UNIVERSITÉ DE MONTRÉAL

OPTIMIZATION OF BROKE RECIRCULATION IN A NEWSPRINT MILL

MICHAL DABROS
DÉPARTEMENT DE GÉNIE CHIMIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CHIMIQUE)
SEPTEMBRE 2002

0-612-76992-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMIZATION OF BROKE RECIRCULATION IN A NEWSPRINT MILL

présenté par : DABROS Michal
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par jury d'examen constitué de :

M. CARREAU Pierre J., Ph.D., président
M. PERRIER Michel, Ph.D., membre et directeur de recherche
M. STUART Paul, Ph.D., membre et codirecteur de recherche
M. GENDRON Sylvain, Ph.D., membre

DÉDICACE

À ma famille.

REMERCIEMENTS

Je tiens à remercier :

Michel Perrier pour son appui et la supervision de ce projet,

Paul Stuart et Fraser Forbes pour leurs conseils et leur aide,

Martin Fairbank, Pascal Hébert et Jérôme Claveau de Abitibi Consolidated Inc. pour leur implication et leur aide,

et enfin, Agnès Dévarieux pour sa patience et les étudiants de la Chaire en Intégration des procédés dans l'industrie papetière, de l'URCPC et du CRIP pour l'ambiance.

RÉSUMÉ

Tâchant de conserver leur avantage concurrentiel, les manufacturiers de papier journal examinent les moyens de réduire le nombre de casses de feuille et de maximiser la productivité de leur procédé. Les variations dans les propriétés de la pâte mixte peuvent déranger la stabilité d'une machine à papier et contribuer aux causes de casse de feuille. Ces fluctuations sont souvent le résultat de la gestion du recyclage des cassés appliquée dans l'usine. Plus spécifiquement, les changements rapides du taux de pâte cassée peuvent induire des variations dans les propriétés de la pâte mixte à la caisse d'arrivée d'une machine à papier.

La gestion de recyclage des cassés peut être améliorée par l'application de l'optimisation basée sur modèles. Les méthodes conventionnelles d'optimisation fonctionnent en minimisant une fonction objectif, habituellement sous contraintes. Les algorithmes standard d'optimisation exigent une fonction objectif analytique, ainsi que ses premières et/ou deuxièmes dérivées. Les modèles de procédé non analytiques (une simulation, par exemple), ne disposent que des valeurs de la fonction objectif. Pour de telles applications, les méthodes de recherche directe servent d'alternative. L'optimisation par recherche directe peut être considérée comme une méthode selon laquelle les variables d'entrée d'un modèle dynamique de processus sont sondées systématiquement de façon à rechercher les régions de fonctionnement quasi-optimales. Un indice de performance est

mesuré dans ces régions et, après un nombre d'expériences suffisant, le régime de fonctionnement qui rend la performance acceptable peut être trouvé.

Cette étude se concentre sur le développement d'un système d'optimisation par recherche directe afin d'améliorer la gestion de recyclage des cassés dans une usine de papier journal. L'usine considérée dans cette étude comporte quatre machines à papier et produit environs 1000 t/d de papier journal à partir de pâtes thermomécanique et desencrée. La pâte cassée est réutilisée dans le procédé avec un taux approprié permettant le maintien d'un inventaire souhaitable dans le circuit de cassés. Dans le processus actuel de recyclage de cassés, la proportion de pâte cassée est souvent ajustée abruptement, entraînant ainsi un impact considérable sur les propriétés de la pâte mélangée alimentée aux machines à papier. L'objectif de cette étude est de développer une simulation dynamique du procédé afin de mesurer ces effets. Deux algorithmes de recherche directe (*Nelder-Mead Simplex* et Algorithmes Génétiques) sont alors appliqués afin de déterminer des profils améliorés de réglage du taux de pâte cassée et de minimiser les variations à la caisse d'arrivée.

La première étape de l'étude a consisté en la construction d'une simulation dynamique de la section de fabrication de papier de l'usine en utilisant le logiciel WinGEMS 5.0. Ce modèle a été validé d'une façon non formelle. L'effet de la gestion actuelle du recyclage de cassés sur les propriétés de la pâte à la caisse d'arrivée des machines à papier a été examiné en utilisant une fonction objectif. Une interface dynamique a été alors construite

pour relier la simulation aux algorithmes de recherche directe qui ont été programmés dans Visual Basic. Le profil du changement dans le taux de recyclage a servi comme degré de liberté pour les algorithmes. Une étude de cas a été exécutée pour une période de 24 heures afin de mesurer la diminution des variations dans les propriétés de pâte mélangée aux caisses d'arrivée.

Au terme de cette étude, nous avons observé que les fluctuations dans les propriétés de la pâte mixte peuvent être réduites de manière significative avec un taux de changement de la proportion de cassés plus graduel. Les deux méthodes de recherche directe ont montré un profil quasi-optimal très semblable de ce taux. Il est recommandé que le processus actuel de recyclage de pâte soit analysé à l'usine et que les résultats de cette étude soient mis en œuvre dans le procédé. Nous supposons ici que la réduction des variations à la caisse d'arrivée entraîne une augmentation de la stabilité globale de la machine à papier et une réduction du nombre de casses de papier.

ABSTRACT

Striving to maintain their competitive advantage, newsprint manufacturers are looking into ways of effectively reducing the number of paper machine breaks and maximizing the run-time of their process. Fluctuations in the pulp furnish can upset the stability of a paper machine and are one of the causes of sheet breaks. These fluctuations are often the result of the broke recirculation strategies applied at the newsprint mill. Specifically, sudden changes to the broke ratio can induce variations in the properties of the mixed pulp at the paper machine headbox.

The broke recirculation strategies can be improved by the application of simulation-based optimization. Conventional optimization schemes work by minimizing an objective function, usually under constraints. Standard optimization algorithms require an analytical objective function as well as its first and/or second derivatives. For process models that are not equation-based (like simulations), only the objective function values are available and the derivatives of the function are not available. For applications like this, direct search methods surface as the alternative. Direct search optimization can be thought of as a method of systematically probing the input variables of a dynamic process model in search of near-optimum operating regions. A performance index is measured in these regions, and given enough experimentation, the operating regime that yields acceptable performance can be found.

