

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par François-Luc Cimelière

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Mathématiques Appliquées

**OPTIMISATION DU TRAITEMENT DE L'ORDRE DE FABRICATION
DANS L'INDUSTRIE TEXTILE**

Soutenue le : 05 novembre 2004

Après avis de :

M.	P. Bruniaux	Maître de conférence	ENSAIT, Lille	Rapporteurs
M.	T. Pham Dihn	Professeur	INSA, Rouen	

Devant la commission d'examen formée de :

M.	F. Vanderbeck	Professeur	Université Bordeaux I	Président
M.	P. Bruniaux	Maître de conférence	ENSAIT, Lille	Rapporteurs
M.	T. Pham Dihn	Professeur	INSA, Rouen	
M.	P. Bousquet	Chef de service	LECTRA	Examineurs
M.	P. Morel	Professeur	Université Bordeaux I	
Mme	C. Ondars	Chef de projet	LECTRA	

AVANT PROPOS

Ce mémoire décrit les travaux de recherche que j'ai effectués dans le cadre d'un contrat de thèse CIFRE entre l'université Bordeaux I et l'entreprise Lectra basée à Cestas.

Je tiens à remercier tout particulièrement Paul Morel pour la confiance qu'il m'a accordée en me proposant cette thèse. Durant ces trois années, il a su me guider par ses précieux conseils tout en me préservant une grande liberté dans mon travail. Les recherches exposées dans les pages qui vont suivre lui doivent beaucoup.

Je voudrais également exprimer ma gratitude à l'ensemble du personnel de Lectra qui m'a cordialement accueilli pendant ces travaux. J'adresse un merci tout particulier à l'équipe Optimino qui me compte dans ses rangs depuis un peu plus de trois ans. Merci donc, à Patrick Ducasse qui a été le premier à me former sur le sujet Optiplan et qui m'a de nombreuses fois éclairé sur les lois de l'informatique. Merci à Christine Ondars responsable de l'équipe qui m'a permis de me consacrer entièrement à mon travail de recherche. Merci à Antoine Georgevail qui, entre autres tâches, travaille avec moi sur l'intégration des nouveaux algorithmes dans Optiplan. Outre ses capacités à corriger mes nombreuses erreurs de programmation, il est également un grand pourvoyeur de blagues pour toute l'équipe. Merci à l'équipe basée en Allemagne Frank Steidl, Helmut Wagner et Heinz Wrosch qui ont initié le projet Optiplan. Merci enfin à tous ceux qui de près ou de loin ont contribué à rendre l'ambiance de travail aussi agréable.

Ce mémoire étant en quelque sorte l'aboutissement d'un certain nombre d'années passées sur les bancs des écoles, grandes ou petites, je tiens également à remercier tous les professeurs qui ont su au fil du temps me donner le goût d'apprendre. J'espère le conserver le plus longtemps possible.

A titre plus privé, je voudrais dire un grand merci à mes parents sans qui bien sûr je ne serais pas-là. Ils ont été et sont toujours mes premiers professeurs, ils sont aussi bien plus que cela...

Merci à celle qui m'accompagne... de m'accompagner.

TABLE DES MATIERES

Introduction	15
Première Partie	19
Chapitre 1 : Présentation du problème	23
1.1 Vocabulaire.....	23
1.2 Problématique.....	27
1.3 Exemple de construction de sections.....	29
1.4 Les coûts	32
1.4.1 Les coûts matières	32
1.4.2 Les coûts de matelassage.....	32
1.4.3 Les coûts de coupe	33
1.4.4 Les coûts placements.....	33
1.4.5 Les coûts d'évacuation	34
1.4.6 Remarques générales sur les coûts	34
1.4.6.1 Les vitesses	34
1.4.6.2 Importance relative des coûts	34
1.5 Les contraintes	36
1.5.1 Découper les quantités souhaitées.....	36
1.5.2 Sur la création des placements	36
1.5.2.1 Création ou non de nouveaux placements ?.....	36
1.5.2.2 Limitation des combinaisons entre tailles.....	37
1.5.2.3 Nombre maximum de tailles dans un placement	37
1.5.2.4 Autres contraintes	37
1.5.3 Contraintes de la salle de coupe	38
1.5.3.1 Nombre maximum de plis par section	38
1.5.3.2 Longueur maximale d'une section.....	38
1.5.3.3 Nombre minimum de plis pour l'utilisation d'un placement.....	38
1.5.4 Respect des stocks de tissu.....	39
1.6 Longueur des nouveaux placements.....	40
1.7 Traitement actuel	42

1.7.1	Résolution pas à pas	42
1.7.2	Choix du meilleur placement	42
1.7.3	Analyse des solutions obtenues.....	43
Chapitre 2 : Modélisation mathématique.....		47
2.1	Notations, satisfaction de l'OF et variables X	47
2.2	Fonction économique, autres variables	48
2.2.1	Définition des variables.....	49
2.2.1.1	Variables $X(j, s, p)$	49
2.2.1.2	Variables $Y(p)$	49
2.2.1.3	Variables $Z(p)$	49
2.2.1.4	Variables $E(j, p)$	50
2.2.2	Définition des coûts.....	50
2.2.2.1	Coûts associés aux variables $X(j, s, p)$	50
2.2.2.2	Coûts associés aux variables $Y(p)$	51
2.2.2.3	Coûts associés aux variables $Z(p)$	51
2.2.2.4	Coûts associés aux variables $E(j, p)$	52
2.2.3	Résumé de la fonction économique	52
2.3	Les contraintes	53
2.3.1	Découper les quantités de l'OF	53
2.3.2	Respecter les stocks disponibles	53
2.3.3	Nombre maximum de plis dans une section.....	54
2.3.4	Contraintes sur la création des placements	55
2.3.5	Nombre minimum de plis par placement	55
2.3.6	Calcul du nombre de paquets pour l'évacuation	57
2.4	Résumé de la mise en équations	58
2.5	Résolution directe	61
2.5.1	Algorithme du Simplexe	61
2.5.2	Méthode de Branch & Bound.....	61

2.5.3	Choix d'un logiciel de résolution.....	63
2.5.4	Cas où la résolution directe est possible.....	64
2.5.5	Cas où la résolution directe est impossible	65
Chapitre 3 : La méthode du recuit simulé.....		67
3.1	Introduction	67
3.2	De la métallurgie aux mathématiques	67
3.3	Fondements théoriques.....	69
3.3.1	Quelques rappels sur les chaînes de Markov	69
3.3.2	Application au recuit simulé	70
3.4	Algorithme.....	72
3.5	Remarques importantes	74
3.5.1	Les paramètres.....	74
3.5.2	La notion de voisinage	74
Chapitre 4 : Méthodes de recherche des placements.....		75
4.1	Introduction	75
4.1.1	Notion de « configuration de placement »	76
4.2	Recherche exhaustive	77
4.2.1	Calcul du nombre de configurations possibles.....	77
4.2.2	Calcul du nombre de placements dans une configuration.....	81
4.2.3	Analyse des cas où la recherche exhaustive est possible	82
4.3	Recherche par « configuration de placement »	84
4.3.1	Problème de l'augmentation du nombre de colonnes de l'OF	84
4.3.2	Sélection d'un échantillon de placements pour une configuration donnée.....	86
4.3.2.1	Placement permettant de découper le plus de plis pour un tissu donné	87
4.3.2.2	Système de poids pour la sélection des placements.....	88
4.4	Recherche par nombre de tailles.....	91
4.4.1	Problème de l'augmentation du nombre de configurations	91
4.4.2	Sélection d'un échantillon de placements pour un nombre de tailles donné ...	92
4.4.2.1	Placement n -tailles permettant de découper le plus de plis pour chaque tissu.....	92
4.4.2.2	Exemple.....	95

4.4.2.3	Système de poids pour la sélection des placements	96
Chapitre 5 :	Nouvelle méthode de résolution	99
5.1	Recherche d'une solution initiale	99
5.2	Construction des sections et recuit simulé	100
5.2.1	Démarche globale.....	100
5.2.2	Paramètres du recuit simulé	101
5.2.2.1	Décroissance de la température	101
5.2.2.2	Initialisation de la température	102
5.2.2.3	Nombre et longueur des paliers de température.....	103
5.3	Résolution d'un sous problème	103
5.3.1	Construction d'un problème relaxé	104
5.3.1.1	Rappel de la modélisation.....	104
5.3.1.2	Suppression des variables Y , Z et E	105
5.3.1.3	Relaxation des contraintes d'intégrité.....	107
5.3.2	Obtention d'une solution réalisable.....	108
5.3.2.1	Troncature des variables	108
5.3.2.2	Utilisation des placements par ordre de coût croissant	109
Chapitre 6 :	Partitionnement des OF de grande taille.....	111
6.1	Introduction	111
6.2	Principe.....	112
6.3	Comment regrouper les tailles ? Méthodes de classification	113
6.3.1	Mise au point d'un indice de dissimilarité entre les tailles.....	113
6.3.2	Méthodes de classification	116
6.3.2.1	Création d'une partition de l'ensemble des tailles d'un ordre de fabrication	117
6.3.2.2	Création d'un arbre hiérarchique	118
6.3.2.3	Comparaison des différentes méthodes sur un exemple	121
6.3.3	Contrôle du nombre de classes dans la partition retenue	122
6.3.3.1	Ne pas « casser » des OF contenant moins de 10 tailles	123
6.3.3.2	Utiliser les meilleurs placements autorisés	123
6.3.3.3	Résoudre si possible de petits sous-ordres.....	123
6.4	Conclusion sur la résolution des OF de grande taille.....	124
Chapitre 7 :	Résolution avec déviations sur les quantités.....	127

7.1	Problématique.....	127
7.2	Méthode d'optimisation d'un quotient de deux fonctions linéaires....	128
7.3	Application à la construction de sections.....	129
7.4	Obtention d'une solution réalisable.....	130
Chapitre 8 : Exemples & Comparaisons.....		133
8.1	Exemple 1 : Résolution de base.....	133
8.2	Exemple 2 : Influence de la sélection des placements réalisables	137
8.3	Exemple 3 : Partitionnement en plusieurs sous-ordres	141
8.4	Exemple 4 : Influence des déviations sur les quantités.....	143
Chapitre 9 : Qualité des solutions.....		147
9.1	Introduction	147
9.2	Coûts matière uniquement.....	147
9.3	Problèmes à solution optimale connue.....	150
9.4	Autres problèmes	151
9.5	Résumé	153
Chapitre 10 : Conclusion.....		155
10.1	SPGO : Une nouvelle méthode de construction des sections	155
10.1.1	Méthodes de sélection des placements.....	156
10.1.2	Résolution d'un sous-problème à chaque itération	156
10.1.3	Partitionnement des OF de grande taille	156
10.1.4	Résolution avec des déviations sur les quantités.....	157
10.2	Avantages et inconvénients de la méthode.....	157
10.3	Perspectives	158
10.3.1	Perfectionnement des algorithmes annexes.....	158
10.3.2	Combinaisons des 2 approches de résolution.....	160
10.3.3	Amélioration des tables d'efficience.....	160
Bibliographie		161

Deuxième Partie	163
Chapitre 1 : Origine et positionnement du problème	167
1.1 Origine du problème	167
1.2 Traitement actuel du problème	168
1.3 Positionnement du problème	169
1.4 Création d'une base de placements	171
Chapitre 2 : Méthode de résolution	173
2.1 Remarques préalables	173
2.2 Les réseaux de neurones	175
2.2.1 Références	175
2.2.2 Pourquoi les réseaux de neurones ?.....	175
2.2.3 Qu'est ce qu'un neurone ?.....	175
2.2.4 Organisation des neurones en réseau	177
2.2.5 Réseau à une couche cachée de sigmoïdes et un neurone de sortie linéaire ..	178
2.2.6 Propriété fondamentale : l'approximation parcimonieuse	179
2.2.6.1 L'approximation universelle.....	179
2.2.6.2 La parcimonie	179
2.2.7 L'apprentissage supervisé des réseaux de neurones	180
2.2.7.1 But de l'apprentissage.....	180
2.2.7.2 Technique d'apprentissage	181
2.2.8 Sélection des variables d'entrées : Méthode du descripteur sonde	182
2.2.8.1 Procédure d'orthogonalisation de Gram-Schmidt.....	183
2.2.8.2 Sélection des variables d'entrées à partir du classement obtenu.....	184
2.2.9 Sélection de modèles : Leave-one-out virtuel	184
2.2.9.1 Elimination des modèles surajustés	186
2.2.9.2 Principe du Leave-one-out virtuel	186
Chapitre 3 : Recherche de paramètres influençant l'efficience	189
3.1 Paramètres globaux	189
3.1.1 Nombre de tailles à placer	189
3.1.2 Laize du placement.....	192

3.1.3	Surface par taille à placer	193
3.1.4	Indice « d'éloignement » des tailles placées	195
3.1.5	Indice de « rectangularité »	196
3.1.6	Nombre de « petites/grandes pièces »	198
3.2	Paramètres spécifiques	200
3.2.1	Placements de chemises	200
3.2.1.1	Variations d'efficience en fonction de la laize.....	200
3.2.1.2	Les pièces importantes.....	202
3.2.1.3	Les bonnes configurations	204
3.2.1.4	Définition des indices	209
3.2.2	Placements de pantalons.....	212
3.2.2.1	Notion de « colonne ».....	212
3.2.2.2	Remarque sur les ceintures	213
3.2.2.3	Définition des indices	215
Chapitre 4 :	Premiers résultats	221
4.1	Démarche	221
4.1.1	Placements mono-tailles et placements multi-tailles	221
4.1.2	Elimination des mauvais résultats avérés.....	221
4.1.3	Initialisation des poids du réseau	222
4.1.4	Ensemble d'apprentissage – Ensemble de validation	222
4.1.5	Sélection des modèles	222
4.2	Placements de pantalons.....	224
4.2.1	Placements mono-taille	224
4.2.1.1	Placements disponibles.....	224
4.2.1.2	50% Apprentissage – 50% Validation	225
4.2.1.3	Diminution du nombre de placements dans l'ensemble d'apprentissage.....	226
4.2.2	Résultats sur tous les placements	229
4.2.2.1	Placements disponibles.....	229
4.2.2.2	Sélection du modèle (nombre de neurones cachés)	229
4.2.2.3	Stratégie de sélection du modèle	232
4.2.2.4	Résultats pour un modèle à 12 neurones cachés.....	232
4.3	Placements de chemises	234
4.3.1	Placements mono-taille	234

4.3.1.1	Placements disponibles.....	234
4.3.1.2	50% Apprentissage – 50% Validation.....	234
4.3.1.3	Diminution du nombre de placements dans l'ensemble d'apprentissage.....	236
4.3.2	Résultats sur tous les placements.....	239
4.3.2.1	Placements disponibles.....	239
4.3.2.2	Sélection du modèle (nombre de neurones cachés).....	239
4.3.2.3	Stratégie de sélection du modèle.....	243
4.3.2.4	Résultats pour un modèle à 13 neurones cachés.....	243
Chapitre 5 : Conclusion.....		245
5.1	Choix des réseaux de neurones.....	245
5.1.1	Architecture.....	245
5.1.2	Apprentissage supervisé.....	245
5.1.3	Variables d'entrées.....	246
5.1.4	Choix du modèle.....	246
5.2	Résultats obtenus.....	246
5.2.1	Placements de pantalons.....	246
5.2.2	Placements de chemises.....	247
5.3	Perspectives.....	248
5.3.1	Contraintes sur le positionnement des pièces.....	248
5.3.2	D'autres types de placements.....	249
5.3.3	Regroupements entre types de placements ?.....	249
Annexe 1: Calcul du gradient de la fonction de coût par		
rétropropagation.....		251
A1.1	Notations.....	251
A1.2	Calcul de $\left(\frac{\partial J^k(w)}{\partial w_j^s} \right)_{1 \leq j \leq n_{NC}}$	253
A1.3	Calcul de $\left(\frac{\partial J^k(w)}{\partial w_{jj'}^i} \right)_{\substack{1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1}}}$	253
A1.4	Cas d'un réseau à une couche cachée.....	255

Annexe 2: Minimisation de la fonction de coût avec la méthode de Levenberg-Marquardt.....	257
Annexe 3: Définition de l'indice d'éloignement d'un placement	259
A3.1 Notations et définition	259
A3.2 Recombinaison des termes	260
A3.3 Calcul de l'indice.....	261
Annexe 4: Exemple d'utilisation de la prédiction d'efficience	263
A4.1 Résolution d'un bordereau de coupe	263
A4.2 Résolution avec Optiplan	264
A4.3 Résolution avec recuit simulé et les tables d'efficience.....	264
A4.4 Résolution avec recuit simulé et prédiction d'efficience	265
Bibliographie	267

Introduction

Depuis une trentaine d'années, LECTRA développe une offre technologique dédiée aux industries fortement utilisatrices de textiles, de cuirs et autres matériaux souples. Leader mondial dans le domaine de la CFAO (Conception et Fabrication Assistée par Ordinateur) LECTRA développe en parallèle :

- Des équipements de production : matelasseurs, machines de découpes...
- Des logiciels de conception et de gestion de production

Dans l'industrie textile, avant la phase de production proprement dite, deux notions sont très importantes :

- Le placement est la disposition des pièces d'un ou de plusieurs articles qui seront découpés ensemble. L'efficacité d'un placement représente la proportion de matière réellement utilisée, elle mesure la qualité de la disposition des pièces.
- L'ordre de fabrication centralise toutes les données nécessaires à la production. En effet, un ordre de fabrication fait bien sûr référence à différents articles à produire (vêtements, tissus automobiles, etc.) dans certaines quantités mais aussi aux tissus qui seront utilisés pour produire ces articles et à différents paramètres de la salle de coupe (coûts horaire, temps fixes, vitesse des machines...).

Le logiciel de LECTRA Optiplan permet d'organiser toutes ces données de manière cohérente et de planifier au mieux la phase de production. Cette planification peut être réalisée soit manuellement soit de manière automatique. Actuellement, le module de traitement automatique des ordres de fabrication d'Optiplan est basé sur l'expertise des gens du métier et sur la reproduction de leur façon de travailler. Les travaux qui sont décrits dans ce rapport proposent une approche plus mathématique du problème. Ils se décomposent en deux parties distinctes :

- La première partie est consacrée à la proposition d'un plan de production optimal pour chaque ordre de fabrication lorsque l'efficacité de chaque placement est supposée connue.

Le premier chapitre de cette partie est consacré à la description de ce qu'est un plan de production pour un ordre de fabrication, mais aussi au mode de traitement automatique actuel utilisé dans Optiplan. On verra dans le deuxième chapitre comment modéliser la construction des sections d'un ordre de fabrication comme un problème d'optimisation linéaire en nombres entiers.

La complexité de ce problème est étroitement liée au nombre de placements qu'il est éventuellement possible d'utiliser pour construire les sections. Si ce nombre est élevé, on choisira une méthode de résolution heuristique basée sur la méthode du recuit simulé qui est décrite brièvement dans le troisième chapitre. La sélection de ces placements « admissibles » peut être effectuée de différentes manières (exhaustives ou non), c'est l'objet du quatrième chapitre.

L'adaptation de l'algorithme du recuit simulé à la construction des sections est détaillée dans le cinquième chapitre, ainsi que le mode de résolution des sous-problèmes traités à chaque itération. Cependant, une des plus grandes difficultés de ce problème industriel est d'obtenir des solutions convenables en des temps limités. On doit adapter la méthode de résolution en scindant les ordres les plus complexes en plusieurs parties. Le sixième chapitre présente un moyen de réaliser cela en créant une partition de l'ensemble des tailles de l'ordre de fabrication. Le septième chapitre propose quant à lui une variation de la méthode de résolution lorsque des écarts sur les quantités à produire sont autorisés.

Alors que quelques exemples typiques sont commentés dans le huitième chapitre, on s'intéresse dans le neuvième chapitre à la qualité des solutions obtenues avec la méthode heuristique par rapport aux meilleures bornes fournies par les logiciels de résolution exacte.

La conclusion sur cette première partie constitue le dixième chapitre.

- La seconde partie traite de l'estimation de l'efficacité des placements non encore réalisés en fonction de leurs caractéristiques (modèle, nombre de tailles...)

Le premier chapitre de cette partie expose les liens entre l'efficacité inconnue des nouveaux placements et la construction des sections traitée dans la première partie. En effet, avoir une bonne estimation de l'efficacité des placements non réalisés permet que le coût des solutions proposées par la méthode d'optimisation de la première partie soit le plus proche possible du coût réel qui ne sera connu que lorsque les placements retenus seront effectivement créés.

Pour estimer l'efficacité des placements, on choisit de construire un modèle statistique en utilisant un historique de placements connus. La nature des variations d'efficacités (non linéaires et parfois brutales) nous ont conduit à utiliser pour cela les propriétés d'approximation des réseaux de neurones. Le deuxième chapitre constitue une brève présentation de l'utilisation de ceux-ci pour des problèmes d'approximation.

Dans notre étude, nous nous sommes essentiellement intéressés aux placements de pantalons et aux placements de chemises. Le troisième chapitre est consacré à la recherche des paramètres les plus pertinents pour expliquer les variations d'efficacités observées. On distinguera des paramètres généraux (globaux) présents aussi bien dans le cas des chemises que des pantalons et des paramètres spécifiques à chacun de ces deux types de placements. Cela nous a donc amené à construire deux modèles statistiques différents.

Le quatrième chapitre présente les premiers résultats obtenus pour chacun de ces deux modèles. On s'intéressera pour chacun d'eux à leur performance de généralisation.

La conclusion sur cette deuxième partie constitue le cinquième chapitre. On y parlera notamment du travail restant à effectuer pour obtenir une estimation efficace de l'efficacité des placements de tous types. En effet, cette partie en est encore au stade expérimental cependant, les premiers résultats obtenus sont très encourageants.

Première Partie

Résolution d'un ordre élémentaire de
fabrication à l'aide de la méthode du recuit
simulé

Chapitre 1 : Présentation du problème

1.1 Vocabulaire

La première partie de ce chapitre a pour but de familiariser le lecteur avec des termes spécifiques à l'industrie textile. Ces termes seront réutilisés ensuite tout au long du document, il est donc important de bien appréhender ce qu'ils représentent pour une bonne compréhension de la suite.

Article : Ensemble de pièces constituant une entité à produire. Un article peut être :

- Une taille d'un modèle de vêtement (pantalon en 42, chemise en 36...)
- Issu d'autres secteurs de l'industrie textile (housse de fauteuil, airbag...)
- ...

Tissu : Le tissu est la matière dans laquelle les pièces d'un article vont être découpées. Toutes les pièces d'un même article ne sont pas forcément découpées dans le même tissu, on parle alors de types de tissu différents (pour une veste on distingue le drap et la doublure par exemple) Le tissu se présente le plus souvent sous forme de rouleaux de longueurs variables.

Laize : Largeur d'un rouleau de tissu. Les placements utilisés pour découper ce tissu doivent donc avoir la même laize. Dans certains cas toutefois, de légères tolérances sont acceptées :

- Placement moins large que le tissu : Pertes de tissu supplémentaires
- Placement plus large que le tissu : Risque de pièces tronquées

Ordre de fabrication (également noté OF par la suite) : Un ordre de fabrication décrit ce que l'on veut produire. C'est à dire la quantité à découper pour chaque élément d'un ensemble d'articles et chaque élément d'un ensemble de matières.

Ordre de fabrication : Produire des vestes

	S	M	L
Rouge	20	12	10
Vert	10	15	15
Bleu	12	10	16

Figure 1 : Exemple d'un ordre de fabrication

Le tableau de la **Figure 1** est un exemple d'ordre de fabrication représenté sous forme de matrice. L'ensemble des articles correspond à 3 tailles différentes d'un modèle de veste, l'ensemble de matières correspond lui à l'ensemble de couleurs (rouge, vert, bleu) que l'on veut obtenir pour les vestes.

Ordre élémentaire : C'est l'ordre de fabrication réduit à un type de tissu. L'ordre élémentaire va permettre de découper une partie des pièces de chaque article de l'ordre de fabrication.

Ordre élémentaire : type tissu "Drap"

	S	M	L
DR1	20	12	10
DR2	10	15	15
DR3	12	10	16

Ordre élémentaire : type tissu "Doublure"

	S	M	L
DB	42	37	41

Figure 2 : Exemple de 2 ordres élémentaires composants un ordre de fabrication

La **Figure 2** représente les 2 ordres élémentaires de l'ordre de fabrication de la **Figure 1**. On remarque que pour l'ordre élémentaire « Doublure », il n'y a plus de notion de couleur (la doublure est la même quelle que soit la couleur de la veste), les 3 lignes de l'ordre de fabrication sont remplacées par une seule qui contient la somme des quantités à produire.

Taille : Un article correspond souvent à un modèle de vêtement dans une taille donnée (42,46...S,M...). Ce n'est pas toujours le cas dans l'industrie du textile automobile ou d'ameublement par exemple. Cependant, afin de simplifier les notations, on appellera taille tout article correspondant à une colonne dans la matrice d'un ordre élémentaire. Le nombre de tailles d'un ordre élémentaire sera alors égal au nombre de colonnes de la matrice des quantités à produire.

Placement : C'est la disposition d'un ensemble de pièces, provenant d'un (mono-taille) ou de plusieurs (multi-tailles) articles, pouvant être découpées dans un même type tissu. La découpe de la matière sera effectuée selon cette disposition qui doit limiter le plus possible les chutes.

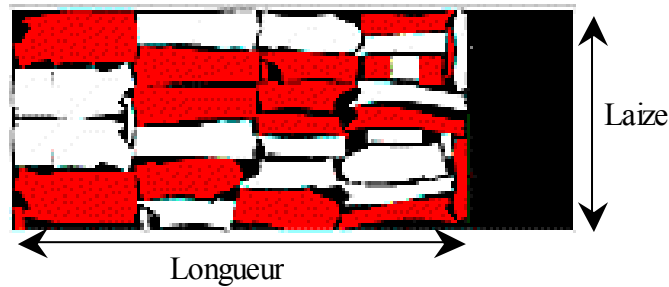


Figure 3 : Exemple de placement

Chutes : Quantité de matière perdue :

- Pour un placement : différence entre la surface utilisée et la surface des pièces découpées.
- Autres chutes possibles : pertes en bout, fin de rouleaux de tissu non utilisables.

Efficienc e d'un placement : C'est le pourcentage d'utilisation de la matière, le rapport entre la surface des pièces placées et la surface utilisée pour les placer :

$$Efficienc e(P) = 100 \times \frac{\sum_i Surface(pièce_i(P))}{Longueur(P) \times laize}$$

Pour limiter les chutes de matières on a donc intérêt à avoir l'efficienc e la plus grande possible pour chaque placement. Selon les domaines considérés, l'efficienc e d'un placement de bonne qualité se situe généralement entre 80 et 90%.

Plis : Représente une « couche » de tissu étalée au dessus d'un placement. Le nombre de plis découpés avec un placement donne le nombre de fois où sont découpés les articles contenus dans ce placement.

Section : Permet de découper une partie des articles d'un ordre élémentaire. Une section représente un placement ainsi que, pour chaque tissu de l'ordre élémentaire, le nombre de plis à découper suivant ce placement.

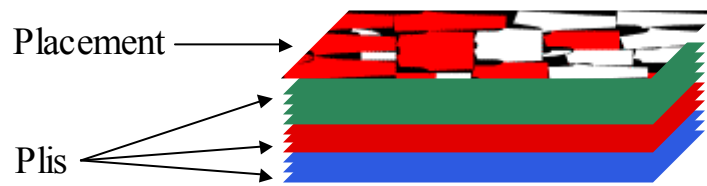


Figure 4 : Exemple de section

Solution de sections : Ensemble de sections permettant de découper toutes les quantités d'un ordre élémentaire.

Bordereau : (ou **bordereau de coupe**) Ancienne dénomination d'un ordre de fabrication ou d'un ordre élémentaire. On parlera encore de **résolution du bordereau** pour l'action de trouver une solution de sections à un ordre élémentaire.

Matelas : C'est l'« objet » qui est réellement découpé dans la salle de coupe. Un matelas est un ensemble de sections (éventuellement une seule section) « collées » les unes à la suite des autres et qui vont être découpées en même temps.

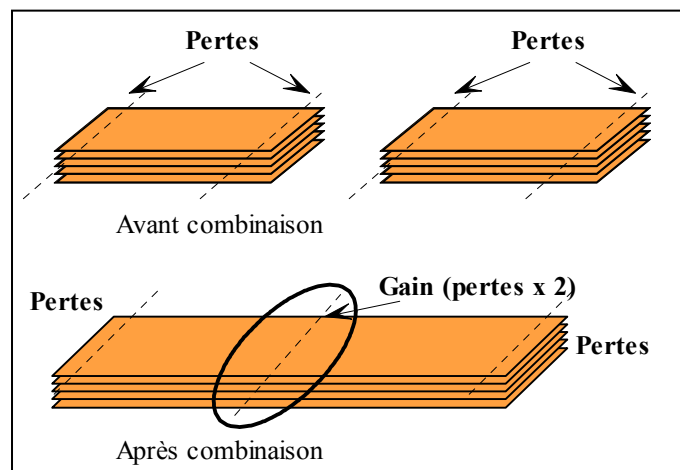


Figure 5 : Exemple de regroupement de 2 sections pour former un matelas

On doit rajouter à chaque extrémité d'un matelas quelques centimètres de tissu (appelés pertes en bout) destinés à fixer celui-ci pendant la découpe. Comme le montre la **Figure 5** le regroupement de plusieurs sections permet notamment de limiter ces pertes.

Plan de matelassage : Il découle d'une solution de sections. Il définit le regroupement des sections en différents matelas. Le plan de matelassage transféré à la salle de coupe permettra de découper toutes les quantités de l'ordre élémentaire correspondant.

1.2 Problématique

Comme le font apparaître les termes explicités ci-dessus, le traitement d'un ordre élémentaire se décompose principalement en 2 étapes. La première consiste à trouver une solution de sections permettant de découper tous les articles demandés dans la matrice de l'ordre élémentaire. Pour cela, on doit trouver un ensemble de placements ainsi que, pour chacun d'eux, le nombre de plis à réaliser dans chaque tissu. Cette phase du traitement est couramment nommée : la construction des sections. La seconde étape consiste à construire un plan de matelassage à partir de la solution de section trouvée précédemment. Chaque matelas correspond à une ou plusieurs sections que l'on regroupe pour limiter les chutes de tissu ou faciliter la découpe. Une fois cette phase de construction des matelas effectuée, le résultat peut directement être transféré dans la salle de découpe où la production proprement dite pourra commencer.

Le but du traitement automatique des ordres de fabrication est de trouver le meilleur plan de matelassage possible. La qualité d'un plan de matelassage dépend bien sûr de plusieurs critères dont voici les principaux :

- Le plan de matelassage doit être adapté à l'organisation de la salle de coupe (on parle de plan de matelassage réalisable)
- Respecter les stocks de tissus disponibles
- Utiliser le moins de matière possible
- Limiter au maximum le temps passé dans la salle de coupe

D'autres critères, spécifiques à chaque utilisateur, peuvent également avoir de l'importance. En règle générale, la problématique est de trouver la solution qui minimise le coût unitaire de l'article à découper tout en respectant les contraintes de la salle de coupe et les stocks disponibles.

En ce qui concerne le respect des stocks de tissus, on ne considère pour chacun d'entre eux que la longueur totale disponible. Cependant, cette longueur est souvent obtenue en additionnant les longueurs de plusieurs rouleaux du même tissu. Cela fait apparaître un autre problème qui est la gestion des fins de rouleaux. En effet, les longueurs totales disponibles ne sont réellement utilisables que si l'on parvient à utiliser tous les rouleaux de tissu au maximum. Ce problème ne sera pas traité ici pour deux raisons :

- La première est que la plupart des utilisateurs ne disposent pas d'informations précises sur la longueur de chacun des rouleaux de tissus.
- Lorsque ces informations existent, on préfère optimiser l'utilisation des rouleaux de tissu après la construction des matelas qui est elle-même postérieure à la construction des sections.

On pourrait imaginer traiter la construction des sections, la construction des matelas et la gestion des rouleaux de tissu ensemble car se sont les trois composantes d'un problème plus global. Cependant, on ne choisit pas cette approche car les problèmes que l'on aurait alors à traiter seraient beaucoup plus complexes et leur résolution serait peu envisageable en pratique.

L'utilisation du module de traitement automatique des OF d'Optiplan permet un gain de l'ordre de 2 à 5%¹ par rapport à un traitement manuel. Il est généralement convenu que 80% de ces gains sont obtenus durant la 1^{ière} phase du traitement des ordres qui est la construction des sections. Il est donc primordial dans le cadre d'une résolution automatique d'accorder un soin tout particulier à la qualité des solutions de sections obtenues car une amélioration moyenne de 1% des coûts des solutions de sections représente un gain de performance compris entre 16 et 40% pour Optiplan. C'est dans cette optique que le travail exposé ici a été réalisé. Son but est d'explorer une nouvelle façon d'aborder le problème de la construction des sections. Les paragraphes suivants présentent plus en détail le problème à résoudre puis le mode de résolution actuel utilisé dans Optiplan.

¹ Le gain réel peut dépendre de l'organisation de la salle de coupe, du type d'articles découpés, mais également du talent et de l'expérience de l'opérateur qui réalise le traitement manuel.

1.3 Exemple de construction de sections

Le petit exemple suivant permet de se familiariser avec les données que l'on manipule lors de la construction des sections mais également de donner un premier aperçu du traitement du problème sans pour autant introduire encore la notion d'optimisation.

On suppose que l'on veut résoudre (trouver une solution de sections) pour l'ordre élémentaire de la **Figure 6**. Cet OF très simple correspond à une demande d'un modèle de pantalon nommé « trouser » dans 4 tailles (S, M, L et XL) et 2 tissus (Blue et Red). Le nombre total de pantalons à découper est 70.

Ordre élémentaire 01					
Trouser	S	M	L	XL	Total
Blue	5	15	20	10	50
Red	0	5	10	5	20
Total	5	20	30	15	70

Figure 6 : Exemple d'ordre élémentaire

On suppose également que l'on dispose déjà d'un placement¹ pouvant être utilisé pour résoudre cet OF. Ce placement décrit dans la **Figure 7** contient une fois les tailles S et XL et 2 fois les tailles M et L. Cela signifie qu'à chaque fois que l'on découpe un pli avec **P1**, 6 pantalons (1S, 2M, 2L et 1XL) sont découpés².

Placement existant				
Nom	S	M	L	XL
P1	1	2	2	1

Figure 7 : Liste des placements existants

¹ Provenant par exemple d'une résolution antérieure d'un ordre contenant ce modèle.

² **P1** est appelé placement 6-tailles.

On choisit dans un premier temps d'utiliser P1. En effet, la création de nouveaux placements représente un coût non négligeable et on a souvent intérêt à utiliser le plus possible les placements existants. Pour chaque tissu de l'ordre élémentaire, on peut calculer le nombre maximum de plis que l'on peut découper avec P1. Pour le tissu bleu et la taille S par exemple, on divise la quantité demandée (5) par le nombre de répétitions de cette taille dans P1¹ (1) : on ne peut pas découper plus de 5 plis dans le tissu bleu avec P1 sous peine de dépasser la quantité demandée pour la taille S. On réitère ce processus pour toutes les tailles et on garde le plus petit quotient.

Ordre élémentaire 01					
Trouser	S	M	L	XL	Total
Blue	5	15	20	10	50
Red	0	5	10	5	20
Total	5	20	30	15	70

Figure 8 : Calcul du nombre de plis possibles avec P1

La Figure 8 montre que le nombre de plis maximum que l'on peut découper avec P1 est :

- 5 pour le tissu bleu
- 0 pour le tissu rouge (pas de demande pour la taille S et le tissu rouge)

On formalise ce calcul en écrivant :

$$NbMaxPlis(P, j) = \underset{\substack{i \in \{tailles\} \\ p_i > 0}}{\text{Min}} \left(\left\lfloor \frac{B_{ij}}{p_i} \right\rfloor \right)$$

où :

P	représente un placement quelconque
j	le j ème tissu de l'ordre
i	la i ème taille de l'ordre
p_i	le nombre de répétitions de la taille i sur P
B_{ij}	la quantité d'articles demandée pour la taille i et le tissu j

¹ Si le nombre de répétitions est supérieur à 0. Dans le cas contraire, on ignore la taille correspondante.

On propose de découper le maximum de plis possibles (ce n'est pas obligatoire) avec P1. On obtient alors une première section puis en soustrayant le nombre de pantalons découpés avec P1 des quantités de l'OF, on obtient la matrice des quantités restant à découper (cf. **Figure 9**).

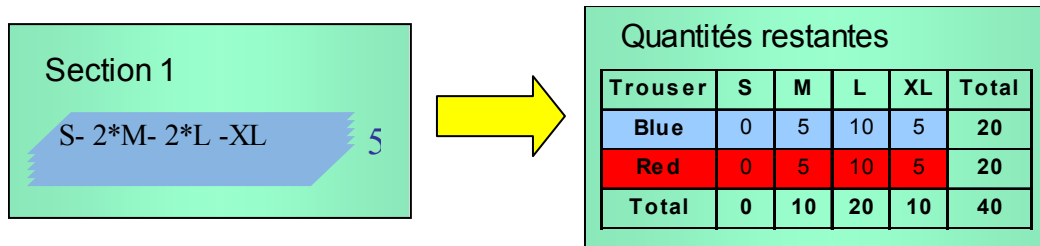


Figure 9 : Calcul des quantités restant à découper en fonction de la section 1

Pour continuer la résolution de l'OF, il faut maintenant utiliser un nouveau placement. On remarque que les quantités à produire sont les mêmes pour les 2 tissus. De plus, les quantités demandées pour la taille L sont exactement le double de celles demandées pour les tailles M et XL. On propose donc de créer le placement P2 (cf. **Figure 10**) contenant une fois les tailles M et XL et 2 fois la taille L, il permet de découper tous les pantalons restants avec 5 plis de bleu et 5 plis de rouge.

Proposition de nouveau placement				
Nom	S	M	L	XL
P2	0	1	2	1

Figure 10 : Nouveau placement à créer

La section 2 représentée sur la **Figure 11** permet d'annuler la matrice des restes, la construction de sections de l'ordre élémentaire est donc achevée.

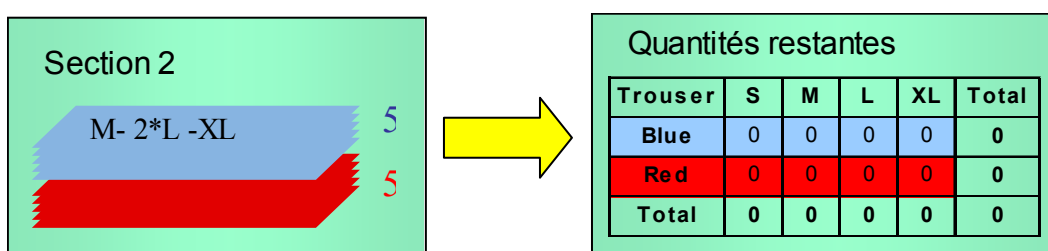


Figure 11 : La section 2 permet de découper toutes les quantités restantes

La **Figure 12** permet de résumer la solution de section que l'on vient de trouver pour cet ordre élémentaire. Cependant, à aucun moment, on ne s'est posé la question de savoir si la solution que l'on construisait satisfaisait d'éventuelles contraintes de production ou si son coût était acceptable.

		Plies	
		Blue	Red
Section 1	P1	5	0
Section 2	P2	5	5

Figure 12 : Résumé de la solution de sections

Les différents coûts et contraintes éventuels sont décrits dans les paragraphes suivants.

1.4 Les coûts

L'objectif de la construction de sections automatique est de fournir une solution de coût le plus faible possible à un ordre élémentaire. Cependant, le coût d'une solution de sections dépend le plus souvent de plusieurs paramètres qui sont parfois antagonistes. Voici le détail des différents types de coûts.

1.4.1 Les coûts matières

Ce sont les coûts liés à l'utilisation du tissu, ils correspondent au prix total de tout le tissu utilisé dans la solution de sections. Pour les calculer, il faut connaître pour chaque tissu, son prix au mètre et la longueur à utiliser. Cette longueur dépend des placements que l'on utilise et du nombre de plis découpés avec chacun d'entre eux. Pour réduire ces coûts, il faut utiliser des placements efficaces afin de perdre le moins de tissu possible. Dans la plupart des cas, les placements les plus efficaces sont ceux qui contiennent le plus de tailles, on a donc intérêt pour réduire les coûts matières à utiliser de longs placements avec peu de plis.

1.4.2 Les coûts de matelassage

L'action d'étaler le tissu au-dessus d'un placement pour former une pile de plis s'appelle le matelassage. En effet, un coût horaire de matelassage est déterminé dès le départ et le coût de

matelassage sera ce prix de l'heure multiplié par le temps de matelassage qui dépend de 2 paramètres :

- Un temps fixe (préparation, installation, etc...) qui est compté à chaque section.
- La vitesse de matelassage, c'est à dire la vitesse à laquelle la machine étale le tissu au-dessus du placement

Le coût de matelassage dépend donc du nombre de sections dans la solution (le temps fixe est compté autant de fois qu'il y a de sections) mais surtout de la longueur totale de tissu à étaler. Comme pour les coûts matières, on a intérêt à utiliser des placements longs avec peu de plis.

1.4.3 Les coûts de coupe

Ces coûts correspondent à l'action de découper les pièces de tissu suivant la disposition décrite par les placements utilisés. Comme pour le matelassage, c'est le temps de coupe qui est mesuré puis multiplié par le coût horaire de coupe. Le temps de coupe dépend lui aussi de 2 paramètres :

- Un temps fixe (préparation, installation, etc...) qui est compté à chaque section.
- La vitesse de coupe, c'est à dire la vitesse à laquelle la machine découpe le tissu au-dessus du placement.

Le coût de coupe dépend d'une part du nombre de sections dans la solution (le temps fixe est compté autant de fois qu'il y a de sections) et de la longueur que la lame (ou le laser) de la machine de coupe doit parcourir d'autre part. On calcule la longueur de coupe de chaque placement utilisé en additionnant le périmètre de toutes les pièces qu'il contient. Pour réduire ces coûts, on a donc intérêt à découper le plus d'articles possible en même temps en utilisant des placements courts avec beaucoup de plis.

1.4.4 Les coûts placements

Lorsque la solution de sections utilise des placements nouveaux, il faut les réaliser pour pouvoir appliquer cette solution. Le coût de création d'un placement dépend principalement de sa complexité, c'est à dire du nombre de tailles qu'il contient, mais aussi d'un coût fixe

associé à la mise en route de la procédure de création de placement. On a ici un dilemme car la création d'un placement contenant beaucoup de tailles coûte cher mais permet le plus souvent d'obtenir une meilleure efficacité et donc de réduire les coûts matières et de matelassage. Cependant, dans la plupart des cas l'habitude des professionnels est d'utiliser les placements existants s'il y en a.

1.4.5 Les coûts d'évacuation

Ces coûts correspondent à l'extraction et à la mise en paquets des pièces découpées. En effet, une fois la coupe effectuée (ou même au fur et à mesure de la coupe), les pièces sont retirées de la machine de coupe puis regroupées par paquets (un paquet contient plusieurs pièces identiques découpées dans un même tissu). Là encore c'est le temps d'évacuation qui va déterminer le prix de l'évacuation en fonction d'un coût horaire. Pour réduire ces coûts, on a intérêt à utiliser des placements contenant peu de tailles (peu de pièces découpées) avec un grand nombre de plis qui sera si possible un multiple du nombre de pièces par paquet.

1.4.6 Remarques générales sur les coûts

1.4.6.1 Les vitesses

Pour calculer les temps de matelassage et de coupe, on utilise les vitesses de matelassage et de coupe. Ces paramètres sont des constantes fournies par l'utilisateur et représentent des vitesses moyennes pour le matelasseur et le découpeur. Dans la réalité, la vitesse n'est pas constante, les machines pour une distance donnée à parcourir passent par une phase d'accélération, une phase de maintien de la vitesse et une phase de décélération. Dans le cas du matelassage par exemple, l'utilisation de longs placements permet de maintenir plus longtemps une vitesse élevée. Dans le cas de la coupe, on ne tient pas compte de l'épaisseur de tissu à découper. Pourtant un grand nombre de plis fait diminuer la vitesse de coupe.

1.4.6.2 Importance relative des coûts

Les paragraphes précédents montrent que le coût d'une solution de section se décompose en plusieurs parties (prix du tissu, coûts de coupe, création des placements...). Certains de ces

coûts sont antagonistes et réduire les uns conduit à augmenter les autres et vice versa. De plus, ils n'ont pas la même importance dans le coût total d'une solution. Comme le montre la **Figure 13**, les coûts matières représentent le plus souvent entre 80 et 90% du prix total mais dépendent beaucoup du prix des tissus employés car ceux-ci peuvent varier dans des proportions très importantes selon les cas. Les coûts de placement et de coupe peuvent être très variables suivant les méthodes de création de placements et de découpe utilisées.

On peut déjà voir se dégager 2 grandes orientations pour le choix d'une solution :

- Si les tissus utilisés sont très chers, on aura tendance à créer les placements les plus efficaces et les plus adaptés aux quantités de l'OF pour économiser au maximum les coûts matières.
- Si les tissus utilisés sont peu chers mais que la création des placements est difficile, on essaiera d'utiliser au mieux les placements existants et, de créer si nécessaire des placements simples (mono-taille par exemple) pour réduire les coûts placements et les coûts de coupe.

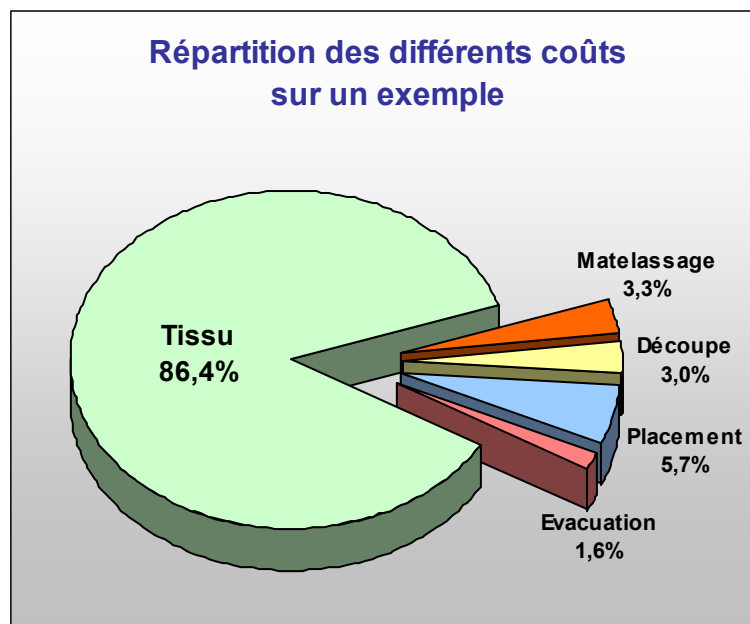


Figure 13 : Importance des coûts matière dans la répartition des coûts

Bien que les coûts matières soient en général beaucoup plus élevés que les autres coûts, il n'est pas toujours bon de se focaliser uniquement sur eux. En effet, une attention particulière à

la qualité des placements utilisés peut permettre une économie de quelques pourcents sur le tissu mais dans le même temps faire doubler les coûts de coupe et de placement car la solution devient plus complexe.

C'est toute la difficulté de ce problème qui repose sur des équilibres entre les coûts très instables. Une résolution efficace devient alors très rapidement impossible manuellement dès que la taille de l'ordre de fabrication augmente.

1.5 Les contraintes

1.5.1 Découper les quantités souhaitées

C'est bien sûr la première contrainte à respecter. La solution de sections doit permettre de découper toutes les quantités de la matrice de l'ordre élémentaire. On verra cependant (cf. *Chapitre 7 : Résolution avec déviations sur les quantités*) qu'il est envisageable dans certains cas de découper un peu plus ou un peu moins que les quantités souhaitées.

1.5.2 Sur la création des placements

Ces contraintes sont les plus nombreuses, elles limitent le choix de propositions de création de nouveaux placements.

1.5.2.1 Création ou non de nouveaux placements ?

Cette contrainte est bien sûr la plus restrictive. On peut effectivement choisir de résoudre l'ordre de fabrication avec uniquement les placements déjà existants. Il peut toutefois arriver que les placements existants ne permettent pas de découper toutes les quantités demandées (cf. *1.3 : Exemple de construction de sections*), si l'on souhaite tout de même obtenir une solution réalisable il faut autoriser la création des placements mono-tailles. En effet, ce sont les plus simples et ils permettent toujours de découper toutes les quantités de l'OF pour une taille donnée. Malheureusement, ce sont rarement les meilleurs placements à utiliser.

1.5.2.2 Limitation des combinaisons entre tailles

Dans un ordre élémentaire, chaque colonne de la matrice correspond à un article à produire.¹ Il peut arriver qu'on ne souhaite pas mélanger tels ou tels articles dans un même placement. Pour un vêtement par exemple on distingue les notions de modèle, variantes et sous-échelle de tailles (un modèle peut contenir plusieurs variantes qui peuvent elles-même contenir plusieurs sous-échelles de tailles). On peut alors imposer les contraintes suivantes :

- Pas de limitation (on peut mélanger n'importe quelle taille dans un même placement)
- Limitation dans un modèle (les placements créés contiennent des tailles d'un même modèle)
- Limitation dans une variante (les placements créés contiennent des tailles d'une même variante)
- Limitation dans une sous-échelle de tailles ((les placements créés contiennent des tailles d'une même sous-échelle de tailles)

1.5.2.3 Nombre maximum de tailles dans un placement

Comme son nom l'indique, cette contrainte fixe le nombre maximum de tailles que peut contenir un placement. D'un point de vue industriel, elle limite la longueur des placements utilisés, d'un point de vue plus mathématique, elle réduit le nombre de solutions réalisables pour l'ordre élémentaire.

1.5.2.4 Autres contraintes

Des contraintes supplémentaires peuvent parfois être activées. Elles résultent le plus souvent de connaissances empiriques et permettent dans certains cas spécifiques de trouver plus facilement les meilleures solutions. En voici quelques-unes :

- Nombre maximum de répétition d'une même taille dans un placement
- Nombre minimum de tailles dans un placement

- Nombre minimum de répétition d'une même taille dans un placement
- Nombre de tailles dans un placement multiple d'une certaine valeur
- Nombre de répétition d'une même taille dans un placement multiple d'une certaine valeur

1.5.3 Contraintes de la salle de coupe

1.5.3.1 Nombre maximum de plis par section

Cette contrainte est très importante, elle provient directement des performances des découpeurs. En effet, une machine de coupe peut découper au maximum une certaine hauteur de tissu qui se convertit facilement en nombre de plis suivant le tissu que l'on découpe. On peut utiliser un placement pour un nombre de plis plus important que ce nombre maximum, mais on est alors obligé de scinder ce nombre de plis en autant de sections que nécessaire pour respecter la contrainte.² Dans une solution de section on peut donc avoir plusieurs sections associées au même placement.

1.5.3.2 Longueur maximale d'une section

La longueur d'une section doit être inférieure à la longueur disponible sur le matelasseur. Cette contrainte revient donc à limiter la longueur des placements (cf. 1.5.2.3 : *Nombre maximum de tailles dans un placement*) par une valeur numérique précise et non pas par un nombre de tailles maximum.

1.5.3.3 Nombre minimum de plis pour l'utilisation d'un placement

Cette contrainte est également le fruit de connaissances empiriques. L'expérience montre en effet qu'il est peu intéressant d'utiliser un long placement (même très efficace) pour un

¹ Rappel : Les articles correspondant le plus souvent à une taille d'un modèle de vêtement, on appelle « taille » toute colonne de la matrice de l'ordre élémentaire.

² Par exemple, si le nombre maximum de plis dans une section est fixé à 30, on peut découper 70 plis avec un même placement si on utilise 3 sections (30, 30 et 10 ou bien 25, 25 et 20...)

nombre de plis très faible. Dans un tel cas, les coûts de création du placement et de coupe engendrés sont le plus souvent supérieurs aux économies de tissu réalisées.

On fixe alors pour chaque placement un nombre minimum de plis à découper pour pouvoir l'utiliser. Ce nombre minimum dépend du nombre de tailles que le placement contient.

Supposons que le nombre maximum de tailles dans un placement soit fixé à 6. La **Figure 14** montre un exemple de table qui régleme l'utilisation des placements en fonction du nombre de tailles qu'ils contiennent.

Nombre de tailles dans le placement	Nombre minimum de plis
6	30
5	25
4	20
3	15
2	10
1	1 (non contraint)

Figure 14 : Nombre minimal de plis

1.5.4 Respect des stocks de tissu

La solution de sections doit respecter les stocks de tissu disponibles. Si les stocks ne paraissent pas suffisants, l'utilisateur peut tout de même choisir de procéder à la résolution en ne tenant plus compte de cette contrainte.

De plus, il peut arriver que pour un tissu demandé dans l'ordre de fabrication, on dispose en stock de plusieurs tissus susceptibles de convenir mais dans des laizes et des quantités différentes¹. Si tel est le cas, la solution devra proposer un choix entre les différents tissus disponibles pour chaque tissu demandé de l'OF.

¹ Dans un ordre on peut demander telle ou telle nuance de rouge par exemple. Il est possible que dans les stocks disponibles il y ait plusieurs rouleaux de tissus de cette couleur mais dans des laizes différentes.

1.6 Longueur des nouveaux placements

On l'a dit, la construction de sections implique la plupart du temps la proposition de création de nouveaux placements. Or, pour calculer le coût de la solution de sections, on doit notamment connaître la longueur totale de tissu utilisée et donc la longueur des nouveaux placements. Cependant, comme ceux-ci n'ont pas encore été réalisés, on ne peut pas connaître leur longueur. Pour palier ce problème, deux solutions sont envisageables dans un premier temps :

1. Réaliser réellement chaque nouveau placement proposé en cours de résolution, on disposerait alors de la vraie longueur du placement.
2. Estimer la longueur de chaque nouveau placement proposé en fonction de règles définies au préalable.

La première solution n'est pas réalisable en pratique. En effet, le processus de réalisation d'un placement, qu'il soit manuel ou automatique¹, peut être long². Le but de la construction de sections est de trouver les placements les plus adaptés pour découper les quantités demandées dans l'OF. Il est donc important de pouvoir « tester » le plus de placements possibles. Dans ces conditions, il n'est pas envisageable, pour des raisons de temps de résolution, de réaliser réellement chaque placement qui pourrait être éventuellement utilisé dans la solution de sections.

On retient donc la deuxième possibilité qui consiste à estimer la longueur de chaque nouveau placement. Comme la superficie des pièces des articles à produire est connue, lorsqu'on propose la création d'un nouveau placement, on peut facilement calculer la superficie totale des pièces qu'il doit contenir. On peut donc calculer la longueur qu'aurait ce placement si la laize du tissu est connue et si l'on suppose qu'il n'y a aucune perte de tissu. Pour obtenir la

¹ Plusieurs logiciels de placement automatique existent sur le marché. Celui de LECTRA s'appelle Diamino Expert.

² La durée dépend de la complexité du placement (c'est à dire du nombre total de pièces à placer). Elle peut varier (dans le cas du placement automatique) de quelques minutes à plusieurs heures dans les cas les plus critiques.

longueur réelle du placement, il faut multiplier cette quantité par un facteur supérieur à 1 qui est en fait l'inverse de l'efficacité :

$$L_{100\%}(P) = \frac{\sum_i Surface(pièce_i(P))}{laize} \quad \text{et} \quad Efficiency(P) = \frac{\sum_i Surface(pièce_i(P))}{Longueur(P) \times laize}$$

$$\Rightarrow Longueur(P) = \frac{\sum_i Surface(pièce_i(P))}{laize \times Efficiency(P)} = L_{100\%}(P) \times \frac{1}{Efficiency(P)}$$

On choisit finalement d'estimer l'efficacité (c'est à dire la qualité) des placements proposés puis d'en déduire alors une estimation de la longueur.

Le critère actuel de l'estimation de l'efficacité est le nombre de tailles contenues dans les placements. En effet, on sait de façon empirique que, plus un placement contient de tailles (de pièces), plus il est efficace (ce n'est pas une règle absolue mais une tendance générale qui se vérifie la plupart du temps). L'utilisateur est alors amené à remplir des tables d'efficacités qui vont fournir pour un nombre de tailles donné une certaine efficacité.

Répétition de taille	1
1 Taille	67,13 %
2 Tailles	73,39 %
3 Tailles	75,00 %
4 Tailles	77,00 %
5 Tailles	79,00 %
6 Tailles	80,00 %
7 Tailles	81,00 %

Figure 15 : Exemple de table d'efficacité

Dans l'exemple de la **Figure 15**, les deux premières valeurs (pour les placements mono-taille et 2-tailles) ont pu être obtenues en calculant les moyennes des efficacités sur des placements existants comparables (contenant le même modèle par exemple). Les autres valeurs sont des estimations empiriques.

Il est clair que cette manière d'estimer l'efficacité (et donc la longueur) des nouveaux placements est loin d'être satisfaisante dans tous les cas. On observe fréquemment de gros

écart entre l'efficacité réelle d'un placement une fois celui-ci réalisé et la valeur correspondante dans la table d'efficacité.

La deuxième partie de ce travail est entièrement consacrée à la recherche d'une technique plus précise d'estimation de l'efficacité des nouveaux placements.

1.7 Traitement actuel

1.7.1 Résolution pas à pas

Le traitement actuel de la construction automatique des sections dans Optiplan s'inspire largement de la résolution manuelle.

La technique mise en place est un algorithme itératif qui, à chaque étape, réalise le processus suivant :

- Trouver le « meilleur » placement à utiliser suivant les quantités restant à découper.
- Utiliser ce placement pour un certain nombre de plis.
- Calculer les quantités restant à découper

L'algorithme s'arrête lorsque toutes les quantités demandées dans l'OF sont découpées. A chaque étape, on construit donc une section (ou éventuellement plusieurs si le nombre de plis est trop important) associée au « meilleur » placement au regard des quantités restantes. C'est un algorithme glouton qui ne permet pas de « retour en arrière » si le choix d'un placement se révèle peu judicieux.

1.7.2 Choix du meilleur placement

A chaque étape de la résolution, il y a donc 2 choix à faire :

- Trouver le placement que l'on va utiliser
- Déterminer pour chaque tissu le nombre de plis à découper avec celui-ci

Plusieurs critères sont utilisés pour choisir le meilleur placement à chaque itération. Le premier est le nombre de tailles, on cherche en priorité un placement contenant le plus de tailles possibles¹. Parmi ceux-ci, on cherche d'abord à utiliser des placements existants, s'il n'y en a pas, on cherche une nouvelle combinaison de tailles qui satisfait les contraintes de création de placements. On calcule le nombre total de plis que l'on peut découper avec cette combinaison de tailles, puis le coût par article découpé que l'on obtiendrait avec ce nombre total de plis. On garde alors le placement qui permet le coût le plus faible. Si on n'a trouvé aucun placement utilisable pour le nombre maximum de tailles, on cherche alors un placement contenant une taille de moins. On réitère l'opération jusqu'à trouver un placement qui convienne².

En ce qui concerne la détermination du nombre de plis, l'utilisateur doit choisir entre 3 stratégies :

1. Découper le nombre maximum de plis par section
2. Découper un multiple du nombre maximum de plis par section
3. Découper autant de plis que possible

Par exemple, si on peut découper au maximum 75 plis pour un placement P donné, et que le nombre maximum de plis autorisé dans une section est 30. Le choix des différentes stratégies implique les actions suivantes :

1. On découpe seulement 30 plis et on construit 1 seule section avec le placement P
2. On découpe 60 plis et on construit 2 sections avec le placement P
3. On découpe 75 plis et on construit 3 sections avec le placement P

1.7.3 Analyse des solutions obtenues

L'avantage de cette méthode de résolution est sa rapidité. En effet, le temps de la construction automatique des sections varie de quelques secondes à quelques dizaines de secondes pour les

¹ Ce nombre est fixé par la contrainte du nombre maximum de tailles dans un placement

² Si aucun placement multi-tailles ne convient, on utilisera un placement mono-taille ce qui est toujours possible.

ordres de fabrication les plus importants. De plus, cette méthode repose sur l'expertise et l'expérience de personnes connaissant très bien cette problématique et permet souvent d'obtenir des solutions assez proches de l'optimum.

L'inconvénient principal de cette méthode est qu'elle repose sur des critères d'optimisation locaux. En effet, à chaque itération, on cherche le meilleur placement à utiliser sans tenir compte des placements que l'on pourrait ou non utiliser pour les itérations suivantes.

