Expérimentation de la réduction d'un modèle de simulation par réseau de neurones : cas d'une scierie

Philippe Thomas, André Thomas

7e Conférence Internationale de MOdélisation et SIMulation MOSIM'08, 31 mars au 2 avril, 2030-2038, Paris, France

« Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination.».

EXPERIMENTATION DE LA REDUCTION D'UN MODELE DE SIMULATION PAR RESEAU DE NEURONES : CAS D'UNE SCIERIE

Philippe THOMAS, André THOMAS

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
Philippe.thomas@cran.uhp-nancy.fr

RESUME : *La simulation est souvent utilisée pour évaluer la pertinence d'un Programme Directeur de Production (PDP) ou pour en évaluer l'impact sur des scénarii d'ordonnancement détaillé de la fabrication. Dans ce cadre, nous nous proposons d'étudier la construction d'un modèle de simulation en essayant d'en réduire la complexité. Pour ce faire nous avons exploité un réseau de neurones, plus particulièrement un perceptron multicouches, afin de remplacer tous les postes de charge qui ne sont pas contraintes de capacité du système de production par une boite noire. Cette approche sera appliquée au cas d'une scierie. Ce papier s'organise comme suit : après une introduction qui rappelle le contexte et les objectifs, nous présenterons la méthode employée et les perceptrons multicouches. Nous poursuivrons en présentant le cas d'application pour finalement présenter le modèle résultant et sa validation.*

MOTS-CLES : *perceptron multicouches, modèle réduit, simulation, réseau de neurones, re-ordonnancement*

1. INTRODUCTION

La simulation du fonctionnement d'un atelier est une aide précieuse pour l'aide à la décision en planification ou en ordonnancement. Elle permet, par exemple, de disposer de l'historique des variations de l'état des machines au cours du temps. On dispose donc des dates de tous les événements et en particulier des dates de début et de fin des opérations. Ces informations peuvent alors être très utiles pour l'établissement d'un ordonnancement prévisionnel (Lopez et Roubellat 2001). Par ailleurs, des événements, aléas, peuvent remettre en cause le PDP ce qui va conduire à une modification de ce dernier et donc, à la nécessité d'effectuer un re-ordonnancement des tâches. Les systèmes temps réel de suivi de fabrication permettant aujourd'hui de collecter très rapidement les informations corrélées à l'apparition de ces aléas dans le système de gestion (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). Enfin, il s'agira de maximiser le taux de charge des goulots. Dans une philosophie de gestion de type « management par les contraintes », celui-ci est le critère principal pour l'évaluation d'un PDP. Pour ces raisons, il est intéressant d'utiliser la simulation dynamique des flux (Thomas et Charpentier 2001).

Actuellement, les simulations sont effectuées avec des modèles de plus en plus complexes, ceci grâce au développement des moyens de calcul informatique. Cependant, les modèles complexes obligent leurs concepteurs à faire face à des problèmes pour lesquels ils n'ont pas été formés, et que Page *et al*. (1999) appellent « *Problems of Scale* ». Aussi, de nombreux auteurs ont rappelé l'importance d'utiliser des modèles de simulation simples (Ward 1989), (Musselman 1993), (Pidd 1996), (Brook et Tobias 2000), (Chwif *et al.* 2006).

D'autre part, si la décision est prise en quelques heures seulement, pendant ce temps même court, l'état de l'atelier a changé (Thomas et Charpentier 2001). Il ne faut donc pas que le temps passé à réaliser les simulations en prenant en compte les nouvelles données pour effectuer le re-ordonnancement soit pénalisant. Aussi, nous posons comme hypothèse que, dans ce contexte, il s'agit de trouver un modèle suffisamment simple pour gagner du temps lors de sa construction, de son paramétrage, de l'exécution des scenarii et de l'interprétation des résultats, mais suffisamment significatif pour minimiser le risque de prendre une mauvaise décision (Thomas et Charpentier 2001).

Par ailleurs, les réseaux de neurones, grâce à leur capacité d'être des « approximateurs universels », ont prouvé leur capacité à extraire de données d'expérimentation des modèles performants, sans avoir à effectuer d'hypothèses sur la forme générale de ces derniers (Thomas *et al.* 1999). Aussi, ils commencent à faire leur apparition dans le cadre général de la supply chain (Shervais *et al.* 2003), (Chiu et Lin 2004).

Notre travail a donc pour but de montrer l'intérêt d'utiliser les réseaux de neurones pour réduire les modèles de simulation, que ce soit pour l'établissement ou l'utilisation de ces modèles que dans le but de faciliter la prise de décision en re-ordonnancement.

Nous allons maintenant présenter la méthode de réduction de modèle proposée ainsi que les perceptrons multicouches utilisés. Nous poursuivrons en présentant le cas d'application qui est une scierie spécialisée dans la

découpe de conifères. Nous finirons en présentant le modèle de simulation obtenu que nous comparerons au modèle décrivant complètement la scierie.

