HAL
open science

Proposition d'une procédure de détermination de temps unitaire de pièces découpées

Hind El Haouzi, Philippe Thomas, André Thomas

To cite this version:

Hind El Haouzi, Philippe Thomas, André Thomas. Proposition d'une procédure de détermination de temps unitaire de pièces découpées. 8ème Conférence Internationale de Modélisation et Simulation, MOSIM'10, May 2010, Hammamet, Tunisie. pp.CDROM. hal-00525896

HAL Id: hal-00525896

https://hal.science/hal-00525896

Submitted on 13 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

8e Conférence Internationale de MOdélisation et SIMulation - MOSIM'10 - 10 au 12 mai 2010 - Hammamet - Tunisie
« Evaluation et optimisation des systèmes innovants de production de biens et de services »

PROPOSITION D'UNE PROCEDURE DE DETERMINATION DE TEMPS UNITAIRE DE PIECES DECOUPEES

Hind EL HAOUZI[1+2], Philippe THOMAS[1], André THOMAS[1]

(1)Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039), Nancy-Université, CNRS
ENSTIB 27 rue du merle blanc, B.P. 1041
88051 Epinal cedex 9 France
(2)Trane company, route de Champagne, 88130 Charmes
hind.elhaouzi@cran.uhp-nancy.fr.

RESUME : *L'entreprise TRANE est un groupe industriel qui fabrique des appareils de réfrigération. Dans ses processus de production elle utilise une cellule de découpe de pièces en tôle qu'elle doit gérer au mieux pour deux raisons : c'est un goulot d'étranglement et le rendement matière est, compte tenu du coût du métal, un objectif prioritaire. Les enjeux sont tels qu'il est donc devenu crucial de trouver le meilleur compromis entre performance (délai) et taux de chutes. Pour ce faire, il est impératif de connaitre les temps unitaires de chaque pièce car ceux-ci sont des paramètres des modèles de calcul des boucles kanban utilisées pour gérer les flux. Nous présenterons ici une procédure générique de détermination des temps unitaires de pièces découpées s'appuyant d'une part sur un modèle linéaire, et d'autre part sur une analyse statistique de données utilisées pour calculer les temps par pièce et dégager les paramètres du processus qui peuvent influencer d'une manière significative le temps estimé. Cette approche a été étendue par l'adjonction d'une méthode récursive afin d'affiner les résultats en prenant en compte toutes les nouvelles données disponibles. Forte de ces informations, l'entreprise pourra en perspective installer un nouveau système de pilotage de flux qui consiste à instrumenter des étiquettes kanban via des puces RFID pour une optimisation dynamique (en ligne) du temps, du taux de chutes et des flux.*

MOTS-CLES : *outils statistiques, Récursivité, approches adaptatives …*

1 INTRODUCTION

Dans le cadre de sa démarche d'amélioration continue, l'entreprise TRANE a pour but d'intégrer ses processus métier de bout en bout (automatisation de procédés manuels, rationalisation de processus et amélioration des délais de commande), pour ainsi trouver des moyens d'automatiser la réponse aux différentes opportunités de marché (inventaire en temps réel et prévisionnel, production juste à temps).

Dans ce contexte, le projet de centralisation sur le site de Golbey de l'ensemble des moyens de production tôlerie (cellule de production) des deux usines françaises a été initié, soit environ 25 poinçonneuses-grignoteuses et presses-plieuses. L'objectif de l'opération a été d'optimiser les flux par épaisseur pour mieux équilibrer les charges, rationaliser les en-cours en travaillant à la commande et non par lots figés. Cette cellule de production devant alimenter les deux usines de Charmes et Golbey, une gestion des approvisionnements a été mise en place en s'appuyant sur le système kanban. Un premier bilan a mis en évidence que des améliorations de ce système devaient être envisagées, en particulier en installant un mode de gestion par kanbans dynamiques, le système initial montrant des limites du fait de la réduction des temps de cycle de fabrication et de la variété des produits constatée entre temps. La première étape de ce projet consiste donc à connaître les temps de découpe de chacune des pièces sur les machines (notés temps de pièce) pour pouvoir recalculer les paramètres des boucles kanban. Or, les pièces ne sont pas poinçonnées seules mais imbriquées [1] ensembles en fonction du plan de production pour des objectifs évidents de rendement matière. Cette opération d'imbrication repose sur des algorithmes différents, selon les technologies machines, pour optimiser en particulier les taux de chutes et le temps de découpe de la tôle (noté temps de tôle). Nous allons voir dans la suite de l'article l'impact de cette opération sur la détermination des temps de pièce.

Cette approche implique que les temps de pièce ne sont pas connus et seuls les temps de tôle sont enregistrés en même temps que sont enregistrées les caractéristiques de la tôle (largeur, longueur, …) et de l'imbrication réalisée (nombre et type de pièces imbriquées, …). La difficulté est donc d'extrapoler les temps de pièce à partir des temps de tôle en créant et en mettant au point l'algorithme adéquat pour traiter le système de données. Notons que l'entreprise possède une base de données

[1] L'imbrication consiste à trouver le rangement le plus économique possible pour un ensemble de pièces sur la feuille de tôle.

(BdD) portant sur 5 années d'exploitation et donc suffisante pour imaginer en extraire des données crédibles.

L'objectif de ce papier est donc de montrer quelles méthodes ont été mises en place pour déterminer ces temps de pièce automatiquement. Le travail présenté ici est assez opérationnel. Cependant la procédure décrite a été rendue la plus générique possible afin d'en faciliter la réutilisabilité. La mise en place d'une approche récursive d'adaptation des temps de pièce estimés a été nécessaire.

Ces temps de pièce, une fois connus, seront implémentés dans un système de simulation à événements discrets qui permettra d'évaluer des solutions de dimensionnement des boucles kanban prédéterminées par calculs (Tardif et Maaseidvaag, 2001).

La suite du texte est organisée comme suit : une description détaillée du processus et de la cellule concernée nécessaire à la compréhension du problème. Dans une troisième partie, l'opération d'imbrication sera détaillée puis suivra une description de la procédure générique proposée. Cette dernière sera ensuite détaillée selon l'approche nécessaire aux machines classiques, puis selon la méthode spécifique à la machine SALVA. Nous poursuivrons en décrivant les apports pour les perspectives de notre projet avant de conclure.

2 DESCRIPTION DES PROCESSUS CONCERNES :

Comme nous l'avons introduit précédemment, la cellule de découpe doit :

- alimenter les lignes d'assemblage de Charmes et de Golbey en tôlerie (soit environ 6 lignes d'assemblage),
- fabriquer 5000 à 6000 pièces / jour pour 3000 types différents,
- assurer une génération de programme dynamique selon les charges machines et le taux de chute (plusieurs gammes),
- gérer plusieurs formats et épaisseurs des tôles.