Un exemple classique de ce problème est illustré par le très célèbre problème d'optimisation dit du « voyageur de commerce ». Le but du voyageur de commerce est de visiter un ensemble de villes en parcourant le moins de distance possible. On s'aperçoit très rapidement que la technique consistant à, partant d'une ville donnée, visiter la ville la plus proche de la ville où l'on se trouve est une mauvaise technique.

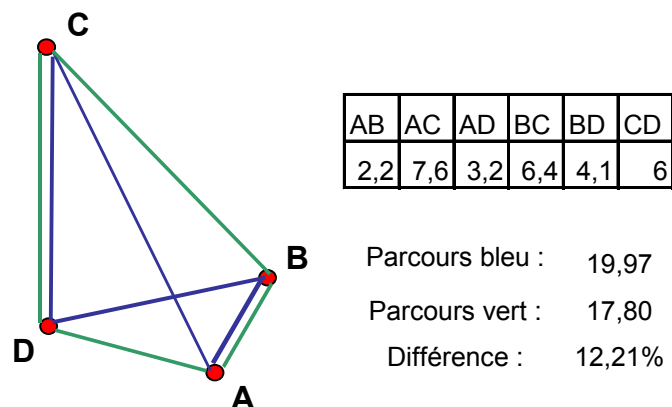


Figure 16 : Problème du voyageur de commerce

La **Figure 16** illustre un problème du voyageur de commerce à 4 villes. Le tracé bleu correspond au parcours obtenu en partant de la ville A et en choisissant à chaque fois la ville la plus proche. Le tracé vert correspond lui au parcours le plus court.

Dans le cas de la construction de sections, si de manière analogue on choisit à chaque étape le meilleur placement à utiliser, on se rend compte que le choix des premiers placements conditionne très fortement tout le reste de la solution. Le graphique de la **Figure 17** représente sur un exemple, le prix par article découpé ainsi que le pourcentage de toutes les quantités de l'OF pour chaque section.

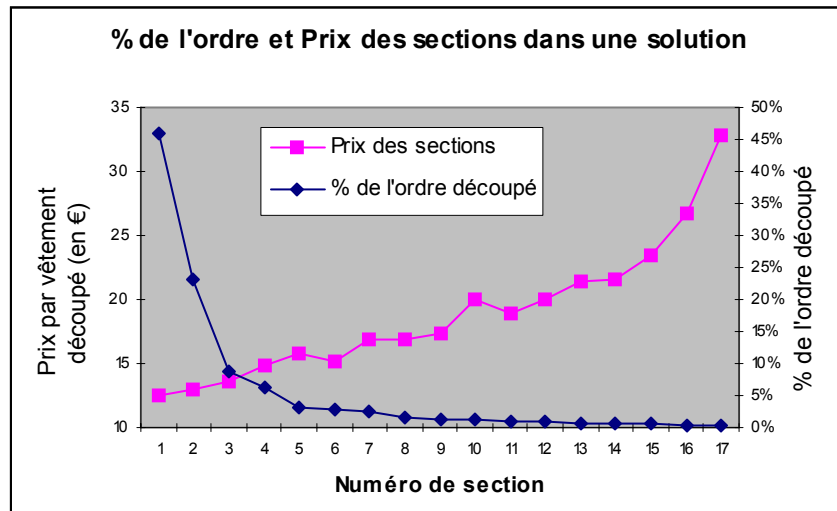


Figure 17 : Influence des premiers placements sur la solution

Les 3 premiers placements permettent de découper près de 80% des quantités totales demandées dans l'OF. Le reste est découpé avec 14 autres placements ! A l'inverse, en ce qui concerne le coût par article découpé, il est faible pour les premiers placements et devient très important pour les derniers. En effet, une fois décomptées les quantités découpées avec les premiers placements, les quantités restant à produire sont le plus souvent dispersées dans la matrice des quantités. Pour les découper, on est alors obligé d'utiliser de nombreux placements avec de faibles quantités (82% des placements sont utilisés pour découper 23% des quantités totales) ce qui implique des coûts unitaires élevés (impossibilité d'utiliser les placements contenant le plus de tailles et qui sont les plus efficaces par exemple).

La façon de déterminer le nombre de plis lorsque l'on décide d'utiliser un placement peut également poser des problèmes. La Figure 18 montre un ordre élémentaire très simple qui peut se résoudre avec les placements P1 et P2 avec respectivement 53 et 17 plis.

	S	M	L	XL
Bleu	70	123	70	87

	S	M	L	XL
P1	1	2	1	1
P2	1	1	1	2

Nb Plis
53
17

Figure 18 : Exemple d'ordre et sa solution avec 2 sections

La **Figure 19** représente les solutions que l'on obtient avec Optiplan suivant les stratégies adoptées pour déterminer le nombre de plis. Le nombre maximum de plis par section étant fixé à 15, les stratégies 1 & 2 (cf. *Découper un multiple du nombre maximum de plis par section*) proposent d'utiliser P1 dans 4 sections pour 60 plis au total. La stratégie 3 utilise P1 au maximum avec 61 plis (donc 5 sections).

		S	M	L	XL		
Stratégie 1 & 2	P1	1	2	1	1	Nb Plis	
	P3	1	0	1	2	60	
	P4	0	1	0	2	10	
	P5	0	0	0	1	3	
						1	
		S	M	L	XL		
Stratégie 3	P1	1	2	1	1	Nb Plis	
	P3	1	0	1	2	61	
	P6	0	0	0	2	9	
	P7	0	1	0	0	4	
						1	

Figure 19 : Solution obtenue avec Optiplan suivant la stratégie retenue

Dans les 2 cas, les quantités restantes obligent à utiliser 3 autres placements pour découper les autres articles. De plus, les 3 placements utilisés contiennent moins de tailles (4, 3 et 1 ou 4, 2 et 1) que P2 et sont donc moins efficaces.

Le problème est qu'aucune des 3 stratégies possibles ne permet d'utiliser P1 avec 53 plis pour pouvoir utiliser P2 et terminer ainsi la résolution.

Ces quelques exemples montrent les limites d'une résolution basée sur des critères d'optimisation locaux qui peuvent compromettre la qualité du reste de la solution. Dans les chapitres qui suivent, on décrit une nouvelle méthode de construction automatique des sections. Le but recherché est d'obtenir de meilleurs résultats par une approche plus globale du problème.

Chapitre 2 : Modélisation mathématique

Le but de ce chapitre est de formuler le problème de la construction de sections comme un problème d'optimisation linéaire en nombres entiers.

2.1 Notations, satisfaction de l'OF et variables X

Dans toute la suite, on appelle B la matrice de l'ordre élémentaire contenant les quantités à découper, $nTissus$ et $nTailles$ respectivement le nombre de lignes et le nombre de colonnes de B .

On représente un placement par un vecteur $p = {}^t(p_1, \dots, p_i, \dots, p_{nTailles})$ à $nTailles$ coordonnées où p_i est le nombre de répétitions de la taille i sur $p \quad \forall 1 \leq i \leq nTailles$.

On note P l'ensemble de tous les placements, existants ou satisfaisant les contraintes de création, éventuellement utilisables pour résoudre l'ordre de fabrication. P sera appelée la bibliothèque complète des placements, elle sera supposée connue dans un premier temps. On verra en détails comment l'obtenir dans le Chapitre 4 : *Méthodes de recherche des placements*.

Pour chaque tissu demandé dans l'OF (pour chaque ligne de B), on dispose d'un certain nombre de tissus disponibles en stocks, de laizes différentes.

On est amené alors à définir les quantités suivantes:

- $\forall j \in \{1 \dots nTissus\}$, $nLaize(j) =$ le nombre de tissus (donc de laizes différentes) disponibles en stock qui correspondent au $j^{\text{ème}}$ tissu de l'OF.
- $\forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\}$, $Laize(j, s) =$ la laize du $s^{\text{ième}}$ tissu disponible pour le $j^{\text{ème}}$ tissu de l'OF.

- $\forall p \in P, \quad laize(p) = \text{la laize du placement } p \text{ (on peut utiliser } p \text{ avec des tissus dont la laize est supérieure ou égale à la laize de } p \text{)}$

Ainsi, si l'on suppose que la bibliothèque P est connue, il faut déterminer pour chaque tissu disponible le nombre de plis que l'on va découper avec chaque placement de P . On définit donc les variables X du problème par :

$$\forall p \in P, \forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\},$$

$$X(j, s, p) = \text{le nombre de plis à découper avec le placement } p, \text{ dans le } s^{\text{ième}} \text{ tissu disponible pour le } j^{\text{ième}} \text{ tissu de l'ordre (noté tissu } (j, s) \text{)}.$$

Il est possible que pour un tissu donné, certains des placements de P ne soient pas utilisables (laize du placement trop grande, trop petite, etc...). On définit pour chaque tissu disponible en stock, l'ensemble des placements admissibles par :

$$\forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\},$$

$$A(j, s) = \{p \in P / p \text{ utilisable pour le tissu } (j, s)\}$$

Découper le nombre d'articles souhaités dans la taille i et le tissu j de l'OF s'écrit alors :

$$\forall i \in \{1 \dots nTailles\}, \quad \forall j \in \{1 \dots nTissus\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in A(j,s)} \{p_i \times X(j, s, p)\} \right\} = B_{ji}$$

2.2 Fonction économique, autres variables

On a vu que la fonction économique se décompose en plusieurs parties. Les coûts liés à l'utilisation du tissu (coûts matières) dépendant des différents placements choisis mais aussi du tissu découpé et de la laize de ce tissu (différence de prix entre les tissus). Les coûts de matelassage, les coûts de coupe, les coûts d'évacuation et les coûts dus à la création et à l'utilisation des placements sont les autres types de coûts à prendre en compte.

Ces différents coûts n'agissent pas de la même façon sur la valeur totale de la fonction économique. En effet, certains sont comptés à chaque pli (matière, matelassage...), d'autres

sont comptés à chaque section (coupe, coûts fixes...), lors de l'utilisation ou non d'un placement (coûts de création) et même en fonction du nombre de paquets lors de l'évacuation. A chaque type de coût, on fait correspondre des variables différentes. On admet dans cette modélisation que le coût d'une solution est une fonction déterministe des variables que l'on définit et que les différents coûts s'expriment linéairement par rapport à ces variables. Dans la pratique cela n'est pas tout à fait exact cependant cette approximation est proche de la réalité et permet de ne pas complexifier outre mesure le problème.

2.2.1 Définition des variables

2.2.1.1 Variables $X(j, s, p)$

Ces variables ont été définies dans le paragraphe précédent car ce sont elles qui permettent de calculer les quantités découpées.

$X(j, s, p)$ est le nombre de plis que l'on découpe avec le placement p , dans le tissu (j, s) . Tous les coûts qui sont comptés pli par pli dépendent de ces variables dans la fonction économique.

2.2.1.2 Variables $Y(p)$

On a vu que le nombre de plis que l'on peut découper en une seule fois est limité par la capacité maximale de la machine de coupe. On peut donc être amené à découper un même placement dans plusieurs sections différentes. On définit :

$\forall p \in P, \quad Y(p) =$ le nombre de sections découpées avec le placement p .

Tous les coûts qui sont comptés section par section dépendent de ces variables dans la fonction économique.

2.2.1.3 Variables $Z(p)$

Lorsque l'on utilise de nouveaux placements, cela engendre des coûts spécifiques. Il faut donc savoir si on utilise tel ou tel placement. On définit :

$$\forall p \in P, \quad Z(p) = \begin{cases} 1 & \text{si } p \text{ est utilisé} \\ 0 & \text{sinon} \end{cases}$$

Les coûts de création des nouveaux placements dépendent des variables binaires Z dans la fonction économique.

2.2.1.4 Variables $E(j, p)$

Lors de la phase d'évacuation des pièces découpées, on doit connaître, le nombre de paquets à évacuer. On définit :

$$\forall p \in P, \quad \forall j \in \{1 \dots nTissus\}, \quad E(j, p) = \text{le nombre de paquets à évacuer sur le placement } p \text{ dans l'ensemble des tissus disponibles pour le } j^{\text{ième}} \text{ tissu de l'ordre.}$$

Les coûts d'évacuation des paquets dépendent de ces variables dans la fonction économique.

2.2.2 Définition des coûts

Ce paragraphe décrit tous les coûts à prendre en compte dans la fonction économique en fonction des variables auxquelles ils s'appliquent.

2.2.2.1 Coûts associés aux variables $X(j, s, p)$

On retrouve ici les coûts matières et les coûts de matelassage (hors temps fixe). On pose:

$$\forall p \in P, \forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\}, \\ Mat(j, s, p) = \text{le coût matière et de matelassage (hors temps fixe), d'un pli avec} \\ \text{le placement } p \text{ dans le tissu } (j, s).$$

Ce coût dépend principalement, de la longueur du placement p et du prix du tissu (j, s) , de la vitesse de matelassage et du prix horaire de matelassage. La totalité des coûts associés aux variables $X(j, s, p)$ s'écrit alors :

$$\left| \begin{array}{l} \text{CoûtsMatière} + \\ \text{CoûtMatelassageSsTpsFixe} \end{array} \right. = \sum_{j=1}^{n\text{Tissus}} \sum_{s=1}^{n\text{Laize}(j)} \sum_{p \in A(j,s)} \text{Mat}(j,s,p) \times X(j,s,p)$$

2.2.2.2 Coûts associés aux variables $Y(p)$

On retrouve ici les coûts de coupe, les coûts dus à tous les temps fixes (matelassage, coupe, évacuation) et les coûts de copie placement. On pose :

$$\forall p \in P,$$

$C_section(p)$ = le coût de coupe, les coûts dus aux temps fixes et les coûts de copie placement d'une section avec le placement p .

Ce coût dépend :

- Du périmètre total des pièces à découper sur le placement p .
- De la vitesse et du prix horaire de coupe.
- Des différents temps fixes et prix horaires.
- De la longueur et du coût de tracé (fixe + au mètre) du placement p .

La totalité des coûts associés aux variables $Y(p)$ s'écrit alors:

$$\left| \begin{array}{l} \text{CoûtsCoupe} + \\ \text{CoûtsTpsFixe} + \\ \text{CoûtTracéPlacement} \end{array} \right. = \sum_{p \in P} C_section(p) \times Y(p)$$

2.2.2.3 Coûts associés aux variables $Z(p)$

On retrouve ici les coûts de création des placements. On pose:

$$\forall p \in P,$$

$Creat(p)$ = le coût de création du placement p .

Ce coût dépend principalement du nombre de tailles contenues dans le placement p . (plus p contient de tailles plus la création sera chère). La totalité des coûts associés aux variables $Z(p)$ s'écrit alors:

$$| \text{CoûtsCréation} = \sum_{p \in P} \text{Creat}(p) \times Z(p)$$

2.2.2.4 Coûts associés aux variables $E(j, p)$

On retrouve ici les coûts d'évacuation des paquets. On pose:

$$\forall p \in P, \forall j \in \{1 \dots nTissus\},$$

$Evac(j, p)$ = le coût d'évacuation d'un paquet sur le placement p correspondant au $j^{\text{ième}}$ tissu du bordereau.

Ce coût dépend :

- Du nombre de pièces à évacuer sur le placement p .
- De la vitesse et du coût horaire d'évacuation des paquets.

La totalité des coûts associés aux variables $E(j, p)$ s'écrit alors :

$$| \text{CoûtsEvacuation} = \sum_{p \in P} \sum_{j=1}^{nTissus} Evac(j, p) \times E(j, p)$$

2.2.3 Résumé de la fonction économique

Avec les notations décrites ci-dessus, le coût d'une solution de sections est donné par la formule:

$$\begin{aligned}
\text{CoûtSolution} &= \begin{cases} \text{CoûtsMatière} + \text{CoûtMatelassageSsTpsFixe} + \\ \text{CoûtsCoupe} + \text{CoûtsTpsFixe} + \text{CoûtTracéPlacement} + \\ \text{CoûtsCréation} + \\ \text{CoûtsEvacuation} \end{cases} \\
&= \begin{cases} \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in A(j,s)} \text{Mat}(j,s,p) \times X(j,s,p) + \\ \sum_{p \in P} C_section(p) \times Y(p) + \\ \sum_{p \in P} \text{Creat}(p) \times Z(p) + \\ \sum_{p \in P} \sum_{j=1}^{nTissus} \text{Evac}(j,p) \times E(j,p) \end{cases}
\end{aligned}$$

2.3 Les contraintes

2.3.1 Découper les quantités de l'OF

C'est bien sûr la première contrainte à vérifier, elle a été explicitée dans *2.1 Notations, satisfaction de l'OF et variables X*. Elle s'écrit :

$$\forall i \in \{1 \dots nTailles\}, \quad \forall j \in \{1 \dots nTissus\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in A(j,s)} \{p_i \times X(j,s,p)\} \right\} = B_{ji}$$

2.3.2 Respecter les stocks disponibles

On doit veiller ici à ce que l'on ne découpe pas plus de tissu que l'on en a en stock. Il faut donc connaître la longueur totale (en mètres) dont on dispose pour chaque tissu. On pose :

$$\forall j \in \{1 \dots nTissus\}, \quad \forall s \in \{1 \dots nLaize(j)\},$$

$dispo(j,s)$ = longueur totale (en mètres) disponible pour le tissu (j,s)

$\forall p \in P$, $longueur(p)$ = la longueur (en mètres) de tissu nécessaire pour un pli du placement p

La contrainte de respect des stocks s'écrit alors :

$$\forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\}, \quad \sum_{p \in A(j,s)} longueur(p) \times X(j,s,p) \leq dispo(j,s)$$

2.3.3 Nombre maximum de plis dans une section

Cette contrainte est imposée par les machines de coupe qui ne peuvent pas découper plus d'un certain nombre de plis à la fois. Ce nombre noté $PliMax$ dépend de la machine de coupe, du tissu découpé, mais aussi de la qualité de découpe souhaitée. Respecter cette contrainte revient en fait à fixer le nombre de sections que l'on va découper avec un même placement en fonction du nombre total de plis que l'on découpe avec ce placement. Cela revient donc à déterminer les variables $Y(p)$ en fonction des variables $X(j,s,p)$.

Soit $nbPli(p) = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p)$ le nombre total de plis découpés avec le placement p .

On a alors: (si $\lfloor x \rfloor$ désigne la partie entière de x)

$$\text{Si } nbPli(p) \text{ n'est pas un multiple de } PliMax : Y(p) = \left\lfloor \frac{nbPli(p)}{PliMax} \right\rfloor + 1$$

$$\text{Si } nbPli(p) \text{ est un multiple de } PliMax : Y(p) = \frac{nbPli(p)}{PliMax}$$

Ceci peut se traduire linéairement par les deux inégalités suivantes:

$$Y(p) \geq \frac{nbPli(p)}{PliMax} \quad \text{et} \quad Y(p) < \frac{nbPli(p)}{PliMax} + 1$$

Finalement, (en transformant l'inégalité stricte en égalité large) la contrainte du nombre maximum de pli par section s'écrit :

$$\forall p \in P, \quad \begin{cases} \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) - PliMax \times Y(p) \leq 0 \\ - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) + PliMax \times Y(p) \leq PliMax - 1 \end{cases}$$

2.3.4 Contraintes sur la création des placements

Ces contraintes sont d'ordre industriel (on ne peut pas découper ce placement car il est trop long, il contient trop de tailles...). Pour que la modélisation soit correcte d'un point de vue mathématique et puisse fournir théoriquement la solution optimale, on considère que la bibliothèque de placements P contient tous les placements qu'il est possible d'utiliser pour résoudre l'ordre élémentaire (appelés par la suite placements réalisables).

On étudiera en détail dans le Chapitre 4 :s'il est toujours possible d'obtenir cette bibliothèque complète et comment procéder dans le cas contraire.

2.3.5 Nombre minimum de plis par placement

On doit ici prendre en compte la contrainte qui impose pour chaque placement un nombre minimum de plis à découper pour pouvoir l'utiliser. Ce nombre minimum dépend du nombre de tailles que le placement contient.

On pose :

$\forall p \in P, PliMin_p =$ Nombre minimum de plis à découper avec le placement p si on l'utilise.

Le nombre total de plis découpés avec le placement p est: $nbPli(p) = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p)$

On doit donc avoir:

$$\forall p \in P, \begin{cases} nbPli(p) \geq PliMin_p \\ \text{ou} \\ nbPli(p) = 0 \end{cases}$$

C'est une contrainte disjonctive, ce sont les variables $Z(p)$ qui vont permettre d'exprimer cette contrainte sous forme linéaire.

On rappelle la définition des variables $Z(p)$:

$$\forall p \in P, \quad Z(p) = \begin{cases} 1 & \text{si } p \text{ est utilisé} \\ 0 & \text{sinon} \end{cases}$$

En utilisant ces variables la contrainte s'écrit alors :

$$\forall p \in P, \quad \begin{cases} nbPli(p) \geq PliMin_p \text{ si } Z(p) = 1 \\ \text{et} \\ nbPli(p) = 0 \text{ si } Z(p) = 0 \end{cases}$$

Ce qui peut s'écrire de la façon suivante:

$$\forall p \in P, \quad \begin{cases} \text{C1: } Maj \times Z(p) - nbPli(p) \leq Maj - PliMin_p \\ \text{et} \\ \text{C2: } -Maj \times Z(p) + nbPli(p) \leq 0 \end{cases}$$

Où Maj est un majorant du nombre maximum de pli qu'il est possible de découper avec n'importe quel placement (par exemple: $Maj = \sum_{i,j} B_{ij}$).

En effet:

si $Z(p) = 1$ (on utilise le placement p)

C1: impose que $-nbPli(p) \leq -PliMin_p$, c'est à dire que $nbPli(p) \geq PliMin_p$ (c'est ce que l'on veut si p est utilisé).

C2: est toujours vérifiée.

Si $Z(p) = 0$ (p n'est pas utilisé)

C1: est toujours vérifiée

C2: impose que $nbPli(p) \leq 0$ et donc que $nbPli(p) = 0$ (le nombre de plis est nul).

La contrainte du nombre minimum de plis s'écrit donc :

$$\forall p \in P, \quad \begin{cases} Maj \times Z(p) - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) \leq Maj - PliMin_p \\ -Maj \times Z(p) + \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) \leq 0 \end{cases}$$

2.3.6 Calcul du nombre de paquets pour l'évacuation

Cette contrainte n'est pas une contrainte intrinsèque du problème, elle sert uniquement à calculer les valeurs des variables $E(j, p)$. Le nombre de paquets (également appelés bûches) à évacuer pour chaque placement et chaque tissu du bordereau dépend, du nombre de pièces à évacuer, des nombres de plis et de sections découpés avec le placement et de la taille des paquets souhaités.

On définit les quantités suivantes :

- $nbPli(j, p) = \sum_{s=1}^{nLaize(j)} X(j, s, p)$: le nombre total de plis découpés avec le placement p pour le $j^{\text{ème}}$ tissu de l'OF.
- $MultiPack$ = Le nombre de plis souhaités par paquets.

On a alors : (si $\lfloor x \rfloor$ désigne la partie entière de x)

$$\text{Si } nbPli(j, p) \text{ n'est pas un multiple de } MultiPack : E(j, p) = \left\lfloor \frac{nbPli(j, p)}{MultiPack} \right\rfloor + 1$$

$$\text{Si } nbPli(j, p) \text{ est un multiple de } MultiPack : E(j, p) = \frac{nbPli(j, p)}{MultiPack}$$

Ceci peut se traduire linéairement par les deux inégalités suivantes:

$$E(j, p) \geq \frac{nbPli(j, p)}{MultiPack} \quad \text{et} \quad E(j, p) < \frac{nbPli(j, p)}{MultiPack} + 1$$

Finalement, la contrainte permettant de calculer les variables $E(j, p)$ s'écrit :

$$\forall p \in P, \forall j \in \{1 \dots nTissus\} \quad \begin{cases} \sum_{s=1}^{nLaize(j)} X(j, s, p) - MultPack \times E(j, p) \leq 0 \\ - \sum_{s=1}^{nLaize(j)} X(j, s, p) + MultPack \times E(j, p) \leq MultPack - 1 \end{cases}$$

2.4 Résumé de la mise en équations

Ainsi posé, le problème de la construction des sections revient à un problème d'optimisation (ici une minimisation) linéaire en nombres entiers.

Les variables à déterminer sont:

- $X(j, s, p) \quad \forall p \in P, \forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\}$
soit : $|P| \times \sum_{j=1}^{nTissus} nLaize(j)$ variables.
- $Y(p) \quad \forall p \in P$ soit : $|P|$ variables.
- $Z(p) \quad \forall p \in P$ soit : $|P|$ variables.
- $E(j, p) \quad \forall p \in P, \forall j \in \{1 \dots nTissus\}$ soit : $|P| \times nTissus$ variables.

Toutes les autres quantités sont des données du problème ou sont calculables une fois la bibliothèque P connue. Le nombre total de variables est donc égal à :

$$|P| \times \left(2 + nTissus + \sum_{j=1}^{nTissus} nLaize(j) \right)$$

Le programme linéaire est le suivant:

$$\begin{aligned}
Z \min = & \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in A(j,s)} Mat(j,s,p) \times X(j,s,p) + \sum_{p \in P} C_section(p) \times Y(p) \\
& + \sum_{p \in P} Creat(p) \times Z(p) + \sum_{p \in P} \sum_{j=1}^{nTissus} Evac(j,p) \times E(j,p)
\end{aligned}$$

avec :

Satisfaction du bordereau :

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in A(j,s)} \{p_i \times X(j,s,p)\} \right\} = B_{ji}$$

Respect des stocks:

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \sum_{p \in A(j,s)} longueur(p) \times X(j,s,p) \leq dispo(j,s)$$

Nombre maximum de plis:

$$\forall p \in P, \quad \left\{ \begin{array}{l} \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) - PliMax \times Y(p) \leq 0 \\ - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) + PliMax \times Y(p) \leq PliMax - 1 \end{array} \right.$$

Nombre minimum de plis:

$$\forall p \in P, \quad \left\{ \begin{array}{l} Maj \times Z(p) - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) \leq Maj - PliMin_p \\ -Maj \times Z(p) + \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j,s,p) \leq 0 \end{array} \right.$$

Calcul du nombre de paquets:

$$\forall p \in P, \forall j \in \{1 \dots nTissus\} \quad \left\{ \begin{array}{l} \sum_{s=1}^{nLaize(j)} X(j,s,p) - MultPack \times E(j,p) \leq 0 \\ - \sum_{s=1}^{nLaize(j)} X(j,s,p) + MultPack \times E(j,p) \leq MultPack - 1 \end{array} \right.$$

Intégrité des variables:

$$\left. \begin{array}{l} \forall p \in P \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \begin{cases} Z(p) \in \{0,1\} \\ Y(p) \in \mathbb{N} \\ E(j,p) \in \mathbb{N} \\ X(j,s,p) \in \mathbb{N} \end{cases}$$

Le nombre total de contraintes est donc réparti de la façon suivante:

- Satisfaction des quantités de l'OF : $nTailles \times nTissus$
- Respect des stocks : $\sum_{j=1}^{nTissus} nLaize(j)$
- Nombre maximum de plis : $2 \times |P|$
- Nombre minimum de plis : $2 \times |P|$
- Calcul du nombre de paquets : $2 \times nTissus \times |P|$

La taille du problème dépend donc des nombres de tailles et de tissus disponibles de l'ordre de fabrication (c'est à dire de la taille de la matrice des quantités) et du nombre de placements utilisables pour la résolution. Si les dimensions de la matrice des quantités dépassent rarement quelques dizaines de colonnes ou de lignes, on verra dans le Chapitre 4 : que le nombre de placements utilisables $|P|$ augmente de façon exponentielle en fonction de $nTailles$.

Pour un exemple de taille modeste on a les données suivantes :

- $nTailles = 7$
- $nTissus = \sum_{j=1}^{nTissus} nLaize(j) = 5$
- $|P| = 881$

Les nombres de variables et de contraintes sont alors respectivement égaux à 10572 et 12374. En revanche, le problème est relativement creux puisque le nombre de coefficients non nuls est 54659. Cela représente moins d'un coefficient sur 2000.

2.5 Résolution directe

Les deux paragraphes suivants donnent un aperçu très succinct des méthodes d'optimisation en programmation linéaire. Pour plus de précision quant à ces techniques, le lecteur pourra se référer par exemple à [Chvatal1983].

2.5.1 Algorithme du Simplexe

La méthode du simplexe est un algorithme permettant de résoudre de manière exacte des problèmes d'optimisation pouvant se mettre sous la forme :

$$\begin{array}{l} \min \mathbf{c}^t \mathbf{x} = z \\ \text{LP : } \left\{ \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+^n \end{array} \right. \end{array}$$

Dans cette formulation, \mathbf{x} représente le vecteur des variables qui sont réelles, positives ou nulles. Le vecteur \mathbf{c} représente le coût associé à chaque variable, la matrice A et le vecteur \mathbf{b} expriment les contraintes que doivent respecter les variables. Si l'on relaxe les contraintes d'intégrité des variables, la modélisation du problème proposée dans le paragraphe précédent peut se mettre sous la forme LP. La résolution exacte de la construction des sections est donc envisageable si l'on peut adapter l'algorithme du simplexe pour résoudre des problèmes en nombres entiers.

2.5.2 Méthode de Branch & Bound

La méthode de Branch & Bound (ou séparation et évaluation) est une technique de résolution de problèmes d'optimisation linéaire en nombres entiers :

$$\begin{array}{l} \min \mathbf{c}^t \mathbf{x} = z \\ \text{MIP : } \left\{ \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \\ \mathbf{x} \in \mathbb{N}^n \end{array} \right. \end{array}$$

L'algorithme consiste à construire un arbre d'énumération de sous-problèmes d'optimisation linéaire en nombres réels. Au nœud racine de l'arbre, le problème résolu est le problème de départ dans lequel les contraintes d'intégrités des variables sont relaxées :

$$\begin{array}{l} \min \mathbf{c}'\mathbf{x} = z \\ \text{LP}_0 : \quad \left| \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+^n \end{array} \right. \end{array}$$

Si la solution de ce problème relaxé n'a que des variables entières, le problème de départ est résolu. En général ce n'est pas le cas, on choisit alors une variable x_j telle que $x_j = a \notin \mathbb{N}$, et on scinde le problème de départ en 2 en posant :

$$\begin{array}{l} \min \mathbf{c}'\mathbf{x} = z \\ \text{LP}_1^1 : \quad \left| \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+^n \\ x_j \leq \lfloor a \rfloor \end{array} \right. \end{array} \qquad \begin{array}{l} \min \mathbf{c}'\mathbf{x} = z \\ \text{LP}_1^2 : \quad \left| \begin{array}{l} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+^n \\ x_j \geq \lfloor a \rfloor + 1 \end{array} \right. \end{array}$$

On découple ainsi le problème jusqu'à obtenir des solutions entières. Cette méthode permet d'explorer implicitement toutes les solutions réalisables du problème. Elle est toutefois plus avantageuse que l'énumération totale des solutions car on peut éliminer des branches de l'arbre qui ne contiennent pas la solution optimale. En effet, si on obtient une solution entière dont le coût est v . Si à un endroit de l'arbre, la solution d'un sous-problème à un coût supérieur à v alors toutes les solutions réalisables issues de ce sous-problème ont un coût supérieur à v , on peut donc se passer d'explorer cette branche de l'arbre. L'algorithme permet aussi de calculer des bornes sur le coût de la solution optimale. En effet, la solution optimale est issue d'un des sous-problèmes précédemment traité. Son coût est donc minoré par le minimum des coûts des solutions des sous-problèmes. Ainsi, après la résolution d'un sous-problème on dispose d'un encadrement sur le coût de la solution optimale :

- Une borne supérieure fournie par la meilleure solution entière trouvée jusqu'à maintenant

- Une borne inférieure égale au minimum des coûts des solutions des sous-problèmes actifs (ceux qui n'ont pas encore de fils)

Chaque sous-problème est du type LP et est résolu avec la méthode du simplexe. On dispose donc désormais d'une méthode de résolution exacte pour le problème de la construction des sections.

2.5.3 Choix d'un logiciel de résolution

Il existe plusieurs logiciels mettant en œuvre les algorithmes du Simplexe et de Branch & Bound. Certains d'entre eux sont gratuits comme [Lp_solve](#) ou [Glpk](#), d'autres sont des logiciels commerciaux : [Cplex](#), [Lindo](#) ou [Xpress-MP](#). Même si ces algorithmes sont en théorie relativement peu compliqués à implémenter, ces logiciels sont souvent le fruit de plusieurs dizaines d'années de recherches sur l'amélioration du temps de résolution et de la stabilité numériques par exemple. Pour cette raison, il n'a pas été envisagé de développer un Simplexe ou un Branch & Bound propre à la construction des sections.

[Lp_solve](#) a été éliminé d'entrée à cause d'une trop grande instabilité numérique dès que la taille des problèmes augmente. On réalise des tests de performance entre [Glpk](#), [Lindo](#), [Xpress-MP](#) et [Cplex](#) sur deux types de problèmes spécifiques à la construction des sections. Sur la **Figure 20**, le graphique de gauche donne les temps moyens de résolution pour les 4 solveurs sur un ensemble de simplexes de grandes tailles qui correspondent à des problèmes de construction de sections dont on a relaxé la contrainte d'intégrité des variables. Le graphique de droite donne les temps moyens de résolution pour des problèmes exacts mais de taille réduite.

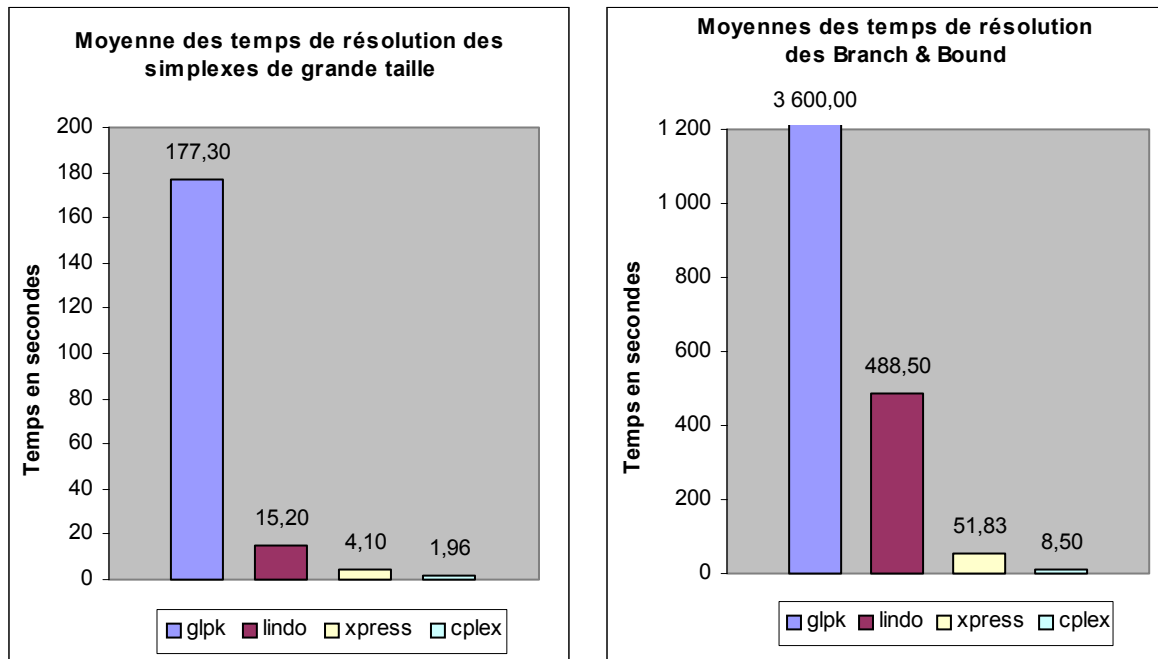


Figure 20 : Comparaisons de différents solveurs sur des problèmes spécifiques

Les graphiques montrent que pour ces deux types de problèmes, [Cplex](#) est de loin le plus rapide surtout en ce qui concerne les problèmes en nombres entiers.

2.5.4 Cas où la résolution directe est possible

L'inconvénient essentiel de la méthode de Branch & Bound est que son temps de résolution peut dans le pire des cas augmenter exponentiellement en fonction de la taille du problème à traiter. Dans certains cas, la solution optimale est trouvée relativement rapidement mais la preuve de l'optimalité demande encore un très grand nombre d'itérations. Mais il arrive aussi que le nombre de variables entières soit si important que l'algorithme cherche en vain une solution réalisable pendant plusieurs heures.

Dans le cas du problème de la construction de sections, la donnée critique est le nombre de placements dans la bibliothèque de placements réalisables P . On peut résoudre le problème de manière exacte lorsque $|P|$ est faible. Cela se produit principalement lorsque la création de nouveaux placements est interdite ; on ne doit alors utiliser dans les sections que des placements préexistants. Ceux-ci sont généralement en nombre limité, quelques dizaines la

plupart du temps. En pratique, ce cas de figure se présente lorsque les coûts de création de placements sont très élevés et que les articles à produire varient peu au cours du temps. La résolution directe est alors très efficace et fournit rapidement la solution optimale du problème.

2.5.5 Cas où la résolution directe est impossible

Lorsque le nombre de placements $|P|$ est trop important pour envisager une résolution directe, deux types d'approches de résolution sont possibles :

- Des méthodes de résolution exactes de types génération de colonnes
- Des méthodes heuristiques de type recuit simulé, recherche tabou, algorithmes génétiques...

D'un point de vue mathématique, les méthodes exactes sont bien sûr les plus attrayantes. Cependant, dans le cas de la construction des sections elles ne semblent pas être les plus adaptées. En effet, la nature du problème fait que le nombre de contraintes est proportionnel au nombre de variables et cela ne facilite pas la procédure de génération de colonnes. De plus, pour la plupart des utilisateurs, obtenir la solution optimale est moins important que de parvenir rapidement à une bonne solution.

Si l'on abandonne l'idée de prouver l'optimalité de la solution fournie, on peut choisir une méthode de résolution heuristique. La méthode du recuit simulé présente l'avantage d'être relativement simple à mettre en place pour des problèmes très divers. De plus, on manipule à chaque itération des solutions réalisables du problème entier cela peut permettre l'interruption de la résolution à n'importe quel moment du traitement. Le chapitre suivant, est une présentation générale de l'algorithme du recuit simulé. Le Chapitre 5 : *Nouvelle méthode de résolution* présente quant à lui l'utilisation de cette méthode pour la résolution du problème de construction des sections.

Chapitre 3 : La méthode du recuit simulé

3.1 Introduction

Le recuit simulé est une méthode heuristique de résolution de problèmes d'optimisation combinatoire. Elle appartient à la famille des algorithmes de recherche locale dits à seuil. La particularité du recuit simulé est d'être une méthode d'optimisation stochastique (basée sur la théorie des probabilités) ce qui a permis de démontrer des résultats théoriques de convergence vers les solutions optimales. En pratique, le recuit simulé fournit des solutions intéressantes à beaucoup de problèmes classiques d'optimisation combinatoire (voyageur de commerce, coloration de graphe...).

Les paragraphes suivants ont pour but de présenter succinctement l'origine, les fondements théoriques et la mise en œuvre pratique du recuit simulé. Ils sont essentiellement tirés de [Aarts 1997], le lecteur pourra s'y référer pour plus de précision.

3.2 De la métallurgie aux mathématiques

Dans le domaine de la métallurgie, le recuit est un procédé thermique permettant d'obtenir des états de basses énergies pour un métal. Ce procédé se décompose en deux étapes :

- Le métal est d'abord porté à température de fusion
- La température est ensuite lentement diminuée jusqu'à ce que les particules du liquide s'ordonnent dans l'état fondamental du métal.

Dans la phase liquide, les particules du métal s'agencent aléatoirement. Au contraire, dans l'état fondamental les particules forment un réseau hautement structuré ce qui confère au métal une très grande solidité. L'énergie du système physique constitué par les particules du métal est alors minimale.

Une modélisation de ce processus a été proposée par Métropolis en 1953. Il a notamment fourni un algorithme permettant de simuler l'évolution du métal jusqu'à l'équilibre thermique :

Etant donné un état i du métal avec l'énergie E_i , l'état suivant j avec l'énergie E_j est obtenu en modifiant légèrement l'état i (déplacement d'une particule de métal par exemple). Si la différence $E_j - E_i$ est négative, l'état j est accepté comme nouvel état. Si la différence d'énergie est strictement positive, l'état j est accepté avec la probabilité : $\exp\left(\frac{E_i - E_j}{k_B T}\right)$ où T est la température du métal et k_B la constante de Boltzmann.

Cette règle d'acceptation des nouveaux états est appelée le critère de Métropolis. Si la température est diminuée suffisamment lentement, le métal peut atteindre l'équilibre thermique à chaque température :

Pour une température T donnée, à l'équilibre thermique, la probabilité pour que le métal soit dans un état i est donnée par : $P_T\{X = i\} = \frac{\exp\left(\frac{-E_i}{k_B T}\right)}{\sum_{\{\text{états } j\}} \exp\left(\frac{-E_j}{k_B T}\right)}$ où X est une variable aléatoire représentant l'état actuel du métal. Cette loi de probabilité est appelée distribution de Boltzmann.

Par analogie avec ce processus physique de recuit, on peut appliquer le critère de Métropolis pour générer une suite de solutions d'un problème d'optimisation combinatoire. L'analogie est basée sur les équivalences suivantes :

- Une solution d'un problème d'optimisation combinatoire est équivalente à un état du métal
- Le prix d'une solution du problème est équivalent à l'énergie d'un état du métal

Dans cette procédure de recuit simulé, la quantité $k_b T$ est remplacée par un paramètre (également appelé température) contrôlant la règle d'acceptation des solutions.

3.3 Fondements théoriques

L'algorithme du recuit simulé peut être modélisé mathématiquement en utilisant la théorie des chaînes de Markov.

3.3.1 Quelques rappels sur les chaînes de Markov

- Une suite de variables aléatoires $(X_n)_{n \in \mathbb{N}}$ définies sur un ensemble fini d'événements Ω est une chaîne de Markov si la réalisation de X_n ne dépend du passé que par la réalisation de X_{n-1} autrement dit si :

$$P(X_n = i_n / X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) = P(X_n = i_n / X_{n-1} = i_{n-1})$$

- On appelle alors matrice de transition la matrice $P(n)$ définie par la relation :

$$P_{ij}(n) = P(X_n = j / X_{n-1} = i)$$

- On dit qu'une chaîne de Markov est homogène si la matrice de transition est la même quel que soit $n : P(X_n = j / X_{n-1} = i) = P_{ij} \quad \forall n \in \mathbb{N}$. Dans ce cas on a :

$$P(X_{n+m} = j / X_n = i) = (P^m)_{ij} \quad \left(\text{où } P^m = \underbrace{P \times \dots \times P}_{m \text{ fois}} \right)$$

- Une chaîne de Markov homogène de matrice de transition P admet une distribution stationnaire si : $\forall i \in \Omega, \exists q_i \geq 0 / q_i = \left(\lim_{n \rightarrow +\infty} P\{X_n = i / X_0 = j\} \quad \forall j \in \Omega \right) = \lim_{n \rightarrow +\infty} P\{X_n = i\}$

de plus, si une telle distribution existe, on a : ${}^t P q = q$

- Une chaîne de Markov homogène est irréductible si : $\forall i, j \in \Omega \quad \exists n > 0 / (P^n)_{ij} > 0$

(on peut passer d'un état i à un état j avec une probabilité strictement positive en un nombre fini d'étapes.)

- Une chaîne de Markov homogène est apériodique si : $\forall i \in \Omega \text{ pgcd}(D_i) = 1$
où $D_i = \{n \in \mathbb{N} / (P^n)_{ii} > 0\}$.
- Si $(X_n)_{n \in \mathbb{N}}$ est une chaîne de Markov homogène irréductible et apériodique, il existe une unique distribution stationnaire q .

3.3.2 Application au recuit simulé

On représente un problème d'optimisation combinatoire par un ensemble fini S de solutions réalisables et une fonction de coût $f : S \rightarrow \mathbb{R}^+$. On pose $S^* = \{i^* \in S / f(i^*) \leq f(i) \quad \forall i \in S\}$ l'ensemble des solutions optimales du problème. On dispose également d'une fonction de voisinage $V : S \rightarrow 2^S$ associant à chaque élément s de S un sous-ensemble $V(s)$ de S de solutions « proches » de s . On génère une suite $(X_n)_{n \in \mathbb{N}}$ de solutions du problème avec la loi de probabilité suivante :

$$\forall i, j \in S \quad \mathbf{P}_c \{X_n = j / X_{n-1} = i\} = \begin{cases} \frac{1}{|V(i)|} \chi_{(V(i))}(j) \times \exp\left(\frac{-(f(j) - f(i))^+}{c}\right) & \text{si } i \neq j \\ 1 - \sum_{l \in S - \{i\}} \frac{1}{|V(i)|} \chi_{(V(i))}(l) \times \exp\left(\frac{-(f(l) - f(i))^+}{c}\right) & \text{si } i = j \end{cases}$$

où: $\chi_{(V(i))}(j) = \begin{cases} 1 & \text{si } j \in V(i) \\ 0 & \text{sinon} \end{cases}$ et $a^+ = \begin{cases} a & \text{si } a > 0 \\ 0 & \text{sinon} \end{cases}$

Autrement dit, on accepte de passer d'une solution i à une solution j si $j \in V(i)$. De plus, la probabilité de passer à la solution j est plus faible si son coût est plus élevé que celui de la solution i . Ainsi définie, la suite $(X_n)_{n \in \mathbb{N}}$ est une chaîne de Markov homogène. De plus si l'on suppose que la fonction de voisinage est telle que l'on peut « passer » de n'importe quelle solution à n'importe quelle autre en un nombre fini d'étapes et que $\forall i \in S, |V(i)| = \Theta$, alors $(X_n)_{n \in \mathbb{N}}$ est également irréductible et apériodique.

Il existe alors une distribution stationnaire unique (qui dépend de c) dont les composants sont donnés par :

$$\forall i \in S, q_i(c) = \frac{\exp\left(-\frac{f(i)}{c}\right)}{\sum_{k \in S} \exp\left(-\frac{f(k)}{c}\right)}$$

On retrouve ici la distribution de Boltzmann décrite dans 3.2. Pour toute solution i , on a

$$\text{donc, } \lim_{n \rightarrow +\infty} P_c \{X_n = i\} = \frac{\exp\left(-\frac{f(i)}{c}\right)}{\sum_{k \in S} \exp\left(-\frac{f(k)}{c}\right)}$$

$$\text{Or, } \lim_{c \rightarrow 0} \frac{\exp\left(-\frac{f(i)}{c}\right)}{\sum_{k \in S} \exp\left(-\frac{f(k)}{c}\right)} = \frac{1}{|S^*|} \chi_{(S^*)}(i) \text{ et donc, } \lim_{c \rightarrow 0} \left(\lim_{n \rightarrow +\infty} P_c \{X_n = i\} \right) = \begin{cases} 0 & \text{si } i \notin S^* \\ \frac{1}{|S^*|} & \text{sinon} \end{cases}$$

$$\text{Finalement, } \lim_{c \rightarrow 0} \left(\lim_{n \rightarrow +\infty} P_c \{X_n \in S^*\} \right) = 1$$

Ce premier résultat sur les chaînes de Markov homogènes permet d'entrevoir, sous certaines conditions d'acceptation des solutions, la notion de convergence du recuit simulé vers une solution optimale du problème. Cependant, lorsque l'on fait tendre c vers 0, à chaque valeur du paramètre, la chaîne de Markov correspondante est de longueur infinie avant d'atteindre la distribution stationnaire.

Afin de n'avoir qu'un nombre fini de résultats pour chaque valeur de c , le recuit simulé peut également être modélisé par une chaîne de Markov non-homogène ayant les mêmes probabilités de transition que précédemment, le paramètre c est simplement remplacé par $(c_n)_{n \in \mathbb{N}}$. Sous certaines conditions sur la décroissance vers 0 de la suite $(c_n)_{n \in \mathbb{N}}$, on peut montrer que l'on obtient le même résultat de convergence que précédemment :

$$\lim_{n \rightarrow +\infty} P \{X_n \in S^*\} = 1.$$

Il est donc possible de prouver la convergence du recuit simulé en utilisant les chaînes de Markov. Cependant, il a été démontré que cette convergence est très lente et que le nombre d'itérations nécessaires pour être arbitrairement proche de la solution optimale est la plupart du temps supérieure à la taille de l'espace des solutions. Dans un tel cas il serait alors plus avantageux d'énumérer toutes les solutions et de retenir la meilleure. En pratique, on devra donc abandonner l'idée de garantir l'optimalité de la solution pour que l'algorithme se termine en un temps raisonnable.

3.4 Algorithme

On génère une suite de chaînes de Markov homogènes pendant lesquelles la température est constante de longueur finie (on les appelle paliers de température). On passe d'une chaîne à la suivante en diminuant la température. Le schéma de la **Figure 21** représente ce processus. On distingue 3 notions de solution dans l'algorithme :

- La solution courante : c'est la dernière solution testée par l'algorithme, elle est différente à chaque itération, on la choisit aléatoirement dans le voisinage de la solution optimale courante
- La solution optimale courante : c'est la dernière solution acceptée par la règle de Métropolis. Une solution peut donc être acceptée comme courante optimale bien qu'elle dégrade le coût de l'actuelle solution optimale courante.
- La solution optimale : c'est la meilleure solution trouvée jusqu'à présent. A chaque nouveau palier de température on repart de cette solution. Elle converge théoriquement vers la solution optimale du problème.

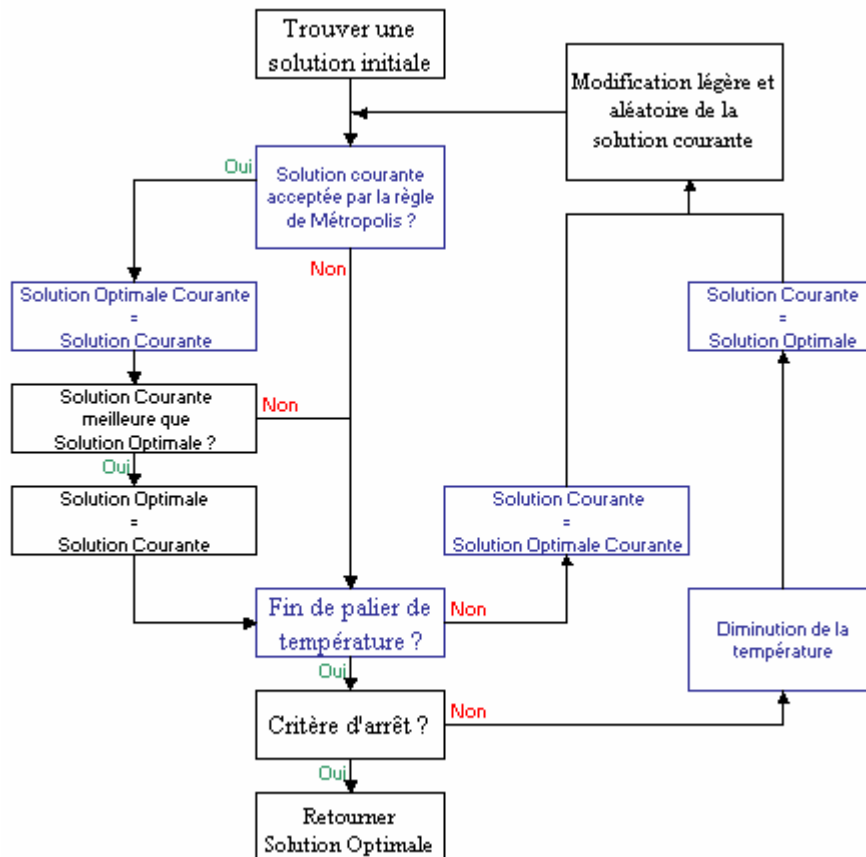


Figure 21 : Schéma général de l'algorithme du recuit simulé

A chaque itération, la solution courante est testée, si elle est acceptée par la règle de Métropolis, elle devient solution courante optimale, et bien sur, si elle améliore la solution optimale elle devient elle-même solution optimale.

Pour passer à l'itération suivante on doit savoir si l'on reste dans le même palier de température (c'est à dire dans la même chaîne de Markov homogène).

- Si c'est le cas, on cherche une nouvelle solution courante aléatoirement (avec équiprobabilité) parmi les voisins de la solution optimale courante.
- Sinon, on diminue la température de manière à rendre moins probable une dégradation de la solution et on cherche une nouvelle solution courante aléatoirement parmi les voisins de la solution optimale.

3.5 Remarques importantes

3.5.1 Les paramètres

La qualité de la convergence vers la solution optimale (ou au moins vers une bonne solution approchée) dépend beaucoup de la valeur des différents paramètres du recuit simulé qui sont :

- La température initiale.
- La fonction de décroissance de la température (comment fait-on décroître la température ?)
- Le nombre de paliers de température que l'on souhaite effectuer
- La longueur de chaque palier (combien fait-on d'itérations pour une température donnée ?)

Il est sûr que si l'on veut avoir la meilleure solution possible, il faut effectuer le maximum de paliers possible et les faire le plus long possible. Cependant, pour satisfaire aux impératifs de temps, il conviendra de régler les paramètres ci-dessus de façon à trouver le meilleur compromis possible entre vitesse d'exécution de l'algorithme et qualité de la solution renvoyée.

3.5.2 La notion de voisinage

Une autre notion primordiale pour l'algorithme du recuit simulé est celle de voisinage. En effet, la fonction de voisinage doit permettre d'assurer l'irréductibilité de la chaîne de Markov. Pour cela on doit pouvoir « passer » de n'importe quelle solution à n'importe quelle autre en un nombre fini d'itérations.

Chapitre 4 : Méthodes de recherche des placements

4.1 Introduction

Dans le Chapitre 2 : Modélisation mathématique on a noté P l'ensemble de tous les placements, existants ou satisfaisant les contraintes de création, éventuellement utilisables pour résoudre l'ordre de fabrication. Ce chapitre, permet de décrire différentes méthodes pour déterminer :

- La bibliothèque complète lorsque c'est possible.
- Une partie seulement de cette bibliothèque lorsqu'une description exhaustive n'est pas envisageable.

Le résultat de chacune de ces méthodes est un ensemble de placements (pouvant contenir plusieurs milliers de placements) que l'on pourra éventuellement utiliser pour résoudre l'ordre de fabrication. C'est cet ensemble qui sera alors noté P . Dans le cas où une description exhaustive de tous les placements réalisables n'est pas possible, on s'interdit d'utiliser certains d'entre eux. La perte de généralité que cela entraîne ne permet alors pas de garantir que la solution fournie est optimale. Cependant, cette concession a déjà été faite par le choix de l'utilisation d'une méthode heuristique telle que le recuit simulé. De plus, comme le temps de résolution est limité, on verra que l'on obtient souvent de meilleurs résultats par une exploration efficace d'un domaine restreint que par une exploration trop partielle d'un domaine plus important.

Lorsque l'on choisit de ne retenir qu'une partie des placements éventuellement utilisables, on essaie de sélectionner ceux qui seront a priori les plus utiles pour la construction de sections. Dans cette optique, on définit dans le paragraphe suivant la notion de configuration de placements.

4.1.1 Notion de « configuration de placement »

$\forall 1 \leq i \leq nTailles$, p_i est le nombre de répétitions de la taille i sur le placement p .

On peut par exemple en déduire le nombre de tailles dans le placement p par :

$nbTailles(p) = \sum_{i=1}^{nTailles} p_i$. Cela fournit un premier moyen de classification des placements entre

eux :

- mono-taille
- 2-tailles
- etc....

Pour décrire explicitement l'ensemble des placements réalisables, cette première classification n'est pas suffisante. En effet, parmi les placements 4-tailles par exemple, on peut trouver des configurations très différentes (une même taille répétée 4 fois, 4 tailles répétées 1 fois, etc....).

Pour plus de précision, on définit la notion de « configuration de placement » :

On dit que 2 placements p^1 et p^2 ont la même configuration s'il existe une permutation σ des entiers $\{1...nTailles\}$ telle que $p_i^1 = p_{\sigma(i)}^2 \quad \forall i \in \{1...nTailles\}$ c'est à dire s'ils ont le même ensemble de répétitions de tailles mais pas nécessairement dans le même ordre. Bien sûr, deux placements qui ont la même configuration contiennent le même nombre de tailles. De plus, le nombre de placements différents ayant la même configuration est égal au nombre de permutations différentes de l'ensemble $\{p_i / 1 \leq i \leq nTailles\}$.

Les 3 placements du tableau de la **Figure 22** sont tous des placements 4-tailles. P1 et P2 ont la même configuration, elle est différente de celle de P3.

	S	M	L	XL
P1	1	2	0	1
P2	1	1	0	2
P3	1	1	1	1

Figure 22 : Exemples de configurations de placements

4.2 Recherche exhaustive

On cherche à décrire dans ce cas de figure tous les placements qu'il est possible d'utiliser pour résoudre l'ordre de fabrication (appelés par la suite placements réalisables). Cela peut être envisagé soit pour des ordres de fabrication de taille modérée (nombre de colonnes faible) soit lorsque les contraintes sur la création de nouveaux placements sont très restrictives. Si aucune de ces conditions n'est satisfaite, le nombre de placements réalisables est énorme. Pour s'en rendre compte, on calcule le nombre de configurations de placement possibles puis, pour chacune d'elles, le nombre de placements qu'elles contiennent.

4.2.1 Calcul du nombre de configurations possibles

On considère le problème suivant :

$$\left| \begin{array}{l} \text{Soient } n \in \mathbb{N} \text{ et } k \in \mathbb{N} / 1 \leq k \leq n \\ \text{Config}_n^k = \left\{ \left\{ p_j / 1 \leq j \leq k \right\} / \left(p_j > 0 \forall j \sum_{j=1}^k p_j = n \right) \right\} \\ \text{L'ensemble des } k\text{-uplet d'entiers strictement positifs dont la somme est } n \\ \forall n \in \mathbb{N} \text{ et } k \in \mathbb{N} / 1 \leq k \leq n \text{ on veut calculer: } f(n, k) = \#(\text{Config}_n^k) \text{ où } \#(A) \text{ est le cardinal de l'ensemble } A \end{array} \right.$$

L'ordre des p_j étant indifférent, on impose sans perte de généralité que : $p_1 \leq p_2 \leq \dots \leq p_k$

Si $\forall n \in \mathbb{N}$, on a $f(n, 1) = 1$ avec $p_1 = n$

Montrons que : $\forall n \in \mathbb{N}$ fixé, $f(n, k) = \sum_{j=1}^{\lfloor n/k \rfloor} f(n - k \times (j-1), k-1) \quad \forall 2 \leq k \leq n$

$$\text{Cas 1: } p_1 = 1 \text{ et } \sum_{j=2}^k p_j = n-1 \text{ avec } p_j \geq 1 \quad \forall j \geq 2$$

$$\text{Cas 2 : } p_1 = 2 \text{ et } \sum_{j=2}^k p_j = n-2 \quad \text{avec } p_j \geq 2 \quad \forall j \geq 2$$

$$\text{Cas } \lfloor \frac{n}{k} \rfloor: \quad p_1 = \lfloor \frac{n}{k} \rfloor \text{ et } \sum_{j=2}^k p_j = n - \lfloor \frac{n}{k} \rfloor \text{ avec } p_j \geq \lfloor \frac{n}{k} \rfloor \quad \forall j \geq 2$$

$$\text{Il n'y a pas plus de cas car: } p_1 \geq \lfloor \frac{n}{k} \rfloor + 1 \Rightarrow \begin{cases} \sum_{j=1}^k p_j \geq k \times (\lfloor \frac{n}{k} \rfloor + 1) > n \\ \text{car } n = k \times \lfloor \frac{n}{k} \rfloor + r \text{ avec } 0 \leq r \leq k-1 \end{cases}$$

$\forall 1 \leq l \leq \lfloor \frac{n}{k} \rfloor$, le nombre de possibilités pour le cas l est égal au nombre de $(k-1)$ -uplets d'entiers supérieurs ou égaux à l dont la somme est $n-l$.

$$\text{Or, } \begin{cases} \sum_{j=2}^k p_j = n-l \\ p_j \geq l \quad \forall j \geq 2 \end{cases} \Leftrightarrow \begin{cases} \sum_{j=2}^k (p_j - (l-1)) = n-l - (k-1) \times (l-1) \\ p_j \geq l \quad \forall j \geq 2 \end{cases} \Leftrightarrow \begin{cases} \sum_{j=2}^k \tilde{p}_j = n - k(l-1) - 1 \\ \tilde{p}_j > 0 \quad \forall j \geq 2 \end{cases}$$

Le nombre de possibilités pour le cas l est donc égal au nombre de $(k-1)$ -uplets d'entiers strictement positifs dont la somme est $n - k(l-1) - 1$ c'est à dire $f(n - k(l-1) - 1, k-1)$

$$\text{Et donc, } f(n, k) = \sum_{j=1}^{\lfloor \frac{n}{k} \rfloor} f(n - k(j-1) - 1, k-1)$$

Finalement, de proche en proche, on peut calculer $f(n, k) \quad \forall n \in \mathbb{N}$ et $k \in \mathbb{N} / 1 \leq k \leq n$.