2. LA REDUCTION DE MODELE

2.1. L'algorithme

La problématique de la réduction de modèle a été initiée par Zeigler (1976) pour qui la complexité d'un modèle est relative au nombre d'éléments, de connexions et de calculs du modèle.
Les premières techniques de simplification en modélisation ont été proposées par Innis et Rexstad (1983). Depuis lors, un certain nombre d'auteurs se sont intéressés à ce problème dans des cadres particuliers. Ainsi, un modèle réduit dans un cas relatif à l'industrie électronique qui utilise le temps de cycle comme indicateur de mesure a été proposé (Leachman 1986). Brooks et Tobias (2000) ont, eux, proposé une approche de simplification de modèle pour les cas où les indicateurs à suivre sont les moyennes des volumes de production « throughput rates ». D'autres cas ont été étudiés notamment par Hung et Leachman (1999), Hwang *et al*. (1999) ou encore Thomas et Charpentier (2005).
L'algorithme de réduction proposé est une extension de celui proposé par Thomas et Charpentier (2005) :

1. Identifier le goulet structurel (Poste de charge (PdC) qui, depuis plusieurs années, est majoritairement contrainte de capacité).
2. Pour la liasse d'Ordre de Fabrication (OF) du PDP considéré, identifier le goulet conjoncturel.
3. Parmi les autres PdC, identifier ceux (postes de synchronisation) satisfaisant aux deux conditions :
   - utilisé conjointement à un poste goulet 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 par un réseau de neurones l'intervalle situé entre chacun des PdC précédemment trouvés.

2.2. Le perceptron multicouches

Les travaux de Cybenko (1989) et de Funahashi (1989) ont montré qu'un réseau de neurones multicouches possédant une seule couche cachée utilisant une fonction d'activation sigmoïdale et une couche de sortie utilisant une fonction d'activation linéaire peut approximer toute fonction non linéaire avec la précision voulue si on utilise suffisamment de neurones dans la couche cachée. Ce résultat important explique la grande popularité de ce type de réseau de neurones que l'on appelle classiquement un perceptron multicouches.
Nous allons ici rappeler la structure du perceptron multicouches utilisé. Le réseau présenté par la figure 1 est composé de neurones interconnectés en trois couches successives. La première couche est composée de neurones « transparents » qui n'effectuent aucun calcul mais simplement distribuent leurs entrées à tous les neurones de la couche suivante appelée couche cachée.

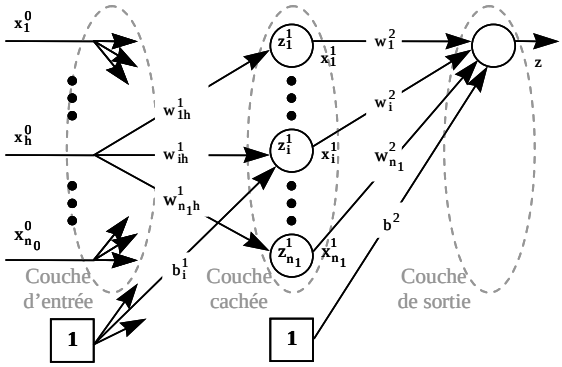

Figure 1. L'architecture du perceptron multicouches

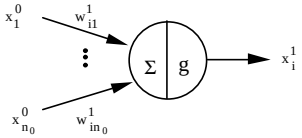

Figure 2. Le neurone i de la couche cachée

Les neurones de la couche cachée (figure 1), dont un exemple peut être représenté par la figure 2, reçois les n_0 entrées $\{x_1^0,\cdots,x_{n_0}^0\}$ de la couche d'entrée avec les poids associés $\{w_{i1}^0,\cdots,w_{in_0}^0\}$. Ce neurone commence par calculer la somme pondéré de ses n_0 entrées :

$$z_i^1 = \sum_{h=1}^{n_0} w_{ih}^1 . x_h^0 + b_i^1 \qquad (1)$$

où b_i^1 est un biais (ou seuil). La sortie du neurone caché est obtenue en transformant la somme (1) par l'intermédiaire de la fonction d'activation g(.) :

$$x_i^1 = g\left(z_i^1\right). \qquad (2)$$

Bien que de nombreuses fonctions d'activations ait été proposées, la fonction g(.) est généralement la tangente hyperbolique (Thomas 1997) :

$$g(x) = \frac{2}{1+e^{-2x}} - 1 = \frac{1-e^{-2x}}{1+e^{-2x}} . \qquad (3)$$

Le neurone de la dernière couche (ou couche de sortie) utilise une fonction d'activation linéaire et n'effectue donc qu'une simple somme pondérée de ses entrées :

$$z = \sum_{i=1}^{n_1} w_i^2 . x_i^1 + b \qquad (4)$$

où w_i^2 sont les poids connectant les sorties des neurones cachés au neurone de sortie et b est le biais du neurone de sortie.
La détermination des poids et biais du réseau qui en sont les paramètres s'effectuent par l'intermédiaire d'un apprentissage supervisé. Cet apprentissage s'effectue en deux phases :

- une phase d'initialisation. Le choix des poids et biais initiaux peut s'effectuer de manière aléatoire ou utiliser des algorithmes plus complexes (Thomas et Bloch 1997). Nous utiliserons ici l'algorithme de Nguyen et Widrow (1990),
- une phase d'apprentissage. Une fois de plus de nombreux algorithmes existent. Nous utiliserons l'algorithme de Levenberg-Marquadt qui se comporte comme un algorithme du Hessien lorsque l'on se trouve loin de la solution et comme un algorithme du gradient lorsque l'on s'en rapproche (Thomas et Bloch 1996).

3. LA SCIERIE

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.
En 2001, cette scierie avait une capacité de 270 000 m^3/an. Elle générait un chiffre d'affaire de 53 millions d'euros et employait 300 personnes.

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 3. 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 parallélépipède rectangle, l'équarri. Cette première étape, qui forme les deux premiers cotés 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 cotés du parallélépipède. 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'ébouteur.