This thesis focuses on developing and testing a direct search optimization system for the broke recycling strategy at a newsprint mill. The mill considered in this study runs four paper machines and produces newsprint at a rate of about 1000 t/d from thermomechanical and de-inked pulp. Broke pulp is recycled back into the process at rates appropriate to maintain a desirable inventory in the broke storage system. The broke recirculation is currently managed in a non-optimal fashion. Namely, the broke ratio is often adjusted abruptly, causing considerable impact on the properties of the mixed pulp fed into the paper machines. The objective of this study is to develop a dynamic simulation of the process in order to quantify the effects of the current broke recirculation policy. Direct search algorithms (Nelder-Mead Simplex and Genetic Algorithms) are then applied in order to find improved profiles of the broke ratio adjustments and to minimize headbox variations.

The first step of the project involved constructing a dynamic simulation of the papermaking section of the mill using WinGEMS 5.0 software and validating the model as required. The effect that the current broke recirculation strategy has on the pulp properties at the paper machine headboxes was examined using an objective function. A dynamic interface was then built to link the simulation to the direct search algorithms that were programmed using Visual Basic. The profile of broke ratio adjustments was left as the degree of freedom for the algorithms. A case study was run for a 24 hour period in order to quantify the decrease in pulp property variations at the paper machine headboxes observed with the improved broke recirculation policy.

The most significant finding of this work is that changing the broke ratio more gradually and over a longer period of time significantly reduces the fluctuations in the paper machine furnish. The two direct search methods found nearly the same near-optimum broke ratio adjustment profiles and yielded very similar results. It is recommended that the current broke recirculation policy be revised at the mill and that the results of this study be implemented in the process. It is hypothesized that with smaller variations at the headbox, the overall machine stability can be increased and the number of paper breaks reduced.

CONDENSÉ EN FRANÇAIS

La modélisation et l'optimisation sont des outils d'analyse de procédé importants dans la recherche, ainsi que dans l'industrie. L'efficacité et la productivité d'un procédé peuvent être augmentées de manière significative en appliquant un système d'optimisation basé sur modèle. Cette technologie trouve de plus en plus son application dans l'industrie papetière, donnant un avantage dans ce secteur d'industrie concurrentiel.

Dans les usines de papier journal, la perturbation la plus importante du procédé est la casse de feuille de papier. Cet événement est, dans la plupart de cas, imprévisible et il est difficile d'en gérer et d'en minimiser les impacts. Les casses interrompent la production de la machine à papier et produisent des quantités importantes de pâte cassée qui doivent être stockées et réutilisées dans le procédé. Le taux de recyclage de la pâte cassée dépend des inventaires dans la section de stockage des cassés. Cette variable est souvent réglée de façon à conserver un niveau désiré dans les réservoirs de pâtes et d'eaux blanches.

La pâte cassée a des propriétés considérablement différentes de ceux des pâtes vierges. Le recyclage des cassés a donc un effet important sur l'équilibre de la pâte mélangée qui alimente les machines à papier. Un changement brusque du taux de recyclage de la pâte cassée peut induire des variations dans les propriétés de la pâte mixte à la caisse d'arrivée d'une machine à papier et perturber la stabilité globale de la machine. Ceci risque de causer une casse de feuille, interrompre la production et perturber davantage le procédé.

Le principal objectif de cette étude est de bâtir une simulation dynamique de l'atelier de fabrication de papier journal et ensuite d'appliquer des méthodes d'optimisation par recherche directe afin d'améliorer la gestion des cassés dans le procédé. Plus spécifiquement, ces algorithmes servent à trouver un profil d'ajustement du taux de recyclage qui réduit l'impact de ce changement sur les propriétés de la pâte mixte aux caisses d'arrivée des machines à papier.

La méthodologie suivante est appliquée :

1. Une simulation dynamique est construite en utilisant le logiciel WinGEMS 5.0. Le modèle comprend les machines à papier, la section de cassés et le circuit d'eaux blanches.
2. Une fonction objectif est établie afin d'exprimer les variations dans les propriétés de la pâte mixte à la caisse d'arrivée de chaque machine à papier.
3. La simulation dynamique est utilisée pour examiner le processus actuel de recyclage de pâte cassée en évaluant la fonction objectif.
4. Une interface dynamique est construite pour relier la simulation aux algorithmes de recherche directe programmés dans Visual Basic

5. Les algorithmes de recherche directe sont utilisés pour trouver le profil quasi-optimum de changement du taux de recyclage. Les résultats sont évalués d'après la valeur de la fonction objectif.
6. La possibilité d'implanter en usine la stratégie améliorée développée dans cette étude et les bénéfices attendus après la mise en œuvre, sont examinés.

L'application considérée dans cette étude est une usine intégrée de papier journal. L'usine produit environ 1000 t/d de papier journal à partir de quatre machines. Les machines à papier sont alimentées par trois types de pâte : la pâte thermomécanique (PTM) la pâte désencrée (DIP) et la pâte cassée recyclée dans le procédé. Les pâtes vierges (PTM et DIP) sont fabriquées sur le même site dans les ateliers de pâte qui partagent le système commun d'eaux blanches avec la section de fabrication de papier. La pâte cassée est réutilisée dans le procédé avec un taux approprié permettant le maintien d'un inventaire souhaitable dans le circuit de cassés.

La pâte est stockée dans les inventaires de PTM, DIP et pâte cassée à une consistance élevée qui varie entre 3 % pour la pâte cassée, 4 % pour la DIP et 6 % pour la PTM. La pâte mixte, qui consiste en une proportion spécifiée de PTM, DIP et pâte cassée, est diluée et alimentée aux machines à papier à une consistance d'environ 1%. Les machines sont divisées en deux parties : la partie humide et la partie sèche. La partie humide se compose de la zone de formation, où la pâte prend la forme d'une feuille, et la partie de presses, où l'eau est retirée mécaniquement de la feuille. Avec une consistance d'environ

45%, la feuille sort de la partie humide et entre dans les séchoirs de la partie sèche. Au terme du procédé, la feuille de papier sort des séchoirs avec un taux d'humidité approximatif de 8 %. Les machines à papier consomment une quantité importante d'eau fraîche.