La figure 1 décrit cette cellule nommée « Front End ». Celle-ci répond à des besoins exprimés dynamiquement par des relances des kanban (représentant 80% des demandes) et le calcul de besoins nets des ordres de fabrication des pièces jugées critiques et nécessitant un suivi particulier. L'imbrication est donc réalisée à partir de ces besoins en commençant d'abord par les pièces critiques avec un objectif de maximisation de rendement matière. Les pièces seront alors découpées dans des formats de tôles différentes (selon le résultat de l'opération de l'imbrication), poinçonnées, puis pliées (Figure 2) sur une seule machine complexe « Salvagnini » dite SALVA ou sur des machines traditionnelles (ALMA) (figure 1) en fonction des capacités disponibles de celles-ci. Un atelier de peinture est finalement utilisé pour traiter certaines pièces qui le nécessitent.

Rappelons que les temps de pièce réels sont donc résultants de la fabrication effective des pièces sur les machines. Nous verrons ultérieurement que le système déjà en place n'est capable de fournir que le temps de tôle. Cette description met en évidence les points sensibles que sont les calculs des temps de pièce sur les machines traditionnelles et sur la machine SALVA. Pour dégager les facteurs influents sur ces temps de pièce (comprendre la relation entre le temps de tôle et les temps de pièce), il nous a paru nécessaire de présenter l'opération de l'imbrication pour les deux types de machines : machines traditionnelles et machine SALVA.

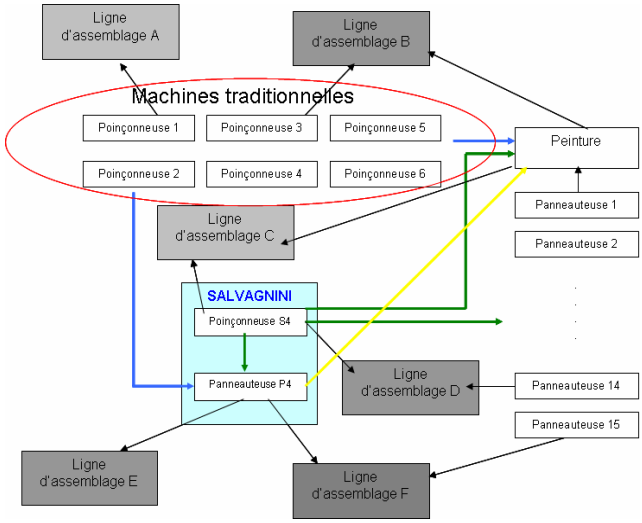


Figure 1 : La cellule Front-END

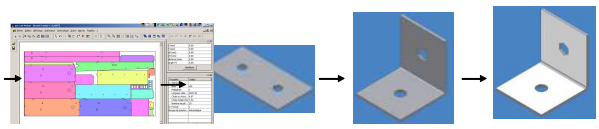


Figure 2 : Processus concerné par l'imbrication et le découpage/poinçonnage des pièces.

3 L'OPERATION DE L'IMBRICATION ET SON IMPACT SUR LE TEMPS

Dans la littérature, nous retrouvons cette notion d'imbrication sous le nom du Bin Packing. En effet, le Bin Packing consiste, comme son nom l'indique, à ranger *m* articles dans *n* boites de manière à minimiser le nombre de boites utilisées. Le problème classique se définit en une dimension, mais il existe de nombreuses variantes en deux ou trois dimensions. En pratique, les utilisateurs sont souvent contraints par le matériel de découpe disponible, i.e. la façon de procéder pour la réalisation d'un plan de découpe est différente. Les contraintes les plus souvent rencontrées sont (T. Saadi, 2008) :

- la contrainte guillotine : l'outil de coupe doit aller d'un bord à l'autre du rectangle ou de la bande à

découper. Ce type de découpe est aussi appelé découpe de bout à bout. On parle aussi du niveau de guillotine. Il s'agit d'une constante K qui limite le nombre de changements de la découpe,

– La contrainte d'orthogonalité : La découpe prend en considération l'orientation des pièces.

3.1 Le cas ALMA :

Pour cette machine, le logiciel propose une solution avec une découpe non-guillotine (figure 3) et sans la contrainte d'orthogonalité. En plus le déchargement manuel des tôles découpées vers le processus aval (poste de pliage) impose de garder une certaine distance entre les pièces à découper pour l'opération de « dégrapage ».

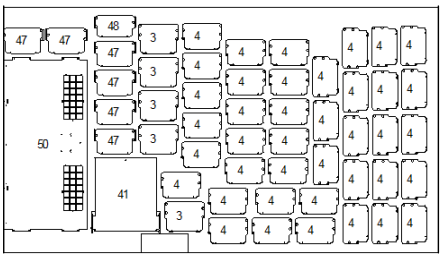


Figure 3 : Exemple d'imbrication réalisée pour une des machines traditionnelles.

Sur cette machine, la découpe se fait par « grignotage ». C'est-à-dire qu'un outil usine d'une manière particulière le tour de la pièce. Si tous les paramètres étaient constants, la détermination des temps pourrait se faire par calcul (périphérie * vitesse d'avance). Cependant d'autres paramètres influent (éléments additionnels de forme, encoche, rainure, chanfrein, …) sur le temps de pièce.

3.2 Le cas de SALVA :

La machine SALVA est une ligne flexible de poinçonnage-cisaillage-panneautage qui supporte 66 outils (poinçons) différents : la découpe se fait essentiellement par 2 cisailles. Cette machine comporte également un magasin de 12 formats de tôle. L'analyse réalisée sur le logiciel d'imbrication a permis de formaliser le type de découpe utilisé. Il s'agit du Two-dimensional Rectangular Multiple Stock Size *Cutting Stock Problem* (Cintra, *et al.*, 2008), avec la contrainte guillotine (figure 4).

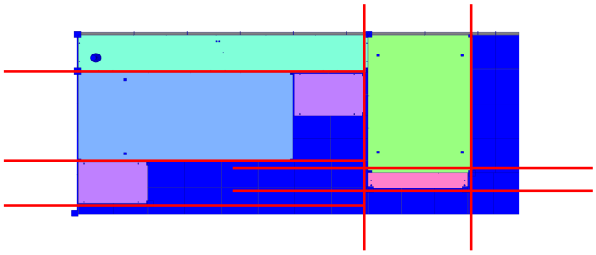


Figure 4 : Exemple d'imbrication réalisée pour la machine SALVA

Le niveau de guillotine est non limité et une contrainte d'orthogonalité est utilisée. Sur la figure 4 sont représentées les différentes guillotines utilisées pour la découpe de cette tôle (traits rouges).

