la comparaison entre deux formes ou servir d'attributs dans une classification de formes.

7.1 Longueur d'un sous-contour apparent

$$longueur(\tilde{A}(C_{k,c})) = \sum_{i=1}^{card(S_{k,c})-1} d(S_{k,c}^{(i)}, S_{k,c}^{(i+1)})$$

où $d(P, Q) = \sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}$ est la distance euclidienne entre deux points P et Q

Cette grandeur est représentative de la longueur du sous-contour vu d'un côté, mais la somme de ces grandeurs sur tout le contour n'est pas significative en raison de la redondance éventuelle de la description.

7.2 Surface d'un peigne réduit

En sommant les éléments de surface trapézoïdaux, il vient :

$$surface(S_{k,c}) = \sum_{i=1}^{card(S_{k,c})-1} \frac{1}{2} \left(X_{S_{k,c}}(i+1) - X_{S_{k,c}}(i) \right) \left(Y_{S_{k,c}}(i) + Y_{S_{k,c}}(i+1) \right)$$

Ce calcul est intéressant lorsqu'il s'agit de calculer la surface récupérable selon une direction du placement, ou pour calculer le talon et le front d'un placement (cf chap2).

La somme des surfaces des peignes de tous les côtés fournit un majorant de la surface de la déficience rectangulaire. L'écart par rapport à la surface réelle de la déficience rectangulaire est très certainement fonction de la redondance.

7.3 Taux de la déficience rectangulaire

La surface d'un peigne exprime l'importance de la zone de concavité du sous contour apparent par rapport au côté du rectangle considéré. Mais cette valeur dépend beaucoup des dimensions du rectangle circonscrit. Afin de normaliser cette grandeur, on peut établir le paramètre suivant (**Figure 3.15**) :

$$concavité(S_{k,c}) = \frac{surface(S_{k,c})}{l_{k,c} \times S_{k,c}^{max}} \leq 1 \quad \text{où } S_{k,c}^{max} \text{ représente la hauteur maximale parmi}$$

les hauteurs de toutes les dents du peigne réduit $S_{k,c}$.

Ce rapport exprime intuitivement le taux de déficience rectangulaire pour le côté c . Plus ce rapport est élevé, plus il y a d'espace vide entre la forme et le côté du rectangle circonscrit.

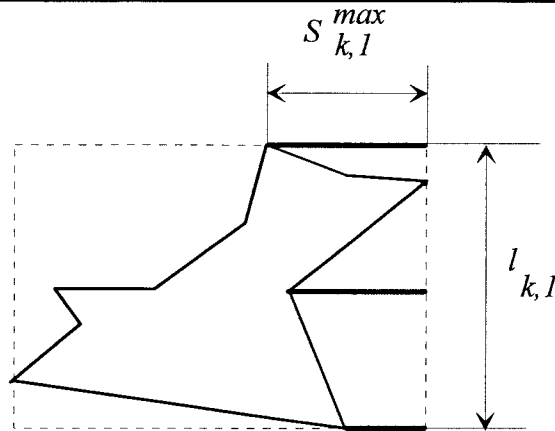
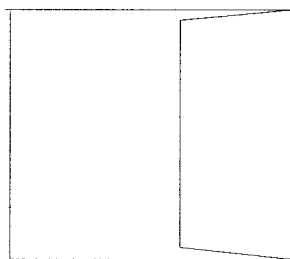
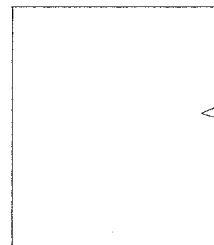


Figure 3.15

Les deux cas de figures particuliers sont :



Zone fortement
concave
 $concavité(S_{k,l}) \cong 1$



Zone faiblement
concave
 $concavité(S_{k,l}) \cong 0$

Figure 3.16

Mais celui-ci présente l'inconvénient d'être indéterminé numériquement lorsque toutes les dents sont nulles (cas du rectangle). On préfère donc le quotient suivant:

$$concavité(S_{k,c}) = \frac{surface(S_{k,c})}{l_{k,c}}$$

Il est alors proche de la hauteur moyenne des dents du peigne simple.

Il en découle:

si $concavité(S_{k,c}) = 0$ alors face c de F_k rectiligne

si $concavité(S_{k,c}) \gg 0$ alors face c de F_k fortement concave

7.4 Complexité d'un peigne

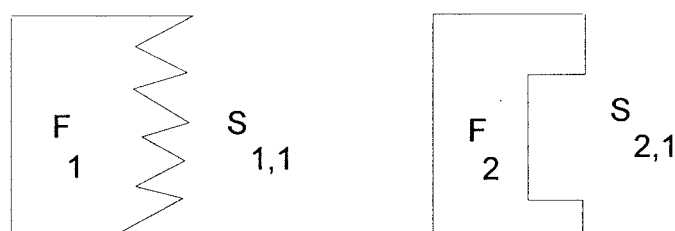
Le degré de concavité précédent reste équivoque lorsqu'il s'agit d'estimer la complexité du peigne, c'est-à-dire la variabilité de la hauteur des dents qui le compose. Au calcul de moyenne

précédent, on peut ajouter la variance, renseignant sur la dispersion des hauteurs par rapport à la moyenne.

$$\text{complexité}(S_{k,c}) =$$

$$\sqrt{\frac{1}{l_{k,c}} \sum_{i=1}^{\text{card}(S_{k,c})} x_{S_{k,c}(i+1)}^{-1} \left[y_{S_{k,c}(i)} + \frac{(y_{S_{k,c}(i+1)} - y_{S_{k,c}(i)})}{(x_{S_{k,c}(i+1)} - x_{S_{k,c}(i)})} (x - x_{S_{k,c}(i)}) - \text{concavité}(S_{k,c}) \right]^2} \geq 0$$

Avec cette formulation, la valeur 0 représente la complexité minimale. Cet indice de dispersion renseigne donc sur la complexité du côté considéré. L'exemple suivant illustre une situation possible, où l'indice de concavité est le même pour deux formes, mais où la complexité est différente.



$$\text{concavité}(S_{1,1}) = \text{concavité}(S_{2,1})$$

$$\text{complexité}(S_{1,1}) > \text{complexité}(S_{2,1})$$

Figure 3.17

On pourrait bien sur étendre la liste de ces facteurs de caractérisation de peignes ou de formes, mais le but ici est plutôt de présenter les traitements relatifs au problème du placement. Néanmoins, quelques exemples typiques ont permis d'apprécier l'usage possible des peignes-réduits à d'autres fins.

VIII Opérateurs de formes unitaires

Dans cette partie, nous proposons de réécrire les opérateurs géométriques de base modifiant l'état d'une seule forme. Les opérateurs faisant partie de cette classe sont la rotation, la symétrie et le peigne dual. Il convenait en effet de tirer profit de la particularité de la description des contours par les peignes ainsi que de la contrainte de droit fil pour redéfinir ces opérateurs qui sont par ailleurs nécessaires à la construction d'un placement. La classe des opérateurs bi-opérandes sera développée dans le chapitre IV

8.1 Rotation de $\pi/2$

L'opération de rotation d'angle $\lambda\pi/2$, $\lambda \in \mathbb{Z}$, de centre le centre d'inertie du rectangle exinscrit peut s'effectuer facilement en réalisant une permutation circulaire des indices du côté associé à chaque peigne réduit.

soit $S_k = \{S_{k,c}, c \in \{0,1,2,3\}\}$

F_k rotation: $\lambda \Rightarrow S_k = \{S'_{k,c} \mid S'_{k,c} = S_{k,c+\lambda[4]}, \lambda \in \mathbb{Z}, c \in \{0,1,2,3\}\}$

Remarque N°1

Compte tenu de l'inséparabilité du modèle de visualisation et du modèle de traitement, la rotation doit s'accompagner d'une rotation du contour de visualisation C_k . Ceci se fait très classiquement au moyen d'une transformation matricielle des coordonnées des points de C_k .

Remarque N°2

La rotation d'un angle quelconque n'est possible qu'en déterminant de nouveau les peignes réduits après une rotation du contour initial C_k de l'angle voulu.

8.2 Symétries par rapport à $Cr_k X$ et $Cr_k Y$

8.2.1 Symétrie par rapport à $Cr_k X$

La figure 3.15 représente explicitement l'opération effectuée:

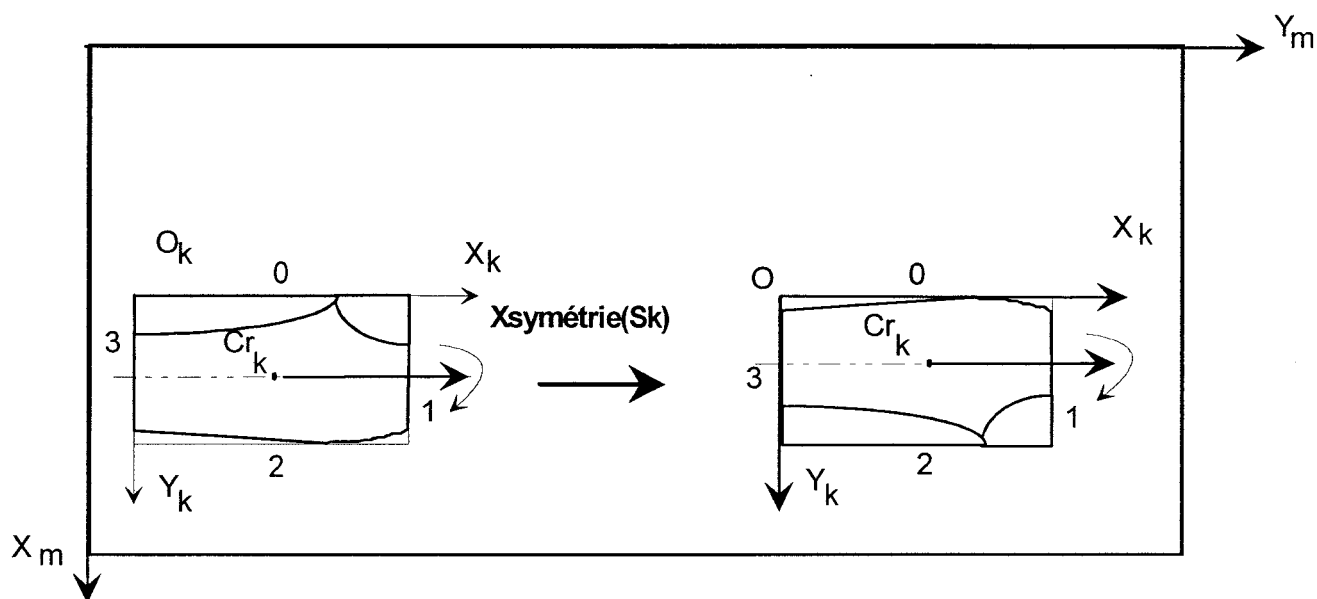


Figure 3.18

Cette opération peut s'écrire

soit $S_k = \{S_{k,c}, c \in \{0,1,2,3\}\}$

F_k Xsymétrie \Rightarrow

$$S_k = \left\{ S_{k,c} \left| \begin{array}{l} \text{si } c \text{ est pair alors } S_{k,c} = S_{k,c+2}[4] \\ \text{si } c \text{ est impair alors } \forall q \in [0, \text{card}(S_{k,c})], S_{k,c} = \left(S_{k,c}(q) \left| \begin{array}{l} x S_{k,c}(q) := l_{k,c} - x S_{k,c}(q) \\ y S_{k,c}(q) \end{array} \right. \right) \right. \right\}$$

8.2.2 Symétrie par rapport à Cr_kY

De la même manière, la figure 3.16 explique la symétrie en OY.

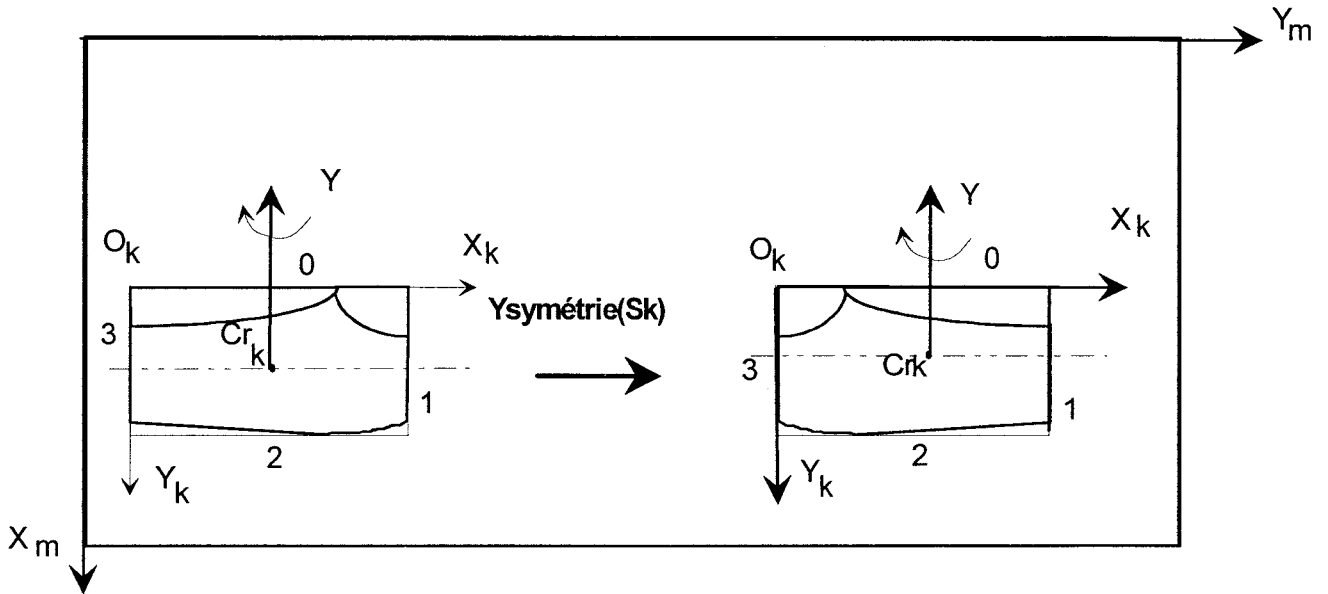


Figure 3.19

Cette opération peut s'écrire

$$\text{soit } S_k = \{S_{k,c}, c \in \{0,1,2,3\}\}$$

$$S''_k = S_k \text{ Ysymétrie} \Rightarrow$$

$$S''_k = \left\{ S''_{k,c} \left| \begin{array}{l} \text{si } c \text{ est pair alors } \forall q \in [0, \text{card}(S_{k,c})], S''_{k,c} = \left(S''_{k,c}(q) \left| \begin{array}{l} x S''_{k,c}(q) \equiv l_{k,c-x} S_{k,c}(q) \\ y S''_{k,c}(q) \equiv y S_{k,c}(q) \end{array} \right. \right) \right. \\ \text{si } c \text{ est impair alors } S''_{k,c} = S_{k,c+2[4]} \end{array} \right. \right\}$$

8.3 Peigne dual et forme duale

L'étude de l'imbrication de deux formes que nous développerons dans le chapitre IV, peut être approchée selon le principe de la mesure de similarité entre le côté d'une forme et le complémentaire d'un des peignes de l'autre forme que nous préférons appeler "peigne dual" au sens des peignes réduits.

8.3.1 Définition

Le peigne dual $S^*_{k,c}$ correspond au peigne qu'aurait le sous-contour $\tilde{A}(C^*_{k,c})$ d'une forme imaginaire F^*_k qui viendrait en correspondance idéale avec le sous-contour apparent strict $\tilde{A}(C_{k,c})$ de la forme initiale. La forme F^*_k est appelée forme duale de F_k selon le côté c .

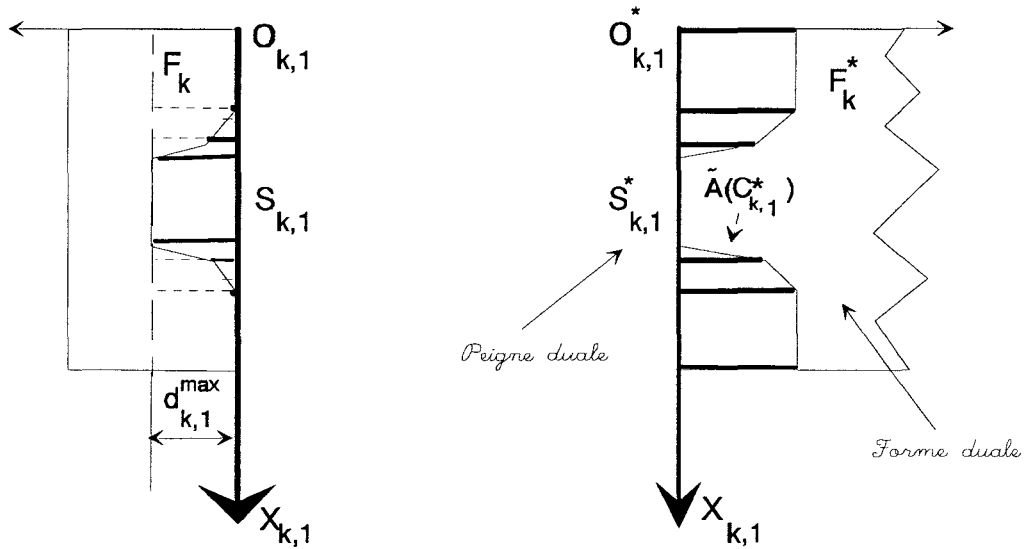


Figure 3.20

Les composantes du peigne dual $S^*_{k,c}$ sont obtenues par:

$$S^*_{k,c} = \left((x, d^*_{k,c}(x)) \mid d^*_{k,c}(x) = d^{max}_{k,c}(x) - d_{k,c}(x) \right)$$

$$\text{avec } d^{max}_{k,c} = \underset{z}{\text{Max}} d_{k,c}(z)$$

Chaque dent duale correspond au complément de la dent du peigne initial par rapport à la hauteur de dent maximale. Ce calcul effectué pour toutes les dents génère bien le peigne réduit dual

Malheureusement, cette notion de peigne dual perd de sa généralité car elle est victime de l'inconvénient majeur lié aux déformations du contour lors de la présence de partie non visible ou de sections d'inflexions paraxiales.

IX Classe des formes irrégulières : *Forme_Modèle*

Nous allons décrire ici la classe des formes irrégulières que nous appellerons *Forme_Modèle*. C'est sous cet aspect que nous avons décidé d'élaborer une structure de données rassemblant toutes les variables nécessaires à la bonne représentation des formes utilisant le concept des peignes de contour. Les instances de cette classe seront supposées être représentées par les peignes réduits. Cependant, nous n'en donnerons, dans un premier temps qu'une description partielle. En effet, à ce stade de l'étude, tous les paramètres utiles ne sont pas encore connus. Nous verrons dans le chapitre suivant les compléments que nous y apporterons.

9.1 Les variables d'instances

Leur affectation est propre à chaque instance créée.

Ce sont:

- Le contour de référence $C(k)$ dont les points constitutifs sont exprimés dans le repère d'édition
- Les quatre peignes réduits $S_{k,c}$, $c \in \{0,1,2,3\}$
- éventuellement les peignes duaux $S^*_{k,c}$
- Les quatre listes $Z_{k,c}$ contenant l'attribut de convexité (0 = convexe, 1= concave) pour chaque sommet des peignes
- Les variables de configuration courante:
 - λ pour l'orientation
 - S_X pour la symétrie
- Les domaines de définition des variables de configuration:
 - l'ensemble *orientations_Admisses*
 - le drapeau de symétrie S_{CrX}
- Les attributs de côtés:
 - $surface(S_{k,c})$, $longueur(\tilde{A}(C_{k,c}))$, $concavité(S_{k,c})$

9.2 Les variables de la classe *Forme_Modèle*

Elles sont uniques pour toutes les instances de la classe, et donc propres à la classe. C'est la précision de description des peignes réduits, c.a.d. la tolérance de surface linéique ϵ .

9.3 Messages et Méthodes

L'application que nous avons développée a été prototypée à l'aide d'un langage de simulation orienté objet : *SMALLTALK*. En conséquence, les procédures algorithmiques que nous sommes amenés à écrire seront nommées selon la syntaxe propre à ce langage. En terme de langage à objets, on appelle cela *un message*. Les objets concernés par ce message seront les *récepteurs*. Eventuellement, un message pourra contenir un ou plusieurs *arguments*. Enfin, la façon dont un objet répond à un message est décrite dans une *méthode*.

La classe *Forme_Modèle* contient une liste de méthodes qui permettent à ses instances de répondre aux messages qui leurs sont envoyés. A ce niveau nous avons défini:

F_k *rotation*: λ

F_k *Xsymétrie*

F_k *Ysymétrie*

X Conclusion

Dans ce chapitre, nous avons présenté une nouvelle façon de décrire les contours des formes irrégulières qui sont mises en jeu dans un système de placement. Le cadre restreint dans lequel nous nous plaçons où les contraintes de placement réduisent l'orientation des formes nous a permis d'élaborer le concept des peignes de formes. En fait, les peignes répondent à une démarche naturelle visant à se limiter à la connaissance du sous-contour du côté que l'on sera amené à mettre en relation avec une autre forme lors de la recherche d'une imbrication locale optimale. Mais la généralisation de leur utilisation à d'autres fins que le placement soulève quelques questions. Les inconvénients que nous avons relevés en analyse, en particulier la perte d'informations des parties non-visibles, la redondance de description, les déformations engendrées au niveau des sections d'inflexion paraxiales et la non-invariance par rotation constituent des obstacles insurmontables pour certaines applications. Certains de ces inconvénients pourraient toutefois être supprimés, si le même principe était appliqué à partir du cercle circonscrit. La visibilité serait améliorée, la redondance nulle, et l'invariance par rotation obtenue.

En tout cas, c'est ce modèle de description qui servira dans la suite de ce travail et tous les traitements ultérieurs viseront à optimiser au mieux les potentialités de ce concept. Alors que les commandes géométriques que l'on peut envoyer à une seule forme ont été décrites, les opérateurs mettant en jeu deux formes restent à définir. Elles sont, et on le comprend bien, essentielles à la réalisation d'un placement combinatoire.

CHAPITRE IV:

IMBRICATION DES FORMES IRREGULIERES

Tout ce qui bouge, bouge parce que d'autres
choses bougent.

St Thomas d'Aquin

I Introduction

L'objectif qui sera suivi tout au long de ce chapitre concerne la construction géométrique du placement. Pour cela, il nous faut présenter les deux fonctions indispensables à sa réalisation: la concaténation et l'imbrication optimale ou sous optimale. La première, sorte de mariage entre deux formes, a pour objet de créer une nouvelle forme dont le contour est issu de l'enveloppe de la réunion de ses "géniteurs". En réalisant consécutivement cette opération avec toutes les formes à produire, on imagine comment le placement atteindra les allures d'une famille. En outre, celle-ci doit se développer à l'intérieur d'un espace borné. Mais pour loger le maximum d'occupants, il faut combiner au mieux chaque couple de formes à la faveur d'un critère local d'imbrication. Nous en choisirons un.

Cependant, la qualité des imbrications n'est pas notre seule préoccupation. Les opérateurs que nous allons présenter constitueront le "noyau" central du traitement pour réaliser la construction d'une famille de formes. L'aspect combinatoire lié à la recherche de la famille optimale implique un usage très fréquent de ces fonctions. Une faible complexité des calculs associés s'impose donc, et constituera une deuxième ligne de mire à cette étude.

Enfin, la problématique de cette étude nous rappelle que la prise en compte des motifs fait partie des contraintes de travail. Il faudra donc intégrer cette spécificité dans chacun des opérateurs présentés. Nous remarquerons par ailleurs que cela diminue le domaine de définition des positions relatives d'imbrication et par conséquent la complexité de la recherche.

II Placement de rectangles paraxiaux non disjonctifs

2.1 Objectif

On a vu dans le chapitre III que les sommets appartenant à chacun des peignes réduits $S_{k,c}$, sont définis par rapport au repère local associé au côté c du rectangle R_k associé à F_k . Par conséquent, au niveau des opérateurs de traitement que nous allons développer, utiles à la construction d'un placement (recherche locale de l'imbrication entre deux formes, concaténation de deux formes), nous serons amenés à exprimer les coordonnées des points mis en jeu dans chaque forme dans un autre repère. Pour cela, il semble naturel de choisir le repère $O_{kul}XY$ comme repère cible. Conformément à nos conventions, l'origine du repère est situé au coin supérieur gauche du rectangle circonscrit à l'enveloppe de la concaténation des deux contours.

Auparavant, nous proposons de caractériser mathématiquement les rectangles à l'aide d'une forme de matrice. A l'aide de cette définition, nous pourrions établir ensuite les traitements nécessaires à la mise en relation de deux rectangles à l'instar de ceux que nous définirons pour les formes irrégulières. Ainsi, lorsque cela sera nécessaire, nous pourrions passer de l'espace des peignes réduits vers l'espace des rectangles par simple changement de repère. De plus, cette formulation permettra de soulager la présentation des calculs intervenant au niveau des relations de traitement des formes irrégulières.

Remarquons enfin que la spécificité du problème et en particulier la contrainte d'orientation affectant chaque forme, nous amène à limiter nos formulations à celles des *rectangles paraxiaux*. De plus, les imbrications --souhaitables-- des formes dans la construction du placement impliquent la *non-disjonctivité* éventuelle des rectangles circonscrits de chaque forme.

2.2 Représentation d'un rectangle paraxial par une matrice

Soit une forme F_k et son rectangle circonscrit R_k

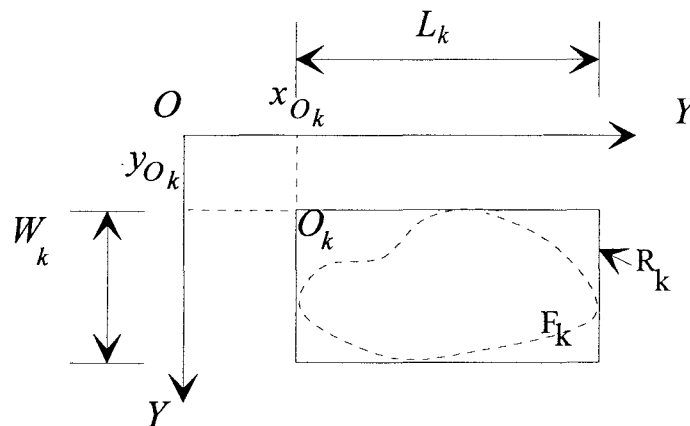


Figure 4.1

On note par $\overline{\overline{OMk}}$ la matrice à coefficients définis sur Z , caractérisant entièrement le rectangle R_k .

$\overline{\overline{OMk}}$ est défini par:

$$\overline{\overline{OMk}} = \begin{bmatrix} m_{0,0}^k & m_{0,1}^k \\ m_{1,0}^k & m_{1,1}^k \end{bmatrix} = \begin{bmatrix} x_{O_k} & x_{O_k} + L_k \\ y_{O_k} & y_{O_k} + W_k \end{bmatrix}$$

où chaque colonne $m_j = \begin{bmatrix} m_{0,j}^k \\ m_{1,j}^k \end{bmatrix}$ de la matrice $\overline{\overline{OMk}}$ correspond aux vecteurs de coordonnées de l'origine \vec{OO}_k ($j=0$) et du coin diamétralement opposé à O_k ($j=1$) dans le repère OXY .

2.3 Propriétés des matrices de rectangle $\overline{\overline{OMk}}$

Dans ce paragraphe, on donne les relations permettant de déterminer les différentes grandeurs caractérisant les rectangles paraxiaux.

2.3.1 dimensions d'un rectangle:

Soit $l_{k,c}$ la longueur d'un côté c du rectangle R_k , alors:

$$\begin{aligned} l_{k,c} &= m_{0,1}^k - m_{0,0}^k && \text{si } c \text{ est pair} \\ l_{k,c} &= m_{1,1}^k - m_{1,0}^k && \text{si } c \text{ est impair} \end{aligned}$$

Les coordonnées des points étant algébriques, il se peut que les longueurs $l_{k,c}$ soient négatives selon la nature du rectangle. On parle alors de *longueur algébrique*.

2.3.2 surface d'un rectangle

On définira aussi la *surface algébrique* SR_k d'un rectangle R_k par :

$$SR_k = l_{k,0} \times l_{k,1}$$

2.3.3 Signe d'un rectangle

Un *rectangle est positif (négatif)* si $SR_k > 0$ respectivement ($SR_k < 0$)

2.3.4 Rectangles particuliers

Un rectangle est *dégénéré* si $SR_k = 0$. Dans ce cas, il existe au moins un côté c tel que $l_{k,c} = 0$ et il s'agit donc d'un segment paraxial ou d'un point.

Un *rectangle est un point* si :

$$\forall c, \quad l_{k,c} = 0$$

Le *rectangle-origine* défini par la matrice $\overline{\text{OMo}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ correspond au point-origine O du repère OXY

Un *rectangle est un segment paraxial* si :

$$\exists c \text{ tel que } l_{k,c} = 0 \text{ ou } l_{k,c+1[4]} = 0.$$

Dans ce cas, le rectangle se réduit à un segment parallèle à l'un des axes du repère OXY .

2.3.5 Extraction des vecteurs particuliers

Soit \overline{OMk} la matrice du rectangle R_k alors *les coordonnées de l'origine O_k du rectangle par rapport à l'origine O* sont données par le vecteur

$$\vec{O O_k} = \begin{pmatrix} m_{0,0}^k \\ m_{1,0}^k \end{pmatrix}$$

De même, *les coordonnées des origines locales $O_{k,c}$ par rapport à l'origine O_k du rectangle* sont données par (cf. chapitre 3 figure 3.4.b):

$$\vec{O_k O_{k,c}} = \left\{ \begin{array}{l} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ si } c = 0 \text{ ou } 3 \\ \begin{pmatrix} l_{k,0} \\ 0 \end{pmatrix} \text{ si } c = 1 \\ \begin{pmatrix} 0 \\ l_{k,1} \end{pmatrix} \text{ si } c = 2 \end{array} \right\} = \begin{pmatrix} l_{k,0} & \text{si } c = 1, 0 \text{ sinon} \\ l_{k,1} & \text{si } c = 2, 0 \text{ sinon} \end{pmatrix}$$

2.4 Positionnement relatif de deux rectangles

Les opérations ensemblistes mettant en jeu plusieurs rectangles imposent la connaissance d'une position relative entre les éléments. Compte-tenu de la paraxialité, la seule donnée nécessaire au positionnement relatif de deux rectangles R_k et R_l sera spécifiée par le vecteur $\vec{O_k O_l} = \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}$ appelé *vect_Pos_Rel*, joignant les origines respectives O_k et O_l de ces rectangles.

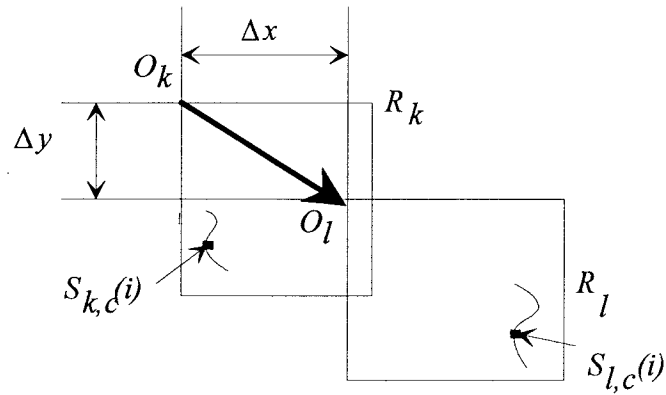


Figure 4.2

2.5 Opérations sur les rectangles

2.5.1 Réunion de deux rectangles

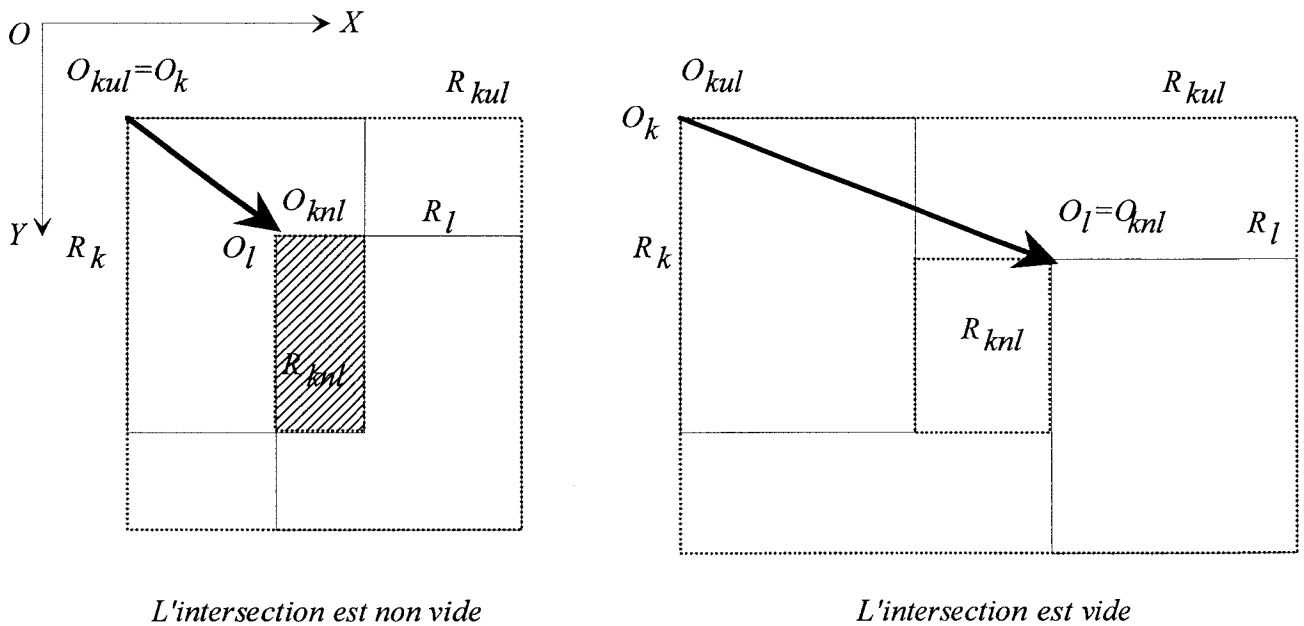


Figure 4.3

Soit $\overline{O M_{k \cup l}}$ la matrice du rectangle R_{kul} circonscrit à la réunion des deux rectangles R_k et R_l . Les sommets du rectangle de la réunion sont tels que l'origine O_{kul} est le point de R_{kul} se situant

à une distance la plus faible de l'origine O , et le sommet diamétralement opposé est celui se situant à une distance euclidienne la plus grande de l'origine O .

On peut alors établir:

$$\begin{aligned} \overline{\overline{OMk \cup l}} &= \overline{\overline{OMk}} \cup \overline{\overline{OMl}} = \begin{bmatrix} x_{ok} & x_{ok} + L_k \\ y_{ok} & y_{ok} + W_k \end{bmatrix} \cup \begin{bmatrix} x_{ol} & x_{ol} + L_l \\ y_{ol} & y_{ol} + W_l \end{bmatrix} \\ &= \begin{bmatrix} \min(x_{ok}, x_{ol}) & \max(x_{ok} + L_k, x_{ol} + L_l) \\ \min(y_{ok}, y_{ol}) & \max(y_{ok} + W_k, y_{ol} + W_l) \end{bmatrix} \end{aligned}$$

Remarque:

En supposant que l'on positionne le rectangle R_k de telle façon que $O_k = O$, alors

$$\overline{\overline{OkMk \cup l}} = \begin{bmatrix} \min(0, \Delta x) & \max(L_k, \Delta x + L_l) \\ \min(0, \Delta y) & \max(W_k, \Delta y + W_l) \end{bmatrix}$$

On obtient ainsi une formulation en coordonnées relatives.

2.5.2 Intersection de deux rectangles

Soit $\overline{\overline{OM_{k \cap l}}}$ la matrice du rectangle R_{knl} issu de l'intersection des deux rectangles R_k et R_l . En reprenant les définitions d'intersection d'intervalles, il vient :

$$\begin{aligned} \overline{\overline{OM_{k \cap l}}} &= \overline{\overline{OMk}} \cap \overline{\overline{OMl}} = \begin{bmatrix} x_{ok} & x_{ok} + L_k \\ y_{ok} & y_{ok} + W_k \end{bmatrix} \cap \begin{bmatrix} x_{ol} & x_{ol} + L_l \\ y_{ol} & y_{ol} + W_l \end{bmatrix} \\ &= \begin{bmatrix} \max(x_{ok}, x_{ol}) & \min(x_{ok} + L_k, x_{ol} + L_l) \\ \max(y_{ok}, y_{ol}) & \min(y_{ok} + W_k, y_{ol} + W_l) \end{bmatrix} \end{aligned}$$

L'expression précédente nous fait constater que l'origine O_{knl} est le sommet de R_{knl} situé à une distance la plus courte de l'origine la plus éloignée de O parmi O_k et O_l

Remarque :

Si $O = O_k$, alors

$$\overline{OkMk \cap l} = \begin{bmatrix} \max(0, \Delta x) & \min(L_k, \Delta x + L_l) \\ \max(0, \Delta y) & \min(W_k, \Delta y + W_l) \end{bmatrix}$$

On convient ici de l'existence d'un rectangle d'intersection, que les deux rectangles R_k et R_l soient disjoints ou non. Selon le cas, l'origine du rectangle R_{knl} sera le sommet situé à gauche ou à droite du rectangle d'intersection (cf. figure 4.3).

2.5.3 Signe des surfaces des rectangles réunion et intersection*a) Réunion*

Quelle que soit la position relative des deux rectangles, on peut montrer que la surface du rectangle de la réunion est positive:

$$\forall \overrightarrow{O O_k} \quad , \quad SR_{k \cup l} = [\max(l_k, \Delta x + L_l) - \min(0, \Delta x)] \times [\max(W_k, \Delta y + W_l) - \min(0, \Delta y)] \geq 0$$

b) intersection

Si les rectangles R_k et R_l sont disjoints, alors R_{knl} se situe à l'extérieur de R_k et R_l . Dans ce cas, O_{knl} est un sommet se situant à droite de R_{knl} , $l_{k \cap l, 0}$ est négative, mais $l_{k \cap l, 1}$ est toujours positive car l'origine O_{knl} a son ordonnée minimale. SR_{knl} est donc négative. On peut donc écrire les règles:

$$\text{si } R_k \cap R_l = \emptyset \text{ alors } SR_{k \cap l} < 0$$

$$\text{si } R_k \cap R_l \neq \emptyset \text{ alors } SR_{k \cap l} \geq 0$$

Si l'intersection se réduit à un rectangle dégénéré, alors S_{knl} est nulle

$$si \text{Int}(R_k \cap R_l) = \emptyset \text{ alors } SR_{k \cap l} = 0$$

Ayant défini les propriétés portant sur les matrices de rectangles paraxiaux, on peut maintenant passer aux opérateurs de concaténation et d'imbrication de formes irrégulières.

2.6 Changement de repère

- Soit R_k et R_l deux rectangles définis par leur matrice respective $\overline{OM_k}$ et $\overline{OM_l}$.

- Soit $\overrightarrow{O_k O_l}$ le vecteur connu de positionnement relatif de ces deux rectangles

Le placement des deux rectangles nous permet de définir le rectangle réunion R_{kul} propriétaire de l'origine image O_{kul} .

Les formulations que nous proposons ici, consistent à déterminer les coordonnées d'un point dans le repère local vers le repère lié au même côté de la réunion de deux formes. Ces formules sont présentées en Annexe D. On retiendra ici que le message envoyé à l'objet sommet d'un peigne réduit, effectuant la méthode algorithmique de changement de repère:

$$S_{k,c(i)/O_{kul}} := S_{k,c(i)/O_k} \text{ change_Repère_De: } \overline{OM_k} \text{ vers: } \overline{OM_{k \cup l}} \text{ côté: } c^1$$

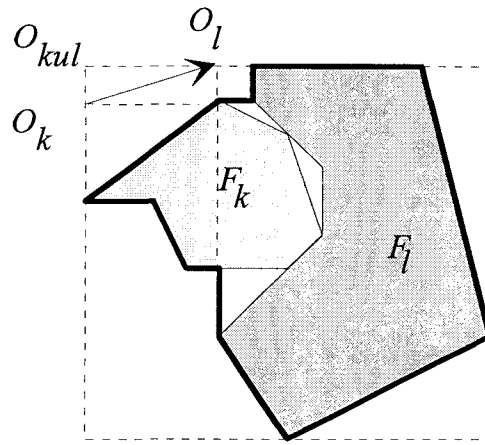
III Concaténation géométrique de deux formes

L'objectif de cette partie est de décrire la forme F_{kul} , résultante de l'enveloppe de la réunion géométrique des deux formes F_k et F_l . Cette opération est un élément-clef pour le domaine de commande, car il permet la construction du placement selon une procédure itérative de concaténation des formes.