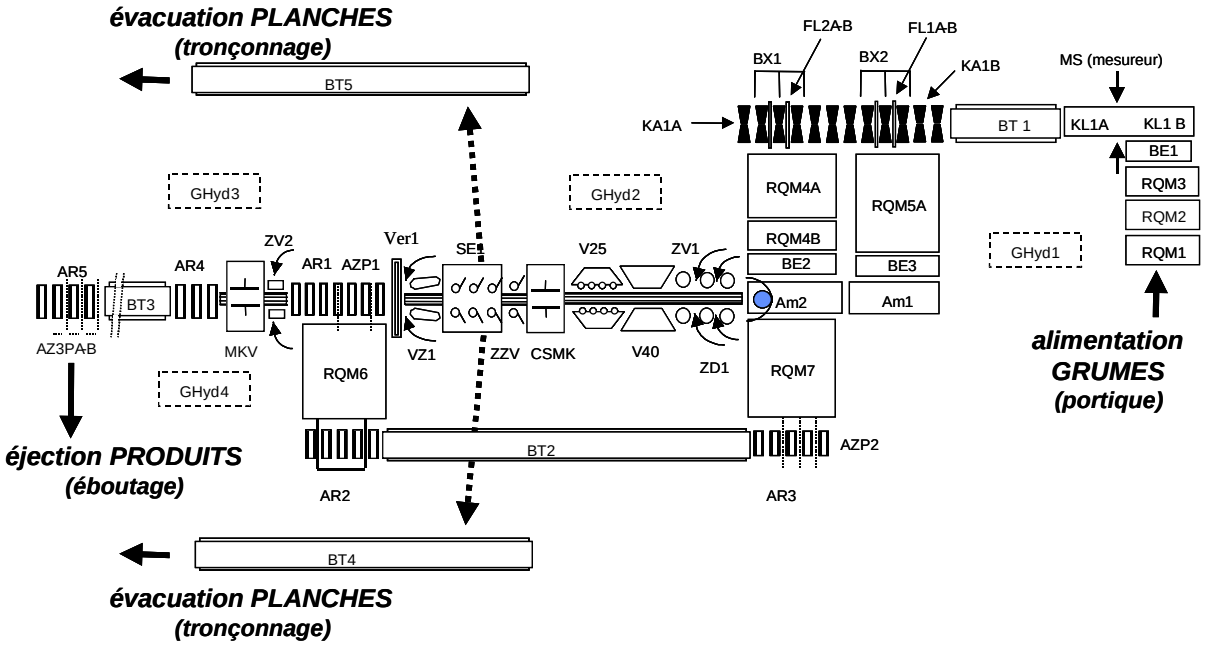

Figure 3. La ligne canter

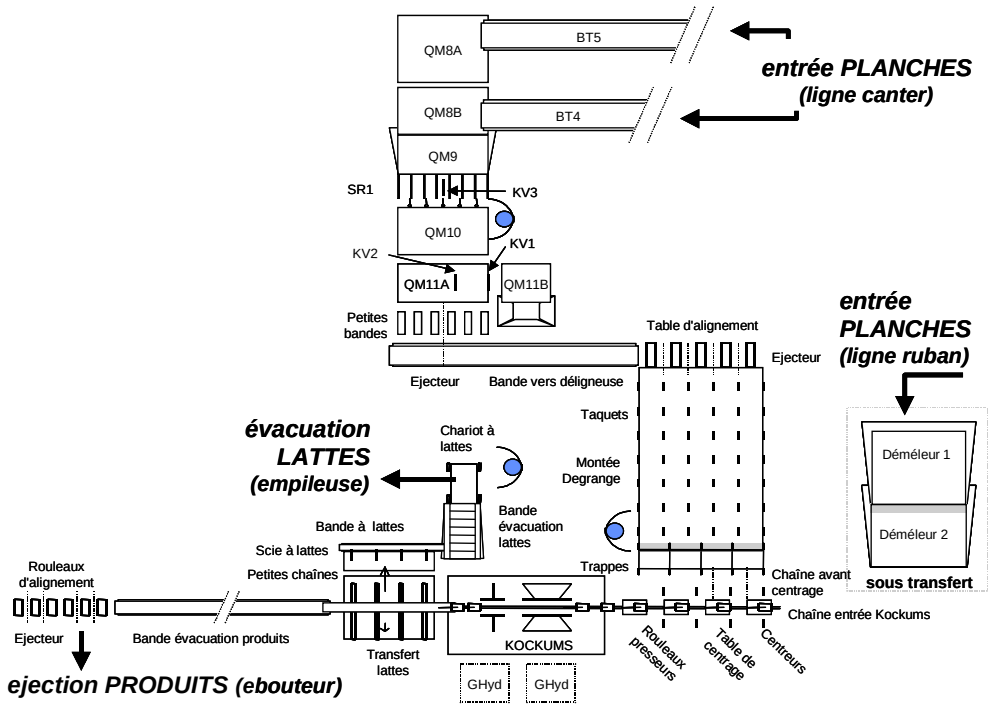

Figure 4. La tronçonneuse-deligneuse (Kockums)