Revenons au problème des configurations de placements. Trois paramètres influent sur le nombre de configurations :

- *nTailles* le nombre de tailles de l'OF (le nombre de colonnes de la matrice des quantités)
- *nbMaxTailles* le nombre maximum de tailles acceptées dans un placement
- *nbMaxRep* le nombre maximum de répétitions d'une même taille

Pour $1 \leq n \leq \text{nbMaxTailles}$ et $1 \leq k \leq n\text{Tailles}$ on veut calculer le nombre de configurations de placements contenant n tailles pour lesquelles k tailles différentes sont répétées (mais pas plus de *nbMaxRep* fois chacune). Cela revient à calculer le nombre de k -uplets d'entiers compris entre 1 et *nbMaxRep* dont la somme est n .

On définit par $nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n, k)$ ce nombre. On déduit du résultat précédent que :

$$\left(\begin{array}{l} \forall 1 \leq n \leq nbMaxTailles \\ \forall 1 \leq k \leq \min(n, nTailles) \end{array} \right) \left\{ \begin{array}{l} nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n, 1) = 1 \\ nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n, k) = \sum_{j=1}^r nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n - k(j-1) - 1, k - 1) \\ \text{avec } r = \min\left(nbMaxRep, \left\lfloor \frac{n}{k} \right\rfloor\right) \end{array} \right.$$

On peut donc calculer de manière récursive le nombre de configurations pour chaque n et chaque k et donc le nombre total de configurations possibles en calculant :

$$\sum_{n=1}^{nbMaxTailles} \sum_{k=1}^{K \max_n} nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n, k) \text{ avec } K \max_n = \min(n, nTailles)$$

$nbConf(n,k)$		k										Total	
		1	2	3	4	5	6	7	8	9	10		
n	1	1											1
	2	1	1										2
	3	1	1	1									3
	4	1	2	1	1								5
	5	1	2	2	1	1							7
	6	1	3	3	2	1	1						11
	7	1	3	4	3	2	1	1					15
	8	1	4	5	5	3	2	1	1				22
	9	1	4	7	6	5	3	2	1	1			30
	10	1	5	8	9	7	5	3	2	1	1		42
Total		10	25	31	27	19	12	7	4	2	1		138

Figure 23 : Exemple de calcul du nombre de configurations de placement

Le tableau de la Figure 23 fournit le détail de ces calculs pour :

$$\left| \begin{array}{l} nTailles = 10 \\ nbMaxTailles = 10 \\ nbMaxRep = 10 \end{array} \right.$$

On peut vérifier par exemple que :

$$\begin{aligned}
 nbConf_{10}^{10}(10,3) &= nbConf_{10}^{10}(9,2) + nbConf_{10}^{10}(6,2) + nbConf_{10}^{10}(3,2) \\
 &= 4 + 3 + 1 \\
 &= 8
 \end{aligned}$$

On peut également s'inspirer de cette méthode de calcul pour décrire précisément chaque configuration de placement. Par exemple, on a 4 configurations possibles correspondant aux placements contenant 9 tailles en tout avec uniquement 2 tailles différentes placées ($n = 9$ et $k = 2$).

- 1 taille placée 1 fois et une autre 8 fois (1,8)
- 1 taille placée 2 fois et une autre 7 fois (2,7)
- 1 taille placée 3 fois et une autre 6 fois (3,6)
- 1 taille placée 4 fois et une autre 5 fois (4,5)

De la même manière, pour $n = 8$ et $k = 2$, 4 configurations sont possibles : (1,7), (2,6), (3,5) et (4,4).

Pour $n = 7$ et $k = 2$, 3 configurations sont possibles : (1,6), (2,5) et (3,4).

Pour trouver toutes les configurations possibles pour $n = 10$ et $k = 3$, on procède de la façon suivante :

La taille qui est répétée le moins de fois peut l'être 1 fois, 2 fois ou 3 fois $\left(\left\lfloor \frac{10}{3} \right\rfloor = 3\right)$

- Si elle est répétée 1 fois, il reste 9 tailles à placer sur 2 tailles différentes. On a donc 4 configurations possibles : (1,1,8), (1,2,7), (1,3,6) et (1,4,5).
- Si elle est répétée 2 fois, il reste 8 tailles à placer sur 2 tailles différentes mais les tailles présentes doivent être répétée au moins 2 fois (sous peine de redondance des configurations). 3 configurations sont possibles : (2,2,6), (2,3,5) et (2,4,4).
- Si elle est répétée 3 fois, il reste 7 tailles à placer sur 2 tailles différentes et les tailles présentes doivent être répétée au moins 3 fois. 1 seule configuration est possible : (3,3,4).

Pour $n = 10$ et $k = 3$, les 8 configurations possibles sont :

- (1,1,8), (1,2,7), (1,3,6), (1,4,5)
- (2,2,6), (2,3,5), (2,4,4)
- et (3,3,4)

Là encore, on peut décrire de manière récursive toutes les configurations de placements possibles.

4.2.2 Calcul du nombre de placements dans une configuration

Une configuration de placements contenant n tailles réparties sur k tailles différentes peut être décrite par un ensemble de $nTailles$ entiers parmi lesquels k d'entre eux sont strictement

positifs : $\{p_j / 1 \leq j \leq nTailles\} / \left(\begin{array}{l} p_j \geq 0 \quad \forall j, \quad \sum_{j=1}^{nTailles} p_j = n \\ \exists j_1, \dots, j_{nTailles-k} / p_{j_i} = 0 \quad \forall 1 \leq i \leq nTailles - k \end{array} \right)$. Le nombre de

placements différents dans une même configuration est égal au nombre de permutations différentes de l'ensemble $\{p_j / 1 \leq j \leq nTailles\}$. Pour calculer ce nombre, il faut tenir compte du fait que certains des p_j peuvent être égaux entre eux. On définit \tilde{k} comme le nombre de valeurs différentes dans $\{p_j / 1 \leq j \leq nTailles\}$. Puis on décrit une configuration de placements

par un ensemble de \tilde{k} couples : $\{(\tilde{p}_j, m_j) / 1 \leq j \leq \tilde{k}\} / \left(\begin{array}{l} \tilde{p}_j \geq 0 \quad \forall j, \quad \sum_{j=1}^{\tilde{k}} m_j \times \tilde{p}_j = n \\ m_j \geq 1 \quad \forall j, \quad \sum_{j=1}^{\tilde{k}} m_j = nTailles \end{array} \right)$ où m_j est

le nombre de fois où l'entier \tilde{p}_j est répétée dans la configuration. On pose

$s_0 = 0$ et $s_i = \sum_{j=1}^i m_j \quad \forall 1 \leq i \leq \tilde{k}$. Avec ces notations, le nombre de placements dans une

configuration est donné par : $\prod_{i=1}^{\tilde{k}} C_{nTailles-s_{i-1}}^{m_i} = \frac{nTailles!}{\prod_{i=1}^{\tilde{k}} (m_i!)}$. En effet, on doit placer au départ m_1

fois la quantité \tilde{p}_1 parmi toutes les tailles disponibles : on a donc $C_{nTailles}^{m_1}$ possibilités. Ensuite,

on place m_2 fois la quantité \tilde{p}_2 mais le nombre de tailles disponibles est désormais $nTailles - m_1$: on a donc $C_{nTailles - m_1}^{m_2}$ possibilités. On procède ainsi jusqu'à \tilde{k} et on obtient le résultat précédent.

Exemple : On suppose que $nTailles = 10$. On calcule le nombre de placements ayant la configuration (1,1,1,2,2,3). Cela revient à dire que pour chaque placement ayant cette configuration, parmi les 10 tailles de l'OF :

- 3 sont répétées 1 fois
- 2 sont répétées 2 fois
- 1 est répétée 3 fois
- 4 ne sont pas présentes sur le placement

	36	38	40	42	44	46	48	50	52	54
P1	0	0	1	2	2	3	1	1	0	0
P2	1	0	2	0	1	2	3	0	0	1

Figure 24 : Exemple de placements de même configuration

Les 2 placements de la Figure 24 ont cette configuration, mais le nombre total des placements

qui l'ont est : $N = \frac{10!}{1!2!3!4!} = 12600$.

La méthode de calcul du nombre de placement ayant une même configuration suggère également un algorithme d'énumération permettant de décrire tous les placements ayant cette configuration.

4.2.3 Analyse des cas où la recherche exhaustive est possible

Comme on l'a vu dans les 2 paragraphes précédents, le nombre total de placements réalisables dépend principalement :

- Du nombre de colonnes dans la matrice des quantités de l'OF : $nTailles$
- Des contraintes sur la création des nouveaux placements et plus précisément :

- Du nombre maximum de tailles autorisé sur un placement : $nbMaxTailles$
- Du nombre maximum de répétitions d'une même taille sur un placement : $nbMaxRep$

On reprend l'exemple de la **Figure 23** avec les valeurs des paramètres suivants :

$$\left| \begin{array}{l} nTailles = 10 \\ nbMaxTailles = 10 \\ nbMaxRep = 10 \end{array} \right.$$

Le nombre de configurations de placements dans ce cas est 138 et le nombre total de placements réalisables est 184 755 la configuration la plus fournie est celle décrite sur la **Figure 24**, elle contient 12600 placements à elle seule. Ce premier exemple montre que des paramètres peu élevés peuvent cependant impliquer un nombre très important de placements réalisables. Il n'est pas envisageable dans ce cas de travailler avec la totalité des placements réalisables.

En revanche, le nombre de colonnes dans la matrice des quantités de l'OF est 2 fois plus petit. On obtient le jeu de paramètres suivants :

$$\left| \begin{array}{l} nTailles = 5 \\ nbMaxTailles = 10 \\ nbMaxRep = 10 \end{array} \right.$$

Bien que le nombre de configurations de placements (112) reste conséquent, le nombre total de placements réalisables n'est plus que 3002, et la configuration la plus fournie contient 120 placements seulement. La diminution du nombre de colonnes influe peu sur le nombre de configurations en revanche, le nombre de placements qu'elles contiennent à beaucoup diminué. Dans ce cas il est possible de travailler avec la totalité des placements réalisables.

Un autre cas favorable se produit lorsque les contraintes sur la création des nouveaux placements sont fortes. On utilise le jeu de paramètres suivant :

$$\left| \begin{array}{l} nTailles = 10 \\ nbMaxTailles = 5 \\ nbMaxRep = 2 \end{array} \right.$$

Il n'y a alors plus que 11 configurations de placements. Le nombre total de placements réalisables est 2342 et la configuration la plus fournie contient 840 placements. Plus les contraintes sur la création des nouveaux placements sont restrictives, plus le nombre de configurations de placements baisse bien qu'elles contiennent toujours un nombre relativement important de placement. Dans ce cas il est également possible de travailler avec la totalité des placements réalisables.

Dans la pratique, il est facile d'avoir un ordre de grandeur du nombre total de placements réalisables en fonction des 3 paramètres décrits ci-dessus. On effectuera une recherche exhaustive lorsque ce nombre n'excède pas quelques milliers ($< 10\,000$). Dans le cas contraire, on effectuera une recherche partielle des placements réalisables.

4.3 Recherche par « configuration de placement »

4.3.1 Problème de l'augmentation du nombre de colonnes de l'OF

On a vu dans le paragraphe précédent que l'augmentation de $nTailles$ (le nombre de colonnes de la matrice de l'OF) implique une forte augmentation du nombre de placements réalisables. Pour plus de précision, on fixe les 2 autres paramètres qui influencent le nombre de placements et on fait varier $nTailles$. On pose donc :

$$\left| \begin{array}{l} nbMaxTailles = 4 \\ nbMaxRep = 2 \end{array} \right.$$

On a alors le choix entre 8 configurations de placements différentes qui sont :

- Placements mono-taille : $c1 = (1)$
- Placements 2-tailles : $c2 = (2)$, $c3 = (1,1)$
- Placements 3-tailles : $c4 = (1,2)$, $c5 = (1,1,1)$

- Placements 4-tailles : $c_6 = (2,2)$, $c_7 = (1,1,2)$, $c_8 = (1,1,1,1)$

On calcule le nombre de placements dans chacune de ces configurations en fonction de $nTailles$. On obtient en posant $N(c) =$ nombre de placements ayant la configuration c :

$$\left| \begin{array}{l} N(c_1) = N(c_2) = C_{nTailles}^1 = nTailles \\ N(c_3) = N(c_6) = C_{nTailles}^2 = \frac{nTailles(nTailles-1)}{2} \\ N(c_4) = C_{nTailles}^1 \times C_{nTailles-1}^1 = nTailles \times (nTailles-1) \\ N(c_5) = C_{nTailles}^3 = \frac{nTailles(nTailles-1)(nTailles-2)}{6} \\ N(c_7) = C_{nTailles}^2 \times C_{nTailles-2}^1 = \frac{nTailles(nTailles-1)}{2} \times (nTailles-2) \\ N(c_8) = C_{nTailles}^4 = \frac{nTailles(nTailles-1)(nTailles-2)(nTailles-3)}{24} \end{array} \right.$$

Finalement, le nombre total de placements réalisables s'exprime comme un polynôme de degré 4 en $nTailles$. Dans ce cas on a :

$$NbPlacements(nTailles) = \frac{1}{24} (nTailles^4 + 10 \times nTailles^3 + 11 \times nTailles^2 + 26 \times nTailles)$$

En réalité, le degré du polynôme donnant le nombre de placements est égal au nombre maximum de tailles acceptées dans un placement : $nbMaxTailles$. Le graphique de la **Figure 25** montre l'augmentation du nombre de placements réalisables lorsque $nTailles$ augmente.

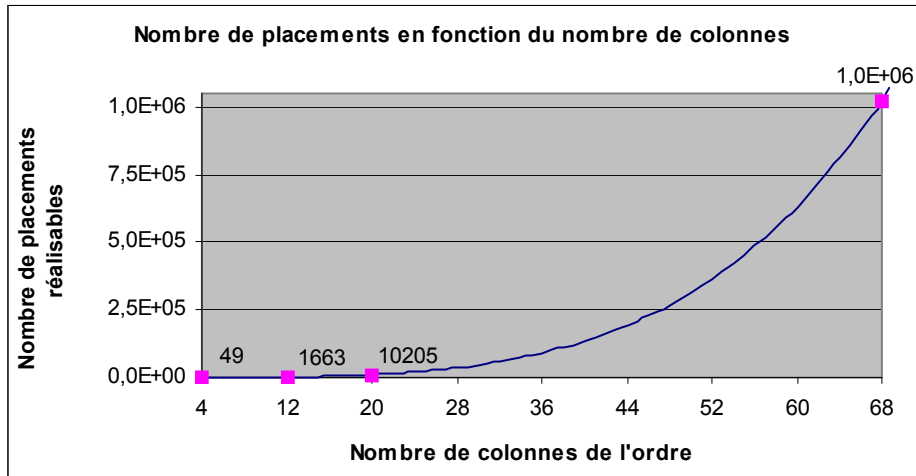


Figure 25 : Evolution du nombre de placements en fonction de $nTailles$

Les contraintes sur la création de nouveaux placements dans ce cas sont fortes et limitent le nombre de configurations à 8. Malgré cela, le nombre de placements réalisables est déjà supérieur à 10 000 pour $nTailles = 20$ et il dépasse le million pour $nTailles = 68$.

Ainsi, lorsque le nombre de configurations est limité par les contraintes de création mais que $nTailles$ est trop grand pour pouvoir effectuer une recherche exhaustive des placements, on choisira de sélectionner une partie seulement des placements réalisables pour chaque configuration. Le paragraphe suivant présente une méthode pour sélectionner les placements qui seront retenus.

4.3.2 Sélection d'un échantillon de placements pour une configuration donnée

Lorsque l'on ne sélectionne qu'un échantillon des placements pour une configuration donnée, on court en effet le risque d'éliminer certains placements qui seraient présents dans la solution optimale théorique. Pour limiter au maximum ces risques, les placements sélectionnés ne seront pas choisis au hasard mais selon les quantités de l'OF. En effet, en étudiant la configuration de la matrice des quantités à produire par taille et par tissu, il faut essayer de retenir les placements qui auront le plus de chance d'être utilisés dans la solution.

4.3.2.1 Placement permettant de découper le plus de plis pour un tissu donné

On cherche à trouver dans une configuration donnée, le placement qui permet de découper le plus de plis (et donc de vêtements) en fonction des quantités demandées pour une ligne (un tissu) de la matrice de l'OF. On a donc :

- Les quantités à découper dans le tissu souhaité : $\{b_j / 1 \leq j \leq nTailles\}$
- Les répétitions de tailles de la configuration de placement : $\{p_j / 1 \leq j \leq nTailles\}$

Le nombre de plis, qu'il est possible de découper avec un placement p de cette configuration,

$$\text{est donné par : } NbPlis(p) = \underset{\substack{1 \leq j \leq nTailles \\ p_j > 0}}{\text{Min}} \left(\left\lfloor \frac{b_j}{p_j} \right\rfloor \right)$$

Il est facile de démontrer que pour découper le plus de plis possibles, il faut classer les quantités à produire b_j ainsi que les répétitions de tailles p_j par ordre décroissant. On associe ensuite à chaque quantité à produire la répétition de taille de même rang dans cette classification¹.

Le tableau de la **Figure 26** représente les quantités b_j pour un tissu d'un OF contenant 6 tailles ainsi que le placement permettant de découper le plus de plis pour la configuration (1,1,1,2). Il permet de découper 16 plis et donc 80 vêtements.

	XS	S	M	L	XL	XXL	
B _j	12	17	23	32	24	19	
P _j	0	0	1	2	1	1	NbPlis
B _j / P _j			23	16	24	19	16

Figure 26 : Exemple de placement permettant de découper le plus de plis pour un tissu donné

Cette méthode de sélection des placements permet d'obtenir un placement pour chaque configuration permise et chaque tissu de l'OF. Cependant, le fait de sélectionner

¹ Comme la règle énoncée est transitive, pour prouver qu'elle est optimale, il faut montrer que l'on augmente le nombre de plis en inversant 2 quantités qui la contredisent.

exclusivement les placements qui vont permettre de découper le plus de vêtements est trop restrictif, parfois la qualité d'un placement se mesure par le fait qu'il permet d'éliminer des restes apparus après l'utilisation de placements précédents. Pour pallier ce problème, nous allons compléter cette première sélection par une deuxième moins déterministe qui va donner plus de liberté pour l'utilisation des placements.

4.3.2.2 Système de poids pour la sélection des placements

Le but est de sélectionner, pour chaque configuration de placement et pour chaque tissu de l'OF non pas un, mais toute une liste de placements (la taille de cette liste peut-être fixée a priori ou donnée par l'utilisateur suivant les spécificités du problème traité).

Le principe consiste à attribuer pour chaque taille de l'OF une probabilité de choisir de placer cette taille dans le placement sélectionné. Cette probabilité est basée sur le nombre de vêtements à découper pour la taille et le tissu considéré. Ainsi, pour une taille et un tissu donné, plus le nombre de vêtements à découper est important plus la probabilité de placer cette taille sur un placement sera importante.

De plus il y a également la question de savoir combien de fois on va faire apparaître une taille sur un placement. Dans une configuration (2,1,1) par exemple, trois tailles différentes apparaîtront sur le placement sélectionné mais une de ces trois tailles sera placée deux fois. L'idée est que plus le nombre de répétitions est important, plus la loi de probabilité trouvée devra favoriser les tailles où le nombre de vêtements à découper est important. On choisit de définir ces probabilités en utilisant les puissances en posant :

$$p(i, k) = \frac{(b_i)^k}{\sum_{l=1}^{n_{Tailles}} (b_l)^k} \quad \left| \begin{array}{l} \text{où } p(i, k) \text{ est la probabilité de choisir la taille } i \text{ pour être répétée } k \text{ fois} \\ \text{sur le placement sélectionné} \end{array} \right.$$

On vérifie facilement qu'il s'agit bien d'une loi de probabilité sur l'ensemble des tailles de l'OF :

$$\forall 1 \leq i \leq nTailles, 0 \leq p(i, k) \leq 1 \text{ et } \sum_{i=1}^{nTailles} p(i, k) = \sum_{i=1}^{nTailles} \left(\frac{(b_i)^k}{\sum_{l=1}^{nTailles} (b_l)^k} \right) = \frac{\sum_{i=1}^{nTailles} (b_i)^k}{\sum_{l=1}^{nTailles} (b_l)^k} = 1$$

Le fait d'élever les b_i à la puissance k permet d'accentuer l'importance des tailles où le nombre de vêtements à produire est important lorsque le nombre de répétitions de la taille sur le placement augmente. Le graphique de la **Figure 27** représente les différentes lois de probabilités obtenues pour les quantités de la **Figure 26** en fonction du nombre de répétitions de la taille choisie¹.

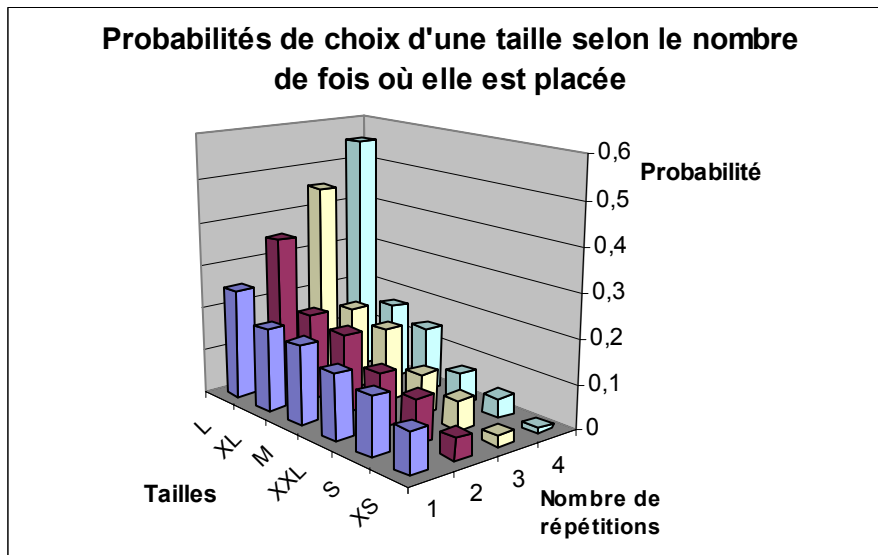


Figure 27 : Différentes lois de probabilités sur le choix des tailles suivant le nombre de répétitions

Pour une seule répétition, on obtient une distribution assez équilibrée, proportionnelle au nombre de vêtements demandés. Lorsque le nombre de répétitions augmente, la probabilité de choisir la taille L (la plus demandée) augmente fortement alors que la probabilité de choisir la taille XS (la moins demandée) se rapproche de zéro. Pour les tailles intermédiaires, les probabilités varient relativement peu.

¹ Pour une meilleure lisibilité du graphique, les différentes tailles de l'ordre sont classées par ordre décroissant du nombre de vêtements demandés.

Grâce à ce système, on peut facilement mettre au point une méthode permettant de trouver pour chaque configuration de placement et chaque tissu de l'OF, une sélection de placements différents.

Pour cela, on fixe le nombre de placements $nbPla$ que l'on veut sélectionner pour chaque configuration et chaque tissu de l'OF. Le nombre total de placements retenus sera donc : $nbPlaTotal = nbPla \times nbConfig \times nTissus$. On reprend l'exemple de la **Figure 26** pour sélectionner un autre placement ayant pour configuration (1,1,1,2). On choisit en premier lieu la taille qui sera répétée 2 fois sur le placement. On calcule la loi de probabilité sur les tailles et sa fonction de répartition pour 2 répétitions. On obtient le tableau de la **Figure 28**:

	XS	S	M	L	XL	XXL
p(i,2)	0,049	0,099	0,181	0,35	0,197	0,124
F(i,2)	0,049	0,148	0,329	0,679	0,876	1

Figure 28 : Loi de probabilité de choix d'une taille répétée 2 fois

On calcule ensuite une réalisation d'une variable aléatoire x qui suit une loi uniforme sur $[0,1]$, on obtient par exemple $0,679 \leq x = 0,85 < 0,876$ c'est donc la taille XL qui sera répétée 2 fois.

On calcule maintenant la loi de probabilité sur les tailles (sauf la taille XL) et sa fonction de répartition pour 1 répétition. On obtient le tableau de la **Figure 29** :

	XS	S	M	L	XXL
p(i,2)	0,117	0,165	0,223	0,311	0,184
F(i,2)	0,117	0,282	0,505	0,816	1

Figure 29 : Loi de probabilité de choix d'une taille répétée une fois

On calcule de nouveau une réalisation d'une variable aléatoire x qui suit une loi uniforme sur $[0,1]$, on obtient par exemple $0,117 \leq x = 0,22 < 0,282$ la taille S sera donc répétée 1 fois.

On réitère ce processus pour les 2 tailles restant à placer en supprimant à chaque fois les tailles déjà choisies. On obtient par exemple le placement P de la **Figure 30** :

	XS	S	M	L	XL	XXL
P	1	1	1	0	2	0

Figure 30 : Exemple de placement sélectionné avec les poids

On répète cette méthode jusqu'à obtenir $nbPla$ placements différents pour chaque configuration et chaque tissu de l'OF. Cette technique permet de réduire significativement le nombre de placements que l'on va retenir. La difficulté est alors de fixer le nombre de placements que l'on souhaite garder pour chaque configuration et chaque tissu de l'OF. En effet, si l'on choisit un nombre de placements trop faible on risque de laisser de côté des placements qui auraient été très utiles. En revanche, si on génère un trop grand nombre de placements, on se rapproche d'une sélection exhaustive avec les mêmes difficultés que cela comporte.

4.4 Recherche par nombre de tailles

4.4.1 Problème de l'augmentation du nombre de configurations

On s'intéresse dans ce paragraphe à l'influence des contraintes de création de nouveaux placements sur le nombre de configurations possibles. On choisit de fixer le nombre de colonnes de l'OF à une valeur faible, on pose : $nTailles = 5$. En revanche, on fait varier le nombre de tailles autorisé dans un placement $nbMaxTailles$, de plus le nombre de répétitions d'une même taille n'est pas limité c'est à dire : $nbMaxRep = nbMaxTailles$.

Le nombre de colonnes étant fixe, le nombre de placements dans une configuration donné est borné par $nTailles! = 120$ ce qui est raisonnable. En revanche, dans ce cas c'est le nombre de configurations qui va fortement augmenter en fonction de $nbMaxTailles$.

On sait calculer le nombre total de configurations de placements avec la formule :

$$\sum_{n=1}^{nbMaxTailles} \sum_{k=1}^{K \max_n} nbConf_{nbMaxTailles, nbMaxRep}^{nTailles}(n, k) \text{ avec } K \max_n = \min(n, nTailles). \text{ Le graphique de la}$$

Figure 31 représente ce nombre en fonction de $nbMaxTailles$.

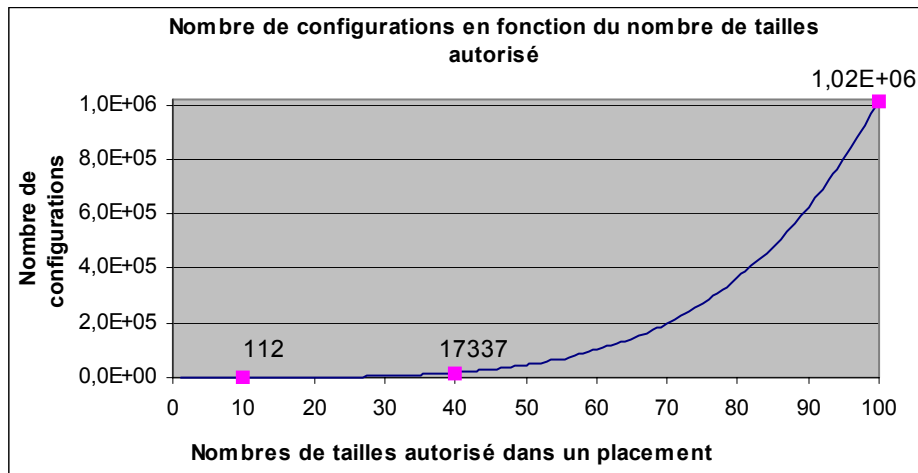


Figure 31 : Evolution du nombre de configurations en fonction de $nbMaxTailles$

Bien que $nTailles$ soit limité à 5, le nombre de configurations de placement est supérieur à 17000 pour $nbMaxTailles = 40$ et il dépasse le million pour $nMaxTailles = 100$.

Ainsi, lorsque le nombre de configurations devient trop important, on ne peut effectuer ni une recherche exhaustive des placements réalisables ni même une recherche par configuration. On choisira alors de retenir une liste de placements uniquement pour chaque nombre de taille autorisé. Le paragraphe suivant présente une méthode pour sélectionner les placements qui seront retenus.

4.4.2 Sélection d'un échantillon de placements pour un nombre de tailles donné

Le but de ce paragraphe est de sélectionner des placements s'accordant au mieux avec les quantités à produire du bordereau mais sans se préoccuper désormais de configurations de placement. Pour cela, on met en place une technique itérative, qui permet de sélectionner un placement contenant $(n+1)$ tailles à partir d'un placement contenant n tailles.

4.4.2.1 Placement n -tailles permettant de découper le plus de plis pour chaque tissu

L'idée est la suivante, à chaque fois que l'on va rajouter une taille sur le placement, le nombre de plis va devenir inférieur (ou égal dans le meilleur des cas) à ce qu'il était au rang précédent. On va donc essayer de trouver la taille du bordereau pour laquelle la réduction du nombre de plis sera la plus faible possible. On pose :

- Les quantités à découper dans le tissu souhaité : $\{b_j / 1 \leq j \leq nTailles\}$
- Les répétitions de tailles d'un placement contenant n tailles : $\{p_j^n / 1 \leq j \leq nTailles\}$

Le nombre de plis, qu'il est possible de découper avec le placement p^n est donné par :

$$NbPlis(p^n) = \underset{\substack{1 \leq j \leq nTailles \\ p_j^n > 0}}{\text{Min}} \left(\left\lfloor \frac{b_j}{p_j^n} \right\rfloor \right)$$

On peut déjà noter que la suite des $(NbPlis(p^n))_{n \geq 1}$ est une suite décroissante, en effet par principe, on a :

$$\forall 1 \leq j \leq nTailles, \forall n \geq 1, \quad p_j^{n+1} \geq p_j^n$$

et donc :

$$\forall 1 \leq j \leq nTailles, \forall n \geq 1, \quad \left\lfloor \frac{b_j}{p_j^{n+1}} \right\rfloor \leq \left\lfloor \frac{b_j}{p_j^n} \right\rfloor$$

$$\text{et finalement, } \forall n \geq 1, \quad \underset{\substack{1 \leq j \leq nTailles \\ p_j^{n+1} > 0}}{\text{Min}} \left(\left\lfloor \frac{b_j}{p_j^{n+1}} \right\rfloor \right) \leq \underset{\substack{1 \leq j \leq nTailles \\ p_j^n > 0}}{\text{Min}} \left(\left\lfloor \frac{b_j}{p_j^n} \right\rfloor \right)$$

Ainsi, pour construire des placements permettant de découper beaucoup de plis, il faut essayer à chaque étape, de perdre le moins de plis possibles. Soit p^n un placement contenant n tailles. Supposons que l'on rajoute une fois la taille j pour obtenir le placement p^{n+1} . Le nombre de plis possibles avec p^{n+1} est alors donné par la formule :

$$NbPlis(p^{n+1}) = \text{Min} \left(NbPlis(p^n), \left\lfloor \frac{b_j}{p_j^n + 1} \right\rfloor \right)$$

En effet, soit c'est la quantité b_j qui est bloquante pour le nombre de plis du fait de l'augmentation de la répétition de la taille j , soit ce sont les autres tailles qui sont bloquantes.

On montre par une récurrence sur n que la stratégie optimale pour obtenir le placement contenant n tailles permettant de découper le plus de plis est de rajouter au placement optimal

$$p^{n-1} \text{ une répétition sur la taille } j^* \text{ qui vérifie : } \left\lfloor \frac{b_{j^*}}{p_j^{n-1} + 1} \right\rfloor = \mathit{Max}_{1 \leq j \leq n \text{Tailles}} \left(\left\lfloor \frac{b_j}{p_j^{n-1} + 1} \right\rfloor \right).$$

En effet, pour $n = 1$, le placement mono-taille qui permet de découper le plus de plis est évidemment qui contient la taille la plus demandée j^* qui vérifie :

$$\left\lfloor \frac{b_{j^*}}{0+1} \right\rfloor = \mathit{Max}_{1 \leq j \leq n \text{Tailles}} \left(\left\lfloor \frac{b_j}{0+1} \right\rfloor \right).$$

Supposons maintenant que p^{n-1} est optimal parmi tous les placements contenant $n-1$ tailles, c'est à dire : $NbPlis(p^{n-1}) \geq NbPlis(\tilde{p}^{n-1}) \quad \forall \tilde{p}^{n-1}$ contenant $n-1$ tailles .

On construit p^n en rajoutant une répétition de la taille j^* qui vérifie :

$$\left\lfloor \frac{b_{j^*}}{p_j^{n-1} + 1} \right\rfloor = \mathit{Max}_{1 \leq j \leq n \text{Tailles}} \left(\left\lfloor \frac{b_j}{p_j^{n-1} + 1} \right\rfloor \right).$$

Si $\left\lfloor \frac{b_{j^*}}{p_j^{n-1} + 1} \right\rfloor = NbPlis(p^{n-1})$ alors $NbPlis(p^n) = NbPlis(p^{n-1})$ et p^n est optimal.

Si $\left\lfloor \frac{b_{j^*}}{p_j^{n-1} + 1} \right\rfloor < NbPlis(p^{n-1})$ alors p^n est optimal parmi les placements issus de p^{n-1}

par construction (on rajoute une répétition de telle manière que le nombre de plis diminue le moins possible). De plus,

$$\forall 1 \leq i \leq n \text{Tailles}, \left\lfloor \frac{b_i}{p_i^{n-1} + 1} \right\rfloor \leq \left\lfloor \frac{b_{j^*}}{p_j^{n-1} + 1} \right\rfloor = NbPlis(p^n)$$

Soit \tilde{p}^{n-1} un placement contenant $n-1$ tailles, différent de p^{n-1} . Montrons que, $\forall \tilde{p}^n$ issu de \tilde{p}^{n-1} on a : $NbPlis(p^n) \geq NbPlis(\tilde{p}^n)$.

\exists une taille i^* telle que $\tilde{p}_i^n = \tilde{p}_i^{n-1} + 1$ (la taille répétée une fois de plus sur \tilde{p}^n que sur \tilde{p}^{n-1}).

Si $p_i^{n-1} \leq \tilde{p}_i^{n-1}$

$$p_i^{n-1} + 1 \leq \tilde{p}_i^n \text{ et donc: } NbPlis(\tilde{p}^n) \leq \left\lfloor \frac{b_{i^*}}{\tilde{p}_i^n} \right\rfloor \leq \left\lfloor \frac{b_{i^*}}{p_i^{n-1} + 1} \right\rfloor \leq NbPlis(p^n)$$

Si $p_i^{n-1} > \tilde{p}_i^{n-1}$, comme les deux placements ont le même nombre de tailles,

\exists une taille k telle que $p_k^{n-1} < \tilde{p}_k^{n-1}$

$$\text{On a alors: } p_k^{n-1} + 1 \leq \tilde{p}_k^{n-1} \text{ d'où } NbPlis(\tilde{p}^n) \leq \left\lfloor \frac{b_k}{\tilde{p}_k^{n-1}} \right\rfloor \leq \left\lfloor \frac{b_k}{p_k^{n-1} + 1} \right\rfloor \leq NbPlis(p^n)$$

Et donc, p^n est optimal parmi tous les placements contenant n tailles.

4.4.2.2 Exemple

La **Figure 32** représente un OF contenant 8 tailles et un seul tissu :

Tailles	44	46	48	50	52	54	56	58
Quantités demandées	60	80	100	125	140	100	116	80

Figure 32 : Exemple de quantités demandées pour un tissu

Les placements obtenus pour 10, 20 et 63 tailles sont :

- $p^{(10)} = (1, 1, 1, 2, 2, 1, 1, 1)$ avec 60 plis (600 vêtements découpés)
- $p^{(20)} = (1, 2, 3, 4, 4, 2, 3, 2)$ avec 33 plis (660 vêtements découpés)
- $p^{(63)} = (5, 6, 8, 10, 11, 8, 9, 6)$ avec 12 plis (756 vêtements découpés)

Cette méthode de sélection des placements permet d'obtenir un placement pour chaque nombre de tailles autorisé et chaque tissu de l'OF. Cependant, pour les mêmes raisons évoquées lors de la sélection des placements par configuration, le fait de retenir uniquement les placements qui vont permettre de découper le plus de vêtements est trop restrictif. On va donc compléter de nouveau cette première sélection par une deuxième moins déterministe qui va donner plus de liberté pour l'utilisation des placements.

4.4.2.3 Système de poids pour la sélection des placements

Le but est de sélectionner, pour chaque nombre de tailles et pour chaque tissu de l'OF non pas un mais toute une liste de placements (la taille de cette liste peut-être fixée a priori ou donnée par l'utilisateur suivant les spécificités du problème traité).

Le principe consiste à attribuer pour chaque taille de l'OF une probabilité de choisir de placer cette taille dans le placement sélectionné. Cette probabilité est basée sur le nombre de vêtements à découper pour la taille et le tissu considéré. Ainsi, pour une taille et un tissu donné, plus le nombre de vêtements à découper est important plus la probabilité de placer cette taille sur un placement sera importante.

Contrairement au cas de la sélection par configuration de placement, le but ici est de sélectionner les placements de manière itérative en ajoutant à chaque fois une répétition de taille au placement obtenu précédemment. Pour cela, on définit les quantités suivantes :

p^n placement contenant n tailles: $\{p_j^n / 1 \leq j \leq nTailles\}$

$$b_j^n = b_j - NbPlis(p^n)$$

$$prob_j^{n+1} = \frac{b_j^n}{\sum_{i=1}^{nTailles} b_i^n}$$

Les quantités b_j^n correspondent au nombre de vêtement restant à découper dans la taille j si l'on utilise le placement p^n au maximum. $prob_j^{n+1}$ est alors la probabilité de rajouter la taille j à p^n pour obtenir p^{n+1} . On vérifie que l'on a bien, pour tout n , une loi de probabilité sur l'ensemble des tailles de l'OF :

$$\forall 1 \leq j \leq nTailles, \quad 0 \leq prob_j^n \leq 1 \text{ et } \sum_{j=1}^{nTailles} prob_j^n = \sum_{j=1}^{nTailles} \left(\frac{b_j^n}{\sum_{i=1}^{nTailles} b_i^n} \right) = \frac{\sum_{j=1}^{nTailles} b_j^n}{\sum_{i=1}^{nTailles} b_i^n} = 1$$

On fixe le nombre de placements $nbPla$ que l'on veut sélectionner pour chaque nombre de tailles et chaque tissu de l'OF. Le nombre total de placements retenus sera donc : $nbPlaTotal = nbPla \times nbMaxTailles \times nTissus$. Pour chaque tissu, on sélectionne itérativement $nbMaxTailles$ placements contenant de 1 à $nbMaxTailles$ tailles, puis on répète cette procédure $nbPla$ fois pour obtenir le nombre de placements souhaité.

Chapitre 5 : Nouvelle méthode de résolution

Ce chapitre a pour but de décrire la démarche globale de résolution du problème de la construction des sections. Durant cette phase, on dispose d'une bibliothèque de placements P qui est le résultat de la recherche des placements réalisables décrite dans le *Chapitre 4 : Méthodes de recherche des placements*. P peut donc représenter l'ensemble de tous les placements réalisables ou bien seulement une sélection d'entre eux. Dans un premier temps, on cherche à démarrer l'algorithme avec une première solution réalisable du problème. La partie la plus importante du chapitre est consacrée à l'adaptation de l'algorithme du recuit simulé au problème spécifique de la construction de sections ainsi qu'à la description du sous-problème à traiter à chaque itération. On met en place une nouvelle méthode de résolution qui permet une optimisation globale du problème de la construction des sections. Dans toute la suite, on fera référence à cette méthode par le sigle SPGO initiales en anglais de « Section Planning Global Optimisation ».

5.1 Recherche d'une solution initiale

Cette première solution doit être un compromis entre qualité et rapidité de calcul. En effet, étant donné le temps limité dont on dispose, il est intéressant de démarrer l'algorithme avec une solution si possible pas trop éloignée de la solution optimale. Cependant, la recherche de cette solution ne doit pas être trop complexe car le temps consacré à calculer cette solution est du temps en moins pour l'algorithme du recuit simulé. On choisit une méthode de construction pas à pas des sections, proche de la méthode de résolution actuelle. A chaque itération on cherche le meilleur placement et on l'utilise dans une section. Cependant, le critère de choix du placement à utiliser est simplifié à l'extrême, on choisit à chaque itération le placement, de la bibliothèque de placements P , permettant de découper le plus de vêtements et on l'utilise au maximum.

On note B_{ji} la quantité demandée dans la taille i $1 \leq i \leq nTailles$ et le tissu j $1 \leq j \leq nTissus$. Le nombre de plis maximum que l'on peut découper avec un placement p est la somme des nombres de plis que l'on peut découper avec p pour chaque tissu, c'est à dire :

$$NbPlis_B(p) = \sum_{j=1}^{nTissus} NbPlis_B(p, j) = \sum_{j=1}^{nTissus} \text{Min} \left(\left\lfloor \frac{B_{ji}}{p_i} \right\rfloor \right)$$

Le nombre de vêtements (ou d'articles) découpés au maximum avec p est ce nombre de plis multiplié par le nombre de tailles de p : $NbVet_B(p) = NbPlis_B(p) \times \sum_{i=1}^{nTailles} p_i$

On appelle p^k le $k^{\text{ième}}$ placement utilisé dans la solution initiale. On calcule la $k^{\text{ième}}$ matrice des quantités résiduelles : $B_{ji}^k = B_{ji}^{k-1} - p_j^k \times NbPlis_{B^{k-1}}(p, j)$.

On choisit alors le placement p^{k+1} qui vérifie : $NbVet_{B^k}(p^{k+1}) = \text{Max}_{p \in P} (NbVet_{B^k}(p))$.

La recherche s'arrête lorsque la matrice des quantités résiduelles est nulle.

5.2 Construction des sections et recuit simulé

5.2.1 Démarche globale

L'idée générale de la résolution est d'utiliser l'algorithme du recuit simulé pour décomposer le problème initial en une résolution successive de sous-problèmes. La bibliothèque P pouvant contenir plusieurs milliers de placements réalisables, un sous-problème correspond à la construction des sections en utilisant uniquement un échantillon P_{ech} de P composé de quelques dizaines de placements au maximum. Le fait de modifier légèrement l'échantillon de placements P_{ech} permet alors de passer d'une solution à une autre dans l'algorithme du recuit simulé. Pour cela on définit les notions de voisinages suivantes :

- Deux bibliothèques de placements sont voisines si elles sont de même taille et si elles diffèrent au plus d'un placement.

- Deux solutions sont voisines si on les obtient avec des bibliothèques de placements voisines.

Ainsi, pour passer d'une solution à une autre dans le recuit simulé, on choisit aléatoirement un placement dans P_{ech} , et on le remplace par un autre placement de P choisi lui aussi de manière aléatoire. On résout alors le problème avec ce nouvel échantillon de placements et on teste la solution trouvée avec la règle de Métropolis.

5.2.2 Paramètres du recuit simulé

5.2.2.1 Décroissance de la température

L'algorithme du recuit simulé se décompose en plusieurs paliers durant lesquels la température T est constante. Lorsque l'on change de palier, T diminue ce qui rend moins probable l'acceptation d'une solution qui dégrade la solution optimale courante. Le schéma de décroissance de la température que l'on choisit est très simple :

$$T_{k+1} = cT_k \quad \begin{cases} 0 < c < 1 \\ T_k \text{ est la température du } k^{\text{ième}} \text{ palier} \end{cases}$$

Comme le montre le graphique de la **Figure 33**, la valeur du paramètre c détermine la vitesse de décroissance de la température.

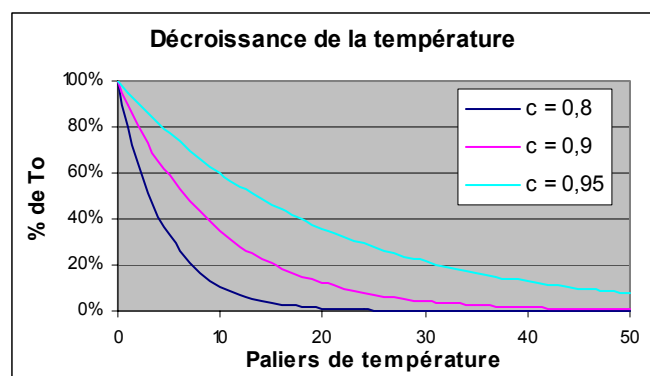


Figure 33 : Schéma de décroissance de T

On peut également choisir d'avoir toujours la même température $T_{end} = t$ pour le dernier palier. Comme le nombre de paliers $NbPaliers$ est donné par l'utilisateur, cela permet de

calculer la valeur de c par : $t = T_0 \times c^{NbPaliers} \Rightarrow c = \left(\frac{t}{T_0}\right)^{\frac{1}{NbPaliers}}$.

Indépendamment du temps de résolution, on a bien entendu intérêt à avoir le plus de paliers possibles car la diminution de température est alors moins brusque.

5.2.2.2 Initialisation de la température

La règle d'acceptation de Métropolis accepte une solution qui dégrade la solution optimale courante avec une probabilité : $\exp\left(-\frac{\Delta f}{T}\right)$. Δf est l'ampleur de la dégradation. Le paramètre de température T est primordial, il contrôle le déroulement de l'algorithme du recuit simulé. Si T est très élevée, toutes les dégradations sont acceptées et l'algorithme n'est plus guidé vers les bonnes solutions. A l'inverse, si T est très faible, aucune dégradation n'est acceptée et on retrouve un algorithme de descente pure avec tous les risques de rester piégé dans des minimums locaux. Comme T diminue lentement au cours du recuit simulé, il est moins préjudiciable de démarrer avec une température initiale trop élevée que trop basse. Cependant, dans ce cas, les premiers paliers de température sont inutiles et l'on perd du temps. On va donc chercher dès le départ à ajuster T le mieux possible au problème traité.

Dans cette optique, on réalise en début d'algorithme une série d'itérations pour lesquelles toutes les dégradations sont acceptées ($T = +\infty$) et on calcule pour cette série la médiane M des différentes dégradations observées. Ensuite, pour fixer la température initiale T_0 on choisit une probabilité p d'accepter en moyenne une dégradation. Si on exprime cette probabilité moyenne par $p = \exp\left(\frac{-M}{T_0}\right)$ on obtient : $T_0 = \frac{-M}{\ln(p)}$.

Il reste toutefois à choisir la probabilité moyenne d'accepter une dégradation. Couramment, les valeurs de p sont choisies assez grandes (entre 0,8 et 1). Cependant, dans notre cas, la solution initiale choisie n'est pas quelconque (cf. 5.1 : Recherche d'une solution initiale) et est

déjà en un sens le résultat d'une première optimisation sommaire. Si la valeur de p est grande, on arrive rarement à améliorer la solution initiale lors des premiers paliers de température car les solutions testées sont trop mauvaises en moyenne. On choisit donc une valeur de p assez faible entre 0,3 et 0,4.

5.2.2.3 Nombre et longueur des paliers de température

La longueur d'un palier de température est le nombre d'itérations réalisées avec la même valeur de T . On choisit d'augmenter la longueur des paliers au fur et à mesure que la température décroît. On favorise ainsi l'exploration de l'espace des solutions lorsque les dégradations sont peu probables et que le nombre de solutions acceptées est plus faible. De façon arbitraire, on impose que le dernier palier de température soit deux fois plus long que le premier. La longueur des autres paliers est alors calculée proportionnellement par :

$$\forall 1 \leq k \leq NbPaliers : LongPalier_k = \left\lfloor LongPalier_1 \left(1 + \frac{k-1}{NbPaliers-1} \right) \right\rfloor$$

Le nombre total d'itérations effectuées par le recuit simulé est alors :

$$NbIter = \sum_{k=1}^{NbPaliers} \left\lfloor LongPalier_1 \left(1 + \frac{k-1}{NbPaliers-1} \right) \right\rfloor$$

$$\Rightarrow NbIter \sim \frac{3}{2} NbPaliers \times LongPalier_1$$

5.3 Résolution d'un sous problème

A chaque itération du recuit simulé, on doit résoudre un sous-problème qui correspond à la construction des sections en utilisant uniquement un échantillon d'au plus quelques dizaines de placements. On impose que les placements mono-tailles soient présents dans cet échantillon afin d'assurer l'existence de solutions pour tous les sous-problème. On se retrouve alors dans le cas d'une construction de sections avec un ensemble de placements limité (cf. 2.5 : Résolution directe). On a vu que l'on pouvait résoudre ce problème exactement en utilisant la méthode de Branch & Bound. Cependant, le temps imparti pour la résolution globale ne doit pas excéder quelques minutes au plus. Le nombre total d'itérations du recuit simulé interdit d'utiliser cette méthode exacte mais lourde en temps de calcul. On va donc

résoudre les sous-problèmes avec une méthode heuristique fournissant de bons résultats et très rapide.

5.3.1 Construction d'un problème relaxé

5.3.1.1 Rappel de la modélisation

On appelle P_{ech} l'échantillon de placements disponible pour résoudre le sous problème, le problème à traiter est le suivant (2.2.3 Résumé de la fonction économique) :

$$Z \min = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat(j, s, p) \times X(j, s, p) + \sum_{p \in P_{ech}} C_section(p) \times Y(p) \\ + \sum_{p \in P_{ech}} Creat(p) \times Z(p) + \sum_{p \in P_{ech}} \sum_{j=1}^{nTissus} Evac(j, p) \times E(j, p)$$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\},$$

$$\sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right\} = B_{ji}$$

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\},$$

$$\sum_{p \in P_{ech}} longueur(p) \times X(j, s, p) \leq dispo(j, s)$$

$$\forall p \in P_{ech},$$

$$\left\{ \begin{array}{l} \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) - PliMax \times Y(p) \leq 0 \\ - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) + PliMax \times Y(p) \leq PliMax - 1 \end{array} \right.$$

$$\forall p \in P_{ech},$$

$$\left\{ \begin{array}{l} Maj \times Z(p) - \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) \leq Maj - PliMin_p \\ - Maj \times Z(p) + \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) \leq 0 \end{array} \right.$$

$$\left. \begin{array}{l} \forall p \in P_{ech}, \forall j \in \{1 \dots nTissus\} \\ \\ \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \left\{ \begin{array}{l} \sum_{s=1}^{nLaize(j)} X(j, s, p) - MultPack \times E(j, p) \leq 0 \\ - \sum_{s=1}^{nLaize(j)} X(j, s, p) + MultPack \times E(j, p) \leq MultPack - 1 \\ \\ Z(p) \in \{0, 1\} \\ Y(p) \in \mathbb{N} \\ E(j, p) \in \mathbb{N} \\ X(j, s, p) \in \mathbb{N} \end{array} \right.$$

5.3.1.2 Suppression des variables Y , Z et E

On sait (cf. 1.4.6.2 : Importance relative des coûts) que les coûts liés à l'utilisation de la matière représentent plus de 80% du coût total d'une solution. Si à l'extrême on suppose que tous les autres coûts (matelassage, coupe, création placement...) sont nuls et que les placements peuvent être utilisés pour n'importe quel nombre de plis alors les variables Y , Z et E qui permettent de calculer les coûts autres que les coûts matières n'ont plus lieu d'être ce qui simplifie beaucoup le problème.

Dans le cas général, les coûts en fonction des variables Y , Z et E ne sont pas nuls mais on choisit de ne pas les prendre totalement en compte dans le problème relaxé.

Les variables Z sont des variables binaires qui correspondent à l'utilisation ou non d'un placement dans la solution. Elles permettent de calculer le coût de création de placement et de contrôler le nombre minimum de plis découper avec un placement s'il est utilisé. On élimine ces variables Z et donc les coûts de création de placement et la contrainte du nombre de plis minimum ne sont plus pris en compte dans le problème relaxé.

Les variables E permettent de calculer le nombre de bûches à évacuer pour chaque tissu et chaque placement. On les élimine dans le problème relaxé et on ne tient plus compte des coûts d'évacuation.

Les variables Y correspondent au nombre de sections réalisées pour chaque placement. Elles permettent de calculer les coûts de coupe ainsi que tous les coûts fixes calculés à chaque section. Pour un placement p donné, on a :

$$Y(p) = \left\lfloor \frac{\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p)}{PliMax} \right\rfloor + \delta \quad \text{où : } \begin{cases} \delta = 1 \text{ si le nombre de plis est multiple de } PliMax \\ \delta = 0 \text{ sinon} \end{cases}$$

On choisit de prendre en compte les coûts en fonction des variables

Y : $\sum_{p \in P_{ech}} C_section(p) \times Y(p)$ par l'approximation linéaire :

$$\frac{1}{PliMax} \sum_{p \in P_{ech}} C_section(p) \times \left(\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} X(j, s, p) \right)$$

Cette approximation s'exprime en fonction des variables X et on pose :

$$Mat_Sec(j, s, p) = Mat(j, s, p) + \frac{C_section(p)}{PliMax}$$

Sur le graphique de la **Figure 34** la courbe bleue représente les coûts réels en fonction des variables Y , la courbe rose représente l'approximation calculée en fonction des variables X .

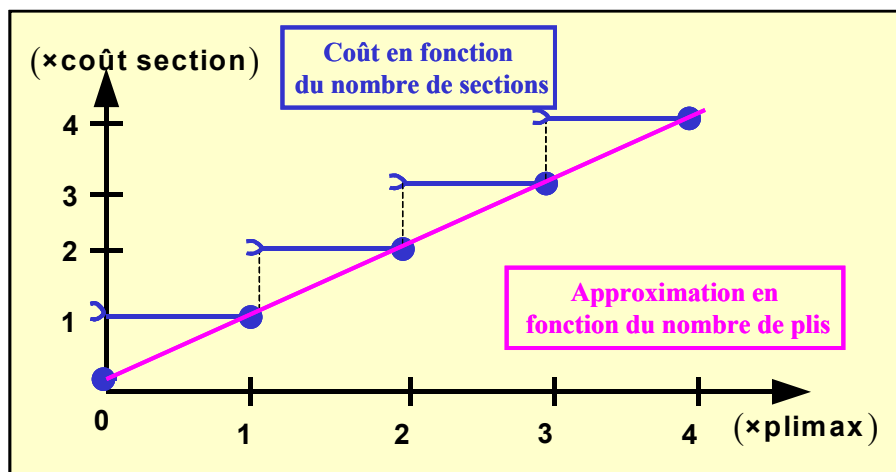


Figure 34 : Approximation des coûts liés aux variables Y

Avec ces modifications, le problème à résoudre est le suivant :

$$Z \min = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat_Sec(j, s, p) \times X(j, s, p)$$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right\} = B_{ji}$$

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \sum_{p \in P_{ech}} longueur(p) \times X(j, s, p) \leq dispo(j, s)$$

$$\left. \begin{array}{l} \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad X(j, s, p) \in \mathbb{N}$$

5.3.1.3 Relaxation des contraintes d'intégrité

Les variables X représentent le nombre de plis découpé avec chaque placement de P_{ech} et pour chaque tissu. Ce sont donc des nombres entiers. Pour que la résolution du sous-problème soit la plus rapide possible, on choisit de relaxer la contrainte d'intégrité des variables X . Le problème à résoudre se réduit alors à :

$$Z \min = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat_Sec(j, s, p) \times X(j, s, p)$$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right\} = B_{ji}$$

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \sum_{p \in P_{ech}} longueur(p) \times X(j, s, p) \leq dispo(j, s)$$

$$\left. \begin{array}{l} \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad X(j, s, p) \geq 0$$

Ce problème est résolu avec la méthode du simplexe. Le résultat obtenu n'est pas une solution réalisable de la construction des sections, le paragraphe suivant décrit la méthode permettant d'obtenir une solution réalisable à partir du résultat du problème relaxé.

5.3.2 Obtention d'une solution réalisable

5.3.2.1 Troncature des variables

Les variables X solutions du problème relaxé sont des variables réelles. Pour obtenir des variables entières on pose :

$$\left. \begin{array}{l} \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\} \tilde{X}(j, s, p) = \lfloor X(j, s, p) \rfloor$$

Le nombre de plis découpés avec le placement p devient : $NbPlis(p) = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \tilde{X}(j, s, p)$

Afin de respecter la contrainte du nombre minimum de plis, on impose :

$$\forall p \in P_{ech}, \quad NbPlis(p) < PliMin_p \Rightarrow \begin{cases} \tilde{X}(j, s, p) = 0 \\ \forall j \in \{1 \dots nTissus\}, \forall s \in \{1 \dots nLaize(j)\} \end{cases}$$

Les variables \tilde{X} permettent de découper une partie de l'ordre de fabrication, on définit une

matrice des restes R par : $\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad R_{ji} = B_{ji} - \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in P_{ech}} \{p_i \times \tilde{X}(j, s, p)\} \right\}$

Le but est maintenant de découper les quantités de R avec les placements de P_{ech} tout en respectant les contraintes de stocks et du nombre minimum de plis par placement.

5.3.2.2 Utilisation des placements par ordre de coût croissant

Dans un premier temps, on calcule pour chaque placement p de P_{ech} , la moyenne sur tous les tissus du coût d'un pli découpé avec p . Ce prix comprend :

- Le coût matière moyen d'un pli
- Le coût de matelassage d'un pli
- Si p n'est pas encore utilisé dans la solution ($NbPlis(p) = 0$) :
 - Le coût de création du placement p
 - Le coût de coupe d'une section de plis de p
 - Le coût des temps fixes de la création d'une nouvelle section

Les placements de P_{ech} sont classés par ordre croissant de ce prix moyen par pli. On va ensuite tenter d'utiliser les placements de P_{ech} dans cet ordre pour découper les quantités de R .

On suppose que P_{ech} contient n_{ech} placements, $\forall 1 \leq k \leq n_{ech}$ on définit les quantités restant à découper avant l'utilisation (ou non) du $k^{ième}$ placement de P_{ech} par R^k . Le nombre de plis que l'on peut découper pour chaque tissu avec p^k suivant les quantités R^k et en respectant les stocks disponibles est donné par :

$$NbPlis_j(p^k) = Min \left(\underset{\substack{1 \leq i \leq nTailles \\ p_i^k > 0}}{Min} \left(\left\lfloor \frac{R_{ji}^k}{p_i^k} \right\rfloor \right); \left\lfloor \frac{\sum_{s=1}^{nLaize(j)} dispo(j,s)}{Longueur(p^k)} \right\rfloor \right) \quad \forall 1 \leq j \leq nTissus$$

Le nombre total de plis que l'on peut découper avec p^k est donc : $\sum_{j=1}^{nTissus} NbPlis_j(p^k)$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\} :$$

- Si $\sum_{j=1}^{n_{Tissus}} NbPlis_j(p^k) \geq Plimin_{p^k}$ on peut utiliser le placement p^k et on a :

$$R_{ji}^{k+1} = R_{ji}^k - NbPlis_j(p^k) \times p_i^k$$
- Sinon on n'utilise pas p^k et : $R_{ji}^{k+1} = R_{ji}^k$

On répète le processus jusqu'à ce que $R \equiv 0$, si les stocks disponibles sont suffisants on est sûr d'y arriver car P_{ech} contient les placements monotailles que l'on peut toujours utiliser (pas de nombre de plis minimum à respecter).

Le résultat obtenu est une solution réalisable du problème de construction des sections. On calcule son coût en déduisant la valeur des variables Y , Z et E des valeurs de \tilde{X} mises à jour à chaque étape de la procédure précédente.