3.1 Définition:

La concaténation des deux formes au sens des peignes réduits est l'opérateur capable de calculer les quatre peignes réduits issus de l'enveloppe de la réunion des sous contours apparents stricts des deux formes et ce pour une position relative donnée, l'intersection des deux contours apparents stricts pouvant être quelconque, c'est-à-dire vide ou non.

¹La syntaxe utilisée est celle du langage *Smalltalk* dans lequel l'application fut prototypée. Cependant, aucune considération sur la classe d'objets de type *Sommets* n'est présentée ici.



$$F_{kul} := F_k \text{ concatène: } F_l \text{ pos_Rel: } \xrightarrow{O_k O_l}$$

Figure 4.4

3.2 Convention:

Soient F_k et F_l deux formes définies par leurs peignes-réduits S_k et S_l , on notera par:

$$F_{kul} := F_k \text{ concatène: } F_l \text{ pos_Rel: } \xrightarrow{O_k O_l} \quad \text{le message envoyé au récepteur instance}$$

de *Forme_Model*, permettant de déterminer la forme dont le contour correspond à l'enveloppe de la concaténation des contours du récepteur F_k et de l'argument F_l .

3.3 Obtention :

Pour chaque côté c , le peigne résultant $S_{kul,c}$ s'obtient en effectuant les étapes suivantes:

Etape 1:

Conversion des coordonnées des points de $S_{k,c}$ et $S_{l,c}$ par rapport à la nouvelle origine $O_{kul,c}$

²La concaténation se situe au niveau des peignes, mais ce sont bien les instances F_k et F_l qui sont utilisées ici, sachant que la base de données associée à chaque forme contient toutes les données nécessaires

Etape 2

Pour chaque côté c , concaténation des points de chaque liste en tenant compte du masquage de certains points lors de l'association. Création de nouveaux points de contrôle pour définir la zone de jonction.

Explicitons chacune ces deux étapes

Etape 1:

Elle consiste tout d'abord à déterminer les rectangles réunion $\overline{\overline{OM_k \cup l}}$ et intersection $\overline{\overline{M_k \cap l}}$ des deux rectangles R_k et R_l par rapport à l'origine O_k . La connaissance de ces matrices permet ensuite de changer d'origine de repère depuis les origines locales O_k et O_l vers l'origine unique O_{kul} . Les coordonnées des points par rapport à l'origine O_{kul} s'obtiendront par les formules de passage définies précédemment.

On rappelle ainsi que les points de $S_{k,c}$ et $S_{l,c}$ seront connus dans le repère $O_{kul}XY$ après avoir effectué les méthodes algorithmiques:

$$S_{k,c(i)}/O_{kul} := S_{k,c(i)}/O_k \text{ change_Repère_De: } \overline{\overline{OM_k}} \text{ vers: } \overline{\overline{O M_{k \cup l}}}$$

$$S_{l,c(j)}/O_{kul} := S_{l,c(j)}/O_l \text{ change_Repère_De: } \overline{\overline{OM_l}} \text{ vers: } \overline{\overline{O M_{k \cup l}}}$$

Etape 2:

Elle consiste à supprimer les points cachés dans la zone d'intersection. Plusieurs possibilités sont offertes:

- a) Première solution

Exploiter la définition des peignes en ne gardant sur la zone d'intersection que les points dont la hauteur comparée est minimale. Le problème est que les sommets des deux peignes réduits ont rarement une projection identique sur le côté. Aussi cela oblige à réaliser une interpolation pour connaître les points non répertoriés dans la liste du peigne réduit à comparer. Une recherche des extrémités du segment portant le point à interpoler doit être effectuée en parcourant la liste, par exemple de façon dichotomique.

Soit L une liste de sommets d'un polygone et soit Q un point d'abscisse connu et d'ordonnée inconnue. La détermination de l'ordonnée de Q doit donc s'effectuer en cherchant à encadrer le point Q entre deux sommets consécutifs P et R de L de telle façon que $x_P \leq x_Q \leq x_R$

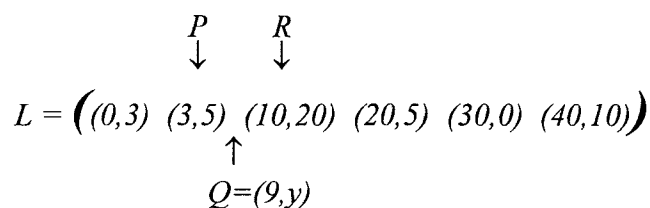


Figure 4.5: Positionnement d'un point d'ordonnée inconnue

Ceci peut être effectué par un algorithme de recherche dichotomique de complexité de l'ordre $O(\text{Log}(n))$, n étant le nombre de points dans la liste L [BRA 87].

Soit $n_{k,c}$ et $n_{l,c}$ le nombre de sommets de $S_{k,c}$ respectivement $S_{l,c}$, et dans le cas où l'intersection met en jeu $i_{k,c}$ et $i_{l,c}$ sommets, la procédure de suppression nécessitera:

$n_{k,c} + n_{l,c}$ tests d'appartenance à l'intervalle d'intersection.

$i_{k,c} \text{Log}(n_{l,c}) + i_{l,c} \text{Log}(n_{k,c})$ tests pour la fouille dichotomique

$(i_{k,c} + i_{l,c})$ calculs d'interpolation linéaire discrète.

$(i_{k,c} + i_{l,c})$ comparaisons de hauteurs

-b) Deuxième solution:

Elle consiste à créer une nouvelle liste S_{kul} en ne retenant que les points visibles. La détection et l'élimination des points non-visibles est effectuée lorsqu'apparaît une alternance de l'étiquetage³ lors de la concaténation d'un nouveau point. L'algorithme effectue donc les opérations suivantes:

³ Étiquetage d'un point: action d'associer une étiquette à un point. L'étiquette est ici a ou b selon l'appartenance du point à une de ces listes: A ou B.

- On détermine le point de A ou de B d'abscisse minimale: soit P ce point, Q le premier point de l'autre liste.

Pas N°4

- si P a même étiquette que le dernier point de $S_{kul,c}$

alors: (*pas de point non-visible*)

- on ajoute P à $S_{kul,c}$

- on supprime P de sa liste de provenance

sinon: (*alternance d'étiquetage et donc changement de forme*)

- si $Y_P < Y_Q$ alors (cas de la figure 4.6):

- on ajoute P à $S_{kul,c}$

-on supprime P de sa liste de provenance

- Tq $X_Q \leq X_{O_{knl,c}} + l_{knl,c}$ supprimer Q

(*on supprime tous les points non-visibles jusqu'à la fin de l'intersection des projections des peignes réduits*)

sinon: (cas de la figure 4.7)

- on ajoute Q à $S_{kul,c}$

-on supprime Q de sa liste de provenance

- Tq $X_P \leq X_{O_{knl,c}} + l_{knl,c}$ supprimer P

(*on supprime tous les points non-visibles jusqu'à la fin de l'intersection des projections des peignes réduits*)

- retour au pas N°3

Pas N°5

arrêt de la concaténation

Soit $(i_{k,c} + i_{l,c})$ le nombre total de points contenus dans l'intervalle d'intersection. La procédure de suppression nous amène ici à calculer:

$\max(n_{k,c}, n_{l,c})$ comparaisons pour la détermination de l'abscisse minimale.

$2^*(n_{k,c} + n_{l,c})$ suppressions ou ajouts de points.

$(i_{k,c} + i_{l,c})$ comparaisons d'abscisses ou d'ordonnées lors de l'alternance d'étiquette

Dans cette méthode on s'affranchit de la recherche dichotomique, qui se révèle très pénalisante dès que le nombre de sommets présents dans l'intersection augmente. Sans approfondir la comparaison de complexité avec la précédente procédure, le choix se portera évidemment vers cette méthode de complexité linéaire.

Modification de l'étape 2

Afin de définir le contour de l'enveloppe de la réunion au niveau de la jonction, nous avons modifié l'algorithme précédent. L'idée consiste à créer des points afin de reconstituer la frontière à la jonction des deux formes. Ici, deux cas peuvent surgir:

1) La projection de l'intersection sur le côté c est non vide ($l_{knl,c} > 0$)

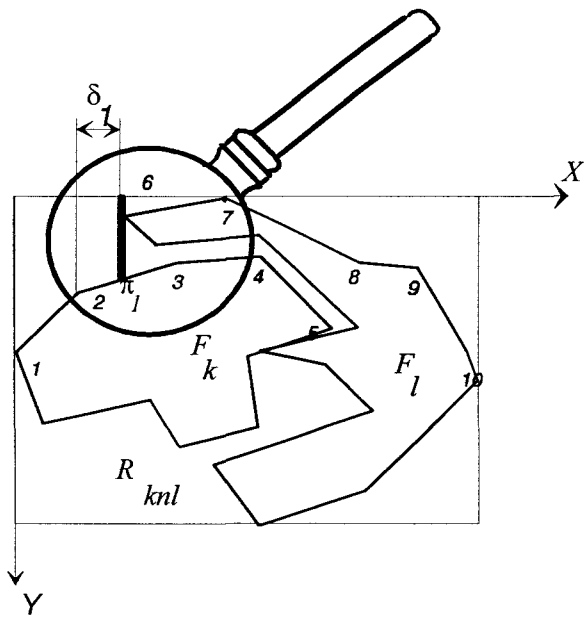


Figure 4.8.a

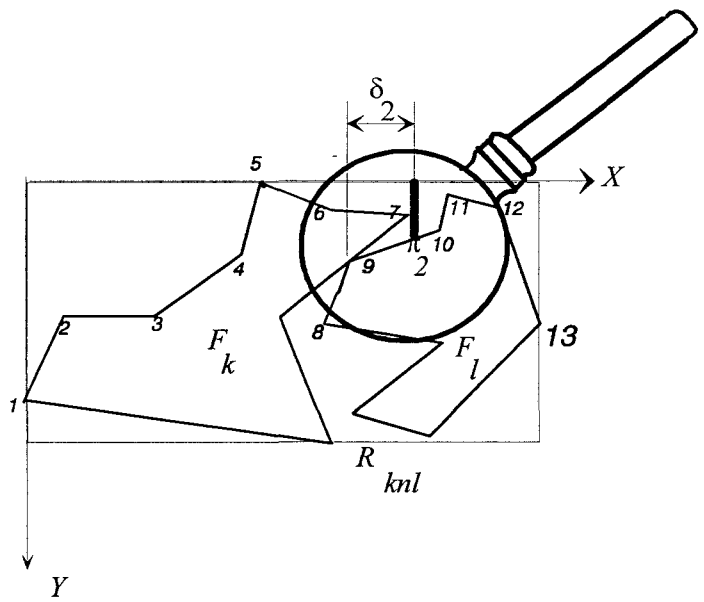


Figure 4.8.b

Lorsque deux formes s'imbriquent, la frontière de la jonction visible des deux contours est rarement située à l'emplacement d'un sommet du peigne. Il convient alors de créer un nouveau point de contrôle caractérisant la frontière. A partir de l'abscisse du point responsable de l'alternance d'étiquetage, le nouveau point π pourra être déterminé simplement en effectuant une interpolation linéaire en cette abscisse sur le segment porteur partiellement occulté par l'intersection. Selon la configuration, deux cas peuvent se présenter:

- La frontière est du type front montant. Les extrémités du segment porteur sont donc le dernier point ajouté dans la liste $S_{kul,c}$ et son successeur dans la liste de provenance. (exemple du point π_1 dans le segment [2,3] de la figure 4.8.a)
- La frontière est du type front descendant. Les extrémités du segment porteur sont donc le dernier point caché et son successeur dans la liste dans la liste de provenance (exemple du point π_2 dans le segment [9,10] de la figure 4.8.b).

L'algorithme détaillé de procédure de suppression de points est donné dans l'Annexe E

2) La projection de l'intersection sur le côté c est vide ($l_{knl,c} < 0$)

Il convient alors de décrire l'espace laissé entre les deux formes en créant de nouveaux points. Pour cela, on insère deux points $\pi_{1,c}$ et $\pi_{2,c}$ entre les listes $S_{k,c}$ et $S_{l,c}$ de telle façon que ces points se trouvent à la limite du côté opposé, caractérisant ainsi un puits dans lequel il est possible d'y introduire une autre forme qui conviendrait. L'espace n'est donc pas perdu.

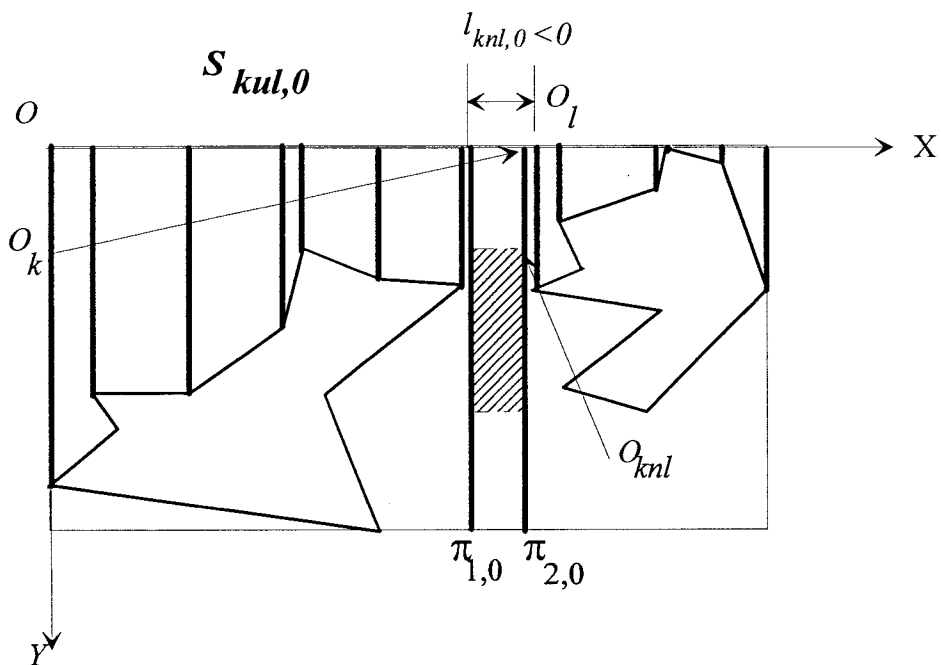


Figure 4.9

Les points $\pi_{1,c}$ et $\pi_{2,c}$ seront construits en prenant leurs abscisses égales à une unité supérieure (resp. inférieure) à l'abscisse du dernier point supprimé lorsque la liste courante devient vide (resp premier point ajouté de l'autre liste). Les ordonnées sont égales à la longueur du côté adjacent

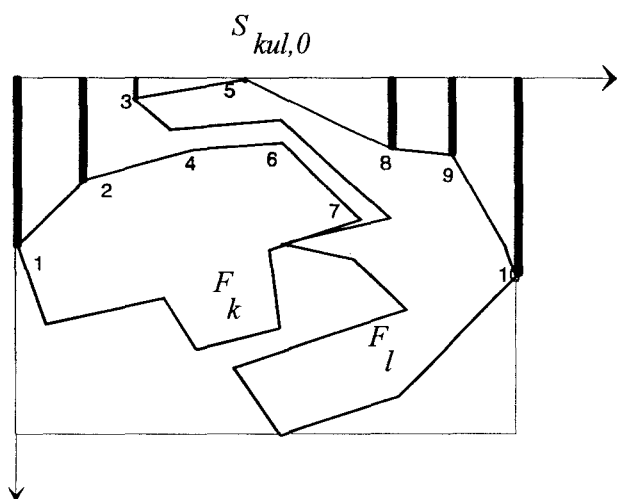
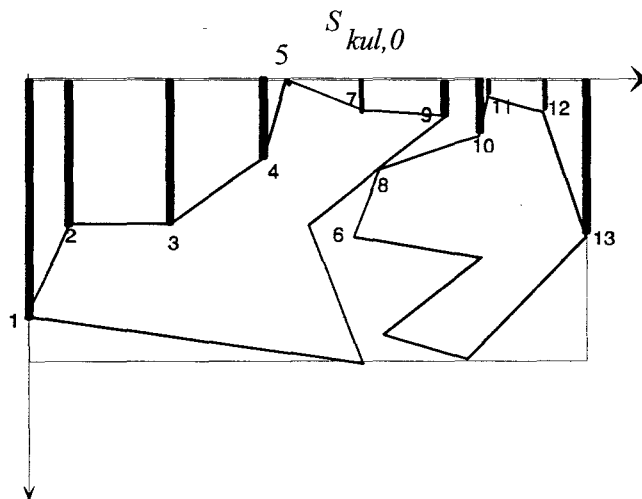


Figure 4.6



1,2,3,... succession des points par abscisses croissantes

Figure 4.7

Pas N°1

On crée deux nouvelles listes de travail telles que :

$$A=S_{k,c} \text{ et } B= S_{l,c}$$

chaque point étant affecté d'une étiquette a ou b

Pas N°2

On introduit dans S_{kul} , le point de A ou de B d'abscisse minimale, et on le supprime de sa liste de provenance.

Pas N°3

- si les deux listes sont vides, aller au pas N°5
- si une liste est vide alors:
 - on ajoute le premier point de l'autre liste à S_{kul} et on le supprime de sa liste de provenance
 - aller au pas N°3

$$\begin{array}{l}
\text{si } S_{k,c} \text{ est avant } S_{l,c}, \text{ alors } \left\{ \begin{array}{l} \pi_{1,c} = (l_{k,c} + 1, l_{k \cup l, c+1}[4]) \\ \pi_{2,c} = (l_{k,c} + |l_{k \cap l, c}| - 1, l_{k \cup l, c+1}[4]) \end{array} \right\} \\
\text{ce qui donne} \\
\text{si } S_{k,c} \text{ est après } S_{l,c}, \text{ alors } \left\{ \begin{array}{l} \pi_{1,c} = (l_{l,c} + 1, l_{k \cup l, c+1}[4]) \\ \pi_{2,c} = (l_{l,c} + |l_{k \cap l, c}| - 1, l_{k \cup l, c+1}[4]) \end{array} \right\}
\end{array}$$

Le codage des espaces vides sera d'un grand intérêt lorsqu'on aura à traiter le placement sur tissu à motifs. En effet, la contrainte de raccord engendre généralement des disjonctions entre les rectangles circonscrits des différentes formes. Les espaces vides doivent donc être représentés pour éviter d'en perdre l'accessibilité.

IV Evaluation de la complémentarité

Nous venons de démontrer dans la partie précédente, qu'il était possible d'obtenir la description par les peignes-réduits d'une forme issue de la concaténation de deux formes. Utile à la construction itérative du placement, elle présuppose la connaissance du vecteur de positionnement relatif entre deux éléments. La détermination de ce vecteur sera donc l'objet des développements suivants.

Les calculs présentés dans cette partie seront toujours motivés par la volonté d'établir une méthode permettant de déterminer les coordonnées relatives d'un couple de formes placées en imbrication optimale vis-à-vis d'un critère. Cependant l'interprétation optimale n'est pas unique car elle dépend en réalité de la nature du problème. Dans le cas de l'optimisation du placement, et si on adopte l'approche bifocale locale-globale, la fonction de but n'est pas systématiquement identique à celle du niveau global. L'imbrication optimale au niveau local signifiera plutôt surface occupée minimale, intersection maximale ou surface inaccessible minimale, alors que globalement on cherchera à minimiser la perte de tissu dans la bande en construction. Autant d'objectifs différents qui invitent à définir autant de métriques différentes. Naturellement, compte-tenu de la nature du code et des contraintes du problème, la recherche d'une imbrication optimale sous-entend la mise en relation de deux peignes issus de deux formes indépendantes, dont il faut déterminer un positionnement relatif satisfaisant les contraintes.

4.1 Optimisation de l'Imbrication

Nous entendons par optimisation de l'imbrication selon une direction, la juxtaposition de deux formes en minimisant une distance d'imbrication I . Cela se traduit par les règles:

$$\underset{\text{positions relatives}}{\text{Min}} \quad I(F_k, F_l, c) \quad \text{et} \quad \text{Int}(F_k \cap F_l) = \{\emptyset\}$$

Elle traduit le fait que la mise en correspondance du côté c d'une forme F_k et du côté $c+2[4]$ d'une forme F_l doit se faire au sens de la minimisation d'un indice I dont il sera fait mention ultérieurement. La contrainte de disjonction doit naturellement être satisfaite, ainsi que le respect des dimensions global de la bande.

L'optimisation de l'imbrication a ici pour objet de déterminer le vecteur de positionnement relatif $\overrightarrow{O_k O_l}$ des deux formes à concaténer. Celui-ci sera donc obtenu par activation de la méthode algorithmique dont le message sera:

$$\overrightarrow{O_k O_l} = F_k \text{ recherche_Imbrication: } F_l \text{ du côté: } c \text{ dans: rectangle_Maximal}$$

4.2 Coordonnées relatives entre deux peignes

La mise en relation de deux peignes nécessite un paramétrage du positionnement relatif entre les origines locales $O_{k,c}$ et $O_{l,c+2[4]}$ tel que:

$$\overrightarrow{O_{k,c} O_{l,c+2[4]}} = \begin{pmatrix} d \\ P_i \end{pmatrix} \text{ si } c \text{ est pair ou } \begin{pmatrix} P_i \\ d \end{pmatrix} \text{ si } c \text{ est impair, avec:}$$

- $d \in Z$ est le **décalage relatif** selon $O_{k,c}X$ entre les origines $O_{k,c}$ et $O_{l,c+2[4]}$ des peignes $S_{k,c}$ et $S_{l,c+2[4]}$. On peut donc écrire:

$$d = XO_{l,c+2[4]} - XO_{k,c} \text{ si } c \text{ est pair.}$$

$$d = YO_{l,c+2[4]} - YO_{k,c} \text{ si } c \text{ est impair.}$$

- $P_i \in Z$ est la profondeur d'imbrication ou de chevauchement des rectangles circonscrits selon $O_{k,c}Y$. Elle est définie par:

$$P_i = YO_{l,c+2[4]} - YO_{k,c} \text{ si } c \text{ est pair.}$$

$$P_i = XO_{l,c+2[4]} - XO_{k,c} \text{ si } c \text{ est impair.}$$

P_i est positif lorsque $O_{l,c+2[4]}$ est placé après $O_{k,c}$ en circulant dans le sens $O_{k,c}Y$, c'est à dire lorsque l'intersection des rectangles R_k et R_l est non vide.

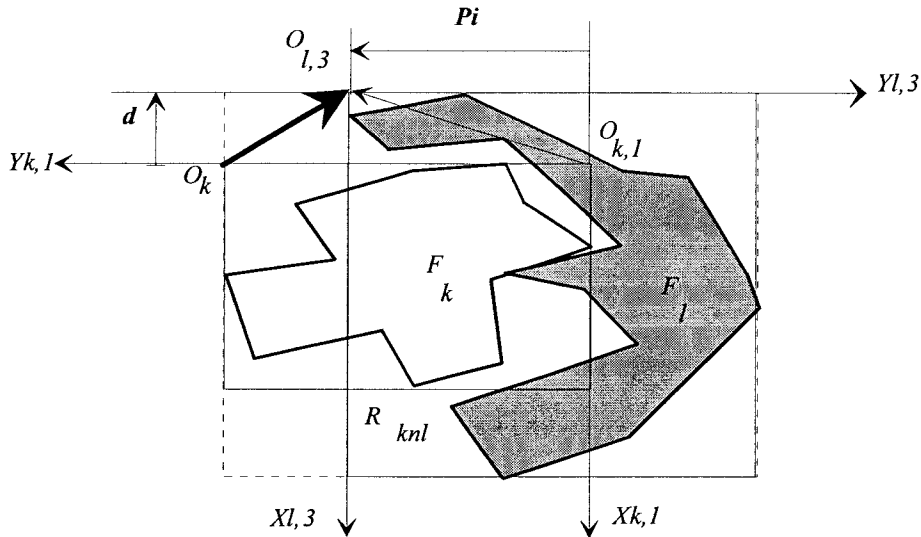


Figure 4.10

4.3 Changement de repère

Selon le numéro du côté de mise en correspondance, les composantes du vecteur $(d, P_i)^t$ sont permutées afin de satisfaire la cohérence de définition de ces grandeurs. Pour convertir le positionnement relatif entre deux peignes désigné par $\overrightarrow{O_{k,c}O_{l,c+2[4]}}$ à un positionnement relatif entre les deux rectangles R_k et R_l exprimé par le vecteur $\overrightarrow{O_kO_l}$, on établira une formule de passage. En utilisant la décomposition vectorielle, il vient:

$$\overrightarrow{O_kO_l} = \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} = \overrightarrow{O_kO_{k,c}} + \overrightarrow{O_{k,c}O_{l,c+2[4]}} - \overrightarrow{O_lO_{l,c+2[4]}}$$

On retrouve ici le vecteur de position relative entre les origines locales:

$$\overrightarrow{O_{k,c}O_{l,c+2[4]}} = \begin{pmatrix} d \\ P_i \end{pmatrix} \text{ si } c \text{ est pair}$$

$$= \begin{pmatrix} P_i \\ d \end{pmatrix} \text{ si } c \text{ est impair}$$

Ainsi que $\overrightarrow{O_k O_{k,c}}$, $\overrightarrow{O_l O_{l,c+2[4]}}$ obtenus directement à partir des matrices de rectangle $\overline{\overline{O M_k}}$ et $\overline{\overline{O M_l}}$

Une fois les grandeurs (d, P_i) connues, on obtiendra le vecteur de positionnement relatif $\overrightarrow{O_k O_l}$ en appliquant le message:

$$\overrightarrow{O_k O_l} := (d, P_i) \text{ peigne_Vers_Rect: } \overline{\overline{O M_k}} \text{ et: } \overline{\overline{O M_l}} \text{ selon_côté: } c$$

4.4 Contrainte de disjonction

Il s'agit de définir les conditions vérifiant la relation $Int(F_k \cap F_l) = \{\emptyset\}$ signifiant que la disjonction est vérifiée tant que l'intérieur de l'intersection des deux formes est vide. Pour cela, les considérations suivantes vont nous permettre d'énoncer les principes de traitement visant à satisfaire cette contrainte tout en cherchant à optimiser les calculs.

Définition:

Les limites extrêmes du non-recouvrement sont établies lorsque les sous-contours de deux formes distinctes sont en contact.

Supposons que les axes $O_{k,c}X$ et $O_{l,c+2[4]}X$ soient superposés (cf. figure 4.11.a). Soient π_1 et π_2 deux points appartenant respectivement à $\tilde{A}(C_{k,c})$ et $\tilde{A}(C_{l,c+2[4]})$ et tels que la distance euclidienne $\overline{\pi_1 \pi_2}$ soit minimale parmi toutes les distances $\overline{\pi_k \pi_l}$ de $\tilde{A}(C_{k,c})$ et $\tilde{A}(C_{l,c+2[4]})$ ayant la même projection sur l'intervalle commun $[a, b]$. On dira que les deux formes sont en contact lorsque, pour une position relative fixée, d , des origines $O_{k,c}$ et $O_{l,c+2[4]}$, la profondeur d'imbrication P_i est égale à $\overline{\pi_1 \pi_2}$ (cf. figure 4.11)

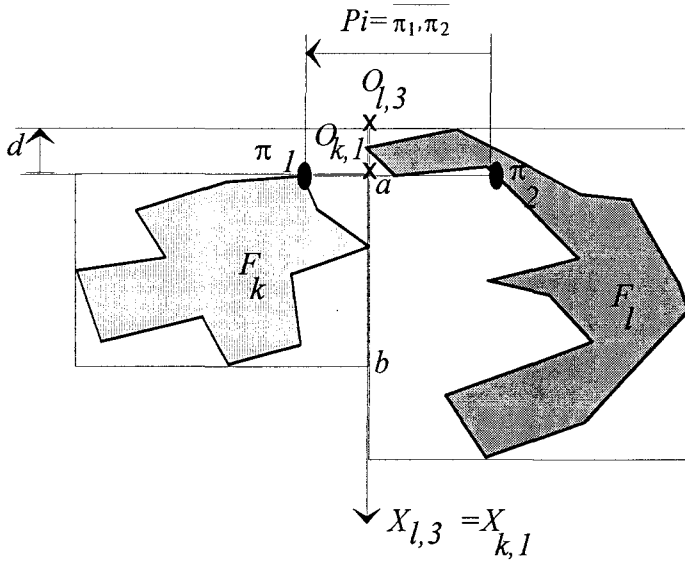


Figure 4.11a

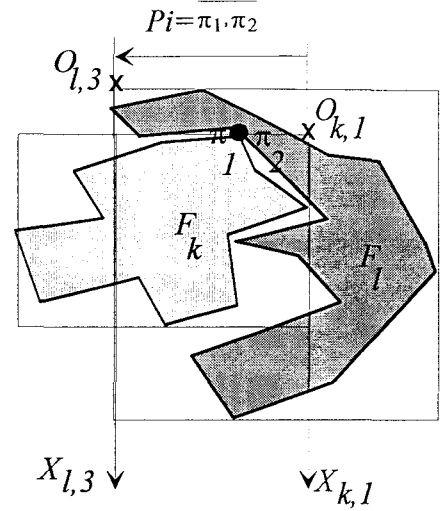


Figure 4.11b

Conséquence :

La contrainte de disjonction sera satisfaite si pour un décalage d constant entre $O_{k,c}$ et $O_{l,c+2[4]}$, la profondeur d'imbrication vérifie l'inéquation:

$$Pi(d) \leq \overline{\pi_1, \pi_2}$$

avec

$$\overline{\pi_1, \pi_2} = \underset{proj \pi_k / [a,b] = proj \pi_l / [a,b]}{Min} \quad d(\pi_k, \pi_l) = \underset{proj \pi_k / [a,b] = proj \pi_l / [a,b]}{Min} (y_{\pi_k} + y_{\pi_l})$$

où y_{π_k} et y_{π_l} sont les hauteurs des sommets π_k et π_l appartenant respectivement aux peignes $S_{k,c}$ et $S_{l,c+2[4]}$.

Cette définition nécessite a priori d'effectuer la recherche de ce minimum pour tous les points des sous-contours dont les projections appartiennent à l'intervalle d'intersection $[a,b]$. Cependant, en exploitant les propriétés géométriques des polygones, nous pourrions simplifier sensiblement la procédure de recherche.

4.5 Simplification de la vérification de la contrainte

Soit $[A,B]$ et $[C,D]$ deux segments appartenant à deux polygones $P1$ et $P2$ disjoints, tels que ces segments soient portés par deux droites $D1$ et $D2$ sécantes.

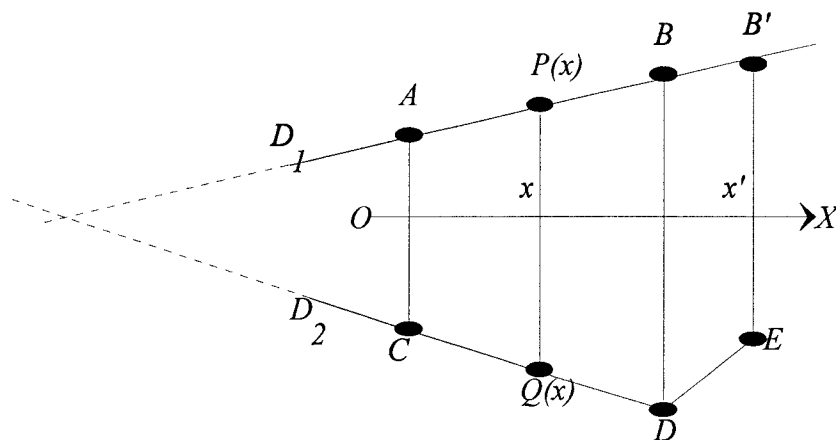


Figure 4.12

Soit B' un autre point de $D1$ et de projection x' sur Ox , et E un autre sommet de $P2$ également de projection x' sur Ox , consécutif à D , tel que D est un point de concavité. Supposons que AC et BD soient perpendiculaires à OX . Soient $P(x)$ et $Q(x)$ deux points de AB et CD de même abscisse x

Axiome 1 :

En supposant que A et B soient deux sommets consécutifs d'un peigne réduit, et que C et D soient les points correspondants dans le sous-contour de l'autre forme:

supposons $\overline{AC} \leq \overline{BD}$
 alors $\forall x$ telque $projA_{OX} \leq x \leq projB_{OX}$, $\overline{AC} \leq \overline{P(x)Q(x)}$

Conséquence :

Soit à calculer:

$$\overline{PQ}_{min} = \underset{x}{Min} \overline{P(x)Q(x)} \text{ pour } P(x) \in [A,B] \text{ et } Q(x) \in [C,D]$$

comme d'après l'axiome 1, $\overline{Min(AC, BD)} \leq \overline{P(x)Q(x)}$ il est donc inutile de calculer toutes les distances $\overline{P(x)Q(x)}$ pour $P(x) \in]A, B[$ et $Q(x) \in]C, D[$ afin de déterminer le minimum de cette distance.

Interprétation

Cette propriété montre qu'il est inutile de calculer la fonction du *Min* pour les points situés entre deux sommets consécutifs d'un peigne-réduit, puisque la distance minimale se situe au niveau de l'un de ces sommets.

Axiome 2 :

Avec les hypothèses précédentes,

$$D \text{ est un point de concavité} \Leftrightarrow \left(\text{et } \begin{array}{l} \overline{BD} > \overline{AC} \\ \overline{BD} > \overline{B'E} \end{array} \right)$$

Conséquence :

Il est inutile de calculer la distance \overline{PQ} , si P (ou Q) est un point de concavité.

Interprétation

Si l'on suppose connue la concavité en un point, le calcul de la fonction *Min* est inutile en ce point.

Finalement, ces résultats conduisent à une réduction du nombre de tests de minimum aux seuls sommets convexes. Il ne reste alors qu'à rassembler à l'intérieur d'un même algorithme le principe de vérification de la contrainte.

4.6 Principe de vérification de la contrainte de disjonction :

Entrée: $\{d, S_{k,c}, S_{l,c+2[4]}\}$

Étape 1:

Déterminer l'intervalle $[a, b]$, projection sur l'axe $O_{k,c}X$ de l'intersection de deux rectangles circonscrits:

$$[a, b] = \left[\max(o, d) ; \min(l_{k,c}, l_{l,c+2[4]} - d) \right]$$

Etape 2:

Pour tous les points convexes P de $S_{k,c}$ d'abscisse x_P appartenant à $[a,b]$, chercher la somme minimale $S_1 = \min_{x_P \in [a,b]} (y_P + y_Q)$, où Q est le point de $S_{l,c+2[4]}$ de même projection sur

$O_{k,c}X$. L'ordonnée Y_Q du point Q est obtenu par interpolation.

Idem pour tous les points convexes Q de $S_{l,c+2[4]}$ pour obtenir $S_2 = \min_{x_Q \in [a,b]} (y_Q + y_P)$,

P est cette fois un point obtenu par interpolation.

On prendra finalement le minimum des deux valeurs obtenues précédemment:

$$Pi_{max}(d) = \text{Min}(S_1, S_2)$$

Remarque :

Dans l'étape 2, on est souvent amené à déterminer par interpolation les coordonnées du point se situant sur la même projection que le sommet appartenant au peigne-réduit opposé. Cette technique nécessite avant tout une recherche (par dichotomie, par exemple) de l'abscisse des sommets encadrant le point à interpoler. Mais cette recherche, relativement fréquente lors du déroulement de l'algorithme, est globalement complexe. On préfère donc utiliser un second principe.

Etape 2 modifiée:

On sélectionne un à un les sommets d'une première liste ($S_{k,c}(i)$ par exemple) :

- Si le sommet sélectionné appartient à l'intervalle d'intersection $[a,b]$ et s'il est convexe (test direct dans la base de données associée à la forme), alors on effectue une incrémentation d'un compteur j pointant sur les sommets $S_{l,c+2[4]}(j)$ de l'autre liste jusqu'à ce que ce que l'un de ces sommets ait son abscisse supérieure à l'abscisse du sommet de la première liste. En prenant aussi le prédécesseur du sommet détecté, on dispose d'un encadrement et on peut ainsi réaliser une interpolation linéaire discrète. On procède ainsi pour tous les sommets i de la première liste inclus dans l'intervalle $[a,b]$. Naturellement le compteur j n'est pas réinitialisé, puisque les valeurs suivantes sont forcément supérieures. Durant la procédure, on mémorise la distance minimale séparant les deux peignes. On réitère le principe en sélectionnant cette fois les sommets de l'autre liste. L'Annexe F présente la recherche de la profondeur d'imbrication maximale.

Cette méthode algorithmique est activée par le message:

$$Pi := F_k \text{ disjonction_avec: } F_l \text{ côté: } c \text{ décalage: } d$$

4.7 Indices de complémentarité

L'algorithme de respect de la contrainte de disjonction que nous venons de développer permet du même coup de lier la variable P_i à la variable d , c'est à dire la profondeur d'imbrication avec le décalage. Pour définir une position relative optimale ou proche de l'optimum pour chaque couple de formes à assembler, il ne reste qu'à déterminer la valeur de d minimisant un critère d'occupation surfacique. Cet objectif nous conduit naturellement à discuter du critère surfacique à retenir et plus tard, de la méthode d'optimisation. Successivement, nous analyserons les différents critères potentiellement utilisables afin d'établir notre choix.

4.7.1 Distance quadratique inter-peigne

La première approche consiste à établir une mesure de la qualité de la mise en correspondance des contours $\tilde{A}(C_{k,c})$ et $\tilde{A}(C_{l,c+2[4]})$. Une technique possible est basée sur l'évaluation d'une distance quadratique moyenne entre la courbe définie par $S_{k,c}$ et celle décrite par le peigne dual $S_{l,c+2[4]}^*$. Le résultat est représentatif de la similarité des contours sur la zone commune.

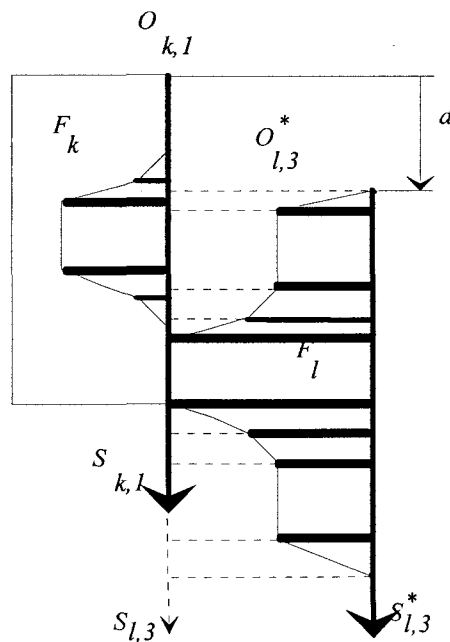


Figure 4.13

En superposant les axes $O_{k,c}X$ et $O_{l,c+2[4]}^*X$, on obtient (figure 4.14).

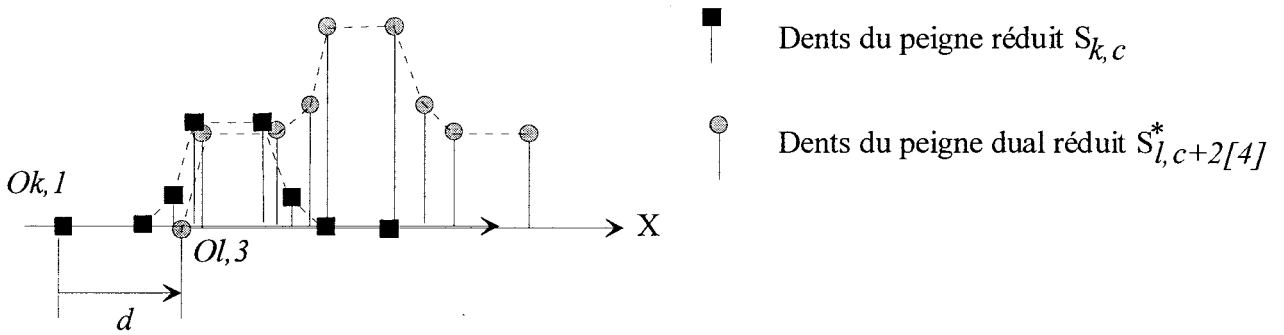


Figure 4.14

Nous désignerons par $I_l(F_k, F_l, c, d)$ la fonction qui évalue la distance quadratique inter-peigne lorsque le côté c de F_k est en correspondance avec le côté $c+2[4]$ de la forme F_l et ce pour un décalage d fixé. L'expression mathématique aura la forme:

$$I_l(F_{k,c}, F_{l,c}, d) = \frac{l}{l_{k \cap l, c}} \times \sum_{\text{proj}P / O_{k,c}X = \text{proj}Q / O_{l,c}X} (y_P - y_Q)^2$$

avec $\left\{ \begin{array}{l} P \in S_{k,c} \\ \text{ou} \\ Q \in S^*_{l,c+2[4]} \end{array} \right\}$

Cet indice calcule la somme quadratique de toutes les différences une à une des dents des peignes $S_{k,c}$ et $S^*_{l,c+2[4]}$. Cette somme est ensuite normalisée par la longueur de l'intervalle d'intersection.