L'eau blanche retirée au niveau des machines est envoyée au circuit d'eaux blanches et utilisée pour des différentes applications dans le procédé, dont les plus important suivent ci-dessous:

- Dilution du courant de la pâte mixte alimenté aux machines à papier ;
- Dilution de la pâte cassée ;
- Dilution de la PTM et de la DIP dans les ateliers de pâte.

Lors d'une casse, la machine à papier continue généralement à fonctionner. La feuille cassée est ramassée au-dessous les machines, diluée jusqu'à une consistance d'environ 0,6 %, et envoyée au système commun de traitement des cassés. Là, la pâte cassée est épaissie à 3 %, nettoyée dans un système de tamis et de classeurs, et stockée dans le réservoir de pâte cassée. L'eau retirée lors de l'épaississement est utilisée dans les ateliers de TMP et de DIP.

Un modèle dynamique est construit pour simuler la fabrication du papier et les casses des machines à papier. Le modèle a été validé de manière informelle en utilisant les données prises à l'usine à l'aide du personnel. Les ateliers de pâte n'ont pas été inclus dans la simulation, et ainsi, les courants de PTM et de DIP constituent des inputs dans le modèle. Les autres paramètres d'entrée importants sont les suivants :

- Débit de la pâte mixte et eau fraîche alimentée à chaque machine à papier ;
- Ratio de PTM, de DIP et de pâte cassée dans le courant de pâte mixte pour chaque machine à papier ;
- Temps des casses aux parties humides et aux parties sèches des quatre machines.

Le principal paramètre de sortie de la simulation est les propriétés de la pâte mélangée à la caisse d'arrivée de chaque machine à papier. Les propriétés suivantes sont observées : débit total, consistance, température, fraction de fibres et de fines, et taux de solides dissous dans la pâte. Afin d'évaluer l'impact du recyclage de la pâte cassée, deux fonctions objectifs ont été établies. Les fonctions expriment les fluctuations dans les propriétés de la pâte à la caisse d'arrivée de chaque machine. Le but de cette étude est de minimiser ces variations en minimisant les fonctions objectifs.

La simulation est connectée à une interface spécialement programmée avec des algorithmes d'optimisation par recherche directe. Ces algorithmes ont pour but d'atteindre l'objectif de l'étude en cherchant des meilleures manières d'ajuster le taux de recyclage des cassés.

Les méthodes conventionnelles d'optimisation fonctionnent en minimisant une fonction objectif, habituellement sous contraintes. Les algorithmes standard d'optimisation exigent une fonction objectif analytique, ainsi que ses premières et/ou deuxièmes dérivées. Les modèles de procédé non analytiques (une simulation, par exemple) ne disposent que des valeurs de la fonction objectif. Pour de telles applications, les méthodes de recherche directe servent d'alternative. L'optimisation par recherche directe peut être considérée comme une méthode selon laquelle les variables d'entrée d'un modèle dynamique de processus sont sondées systématiquement de façon à rechercher les régions de fonctionnement quasi-optimales. Un indice de performance est mesuré dans ces régions et, après un nombre d'expériences suffisant, le régime de fonctionnement rendant la performance acceptable peut être trouvé.

Les deux méthodes de recherche directe appliquées dans cette étude sont la méthode de *Nelder-Mead Simplex* et les Algorithmes Génétiques. La méthode de *Nelder-Mead Simplex* est une formule algorithmique développée dans les années 1960. Grâce à sa simplicité et à sa fiabilité relative, elle est considérée la plus populaire et utilisée de toutes les méthodes de recherche directe. Les Algorithmes Génétiques sont basés sur les principes de la génétique et de la sélection naturelle, et sont devenus connus pour leur robustesse et leur résistance structurale à converger dans des minima locaux.

Une étude de cas a été réalisée pour évaluer et quantifier la possibilité de diminuer les variations dans les propriétés de la pâte aux caisses d'arrivée des machines à papier. Se basant sur les données prises à l'usine, une journée typique (24 heures d'opération) a été simulée avec le modèle dynamique. Le processus de recyclage des cassés a été reproduit comme il avait été fait à l'usine, et les fluctuations aux caisses d'arrivée ont été observées et évaluées par les fonctions objectifs. Nous supposons ici que ces variations pourraient sensiblement diminuer si les réglages du ratio des cassés avaient été faits plus graduellement. Les deux algorithmes de recherche directe ont été utilisés pour trouver le profil quasi-optimum de ces ajustements. Les fluctuations aux caisses d'arrivée pendant la simulation des cas optimalisés ont été mesurées en utilisant la fonction objectif et comparée aux résultats obtenus pour la simulation du cas initial.

Au terme de l'étude de cas, les résultats ont montré qu'en changeant le taux de recyclage des cassés plus graduellement, les variations des propriétés de la pâte mixte à la caisse d'arrivée peuvent être amorties considérablement. Les valeurs des fonctions objectifs ont été réduites sensiblement pour chaque machine à papier. Les deux algorithmes directs de recherche qui ont été utilisés ont fourni presque les mêmes résultats pour le profil quasi-optimum de changement du radio de pâte cassée. Toutefois, il a été observé que le temps de simulation a été considérablement plus petit pour la méthode de *Nelder-Mead Simplex*.

Les résultats de cette étude offrent une nouvelle approche dans la gestion des cassés à l'usine, ainsi qu'une perspective sur les bénéfices potentiels qui pourraient être dérivés de sa mise en œuvre. Il a été confirmé auprès du personnel de l'usine que le profil quasi-optimum de réglage du ratio de cassés trouvé dans cette étude peut être implanté dans le système de contrôle sans grand difficulté. Une étude de mise en œuvre serait nécessaire pour vérifier si les variations aux caisses d'arrivée peuvent être diminuées dans le vrai procédé et pour quantifier les bénéfices. Une diminution de la fréquence de casses est prévue comme conséquence d'une augmentation de la stabilité générale des machines à papier.

