



HAL
open science

Réduction et génération automatique de modèle Arena© en exploitant des arbres de régression : cas d'une scierie

Philippe Thomas, Marie-Christine Suhner, André Thomas

► To cite this version:

Philippe Thomas, Marie-Christine Suhner, André Thomas. Réduction et génération automatique de modèle Arena© en exploitant des arbres de régression : cas d'une scierie. 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14, Nov 2014, Nancy, France. hal-01086859

HAL Id: hal-01086859

<https://hal.science/hal-01086859>

Submitted on 25 Nov 2014

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.

REDUCTION ET GENERATION AUTOMATIQUE DE MODELE ARENA© EN EXPLOITANT DES ARBRES DE REGRESSION : CAS D'UNE SCIERIE

^{1,2}Philippe THOMAS, ^{1,2}Marie-Christine SUHNER, ^{1,2}André THOMAS

¹Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy cedex, France
²CNRS, CRAN, UMR7039, France

philippe.thomas@univ-lorraine.fr, marie-christine.suhner@univ-lorraine.fr, andre.thomas@univ-lorraine.fr

RESUME : *La simulation est souvent utilisée pour évaluer la pertinence d'un Plan Directeur de Production (PDP) ou pour en évaluer l'impact sur des scénarii d'ordonnancement détaillé de la fabrication. Cependant, construire de tels modèles de simulation reste une tâche difficile. L'objectif de ce papier est de réduire la complexité de cette tâche et de la rendre partiellement automatisée en combinant des approches discrètes et continues conduisant à des modèles réduits plus simples et rapides. Le principe est de focaliser l'étape de modélisation selon une approche discrète sur les goulots d'étranglement en accord avec la théorie des contraintes, les autres éléments de l'atelier étant modélisés de manière plus grossière à l'aide d'un modèle continu dont l'unique but est de déterminer comment les goulots sont alimentés. Ce modèle continu est construit en exploitant un arbre de régression qui combine une séquence de tests logiques. Un tel arbre de régression peut alors très simplement, voir automatiquement, être transformé en modèle Arena©. Cette approche est appliquée au cas d'une scierie et les résultats sont comparés avec ceux obtenus dans un travail précédent exploitant des réseaux de neurones.*

MOTS-CLES : *Arbres de décisions; CART; modèle réduit; simulation; chaîne logistique.*

1 INTRODUCTION

L'évaluation des scénarios de planification ou d'ordonnancement est un point crucial pour le contrôle de chaînes logistiques internes ou externes. Dans ce cadre, la simulation est un outil important qui permet, entre autre, de mettre en évidence l'évolution de l'état des ressources, des en-cours ou des files d'attente. Ces informations sont particulièrement utiles pour l'établissement d'un ordonnancement prévisionnel (Lopez and Roubellat, 2001).

Par ailleurs, des événements, aléas, peuvent remettre en cause le Plan Directeur de Production (PDP) rendant nécessaire d'effectuer un ré-ordonnancement des tâches. Les informations relatives à ces aléas sont collectées très rapidement par les systèmes temps réel de suivi de fabrication (Khouja 1998), la difficulté consiste alors dans le traitement de cette masse d'informations afin de prendre des décisions rapidement (Pritsker et Snyder 1994, Roder 1994). A ce niveau de planification, l'équilibrage de charge ou de capacité est obtenu par la fonction de management de la capacité des ressources critiques ou « Rough-Cut Capacity Planning » qui concerne principalement les goulots d'étranglement (Vollmann *et al.* 1992). Goldratt et Cox, dans "Le But" (1992) proposent de piloter l'ensemble de la chaîne logistique en contrôlant les goulots d'étranglement ce qui a conduit à la théorie des contraintes (ToC).

Dans ce cadre, l'utilisation de la simulation par événements discrets utilisées pour évaluer la manière dont

évoluent les flux matières est particulièrement utile (Thomas and Charpentier, 2005). Cependant, construire des modèles de simulation de cas industriels réels est une tâche très complexe, souvent longue, et le modèle résultant peut s'avérer trop lourd ou trop complexe pour être réellement utile (Page *et al.* 1999).

De nombreux travaux ont montré l'intérêt d'exploiter des modèles de simulation plus simples (réduits/agrégés) (Ward 1989; Brooks and Tobias 2000; Chwif *et al.* 2006). Différentes approches de simplification de modèles ont été proposées, comme par exemple, l'utilisation de modèles continus exploitant une estimation du gradient pour les systèmes stochastiques afin d'approcher le comportement discret d'un système manufacturier (Ho, 1987; Suri and Fu, 1994). D'autres auteurs exploitent des métas modèles (régression linéaire, splines, Kriging, ...) pour construire des modèles de simulation (Kleijnen et Sargent, 2000). Les réseaux de neurones ont également été exploités dans ce cadre (Thomas and Thomas 2011, Thomas *et al.* 2011).

Un autre outil, les arbres de régression, est également capable de trouver un modèle performant pour réduire des modèles de simulation. Les arbres de régression et de classification sont des machines d'apprentissage capables d'extraire un modèle à partir de données. Ce modèle est obtenu en partitionnant l'espace des données et en associant un modèle de prédiction très simple à chacun de ces sous-espaces (Breiman *et al.* 1984). Cet outil est donc capable d'extraire un modèle directement à partir d'un jeu de données à l'instar des réseaux de neurones, mais contrairement à ces derniers, le modèle obtenu

nu n'est pas une boîte noire mais une succession de tests logiques dont il est possible d'extraire de la connaissance.

L'objectif principal de ce papier est d'évaluer la capacité des arbres de régression à être utilisés pour réduire la taille, et donc, la complexité, des modèles de simulation. Ces modèles étant constitués d'une succession de tests logiques, ils peuvent de surcroît être très simplement transformés en modèle Arena©.

La section suivante présentera l'approche de réduction de modèles proposée ainsi que les arbres de régression. La troisième partie concernera la présentation de l'application industrielle considérée, qui sera ici le cas d'une scierie. Le modèle de simulation réduit de la scierie sera finalement présenté dans la partie 4 et les résultats obtenus seront comparés avec ceux obtenus dans des travaux précédents. La transformation de l'arbre de régression en modèle Arena© sera également présentée avant de conclure.

2 LE MODELE REDUIT

2.1 L'algorithme

La complexité d'un modèle dépend du nombre d'éléments, de connexions et de calculs dans le modèle. Zeigler (1976) note différentes approches pour simplifier un modèle de simulation discret :

- En remplaçant une partie du modèle par une variable aléatoire,
- En dégradant le nombre de valeurs que peuvent prendre une variable,
- En regroupant différentes parties du modèle ensemble.