Par cette même approche, on peut aussi chercher à minimiser la dissemblance d'appariement, c'est à dire la surface moyenne comprise entre les deux portions de contours de projection commune. Ceci part du raisonnement intuitif visant à minimiser les trous situés entre les formes pour maximiser le rapport $\frac{\text{surface utile}}{\text{surface occupée}}$. Ce calcul est donné par :

$$I_2(F_k, F_l, c, d) = \frac{1}{l_{k \cap l, c}} \times \sum_{\substack{[(proj P / O_{k,c} X = proj R / O_{k,c} X) \\ < \\ (proj Q / O_{k,c} X = proj S / O_{k,c} X)] \\ \text{avec } P, Q, R, S / \\ \text{et} \\ (P, Q \in S_{k,c}) \wedge (R, S \in S^*_{l,c+2[4]}} \left| \frac{(y_P - y_Q)(x_P - x_Q)}{2} \right| - \left| \frac{(y_R - y_S)(x_R - x_S)}{2} \right| \geq 0$$

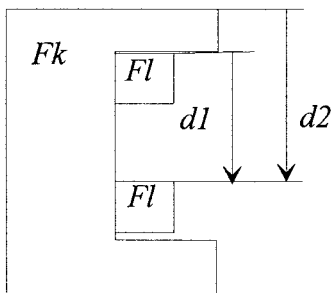
Ces deux mesures, bien que qualitativement intéressantes, conduisent à une complexité peu engageante d'autant qu'elles sont à calculer pour toutes les valeurs potentielles du décalage d .

4.7.2 Surface du rectangle d'intersection

Cet indice répond à l'approche qui consiste à mesurer la surface algébrique du rectangle d'intersection R_{knl} . Celle-ci se calcule facilement après avoir vérifié la contrainte de disjonction et donc obtenu la valeur de Pi .

$$I_3(F_k, F_l, c, d, Pi) = SR_{k \cap l}(Pi, d) \in \mathbb{Z}$$

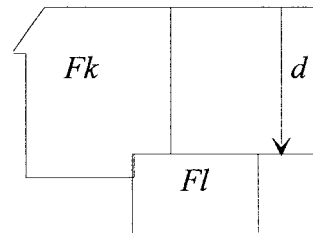
Pour chaque valeur de d et de Pi correspondante, et après un changement de repère permettant d'obtenir le vecteur de positionnement relatif, on peut en déduire la surface du rectangle d'intersection $SR_{k \cap l}$. Cependant, si la maximisation de cet indice algébrique favorise effectivement l'imbrication de deux formes, par contre elle peut également engendrer un indéterminisme lorsque, pour un sous-ensemble des valeurs possibles de d , $SR_{knl,c}$ est maximal. De plus, la maximisation de cet indice peut parfois fournir un résultat indésirable, si l'on ne tient pas compte des dimensions de l'association obtenue (cf. figure 4.15).



Indéterminisme sur d

Figure 4.15.a

$$I(d1) = I(d2)$$



Situation indésirable

Figure 4.15.b

4.7.3 Couloir de pénétration

La situation présentée figure 4.15.a constitue un cas particulier, dont on peut s'affranchir en convenant de choisir le décalage d minimum. Mais cette règle n'est pas toujours satisfaisante pour les cas où le premier minimum ne correspond pas à un tassement maximal. La figure 3.16 présente une telle situation. Là, les décalages $d1$ et $d2$ donnent une valeur maximale de SR_{knl} . Avec la règle précédente, c'est $d1$ qui est retenue. Pourtant $d2$ occasionnerait un tassement supérieur. Pour favoriser le tassement, on peut chercher à maximiser la surface du "couloir de pénétration", dessiné par le rectangle de la plus petite forme.

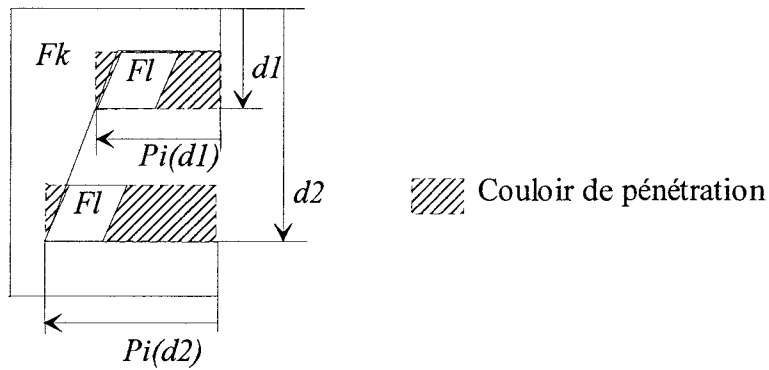


Figure 4.16

On définit donc le couloir de pénétration par :

$$I_4(F_k, F_l, c, d, Pi) = l_{k \cap l, c} \times Pi(d) \in Z$$

Si I_4 fait apparaître plusieurs maxima locaux, on garde la valeur correspondante au décalage minimal.

Mais cette distance ne supprime pas les situations indésirables du type escalier, illustré par la figure 4.15. Pour cela, on doit tenir compte des dimensions de l'association, ce que propose la surface du rectangle circonscrit de la concaténation.

4.7.4 Surface du rectangle de la concaténation

Cet indice mesure la surface algébrique du rectangle circonscrit à la réunion des deux formes SR_{kul} . Cette mesure pourra se calculer par la relation :

$$I_5(F_k, F_l, c, d, Pi) = SR_{k \cup l} \in Z$$

Pour une valeur de d , et après avoir vérifié la contrainte de disjonction, la connaissance de $P_i(d)$ permet de calculer $\vec{O_k O_l} = \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}$ afin d'en déduire la matrice $\overline{\overline{O_k M_{k \cup l}}}$ et finalement SR_{kul} .

Comme pour le rectangle d'intersection, l'optimisation de la surface du rectangle de la concaténation peut compter plusieurs minima locaux, induisant de nouveau un indéterminisme sur la valeur de d . C'est le cas par exemple lorsqu'une petite forme vient se loger complètement à l'intérieur du rectangle circonscrit d'une grande forme.

Les considérations de tassement concernant le couloir de pénétration sont également applicables à cette approche. On propose donc l'amélioration suivante.

4.7.5 Surface modifiée du rectangle circonscrit

Cette distance exprime la différence entre la surface du rectangle circonscrit et la surface du couloir de pénétration. On la définit donc par :

$$I_6(F_k, F_l, c, d, P_i) = I_5(\cdot) - I_4(\cdot) \in \mathbb{Z}^+$$

Cet indice a l'avantage de tenir compte des dimensions de l'association de deux formes tout en évitant des situations fréquentes d'indéterminisme. La minimisation de I_6 tend à minimiser la surface du rectangle circonscrit, tout en maximisant le couloir de pénétration. C'est cet indice qui sera utilisé.

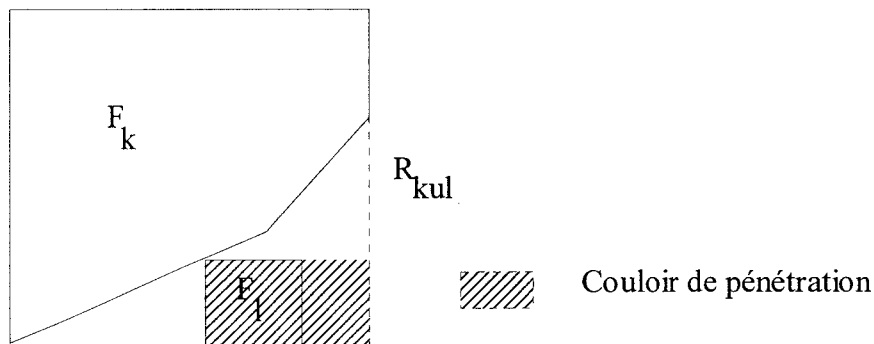


Figure 4.17

V Optimisation locale

5.1 Choix du critère

L'opération consiste à optimiser un des indices choisis parmi ceux évoqués précédemment. Selon l'indice, il s'agit de le minimiser : $I1, I2, I5, I6$ ou de le maximiser : $I3, I4$ par rapport à ses variables d et Pi .

Pour notre problème, s'agissant d'adopter une stratégie locale-globale, l'indice $I6$ a été choisi compte tenu de sa généralité.

Nous chercherons donc à déterminer Pi et d tel que:

$$I_6^{min}(F_k, F_l, c) = \underset{d, Pi}{MIN} I_6(F_k, F_l, c, d, Pi)$$

Nous sommes donc en présence de l'optimisation d'une fonction discrète à deux variables. Cependant, dans le cas du placement sur tissu uni on peut affirmer que pour minimiser l'espace occupé, les formes doivent être naturellement en contact l'une contre l'autre. $Pi(d)$ étant connu après avoir vérifié la contrainte de disjonction entre les couples de formes à imbriquer successivement, l'optimisation ne porte plus que sur la variable d . Ainsi, les variables Pi et d sont couplées. On a donc:

$$I_6^{min}(F_k, F_l, c) = \underset{d \in [\text{domaine de recherche}]}{MIN} I_6(F_k, F_l, c, Pi_{max}(d), d)$$

L'optimisation de l'imbrication suppose que le calcul de l'indice de complémentarité ainsi que la vérification de la contrainte de disjonction pour chaque position candidate ont été effectués. La complexité de la recherche est évidemment proportionnelle au nombre de positions candidates, c'est à dire de l'espace de recherche qu'il convient naturellement de diminuer.

5.2 Réduction du domaine de recherche

Dans la suite, on aura donc à exprimer un ensemble de relations dans le but unique de réduire le nombre d'itérations dans la recherche du minimum de l'indice de complémentarité retenu. Cependant, notre application ayant vocation de pouvoir traiter les deux types de tissu, uni ou à motif, il convient donc ici de faire la distinction. Selon le cas, les méthodes d'optimisation seront différentes puisque les domaines de définition seront eux-mêmes différents.

5.3 Réduction de l'intervalle de recherche dans le cas de matériau uni

5.3.1 Bornes limites de l'intervalle de recherche

Supposons que l'on cherche à minimiser la surface $SR_{k \cup l}$ du rectangle circonscrit de la réunion de F_k et F_l et que celles-ci soient en contact par leurs côtés respectifs c et $c+2[4]$.

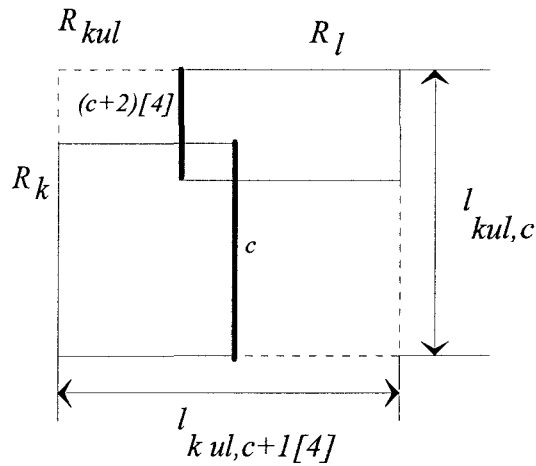


Figure 4.18

Selon la position relative des deux rectangles, les dimensions du rectangle exinscrit résultant de la concaténation des deux formes peuvent avoir comme valeurs limites:

$$\max(l_{k,c}, l_{l,c}) \leq l_{k \cup l,c} \leq l_{k,c} + l_{l,c}$$

$$\max(l_{k,c+1}, l_{l,c+1}) \leq l_{k \cup l,c+1} \leq l_{k,c+1} + l_{l,c+1}$$

Plaçons maintenant ces deux formes dans une situation engendrant une surface du rectangle circonscrit que l'on soit sûr de pouvoir améliorer, c'est-à-dire diminuer.

i) $P_i = 0$ et absence de débordement

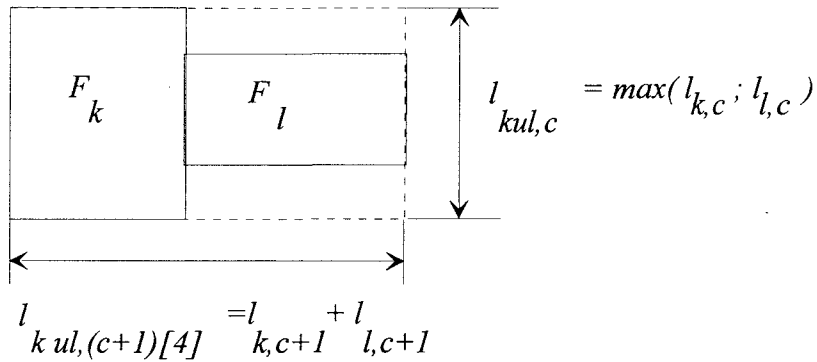


Figure 4.19

La surface correspondante à cette situation constitue alors un premier majorant $SR_{k \cup l}$ tel que

La surface correspondante à cette situation constitue alors un premier majorant $SR_{k \cup l}$ tel que

$$SR_{k \cup l}^1 = (l_{k,c+1} + l_{l,c+1}) \times \max(l_{k,c}, l_{l,c})$$

ii) profondeur d'imbrication maximale et débordement maximal.

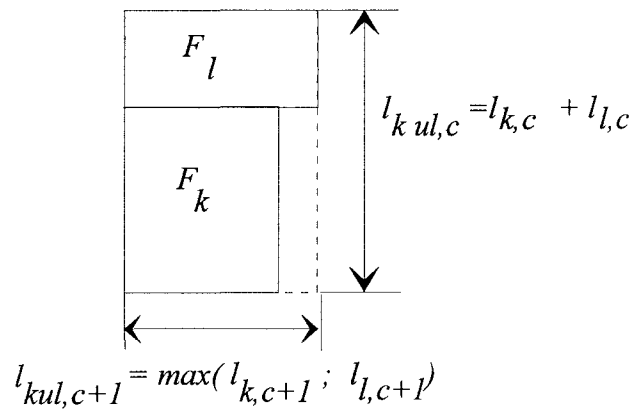


Figure 4.20

La surface du rectangle circonscrit de cette association est

$$SR_{k \cup l}^2 = (l_{k,c} + l_{l,c}) \times \max(l_{k,c+1}, l_{l,c+1})$$

En prenant le minimum des deux surfaces précédentes, on en déduit un majorant de $SR_{k \cup l}$, qu'il est possible d'obtenir par placement trivial sans aucune recherche d'imbrication.

$$SR_{k \cup l}^{maj} = \min(SR_{k \cup l}^1, SR_{k \cup l}^2)$$

Dans l'hypothèse où l'on puisse trouver un décalage d non trivial pour lequel la largeur du rectangle de la réunion soit minimale ($\max(l_{k,c+1}, l_{l,c+1})$) c'est-à-dire pour une profondeur d'imbrication maximale, alors la longueur doit être telle que la surface du rectangle $SR_{k \cup l}$ reste inférieure au majorant précédent. Si l'hypothèse n'est pas vérifiée, alors la longueur sera de toute façon plus faible que celle proposée. Ce qui s'écrit:

$$SR_{k \cup l} \leq SR_{k \cup l}^{maj}, \quad Pi = \min(l_{k,c+1}, l_{l,c+1})$$

En remplaçant $SR_{k \cup l}$ par le produit *longueur*largeur* on a:

$$l_{k \cup l, c}(d) \times \max(l_{k,c+1}, l_{l,c+1}) \leq SR_{k \cup l}^{maj}$$

Ceci nous permet de générer une contrainte sur $l_{k \cup l, c}$ telle que:

$$l_{k \cup l, c}(d) \leq \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})}$$

Selon le signe de d , on aura:

$$\text{si } d < 0, \text{ alors } l_{k,c} - d \leq \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} \Leftrightarrow d \geq -\frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} + l_{k,c}$$

$$\text{si } d \geq 0, \text{ alors } l_{l,c} + d \leq \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} \Leftrightarrow d \leq \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} - l_{l,c}$$

finalemant

$$d \in \left[-\frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} + l_{k,c}, \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} - l_{l,c} \right]$$

On a exprimé ici l'intervalle de recherche le plus large qu'il sera permis à d de balayer

5.3.2 Mise en correspondance des creux et des bosses

L'idée a déjà été proposée dans [DEL 91]. Elle consiste à penser que l'imbrication optimale de deux formes est souvent obtenue lorsque l'on a une mise en relation mutuelle des bosses et des creux de chaque forme. Cette réduction du domaine de recherche tire donc profit de cette idée en utilisant les définitions proposées concernant les segments bosses et segments creux:

Il existe un segment bosse si les extrémités de ce segment sont des points de convexité.

Il existe un segment creux si les extrémités de ce segment sont des points de concavité.

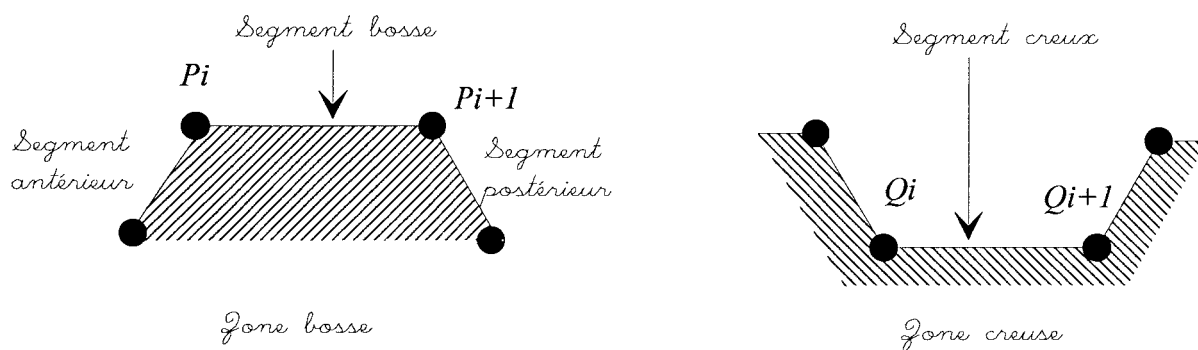


Figure 4.21

On définit également la zone creuse et la zone bosse par la portion polygonale composée du segment creux (ou bosse) et des segments immédiatement antérieurs et postérieurs.

A partir de la connaissance de ces segments-creux et bosses de chacune des formes, il s'agit alors de déterminer les valeurs de "d" permettant toutes les mises en correspondance d'un creux d'une forme avec la bosse de l'autre forme et vice et versa.

D'autre part, un contact entre deux polygones met en commun au moins un point de convexité du premier et un segment de droite du second [AVN 88].

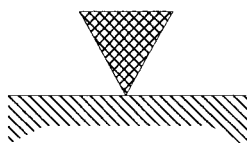


Figure 4.22

Par conséquent, pour éviter tout risque d'oubli de position relative potentiellement optimale, il sera nécessaire de mettre chaque extrémité convexe d'un segment bosse en relation avec tous les

segments constituant la zone creuse, et ceci en effectuant un glissement relatif des deux formes. Cependant, une réduction du nombre d'essais peut être réalisée en prenant note des considérations suivantes :

1) -En première approximation, on se contentera de ne mettre en correspondance que les segments bosses de hauteur maximale (ceux dont au moins une des extrémités est tangente au côté du rectangle circonscrit) avec toutes les zones creuses de l'autre forme. La recherche s'effectue successivement pour les deux formes.

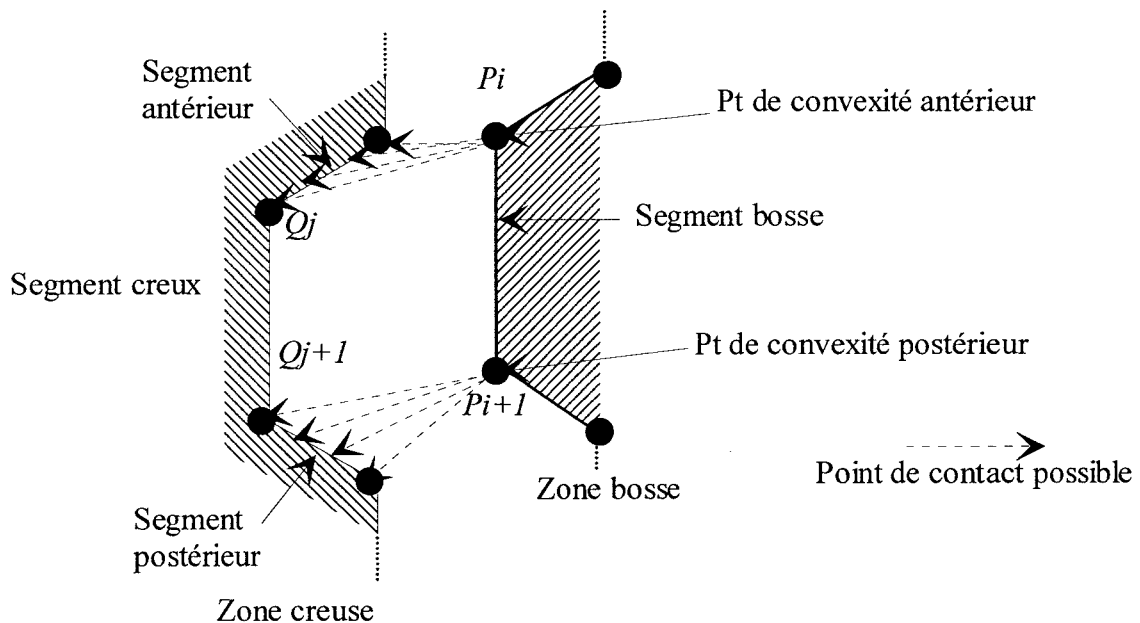


Figure 4.23

2) D'autre part, sur l'intervalle borné par la projection des extrémités de la zone creuse, la circulation d'un point de contact sur cette zone décrit donc une trajectoire convexe. La profondeur d'imbrication optimale sur cette zone est donc l'extremum local. En supposant que le point de contact entre les deux formes soit situé dans cette zone creuse, l'indice d'imbrication présente donc aussi un extremum local.

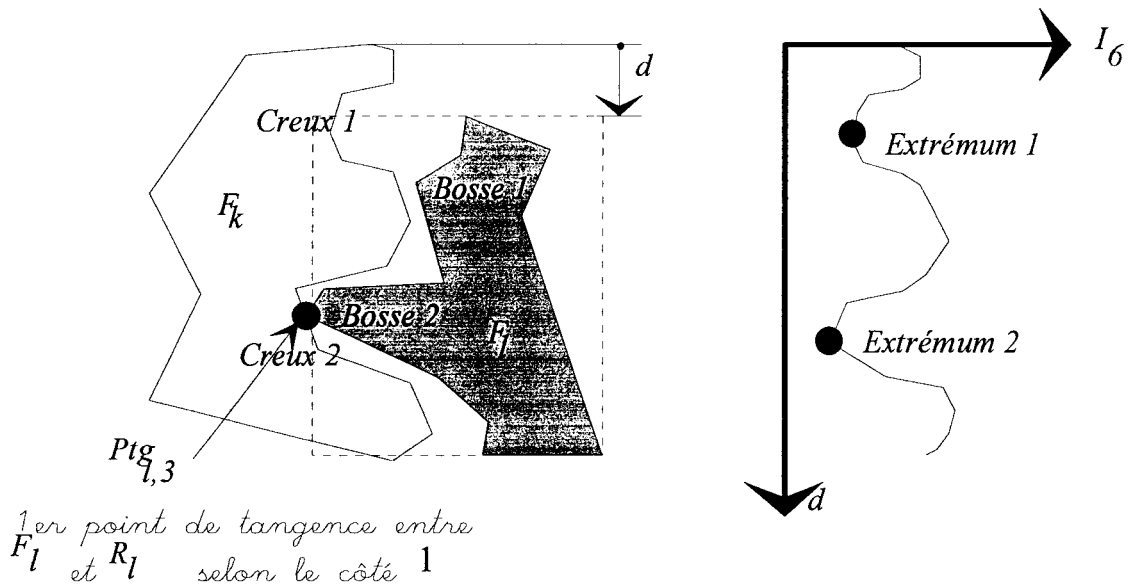


Figure 4.24 : Circulation de la bosse prédominante de F_1 le long du côté 1 de F_k

Par conséquent, lors de la recherche du glissement, il conviendra d'arrêter la recherche concernant cette zone creuse, dès la détection du minimum local de l'indice de qualité d'imbrication I_6 . Dans cette approximation, on s'affranchit évidemment des possibilités de contact en dehors de la zone creuse courante, ce qui modifierait la convexité locale de la fonction I_6 au voisinage de la zone creuse.

3) Enfin, on alterera volontairement la précision de la recherche, en échantillonnant celle-ci avec une période de glissement de Tr pixel. Le choix se portera raisonnablement vers une fraction entière de $1/9$ de la longueur de la zone creuse. Cette valeur a été choisie car elle engendre en moyenne seulement 5 calculs d'indice par zone creuse avant de trouver l'extremum, ce qui est acceptable d'un point de vue complexité.

N'oublions pas non plus la contrainte de laize qui réduit encore l'intervalle de recherche, en particulier lorsque la largeur de l'association de deux formes avoisinent la largeur de la matière.

La recherche de l'optimum local s'effectuera donc par l'algorithme présenté dans l'Annexe G. Cet algorithme énonce la procédure de mise en correspondance de la bosse principale dans $S_{l,c}$ avec les zones creuses de $S_{k,c+2}$. Cette procédure est ensuite renouvelée en permutant les rôles de $S_{k,c}$ et $S_{l,c+2}$. La valeur minimale de l'indice I_6 est alors choisie en comparant les deux valeurs I_{min1} et I_{min2} obtenue lors de l'exécution des deux itérations algorithmiques successives.

5.4 Recherche sous contrainte de motif

Les considérations de réduction de domaine de recherche sont également induites par la présence d'un motif sur le tissu. A partir des notations relatives à la contrainte de positionnement relatif adoptées dans le chapitre II, on peut établir de nouvelles relations réduisant le nombre de positions relatives autorisées d'une forme par rapport à celle qui lui sera connectée lors de l'assemblage.

Considérons l'exemple de la figure 4.25. Supposons que les formes F_0 et F_4 viennent en contact dans le vêtement et par conséquent mettent en relation leur point de raccord respectif. Nous devons ainsi connaître la position relative de la forme F_4 par rapport au placement intermédiaire P_u , obtenu par concaténation successive des formes F_0, F_1, F_2, F_3 .

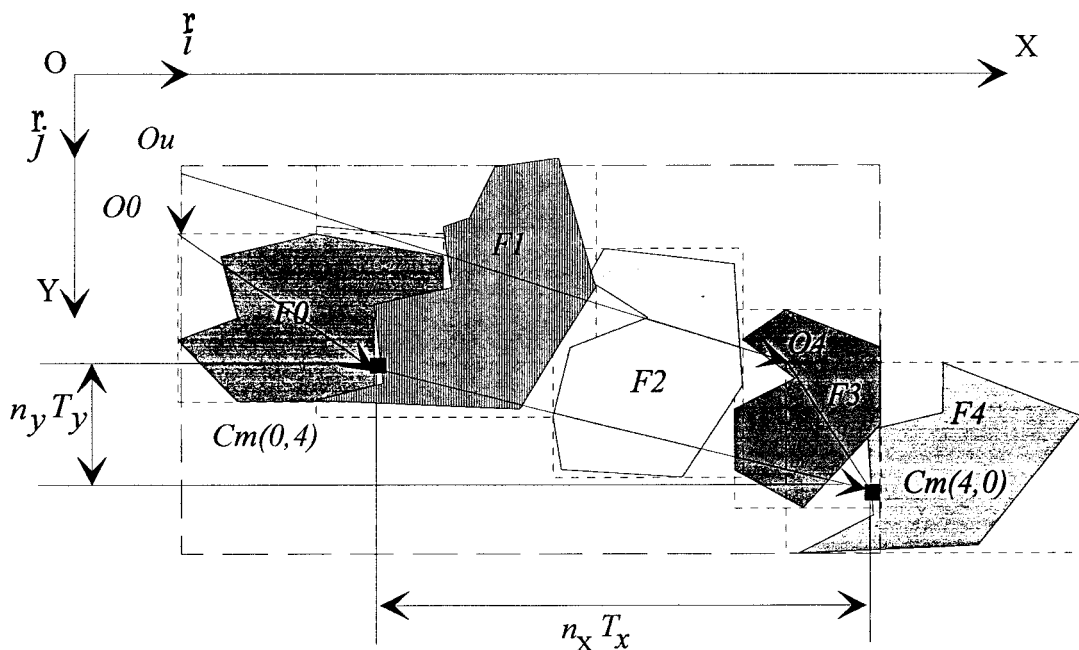


Figure 4.25

Nous pouvons alors écrire:

$$\vec{O_u O_4} + \vec{O_4 cm(4,0)} = \vec{O_u O_0} + \vec{O_0 cm(0,4)} + \begin{pmatrix} n_x T_x \\ n_y T_y \end{pmatrix} \times (\vec{i}, \vec{j})$$

soit:

$$\vec{O_u O_4} = \vec{O_u O_0} + \vec{O_0 cm(0,4)} + \begin{pmatrix} n_x T_x \\ n_y T_y \end{pmatrix} \times (\vec{i}, \vec{j}) - \vec{O_4 cm(4,0)}$$

Plus généralement, si F_i et F_n sont deux formes en raccord, nous avons

$$\begin{aligned}\vec{O_u O_n} &= \vec{O_u O_i} + O_i \vec{cm}(i, n) + \begin{pmatrix} n_x T_x \\ n_y T_y \end{pmatrix} \times (\vec{i}, \vec{j}) - O_n \vec{cm}(n, i) \\ &= \begin{pmatrix} f_x(i, n, n_x) \\ f_y(i, n, n_y) \end{pmatrix} \\ n_x, n_y &\in N^2\end{aligned}$$

\vec{i}, \vec{j} sont les deux vecteurs unitaires de la base orthogonale liée au repère (O, X, Y)

Les fonctions f_x and f_y sont discrètes de pas de discrétisation T_x (respect. T_y), croissante en n_x , (respectivement n_y). Le vecteur $\vec{O_u O_i}$ est remis à jour à chaque modification du placement et mémorisé. Les vecteurs $O_i \vec{cm}(i, n)$ et $O_n \vec{cm}(n, i)$ font partis de la base de données associée à chaque forme.

Cette relation nous amène à traduire son équivalence dans le repère local lié au peigne réduit du côté mis en relation. Nous précisons donc l'expression du couple (d, Pi) en fonction des relations précédentes et du côté c (cf. figure 4.26).

- si c est pair,

$$\begin{pmatrix} d(n_x) \\ Pi(n_y) \end{pmatrix} = O_u \vec{O_n} - O_u \vec{O_{u,c+}} + O_n \vec{O_{n,c+2}}$$

alors,

$$\begin{pmatrix} d(n_x) \\ Pi(n_y) \end{pmatrix} = \begin{pmatrix} f_x(i, n, n_x) \\ f_y(i, n, n_y) \end{pmatrix} - O_u \vec{O_{u,c+}} + O_n \vec{O_{n,c+2}}$$

-si c est impair,

$$\begin{aligned}\begin{pmatrix} Pi(n_x) \\ d(n_y) \end{pmatrix} &= O_u \vec{O_n} - O_u \vec{O_{u,c+}} + O_n \vec{O_{n,c+2}} \\ &= \begin{pmatrix} f_x(i, n, n_x) \\ f_y(i, n, n_y) \end{pmatrix} - O_u \vec{O_{u,c+}} + O_n \vec{O_{n,c+2}}\end{aligned}$$

Notons que les vecteurs $\vec{O_u O_{u,c}}$, $\vec{O_n O_{n,c+2}}$ sont connus pour chaque forme simple ou composée, grâce aux matrices de rectangle associées \overline{OMu} et \overline{OMn}

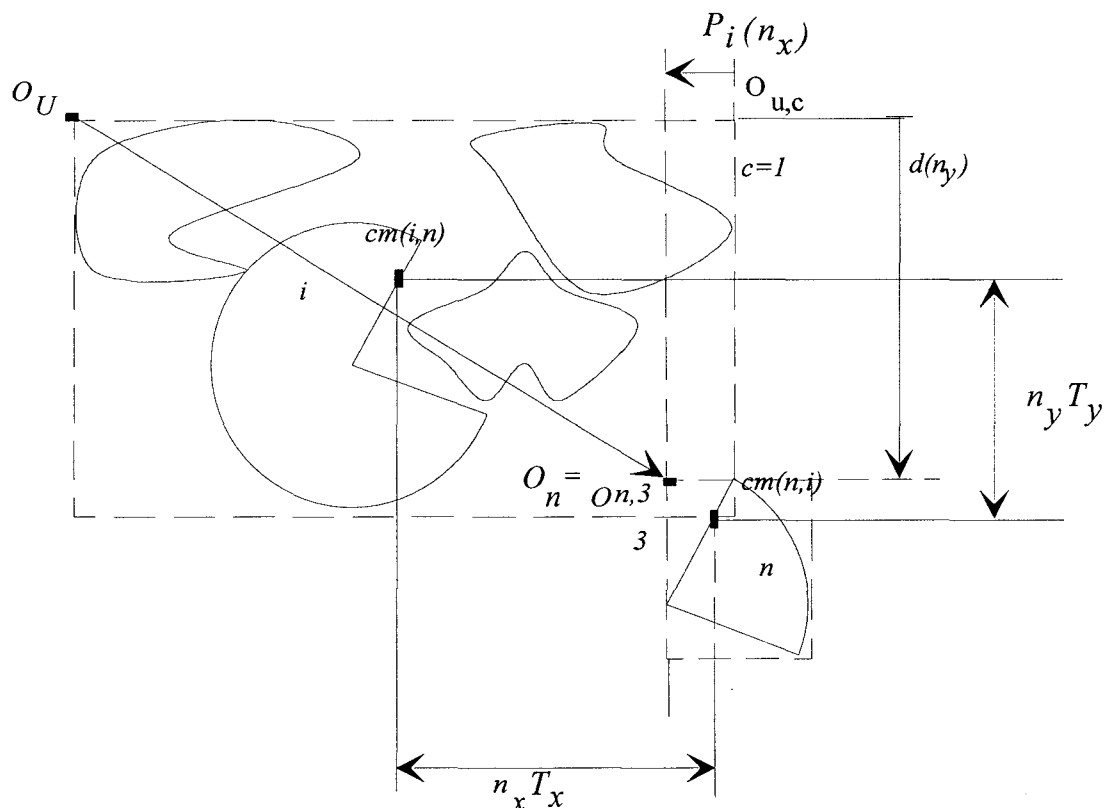


Figure 4.26

Nous voyons que d devient dépendant de n_x (ou n_y si c est impair). L'indice de complémentarité devient alors fonction de $d(n_x)$ ou $d(n_y)$. La valeur de référence pour P_i sera celle qui minimise la distance $I_6(P_u, F_n, c, d(n_x/y), P_i)$ quand la forme F_n est en contact avec la forme composée P_u . Ensuite, nous augmentons n_y (n_x si c est impair) jusqu'à ce que $P_i(n_y)$ (ou $P_i(n_x)$) soit proche de $P_{i_{max}}$ par valeur inférieure (pour assurer la contrainte de disjonction). La connaissance du couple final (d, P_i) permet ensuite d'obtenir le vecteur de positionnement relatif après avoir effectué un changement de repère.

La complexité de la procédure d'imbrication locale est sensiblement réduite, car l'intervalle de recherche a été discrétisé par les seules valeurs autorisées de d satisfaisant la contrainte. Pour une lecture détaillée de la procédure d'optimisation de l'imbrication dans le cas des tissus à motif, nous nous reporterons à l'Annexe H.

VI Complément à la classe *Forme_Modèle*

Nous présentons ici les variables d'instances ajoutées à la classe *Forme_Modèle*. Celles-ci sont relatives aux traitements que nous avons présentés dans ce chapitre.

Pour une forme F_k , ce sont:

- La matrice de rectangle \overline{OMk}
- Le côté de mise en contact avec une autre forme: c
- Le vecteur de coordonnées relatives entre F_k et la forme qui lui est associée dans la concaténation: $vect_Pos_Rel$.

VII Influence de l'approximation polygonale

7.1 Objectif

Nous analysons ici l'influence du réglage de la précision de l'approximation polygonale sur la qualité des résultats obtenus dans l'utilisation des fonctions d'imbrication et de concaténation ainsi que les temps de réponse associés. En effet, pour une forme donnée, le nombre de sommets présents dans les peignes réduits induit une répercussion, d'une part sur la représentation du contour de la forme et donc sur la qualité de l'imbrication, d'autre part sur le nombre d'opérations engendrées au niveau des procédures d'imbrication et de concaténation. Il convient donc dans cette partie de constater à travers quelques essais numériques l'influence du paramètre ε^4 sur la surface du rectangle exinscrit issu de la concaténation de plusieurs couples de formes.

7.2 Description de l'essai

7.2.1 choix des formes

Pour cet essai nous avons retenu trois couples différents de formes A, B et C, composés d'une forme principale ou maîtresse (A_m , B_m et C_m) et d'une forme secondaire ou esclave (A_e , B_e et C_e). Chaque couple permet, grâce à une large concavité de la forme principale, l'imbrication d'une

⁴ On rappelle que ε est le coefficient de surface linéique permettant de calculer le seuil surfacique maximal d'approximation polygonale. Il exprime le nombre de pixels qu'il est possible de retrancher à la surface du peigne pour réaliser l'approximation par unité de longueur du peigne

forme dans l'autre. Pour les deux premiers couples, la forme esclave est un rectangle. Pour ce dernier, le nombre de sommets par côté vaut donc deux. Ceci permet d'évaluer les caractéristiques du résultat relativement à la seule forme maîtresse, indépendamment des positions occupées par le rectangle. La forme A_m a été dessinée de sorte que le côté en imbrication soit très bruité et très "accidenté". La forme B_m présente une concavité en arc de cercle. Pour le troisième couple, nous avons voulu tester les variations engendrées sur deux formes possédant une bonne complémentarité. Nous présentons en *annexe J* les trois couples, ainsi que la représentation sous forme graphique des peignes réduits pour 3 valeurs de $\varepsilon = (0.5, 1, 3)$.

7.2.2 algorithme d'essai

Dans cet essai, nous affectons pour chaque forme une orientation fixe et nous imposons l'imbrication selon le côté N° 2 de la forme maîtresse. Pour chaque valeur de ε , nous effectuons alors une procédure de recherche d'imbrication suivie d'une concaténation. On extrait ensuite la surface corrigée optimale du rectangle exinscrit à la concaténation, c'est à dire la fonction de coût I_G minimale. Pour mesurer la complexité, on mesure le temps associé à 200 séquences identiques à la précédente afin d'obtenir une précision significative.

7.3 Résultats

7.3.1 Présentation

Dans le tableau suivant, nous rapportons les caractéristiques des peignes mis en jeu dans l'imbrication ainsi que les mesures effectuées pour chaque couple de formes pour six valeurs du paramètre de précision ($\varepsilon = 0, 0.2, 0.5, 1, 2, 3$ pixels), c'est à dire:

- la surface corrigée du rectangle exinscrit,
- le rendement de l'imbrication,
- le rapport entre le temps mis par l'algorithme pour l'essai et le temps mis par l'algorithme lorsque ε vaut 0,
- le nombre de sommets dans les peignes réduits mis en jeu.

ε	0	0.2	0.5	1	2	3
Couple A:						
$I_6 =$	40 604	40 604	40 604	40 604	40 604	40 604
rendement	51.57%	51.57%	51.57%	51.57%	51.57%	51.57%
$t(\varepsilon)/t(\varepsilon=0)$	100%	17%	9.1%	6%	6%	4%
nb de sommets $A_m \& A_e$	240&2	103&2	68&2	43&2	43&2	27&2
Couple B:						
$I_6 =$	50 218	50 218	50 218	50 218	50 598	53 258
rendement	66.13	66.13	66.13	66.13	65.73	63.09
$t(\varepsilon)/t(\varepsilon=0)$	100%	3.7%	2.7%	2.6%	2.4%	2.3%
nb de sommets $B_m \& B_e$	322&2	26&2	17&2	15&2	14&2	12&2
ε	0	0.2	0.5	1	2	3
Couple C:						
$I_6 =$	38 330	38 330	38 330	61 634	61 634	61 634
rendement	92%	92%	92%	75%	75%	75
$T(\varepsilon)/T(\varepsilon=0)$	100%	13%	5%	4%	4%	4%
nb de sommets $C_m \& C_e$	282&9	35&9	7&9	6&9	6&9	6&9

Figure 4.27

7.3.2 Analyse

Les résultats font apparaître la nécessité de trouver un compromis entre la finesse de représentation et la complexité de la recherche d'imbrication. Les deux objectifs étant contradictoires, le moyen terme est souvent dépendant de la nature du contour, et une valeur de ε peut convenir pour un type de forme et moins pour un autre.

- Dans le cas du couple A, le nombre de sommets diminue longtemps avant que la qualité de l'imbrication soit dégradée. Au-delà de $\varepsilon = 10$ (non représenté ici), les points formant des petites concavités sont fusionnées, et le rectangle secondaire ne pénètre plus suffisamment, augmentant sensiblement le coût.