Partant de ce résultat, plusieurs facteurs peuvent influer sur les temps de pièce. Entre autres, la géométrie de la pièce, l'orientation de la pièce (dépendante de la solution proposée par le logiciel), format de la tôle, les pièces « voisines », pour dégager la pièce… Ceci rend une analyse statistique pour la détermination des temps de pièce très difficile (voire infaisable).

4 PROCEDURE DE DETERMINATION DES TEMPS DE PIECE

Quelle que soit la machine utilisée, le processus de détermination des temps de pièce commence par les mêmes étapes :

4.1 Extraction des données

Extractions de toutes les pièces par tôle :

– extraire toutes les tôles lancées en production de ORACLE et CINCOM (respectivement les ERP de Charmes et de Golbey),

– relever les caractéristiques géométriques de chacune d'elles (longueur, largeur, nombre de plis, nombre de rotations, etc…).

Décomposer chaque tâche réalisée sur chaque machine :

– définir la cartographie du process

– trouver les tâches qui sont affectées à chaque tôle ou chaque chariot. Exemple (Tirer chariot de l'IPK (stock d'en-cours géré par kanban), appel du programme, charger tôle sur machine, …),

– trouver les tâches qui ne sont réalisées que quelques fois par jour. Exemple (Changement d'outils, changement de paquet de tôles, trier le dossier, …),

– préparer des feuilles de saisie.

Calcul des temps de tôle :

– mesurer la capabilité (Minitab),

– chronométrer les tâches,

– procéder à l'analyse des données (Minitab),

– calculer les temps de tôle.

4.2 Heuristique de calcul

La figure 5 présente un algorithme le plus générique possible pour le calcul des temps de pièce en fonction des contraintes de la machine. Ce calcul est différent selon que la découpe se fait selon une contrainte guillotine (SALVA) ou par grignotage (machine traditionnelle). Ces deux approches sont détaillées dans les chapitres suivants. L'heuristique E* est détaillée dans la partie

6 quand l'algorithme de programmation linéaire est présenté dans le chapitre 5. L'ensemble de cette procédure est programmée dans un outil spécifique.

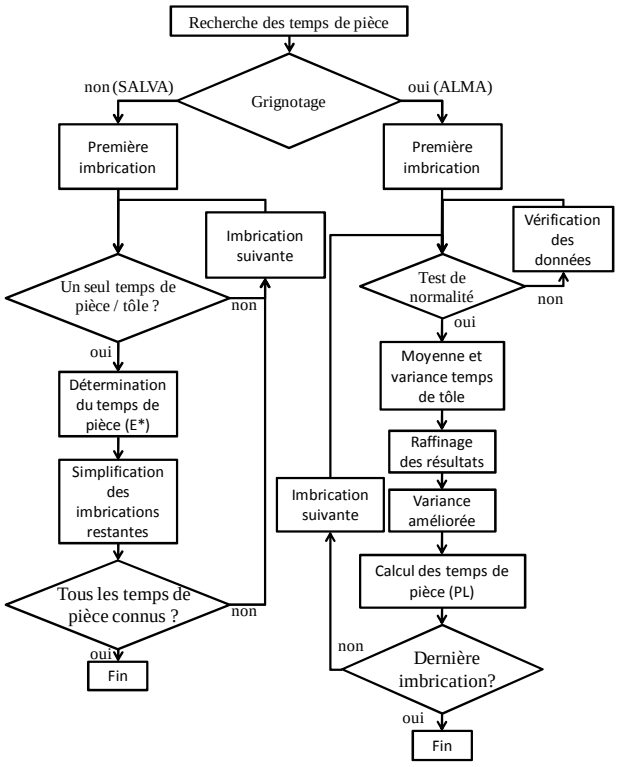


Figure 5 : Méthode générique de calcul des temps de pièce

5 APPLICATION DE LA DEMARCHE SUR ALMA

La première étape a été de vérifier si les temps de tôle réels correspondaient bien aux temps par tôle que le logiciel d'imbrication calculait. Pour cela des chronométrages ont été réalisés. Puis pour faire l'analyse comparative des données issues de l'échantillon de temps observés, un test de normalité a permis de confirmer l'hypothèse nulle « les données observées suivent bien une loi normale » de dispersion compatible. Le T test de Student a permi de valider l'hypothèse précédente et donc que les deux distributions ont les mêmes moyennes. Les résultats ont montré que les temps de tôle Alma sont donc acceptables à 95%.
En supposant que le modèle soit linéaire on a utilisé une régression simple pour formuler un premier modèle de temps de tôle (Voir encadré ci-dessous). Cette régression nous donne un modèle caractérisé par un R^2 = 82,1%, ce qui est correct dans le sens où un R^2 > 80% est généralement acceptable. Pour autant dans l'entreprise et sur les autres machines, les modèles sont caractérisés par un R^2 > 90%.

Le modèle linéaire du temps de pièce obtenu est donc le suivant :

Temps de tôle = 5,09 + 0,000033 Longueur + 0,000068 Largeur + 0,00665 Chanfreins + 0,0111 Encoches + 0,0149 Deformations + 0,000005 Evid. rect. + 0,000013 Evid. circ. + 0,00479 Trous + 0,0216 Formes simples + 0,000637 Evid. pré-def.

Avec: S = 0,944161 R-Sq = 82,1% R-Sq(adj) = 81,9%

Malgré la cohérence de ces résultats, on a cherché à obtenir un modèle plus précis sans pour autant ajouter des facteurs.

Ces résultats ont été raffinés par une régression de type surface de réponse, puis par l'usage d'un modèle « Full quadratic » qui utilise tous les facteurs mis au carré et toutes les interactions. Il a suffi alors de supprimer de l'analyse, les termes dont la valeur de P est supérieure à 0,05 et de relancer l'analyse. En répétant cette opération une ou deux fois, le modèle a été purgé des facteurs non influents

La conversion des modèles à la tôle au modèle à la pièce se faisait simplement en multipliant la constante par le ratio entre la surface de la pièce et la surface de la tôle.
Mais en procédant de la sorte le modèle quadratique montre d'importants disfonctionnements dû aux interactions entre les facteurs.
Pour remédier au problème, nous avons fais un compromis entre le modèle linéaire et le modèle Full Quadratic en identifiant les interactions importantes dans le modèle quadratique puis les ajouter au modèle linéaire simple.

En testant le nouveau modèle, de la même manière que précédemment, nous obtenons un R^2 = 86.2%.

Pour contrôler la fiabilité des modèles, nous avons éprouvé le système de calcul sur une semaine complète de production (figure 6).