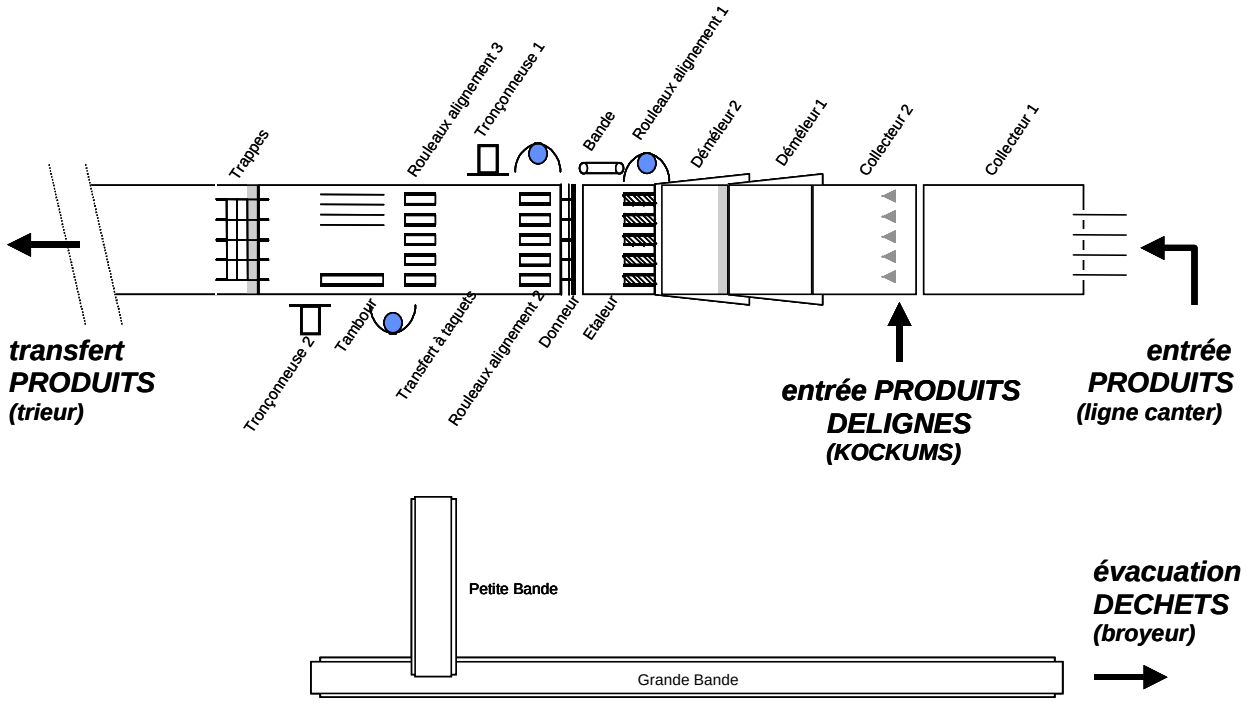

Figure 5. L'ébouteur

La figure 4 présente la deuxième partie de la ligne de sciage qui correspond au tronçonnage délignage et dont la machine principale est la scie Kockums. Seul 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 5. En entrée de l'ébouteur se trouvent deux collecteurs (collecteur 1 et collecteur 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 démêlées et mises sur un tapis à taquets.

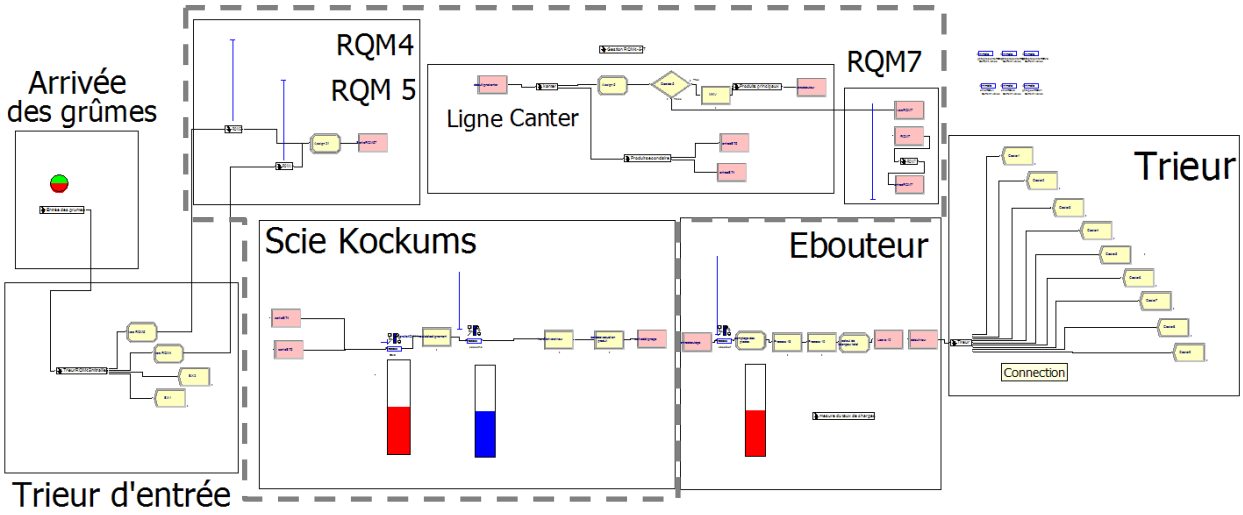

Figure 6. Le modèle complet

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 2001) ont montré que c'est cette machine, l'ébouteur, qui est le poste goulot de la chaîne de sciage.

4. LES MODELES DE SIMULATION

4.1. Le logiciel Arena®

Arena® est développé par la société Rockwell Software®. C'est un logiciel de simulation de systèmes à événement discrets basé sur la théorie des files d'attentes et s'appuyant sur le logiciel Siman®.
Il peut être associé à des logiciels aussi divers que Excell® ou Visual Basic® ce qui facilite le développement de projets complexes.