Innis et al. (1983) ont listé 15 techniques de simplification relatives à quatre étapes principales :

- Hypothèse : identifier les parties importantes du système,
- Formulation : définir le modèle
- Codage : construction du modèle
- Expérimentation.

Différentes approches de réduction de modèle ont été proposés, une courte revue bibliographique de ces approches a été proposée dans (Thomas et al. 2011).

L'algorithme de réduction de modèle proposé ici est une modification de celui proposé par Thomas et Thomas (2011) qui est basé sur l'exploitation de modèles neuronaux. Nous allons ici associer un modèle à événements discrets et un modèle continu (les arbres de décisions) dans le but de construire un modèle de simulation (figure 1). L'objectif est de maximiser le taux d'utilisation des goulots d'étranglement et, dans le même temps, de simplifier la phase de modélisation. Les principales étapes sont les suivantes :

1. Identifier le goulot structurel (Poste de Charge (PdC) qui, depuis plusieurs années, est majoritairement contraint en capacité).
2. Pour la liasse d'Ordres de Fabrication (OF) du PDP considéré, identifier le goulot conjoncturel.

3. Parmi les autres PdC, identifier ceux (postes de synchronisation) satisfaisant aux deux conditions :
 - utilisé conjointement à un poste goulot par au moins un OF,
 - largement utilisé lorsque l'ensemble des OF est considéré.
4. Si tous les OF ont été considérés, passer à 5. sinon retourner en 3.
5. Modéliser à l'aide d'un arbre de décision les intervalles entre les différents PdC déterminés aux étapes précédentes.

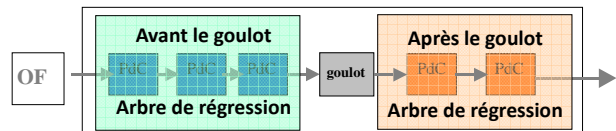


Figure 1 : Algorithme de réduction de modèle

Les postes de charge (PdC) représentés dans le modèle sont donc, soit le goulot conjoncturel, soit le goulot structurel, soit les PdC indispensables pour la synchronisation des OF. Tous les autres PdC sont inclus dans les blocs agrégés précédant ou suivant les goulots. Le goulot conjoncturel est le PdC saturé pour le PDP et l'ordonnancement prédictif considéré, ce qui signifie que pour le PDP considéré, il utilise l'ensemble de sa capacité de charge. Le goulot structurel est le PdC qui, dans le passé, s'est le plus souvent trouvé dans cette condition. Pour le PDP considéré, les goulots structurel et conjoncturel peuvent être confondus. En effet, pour un PDP spécifique, il n'existe qu'un et un seul goulot d'étranglement, le poste le plus chargé, mais ce poste peut être différent du goulot traditionnel. Les PdC de synchronisation sont les ressources utilisées conjointement avec le goulot pour au moins un OF. Afin de minimiser le nombre de postes de synchronisation, uniquement le PdC le plus commun au goulot pour l'ensemble des OF est sélectionné.

2.2 Les arbres de régression

Les arbres de décision sont des modèles séquentiels combinant une séquence de tests simples. Un arbre de régression est similaire à un arbre de classification, à la différence près que la variable de sortie est une variable continue et un modèle de régression très simple est associé à chaque nœud afin de fournir la valeur prédite de la sortie (Loh 2011).

Le premier arbre de régression (AID) a été proposé par Morgan et Sonquist (1963). Le principe de cet arbre de régression est de partir d'un nœud racine, et pour chaque nœud, de trouver le test sur les données d'entrée qui réduit la somme des impuretés dans ses deux nœuds enfants. Breiman et al. (1984) ont proposé l'algorithme Classification and Regression Tree (CART) qui fonctionne sur le même principe. La principale différence entre AID et CART est le choix des processus d'estimation et de « pruning ». CART est basé sur une généralisation de la variance binomiale appelée indice

Gini. D'autres algorithmes ont été proposés dans le passé comme C4.5 (Quinlan 1996), CART (Breiman et al. 1984)... L'algorithme utilisé ici est le CART.

Le processus de croissance de l'arbre est récursif et la croissance des branches s'arrête lorsque (Lewis 2000) :

- Il n'y a qu'une seule observation dans le nœud terminal,
- Toutes les observations dans le nœud terminal possèdent une distribution identique rendant toute division impossible,
- Une limite externe sur le nombre de niveau de l'arbre ou sur un seuil de décroissance de l'impureté est atteinte.

L'arbre ainsi obtenu présente généralement un problème de surapprentissage qui nécessite une phase de pruning. Cette phase de pruning est basée sur l'indice d'impureté exploité durant la phase de croissance (Gini) associé à un terme de pénalité correspondant au nombre de nœuds terminaux. La sélection de l'arbre de décision final est obtenue à l'aide d'un processus de validation croisée (Breiman et al. 1984).

3 LE CAS D'APPLICATION

Le cas d'application étudié est une scierie, appartenant au premier scieur français, située dans le massif des Vosges en France et spécialisée dans la transformation de conifères en planches.

La ligne de sciage peut être décomposée en trois parties principales. Afin d'en comprendre le fonctionnement,

nous allons suivre le cheminement d'une grume de son entrée dans la chaîne, à sa sortie sous forme de diverses planches.

La première partie de la ligne de sciage est constituée de la ligne Canter présentée à la figure 2. La grume que l'on doit traiter entre dans la chaîne par la droite par l'intermédiaire des convoyeurs RQM1 RQM2 et RQM3. En fonction de ses caractéristiques (scanner MS), notre grume va être dirigée soit vers RQM4 soit vers RQM5 qui font office de stocks. Elle va ensuite être amenée sur la première machine Canter puis sur la déligneuse CSMK qui a pour but de transformer notre grume en un parallélepède rectangle, l'équarri. Cette première étape, qui forme les deux premiers côtés de l'équarri, produit 2 planches (appelées produits secondaires) qui seront expulsées par l'intermédiaire des convoyeurs BT4 et BT5. Le reste de la grume va être éjecté sur RQM6, va subir une rotation de 90° pour être enfin stocké sur RQM7 en attendant de repasser sur la même machine pour faire les deux autres côtés du parallélepède. A l'issue du deuxième passage, outre l'équarri, on obtient deux autres produits secondaires qui sont également évacués par les convoyeurs BT4 et BT5 en direction de la Tronçonneuse Déligneuse dont l'élément principal est la scie Kockum. L'équarri est ensuite déligné sur la déligneuse MKV en trois planches supplémentaires que nous appellerons produits principaux. Ces produits principaux sont directement dirigés vers l'éboueur.