Chapitre 6 : Partitionnement des OF de grande taille

6.1 Introduction

La taille des ordres de fabrication que l'on a à résoudre peut augmenter dans certains cas de façon assez importante. Il y a deux raisons différentes d'augmentation de la taille du problème.

- Augmentation du nombre de tissus
- Augmentation du nombre de tailles

Dans certains cas, on peut trouver des OF contenant une centaine de tissus différents et une centaine de tailles différentes (voire plus dans quelques cas extrêmes). Bien sur, plus la taille du problème est importante plus sa résolution est difficile. Ceci est d'autant plus vrai que la contrainte "durée de résolution" est très forte. En effet, les méthodes de production chez la majeure partie des clients de LECTRA nous impose une durée de résolution allant de quelques secondes à quelques minutes au plus pour chaque problème.

Ces deux types d'augmentation de la taille du problème n'influent pas du tout de la même manière sur sa résolution. En effet, à nombre de tailles égal, le fait d'ajouter un ou plusieurs tissus à un bordereau de change rien au nombre de placements réalisables (on suppose qu'il n'y a pas de contrainte sur le nombre minimum de plis). On a vu en revanche que le nombre de placements réalisables augmente de façon quasi¹ exponentielle en fonction du nombre de colonnes de la matrice des quantités (cf. 4.3.1 : Problème de l'augmentation du nombre de colonnes de l'OF).

¹ Le nombre de placements réalisables s'exprime en fonction du nombre de colonnes de l'ordre par un polynôme de degré *NbTaillesMax*. Le degré de ce polynôme peut donc être très élevé si le nombre de tailles autorisé dans un placement est grand.

On a vu au paragraphe 4.3 que l'on pouvait contourner ce problème en ne sélectionnant qu'une partie des placements réalisables dans la bibliothèque de placements testée par le recuit simulé. Cependant, sans doute à cause de la faible proportion de placements sélectionnés par rapport au nombre total de placements réalisables, les résultats de cette technique sont souvent médiocres pour les OF les plus imposants. L'objet de ce chapitre est d'envisager une adaptation de la méthode de résolution afin de pouvoir traiter les plus gros ordres de fabrication avec des solutions de qualité tout en restant dans des temps de résolution "acceptables".

6.2 Principe

L'idée est très simple, elle consiste à "casser en plusieurs morceaux" un ordre de fabrication qu'il serait difficile de résoudre directement. Ce type de stratégie a déjà été envisagé par le passé. Il s'agissait alors de résoudre l'OF "tissu par tissu" ce qui revenait en fait à résoudre plusieurs OF ne comportant chacun qu'un seul tissu. Le but ici est différent et consiste à casser la matrice des quantités à produire non pas tissu à tissu (ligne à ligne) mais plutôt taille à taille (colonne à colonne). Plus précisément, on va considérer que l'ensemble de toutes les tailles de l'OF est constitué de plusieurs groupes de tailles, chaque groupe contenant plusieurs tailles. L'ordre de fabrication initial se décompose alors en plusieurs "sous-ordres" que l'on peut résoudre indépendamment les uns des autres avec pour chacun d'eux une sélection des placements réalisables. On choisit de "casser" l'ensemble des tailles plutôt que l'ensemble des tissus car c'est l'augmentation du nombre de tailles qui fait exploser le nombre de placements réalisables.

Un des avantages de ce principe est d'éliminer totalement la résolution des gros ordres de fabrication car on ne résout plus que des sous-ordres de taille "raisonnable". De plus, chacun d'entre eux sera résolu de manière plus efficace car les solutions réalisables seront beaucoup moins nombreuses¹.

¹ Lorsque l'ensemble des solutions réalisables croît de façon exponentielle en fonction de la taille n d'un problème, il vaut mieux résoudre n problèmes plutôt qu'un seul n fois plus gros.

En revanche, l'utilisation cette méthode entraîne de nouveau une perte de généralité. En effet, il se peut que dans la solution optimale théorique, un placement contenant des tailles appartenant à des groupes de tailles différents soit utilisé. Dans ce cas, on ne pourra jamais trouver la solution optimale théorique en faisant la résolution groupe à groupe. Cependant, comme pour le cas de la sélection non exhaustive des placements, on verra qu'en pratique cette méthode fournit de bien meilleurs résultats que lorsque l'OF est résolu en une seule fois.

6.3 Comment regrouper les tailles ? Méthodes de classification

On a vu au paragraphe précédent que séparer les tailles de l'ordre de fabrication initial en plusieurs groupes de tailles entraînait une perte de généralité. Il faut donc essayer, de faire les "meilleurs" groupes possibles afin de limiter au maximum les effets de cette perte de généralité. Pour cela, on crée un critère de ressemblance entre les tailles de l'OF (plus exactement un indice de dissimilarité entre les vecteurs colonnes de la matrice des quantités) qui permettra de décider si telle taille doit être regroupée avec telle autre.

Pour prendre ces décisions, on fait appel à des méthodes statistiques et plus précisément, aux méthodes de partitionnement.

6.3.1 Mise au point d'un indice de dissimilarité entre les tailles

Les quantités à produire sont représentées sous la forme d'une matrice à n_{Tissus} lignes et à $n_{Tailles}$. Chaque colonne de cette matrice est représentée par un vecteur colonne à n_{Tissus} coordonnées et correspond aux quantités à produire dans une taille de l'OF.

Une distance d sur \mathbb{R}^n est une application qui vérifie les propriétés suivantes:

$$\begin{aligned}
 (P1) \quad d(i, j) &\geq 0 & \forall i, j \in \mathbb{R}^n \\
 (P2) \quad d(i, i) &= 0 & \forall i \in \mathbb{R}^n \\
 (P3) \quad d(i, j) &= d(j, i) & \forall i, j \in \mathbb{R}^n \\
 (P4) \quad d(i, j) &\leq d(i, h) + d(h, j) & \forall i, j, h \in \mathbb{R}^n
 \end{aligned}$$

Un indice de dissimilarité vérifie (P1), (P2) et (P3) mais pas forcément (P4).

Les premiers indices de dissimilarité qui viennent à l'esprit sont alors tout simplement les distances sur \mathbb{R}^n :

$$\forall u, v \in \mathbb{R}^n, \quad d_q(u, v) = \left(\sum_{i=1}^n |u_i - v_i|^q \right)^{1/q} \quad \forall q \geq 1$$

La plus simple à utiliser dans notre cas semble être pour $q = 1$:

$$\forall u, v \in \mathbb{R}^n, \quad d_1(u, v) = \sum_{i=1}^n |u_i - v_i|$$

Cependant, dans notre cas, cette distance n'est pas adaptée. Pour s'en apercevoir, on considère l'ordre de fabrication de la **Figure 35** :

	A	B	C	D
Rouge	12	10	20	24
Bleu	7	6	12	14

Figure 35 : Exemple où la distance entre tailles n'est pas adaptée

La solution optimale pour résoudre ce bordereau est :

- Utiliser le placement (1,0,0,2) avec (12 et 7) plis
- Utiliser le placement (0,1,2,0) avec (10 et 6) plis

Imaginons que l'on désire séparer cet OF en deux, avec la distance d_1 on regroupe les tailles A et B puis les tailles C et D. On ne peut pas alors retrouver la solution optimale en résolvant les deux sous-ordres séparément. En effet, la distance d_1 ne prend en compte que l'écart absolu entre les différentes quantités à produire alors que l'on doit aussi tenir compte du fait que, sur un placement on peut répéter plusieurs fois une même taille. Ainsi, il peut être très intéressant de regrouper des tailles présentant des quantités à produire très différentes si l'on veut limiter au maximum les mauvais effets de la décomposition de l'ordre de fabrication initial.

Dans l'exemple de la **Figure 35**, il faudrait regrouper les tailles A et D et les tailles B et C. Le but est maintenant de définir un indice de dissimilarité mieux adapté que les distances classiques. Pour cela on commence par définir les quantités suivantes:

$$s_{h,k}(u,v) = \text{Max} \left(\sum_{i=1}^{nTissus} \left[\text{Min} \left(\frac{u_i}{h}, \frac{v_i}{k} \right) \right]; \sum_{i=1}^{nTissus} \left[\text{Min} \left(\frac{v_i}{h}, \frac{u_i}{k} \right) \right] \right) \quad \left| \begin{array}{l} \forall h, k \in \mathbb{N}^* \\ \forall u, v \in \mathbb{R}^{nTissus} \end{array} \right.$$

Imaginons un placement qui ne contienne que les tailles u et v . Si on place la taille u h fois et la taille v k fois, pour chaque tissu $i \leq nTissus$ le nombre de plis que l'on peut faire au maximum avec ce placement est : $\left[\text{Min} \left(\frac{u_i}{h}, \frac{v_i}{k} \right) \right]$.

Alors le nombre total de plis que l'on peut faire avec ce placement est : $\sum_{i=1}^{nTissus} \left[\text{Min} \left(\frac{u_i}{h}, \frac{v_i}{k} \right) \right]$

Inversement, si on place la taille u k fois et la taille v h fois, le nombre total de plis devient:

$$\sum_{i=1}^{nTissus} \left[\text{Min} \left(\frac{v_i}{h}, \frac{u_i}{k} \right) \right].$$

Ainsi, $s_{h,k}(u,v)$ correspond au nombre de plis maximum que l'on peut faire avec un placement ne contenant que les tailles u et v sachant que une de ces tailles est placée k fois et l'autre h fois.

On définit maintenant des quantités qui se rapprochent d'un indice de dissimilarité:

$$d_{h,k}(u,v) = 1 - \frac{(h+k) \times s_{h,k}(u,v)}{\sum_{i=1}^{nTissus} u_i + v_i} \quad \left| \begin{array}{l} \forall h, k \in \mathbb{N}^* \\ \forall u, v \in \mathbb{R}^{nTissus} \end{array} \right.$$

$(h+k) \times s_{h,k}(u,v)$ représente le nombre total de vêtements que l'on découpe avec le placement précédent qui contient $(h+k)$ tailles.

$\sum_{i=1}^{nTissus} (u_i + v_i)$ représente le nombre total de vêtements à produire dans les tailles u et v .

Et donc, $d_{h,k}(u,v)$ est la proportion de vêtements qui resterait à produire dans les tailles u et v si l'on utilisait le placement précédent au maximum.

Finalement, on pose:

$$\forall \lambda \in \mathbb{N}^*$$

$$D_\lambda(u,v) = \underset{\substack{k \leq \lambda \\ h \leq k}}{\text{Min}}(d_{h,k}(u,v)) \quad \forall u, v \in \mathbb{R}^{nTissus}$$

Dans la pratique, λ sera égal au nombre maximum de fois où l'on peut répéter une même taille sur un placement $NbMaxRep$. Et dans toute la suite on assimilera $D_{NbMaxRep}$ à D . Avec ces notations, $D(u,v)$ correspond à la proportion minimale de vêtements qui resterait à produire dans les tailles u et v si l'on utilise un placement ne contenant que ces deux tailles. On vérifie facilement que (P1), (P2) et (P3) sont vérifiées, D est donc un indice de dissimilarité entre les différentes tailles d'un OF. Il dépend également des contraintes de création des nouveaux placements. Par construction, D est bien adaptée au problème. Deux tailles sont proches au sens de D si l'on peut découper une grande partie des vêtements demandés dans ces deux tailles avec un même placement. On a alors intérêt à placer ces tailles dans un même groupe.

6.3.2 Méthodes de classification

Le but des méthodes de classification est de trouver parmi un ensemble d'individus¹ des groupes qui présentent des analogies ou des caractéristiques communes. Il existe un grand nombre de méthodes de classification qui se regroupent essentiellement en deux grandes catégories:

- Les **méthodes de partitionnement**, fournissant une partition en k classes de l'ensemble des individus (k étant fixé au préalable)

¹ Le terme individu est à prendre ici au sens statistique, il peut désigner des personnes ou tout autre objet dont on connaît certaines caractéristiques

- Les méthodes de *classification hiérarchique*, fournissant un arbre hiérarchique, c'est à dire une succession de partitions de l'ensemble des individus emboîtées les unes dans les autres.

Dans notre cas, nous avons besoin d'une partition et d'une seule de l'ensemble des tailles du bordereau à traiter. La première idée est donc d'utiliser une méthode de partitionnement. Le paragraphe suivant décrit une de ces méthodes. On verra qu'elle n'est pas satisfaisante pour la partition des ordres de fabrication. Le paragraphe 6.3.2.2 décrit ensuite plusieurs méthodes de classification hiérarchique. C'est l'une d'entre elles qui sera modifiée pour s'adapter au cas de la construction des sections.

6.3.2.1 Création d'une partition de l'ensemble des tailles d'un ordre de fabrication

La méthode de partitionnement mise en application ici est appelée P.A.M (Partitionning Around Medoids) décrite dans [Kaufman1990]. Cette méthode est une généralisation de la méthode des centres mobiles qui est seulement valable lorsque l'on dispose d'une distance entre les individus. En effet, la méthode des centres mobiles fait appel à la notion de centre de gravité qui nécessite une distance. Une fois fixé le nombre de classes k que l'on souhaite, la méthode P.A.M se décompose en deux phases:

- La première appelée BUILD consiste à trouver k tailles représentatives parmi l'ensemble des tailles du bordereau. La donnée de ces k tailles détermine une partition de l'ensemble des tailles car alors, chaque taille qui ne fait pas partie des k tailles représentatives est associée à la taille représentative la plus proche (au sens de l'indice de dissimilarité).
- La seconde appelée SWAP a pour but d'améliorer la première sélection de tailles représentatives en testant, pour toutes les tailles représentatives, s'il est avantageux de la remplacer par une taille non représentative.

L'utilisation de cette méthode pour créer une partition de l'ensemble des tailles présente cependant un inconvénient, la taille (c'est à dire le nombre d'individus) dans chaque classe de la partition finale peut être très variable et il arrive souvent que l'on ait de grosses différences d'une classe à l'autre. Or, ce qui est intéressant pour la résolution d'un ordre de fabrication,

c'est plutôt d'avoir des classes assez équilibrées en nombre de tailles afin de créer des sous-ordres pas trop "gros" (car c'est le but de l'opération) mais pas trop petits non plus (si l'on a trop peu de tailles dans un sous-ordre, on ne pourra utiliser que de petits placements peu avantageux pour le résoudre). Cependant, avec cette méthode, on n'a aucun contrôle sur la partition finale, voilà pourquoi on essaie plutôt de mettre en place un algorithme de classification hiérarchique.

6.3.2.2 Création d'un arbre hiérarchique

Il existe principalement deux sortes de classification hiérarchique:

- Les méthodes dites "**ascendantes**" où l'on part de la partition composée des singletons en agrégeant peu à peu les individus.
- Les méthodes dites "**descendantes**" où l'on part de la partition composée d'une seule classe contenant tous les individus pour séparer peu à peu les individus.

Les méthodes ascendantes sont en général plus utilisées car les algorithmes mis en œuvre sont moins complexes.

La méthode de classification hiérarchique mise en application ici est la méthode ascendante appelée AGNES décrite dans [Kaufman1990].

Tout d'abord, on définit une notion de dissimilarité entre deux classes à partir de l'indice de dissimilarité défini au paragraphe 6.3.1 :

Soient A et B deux classes d'une partition P de l'ensemble des tailles d'un ordre:

On pose:
$$d(A, B) = \frac{1}{|A||B|} \sum_{\substack{i \in A \\ j \in B}} D(i, j) \quad (|A| \text{ désigne le nombre d'éléments de } A)$$

$d(A, B)$ représente en quelque sorte la moyenne des dissimilarités entre les individus des classes A et B .

Le résultat de la méthode AGNES est une suite de n Tailles partitions imbriquées les unes dans les autres :

- A l'étape 0, la partition P^0 est la partition en $nTailles$ classes de l'ensemble des tailles. C'est la partition composée des singletons.
- A l'étape k ($1 \leq k \leq nTailles - 1$), la partition P^k contient $(nTailles - k)$ classes. La partition P^{k+1} de l'étape $k+1$ est engendrée par la fusion des 2 classes de P^k les plus proches au sens de l'indice de dissimilarité entre classes.

Du fait de la définition de la dissimilarité entre deux classes d'une partition et de la manière dont on obtient les classes successives, on n'a pas besoin à chaque étape de recalculer toutes les dissimilarités entre classes. On peut en effet se servir de la relation suivante:

$$\begin{aligned}
 d(A \cup B, C) &= \frac{1}{|A \cup B| |C|} \sum_{\substack{i \in A \cup B \\ j \in C}} D(i, j) \\
 &= \frac{1}{(|A| + |B|) \times |C|} \left(\sum_{\substack{i \in A \\ j \in C}} D(i, j) + \sum_{\substack{i \in B \\ j \in C}} D(i, j) \right) \\
 &= \frac{1}{(|A| + |B|) \times |C|} (|A| |C| d(A, C) + |B| |C| d(B, C)) \\
 &= \frac{1}{|A| + |B|} (|A| d(A, C) + |B| d(B, C))
 \end{aligned}$$

Malheureusement, cette méthode appliquée ainsi présente souvent le même désavantage que la précédente et on se retrouve souvent avec des classes très déséquilibrées. Cependant ici, on "voit" peu à peu les partitions se construire et les classes "grossir", on peut donc espérer avoir un meilleur contrôle sur le nombre d'éléments à l'intérieur de chaque classe. En effet, on veut éviter que dans une partition donnée, les écarts entre les nombres d'éléments de chaque classe soient trop importants. Une stratégie possible pour réaliser cela serait d'empêcher que les classes contenant déjà beaucoup d'éléments se regroupent dans la partition suivante. L'idée est alors de modifier la formule de dissimilarité entre les classes.

On pose les relations suivantes:

$$d(C_i^0, C_j^0) = D(i, j) \quad \forall 1 \leq i, j \leq nTailles$$

$$d(C_i^k \cup C_j^k, C_l^k) = \frac{1}{\sqrt{|C_i^k| + |C_j^k|}} (|C_i^k| d(C_i^k, C_l^k) + |C_j^k| d(C_j^k, C_l^k)) \quad \begin{array}{l} \forall 1 \leq k \leq nTailles - 1 \\ \forall 1 \leq i, j \leq nTailles - k \end{array}$$

La présence de la racine carrée au dénominateur permet de favoriser le regroupement de classes comprenant peu d'éléments. En effet prenons par exemple trois classes A , B et C telles que $|A| = |B|$ et $d(A, C) = d(B, C)$:

Avec la formule initiale on obtient:

$$\begin{aligned} d(A \cup B, C) &= \frac{1}{|A| + |B|} (|A| d(A, C) + |B| d(B, C)) \\ &= \frac{2|A|}{2|A|} d(A, C) \\ &= d(A, C) \quad (= d(B, C)) \end{aligned}$$

Avec la formule modifiée on obtient:

$$\begin{aligned} d(A \cup B, C) &= \frac{1}{\sqrt{|A| + |B|}} (|A| d(A, C) + |B| d(B, C)) \\ &= \frac{2|A|}{\sqrt{2|A|}} d(A, C) \\ &= \sqrt{2|A|} \times d(A, C) \quad (= \sqrt{2|B|} \times d(B, C)) \end{aligned}$$

Avec cette formule, plus les classes comportent d'individus, plus les dissimilarités entre classes sont importantes. Ainsi, les regroupements des classes comprenant le moins d'éléments seront facilités. L'inconvénient de cette méthode est que l'on n'a pas de formule directe pour calculer les dissimilarités entre classes mais ce n'est pas forcément un handicap trop lourd si l'on tient à jour le tableau des dissimilarités entre classes au fur et à mesure des regroupements.

6.3.2.3 Comparaison des différentes méthodes sur un exemple

Le tableau de la **Figure 36** représente les quantités demandées pour un OF contenant 14 tailles différentes :

Tailles Tissus	34	36	38	40	42	44	46	48	50	52	54	56	58	60	Total
Bleu	10	13	21	19	17	32	43	29	12	7	21	18	24	9	275
Jaune	9	23	17	5	13	21	34	36	15	9	18	23	17	11	251
Marine	12	19	34	16	9	17	27	22	6	12	13	36	22	15	260
Marron	13	14	19	18	12	31	33	17	5	10	15	21	27	18	253
Mauve	15	12	27	3	14	43	37	21	8	11	20	25	17	8	261
Total	59	81	118	61	65	144	174	125	46	49	87	123	107	61	1300

Figure 36 : Exemple d'OF contenant 14 tailles différentes

On décide de partitionner l'ensemble des tailles en 4 classes, on compare les partitions obtenues avec différentes méthodes de classification :

- Méthode P.A.M.

$$C_1 = \{34, 42, 52, 58\} \quad C_2 = \{36, 38, 56, 60\}$$

$$C_3 = \{40\} \quad C_4 = \{44, 46, 48, 50, 54\}$$

- Méthode AGNES

$$C_1 = \{34, 38, 52, 56, 58, 60\} \quad C_2 = \{36, 42, 44, 46, 48, 54\}$$

$$C_3 = \{40\} \quad C_4 = \{50\}$$

- Méthode du lien minimum

Consiste à poser : $d(A \cup B, C) = \text{Min}(d(A, C); d(B, C))$

$$C_1 = \{34, 36, 38, 44, 46, 48, 52, 54, 56, 58, 60\}$$

$$C_2 = \{40\} \quad C_3 = \{42\} \quad C_4 = \{50\}$$

- Méthode du lien maximum

Consiste à poser : $d(A \cup B, C) = \text{Max}(d(A, C); d(B, C))$

$$\begin{array}{ll} C_1 = \{34, 38, 52, 56, 58, 60\} & C_2 = \{36, 48, 50\} \\ C_3 = \{40\} & C_4 = \{42, 44, 46, 54\} \end{array}$$

- Méthode « AGNES modifiée »

$$\begin{array}{ll} C_1 = \{34, 38, 52, 56\} & C_2 = \{36, 48, 50\} \\ C_3 = \{40, 58, 60\} & C_4 = \{42, 44, 46, 54\} \end{array}$$

Seule la méthode « AGNES modifiée » a permis d'obtenir 4 classes équilibrées en terme de nombre d'individus. La **Figure 37** représente le dendrogramme obtenu avec cette méthode. La hiérarchie est évaluée par l'indice de dissimilarité définie entre les classes.

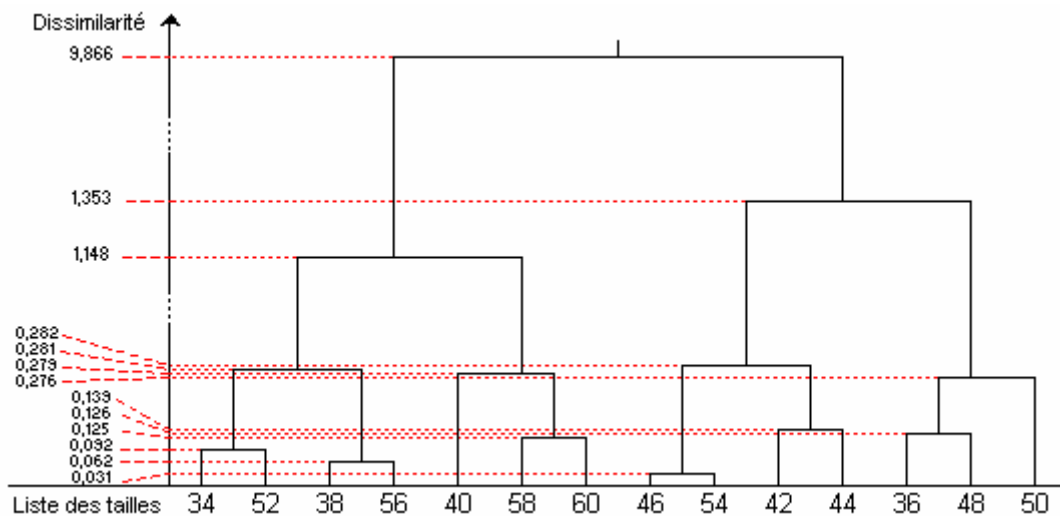


Figure 37 : Arbre hiérarchique de classification de l'ensemble des tailles

On observe des sauts importants de la valuation entre les parties contenant des nombres d'individus différents. C'est cela qui permet d'obtenir une partition en 4 classes équilibrées.

6.3.3 Contrôle du nombre de classes dans la partition retenue

L'application de la méthode AGNES modifiée fournit donc une hiérarchie de l'ensemble des tailles c'est à dire une suite de $nTailles$ partitions imbriquées. Cependant, pour la résolution

des sous-ordres, on ne doit travailler qu'avec une seule partition. Il faut donc, déterminer, à l'intérieur de la hiérarchie obtenue, la partition avec laquelle on va effectuer la résolution.

On choisit pour cela des critères relativement empiriques qui dépendent des données du problème à résoudre.

6.3.3.1 Ne pas « casser » des OF contenant moins de 10 tailles

Lorsque le nombre de tailles différentes présentes dans l'ordre de fabrication est inférieur ou égal à 10 on choisit de le résoudre directement. En effet, c'est le nombre de tailles en dessous duquel la résolution directe est très efficace car la génération exhaustive des placements réalisables est souvent possible et utile.

6.3.3.2 Utiliser les meilleurs placements autorisés

On sait que, en règle générale, l'efficacité des placements augmente lorsque le nombre de tailles placées augmente. Si *NbMaxTailles* est le nombre maximum de tailles autorisées dans un placement, les placements contenant *NbMaxTailles* tailles sont la plupart du temps les plus efficaces et il est donc avantageux de les utiliser dans la résolution. Ainsi, lorsqu'on décide de "casser" un OF en plusieurs sous-ordres, il faut veiller à ce que ceux-ci contiennent suffisamment de tailles pour pouvoir utiliser les placements les plus longs autorisés. Pour cela, on impose que le nombre de tailles minimum dans un sous-bordereau soit égal à *NbMaxTailles* sauf dans le cas où *NbMaxTailles* > 20 .

6.3.3.3 Résoudre si possible de petits sous-ordres

A partir du moment où le nombre de tailles présentes dans un sous-ordre est suffisant ($> NbMaxTailles$), l'expérience montre qu'il vaut mieux la plupart du temps résoudre plusieurs petits OF qu'un seul plus important. On cherchera donc à résoudre les plus petits OF possibles permettant de respecter la contrainte décrite au 6.3.3.2

La sélection de la partition retenue dans la hiérarchie se fait de la façon suivante:

- Si le nombre de tailles n'est pas supérieur à 10, on résout directement.

- Sinon, on part du "bas" de l'arbre, de la partition composée des singletons et on retient la première partition qui satisfait la contrainte décrite au 6.3.3.2.

6.4 Conclusion sur la résolution des OF de grande taille

On dispose désormais de deux méthodes permettant de contourner les problèmes liés à l'augmentation de la taille des OF à traiter:

- Les méthodes permettant une sélection non exhaustive des placements réalisables.
- Une procédure de classification des tailles permettant de se ramener à une résolution de petits OF.

Ces deux méthodes ne sont pas parfaites et sont à l'origine d'une perte de généralité par rapport à la formulation théorique du problème. En effet, lorsque l'on résout un OF en une seule fois avec une sélection exhaustive des placements, on n'est pas assuré de trouver la solution optimale à la fin des itérations du recuit simulé, mais on est sûr que cette solution optimale est "atteignable" car on a accès à l'intégralité de l'espace des solutions réalisables. A l'inverse, lorsque l'on effectue une sélection non exhaustive des placements ou que l'on résout des sous-ordres, on abandonne une partie de l'espace des solutions réalisables, on n'a donc pas l'assurance que la solution optimale fasse partie des solutions réalisables que l'on peut atteindre. Cependant, ces méthodes sont très utiles en pratique car elles permettent de réduire considérablement l'espace des solutions, elles permettent de trouver des solutions de meilleure qualité dans des temps plus courts.

La démarche générale de résolution d'un bordereau donné devient donc la suivante:

- Création d'une partition de l'ensemble des tailles afin de résoudre une suite de sous-bordereaux. (le nombre de sous-bordereaux varie selon le nombre de tailles du bordereau initial et le nombre de tailles maximum autorisé sur un même placement)
- Pour chacun des sous-bordereaux créés, (il peut n'y en avoir qu'un):
 - Effectuer la génération des placements:

- Génération exhaustive si le nombre de tailles du sous-bordereau n'est pas trop important.
- Génération non exhaustive sinon.
- Résolution du sous-bordereau en utilisant les placements retenus.
- Regrouper les différentes solutions des sous-bordereaux en une solution du bordereau initial.

Chapitre 7 : Résolution avec déviations sur les quantités

7.1 Problématique

Il arrive parfois que des déviations sur les quantités d'articles découpés soient autorisées. En effet, dans certains cas, les utilisateurs acceptent de découper un peu plus ou un peu moins que les quantités exactes de l'ordre de fabrication afin de réduire si possible le coût moyen par article découpé. Cette méthode est appelée OVERCUT si l'on découpe des articles en plus et UNDERCUT si l'on découpe des articles en moins.

Lorsque l'on utilise cette méthode, le but n'est plus de minimiser le coût de tous les articles produits mais le coût *moyen* des articles produits. Or, ce nouvel objectif n'est plus un objectif linéaire par rapport aux variables du problème. Cependant, le seul endroit où intervient la linéarité du problème dans la résolution c'est pendant le traitement du problème relaxé (cf. 5.3.1) par la méthode du simplexe. En effet, le problème relaxé à résoudre à chaque itération lorsque que les déviations ne sont pas autorisées est le suivant :

$$Z \min = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat_Sec(j, s, p) \times X(j, s, p)$$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad \sum_{s=1}^{nLaize(j)} \left\{ \sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right\} = B_{ji}$$

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \sum_{p \in P_{ech}} longueur(p) \times X(j, s, p) \leq dispo(j, s)$$

$$\left. \begin{array}{l} \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad X(j, s, p) \geq 0$$

A l'inverse, lorsque l'on autorise les déviations, le problème devient le suivant :

$$Z \min = \frac{\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat_Sec(j, s, p) \times X(j, s, p)}{\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} NbTailles(p) \times X(j, s, p)}$$

$$\left. \begin{array}{l} \forall i \in \{1 \dots nTailles\} \\ \forall j \in \{1 \dots nTissus\} \end{array} \right\}, \quad \begin{cases} \sum_{s=1}^{nLaize(j)} \left(\sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right) \geq B_{ji}^- \\ \sum_{s=1}^{nLaize(j)} \left(\sum_{p \in P_{ech}} \{p_i \times X(j, s, p)\} \right) \leq B_{ji}^+ \end{cases}$$

$$\left. \begin{array}{l} \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad \sum_{p \in P_{ech}} longueur(p) \times X(j, s, p) \leq dispo(j, s)$$

$$\left. \begin{array}{l} \forall p \in P_{ech} \\ \forall j \in \{1 \dots nTissus\} \\ \forall s \in \{1 \dots nLaize(j)\} \end{array} \right\}, \quad X(j, s, p) \geq 0$$

où :

$\forall p \in P_{ech}$, $NbTailles(p)$ est le nombre de tailles contenues dans le placement p

$\forall 1 \leq i \leq nTailles, \forall 1 \leq j \leq nTissus$, $\begin{cases} B_{ji}^+ \text{ est le nombre maximum d'articles} \\ B_{ji}^- \text{ est le nombre minimum d'articles} \end{cases}$
à produire dans la taille i et le tissu j .

Ce problème n'est pas linéaire par rapport aux variables X , on ne peut donc pas le résoudre directement avec la méthode du simplexe. Le paragraphe suivant, propose une méthode pour linéariser ce problème avant de le résoudre.

7.2 Méthode d'optimisation d'un quotient de deux fonctions linéaires

On se donne le problème suivant:

$$P 1 \left\{ \begin{array}{l} \min f(\mathbf{x}) = \frac{\mathbf{c}\mathbf{x} + c_0}{\mathbf{d}\mathbf{x} + d_0} \\ \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+^n \end{array} \right.$$

où: $\left\{ \begin{array}{l} \mathbf{c} \text{ et } \mathbf{d} \text{ sont des vecteurs lignes, } \mathbf{x} \text{ et } \mathbf{b} \text{ des vecteurs colonnes } A \text{ une matrice} \\ c_0 \text{ et } d_0 \text{ sont des scalaires.} \end{array} \right.$

De plus, on impose: $\mathbf{d}\mathbf{x} + d_0 > 0 \quad \forall \mathbf{x} \geq 0$

Considérons maintenant le problème d'optimisation linéaire classique suivant:

$$P 2 \left\{ \begin{array}{l} \min g(\mathbf{y}, t) = \mathbf{c}\mathbf{y} + c_0 t \\ \mathbf{A}\mathbf{y} - \mathbf{b}t \leq 0 \\ \mathbf{d}\mathbf{y} + d_0 t = 1 \\ \mathbf{y} \in \mathbb{R}_+^n, t > 0 \end{array} \right.$$

où: $\left\{ \begin{array}{l} \mathbf{y} \text{ est un vecteur colonne} \\ t \text{ est un scalaire} \end{array} \right.$

On montre que $P1 \Leftrightarrow P2$. Cette proposition est citée brièvement dans [Hillier1995]. Pour plus de détails sur l'optimisation de fonctions fractionnelles, le lecteur pourra se reporter à [Schaible1981].

7.3 Application à la construction de sections

Le problème de minimisation du coût par article découpé est un problème de type P1. On construit donc un problème de type P2 équivalent au premier :

$$Z \min = \sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} Mat_Sec(j, s, p) \times Y(j, s, p)$$

$$\left. \begin{array}{l}
\forall i \in \{1 \dots nTailles\} \\
\forall j \in \{1 \dots nTissus\} \\
\forall j \in \{1 \dots nTissus\} \\
\forall s \in \{1 \dots nLaize(j)\} \\
t > 0 \quad \text{et} \quad \left. \begin{array}{l}
\forall p \in P_{ech} \\
\forall j \in \{1 \dots nTissus\} \\
\forall s \in \{1 \dots nLaize(j)\}
\end{array} \right\}, \quad Y(j, s, p) \geq 0
\end{array} \right\}
\begin{array}{l}
\left. \begin{array}{l}
\sum_{s=1}^{nLaize(j)} \left(\sum_{p \in P_{ech}} \{p_i \times Y(j, s, p)\} \right) + t \times B_{ji}^- \leq 0 \\
\sum_{s=1}^{nLaize(j)} \left(\sum_{p \in P_{ech}} \{p_i \times Y(j, s, p)\} \right) - t \times B_{ji}^- \leq 0 \\
\sum_{p \in P_{ech}} longueur(p) \times Y(j, s, p) \leq dispo(j, s) \\
\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} NbTailles(p) \times Y(j, s, p) = 1
\end{array} \right\}
\end{array}$$

Après la résolution de ce programme linéaire par la méthode classique du simplexe, on récupère la valeur des variables X par la relation:

$$X(j, s, p) = \frac{Y(j, s, p)}{t} \quad \forall 1 \leq j \leq nTissus, \forall 1 \leq s \leq nLaizes(j), \forall p \in P_{ech}$$

De plus, en pratique le cas $t = 0$ est impossible car la contrainte :

$$\left. \begin{array}{l}
\forall i \in \{1 \dots nTailles\} \\
\forall j \in \{1 \dots nTissus\}
\end{array} \right\}, \quad \sum_{s=1}^{nLaize(j)} \left(\sum_{p \in P_{ech}} \{p_i \times Y(j, s, p)\} \right) - t \times B_{ji}^- \leq 0$$

impose: $t = 0 \Rightarrow Y(j, s, p) \equiv 0$ mais la contrainte $\sum_{j=1}^{nTissus} \sum_{s=1}^{nLaize(j)} \sum_{p \in P_{ech}} NbTailles(p) \times Y(j, s, p) = 1$

ne permet pas que tous les $Y(j, s, p)$ soient nuls.

7.4 Obtention d'une solution réalisable

On a vu comment résoudre le problème relaxé correspondant à la minimisation du coût moyen par article découpé. Il reste à adapter la méthode d'obtention d'une solution réalisable (cf. 5.3.2 : Obtention d'une solution réalisable) pour qu'elle convienne au cas d'une résolution où des déviations sur les quantités à produire sont autorisées.

Dans le cas d'une résolution sans déviation, la méthode se décompose ainsi:

- Récupération de la partie entière de la solution du problème relaxé
- Annuler l'utilisation des placements dont le nombre de plis est insuffisant
- Ordonner les placements de l'échantillon par ordre croissant de coût
- Création de la matrice des restes
- Pour tous les placements (du moins cher au plus cher)
 - Tant que le bordereau des restes n'est pas vide
 - Peut-on l'utiliser en respectant les contraintes?
 - Si oui
 - ✓ Utiliser le placement le plus possible
 - ✓ Actualiser la matrice des restes
 - ✓ Actualiser la solution
- Renvoyer la solution

L'obtention d'une solution réalisable est garantie car les placements mono-taille sont présents dans l'échantillon et il n'y a pas de contrainte sur leur utilisation.

Il faut maintenant généraliser cette fonction afin qu'elle puisse également s'appliquer à une résolution avec déviations. Pour cela, au lieu d'avoir une matrice des restes, on distingue deux matrices des quantités résiduelles:

- Une matrice des quantités minimum, qui donne pour chaque tissu et chaque taille, la quantité minimum encore à produire pour satisfaire les quantités basses de l'ordre de fabrication (UNDERCUT)
- Une matrice des quantités maximum, qui donne pour chaque tissu et chaque taille, la quantité maximum que l'on peut encore produire tout en satisfaisant les quantités hautes de l'ordre de fabrication (OVERCUT)

La démarche se résume alors de la façon suivante :

- Récupération de la partie entière de la solution du problème relaxé
- Annuler l'utilisation des placements dont le nombre de plis est insuffisant
- Ordonner les placements de l'échantillon par ordre croissant de coût
- Création de la matrice des quantités minimum
- Création de la matrice des quantités maximum
- Pour tous les placements (du moins cher au plus cher)
 - Tant que la matrice des quantités minimum n'est pas nulle
 - Peut-on l'utiliser en respectant les contraintes?
 - Si oui
 - ✓ Utiliser le placement de façon à satisfaire les quantités minimales sans aller au-delà des quantités maximales
 - ✓ Actualiser la matrice des quantités minimum
 - ✓ Actualiser la matrice des quantités maximum
 - ✓ Actualiser la solution
- Renvoyer la solution

Chapitre 8 : Exemples & Comparaisons

Dans ce chapitre, quatre exemples de résolution sont présentés. Chacun d'eux correspond à un type de problèmes rencontrés fréquemment en pratique.

- L'ordre de fabrication du premier exemple est de taille réduite, on peut donc effectuer une résolution simple (détaillée dans le Chapitre 5 :) avec une sélection exhaustive des placements.
- Dans le deuxième exemple, on traite deux OF pour lesquels la sélection exhaustive des placements n'est pas possible car le nombre de tailles est trop élevé. Cependant, on n'effectue pas le partitionnement en plusieurs sous-ordres, on essaie alors d'évaluer l'influence du nombre de placements sélectionné pour la résolution sur le résultat final.
- Le troisième exemple correspond à un cas très général pour lequel on réalise le partitionnement en plusieurs sous-ordres. Chacun d'eux est ensuite traité séparément avec une sélection exhaustive ou non des placements selon les cas.
- Dans le dernier exemple, on évalue l'influence d'un pourcentage de déviations autorisées sur le coût moyen par vêtement des solutions.

8.1 Exemple 1 : Résolution de base

Le tableau de la **Figure 38** représente les quantités demandées d'un ordre de fabrication de pantalons. Celui de la **Figure 39** donne les principaux paramètres concernant la création des nouveaux placements ainsi que leur utilisation.

	TROUSER										
	1	3	5	7	9	11	13	15	17	19	
rouge	5	12	7	19	14	17	25	23	11	9	142
bleu	3	9	14	21	31	14	17	26	14	12	161
jaune	8	7	23	19	24	16	17	15	6	8	143
vert	12	10	18	24	19	21	17	15	13	5	154
	28	38	62	83	88	68	76	79	44	34	600

Figure 38 : Ordre de fabrication de pantalons en 10 tailles et 4 couleurs

Nb maximum de tailles par placement	3	
Nb répétitions d'une même taille	2	
Nb minimum de plis	1 taille	1
	2 tailles	7
	3 tailles	10
Nb maximum de plis par section	40	

Figure 39 : Paramètres de résolutions de l'OF

On résout cet OF avec le module automatique de construction de sections d'Optiplan. On obtient après 2 secondes de traitement la solution détaillée dans le tableau de la **Figure 40**. La résolution est ensuite effectuée en utilisant la méthode SPGO. Etant donné la taille de l'ordre de fabrication et les contraintes sur la création des placements, l'OF n'est pas partitionné en sous-ordres et on effectue une sélection exhaustive des placements réalisables. On n'a donc pas de perte de généralité, la solution optimale du problème se trouve dans l'espace des solutions explorées. On arrête la résolution après 3 minutes 20 secondes, la meilleure solution trouvée est détaillée dans le tableau de la **Figure 41**. Ces deux solutions sont comparées dans le tableau de la **Figure 42**.

Placement N°	TAILLES									TISSUS				
	1	3	5	7	9	11	13	15	17	19	Rouge	Bleu	Jaune	Vert
1				1	1			1			14	21	15	15
2			1				1	1			7	14	16	17
3		1								1	11	9	6	10
4	1									1	5	3	8	5
5					2						0	5	4	2
6							1	1			10	0	0	0
7								1	1		8	3	0	0
8	1			1							0	0	0	7
9										1	4	9	0	0
10				1							5	0	4	2
11			1								0	0	7	1
12										1	0	5	0	3
13							1				0	0	0	4
14									1		1	2	0	0
15		1									1	0	1	0
16					1						0	0	1	0
17								1			0	0	1	0

Figure 40 : Solution fournie par Optiplan

Placement N°	TAILLES									TISSUS				
	1	3	5	7	9	11	13	15	17	19	Rouge	Bleu	Jaune	Vert
1	1		1					1			5	3	8	12
2				2			1				5	5	9	11
3		1					1		1		12	9	7	10
4				1	1					1	9	11	1	2
5							1	1		1	9	12	8	5
6		1			1						2	11	15	6
7					1					1	2	3	5	11
8					1						1	6	3	0
9								1			11	2	1	0
10									1		2	5	0	0

Figure 41 : Solution fournie par SPGO

La première différence observée entre ces 2 solutions est le nombre de placements utilisés. La solution d'Optiplan en utilise 17, alors que celle fournie par SPGO n'en utilise que 10. Il s'ensuit logiquement une baisse des coûts placement et de coupe de plus de 35% car le nombre de sections à découper passe de 19 à 10. Les placements monotailles, qui sont les moins efficaces sont également moins employés dans la solution fournie par SPGO que dans celle d'Optiplan. Cela permet un léger gain en coût matière de l'ordre de 0,5%. La baisse du nombre de sections fait également diminuer les coûts de matelassage en diminuant les temps fixes nécessaires à chaque changement de sections. Globalement, le gain obtenu par la méthode SPGO en un peu plus de 3 minutes est de 4,97%.

L'examen de la répartition des coûts, montre que comme c'est le cas la plupart du temps, les coûts matières représentent plus de 80% du coût total pour les 2 solutions. Cependant, Les autres coûts passent de 15,61% du coût total pour la solution d'Optiplan à 11,59% du coût total pour la solution de SPGO. Cela représente une diminution de près de 25% de ces coûts et c'est principalement ici que le gain est obtenu. En effet, Optiplan fournit généralement de bonnes solutions en ce qui concerne l'utilisation matière. La méthode SPGO a pu fournir une solution encore un peu meilleure en coûts matière tout en diminuant de 25% les autres coûts.

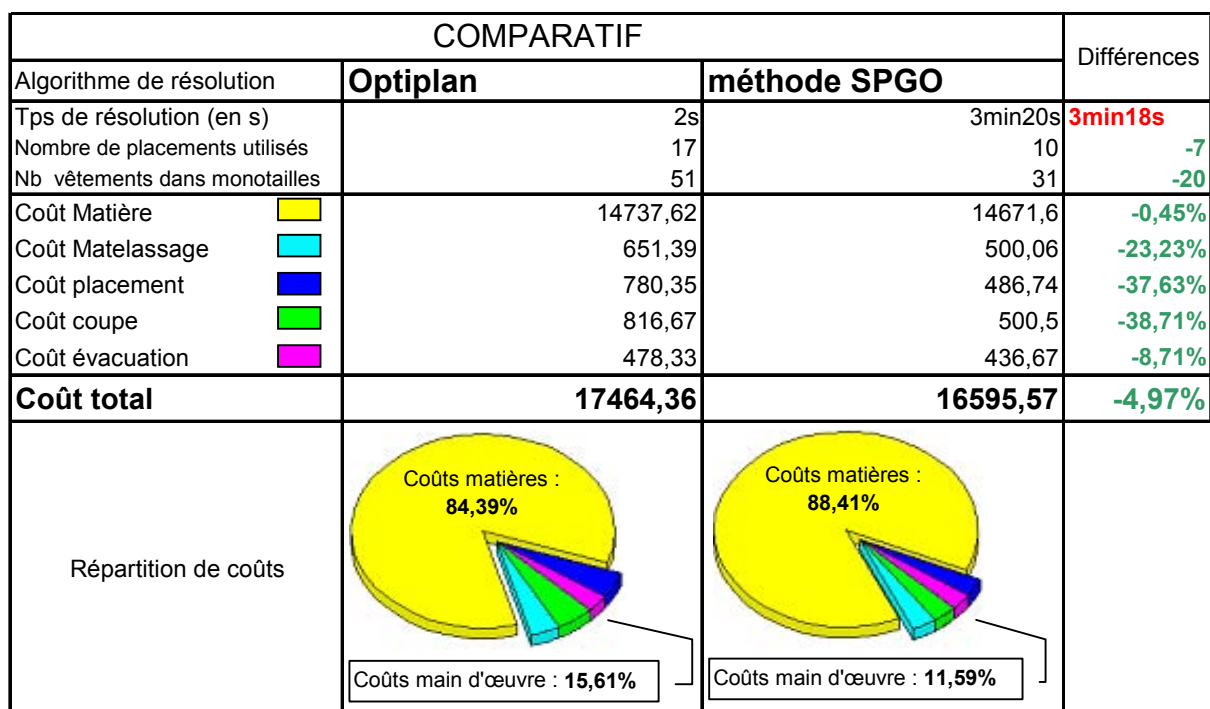


Figure 42 : Comparaison des solutions obtenues

Les 2 graphiques de la **Figure 43** donnent le coût moyen de chaque section ainsi que le pourcentage des quantités totales demandées dans l'OF que l'on découpe avec. Le premier graphique représente la solution fournie par Optiplan, l'allure des courbes est caractéristique, la solution utilise 17 placements mais les 3 premiers découpent plus de 70% des quantités totales alors que les 10 derniers placements ne permettent de découper que 10% de ces quantités. Parallèlement à cela, le coût moyen des sections augmente très fortement pour les placements les moins utilisés. Le graphique de droite représente la solution fournie par SPGO. L'utilisation des placements utilisés est beaucoup mieux répartie ; par exemple, 6 placements sur 10 permettent de découper plus de 10% des quantités totale. Par voie de conséquence, le

coût moyen des sections augmente de façon plus modérée. Ces graphiques permettent de distinguer les deux approches de résolution. La première est une résolution pas à pas qui permet de trouver de très bonnes sections au début de la résolution mais on court le risque de ruiner le reste de la solution en créant de grandes disparités dans les quantités restant à produire ce qui oblige à utiliser beaucoup de placements pour peu d'articles découpés. Dans le deuxième cas, la résolution est vue globalement, on compare des solutions entières et non pas uniquement des sections. Cela empêche alors d'utiliser un placement qui serait très bon mais qui impliquerait ensuite de ne créer que des sections dont le coût est prohibitif.

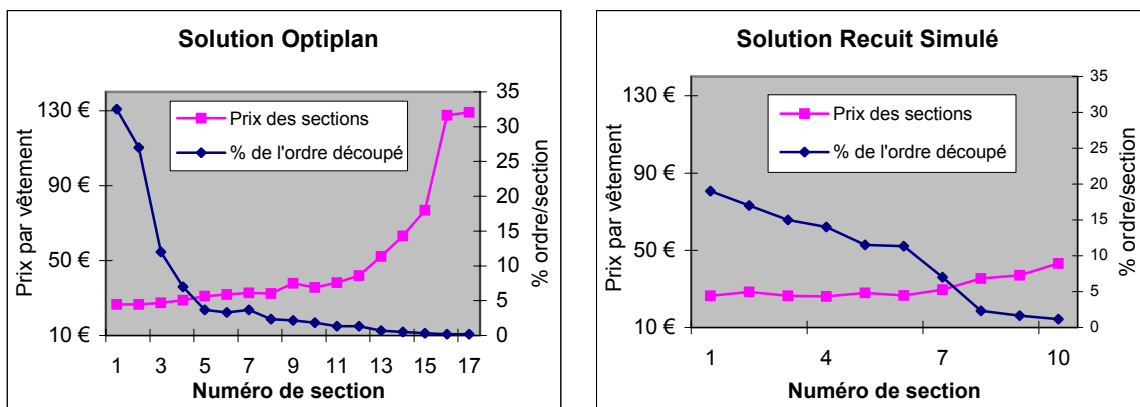


Figure 43 : Comparaison de l'évolution du prix des sections

L'exemple présenté ici représente de façon assez marquée les différentes approches de résolution du problème de la construction des sections. De manière plus générale, l'écart de performance entre les deux méthodes est moins important. Le gain obtenu par la méthode SPGO par rapport à Optiplan est compris entre 1% et 2% en moyenne selon les données du problème à résoudre.

8.2 Exemple 2 : Influence de la sélection des placements réalisables

Pour mesurer l'influence de la phase de sélection des placements que l'on pourra utiliser dans la résolution on revient à l'OF dont les quantités demandées sont décrites dans le tableau de la Figure 36. Cet ordre de fabrication contient 14 tailles et 5 tissus différents. Les contraintes sur la création des nouveaux placements sont les suivantes :

- Nombre maximum de tailles autorisées sur un placement : 7
- Nombre maximum de répétitions d'une même taille sur un placement : 2

On teste 3 manières différentes de sélectionner les placements réalisables. Les résultats obtenus sont présentés dans le tableau de la **Figure 44**. Ces trois solutions sont très comparables en terme de coût. (+0,12% pour la deuxième solution, -0,06% pour la troisième solution par rapport à la première solution qui correspond à une sélection exhaustive des placements). La différence de temps de résolution entre la première et les deux autres solutions correspond à l'énumération des 74 770 placements qui prend beaucoup de temps. De plus, le placement (0,0,1,0,0,1,2,1,0,0,1,0,1,0) est commun aux trois solutions, il permet de découper à lui seul près de la moitié des quantités de l'OF.

Type de Sélection	Exhaustive	Non Exhaustive	
		Nombre de placements par configuration et par tissu	
		20	120
Nombre total de placements sélectionnés	74770	1710	8319
Durée de résolution	6min26s	4min17s	4min21s
Coût de la solution	45916,61	45971,5	45889,1

Figure 44 : Résultats pour différentes méthodes de sélection des placements

La restriction de l'ensemble des placements au départ semble donc une bonne solution à l'augmentation du nombre de tailles des ordres de fabrication. On peut en effet obtenir quasiment les mêmes résultats (en terme de coûts) avec près de 50 fois moins de placements! De plus, dans cet exemple, la sélection exhaustive était encore réalisable, mais dans bien des cas cela est tout simplement trop complexe et il faut alors l'abandonner.

La difficulté de cette approche du problème est maintenant de déterminer un "bon" nombre de placements à sélectionner par configuration de placement et par tissu de l'OF. En effet, si l'on choisit un nombre de placements trop faible on risque de laisser de côté des placements qui auraient été très utiles. En revanche, si on sélectionne un trop grand nombre de placements, l'exploration de l'espace des solutions réalisables par le recuit simulé sera peu efficace dans un temps de quelques minutes. Dans la pratique, on considère qu'au-dessus de 10 000 placements, il devient difficile d'avoir une bonne exploration dans cet intervalle de durée.

L'ordre de fabrication représenté sur la **Figure 45** est un exemple de cas où la sélection exhaustive des placements est impossible. En effet, cet OF contient 58 tailles différentes provenant de deux modèles. Même avec des contraintes sur la création des nouveaux placements, le nombre de placements réalisables est énorme¹.

Tailles Tissus	23	24	25	26	27	28	29	30	31	32	44	46	48	50	52	54	56	58	60	62	88	90	94	98	102	106	110	114	118
Rouge	10	20	25	30	20	50	45	46	20	10	60	80	100	125	140	100	116	80	30	20	10	25	35	60	40	30	10	5	2
Orange	25	30	35	55	25	20	10	10	5	5	30	120	150	220	250	220	160	84	62	31	25	40	60	60	45	37	18	11	9
Beige	25	55	67	82	74	68	51	26	12	8	66	79	95	145	153	127	118	126	104	87	1	5	8	10	8	4	1	1	1
Marron	3	8	12	15	9	7	4	2	1	1	12	24	28	55	24	23	21	14	10	7	2	6	7	16	3	1	1	2	1
Gris	25	28	45	55	42	40	30	10	5	5	16	28	38	82	64	50	40	32	10	10	5	16	22	28	16	10	5	5	5
Total	88	141	184	237	170	185	140	94	43	29	184	331	411	627	631	520	455	336	216	155	43	92	132	174	112	82	35	24	18

23	24	25	26	27	28	29	30	31	32	44	46	48	50	52	54	56	58	60	62	88	90	94	98	102	106	110	114	118	Total	
(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	
16	20	25	40	65	42	44	30	20	5	35	75	85	110	140	110	100	95	75	60	35	45	55	70	55	40	30	20	10	2896	
5	8	12	14	16	10	8	4	2	1	10	20	25	35	55	50	41	25	22	10	8	10	15	18	11	8	4	3	1	2303	
5	8	17	19	24	26	20	13	10	6	37	49	82	94	81	34	22	13	6	4	12	24	36	48	36	24	12	6	6	2381	
11	22	26	34	42	31	18	16	13	4	12	16	22	24	26	18	19	14	12	14	2	6	17	22	22	10	6	8	4	810	
28	32	32	34	30	32	28	26	15	10	50	60	80	110	70	65	50	50	42	30	12	24	28	38	20	18	12	10	8	1811	
65	90	112	141	177	141	118	89	60	26	144	220	294	373	372	277	232	197	157	118	69	109	151	196	144	100	64	47	29	10201	

Figure 45 : Exemple d'OF pour lequel la sélection exhaustive est impossible

On réalise plusieurs résolutions de cet OF avec pour chacune d'entre elles un nombre de placements sélectionnés différents. Les résultats sont détaillés dans le tableau de la **Figure 46**.

¹ Si le nombre maximum de tailles autorisées sur un placement est 8 et si le nombre maximum de répétitions d'un même taille est 2. Alors le nombre total de placements réalisables est supérieur à 5 milliards !

Nombre de placements par configuration et par tissu	Nombre total de placements sélectionnés	Coût de la solution initiale	Coût de la solution finale	Amélioration entre la solution initiale et la solution finale
1	173	670,103	665,222	0,73%
5	630	659,698	654,680	0,76%
8	975	656,434	651,396	0,77%
10	1 200	655,924	651,820	0,63%
12	1 433	654,598	649,356	0,80%
15	1 761	657,046	652,911	0,63%
20	2 311	653,221	652,847	0,06%
30	3 408	653,986	653,229	0,12%
50	5 603	652,659	651,001	0,25%
100	11 088	651,517	651,332	0,03%
150	16 540	650,824	650,507	0,05%
300	32 412	649,395	648,764	0,10%

Figure 46 : Résultats en fonction du nombre de placements sélectionnés au départ

La 2^{ème} colonne donne le nombre total de placements sélectionnés. Les valeurs de la 3^{ème} colonne correspondent aux coûts des solutions initiales trouvées dans chacun des cas. Ces valeurs sont à comparer avec les coûts obtenus pour les solutions finales à la fin de la résolution. Le graphique de la **Figure 47** représente les coûts des solutions initiales et finales trouvées en fonction du nombre de placements sélectionnés. La partie colorée en orange correspond donc au gain réalisé durant le recuit simulé.

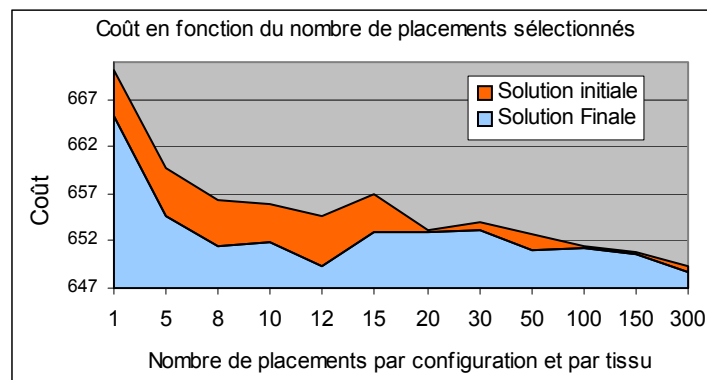


Figure 47 : Evolution des gains entre solution initiale et solution finale

On observe que, en règle générale, plus le nombre de placements sélectionnés augmente, plus la première solution trouvée sera bonne. Cela s'explique car la première solution est calculée en fonction de la sélection de placements retenue. Donc, plus cette sélection est importante plus le choix est large pour le calcul de la solution initiale.

En revanche, le pourcentage d'amélioration que l'on obtient au fil des itérations du recuit simulé à partir de cette solution initiale a plutôt tendance à diminuer lorsque le nombre de placements sélectionnés augmente. En effet, un espace des solutions réalisables important est difficile à explorer dans un temps limité. En outre, on peut supposer que plus la première solution est bonne plus on est proche de la solution optimale, et les améliorations sont moins franches lorsque l'on se rapproche de celle-ci.

En ce qui concerne le coût de la solution finale, la tendance est assez difficile à appréhender. En effet, dans certains cas avec peu de placements au départ et une première solution moyenne, on arrive à une très bonne solution finale grâce à une bonne exploration de l'espace des solutions lorsque 1433 placements sont sélectionnés par exemple. Cependant, le meilleur résultat est obtenu en sélectionnant plus de 30 000 placements ce qui a permis de trouver une excellente solution initiale. Il faut de toute façon souligner que comme le nombre d'itérations du recuit simulé est limité à quelques milliers au maximum la solution finale est extrêmement sensible à la solution initiale.

En règle générale, il est difficile de déterminer le nombre de placements à sélectionner qui va permettre d'obtenir la meilleure solution possible. C'est une des limites de ce système de sélection non exhaustive des placements qui reste cependant très utile et même indispensable lorsque la sélection exhaustive est impossible.

8.3 Exemple 3 : Partitionnement en plusieurs sous-ordres

Le tableau de la **Figure 48** reprend l'ordre de fabrication de la **Figure 45**, mais on choisit cette fois de partitionner l'ensemble des tailles pour résoudre plusieurs sous-ordres de manière indépendante. Le nombre de tailles maximum autorisées sur un placement est 8. On impose donc les sous-ordres créés comportent au minimum 8 tailles.