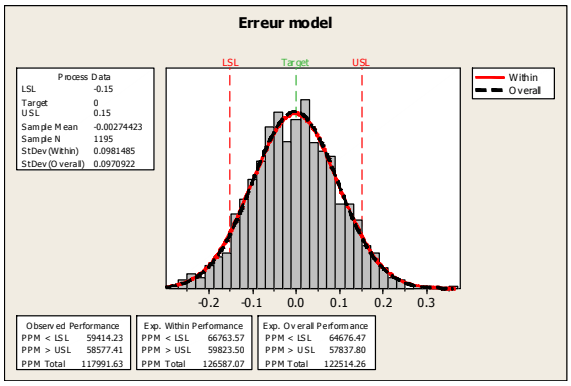


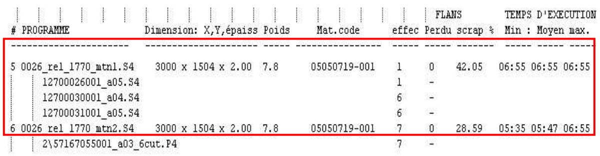
Figure 6 : test de validation des données calculées

Ce graphique montre la répartition des erreurs (ratio temps calculé/ temps réel, exprimé en pourcentage) sur une semaine de production. Cette répartition est normale, la moyenne est de -0,27%. On peut voir également que le

modèle admet 117 991 défauts par million d'opportunités. Concrètement, cela signifie que le pourcentage d'erreur à la pièce est compris entre -15% et 15% dans 88,3% des cas, ce qui est très largement acceptable pour l'usage escompté.

6 METHODE DE CALCUL DE TEMPS DE PIECE SUR SALVA

Sur cette machine, trois opérations sont possibles. Dans un premier temps, la tôle subit une opération de poinçonnage (la plaque est percée aux endroits prévus), puis elle est découpée en plusieurs pièces (opération de découpe). Une opération de pliage est ensuite appliquée sur chacune des pièces. Plusieurs pièces peuvent donc être fabriquées à partir d'une seule tôle. Comme nous l'avons dit précédemment un logiciel propre à la machine traite les données d'ORACLE et de CINCOM (les deux ERP) relatives aux relances Kanban (détermination des références pièces arrivant sur la machine) afin de trouver la bonne adéquation entre nombre de références par tôles et taux de chute. En effet, le taux de chute augmente quand le nombre de références par tôle diminue.



# PROGRAMME	Dimension: X,Y,épaiss	Poids	Mat.code	effec	FLANS Perdu	scrap %	TEMPS D'EXECUTION Min	: Moyen	max.
5 0026_rel_1770_mtn1.S4	3000 x 1504 x 2.00	7.8	05050719-001	1	0	42.05	06:55	06:55	06:55
12700026001_a05.S4				1	-				
12700030001_a04.S4				6	-				
12700031001_a05.S4				6	-				
6 0026_rel_1770_mtn2.S4	3000 x 1504 x 2.00	7.8	05050719-001	7	0	28.59	05:35	05:47	06:55
2\57167055001_a03_6cut.P4				7	-				

Figure 7 : Le fichier d'extraction des temps de production issus de la machine

Le logiciel permet d'obtenir les temps par tôle (et non par pièce). Ces données sont consignées dans des fichiers texte, listant l'ensemble des imbrications pour une période, avec pour chacune d'elle, la liste des pièces fabriquées (figure 7). La première ligne de ce fichier décrit la découpe d'une tôle. On y retrouve la référence de l'imbrication, les dimensions de la tôle, le code matière, le nombre de réalisations, le taux de déchet ainsi que les temps d'exécution moyen minimal et maximal qui sont ici les mêmes puisque une seule tôle de ce type est découpée. Les 3 lignes suivantes décrivent les pièces qui ont été imbriquées dans cette tôle. On constate ici que cette imbrication inclue 1 pièce 12700026001, 6 pièces 12700030001 et 6 pièces 12700031001. Plusieurs années de production sont ainsi archivées et exploitables pour extraire de la connaissance de ces données. Dans ce cadre, une première approche intuitive pourrait être d'essayer d'exploiter des stratégies d'extraction de connaissance basées sur l'intelligence artificielle telles que réseaux de neurones, réseaux Bayésiens, logique floue… Cependant, du fait du grand nombre de pièces traitées et de leur diversité, de la grande variabilité relative aux types de données disponibles pour chaque pièce (figure 11), et de la grande dispersion statistique de ces données due majoritairement à la réalisation de l'imbrication, ces approches ne semblent pas les plus pertinentes.

6.1 Détermination hors ligne des temps

A partir des données précédemment citées, (temps par tôle, type de pièce, …) L'algorithme suivant (figure 8) a été mis en œuvre.

La première étape est l'extraction de toutes les informations relatives à la découpe des tôles présentes dans la BdD modélisées sous forme d'équations linéaires :

$$\sum_{i=1}^{n} a_{ij}.t_{ij} = T_j \quad \text{avec } j = 1, \cdots, m \tag{1}$$

où T_j représente le temps de tôle (découpe) ou de l'imbrication j (parmi les m imbrications archivées dans la BdD), t_{ij} représente le temps de pièce (découpe) i de la tôle j quand a_{ij} représente la quantité de pièces i présentes dans la tôle j.

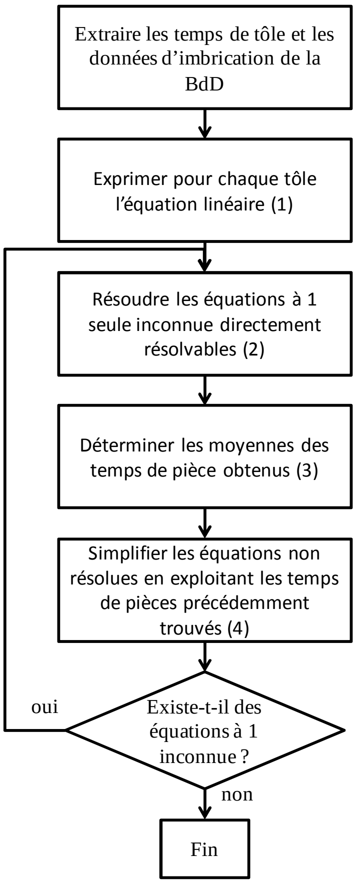


Figure 8 : Détail de l'algorithme E* de la figure 5

Dans cette équation (1), les T_j et a_{ij} sont connus et nous cherchons à déterminer les t_{ij} inconnus. Nous obtenons donc un système à m équations (m tôles) et n inconnues (n types de pièces). Nous pouvons tirer de la figure 7 un exemple de ces équations :

1 * 12700026001_a05.S4+6 * 12700030001_a04.S4+ 6 * 12700031001_a05.S4 = 6 minutes 55 secondes