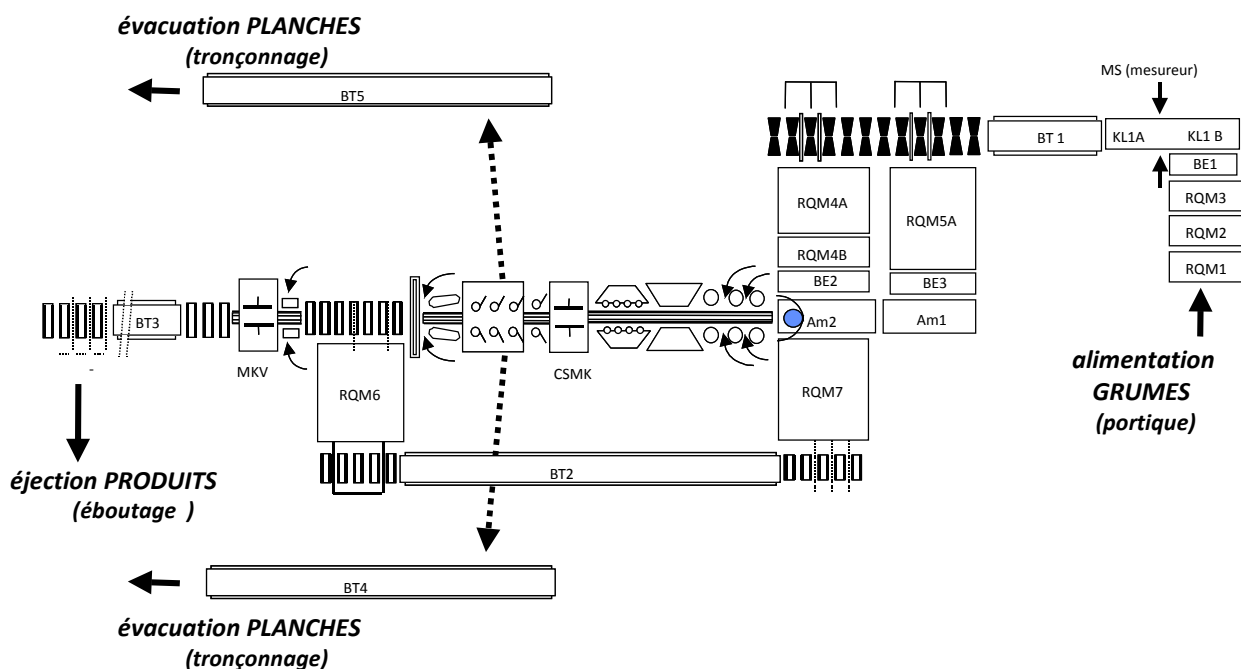


Figure 2. La ligne Canter

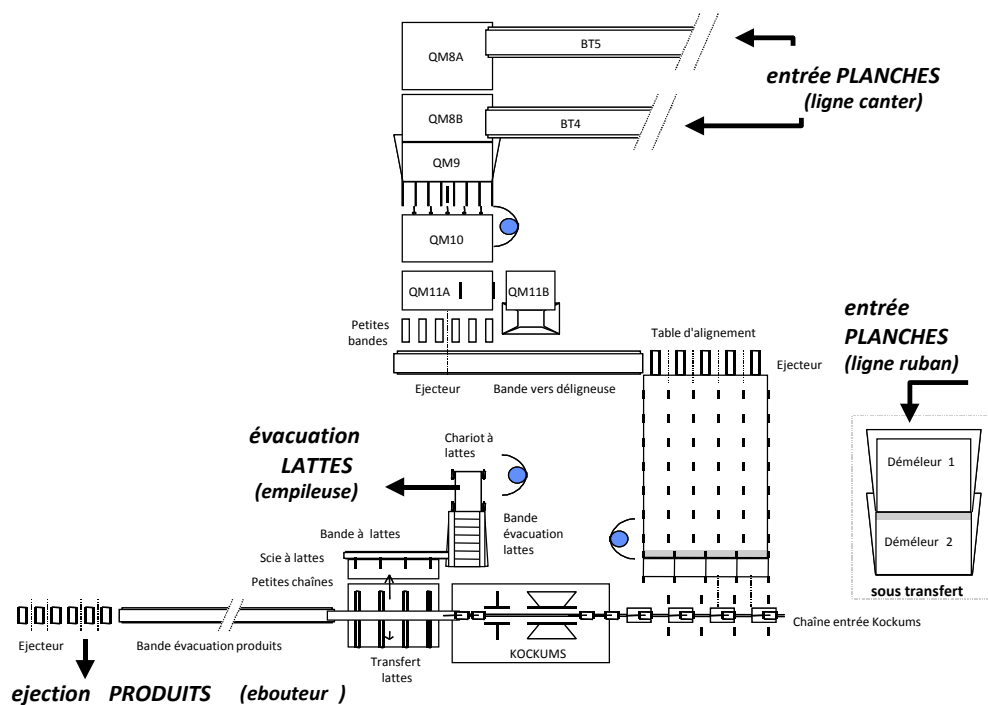


Figure 3. La tronçonneuse-déligneuse (Kockums)