Tailles Tissus	23	24	25	26	27	28	29	30	31	32	44	46	48	50	52	54	56	58	60	62	88	90	94	98	102	106	110	114	118
Rouge	10	20	25	30	20	50	45	46	20	10	60	80	100	125	140	100	116	80	30	20	10	25	35	60	40	30	10	5	2
Orange	25	30	35	55	25	20	10	10	5	5	30	120	150	220	250	220	160	84	62	31	25	40	60	60	45	37	18	11	9
Beige	25	55	67	82	74	68	51	26	12	8	66	79	95	145	153	127	118	126	104	87	1	5	8	10	8	4	1	1	1
Marron	3	8	12	15	9	7	4	2	1	1	12	24	28	55	24	23	21	14	10	7	2	6	7	16	3	1	1	2	1
Gris	25	28	45	55	42	40	30	10	5	5	16	28	38	82	64	50	40	32	10	10	5	16	22	28	16	10	5	5	5
Total	88	141	184	237	170	185	140	94	43	29	184	331	411	627	631	520	455	336	216	155	43	92	132	174	112	82	35	24	18

23	24	25	26	27	28	29	30	31	32	44	46	48	50	52	54	56	58	60	62	88	90	94	98	102	106	110	114	118	Total	
(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	2896
16	20	25	40	65	42	44	30	20	5	35	75	85	110	140	110	100	95	75	60	35	45	55	70	55	40	30	20	10	2303	
5	8	12	14	16	10	8	4	2	1	10	20	25	35	55	50	41	25	22	10	8	10	15	18	11	8	4	3	1	2381	
5	8	17	19	24	26	20	13	10	6	37	49	82	94	81	34	22	13	6	4	12	24	36	48	36	24	12	6	6	810	
11	22	26	34	42	31	18	16	13	4	12	16	22	24	26	18	19	14	12	14	2	6	17	22	22	10	6	8	4	1811	
28	32	32	34	30	32	28	26	15	10	50	60	80	110	70	65	50	50	42	30	12	24	28	38	20	18	12	10	8	10201	
65	90	112	141	177	141	118	89	60	26	144	220	294	373	372	277	232	197	157	118	69	109	151	196	144	100	64	47	29	10201	

Figure 48 : Partitionnement de l'ensemble des tailles de l'ordre de fabrication

On crée un arbre hiérarchique de l'ensemble des tailles (cf. Chapitre 6 : Partitionnement des OF de grande taille). En remontant progressivement dans la hiérarchie trouvée, la première partition respectant la condition de 8 tailles au minimum par classe est la suivante :

$$C_1 = \{23_1, 24_1, 25_1, 26_1, 27_1, 28_1, 29_1, 60_1, 62_1, 44_2, 46_2, 48_2, 50_2, 52_2\} \quad (14 \text{ tailles})$$

$$C_2 = \{30_1, 31_1, 32_1, 54_2, 56_2, 58_2, 60_2, 62_2, 88_2, 90_2, 94_2, 98_2, 102_2, 106_2, 110_2\} \quad (15 \text{ tailles})$$

$$C_3 = \{44_1, 46_1, 48_1, 50_1, 52_1, 54_1, 56_1, 58_1\} \quad (8 \text{ tailles})$$

$$C_4 = \{88_1, 90_1, 94_1, 98_1, 102_1, 106_1, 110_1, 114_1, 118_1\} \quad (9 \text{ tailles})$$

$$C_5 = \{23_2, 24_2, 25_2, 26_2, 27_2, 28_2, 29_2, 30_2, 31_2, 32_2, 114_2, 118_2\} \quad (12 \text{ tailles})$$

Ces 5 classes sont repérées par des couleurs différentes dans le tableau de la Figure 48. Pour les sous-ordres correspondant aux classes C_3 et C_4 on choisit une sélection exhaustive des placements. Pour les autres, étant donné le nombre de tailles supérieur qu'ils contiennent, on choisit une sélection non exhaustive des placements. On résout ensuite chaque sous-ordre. Les résultats obtenus sont présentés dans le tableau de la Figure 49 :

N° de classe	Nb de placements sélectionnés	Temps de résolution	Coût de la solution
1	4132	2min05s	183,529
2	4179	1min29s	144,1
3	3833	52s	186,09
4	4720	53s	40,968
5	3934	1min09s	87,817
Total	20798	6min28s	642,504

Figure 49 : Résultat obtenu pour chaque sous-ordre

Ces résultats sont à comparer avec ceux présentés sur la **Figure 46**, qui correspondent à une résolution directe, avec une sélection non exhaustive des placements. Le meilleur résultat obtenu alors était 648,764 avec une sélection de 32412 placements après près de 12 minutes de résolution. Ainsi, le partitionnement des tailles de l'OF permet d'obtenir deux fois plus rapidement une solution meilleure de près de 1% avec 50% de placements disponibles en moins.

Cette méthode paraît donc être la plus appropriée pour traiter les OF contenant un grand nombre de tailles. En effet, elle permet de réduire considérablement l'espace des solutions réalisables en autorisant de plus une meilleure recherche des solutions sur chacun des sous-ordres. En effet, comme ceux-ci contiennent peu de tailles par rapport à l'ordre de fabrication initial, on peut souvent les résoudre avec une sélection exhaustive des placements réalisables pour chaque sous-ordre. De plus, à nombre d'itération égal, le temps de résolution est plus court.

8.4 Exemple 4 : Influence des déviations sur les quantités

Dans ce paragraphe, on mesure l'effet des déviations sur les quantités à produire sur les coûts moyens par vêtement. Pour cela, on résout une nouvelle fois l'OF dont les quantités à produire sont présentées dans le tableau de la **Figure 36**. On définit pour toutes les quantités à produire un même pourcentage de déviation autorisé α . Ainsi, si pour une taille i et un tissu j la quantité demandée initialement est B_{ij} , la quantité effectivement découpée D_{ij} dans la solution devra satisfaire : $(1-\alpha)B_{ij} \leq D_{ij} \leq (1+\alpha)B_{ij}$. On augmente progressivement le pourcentage de déviation autorisée α afin d'observer l'évolution du coût moyen par vêtement.

Les résultats obtenus, en utilisant la méthode de résolution décrite au *Chapitre 7 : Résolution avec déviations sur les quantités*, sont présentés dans le tableau de la **Figure 50**. Pour chaque valeur de α , on note :

- Le nombre de vêtements découpés en plus : OVERCUT
- Le nombre de vêtements découpés en moins : UNDERCUT
- L'écart entre le nombre total de vêtements découpés et le nombre total de vêtements demandés initialement
- La somme de tous les écarts entre les quantités découpées et les quantités demandées initialement
- Le coût moyen par vêtement et la proportion de ce coût par rapport à une résolution sans déviation

Déviations autorisées sur les quantités	OVERCUT: (nb de vêtements suppl.)	UNDERCUT: (nb de vêtements en moins)	Ecart nb. Total de vêtements	Ecarts cumulés	Coût Moyen par vêtement	Coût de la solution / Coût sans déviation
0%	0	0	0	0	37,081	100,000%
1%	0	0	0	0	37,081	100,000%
2%	5	7	-2	12	36,994	99,765%
3%	12	15	-3	27	36,836	99,339%
4%	21	19	2	40	36,814	99,280%
5%	25	30	-5	55	36,577	98,641%
6%	32	32	0	64	36,478	98,374%
7%	42	23	19	65	36,52	98,487%
8%	37	48	-11	85	36,299	97,891%
9%	45	38	7	83	36,355	98,042%
10%	47	42	5	89	36,096	97,344%
11%	54	52	2	106	36,196	97,613%
12%	55	56	-1	111	36,22	97,678%
13%	55	69	-14	124	36,015	97,125%
14%	63	62	1	125	36,047	97,212%
15%	65	62	3	127	35,93	96,896%
16%	60	92	-32	152	35,98	97,031%
17%	51	108	-57	159	35,908	96,837%
18%	85	78	7	163	35,913	96,850%
19%	70	97	-27	167	35,886	96,777%
20%	93	98	-5	191	35,726	96,346%

Figure 50 : Influence du % de déviation autorisé sur le coût moyen des vêtements

Pour illustrer ces résultats, le graphique de la **Figure 51** représente la proportion du coût moyen par vêtement par rapport à une résolution sans déviation en fonction de la valeur de α .

La tendance générale observée correspond bien à une diminution du coût moyen par vêtement lorsque le pourcentage de déviation autorisé augmente ce qui était le but recherché. On a par exemple un gain de plus de 3,5% lorsque le pourcentage de déviation autorisé est de 20% par rapport à la résolution avec les quantités exactes. La diminution peu régulière de ces coûts peut-être due aux caractéristiques intrinsèques des problèmes traités mais aussi au caractère non déterministe de l'algorithme du recuit simulé. On remarque également une plus grande quantité d'UNDERCUT que d'OVERCUT en moyenne. Cela est principalement dû à la méthode d'obtention d'une solution réalisable (cf. 7.4) après la résolution du problème relaxé qui ne traite pas tout à fait de la même façon les vêtements découpés en plus et ceux découpés en moins.

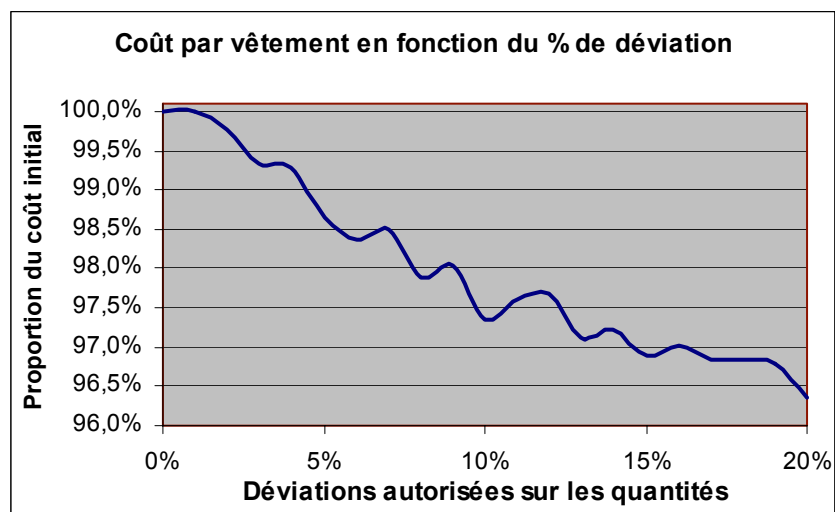


Figure 51 : Diminution du coût moyen avec l'augmentation des déviations autorisées

Chapitre 9 : Qualité des solutions

9.1 Introduction

Afin d'obtenir des résultats dans des temps raisonnables (quelques minutes au plus), on a plusieurs fois dans les chapitres précédents relaxé des contraintes, morcelé le problème en sous problèmes, considéré uniquement des sous-ensembles de solutions réalisables. Toutes ces méthodes permettent d'obtenir des solutions réalisables dans des temps assez courts mêmes pour de très gros problèmes. Cependant, on n'est jamais sûr d'avoir atteint la solution réellement optimale du problème. Plusieurs tentatives ont été faites afin d'obtenir avant ou pendant la résolution une borne inférieure de la solution optimale. Cela est possible lorsque la sélection des placements est exhaustive en relaxant des contraintes et en supprimant des variables. Cependant, on ne l'utilise pas car la qualité de cette borne (est-elle proche ou éloignée de la solution optimale ?) dépend beaucoup des données des problèmes considérés ce qui ne permet pas de statuer sur la qualité d'une solution obtenue avec SPGO.

Afin d'avoir tout de même une indication sur la qualité des solutions, nous allons comparer pour une sélection de problèmes représentatifs, les résultats obtenus par la méthode SPGO à ceux donnés par un logiciel de résolution exacte utilisant la méthode de Branch & Bound : Cplex.

9.2 Coûts matière uniquement

Dans le cas général il est impossible en pratique de résoudre le problème de la construction des sections directement à l'aide d'un algorithme de Branch & Bound classique car les temps de résolution sont prohibitifs. Ceci est notamment les cas lorsque le nombre de placements est important. On peut toutefois créer des exemples pour lesquels la fonction économique est composée exclusivement des coûts matières (et aussi les coûts de matelassage). Les variables Y , Z et E sont alors supprimées du problème. Même si le nombre de variables reste élevé, cela

réduit considérablement le nombre de contraintes dans le problème qui est alors, en principe, plus facile à résoudre pour des logiciels de programmation linéaire type Cplex.

On compare les résultats obtenus avec une résolution directe par Cplex et ceux fournis par la méthode SPGO sur 5 exemples décrits dans le tableau de la **Figure 52**.

La première colonne du tableau donne la taille du fichier MPS donné à Cplex, c'est un bon indicateur de la taille des problèmes à traiter. Les autres colonnes apportent plus de précisions :

- Le nombre de placements réalisables sélectionnés pour la résolution
- Le nombre de variables X , Y et Z (aucunes variables Y et Z pour la minimisation des coûts matières)
- Le nombre de contraintes (respect des quantités demandées pour la minimisation des coûts matières)

	Taille MPS (Mo)	Nombre de Placements	Nombre de Variables X	Nombre de Variables Y	Nombre de Variables Z	Nombre de contraintes
1	24,30	29 330	146 650	0	0	70
2	7,48	9 947	49 735	0	0	70
3	1,91	2 849	14 245	0	0	70
4	2,88	8 271	16 542	0	0	18
5	0,94	3 060	6 120	0	0	18

Figure 52 : 5 exemples de résolution avec coût matière uniquement

Le tableau de la **Figure 53** permet de comparer les solutions obtenues dans chacun de ces cas.

	Résultat SPGO		Résultat Cplex				Amélioration possible ?
	Coût	Temps	Coût	Temps	Meilleure Borne	Coût LP	
1	42320	9min45s	42263	2h43min47s	42241	42241	0,19%
2	42853	9min45s	42808	9min11s	42791	42791	0,14%
3	43418	4min27s	43406	4min16s	43388	43388	0,07%
4	28256	3min11s	28218	13min	28204	28204	0,18%
5	29012	2min56s	28994	25min	28985	28985	0,09%

Figure 53 : Comparaison des solutions obtenues avec Cplex et avec SPGO

Les 2 premières colonnes du tableau donnent le coût et le temps de résolution des solutions obtenues avec la méthode SPGO. Pour les résultats obtenus avec Cplex, la colonne « coût » correspond à la meilleure solution trouvée (pas nécessairement optimale) et la colonne « temps » donne la durée de résolution avant que Cplex soit stoppé. Deux colonnes sont également rajoutées :

- La colonne « meilleure borne » est le meilleur (le plus grand) minorant de la solution optimale que Cplex a obtenue avant l'arrêt de la résolution.
- La colonne « coût LP » est le coût de la résolution du problème en nombres réels (coût du simplexe résolu au nœud racine du Branch & Bound).

La colonne « amélioration possible » est la différence en pourcentage entre le coût de la solution obtenue avec la méthode SPGO et la meilleure borne trouvée par Cplex. Ce pourcentage représente l'écart maximal entre la solution obtenue avec SPGO et la solution optimale du problème.

Ce type de problèmes se caractérise par des solutions réalisables dont le coût est très proche du coût des solutions en nombres réels. Dans les 5 cas, Cplex trouve une solution réalisable légèrement meilleure que celles fournies par la méthode SPGO. Cependant, bien que ces solutions aient des coûts très proches de la borne sur la solution optimale, Cplex peine à trouver la solution réellement optimale ou à prouver que celle déjà trouvée est optimale. En effet, dans tous les cas la meilleure borne trouvée reste le coût de la solution du problème en nombres réels. Cependant, l'écart maximal entre le coût de la solution fournie par le recuit

simulé et le coût de la solution optimale reste très faible. Cet écart est inférieur à 0,2% de la meilleure borne dans les 5 cas.

9.3 Problèmes à solution optimale connue

Sur certains problèmes de taille réduite, Cplex parvient à résoudre le problème de manière exacte et renvoie la solution optimale. Cette situation survient lorsque l'on n'autorise pas la création de nouveaux placements. On résout le problème en utilisant un ensemble de placements existants qui est très souvent de taille limitée. Le tableau de la **Figure 54** décrit plusieurs exemples de ce type :

	Taille MPS (Mo)	Nombre de Placements	Nombre de Variables X	Nombre de Variables Y	Nombre de Variables Z	Nombre de contraintes
1	0,05	35	175	35	35	175
2	0,05	35	175	35	35	175
3	0,02	14	70	14	14	76
4	0,02	14	70	14	14	76
5	0,06	41	205	41	41	194
6	0,04	30	150	30	30	140
7	0,04	30	150	30	30	140
8	0,07	47	235	47	47	233
9	0,37	624	624	624	624	2 505
10	0,08	145	145	145	145	589
11	0,04	57	114	57	57	246
12	0,08	92	276	92	92	398
13	0,25	300	900	300	300	1 230

Figure 54 : Exemples pour lesquels on connaît la solution optimale

Le tableau de la **Figure 55** présente les résultats obtenus :

	Résultat SPGO		Résultat Cplex			Amélioration possible ?
	Coût	Temps	Coût	Temps	Borne	
1	29630	3min15s	29615	2h47min	optimum	0,07%
2	20958	3min34s	20955	52min	optimum	0,01%
3	8447	3min02s	8447	2s	optimum	0,00%
4	21819	2min34s	21696	4s	optimum	0,56%
5	19585	3min19s	19527	1h9min	optimum	0,30%
6	80638	4min58s	80616	13min50s	optimum	0,03%
7	119527	4min55s	119445	2h44min	optimum	0,07%
8	42855	4min25s	42855	19min44s	optimum	0,00%
9	14402	1min	14347	2min10s	optimum	0,38%
10	14452	1min37s	14452	2min21s	optimum	0,00%
11	32485	2min11s	32485	27min47s	optimum	0,00%
12	2706	1min35s	2706	1min10s	optimum	0,00%
13	2668	5min54s	2668	1h29min	optimum	0,00%

Figure 55 : Comparaison des solutions obtenues avec Cplex et avec SPGO

Dans la colonne « temps » de SPGO, les valeurs en bleu correspondent à une résolution plus rapide que Cplex. Dans la colonne « amélioration possible », les valeurs en bleu correspondent aux cas pour lesquels la solution optimale est trouvée avec SPGO.

Dans ces exemples Cplex renvoie à chaque fois la solution optimale cependant, les temps de résolution varient de quelques secondes à plusieurs heures. La solution optimale est trouvée relativement rapidement dans tous les cas, mais la confirmation de l'optimalité qui nécessite l'exploration de l'arbre peut être très longue. SPGO trouve la solution optimale environ une fois sur deux (6 fois sur 13) et, dans le cas contraire, il renvoie des solutions très proches :

- Amélioration possible < 1% dans tous les cas
- Amélioration possible < 0,1% dans 4 cas sur 7

Les écarts de temps sont eux beaucoup plus faibles puisque les temps de résolution pour la méthode SPGO sont compris entre 1 et 6 minutes.

9.4 Autres problèmes

Ces problèmes sont ceux rencontrés le plus fréquemment. Toutes les variables et toutes les contraintes sont présentes et le nombre de placements disponibles est suffisamment important

pour rendre la résolution exacte avec Cplex très difficile. Les tableaux des **Figure 56** et **Figure 57** décrivent plusieurs exemples de ce type ainsi que les résultats obtenus avec Cplex et avec SPGO.

	Taille MPS (Mo)	Nombre de Placements	Nombre de Variables X	Nombre de Variables Y	Nombre de Variables Z	Nombre de contraintes
1	4,17	2 509	12 545	2 509	2 509	10 010
2	1,00	665	3 325	665	665	2 729
3	0,17	119	595	119	119	546
4	1,45	881	4 405	881	881	3 558
5	0,28	193	965	193	193	817

Figure 56 : Exemple de problèmes dans le cas général

	Résultat SPGO		Résultat Cplex			Amélioration possible ?
	Coût	Temps	Coût	Temps	Borne	
1	48410	4min20s	49161	86h08min	43850	9,42%
2	46492	3min3s	47087	37min	45508	2,12%
3	48066	9min50s	47981	17h	47548	1,08%
4	18588	2min54s	18608	63h26min	17573	5,46%
5	42747	4min22s	43515	15h35min	42430	0,74%

Figure 57 : Comparaison des solutions obtenues avec Cplex et avec SPGO

On a du mal à statuer sur ces exemples de la qualité ou non des solutions fournies par la méthode SPGO. En effet, les bornes inférieures trouvées sont assez éloignées des valeurs des solutions obtenues SPGO. Cependant, ces solutions sont meilleures que celles trouvées par Cplex mêmes après de très longs temps de résolution.

Le fait que les solutions trouvées soient éloignées des bornes ne veut pas nécessairement dire que ces solutions sont mauvaises. En effet, les bornes inférieures fournies dans un algorithme de Branch & Bound sont trouvées en résolvant des problèmes en nombres réels or, des contraintes disjonctives telle que les contraintes du nombre de pli minimum sur un placement sont sans effet lorsque les variables qui apparaissent dans ces contraintes ne sont plus entières.

9.5 Résumé

Pour les cas où l'on a une idée très précise du coût des solutions optimales, les solutions fournies par SPGO sont très bonnes. Malheureusement, dans le cas général il est plus difficile de prouver la qualité des solutions. Cependant les solutions trouvées sont souvent meilleures que celle fournies par Cplex et ceci dans des temps bien plus courts.

Chapitre 10 : Conclusion

Le but de cette partie était de proposer une nouvelle méthode de résolution pour le problème de la construction des sections dans le logiciel Optiplan. Les paragraphes suivants permettent de faire le bilan de ce qui a été réalisé, en analysant les avantages et les inconvénients des algorithmes proposés. Dans le dernier paragraphe, quelques pistes d'amélioration sont suggérées.

10.1 SPGO : Une nouvelle méthode de construction des sections

Le premier travail effectué a été la modélisation mathématique de la construction des sections en un problème d'optimisation linéaire en nombres entiers. Cette modélisation nous a alors conduit à distinguer deux cas :

- Le nombre de placements utilisables est faible
- Le nombre de placements utilisables est grand

En pratique, le premier cas survient la plupart du temps lorsque la création de nouveaux placements est interdite. Les sections doivent alors être composées des placements existants qui sont souvent peu nombreux. On peut envisager alors une résolution directe en utilisant le module de résolution de problèmes linéaires en nombres entiers de Cplex.

Cependant, le cas général reste le deuxième. On a mis en place une méthode de résolution heuristique basée sur l'algorithme du recuit simulé. La stratégie est alors légèrement modifiée. On cherche la meilleure solution possible dans un temps limité plutôt que la solution théoriquement optimale du problème. La résolution se déroule alors en plusieurs étapes.

10.1.1 Méthodes de sélection des placements

Avant de commencer la résolution proprement dite, on sélectionne l'ensemble des placements qui seront susceptibles d'être utilisés dans la solution. Suivant les caractéristiques du problème traité, plusieurs méthodes de sélection sont possibles :

- Sélection de tous les placements satisfaisant les contraintes de production s'ils ne sont pas trop nombreux.
- Sélection des placements suivant leur configuration des répétitions de tailles lorsque le nombre de tailles de l'OF est grand mais que les contraintes sur la création des placements existent.
- Sélection des placements suivant le nombre de tailles qu'ils contiennent lorsque les contraintes sur la création des placements sont inexistantes ou très souples.

Bien évidemment, on a intérêt à faire une sélection exhaustive lorsque c'est possible car il n'y a pas de perte de généralité.

10.1.2 Résolution d'un sous-problème à chaque itération

Lorsque la sélection des placements est faite, on cherche une solution initiale puis la procédure de recuit simulé est lancée. A chaque itération, le problème est résolu pour un sous-ensemble de placements. Cette résolution doit être rapide pour pouvoir réaliser un nombre suffisant d'itérations. Pour cela, on résout à l'aide de Cplex un problème relaxé où plusieurs contraintes, dont l'intégrité des variables, sont abandonnées. Une procédure heuristique permet ensuite de transformer la solution renvoyée par Cplex de ce problème relaxé en une solution réalisable du problème de la construction des sections.

10.1.3 Partitionnement des OF de grande taille

Lorsque le nombre de tailles de l'ordre de fabrication est important, le nombre de placements réalisables est souvent énorme. La sélection non exhaustive des placements en fonction des configurations ou du nombre de tailles ne donne pas toujours de bons résultats. L'idée a été de résoudre les OF de grandes tailles morceau par morceau en construisant plusieurs sous-ordres. Pour cela, on cherche une partition de l'ensemble des tailles puis chaque classe de cette

partition est résolue séparément. Un indice de dissimilarité entre les différentes tailles a été créé afin de fournir un critère de regroupement. Pour construire les sous-ordres, on utilise une méthode de classification hiérarchique que l'on adapte pour avoir des classes de tailles comparables.

10.1.4 Résolution avec des déviations sur les quantités

Lorsque des déviations sur les quantités à découper sont autorisées, on ne cherche plus à minimiser le coût total d'une solution mais plutôt le coût moyen par article découpé. La fonction objectif du problème n'est plus linéaire mais se présente comme le quotient de 2 fonctions linéaires par rapport aux variables du problème. Dans un premier temps, on transforme ce problème en un problème linéaire avec un changement de variables. Ensuite la procédure d'obtention d'une solution réalisable est modifiée pour tenir compte au mieux des libertés sur le nombre d'articles à produire.

10.2 Avantages et inconvénients de la méthode

La méthode développée dans ces pages est plus globale que la méthode actuelle de résolution itérative section à section. Les résultats obtenus sont meilleurs en moyenne de 1 à 2%. Bien que ces gains puissent paraître limités, il faut les comparer au gain que permet de réaliser le logiciel Optiplan par rapport à une résolution manuelle qui sont de l'ordre de 2 à 5%. On a également constaté que, sur des problèmes dont la solution optimale est connue, les solutions fournies par la nouvelle méthode sont très proches de cet optimum. SPGO semble également plus sensible aux variations des paramètres ou des contraintes du problème traité.

Une autre caractéristique de la résolution avec la méthode SPGO est que l'on crée une solution complète du problème à chaque itération. Cela peut être vu comme un avantage car en pratique, il arrive que l'utilisateur ne soit pas satisfait de la solution fournie par le module automatique pour telle ou telle raison. La création d'un grand nombre de solutions permet d'en sauvegarder plusieurs au cours de l'algorithme. L'utilisateur a alors plus de choix pour sélectionner celle qui l'intéresse.

L'inconvénient majeur de la nouvelle méthode est qu'elle nécessite plus de temps de résolution. Les temps de calcul pour la méthode actuelle varient le plus souvent de quelques secondes à quelques dizaines de secondes. Dans le cas de SPGO, l'intervalle de durée est plutôt de l'ordre de quelques dizaines de secondes à une dizaine de minutes pour les OF les plus importants.

Cette augmentation de durée peut paraître rebutante au premier abord. Cependant, au regard du gain obtenu, cela n'est pas nécessairement prohibitif. De plus, il faut considérer que la construction des sections n'est qu'une étape du processus de production. En effet, la solution fournie implique le plus souvent la création de nouveaux placements. Or le processus de création de placements, qu'il soit manuel ou automatique, peut durer de quelques minutes à plusieurs heures pour les combinaisons de tailles les plus complexes. Enfin, les processus de production proprement dite comme le matelassage, la coupe l'évacuation peuvent également durer plusieurs heures. Dans ces conditions, il ne semble pas aberrant de consacrer quelques minutes à la construction des sections qui peuvent peut-être permettre de gagner une heure dans la salle de coupe.

10.3 Perspectives

A ce stade, la construction des sections à l'aide de la méthode SPGO permet de résoudre efficacement les plus petits comme les plus gros ordres de fabrication dans des temps de calcul raisonnables. Plusieurs axes d'amélioration peuvent toutefois être envisagés.

10.3.1 Perfectionnement des algorithmes annexes

A plusieurs reprises au cours de la résolution, on fait appel à des procédures heuristiques qui peuvent dans certains cas être perfectibles.

Lors de la sélection non exhaustive des placements par exemple, on choisit un échantillon de placements qui sont les plus susceptibles d'être utilisés dans la solution optimale. Pour cela on tient compte des quantités souhaitées pour chaque taille et chaque tissu. Pour être le plus efficace possible, il faudrait tenir compte du fait que lorsque l'on utilise un placement, les quantités restant à découper changent en conséquence. Une amélioration possible de la

procédure de sélection des placements serait donc de répondre à la question : « Quel placement a-t-on intérêt à utiliser sachant qu'on utilise cet autre placement avec tel nombre de plis ? »

A chaque itération du recuit simulé, la solution du problème relaxé peut violer plusieurs contraintes du problème de départ. Il y a bien sûr l'intégrité des variables mais aussi le nombre minimum de plis pour l'utilisation d'un placement par exemple. Une procédure permet ensuite de passer de cette solution à une solution réalisable pour la construction des sections. Lorsque les contraintes relaxées sont très restrictives, la solution du problème relaxé peut-être très éloignée d'une solution réalisable. L'optimisation qui a permis de calculer la solution du problème relaxé perd alors de son efficacité puisque la solution réalisable finalement renvoyée en sera très différente. L'amélioration consisterait alors à tenir compte d'une manière ou d'une autre de certaines contraintes dans le problème relaxé et à perfectionner la méthode d'obtention d'une solution réalisable pour que celle-ci soit la plus proche possible de la solution du problème relaxé.

Un point également clé de la résolution est la phase qui consiste à sélectionner un placement dans le sous-ensemble et à le remplacer par un autre provenant de la sélection exhaustive ou non réalisée en début d'algorithme. Actuellement, ces choix sont aléatoires aussi bien pour trouver celui qui va disparaître que pour trouver celui qui va le remplacer. Plusieurs tentatives ont été faites pour tenter de guider plus efficacement et plus rapidement l'algorithme vers les meilleures solutions :

- Elimination d'un placement peu ou pas utilisé dans le sous-ensemble
- Elimination d'un placement fortement utilisé
- Choix d'un placement contenant le même nombre de tailles que celui qui est supprimé
- Recherche des tailles les moins présentes dans les placements du sous-ensemble.
Choix d'un placement contenant ces tailles
- ...

Aucune de ces stratégies n'a donné de résultats significativement meilleurs que le simple choix aléatoire. Cependant, peut-être faut-il continuer à explorer cette voie car une chose est

sûre, le temps imparti est insuffisant pour explorer correctement l'ensemble de l'espace des solutions. Toute méthode permettant de se restreindre à des zones où les chances de trouver de bonnes solutions sont plus grandes serait la bienvenue.

10.3.2 Combinaisons des 2 approches de résolution

Il est également envisageable de combiner les 2 approches de résolution du problème de la construction de sections. En effet, le partitionnement de l'ensemble des tailles afin de résoudre plusieurs sous-ordres indépendamment peut également être utile pour la méthode de résolution actuelle. On pourrait aussi utiliser la méthode SPGO avec, comme solution initiale, la solution fournie par les algorithmes actuels d'Optiplan. On pourrait ainsi selon les cas et le temps disponible réaliser une résolution rapide ou une résolution approfondie. Enfin, on peut même imaginer résoudre chaque sous-problème correspondant à un sous-ensemble de placements avec les algorithmes actuels d'Optiplan en remplacement de la résolution du problème relaxé et de la procédure d'obtention d'une solution réalisable. Comme on le voit, on dispose désormais qu'un grand nombre de « briques de résolution » que l'on peut assembler différemment selon le temps et les moyens disponibles.

10.3.3 Amélioration des tables d'efficience

Parallèlement à cela, il existe un autre problème qui est commun aux deux approches de résolution et qui nuit à la qualité des solutions proposées. En effet, la plupart du temps la construction des sections conduit à proposer la création de nouveaux placements. Ces placements ont été jugés intéressants sur la base de tables d'efficience qui donnent pour un nombre de tailles donné dans un placement une estimation de son efficience. Ces tables d'efficience sont trop souvent assez sommaires. La qualité d'une solution peut alors totalement être remise en cause si les estimations de l'efficience des nouveaux placements se révèlent inexactes. Ce problème est l'objet de toute la seconde partie de cette thèse qui est donc consacrée à la mise au point d'une technique permettant d'estimer l'efficience d'un nouveau placement en fonction du nombre de tailles qu'il contient, de sa laize et d'autres paramètres qui le caractérisent.

Bibliographie

- [Aarts 1997]: Emile Aarts, Jan Karel Lenstra. *Local search in combinatorial optimization*. John Wiley and Sons – Interscience Series in Discrete Mathematics and Optimization, 1997, p91 à 120.
- [Chvatal1983]: Vasek Chvatal, *Linear Programming*. W.H. Freeman and Co. 1983
- [Hillier1995]: Frederick S. Hillier & Gerald J. Lieberman. *Introduction to operations research*. Sixth edition, Mc Graw-Hill 1995. Fractional programming p572, 573.
- [Kaufman1990]: Léonard Kaufman, Peter J. Rousseeuw. *Finding Groups In Data – An introduction to Cluster analysis*. John Wiley and Sons, 1990.
- [Schaible1981]: S. Schaible, *A survey of fractional programming* in S. Schaible and W.T. Ziemba eds., *Generalized concavity in optimization and economics*. Academic Press, New York, 1981. p417 à 440.

Deuxième Partie

Recherches sur la prévision de l'efficience des
placements réalisés avec DIAMINO EXPERT

Chapitre 1 : Origine et positionnement du problème

1.1 Origine du problème

On l'a vu dans la première partie, un placement est un regroupement de différentes pièces d'un même type tissu qui seront découpées ensemble. Ces pièces appartiennent à un ou plusieurs articles qui peuvent être par exemple plusieurs tailles d'un même modèle de vêtement. La matière découpée est disponible sous forme de rouleaux de tissus de largeur constante appelée laize. La construction d'un placement est un processus soit manuel soit automatique qui doit fournir pour un ensemble de pièces donné un agencement permettant de limiter au maximum les chutes de tissu. Comme la laize est connue, cela revient à minimiser la longueur nécessaire pour disposer toutes les pièces. Cette longueur de placement ainsi définie, dépend bien sûr du nombre et de la superficie des pièces qui le composent. Pour mesurer la qualité d'un placement, on calcule son efficacité qui est le rapport entre la superficie des pièces placées et la superficie de tissu utilisée.

Le processus de création d'un placement, qu'il soit manuel ou automatique peut durer de quelques minutes à plusieurs dizaines de minutes pour les placements contenant le plus de pièces. Dans ces conditions, il n'est pas envisageable de créer effectivement les centaines de placements que l'on voudrait tester pendant la construction des sections. Notre objectif est donc de mettre au point une technique d'estimation de l'efficacité d'un placement sachant le type de vêtement qu'il contient, le nombre de tailles, la largeur de tissu disponible, etc....

Avoir une estimation fiable de l'efficacité d'un placement alors qu'il n'est pas encore réalisé permettrait de rendre beaucoup plus efficace la phase d'optimisation de la construction des sections. En effet, le coût d'une solution de sections dépend pour une très grande part de l'efficacité des placements que l'on utilise. Pour déterminer si tel ou tel placement est

intéressant, on a donc tout intérêt à avoir une estimation aussi fine que possible de son efficience.

Le problème de l'optimisation du placement de formes dans une surface est un problème connu de longue date et beaucoup de travaux y font référence dans la littérature scientifique. Le problème du placement automatique fait partie d'une classe de problèmes désignés sous l'appellation « Cutting & Packing problems » en anglais. Une partie des travaux de [Victor J. Milenkovic](#) traite de ce problème dont il donne une brève description théorique dans [Milenkovic1991]. Pour plus de précision, le lecteur pourra également se référer à [Li1994] ou à [Bounsaythip1998]. En revanche, l'estimation du résultat de cette optimisation en fonction de paramètres liés aux formes à placer ou à la surface de placement est un problème nouveau.

1.2 Traitement actuel du problème

Dans Optiplan, pour déterminer l'efficience des nouveaux placements, on cherche à répartir les placements susceptibles d'être créés en plusieurs catégories. On sait de façon empirique que, pour un modèle donné, les placements contenant beaucoup de tailles sont les plus efficaces. En conséquence, le premier critère permettant de différencier les placements est le nombre de tailles qu'ils contiennent :

- Placements mono-taille
- Placements 2-tailles
- etc....

Le fait de fixer une efficience pour chaque nombre de taille permet de construire ce que l'on appellera par la suite une table d'efficience.

On peut ensuite raffiner ces catégories en considérant que l'efficience dépend également de la laize du tissu découpé. On crée alors plusieurs tables d'efficience qui correspondent chacune à un intervalle de laize donné. On peut par exemple créer les trois tables suivantes :

- Laize 100cm à 119cm..... table 1

- Laizes 120cm à 139cm..... table 2
- Laizes supérieures ou égales à 140cm..... table 3

L'efficacité qui sera ensuite appliquée pour les nouveaux placements dans chaque catégorie peut être déterminée de 2 façons différentes :

- Manuellement par l'utilisateur : Si celui-ci a une idée précise de l'efficacité des placements qui pourront être retenus dans chaque catégorie
- Par calcul de l'efficacité moyenne des placements déjà existants dans chaque catégorie pour les modèles considérés.

Cette deuxième méthode constitue donc une tentative d'analyse statistique des placements existants. L'inconvénient majeur de cette approche est qu'elle est très sommaire et il arrive parfois que la différence entre l'efficacité estimée d'un placement proposé par OPTIPLAN et l'efficacité « réelle » de ce placement après sa réalisation soit très grande. La qualité de la solution de sections fournie par OPTIPLAN est alors détériorée par ce manque de précision des efficacités des nouveaux placements.

1.3 Positionnement du problème

On choisit d'aborder ce problème également par une approche statistique mais qui soit si possible beaucoup plus précise. On va tenter de dégager les critères, les paramètres, qui influent sur l'efficacité en étudiant des placements déjà existants. Le modèle statistique résultant de l'étude de placements connus devra ensuite permettre d'estimer l'efficacité de nouveaux placements. Cependant, on ne tente pas (au moins dans un premier temps) de trouver un modèle général utilisable pour tous les types de placements¹. On préférera plutôt construire un modèle différent pour chaque type de placements (pantalons, chemises, lingerie...). En revanche, il pourra être utile de s'intéresser à des regroupements éventuels

¹ La problématique du placement est très générale. Ainsi la géométrie, la taille ou le nombre des formes à placer peuvent être très différentes selon le domaine que l'on traite. En conséquence, l'efficacité des placements ne varie pas toujours selon les mêmes critères.

entre types. En effet, afin de ne pas multiplier le nombre de modèles statistiques différents, il sera important (dans un second temps) d'analyser des paramètres influençant l'efficacité pouvant être communs à plusieurs types de placements. On pourra ainsi limiter le nombre de modèles statistiques construits en identifiant des grandes catégories de placements à l'intérieur desquelles l'efficacité varie selon les mêmes règles¹.

¹ Par exemple, si les placements contenant des airbags se comportent de la même manière que les placements contenant des canapés on regroupera ces deux types dans une même catégorie

1.4 Création d'une base de placements

La création de nouveaux placements est un élément très important dans le processus de l'industrie textile. L'obtention de placements efficaces permet de limiter au maximum les pertes de tissus.

La création du placement (l'optimisation du positionnement des pièces) peut être soit manuelle soit automatique. Dans le cas d'une création manuelle, l'optimisation est faite par le placeur qui a généralement une grande expérience dans son domaine. Parallèlement à cela, plusieurs sociétés proposent des logiciels de placement automatique. Bien que ce problème d'optimisation de placements de formes ait été abordé de nombreuses fois dans la littérature scientifique, il est très difficile de le résoudre en pratique (dans certains cas, les placeurs manuels obtiennent de meilleurs résultats). Cependant, grâce notamment à l'amélioration des capacités de calcul des ordinateurs, ces logiciels de placement automatique sont de plus en plus efficaces.

Le logiciel de placement automatique de LECTRA s'appelle Diamino Expert, c'est lui que l'on a utilisé pour créer la base de placement. En effet, l'utilisation du logiciel permet l'obtention de résultats reproductibles et en principe, moins subjectifs.

On touche cependant ici à un point sensible de la démarche, pour un ensemble de pièces donné (sauf cas trivial) on ne connaît pas le placement optimal correspondant, on ne connaît que le résultat de Diamino (le problème est le même pour le placement manuel) sans indication sur l'écart éventuel entre ce résultat et le meilleur placement possible. Le risque est alors de prendre les variations de performance du logiciel pour des caractéristiques intrinsèques des pièces à placer. Les résultats fournis par Diamino Expert dépendent bien sûr du temps alloué pour l'optimisation mais aussi de la version du logiciel dont les algorithmes sont régulièrement améliorés. Afin de construire le modèle statistique le plus fiable possible, tous les placements de la base sont donc créés avec la même version et avec un temps d'optimisation comparable.

Il peut arriver également que les résultats de Diamino ne soient pas tous cohérents entre eux.¹ Ainsi, si l'on peut améliorer manuellement de façon évidente l'efficacité d'un placement, on choisit de le faire plutôt que de conserver un placement dont l'efficacité est trop éloignée de l'efficacité optimale.

Une fois ces remarques faites, on construit un grand nombre de placements avec Diamino Expert en faisant varier au maximum leurs caractéristiques :

- Catégorie de vêtements (chemises, pantalons...)
- Plusieurs modèles pour chaque type de vêtements
- Nombre de tailles à placer (une taille est un ensemble de pièces)
- Laize (largeur de tissu disponible pour placer les pièces)

¹ Exemple de résultat incohérent : Pour une laize donnée Diamino Expert propose une bonne efficacité correspondant à une longueur x . Pour une laize un peu supérieure (+1 cm par exemple) on attend une longueur maximum de x (il suffit de reproduire le placement précédent), cependant il arrive qu'on obtienne une longueur supérieure à x et donc une efficacité moins bonne que ce que l'on pourrait attendre.

Chapitre 2 : Méthode de résolution

2.1 Remarques préalables

L'observation de la base de placements fournie par DIAMINO permet de faire les premières constatations suivantes :

- Les variations d'efficacité ne sont pas linéaires en fonction des paramètres des placements (par exemple : l'efficacité augmente beaucoup entre les placements contenant 1 seule taille et ceux contenant 2 tailles, en revanche l'augmentation est beaucoup plus faible entre 2, 3 et 4 tailles cf. **Figure 58**).

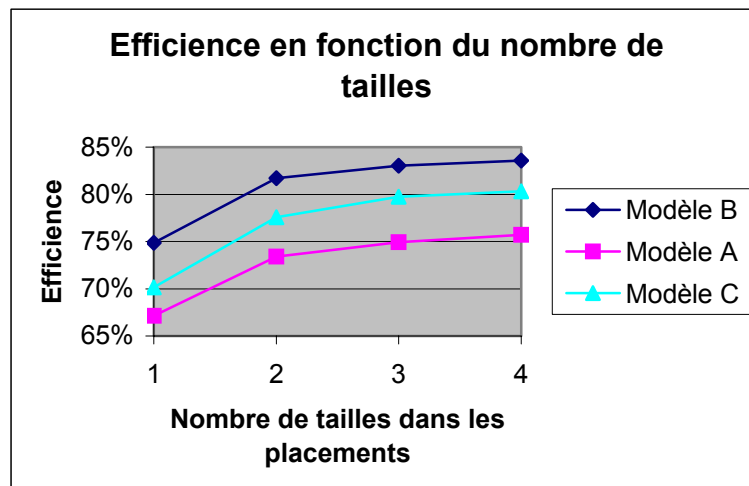


Figure 58 : Efficacité moyenne de 3 modèles de pantalons en fonction du nombre de tailles à placer

- Une toute petite variation des paramètres peut parfois entraîner un changement conséquent d'efficacité (particulièrement en ce qui concerne la largeur de tissu disponible qui permet ou non telle ou telle disposition des pièces cf. **Figure 59**).

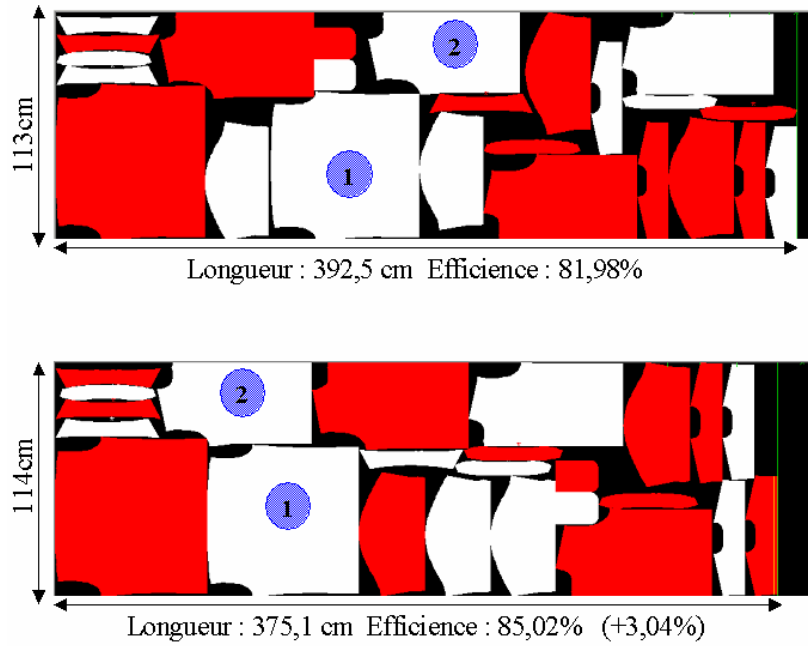


Figure 59 : Ecart important d'efficience pour une faible augmentation de la laize

2.2 Les réseaux de neurones

2.2.1 Références

Cette partie a pour but de décrire en quelques pages l'outil de modélisation que constituent les réseaux de neurones. Cependant, ce mémoire n'a pas l'ambition de fournir un exposé exhaustif des différentes méthodes neuronales. Pour plus de précisions, le lecteur pourra se référer avantageusement à [Dreyfus 2002] ainsi qu'aux autres références citées au fil des paragraphes suivants.

La démarche que nous avons suivie a été de recueillir un ensemble de techniques connues pour les adapter au cas de la prédiction d'efficacité des placements. Ce travail permet également de procurer aux réseaux de neurones un nouvel exemple concret d'application.

2.2.2 Pourquoi les réseaux de neurones ?

Les remarques précédentes nous ont conduit à ne pas utiliser les méthodes statistiques classiques (comme la régression par exemple).

Les réseaux de neurones n'ont pas les mêmes limitations et il est possible de les utiliser comme des approximateurs universels pour toute fonction bornée suffisamment régulière.

De plus l'approximation réalisée par un réseau de neurones est non linéaire par rapport à ces paramètres, le nombre de paramètres requis pour obtenir une précision donnée est moins important que dans le cas d'une approximation linéaire par rapport à ces paramètres.

On utilise un réseau non bouclé (pas de retour en arrière) contenant une couche de neurones cachés à fonction d'activation sigmoïde et un neurone de sortie linéaire.

Les entrées du réseau correspondent aux différents paramètres influençant l'efficacité et la sortie du réseau est la prédiction d'efficacité correspondant aux entrées.

2.2.3 Qu'est ce qu'un neurone ?

Un neurone est une fonction algébrique non linéaire, paramétrée, à valeurs bornées.

On représente souvent les neurones sous la forme d'un schéma de ce type (**Figure 60**):

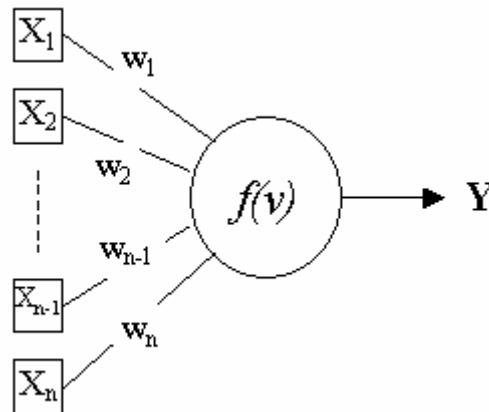


Figure 60 : Schéma d'un neurone

- ✓ Les X_i sont les variables d'entrées du neurone
- ✓ Les w_i sont les paramètres attachés à chaque variable d'entrée, appelés poids
- ✓ Les X_i et les w_i déterminent le potentiel v du neurone
- ✓ La fonction d'activation f permet de calculer la sortie $Y=f(v)$ du neurone

La potentiel v est une combinaison linéaire entre les entrées X_i et les poids w_i .

Le potentiel le plus souvent utilisé est la somme pondérée à laquelle s'ajoute un terme constant ou biais :

$$v = w_0 + \sum_{i=1}^n w_i x_i$$

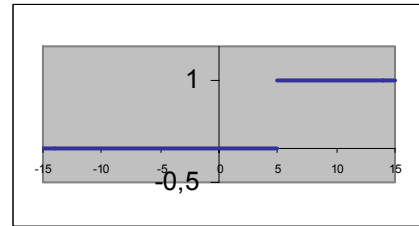
La sortie Y du neurone est donc la valeur de la fonction d'activation f pour le potentiel v correspondant :

$$Y = f(v) = f\left(w_0 + \sum_{i=1}^n w_i x_i\right)$$

La fonction d'activation f peut prendre des formes différentes selon les applications que l'on cherche à réaliser avec les neurones (cf. **Figure 61**).

Discontinue:

$$f(v) = \begin{cases} 1 & \text{si } v > \text{seuil} \\ 0 & \text{sinon} \end{cases} = \begin{cases} 1 & \text{si } w_0 + \sum_{i=1}^n w_i x_i > \text{seuil} \\ 0 & \text{sinon} \end{cases}$$



Continue:

$$f(v) = \text{ArcTan}(v) = \text{ArcTan}\left(w_0 + \sum_{i=1}^n w_i x_i\right)$$

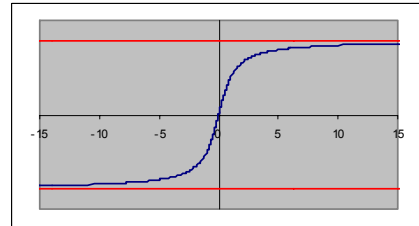


Figure 61 : Exemple de 2 fonctions d'activation possibles

2.2.4 Organisation des neurones en réseau

Comme on vient de le voir, un neurone réalise simplement une fonction non linéaire, paramétrée de ses variables d'entrée. L'intérêt des réseaux de neurones réside dans les propriétés qui résultent de leur association en réseaux, c'est à dire de la *composition* des fonctions non linéaires réalisées par chacun des neurones. On distingue deux types de réseaux de neurones : les réseaux non bouclés et les réseaux bouclés. Dans toute la suite, on s'intéressera uniquement aux réseaux de neurones non bouclés.

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones « connectés » entre eux, l'information circulant des entrées vers les sorties sans « retour en arrière ».

La seule contrainte sur le graphe des connections d'un réseau de neurones non bouclé est qu'il ne contienne pas de cycle. Néanmoins, la très grande majorité des applications des réseaux de neurones mettent en jeu des « réseaux à couches » tel que celui représenté sur la **Figure 62** :

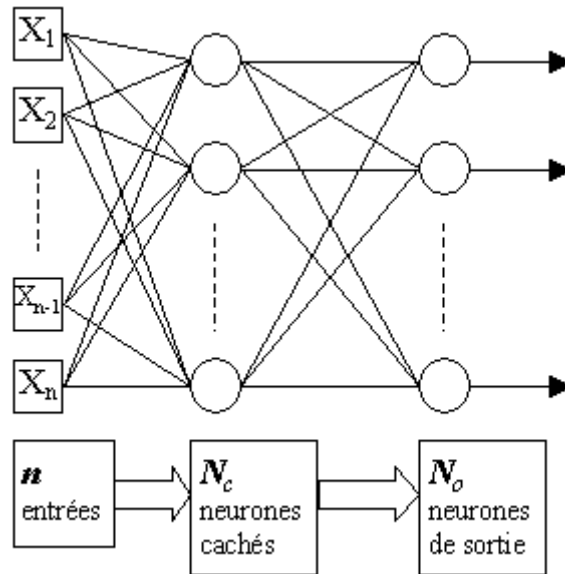


Figure 62 : Exemple de réseau de neurones non bouclé à une couche cachée

2.2.5 Réseau à une couche cachée de sigmoïdes et un neurone de sortie linéaire

Cette configuration représentée sur la Figure 63 est un cas très important en pratique dans l'utilisation des réseaux de neurones non bouclés.

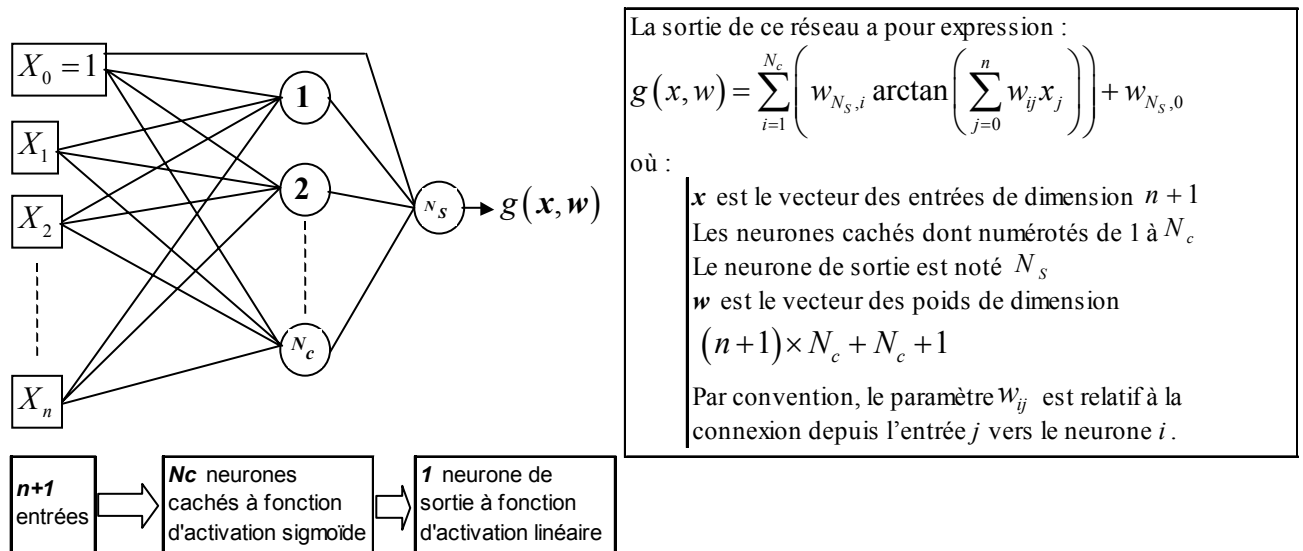


Figure 63 : Exemple de réseau à une couche de neurones cachés à fonction d'activation sigmoïde

La sortie du réseau $g(x, w)$ est une fonction linéaire des poids de la dernière couche de connexions qui relie les N_c neurones cachés au neurone de sortie.

Elle est une fonction non linéaire des poids de la première couche de connections qui relie les $n+1$ entrées du réseau aux N_c neurones cachés. Dans toute la suite on ne considérera que ces types de réseaux de neurones.

2.2.6 Propriété fondamentale : l'approximation parcimonieuse

Les propriétés des réseaux de neurones non bouclés suivantes sont décrites dans [Monari 1999] pages 7 et 8 et dans [Dreyfus 2002] pages 10 et 11.

2.2.6.1 L'approximation universelle

Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

Cette propriété est notamment vraie pour les réseaux de neurones à fonction d'activation sigmoïde présentés sur la **Figure 63**.

2.2.6.2 La parcimonie

La parcimonie représente la capacité d'un modèle à obtenir de bons résultats avec un nombre restreint de paramètres ajustables. Ainsi, on dit qu'une approximation est plus parcimonieuse qu'une autre si pour obtenir la même précision on utilise moins de paramètres.

On peut alors montrer qu'une approximation non linéaire par rapport à ses paramètres ajustables est plus parcimonieuse qu'une approximation linéaire par rapport à ses paramètres ajustables. La sortie des réseaux de neurones représentés sur la **Figure 63** est non linéaire par rapport aux poids reliant les entrées à la couche de neurones cachés. Une approximation avec ce type de réseaux est donc plus parcimonieuse qu'une approximation polynomiale par exemple. De plus, pour une précision donnée et un nombre de neurones cachés fixé, le nombre de paramètres ajustables est seulement proportionnel au nombre de variables de la fonction à approcher.

On voit donc là tout l'intérêt d'utiliser les réseaux de neurones comme approximateurs dès que le nombre de variables d'entrées est important (typiquement >3).

2.2.7 L'apprentissage supervisé des réseaux de neurones

2.2.7.1 But de l'apprentissage

On appelle « apprentissage » des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée.

On a vu, qu'un réseau de neurones réalise une relation algébrique (non linéaire) entre ces entrées et sa sortie (cas d'un seul neurone de sortie). On peut donc affecter à un tel réseau la tâche qui consiste à réaliser une fonction algébrique non linéaire. Celle-ci peut-être :

- Connue analytiquement : le réseau réalise alors une tâche *d'approximation de fonction*
- Inconnue analytiquement, mais pour laquelle on dispose de valeurs, en nombre fini, qui sont entachées de bruit (si ces mesures résultent de mesures effectuées sur un processus physique, économique, etc.) : le réseau réalise alors une tâche de *modélisation statique* ou de *régression*.

C'est essentiellement pour ce dernier type d'applications que sont mis en œuvre les réseaux de neurones à apprentissage supervisé.

On connaît donc en certains points la valeur que doit avoir la sortie du réseau en fonction des entrées correspondantes : c'est en ce sens que l'apprentissage est « supervisé ».

2.2.7.2 Technique d'apprentissage

Soit x le vecteur des entrées x_j , soit w le vecteur des poids w_i du réseau

On pose $g(x, w)$ = la sortie du réseau associée à x et w

On dispose d'un ensemble de vecteurs d'entrées: $x_k / k \in \{1 \dots N\}$

pour lesquelles on connaît la grandeur qui nous intéresse: $y_k / k \in \{1 \dots N\}$

On cherche par exemple à minimiser en w (cas de l'erreur quadratique): $J(w) = \sum_{k=1}^N (y_k - g(x_k, w))^2$

La fonction de coût J n'est pas quadratique par rapport aux poids w car la sortie du réseau $g(x, w)$ n'est pas linéaire par rapport à ceux-ci. Pour minimiser J , il faut donc avoir recours à des méthodes itératives qui modifient les paramètres du réseau en fonction du gradient de la fonction de coût.

2.2.7.2.1 Calcul du gradient par rétropropagation

On veut calculer la dérivée partielle de la fonction de coût J par rapport à chacun des poids du réseau w_i . La méthode de rétropropagation consiste à calculer ces dérivées partielles de manière récursive. En effet, dans un réseau à couches, les dérivées partielles par rapport aux poids de la couche i peuvent s'exprimer en fonction des dérivées partielles de la couche $i+1$.

L'**annexe 1** présente en détail cette technique de calcul du gradient de la fonction de coût.

2.2.7.2.2 Modification des paramètres en fonction du gradient de la fonction de coût

Une fois le gradient de la fonction de coût connu, on cherche à modifier les poids du réseau en conséquence. Le but étant bien sûr de faire décroître au maximum la fonction de coût.

On a alors le choix entre de nombreuses méthodes itératives de minimisation et notamment :

- **Les méthodes du premier ordre : gradient simple**

A l'itération i , on calcule les nouvelles valeurs des poids par la formule suivante :

$$w(i) = w(i-1) - \mu_i \nabla J(w(i-1)) \quad \text{avec } \mu_i > 0$$

- **Les méthodes du second ordre**

Elles utilisent le fait que l'on pourrait atteindre le minimum de J en une seule itération si l'on connaissait la matrice des dérivées secondes (matrice hessienne notée H) au point de minimum w^* . En effet, le développement de Taylor de la fonction J au point w au voisinage du minimum w^* permet d'écrire : $w^* = w - H(w)^{-1} \times \nabla J(w)$ (sous réserve que la matrice H soit inversible).

Bien sur cette formule n'est pas applicable puisque w^* n'est pas connu. Elle permet toutefois de mettre en place plusieurs algorithmes d'approximation itérative de la matrice hessienne.

Les méthodes du second ordre bien que plus lourdes à mettre en place sont beaucoup plus efficaces (en ce qui concerne le nombre d'itérations nécessaires) que les méthodes du premier ordre. On choisit d'utiliser l'une d'entre elles : la méthode de Levenberg-Marquardt dont une description détaillée est donnée dans l'**annexe 2**

2.2.8 Sélection des variables d'entrées : Méthode du descripteur sonde

La tâche la plus difficile lorsque l'on veut construire un modèle de prédiction, c'est de trouver les variables d'entrées qui influent sur la grandeur que l'on veut estimer. La démarche consiste généralement à sélectionner un ensemble de variables d'entrée potentielles qui peuvent raisonnablement avoir une influence sur la sortie. On essaie ensuite de mesurer l'influence réelle de chacune de ces variables afin de ne garder que les plus significatives.

La méthode du descripteur sonde (cette méthode est décrite en détail dans [Stoppiglia 1998]) permet de réaliser cela lorsque le modèle souhaité est un modèle linéaire par rapport à ces paramètres. On sait que ce n'est pas le cas des réseaux de neurones, cependant il est très difficile de mesurer l'influence réelle d'une variable dans un modèle non linéaire par rapport à ces paramètres. Pour cela on effectuera la même sélection des variables pour un modèle non linéaire que pour un modèle linéaire.

Dans un premier temps, on classe les variables d'entrée par ordre décroissant d'influence sur la sortie :

2.2.8.1 Procédure d'orthogonalisation de Gram-Schmidt.

On considère $(\mathbf{x}_i^k)_{1 \leq k \leq N}$ $1 \leq i \leq N_E$ l'ensemble des N_E vecteurs de \mathbb{R}^N correspondant aux N_E variables d'entrées connus pour les N exemples.

De même $(Y^k)_{1 \leq k \leq N}$ est le vecteur de \mathbb{R}^N correspondant aux valeurs des observations pour les N exemples. On définit θ_i comme l'angle entre les vecteurs \mathbf{x}_i et \mathbf{Y} .

- Si $|\theta_i| = 0$: les deux vecteurs sont colinéaires et la sortie \mathbf{Y} est proportionnelle à \mathbf{x}_i .
- Si $|\theta_i| = \frac{\pi}{2}$: les deux vecteurs sont orthogonaux et la sortie \mathbf{Y} est indépendante de \mathbf{x}_i .