4.2. Le modèle complet

Des travaux précédents (Thomas et Charpentier 2001) ont permis de construire le modèle complet de la ligne de sciage. Ce modèle est présenté à la figure 6.
Ce modèle intègre un module de gestion de l'arrivée des grumes suivant un processus poissonien. Dans ce module, les caractéristiques de la grume qui sont relevées par le mesureur (figure 3) sont associées à celle-ci. Un deuxième module, le « trieur d'entrée » permet d'orienter la grume soit vers RQM4 soit vers RQM5 en fonction de ses caractéristiques, soit d'éjecter la grume du process si celle-ci n'entre pas dans les caractéristiques de la gamme (grume trop grosse ou trop petite) ou si elle a été mitraillée. Les grumes sont donc ensuite envoyées sur RQM4 ou RQM5 qui servent de stocks d'entrée à la ligne canter (de même que RQM7).

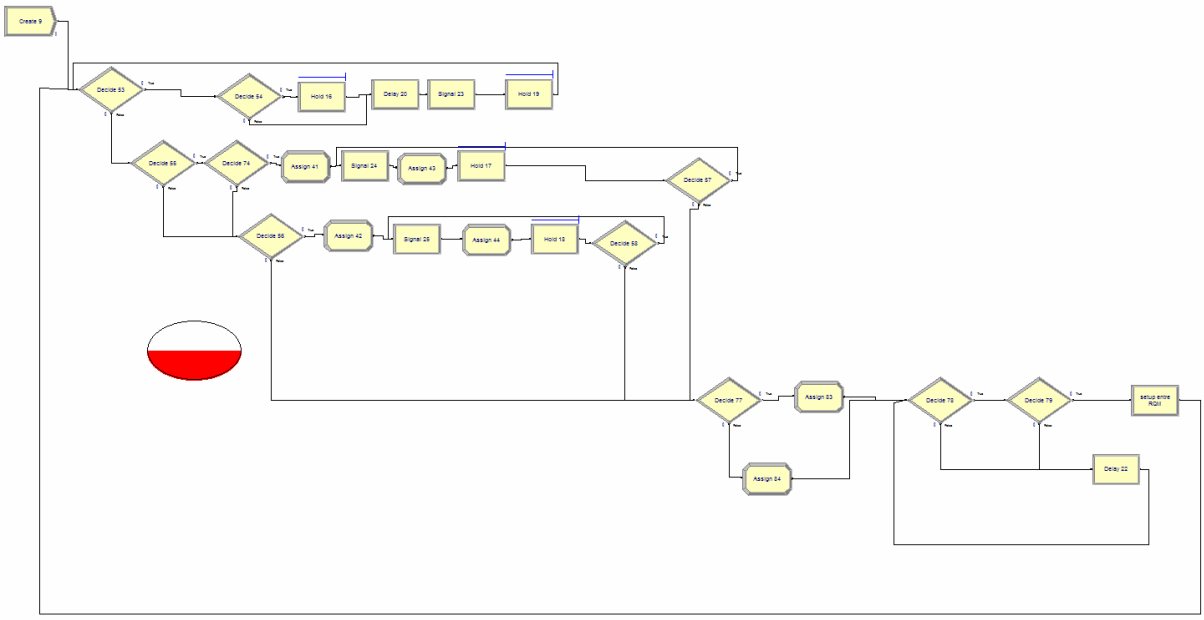

Figure 7. Gestion RQM4-5-7

La gestion de la ligne Canter et du passage sur RQM7 font l'objet de deux modules supplémentaires. La modélisation de la ligne Canter emploie deux sous modèles qui permettent de gérer le traitement des produis principaux et celui des produits secondaires. Nous avons donc trois sorties pour la ligne Canter qui sont reliées, soit aux entrées de la scie kockums pour les produits secondaires, soit directement à l'entrée du module ébouteur pour les produits principaux. Un sous modèle, présenté figure 7, permet de gérer les règles de priorité pour le choix du stock d'entré qui approvisionne la ligne Canter, soit RQM4, soit RQM5, soit RQM7. Ce sous modèle donne une bonne idée de la complexité des différents sous modèles qui ont été construits pour gérer la ligne Canter notamment. Les autres modules, bien que représentant le cœur de la ligne de sciage avec la scie kockums et l'ébouteur, sont des modules assez simples à construire. Enfin, un dernier module permet de gérer et de simuler le tri et le rangement des produits finis et il n'a pas d'incidence réelle sur le fonctionnement de la ligne de sciage.