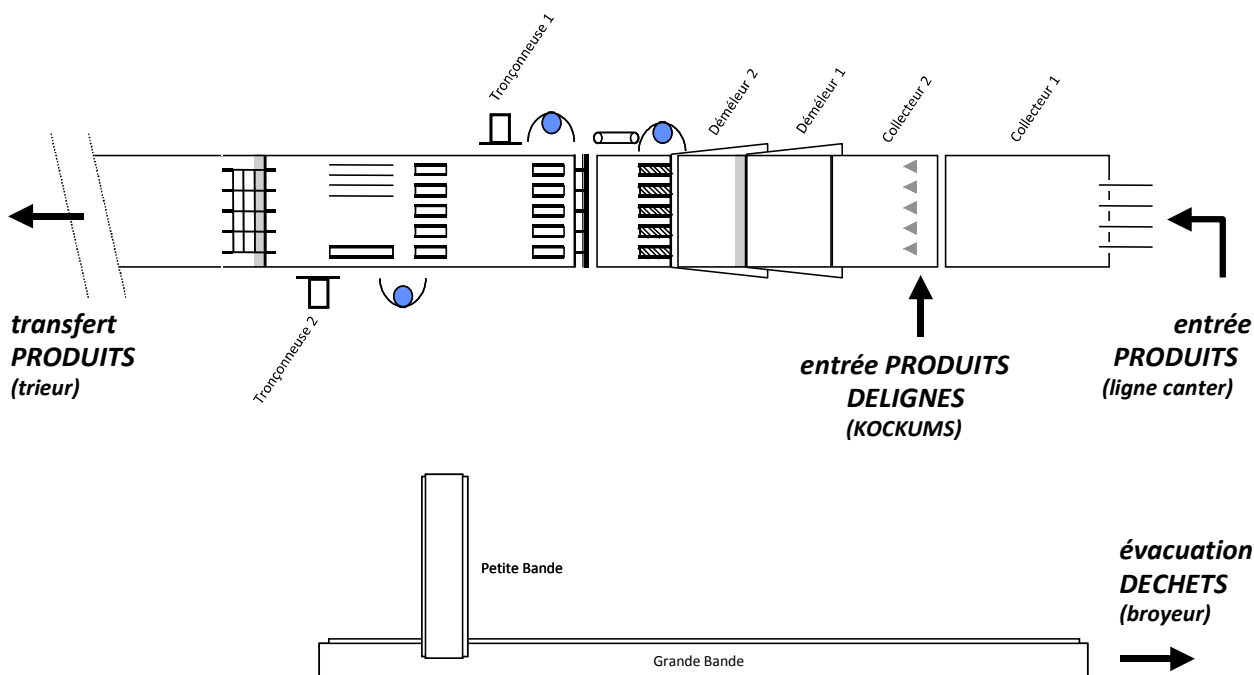


Figure 4. L'ébouteur

La figure 3 présente la deuxième partie de la ligne de sciage qui correspond au tronçonnage déligneuse et dont la machine principale est la scie Kockums. Seuls les produits secondaires passent sur cette partie de ligne. L'entrée des produits secondaires se fait par l'intermédiaire des convoyeurs BT4 et BT5. Ils sont tronçonnés en QM11 puis arrivent par le tapis sur la déligneuse (Kockums) qui optimise la planche en fonction des produits demandés. La table d'alignement permet de servir de stock d'entrée de la scie Kockums.

Les produits secondaires sont finalement expédiés par l'intermédiaire de la bande d'évacuation en direction de la troisième partie de la ligne, l'ébouteur qui est présenté figure 3.

En entrée de l'ébouteur se trouvent deux collecteurs (collecteurs 1 et 2) qui permettent l'admission sur la ligne respectivement des produits secondaires en provenance de la scie Kockums, et des produits principaux en provenance directe de la ligne Canter. Les pièces sont démantelées et mises sur un tapis à taquets.

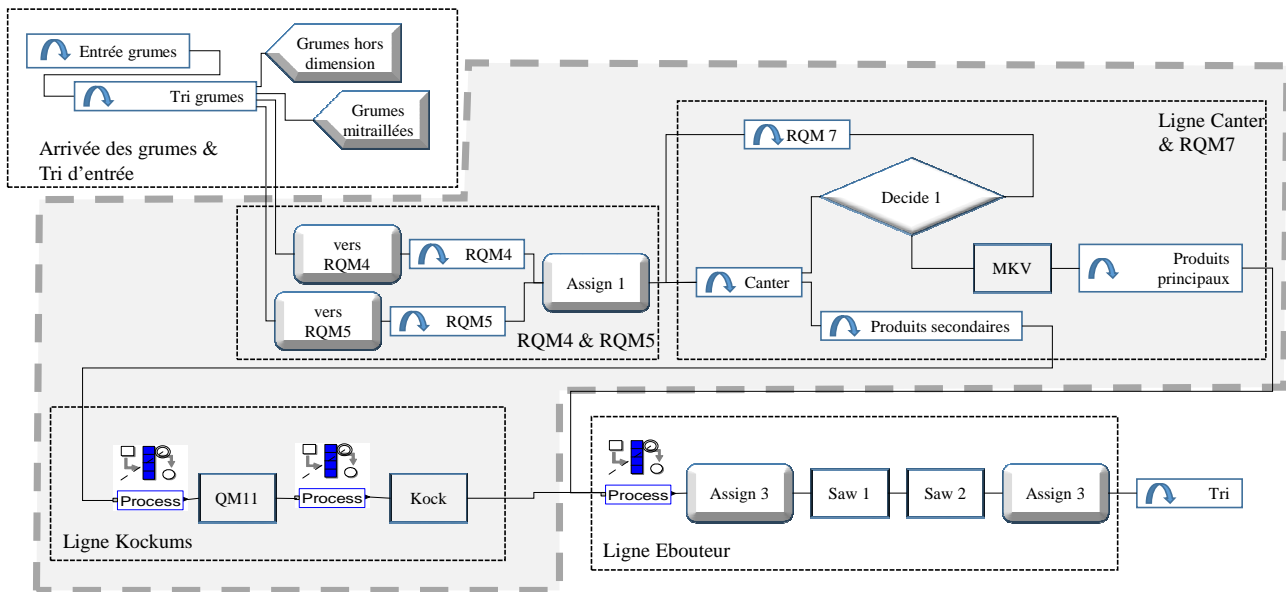


Figure 5 : Le modèle complet

La tronçonneuse 1 va permettre d'effectuer une coupe de purge des défauts tandis que la tronçonneuse 2 permet de donner sa longueur finale au produit. Des travaux précédents (Thomas et Charpentier 2005) ont montré que c'est cette machine, l'ébouteur, qui est le poste goulot de la chaîne de sciage, et comme expliqué précédemment, de la productivité de l'ébouteur dépend la productivité de toute la ligne. Or cette productivité de l'ébouteur dépend de décisions prises en entrées de la ligne Canter. Aussi, l'impact d'une décision est observé bien trop tardivement pour pouvoir être corrigé. D'où, pour en évaluer l'impact, un outil de simulation est nécessaire.

LES MODELES DE SIMULATION

Comme expliqué précédemment, le goulot d'étranglement de cette ligne de production est l'ébouteur. De ce fait, la modélisation du fonctionnement des stocks d'entrées RQM4, RQM5, et RQM7, ainsi que celui des lignes Canter et Kockums n'est pas nécessaire et ne fournit aucune information « utile » (critique) pour l'évaluation du PDP considéré. Aussi, dans le modèle complet présenté figure 5, tous les éléments entourés en pointillés ne sont pas indispensables au modèle et peuvent être remplacés par un modèle continu, ici un arbre de régression.

Suivant l'algorithme présenté partie 2, un modèle réduit de la scierie peut être construit. Afin de le construire la partie continue du modèle qui va remplacer tous les éléments non nécessaires précédemment décrits, il est nécessaire d'identifier les variables explicatives. Thomas et al. (2011) ont collecté les variables d'entrées qui peuvent être classées en trois catégories :

- Données relatives au produit (ici la grume),
- Données relatives au process,
- Données relatives à la nomenclature (ici le plan de coupe).