VIII Conclusion

Les fonctionnalités qui viennent d'être développées dans ce chapitre vont permettre de mener à bien la construction d'un amalgame de formes selon un principe de concaténation itérative. Elles sont bien sûr particulières à la méthode de représentation des formes par les peignes réduits. De plus, la spécificité de leur définition semble réduire leur utilisation exclusivement au problème de placement paraxial.

Mais il est raisonnable de croire que la philosophie générale de ce concept pourrait être conservée en prolongeant la définition des peignes-réduits au cas du cercle circonscrit. On a vu dans le chapitre III les avantages que procureraient cette extension. La construction d'un placement non-paraxial serait alors envisageable avec son cortège d'applications (cuir, peau, matériaux synthétiques...). Dans tous les cas, pour le présent problème, les peignes de contours se révèlent être une alternative possible, même s'il est vrai que les méthodes de description de formes proposées dans la littérature répondent dans certains cas au problème posé.

Le crédit apporté par les peignes-réduits se situe surtout au niveau de la faible complexité des calculs engendrés lors de la phase de recherche d'imbrication locale, et ce, moyennant quelques considérations sur les bosses et les creux et quelques artifices d'optimisation. Ce résultat positif est tout à fait essentiel lorsque l'on prend conscience que ces opérateurs constituent le noyau de calcul dans la procédure d'optimisation combinatoire.

L'efficacité des fonctions de d'imbrications doit être complétée par un réglage judicieux de l'approximation polygonale. On a constaté que ce réglage se justifie au niveau de qualité du placement local mais aussi à cause de l'impact inévitable sur le placement global. Cependant, moyennant quelques essais réalisés à partir de couples de formes génériques, une valeur particulière du paramètre ε pouvait garantir un ajustement fiable du compromis coût/complexité.

Pourtant une question subsiste. Pouvons-nous faire mieux au niveau de la complexité de la recherche d'imbrication locale ? La réponse se trouve peut-être dans d'autres points de vue. On peut toujours imaginer d'autres principes de recherche d'imbrication, en associant aux formes des propriétés électriques ou magnétiques par exemple. On pourrait considérer les sous-contours de deux formes à imbriquer comme des lignes de potentiels opposés qu'il s'agirait d'approcher au mieux afin d'engendrer un champ électrique maximal. Autant de pistes qui mériteraient peut-être quelques investigations.

CHAPITRE V

METHODES D'OPTIMISATION COMBINATOIRE POUR LA RECHERCHE DU PLACEMENT

Mon intérêt réside dans le futur car
je me prépare à y passer le reste de
ma vie.

C.F. Kettering

I INTRODUCTION

Nous allons maintenant consacrer ce chapitre à la présentation de la résolution du problème de placement. Nous entendons ici par résolution, la recherche d'une ou plusieurs solutions de coût minimum. Il s'agit donc d'optimisation.

Dans le chapitre 2, nous avons décrit le problème du placement en précisant les contraintes qui le régissent et l'objectif que l'on se fixe. Les contraintes définissent les degrés de liberté du système (les formes ou les groupements de formes), et la fonction de coût donne une mesure pour évaluer, comparer, valider un état intermédiaire du placement ou une solution. Mais pour que le problème soit bien défini, il faut décrire l'espace de configuration, c'est-à-dire les différentes situations qu'un placement peut atteindre. Nous avons choisi l'espace d'états, au sens où on l'entend en Intelligence Artificielle, c'est à dire un graphe constitué d'un ensemble de noeuds et d'arcs symbolisant états et coûts de chemin. Pour passer d'un état à un autre, un certain nombre de transformations sont nécessaires. Les transformations utilisées en construction de placement ont été complètement redéfinies dans le chapitre IV, en raison du concept utilisé dans le chapitre III pour la description des formes. Bon nombre de contraintes étant implicitement vérifiées lors de la construction, et un découplage entre le niveau local du placement et le niveau global de décision permettent de réduire du même coup le nombre d'états engendrés. Cependant, celui-ci reste important, tout au moins trop important pour espérer envisager une méthode de recherche exhaustive. S'agissant de trouver une solution sous-optimale à un problème de nature discrète, mais avec l'assurance que celle-ci soit proche de la meilleure, on se propose donc de comparer deux méthodes différentes d'optimisation combinatoire. La première, appelée Recuit Simulé, est une méthode d'optimisation stochastique. La deuxième utilise l'algorithme "Aε" emprunté à la recherche en graphe en Intelligence Artificielle. Leur point commun, ce qui les rend

comparables, c'est leur convergence asymptotique vers l'admissibilité. Si la première peut en théorie trouver l'optimum global, les contraintes temporelles nous amènent à interrompre la recherche avant l'aboutissement à l'optimum. On estimera alors sa capacité à fournir une solution satisfaisante en temps limité. La deuxième méthode est par définition ϵ -admissible, mais peut aussi prétendre à l'admissibilité si le paramètre de réglage $\epsilon=0$. Mais là encore, la réalité industrielle nous oblige à être moins gourmand. En trouvant un terrain d'entente, une mesure de l'efficacité de recherche, on pourra alors comparer nos deux méthodes et faire un choix qui paraît a priori difficile et définir les modalités d'utilisation.

II Recuit simulé et représentation du problème: 1er essai

2.1 Introduction

Nos premières investigations sur les algorithmes d'optimisation combinatoire nous guidèrent vers la méthode du recuit simulé, dont la facilité (toute relative) d'implantation nous séduisait. L'expérience proposée par [PRE 83] nous a permis de constater que cette méthode convenait bien à ce genre de problème très combinatoire, à partir du moment où le temps ne représentait pas une contrainte de l'application.

Mais selon sa méthode, il fallait à l'issue d'une session de recherche, utiliser une heuristique d'ajustement afin de rétablir la contrainte de disjonction qui n'était pas satisfaite de façon rigoureuse. Des chevauchements résiduels persistaient du fait que la solution n'était qu'un minimum local, et donc, que le terme représentant les chevauchements n'était pas minimisé de façon absolue.

Les opérateurs de concaténation que l'on a pu construire avec le concept de peigne de contour, nous invita alors à voir le recuit simulé sous un autre jour. On pouvait ici, pour chaque nouvelle configuration proposée, satisfaire toutes les contraintes de placement avant d'évaluer la fonction de coût, en l'occurrence les pertes. Avec ces outils, on allait pouvoir utiliser le recuit simulé et modifier le réglage des paramètres afin d'accélérer son cycle de recherche. Même si le résultat final ne représentait qu'un minimum local, l'important était ici de réduire le nombre d'itérations. L'avantage que l'on avait par rapport à l'algorithme de [PRE 83] était que, quel que soit le résultat final, toutes les contraintes étaient satisfaites.

Nous en donnons ici le principe initial:

2.2) Représentation

Toutes les formes sont introduites dans une liste appelée "STOCK" dans l'ordre de leur concaténation.

Chaque forme $STOCK(k)$, $1 \leq k \leq N$, est caractérisée par son vecteur de configuration: $\begin{pmatrix} \lambda \\ S_x \\ c \end{pmatrix}$ où (λ, S_x, c) est un triplet prenant ses valeurs sur le domaine des paramètres $\mathcal{P}(k) = orientations_Admises(k) \times \{0\} \cup S_{Crx}(k) \times \{0,1,2,3\}$

L'ordre initial de la liste STOCK ainsi que l'affectation de chaque vecteur d'état est arbitraire. Les formes sont concaténées de façon optimale transversalement tant que la largeur est inférieure à celle de la matière. Ensuite, elles sont concaténées longitudinalement.

$$STOCK = \left\{ \begin{array}{c} stock(1) \\ \left[\begin{array}{c} \lambda = 0 \\ S_x = 0 \\ c = 2 \end{array} \right] \end{array} , \begin{array}{c} stock(2) \\ \left[\begin{array}{c} \lambda = 0 \\ S_x = 0 \\ c = 2 \end{array} \right] \end{array} , \dots \dots \dots \begin{array}{c} stock(N) \\ \left[\begin{array}{c} \lambda = 0 \\ S_x = 0 \\ c = 1 \end{array} \right] \end{array} \right\}$$

2.3 Générateur de configuration aléatoire

L'idée est ici de construire un générateur de configurations aléatoires où les contraintes sont vérifiées au niveau intrinsèque des opérateurs.

Les changements de configurations s'effectuent selon la séquence:

Etape 1: Sélection aléatoire d'une forme $Stock(k)$

Etape 2: Choix aléatoire d'une perturbation élémentaire sur la forme $Stock(k)$ parmi les propositions suivantes:

a) un changement de l'orientation en sélectionnant une valeur de λ dans l'ensemble $orientations_Admises(k)$. On effectue: $Stock(k)$ **rotation:** λ

b) une symétrie par rapport à l'axe longitudinal si cela est autorisé (si $S_{Crx} = 1$). On effectue: $stock(k)$ **XSymétrie.**

c) changement de coté de mise en concaténation si cela est possible compte-tenu des contraintes de largeur. En réalité, on prend comme nouvelle valeur de c , la valeur suivante de la valeur courante dans la liste $(0,1,2,3)$:

$$c := c + 1 \text{ (modulo 4)}$$

d) une permutation de la forme $Stock(k)$ avec $Stock(k-1)$

S'il s'agit de la première forme ($k=1$), celle-ci est permutée avec la dernière.

Etape 3: Construction

Le placement est remis à jour seulement à partir de la forme $Stock(k-1)$ si la perturbation était une permutation, ou à partir de $Stock(k)$ sinon. On effectue donc:

si *permutation* alors $i := k-1$ sinon $i := k$

Pour $j = i$ jusqu'à N faire

$vect_Pos_Rel := P(j-1)$ **recherche_Imbrication:** $Stock(j)$ **du_Côté:** c

$P(j) := P(j-1)$ **concatène:** $Stock(j)$ **pos_Rel:** $vect_Pos_Rel$

2.4 Optimisation

Pour chaque configuration, on évalue la fonction à optimiser, c'est-à-dire ici la longueur du placement total.

$$E = \text{Longueur}(P(N))$$

L'acceptation de chaque nouvelle configuration $P(N)$ est soumise au test de Metropolis mis en oeuvre dans l'algorithme du recuit simulé. Une présentation détaillée sera donnée plus loin. Les paramètres caractérisant le modèle du recuit simulé ont été réglés expérimentalement afin de réduire le temps au mieux. Les résultats sont intéressants pour un nombre limité de formes.

2.5 Vers une autre représentation

Mais l'avantage apporté par le générateur de configurations satisfaisant les contraintes ne suffit pas ! Nous nous sommes aperçus rapidement, que l'obtention d'une bonne solution induit un temps d'attente très long lorsque l'instance du problème (c'est à dire le nombre de formes) augmente. Que faire pour que le nombre de formes reste raisonnable tout au long du travail? Le travail en bande est une solution au problème: on divise l'espace des configurations en réduisant l'espace de placement. Mais ceci oblige de formaliser le problème de placement d'une autre façon: celle décrite dans le chapitre II: "Placement en bande". A partir de là, comment représenter l'espace de configuration ? Il faut trouver une représentation qui permette de développer toutes les configurations potentielles de la bande en mettant en jeu toute la variété des formes du carnet de commande.

L'opération de concaténation que l'on s'est attaché à construire, induit une relation de préséance entre les formes à associer. On concatène selon un côté précis de la forme réceptrice déjà placée. Permuter deux formes nécessite une séparation et une nouvelle concaténation selon le nouvel ordre. On voit ici qu'une représentation possible est le graphe d'état. Avec ses arcs et noeuds, il restitue aussi cette notion d'ordre. L'absence de circuit (car une forme placée ne peut être enlevée) et de cycle (il n'est pas possible d'obtenir le même placement avec des formes différentes ou des opérations différentes) lui confèrent une structure arborescente. Nous allons donc représenter notre problème sous cette forme.

Finalement une bande n'est alors qu'une réduction du graphe entier, en limitant les développements de l'arborescence à un certain rang. Ce rang, c'est-à-dire celui des noeuds terminaux, dépend de la variation de la fonction de coût le long du chemin parcouru.

III Représentation en graphe

Dans cette représentation, on structure l'ensemble des états du problème par un graphe non-orienté. Chaque sommet (ou noeud) du graphe est un état discret du problème; il y a un arc du noeud u au noeud v si on dispose d'un opérateur permettant de transformer l'état u en l'état v .

3.1 L'espace des configurations

Résumons les principales définitions de notre problème, représenté en graphe:

Un état est une description de la situation initiale (aucune forme n'est placée sur la matière) ou d'une situation intermédiaire quelconque rencontrée lors de sa résolution. Chacun d'eux est réalisé par des concaténations successives de plusieurs formes positionnées en imbrication locale optimale relativement au critère *de la surface corrigée du rectangle exinscrit* (I_G) défini au chapitre IV.

Etant donnée la recherche du placement de la i -ème bande P_i , on désigne par u l'état correspondant au placement intermédiaire $P_{i,u}$. $P_{i,u}$ est une instance de la classe *Forme_Model*, obtenue par concaténation successive d'un sous-ensemble de formes référencées dans l'encours de placement EP . Dans ce qui suit, l'indice i sera supprimé pour simplifier l'écriture.

A chaque noeud u , on associe le coût $g(u)$ (cf. chapitre II) représentant la chute de matière par rapport aux dimensions de la bande, ce qui donne:

$$g(u) = S_{occupée} - S_{utile} = L_u \times W_M - \sum_{F_k \in P_u} S_k \quad \text{avec}$$

L_u : longueur du placement P_u

W_M largeur matière

S_k surface de la forme F_k

De même, on désigne par

$$A_u = (a_{0,u} \ a_{1,u} \ \dots \ a_{k,u} \ \dots \ a_{N,u})$$

la liste des effectifs restant à produire au noeud u pour les N types de formes, cette information étant utile pour tenir à jour la productibilité de chaque forme.

L'ensemble des états, que l'on notera U , est appelé, selon la sensibilité scientifique de chacun, l'espace de représentation, l'espace de configuration ou l'espace de recherche.

Il s'agit ici d'un ensemble discret, et fini, puisque toute transformation d'état vers les états finaux consiste à ajouter une forme supplémentaire, et que l'ensemble des formes référencées dans l'encours EP est fini. D'autre part, les limites physiques imposées par les contraintes de placement en bande engendrent un cloisonnement du graphe sur un sous-ensemble des états accessibles, c'est-à-dire un sous-graphe.

L'espace de configuration U est structuré par un ensemble d'applications susceptibles de transformer un état en un autre état. Au niveau du placement, la transformation est réalisée au moyen des opérateurs géométriques définis dans le chapitre précédent, et que l'on rappelle ci-dessous.

Soit F_k une instance de la classe des formes *Forme_Model*, les méthodes d'instances opérant sur la classe sont :

F_k rotation: λ , λ prenant ses valeurs sur l'ensemble *orientations_Admisses(k)* $\subset \{0,1,2,3\}$. Cette méthode oriente l'axe $O_k X_k$ de la forme d'un angle $\lambda\pi/2$ par rapport à la matière.

F_k Xsymétrie ; Cette méthode réalise une symétrie de la forme F_k par rapport à l'axe CrX_k . Cette transformation est autorisée si le paramètre $S_x = 1$.

vect_Pos_Rel := F_k recherche_Imbrication: F_l du côté: c dans: rectangle_Maximal. Cette méthode renvoie le vecteur de positionnement relatif

$vect_Pos_Rel = \xrightarrow{O_k O_l}$ des deux formes, afin que celles-ci soient en imbrication optimale relativement au côté c de F_k , tout en vérifiant les contraintes de largeur et de longueur de bande spécifiées par le paramètre *rectangle_Maximal*.

Soit u et v deux noeuds du graphe U , tel que v est un successeur de u , et soit P_u et P_v les deux placements associés, alors le placement P_v sera réalisé par la concaténation du placement intermédiaire P_u et d'une nouvelle forme F_k tel que P_v est obtenu par:

$P_v := P_u \text{ concatène: } F_k \text{ pos_Rel: } vect_Pos_Rel$. Cette méthode concatène les deux formes et crée une nouvelle instance dont le contour correspond à l'enveloppe de la réunion des deux formes.

Toute transformation d'état engendrée par une application se traduira donc au niveau du placement par une combinaison de ces opérations. Pour simplifier le formalisme, on emploiera une application unique à plusieurs paramètres. On désignera par :

$$\begin{aligned} U &\rightarrow U \\ u &\rightarrow v = O_{k,\lambda,S_x,c}(u), \end{aligned} \quad \text{l'application de } U \text{ sur } U, \text{ dont les 4 paramètres}$$

(k,λ,S_x,c) sont définis sur:

$$[0,N] \times \mathcal{P}(k) = [0,N] \times \text{orientations_Admises}(k) \times \{0\} \cup S_{CrX}(k) \times \{0,1,2,3\}$$

$\mathcal{P}(k)$ est appelé *espace des paramètres* de l'application O pour la forme F_k

Cette application effectue la séquence suivante :

$O_{k,\lambda,S_x,c}(u)$

- 1) Si $a_{k,u} > 0$ alors "forme F_k productible"
 - 1.1) F_k orientation: λ
 - 1.2) Si $S_x=1$ alors F_k Xsymétrie
 - 1.3) $vect_Pos_Rel = Pu$ recherche **Imbrication F_k du côté: c dans: *rectangle_Maximal***
 - 1.4) $vect_Pos_Rel \neq nil$ alors
 - 1.5.1) $P_v := Pu$ concatène: F_k pos_Rel: $vect_Pos_Rel$
 - 1.5.2) création du noeud v associé à P_v
 - 1.5.3) calculer $g(v)$ "calcul du coût"
 - 1.5.4) $a_{k,v} := a_{k,u-1}$ "mise à jour de l'effectif de la forme F_k ".

Cette séquence exprime que si la forme F_k est disponible au noeud u ($a_{k,u} > 0$), elle subit les opérations de rotation et éventuellement de symétrie. Une recherche d'imbrication

avec le sous-placement antécédent génère le vecteur de positionnement relatif nécessaire à la concaténation.

La variation de tous les paramètres sur l'espace $[0, N] \times \mathcal{P}(k)$ engendre l'ensemble fini des opérateurs O défini par :

$$O = \left\{ o_{k, \lambda, sx, c} / (k, \lambda, sx, c) \in [0, N] \times \mathcal{P}(k) \right\}$$

Le couple $G(U, O)$ est le *graphe d'états*. Pour $(u, v) \in U^2$, on aura un *arc* (u, v) si et seulement si pour un quadruplet $(k, l, s_x, c) \in [0, N] \times \mathcal{P}(k)$, $o_{k, \lambda, sx, c}(u)$ existe. v est un *successeur* de u . On note $O(u)$ l'ensemble de tous les successeurs engendrés sur l'espace des paramètres $[0, N] \times \mathcal{P}(k)$.

3.2 Représentation implicite

Le problème à résoudre n'est pas formulé par la donnée explicite du graphe $G(U, S)$, mais plutôt par les données suivantes:

- Un état initial u_0 associé au placement vide $P_0 = \emptyset$.
- L'ensemble des opérateurs O .
- Un coût $g(u)$ associé à chaque état.
- Un sous ensemble $T \subset U$ des états terminaux.

Ces états sont définis implicitement par la relation d'équivalence:

$$t \in T \Leftrightarrow \left\{ \begin{array}{l} O(t) = \{\emptyset\} \\ ou \\ \forall v \in O(t): g(v) \geq g(t) \end{array} \right\} \text{ cette relation signifie que } t \text{ est un état}$$

terminal s'il ne possède aucun successeur ou si tous ses successeurs ne vérifient pas la décroissance de la fonction de coût g . Dans le second cas, le placement correspondant à cette situation ne satisfait pas les conditions de pertes minimales relativement à la bande.

Ce graphe, défini implicitement est nécessairement connexe et possède une seule racine u_0 . Ne possédant pas de cycle, il s'agit donc d'une arborescence K -aire. Les noeuds terminaux sont les feuilles de l'arborescence (**cf figure 1**). K représente le nombre maximal de successeurs directs que peut avoir un noeud. Ici, K est égal à la cardinalité du plus grand ensemble des paramètres:

$$K = \text{Max}_{\substack{F_{k \in EP} \\ a_k \neq 0}} (\text{card } \mathcal{P}(k)) \times \text{card}[0, N]$$

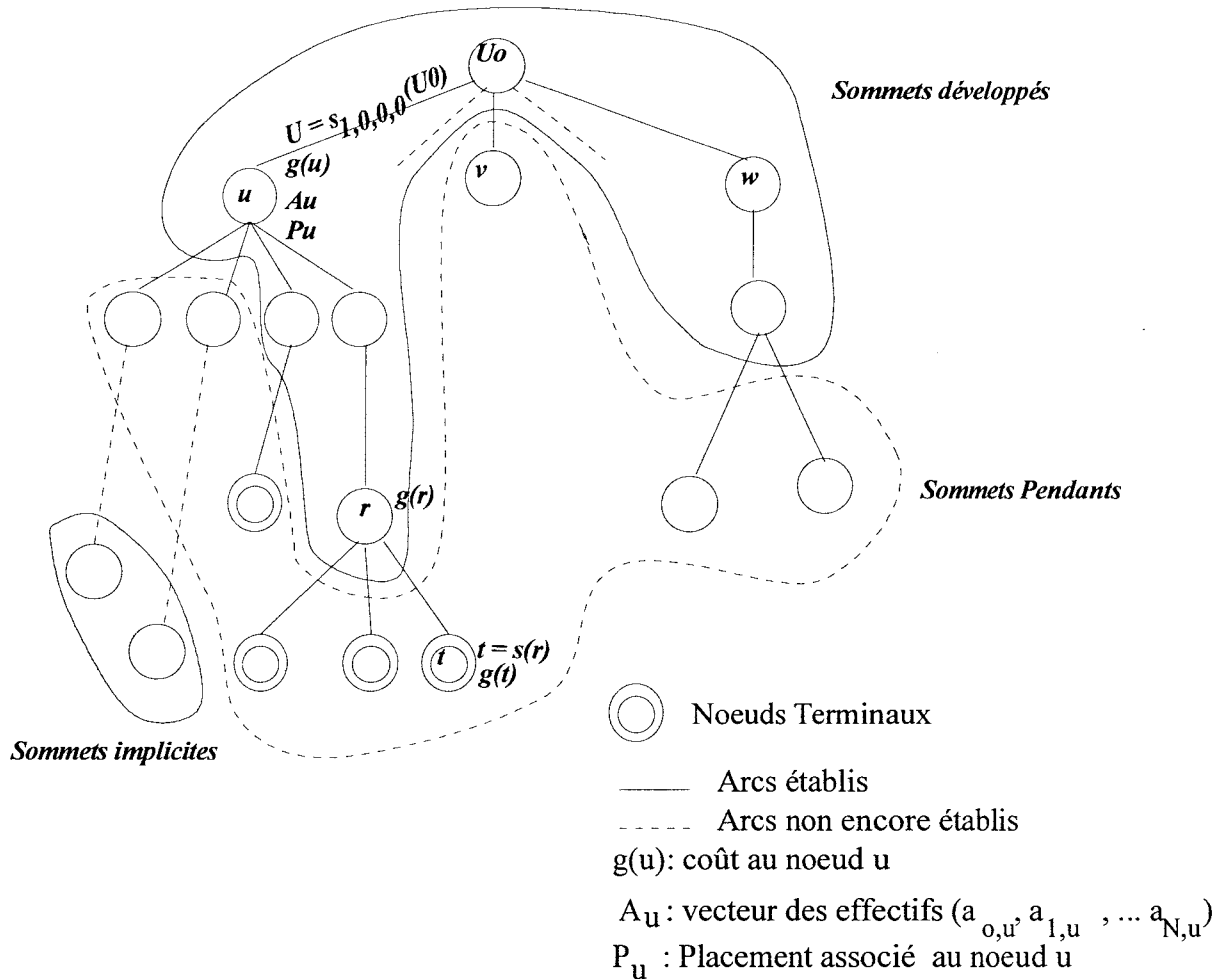


Figure 1

A un moment quelconque de l'exploration, on distingue:

- Les sommets développés ou partiellement développés dont la totalité ou une partie des successeurs a été engendrée.
- Les sommets pendants: ont été explicités (le placement a été déterminé et évalué), mais leurs successeurs ne sont pas connus.
- Les sommets implicites: Ils descendent des sommets pendants, mais ne sont pas encore connus.

L'arbre s'arrête donc aux états terminaux dont le placement couvre une bande au sens du chapitre II. En cela, ce graphe est un sous graphe du graphe des états accessibles sans la contrainte de longueur de bande. (**cf figure 2**). Cette limitation est le reflet du choix d'une stratégie "diviser pour régner", où l'on se contente de traiter des petits placements pour éviter l'explosion combinatoire.

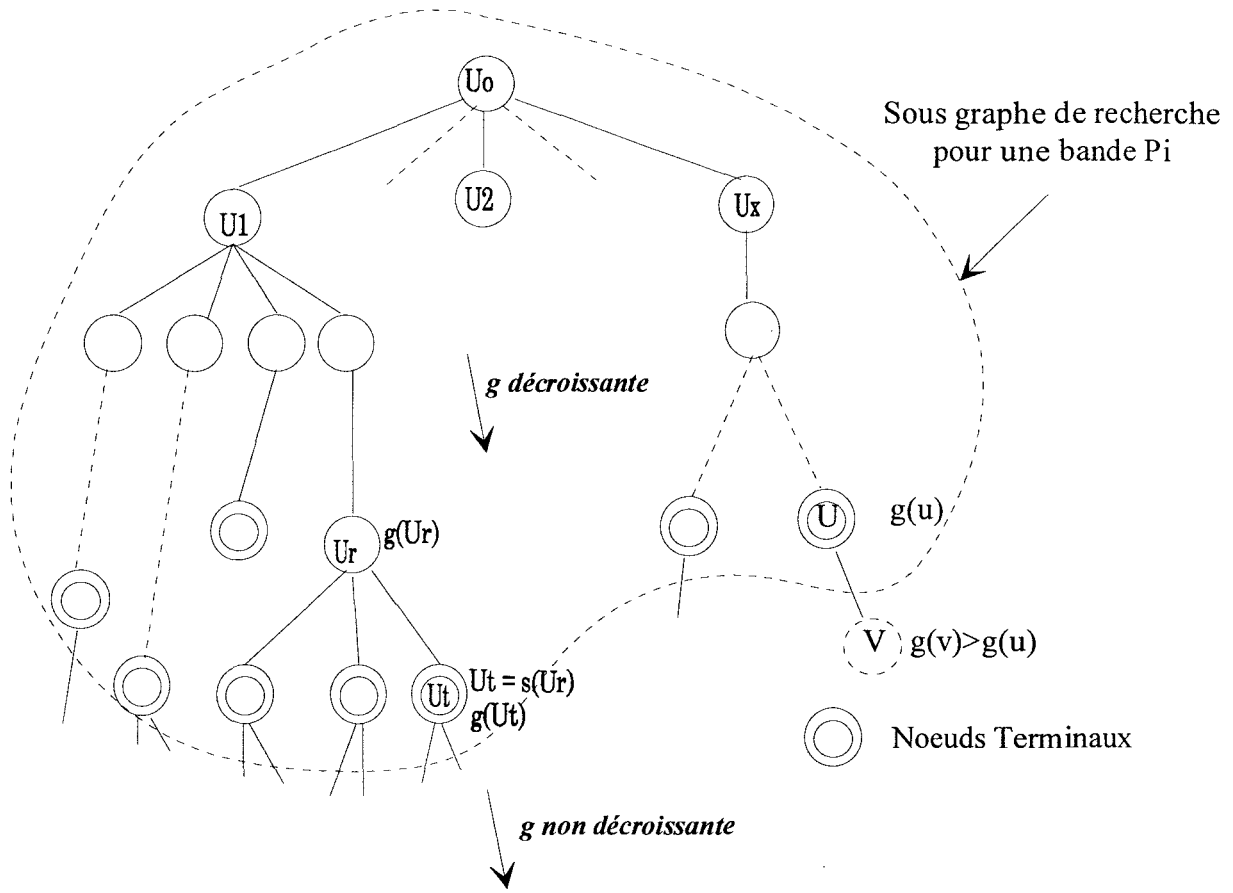


Figure 2

Remarquons que le rang d'un noeud (nombre d'arcs depuis u_0), correspond au nombre de formes présentes dans la bande relativement à ce noeud. Tous les noeuds terminaux ne sont donc pas de rang identique.

IV Variation de la fonction de coût et topologie de l'espace de configuration

A l'espace de configuration présenté précédemment, on peut lui associer une courbe matérialisant l'évolution de la fonction $g(u)$ selon les différentes combinaisons de placement, c'est-à-dire les noeuds. Chaque placement correspond à un noeud du sous-graphe pour une bande P_i (cf figure 3).

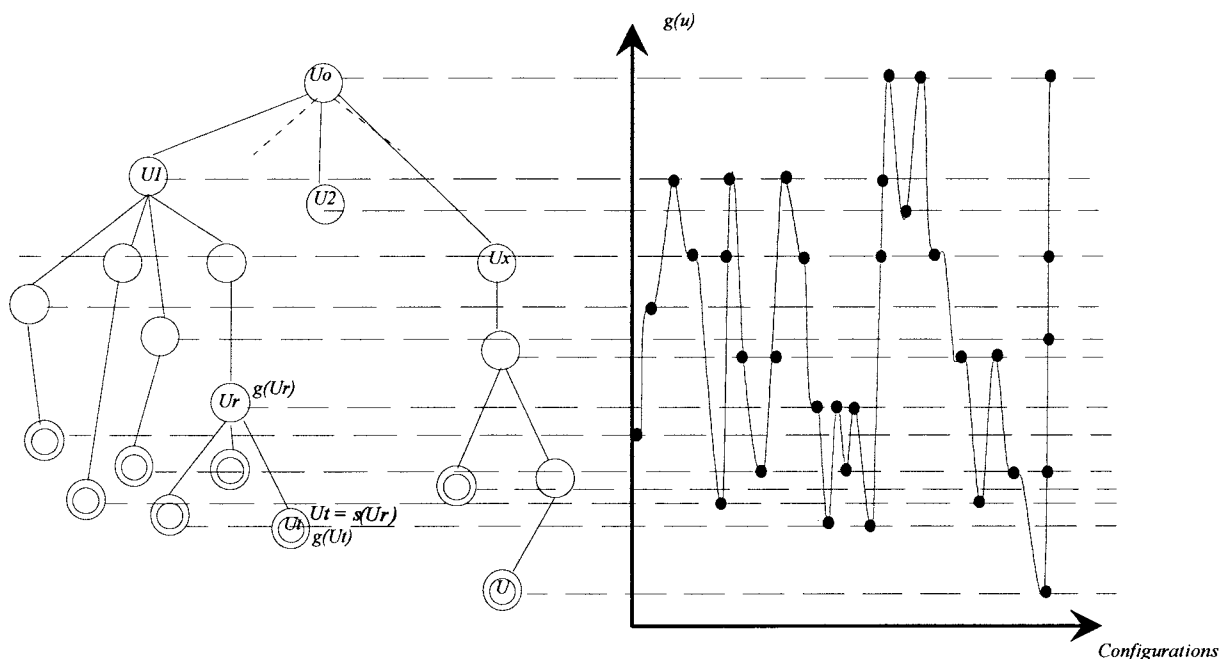


Figure 3

Evidemment, la courbe $g(u)$ n'a de sens que si l'espace des configurations est doté d'une topologie munie d'une relation de voisinage entre les configurations et d'une distance. Celle-ci découle naturellement de la représentation arborescente en considérant que deux configurations sont voisines (ou connexes) lorsqu'elles sont reliées par un seul arc. La distance entre deux configurations non connexes est définie comme le nombre minimal de changements élémentaires nécessaires pour passer d'une configuration à une autre (sur la figure 3, la distance $d(Ur, U2)$ entre deux configurations Ur et $U2$ vaut 4).

D'autre part, lors de la recherche en graphe, la remontée vers le prédécesseur d'un noeud, signifie qu'une forme vient d'être supprimée. Ceci est réalisé virtuellement puisque aucun opérateur de soustraction de forme n'a été prévu. Le graphe est donc non-orienté et chaque configuration est réversible. Cette propriété permet donc de décrire toutes les valeurs potentielles de la fonction de coût, signifiant que tous les états du placement sont donc accessibles.

Si ce paysage de coût (on parle aussi de paysage d'énergie) présentait un seul minimum global et aucun minimum local, le problème d'optimisation serait très simple, et il pourrait être résolu par la méthode d'amélioration itérative classique. Toute la difficulté du problème d'optimisation réside dans le fait que la fonction de but $g(u)$ comporte un grand nombre de vallées plus ou moins profondes et plus ou moins proches qui correspondent à des minima locaux.

Le nombre de minima locaux, donc de noeuds terminaux, dépend de l'espace des paramètres $\beta(k)$ propre à chaque forme et du nombre N de formes différentes. Il est clair que si une forme a peu de contraintes spécifiques, l'espace des paramètres $\beta(k)$ est grand et les successeurs engendrés sont nombreux. De même, le nombre N de modalités différentes dans l'encours de placement élargit sensiblement l'arborescence. Il est à noter cependant, que le choix que nous avons fait s'agissant de la stratégie "locale-globale", confère au problème une sensible simplification de complexité. Celle-ci se verrait grandie si toutes les positions relatives entre les formes étaient envisageables.

Un calcul de mauvais cas, nous permet d'évaluer le nombre de placements en bande différents, engendrés pour un ensemble de N formes distinctes. Supposons que chaque forme n'ait aucune contrainte spécifique. Il vient alors:

$$\begin{aligned} \text{card}(O) &= N \times \text{card}(\text{orientations_Admissibles}) \times \text{card}(\{0,1\}) \times \text{card}(\{0,1,2,3\}) \\ &= N \times 4 \times 2 \times 4 \\ &= N \times 32 \end{aligned}$$

Chaque noeud peut donc avoir au pire $32 \times N$ successeurs. Ce qui fait au rang 1: $32 \times (N+1)$ sommets, au rang 2: $(32 \times N)^2$ sommets,.....au rang r : $(32 \times N)^r$ sommets. Supposons que l'on dispose de $N=15$ modalités de forme de dimensions semblables ($l \times w$) et d'effectif important pour que chaque type de forme puisse remplir complètement une bande et que l'on espère un rendement moyen de 70% de la surface occupée pour la bande considérée. Les formes étant semblables, les noeuds terminaux seront tous sensiblement au même rang R , égal au nombre de pièces que l'on peut introduire dans le rectangle de matière $l \times W_M$. On aura donc $R \approx 0.7 \times W_M/w$. Choisissons $W_M/w = 10$, ce qui donne $R=7$. Le nombre total de noeuds explicités sera donc $\frac{(32 \times N)^{R+1} - (32 \times N)}{(32 \times N) - 1} = \frac{(32 \times 15)^{7+1} - (32 \times 15)}{(32 \times 15) - 1} \approx 5.9 \times 10^{18}$. Si ce calcul montre que même si l'on est ramené à un problème de complexité en $O((K \times N)^R)$ en cloisonnant le graphe (au lieu de $O(e^N)$ pour les problèmes NP-complet), le nombre de noeuds à expliciter reste trop élevé pour espérer utiliser une recherche exhaustive. En pratique, le compromis qualité/temps nous conduit à nous satisfaire d'une bonne solution avec l'assurance que la meilleure solution soit de coût relativement proche. Pour atteindre cet objectif au bout d'un temps de calcul raisonnable, il est nécessaire d'avoir recours à des méthodes heuristiques.

Ces méthodes doivent permettre, face à l'impossibilité d'explorer la totalité des configurations du placement, d'évoluer vers une bonne solution en envisageant un nombre limité de combinaisons, dont le choix est guidé, soit par le hasard s'il s'agit d'une méthode stochastique, soit par une stratégie de parcours de graphe.

V Méthode stochastique: le recuit simulé

Cette méthode, issue de la thermodynamique, s'inspire des méthodes d'amélioration itérative dont nous donnons tout d'abord le principe.

5.1 Amélioration itérative aléatoire

Cette méthode, appelée aussi méthode de la descente la plus rapide (dans la courbe de coût) est dérivée de celle du gradient. Elle consiste, à partir d'une solution initiale quelconque à tester une nouvelle configuration aléatoire à partir d'une modification simple ; si cette modification entraîne une diminution de la fonction de coût, on l'adopte et on réitère le processus sur la nouvelle configuration, sinon on garde l'ancienne et on essaie une autre modification. On s'arrête quand aucune modification ne peut apporter d'amélioration ou bien à la suite d'un grand nombre d'essais négatifs consécutifs. La configuration initiale est choisie aléatoirement ou peut être la solution d'une heuristique. Dans le cas de notre problème de placement et en reprenant la représentation choisie précédemment, la méthode revient à se diriger d'un noeud à un noeud voisin de coût plus faible. Partant d'un noeud u , ce principe nous amène à choisir systématiquement un noeud v successeur de u , puisque la fonction $g(u)$ est décroissante. Si v est un noeud terminal, la procédure s'arrête puisqu'il n'y a plus de voisin moins coûteux (cf. **figure 4**). La recherche tombe dans une vallée qui n'est pas forcément un minimum global, et dont elle ne peut sortir. La seule issue consiste à redémarrer une nouvelle procédure de noeud initial identique ou différent sans certitude d'aboutir à l'optimum global. Des théorèmes de convergence vers l'optimum global ont été établis pour cette méthode [BON 87], cependant des difficultés subsistent pour échapper aux minima locaux.

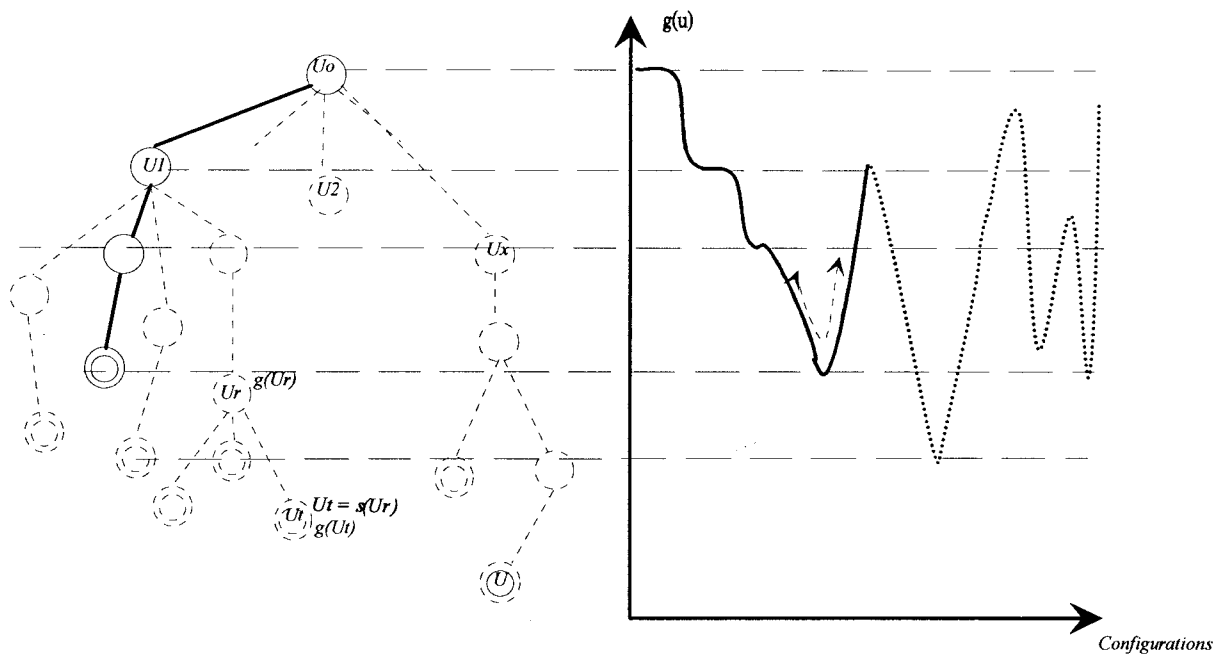


Figure 4

On constate donc que la seule façon d'atteindre un placement de plus faible coût consiste à remonter la vallée pour espérer redescendre ensuite vers un minimum plus bas ; ce que fait naturellement le recuit simulé.

5.2 Le recuit simulé

5.2.1 Analogie avec l'optimisation combinatoire

L'idée du recuit simulé a été développée simultanément et indépendamment chez IBM par S. Kirkpatrick, C.D. Gellat Jr et M.P. Vecchi [KIR 82], en Tchécoslovaquie par V. Cerny [CER 82] et enfin à l'ESPCI [SIA 84]. Kirkpatrick reprit l'algorithme de N. Metropolis [MET 53] qu'il appliqua à des problèmes concrets d'optimisation (placement de composants, routage de circuit...). Depuis cette méthode a fait l'objet d'une importante utilisation, comme celles citées dans [SIA 88] et [LAA 87].