4.3. Le modèle réduit

Comme nous venons de le voir, la réalisation d'un modèle de simulation complet pour une installation finalement assez simple est déjà une tâche assez complexe et produit un modèle lourd à gérer.
Or, nous savons que le poste goulot d'étranglement de cette ligne de sciage est l'ébouteur (Thomas et Charpentier 2001). Donc, un pilotage optimal de la ligne nécessite d'optimiser la charge du poste goulot, en accord avec le principe donné par la méthode O.P.T. que « toute heure perdue sur un poste goulot est une heure perdue pour le système. » (Goldratt et Fox 1992), (Marris 1994).
Dans ce cadre, il semble bien que la modélisation des stocks RQM4 et 5, de la ligne Canter, du stock RQM7, de la gestion des RQM4-5 et 7 et même de la scie kockums soit superflue. Donc tout la partie entourée par des tirets gris sur la figure 6 (et qui comprend en particulier le sous modèle de la figure 7) n'apporte pas d'informations directe et utile pour l'évaluation d'un PDP ou la mise en place d'un re-ordonnancement pour lesquels nous aurions des indicateurs de performance tels que délai global ou taux de charge du goulot... En effet, seul les instants d'arrivée des produits en entrée de l'ébouteur vont nous permettre de simuler la charge de ce dernier. C'est pourquoi, nous proposons de remplacer toute la partie entourée de tiret gris sur la figure 6 par un perceptron multicouches qui va transformer les informations en entrée du modèle et fournies par les modules « arrivée des grumes » et « trieur d'entrée » en des instants d'arrivées de produits en entrée de l'ébouteur sans que les chemins suivis ou les transformations subies soient apparentes.
Pour ce faire, il est nécessaire de collecter les données disponibles en entrée du système. Tout d'abord, chaque grume porte en elle, un certain nombre d'information qui sont collectées par le scaner en entrée de ligne Canter. Cette étape nous fournit les informations dimensionnelles de la grume que sont sa longueur (longueur) et trois valeurs de diamètre, diamètre en son extrémité la plus fine (diamPetitBout), diamètre en son extrémité la plus grosse (diamGrosBout), et le diamètre moyen (diamMoyen). Ces informations, nous indiquent entre autre immédiatement si la grume doit être dirigée vers RQM4 ou vers RQM5 ce qui est une information supplémentaire (RQM).
Outre ces informations dimensionnelles de la grume, nous pouvons disposer d'informations sur l'état de la ligne à l'instant de l'entrée de la grume considérée. Les informations relevées sont la taille du stock d'entrée de l'ébouteur (Q_ebouteur), le taux d'utilisation de l'ébouteur (taux_ebouteur) et le nombre de grumes présentes dans le système entre les entrées des RQM4 et 5 et la sortie de la ligne Canter, donc principalement une image de la somme des stocks sur RQM4, RQM5 et RQM7 et des grumes en cours de traitement sur les différentes machines de la ligne Canter (Q_RQM).
Le dernier type d'information disponible est celui lié au plan de coupe des grumes. En effet chaque grume va être découpée en *n* produits qui peuvent être des produits secondaires ou principaux. Le plan de coupe qui nous a intéressé ici est un plan ou 7 produits sont coupés dans chaque grume :

- 2 produits secondaires lors du premier passage de la grume sur la sur déligneuse CSMK (figure 3),
- 2 produits secondaires lors du deuxième passage de la grume sur la déligneuse CSMK (figure 3),
- 3 produits principaux après passage de l'équarri sur la déligneuse MKV.

Ces 7 produits peuvent donc être décomposés en trois types de produits selon l'endroit de la découpe, CSMK ou MKV, et l'instant de la découpe, premier ou deuxième passage (type_piece). La dernière donnée est l'épaisseur (en mm) du produit réalisé qui sert aussi de dénomination. Dans notre cas, nous ne produisons que deux types de produits, les produits principaux sont du 75, les secondaires sont du 25 (produit).
Nous disposons donc d'un jeu de dix variables d'entrées (longueur ; diamGrosBout ; diamMoyen ; diamPetitBout ; produit ; type_piece ; Q_ebouteur ; taux_ebouteur ; Q_RQM ; RQM) que l'on va pouvoir associer aux 12775 pièces produites. Ces dix variables vont constituer les dix entrées du perceptron multicouches.
Ces 12275 pièces vont mettre un certain temps entre le moment où elles entrent dans le système sous forme de grume et le moment où elles entrent dans le stock d'entrée de l'ébouteur. C'est ce temps ΔT que nous cherchons à déterminer. C'est donc lui qui constituera la sortie de notre réseau de neurones.
Le nombre de neurones dans la couche cachée à été déterminé par essai - erreur. Ces différents essais nous ont conduits à choisir de placer 30 neurones dans la couche cachée.
La détermination des paramètres du réseau de neurone s'effectuant par apprentissage supervisé, il est nécessaire de diviser la base de données précédemment constituée

en deux jeux. Le premier, appelé jeu d'apprentissage, va nous servir à effectuer l'apprentissage proprement dit. Le deuxième, appelé jeu de validation, nous servira à vérifier que l'apprentissage s'est correctement passé, en particulier, à vérifier qu'il n'y a pas eu de phénomène de sur-apprentissage.

4.4. Les résultats

Nous allons maintenant présenter les résultats obtenus avec le perceptron multicouches comparativement à ceux fournis par le modèle complet.
Comme nous l'avons dit précédemment, nous avons divisé notre base de données en deux jeux différents. Cette division s'est effectuée de manière aléatoire et nous avons utilisées les informations en provenance de 6365 pièces pour constituer le jeu d'apprentissage et 6410 pour le jeu de validation. Nous allons commencer par présenter les résultats obtenus sur le jeu de données d'apprentissage.

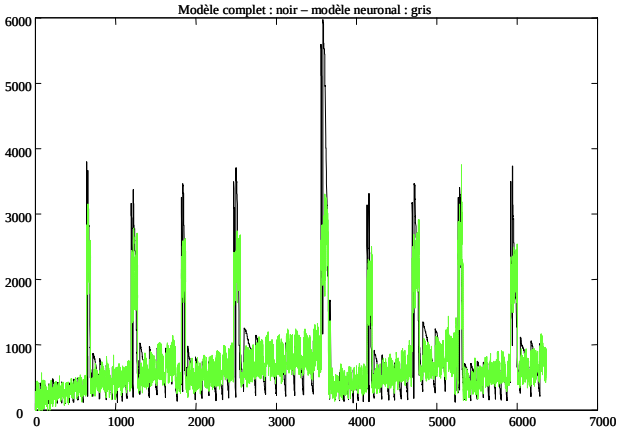

Figure 8. Temps de transfert – jeu d'apprentissage

La figure 8 présente les temps (en ordonnée) qu'ont mis les 6365 pièces (en abscisse) constituant le jeu d'apprentissage 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 noir les temps fournis par le modèle complet qui nous sert de référence, et en gris les temps fournis par le modèle neuronal. 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 le modèle neuronal sont tout à fait similaires à ceux donnés par le modèle complet.
La figure 9, qui présente l'erreur effectuée par le modèle neuronal, comparativement au modèle complet confirme bien ces premières constatations. Nous pouvons tout de même noter que le modèle neuronal présente quelques difficultés à estimer de manière très précise les temps mis par quelques pièces qui ont passées un temps assez long sur la chaîne. Cela peut s'expliquer par la non prise en compte de certaines variables pouvant avoir une incidence particulière sur ces données comme, la présence de plus ou moins de grumes mitraillées, la répartitions des grumes sur les différents RQM, …