Les données relatives au produit sont principalement des données dimensionnelles, longueur (lg) et trois valeurs de diamètres (diaPB, diaGB, et diaMOY).

Les données relatives au process sont les données du process collectées à l'arrivée de la grume en entrée de ligne et concernent la longueur de la file d'attente en entrée du goulot et le taux d'utilisation de ce même goulot, ici l'ébouteur (Q_{trim} , et U_{trim} , respectivement). Le nombre de grumes présentes dans le process entre l'entrée et la sortie de la ligne Canter (Q_{RQM}) et le niveau des stocks d'en cours (Q_{RQM4} , Q_{RQM5} et Q_{RQM7}) sont également nécessaires.

Les données relatives à la nomenclature, ici le plan de coupe, indique que les grumes doivent être tronçonnées en produits principaux et secondaires. Cette information est donnée par le type de pièce (T_{piece}).

En conséquence, les variables d'entrée du modèle sont : Lg , $diaGB$, $diaMOY$, $diaPB$, T_{piece} , Q_{trim} , U_{trim} , Q_{RQM} , Q_{RQM4} , Q_{RQM5} et Q_{RQM7} . Dans notre application, 12775 produits sont simulés à l'aide du modèle complet de la figure 5 qui nous servira de référence, l'objectif étant d'obtenir un comportement du modèle réduit équivalent à celui du modèle complet de référence. Ce jeu de données est divisé en deux, une partie pour l'identification et une autre pour la validation du modèle. L'étape suivante est de déterminer la variable de sortie. L'objectif de ce modèle continu est de déterminer comment le goulot d'étranglement sera approvisionné. De ce fait, notre objectif est de modéliser le temps d'écoulement des produits entre l'entrée de la grume dans le process et son arrivée sous forme de planche dans la file d'attente du goulot (ΔT). Pour les problèmes de régression, l'objectif est de réduire la distance entre le modèle et les données. Le critère RMSE (Recursive Mean Square Error) peut être utilisé pour évaluer cette distance :

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2} . \quad (1)$$

où N représente le nombre de données, y_n est la $n^{\text{ème}}$ donnée relevée, et \hat{y}_n sa valeur prédite.

La figure 6 présente l'évolution du RMSE en fonction du nombre de nœuds inclus dans l'arbre. Cette figure montre classiquement que même si le RMSE décroît continuellement en fonction du nombre de nœuds sur le jeu de données d'identification, il n'en est pas de même

sur le jeu de validation où cette valeur commence par décroître avant d'augmenter à nouveau. Cette évolution illustre le problème de surapprentissage et la sélection du meilleur modèle doit être effectuée en utilisant une stratégie de validation croisée. Dans notre cas, le meilleur modèle comprend 296 nœuds.

Ce meilleur modèle doit être comparé avec celui obtenu dans des travaux précédents qui exploitaient un modèle neuronal (Thomas et al. 2011).

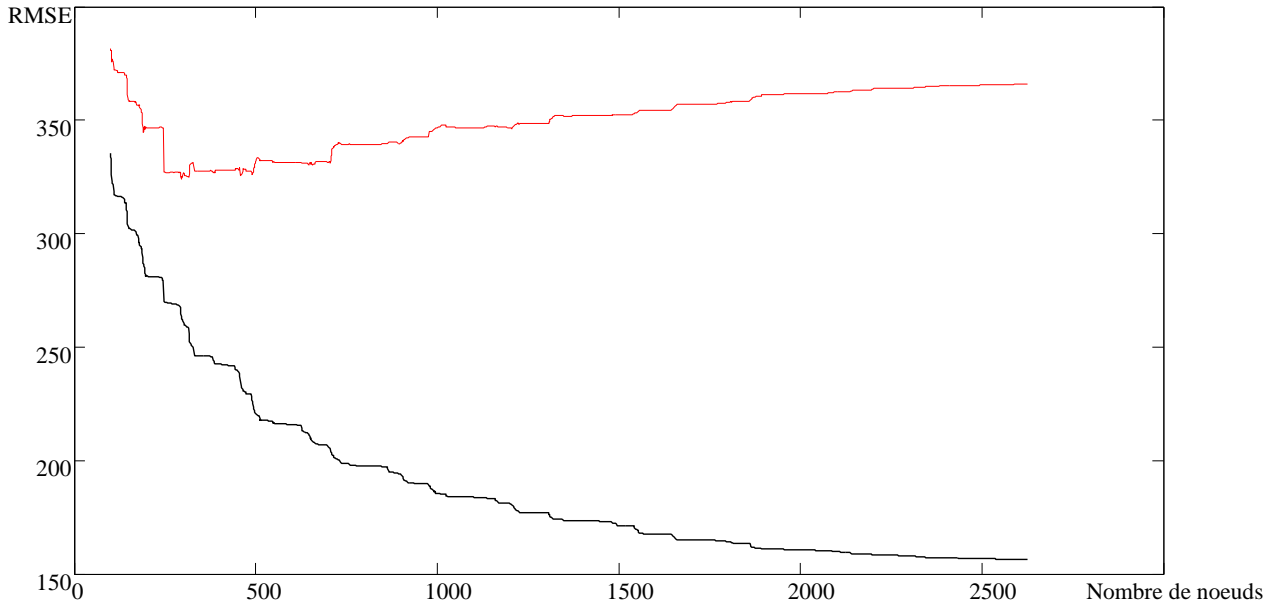


Figure 6 : Evolution du RMSE en fonction du nombre de nœuds – jeu de donnée d'identification en gras noir – jeu de données de validation en rouge