En conclusion, l'étude offre de nombreuses nouvelles voies de recherche dans le domaine. Il est recommandé également que le processus entier de gestion de casses soit analysé pour évaluer d'autres possibilités d'optimisation et d'amélioration du procédé.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

Modeling and optimization are process analysis and improvement tools that are becoming increasingly more important in the manufacturing industry. They are also an active research field in areas such as chemical or electrical engineering. It has been shown in many applications that a process' efficiency and productivity can be greatly enhanced with the help of a reliable modeling and optimization schemes (Hall and Bowden, 1997; Fraleigh, 1999, Xiong and Jutan, 2000).

Control strategies at pulp and paper mills are for the large part simple, and it might even be said that the sector lags other industries in the implementation of advanced process enhancement and optimization technologies. This thesis is concerned with the dynamic simulation of the papermaking section of an integrated newsprint mill, and the subsequent application of two direct search methods for the purpose of optimizing the current broke recycling strategies.

Following the problem statement and motivation behind the work, as well as the thesis scope and objectives, this chapter will present a short background of the newsprint process and optimization methods that will be explored, and lastly, outline the thesis conventions.

1.1 Problem Statement and Motivation

Paper breaks are a major disturbance in the papermaking process. They are largely unpredictable and difficult to handle properly in order to minimize impact on the stability of the papermaking process. The rejected paper (called "broke") is diluted and repulped, and recycled back into the process at a controlled rate. Broke pulp has considerably different properties than fresh pulp and as a consequence, the recycling strategy can greatly affect machine stability. Namely, sudden changes to the broke recycle rate can induce variations in the properties of the mixed pulp at the paper machine headbox, which has been shown to decrease the stability of the machine (Orccotoma, *et al.*, 1997).

The rate of broke recycling (the *broke ratio*) is a process variable controlled by the process operators with the objective of managing the inventories in the broke storage system. Depending on operational factors, the broke ratio is typically adjusted manually. A dynamic simulation has been constructed in order to examine the effect that these adjustments have on the pulp properties at the headbox of the paper machines at an integrated newsprint mill. It is hypothesised that if the broke ratio changes are made smoother and less sudden, the variations in the headbox conditions can be significantly reduced. Direct Search algorithms are used in order to find the improved change profiles for the broke ratio adjustments. The ultimate conclusion that will be sought is that the fluctuations at the headbox can be dampened, and thus the frequency of paper breaks will be reduced owing to improved overall machine stability.

1.2 Thesis Objectives and Scope

The main objective of this thesis is to perform a proof-of-concept study into the optimization of the broke recirculation strategy at an integrated newsprint mill. Optimization technologies have not been widely applied in the pulp and paper industry, and this work seeks to explore the optimization of broke recirculation strategies using two Direct Search methods. Implementation issues and recommendations for future work will be presented at the end of the thesis.

A detailed study of the newsprint process and the development of the steady-state and dynamic simulations of the papermaking section are presented in Chapter 2. The chapter also discusses the validation of the simulation using data collected at the mill. Chapter 3 begins with the theory of the Nelder-Mead Simplex method and Genetic Algorithms. The current mill operating strategy and problem statement are then presented in detail. Finally, the chapter discusses the development of the objective function and the solution approach. Chapter 4 is dedicated to the case study. The chapter introduces the setup of the study, the data used and the assumptions that were considered. Results are presented, followed by a discussion of the case study itself and its implementation issues. The final chapter summarizes the work done and outlines some recommendations for future work.

1.3 Background

Dynamic simulations are useful process analysis tools that can be used for the purpose of optimization. A thorough knowledge of the process is required alongside the construction of the simulation. This section will introduce the newsprint process treated in this study, and then present the methods of direct search optimization that will be used.

1.3.1 Newsprint Making Process

Newsprint production is one of Quebec's largest industry sectors. The process is continuous, and large modern paper machines can produce paper at rates of well over 500 tons per day. Paper is made out of pulp which is a suspension of wood fibres and water. Wood fibres are obtained from wood chips either by mechanical or chemical methods.

The mill considered in this work is an integrated newsprint mill, meaning that the pulping operations are linked directly to the papermaking operations through an integrated process water system. Three types of pulp are used in the process: thermomechanical pulp (TMP), deinked pulp (DIP) from recycled paper, and broke pulp. They are mixed at prescribed ratios, and the mixture comprises the feed into the paper machines. Paper machines are designed to accept the mixed pulp, form it into a sheet and dry it. While being formed, the sheet travels across the machines at speeds up to 1500 m/min. The process is fragile, and occasionally, the sheet ruptures (a "break"). The broke pulp is diluted, repulped, stored and eventually recycled back into the process.

1.3.2 Direct Search Methods

Standard optimization methods make use of an analytical (equation based) model of a process and an objective function that deterministically describes the performance of the system. For processes described by non-analytical models, where no derivative information is available for the response surface, conventional function minimization methods are not applicable (Wright, 1998; Hall and Bowden, 1997). Problems of this nature need to be solved using iterative optimization techniques, such as Direct Search methods.

Direct search simulation optimization can be thought of as a technique of methodically finding a combination of input parameters for a given process, which yields the best attainable response according to an objective function defined in terms of some chosen simulation output variables (Humphrey and Wilson, 1998). These methods employ systematic iterative experimentation to perturb the process (or the process' model) to different operating regions and evaluate the performance index in these regions (Monder, 2000). With enough experimentation, the algorithm arrives at a solution that is relatively close to the optimum. Because of their conceptual simplicity, direct search methods have been applied to processes in a variety of fields. The value of the response function is all that is required. Their main disadvantage is heavy computational requirement due to the iterative procedure.