5.2.2 Algorithme de Metropolis

L'algorithme du recuit simulé s'inspire largement de la méthode d'amélioration itérative, mais en la modifiant de façon fondamentale. Au lieu de rejeter une modification entraînant une augmentation de la fonction de coût, on l'accepte avec une certaine probabilité.

On espère ainsi, pouvoir remonter la fonction de coût pour sortir de la vallée, passer un col pour redescendre ensuite dans une vallée plus basse.

Le choix de la représentation nous amène à adopter le principe de Metropolis au parcours de graphe suivant:

On effectue une transformation simple passant du noeud u au noeud v . On calcule $\Delta g = g(v) - g(u)$, la variation de la fonction de coût g , et on accepte la transformation avec la probabilité $p(u,v)$ telle que :

$$(1) p(u,v) = 1 \text{ si } \Delta g \leq 0$$

$$(2) p(u,v) = e^{-\frac{\Delta g}{T}} \text{ sinon.}$$

Lorsque la variation $\Delta g \leq 0$, le nouveau placement propose une amélioration de la fonction de coût g , les chutes sont plus faibles, et donc la nouvelle configuration doit être acceptée. On lui affecte alors la probabilité maximale de 1.

Si la variation $\Delta g > 0$, la probabilité d'acceptation du nouveau placement tient compte de l'amplitude de l'augmentation de Δg (une forte augmentation est moins favorable) et du degré d'avancement de la simulation, c'est à dire le paramètre T (T est ici un paramètre de contrôle analogue à la température dans le processus réel de recuit thermique). Pour déterminer si une transformation est acceptée, on tire un nombre aléatoire p sur $[0,1]$ et on compare avec $p(u,v)$:

Si $p < p(u,v)$ la configuration v est acceptée sinon elle est rejetée et on essaie un autre noeud.

D'où l'algorithme: (cf figure 7)

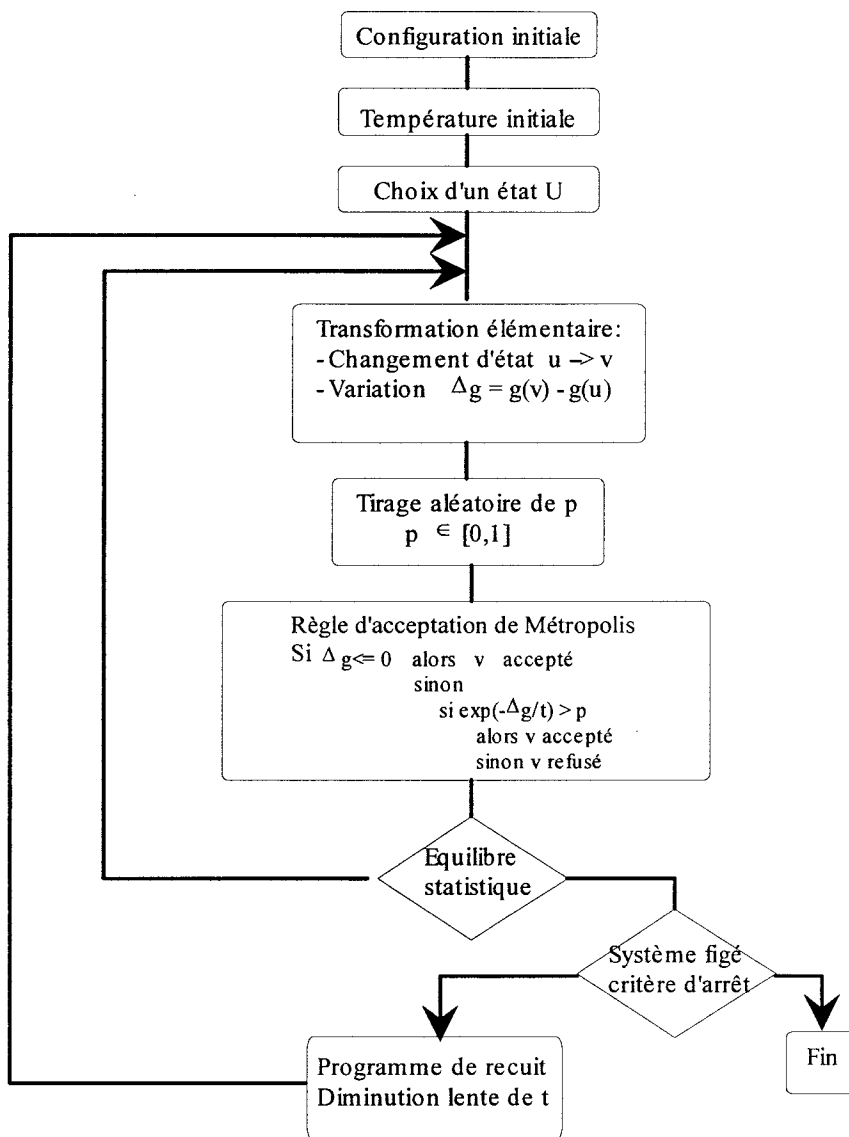


Figure 7 [SIA 88]

5.2.3 Comportement du recuit simulé sur le graphe

L'utilisation du recuit simulé pour un problème décrit par une représentation en graphe arborescent semble originale. Nous examinerons donc son comportement général et son efficacité pour cette forme de représentation.

Au niveau du graphe, l'heuristique revient en fait à "voyager" dans l'arborescence en "liberté contrôlée". Supposons que l'on vienne de traverser un noeud u situé au rang r de l'arbre. Effectuer une transformation élémentaire consiste à choisir un voisin direct de u (successeur ou prédécesseur) au sens de la topologie définie dans l'espace de configuration. On doit alors choisir aléatoirement parmi les deux possibilités suivantes:

- 1) On remonte vers le prédécesseur de u .
- 2) On engendre un successeur $O_{k,\lambda,sx,c}(u)$ en tirant aléatoirement un quadruplet de paramètres sur l'espace $[0,N] \times \mathcal{P}(k)$.

La fonction de coût $g(u)$ étant décroissante quand r croît, le choix 1) génère systématiquement un accroissement des chutes et l'acceptation du prédécesseur dépend toujours du tirage de p : Le retour arrière est toléré sous les hypothèses probabilistes de Boltzman¹. A T élevée la probabilité $p(u,v) = e^{-\frac{\Delta g}{T}}$ est proche de 1. Les remontées ne sont donc pas rares et le rang du noeud visité peut être quelconque. L'état visité peut se situer à tous les niveaux de l'arborescence et sur toute la largeur de l'arbre.

Cependant une préférence est donnée à la descente, puisque si le choix 2) est tiré, le successeur est systématiquement accepté. Lorsque le paramètre de contrôle T diminue, les remontées sont de plus en plus difficiles, et donc le rang moyen des noeuds visités est de plus en plus élevé. Les prédécesseurs étant de plus en plus rejetés, les noeuds visités sont majoritairement ceux du bas de l'arborescence. On affine la recherche au voisinage des minima locaux. Puis, le système se fige sur un noeud terminal. Comme durant le recuit, les remontées étaient en probabilité plus difficiles depuis les bons placements, le système évolue statistiquement vers le minimum global.

5.3 Paramètres de contrôle du recuit simulé

La présentation de l'algorithme de Metropolis peut paraître incomplète dans la mesure où un certain nombre de paramètres restent à préciser. En particulier, les conditions de l'équilibre statistique, le critère d'arrêt, la loi de décroissance en T et la configuration initiale sont à définir pour spécifier le modèle du recuit simulé.

¹Du nom du physicien qui formula le comportement statistique des molécules dans un gaz

5.3.1 Placement initial

Le choix de la configuration initiale n'intervient pas pour la qualité du résultat final, cependant une bonne configuration de départ permet de démarrer avec une valeur de T plus basse, reflétant que le système se trouve dans un désordre limité. Nous faisons le choix d'appliquer ici la méthode de l'amélioration itérative déjà vue. Cela revient à choisir aléatoirement une succession de formes affectées d'un vecteur de paramètres lui même aléatoire, et l'on concatène l'ensemble de ces formes. Remarquons que l'amélioration itérative consiste à appliquer le recuit simulé à T nulle (ou infinitésimale pour éviter l'indétermination sur le calcul de $p(u,v)$). La recherche du placement initial consistera donc à appliquer une session de l'algorithme du recuit avec $T \approx 0$.

5.3.2 Température initiale T_0

Pour éviter que le processus d'optimisation ne soit piégé dans un minimum local, T_0 doit être suffisamment élevé pour que toutes les configurations soient équiprobables au début de la recherche, y compris les plus coûteuses. C'est à dire que, à T_0 , la probabilité $p(u,v)$ doit tendre vers 1, même si la variation Δg est un accroissement important. Cela revient à écrire que pour une variation Δg_{max} , T_0 est telle que $(e^{-\frac{\Delta g_{max}}{T_0}}) \rightarrow 1$ soit encore $(\frac{\Delta g_{max}}{T_0}) \rightarrow 0$.

Lorsqu'on connaît Δg_{max} , [SIA 88] préconise de choisir:

$$T_0 = \mu_1 \times \Delta g_{max}, \quad \text{avec } \mu_1 \gg 1 \quad (\text{typiquement } 10).$$

Avec la connaissance des formes présentes dans l'encours de placement, et compte tenu de la définition de $g(u)$, la variation maximale de Δg_{max} correspond au retrait de la forme de surface la plus élevée, c'est à dire $\Delta g_{max} = (\underset{\substack{F_k \in EP \\ a_k \neq 0}}{\text{Max}} (S_k))$. Quelques essais

présentés en résultats nous ont amené à retenir $T_0 = \mu_1 \underset{\substack{F_k \in EP \\ a_k \neq 0}}{\text{Max}} (S_k)$, avec $\mu_1 = 1$. Nous

remarquerons, d'après les résultats fournis dans la suite, que la vitesse de croissance moyenne du rendement est optimal pour $\mu_1 = 1$. Une valeur plus élevée de μ_1 manifeste une vitesse de convergence moins élevée, une valeur plus faible gèle la construction du placement trop rapidement et piège la recherche dans un minimum local trop vite.

5.3.3 Changement d'état

La transformation d'état s'effectuera au moyen de la procédure suivante:

Soit $CHEMIN(v)$ la liste des antécédents du noeud courant v (y compris v).

Si u est le prédécesseur de v , on a $CHEMIN(v) = (\text{racine}, \dots, u, v)$

Transforme_Elem(v)

- 1) $nb_transitions := nb_transitions + 1$;
 - 2) tirer un nombre aléatoire $NA1 \in [0, 1]$;
 - 3) si $NA1 > 0.5$
 - alors
 - 3.1) tirer un nombre aléatoire $NA2 \in [0, 1]$;
 - 3.2) si $NA2 < e^{-\frac{g(v) - g(u)}{T}}$
 - alors 3.2.1) supprimer v dans $CHEMIN(v)$ "remontée acceptée"
 - sinon
 - 3.3) tirer aléatoirement un quadruplet (k, λ, s_x, c) dans $[0, 1, ..N] \times \mathcal{A}(k)$;
 - 3.4) expliciter $w := s_{k, \lambda, s_x, c}(v)$;
 - 3.5 si w existe alors ajouter w dans $CHEMIN(v)$;
 - 3.6 $nb_engendré := nb_engendré + 1$;
- Fin

Remarquons que le rang du noeud courant v correspond à la taille de la liste $CHEMIN(v)$

Nous avons choisi d'attribuer au changement d'état une probabilité identique à la montée ou à la descente dans le graphe. Tous les états accessibles ne sont donc pas équiprobables car l'accès au seul prédécesseur a 50% de chances (notons $p^{\uparrow} = 0.5$) et les successeurs se partagent les 50% restant (notons $p^{\downarrow} = 0.5$). Au début, le placement évolue alors beaucoup, des pièces sont ajoutées ou ôtées de façon très aléatoire. Mais lorsque T diminue, la probabilité d'acceptation des montées diminue, et donc la probabilité de transition vers un prédécesseur diminue aussi, au profit des successeurs. Au fur et à mesure que l'on descend dans l'arbre, on se rapproche des noeuds terminaux qui n'ont pas de successeurs. Donc le système est piégé, il est gelé. En pratique cela se caractérise par le fait que l'on essaie d'enlever la dernière forme de la bande, qu'on choisie une autre forme ou la même affectée d'un nouveau vecteur de configuration. Lorsqu'il n'y a plus de successeur, le système est gelé: le rang se stabilise. On voit donc que la probabilité de transition d'états dépend du paramètre de contrôle T dont la politique de variation est essentielle.

5.3.4 Loi de décroissance du paramètre de contrôle T

On peut montrer que la convergence a lieu lorsque l'on part d'une température suffisamment élevée et que l'on fait décroître celle-ci par palier, de sorte qu'au palier k , $T_k = \mu_2 \times T_{k-1}$ avec $0,5 < \mu_2 < 1$. La valeur de μ_2 constitue la raison de la suite géométrique des T_k . Si μ_2 est faible, les résultats sont analogues à ceux de la trempe thermique c'est à dire ici que le placement se stabilise dans une configuration qui n'est pas forcément la plus satisfaisante. On peut alors observer des désordres locaux, des formes dans une mauvaise position ou pouvant être remplacée par une forme plus adaptée. Si μ_2 est trop proche de 1, la décroissance est très lente. Pour notre problème, on a obtenu de bons résultats avec $\mu_2 = 0,91$. En réalité, le taux de décroissance μ_2 est très lié au nombre d'essais effectués dans chaque palier pour atteindre l'équilibre statistique.

5.3.5 Equilibre statistique

Il détermine le moment où l'on doit changer de température. La théorie montre que, à température constante, l'équilibre est atteint lorsque l'on a stationnarité de la distribution de probabilité de la chaîne de Markov. Celle ci est constituée de la suite des probabilités associées à chaque configuration visitée. Si la température est constante, cette dernière est alors homogène. Cela signifie que l'on a alors toutes chances de repérer la meilleure solution si l'on fait tendre le nombre de transitions vers l'infini. La méthode converge alors asymptotiquement vers l'optimum global. En pratique, on essayera un nombre limité de configurations par palier, si l'on se satisfait d'une solution quasi-admissible. Il s'agit donc d'une heuristique du recuit. On introduit alors un paramètre supplémentaire. Soit M , la longueur des chaînes de Markov, égal au nombre de configurations pour le palier à la température T . [Aar 85] montrent que moyennant quelques approximations et si le taux de décroissance de T est suffisamment proche de 1, la longueur des chaînes de Markov peut être égale au nombre d'états accessibles depuis la configuration courante. Il vient que:

$$M = \underset{u}{\text{Max card}} S(u) + 1.$$

En ce qui nous concerne, M peut s'écrire:

$$M = \underset{k}{\text{Max}} [\text{card} [0, N] \times \text{card} P(k)] \leq N \times 32,$$

$$a_k \neq 0$$

N étant le nombre de formes productibles présentes dans EP . Selon le degré de contrainte des formes de l'encours de placement, M sera plus ou moins élevé.

5.3.6 Critère d'arrêt

Il caractérise le fait qu'aucune transformation n'est plus acceptée et signale la fin du traitement. On peut ainsi décider de s'arrêter lorsque le rang moyen des états visités reste relativement constant, c'est-à-dire que le chemin de l'arbre n'évolue plus. Ceci se manifeste lorsque $\overline{card(CHEMIN)} \approx cte$ sur deux paliers de température successifs. Mais il peut être coûteux en temps de s'attacher à ce seul critère. Une autre possibilité consiste à se satisfaire de la solution de coût minimum obtenue lorsque la vitesse de décroissance relative de la fonction de coût est inférieure à une vitesse seuil V_{min} . Nous adoptons ici le principe énoncé dans le chapitre II concernant la contrainte temporelle. Un ralentissement de l'efficacité de la recherche arrêtera donc l'algorithme. Enfin, lorsque durant la recherche, par chance une bonne solution vient à surgir rapidement, il conviendra de se satisfaire de la solution proposée si celle-ci offre un rendement supérieur à un seuil de rendement maximal ρ_{max} que l'utilisateur pourra spécifier en fonction de ses desiderata.

L'arrêt de la recherche est donc provoqué par la validation de la condition suivante:

$$\left(\frac{\overline{card(CHEMIN)}_{k+1} - \overline{card(CHEMIN)}_k}{\overline{card(CHEMIN)}_k} \right) < \left(\Delta r/r \right)_{min}$$

ou

$$(nb_noeuds_engendrées - e_n) > \frac{g_{n-1} - g_n}{g_{n-1} \times V_{min}}$$

ou

$$(\rho > \rho_{max})$$

avec :

- ♦ $\overline{card(CHEMIN)}_{k+1}$ rang moyen sur le palier k+1.
- ♦ k = indice du palier à T constante.
- ♦ $(\Delta r/r)_{min}$ la variation relative minimale du rang moyen (typiquement 1%).
- ♦ $V_{min} = 1/100.000$ (diminution de 1% du coût pour 1000 transitions).
- ♦ e_n le nombre de noeuds explicités pour obtenir la dernière solution de coût minimal g_n .

♦ ρ_{\max} le rendement maximal. Une valeur élevée caractérise la volonté de trouver une solution de bonne qualité. Cet objectif ne sera atteint que si les autres conditions d'arrêt ne sont pas actives avant l'activation de celle-ci. Une valeur trop faible provoquera sans doute un arrêt prématuré.

5.4 Résultats

5.4.1 Constitution de l'encours de placement type

Dans ce qui suit, on se propose d'observer les résultats obtenus à l'aide de cette méthode. Pour cela, nous avons décidé de la composition d'un ordre de coupe, que l'on gardera tout au long de ce chapitre. L'*encours de placement EP* est constitué de:

Formes F_k	A	B	C	D	E	F
effectif a_k	10	10	15	15	15	15
surface A_k (en pixels ²)	35068	30167	1595	4412	1968	1199
longueur (en pixel)	369	371	75	168	153	60
largeur (en pixel)	144	124	33	27	12	21
orientations permise	$\{0, \pi\}$	$\{0, \pi\}$	$\{0, \pi/2, \pi, 3\pi/2\}$	$\{0, \pi/2\}$	$\{0, \pi/2\}$	$\{0, \pi/2\}$
symétrie S_{crx}	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$

Tableau 1 : constitution de l'encours de placement EP

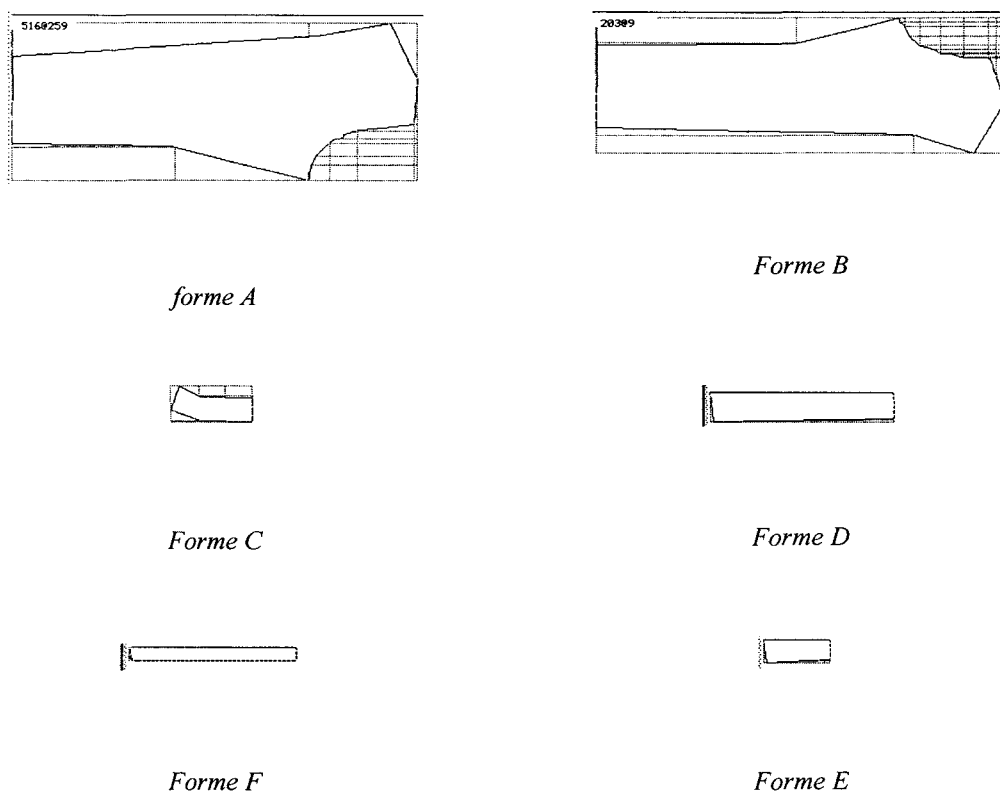


figure 9: Contours des 6 formes du carnet de commande

5.4.2 Etude d'un exemple

L'évaluation des pertes est ici présentée sous la forme du rendement $\rho = \frac{\text{surface utile}}{\text{surface occupée}}$. Cet indice facilite la comparaison, car il est indépendant de la longueur de la bande qui peut varier à $\alpha\%$ près d'une solution à une autre.

La complexité de la recherche est mesurée par le nombre de noeuds engendrés, reflétant exactement le nombre de procédures de recherche d'imbrications. C'est en effet ces dernières qui occupent le calculateur dans sa majeure partie, et le nombre de noeuds engendrés est très significatif du temps de calcul.

Les résultats présentés concernent la réalisation de la première bande productible à l'aide des formes référencées dans l'encours de placement *EP*.

L'exemple suivant fait apparaître la progression des rendements des différents minima locaux obtenus au cours d'un essai où nous avons fixé :

- ◆ Température initiale: coefficient $\mu_1 = 3$.
- ◆ Décroissance en température: $\mu_2 = 0,91$.
- ◆ Rendement maximal $\rho_{\max} = 80\%$.
- ◆ Laize de la matière : 500 pixels.
- ◆ L'ensemble des connexités admises a été paramétré à $\{0,2\}$ (concaténation des formes par le dessus ou le dessous)

Dans cet exemple, c'est la diminution de la vitesse de décroissance en coût qui arrêta la recherche.

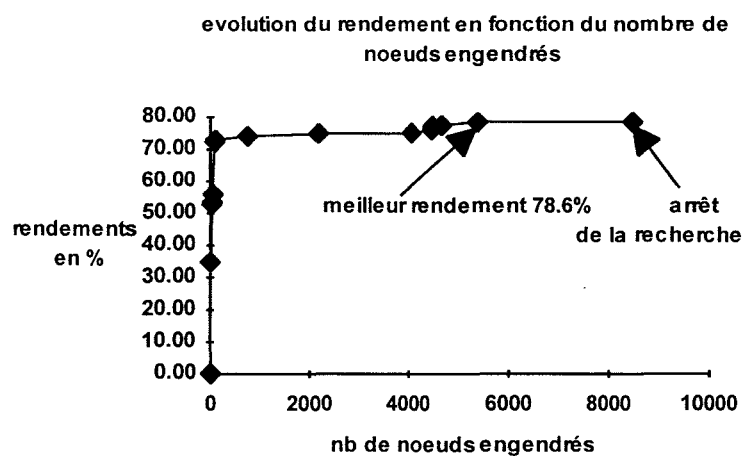


Figure 10

Interprétation: la courbe précédente est typique au regard de tous les essais effectués avec cette méthode. La variation du rendement est très forte au début de la recherche, symbolisant la descente rapide vers le bas de l'arbre lors de la réalisation de la solution initiale à température nulle. Ensuite, la température est brusquement augmentée à la valeur indiquée précédemment (ici $\mu_1 * S_{\max} = 3 * 35068 = 105204$), pour décroître progressivement. Le rendement évolue donc plus lentement pendant le recuit simulé.

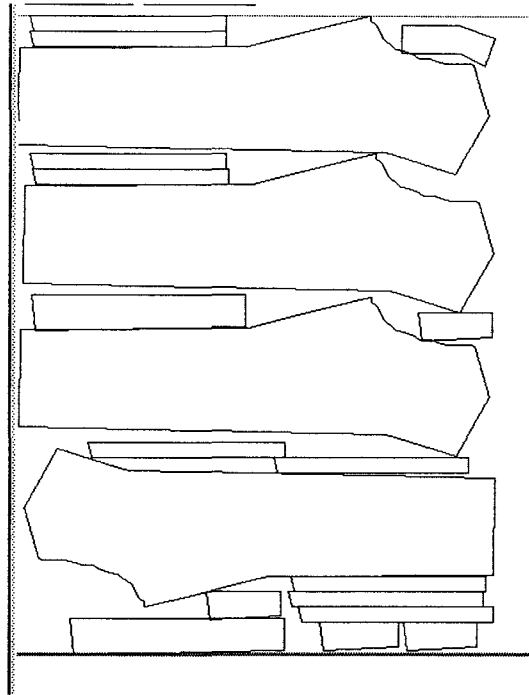


Figure 11: placement correspondant à la solution finale (78.6%)

Remarque: Un ensemble de côtés de contact admissibles porté à $\{0,1,2,3\}$ (concaténation possible selon les quatre côtés) aurait sans doute permis d'obtenir un rendement plus élevé (récupérations de l'espace disponible sur les côtés) au prix d'un nombre de noeuds engendrés plus élevé.

5.4.3 Ajustement des paramètres

a) Ajustement de la vitesse de décroissance en T

Nous faisons apparaître ici, les différents résultats obtenus pour le même encours de placement, en faisant varier les différents paramètres contrôlant le recuit simulé. Pour tous ces essais, c'est la diminution de la vitesse de décroissance relative qui arrêta la recherche.

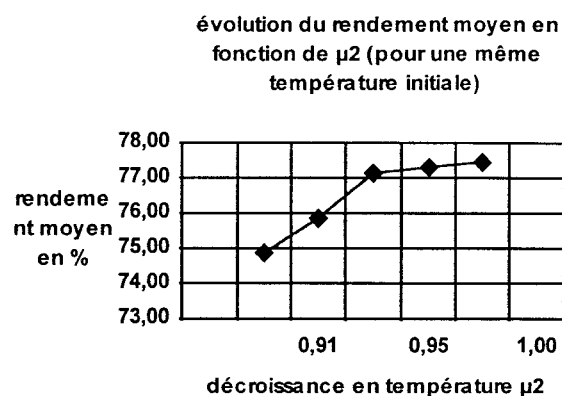
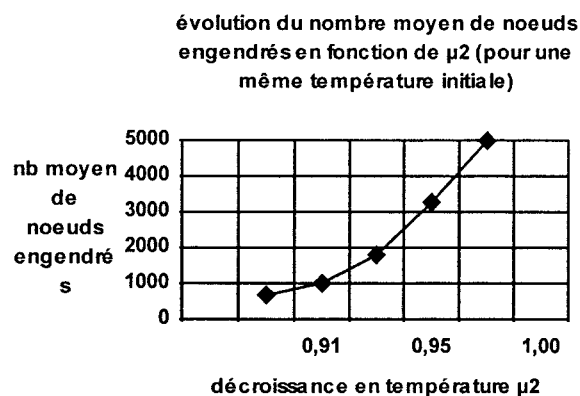


Figure 12 : évolution du nombre d'états engendrés en fonction de la décroissance en température

Figure 13 : évolution du rendement moyen en fonction de la décroissance en température

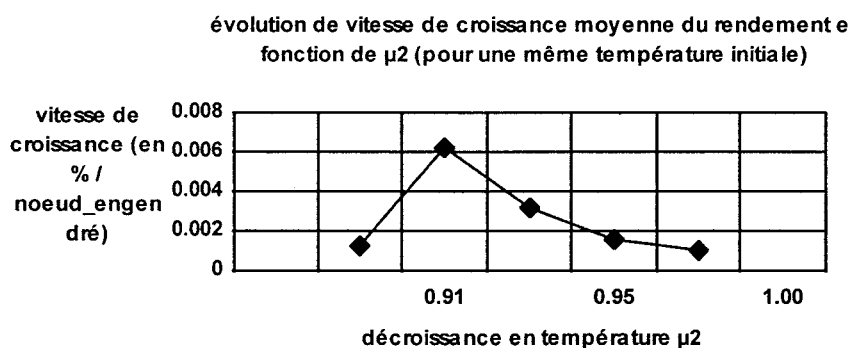
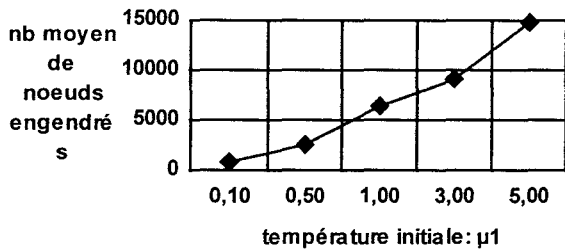


Figure 14 : évolution de la vitesse de croissance du rendement (entre la solution initiale et la solution finale) en fonction du taux de décroissance en température

Ces différentes courbes font apparaître des résultats classiques. Le nombre de noeuds à expliciter est très élevé lorsque la décroissance en température est lente (cf. figure 12). Parallèlement, les rendements augmentent aussi, mais beaucoup plus lentement (cf. figure 13). Le rapport entre l'augmentation de rendement réalisé grâce au recuit simulé (rendement final - rendement solution initiale) et le nombre de noeuds engendrés nécessaire à l'obtention de ce résultat, fait remarquer une valeur optimale de la décroissance en température. $\mu_2 = 0,91$ semble donner un bon compromis temps/rendement.

b) Ajustement du paramètre T_0

évolution du nombre moyen de noeuds engendrés en fonction de μ_1 (pour une même décroissance en température $\mu_2=0.95$)



évolution du rendement moyen en fonction de μ_1 (pour une même décroissance en température $\mu_2=0.95$)

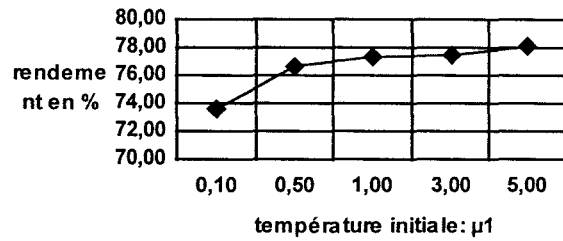


Figure 17

Figure 16

Les courbes précédentes montrent un résultat classique. Plus la température initiale est élevée, plus la recherche est longue pour obtenir un rendement intéressant (cf. figure 16 et 17). En effectuant le même rapport que dans le cas précédent, on constate que le compromis vitesse/temps semble optimal pour $\mu_1=1$

évolution de la vitesse de croissance en fonction de μ_1 (pour un même décroissance en température $\mu_2=0.95$)

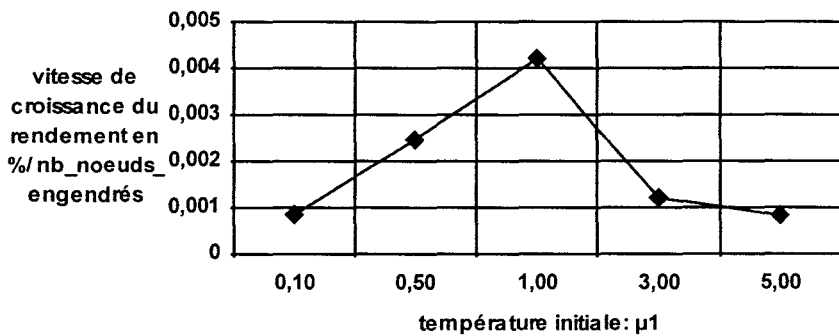


Figure 18

5.5 Conclusions sur la méthode du recuit simulé

Dans un premier temps nous avons relaté l'expérience effectuée avec le recuit simulé en développant un générateur de configuration aléatoire. Sans remettre en cause le principe, nous nous sommes aperçus que la nécessité de traiter des instances plus importantes du problème nous obligeait à revoir la représentation.

Nous nous sommes ensuite orientés vers une adaptation de la méthode du recuit simulé au parcours d'arbre. La modélisation du placement de formes à travers un graphe où tous les états sont acceptables du point de vue des contraintes, nous permet d'interpréter sous un autre angle le déroulement du recuit simulé. D'autre part, les résultats obtenus sont différents de ceux présentés dans [PRE 83] où précisément les contraintes n'étaient pas implicites, mais explicitées dans la fonction de coût. Le nombre d'itérations nécessaires était alors très élevé pour obtenir un placement satisfaisant les contraintes, et la convergence déjà assez lente dépendait en plus de la finesse du pas de variation accordé à la translation des formes, et de la richesse du domaine de commande. En ce qui nous concerne, nous avons préféré un arrêt plus rapide de la recherche tout en s'attachant à caler les paramètres de contrôle afin que la vitesse de convergence soit maximale. Pour notre cas, nous avons constaté que les valeurs de $\mu_1=1$ et $\mu_2=0.91$ semblent donner les meilleurs résultats aux vues des expérimentations donnant lieu aux relevés précédents. Naturellement, ces valeurs n'ont pas un caractère universel. Une variation sensible de l'encours de placement va déplacer la valeur optimale de μ_1 et μ_2 . Cependant, à titre d'essais, nous avons essayé de moduler l'encours de placement et avons remarqué une certaine constance de l'efficacité de recherche.

Par rapport à la première implantation du recuit simulé, la deuxième méthode présentait un domaine de commande moins riche. Par exemple, les permutations n'étant pas implémentées, on a remarqué que cette absence était ressentie au niveau de la vitesse de convergence. Il serait peut-être souhaitable de construire des macro-opérations visant à franchir plusieurs arcs lors du parcours de l'arbre afin de simuler certaines opérations, la permutation par exemple..

Mais globalement, les résultats obtenus correspondent aux attentes, compte tenu d'une volonté initiale de tendre asymptotiquement vers le minimum global, tout en acceptant de s'arrêter avant l'obtention de celui-ci.

VI Parcours de graphe: l'algorithme ε -admissible

6.1 Introduction

La modélisation du problème de placement sous forme arborescente nous conduit naturellement à étudier les méthodes de parcours d'arbre proposées dans bon nombre de manuels traitant de l'intelligence artificielle. Mais la multitude des techniques potentiellement utilisables invite à faire le choix d'une méthode particulière. Nous procédons dans un premier temps à la genèse de ce choix. Nous expliciterons ensuite l'implantation de l'algorithme choisi ainsi que ses aménagements.

6.2 Choix de l'algorithme

Etant donné un espace de configuration (notre graphe $G(U,S)$), la recherche d'un chemin sur le graphe peut se caractériser en première approche par le sens de parcours et la stratégie d'exploration.

Le sens du parcours désigne le sens général de la recherche selon que l'on démarre d'un état initial vers l'état terminal solution du problème, ou que l'on part du but à atteindre et que l'on cherche la séquence d'opérateur permettant d'atteindre ce but depuis un état connu. Le premier s'appelle chaînage-avant, celui que l'on a retenu compte-tenu que notre graphe est défini implicitement et que les états terminaux sont ceux que l'on cherche. Le deuxième représente le chaînage-arrière, plus adapté lorsqu'il s'agit par exemple de démontrer un théorème.

La stratégie d'exploration spécifie le mode de contrôle de la recherche [DOM 88]

1) La recherche "incrémentale" consiste à sélectionner un état relativement à un critère sans jamais revenir en arrière. Le choix d'un état n'est jamais remis en cause. Il s'agit d'un processus de recherche sans mémorisation et sans retour, d'une stratégie de contrôle irrévocable [NIL 82], ou d'un algorithme déterministe. L'amélioration itérative aléatoire, le "hill climbing", l'algorithme glouton en sont des exemples.

2) La recherche avec mémorisation et retours limités ("backtrack") consiste à explorer un chemin jusqu'à un noeud terminal. Si celui-ci n'est pas satisfaisant, on remonte jusqu'au dernier choix effectué pour prendre une autre direction. Nous profitons de cette définition pour ajouter que la méthode du recuit simulé au sens où nous l'avons interprété, constitue une variante de cette classe d'algorithme. On y pratique également des retours sur le chemin en mémoire, mais avec une probabilité de transition variable. Il s'agit ici d'un retour arrière probabiliste.

3) La recherche avec mémorisation du graphe ou recherche en graphe se caractérise par un contrôle révisable des chemins. Le principe est de conserver la trace de tous les noeuds rencontrés (ou seulement les plus prometteurs). Au fur et à mesure de l'exploration de l'arbre, tous les états générés sont mémorisés, afin de pouvoir remettre en cause à tout moment le chemin suivi. Lorsqu'une solution présente un coût non satisfaisant, il est alors possible de choisir parmi les noeuds mémorisés un autre noeud avec l'espoir d'améliorer le coût final. Il s'agit aussi d'une stratégie non déterministe.

La recherche en graphe offre naturellement des avantages sur la qualité du résultat final (par rapport à la stratégie incrémentale) et sur la complexité de la recherche (par rapport au retour arrière). Mais les politiques de recherche peuvent varier aussi selon l'information dont on dispose pour l'évaluation d'un chemin à suivre.

6.3 Politique de recherche en graphe

6.3.1 Recherches

Si la recherche est non informée, on peut utiliser la recherche en profondeur d'abord ou en largeur d'abord.

Mais lorsque les arcs sont valués, on peut procéder alors à une recherche informée. Si l'information se limite à la connaissance du passé, on parle de recherche à coût uniforme. Par contre, on peut également introduire une "estimation de l'avenir". La fonction coût s'exprime alors par la somme de deux composantes (coût du parcours passé + estimation du coût futur = $g(u)+h(u)$). Ce principe est à la base des algorithmes A ainsi que ses dérivés (A^* , $A\epsilon$).

6.3.2 L'algorithme A

A chaque noeud on associe deux informations:

- Le coût du chemin parcouru $g(u)$ comme pour l'algorithme de coût uniforme
- Le coût du chemin à parcourir. Supposons que $h^*(u)$ soit le coût minimal d'un chemin allant de u à la solution de coût minimal du sous graphe descendant de u , alors $h(u)$ est une estimation de $h^*(u)$. Soit $f(u)=g(u)+h(u)$ une estimation du coût du chemin entre la racine et le noeud terminal de coût minimal accessible depuis u et passant par u . Soit NDP l'ensemble des noeuds pendants engendrés lors de la recherche. L'algorithme qui consiste à développer en premier les noeuds sélectionnés par valeurs de f croissantes dans NDP est appelé algorithme A.

6.3.3 L'algorithme A*

L'algorithme A* est un algorithme A muni de la propriété d'admissibilité.

On dit que l'heuristique h vérifie la condition d'admissibilité si elle est minorante:

$\forall u, h(u) \leq h^*(u)$. Une démonstration de ce théorème est donnée dans [FAR 87].

Conséquence: A* trouve le chemin optimal (s'il en existe un).

L'algorithme se caractérise également par deux autres propriétés:

Terminaison d'A*

Pour toute h , application de U dans \mathfrak{R} , et pour tout G , graphe fini à valuations positives ou nulles, A* s'arrête soit sur un état terminal, soit faute d'état pendant ($NDP = \emptyset$) s'il n'existe pas de chemin de u_0 à $t \in T$.

Complexité d'A*

Pour une heuristique minorante, le sous graphe G' auquel se limite la recherche est restreint aux seuls états u tel que $U' = \{u \in U \mid g^*(u) + h(u) \leq f^*(u_0)\}$. $f^*(u_0)$ est le coût du chemin minimal joignant u_0 et $u \in T$. $Card(U')$ est le nombre de noeuds dans le sous graphe G' . Pour une heuristique h minorante et si de plus elle est monotone: ($\forall u, \forall v \in S(u), h(u) - h(v) \leq k(u, v) = \text{coût pour passer de } u \text{ à } v$), alors A* effectue au plus $Card(U')$ développements. Pratiquement, si r^* est le rang de la seule solution optimale et pour un arbre K-aire uniforme (ici $K = \underset{a_{k \neq 0}}{\underset{k}{\text{Max}}} [card [0, N] \times card P(k)]$), la complexité est de la forme exponentielle $O(K^{r^*})$ [POH 77].

Pour le même modèle avec $\delta \in \mathfrak{R}^+$,

a) Si $(1-\delta)h^* \leq h \leq h^*$, alors A* est exponentiel en $O(K^{\delta r^*})$

b) Si $h^* - \delta \leq h \leq h^*$, alors A* est linéaire en $O(r^* K^\delta)$ [GAS 79].

Mais le théorème général de [PEA 83] établit qu'il est rarement possible d'obtenir mieux que la complexité exponentielle de la longueur du meilleur chemin. L'admissibilité est donc une propriété coûteuse. Une solution approximative satisfaisant le compromis admissibilité/complexité peut être obtenu à partir de l'algorithme A ϵ . Il présente l'avantage de pouvoir régler ce compromis.

6.4 L'algorithme A_ε

6.4.1 Principe

Comme pour l'algorithme A^* , l'algorithme A_ε est basé sur le principe de l'estimation du coût à l'arrivée à un noeud terminal ($f(u) = g(u) + h(u)$). On dit qu'un algorithme est ε -admissible si et seulement s'il fournit une solution acceptable "t" de coût :

$$f(t) \leq \text{seuil}, \text{ où } \text{seuil} = (1 + \varepsilon)f^*(u_0).$$

$f^*(u_0) = g(u_0) + h^*(u_0)$ est le minimum du coût des chemins allant de u_0 à la solution optimale. Comme $f^*(u_0)$ n'est pas connu a priori, on utilise un estimateur minorant $f(u) = g(u_0) + h(u)$. ε représente le maximum de l'écart relatif que tolère l'utilisateur entre la solution optimale et la solution qu'il recherche.