Une fois tout ce système d'équations écrit, nous recherchons toutes les équations directement résolvables, c'est-à-dire toutes les équations à une seule inconnue t_{ij}, ce qui revient à dire toutes les tôles ne comprenant qu'un seul type de pièce :

$$t_{ij} = \frac{T_j}{a_{ij}} \qquad (2)$$

Pour une même pièce *i*, nous avons utilisé *l* équations correspondant à *l* tôles incluant uniquement des pièces *i*. De ce fait, nous pouvons déterminer un temps moyen pour la pièce *i* :

$$\overline{t_i} = \frac{\sum_{j=1}^{l} t_{ij}}{l} \qquad (3)$$

Ces *k* temps $\overline{t_i}$ ainsi déterminés, sont utilisés pour simplifier les équations non résolues dont les tôles comprennent certaines pièces dont le temps moyen est connu :

$$\sum_{i=1}^{n-k} a_{ij}.t_{ij} + \sum_{h=1}^{k} a_{hj}.\overline{t_i} = T_j \quad \text{avec } j = 1, \cdots, m' \qquad (4)$$

où *m'* est le nombre d'équations non encore résolues et *k* le nombre de temps moyen déjà trouvés. Parmi les *m'* équations restantes, certaines ne comprennent plus qu'une seule inconnue. Nous pouvons alors les résoudre, trouver d'autres temps moyen $\overline{t_i}$ qui simplifient d'autres équations (4) et boucler ainsi sur ces différentes étapes jusqu'à ce qu'il n'existe plus d'équations résolvables. Nous aurons ainsi estimé tous les temps déterminables.

L'outil de calcul développé a été utilisé pour un jeu de données d'une année composé de 12 fichiers mensuels et représentant plus de 22000 imbrications comportant près de 3000 pièces. Sur la base de ce jeu de données, l'outil a permis de déterminer le temps d'environ 76% des pièces produites par la machine.

Le modèle ne permet pas de trouver le temps de pièce pour toutes les pièces pour plusieurs raisons :

1. pour *n* inconnues nous avons en entrée *m* équations avec *m # n*,
2. le temps consacré à la chute pour dégager les pièces n'est pas pris en compte,
3. la propagation des erreurs n'est pas bien maitrisée,
4. les pièces révisées ou légèrement modifiées sont considérées comme différentes.

L'analyse de ces résultats sur minitab est schématisée dans la figure 9. Nous constatons que les temps ne suivent pas une loi normale (*P*<0,005) et que la moyenne est de 86 secondes (l'expert avait estimé celle-ci à 60s) avec un écart type grand.

Il est donc possible d'améliorer cette détermination des temps en prenant en compte les points 1 à 4 précédemment cités.

6.2 Amélioration du modèle

La source la plus importante d'erreur dans l'approche de détermination des temps précédemment décrite est due à la non prise en compte des temps de chute. En effet, la chute, qui peut représenter une proportion importante de la tôle à découper, doit être éliminée par grignotage (opération d'usinage par coupe alternative de la tôle), ce qui ne peut pas se faire instantanément.

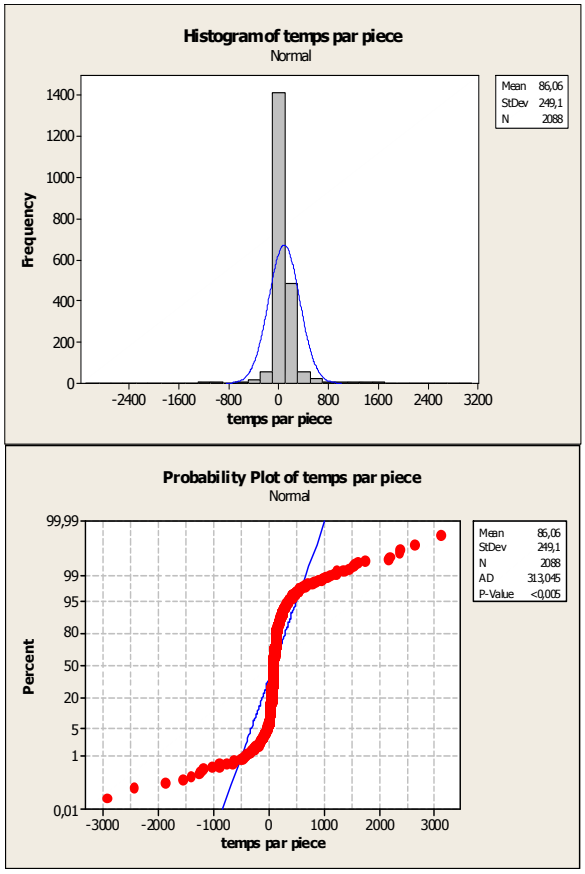


Figure 9 : Analyse des résultats du modèle de calcul de la machine Salva

nombre de la pièce 5715 4599	temps par tôle	scrap*surfacetôle
1	126	4,469162957
2	139	4,290297793
3	146	4,111432629
4	158	3,932567465
5	162	3,753702301
6	173	3,574837137
7	183	3,395971973
8	190	3,217106809
9	205	3,042289842
10	223	2,861982217
11	226	2,678045174
12	234	2,519932836
13	244	2.335995793

Tableau 1 : Extrait du tableau de 24 imbrications d'une même pièce

Afin d'estimer le temps de chute moyen en seconde par m^2, nous avons réalisé des tests d'imbrication sur 3 types de pièces parmi les plus représentatives en positionnant de 1 à 24 fois la même pièce dans une tôle, le reste de la tôle constituant la chute. En exemple, le tableau 1 présente 13 des 24 imbrications réalisées en plaçant pour

chaque tôle de 1 à 24 fois la pièce n° 5 715 4599. En supposant que le modèle soit linéaire, comme sur les machines traditionnelles, on a utilisé une régression simple pour formuler un premier modèle :

$$\begin{cases} T_j = 243 + 5,2.n_{pièce} - 28.s_{chute} \\ S = 3,94664 \quad R - S_q = 00,7\% \quad R - S_q(adj) = 99,7 \end{cases} \quad (5)$$

où $n_{pièce}$ représente le nombre de pièces identiques présentes dans la tôle, s_{chute} représente la surface de la chute dans la tôle et T_j le temps de tôle.

Cette régression nous donne un modèle caractérisé par un coefficient de corrélation $R^2 = 99,7\%$, ce qui est très bon.

Cependant, partant du constat que le temps total de la tôle est constitué uniquement du temps de chute additionné du temps consacré à la fabrication des pièces, nous avons décidé de modifier le modèle en utilisant une régression passant par l'origine. Ceci permet de partager le bruit entre les chutes et les pièces. Ce type d'approche est utilisé dans de nombreux problèmes pratiques, lorsque l'on étudie la relation qui peut exister entre 2 quantités Y et X, et que nous savons que si X = 0 alors Y = 0 elle est aussi utilisé pour les systèmes d'étalonnage.