There are many popular direct search methods: Evolution Strategies, Genetic Algorithms, Tabu Search, Nelder-Mead Simplex, Simulated Annealing. Each technique has its advantages and disadvantages, and there have been studies comparing the methods (such as Hall and Bowden, 1997). In this thesis, the Nelder-Mead Simplex and Genetic Algorithms will be used to optimize the broke recirculation strategy at an integrated newsprint mill. The Nelder-Mead Simplex, a simplex-based method developed in the 1960's, is by far the most popular and widely used direct search method owing to its simplicity and relative reliability (Wright, 1998). Genetic Algorithms are based on the principles of genetics and natural selection, and have become known for their robustness and structural resistance to getting trapped in local minima (Bingül, *et al.*, 2000).

1.4 Thesis Conventions

The term *process model* is used in this thesis in reference to the dynamic simulation. Inputs are those variables that enter the simulation from the simulation worksheet through the interface. Outputs are those variables that leave the simulation and are recorded in the simulation worksheet.

The term *pulp* refers to the solution of fibrous material dispersed in water and containing some dissolved solids. In accordance to the standard in the pulp and paper industry, the word *consistency* refers to the pulp's weight concentration in fibrous material. The fibrous material is assumed to consist of fibres and fines, where fines are typically those fibres that are less than 0.5 mm in length.

Throughout the thesis, all terms and abbreviations are explained on first usage.

CHAPTER II

MODEL

Computer simulation is a powerful analytical modeling tool that is increasingly used in research and industry for process design, analysis and optimization (Croteau and Roche, 1987; Hall and Bowden, 1997). The main purpose of their application is to aid design, increase production efficiency or decrease environmental impact. A dynamic simulation is an example of a process model that, when validated, can provide extensive insight into a process operation and can serve as a reliable tool for process analysis or improvement.

This chapter discusses the modeling of the newsprint process considered in this study. After a description of the process in Section 2.1, Section 2.2 focuses on the creation of a dynamic simulation of a newsprint mill for the purpose of direct search optimization. The overall strategy, input and output variables and the steps taken to build the steady state and dynamic simulations are presented. Lastly, Section 2.3 describes the validation of the model.

2.1 Introduction and Process Description

The application considered in this study is the Belgo newsprint mill belonging to Abitibi-Consolidated Inc., which is located in Shawinigan, Québec. The mill produces newsprint at the rate of about 1000 t/d. Four paper machines of differing age and characteristics are fed by three thermo-mechanical pulp (TMP) lines and a de-inked recycled pulp (DIP) line. The four machines share a common whitewater system and a common broke system that recycles the broke into the pulp stream. The mill's general flowsheet, outlining typical values of some important flow rates and average production rates, is presented in Figure 2.1.

The mill is highly integrated, implying that the whitewater system is shared by the pulping sections (TMP and DIP) and the papermaking plant. Generally, the whitewater withdrawn from the paper and broke in the papermaking section is used by the pulp plants. Broke pulp is stored and recycled back into the system, impacting the operation of the paper machines.

The following subsections describe the process in more detail by focusing on the different circuits of the mill.

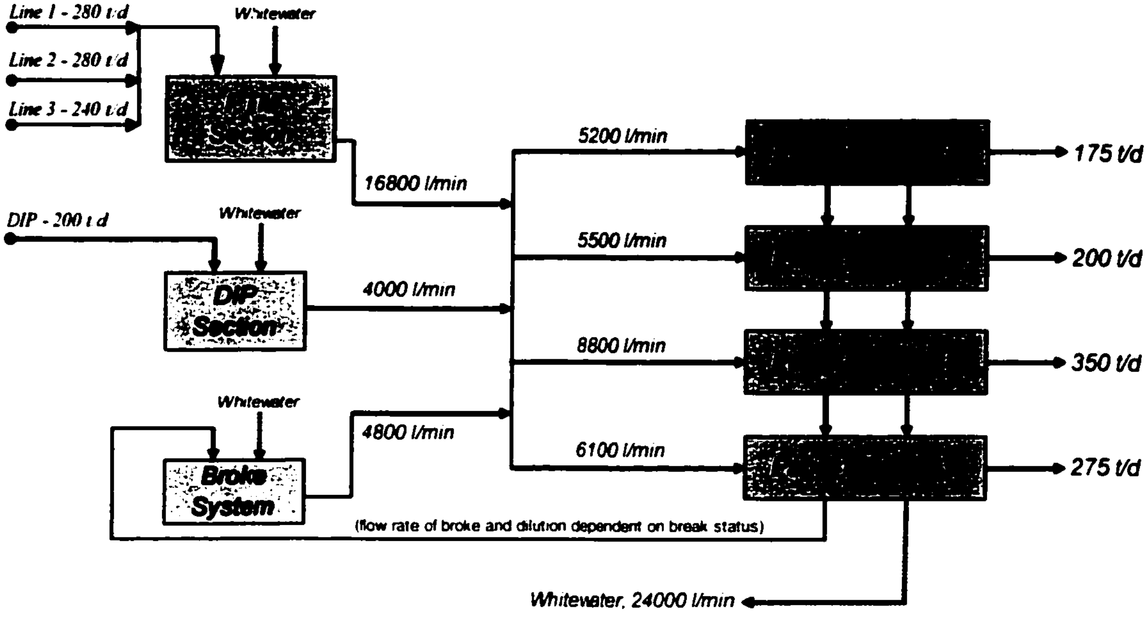


Figure 2.1: General flowsheet of the Abitibi-Consolidated Inc. – Belgo newsprint mill in Shawinigan.

2.1.1 Pulping and Pulp Preparation Sections

The pulp is refined and prepared on site in the TMP and DIP mills. It is then stored at elevated consistency and diluted on the way to the paper machines. The pulping sections use large amounts of whitewater coming from the papermaking part of the plant.

The TMP plant is comprised of three lines working independently. Chips, at 47% consistency are fed to the double-stage refiners. The refiners consume large amounts of electrical energy and whitewater to convert the wood chips into a free-flowing pulp solution. The refined pulp is stored in latency chests and then passed through screens. The rejects from the screens are fed to the rejects refiner. The accepts stream passes through cleaners and makes up the dilute thermomechanical pulp that is sent to the disk thickeners, which raise the pulp's consistency to about 6.5%. The whitewater withdrawn from the thickeners is used primarily for upstream pulp dilution in the TMP plant, as well as in the DIP plant. The thickened pulp is stored at this elevated consistency in the '200T' TMP storage tank. From there it is fed to the machines following dilution using whitewater from the papermaking section.