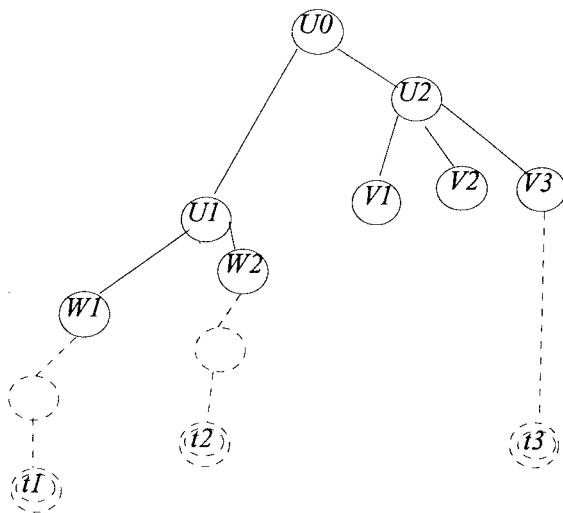
Donc, au moment du choix d'un noeud v à développer, on cherchera en priorité parmi les successeurs acceptables de u venant d'être engendrés tel que :

$$v \in \text{Succ_Acc} \Leftrightarrow v \in S(u) \text{ et } f(v) \leq (1 + \varepsilon)f(u_0).$$

Notons que succ_Acc est un sous ensemble de NDP . Le seuil d'acceptabilité est remis à jour chaque fois que l'on trouve un plus grand minorant, i.e. $f(u_0)$ plus près de $f^*(u_0)$. Ce plus grand minorant est donné par le noeud pendant de f minimal i.e. $f(\hat{u})$, \hat{u} tête de liste de NDP . Si au moment de développer un nouveau noeud, l'ensemble Succ_Acc est vide, on choisira alors un noeud dans NDP . Ce cas de figure correspond à un retour arrière dans le graphe pour développer un autre chemin.

Le choix d'un noeud à développer s'effectuera au moyen d'une procédure de choix Choix_dans(.) où (.) est l'argument de la procédure pouvant être les listes NDP ou Succ_Acc . L'idée consiste alors à choisir le noeud pouvant prétendre atteindre une solution acceptable pour une complexité limitée. Pour cela, on calcule la fonction de choix $f_c(u) = f(u) + \gamma * h_c(u)$. Le terme $h_c(u)$ est une estimation du nombre d'arc du chemin le plus court entre u et un noeud terminal. Le noeud à développer est celui dont la valeur $f_c(u)$ est minimale. Le paramètre γ permet également de régler le compromis admissibilité/complexité dans le choix d'un chemin. Plus γ est grand, plus on favorise le choix des chemins courts mais qui ne sont pas nécessairement de coût moindre.

Il arrive qu'aucun successeur d'un noeud venant d'être développé ne soit acceptable. Cependant, s'il existe parmi eux un état qui soit proche de l'acceptabilité et qu'une solution soit à quelque rang en dessous de cet état, on peut alors tenter de continuer le développement en profondeur du meilleur des états présents dans NDP , c.a.d. \hat{u} . Cet investissement sert alors à augmenter le seuil d'acceptabilité pour que le chemin précédemment exploré redevienne acceptable. C'est une tactique de persévérance.

Tactique de persévérance

1) $f(u_1) < f(u_2)$, mais u_2 a été choisi car moins complexe ($f_c(u_2) < f_c(u_1)$)

2) u_2 vient d'être développé mais $f(v_1), f(v_2), f(v_3) > \text{Seuil}$ et ne sont donc pas acceptables

3) On développe u_1 (meilleur en coût que u_2) pour augmenter le seuil.

4) si $f(v_1), f(v_2), f(v_3) \leq \text{seuil}$, on peut continuer à descendre par u_2

Figure 19

6.4.2 Algorithme

On rappelle ici qu'un noeud est un placement de formes concaténées et que le passage d'un noeud à un de ses successeurs est représenté par un arc symbolisant l'ajout d'une forme supplémentaire selon une certaine configuration.

L'algorithme utilise les listes suivantes:

NDP: liste des noeuds pendants (placements inachevés) ordonnée selon les valeurs croissantes de f , et décroissantes de g en cas d'égalité sur f .

Successeurs: liste des successeurs du noeud venant d'être développé (placements complétés mais inachevés). Cette liste est rangée selon les f croissant et g décroissant.

Succ: liste de sauvegarde des successeurs du noeud venant d'être développé pendant la boucle de persévérance.

Succ_Acc: liste des successeurs acceptables du noeud venant d'être développé (placements complétés dont la chute estimée est acceptable).

Solutions: liste des noeuds terminaux rangés selon les g croissants (bandes finales successives)

A ϵ entrée: ϵ 1) Initialisation:1.1 $NDP := (u_0)$ 1.2 $g(u_0) := Wm * \underset{\substack{F_k \in EP \\ a_{k,0} \neq 0}}{\text{Max}} L_k;$ 1.3 $f(u_0) := g(u_0) + h(u_0);$ 1.4 $\text{Seuil} := (1 + \epsilon)f(u_0);$ 1.5 $\text{Solutions} := \emptyset;$ 1.6 $\text{développer}(u_0);$ 1.7 $\text{Succ_Acc} := (v \in \text{Successeurs} / f(v) \leq \text{Seuil});$ 2) Boucle principaleTantque $NDP \neq \emptyset$ et $\forall t \in \text{Solutions}: f(t) > \text{seuil}$ 2.1 si $\text{Succ_Acc} \neq \emptyset$ alors $u := \text{Choix_dans}(\text{Succ_Acc})$
sinon $u := \text{Choix_dans}(NDP);$ 2.2 $\text{développer}(u);$ 2.3 $\text{Succ} := \text{Successeurs} \{ \text{Sauvegarde de Successeurs} \}$ 2.4 Tantque Persévère et $\text{Successeurs} \neq \emptyset$ et $\forall v \in \text{Successeurs}: f(v) > \text{seuil}$ 2.4.1 $\text{développer}(\hat{u}) \{ \hat{u} \text{ en tête de } NDP \}$ *FinTantque;*2.5 $\text{Successeurs} := \text{Succ} \{ \text{Restitution de Successeurs} \}$ 2.6 $\text{Succ_Acc} := (v \in \text{Successeurs} / f(v) \leq \text{seuil});$ 2.7 $\forall u \in NDP, \forall v \in NDP, \text{ si } f(u) = f(v) \text{ alors supprimer } v;$ 2.8 $\forall u \in NDP, \text{ si } f(u) > f(\hat{t}) \text{ alors supprimer } u \{ \hat{t} \text{ tête de la liste Solutions} \};$ 2.9 Tantque $\text{Card}(NDP) > 400$ supprimer l'élément en queue de NDP *FinTantque.*3) Sortie3.1 $\text{sortir } \hat{t} \in \text{Solutions} \{ \hat{t} \text{ tête de la liste Solutions} \}$ 3.2 $\epsilon' := [(1 + \epsilon) / \text{Seuil}] * f(\hat{t}) - 1$ 3.3 $\text{afficher } P_{\hat{t}} \text{ et } \epsilon' \{ P_{\hat{t}} = \text{le placement associé au noeud } \hat{t}, \epsilon' \text{ l'écart relatif de la solution } \hat{t}, \epsilon' \leq \epsilon. \}$ *fin.*

Cet algorithme s'arrête dès qu'une solution est acceptable ou lorsqu'il n'y a plus de noeud pendant. Lors de l'arrêt de la recherche, on peut établir l'écart ϵ' réel de la solution obtenue par rapport à l'objectif. Ceci est calculé en 3.2).

L'algorithme fait appel à la procédure $\text{développer}(u)$ exécutant la séquence suivante:

- Après avoir supprimé le noeud à développer de la liste NDP , tous les successeurs sont engendrés s'ils existent. Les coûts estimés sont ensuite calculés et les successeurs sont affectés à Successeurs et NDP .

Si *Successeurs* est vide, c'est donc que le noeud développé est un noeud solution. Le seuil d'acceptabilité est ensuite remis à jour en prenant la valeur de f en tête de liste.

développer(u)

- 1) Supprimer u de NDP;
 - 2) *Successeurs* := \emptyset ;
 - 3) $nb_noeud_devel := nb_noeud_devel + 1$;
 - 4) Pour $(k, \lambda, S_x, c) \in [0, N]^* \mathcal{A}(k)$
 - 4.1) Si $a_{k,u} \neq 0$

alors

 - 4.1.1 $v := s_{k, \lambda, S_x, c}(u)$;
 - 4.1.2 $nb_noeud_engend := nb_noeud_engend + 1$;
 - 4.1.3 Si $v \neq \emptyset$

alors

 - 4.1.3.1 $f(v) := g(v) + h(v)$;
 - 4.1.3.2 ajouter v dans *Successeurs* et dans NDP
- FinPour*;
- 5) si *Successeurs* = \emptyset *alors* ajouter u dans *Solutions*;
 - 6) $Seuil := \text{Max}\{Seuil; (1+\epsilon)f(u)\}$ { u tête de liste de NDP }

fin

6.4.3 Heuristique d'estimation de coût h

L'estimation du coût est sans doute l'opération la plus délicate de l'implantation de l'algorithme. C'est d'elle dont dépend la qualité du placement solution et la complexité pour l'obtenir.

Compte-tenu de la définition de $g(u)$ ¹ qui comptabilise les pertes par rapport au rectangle de la matière consommée, l'idée consiste à estimer la surface totale des formes qu'il est possible de placer sur l'espace restant disponible, c'est à dire le remplissage espéré. Soit $h(u)$ ce remplissage. $h(u)$ sera donc négative, signifiant que l'on ajoute une perte négative à la perte existante $g(u)$. La perte totale estimée sera bien $f(u) = g(u) + h(u)$.

Définition de la surface récupérable A_r

La surface récupérable $A_r(u)$ est le complémentaire de l'enveloppe du placement courant P_u par rapport au cadre de matière fixé par les limites (W_m, L_m) .

¹ ici $g(u)$ est nécessairement additif pour pouvoir être ajouté à la perte estimée $h(u)$. Le choix de la perte surfacique s'impose.

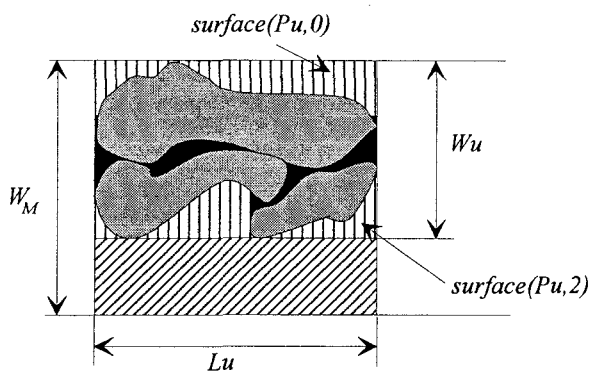


Figure 20

On admettra l'approximation suivante:

$$Ar(u) \approx (W_M - W_u) * L_u + surface(P_u, 0) + surface(P_u, 2)$$

où $surface(P_u, c)$ est la surface du peigne réduit du côté c de P_u .

Cette surface évalue l'espace qu'il est possible d'utiliser en plaçant des formes au-dessus ou au-dessous du groupement courant.

Comme la fonction $h(u)$ doit être minorante et puisque $h(u) \leq 0$, on cherchera à avoir $|h(u)|$ majorant. Le remplissage doit être optimiste. On peut par exemple convenir que:

$$h(u) = -Opt * Ar(u) \quad \text{où } Opt \text{ est un coefficient de l'ordre de } 0,9.$$

Un remplissage supérieur à 86% est assez rarement atteint en placement textile. Nous retiendront donc $Opt = 0.86$. Cette valeur doit bien entendu être ajustée en fonction de la nature des formes et des rendements moyens obtenus manuellement.

Mais cet optimisme doit être tempéré pour ne pas trop s'éloigner de l'heuristique parfaite $h^*(u)$. Si les formes présentes dans le carnet de commande sont de tailles importantes, ou que la surface totale des petites pièces est faible, les espaces internes risquent d'être importants et l'absence des petites pièces ne permettra pas de les combler. Il convient donc de compenser cet optimisme par un coefficient de correction appelé *Correcteur*.

Ce coefficient tient compte ici de la surface des petites formes. Si le total de la surface des petites formes est supérieur ou égal à 1/3 de la surface à remplir ($W_M * L_u$), le remplissage sera probablement bon et *Correcteur* = 1. Si ce rapport est inférieur à 1/20, le remplissage sera sans doute moins serré et donc *Correcteur* est fixé à 0.85. Entre ces deux rapports, on a imposé une simple évolution linéaire.

Ce choix est arbitraire et la simplicité de l'allure de cette fonction permettait de faciliter les premiers tests. Il se trouve que les résultats obtenus sont encourageants. Les valeurs caractéristiques de cette fonction ont été établies expérimentalement. Remarquons toutefois que la fonction de correction doit normalement satisfaire certaines conditions afin que la fonction f soit croissante. Cependant, il ne nous a pas été permis de le démontrer.

Notons enfin qu'une forme est arbitrairement considérée "petite" lorsque sa surface est inférieure à 1/3 de la surface de la plus grande forme restante à placer.

La fonction de correction aura donc le profil suivant:

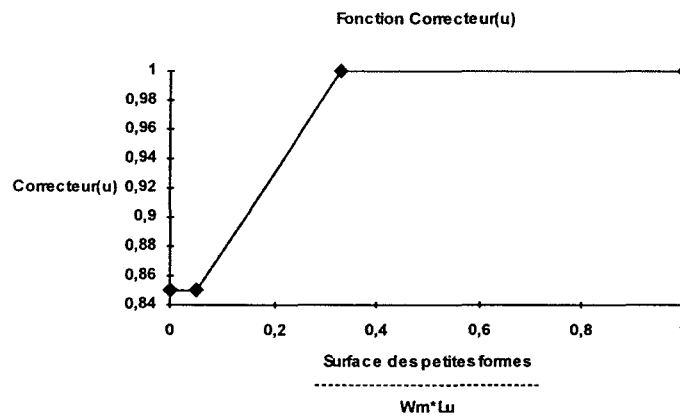


Figure 21

Finalement:

$$h(u) = -A_r(u) * Opt * Correcteur(u), \text{ avec } Opt = 0.86$$

Remarquons que $Correcteur(u)$ est une fonction décroissante par rapport au rang. En effet, au cours d'une descente en profondeur de l'arbre deux tendances cohabitent:

- Le nombre de petites formes tend à diminuer, faisant diminuer aussi la surface totale de cette classe
- Le nombre de grandes formes décroît aussi, faisant ainsi descendre le seuil de sélection des petites formes.

On peut donc écrire

$$\forall v \in S(u), Correcteur(v) \leq Correcteur(u)$$

Cette propriété sera utile pour démontrer la croissance de $f(u)$.

Le modèle retenu pour calculer le coefficient correcteur est simple et ne prend donc pas en compte toutes les situations. A ce niveau, la connaissance de l'expert est essentielle, et l'expérimentation obligatoire. Les valeurs présentées ici donnent de bons résultats, mais n'ont pas, on s'en doute, un caractère universel. L'idéal serait sans doute de prendre en compte d'autres paramètres significatifs, des attributs de formes comme la complexité des contours dont on suppose l'influence.

Mais le propre de l'algorithme A est aussi de s'affranchir d'une connaissance approfondie du paysage de coût, grâce au principe de remise en cause qui lui est propre. Et puis, si le paysage de coût était connu, le secret du placement n'en serait pas un et un bon algorithme déterministe ferait sans doute mieux.

Dans l'attente de trouver une technique d'optimisation combinatoire plus efficace, on pourrait suggérer une heuristique $h(u)$ adaptative en fonction des coûts des solutions obtenues durant le passé de la recherche, afin que l'écart $h^*(u)-h(u)$ deviennent de plus en plus faible. On entrevoit ici les possibilités offertes par l'apprentissage au service de l'optimisation. Quelques applications de l'utilisation de l'apprentissage pour la recherche combinatoire sont présentées dans [PER 90].

6.4.4 Heuristique de calcul de la longueur de chemin

Cette heuristique, appelée $h_c(u)$ estime le nombre d'arcs compris entre le noeud u et le noeud terminal le plus proche. Connaissant le remplissage estimé $|h(u)|$, c'est à dire la surface que l'on va pouvoir couvrir, on peut calculer le nombre minimal de formes qu'il est possible de placer dans cette surface de remplissage. La procédure consiste donc à soustraire du remplissage courant, la surface de la plus grande forme restant à placer, si celle-ci est logeable. Sinon on essaie avec une forme plus petite. La procédure est vivante jusqu'à ce que la variable *Remplissage* soit négative.

$h_c(u)$

1) Initialisation

- 1.1) $h_c := 0;$
- 1.2) *Remplissage* := $|h(u)|;$
- 1.3) *Booléen* := *vrai*;

2) Tantque Booléen

- 2.1) $B := \underset{\substack{F_k \in EP \\ a_{k,u} \neq 0}}{\text{Min}} (\text{Remplissage} - S_k)$
- 2.2) si $B > 0$ alors
 - 2.2.1) *Remplissage* := $B;$
 - 2.2.2) $h_c := h_c + 1;$
- sinon *Booléen* := *Faux*;

FinTantque;

3) sortir $h_c(u)$

Fin

6.4.5 Tactique de recherche persévérante

La boucle 2.4 de l'algorithme principal met en oeuvre cette tactique. La condition booléenne *Persévère* est vraie si $\frac{f(\hat{u}) - \text{Seuil}}{\text{Seuil}} < 10\%$ (\hat{u} en tête de la liste *Succ*), et que le nombre d'itérations de cette boucle est ≤ 3 . On mettra en oeuvre la tactique si la différence relative entre le coût estimé du meilleur successeur et le seuil d'acceptabilité n'excède pas 10% et que la boucle d'exécution de la tactique n'a pas été sollicitée plus de trois fois consécutives. Des valeurs plus élevées de ces paramètres engageraient un investissement trop lourd pour rendre un chemin acceptable qui pourrait finalement se révéler peu prometteur.

6.4.6 Limitation des noeuds développés

Revenons un instant à la cellule de découpe. On l'a vu dans le chapitre I que le module de placement devait avoir un rythme de production de placement en rapport avec le rythme de découpe de la table. Or, pour diverses raisons (ralentissement de la découpe, rythme rapide du module de placement en raison d'un choix de ε élevé) il se peut que l'encours de coupe devienne important. Le temps supplémentaire accordé au placement peut donc être utilement consacré à améliorer les placements suivants. On peut donc diminuer ε . Au niveau de notre algorithme, cela se traduit inévitablement par une diminution du seuil d'acceptabilité. Les développements en profondeur sont plus rares et l'exploration s'effectue plus généralement en largeur. La liste *NDP* s'élargie aussi, occupant vite une mémoire limitée avec des noeuds qui sont pour la plupart inintéressants. L'idée consiste donc à élaguer le graphe.

Pour ce faire, plusieurs règles cohabitent:

1) Limiter arbitrairement la taille de *NDP*, en retirant systématiquement les noeuds aux valeurs de f les plus élevées pour revenir en deçà de la taille maximale. Cette option a été implantée à la ligne 2.9 de l'algorithme $A\varepsilon$. Mais ce choix peut conduire à des suppressions de noeuds prometteurs si la taille maximale est trop faible. Nous avons ici pris une valeur prudente de 400 noeuds.

2) Supprimer les noeuds dont les placements sont semblables. De telles situations apparaissent lorsqu'il y a existence de configurations symétriques. Ceci peut être évité en supprimant les noeuds à valeur de f identiques (ligne 2.7 de l'algorithme $A\varepsilon$).

3) Supprimer ceux dont le coût estimé est supérieur au coût estimé du meilleur noeud terminal obtenu précédemment. $NDP := NDP - \{v/f(v) > f(\hat{t})\}$ (ligne 2.8 de l'alg $A\varepsilon$). Cette option n'est envisageable qu'avec la certitude que f soit croissante. Cette propriété très importante est démontrée dans l'annexe I

6.4.7 Fonction de choix f_c

On a vu dans le § 5.4.1 que le choix d'un noeud pendant acceptable ou non, était lié à la minimisation de la fonction $fc(u)=f(u)+\gamma h_c(u)$ en parcourant les noeuds de la liste sélectionnée (*NDP* ou *Succ_Acc*). Le paramètre γ favorise, lorsqu'il est élevé, le choix des chemins courts. Or, compte tenu des options retenues pour la limitation des noeuds à développer, l'obtention d'une solution est nécessaire au début de l'algorithme afin de tirer pleinement parti de l'option 3. Pour cela, on pourra faire dépendre γ du nombre de noeuds terminaux contenus dans l'ensemble des solutions. La fonction suivante a été retenue:

$$\gamma = \frac{6000}{\ln(2 + \text{card}(\text{Solutions}))}$$

Au début de la recherche, le nombre de noeuds terminaux est nul, la valeur de γ est élevée ($6000/\ln 2 \approx 8650$) afin de favoriser le choix des chemins courts (les noeuds à valeur de h_c faible). A mesure de l'obtention de solutions intermédiaires garantissant l'existence d'un résultat, on diminue lentement la valeur de γ (par une fonction logarithmique), en vue de développer des chemins à coûts estimés plus faibles mais souvent plus longs. Cette tactique consiste en résumé à fouiller plus attentivement l'arbre pour localiser une bonne solution après avoir assuré l'acquisition de solutions de qualité moyenne.

6.5 Résultats avec $A\epsilon$

On reprend ici la même composition de l'ordre de coupe que pour les essais avec la méthode du recuit simulé (cf § 4.5). Les critères d'évaluation sont toujours les suivants:

Rendement: $\rho = \frac{\text{surface utile}}{\text{surface occupée}}$, pour le noeud terminal en tête de la liste

Solutions, c'est-à-dire la meilleure bande obtenue lors de la session de recherche.

Pertes surfaciques: $\text{surface occupée} - \text{surface utile}$

Complexité: nombre de noeuds explicités
nombre de noeuds développés.

Vitesse de décroissance de la perte surfacique:
$$\frac{\text{perte solution initiale} - \text{perte solution finale}}{\Delta \text{nb_noeud_eng}}$$

Vitesse de croissance du rendement:
$$\frac{\text{rendement solution finale} - \text{rendement solution initiale}}{\Delta \text{nb_noeud_eng}}$$

Les résultats rassemblés dans les tableaux suivants sont ceux obtenus pour différentes valeurs de ϵ . Ils concernent la réalisation du premier placement élémentaire productible à

l'aide des formes référencées dans l'encours de placement EP , et pour un ensemble de connexité limité aux concaténations par le dessus et le dessous: $Connexité_Admises = \{0,2\}$.

1 Qualité des placements

Dans un premier temps, les courbes de l'évolution du rendement et des pertes surfaciques sont présentées. Nous tenons ici à présenter les deux critères d'efficacité, car ils sont parfois légèrement décorrélés en raison du paramètre longueur qui intervient. Ils confirment la tendance d'une qualité supérieure lorsque ϵ décroît

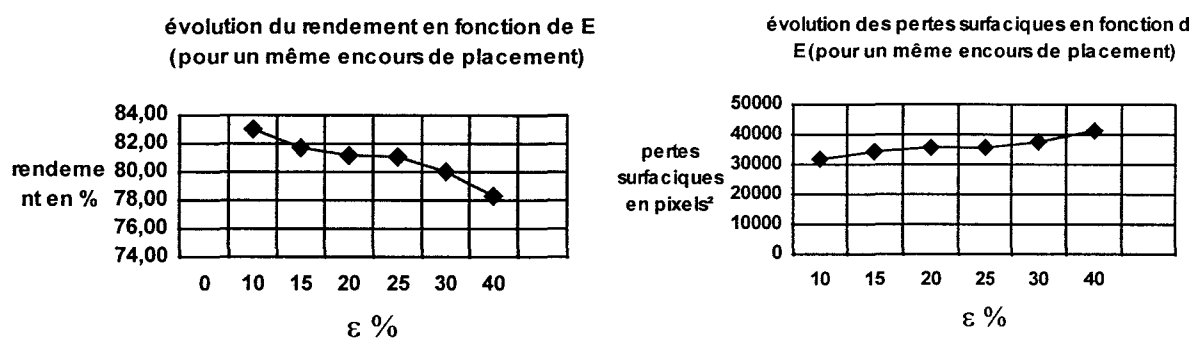


Figure 22

2) Vitesse d'obtention

Les courbes suivantes montrent l'évolution du nombre d'itérations nécessaires à l'obtention de la solution finale en fonction de l'écart relatif par rapport à la solution admissible. Ces indications renseignent évidemment sur le temps de calcul, puisque ce dernier est quasiment proportionnel au nombre de noeuds explicités.

L'évolution subit une brusque variation au passage de $\epsilon=10\%$ à $\epsilon=15\%$. C'est le prix à payer pour obtenir une solution légèrement meilleure. Cette transition brutale du nombre d'essais pour l'obtention de la solution peut être attribué à des irrégularités du paysage de coût, lequel dépend de la constitution de l'encours de coupe. Pour notre exemple, pour ϵ au-delà de 10%, les temps de calcul sont comparables.

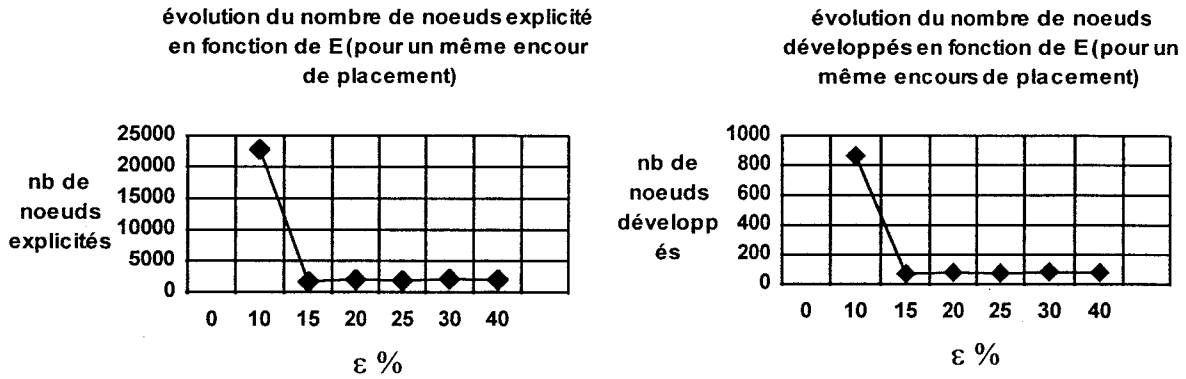


Figure 23

3) Efficacité

Enfin, les courbes suivantes présentent l'efficacité de la recherche entre la solution initiale et la solution finale, en fonction du paramètre ϵ . Lorsque ϵ est faible, la recherche est longue et les vitesses de variation du rendement sont basses. De même, pour une valeur élevée de ϵ , la recherche se limite souvent à la première solution trouvée c'est-à-dire à la solution initiale; la variation de rendement devient nulle et donc la vitesses de croissance aussi. Comme pour le recuit simulé, la vitesse de croissance du rendement passe par un maximum. Ce maximum apparaît ici pour $\epsilon = 15\%$, valeur déjà favorable pour la complexité et le rendement. Mais cette valeur n'est pas significative et correspond a priori à l'exemple-type. Une modification de l'encours de placement ou des paramètres de l'algorithme (fonctions $h(u)$, $f_C(u)$) déplace sensiblement ce maximum. De cette analyse, on peut cependant dégager une tendance générale, et pour des paramètres constants et un carnet de commande ordinaire, il est possible de renouveler l'étude pour en déduire le réglage optimal.

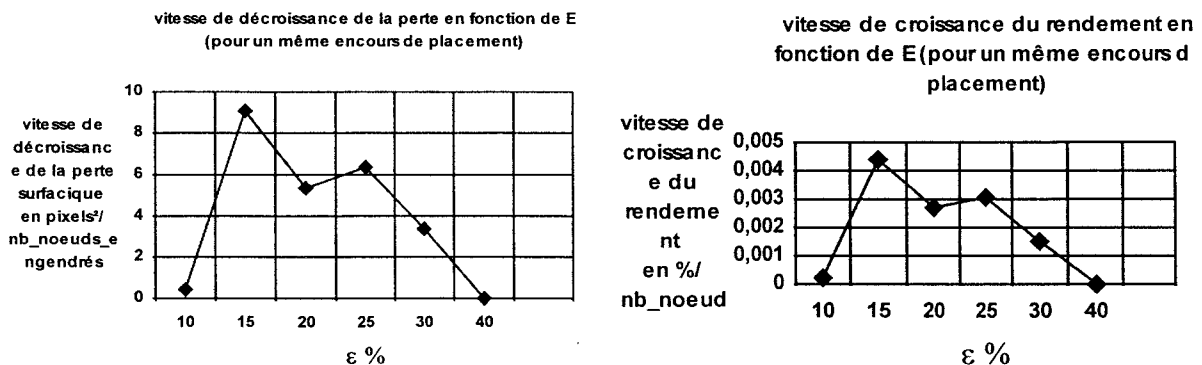


Figure 24

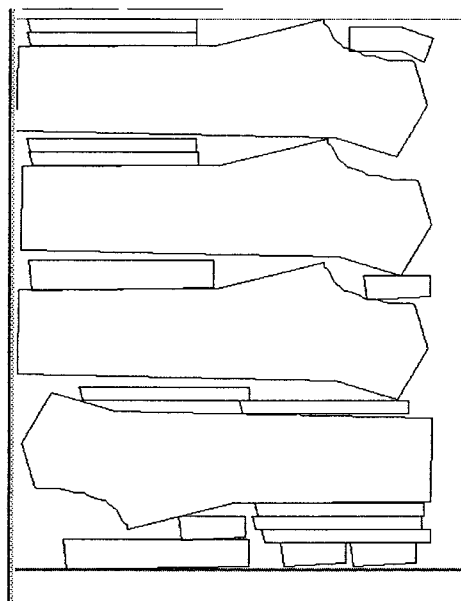


Figure 25: Placement réalisé par $A\epsilon$ avec $\epsilon = 15\%$ - rendement = 81,07% - nb de noeuds engendrés = 1715

VII Comparaison des deux algorithmes d'optimisation

Dans cette partie, nous proposons à titre de comparaison, une analyse montrant les différences séparant les deux algorithmes que nous avons testés. En supposant que le carnet de commande soit identique dans les deux cas, cette comparaison se trouve facilitée par le fait que l'espace de représentation des solutions est semblable.

7.1 Analyse des résultats

Si ces deux méthodes élaborent a priori de bonnes solutions, les principes de recherche sont cependant fondamentalement différents et les résultats comparés sur la base d'un même temps de recherche sont également nuancés. En représentant sur une même courbe l'évolution du rendement en fonction du nombre de noeuds engendrés pour obtenir les différentes solutions, on constate alors que la méthode du recuit simulé se révèle moins efficace que l'algorithme $A\epsilon$.

comparaison de l'évolution du rendement
selon les deux méthodes d'optimisation
(pour un même encours de placement)

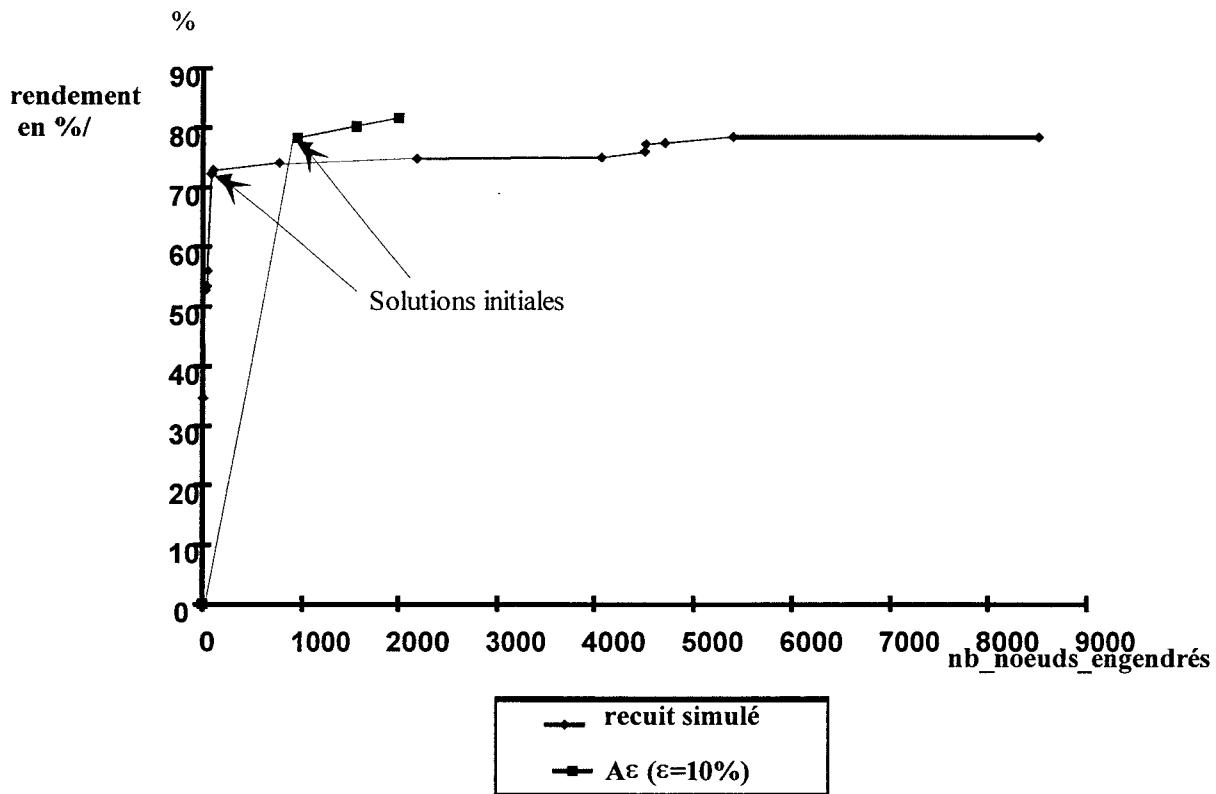


Figure 26

On remarquera en effet sur la *figure 26*, que pour un nombre de noeuds engendrés 4 fois supérieur, la méthode du recuit simulé atteint un rendement en deçà de celui obtenu pour $A\epsilon$. Une autre comparaison, effectuée à partir d'un carnet de commande comportant une variété plus large dans la géométrie des formes, confirme cette tendance de façon encore plus sensible. Nous avons donc cherché à expliquer ces différences.

7.2 analyse des trajectoires de parcours de graphe

7.2.1 Le recuit simulé

Dans le recuit simulé, on ne retient que les noeuds faisant partie du chemin compris entre la racine u_0 de l'arbre (aucune forme placée) et le noeud courant visité. En observant le déroulement de la recherche à travers l'écran de visualisation, on a pu remarquer, à température élevée, qu'un nombre important de bonnes configurations intermédiaires sont construites, mais souvent démontées lorsque, par le jeu du hasard, plusieurs remontées successives dans l'arbre sont acceptées. Ceci amène un double constat. D'une part que tout le

temps qui a servi à trouver une solution qui paraissait a priori prometteuse a été perdu par une séquence aléatoire malheureuse, et que l'absence de mémorisation des ces solutions ne permettra pas d'y revenir de façon directe. On doit ici remettre en cause la représentation sous forme d'arbre. Une modification locale, c'est à dire le choix d'un autre noeud intermédiaire, n'est possible qu'après avoir coupé la partie aval de la branche en ce noeud, laquelle ne sera peut être jamais reconstruite!. La représentation sous forme d'arbre ne permet pas la modification locale portant sur une seule pièce sans modifier le sous ensemble de formes placées après.

A l'inverse, on remarque parfois, toujours à température élevée, que malgré un mauvais choix des noeuds de bas rang, il arrive que l'algorithme continue obstinément à développer une même branche peu prometteuse en essayant de trouver les meilleurs de ses feuilles. Concrètement, les dernières formes placées dans la bande peuvent être bien positionnées, alors que les premières génèrent une chute importante non récupérable, en particulier si les mises en contact ne sont possibles que dans la direction transversale. Là encore, cette perte de temps pourrait être mise à profit pour explorer d'autres régions du graphe. On regrettera donc une fouille anarchique au début de la recherche, qui par principe de l'algorithme est non informée. La trajectoire est donc non dirigée, et les régions visitées de l'arbre sont réellement aléatoires.

Lorsque la température de contrôle baisse, la probabilité d'acceptation des remontées devient plus faible. C'est à ce moment que le système choisit sa région de travail dans l'arbre de recherche. Le rang moyen de visite des noeuds augmente. Hélas, il arrive que le choix de la région ne soit pas heureux. C'est le cas par exemple lorsque, au début du placement, une petite forme mal orientée précède une grande forme. Il faudra attendre une séquence aléatoire favorable pour que toutes les formes suivantes soient supprimées (remontées dans l'arbre), et que la suppression de la grande forme soit acceptée (plus difficile lorsque la température diminue), et enfin que la petite forme soit également enlevée. A l'évidence, cela peut être long pour voir se réaliser tous ces hypothétiques événements. On peut ici douter du choix de la fonction de but, qui ne différencie pas deux configurations contenant les mêmes formes mais placées différemment, la chute étant identique dans les deux cas. La fonction de coût devrait donc nécessairement contenir un membre supplémentaire tenant compte par exemple du rendement associé au rectangle exinscrit des seules formes placées. On rejoint alors les principes adoptés par [PRE 83] et [DAG 90] concernant la fonction de but à membres multiples.

D'autre part, lorsque le carnet de commande contient majoritairement des petites pièces, la probabilité de choix de celles-ci est donc plus élevée. Les expérimentations successives montrant qu'une bonne solution finale contient un mélange de grandes et de

petites formes, il va de soit que cette situation est atteinte après un nombre important de configurations aléatoires successives.

On pourra regretter l'absence de certaines commandes de changement de configurations telles que la permutation de deux formes. On peut en effet penser que dans certains cas, cela offrirait des changements directs de configurations locales pour atteindre des situations plus adaptées.

Si cette méthode converge en théorie vers l'optimum globale, l'algorithme simulé biaise singulièrement cette propriété dès lors que l'espace de représentation, la fonction de coût ou le générateur de configuration ne sont pas adéquates. Quant à la température initiale, le placement initial, la loi de décroissance de la température ou encore le nombre de configurations par palier, ces paramètres n'agissent essentiellement, à condition de respecter les règles établies, que sur la vitesse de convergence.

7.2.2 L'algorithme A_ϵ

A l'inverse du recuit simulé, l'algorithme A_ϵ tire profit des informations dont il dispose pour connaître et choisir les chemins à explorer en priorité. La difficulté essentielle pour la mise au point de cet algorithme est sans conteste l'élaboration des différentes heuristiques utilisées pour estimer d'une part les coûts entre le noeud courant (placement intermédiaire) et le meilleur noeud terminal du sous graphe descendant (bande finale corrélative), d'autre part l'estimation de la longueur du chemin pour arriver à ce noeud terminal et enfin l'heuristique de choix dans les ensembles de placements potentiels. On se propose ici d'analyser l'impact de ces derniers sur l'efficacité de l'algorithme A_ϵ .

a) L'heuristique d'estimation h

Au départ, u_0 est développé. Tout ses fils se voient affectés une estimation de la chute minimale à l'arrivée. Si l'estimation n'est pas très fine, l'ordre des noeuds dans la liste NDP n'est pas le même que l'ordre réel que l'on aurait si l'on connaissait les descendants terminaux avec les coûts réels. Le choix d'un nouveau noeud à développer peut être entaché d'une erreur de jugement a priori et amener l'algorithme à explorer une branche qui plus tard sera remise en question. En fait, on n'est jamais sûr d'avoir le bon ordre, mais plus l'estimation est fine, plus l'ordre est correct. C'est pour cela, qu'au niveau des essais, nous avons vérifié que les coûts réels et ceux estimés à l'arrivée d'un noeud terminal étaient cohérents au niveau de l'ordre ($f(t1) < f(t2) \Leftrightarrow g(t1) < g(t2)$) et de la minoration ($f(t1) \leq g(t1)$). Malheureusement, rien ne prouve qu'il en était de même au niveau des noeuds ascendants.

Certes, ce type d'algorithme permet de corriger cette erreur de jugement lorsque, NDP ne contenant que des noeuds de rang élevé, le coût réel $g(u)$ devient prépondérant,

l'erreur s'atténue et l'ordre s'établit convenablement. Il faudra bien sûr que toutes les branches soient suffisamment développées pour atteindre cette situation, impliquant un temps de recherche prohibitif et un nombre de noeuds engendrés que l'espace mémoire ne pourra pas loger. On rejoint alors le schéma d'une recherche à *coût uniforme* (sans estimation) avec une perte de temps supplémentaire.