On peut estimer facilement le carré du cosinus des angles θ_i . En effet :

$$\left| \begin{array}{l} \langle \mathbf{x}_i, \mathbf{Y} \rangle^2 = \|\mathbf{x}_i\|^2 \times \|\mathbf{Y}\|^2 \times \cos^2(\theta_i) \\ \Rightarrow \cos^2(\theta_i) = \frac{\langle \mathbf{x}_i, \mathbf{Y} \rangle^2}{\|\mathbf{x}_i\|^2 \times \|\mathbf{Y}\|^2} = \frac{((\mathbf{x}_i)^T \mathbf{Y})^2}{((\mathbf{x}_i)^T \mathbf{x}_i) \times ((\mathbf{Y})^T \mathbf{Y})} \end{array} \right. \quad \begin{array}{l} \text{où } \langle \cdot \rangle \text{ représente le produit scalaire} \\ \text{et } T \text{ l'opérateur de transposition} \end{array}$$

Plus $\cos^2(\theta_i)$ est proche de 1 plus la variable \mathbf{x}_i est pertinente dans un modèle linéaire pour expliquer \mathbf{Y} . Pour classer les variables d'entrées par ordre de pertinence décroissante, on procède de la façon suivante :

1. Choisir \mathbf{x}_{i^*} telle que $\cos^2(\theta_{i^*}) = \min_i \{\cos^2(\theta_i)\}$
2. Projeter les vecteurs de toutes les variables d'entrée restantes et de la sortie dans le sous espace orthogonal à \mathbf{x}_{i^*} .
3. On réitère l'opération 1 dans ce sous espace jusqu'à ce que toutes les variables d'entrées aient été classées.

2.2.8.2 Sélection des variables d'entrées à partir du classement obtenu

Lors de l'étape précédente, on a obtenu un classement des entrées par pertinence décroissante. Il faut donc maintenant trouver un rang critique au-delà duquel on rejète toutes les variables proposées. Si ce rang est choisi trop faible, on éliminera des variables utiles pour décrire la grandeur à estimer ce qui évidemment n'est pas souhaitable. A contrario, si le rang est choisi trop élevé, on conservera des variables non pertinentes ce qui peut dégrader les performances du modèle.

Pour trouver ce rang, on applique le principe suivant :

On ajoute aux variables candidates, une variable supplémentaire (appelée descripteur sonde) dont les valeurs sont calculées aléatoirement. On procède au classement des variables vu précédemment en incluant cette dernière et on élimine toutes les variables moins bien classées que le descripteur sonde.

En fait, le rang du descripteur sonde dépend de la réalisation de la variable aléatoire et on ne peut jamais écarter totalement le risque de garder une variable d'entrée non pertinente ou inversement. Pour cela, on se donne un risque $\alpha \in]0,1[$ d'accepter une variable non pertinente. A chaque itération de la procédure de classement décrite plus haut, on calcule la probabilité $P(\cos^2(\theta_{i^*}) \leq \cos^2(\theta_{sonde}))$ que le descripteur sonde soit plus pertinent que x_{i^*} . Si $P(\cos^2(\theta_{i^*}) \leq \cos^2(\theta_{sonde})) < \alpha$ on garde la variable x_{i^*} et on continue, sinon on rejète x_{i^*} ainsi que toutes les autres variables non encore sélectionnées. La probabilité $P(\cos^2(\theta_{i^*}) \leq \cos^2(\theta_{sonde}))$ est obtenue en calculant la fonction de répartition de la variable aléatoire $\cos^2(\theta_{sonde})$.

2.2.9 Sélection de modèles : Leave-one-out virtuel

Le but de ce paragraphe est de présenter une méthode (décrite dans [Monari 1999]) permettant de choisir parmi plusieurs modèles celui qui sera le « meilleur ». Etant donnée la structure de réseau de neurones que l'on s'est fixé (une couche cachée), choisir entre plusieurs modèles revient à choisir le nombre de neurones dans la couche cachée.

Une fois l'apprentissage du réseau effectué, l'idée la plus naturelle pour mesurer l'efficacité du modèle obtenu est de calculer l'erreur quadratique moyenne obtenue sur l'ensemble d'apprentissage (EQMA) :

$$E_A = \sqrt{\frac{1}{N} \sum_{k=1}^N (\mathbf{y}^k - g(\mathbf{x}^k, \mathbf{w}))^2} = \sqrt{\frac{1}{N} \sum_{k=1}^N (r^k)^2} \quad EQMA$$

En fait, on sait que cette erreur peut être rendue arbitrairement petite en augmentant le nombre de neurones dans la couche cachée. Ce n'est donc pas un bon critère pour comparer différents modèles de réseaux. En effet, le but principal de la construction d'un modèle de prédiction n'est pas de s'ajuster le mieux possible aux exemples connus, mais de prédire la grandeur souhaitée pour des exemples n'ayant pas participé à l'apprentissage. Pour cela on aimerait calculer l'erreur théorique de généralisation qui est l'erreur quadratique moyenne sur l'ensemble de tous les exemples possibles. Cette quantité n'est pas calculable car l'ensemble des exemples est le plus souvent infini, cependant pour un ensemble d'exemples n'ayant pas participé à l'apprentissage on peut calculer la quantité suivante :

$$E_V = \sqrt{\frac{1}{N_V} \sum_{k=1}^{N_V} (r_V^k)^2} \quad E_V$$

où N_V est le nombre d'exemples n'ayant pas servi à l'apprentissage
 r_V^k est l'erreur commise pour la prédiction de l'exemple k

En augmentant le nombre de neurones cachés on augmente le nombre de poids à ajuster et cela permet de s'ajuster plus finement aux données d'apprentissage. Cependant, on court alors le risque que le réseau s'ajuste au bruit dont les exemples d'apprentissage sont fatalement entachés. Un tel réseau présentera des différences importantes entre E_V et E_A . A l'inverse, si le nombre de neurones cachés est faible, E_V et E_A seront proches mais E_A sera élevé. Ce problème est appelé le dilemme « biais-variance » : il faut trouver le meilleur compromis possible entre complexité du modèle et performance de généralisation.

2.2.9.1 Elimination des modèles surajustés

On définit la matrice Jacobienne $Z \in \mathbb{R}^{N \times q}$ par $Z = \left(\frac{\partial g(\mathbf{x}^k, \mathbf{w})}{\partial w_i} \right)_{\substack{1 \leq k \leq N \\ 1 \leq i \leq q}}$. Chaque colonne z_i de Z

correspond donc au vecteur des dérivées partielles de la sortie du réseau par rapport au poids w_i pour tous les exemples \mathbf{x}^k .¹ Si Z n'est pas de rang plein (si $rg(Z) < q$ avec $q \leq N$) cela signifie que les variations de la sortie du réseau par rapport à certains poids peuvent s'exprimer comme une combinaison linéaire des variations par rapport aux poids restants. Le modèle considéré possède certainement trop de poids à ajuster, il y a donc un risque de surajustement.

Cela fournit un premier critère pour choisir entre plusieurs modèles. On éliminera systématiquement les modèles qui, après apprentissage, n'ont pas leur matrice Jacobienne de rang plein.

2.2.9.2 Principe du Leave-one-out virtuel

La méthode du Leave-one-out est une technique destinée à estimer l'erreur théorique de généralisation impossible à calculer. On dispose de N exemples pour lesquelles la grandeur à estimer est connue. On réalise alors N apprentissages avec $N-1$ exemples pour chacun d'entre eux (pour l'apprentissage k , l'exemple k n'a pas été utilisé). On note alors $r_k^{(-k)}$ l'erreur de prédiction sur l'exemple k dans le modèle où il n'a pas été utilisé.

Le score du Leave-one-out est alors donné par : $E_L = \sqrt{\frac{1}{N} \sum_{k=1}^N (r_k^{(-k)})^2}$

On choisit en principe de retenir le modèle pour lequel le score de Leave-one-out est le plus faible. L'inconvénient de cette technique est qu'elle est très lourde en temps de calcul puisque pour un ensemble de N exemples on doit réaliser N apprentissages différents.

¹ Il est facile de calculer Z car l'on sait calculer que le gradient de la fonction de coût J est :

$$\frac{\partial J^k}{\partial w_i} = -2(y^k - g(\mathbf{x}^k, \mathbf{w})) \frac{\partial g(\mathbf{x}^k, \mathbf{w})}{\partial w_i} = -2r^k \times z_{i,k}$$

L'idée du Leave-one-out virtuel est de réaliser un seul apprentissage avec tous les exemples disponibles puis d'estimer pour chacun d'entre eux, leur influence sur le modèle obtenu (ie. l'erreur de prédiction $r_k^{(-k)}$ que l'on obtiendrait pour l'exemple k si celui-ci n'avait pas participé à l'apprentissage).

Il est montré dans le chapitre 3 de [Monari 1999] que si le retrait d'un exemple n'a qu'un faible impact sur le modèle alors on peut estimer $r_k^{(-k)}$ par la formule :

$$r_k^{(-k)} = \frac{r_k}{1 - h_{kk}} \quad \text{où} \quad \left\{ \begin{array}{l} h_{kk} = (z^k)^T (Z^T Z)^{-1} z^k \quad (\text{appelé levier de l'exemple } k) \\ Z \text{ matrice Jacobienne} \\ z^k \text{ } k^{\text{ième}} \text{ colonne de } Z \end{array} \right.$$

On peut alors estimer l'erreur de généralisation par le score de Leave-one-out virtuel :

$$E_p = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(\frac{r_k}{1 - h_{kk}} \right)^2} \quad E_p$$

Chapitre 3 : Recherche de paramètres influençant l'efficacité

Comme annoncé dans l'introduction, on ne cherche pas à construire un modèle statistique unique qui serait valable pour tous les types de placements. En effet, les variations d'efficacité sont très différentes suivant les domaines considérés (chemises, lingerie, ameublement...) Dans cette optique, nous nous sommes intéressés dans cette étude à essentiellement deux types de placements :

- Les placements de chemises
- Les placements de pantalons

Cependant, malgré les différences, certains paramètres ont toujours une influence comparable sur l'efficacité des placements. On les appellera les paramètres globaux. Ils sont décrits dans la première partie de ce chapitre. A l'inverse, dans la deuxième partie, on recherche des paramètres qui influencent spécifiquement l'efficacité des placements de chemises puis des placements de pantalons.

3.1 Paramètres globaux

3.1.1 Nombre de tailles à placer

C'est le critère retenu actuellement dans Optiplan pour déterminer des tables d'efficacités pour les nouveaux placements (cf. 1.2 : Traitement actuel du problème)

L'expérience montre généralement que plus un placement contient de tailles (et donc de pièces) à placer plus il est efficace.

Les deux graphiques de la **Figure 64** donnent une première indication sur l'influence du nombre de tailles sur l'efficacité des placements. Le premier correspond à des placements de trois modèles de pantalons, le second à des placements de deux modèles de chemises. Pour chacun des cinq modèles considérés, on dispose de nombreux placements contenant de 1 à 6

tailles. Les courbes représentent l'efficacité moyenne de ces placements en fonction du nombre de tailles qu'ils contiennent.

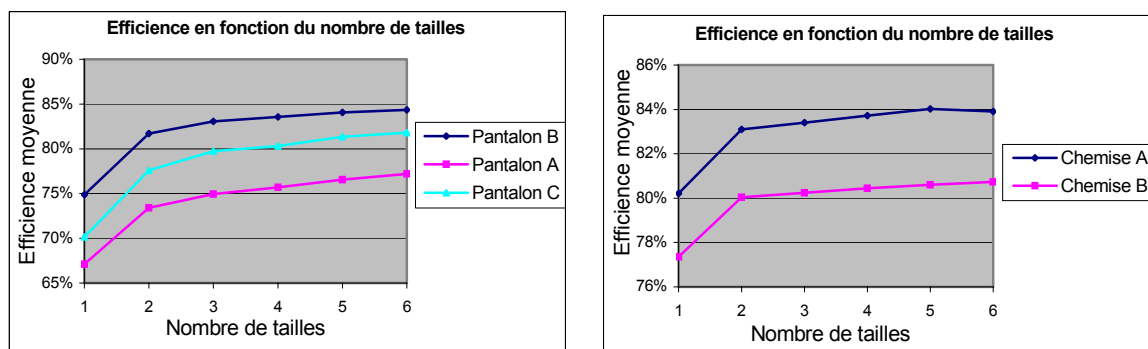


Figure 64 : Efficacité moyenne de différents modèles en fonction du nombre de tailles

L'échelle des deux graphiques n'est pas la même, les variations d'efficacité ont une plus grande amplitude dans le cas des pantalons que dans le cas des chemises. On observe également des écarts entre modèles de même type de vêtement. Pourtant, malgré ces différences, l'influence du nombre de tailles sur l'efficacité des placements semble comparable dans tous les cas car les courbes ont une allure similaire. En effet, pour les cinq modèles considérés, la règle empirique (qui considère que plus un placement contient de tailles, plus il est efficace) n'est contredite qu'une seule fois.¹ Cependant, cette influence ne semble pas linéaire quel que soit le modèle. On note une grande augmentation de l'efficacité lorsque l'on passe de placements monotailles à des placements « 2 tailles » en revanche, l'augmentation est beaucoup moins rapide pour les placements contenant plus de 2 tailles. De plus, les courbes de ces graphiques sont basées sur les moyennes des efficacités de nombreux placements, les variations d'efficacité entre des placements contenant le même nombre de tailles peuvent être de natures très différentes. Pour illustrer cela, le tableau de la **Figure 65** présente (dans le cas des pantalons uniquement) les écarts types des efficacités à l'intérieur d'une même catégorie (même modèle, même nombre de taille).

¹ Pour le modèle de Chemise A, on passe d'une efficacité moyenne de 84,02% pour les placements « 5 tailles » à une efficacité de 83,91% pour les placements « 6 tailles ».

Modèle \ Tailles	Tailles		
	1	2	3
B	6,89	2,13	1,16
A	4,55	2,24	1,52
C	6,16	2,87	1,84

Figure 65: Ecart Type (en % d'efficacité) des résultats en fonction du nombre de tailles et des modèles

Pour chacun des modèles, l'écart type est beaucoup plus important pour les placements monotailles. En effet, il est important de noter que les placements monotailles, ne sont pas nécessairement « mauvais », certains peuvent être plus efficaces que des placements 2-tailles ou 3-tailles. Ils sont moins bons en moyenne car pour certains d'entre eux en revanche, l'efficacité est « catastrophique ».¹

Pour expliquer ce phénomène, on peut avancer l'argument suivant :

Dans un placement mono-taille, le nombre de pièces à placer est généralement faible. Si ses quelques pièces peuvent s'agencer correctement dans la laize du tissu, on obtient de très bons résultats (ou à l'inverse de très mauvais résultats). Quand le nombre de pièces à placer augmente (placements 2-tailles, 3-tailles ou plus) les meilleurs placements ne sont que très légèrement plus efficaces que les meilleurs mono-tailles par contre le nombre de pièces important permet d'atténuer les mauvais cas de figure et d'éviter presque à coup sûr les très mauvaises efficacités.

Ces quelques remarques permettent d'affirmer que le nombre de tailles contenues dans un placement n'est pas suffisant pour expliquer toutes les variations d'efficacité observées. Cependant, ce paramètre reste un critère important dans la recherche d'un modèle statistique de prédiction d'efficacité.

¹ Pour le modèle de pantalon B, le meilleur placement mono-taille a une efficacité de 84,91% contre 85,28% pour le meilleur placement 2 et 3 tailles confondus. En revanche, le plus mauvais mono-taille a une efficacité de 62,32% contre 74,77% pour le plus mauvais 2-3 tailles.

3.1.2 Laize du placement

La laize du placement est la largeur disponible pour placer les pièces. Elle correspond à la largeur du tissu dans lequel ces pièces seront découpées. Comme le montre la **Figure 59**, la laize a une grande influence sur l'efficacité des placements car elle conditionne directement les combinaisons qui vont être possibles ou non pour disposer les pièces. Ces variations très brutales de l'efficacité (qui surviennent lorsque la laize augmente et qu'une disposition efficace des pièces devient possible), sont très liées au type de placement que l'on réalise (chemises, pantalons, etc.) on verra au chapitre suivant (cf. Paramètres spécifiques) comment tenter de les prendre en compte.

On s'intéresse dans ce paragraphe à une (éventuelle) influence plus globale de la laize sur l'efficacité des placements. Les deux graphiques de la **Figure 66** représentent l'efficacité moyenne des placements en fonction de différentes laizes, le premier correspond aux 3 modèles de pantalons et le second aux 2 modèles de chemise étudiés.

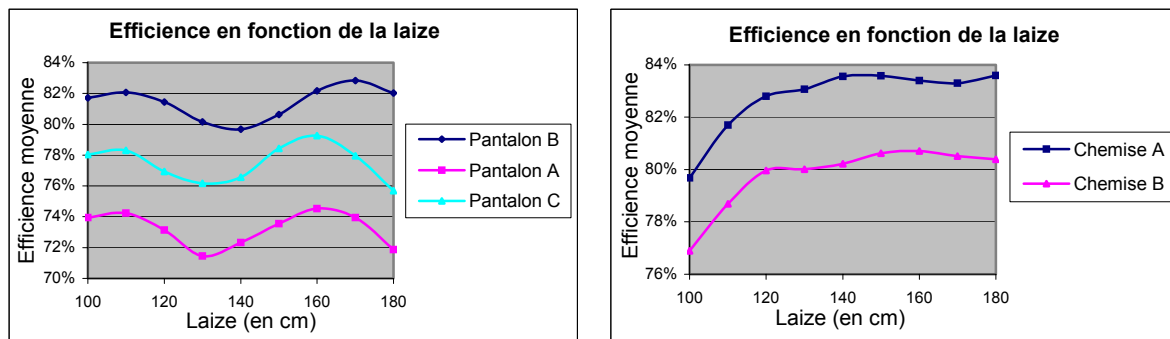


Figure 66 : Efficacité moyenne des placements en fonction de la laize

Contrairement aux graphiques correspondant au nombre de tailles (cf. **Figure 64**), l'influence de la laize semble de nature différente entre les modèles de pantalons et les modèles de chemises. Pour les modèles de pantalons, on observe en moyenne une bonne efficacité pour les petites laizes (100 – 120 cm) ainsi que pour les grandes laizes (160 – 180 cm). Pour les trois modèles étudiés, l'efficacité est moins bonne en moyenne pour les laizes intermédiaires (130 – 140 cm). Ce phénomène est spécifique aux placements de pantalons. En effet, pour ces derniers, la disposition des jambes de pantalons détermine en grande partie la qualité du

placement. Or, comme celles-ci sont en nombre pair (dans le cas de plis simples), il est intéressant de pouvoir les « empiler » dans la laize en nombre pair. Les bonnes efficacités correspondent le plus souvent à l'empilement de deux jambes pour les petites laizes et à quatre jambes pour les grandes laizes. Les mauvaises efficacités sont obtenues pour les laizes intermédiaires lorsqu'on ne peut empiler que trois jambes. Nous reviendrons en détail sur ce point dans le chapitre suivant (cf.3.2.2 Placements de pantalons).

Pour les modèles de chemises, on obtient une mauvaise efficacité en moyenne pour les petites laizes (100 – 110 cm). Pour les laizes supérieures, on n'observe pas réellement de tendance marquée. Pour expliquer cela, il faut remarquer que les modèles de chemises sont notamment composés de plusieurs grandes pièces (le dos, les deux devants). Pour placer celles-ci, on dispose de moins de liberté lorsque la laize disponible est faible et cela conduit à de moins bonnes efficacités en moyenne.

Comme on le voit, la laize a une influence sur l'efficacité des placements mais celle-ci est difficile à cerner d'un point de vue global principalement pour deux raisons :

- Au cas par cas, brusque variations d'efficacité pour de faible augmentation de laize
- Au niveau global, influence différente selon les types de placements considérés (allure des courbes différentes pour les placements de chemises et de pantalons)

Pour ces raisons, on tentera de trouver des paramètres spécifiques (cf. 3.2 Paramètres spécifiques) à chaque type de placement pour rendre compte avec plus de précisions de l'influence de la laize sur l'efficacité des placements.

3.1.3 Surface par taille à placer

La gradation est le terme qui désigne la déclinaison d'un modèle de vêtement en plusieurs tailles. Une chemise par exemple peut exister dans les tailles S, M, L et XL. La surface de tissu nécessaire pour fabriquer taille XL est plus importante que celle nécessaire pour une taille S. Dans un placement multi-tailles, on peut avoir plusieurs « grandes » tailles, plusieurs « petites » tailles ou une combinaison de petites et des grandes tailles. On s'intéresse dans ce

paragraphe à l'éventuelle influence de cette notion de « petites » ou « grandes » tailles sur l'efficacité des placements.

Une première idée pour l'approche de ce problème est de comparer la surface des pièces à placer. En effet, pour un placement mono-taille, les grandes tailles sont bien sûr celles dont la surface des pièces est la plus grande. Evidemment, la surface totale des pièces d'un placement dépend du nombre de tailles qu'il contient. Pour cela, on calcule la surface moyenne des pièces par taille placée en faisant le rapport entre la surface totale des pièces et le nombre de

$$\text{tailles} : S_{/taille}(P) = \frac{\text{Surface}(P)}{\text{NbTailles}(P)}$$

Les graphiques de la **Figure 67** représentent l'efficacité moyenne des placements en fonction de la surface par taille placée. Pour plus de lisibilité, on considère uniquement des placements 2-tailles et on regroupe les placements en quantiles pour lesquels on calcule la surface par taille placée moyenne et l'efficacité moyenne correspondante.

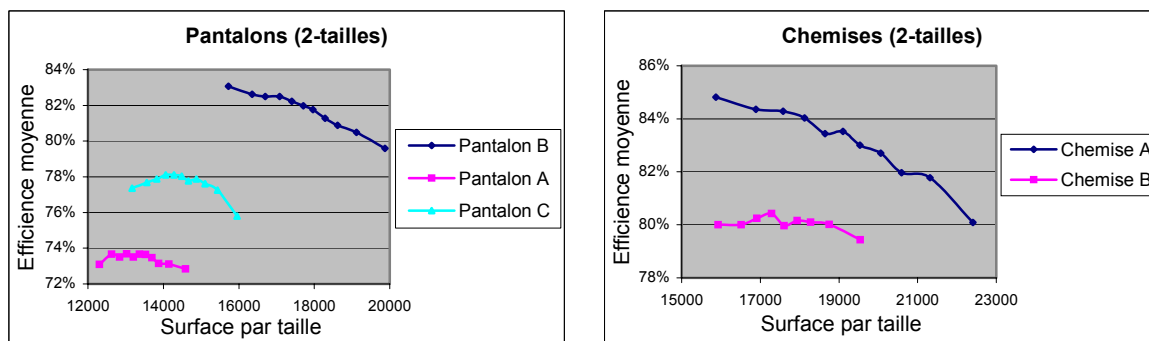


Figure 67 : Efficacité moyenne de placements 2-tailles en fonction de la surface par taille

On observe des comportements assez différents selon les cas. Pour le modèle de pantalon B et le modèle de chemise A, l'efficacité moyenne décroît au fur et à mesure que la surface par taille augmente (autrement dit, mélanger deux grandes tailles ne donne pas de bons résultats).

Pour le modèle de pantalon A et le modèle de chemise B, on n'observe pas de tendance marquée, on remarque cependant que la plus faible efficacité moyenne est obtenue pour les valeurs les plus grandes de surfaces par tailles. C'est la même chose pour le modèle de

pantalon C avec une dégradation de l'efficacité plus marquée pour les plus grandes surfaces par taille.

L'analyse de ces courbes, ne permet pas d'avoir une vision totalement claire sur l'influence de la surface par taille sur l'efficacité des placements. En effet, l'influence ne semble pas être de même nature à l'intérieur même des catégories de placements (allure des courbes différentes pour les modèles de pantalons par exemple). La seule caractéristique commune à toutes les courbes est la mauvaise efficacité obtenue en moyenne lorsque l'on mélange les plus grandes tailles entre elles.

3.1.4 Indice « d'éloignement » des tailles placées

On a vu dans le paragraphe précédent, qu'il était difficile de décrire une influence claire de la surface par taille sur l'efficacité des placements. Cependant, comme le montre la **Figure 68**, deux placements ayant la même surface par taille (ou presque égale) peuvent correspondre à des configurations très différentes :

	36	38	40	42	44	46	48
Placement 1				2			
Placement 2	1						1

Figure 68 : Deux placements différents ayant la même surface par taille

- Le placement 1 contient 2 fois la taille 42
- Le placement 2 contient 1 fois la taille 36 et une fois la taille 48

La surface des pièces de ces deux placements est quasiment identique (l'augmentation de surface d'une taille à l'autre est le plus souvent constante ou quasiment constante). La question est de savoir laquelle de ces 2 configurations fournira la meilleure efficacité en moyenne. Pour cela, on définit pour chaque placement une notion « d'éloignement » des tailles qu'il contient. L'indice créé est compris entre 0 et 1, il vaut 0 lorsque le placement ne contient qu'une taille (éventuellement répétée plusieurs fois) du vêtement. L'indice vaut 1

lorsque les tailles à placer sont pour moitié les plus petites tailles du vêtement et pour moitié les plus grandes tailles du vêtement.¹

Les 2 graphiques de la **Figure 69** représentent l'efficacité moyenne de placements 2-tailles en fonction de l'indice d'éloignement des tailles.

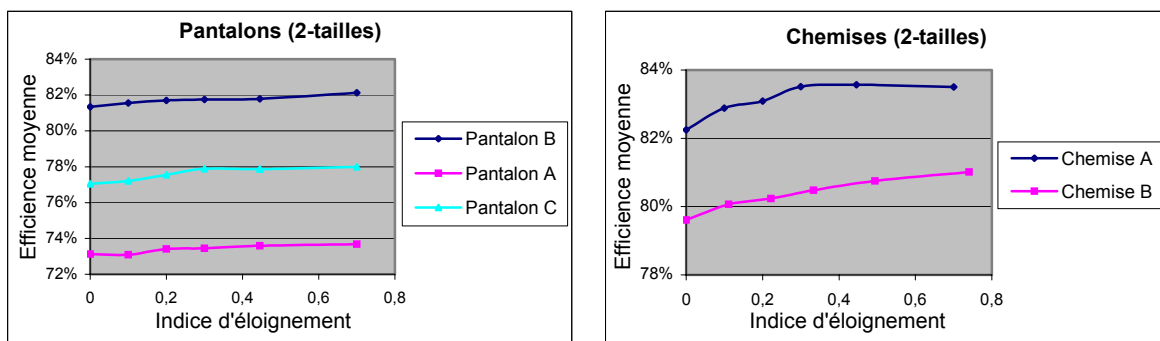


Figure 69 : Efficacité moyenne de placements 2-tailles en fonction de l'indice d'éloignement

Pour tous les modèles considérés, on observe une augmentation de l'efficacité lorsque l'indice d'éloignement augmente. Cette augmentation est de l'ordre de 1% d'efficacité entre les plus basses et les plus hautes valeurs de l'indice.

3.1.5 Indice de « rectangularité »

La problématique du placement consiste à disposer de la meilleure façon possible des pièces de tissu sur une surface rectangulaire², on peut légitimement penser que plus les pièces que l'on a à placer sont « proches » d'un rectangle plus on a de chances d'obtenir une bonne efficacité.

L'idée est de définir pour chaque pièce un « rectangle englobant minimum » qui est en fait le rectangle de plus faible surface pouvant contenir la pièce en question. Pour les placements considérés, les rotations de pièces sont interdites on impose donc que les cotés du rectangle

¹ Lorsque le nombre de tailles est impair, le fait de « bouger la taille du milieu » ne modifie pas l'indice d'éloignement du placement. L'**annexe 3** présente en détail cette notion d'éloignement des tailles placées.

² Il existe des domaines où le support n'est pas rectangulaire mais quelconque. C'est notamment le cas dans l'industrie du cuir où l'on travaille avec des peaux. Cependant ce n'est pas l'objet de notre étude ici.

minimal soient horizontaux et verticaux. On obtient donc ce rectangle minimal en trouvant la plus grande longueur et la plus grande largeur de la pièce à placer (cf. **Figure 70**).

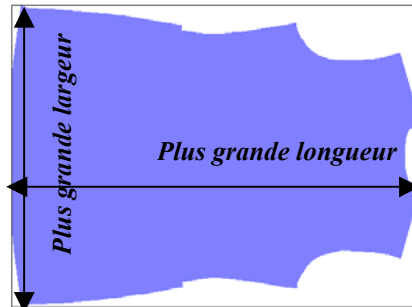


Figure 70 : Rectangle englobant minimum pour un dos de chemise

On définit alors l'indice de rectangularité d'une pièce comme le rapport entre la surface de la pièce et la surface de son rectangle englobant :

$$I_{Rect}(pièce) = \frac{Surface(pièce)}{Long_{max}(pièce) \times Larg_{max}(pièce)}$$

Plus cet indice est proche de 1 et plus la pièce en question est « proche » d'un rectangle. Pour élargir cette notion à l'échelle du placement, on définit pour un placement P donné la quantité suivante :

$$\begin{aligned} I_{Rect}(P) &= \frac{\sum_{pièce \in P} Surface(pièce)}{\sum_{pièce \in P} Long_{max}(pièce) \times Larg_{max}(pièce)} \\ &= \frac{Surface(P)}{\sum_{pièce \in P} Long_{max}(pièce) \times Larg_{max}(pièce)} \end{aligned}$$

Cette quantité correspond en fait à la moyenne des indices de rectangularité de chaque pièce du placement, pondérée par la surface du rectangle englobant correspondant. L'idée de cette pondération est que plus une pièce est grande (son rectangle englobant l'est également), plus la forme de cette pièce a une influence sur le placement. Le tableau suivant (cf. **Figure 71**) présente les indices de rectangularité moyens et les efficacités moyennes correspondantes pour chacun des modèles de vêtements :

	Chemises		Pantalons		
modèle	A	B	A	C	B
indice moyen de "rectangularité"	86,50%	75,81%	71,93	72,93	77,35
efficacité moyenne	82,74%	79,78%	73,22	77,49	81,42

Figure 71: Efficacité et Indice de rectangularité

L'indice de rectangularité semble avoir une réelle influence sur l'efficacité des placements en effet, plus cet indice est élevé et plus l'efficacité moyenne des placements est élevée. De plus, l'utilisation de cet indice permet de différencier des modèles de vêtement de même type (différences entre deux modèles de chemises par exemple). Des paramètres tels que le nombre de tailles où la laize ne permettent pas de faire ces distinctions.

3.1.6 Nombre de « petites/grandes pièces »

On sait que dans la plupart des cas, la disposition des plus grandes pièces du modèle détermine en grande partie la longueur du placement car les petites pièces sont placées autant que possible dans les espaces laissés libres une fois les grandes pièces placées.

L'idée est alors que le nombre de petites pièces dans un placement (par rapport au nombre de grandes pièces) peut influencer l'efficacité. On imagine intuitivement que plus le nombre de petites pièces est important plus il est possible de remplir les espaces libres et donc d'améliorer l'efficacité.

La première tâche est donc de déterminer les pièces considérées comme petites et celles considérées comme grandes. Pour cela, on classe les pièces d'un placement par ordre décroissant de superficie. On calcule alors pour toutes les pièces (sauf la première) le rapport entre la superficie de la pièce et celle de la pièce qui la précède (rapport ≤ 1). La pièce pour laquelle ce rapport est le plus faible sera considérée comme petite ainsi que toutes les pièces qui la suivent. Dans l'exemple de la **Figure 72**, les cinq premières pièces sont considérées comme grandes et toutes les autres comme des petites.

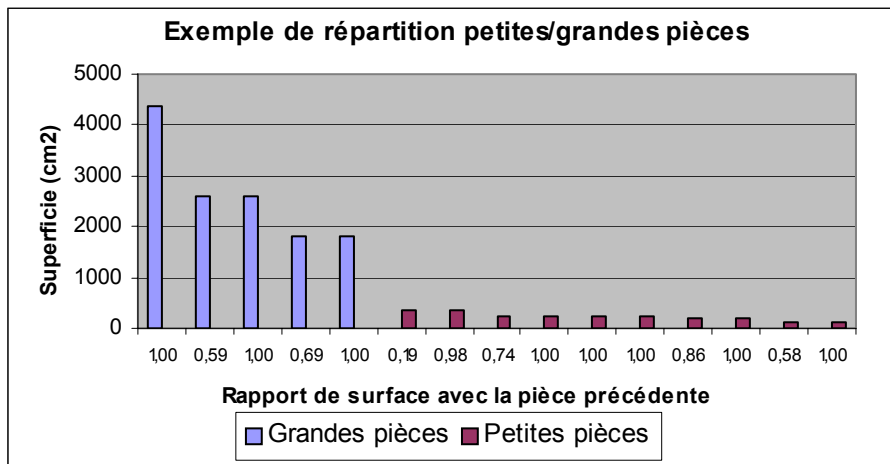


Figure 72 : Illustration de la détermination du nombre de petites pièces (modèle de chemise)

Pour tenter de mesurer l'influence de la proportion de petites pièces sur l'efficacité on utilise le modèle de pantalon B (75% de ces pièces sont petites) que l'on modifie en supprimant quelques-unes de ces petites pièces (le modèle B modifié contient 68.75% de petites pièces). On compare ensuite les efficacités obtenues dans les deux cas. Sur un échantillon de 90 placements monotailles, l'efficacité moyenne pour le modèle B est de 74,87% alors qu'elle passe à 72,85% pour le modèle B modifié (avec moins de petites pièces)¹

Une variante de ce paramètre est de considérer le rapport entre la superficie cumulée des petites pièces et celle des grandes pièces (au lieu du rapport entre les cardinaux). On a ainsi une évaluation plus quantitative de la proportion petite/grande pièce.

¹ Il faut noter que la suppression de ces petites pièces n'a pas fait diminuer l'indice de rectangularité ce qui aurait également pu expliquer une diminution de l'efficacité.

3.2 Paramètres spécifiques

3.2.1 Placements de chemises

3.2.1.1 Variations d'efficacité en fonction de la laize

On sait que la laize a une très grande influence sur l'efficacité des placements. Cependant, cette influence est très difficile à appréhender car de très petites augmentations de la laize peuvent provoquer de brusques améliorations d'efficacité. Les deux graphiques de la **Figure 73** illustrent ce phénomène. Ils correspondent à l'efficacité de placements mono-taille¹ pour les modèles de chemise A et B en fonction de la laize.

¹ On observe le même phénomène pour les placements multi-tailles bien que l'amplitude des sauts d'efficacité soit moins importante.

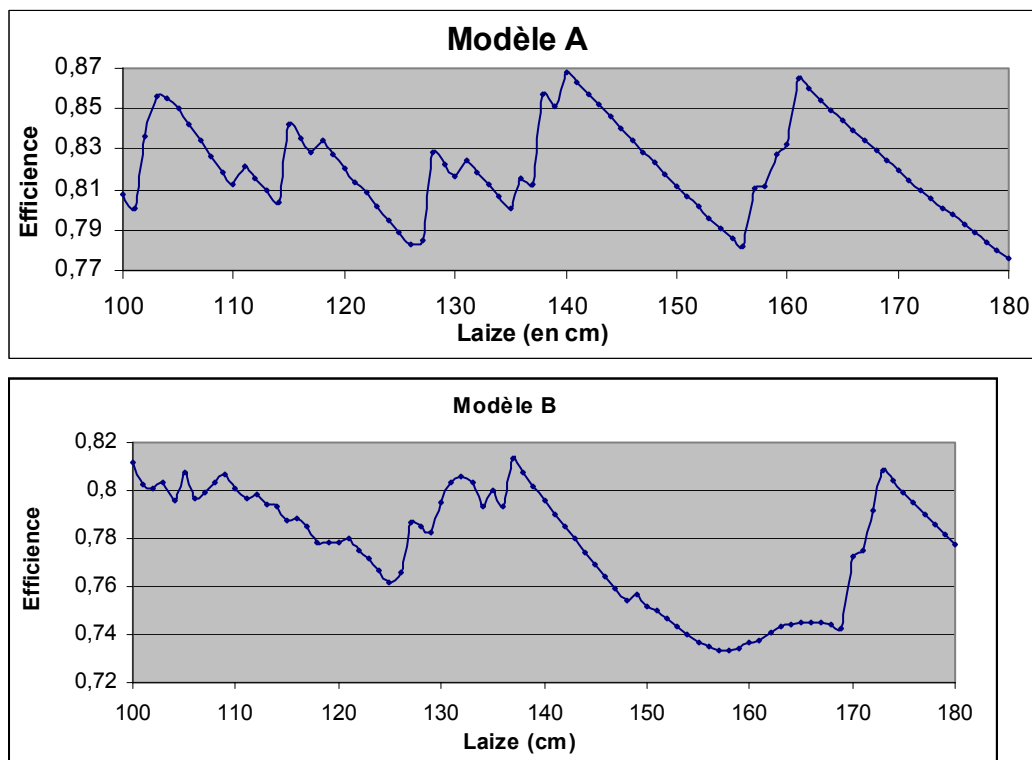


Figure 73 : Evolution de l'efficacité en fonction de la laize pour 2 placements de chemise mono-taille

Pour le modèle A, l'efficacité varie de moins de 78% à presque 87%, on observe des sauts d'efficacité importants en voici quelques exemples :

- 127-128 cm 78,45% - 82,89% + 4,44%
- 137-138 cm 81,23% - 85,75% + 4,52%
- 156-161 cm 78,16% - 86,49% + 8,33%

Pour le modèle B, les variations d'efficacité sont de moindre amplitude, l'efficacité varie de 73% à 82%, et l'aspect en « dents de scie » est moins évident. Cependant, on observe également des sauts d'efficacité :

- 126-127 cm 76,55% - 78,7% + 2,15%
- 136-137 cm 79,36% - 81,3% + 1,94%
- 169-173 cm 74,29% - 80,85% + 6,56%

On observe également qu'après la plupart des brusques augmentations d'efficacité, celle-ci décroît ensuite de façon quasi-linéaire.

Ces sauts d'efficacité sont directement liés au positionnement des pièces dans la laize. Si pour une laize donnée, on obtient une très bonne efficacité grâce à une bonne disposition des pièces, une faible augmentation de la laize ne permettra pas de changer (de façon importante) cette disposition par contre, l'augmentation de la surface du placement provoquée par l'augmentation de la laize induit une baisse de l'efficacité. Ces cas de figure correspondent aux parties quasi-linéaires des courbes. L'efficacité décroît avec l'augmentation de la laize jusqu'à ce qu'une autre bonne disposition des pièces soit possible, c'est là que l'on observe un saut d'efficacité.

On voit tout de suite que entre deux sauts, il est relativement facile de prédire l'efficacité des placements, la difficulté est donc par conséquent de connaître les laizes où ces sauts vont se produire.

3.2.1.2 Les pièces importantes

La **Figure 74** présente la liste des pièces pour les deux modèles de chemise. Dans les deux cas, les plus grandes pièces comme elles sont définies dans le paragraphe 3.1.6 sont :

- Le dos
- Les 2 devants
- Les 2 manches

Pour les deux modèles, ces 5 pièces représentent plus de 80% de la surface totale à placer. Leur positionnement détermine en grande partie la longueur (et donc l'efficacité) du placement.



Figure 74 : Liste des pièces pour les deux modèles de chemise étudiés

On note que l'indice de rectangularité (cf. 3.1.5 Indice de « rectangularité ») des pièces du modèle A est supérieur à celui des pièces du modèle B (0,865 contre 0,758). Cela donne une piste pour expliquer que les sauts d'efficacité sont moins importants pour le modèle B que pour le modèle A. En effet, si un indice de rectangularité élevé permet souvent d'obtenir de bonnes efficacités, il réduit en revanche considérablement la notion d'imbrication entre les pièces. Dans le cas particulier du modèle B, on voit clairement qu'on aura tout intérêt à placer côte à côte le dos et les manches par exemple afin de compenser les pentes respectives du bord de ces pièces.

Dans le but de mieux repérer les laizes pour lesquelles on observe de sauts d'efficacité, on s'intéresse dans le paragraphe suivant aux configurations de ces 5 pièces qui permettent d'obtenir une bonne efficacité.

3.2.1.3 Les bonnes configurations

On étudie en fonction de la laize du placement, les dispositions possibles des 5 grandes pièces des modèles de chemise. En particulier, on s'intéresse aux empilements verticaux éventuellement possible dans la hauteur de laize. Les 6 cas de figure suivants sont classés par ordre de laize croissant.

3.2.1.3.1 Un dos dans la laize

Il est évidemment obligatoire que la laize soit supérieure à la largeur du dos de la chemise pour que le placement soit possible. On s'intéresse aux laizes pour lesquelles cette condition est respectée.¹

3.2.1.3.2 Deux devants dans la laize

Dans le cas des deux plus grandes tailles du modèle A, on ne peut pas pour les plus petites laizes disposer les 2 devants l'un par-dessus l'autre. Le tableau de la **Figure 75** montre les sauts d'efficiences correspondants.

Dans les deux cas, on remarque un saut d'efficacité très important dès que les deux devants peuvent être placés l'un par-dessus l'autre.

Taille	58		60	
Largeur de deux devants (cm)	101,8		105,8	
Laizes observées	101	102	105	106
Efficiences	67,84%	80,24%	67,45%	79,73%
Saut		12,40%		12,28%

Figure 75 : Sauts d'efficacité lorsque les deux devants peuvent être placés l'un par-dessus l'autre

¹ On étudie l'efficacité des placements monotailles pour des laizes comprises entre 100 et 180 cm avec une mesure tous les cm.

La **Figure 76** représente les placements de la taille 58 du modèle de chemise A en laize 101cm puis en laize 102cm. Le fait de pouvoir superposer les 2 devants dans la laize permet de réduire la longueur de tissu nécessaire et donc d'améliorer l'efficacité.

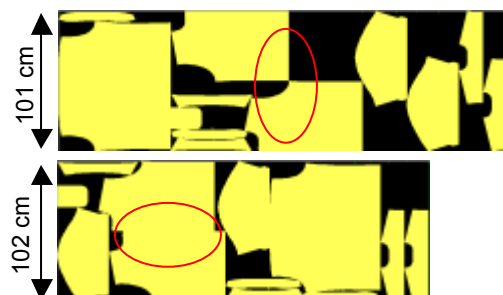


Figure 76 : Positionnement des 2 devants selon la laize disponible

3.2.1.3.3 Un devant et une manche dans la laize

Le tableau de la **Figure 77** présente les efficacités des mono-tailles réalisés avec le modèle A pour certaines laizes. Pour chaque taille, la première valeur de la laize est inférieure à la largeur cumulée d'un devant et d'une manche, la deuxième valeur de la laize est supérieure. Dans tous les cas, on observe un saut d'efficacité.

Taille	50		52		54		56		58		60	
Largeur d'un devant et d'une manche (cm)	103,6		107,6		111,5		115,4		119,4		123,3	
Laizes observées	100	104	104	108	108	112	112	116	116	120	120	123
Efficacités	74,97%	84,54%	75,05%	79,82%	74,62%	79,56%	75,25%	79,33%	75,39%	79,06%	75,55%	79,30%
Saut		9,57%		4,77%		4,94%		4,08%		3,67%		3,75%

Figure 77 : Sauts d'efficacité lorsqu'un devant et une manche peuvent être placés l'un par-dessus l'autre

La **Figure 78** représente les placements de la taille 50 du modèle de chemise A en laize 100cm puis en laize 104cm. Le fait de pouvoir superposer une manche et un devant dans la laize permet également de réduire la longueur de tissu nécessaire et donc d'améliorer l'efficacité.

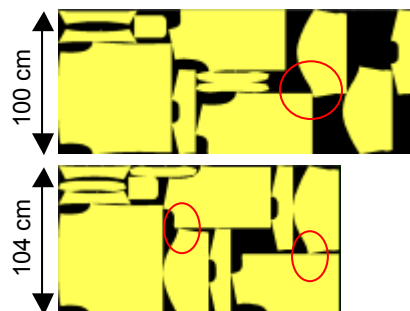


Figure 78 : Positionnement d'une manche et d'un devant selon la laize disponible

Dans le cas du modèle B, les pièces sont un peu moins larges et l'on peut toujours placer un devant et une manche pour toutes les laizes étudiées.

3.2.1.3.4 Un dos et un devant dans la laize

Le tableau de la **Figure 79** présente les efficacités des mono-tailles réalisés avec le modèle A pour certaines laizes. Pour chaque taille, la première valeur de la laize est inférieure à la largeur cumulée du dos et d'un devant, la deuxième valeur de la laize est supérieure. Pour les deux premières tailles, il n'y a pas de différence significative, pour toutes les autres, on observe un saut d'efficacité.

Taille	44		46		48		50		52		54		56		58		60	
Largeur d'un dos et d'un devant (cm)	102,7		108,7		114,7		120,7		126,7		132,7		138,7		144,8		150,8	
Laizes observées	101	103	107	109	113	115	119	121	125	127	131	133	137	139	143	145	149	151
Efficacités (en%)	81,25	81,39	80,09	80,05	79,1	85,86	78,09	83,91	81,77	84,16	81,19	83,48	80,22	82,53	79,33	83,3	78,49	82,55
Saut (en %)		0,14		-0,04		6,76		5,82		2,39		2,29		2,31		3,97		4,06

Figure 79 : Sauts d'efficacité lorsque le dos et un devant peuvent être placés l'un par-dessus l'autre

La **Figure 80** représente les placements de la taille 48 du modèle de chemise A en laize 113cm puis en laize 115cm. Là encore, le gain d'efficacité est obtenu lorsqu'il est possible de superposer le dos et un devant dans la laize.

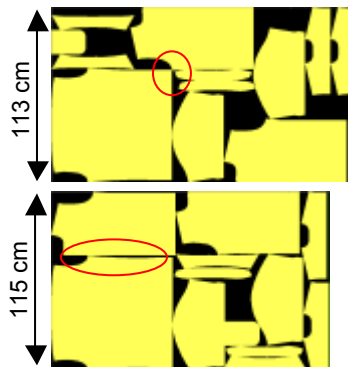


Figure 80 : Positionnement du dos et d'un devant selon la laize disponible

On cherche à réaliser le même genre de tableau pour le modèle B (cf. **Figure 81**). On trouve en effet des laizes pour lesquelles il y a un saut d'efficacité. On remarque cependant que les laizes qui permettent une bonne efficacité sont dans tous les cas inférieures à la largeur cumulée du dos et d'un devant (environ 5 cm).

Taille	40		42		44		46		48		50		52	
Largeur d'un dos et d'un devant (cm)	107,7		110,6		113,6		116,4		120,9		125,4		129,9	
Laizes observées	101	102	104	105	107	108	110	111	114	116	119	120	123	125
Efficacités (en%)	74,93	77,96	74,76	78	74,6	78,34	74,48	78,33	74,42	79,85	74,01	79,18	74,92	79,07
Saut (en %)		3,03		3,24		3,77		3,85		5,43		5,17		4,15
Imbrication (en cm)		5,7		5,6		5,6		5,4		4,9		5,4		4,9

Figure 81 : Sauts d'efficacité lorsqu'on peut imbriquer le dos et un devant

En fait, comme le montre la **Figure 82**, de part leur forme respective, le dos et le devant ont pu s'imbriquer. Le saut d'efficacité survient pour une laize plus faible que ce que l'on attendait au départ.

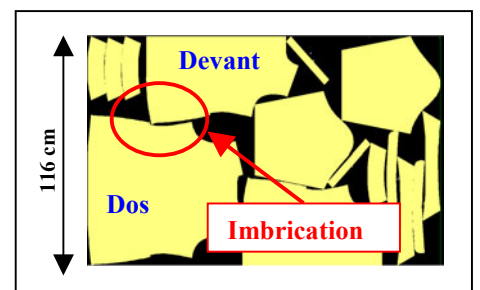


Figure 82 : Exemple d'imbrication permettant une bonne efficacité

3.2.1.3.5 Un dos et deux devants dans la laize

Le tableau de la **Figure 83** met en évidence les sauts d'efficacité observés pour le modèle A lorsqu'un dos et deux devants peuvent s'imbriquer dans la laize. On remarque un saut d'efficacité pour toutes les tailles avec toutefois une augmentation du phénomène d'imbrication pour les grandes tailles. En effet, les règles de gradation peuvent faire légèrement évoluer la forme de certaines pièces pour les différentes tailles d'un vêtement. Les imbrications possibles peuvent donc varier en fonction de la taille considérée.

Taille	42		44		46		48		50		52		54		56	
Largeur d'un dos et de 2 devants (cm)	131,6		139,6		148		155,6		163,6		171,6		179,6		187,6	
Laizes observées	130	131	138	139	145	147	152	154	159	160	163	166	171	172	177	180
Efficacités (en%)	79,04	82,47	77,4	82,45	77,35	81,75	77,19	84,5	78,97	84,56	78,91	83,57	78,26	82,25	79,05	82,45
Saut (en %)		3,43		5,05		4,4		7,31		5,59		4,66		3,99		3,4
Imbrication (en cm)		0,6		0,6		1		1,6		3,6		5,6		7,6		7,6

Figure 83 : Sauts d'efficacité pour l'imbrication d'un dos et de 2 devants pour le modèle A

Le tableau de la **Figure 84** montre que dans le cas du modèle B, on observe également des sauts d'efficacité avec des imbrications importantes pour toutes les tailles.

Taille	36		38		40		42		44		46		48		50		52	
Largeur d'un dos et de 2 devants (cm)	140,3		144,2		148,2		152,1		156,1		159,8		165,8		171,8		177,8	
Laizes observées	126	130	130	132	133	139	137	143	141	147	146	151	152	157	158	162	163	167
Efficacités (en%)	76,6	79,5	76,2	80,2	76,4	81,4	75,4	81,8	76,2	81,8	77,9	81,7	77,8	81,6	77,8	81,3	78,0	80,5
Saut (en %)		2,94		4,03		5,07		6,39		5,67		3,8		3,77		3,52		2,48
Imbrication (en cm)		10,3		12,2		9,2		9,1		9,1		8,8		8,8		9,8		10,8

Figure 84 : Sauts d'efficacité pour l'imbrication d'un dos et de 2 devants pour le modèle B

3.2.1.3.6 Un dos deux devants et une manche dans la laize

Le tableau de la **Figure 85** présente les sauts d'efficacité observés pour le modèle B lorsque l'on peut imbriquer un dos deux devants et une manche dans la laize. Cela concerne uniquement les 3 plus petites tailles car pour les tailles supérieures, cette combinaison de pièces n'est pas possible pour les laizes étudiées. Dans les trois cas on note des sauts d'efficacité avec des imbrications variables.

Taille	36		38		40	
Largeur d'un dos de 2 devants et d'une manche (cm)	180,6		186		191,5	
Laizes observées	169	173	173	177	177	179
Efficacités (en%)	74,3	80,9	74,5	80,8	74,5	80,6
Saut (en %)		6,57		6,37		6,02
Imbrication (en cm)		7,6		9		12,5

Figure 85 : Saut d'efficacité pour l'imbrication d'un dos, de 2 devants et d'une manche dans la laize

On ne peut pas réaliser non plus cette combinaison avec le modèle A dans les laizes étudiées.

3.2.1.4 Définition des indices

Le paragraphe précédent a permis de déceler des combinaisons de pièces qui, lorsqu'elles étaient possibles dans la laize disponible, permettaient d'obtenir des efficacités élevées. On peut calculer la laize théoriquement nécessaire pour que telle ou telle combinaison soit possible en cumulant la largeur des pièces concernées. Il arrive aussi parfois que le saut d'efficacité survienne pour une laize inférieure lorsque des phénomènes d'imbrication se produisent.

Pour tenir compte de ces deux paramètres, on définit 2 indices de la manière suivante :

Soit l la laize disponible pour le placement

On indice par i : $1 \leq i \leq 6$ les six combinaisons de pièces retenues par ordre croissant de largeur.

On appelle $(L_i)_{1 \leq i \leq 6}$ les laizes théoriquement nécessaires pour réaliser chacune des combinaisons.

Il existe i^* tel que: $L_{i^*} \leq l < L_{i^*+1}$

On pose:

$$I_{\text{prec}} = l - L_{i^*} = \underset{\substack{1 \leq i \leq 6 \\ l - L_i \geq 0}}{\text{Min}} (l - L_i)$$

$$I_{\text{suiv}} = L_{i^*+1} - l = \underset{\substack{1 \leq i \leq 6 \\ l - L_i < 0}}{\text{Min}} (L_i - l)$$

$I_{Préc}$ est la différence entre la laize disponible et la largeur nécessaire de la plus grande combinaison de pièces réalisable.

I_{Suiv} est la largeur qu'il faudrait théoriquement rajouter à la laize disponible pour pouvoir réaliser la plus petite combinaison de pièces non encore réalisable.

La **Figure 86** illustre ces 2 indices sur un exemple.

Ces indices permettent de connaître la position relative d'une laize donnée par rapport à deux autres laizes qui permettent en principe d'obtenir de bonnes efficacités.

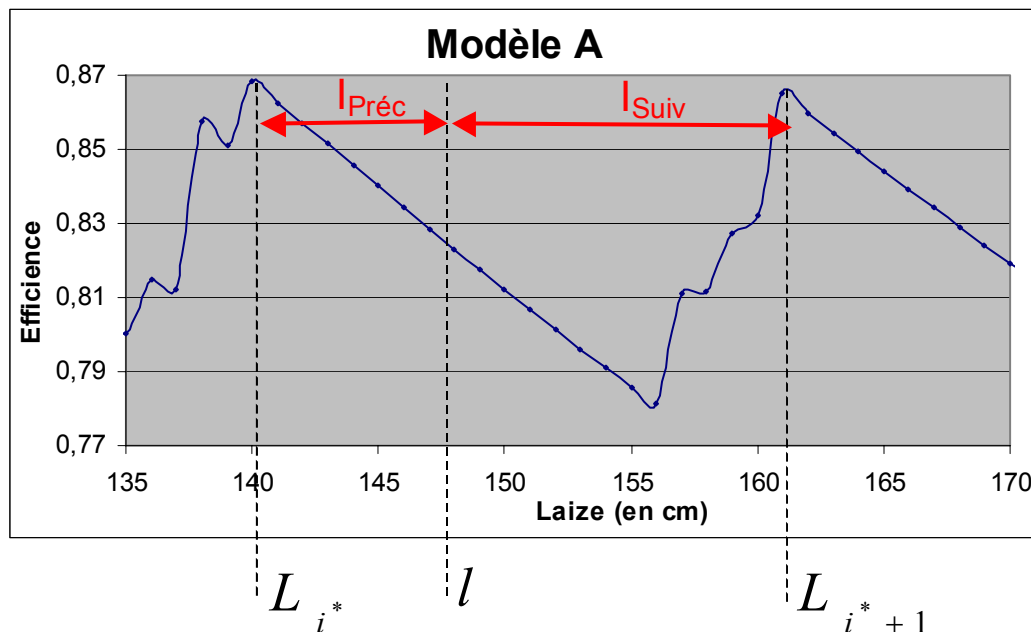


Figure 86 : Illustration des indices relatifs aux bonnes combinaisons de pièces

Cependant, certaines valeurs sont délicates à interpréter. Lorsque $I_{Préc}$ est grand et que I_{Suiv} est petit, on se trouve dans les valeurs de laize où le saut d'efficacité survient grâce à une nouvelle combinaison de pièces. L'indice de rectangularité des pièces contenues dans cette combinaison peut permettre de savoir pour quelle laize le saut va survenir. En effet, si l'indice de rectangularité est grand, il y aura très peu d'imbrications possibles entre les pièces et le saut d'efficacité sera probablement très brutal et il se produira pour une laize très proche

de la valeur théorique L_{i+1}^* . A l'inverse, si l'indice de rectangularité est petit, les imbrications entre pièces seront possibles, l'augmentation de l'efficience se fera de façon moins brutale mais aura lieu pour des laizes inférieures à L_{i+1}^*

3.2.2 Placements de pantalons

3.2.2.1 Notion de « colonne »

On l'a dit, dans un placement de pantalons, les pièces les plus importantes sont celles servant à fabriquer les jambes des pantalons. Au nombre de 4 par pantalon (pour un placement en pli simple) elles représentent à elles seules environ 80% de la quantité totale de tissu à découper et elles vont en grande partie déterminer l'efficacité du placement réalisé.

En effet, si on arrive à placer ces pièces de façon satisfaisante (sans perdre trop de place) il y a de grandes chances que le placement réalisé final ait une bonne efficacité car les pièces restantes sont de plus petites tailles et on arrive le plus souvent à les placer dans les espaces vides créés.

De plus, on remarque que ces pièces servant à fabriquer les jambes du pantalon s'organisent souvent en « colonnes » dont la hauteur et la largeur correspondent respectivement à la laize de tissu disponible et à la longueur des jambes du pantalon.

L'exemple suivant (cf. **Figure 87**) montre un placement contenant 3 tailles du modèle B en laize 135cm.

On distingue ici 4 colonnes qui contiennent chacune 3 « jambes ».

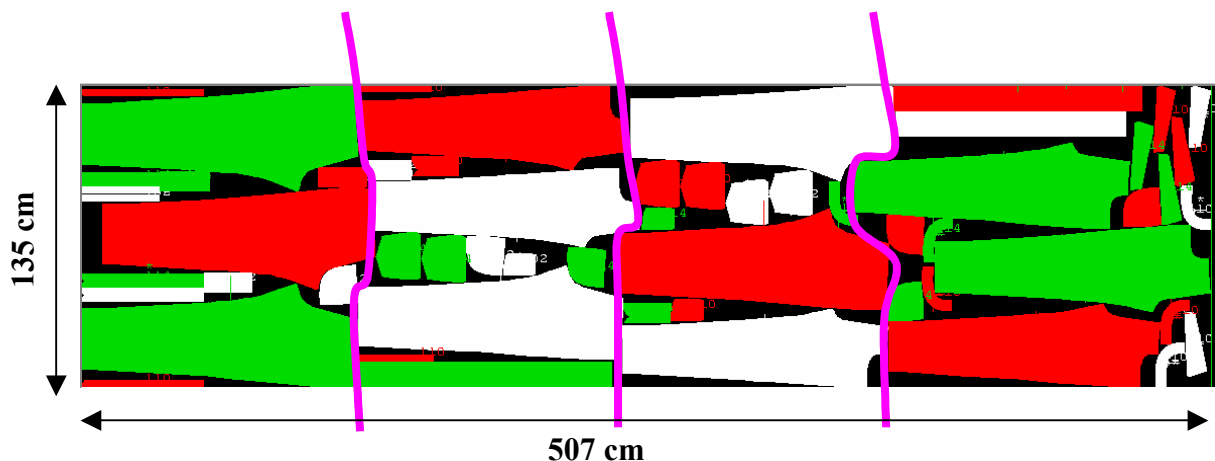


Figure 87 : Exemple de disposition en colonnes dans un placement de pantalons

Il est clair que si l'on arrive à savoir de combien de colonnes on a besoin pour pouvoir placer toutes les « jambes », on aura une bonne idée de la longueur totale du placement et donc de son efficacité.

La difficulté est alors d'obtenir ce nombre de colonnes. Il faut tenir compte pour cela du nombre de « jambes » à placer et de leur largeur, de la laize de tissu disponible mais aussi de possibles imbrications entre les pièces. En effet, le phénomène décrit ci-dessus laisse penser qu'il ne peut y avoir qu'un nombre entier de colonnes mais ce n'est pas toujours le cas.

L'exemple suivant (cf. **Figure 88**) montre le même placement que précédemment mais en laize 125cm :

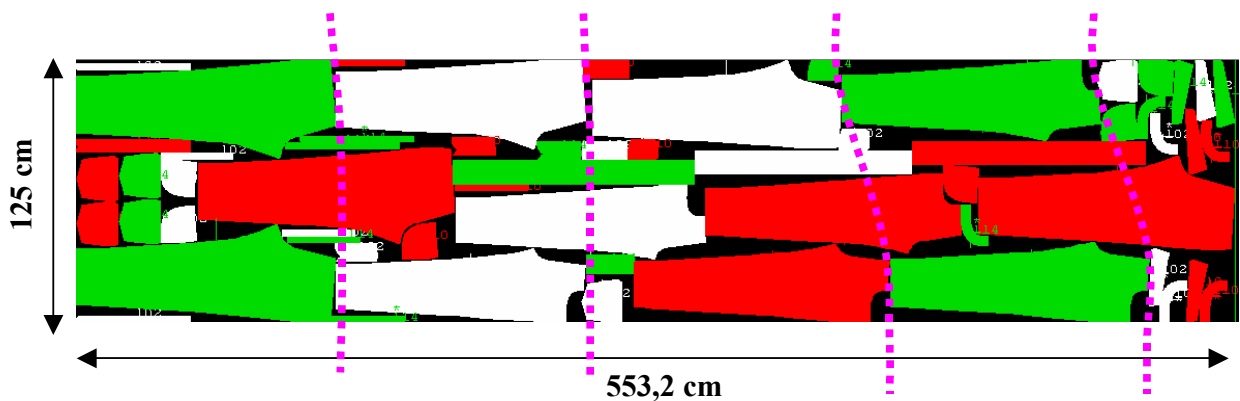


Figure 88 : Exemple de placement où la disposition en colonnes n'est pas évidente

Dans cet exemple, la laize du tissu est trop petite pour que toutes les « jambes » puissent être placées dans 4 colonnes, on s'attend alors à voir apparaître 5 colonnes mais en réalité, grâce à l'imbrication des pièces, en décalant légèrement les « jambes » qui se trouvaient au milieu de la laize, la longueur du placement ne correspond pas à 5 mais plutôt à 4,5 colonnes.