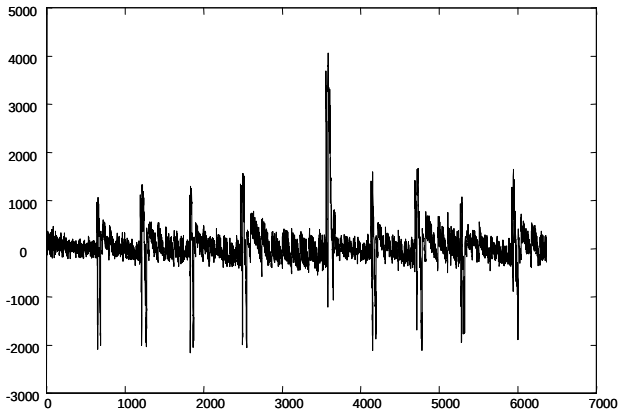

Figure 9. Erreur effectuée sur le jeu d'apprentissage

L'erreur moyenne effectuée sur le jeu d'apprentissage est cependant très faible puisqu'elle est seulement de 7,9 secondes (avec cependant un écart type important : 430 caractéristique des quelques grandes erreurs mal modélisées).

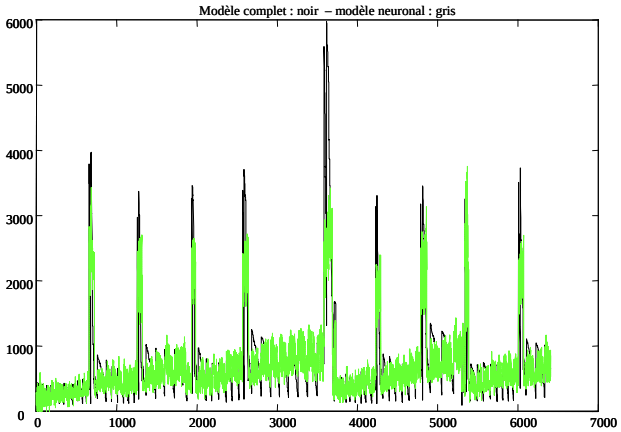

Figure 10. Temps de transfert – jeu de validation

Ces résultats obtenus sur le jeu d'apprentissage prouvent que l'utilisation d'un réseau de neurones permet d'estimer les temps recherchés. Il est toutefois nécessaire de vérifier que ces résultats peuvent s'étendre à d'autres données. La figure 10 présente les temps (en ordonnée) qu'ont mis les 6410 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 noir les temps fournis par le modèle complet qui nous sert de référence, et en gris les temps fournis par le modèle neuronal. Les temps sont toujours indiqués en secondes.
Ces résultats, très similaires à ceux obtenus sur le jeu d'apprentissage, prouvent que le modèle neuronale peut tout à fait s'adapter à d'autres données que celles du jeu d'apprentissage sans présenter de phénomènes de sur-

apprentissage. L'erreur moyenne reste très faible (9,4 secondes) quand à l'écart type, il reste dans les mêmes proportions que pour le jeu d'apprentissage à 448 et ce pour les mêmes raisons.

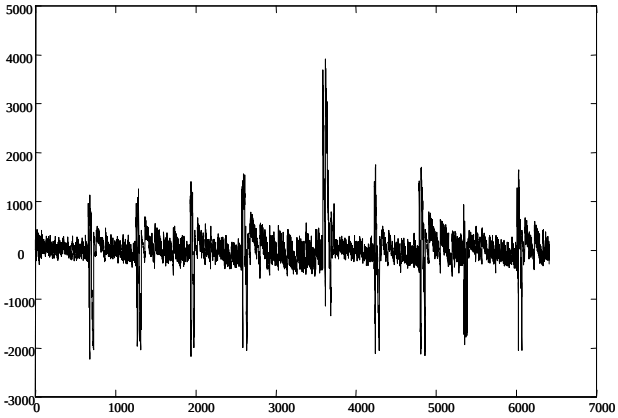

Figure 11. Erreur effectuée sur le jeu de validation

L'ensemble de ces résultats montrent bien qu'il est tout à fait possible de remplacer une partie importante d'un modèle de simulation complexe par un modèle neuronal beaucoup plus simple à construire.

5. CONCLUSION

Dans cet article, nous avons présenté une nouvelle approche de réduction de modèle de simulation en utilisant un réseau de neurones pour modéliser, sous forme d'une boîte noire, le comportement de la partie de la chaîne ne présentant pas ou peu d'intérêt.
Cette approche a été appliquée au cas d'une ligne de sciage.
Les résultats ont montré que :

- les deux jeux de données (apprentissage et validation) donnent des résultats similaires,
- la moyenne de l'erreur (inférieur à 10 secondes) est petite comparativement à la dynamique du processus (de l'ordre de 2000 secondes).