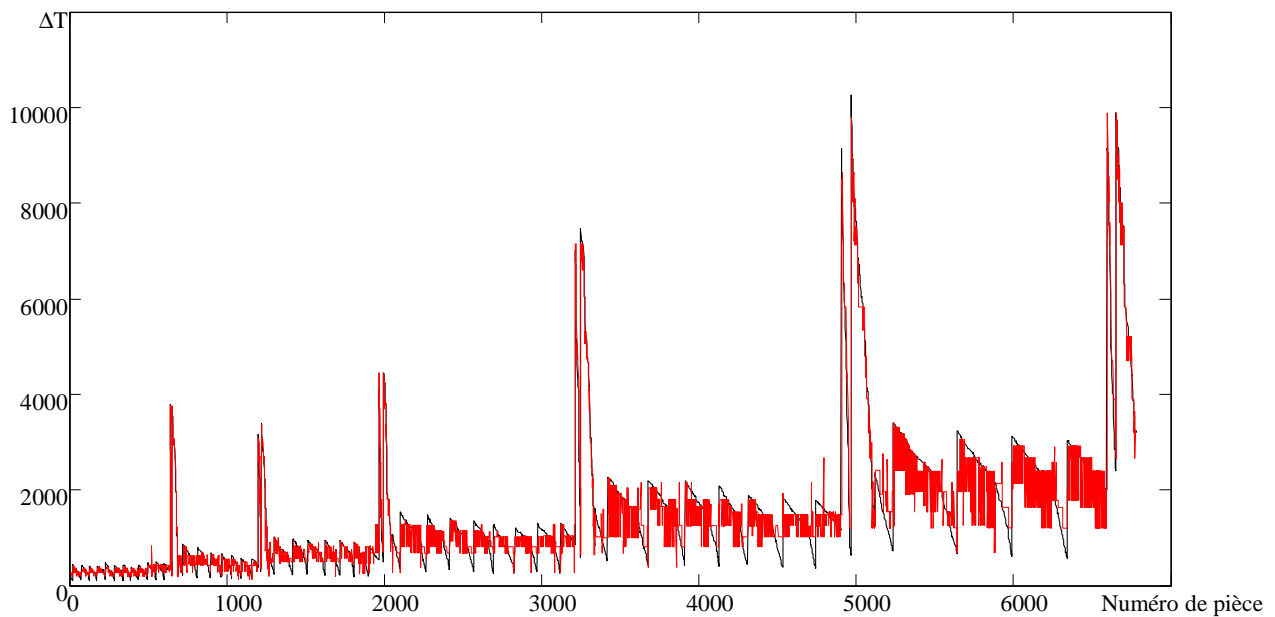


Figure 7 : Temps d'écoulement sur le jeu de validation – modèle complet en noir épais – modèle réduit en rouge

Ces résultats sont présentés table 1 et montrent que l'utilisation d'un arbre de régression permet d'améliorer de plus de 20% les résultats comparativement à ceux obtenus avec un modèle neuronal.

	RMSE	
	Identification	Validation
Modèle neuronal	408,45	413,93
Modèle arbre de régression	263,47	324,10

Table 1 : Comparaison du RMSE pour les modèles neuronaux et arbres de régression

La figure 7 présente les temps (en ordonnée) qu'ont mis les 6781 pièces (en abscisse) constituant le jeu de validation entre le moment où la grume correspondante est entrée dans la chaîne et le moment où la pièce est entrée

dans le stock de l'ebouteur. Nous avons en bleu les temps fournis par le modèle complet qui nous sert de référence, et en vert les temps fournis par l'arbre de régression. Les temps sont fournis en secondes.

Nous pouvons constater que les différentes pièces ont des comportements assez différents car si la majorité des pièces mettent moins de 1000 secondes pour traverser la ligne Canter et passer sur la scie Kockums (pour les pièces secondaires), un certain nombre mettent cependant plus de 3000 secondes pour faire le même trajet. Nous voyons d'autre part que les temps fournis par l'arbre de régression sont tout à fait similaires à ceux donnés par le modèle complet. La figure 8 compare l'erreur obtenue sur le jeu de validation par le meilleur modèle neuronal en bleu et le meilleur arbre de régression en vert.

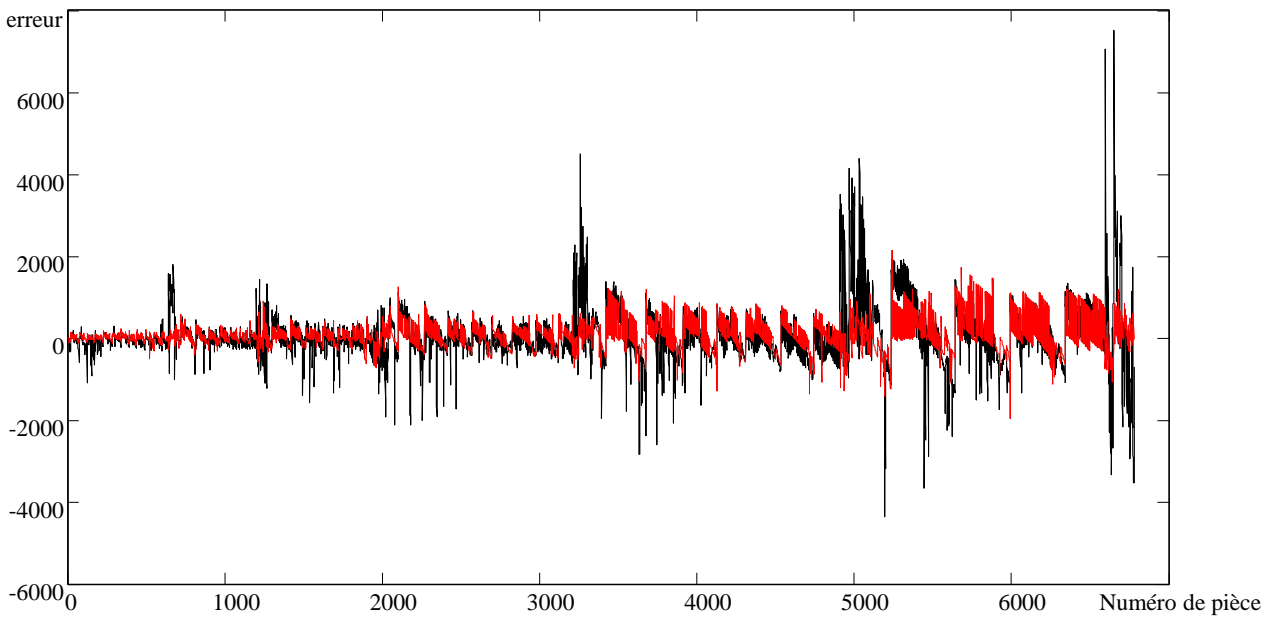


Figure 8 : Erreur effectuée par le modèle neuronal en noir épais et l'arbre de régression en rouge