Cette approche a conduit à la construction de 3 modèles en fonction des 3 types de pièces considérées :

$$\begin{cases} T_{j,1} = 14,5.n_{pièce_1} + 24,7.s_{chute} \\ T_{j,2} = 27,3.n_{pièce_2} + 25,7.s_{chute} \\ T_{j,3} = 34,9.n_{pièce_3} + 26,8.s_{chute} \end{cases} \quad (6)$$

où $n_{pièce_i}$ est le nombre de pièces de type *i* placées dans la tôle, et s_{chute} est la surface de la chute dans la tôle.

Nous considérons ces modèles (6) comme acceptable étant donné que les coefficients de corrélation obtenus sont du même ordre de grandeur que ceux obtenus pour (5). Ceci nous permet de déterminer un temps moyen par m^2 de chute :

$$\overline{t_c} = 25 \sec / m^2 \quad (7)$$

La prise en compte de ce temps de chute permet de remplacer l'équation (1) par :

$$\sum_{i=1}^{n} a_{ij}.t_{ij} + \left(scrap_j.S_j\right).\overline{t_c} = T_j \quad \text{avec } j = 1, \cdots, m \quad (8)$$

où $scrap_j$ représente le pourcentage de perte dans la tôle (chute) quand S_j représente la surface de la tôle.

La deuxième source d'erreur importante est la grande dispersion des temps de tôles pour des imbrications identiques. En effet, un certain nombre d'incidents peuvent intervenir sur la machine rendant certains temps de tôle aberrants. La difficulté est que les incidents à l'origine de cette dispersion ne sont pas connus. De ce fait, il est nécessaire d'éliminer toutes les équations, qui pour une même imbrication, conduisent à des temps de tôle trop variable. Le rejet d'une imbrication se fait par la relation :

$$\text{si} \begin{cases} \left(T_{\max} \geq T_{moy} + \beta.T_{moy}\right) \\ \text{et} \\ \left(T_{\min} \leq T_{moy} - \beta.T_{moy}\right) \end{cases} \text{alors rejet} \quad (9)$$

où T_{max} = max(T_j) ; T_{min}=min(T_j) et T_{moy} = moyenne(T_j) avec $j = 1 \ldots h$, et h est le nombre de tôles réalisant la même imbrication. β est un pourcentage choisi par expertise (ici 15%). Si une décision de rejet est prise, toutes les équations associées à l'imbrication considérée sont rejetées.

Une étape préliminaire à la mise en place de l'algorithme de la figure 8 va donc être de supprimer toutes les équations rejetées du système d'équations. Une fois ce travail fait, l'algorithme se déroule de manière identique en remplaçant l'équation (1) par (8) ce qui conduit à remplacer (2) par :

$$t_{ij} = \frac{T_j - \left(scrap_j.S_j\right).\overline{t_c}}{a_{ij}} \quad (10)$$

La détermination des temps moyens $\overline{t_i}$ se fait toujours par (3). Il est possible de déterminer également l'écart type du temps de la pièce *i* :

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{l}\left(t_{ij} - \overline{t_i}\right)^2}{l}} \quad (11)$$

De manière à éviter la prise en compte de valeurs aberrantes, il est possible de tester l'appartenance de toute donnée t_{ij} à l'intervalle $\left[-3\sigma_i ; +3\sigma_i\right]$ et d'écarter cette donnée pour le calcul de $\overline{t_i}$ si t_{ij} n'appartient pas à l'intervalle.

Pour l'étape suivante de l'algorithme, il est nécessaire de remplacer l'équation (4) par :

$$\sum_{i=1}^{n-k} a_{ij}.t_{ij} + \sum_{h=1}^{k} a_{hj}.\overline{t_i} = T_j - \left(scrap_j.S_j\right).\overline{t_c} \quad (12)$$

avec *j*=1, … *m'* et où *m'* est toujours le nombre d'équations non encore résolues, *k* est le nombre de temps moyens déjà trouvés. Tout le reste de l'algorithme reste identique.

Avec les améliorations apportées, le modèle, permet maintenant, de trouver 80% des temps de pièces. Il est capable de détecter un certain nombre de données aberrantes selon plusieurs critères qui peuvent être définis par l'utilisateur.

Les graphiques de la figure 10 montrent une distribution des temps avec une répartition très représentative. Cependant le critère (P <0,005) ne confirme toujours pas l'hypothèse nulle « les données observées suivent bien une loi normale », ce qui signifie qu'il existe encore des données aberrantes qui n'ont pas été détectées, et qui remettent en cause la validité de tous les résultats. Ce phénomène devrait s'estomper dans le temps avec le volume de données (voir paragraphe suivant), ce qui semble se dessiner dans la figure 11, puisque, comme attendu, plus nous disposons de données pour évaluer un temps de pièce particulier, plus ce dernier sera cohérent et les temps de pièce incohérents trouvés (temps de pièce négatifs par exemple) sont toujours associés à des populations très petites.