On verra un peu plus loin comment tenir compte de cette notion de colonne pour estimer l'efficacité des placements.

3.2.2.2 Remarque sur les ceintures

Les « colonnes » décrites ci-dessus ont été définies comme étant l'empilement dans la laize des pièces servant à fabriquer les jambes des pantalons. L'argument pour cela était que

l'arrangement de ces jambes influait grandement sur la qualité finale du placement. Cependant, sur la plupart des modèles de pantalons, la pièce servant à fabriquer la ceinture est presque aussi longue qu'une « jambe » pour une largeur d'environ 10 cm. Lorsque l'on veut déterminer le nombre de « colonnes » d'un placement sans tenir compte de cette pièce (on la considère comme une petite pièce que l'on pourra placer dans les espaces vides) il arrive souvent que l'on fasse une mauvaise estimation.

Les deux exemples suivants (cf. **Figure 89**) reprennent le placement précédent où l'on a placé uniquement les « jambes » puis les « jambes » et les « ceintures ».

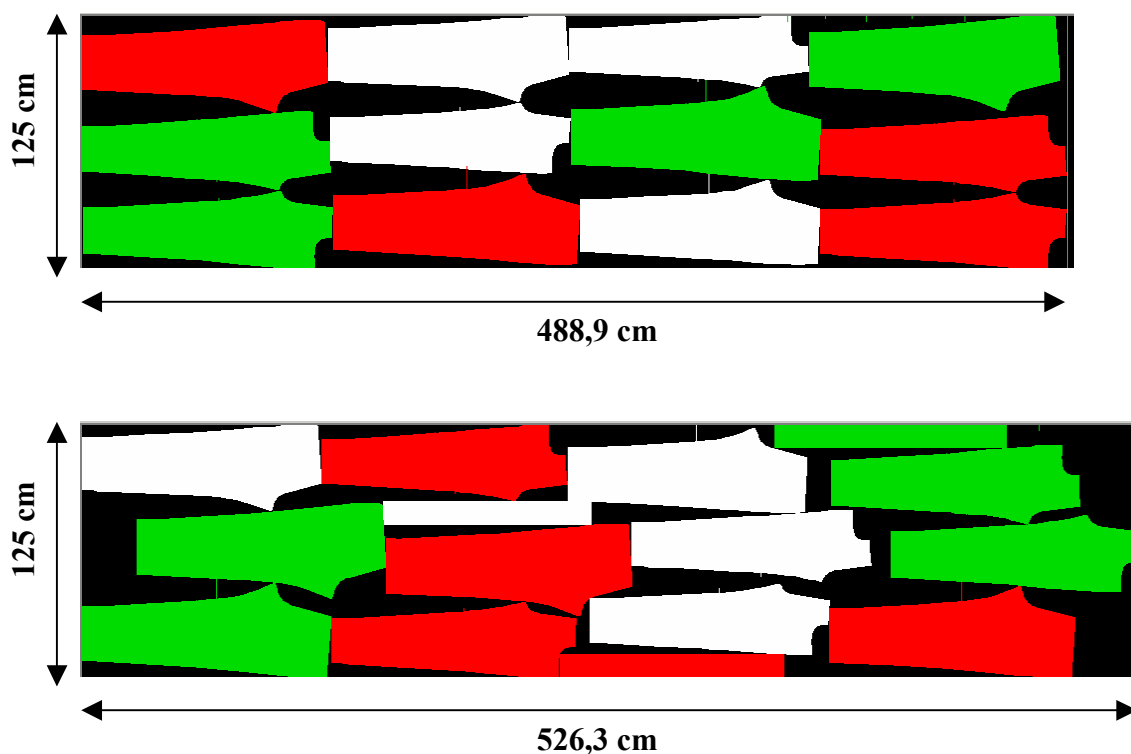


Figure 89 : Influence des ceintures sur la disposition en colonnes

Le fait de rajouter ces trois pièces a provoqué un allongement de presque 40 cm du placement. Les pièces servant à fabriquer les ceintures peuvent donc rarement être placées dans les espaces vides car elles sont trop longues, il faudra donc tenir compte de ces pièces (lorsqu'elles existent) lors de l'estimation du nombre de « colonnes » d'un placement.

3.2.2.3 Définition des indices

3.2.2.3.1 « Hauteur totale nécessaire »

Le nombre de « colonnes » d'un placement varie bien sûr avec la laize de tissu disponible.

Dans un premier temps, on calcule la laize qui serait nécessaire pour pouvoir empiler toutes les « jambes » (et les ceintures) les unes par-dessus les autres en ne formant qu'une seule « colonne ». La **Figure 90** représente ce genre d'empilement :

Afin d'avoir une première estimation du nombre de « colonnes » d'un placement, on divise cette hauteur par la laize du tissu disponible.

Reprenons l'exemple précédent avec la laize 125 cm, on obtient :

$$n_{th} = \frac{530,2}{125} = 4,2416$$

Ce nombre représente le nombre théorique de « colonnes » que l'on obtiendrait si l'on pouvait « remplir » chaque colonne au maximum, il donne une indication sur la longueur finale du placement mais ne correspond à rien dans la réalité. Il est clair que plus le nombre réel de colonnes sera proche de ce nombre (il est toujours supérieur) et plus l'efficacité du placement sera grande.

3.2.2.3.2 Estimation du nombre de colonnes

Afin d'avoir une meilleure idée du nombre réel de « colonnes » d'un placement, on considère les « rectangles englobant minimum » (cf. 3.1.5 Indice de « rectangularité ») des pièces servant à déterminer le nombre de « colonnes » (jambes et éventuellement ceinture) et on cherche pour une laize de tissu donnée, le nombre minimum de « colonnes » nécessaires pour placer tous ces rectangles englobant. Ceci peut se faire soit par une énumération des cas possibles (impossible pour les placements contenant beaucoup de pièces) ou par la résolution

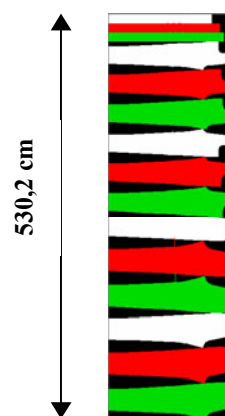


Figure 90 :

Hauteur totale nécessaire

d'un petit problème de programmation linéaire. Le nombre obtenu est donc une borne supérieure du nombre de colonnes réel car la disposition des rectangles est (dans le pire des cas) réalisable pour les pièces réelles.

Pour l'exemple précédent, on obtient : $n_{Max} = 5$

Ce nombre permet de mesurer l'écart entre le nombre de colonnes théorique calculé avec la « hauteur totale nécessaire » et le nombre de colonnes que l'on pourra effectivement obtenir. L'idée est que plus cet écart est grand, moins l'agencement des « grandes » pièces (jambes et ceinture) est bon et finalement, moins l'efficacité est bonne. Il faut cependant nuancer cette remarque car pour un placement court (un mono-taille par exemple) il y a une grande différence entre 1 colonne et 2 colonnes, alors que pour un placement plus long une différence de une colonne (par exemple 5 colonnes au lieu de 6) est moins importante.

L'exemple qui suit (cf. **Figure 91**) montre que le passage d'un placement « 1 colonne » (toutes les pièces tiennent dans la laize) à un placement « 2 colonnes » peut avoir un effet très néfaste sur l'efficacité.

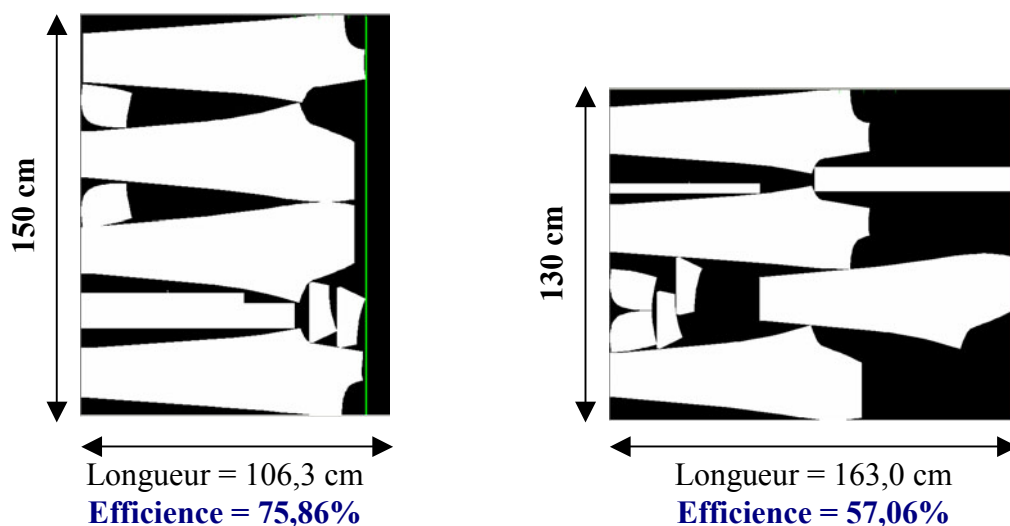


Figure 91 : Dégradation de l'efficacité entre 1 colonne et 2 colonnes

Pour tenir compte de cela, on divise l'écart précédent par le nombre de tailles du placement $NbTailles(P)$.

L'indice finalement utilisé est donc le suivant :
$$I_{NbColonnes}(P) = \frac{n_{Max}(P) - n_{Th}(P)}{NbTailles(P)}$$

3.2.2.3.3 Prise en compte des imbrications entre pièces

Dans le paragraphe précédent, on ne tenait pas compte des imbrications possibles entre pièces (on les identifie à des rectangles) il arrive donc parfois que grâce à ces imbrications, le nombre de « colonnes » réel d'un placement soit inférieur à n_{Max} . Pour prendre en compte ce phénomène (d'autant plus fréquent que le nombre de colonnes est grand), on associe au premier indice $I_{NbColonnes}$ un indicateur qui représente l'augmentation qu'il faudrait faire sur la laize du tissu pour que le nombre de colonnes diminue.

En effet, si pour une laize l donnée on obtient un $n_{Max} = 5$ mais que pour une laize de $l+1$ on obtient un $n_{Max} = 4$ les imbrications entre pièces permettront certainement de n'avoir que 4 colonnes dès la laize de tissu l . A l'inverse, si on est éloigné du passage au nombre de colonnes inférieur, le nombre réel de colonnes sera la plupart du temps égal à n_{Max} . On définit donc un indice I_{Adjust} par :

$$I_{Adjust}(P) = \frac{laize_{n_{Max}-1}(P) - laize(P)}{laize(P)}$$

où $laize_{n_{Max}-1}$ est la laize minimum nécessaire pour avoir une estimation du nombre de colonnes égale à $n_{Max} - 1$.

3.2.2.3.4 Interprétation graphique

Le graphique de la **Figure 92** présente la diminution du nombre de colonnes d'un placement en fonction de l'augmentation de la laize. La courbe bleue représente le nombre de colonnes théorique : n_{Th} (décroissance en $\frac{1}{x}$). La courbe rose représente le nombre de colonnes nécessaires pour placer les rectangles englobant des grosses pièces (jambes + ceinture) : n_{Max} (décroissance en escalier, fonction discontinue). La courbe turquoise représente le nombre réel de colonnes pour placer toutes les grandes pièces.¹

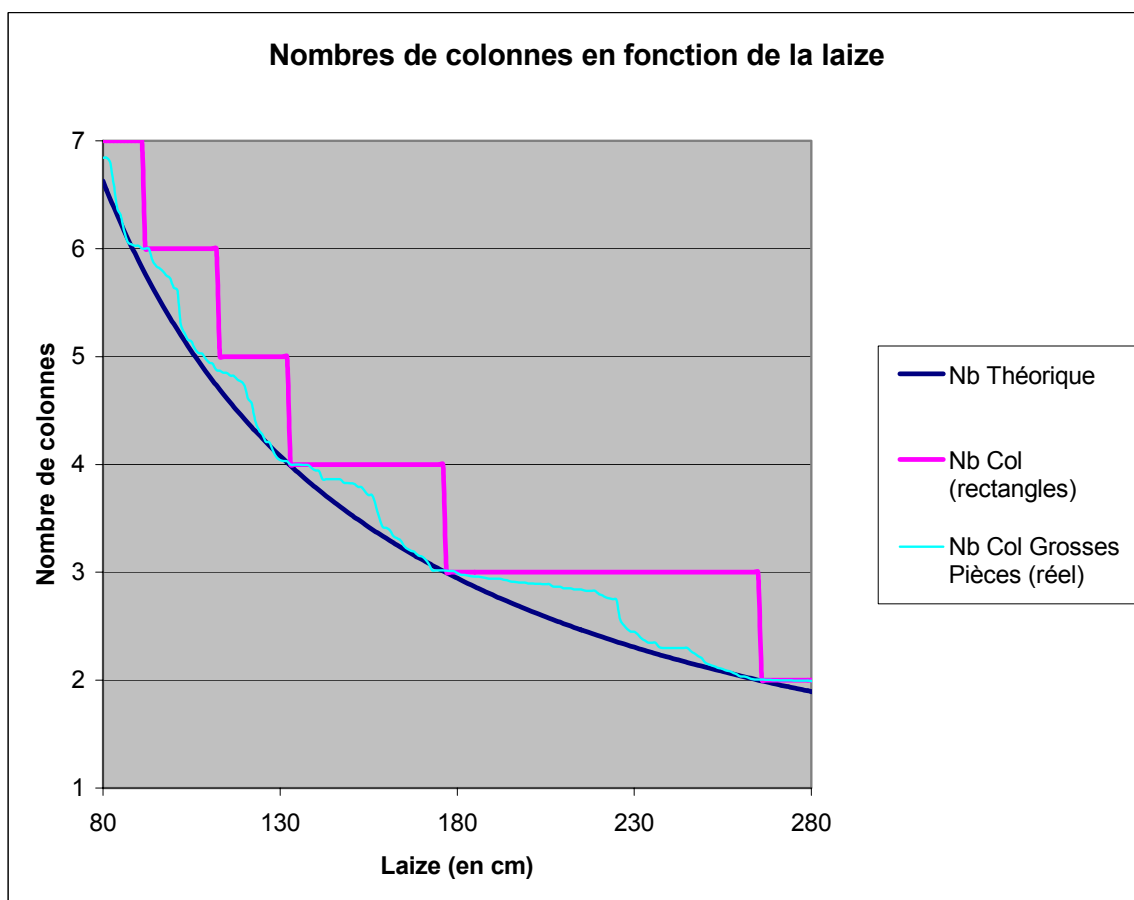


Figure 92 : Décroissance du nombre de colonnes en fonction de la laize

¹ Ce nombre n'est bien sûr pas connu lors de l'estimation de l'efficacité d'un placement. On le calcule une fois le placement réalisé en divisant la longueur du placement par la longueur moyenne des jambes du pantalon.

On observe que le nombre réel de colonnes est compris entre n_{Th} et n_{Max} . De plus, au début de chaque palier (pour la courbe de n_{Max}) le nombre de colonnes réel est très proche de n_{Th} et de n_{Max} . Ensuite, jusqu'au milieu de chaque palier, le nombre réel de colonnes s'éloigne à la fois de n_{Th} et de n_{Max} puis, en fin de palier il est à nouveau très proche de n_{Th} mais également proche de $n_{Max}-1$.

La **Figure 93** montre que les indices $I_{NbColonnes}$ et I_{Ajust} permettent de se situer entre les valeurs de n_{Th} et de n_{Max} . Ils permettent de rendre compte de deux critères opposés :

- Plus l'écart entre n_{Max} et n_{Th} est grand moins l'efficacité est élevée.
- Plus l'écart entre $laize_{n_{Max}-1}$ et $laize$ est faible plus l'efficacité est élevée.

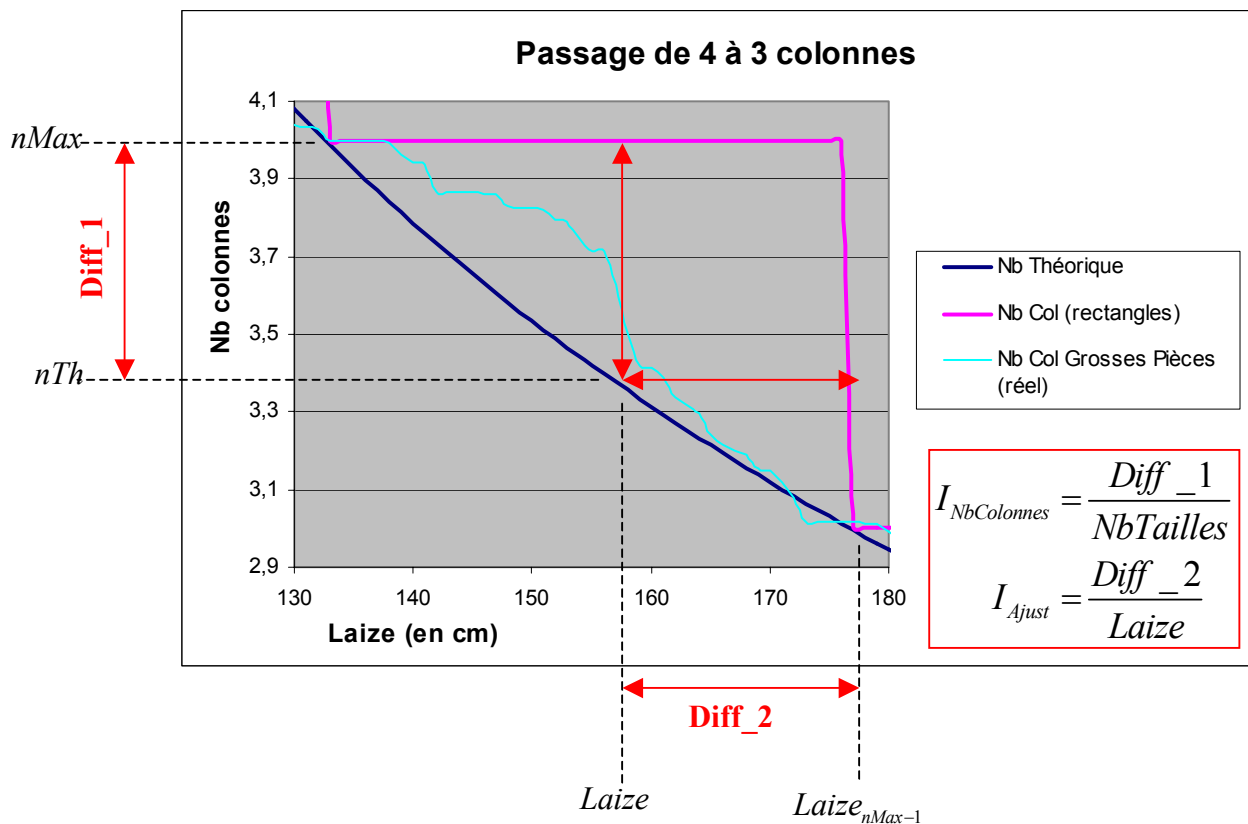


Figure 93 : Interprétation graphique des indices pour le passage de 4 à 3 colonnes

Chapitre 4 : Premiers résultats

4.1 Démarche

4.1.1 Placements mono-tailles et placements multi-tailles

L'étude de l'efficience des placements a principalement porté sur 2 types de vêtements, les modèles de chemises et les modèles de pantalons. On tente donc de construire deux modèles neuronaux distincts pour chacun des cas. L'objectif est bien sûr de prédire l'efficience de placements contenant un nombre de tailles quelconque, cependant l'étude des placements mono-taille est intéressante en soi car ceux-ci forment en quelque sorte les « briques » dont seront constitués les placements multi-tailles et de plus, les variations d'efficience les plus fortes sont observées pour les placements mono-taille. Ainsi, pour chacun des cas, on étudiera en premier lieu les placements mono-taille seuls, puis l'ensemble des placements.

Dans le cas de l'étude des placements mono-taille, on utilise des réseaux de neurones à une couche cachée comprenant 6 neurones cachés à fonction d'activation sigmoïde. Le but n'est pas de trouver le meilleur modèle possible pour les placements mono-taille, mais plutôt d'apprécier de façon rapide la qualité d'estimation que l'on peut obtenir sur des placements simples en fonction du nombre de mesures (les placements dont on connaît l'efficience) disponibles.

4.1.2 Elimination des mauvais résultats avérés

L'efficience des placements étudiés est calculée avec Diamino Expert. On ajuste l'efficience des placements pour lesquels le résultat fourni par Diamino est manifestement éloigné de la meilleure efficience possible, qui est inconnue. En effet, dans quelques cas, pour une laize donnée Diamino propose une bonne efficience correspondant à une longueur x . Pour une laize un peu supérieure (+1 cm par exemple) on attend une longueur maximale de x car il suffit de reproduire le placement précédent. Cependant il arrive qu'on obtienne une longueur

supérieure à x . Dans ce cas, on prendra comme efficacité du nouveau placement, l'efficacité correspondant à une longueur x .

4.1.3 Initialisation des poids du réseau

La phase d'apprentissage du réseau de neurones est une méthode locale de minimisation de l'erreur quadratique. Le résultat de l'apprentissage dépend donc de l'initialisation des paramètres qui sont les poids du réseau. Ainsi, pour chaque modèle de réseau testé, on effectue plusieurs apprentissages avec des initialisations des poids différentes.

4.1.4 Ensemble d'apprentissage – Ensemble de validation

Afin de pouvoir valider le modèle neuronal que l'on cherche à construire, on choisit de ne pas utiliser tous les exemples de placements pour réaliser l'apprentissage du réseau. L'ensemble des placements dont on dispose est alors scindé aléatoirement en deux sous-ensembles :

- L'ensemble d'apprentissage
- L'ensemble de validation

La proportion des placements dans chacun des sous-ensembles peut-être variable (cf. 4.2.1.3 et 4.3.1.3). L'ensemble de validation permet de comparer l'estimation du réseau à l'efficacité réelle des placements n'ayant pas participé à l'apprentissage. On peut alors calculer l'erreur de validation E_v .

4.1.5 Sélection des modèles

On s'est limité dans la recherche de modèle neuronal aux réseaux de neurones à une couche cachée à fonction d'activation sigmoïde. La sélection du modèle à retenir revient alors à déterminer le nombre de neurones dans la couche cachée.

Par analogie aux modèles de régression linéaire, on peut définir les quantités suivantes :

$$SS_T = \sum_{i=1}^N (y^i - \bar{y})^2 \quad \left(\text{où } \bar{y} = \frac{1}{N} \sum_{i=1}^N y^i \right)$$
 traduit la dispersion totale des valeurs d'efficience observées par rapport à la moyenne.

$$SS_R = \sum_{i=1}^N (y^i - g(x^i, \mathbf{w}))^2 = \sum_{i=1}^N r_i^2$$
 c'est la somme des carrés des différences (appelées résidus) entre les valeurs observées et les sorties du réseau. SS_R est la quantité minimisée lors de l'apprentissage du réseau.

$$SS_E = \sum_{i=1}^N (g(x^i, \mathbf{w}) - \bar{y})^2$$
 c'est la somme des carrés des écarts entre les sorties du réseau et la moyenne.

Si l'apprentissage s'est bien passé, le SS_R de l'ensemble d'apprentissage doit être faible par rapport au SS_T . Cependant, contrairement au cas de la régression linéaire on n'a pas nécessairement : $SS_T = SS_R + SS_E$.

Pour sélectionner le meilleur modèle possible, on dispose de plusieurs autres indicateurs :

$$EQMA = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} r_i^2}$$
 C'est l'erreur quadratique moyenne sur l'ensemble d'apprentissage. Elle fournit une indication sur la qualité de l'apprentissage, mais peut être rendue arbitrairement petite en augmentant le nombre de neurones cachés.

$$E_p = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} \left(\frac{r_i}{1-h_i} \right)^2}$$
 C'est le score de leave-one-out virtuel obtenu sur l'ensemble d'apprentissage. C'est une approximation de la performance de généralisation du modèle que l'on ne connaît pas. On a donc

intérêt à choisir un modèle pour lequel ce score est le plus bas possible.

$$E_v = \sqrt{\frac{1}{N_v} \sum_{i=1}^{N_v} r_i^2}$$

C'est l'erreur quadratique moyenne sur l'ensemble de validation n'ayant pas participé à l'apprentissage du réseau. Si l'ensemble de validation contient de nombreux exemples, elle fournit une bonne approximation de la performance de généralisation du modèle. Cela permet de vérifier la validité du score de leave-one-out virtuel.

$$\mu = \frac{1}{\sqrt{q \times N_A}} \sum_{i=1}^{N_A} \sqrt{h_i}$$

Cette quantité est comprise entre 0 et 1. Elle caractérise la distribution des leviers h_i (cf. 2.2.9.2), μ vaut 1 si tous les leviers sont égaux à q/N_A . Lorsque μ s'éloigne de 1, cela signifie que certains exemples ont une influence très forte sur les paramètres du modèle. On préférera plutôt un modèle équilibré avec un μ plus proche de 1.

4.2 Placements de pantalons

4.2.1 Placements mono-taille

4.2.1.1 Placements disponibles

On dispose de 3 modèles de pantalons différents comportant 10 tailles chacun.

On peut donc créer 30 placements monotailles différents. Chaque placement est décliné dans différentes laizes variant de 100 cm à 180 cm par palier de 1 cm. On obtient donc 81 exemplaires de chaque placement (contenant la même taille à placer) et finalement, 2430 placements monotailles pour les modèles de pantalons.

On calcul l'efficience de ces placements à l'aide de Diamino Expert.

4.2.1.2 50% Apprentissage – 50% Validation

On sépare de façon aléatoire l'ensemble des données en 2 ensembles de tailles égales. 1215 placements servent à réaliser l'apprentissage du réseau de neurones et les 1215 autres servent à valider le modèle obtenu.

Le tableau de la **Figure 94** montre les résultats obtenus après l'apprentissage.

	Ensemble d'apprentissage	Ensemble de validation
SS_T	5,6037	5,6887
SS_R	0,0825	0,0876
$EQMA$	0,824 (en% d'efficience)	
E_p	0,860 (en% d'efficience)	
E_v		0,849 (en% d'efficience)

Figure 94 : Résultat de l'apprentissage et performance de généralisation

L'apprentissage a été efficace car le SS_R de l'ensemble d'apprentissage est faible par rapport au SS_T . Cela correspond à une erreur quadratique moyenne de 0,824% d'efficience sur l'ensemble d'apprentissage. Il semble donc y avoir un très léger surajustement pour les exemples ayant participé à l'apprentissage car l'estimation de la performance de généralisation E_v fournie par l'ensemble de validation est de 0,849% d'efficience. L'estimation de la performance de généralisation E_p obtenue par le leave-one-out virtuel est 0,86% d'efficience ce qui est proche de E_v .

Le graphique de la **Figure 95** regroupe les placements de l'ensemble de validation selon leur erreur de prédiction. On remarque dans un premier temps que l'on a quasiment un équilibre entre sur-estimation et sous-estimation (52% des placements ont eu leur efficience sous-estimée et 48% sur-estimée).

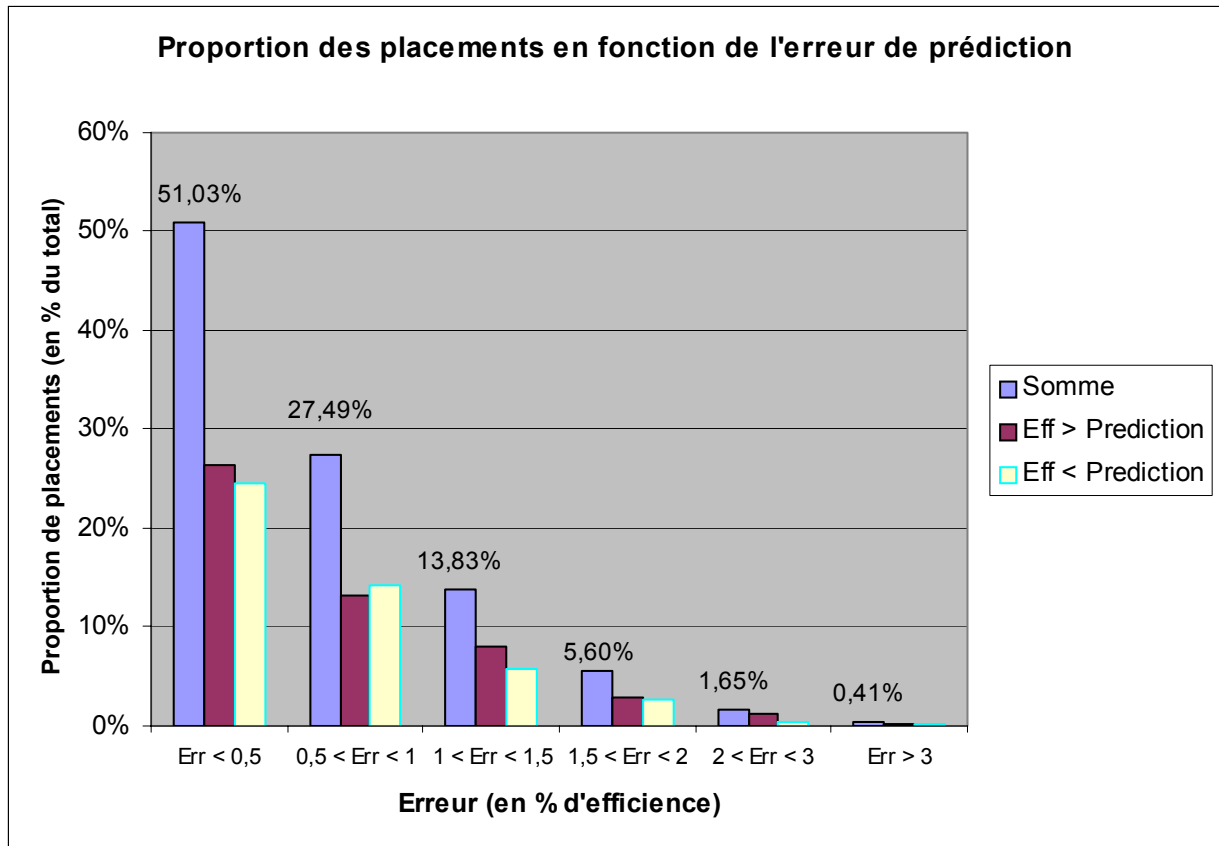


Figure 95 : Regroupement des placements par erreur de prédiction similaire

En résumé :

- Dans 78,52% des cas, l'erreur de prédiction est inférieure en valeur absolue à 1%.
- Dans 2,06% des cas, l'erreur de prédiction est supérieure en valeur absolue à 2%.

4.2.1.3 Diminution du nombre de placements dans l'ensemble d'apprentissage

On sépare de nouveau l'ensemble des placements en deux mais on diminue le nombre de placements utilisés pour l'apprentissage (à l'inverse le nombre de placements destinés à la validation augmente). En effet, dans le cas d'une utilisation réelle, il est intéressant d'obtenir la meilleure performance de généralisation possible avec le moins d'exemples de placements existants.

On réalise l'apprentissage du réseau dans les 3 cas suivants :

- Pourcentage de placements dans l'ensemble d'apprentissage : 33% (noté App33-Val67)
- Pourcentage de placements dans l'ensemble d'apprentissage : 10% (noté App10-Val90)
- Pourcentage de placements dans l'ensemble d'apprentissage : 5% (noté App5-Val95)

Le tableau de la **Figure 96** présente les résultats sur l'ensemble de validation dans chacun des cas avec comme témoin les résultats obtenus lorsque l'ensemble de validation est de même taille que l'ensemble d'apprentissage :

%Apprentissage/ %Validation	Nombre de placements servant à l'apprentissage	E_p	E_v
50/50	1215	0,860%	0,849%
33/67	810	0,842%	0,892%
10/90	243	0,989%	1,002%
5/95	122	0,935%	1,225%

Figure 96 : Performance de généralisation suivant la taille de l'ensemble d'apprentissage

Comme on pouvait s'y attendre, plus le nombre de placements dans l'ensemble d'apprentissage est faible, plus la performance de généralisation (sur l'ensemble de validation) est mauvaise (l'erreur quadratique moyenne augmente). Le score de leave-one-out virtuel qui doit être une estimation de la performance (théorique) de généralisation est relativement proche de l'erreur quadratique moyenne dans les 3 premiers cas. On note cependant qu'il la sous-estime toujours sauf lorsque l'ensemble d'apprentissage est le plus grand.

Le graphique de la **Figure 97** reprend la colonne « somme » de la **Figure 95** dans les 4 cas précédents.

La proportion de placement dont l'erreur est supérieure à 2% d'efficacité en valeur absolue a presque été multipliée par 5 (2,06% contre 9,78%) lorsque l'ensemble d'apprentissage passe de 50% à 5% du nombre total de placements. Dans le même temps, la proportion de placements dont l'erreur de prédiction est inférieure à 0,5% d'efficacité passe de 51,03% à 41,26%. Les regroupements intermédiaires ont pour leur part peu évolués (de 46,92% à 48,96%).

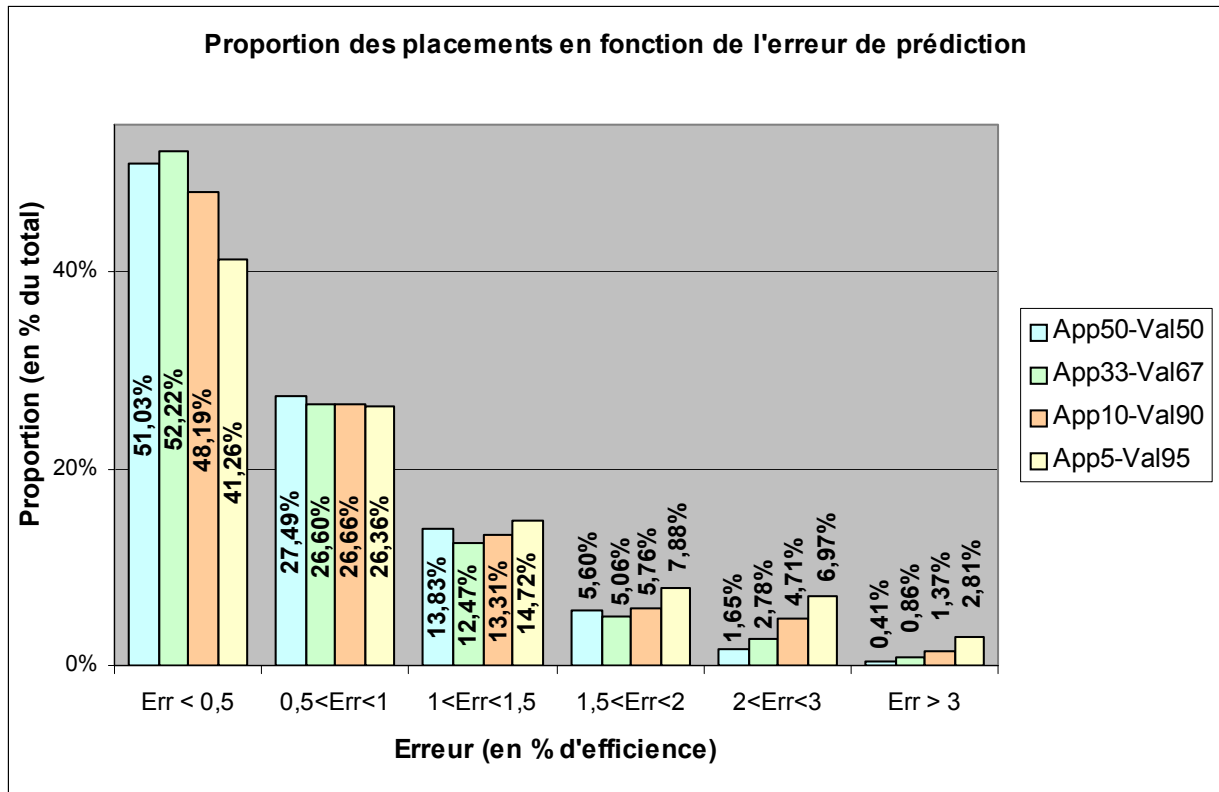


Figure 97 : Répartition des placements en fonction de l'erreur pour différentes tailles de l'ensemble d'apprentissage

Les performances du modèle se dégradent donc lorsque l'on réduit le nombre de placements dans l'ensemble d'apprentissage mais pas de façon catastrophique puisque même dans le pire des cas, on a plus de 2/3 des placements pour lesquels l'erreur de prédiction est inférieure à 1% en valeur absolue.

Plus généralement, les modèles construits pour les placements de pantalons monotailles sont de très bonne qualité grâce notamment à la notion de colonne (cf. 3.2.2.1 Notion de « colonne ») qui s'avère particulièrement efficace pour de petits placements.

4.2.2 Résultats sur tous les placements

4.2.2.1 Placements disponibles

Pour chacun des 3 modèles de pantalon disponibles, on étudie l'efficacité de placements contenant 1, 2 et 3 tailles :

- 10 placements mono-taille (exhaustif)
- 55 placements 2-tailles (exhaustif)
- 30 placements 3-tailles (non exhaustif)

Chaque placement est décliné dans différentes laizes variant de 100 cm à 180 cm par palier de 5 cm. On obtient donc 17 exemplaires de chaque placement (contenant le même ensemble de tailles à placer). On obtient finalement $4845 (= 3 \times (10 + 55 + 30) \times 17)$ placements en tout.

On sépare de façon aléatoire l'ensemble des données en 2 ensembles de tailles (presque) égales. 2445 placements servent à réaliser l'apprentissage du réseau de neurones et les 2400 autres servent à valider le modèle obtenu.

4.2.2.2 Sélection du modèle (nombre de neurones cachés)

On réalise différents apprentissages du réseau de neurones avec différentes initialisations des poids pour des modèles de complexité croissante. Le graphique de la **Figure 98** présente les résultats moyens obtenus pour chaque complexité.

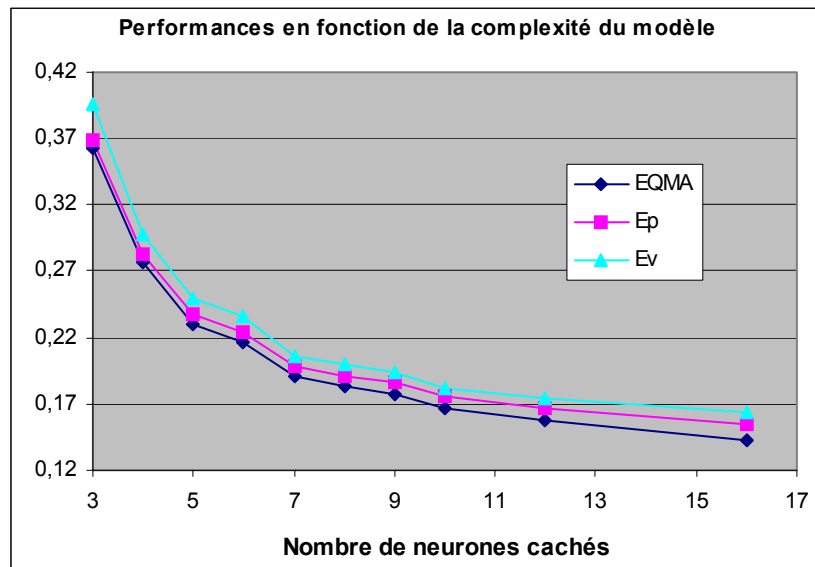


Figure 98 : Evolution des différents indicateurs selon le nombre de neurones cachés

Les indicateurs se classent dans le même ordre dans tous les cas : $EQMA < E_p < E_v$. Ils décroissent avec l'augmentation du nombre de neurones cachés, cependant on note que E_p est très proche de $EQMA$ pour les modèles les moins complexes et s'en éloigne pour les modèles les plus complexes. A l'inverse, E_p et E_v sont plus proches pour les modèles les plus complexes.

Le graphique de la **Figure 99** représente E_v en fonction de E_p pour tous les apprentissages effectués. Les couleurs correspondent au nombre de neurones cachés (notés NC) des modèles.

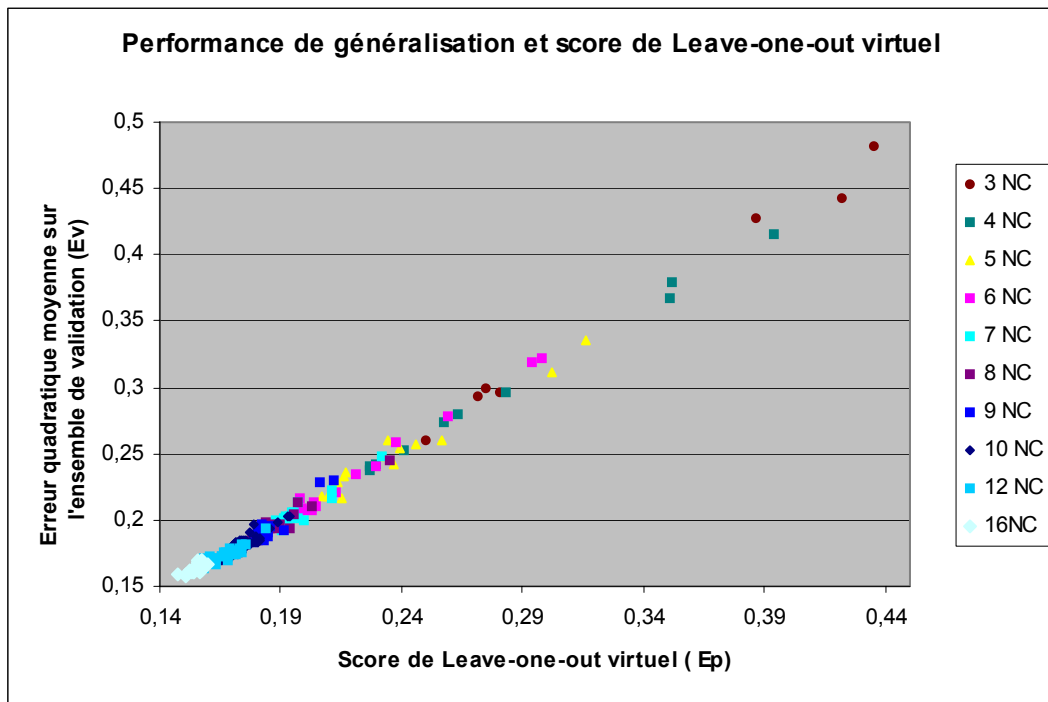


Figure 99 : E_p estimateur de la performance de généralisation du modèle ?

Même si E_p semble toujours légèrement sous-estimer la performance du modèle¹, le graphique montre qu'il y a une forte corrélation entre les deux. On a donc tout intérêt à choisir un modèle pour lequel le score de leave-one-out virtuel est le plus faible possible.

Le graphique de la **Figure 100** représente la différence $E_v - E_p$ en fonction de μ . Il semble que E_v soit plus proche de E_p lorsque μ est grand. Dans ce cas, E_p serait un meilleur estimateur de la performance de généralisation. Cela peut permettre de choisir entre plusieurs modèles ayant la même valeur de E_p .

¹ L'ensemble de validation étant important (2400 exemples) on considère que E_v est une bonne approximation de la performance de généralisation du modèle.

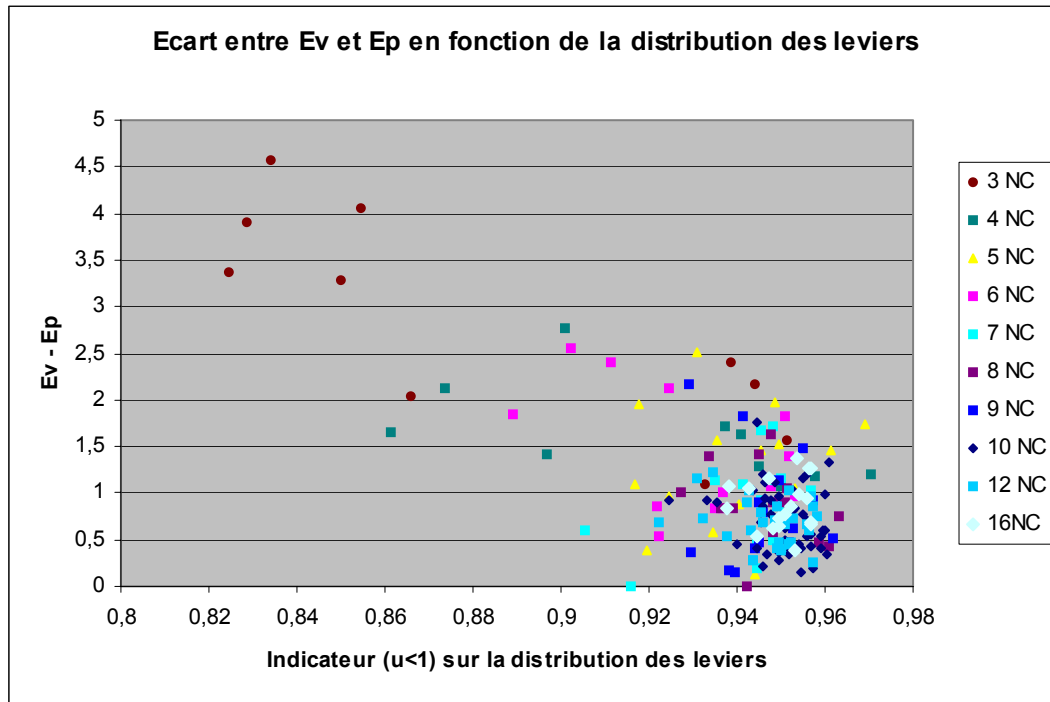


Figure 100 : Influence de la distribution des leviers μ

4.2.2.3 Stratégie de sélection du modèle

Les paragraphes précédents montrent que le critère le plus important pour évaluer la qualité d'un modèle est la valeur du score de leave-one-out virtuel E_p . Si plusieurs modèles ont une valeur de E_p quasiment égale, on a alors intérêt à conserver celui pour lequel la valeur de μ est la plus proche de 1. Enfin, entre deux modèles équivalents en termes de E_p et μ on choisit bien sûr le moins complexe (celui qui a le moins de neurones cachés).

4.2.2.4 Résultats pour un modèle à 12 neurones cachés

Le tableau de la **Figure 101** présente les résultats obtenus après l'apprentissage. La somme des carrés des résidus SS_R est faible par rapport à la dispersion totale. Cela montre que l'apprentissage du réseau a été efficace. De plus, le score de leave-one-out virtuel E_p est proche de E_v , on dispose donc d'une bonne estimation de la performance de généralisation du modèle.

	Ensemble d'apprentissage	Ensemble de validation
SS_T	5,9126	6,1921
SS_R	0,1313	0,1563
$EQMA$	0,733 (en% d'efficience)	
E_p	0,777 (en% d'efficience)	
E_v		0,807 (en% d'efficience)

Figure 101 : Performance du modèle à 12 neurones cachés

Le graphique de la **Figure 102** regroupe les placements de l'ensemble de validation selon leur erreur de prédiction.

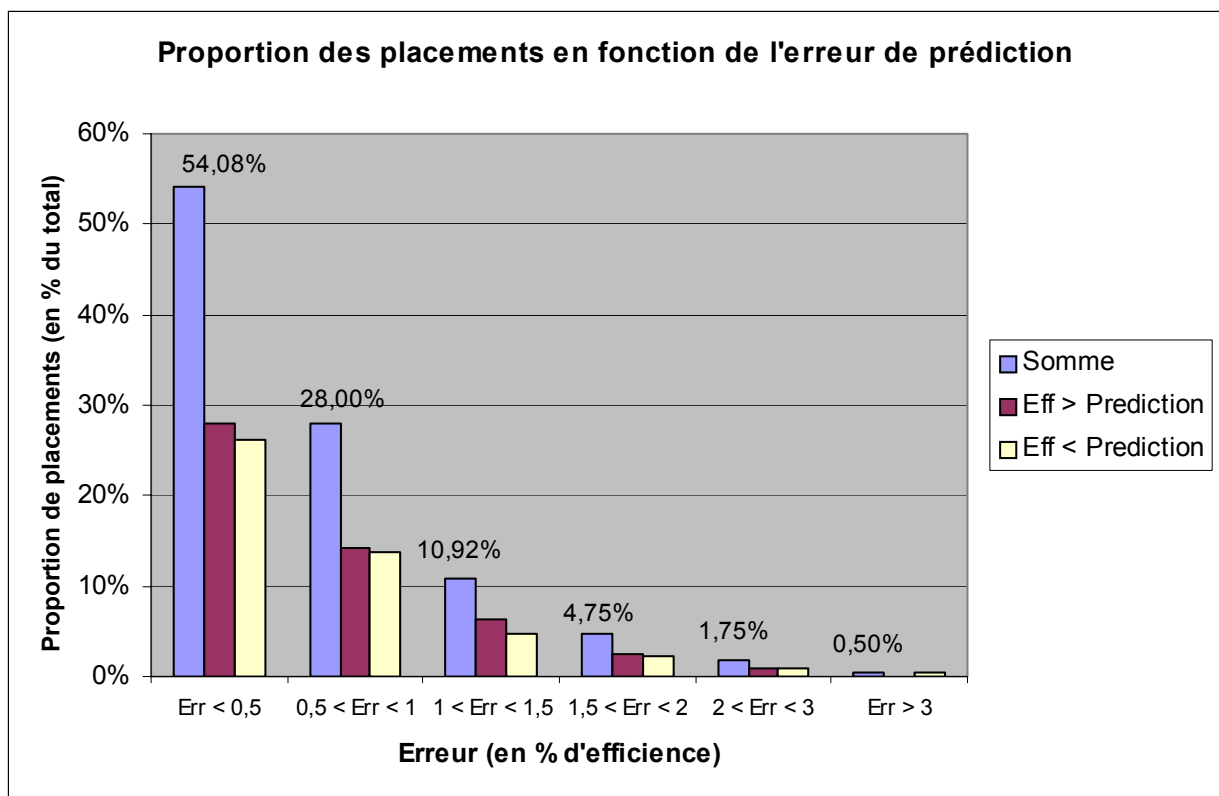


Figure 102 : Regroupement des placements par erreur de prédiction similaire

En résumé :

- Dans 82,08% des cas, l'erreur de prédiction est inférieure en valeur absolue à 1%.
- Dans 2,25% des cas, l'erreur de prédiction est supérieure en valeur absolue à 2%.

4.3 Placements de chemises

4.3.1 Placements mono-taille

4.3.1.1 Placements disponibles

On dispose de 2 modèles de chemise différents comportant respectivement 10 tailles et 9 tailles.

On peut donc créer 19 placements monotailles différents. Chaque placement est décliné dans différentes laizes variant de 100 cm à 180 cm par palier de 1 cm. On obtient donc 81 exemplaires de chaque placement (contenant la même taille à placer) et finalement, 1539 placements monotailles pour les modèles de chemise.

On calcule l'efficacité de ces placements à l'aide de Diamino Expert.

4.3.1.2 50% Apprentissage – 50% Validation

On sépare de façon aléatoire l'ensemble des données en 2 ensembles de tailles (presque) égales. 766 placements servent à réaliser l'apprentissage du réseau de neurones et les 773 autres servent à valider le modèle obtenu.

Le tableau de la **Figure 103** présente les résultats de l'apprentissage.

	Ensemble d'apprentissage	Ensemble de validation
SS_T	0,6781	0,6515
SS_R	0,0640	0,0889
$EQMA$	0,914 (en% d'efficacité)	
E_p	1,005 (en% d'efficacité)	
E_v		1,072 (en% d'efficacité)

Figure 103 : Résultats de l'apprentissage et performance de généralisation

On remarque tout d'abord que la dispersion totale des efficacités par rapport à la moyenne SS_T est beaucoup plus faible pour les modèles de chemise que pour les modèles de pantalons (cf. 4.2.1.2). L'amplitude des variations d'efficacité pour les modèles de chemises est donc

moins importante que pour les modèles de pantalons. L'apprentissage a également été moins efficace car le rapport SS_R / SS_T est de 0,094 alors qu'il valait 0,0147 dans le cas des placements de pantalons. La différence entre l'erreur quadratique moyenne ($EQMA = 0,914\%$) et l'erreur sur l'ensemble de validation ($E_v = 1,072\%$) montre un surajustement également plus important. Cependant, l'estimation de la performance de généralisation E_p obtenue par le leave-one-out virtuel est de 1,005% d'efficacité ce qui est relativement proche de E_v .

Il semble donc que l'on ait plus de difficulté à estimer les petites variations d'efficacité des placements de chemises. Le graphique de la **Figure 104** regroupe les placements de l'ensemble de validation selon leur erreur de prédiction.

On remarque dans un premier temps que l'on a quasiment un équilibre entre sur-estimation et sous-estimation (48,5% des placements ont eu leur efficacité sous-estimée et 51,5% sur-estimée).

En résumé :

- Dans 75,93% des cas, l'erreur de prédiction est inférieure en valeur absolue à 1%.
- Dans 8,15% des cas, l'erreur de prédiction est supérieure en valeur absolue à 2%.

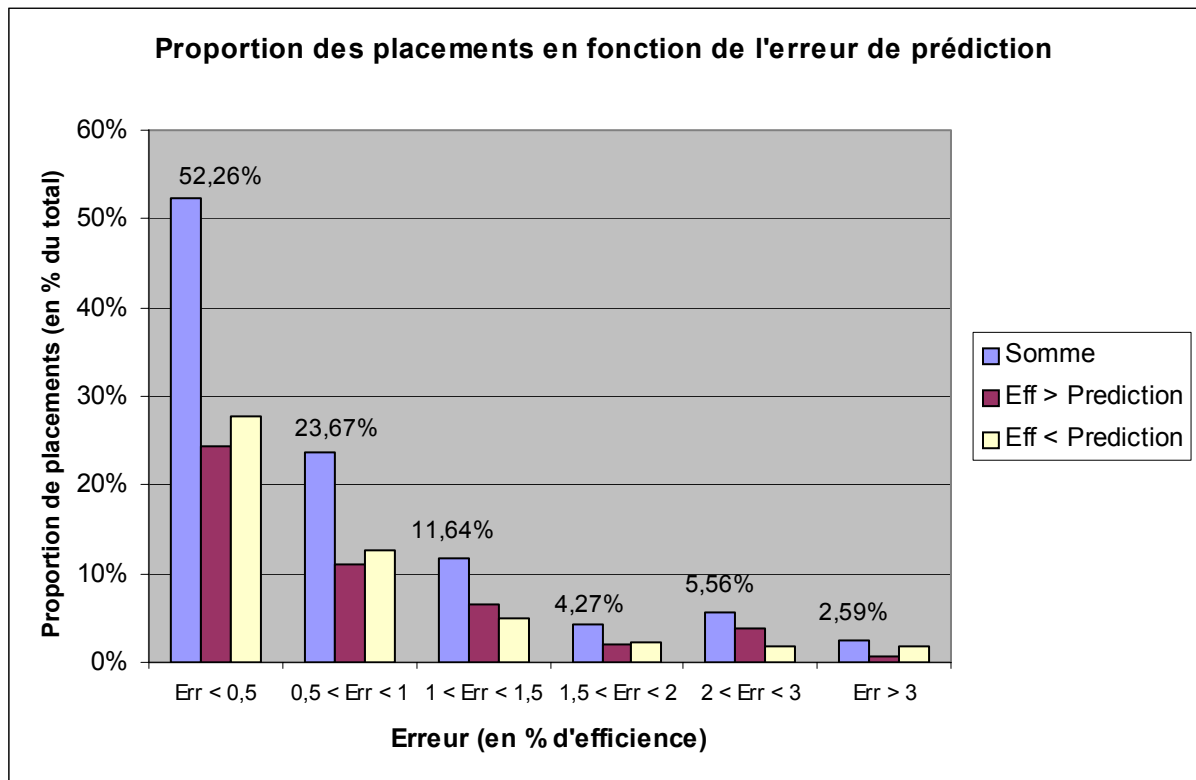


Figure 104 : Regroupement des placements par erreur de prédiction similaire

4.3.1.3 Diminution du nombre de placements dans l'ensemble d'apprentissage

On sépare de nouveau l'ensemble des placements en deux mais on diminue le nombre de placements utilisés pour l'apprentissage (à l'inverse le nombre de placements destinés à la validation augmente). En effet, dans le cas d'une utilisation réelle, il est intéressant d'obtenir la meilleure performance de généralisation possible avec le moins d'exemples de placements existants.

On réalise l'apprentissage du réseau dans les 3 cas suivants :

- Pourcentage de placements dans l'ensemble d'apprentissage : 33% (App33-Val67)
- Pourcentage de placements dans l'ensemble d'apprentissage : 10% (App10-Val90)
- Pourcentage de placements dans l'ensemble d'apprentissage : 5% (App5-Val95)

Le tableau de la **Figure 105** présente les résultats sur l'ensemble de validation dans chacun des cas avec comme témoin les résultats obtenus lorsque l'ensemble de validation est de même taille que l'ensemble d'apprentissage :

%Apprentissage/ %Validation	Nombre de placements servant à l'apprentissage	E_p	E_v
50/50	766	1,005	1,072
33/67	513	1,0387	1,1479
10/90	154	1,148	1,427
5/95	77	0,774	1,839

Figure 105 : Performance de généralisation selon la taille de l'ensemble d'apprentissage

Là encore (cf. 4.2.1.3), plus le nombre de placements dans l'ensemble d'apprentissage est faible, plus la performance de généralisation sur l'ensemble de validation est mauvaise (E_v augmente) et plus l'estimation cette performance de généralisation par le score de leave-one-out virtuel E_p est éloignée de E_v .

Le graphique de la **Figure 106** reprend la colonne « somme » de la **Figure 104** dans les quatre cas précédents :

La proportion de placements dont l'erreur est supérieure à 1,5% d'efficacité en valeur absolue a été plus que multipliée par 2 (12,42% contre 29,05%) lorsque l'ensemble d'apprentissage passe de 50% à 5% du nombre total de placements. Dans le même temps, la proportion de placements dont l'erreur de prédiction est inférieure à 0,5% d'efficacité passe de 52,26% à 33,13%. Les regroupements intermédiaires ont pour leur part peu évolués (de 35,32% à 37,82%).