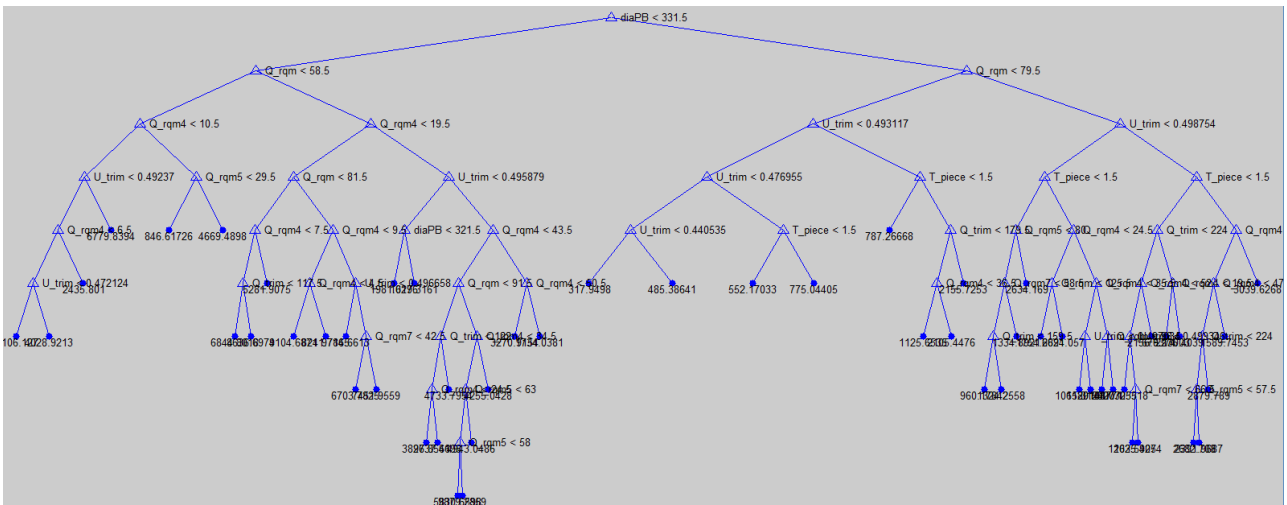


Figure 9 : Le modèle arbre de régression (extrait)

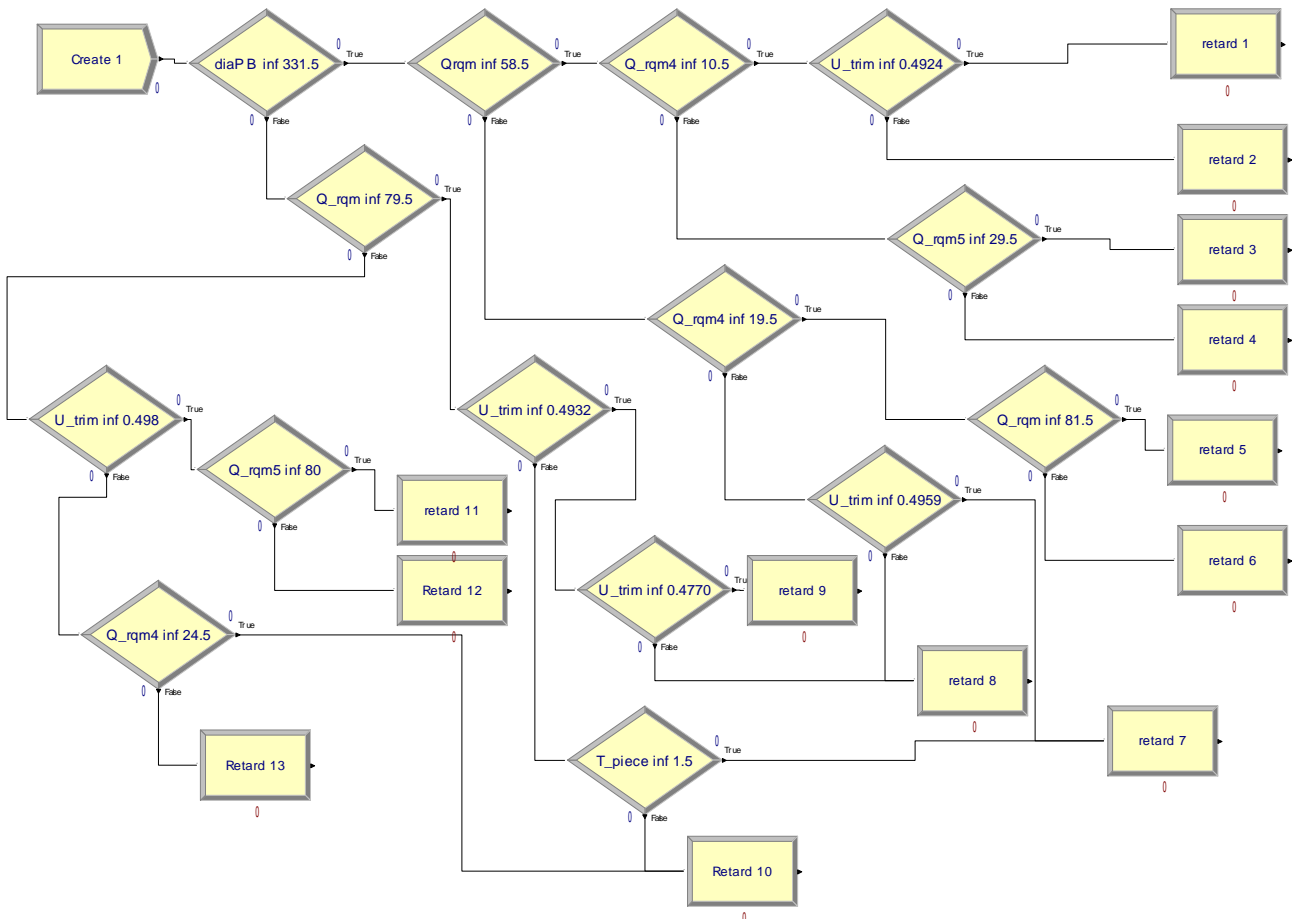


Figure 10 : Traduction de l'arbre de régression en modèle Arena©

Cette figure montre que l'arbre de décision, sur les mêmes données, a été capable de bien mieux prendre en compte les grandes variations de temps d'écoulement et donne une erreur beaucoup plus faible que le modèle neuronal.

La figure 9 présente le modèle arbre de régression ainsi construit. Cette figure présente un autre avantage des arbres de régression par rapport aux réseaux de neurones. En effet, si ces derniers ont des pures boîtes noires desquelles aucune connaissance ne peut être extraite, les arbres de régression sont eux une succession de tests logiques partant des plus informatifs (ici $\text{diaPB} < 331,5$) aux moins informatifs jusqu'aux feuilles du modèle qui fournissent des modèles de régression très simples, ici des valeurs moyennes de temps d'écoulement. Un tel modèle peut aisément, et de manière automatique, être transformé en un modèle à événements discrets incorporable dans des outils du type Arena©, un tel modèle pouvant être représenté par une succession de blocs « décide ».