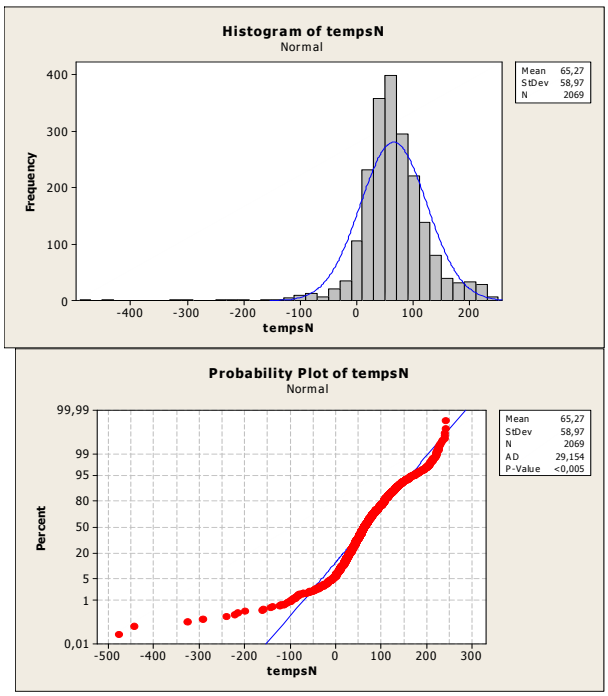


Figure 10 : Analyse des résultats du modèle amélioré de la machine Salva

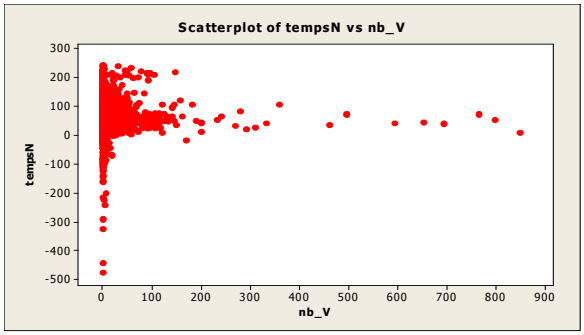


Figure 11 : Analyse des temps par pièce selon le nombre d'observations

6.3 Récursivité

La réalisation de la détermination des temps de pièce est une tâche lourde, et par nature, jamais finie. Pour ce faire, l'historique des données disponible permet de mettre en place une première estimation. Ces temps de pièce estimés sont utilisés pour produire de nouvelles pièces et donc permettent de récolter de nouvelles données. L'exploitation de ces nouvelles données assure d'affiner les résultats obtenus et même, d'adapter les temps de pièce estimés à une éventuelle dérive du processus pouvant être due à des évolutions, à l'usure…

De plus, un des éléments extrêmement important pour la qualité de l'estimation des temps de pièces est la bonne détermination du temps de chute $\overline{t_c}$. Ainsi, toute nouvelle imbrication peut donc servir non seulement à affiner les temps de pièces, mais aussi à affiner le temps de chute.

En présence des données relatives à une nouvelle imbrication, deux cas peuvent donc apparaitre :

- soit, une ou plusieurs des pièces de l'imbrication ne possèdent pas encore de temps de pièce moyen $\overline{t_i}$. Dans ce cas, il est nécessaire d'appliquer la procédure décrite au paragraphe 5.2,
- soit, toutes les pièces de l'imbrication possèdent déjà un temps moyen estimé $\overline{t_i}$. Dans ce cas, cette imbrication peut être utilisée pour raffiner le temps de chute $\overline{t_c}$ et les temps de pièce $\overline{t_i}$.

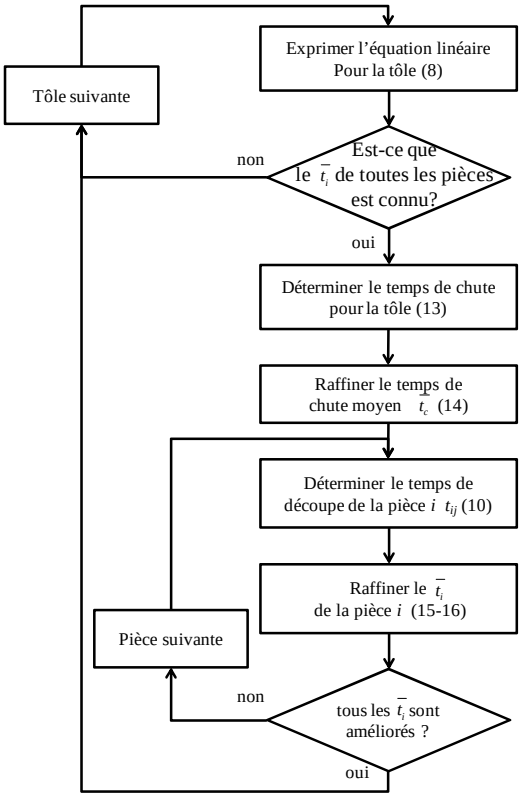


Figure 12 : Algorithme récursif de détermination des temps $\overline{t_c}$ et $\overline{t_i}$

La procédure pour améliorer de manière récursive les $\overline{t_c}$ et $\overline{t_i}$ est décrite par l'algorithme présenté figure 12.

La première étape consiste à déterminer l'équation linéaire de la tôle j par (8). Ensuite, si tous les $\overline{t_i}$ ne sont pas déjà estimés, il est nécessaire de le faire, et on applique l'algorithme de la figure 8 avant de passer à la tôle suivante. Si, par contre, tous les $\overline{t_i}$ sont déjà estimés, alors cette imbrication peut être employée pour améliorer l'estimation des temps connus. Il va alors être nécessaire de déterminer le temps de chute $\overline{t_c}$ de la tôle j :

$$t_{c_j} = \frac{T_j - \sum_{i=1}^{n} \overline{t_i}}{\left(scrap_j.S_j\right)} \qquad (13)$$

Ce temps de chute t_{c_j} peut alors être utilisé pour améliorer le temps de chute moyen $\overline{t_c}$ de manière récursive :

$$\overline{t_c}(k) = \overline{t_c}(k-1) + \frac{1}{k}\left(t_{c_j} - \overline{t_c}(k-1)\right) \qquad (14)$$

où k-1 est le nombre de tôles ayant permis de construire l'estimation $\overline{t_c}$ avant l'utilisation de l'imbrication considérée.

Une fois cette étape réalisée, nous pouvons, pour chaque pièce i présente dans l'imbrication, effectuer le même travail que pour le temps de chute. Il faut tout d'abord déterminer le temps t_{ij} par (10). On peut noter que dès cet instant, on peut utiliser le nouveau $\overline{t_c}$ trouvé à l'étape précédente. Une fois le temps t_{ij} déterminé, il ne nous reste plus qu'à améliorer le $\overline{t_i}$ moyen :

$$\overline{t_i}(k) = \overline{t_i}(k-1) + \frac{1}{k}\left(t_{ij} - \overline{t_i}(k-1)\right) \qquad (15)$$

où k-1 est le nombre de tôles ayant permis de construire l'estimation $\overline{t_i}$ de la pièce i considérée avant l'utilisation de l'imbrication considérée. Pour compléter la connaissance que l'on a du temps de la pièce i, nous pouvons également en déterminer le nouvel écart type $\sigma_i(k)$:

$$\sigma_i^2(k) = \sigma_i^2(k-1) + \frac{1}{k}\left(\varepsilon^2(k) - \sigma_i^2(k-1)\right) \qquad (16)$$

avec $\varepsilon(k) = t_{ij} - \overline{t_i}(k-1)$ qui est l'erreur de prédiction.

6.4 Adaptabilité

Les algorithmes présentés dans les paragraphes précédents nous permettent d'exploiter la connaissance présente dans les BdD et d'améliorer cette connaissance au fur et à mesure que de nouvelles données sont disponibles. Cependant, la machine SALVA, comme toutes les installations industrielles, est un système évolutif. Elle vieillit, elle s'use, elle est réparée, elle subit des modifications plus ou moins importantes… De même, les pièces que cette machine produit peuvent évoluer. Certaines, peuvent disparaître de la production, d'autres être nouvellement conçues, d'autres enfin présenter des plans légèrement modifiés.