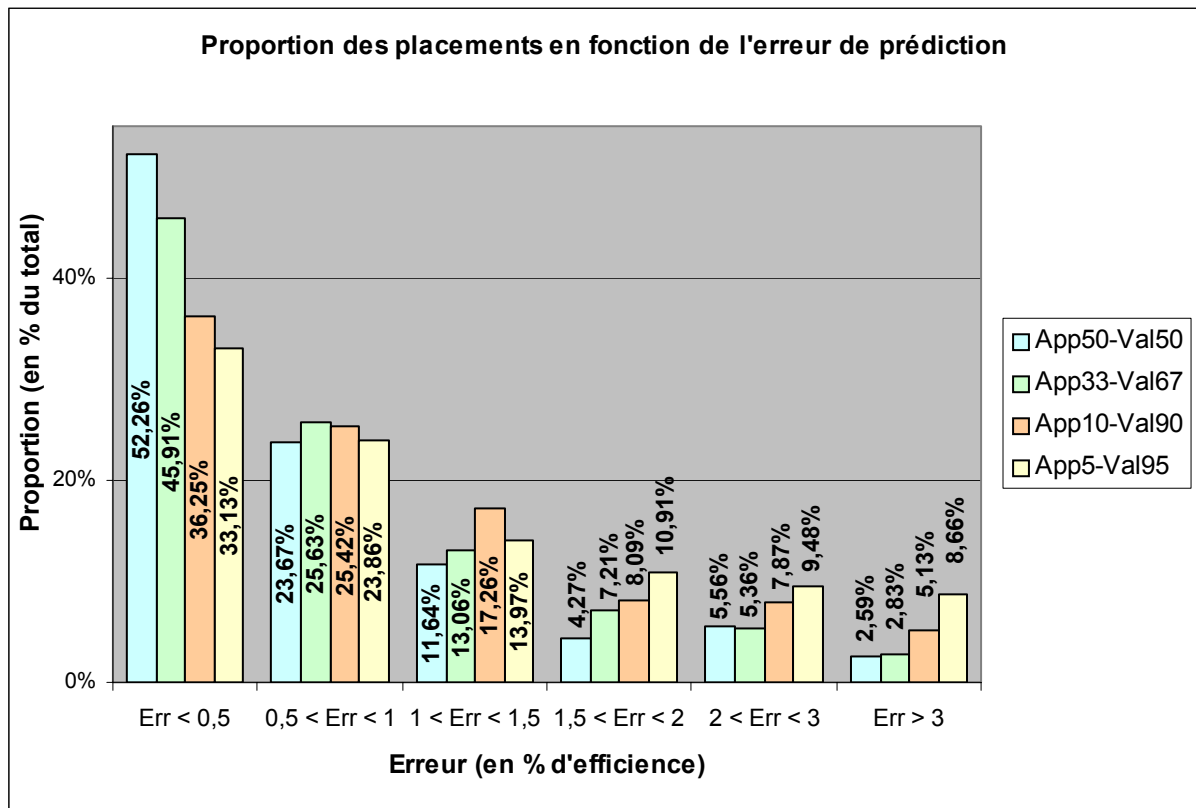


Figure 106 : Répartition des placements en fonction de l'erreur pour différentes tailles de l'ensemble d'apprentissage

Les performances du modèle se dégradent donc lorsque l'on réduit le nombre de placements dans l'ensemble d'apprentissage en particulier dans le dernier cas où le score de leave-one-out virtuel n'est plus une approximation de la performance de généralisation.

Plus généralement, les modèles construits pour les placements de chemise monotailles sont de qualité moyenne. En effet, dans le meilleur des cas, il reste plus de 8% des placements pour lesquels l'erreur de prédiction est supérieure à 2% d'efficience en valeur absolue.

4.3.2 Résultats sur tous les placements

4.3.2.1 Placements disponibles

Pour les 2 modèles de chemise disponibles, on étudie l'efficacité de placements contenant 1, 2 et 3 tailles. Le premier modèle contient 10 tailles différentes et le second seulement 9. Le détail des placements réalisés est le suivant :

- 19 placements mono-taille (10 + 9 exhaustif)
- 100 placements 2-tailles (55 + 45 exhaustif)
- 204 placements 3-tailles (120 + 84 non exhaustif)

Chaque placement est décliné dans différentes laizes variant de 100 cm à 180 cm. Pour les placements mono-taille, la laize évolue de centimètre en centimètre. On obtient donc 81 exemplaires de chaque placement mono-taille soit 1539 placements. Pour les placements multi-tailles, on fait varier la laize par palier de 5 cm. On obtient donc 17 exemplaires de chaque placement (contenant le même ensemble de tailles à placer). On obtient alors, 5168 placements contenant 2 ou 3 tailles. Le nombre total de placements réalisés est donc 6707.

On sépare de façon aléatoire l'ensemble des données en 2 ensembles de tailles (presque) égales. 3354 placements servent à réaliser l'apprentissage du réseau de neurones et les 3353 autres servent à valider le modèle obtenu.

4.3.2.2 Sélection du modèle (nombre de neurones cachés)

On réalise différents apprentissages du réseau de neurone avec différentes initialisations des poids pour des modèles de complexité croissante. Le graphique de la **Figure 107** présente les résultats moyens obtenus pour chaque complexité.

L'erreur quadratique moyenne sur l'ensemble d'apprentissage (EQMA) et le score de leave-one-out virtuel (E_p) décroissent avec l'augmentation du nombre de neurones cachés. E_p est toujours supérieur à EQMA et la différence entre les deux augmente en fonction de la complexité. La performance de généralisation du modèle (identifiée à E_v) est très proche de E_p pour les complexités les moins élevées. En revanche, on observe une dégradation de la

performance de généralisation pour les modèles les plus complexes pour lesquels E_p n'est plus une bonne approximation de E_v .

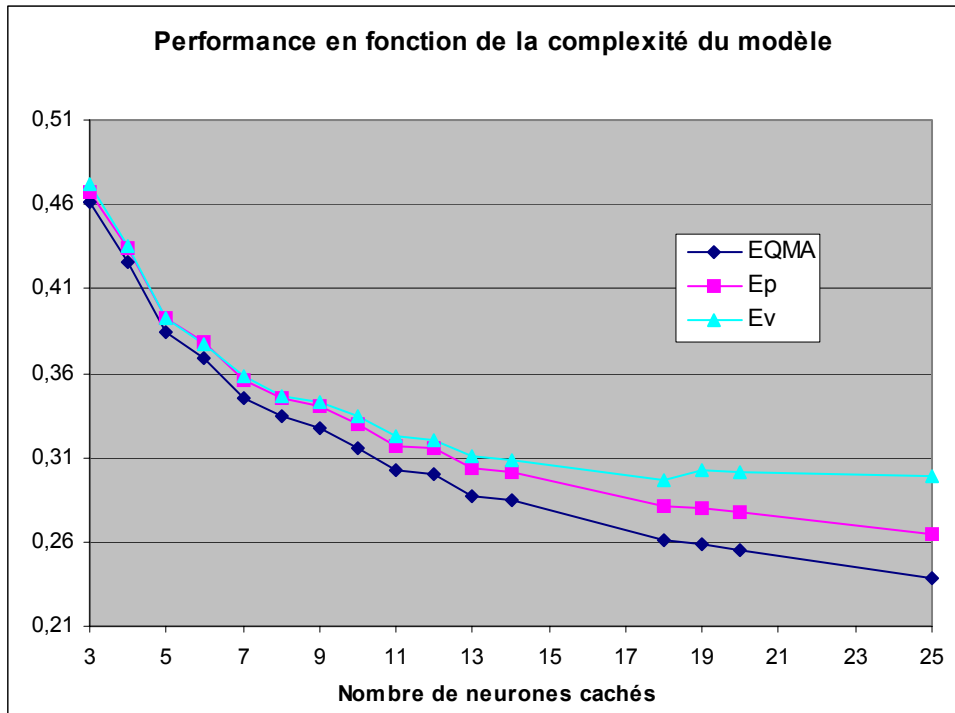


Figure 107 : Evolution des différents indicateurs selon le nombre de neurones cachés

Le graphique de la Figure 108 représente E_v en fonction de E_p pour tous les apprentissages effectués. Les couleurs correspondent à la complexité des modèles. Pour les complexité allant jusqu'à 14 neurones cachés, E_p est une bonne estimation de E_v , les valeurs sont réparties (hormis 2 cas) de part et d'autre de la première bissectrice (en noir). En revanche, pour les complexité supérieure (à partir de 18 neurones cachés) E_v est toujours supérieure à E_p .

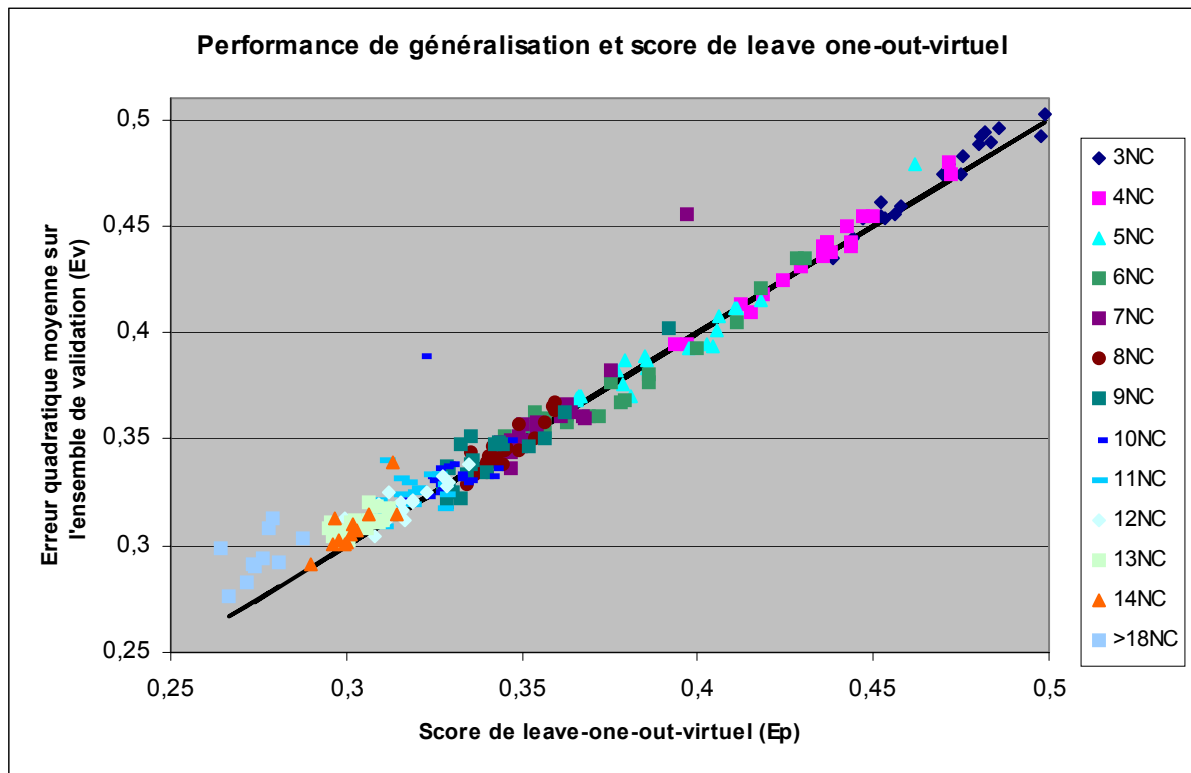


Figure 108 : E_p estimateur de la performance de généralisation du modèle ?

Le graphique de la **Figure 109** représente la différence $E_v - E_p$ en fonction de μ . Contrairement au cas des placements de pantalon, μ ne semble pas avoir une influence marquée sur la différence $E_v - E_p$ (on observe de faibles valeurs de μ avec de faibles valeurs de $E_v - E_p$ et inversement). On ne pourra pas utiliser la quantité μ pour choisir entre plusieurs modèles ayant des scores de leave-one-out virtuel similaires.

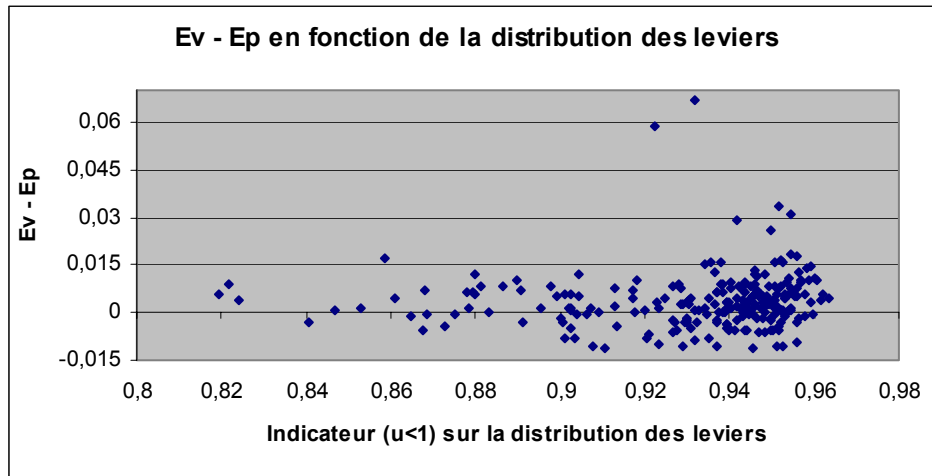


Figure 109 : Influence de la distribution des leviers μ ?

Le graphique de la **Figure 110** représente la différence $E_v - E_p$ en fonction de $E_p - EQMA$. Pour les complexités faibles, on n'observe pas réellement de tendance marquée. En revanche, pour les complexités élevées (supérieures à 14 neurones cachés) l'écart $E_v - E_p$ grandit presque proportionnellement à l'écart $E_p - EQMA$.

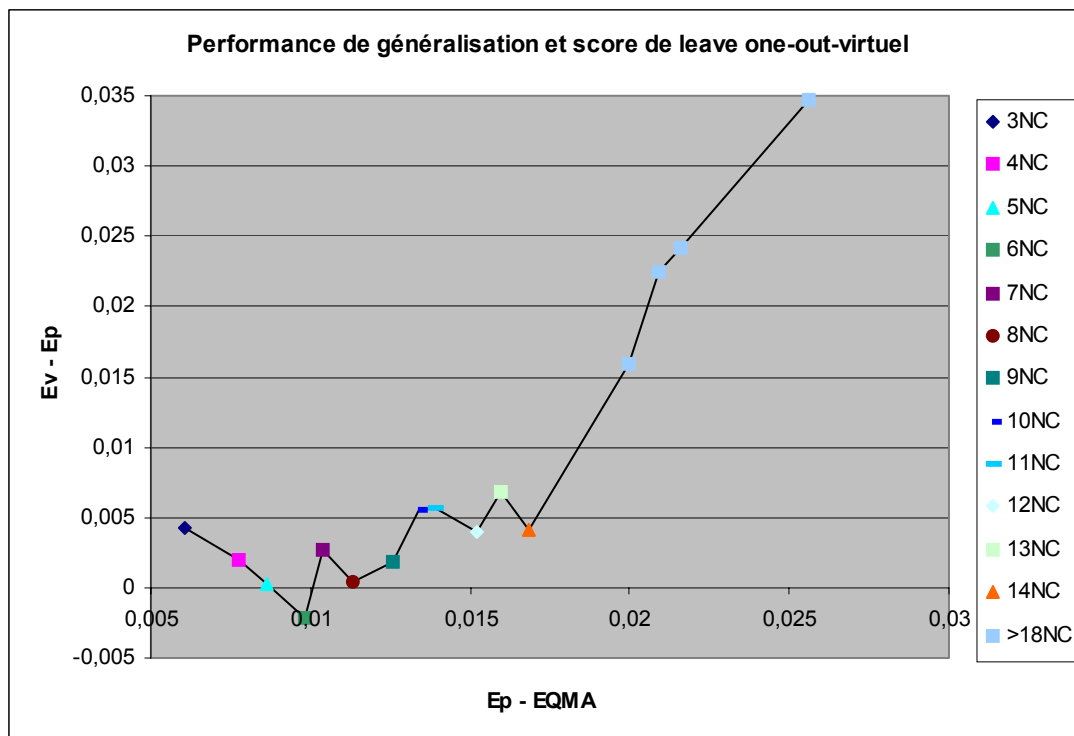


Figure 110 : Relation entre les écarts $E_p - EQMA$ et $E_v - E_p$

4.3.2.3 Stratégie de sélection du modèle

Comme précédemment, on a intérêt dans un premier temps à sélectionner les modèles pour lesquels le score de leave-one-out virtuel E_p est le plus faible. Si plusieurs modèles ont une valeur de E_p quasiment égale, on a alors intérêt à conserver celui pour lequel la valeur de $E_p - EQMA$ est la plus faible. Enfin, entre deux modèles équivalents en termes de E_p et $E_p - EQMA$ on choisit bien sûr le moins complexe (celui qui a le moins de neurones cachés).

4.3.2.4 Résultats pour un modèle à 13 neurones cachés

Le tableau de la présente les résultats obtenus après l'apprentissage. Là encore, l'apprentissage du réseau a été moins efficace que dans le cas des modèles de pantalons (cf. 4.2.2.4). En effet, le rapport SS_R / SS_T vaut 0,0788 alors qu'il était égal à 0,222 précédemment. En revanche, le score de leave-one-out virtuel E_p est proche de E_v on dispose tout de même d'une bonne estimation de la performance de généralisation du modèle.

	Ensemble d'apprentissage	Ensemble de validation
SS_T	2,5747	2,6880
SS_R	0,2029	0,2443
$EQMA$	0,778 (en% d'efficience)	
E_p	0,818 (en% d'efficience)	
E_v		0,854 (en% d'efficience)

Figure 111 : Performances du modèle à 13 neurones cachés

On constate malgré tout que les performances de généralisation sont très proches en valeurs absolues entre les deux types de modèles :

- E_p : 0,777% pour les pantalons et 0,818% pour les chemises
- E_v : 0,807% pour les pantalons et 0,854% pour les chemises

En effet, on arrive moins bien à estimer les variations d'efficience des modèles de chemise, mais comme celles-ci sont, en moyenne, plus faibles que celles des placements des pantalons, les erreurs absolues sur les estimations d'efficience sont comparables en moyenne.

Le graphique de la **Figure 112** regroupe les placements de l'ensemble de validation selon leur erreur de prédiction.

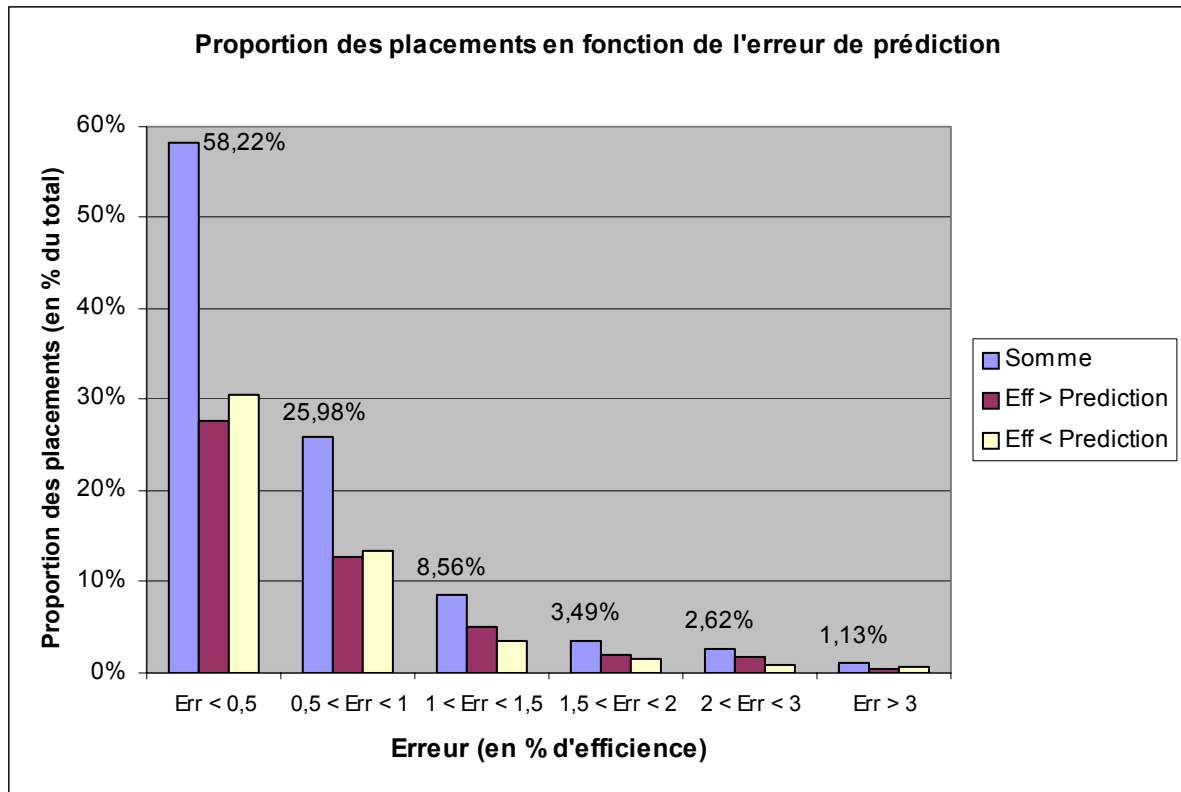


Figure 112 : Regroupement des placements par erreur de prédiction similaire

En résumé :

- Dans 84,2% des cas, l'erreur de prédiction est inférieure en valeur absolue à 1%.
- Dans 3,75% des cas, l'erreur de prédiction est supérieure en valeur absolue à 2%.

Le pourcentage de placements pour lesquels l'erreur est inférieure à 1% est légèrement supérieur dans le cas des chemises (84,2% contre 82,08% cf. 4.2.2.4). En revanche, le pourcentage de placements pour lesquels l'erreur est supérieure à 2% est beaucoup plus important (3,75% contre 2,25%).

Chapitre 5 : Conclusion

5.1 Choix des réseaux de neurones

Le choix des réseaux de neurones comme outil statistique semble particulièrement bien adapté au problème de l'estimation de l'efficacité des placements. En effet, la non-linéarité par rapport à leurs paramètres des modèles neuronaux permet, en plus d'apporter la propriété d'approximation parcimonieuse, de s'adapter plus facilement aux brusques variations éventuelles de l'efficacité des placements.

5.1.1 Architecture

Notre but est d'avoir une approximation de l'efficacité d'un placement en fonction de ses caractéristiques. On cherche donc à estimer le mieux possible la fonction de régression de l'efficacité en fonction des variables d'entrées que l'on juge pertinentes. On utilise pour cela les données expérimentales dont on dispose, on réalise alors ce qui est appelé dans [Dreyfus 2002] une modélisation « boîte noire ». L'architecture retenue pour ce type d'utilisation est représentée sur la **Figure 63**. C'est un réseau de neurones non bouclé, contenant une couche de neurones cachés à fonction d'activation sigmoïde. Lorsque le nombre de variables d'entrées n est fixé, la complexité du modèle dépend alors uniquement du nombre de neurones N_c dans la couche cachée.

5.1.2 Apprentissage supervisé

La phase d'apprentissage permet d'ajuster les poids du réseau aux données expérimentales dont on dispose. Pour cela, on choisit de minimiser l'erreur quadratique entre l'efficacité réelle et la sortie du réseau pour tous les exemples. Contrairement aux modèles linéaires par rapport à leurs paramètres, il n'existe pas pour les réseaux de neurones utilisés de méthode permettant de calculer directement les poids pour lesquels l'erreur quadratique est minimale. On utilise donc la méthode itérative de Levenberg-Marquardt qui permet de minimiser

l'erreur quadratique en calculant une estimation du gradient et de la matrice hessienne de cette erreur en fonction des poids du réseau.

5.1.3 Variables d'entrées

Le choix des variables d'entrées est déterminant dans la conception d'un modèle de prédiction d'efficience. La première phase consiste à retenir des variables susceptibles d'avoir une influence sur l'efficience des placements. Parmi les paramètres influençant l'efficience, on distingue des paramètres globaux valables quel que soit le type de placements et des paramètres spécifiques à chaque type de placements (chemises, pantalons...). Ensuite on ne retient que les variables qui sont réellement pertinentes pour estimer l'efficience des placements. On utilise pour cela la méthode du descripteur sonde décrite dans [Stoppiglia 1998].

5.1.4 Choix du modèle

Lorsque le nombre de variables d'entrées est fixé, choisir un modèle revient à choisir le nombre de neurones dans la couche cachée. Grâce à la propriété d'approximation universelle des réseaux de neurones, on peut rendre l'erreur quadratique arbitrairement petite sur l'ensemble d'apprentissage en augmentant le nombre de neurones cachés. Cependant, on court alors le risque d'obtenir un modèle surajusté qui aura une mauvaise performance de généralisation. Le score de leave-one-out virtuel E_p est donc le principal critère de sélection entre les modèles car il permet d'estimer cette performance de généralisation tout en utilisant la totalité des exemples disponibles dans l'ensemble d'apprentissage.

5.2 Résultats obtenus

5.2.1 Placements de pantalons

Les résultats obtenus pour les placements de pantalons sont très satisfaisants. En effet, pour l'exemple du paragraphe 4.2.2.4 par exemple, l'erreur résiduelle sur l'ensemble des placements n'ayant pas participés à l'apprentissage ne représente que 2,52% de la dispersion totale par rapport à la moyenne. La notion de colonne définie dans le paragraphe 3.2.2.1

semble être déterminante dans la qualité de ces résultats. Ceci est notamment vrai pour les placements mono-taille pour lesquels l'erreur résiduelle sur l'ensemble de validation représente seulement 1,54% de la dispersion totale par rapport à la moyenne pour l'exemple du paragraphe 4.2.1.2.

5.2.2 Placements de chemises

Les résultats obtenus pour les placements de chemises sont légèrement en retrait. En effet, pour l'exemple du paragraphe 4.3.2.4 par exemple, l'erreur résiduelle sur l'ensemble des placements n'ayant pas participé à l'apprentissage représente 9,09% de la dispersion totale par rapport à la moyenne. La notion de bonne configuration pour le positionnement des pièces définie dans le paragraphe 3.2.1.3 est moins efficace que la notion de colonne pour les pantalons. L'exemple du paragraphe 4.3.1.2 pour les placements mono-taille le confirme puisque l'erreur résiduelle sur l'ensemble de validation représente 13,65% de la dispersion totale par rapport à la moyenne.

Si l'on peut s'estimer satisfait de la précision obtenue dans le cas des pantalons, la moins bonne qualité des prédictions dans le cas des chemises peut être interprétée de plusieurs manières différentes :

1. Un paramètre influençant l'efficacité des placements de chemises n'a pas été trouvé
2. L'incertitude sur le résultat de Diamino Expert est plus grande dans le cas des chemises que dans le cas des pantalons

Il est également probable que ces deux facteurs sont conjugués.

Il faut néanmoins dire que même dans le cas des chemises, la précision obtenue est bien meilleure qu'avec la méthode actuelle où seul le nombre de tailles contenues dans le placement est pris en compte. En effet, le pourcentage de placements pour lesquels l'erreur de prédiction est inférieure à 1% est supérieur à 80% dans le cas des pantalons (cf. **Figure 102**) mais aussi dans le cas des chemises (cf. **Figure 112**). Enfin, un premier aperçu du bénéfice obtenu en combinant la construction des sections à l'aide du recuit simulé avec l'estimation de l'efficacité des nouveaux placements par les réseaux de neurones est présenté dans l'Annexe 4:.

5.3 Perspectives

L'étude décrite ici représente une première approche pour traiter le problème de la prédiction de l'efficacité des placements. Il reste encore des points à compléter avant d'obtenir une solution globale permettant d'avoir une estimation pour n'importe quel placement. Les principaux axes de développement sont décrits ci-dessous.

5.3.1 Contraintes sur le positionnement des pièces

Le positionnement des pièces lors de la création d'un placement n'est pas toujours libre. Les contraintes sur le positionnement des pièces peuvent être de plusieurs ordres :

- Retournements : A t-on le droit de retourner les pièces (verticalement ou horizontalement) ?
- Rotations : Peut-on faire tourner les pièces sur elles-mêmes, si oui avec quel pas, et jusqu'à quel angle ?
- Espacement : Doit-on préserver un espace entre deux pièces adjacentes, si oui quelle est la largeur minimale de cet espace ?
- Motif : Le tissu a-t-il un motif à respecter ?

Il est clair que toutes ces contraintes ont une influence sur l'efficacité des placements. En effet, plus celles-ci sont nombreuses et restrictives, moins on a de choix pour le positionnement des pièces et moins on a de chances d'obtenir une bonne efficacité.

Cependant, pour l'instant aucune évaluation n'a été faite sur les proportions de l'influence de chacune de ces contraintes sur l'efficacité des placements. En effet, tous les placements réalisés jusqu'à maintenant l'ont été avec les mêmes contraintes (Pas de rotation, pas de retournement, pas d'espacement obligatoire, pas de motif).

Intuitivement, on imagine que certaines d'entre elles vont être plus faciles à prendre en compte que les autres. Par exemple, pour l'espacement obligatoire entre les pièces, on peut facilement considérer que l'on a à placer des pièces légèrement plus grandes, estimer

l'efficacité d'un placement contenant ces pièces plus grandes (sans espace obligatoire) et ensuite estimer l'efficacité réelle du placement.

Pour d'autres, il est plus difficile de faire des conjectures sans une analyse statistique préalable.

5.3.2 D'autres types de placements

Dans cette étude, on s'est intéressé essentiellement aux placements de chemise et de pantalon. Il existe beaucoup d'autres types de placements.

Certains dans l'industrie de l'habillement :

- Costumes
- Vêtements d'enfant
- Lingerie

Mais il y en a également dans d'autres secteurs d'activité :

- Ameublement (fauteuils, canapés...)
- Automobile (sièges, airbags...)

Il reste à vérifier dans un premier temps que les paramètres globaux, définis dans le paragraphe 3.1, qui se sont avérés utiles pour les chemises et les pantalons sont également adaptés à la prédiction d'efficacité de ces nouveaux types de placements. D'autre part, y a-t-il pour chacun de ces nouveaux types de placements des paramètres spécifiques (cf. paragraphe 3.2) permettant d'améliorer la qualité des estimations ?

5.3.3 Regroupements entre types de placements ?

On a vu que les variations d'efficacité ne suivaient pas les mêmes règles pour les placements de chemise et les placements de pantalon. En est-il de même pour tous les types de placements évoqués ci-dessus ?

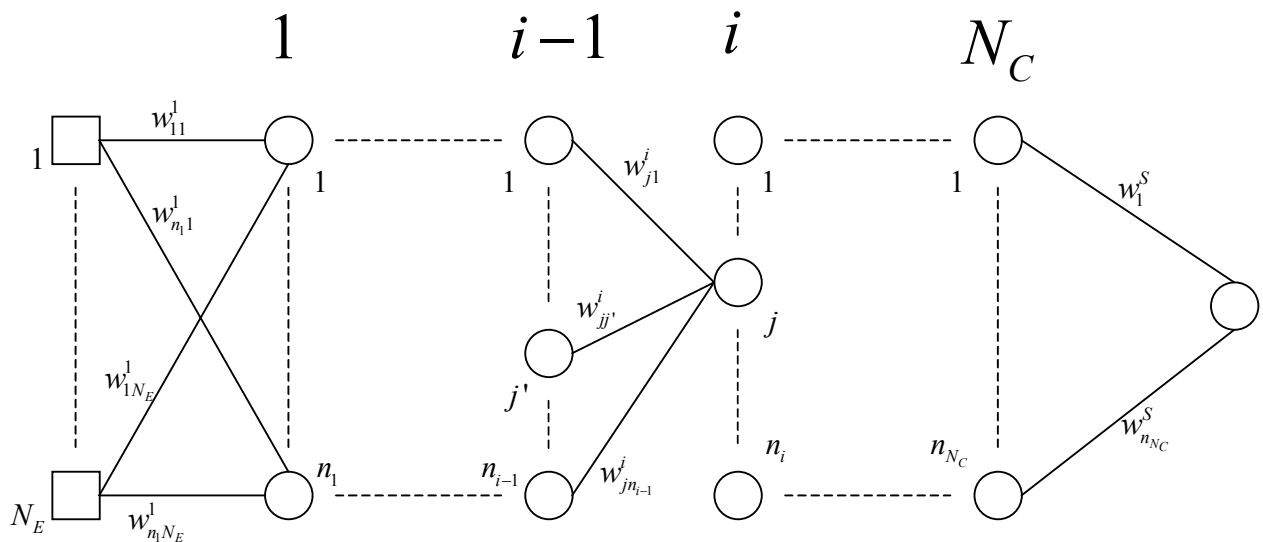
En effet, afin de ne pas multiplier sans fin le nombre de modèles statistiques différents, on a intérêt à limiter au maximum le nombre de paramètres spécifiques à chaque type de placements. L'idée consisterait alors à construire des grandes familles de placements à l'intérieur desquelles les variations d'efficience auraient les mêmes règles. Ces grandes familles pourraient alors contenir plusieurs types de placements différents.

Annexe 1: Calcul du gradient de la fonction de coût par rétropropagation

A1.1 Notations

On se place dans le cas d'un réseau à couches où tous les neurones cachés ont la même fonction d'activation. Le neurone de sortie est linéaire.

Le schéma de la **Annexe1 Figure 1** présente l'ossature d'un tel réseau :



Annexe1 Figure 1 : schéma du réseau utilisé

Soit \mathbf{x} le vecteur des entrées, soit \mathbf{w} le vecteur des poids du réseau

On pose $g(\mathbf{x}, \mathbf{w})$ = la sortie du réseau associée à \mathbf{x} et \mathbf{w}

On dispose d'un ensemble de vecteurs d'entrées: $\mathbf{x}^k / k \in \{1 \dots N\}$

pour lesquelles on connaît la grandeur qui nous intéresse: $y^k / k \in \{1 \dots N\}$

On cherche par exemple à minimiser en \mathbf{w} (cas de l'erreur quadratique): $J(\mathbf{w}) = \sum_{k=1}^N (y^k - g(\mathbf{x}^k, \mathbf{w}))^2$

$J(\mathbf{w}) = \sum_{k=1}^N (E^k)^2$ où $E^k = y^k - g(\mathbf{x}^k, \mathbf{w})$ est l'erreur de prédiction faite pour l'exemple k

Dans toute la suite, l'indice k indique que les entrées du réseau sont celles qui correspondent à l'exemple k . On définit :

Le couple (i, j) $\begin{cases} 1 \leq i \leq N_C \\ 1 \leq j \leq n_i \end{cases}$ le $j^{\text{ième}}$ neurone de la $i^{\text{ième}}$ couche cachée.

x_j^k $1 \leq j \leq N_E$ la valeur de la $j^{\text{ième}}$ entrée pour l'exemple k .

$w_{jj'}^i$ $\begin{cases} 1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1} \end{cases}$ le poids correspondant à la connexion entre les neurones $(i-1, j')$ et (i, j) .

w_j^S $1 \leq j \leq n_{N_C}$ le poids correspondant à la connexion entre le neurone de sortie et le neurone (N_C, j)

v_{ij}^k $\begin{cases} 1 \leq i \leq N_C \\ 1 \leq j \leq n_i \end{cases}$ le potentiel du neurone (i, j) .

$s_{ij}^k = f(v_{ij}^k)$ $\begin{cases} 1 \leq i \leq N_C \\ 1 \leq j \leq n_i \end{cases}$ la valeur de sortie du neurone (i, j) .

v_S^k le potentiel du neurone de sortie.

Comme le neurone de sortie est linéaire, on a : $g(\mathbf{x}^k, \mathbf{w}) = v_S^k$.

Pour minimiser la fonctionnelle J en \mathbf{w} , on définit le gradient de J par rapport au vecteur des

$$\text{poids } \mathbf{w} : \nabla J(\mathbf{w}) = \left(\left(\frac{\partial J(\mathbf{w})}{\partial w_{jj'}^i} \right)_{\substack{1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1}}}, \left(\frac{\partial J(\mathbf{w})}{\partial w_j^S} \right)_{1 \leq j \leq n_{N_C}} \right)$$

On pose $J^k(\mathbf{w}) = (E^k)^2 = (y^k - g(\mathbf{x}^k, \mathbf{w}))^2$ la contribution de l'exemple k à l'erreur totale.

On a :

$$J(w) = \sum_{k=1}^N J^k(w) \quad \text{et donc : } \nabla J(w) = \sum_{k=1}^N \nabla J^k(w) = \sum_{k=1}^N \left(\left(\frac{\partial J^k(w)}{\partial w_{ij}^j} \right)_{\substack{1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1}}}, \left(\frac{\partial J^k(w)}{\partial w_j^S} \right)_{1 \leq j \leq n_{N_C}} \right)$$

A1.2 Calcul de $\left(\frac{\partial J^k(w)}{\partial w_j^S} \right)_{1 \leq j \leq n_{N_C}}$

La fonction de coût partiel ne dépend du paramètre w_j^S que par l'intermédiaire du potentiel v_S^k du neurone (N_C, j) . On applique la formule de dérivée de fonctions composées et on obtient :

$$\frac{\partial J^k}{\partial w_j^S} = \frac{\partial J^k}{\partial v_S^k} \times \frac{\partial v_S^k}{\partial w_j^S}$$

Or, $J^k(w) = E^k(w)^2 = (y^k - g(x^k, w))^2 = (y^k - v_S^k)^2$ pour le neurone de sortie.

$$\text{Donc, } \frac{\partial J^k(w)}{\partial v_S^k} = -2(y^k - v_S^k) = -2E^k$$

On a : $v_S^k = \sum_{j'=1}^{n_{N_C}} w_{j'}^S \times s_{N_C j'}^k$, et donc : $\frac{\partial v_S^k}{\partial w_j^S} = s_{N_C j}^k$ c'est à dire la sortie du neurone (N_C, j)

Finalement, la dérivée de la fonction de coût partiel par rapport à un poids correspondant à une connexion avec le neurone de sortie du réseau est :

$$\frac{\partial J^k}{\partial w_j^S} = -2E^k \times s_{N_C j}^k$$

A1.3 Calcul de $\left(\frac{\partial J^k(w)}{\partial w_{ij}^j} \right)_{\substack{1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1}}}$

La fonction de coût partiel ne dépend du paramètre $w_{jj'}^i$ que par l'intermédiaire du potentiel v_{ij}^k du neurone (i, j) . On applique la formule de dérivée de fonctions composées et on obtient :

$$\frac{\partial J^k}{\partial w_{jj'}^i} = \frac{\partial J^k}{\partial v_{ij}^k} \times \frac{\partial v_{ij}^k}{\partial w_{jj'}^i}$$

On a : $v_{ij}^k = \sum_{j''=1}^{n_i} w_{jj''}^i \times s_{i-1j''}^k$ et donc : $\frac{\partial v_{ij}^k}{\partial w_{jj'}^i} = s_{i-1j'}^k$, c'est à dire la sortie du neurone $(i-1, j')$

On pose: $\frac{\partial J^k}{\partial v_{ij}^k} = \delta_{ij}^k$

De plus, $\forall 1 \leq p \leq n_{i+1}$ on a : $v_{i+1p}^k = \sum_{j=1}^{n_i} w_{pj}^{i+1} \times f(v_{ij}^k)$ c'est à dire que le potentiel v_{ij}^k apparaît dans chaque calcul des potentiels v_{i+1p}^k qui correspondent à la couche de neurones suivants.

On peut donc poser, par composition de fonctions :

$$\begin{aligned} \delta_{ij}^k &= \frac{\partial J^k}{\partial v_{ij}^k} = \sum_{p=1}^{n_{i+1}} \frac{\partial J^k}{\partial v_{i+1p}^k} \times \frac{\partial v_{i+1p}^k}{\partial v_{ij}^k} \\ &= \sum_{p=1}^{n_{i+1}} \delta_{i+1p}^k \times \frac{\partial \left(\sum_{j'=1}^{n_i} w_{pj'}^{i+1} \times f(v_{ij'}^k) \right)}{\partial v_{ij}^k} \\ &= \sum_{p=1}^{n_{i+1}} \delta_{i+1p}^k \times \left(w_{pj}^{i+1} \times f'(v_{ij}^k) \right) \\ &= f'(v_{ij}^k) \times \sum_{p=1}^{n_{i+1}} \delta_{i+1p}^k w_{pj}^{i+1} \end{aligned}$$

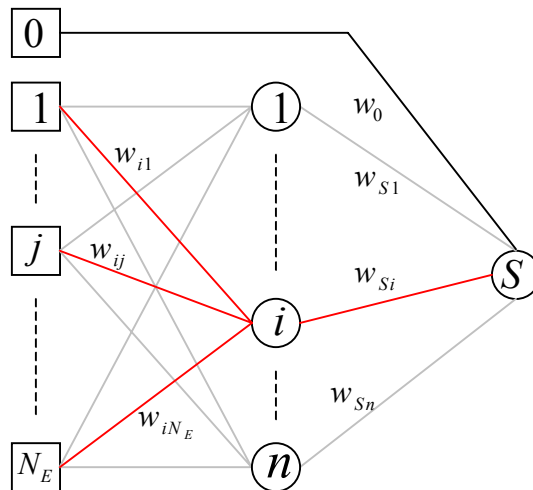
On voit alors facilement que l'on peut calculer de manière récursive tous les $(\delta_{ij}^k)_{\substack{1 \leq i \leq N_C \\ 1 \leq j \leq n_i}}$ en commençant par ceux de la dernière couche de neurones cachés (celle avant la sortie du réseau). C'est ce qu'on appelle la rétropropagation.

On obtient donc :
$$\frac{\partial J^k(w)}{\partial w_{ij}^i} = \delta_{ij}^k \times s_{i-1,j}^k \quad \begin{cases} 1 \leq i \leq N_C \\ 1 \leq j \leq n_i \\ 1 \leq j' \leq n_{i-1} \end{cases}$$

A1.4 Cas d'un réseau à une couche cachée

On se place dans le cas d'un réseau à une couche cachée avec un paramètre w_0 supplémentaire correspondant au biais que l'on veut appliquer à la sortie. La variable d'entrée 0 est constante égale à 1.

On obtient un réseau de cette forme :



Annexe1 Figure 2 : Réseau à une couche cachée et un neurone de sortie

On a :

$$\frac{\partial J^k}{\partial w_0} = -2E^k \times x_0$$

$$\frac{\partial J^k}{\partial w_{Si}} = -2E^k \times s_i^k \quad 1 \leq i \leq n$$

$$\frac{\partial J^k}{\partial w_{ij}} = -2E^k \times f'(v_i^k) w_{Si} x_j \quad \begin{cases} 1 \leq i \leq n \\ 1 \leq j \leq N_E \end{cases}$$

Dans ce cas on peut calculer directement toutes les composantes du gradient.

$$\begin{aligned} \frac{\partial J}{\partial w_0} &= -2x_0 \sum_{k=1}^N E^k \\ \frac{\partial J}{\partial w_{Si}} &= -2 \sum_{k=1}^N E^k f(v_i^k) \quad 1 \leq i \leq n \\ \frac{\partial J}{\partial w_{ij}} &= -2x_j w_{Si} \sum_{k=1}^N E^k f'(v_i^k) \quad \begin{cases} 1 \leq i \leq n \\ 1 \leq j \leq N_E \end{cases} \end{aligned}$$

Annexe 2: Minimisation de la fonction de coût avec la méthode de Levenberg-Marquardt

On veut minimiser en \mathbf{w} la fonction de coût : $J(\mathbf{w}) = \sum_{k=1}^N (y^k - g(\mathbf{x}^k, \mathbf{w}))^2$

On suppose que son gradient $\nabla J(\mathbf{w})$ est connu.

On définit la matrice hessienne $H(\mathbf{w})$ correspondant aux dérivées secondes de J par rapport aux poids.

Le développement de Taylor de la fonction J au point \mathbf{w} au voisinage du minimum \mathbf{w}^* permet d'écrire : $\mathbf{w}^* = \mathbf{w} - H(\mathbf{w}^*)^{-1} \times \nabla J(\mathbf{w})$ (sous réserve que la matrice H soit inversible).

La méthode de Levenberg-Marquardt s'inspire de cette formule et consiste à l'itération i de l'algorithme à modifier le vecteur des poids selon la règle suivante :

$$\mathbf{w}(i) = \mathbf{w}(i-1) - [H(\mathbf{w}(i-1)) + \mu_i I]^{-1} \nabla J(\mathbf{w}(i-1))$$

On calcule le terme général de $H(\mathbf{w})$:

$$\begin{aligned} H(\mathbf{w})_{ij} &= \frac{\partial^2 \left(\sum_{k=1}^N (E^k)^2 \right)}{\partial w_i \partial w_j} = \sum_{k=1}^N \frac{\partial^2 \left((E^k)^2 \right)}{\partial w_i \partial w_j} \\ &= \sum_{k=1}^N \frac{\partial \left(\frac{\partial \left((E^k)^2 \right)}{\partial w_j} \right)}{\partial w_i} = \sum_{k=1}^N \frac{\partial \left(2E^k \frac{\partial E^k}{\partial w_j} \right)}{\partial w_i} \\ &= 2 \sum_{k=1}^N \left(\frac{\partial E^k}{\partial w_i} \frac{\partial E^k}{\partial w_j} + E^k \frac{\partial^2 E^k}{\partial w_i \partial w_j} \right) \end{aligned}$$

E^k est l'erreur de prédiction pour l'exemple k .

Le deuxième terme étant proportionnel à l'erreur on choisit de le négliger.

On obtient alors une matrice symétrique définie positive \tilde{H} qui est une approximation de la matrice hessienne.

On définit $\left(\frac{\partial E^k}{\partial \mathbf{w}}\right)$ le vecteur des dérivées partielles de l'erreur sur l'exemple k par rapport aux poids du réseau.

$$\text{On obtient alors : } \tilde{H}(\mathbf{w}) = 2 \sum_{k=1}^N \left(\frac{\partial E^k}{\partial \mathbf{w}}\right) \times \left(\frac{\partial E^k}{\partial \mathbf{w}}\right)^T$$

La règle exacte de modification des poids est donc :

$$w(i) = w(i-1) - \left[\tilde{H}(w(i-1)) + \mu_i I \right]^{-1} \nabla J(w(i-1))$$

La matrice \tilde{H} est symétrique définie positive donc $\tilde{H}(w(i-1)) + \mu_i I$ l'est également. On peut alors l'inverser assez facilement par la méthode de Cholesky.

Annexe 3: Définition de l'indice d'éloignement d'un placement

A3.1 Notations et définition

Le tableau ci-dessous représente un exemple de placement.

Le modèle contient 10 tailles différentes, (du 36 au 54) et le placement contient 5 tailles (1*36 ; 1*42 ; 2*48 ; et 1*54)

Tailles	36	38	40	42	44	46	48	50	52	54
Indices tailles	1	2	3	4	5	6	7	8	9	10
Tailles placées	1			1			2			1

Annexe3 Figure 1 : Exemple de placement

On pose: N = le nombre de tailles différentes dans le vêtement (ici $N = 10$)

n = le nombre de tailles dans le placement (ici $n = 5$)

On définit: $\{x_i\}_{1 \leq i \leq n}$ = indice de la $i^{\text{ième}}$ taille du placement

avec $1 \leq x_1 \leq \dots \leq x_i \leq x_{i+1} \leq \dots \leq x_n \leq N$

Avec ces notations, le placement précédent (noté P) peut s'écrire : $P = (1, 4, 7, 7, 10)$

On définit alors l'éloignement entre les pièces par la formule : $\delta_n^N(P) = \sum_{i=1}^{n-1} \left(\sum_{j=i+1}^n (x_j - x_i) \right)$

Ce qui correspond à la somme des éloignements entre toutes les pièces prises 2 à 2.

A3.2 Recombinaison des termes

On cherche à simplifier cette écriture :

$$\begin{aligned}
\delta_n^N(P) &= \sum_{i=1}^{n-1} \left(\sum_{j=i+1}^n (x_j - x_i) \right) \\
&= \sum_{i=1}^{n-1} \left(\sum_{j=i+1}^n x_j \right) - \sum_{i=1}^{n-1} (n-i) x_i \\
&= \sum_{j=2}^n \left(\sum_{i=1}^{j-1} x_j \right) - \sum_{i=1}^{n-1} (n-i) x_i \quad (\text{en inversant les indices de sommation}) \\
&= \sum_{j=2}^n (j-1) x_j - \sum_{i=1}^{n-1} (n-i) x_i \\
&= (n-1)(x_n - x_1) + \sum_{i=2}^{n-1} (2i - n - 1) x_i \\
&= (n-1)(x_n - x_1) + \sum_{i=2}^{\lfloor \frac{n}{2} \rfloor} (2i - n - 1) x_i + \sum_{i=\lfloor \frac{n}{2} \rfloor + 1}^{n-1} (2i - n - 1) x_i \\
&= (n-1)(x_n - x_1) + \sum_{i=2}^{\lfloor \frac{n}{2} \rfloor} (2i - n - 1) x_i + \sum_{j=2}^{n - \lfloor \frac{n}{2} \rfloor} (n - 2j + 1) x_{n+1-j} \quad (\text{en posant } j = n + 1 - i)
\end{aligned}$$

Si n est pair, $n - \lfloor \frac{n}{2} \rfloor = \lfloor \frac{n}{2} \rfloor = \frac{n}{2}$ et donc :

$$\begin{aligned}
\delta_n^N(P) &= (n-1)(x_n - x_1) + \sum_{i=2}^{\frac{n}{2}} (n - 2i + 1) (x_{n+1-i} - x_i) \\
&= \sum_{i=1}^{\frac{n}{2}} (n - 2i + 1) (x_{n+1-i} - x_i)
\end{aligned}$$

Si n est impair, $n - \lfloor \frac{n}{2} \rfloor = \lfloor \frac{n}{2} \rfloor + 1$ et donc :

$$\begin{aligned}
\delta_n^N(P) &= (n-1)(x_n - x_1) + \sum_{i=2}^{\lfloor \frac{n}{2} \rfloor} (n-2i+1)(x_{n+1-i} - x_i) + \left(n-2 \left(\left\lfloor \frac{n}{2} \right\rfloor + 1 \right) + 1 \right) x_{n+1-\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right)} \\
&= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n-2i+1)(x_{n+1-i} - x_i) + \left(n - \left(2 \left\lfloor \frac{n}{2} \right\rfloor + 1 \right) \right) x_{\left\lfloor \frac{n}{2} \right\rfloor + 1} \\
&= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n-2i+1)(x_{n+1-i} - x_i) \quad \left(n - \left(2 \left\lfloor \frac{n}{2} \right\rfloor + 1 \right) = 0 \quad \forall n \text{ impair} \right)
\end{aligned}$$

Finalemment, $\delta_n^N(P) = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (n-2i+1)(x_{n+1-i} - x_i) \quad \forall n$

Cette formulation présente l'avantage de ne faire apparaître qu'une seule fois les termes x_i dans la sommation.

A3.3 Calcul de l'indice

$\delta_n^N(P)$ est minimum ($=0$) si tous les $(x_{n+1-i} - x_i)_{1 \leq i \leq \lfloor \frac{n}{2} \rfloor}$ sont nuls c'est à dire si $x_1 = x_n$.

On retrouve le fait que l'éloignement entre les tailles est nul si le placement ne contient qu'une seule taille du vêtement (éventuellement répétées plusieurs fois).

A l'inverse, $\delta_n^N(P)$ est maximum si : $\begin{cases} x_{n+1-i} = N \\ x_i = 1 \end{cases} \quad \forall 1 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor$

Cela correspond, pour un placement contenant n tailles à ce que :

On place $\left\lfloor \frac{n}{2} \right\rfloor$ fois la première taille

On place $\left\lfloor \frac{n}{2} \right\rfloor$ fois la dernière taille (si n est impair le fait de déplacer la taille correspondant à l'indice $\left\lfloor \frac{n}{2} \right\rfloor + 1$ ne modifie pas la valeur de $\delta_n^N(P)$)

On a alors :

$$\begin{aligned}
 \delta_n^N(P) &= \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} (n-2i+1)(N-1) \\
 &= (N-1) \left\{ \left\lfloor \frac{n}{2} \right\rfloor (n+1) - 2 \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} i \right\} \\
 &= (N-1) \left\{ \left\lfloor \frac{n}{2} \right\rfloor (n+1) - 2 \frac{\left\lfloor \frac{n}{2} \right\rfloor \times \left(\left\lfloor \frac{n}{2} \right\rfloor + 1 \right)}{2} \right\} \\
 &= (N-1) \left\lfloor \frac{n}{2} \right\rfloor \left(n - \left\lfloor \frac{n}{2} \right\rfloor \right) \quad \left(= \delta \max_n^N = \begin{cases} (N-1) \frac{n^2}{4} & \text{si } n \text{ est pair} \\ (N-1) \frac{n^2-1}{4} & \text{si } n \text{ est impair} \end{cases} \right)
 \end{aligned}$$

On veut que l'indice d'éloignement ne dépendent ni du nombre de tailles contenues dans placement ni du nombre de tailles différentes du modèle. On pose alors :

$I_{\text{éloignement}}(P) = \frac{1}{\delta \max_n^N} \delta_n^N(P)$ ainsi l'indice est compris entre 0 et 1 quelque soit le placement ou le modèle.

Annexe 4: Exemple d'utilisation de la prédiction d'efficience

A4.1 Résolution d'un bordereau de coupe

La figure ci-dessous représente un exemple de bordereau de coupe. Le modèle à découper est un modèle de pantalon (le modèle B) comportant 10 tailles différentes. Les 3 tissus du bordereau sont disponibles en laize 148cm.

				trouser_fl											
				v											
				88	90	92	94	98	102	106	110	114	118		
1	1	rouge	148,0 cm	5	17	23	47	20	12	26	17	3	6	176	
2		jaune	148,0 cm	9	14	16	31	35	21	18	23	15	2	184	
3		bleu	148,0 cm	8	24	19	28	33	21	17	16	6	8	180	
				22	55	58	106	88	54	61	56	24	16	540	

Annexe4 Figure 1 : Bordereau à résoudre

On veut résoudre ce bordereau avec des placements contenant jusqu'à trois tailles.

On connaît l'efficience moyenne des placements de ce modèle pour chaque nombre de tailles :

- 1 taille : 74,68%
- 2 tailles : 81,72%
- 3 tailles : 83,08%

On a également à disposition un réseau de neurones pour lequel l'apprentissage a été réalisé avec un ensemble de 1615 placements contenant ce modèle.¹

¹ Aucun placement de l'ensemble d'apprentissage n'est en laize 148cm. Les placements qui serviront à résoudre ce bordereau ne sont donc pas dans l'ensemble d'apprentissage de réseau de neurones.

On résout ce bordereau de différentes manières.

A4.2 Résolution avec Optiplan

Optiplan résout ce bordereau en utilisant 13 nouveaux placements, lors de la résolution, les quantités de tissus nécessaires sont estimées en utilisant les tables d'efficience. Ensuite, on réalise les placements proposés par Optiplan et l'on peut alors calculer les quantités réelles de tissu nécessaires. Les résultats obtenus sont présentés dans le tableau suivant :

Placements	Nombre	Quantité tissu estimée (m)	Quantité tissu réelle (m)	Différence
1 taille	7	76,06	78,05	1,99
2 tailles	3	181,67	185,45	3,78
3 tailles	3	518,13	523,37	5,24
Total	13	775,85	786,86	11,01

Annexe4 Figure 2 : Résultats obtenus avec Optiplan

La résolution avec les tables d'efficience a sous-estimé la quantité de tissu nécessaire pour résoudre le bordereau (l'efficience des placements a été surestimée en moyenne).

A4.3 Résolution avec recuit simulé et les tables d'efficience

L'algorithme utilisant le recuit simulé résout ce bordereau en utilisant 14 nouveaux placements, lors de la résolution, les quantités de tissus nécessaires sont estimées en utilisant les tables d'efficience. Ensuite, on réalise les placements proposés et l'on peut alors calculer les quantités réelles de tissu nécessaires. Les résultats obtenus sont présentés dans le tableau suivant :

Placements	Nombre	Quantité tissu estimée (m)	Quantité tissu réelle (m)	Différence
1 taille	4	16,39	16,93	0,54
2 tailles	1	21,37	23,25	1,88
3 tailles	9	729,44	741,28	11,84
Total	14	767,20	781,46	14,26

Annexe4 Figure 3 : Résultats obtenus avec le recuit simulé et les tables d'efficience

La résolution avec le recuit simulé et les tables d'efficacité donne en théorie une très bonne solution (767m estimés au lieu de 775m pour Optiplan), mais la quantité de tissu nécessaire a été sous-estimée de manière plus importante encore que précédemment, et le gain réel par rapport à Optiplan est inférieur à 1% (781m au lieu de 786m).

A4.4 Résolution avec recuit simulé et prédiction d'efficacité

L'algorithme utilisant le recuit simulé résout ce bordereau en utilisant 13 nouveaux placements, lors de la résolution, les quantités de tissus nécessaires sont estimées en utilisant les prédictions d'efficacité obtenues à l'aide du réseau de neurones. Ensuite, on réalise les placements proposés et l'on peut alors calculer les quantités réelles de tissu nécessaires. Les résultats obtenus sont présentés dans le tableau suivant :

Placements	Nombre	Quantité tissu estimée (m)	Quantité tissu réelle (m)	Différence
1 taille	4	14,23	14,17	-0,06
2 tailles	1	24,18	24,28	0,10
3 tailles	8	736,08	738,36	2,28
Total	13	774,49	776,81	2,32

Annexe4 Figure 4 : Résultats obtenus avec recuit simulé et prédiction d'efficacité

La quantité de tissu estimée en utilisant les estimations du réseau de neurones est moins bonne que celle obtenue avec le recuit simulé et les tables d'efficacité. En revanche, la quantité réelle de tissu nécessaire est très proche de cette estimation, cette solution est bien la meilleure des 3 :

	Optiplan (Table)	Recuit + Table d'efficacité	Recuit + Prédictions d'efficacité
Quantité tissu nécessaire (m)	786,86	781,46	776,81
Gain (%)	0,00%	0,69%	1,29%

Annexe4 Figure 5 : Résumé des solutions obtenues

Bibliographie

- [Bounsaythip1998] : C. Bounsaythip. *Algorithmes Heuristiques et Evolutionnistes : Application à la Résolution du Problème de Placement de Formes Irrégulières*. Thèse de l'USTL, NO: 2336, soutenue le 9 Octobre 1998. Disponible sur Internet à l'adresse : <http://www.vtt.fi/tte/staff/bon/thesis/these.html>
- [Dreyfus 2002]: Gérard Dreyfus, J. M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria, L.Hérault. Réseaux de neurones – Méthodologie et applications. Paris : Eyrolles, 2002, 386 p.
- [Li1994] : Z. Li, *Compaction Algorithms for Non-convex Polygons and Their Applications*, PhD thesis in Computer Science, Div. of Applied Sciences, Harvard University, Cambridge, MA, May 1994.
- [Milenkovic1991] : V. J. Milenkovic, K.M. Daniels and Z. Li. Automatic Marker Making. in *Proceedings of the 3rd Canadian Conference on computational Geometry*, ed. T. Shermer, Simon Fraser University, Vancouver, B.C., August 6-10, 1991, pp. 243-246.
- [Monari 1999] : Gaetan Monari
Sélection de modèles non linéaires par leave-one-out. Etude théorique et application des réseaux de neurones au procédé de soudage par points. Thèse de Doctorat de l'Université Pierre et Marie Curie - Paris VI (Novembre 1999). Disponible sur Internet à l'adresse : <http://www.neurones.espci.fr/>

- [Oussar 1998] : Yacine Oussar
Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus. Thèse de Doctorat de l'Université Pierre et Marie Curie - Paris VI (Juillet 1998). Disponible sur Internet à l'adresse : <http://www.neurones.espci.fr/>
- [Stoppiglia 1998] : Hervé Stoppiglia
Méthodes statistiques de sélection de modèles neuronaux - applications financières et bancaires. Thèse de Doctorat de l'Université Pierre et Marie Curie - Paris VI (Décembre 1997). Disponible sur Internet à l'adresse : <http://www.neurones.espci.fr/>

OPTIMISATION DU TRAITEMENT DE L'ORDRE DE FABRICATION DANS L'INDUSTRIE TEXTILE

Résumé :

Dans l'industrie textile un placement est un ensemble d'articles découpés ensemble. Un plan de production est un ensemble de placements découpés plusieurs fois permettant de satisfaire un ordre de fabrication.

La première partie est consacrée à la proposition d'un plan de production optimal pour chaque ordre de fabrication lorsque l'efficacité de chaque placement est supposée connue. Ce problème se modélise en un programme d'optimisation linéaire en nombres entiers. La complexité de ces problèmes ainsi que la volonté d'obtenir des solutions très vite conduisent à utiliser une méthode de résolution heuristique de type recuit simulé.

La seconde partie traite de l'estimation de l'efficacité des placements non encore réalisés en fonction de leurs caractéristiques (modèle, nombre de tailles...) L'étude de placements existants permet de construire un modèle statistique de prédiction d'efficacité en utilisant les propriétés d'approximation des réseaux de neurones.

Mots clés : *Industrie textile, Ordre de fabrication, Placement, Efficacité, Optimisation, Recuit simulé, Estimation, Réseaux de neurones.*

CUT ORDER PLANNING OPTIMISATION IN THE TEXTILE INDUSTRY

Abstract :

In the textile industry, a marker is a set of items that will be cut together. A section planning solution is a set of markers cut several times suitable for cutting the quantities of a manufacturing order.

The first part is dedicated to the search of an optimal section planning solution for every manufacturing order when the marker efficiency is known. This problem is modeled like an integer linear optimisation program. Due to the complexity of this kind of problems and the will to obtain solutions very fast, the problem is solved using an heuristic method: the simulated annealing.

The second part is dedicated to efficiency forecast for unmade markers depending on their characteristics (model, number of sizes...) A statistical efficiency forecast model is built, according to existing markers and using the approximation properties of neural networks.

Keywords : *Textile industry, Manufacturing order, Marker, Efficiency, Optimisation, Simulated annealing, Estimation, Neural networks.*