L'autre inconvénient d'une mauvaise estimation h concerne l'élagage de l'arbre. Lorsque celui-ci jouera son office, les noeuds placés en bout de liste NDP seront supprimés alors qu'ils étaient peut-être prometteurs.

Concernant la fonction d'estimation que nous avons choisie, nous ne pouvons pas assurer qu'elle conserve l'ordre total entre les rangs. Tout au plus, on a réglé (empiriquement) les coefficients qui participent à la fonction d'estimation de sorte que l'estimation soit minorante, propriété nécessaire pour garantir l' ϵ -admissibilité. Mais les résultats pratiques nous ont permis de nous satisfaire du choix.

Les remarques précédentes nous amènent à penser, d'une façon générale, qu'une méthodologie à appliquer pour modéliser la fonction d'estimation nous a beaucoup manquée. On peut espérer qu'en garantissant un ensemble de propriétés nécessaires à $A\epsilon$ et moyennant la construction d'une structure d'estimation auto-adaptative, la fonction h serait plus fine et donc l'efficacité accrue.

b) l'heuristique d'estimation hc

La fonction hc n'a aucune influence sur le rendement du résultat final en fin de recherche. En revanche, elle est d'une importance capitale au regard du temps de recherche. C'est d'elle dont dépend le choix des noeuds pendants dans l'ensemble NDP en cas de retour arrière dans l'arbre, et de façon moins significative, lors d'un choix dans l'ensemble des successeurs acceptables. Le choix du modèle de calcul de $hc(u)$ présente une difficulté majeure qui est d'estimer la longueur du chemin minimum au rang les plus faibles. Selon notre méthode, (décompte des surfaces des plus grandes formes logeables pour annuler le remplissage estimé) le problème se pose lorsque, au début de la recherche, les noeuds se voient presque tous affecter la même valeur de hc : le remplissage estimé étant important, le nombre de pièces logeables est souvent le même selon que l'on place une grande ou une petite forme en premier. Une amélioration possible consisterait en fait à considérer qu'on ne placera plus de formes de surfaces supérieures à celle qui vient d'être placée. Ainsi, dès le début, une petite forme sera succédée que par des petites formes, et le compteur final sera donc plus élevé que pour une pièce initiale de grande taille. La logique qui consiste à penser que le placement des grandes pièces accélère le remplissage est donc respecté.

c) la fonction de choix f_c

Le modèle choisi pour cette fonction est non linéaire. Elle attribue une importance décroissante à l'estimation de la longueur du chemin au fur et à mesure que le nombre de solutions proposées devient important. Mais le coefficient constant intervenant dans le poids affecté à hc a été ici déterminé expérimentalement. Celui-ci a été fixé de sorte que le membre associé à hc soit deux fois plus important en début de recherche (aucune solution). Il convient pour une certaine valeur de $f(u_0)$, pour les exemples traités. Il est donc raisonnable de penser que ce réglage ne correspond pas à tous les types de situations. Au lieu d'une pondération absolue, on peut imaginer une pondération μ , solution de l'équation:

$$2 * f(u_0) = \mu * hc(u_0) / \ln(2 + \text{card}(\text{Solutions})),$$

avec $\text{card}(\text{Solutions}) = 0$ en début de recherche.

Ainsi, quelle que soit la nature du problème on attachera une importance double à la faible longueur du chemin plutôt qu'à la perte estimée. On développe donc en priorité les chemins moins complexes.

VIII Conclusion

Comme en témoignent les résultats précédents, l'algorithme $A\epsilon$ constitue un très bon compromis admissibilité/complexité. Il présente de plus l'avantage d'être ajustable sur le plan de l'objectif à atteindre. Cette propriété est indispensable face à la flexibilité attendue de notre module de placement, et à ce titre devient un outil de choix. Cependant il présuppose une connaissance approfondie du problème traité, en l'occurrence pour établir la fonction d'estimation du coût d'un chemin. Les résultats obtenus sont encourageants, mais quelques améliorations sont encore envisageables. Pour un choix judicieux des paramètres internes aux fonctions h et f_c et de l'écart toléré ϵ , les résultats sont meilleurs que pour le recuit simulé en temps et en rendement, cette dernière méthode souffrant de quelques inconvénients pénalisant l'efficacité de la recherche. Globalement, l'avantage va au profit de $A\epsilon$ qui en plus a le mérite d'être constant dans son efficacité, à l'inverse du recuit simulé qui par principe est tributaire du hasard.

Conclusion Générale

Nous avons essayé à travers ce mémoire, de rassembler les connaissances ainsi que les réflexions qui nous ont permis de mettre en oeuvre un logiciel d'optimisation du placement de formes irrégulières. Certes, le logiciel mis au point, n'est qu'une maquette d'un produit industriel, un prototype qu'il s'agirait d'adapter à l'environnement matériel dans lequel il évoluerait. En fait, nous avons préféré concentrer nos efforts sur les principes algorithmiques et les méthodes d'optimisation plutôt que de définir un produit dédié à un système existant de découpe. C'est pourquoi le logiciel développé a été écrit dans un langage qui, par ses facultés interactives et la réutilisabilité des composants logiciels prédéfinis, favorise la mise au point de méthodes algorithmiques: SMALLTALK. Ses performances moyennes en vitesse de traitement ne représentèrent qu'un handicap relatif face à l'aisance qu'il procurait pour comparer les méthodes algorithmiques que nous avons testées. D'ailleurs, on a comparé les résultats sur la base du nombre d'opérations ou de noeuds engendrés dans l'espace de représentation pour une efficacité donnée, plutôt qu'à travers une mesure du temps perdant toute objectivité dès lors que le calculateur ou le compilateur change.

Cependant, l'utilisation de notre algorithme de placement à des fins industrielles demande peu d'adaptations. En effet, nous nous sommes attachés à ce que ses fonctionnalités soient proches du cahier des charges du système de placement que nous nous étions définis dans le chapitre I. Naturellement, on peut espérer fermement accroître les performances temporelles de l'algorithme en programmant cette application dans un langage compilé tel que le langage C.

A travers l'analyse des systèmes d'optimisation du placement existant, on a souvent pu constater que la modélisation des formes était souvent considérée comme un détail du problème et constituait rarement une préoccupation pourtant essentielle. Par ailleurs, notre approche bifocale de l'optimisation nous a conduit à poser le problème de la description géométrique des contours des formes mis en jeu. C'est pourquoi nous avons ici consacré deux chapitres pour présenter ce que nous ressentions comme une lacune à propos du placement: la représentation des formes irrégulières et la construction du placement. Le concept des *peignes de contours* nous a offert la possibilité de décrire précisément les formes selon quatre vues, tout en préservant les propriétés intéressantes du rectangle vis à vis des contraintes du placement paraxial. Le résultat est donc favorable en terme de complexité lors du processus de mise en imbrication. D'autre part, la concaténation unissant les contours des deux formes a permis de voir la réalisation d'un placement comme une construction progressive d'une forme

ayant la même définition et les mêmes propriétés que ses éléments constitutifs. Cette méthode de travail facilite l'évaluation des caractéristiques du placement grâce à la richesse des informations dont on dispose à toutes les étapes de la construction.

Les peignes de contours sont, on l'a vu, bien adaptés à la spécificité du placement paraxial sur feuille de matériau plan et rectangulaire. On peut également croire que, moyennant quelques aménagements des séquences d'imbrications et peut-être aussi au niveau de la définition, qu'il est envisageable de les utiliser pour le placement sur feuille de matériau non rectangulaire. La description de l'intérieur du contour de la matière serait assurée par une hiérarchie de peignes codant toutes les portions de contours même s'il sont à visibilité réduite. L'évaluation de l'imbrication d'une forme avec la matière se calculerait selon les mêmes méthodes.

Il serait aussi sans doute intéressant d'étendre le principe de description par peignes de contours au cas de la déficience circulaire, de façon à s'affranchir des difficultés liées au développement de l'opération de rotation. Cela ouvrirait la porte au placement constructif sur peaux, où les formes à découper peuvent prendre des orientations quelconques. L'opération de rotation d'angle quelconque permettrait par ailleurs de biaiser légèrement la contrainte du droit fil au niveau du placement textile.

Dans ce mémoire, il est fait abstraction de la résolution de la contrainte d'évitement des défauts de la matière. Cependant nous avons mené dans [ROU 93.c] une réflexion visant à introduire la prise en compte des défauts au niveau de l'imbrication locale. Sans modification supplémentaire des opérateurs qui sont présentés dans ce rapport, nous réussissons à assurer l'évitement des zones de défauts en déterminant, comme forme initiale, la forme issue de la concaténation de l'origine de la bande et des tous les contours des défauts. Ensuite, les formes à produire sont simplement concaténées à la forme initiale, et l'algorithme d'optimisation se déroule à l'instar de celui utilisé pour le placement sur tissu uni.

La cellule de découpe qui nous a servi de support, doit travailler en "juste à temps", c'est-à-dire respecter les délais fixés au moment de la commande. La production doit donc s'adapter au rythme de la demande. D'autre part, le module de placement doit être en mesure de sortir les plans de coupe à fournir à l'atelier d'assemblage, de sorte que l'encours de coupe soit minimal pour éviter les coûts de stockage, mais non nul pour assurer un travail constant de la table de découpe. Le volume de commandes variant dans le temps, nous avons donc cherché à garantir au mieux la flexibilité de la production. Mais, dans un problème hautement combinatoire, la qualité de la solution est souvent tributaire du temps de recherche. Les rendements des bandes de placement dépendent alors du temps qui est laissé à l'algorithme pour fournir le plan de coupe. Il nous fallait donc pouvoir régler les temps d'obtention des placements sans trop dégrader les rendements matière. Le rapport temps/rendement est un compromis qu'il fallait maîtriser. Les résultats que nous avons obtenus avec l'algorithme du recuit simulé d'une part, et la recherche en graphe avec Aε d'autre part sont encourageants. Ils

ont l'avantage de converger vers une solution, de façon plus ou moins rapide selon les valeurs données à certains paramètres de contrôle.

Notre jugement donne une préférence à l'algorithme $A\epsilon$ qui, malgré quelques difficultés de mise en oeuvre, offre par ailleurs une évaluation de la distance à l'optimum global, et donc un réglage plus facile du paramètre de contrôle ϵ . Ici, ce réglage est cependant effectué de façon empirique. La valeur affichée de ϵ dépend de l'opérateur, qui estime le rythme que doit prendre l'algorithme de placement. Ceci est relativement pointu et nécessite une surveillance constante lorsque le carnet de commandes varie très fortement. Améliorer le contrôle du précédent compromis consisterait, par exemple à utiliser la distance à l'optimum comme retour d'information et à se servir de cette donnée dans une boucle d'asservissement du processus d'optimisation. On y afficherait alors le temps d'optimisation comme consigne, laquelle serait calculée à partir de la constitution du carnet de commandes.

Il est vrai que les résultats obtenus avec les algorithmes $A\epsilon$ ou du recuit simulé demande encore quelques aménagements au niveau du logiciel pour diminuer les temps de recherche. En particulier, une session comportant un bon nombre de modèles différents engendre une augmentation de la largeur de l'arbre des solutions. On serait par exemple amené à grouper les formes "semblables" afin de diminuer la variété. On voit alors apparaître un problème de classement des formes selon leurs ressemblances géométriques.

Réduire les temps de calcul est un lot commun à tous les développements théoriques et pratiques en matière industrielle. Même si les calculateurs voient leurs limites en rapidité sans cesse reculer, il est pourtant certain que les problèmes combinatoires NP-Complet resteront, dans l'état actuel des connaissances, de complexité exponentielle et donc "non-admissibles" au sens de l'optimisation en un temps raisonnable. Tous les moyens sont donc louables dès lors qu'ils augmentent un peu les chances d'obtenir la meilleure ou tout au moins une bonne solution en un temps acceptable. Nous avons, par notre méthode d'élagage de graphe, rallié cette politique. Mais nos multiples expérimentations nous ont montré deux choses:

- d'une part que beaucoup de combinaisons infructueuses pourraient être évitées. La mémorisation des situations a posteriori illogiques, le réglage adaptatifs de certains paramètres, la précision de l'estimation des coûts permettraient, on l'imagine, d'améliorer très sensiblement la qualité ou l'efficacité de l'algorithme.

- d'autre part que nos propres observations nous aiguillaient dans les réglages que nous avons effectués ainsi que dans les restrictions que nous avons amenées dans les choix de configurations dont l'impact sur la qualité finale était très limité.

Nous venons ici simplement de pressentir les améliorations qu'offrirait l'apprentissage au sein d'un algorithme d'optimisation.

-
- [AAR 85]: E.H.L. AARTS, P.M.J. LAARHOVEN, Proc. Int. Conf. on C.A.D., Santa Clara, November 1985.
- [ADA 69] : M. ADAMOWICZ, "The optimal two-dimensional allocation of irregular, multiply-connected shapes with linear, logical and geometric constraints", Ph D thesis, Dept of E-E, N.Y.U. (Dec 1969)
- [ADA 70.a] : M. ADAMOWICZ and A. ALBANO, "An approach to the solution of the template layout problem", a preliminary report, consiglio Nazionale delle ricerche, istituto di elaborazione della informazione, Pisa, Italy, nota interna, B 70/12, may 1970
- [ADA 70.b] : M. ADAMOWICZ and A. ALBANO, "A solution of the rectangular layout problem", a preliminary report, consiglio Nazionale delle ricerche, istituto di elaborazione della informazione, Pisa, Italy, nota interna, B 70/12, may 1970
- [ADA 70.c] : M. ADAMOWICZ and A. ALBANO, "The clustering of two dimensionnal shapes", a preliminary report, consiglio Nazionale delle ricerche, istituto di elaborazione della informazione, Pisa, Italy, nota interna, B 70/12, may 1970
- [ADA 71.a] : M. ADAMOWICZ and A. ALBANO, "A two-stage solution of the cutting-stock problem", Information processing 71, North Holland publishing company (Proc IFIP congress 71), Amsterdam (1972) pp 1086-1091
- [ADA 71.b] : M. ADAMOWICZ, "Optimal allocation of two-dimensional shapes", in Proc Int'l Design, Automation Conf, Toronto, Canada, 1971
- [ADA 76.a] : M. ADAMOWICZ and A. ALBANO, "Nesting two dimensional shapes in rectangular modules", Computer Aided Design, vol 8, N°1, january 1976
- [ADA 76.b] : M. ADAMOWICZ and A. ALBANO, "A solution of the rectangular cutting stock problem", IEEE Trans. on Syst. Scien. and Cyber, SMC-6, pp 302-310, 1976
- [ALB 77]: A. ALBANO, "A method to improve two dimensional layout", Computer Aided Design, vol 9, N°1, january 77

-
- [ALB 80] : A. ALBANO and G. SAPUPPO, "Optimal allocation of two-dimensional irregular shapes using heuristic search method", IEEE Trans. on Syst., Man and Cybern., volSMC-10, N° 5, may 1980
- [AME 89] : H. AMET et R. SCHOTT, "Déplacement exact d'un polygone au sein d'un ensemble de polygones", Université de Nancy 1, campus scientifique, CRIN, BP 239, 54 506 Vandoeuvre lès Nancy, cedex, 1989
- [ART 66] : R.C. ART, "An approach to the two-dimensional irregular cutting stock problem", IBM Cambridge Scientific Center, Cambridge, MA, Rep N° 320-2006, 1966
- [AVN 88.a] : F. AVNAIM and J.D. BOISSONNAT, "A practical exact motion-planning algorithm for polygonal objects amidst polygonal obstacles", rapport de recherche I.N.R.I.A. , Aug 1988, France
- [AVN 88.b] : F. AVNAIM and J.D. BOISSONNAT, "Polygon placement under translation and rotation", rapport de recherche I.N.R.I.A. n°889, Aug 1988, France
- [BAL 81]: BALLARD D.H., "Strip trees: A hierarchical representation for curves", comm ACM 24, 5, mai 1981, pp 310-321
- [BAR 88]: BARUCH O., LOEW M.H., "Segmentation of two dimensional boundaries using the chain code", Pattern Recognition, 21(6), 1988, pp 581-589
- [BEN 75]: BENNET J.R., MAC DONALD J.S., "On the measurement of curvature in a quantized environment", IEEE Trans. on Computers, 24(8), august 1975, pp 803-820
- [BER 79]: BERTHOD M. et JANCENNE P., "Le traitement des tracés manuscrits sur une table graphique, actes du 2ème congrès AFCET RFIA, volumes 2, pp 195-209, France, Toulouse, 1979
- [BOB 89] : H. LAPLANTE, "Volume doesn't matter with automation", Bobbin, june 1989, pp 76 - 79
- [BON 87]: C. BONNEMOY, "Sur quelques aspects de l'utilisation de méthodes déterministes en milieu stochastiques et inversement", Thèse d'état, Université Clermont II, septembre 1987.

-
- [BON 90]: P. BONNET, Cours de traitement d'image, Université des Sciences et des Technologies de Lille, 1990
- [BON 91]: C. BONNEMOY, S. B. HAMMA, "La méthode du recuit simulé: optimisation globale dans R^n ", APII, 1991-25, pp 477-496
- [BRA 87]: BRASSARD G., BRATLEY P. Algorithmique: conception et analyse, Ed. Masson, Paris, 1987.
- [BRI 79]: BRIBIESCA E. and GUZMAN A., "How to describe pure form and how to measure differences in shapes using shape numbers", Proc., PRIP, august 1979, pp 427-436
- [BUR 84]: M BURSTEIN and M.N. YOUSSEF, "Timing influence layout design", Proceeding: 22th Design Automation Conference, pp 124-130, june 1985
- [CAO 92]: X. B. CAO, "Contribution à l'élaboration des systèmes d'aide à la conception à base de connaissances ayant des facultés d'apprentissage. Application à la confection" Thèse d'automatique de 3ème cycle, Université de Valenciennes, septembre 1992
- [CAT 76]: A. CASTANI, C. CAVAGNA, U. CUGINI and P. MORO, "A computer aided design system for pattern grading and marking in the garment industry", CAD 1976 proc., London, pp 159-165, 1976
- [CER 82]: V. CERNY, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm, J. of Optimization Theory and Applications N°45, p41., 1985 (accepté pour publication en 1982).
- [CHA 91]: CHASSERY J.M., MONTANVERT A., Géométrie discrète, Hermes, Traité des Nouvelles Technologies, Paris, 1991.
- [CHO 79]: W.W. CHOW, "Nesting of a single shape on a strip", Int. Journ. Prod. Res., vol 17, N°4, pp 305-322, 1979
- [CLE 87]: T. CLERC et M. SAGDAL, "Prototype d'un système de placement intelligent dans le plan", MICAD 87, pp 91-103

-
- [COF XX] : E.G. COFFMAN, M.R. GAREY and D.S. JOHNSON., "Approximation algorithms for bin-packing: an updated survey", communication privée.
- [CON 91] : "Atelier de coupe : le parent pauvre de l'organisation" CONFECTION 2000, N° 121, Avril 1991, pp 74- 75
- [DAG 88] : C.H. DAGLI, "Knowledge-based systems for cutting stock problems", Department of Engineering Management University of Missouri-Rolla, working paper series 88-01-01, submitted to the European Journal of Operational Research, july 1988
- [DAG 90]: C.H. DAGLI and A. HAJAKBARI, "A parallel architecture for stock cutting problem", IEEE International conference on PARBASE-90 : database, parallel architecture and their applications,
- [DEB 78]: De BOOR C., A practical guide to Splines, New York, Springer-Verlag, 1978
- [DEL 89] : J.L. DELAPORTE, "Intégration des fonctions de conception et de préparation de la fabrication pour les entreprises de découpe", Thèse de doctorat de 3ème cycle, Université de Valenciennes et du Hainaut Cambrasis, 1989
- [DIV 73] : J.C. DIVERREZ, B. Du Bois de MEYRIGNAC, R. SOENEN, "Recherche d'un classement hierarchique de surfaces planes et application à un probleme de placement selon une méthode heuristique, revue RAIRO, novembre 1973, J-3, pp 93- 106
- [DMI 91] : B. DMITROWICZ, D WOLF, R. HUSSON, "Une approche méthodologique de l'optimisation de découpe dans l'industrie du bois", 23ème CIRP, Séminaire Internatinal sur les systèmes de production, Nancy, 6-7 Juin 1991, Tome II
- [DOD 91] : M. DODONA, "Contribution à la détection de motifs complexes d'orientation quelconques dans les images. Mise en oeuvre sur un système multiprocesseur", Thèse de l'Institut National Polytechnique de Toulouse, Novembre 1991.
- [DOM 88]: C.H. DOMINE, Techniques de l'Intelligence Artificielle: un Guide Structuré, BO-Pré, Dunod Informatique, 1988
- [DOR 84]: DORST L., SMEULDERS W.M., "Discrete representation of straight lines", IEEE trans. on PAMI, 6(4), 1984, pp 450-463

-
- [DRA 86] : A. DRAOU, " Outils d'aide au placement sous contraintes. Application à la confection", thèse de doctorat, spécialité en Automatique et traitement du signal, Université de Valenciennes et du Hainaut-Cambrasis, 1986
- [DUB 88] : P. DUBOIS, "L'industrie de l'habillement : l'invasion face à la crise", La documentation Française, notes et études documentaires, 1988, pp 69 - 88.
- [DUN 84] : DUNLOP, AGRAWAL, DEUTSCH, JUKL, KOSAK and WIESEL, "Chip layout optimization using critical path weighting, Proceeding: 21th Design Automation Conference, pp 133-136, june 1984
- [ELO 92] : A. ELOMRI, "Méthode d'optimisation dans un contexte productique", thèse de 3ème cycle, Laboratoire d'Automatique et Productique, Université de Bordeaux I
- [FAR 87]: H. FAREENY, M. GHALLAB, Eléments d'Intelligence Artificielle, Hermes Publishing, 1987.
- [FAR 88] : A.A. FARLEY, "Mathematical programming models for cutting stock problems in the clothing industry", J. Op. Res. Soc. , vol 39, N°1, pp 41-53, 1988
- [FRE 61]: H.FREEMAN , "On the encoding of arbitrary géométric configurations", Computer Methods in Image Analysis, Aggrawal & al. ed., IEEE Press, 1977. Reprinted from IRE Trans. Electron Compu, 10, june 1961, pp 260-268
- [FRE 74.a] : H. FREEMAN, "On the packing of arbitrary-shaped templates", division of applied science, New York University, internal report, 1974
- [FRE 74.b]: H.FREEMAN , "Computer processing of line-drawing images", Computer Surveys, 6(1), March 1974, pp 57-97
- [FRE 75]: H.FREEMAN , R.SHAPIRA , "Determining the minimum area encasing rectangle for an arbitrary closed curve", Comm of ACM, 18(7), 1975, pp 409-413
- [GAR 81] : M.R. GAREY and D.S JOHNSON, "Approximation algorithms for bin-packing problems: a survey", in Analysis and Design of Dlgorithms in Dombinatorial Dptimization, Ausiello G and Lucertini M., 1981

-
- [COF XX] : E.G. COFFMAN, M.R. GAREY and D.S. JOHNSON:, "Approximation algorithms for bin-packing: an updated survey", communication privée.
- [CON 91] : "Atelier de coupe : le parent pauvre de l'organisation" CONFECTION 2000, N° 121, Avril 1991, pp 74- 75
- [DAG 88] : C.H. DAGLI, "Knowledge-based systems for cutting stock problems", Department of Engineering Management University of Missouri-Rolla, working paper series 88-01-01, submitted to the European Journal of Operational Research, july 1988
- [DAG 90]: C.H. DAGLI and A. HAJAKBARI, "A parallel architecture for stock cutting problem", IEEE International conference on PARBASE-90 : database, parallel architecture and their applications,
- [DEB 78]: De BOOR C., A practical guide to Splines, New York, Springer-Verlag, 1978
- [DEL 89] : J.L. DELAPORTE, "Intégration des fonctions de conception et de préparation de la fabrication pour les entreprises de découpe", Thèse de doctorat de 3ème cycle, Université de Valenciennes et du Hainaut Cambrais, 1989
- [DIV 73] : J.C. DIVERREZ, B. Du Bois de MEYRIGNAC, R. SOENEN, "Recherche d'un classement hierarchique de surfaces planes et application à un probleme de placement selon une méthode heuristique, revue RAIRO, novembre 1973, J-3, pp 93- 106
- [DMI 91] : B. DMITROWICZ, D WOLF, R. HUSSON, "Une approche méthodologique de l'optimisation de découpe dans l'industrie du bois", 23ème CIRP, Séminaire Internatinal sur les systèmes de production, Nancy, 6-7 Juin 1991, Tome II
- [DOD 91] : M. DODONA, "Contribution à la détection de motifs complexes d'orientation quelconques dans les images. Mise en oeuvre sur un système multiprocesseur", Thèse de l'Institut National Polytechnique de Toulouse, Novembre 1991.
- [DOM 88]: C.H. DOMINE, Techniques de l'Intelligence Artificielle: un Guide Structuré, BO-Pré, Dunod Informatique, 1988
- [DOR 84]: DORST L., SMEULDERS W.M., "Discrete representation of straight lines", IEEE trans. on PAMI, 6(4), 1984, pp 450-463

-
- [DRA 86] : A. DRAOU, " Outils d'aide au placement sous contraintes. Application à la confection", thèse de doctorat, spécialité en Automatique et traitement du signal, Université de Valenciennes et du Hainaut-Cambrais, 1986
- [DUB 88] : P. DUBOIS, "L'industrie de l'habillement : l'invasion face à la crise", La documentation Française, notes et études documentaires, 1988, pp 69 - 88.
- [DUN 84] : DUNLOP, AGRAWAL, DEUTSCH, JUKL, KOSAK and WIESEL, "Chip layout optimization using critical path weighting, Proceeding: 21th Design Automation Conference, pp 133-136, june 1984
- [ELO 92] : A. ELOMRI , "Méthode d'optimisation dans un contexte productique", thèse de 3ème cycle, Laboratoire d'Automatique et Productique, Université de Bordeaux I
- [FAR 87]: H. FAREENY, M. GHALLAB, Eléments d'Intelligence Artificielle, Hermes Publishing, 1987.
- [FAR 88] : A.A. FARLEY, "Mathematical programming models for cutting stock problems in the clothing industry", J. Op. Res. Soc. , vol 39, N°1, pp 41-53, 1988
- [FRE 61]: H.FREEMAN , "On the encoding of arbitrary géométric configurations", Computer Methods in Image Analysis, Aggrawal & al. ed., IEEE Press, 1977. Reprinted from IRE Trans. Electron Compu, 10, june 1961, pp 260-268
- [FRE 74.a] : H. FREEMAN, "On the packing of arbitrary-shaped templates", division of applied science, New York University, internal report, 1974
- [FRE 74.b]: H.FREEMAN , "Computer processing of line-drawing images", Computer Surveys, 6(1), March 1974, pp 57-97
- [FRE 75]: H.FREEMAN , R.SHAPIRA , "Determining the minimum area encasing rectangle for an arbitrary closed curve", Comm of ACM, 18(7), 1975, pp 409-413
- [GAR 81] : M.R. GAREY and D.S JOHNSON, "Approximation algorithms for bin-packing problems: a survey", in Analysis and Design of Dlgorithms in Dombinatorial Dptimization, Ausiello G and Lucertini M., 1981

-
- [GUR 69] : O. GÜREL, "Circular graph of marker layout", IBM techn, report N° 320-2965, feb 1969
- [HAI 66] : M.J. HAIMS, "On the optimum two-dimensionnal allocation problem", Ph D dissertation, dept of elec enrg, New York University, Bronx, techn rept 400-136, june 1966
- [HAI 70] : M.J. HAIMS and H. FREEMAN, "A multistage solution of the template-layout problem", IEEE Trans. on Syst. Scien. and Cybern., vol ssc-6, n°2, April 1970
- [HIN 80] : A.I. HINXMAN, "The trim loss and assortment problems: a survey", European Journal of Operational Research, vol 5, pp 8 -18, 1980
- [HUA 75]: HUANG T.S., "Bounds on the bit rate of linéar run length codes", ITIT, nov 75
- [ISR 80] : S. ISRANI and J. SANDERS, "Two-dimensional cutting stock problem research: a review and a new rectangular layout algorithm", Journal of Manufacturing Systems, vol 1, n°2, pp 169-182, 1982
- [KIR 82]: S. KIRKPATRICK, C.D. GELATT, M.P. VECCHI, "Optimization by simulated annealing", Science, 220, pp 671-680, 1982.
- [KUN 81] : G. KUNTZ, J.P. UHRY, "Optimisation des pertes de matière en industrie textile", Actes du congrès de l'AFCET 1981, Edition Hommes et Techniques
- [KUNTZ XX] : G. KUNTZ, C. LACOTTE, F. PREMPTI and J.P. UHRY, "sur quelques aspects du placement automatisé", bulletin szpécial, placement et découpe, mission pour la conception assistée par ordinateur, ZIRST, chemin du pré-carré, 38240 MEYLAN
- [LAA 87]: P.M.J. LAARHOVEN and E.H.L. AARTS, D., Simulated Annealing: Theory and Appllications, Reidel Publishing Company, Holland, 1987
- [LAU 85]: LAURIOL, "L'organisation du travail et des fabrications dans les ateliers de coupes", , Edition Vauclair, Modes et Techniques, 1985.

-
- [Le BOR 82] : P. Le BORGNE et T. MARTIN MARTIN, "RIMBAUD: logiciel de Recherche d'IMBrication AUtomatique pour la Découpe", rapport de projet de fin d'étude, ENSIMAG, juin 1982.
- [LIU 91]: LIU S., THONNAT M., "Détermination du contour apparent d'un objet en vue de la description de sa forme", actes du 8ème congrès AFCET RFIA, Lyon, 25-29 novembre 1991, pp 1323-1334
- [LOR 90] : LORIOT-VESTRA, "Système de placement interactif sur tissu à motifs", brevet d'invention, 1990
- [LUX 85]: LUX A., Algorithmique et contrôle en vision par ordinateur, Thèse d'état, université scientifique et médicale de Grenoble, INPG, 1985
- [MAD 80] : O.B.J. MADSEN , "References concerning the cutting stock problem", working paper, IMSOR, The Technical University of Denmark, 1980
- [MAO 85] : S. MAOUCHE, P. VIDAL, CICHOWLAS, "Flexible cutting frame in real time", 1er C.I.N.T.T., Mulhouse 1985.
- [MAO 92] : S. MAOUCHE, "A continuous cutting system in the suit make up industry", XI-th International Conference on Systems Science, Wroclaw, Poland, sept 1992
- [MAR XX] : S. MARTELLO et P. TOTH, "Algorithms for knapsack problem", communication privée
- [MET 53]: N. METROPOLIS, A.W. ROSENBLUTH, A.H. TELLER, E. TELLER, "Equation of state calculations by fast computing machines, J. Chemical. and Physique-21,p 1087, 1953.
- [MON 68]: U.MONTANARI , "A method for obtaining skeletons using a quasi-euclidean distance", journal of ACM, 15, 1968, pp 600-624
- [MOR 69]: G.R. MOREAU and P.J. De SAINT HARDAVIN, "Marker layout problem: an experimental attempt", IBM New York, N.Y., Rep N° 320-2978, 1969
- [MOR 73]: G. MOREAU, "Méthodes pour la résolution des problèmes d'optimisation de découpe", Thèse de Docteur Ingénieur, Université Claude Bernard, Lyon, 1973

-
- [NIL 71]: N. NILSSON, Problem-Solving Methods in Artificial Intelligence, Mc Graw Hill, 1971.
- [NIL 82]: N. NILSSON, Principles of Artificial Intelligence, , Springer Verlag, 1982
- [OKA 91]: A. OKAT, "Contribution à la conduite d'une cellule de découpe au laser", Thèse de 3ème cycle, Université des Sciences et des technologies de Lille, Novembre 1991
- [PAV 74]: PAVLIDIS T. and HOROWITZ S.L., "Segmentation of plane curves", IEEE Trans. on Computer, 23 (8), august, 1974
- [PAV 77]: PAVLIDIS T., Structural Pattern Recognition, New York, Springer-Verlag, 1977
- [PEA 83]: J. PEARL, "Knowledge versus search: a quantitative analysis using A*", Artificial Intelligence, 20,1-13; 1983.
- [PEA 83]: J. PEARL, "Studies in semi-admissible heuristics, IEEE Trans. PAMI, 4,b4, 392-400, july 1982.
- [PER 78] : L. M. PEREIRA, "Artificial intelligence techniques in automatic layout design, Artificial Intelligence and Pattern recognition on computer aided design, North Holland, 1978, pp 159-173
- [PER 90]: M.A. PERKOWSKI, P. DYSKO, B. FALKOWSKI, "Two Learning Methods For a Tree-Search Combinatorial Optimizer, IEEE, PCCC 90, pp 606-613, 1990.
- [POH 77]: POHL I., "Practical and Theoretical considerations in heuristic search algorithms", Machine Intelligence,8, Elcock and Michie (EDS), Horwood, 55-71, 1977.
- [PRE 83] : F. PREMPTI, "Méthodes stochastiques dans les problèmes de placement", Thèse de 3ème cycle, Recherche Opérationnelle, Université scientifique et médicale de Grenoble, Juillet 1983
- [RAM 72]: RAMER V., "An iterative procedure for the polygonal approximation of planes curves. Computer Vision, Graphics and Image Processing, 1972

-
- [RAY 91]: RAY B. K. and RAY K.S., "A new approach to polygonal approximation", Pattern Recognition letters, 12, 1991, pp 229-234, North Holland
- [RED 88] : A. REDLINGER, "Contribution au problème du placement optimal. Algorithmes analytiques et combinatoires", Thèses de doctorat d'état, Université des sciences et des technologies du Languedoc, Montpellier, juin 1988
- [ROB 65]: ROBERT L.G., Machine perception of three-dimensional solids, chapter optical and electro-optical information processing, J.P. Tippet et Al Ed, Campridge, Massachussets MIT Press, 1965
- [SAP 76] : G. SAPUPPO, "Allocazione di forme irregolari su risorse rettangolari", Tesi di laurea, istituto di scienze dell' Informazione, Universita di Pisa, 1976
- [SAV 87] : G. SAVARD, "Les problèmes de découpe, un survol de leurs formulations et de leurs méthodes de résolution", rapport de recherche G-87-06, GERAD, Ecole des Hautes Etudes Commerciales, Montréal, Québec, Canada, 1987
- [SIA 84]: P. SIARRY, G. DREYFUS, "An application of Physical methods to the computer aided design of electronic circuits", J. Physique Lett-45, 1984, 1-39.
- [SIA 88]: P. SIARRY et G. DREYFUS, La méthode du recuit simulé: théorie et applications, Ecole Supérieure de Physique et de Chimie de la Ville de Paris, IDSET, Paris, 1988.
- [SIL 64] : M. SILVERMAN, "Marker layout manual", Hull, Massachussets, 1964
- [SOE 77]: R. SOENEN, "Conduite automatisée d'une utité de coupe en confection", Automatisation, pp 118-128, mai 1977
- [TAN 73] : M. TANAKA and T. WACHI, "Computerized marker making", J. Textil machinery society of japan, vol 19, pp 74-81, 1973
- [TAN 87]: P.M.TANCREZ, "Un système de vision linéaire pour l'inspection et la caractérisation en temps réel de produit plat en défilement continu", Thèse De Doctorat, Université des Sciences et Technologies de Lille, 1987

-
- [TOU 85]: G. TOULOUSE, "Applications de la physique statistique aux problèmes complexes", ESPCI, 1985.
- [VIR 76] : E. VIRGILIETTI, "Graphics in ship design", Sperry Univac Symposium: Computers in ship building, Roma 1976
- [WAL 84]: WALL K. and DANIELSON P., "A fast sequential method for polygonal approximation", Computer Vision, Graphics and Image Processing, 28, pp 220-227, 1984
- [ZAH 72]: ZAHN C.T., ROSKIES R.Z., "Fourier descriptors for plane closed curves", IEEE Trans. on Computers, 21(3), 1972, pp 269-281
- [ZIS 84] : V. ZISSIMOPOULOS, "Problèmes de découpe: algorithmes ε -approchants", thèse de 3ème cycle de l'université de Paris-Sud, 1984

-
- [ROU 92]: G. ROUSSEL, S. MAOUCHE, P. VIDAL, "A new description of irregular shapes boundary. Application to the nesting degree processing and shapes clustering for the optimal two dimensional marking problem in clothing industry", XI conference on Systems Sciences, Wroclaw, Poland, sept 1992.
- [ROU 93.a]: G. ROUSSEL, S. MAOUCHE, "Automatic lay-planning for irregular shapes on plain fabric. Search in direct graph and $A\varepsilon$ -admissible resolution", 16th IFIP conference on System Modeling and Optimization, july 1993, Compiègne, France.
- [ROU 93.b]: G. ROUSSEL, S. MAOUCHE, "Improvements about Automatic lay-planning for irregular shapes on plain fabric.", A paraître, IEEE/SMC'93 Conference, October 1993, Le Touquet, France.
- [ROU 93.c]: G. ROUSSEL, S. MAOUCHE, "Intelligent Lay-Planning System For Irregular Shapes And Sheet With Patterns And Flaws. Resolution by ε -Admissible Tree Search", A paraître, 24th ISIR 93 Conference, November 1993, Tokyo, Japon.
-

Topologie de l'espace discret

Convenons que dans l'espace continu, les formes traitées dans notre application sont des régions connexes séparées par des courbes simplement fermées. ¹ (Figure A.1)

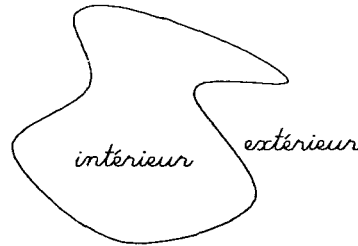


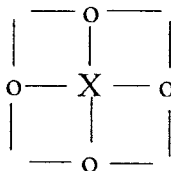
Figure A.1: Courbe simplement fermée

Dans ce qui suit, nous emploierons la métrique utilisant la distance dite "Chess board distance" ou distance du maximum s'exprimant par:

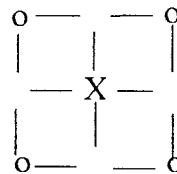
$$\forall P, Q \in \Sigma$$

$$d_8(P, Q) = \text{MAX}(|i_p - i_q|; |j_p - j_q|)$$

Avec cette distance, on obtient 8 points Q, voisins de P, vérifiant : $d_8(P, Q) \leq 1$. Les quatre adjacents par les côtés sont appelés voisins directs, les quatre autres sont les voisins indirects. On parle alors de 8-voisinage.



Les quatres voisins directs



Les quatres voisins indirects

Figure A.2

Compte tenu des considérations précédentes :

Un *chemin 8-connexe* (cf. Figure A.3) compris entre deux points P et Q est une suite de points du maillage de Σ :

$$P \equiv P_0, P_1, \dots, P_n \equiv Q \quad \text{tels que } \forall 1 \leq i \leq n, \quad d_8(P_{i-1}, P_i) \leq 1$$

¹Une courbe est simplement fermée si elle n'a aucun point d'intersection avec elle même.