The DIP plant makes pulp out of recycled paper. The paper (at about 8% moisture) is first diluted to about 1% consistency in pulpers, using cloudy whitewater originating from the DIP thickeners and lean water from the papermaking section. The diluted pulp passes through cleaners and screens, after which it is then thickened to about 4% consistency by the DIP disk thickener. The thickened pulp is stored in the '50T DIP' storage tank, from where it is sent to the mixed pulp chests of the paper machines.

2.1.2 Paper Machines

The mill consists of four paper machines (Machines 6, 7, 8 and 9) of varying age, capacity and characteristics. Regular newsprint as well as special order paper can be made depending on the properties of the mixed pulp and other factors. The machines share a common broke and whitewater collection system.

Figure 2.2 shows a schematic of a paper machine. The pulp mixture (TMP, DIP, broke), diluted to about 3.2%, is stored in the mixed pulp tank located ahead of each machine. The pulp mixture typically consists of about 60% TMP, 20% DIP and 20% broke, but it varies according to the mode of operation (full operation or break operation), inventories, type of paper produced and paper machine. At the fan pump, the pulp stream is mixed with rich whitewater from the wire pit. The mixed stream (now at about 1% consistency) goes though a series of screens before it arrives at the machine headbox.

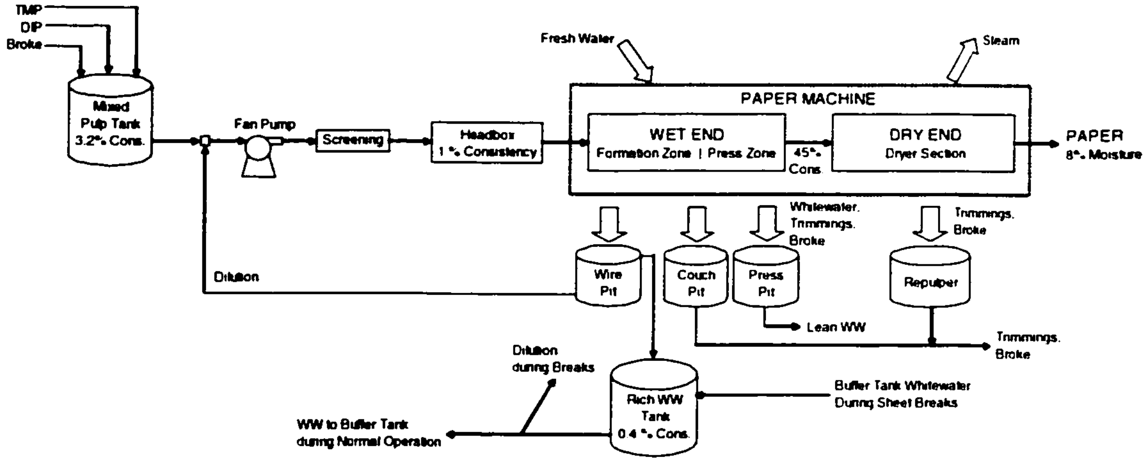


Figure 2.2: Schematic of a newsprint machine.

From the headbox, the pulp enters the formation zone of the paper machine as a wet web. Whitewater from the formation zone is collected below the paper machine in the wire pit and the couch pit. Leaving the formation zone at about 20% consistency, the paper web enters the press section, where it is compressed and more water is mechanically removed.

Whitewater from the press section is collected in the press pit, from where it is sent to the whitewater system (discussed in Section 2.1.4). The forming zone and the press section constitute the wet-end of the paper machine. At about 45% consistency, the sheet leaves the wet-end and enters the paper machine dryers (the dry-end). Leaving the dryers are steam and the final paper product at 7-8 % humidity, depending on the machine.

Fresh water is used in the showers located throughout the various sections of the machine. In addition, lean white water is used in the showers on Machines 8 and 9.

During normal operation, the whitewater collected in the wire and couch pits is used to dilute the incoming pulp. The surplus whitewater overflows to the rich whitewater tank and from there it is sent to the buffer whitewater tank, which is common to all paper machines. The water from the press section is also sent to the whitewater system, which will be further discussed in section 2.1.4. Lastly, some of the whitewater collected at the machines is used as vacuum pump seal water.

During a paper break, the unfinished paper sheet falls to the pits if the break occurs in the wet-end, and in the repulper in case of a dry-end break. During these conditions, no whitewater is sent to the buffer tank, and it is used instead do dilute the broke. A stream of additional whitewater from the buffer tank is activated to help the dilution. The next section describes further the broke system.

2.1.3 Broke Circuit

A break on the paper machine occurs when the sheet ruptures while being formed. Paper breaks happen for many reasons and are largely unpredictable. The machine is typically maintained in operation for the duration of the break while action is taken to restore the paper production. A break can last from a few minutes to a few hours, depending on the circumstances of the problem. Meanwhile, the unfinished product (the broke) is collected, repulped, diluted and recycled back into the process.

Figure 2.3 shows a schematic of the common broke system at the Belgo newsprint mill. The broke coming from the machines is sent to the broke handling tank (the '30T' tank). In the model, wet-end broke is collected and diluted to about 0.6% consistency in the couch pit. Dry-end breaks are diluted to a similar consistency in the repulper before being forwarded to the 30T tank. During normal machine operation, a small continuous flow of diluted wet paper trimmings is collected in the couch pit and sent to the broke handling system. This amount represents approximately 3% of the total paper production in a modern newsprint mill (Orccotoma, *et al.*, 1997). In addition, rejected paper rolls are repulped and diluted and sent into the 30T tank at about 2.3% consistency.