Toutes ces évolutions peuvent être prises en compte lors de l'évaluation récursive des temps de pièce. Cependant, quand l'estimation d'un temps moyen $\overline{t_i}$ est déjà basée sur 999 imbrications passées par exemple, de (15), nous voyons tout de suite qu'une nouvelle imbrication ne va jouer que pour 1/1000ème dans l'évolution du $\overline{t_i}$ considéré ce qui rend toute adaptation à une évolution du processus illusoire.

Pour éviter ce phénomène, nous proposons d'introduire dans les équations (14), (15) et (16), un facteur d'oubli inspiré des stratégies d'apprentissage.

Ce facteur d'oubli doit avoir pour but de favoriser la partie évolution de l'équation au détriment de la partie mémorisation de l'historique. Dans les algorithmes d'apprentissage de type Recursive Prediction Error (RPE) (Chen *et al*., 1990), le facteur d'oubli est attaché à l'adaptation de la matrice de variance-covariance de telle sorte que cette matrice ne devienne jamais trop petite et qu'elle autorise toujours une évolution des paramètres. L'adjonction du facteur d'oubli dans l'équation (14) est obtenue en introduisant un terme multiplicatif $\lambda(k)$:

$$\overline{t_c}(k) = \overline{t_c}(k-1) + \frac{1}{\lambda(k).k}\left(t_{c_j} - \overline{t_c}(k-1)\right) \qquad (17)$$

Ce terme $\lambda(k)$ doit être choisi dans l'intervalle $]0;1]$ pour permettre un phénomène d'oubli. Quand $\lambda(k) = 1$, nous retrouvons un système sans oubli.

Afin de ne pas avoir d'impact quand peu de données sont utilisées il faut donc construire un facteur d'oubli proche de 1 quand k est petit et au contraire, pour faciliter l'adaptation quand beaucoup de données ont déjà été utilisées, il est nécessaire d'avoir un facteur d'oubli qui s'éloigne de 1. Pour les algorithmes RPE, certains auteurs préconisent l'utilisation d'une forme exponentielle en exploitant l'équation :

$$\alpha(k) = \alpha(0).\alpha(k-1) + (1 - \alpha(0)) \qquad (18)$$

Cette équation à une évolution exactement contraire à l'évolution recherchée ici. De ce fait, nous proposons d'utiliser comme facteur d'oubli la fonction :

$$\lambda(k) = 1 + \alpha(0) - \alpha(k) \qquad (19)$$

La figure 13 présente 3 exemples d'évolution de facteur d'oubli en fonction de 3 valeurs initiales de α.

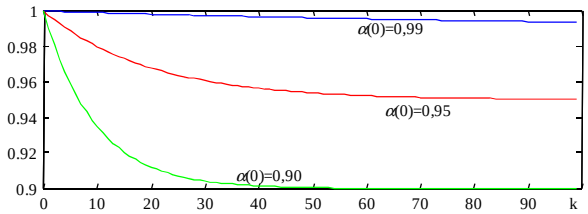


Figure 13 : Evolution du facteur d'oubli $\lambda(k)$ pour 3 valeurs de $\alpha(0)$: 0,99 ; 0,95 ; 0,90

Afin d'éviter que des imbrications ne prennent trop d'importance ponctuellement dans la détermination des temps moyen, l'ensemble de la procédure d'affinage récursif des temps devra être menée sur un ensemble d'imbrications collectées à la suite d'une journée ou d'une semaine de travail et non au fil de l'eau.

7 APPORTS POUR LES PERSPECTIVES DE L'ENTREPRISE ET CONCLUSION :

Après comparaison entre les temps calculés et les temps réels, l'entreprise a conclu à des résultats plutôt convaincants. Les deux modèles montrent, dans le temps, le même comportement à l'égard des différents facteurs. Connaissant désormais les temps opératoires par pièce tout au long du processus de production, un modèle de simulation va pouvoir être implémenté. Celui-ci permettra d'évaluer des solutions de dimensionnements des boucles kanban. En effet, dans la suite du projet, l'objectif est de mettre en place un système de kanban dynamique. A savoir, en fonction de la demande réelle, des capacités effectives de production, etc… le nombre de kanbans dans les boucles ainsi que le nombre de pièces par kanban seront calculés et évalués par simulation. Ce seront les kanbans qui, par l'usage de tags de type RFID, porteront eux même les informations conjoncturelles liées aux événements de production. Ils communiqueront au système central ces informations et si nécessaire, les paramètres des boucles kanban seront redéfinis. Deux expériences dans ce domaine ont déjà été menées dans notre équipe de recherche (El Haouzi. H 2008) (Klein. T, 2008).

D'une manière générale, l'objectif de l'entreprise est d'utiliser le plus possible ces technologies de l'information sans fil et de les associer soit aux produits, soit à des éléments du système physique associés aux produits afin de mettre en permanence à disposition des décideurs (localement ou dans le système central) des données nécessaires au réajustement temps réel des paramètres de production.

Cet article a donc pu montrer une étape du projet consistant à déterminer de manière automatique les temps par pièce sur les différentes machines d'une cellule de production.

REMERCIEMENT :

Nous remercions M. François Bertrand pour son travail sur la partie calcul des temps pour les machines traditionnelles dans le cadre de son contrat d'apprentissage à l'entreprise Trane.

BIBLIOGRAPHIE

Chen, S., S.A. Billings and P.M. Grant, 1990. Non linear system identification using neural networks. *International Journal of Control*, 51(6), p. 1191-1214.

Cintra, G.F., F.K. Miyazawa , Y. Wakabayashi and E.C. Xavier, 2008. Algorithms for two dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1), p. 61-85.

Bayard, D., 1996. Savoir organisationnel, savoir théorique et situation : le contrôle statistique sur échantillons. *Entreprises et Histoire*, 13, p. 67-81.

EL Haouzi, H., 2008. *Approche méthodologique pour l'intégration des systèmes contrôlés par le produit dans un environnement de JàT*. Thèse de Doctorat, Université Henri Poincaré Nancy I, France.

Grappin, R., 1976. *Guide pour l'évaluation des méthodes d'analyses de routine I.N.R.A.*, Station Expérimentale Laitière - 39800 Poligny.

Klein, T., 2008. *Le Kanban Actif pour Assurer l'Interopérabilité Décisionnelle Centralisée / Distribuée*. Thèse de Doctorat, Université Henri Poincaré Nancy I, France.

Nations, U, 1994. *Industrial Commodity Statistics Yearbook* (ISBN: 9210612094).

Tardif, V. and L. Maaseidvaag, 2001. An Adaptive Approach to Controlling Kanban Systems. *European Journal of Operation Research*, 132, p. 411-424.