A titre d'illustration, la figure 10 présente la traduction des quatre premiers niveaux de l'arbre de régression en modèle Arena©. On voit ainsi que chaque nœud de l'arbre peut être traduit par une fonction « décide » quand les feuilles de l'arbre sont, elles, traduites par une fonction « process » permettant de simuler un temps d'écoulement. Ce modèle est bien évidemment beaucoup plus simple que la traduction de l'arbre complet qui comprend 296 nœuds, mais il permet de montrer que

l'exploitation d'un arbre de régression permet de construire automatiquement un modèle à événement discret directement à partir d'un ensemble de données, ce qui n'est pas possible avec un modèle neuronal. Le principal désavantage des arbres de régression pour ce type d'application est leur faible adaptabilité. En effet, les processus industriels sont des systèmes « vivants » qui changent avec le temps suite à divers types d'événements (modification de machines, de gammes...). Une telle évolution va faire apparaître une dérive entre le comportement du processus et celui de son modèle. Si dans le cas d'un modèle neuronal, une telle dérive peut être compensée par des phases de réapprentissage périodique qui ne remettront pas en cause la structure même du modèle, il n'en est pas de même des arbres de décisions qui nécessiteront un réapprentissage aussi bien de la structure du modèle que des valeurs des différents tests.

4 CONCLUSION

L'utilisation d'arbres de régression pour construire des modèles réduits de simulation est étudiée ici. Les arbres de régression sont utilisés pour modéliser de manière continue les éléments d'un processus de fabrication qui ne sont pas contraints en capacité. Cette approche a été appliquée à la modélisation d'une scierie. Les résultats ont montré qu'un modèle réduit construit sur ce principe

donne des résultats équivalents à ceux fournis par un modèle complet. De surcroît, les résultats obtenus ont été comparés à ceux obtenus dans une précédente étude et exploitant des réseaux de neurones. Cette comparaison montre que les arbres de régression donnent de meilleurs résultats sur cette application. Il semble donc pertinent d'exploiter des arbres de régression pour modéliser certaines parties du processus plutôt que de construire un modèle complet. Par ailleurs, les arbres de régression étant constitués d'une succession de tests logiques simples, il est possible d'en extraire de la connaissance et même de les traduire automatiquement en modèle à événements discrets.

Sachant que la construction d'un arbre de régression est une tâche quasi automatique qui nécessite de la part du modélisateur uniquement de collecter et sélectionner les données, une telle approche de modélisation est plus simple et plus rapide comparativement à la construction d'un modèle complet. Une telle approche permet au modélisateur de se focaliser sur la modélisation des goulots d'étranglement qui sont, d'un point de vue gestion des flux, les postes de charges principaux.

Le principal défaut de cette approche est le manque d'adaptabilité des arbres de régression qui en présence d'une évolution du processus nécessiteront un réapprentissage complet du modèle contrairement aux réseaux de neurones. Mais au moins dans un premier temps, ils sont envisageables pour des infrastructures industrielles stables (industries de process), mais dans nos futurs travaux, nous nous intéresserons à l'adaptabilité des arbres de régression afin de faire face à l'évolutivité des processus. Nous chercherons également à valider cette approche sur d'autres types d'applications et en particulier sur des cas de modélisation de chaînes logistiques externes dont une au moins des entreprises appartient à différentes chaînes logistiques.

REFERENCES

- Breiman L., Friedman J.H., Olshen R.A., Stone C.J., 1984. Classification and regression trees, Chapman & Hall, Boca Raton, USA.
- Brooks, R.J., Tobias, A.M., 2000. Simplification in the simulation of manufacturing systems. *International Journal of Production Research*, 38(5): 1009-1027.
- Chwif, L., Paul, R.J., Pereira Barretto, M.R., 2006. Discrete event simulation model reduction: A causal approach. *Simulation Modelling Practice and Theory*, 14: 930-944.
- Goldratt., E., Cox, J., 1992. *The Goal: A process of ongoing improvement*, North River Press; 2nd Revised edition, Great Barrington, USA.
- Ho, Y.C., 1987. Performance evaluation and perturbation analysis of discrete event dynamics systems. *IEEE Transaction on Automatic Control*, 32(7), 563-572.
- Innis, G.S., Rexstad, E., 1983. Simulation model simplification techniques. *Simulation*, 41: 7-15.
- Khouja, M., 1998. An aggregate production planning framework for the evaluation of volume flexibility. *Production Planning and Control*, 9(2), 127-137.
- Kleijnen, J.P.C., Sargent, R.G., 2000. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120, 14-29.
- Lewis R.J., 2000. An introduction to classification and regression tree (CART) analysis. Annual Meeting of the Society for Academic Emergency Medicine, San Francisco, California, May 22-25
- Loh, W.Y., 2011. Classification and regression trees, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1, 14-23.
- Lopez, P., and Roubellat, F. 2001. *Ordonnement de la production*, Hermès, Paris.
- Morgan, J.N., Sonquist, J.A., 1963. Problems in the analysis of survey data, and a proposal. *J. Am. Stat. Assoc.*, 58, 415-434.
- Page, E.H., Nicol, D.M., Balci, O., Fujimoto, R.M., Fishwick, P.A., L'Ecuyer, P., and Smith, R., 1999. An aggregate production planning framework for the evaluation of volume flexibility. *Winter Simulation Conference*, 1509-1520.
- Pritsker, A., and Snyder, K., 1994. *Simulation for planning and scheduling*, APICS, August.
- Quinlan, J.R., 1996. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, 77-90.
- Roder, P., 1994. Visibility is the key to scheduling success, *APICS Planning and Scheduling*, August
- Suri, R., Fu, B.R., 1994. On using continuous flow lines to model discrete production lines. *Discrete Event Dynamic Systems*, 4, 129-169.
- Thomas, A., Charpentier, P., 2005. Reducing simulation models for scheduling manufacturing facilities. *European Journal of Operational Research*, 161(1): 111-125.
- Thomas, P., Thomas, A., 2011. Multilayer Perceptron for Simulation Models Reduction: Application to a Sawmill Workshop. *Engineering Applications of Artificial Intelligence*, 24, 646-657.
- Thomas, P., Thomas, A., Suhner, M.C., 2011. A neural network for the reduction of a Product Driven System emulation model. *Production Planning and Control*, 22, 767-781.
- Vollmann, T.E., Berry, W.L., Whybark D.C., 1992. *Manufacturing, Planning and Systems Control. The Business One Irwin*.
- Ward, S.C., 1989. Argument for constructively simple models. *Journal of the Operational Research Society*, 40(2): 141-153.
- Zeigler, B.P., 1976. *Theory of modelling and simulation*. Wiley, New York.