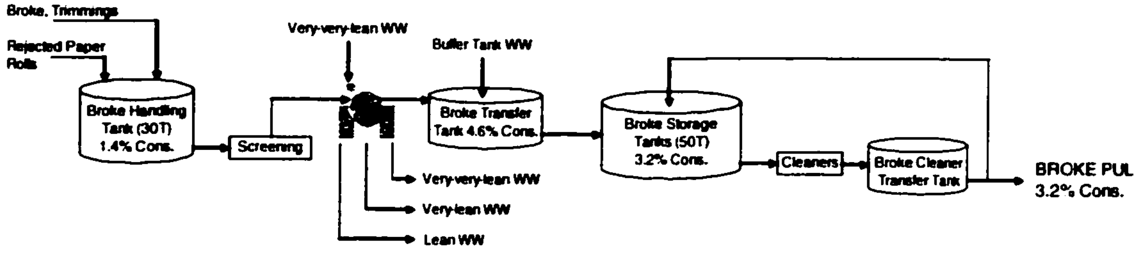


Figure 2.3: Schematic of the broke system at the Belgo newsprint mill.

The level of the broke collection tank is usually maintained low (around 30%) in anticipation of a break. The outlet from the 30T tank, at about 1.4% consistency, passes through a screening system and continues to the broke thickeners. The purpose of the two thickeners is to thicken the broke before storage. The pulp exits the thickeners at 4.6% and enters the broke transfer tank where it is diluted by buffer whitewater to the final storage consistency of 3.2%. At this consistency, the broke is forwarded to the two '50T' broke storage tanks.

Exiting the 50T tanks, the pulp passes through a three-stage cleaner system. The accepts from the primary cleaner are collected in the broke-cleaner transfer tank and constitute the broke that is used in the mixed pulp going to the machines. The accepts from the secondary cleaner are returned to the 30T tank. The accepts from the tertiary cleaner return to the secondary cleaner, while the rejects flow back to the TMP refiners.

2.1.4 Whitewater Circuit

The whitewater system is a complex circuit with fast-changing flow rates and many interactions taking place on a continuous basis. The system is somewhat divided into two parts: the papermaking section whitewater and the pulp plant whitewater. Generally, whitewater flows from the papermaking areas to the pulp plants. A small amount of cloudy whitewater from the TMP thickeners is sent to the papermaking lean whitewater system (see Section 2.2.1).

Three types of whitewater are collected from the broke thickeners: lean, very-lean and very-very- lean, depending on where the whitewater is withdrawn from the thickeners. An additional stream of lean whitewater comes from the press sections of the paper machines. The majority of lean whitewater is used in the TMP plant (~10000 l/min). It is mixed with the cloudy water from the TMP disk thickeners and sent to the TMP whitewater reservoir, discussed below. The remaining exit stream from the lean water tank passes through a series of screens (*Sinclair* and *Albany Gravity*) and subsequently returns to the tank. A small portion of the Sinclair filter accepts is used in the showers on Paper Machine 9. The rest of the accepts and the rejects are sent to the TMP plant.

The lean water tank overflows to the very-lean whitewater tank. Around 2000 l/min of very-lean water is used in the DIP plant, while the overflow is sent to the secondary treatment plant. Very-very-lean water is reused as shower water for the broke thickeners.

The buffer whitewater tank is the major whitewater reservoir in the papermaking section. During normal operation, whitewater from the wet-end of the paper machines is collected in this tank at around 24000 l/min. The major outlets from the buffer tank are as follows:

- Dilution of broke during a paper machine break (up to 20000 l/min per machine);
- Broke pulp dilution at the broke transfer tank, the 50T tanks and at the entry of the tertiary cleaners (~2000 l/min);
- TMP plant dilution (~ 12000 l/min);
- TMP dilution at the exit of the 200T storage tank (~ 6000 l/min);
- DIP dilution (~ 2000 l/min)

A certain level is maintained in the tank at all times in anticipation of a sheet break. Usually the tank's volume varies between 30% and 70%. To prevent tank overflow in case of a prolonged no-break situation, a control system is put in place whereby an increased amount of whitewater is withdrawn from the buffer tank and used in the TMP plant. In exchange, less paper machine lean whitewater is sent to the TMP plant and more of it overflows to the very-lean tank and ultimately to secondary treatment. The relationship between the level of the buffer tank (L_{BT}, %) and the flow rate of very-lean water sent to secondary treatment (Q_{ST}, l/min) is described by the following equation, taken from mill data:

$$
\begin{aligned}
Q_{ST} &= 200L_{BT} - 5800 && \text{for } L_{BT} \geq 30\% \\
Q_{ST} &= 0 && \text{for } L_{BT} < 30\% \qquad (2.1)
\end{aligned}
$$

The TMP whitewater reservoir is the heart of the pulp plant whitewater system. Cloudy water from the disk thickeners along with lean water from the broke thickeners enters the tank. Water from the tank is used in the following units and streams:

- Dilution of pulp coming out of the three main refiners and the reject refiner;
- Latency chests on all the TMP lines;
- Dilution of streams before all the cleaners.

2.2 Simulation Development

A dynamic simulation of the papermaking section of the newsprint mill was developed in order to test the possibilities to optimize the process operation during and following paper breaks. The simulation includes the paper machines, the broke system and the whitewater circuit. The general purpose of the model is to simulate the paper breaks, evaluate the impact of broke recycling on the stability of the machine headbox feed, and to test improved broke recirculation operating strategies, found using direct search optimization techniques.

In general, process simulators can be divided into two groups: sequential and simultaneous (equation based) simulators. The basic difference is that while simultaneous simulators use a matrix of mass and energy balance equations and their derivatives to find an analytical solution to the entire system simultaneously, sequential simulators solve these balances in series for each operating unit included in the process model. The calculations are done iteratively according to a predefined order until convergence criteria are met.

The dynamic model for this work was developed using WinGEMS 5.0 software from Pacific Simulation of Moscow, Idaho. It is a sequential simulator specifically tailored for pulp and paper processes. This section will first discuss the overall strategy used for the development of the model and then focus on the construction of the steady-state and dynamic simulations and the selection of inputs and outputs.