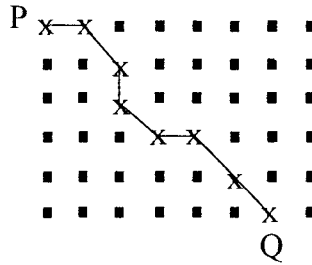


Figure A.3

Une *composante connexe* (cf. Figure A.4) est un ensemble de points connexes entre eux par au moins un chemin connexe:

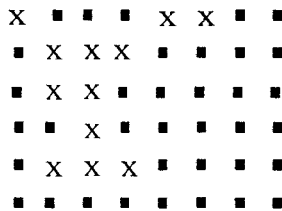
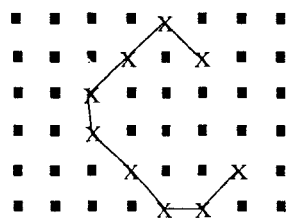


Figure A.4

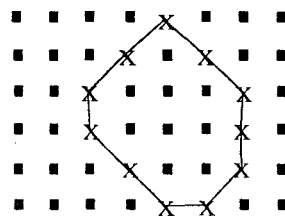
Un *arc* (cf. Figure A.5.a) est une composante connexe ordonnée par la suite P_0, P_1, \dots, P_n , d'extrémité P_0 et P_n tels que si $1 \leq i \leq n-1, n \in \mathbb{N}^*$, P_i a exactement deux voisins P_{i-1} et P_{i+1}

Une *courbe fermée* (cf. Figure A.5.b) est une composante connexe ordonnée par la suite P_0, P_1, \dots, P_n , de point initial quelconque, telle que $\forall i, P_i$ a exactement deux voisins $P_{i-1[n]}$ et $P_{i+1[n]}$ ², ($n \geq 4$ en 8-connexité)



arc 8-connexe

Figure A5.a



courbe 8-connexe

Figure A.5.b

²[n] : modulo n

Codes de description de contours

B.1 Les codes exacts

B.1.1 Codes de FREEMAN [FRE 61]

Il est sans doute le plus universel. On choisit, pour un contour donné, un point de départ $\mathcal{P}_0(x_0, y_0)$; puis on passe d'un point à un autre point connexe à l'aide du code de direction jusqu'au retour en \mathcal{P}_0 . Dans un maillage 8-connexes, l'étiquetage des huit directions est:

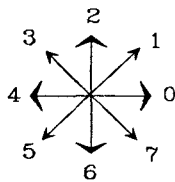


Figure B.1.a

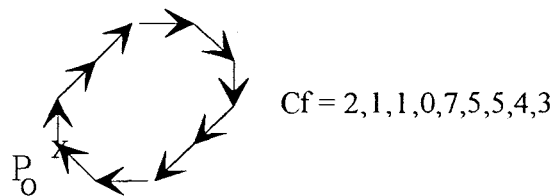


Figure B.1.b

De nombreux algorithmes ont été développés pour réaliser les traitements géométriques classiques: rotation d'un angle quelconque, union de deux contours, intersection de deux contours, homothétie, convexification, mise en correspondance de deux contours ou d'un sous-contour dans un contour, reconnaissance de forme etc... Cependant l'opération d'évaluation de complémentarité entre deux contours selon une direction fixée semble complexe. En utilisant les opérations primitives existantes, l'algorithme d'évaluation de la complémentarité aurait été le suivant:

{Evaluation de l'imbrication du contour C_i dans le contour C_j selon une direction δ (figure B.2)}

Etape 1 : Tant que $C_i \cap C_j = \{\emptyset\}$, translater C_i vers C_j selon la direction d

Etape 2 : Réaliser $C_i \cup C_j$

Etape 3 : évaluer par exemple la surface du rectangle exinscrit de $C_i \cup C_j$

{fin de l'évaluation d'une imbrication }

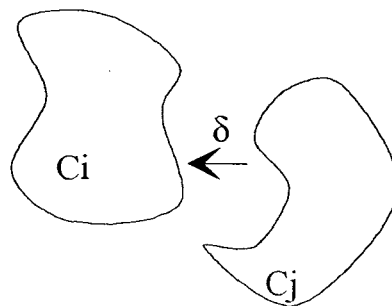


Figure B.2

Quelques codes dérivés du code de Freeman sont apparus:

- Code de Freeman par différence des directions : le code d'une frontière est défini par le point initial P_0 , la première direction α_0 et la suite des changements de directions $(\beta_i)_{i=1..n}$, $\beta_i = \alpha_i - \alpha_{i-1}$

Le code pour la figure B.1.b est : $P_0 = (1,4)$, $\alpha_0 = 2$, $(\beta_i) = (-1, 0, 1, 6, -1, 0, -1, -1, -1)$

- Code de Freeman dérivé : il consiste à détecter les convexités, les points d'inflexion, les concavités et de remplacer chaque transition par un nombre, respectivement 1,2 ou 3.

B.1.2 Codage par transition [HUA 75]

Chaque ligne de la matrice image est codée en fonction des transitions de niveaux de gris intervenant dans cette ligne. Le code consiste en une suite de couples (abscisse, couleur).

				1	1	1	0	0	8	
1	0	0	0	1	1	1	0	0		(4,1);(6,0)
	0	0	0	1	1	1	0	0		(3,1);(6,0)
	0	0	0	1	1	1	1	0		(3,1);(7,0)
	0	0	0	1	1	1	1	1		(3,1);(8,0)
	0	0	0	1	1	1	1	1		(3,1);(8,0)
	0	1	1	1	1	1	1	1		(1,1);(8,0)
	0	1	1	1	1	1	0	0		(1,1);(6,0)
8	0	0	0	0	0	0	0	0		0

Figure B.3

Ce codage privilégie la description dans la direction horizontale.

B.1.3 Signature polaire

Si on a extrait le contour d'un objet et déterminé son centre de gravité, on peut tracer le courbe $r = f(\theta)$ quand θ varie de 0 à 2π . Cette courbe s'appelle la signature polaire (**figure B.4**).

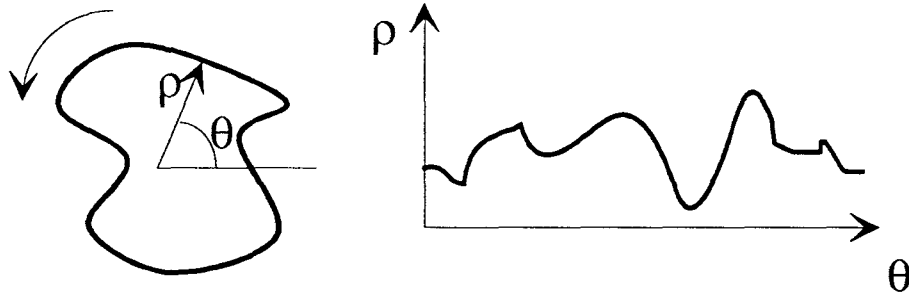


Figure B.4

En pratique, cette méthode a deux inconvénients majeurs:

- Pour des formes présentant des concavités, il se peut que le rayon soit multivalué. Ce qui oblige souvent de ne garder que le rayon maximum, mais le code n'est plus exact.

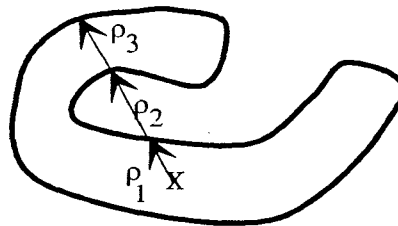


Figure B.5

- La résolution de description est souvent moins bonne que dans le code source, car on est obligé de choisir un pas angulaire supérieur à la plus grande variation angulaire engendrée par deux pixels de plus faible rayon polaire.

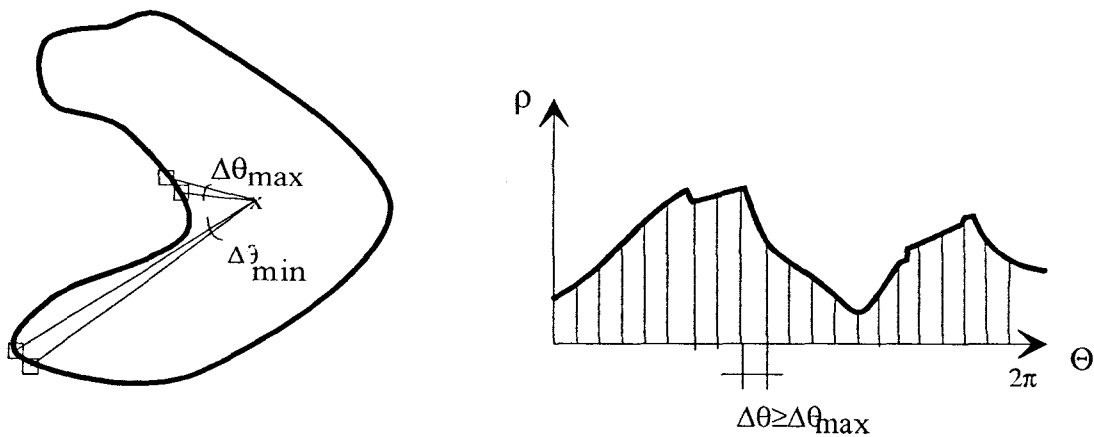


Figure B.6

On représente les variations de la courbe K du contour en fonction de l'abscisse curviligne s [BEN 75] par:

$$K(s) = \frac{d\Theta}{ds} \quad \text{avec} \quad \Theta(P_j) = \text{ArcTg} \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$$

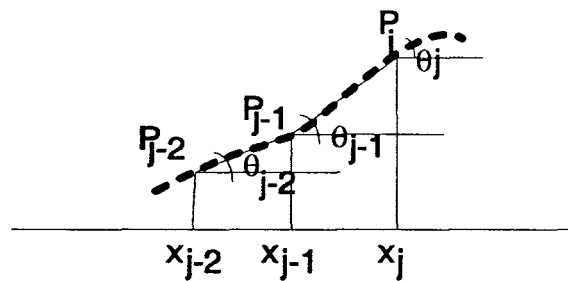


Figure B.7

On se contente généralement de calculer : $k(P_j) = \Theta(P_j) - \Theta(P_{j-1})$. Ce codage est souvent une étape intermédiaire pour ensuite sélectionner les points caractéristiques de la courbe. Mais cette idée pose des problèmes dans l'espace discret, car par exemple pour un cercle à courbure constante, il faudrait retenir tous les points ou aucun. Il faut alors redéfinir la notion de courbure discrète [LUX 85]

B.2 Les représentations approchées

B.2.1 Nombre de forme [BRI 79]

Le principe consiste à extraire le code de Freeman dérivé à partir de la forme orientée de telle façon que le rectangle circonscrit servant de cadre à l'espace de discrétisation soit d'excentricité maximale. L'ordre du nombre de forme correspond au périmètre discret du rectangle circonscrit. En contrôlant le pas de discrétisation, on contrôle aussi l'ordre du nombre et donc la finesse de la description.

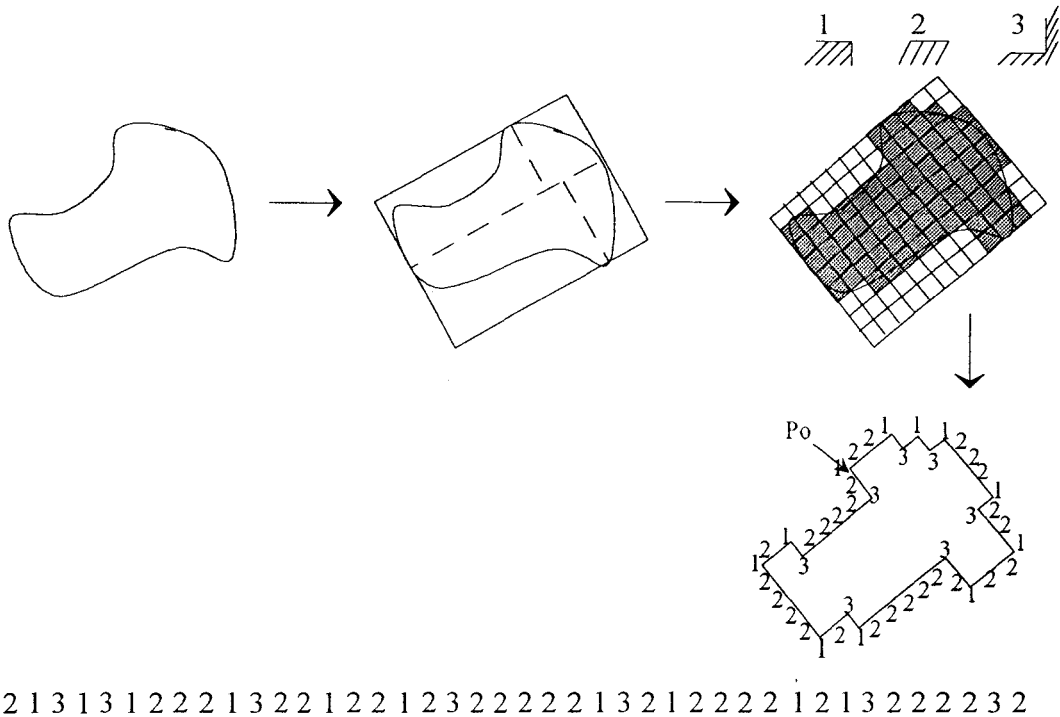


Figure B.8

Ce code est essentiellement intéressant pour mesurer la similarité entre deux formes.

B.2.2 Strip trees [BAL 81]

Le "Strip trees" est une représentation hiérarchique du contour d'une forme. L'algorithme consiste à réaliser une approximation de chaque portion de contour curviligne par une bande de largeur inférieure ou égale à la largeur w^* fixée par l'utilisateur. Le principe de digitalisation est le suivant:

Soit une courbe polygonale définie par ses sommets $[P_0, P_1, P_2, \dots, P_n]$. On commence par chercher le plus petit rectangle ayant deux de ses côtés parallèles à la ligne L joignant les extrémités, et couvrant tous les points de la courbe entre ces extrémités. Le rectangle est la bande racine du strip trees. Prendre ensuite un des points P_k touchant les côtés parallèles à L . Recommencer la procédure pour chaque sous liste de part et d'autre de ce point. $[P_0 \dots P_k]$ et $[P_k \dots P_n]$. Ces deux sous contours appartiennent à deux sous bandes, filles de la bande racine. La procédure est terminée quand toutes les bandes terminales ont une largeur inférieure à W^* . La représentation obtenue est un arbre binaire. Le contrôle de la résolution par le paramètre W^* et la richesse des primitives ensemblistes et géométriques devrait réserver à cette représentation un avenir prometteur.

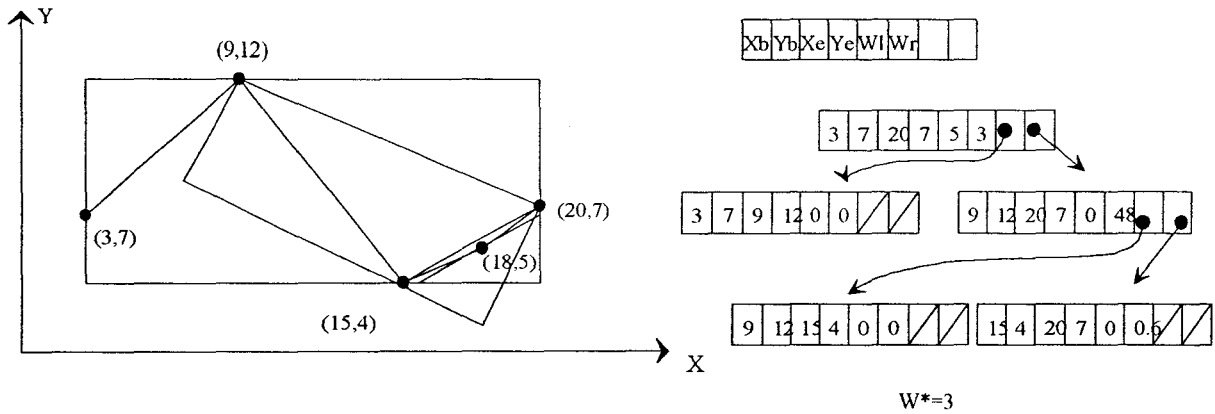


Figure B.9

B.2.3 Squelette de formes [MON 69]

Si la forme contient des parties relativement fine, elle peut être avantageusement décrite par son squelette. Le squelette est défini, en terme de distance d'un point P à une région R telle que:

$$ds(P,R) = \text{INF} \{d(P,Z), Z \in R\}$$

Les distances les plus utilisées sont les distances euclidienne, city block et chessboard distance. Le squelette est alors l'ensemble des couples $(P, d_s(P,C))$ où $d_s(P,C)$ est la distance de P au contour C. Si la distance est euclidienne, le squelette d'une forme est défini comme étant le lieu des centres de tous les cercles les plus grands pouvant être contenus dans la forme et balayant toute la surface de celle-ci. Chaque noeud du squelette est étiqueté par le rayon de son cercle de définition. L'extraction du squelette d'une forme permet de fournir une évaluation des creux et des bosses élémentaires, ainsi que les directions d'imbrication de celle-ci [DEL 89]

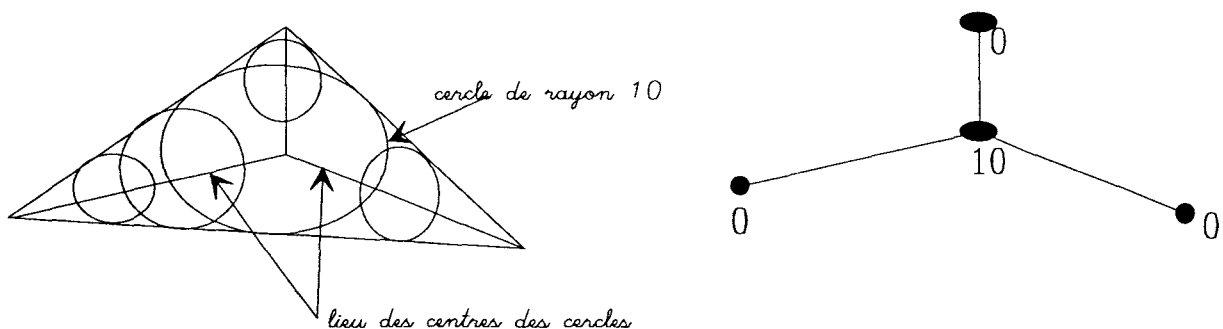


Figure B.10

Cette notion de squelettisation a déjà ses applications en traitement d'image en vue de la reconnaissance des formes, mais l'utilisation de cette méthode se heurte à l'absence d'algorithme et d'opérateur de traitement de ces squelettes. Des algorithmes efficaces de calcul des squelettes de formes sont donnés dans [PAV 77].

B.2.4 Descripteur de Fourier [ZAH 72]

Le principe consiste d'abord à déterminer la fonction angulaire curviligne $\Phi(l)$, qui est le cumul de la courbure $K(l) = d\theta/dl$, depuis le point initial de la courbe.

$$\Phi(l) = \delta\theta + \int_0^l K(x) dx \quad 0 \leq l \leq L$$

$\delta\theta$ étant l'angle de la tangente au point initial.

Ensuite la fonction est normalisée pour que son domaine de définition devienne $[0, 2\pi]$. Elle devient alors $\Phi^*(l)$ telle que :

$$\Phi^*(t) = \Phi\left(\frac{L \times t}{2\pi}\right) + t \quad t \in [0, 2\pi]$$

Ainsi Φ^* est invariante par translation, rotation et pour tout changement de périmètre L . De plus $\Phi^*(0) = \Phi^*(2\pi) = 0$.

Les descripteurs de Fourier sont alors les coefficients (A_k, α_k) obtenus après transformation par série de Fourier tel que

$$\Phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k)$$

Où A_k et α_k sont les amplitudes et phases harmoniques de rang k

Cette technique est surtout utilisée pour la comparaison de deux formes, indépendamment de l'orientation, de la position, ou du rapport d'homothétie qui peut les différencier.

B.2.5 Approximation polynomiale

Le codage consiste à trouver des coefficients numériques caractérisant une courbe analogique Γ la plus proche du contour réel C_k . Si les A_i et les B_i sont les coefficients de la description paramétrique de la courbe Γ par rapport à son repère d'édition, nous avons:

$$X(t) = \sum_i A_i \times t^i$$

$i \in [0, n]$, n étant le degré du
polynôme d'approximation

$$Y(t) = \sum_i B_i \times t^i$$

Des méthodes performantes ont été mises au point comme les courbes de Bezier ou les approximants de Spline [Deb 78]. Mais les calculs sont souvent longs, l'ordre des polynômes est souvent élevé pour des formes complexes surtout localement au niveau des points de cassure.

Algorithme d'approximation polygonale pour l'obtention des peignes de contour réduit

Soit $S_{k,c}$ le peigne constitué à partir de tous les points définis dans le peigne $D_{k,c}$ du sous contour $\tilde{A}(C_{k,c})$. Soit $V_{k,c}$ et $A_{k,c}$, les listes de vecteurs et surface algébriques. Soit enfin la variation de surface linéique maximale, on désigne par $S_{k,c}$ *approx_Poly*: ε le message correspondant à la méthode algorithmique d'approximation polygonal suivant:

```

 $S_{k,c}$  approx_Poly:  $\varepsilon$ 
début
  Pour i:= 1 à card( $S_{k,c}$ )-1
    faire
       $V_{k,c}(i) := S_{k,c}(i+1) - S_{k,c}(i)$  {calcul des vecteurs  $V_{k,c}(i)$ }
  finPour;
  Pour j:= 1 à card( $V_{k,c}$ )-1
     $A_{k,c}(j) := X_{V_{k,c}(j)} \times Y_{V_{k,c}(j+1)} - X_{V_{k,c}(j+1)} \times Y_{V_{k,c}(j)}$  {calculs de la 3ème
    composante
    du produit vectoriel}
  finPour;
  booléen := vrai; A:= 0;
  Tantque booléen
     $A_{\min} := \text{Un\_Majorant\_de\_} A_{k,c}$  ;
    Pour n:= 1 à card( $A_{k,c}$ )
      faire
        si  $A_{\min} > A_{k,c}(n)$  {recherche du  $A_{k,c}(n)$  minimal}
          alors
             $A_{\min} := A_{k,c}(n)$ ;
            suppr := n; {repérage de n}
          finSi
        finPour ;
      si  $A + A_{\min} < l_{k,c} \times \varepsilon$ 
        faire
           $A := A + A_{\min}$  ;
          supprimer  $A_{k,c}(\text{suppr})$ ;
          calculer  $V := V_{k,c}(\text{suppr}) + V_{k,c}(\text{suppr} + 1)$ ;
          remplacer  $V_{k,c}(\text{suppr})$  par  $V$ ;
          supprimer  $V_{k,c}(\text{suppr} + 1)$  et  $S_{k,c}(\text{suppr} + 1)$ ;
          booléen := vrai
        sinon booléen := faux
      finSi
    finTantque
  fin

```

Formules de changement de repère

Les formules suivantes permettent de passer du repère local $O_{k,c}XY$ lié au côté d'une forme F_k au repère local $O_{k,c}XY$ lié au même côté de la concaténation de deux formes F_{kul} . Elles sont valables pour les deux formes F_k ou F_l mais nous les présentons ici pour F_k .

Ces calculs pourront être effectués par une unique méthode dont le message s'écrira:

$$S_{k,c(i)}/O_{kul} := S_{k,c(i)}/O_k \text{ change_Repère_De: } \overline{\overline{OM_k}} \text{ vers: } \overline{\overline{OM_{k \cup l}}} \text{ côté: } c$$

$$\forall S_{k,c(i)} \in S_{k,c} \quad , \quad 1 \leq i \leq \text{card}(S_{k,c})$$

$$\overrightarrow{O_{k \cup l, c} S_{k,c(i)}} = \overrightarrow{O_{k \cup l, c} O_{k,c}} + \overrightarrow{O_{k,c} S_{k,c(i)}} = (1) + (2)$$

En développant (1) on a :

$$\begin{aligned} \overrightarrow{O_{k \cup l, c} S_{k,c(i)}} &= \overrightarrow{O_{k \cup l, c} O_{k \cup l}} + \overrightarrow{O_{k \cup l} O} + \overrightarrow{O O_k} \\ &\quad (1.1) \qquad (1.2) \qquad (1.3) \\ &+ \overrightarrow{O_k O_{k,c}} + \overrightarrow{O_{k,c} S_{k,c(i)}} \\ &\quad (1.4) \qquad (2) \end{aligned}$$

En changeant le sens de (1.1) et (1.2), on a:

$$\begin{aligned} \overrightarrow{O_{k \cup l, c} S_{k,c(i)}} &= \overrightarrow{O_k O_{k,c}} + \overrightarrow{O_{k,c} S_{k,c(i)}} + \overrightarrow{O O_k} \\ &- \overrightarrow{O_{k \cup l} O_{k \cup l, c}} - \overrightarrow{O O_{k \cup l}} \end{aligned}$$

d'où

$$\begin{aligned} \begin{pmatrix} X S_{k,c(i)} \\ Y S_{k,c(i)} \end{pmatrix} / O_{k \cup l} &= \begin{pmatrix} l_{k,0} \text{ si } c=1, 0 \text{ sinon} \\ l_{k,1} \text{ si } c=2, 0 \text{ sinon} \end{pmatrix} + \begin{pmatrix} X S_{k,c(i)} \\ Y S_{k,c(i)} \end{pmatrix} / O_{k \cup l} + \begin{pmatrix} m_{0,0}^k \\ m_{0,1}^k \end{pmatrix} \\ &- \begin{pmatrix} l_{k \cup l,0} \text{ si } c=1, 0 \text{ sinon} \\ l_{k \cup l,1} \text{ si } c=2, 0 \text{ sinon} \end{pmatrix} - \begin{pmatrix} m_{0,0}^{k \cup l} \\ m_{0,1}^{k \cup l} \end{pmatrix} \end{aligned}$$

Ces vecteurs sont extraits des matrices $\overline{\overline{O_k M_k}}$, $\overline{\overline{O_k M_{k \cup l}}}$

Algorithme de suppression de points cachés et de reconstruction de la frontière de jonction

Soit $S_{kul,c}$ la liste issue de la concaténation des listes $S_{k,c}$ et $S_{l,c}$, dans laquelle on suppose l'existence de points non visibles depuis le côté c du rectangle circonscrit R_{kul} . $S_{k,c}$ et $S_{l,c}$ sont des listes ordonnées dans l'ordre des abscisses croissantes. Les points de ces listes ont subi un changement de repère en fonction de la position relative des deux formes. L'algorithme est donc:

Entrée : $S_{k,c}$ et $S_{l,c}$

Pour $I := 1$ à $\text{card}(S_{k,c})$ ajouter $S_{k,c}(i)@'a'$ dans A

{création des étiquettes A et B }

Pour $J := 1$ à $\text{card}(S_{l,c})$ ajouter $S_{l,c}(j)@'b'$ dans B

Si $\text{abscisse}(A(1)) < \text{abscisse}(B(1))$

alors ajouter $A(1)$ à $S_{kul,c}$;

supprimer $A(1)$ de $S_{k,c}$

{Initialisation de la liste $S_{kul,c}$ avec un premier point}

sinon ajouter $B(1)$ à $S_{kul,c}$;

supprimer $B(1)$ de $S_{l,c}$

TantQue $A \neq \{\emptyset\}$ ou $B \neq \{\emptyset\}$

faire

si $A \neq \{\emptyset\}$ alors Tantque $B \neq \{\emptyset\}$ ajouter $B(1)$ à $S_{kul,c}$

supprimer $B(1)$ de B

{une des listes est vide, on termine avec la liste non vide}

si $B \neq \{\emptyset\}$ alors Tantque $A \neq \{\emptyset\}$ ajouter $A(1)$ à $S_{kul,c}$

supprimer $A(1)$ de A

si $A \neq \{\emptyset\}$ et $B \neq \{\emptyset\}$ alors

si $\text{abscisse}(A(1)) \leq \text{abscisse}(B(1))$

alors $P := A(1)$; $Q := B(1)$

{recherche du prochain point à traiter}

sinon $P := B(1)$; $Q := A(1)$

si $\text{étiquette}(P) = \text{étiquette}(\text{dernier point}(S_{kul,c}))$

{pas de changement d'étiquetage, on ajoute le point à la liste $S_{kul,c}$ }

alors ajouter P à $S_{kul,c}$

supprimer P de $\text{provenance}(P)$

sinon

si $Y_P < Y_Q$ alors

{si changement d'étiquetage et le point P est visible, alors on l'ajoute à $S_{kul,c}$ et on supprime tous les points de l'autre liste appartenant à l'intervall d'intersection}

ajouter P à $S_{kul,c}$

supprimer P de $\text{provenance}(P)$

TantQue $X_Q \leq X_{O_{knl,c}} + l_{knl,c}$

supprimer Q

sinon

{si changement d'étiquetage et le point Q est visible, alors on l'ajoute à $S_{kul,c}$ et on supprime tous les points de l'autre liste appartenant à l'intervall d'intersection}

ajouter Q à $S_{kul,c}$

supprimer Q de $\text{provenance}(Q)$

TantQue $X_P \leq X_{O_{knl,c}} + l_{knl,c}$

supprimer P

Algorithme de calcul de la profondeur d'imbrication

Soient $S_{k,c}$ et $S_{l,c+2[4]}$ deux peignes réduits de deux formes F_k et F_l distinctes. L'algorithme suivant s'attache à déterminer la profondeur maximale d'imbrication des deux formes pour un décalage d donné des origines locales $O_{k,c}$ et $O_{l,c+2[4]}$ liées à chacun des deux peignes. Il est appelé par le message:

$Pi := S_{k,c}$ *disjonction_avec:* $S_{l,c+2[4]}$ *décalage:* d

soit i, j deux entiers.

Entrée: $\{S_{k,c}, S_{l,c+2[4]}\}$

$j=1, i=1, P_{imax} = \text{une_valeur_élevée}$

Tantque1 $x_{S_{k,c}}(i) \leq b$

faire:

si $(x_{S_{k,c}}(i) \geq a) \wedge (S_{k,c}(i) \text{ convexe})$

alors

Tantque $x_{S_{k,c}}(i) \leq (x_{S_{l,c+2[4]}}(j) - d)$

faire $j := j + 1$ { on encadre $x_{S_{k,c}}(i)$ sur l'intervalle $[x_{S_{l,c+2[4]}}(j-1), x_{S_{l,c+2[4]}}(j)]$ }

finTq

$P_{imax} = \text{Min}(P_{imax}; y_{S_{k,c}}(i) + y_{S_{l,c+2[4]}}(j-1) +$

$$\left[\frac{y_{S_{l,c+2[4]}}(j) - y_{S_{l,c+2[4]}}(j-1)}{x_{S_{l,c+2[4]}}(j) - x_{S_{l,c+2[4]}}(j-1)} \times \left[(x_{S_{l,c+2[4]}}(j) - d) - x_{S_{k,c}}(i) \right] \right]$$

$i := i + 1$

finSi

finTq

1

Ensuite on cherche P_{imax} en recommençant la même étape pour les points de $S_{l,c+2[4]}$ et on détermine le minimum des deux valeurs .

¹ $S_{k,c}(i)$ est un point convexe d'ordre 1 si $\vec{S_{k,c}(i-1)S_{k,c}(i)} \wedge \vec{S_{k,c}(i)S_{k,c}(i+1)}$ est orienté négativement dans la direction perpendiculaire au plan formé par les deux vecteurs. La recherche de convexité peut se faire soit en ligne, soit elle peut être obtenue directement dans la structure de donnée associée à chaque forme.

algorithme de recherche du décalage optimal (tissu uni)

Cet algorithme détaille la procédure de recherche du décalage optimal d_{opt} dans le cas d'un tissu uni. On ne présente ici que l'algorithme où l'on essaie d'imbriquer la zone bosse prédominante de F_l dans les zones creuses de F_k . Naturellement, la procédure reste identique dans le cas opposé. Il suffit de permuter $S_{k,c}$ et $S_{l,c+2}[4]$ dans l'algorithme, et de remplacer I_{min1} et d_{opt1} par I_{min2} et d_{opt2} . (Remarque: la variable $X_{ptgl,c+2}$ de l'algorithme n'est autre que l'abscisse du premier point de tangence du segment bosse dans le côté $(c+2)[4]$ de F_l).

$$\text{Entrée} = \{ F_k, F_l, c, SR_{k \cup l}^{maj} \}$$

$$\text{borne_Inf} = -\frac{SR_{k \cup l}^{maj}}{\max(I_{k,c+1}, I_{l,c+1})} + I_{k,c}; \quad \text{borne_Sup} = \frac{SR_{k \cup l}^{maj}}{\max(I_{k,c+1}, I_{l,c+1})} - I_{l,c};$$

$$I_{min1} := \text{une_Valeur_Elevée}; \quad I_1 := I_{min1} - 1;$$

Pour $i := 2$ jusque $(\text{card}(S_{k,c}) - 1)$

faire:

si $(S_{k,c}(i) \text{ concave}) \wedge (S_{k,c}(i+1) \text{ concave})$ { détection d'un segment creux de $\tilde{A}(C_{k,c})$ }

alors

$$\text{longueur_Zone_Creuse} := X_{S_{k,c}(i+2)} - X_{S_{k,c}(i-1)};$$

$$d := \text{Max}(X_{S_{k,c}(i-1)} - X_{Ptgl,c+2}, \text{borne_Inf}); \quad \left\{ \begin{array}{l} \text{On positionne la bosse au} \\ \text{début de la zone creuse ou} \\ \text{au début de l'intervalle} \\ \text{autorisé si celui est plus élevé} \end{array} \right\}$$

$$\text{Tantque } (I_1 < I_{min1}) \wedge (d \leq \text{borne_Sup}) \quad \left\{ \begin{array}{l} \text{Tantqu'il y a décroissance} \\ \text{de l'indice de complémentarité} \\ \text{et que l'on ne dépasse pas les} \\ \text{limites de l'intervalle de recherche} \end{array} \right\}$$

faire

$$I_{min1} = \text{Min}(I_{min1}, I_1);$$

$$d_{opt1} = d;$$

$$Pi_{max1} = S_{k,c} \text{ disjonction_avec: } S_{l,c+2} \text{ décalage: } d;$$

$$I_1 = I_6(F_k, F_l, c, d, Pi_{max1});$$

$$d := d + \left\lfloor \frac{\text{longueur_Zone_Creuse}}{9} \right\rfloor$$

finTantque

finSi

finPour

$$\text{Sortie: } \{ I_{min1}, d_{opt1}, Pi_{max1} \}$$

algorithme de recherche du décalage optimal (tissu à motifs)

Cet algorithme détaille la procédure de recherche du décalage optimal d_{opt} dans le cas d'un tissu à motif. La recherche de la position optimale se fait en balayant un interval segmenté par le contrainte de périodicité du motif.

$$\text{Entrée} = \{ F_k, F_l, SR_{k \cup l}^{maj} \}$$

$$\text{borne_Inf} = -\frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} + l_{k,c}; \quad \text{borne_Sup} = \frac{SR_{k \cup l}^{maj}}{\max(l_{k,c+1}, l_{l,c+1})} - l_{l,c};$$

$$I_{min1} := \text{une_Valeur_Elevée}; \quad I := I_{min1} - 1;$$

Si c pair alors

$n_x := 0$; { n_x nombre de périodes du motif dans la direction Ox }
 jusque $d(n_x) \geq \text{borne_Inf}$ faire $n_x := n_x + 1$ finFaire; { on se positionne au plus près du }
 { début de l'interval de recherche }

jusque $(d(n_x) \geq \text{borne_Sup})$ ou $(n_x \leq \lfloor \frac{E_x}{T_x} \rfloor)$ { jusqu'à ce qu'on ait atteint l'autre }
 { extrémité de l'interval de recherche }
 { ou les limites de l'espace }

faire

calculer $Pi_{max}(d(n_x))$;

si $c = 0$ alors

$$n_y = \left\lfloor \frac{E_y}{T_y} \right\rfloor + 1;$$

jusque $Pi(n_y) \leq Pi_{max}(d(n_x))$ faire $n_y := n_y - 1$ finFaire;
 { on diminue $Pi(n_y)$ jusqu'à }
 { satisfaire la contrainte de disjonction }

sinon

$$n_y = 0;$$

$$\text{jusque } Pi(n_y) \leq Pi_{max}(d(n_x)) \wedge (n_y \leq \left\lfloor \frac{E_y}{T_y} \right\rfloor)$$

$$\text{faire } n_y := n_y + 1;$$

{ on incrémente $Pi(n_y)$ jusqu'à }
 { satisfaire la contrainte de disjonction }

finSi

calculer $I := I_6(F_k, F_l, c, d(n_y), Pi(d(n_y)))$

si $I < I_{min}$ alors

$$d_{opt} := d(n_x); \quad Pi_{opt} := Pi(n_y);$$

$$I_{min} := \text{Min}(I, I_{min})$$

finSi

$$n_x := n_x + 1$$

FinFaire

FinSi

Si c impair

alors

$n_y := 0;$

jusque $d(n_y) \geq borne_Inf$ faire $n_y := n_y + 1$ finFaire; { on se positionne au plus près du }
 début de l'intervalle de recherche }

jusque $(d(n_y) \geq borne_Sup) \wedge (n_y \leq \left\lfloor \frac{E_y}{T_y} \right\rfloor)$ { jusqu'à ce qu'on ait atteint l'autre }
 extrémité de l'intervalle de recherche }

faire

calculer $Pi_{max}(d(n_y));$

si $c = 3$ alors

$n_x = \left\lfloor \frac{E_y}{T_y} \right\rfloor + 1;$

jusque $Pi(n_x) \leq Pi_{max}(d(n_y))$ faire $n_x := n_x - 1$ finFaire;

{ on décrémente $Pi(n_x)$ jusqu'à }
 satisfaire la contrainte de disjonction }

sinon

$n_y = 0;$

jusque $Pi(n_x) \leq Pi_{max}(d(n_y)) \wedge (n_x \leq \left\lfloor \frac{E_x}{T_x} \right\rfloor)$

faire $n_x := n_x + 1$ finFaire;

{ on diminue $Pi(n_x)$ jusqu'à }
 satisfaire la contrainte de disjonction }

finSi

calculer $I := I_6(F_k, F_l, c, Pi(d(n_y)), d(n_y))$

si $I < I_{min}$ alors

$d_{opt} := d(n_y); Pi_{opt} := Pi(n_x);$

$I_{min} := Min(I, I_{min})$

finSi

$n_y := n_y + 1$

FinFaire

FinSi

Fin

Sortie : { $I_{min}, d_{opt}, Pi(n_x)$ si c est pair, $Pi(n_y)$ si c est impair }

Cet algorithme possède une limitation relative aux valeurs de la période du motif. Si T_x et T_y sont de l'ordre de grandeur de la dimension des pièces, alors certaines recherches risquent d'être infructueuses car aucune valeur de " d " ne va appartenir à l'intervalle $[borne_Inf, borne_Sup]$. Il convient dans ce cas d'élargir cet intervalle.

Elagage des noeuds pendants non ε -admissibles

I.1 Proposition

Supposons que $f=g+h$ soit croissante et que l'on connaisse une solution de coût estimé $f(t)$. Nous allons montrer qu'il est alors inutile de développer les noeuds pendants dont le coût $f(u)$ est tel que $f(u) > f(t)$

I.2 Démonstration

Soit $f(t)$ le coût estimé d'une solution connue à un noeud terminal t de l'arborescence. Si f est croissante le long d'un chemin passant par u menant de la racine u_0 à un noeud terminal, alors:

$$\forall v \in \hat{S}(u) \text{ on a } f(v) > f(u) \quad (\hat{S}(u) = \text{ensemble des descendants de } u).$$

Supposons donc que l'on connaisse un noeud pendant u de coût estimé $f(u) > f(t)$, alors:

$\forall v \in \hat{S}(u)$ on a $f(v) > f(t)$. Ce qui signifie que le coût à l'arrivée du chemin passant par u sera forcément plus élevé que le coût de la solution t . Il n'est donc pas utile de développer u . Ici, $f(t)$ constitue un majorant des coûts des chemins potentiellement admissibles.

Cette propriété permet d'éviter de conserver dans la liste des noeuds à développer, ceux qui ne laissent aucune chance d'amélioration de la fonction de coût.

Couples de formes utilisées dans l'algorithme d'évaluation de l'influence de l'approximation polygonale

ϵ

0.5

1

3

Couple A

A_m

Surface: 16 284 px²

longueur: 240 px

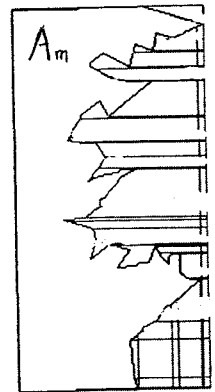
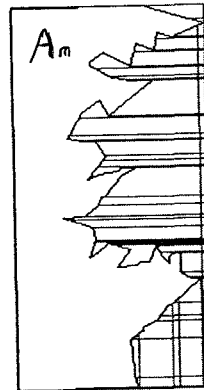
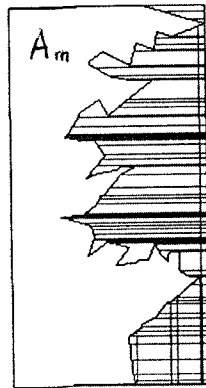
largeur: 121 px

A_e (rectangle)

Surface: 5996 px²

longueur: 59 px

largeur: 103 px



nb sommets côté 2 = 68

nb sommets côté 2 = 43

nb sommets côté 2 = 27

Couple B

B_m

Surface: 29 242 px²

longueur: 322 px

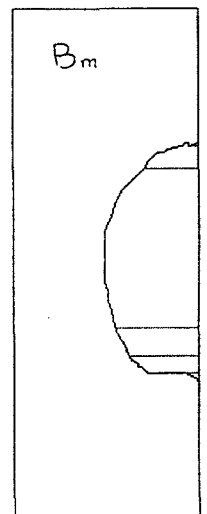
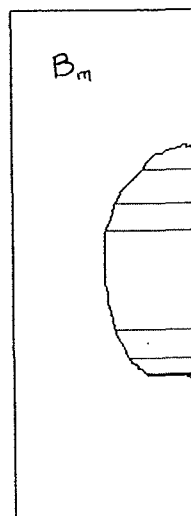
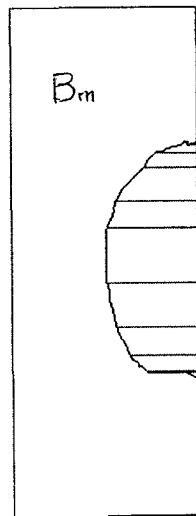
largeur: 115 px

B_e (rectangle)

Surface: 5996 px²

longueur: 59 px

largeur: 103 px



nb sommets côté 2 = 17

nb sommets côté 2 = 15

nb sommets côté 2 = 12