Ces résultats montrent bien l'intérêt d'utiliser un réseau de neurones afin de modéliser une partie du process plutôt que de construire un modèle complet. Cette approche, plus simple et plus rapide, permet de se focaliser sur le cœur du problème que sont les goulets d'étranglement (ici l'ébouteur).
Dans nos travaux futurs, nous allons mettre l'accent sur la détermination de la structure optimale du réseau de neurones, en particulier, du choix des entrées du réseau, et sur la validation de notre approche à des cas d'application différents.

REFERENCES

Brooks R.J., and A.M. Tobias, 2000. Simplification in the simulation of manufacturing systems. *Int. J. Prod. Res.*, 38(5), p. 1009-1027.

Charpentier P., and A. Thomas, 2001. Model reducing method for the scheduling decision-making. *Proceedings of the 1993 Winter Simulation Conference*, p. 58-64.

Chiu M., and G. Lin., 2004. Collaborative supply chain planning using the artificial neural network approach. *Journal of Manufacturing Technology Management*, 15(8), p. 787-796.

Chwif L., R.J. Paul, and M.R. Pereira Barretto, 2006. Discret event simulation model reduction: A causal approach. *Simulation Modelling Practice and Theory*, 14, p. 930-944.

Cybenko G, 1989. Approximation by superposition of a sigmoïdal function. *Math. Control Systems Signals*, 2(4), p. 303-314.

Funahashi K, 1989. On the approximate realisation of continuous mapping by neural networks. *Neural Networks*, 2, p. 183-192.

Goldratt E., and J. Cox, 1992. *The Goal : A process of ongoing improvement*, North River Press; 2nd Revised edition, Great Barrington, USA, ISBN : 978-0884270614.

Hung Y.F., and R.C. Leachman, 1999. Reduced simulation models of wafer fabrication facilities. *Int. J. Prod. Res.*, 37, p. 2685-2701.

Hwang J.S., S. Hsieh, and H.C. Chou, 1999. A Petri net based structure for AS/RS operation modeling. *Int. J. Prod. Res.*, 36, p. 3323-3346.

Innis G.S., and E. Rexstad, 1983. Simulation model simplification techniques. *Simulation*, 41, p. 7-15.

Khouja M., 1998. An aggregate production planning framework for the evaluation of volume flexibility. *Production Planning and Control*, 9(2), p. 127-137.

Leachman R.C., 1986. *Preliminary design and development of a corporate level production planning system for the semi conductor industry*, Eds Optimization in industry, Chichester, UK.

Lopez P., and F. Roubellat, 2001. *Ordonnancement de la production*, Hermès, Paris, ISBN : 2-7462-0184-4.

Marris P., 1994. *Le Management par les contraintes en gestion industrielle. Trouver le bon déséquilibre*, Editions d'Organisation, Paris, http://www.management-par-les-contraintes.com/.

Musselman K.J., 1993. Guideline for simulation project success. *Proceedings of the 1993 Winter Simulation Conference*, p. 58-64.

Nguyen D., and B. Widrow, 1990. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptative weights. *Proc. of the International Joint Conference on Neural Networks IJCNN'90*, 3, p. 21-26.

Page E.H., D.M. Nicol, O. Balci, R.M. Fujimoto, P.A. Fishwick, P. L'Ecuyer, and R. Smith, 1999. An aggregate production planning framework for the evaluation of volume flexibility. *Proceedings of the 1999 Winter Simulation Conference*, p. 1509-1520.

Pidd M., 1996. Five simple principles of modelling. *Proceedings of the 1996 Winter Simulation Conference*, p. 721-728.

Pritsker A., and K. Snyder, 1994. Simulation for planning and scheduling. *APICS*, August.

Roder P., 1994. Visibility is the key to scheduling success. *APICS Planning and Schedulling*, August.

Shervais S.; T.T. Shannon, and G.G. Lendaris, 2003. Intelligent supply chain management using adaptive critic learning. *IEEE Trans. On Systems, Man and Cybernetics, Part A Systems and Human*, 33(2), p. 235-244.

Thomas A., and Charpentier P., 2001. De la pertinence de modèles réduits pour la prise de decision en réordonnancement. *Proceeding of the 2nd International Conference on Integrated Design and Production CPI'01*, Fès, Maroc.

Thomas A., and Charpentier P., 2005. Reducing simulation models for scheduling manufacturing facilities. *European Journal of Operational Research*, 161(1), p. 111-125.

Thomas, P., 1997. *Contribution à l'identification de systèmes non linéaires par réseaux de neurones.* Thèse de Doctorat, Université Henri Poincaré Nancy1, France.

Thomas P., and G. Bloch, 1996. From batch to recursive outlier-robust identification of non-linear dynamic sys-tems with neural networks. *Proc. of the IEEE International Conference on Neural Networks ICNN'96*, Washington D.C., USA, 1, p. 178-183.

Thomas P., and G. Bloch, 1997. Initialization of one hidden layer feed-forward neural networks for non-linear system identification. *Proc. of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics WC'97*, Belin, Germany, p. 295-300.

Thomas P., G. Bloch, F. Sirou, and V. Eustache, 1999. Neural modeling of an induction furnace using robust learning criteria. *Journal of Integrated Computer Aided Engineering*, 6(1), p. 5-23.

Ward S.C., 1989. Argument for constructively simple models. *Journal of the Operational Research Society*, 40(2), p. 141-153.

Zeigler B.P., 1976. *Theory of modelling and simulation*, Wiley, New York.