2.2.1 Overall Strategy

The papermaking section of the mill was simulated in dynamic mode using WinGEMS 5.0 software. The simulation contains the following major sections: the four paper machines, the broke system, and the whitewater system. The TMP and DIP mills were omitted in the simulation for three reasons: a) lack of substantial relevance of the operation within the pulp-making areas to the objectives of this work; b) major simulation and convergence problems and lack of robustness resulting from the presence of numerous looping water streams within these sections; c) relative ease of separating the pulp mills from the papermaking section despite the integration between the two sections. The latter assumption will be discussed in more detail below.

The simulation was constructed with a substantial amount of detail based on the mill process diagrams and discussions with mill personnel. All of the major units and streams were included. The flow sheet of the simulation is included in Appendix I. The general approach was to simulate the approximate values of flow rates, consistencies, temperature, fibre and fine ratios and dissolved solids concentration. For the majority of cases, these data were taken out of the Process Information ("PI") system at the mill. In several instances, input from mill personnel was also used to make the model as realistic as possible. While no formal validation was done on the simulation, an informal validation study was performed and is included in Section 2.3.

The four paper machines are slightly different from one another, but the general setup is similar. Appendix I contains simulation compound block diagrams of the four paper machine sections. Leaving the mixed pulp tank at about 3.4% consistency, the pulp is diluted to about 1% by whitewater from the wire pit tank. The diluted pulp is passed through a dual screening system. The accepts continue into the headbox, from which the pulp enters the paper machine forming section while the rejects are sent to the broke system. Fresh water is supplied to each machine at average flow rates provided by the mill personnel. In addition, some fresh water is supplied to the machines headbox showers. Whitewater streams from the forming and press sections of the wet-end are collected in the wire pit and press whitewater tanks, respectively. Water from the wire pit is used primarily for diluting the original mixed pulp, while the overflow stream enters the rich whitewater tank. A continuous small flow of the rich whitewater is used for diluting paper trimmings and pulp at the dual screening system. Most of the water leaving the rich whitewater tank is sent to the whitewater buffer tank when the machine is operating. During a break, the rich whitewater tank is supplemented with additional whitewater from the buffer tank and used to dilute the broke. For the purposes of the simulation, the couch pit is used for collecting and diluting the trimmings from the machines (a small flow at about 10% consistency) during normal operation and broke pulp while simulating wet-end paper breaks. Breaks in the dry-end are simulated by diverting the pulp to the repulpers where it is diluted with rich whitewater. Water from the press sections of the machines, collected in press whitewater tanks, is sent to the whitewater circuit. From the dryers, water exits in the form of steam from the paper machine hoods and constitutes an output from the simulation. In addition, Machines 6, 7 and 8 collect seal water for vacuum pumps. The excess of that water is turned towards the rich whitewater tank on Machines 6 and 7 and to the press water tank for Machine 8.

The diluted trimmings and broke pulp are sent to the common broke circuit. The simulation contains all the major units of the broke system: the broke collection tank (30T), the screening system, the thickeners, the broke storage tanks (50T) and the three-stage cleaner system. Repulped and diluted off-spec paper is continuously added to the 30T tank. Broke pulp is withdrawn out of the 50T storage tanks at the flow rate required for the four machines.

The broke system is directly linked with the whitewater circuit. Broke pulp is diluted in several places by whitewater coming from the buffer tank. Meanwhile, water withdrawn from the broke at the thickeners supplies the whitewater system through the lean, very-lean and very-very-lean water tanks. In addition to the tanks, the whitewater system also contains filters that treat the press water coming from the machines, as well as the cloudy whitewater originating from the TMP section disk filters.

As the water circuit is shared throughout the mill, the whitewater section of the simulation acts as the interface between the pulp shops and the papermaking part of the plant. In general, as in most paper mills, the cleaner whitewater from the papermaking half of the plant flows back towards the pulp shops. The following are the major whitewater streams going towards the pulp preparation sections:

- Buffer whitewater flowing towards the TMP refiners;
- Buffer whitewater flowing towards the TMP disk thickeners and storage tank;
- Buffer whitewater flowing towards the DIP plant;
- Lean whitewater flowing towards the TMP plant;
- Very lean whitewater flowing towards the DIP plant;
- Accepts and rejects from lean water filters (*Sinclair* filters) flowing towards the TMP disk thickeners and storage tank

Only a single whitewater stream listed below flows in the opposite direction, from the TMP disk thickeners towards the lean whitewater system in the papermaking section. Its parameters were fixed at constant values based on analysis of average mill data and advice given by the mill personnel and are as follows: 9000 l/min, 0.05% consistency, 56°C, 10% fibres (90% fines), 0.325 mass % dissolved solids. Having only a single whitewater stream entering from the pulp shops into the papermaking section made it simple to omit the TMP and DIP systems in the simulation without significantly affecting the precision of the model.

The procedure taken to build and run the steady state simulation and to transform it to a dynamic model is presented below.

2.2.2 Steady-State Simulation

A steady-state simulation was constructed first and served as a tool to test the mass and energy balances of the mill at near-steady-state conditions, and as the prerequisite step to incorporating dynamics into the model. The WinGEMS 5.0 software contains typical blocks found in process simulators: mix and split blocks, tanks and screens. Fibre fractionation by length at the machines and thickeners was simulated using component separation options available with the split block. Cleaners and filters were modeled using screens.

The steady-state simulation was built to represent the mill's operation at full production (no paper breaks) at average values of flow rate, consistency, temperature, fibre classification and total dissolved solids (TDS) content for the major process streams. A general mass balance was performed at first to verify the existing data collected at the mill and to supplement the missing data. The results for the most important streams are tabulated in Appendix II.

In steady-state mode, all of the tank inventories are assumed to be constant with time. This implies that the flow rate into a tank is equal to the flow rate out of it. A mass balance problem was encountered in the broke system since the mixed pulp streams contained a fixed ratio of broke pulp but no actual breaks were simulated in the steady-state simulation. As a result, no pulp was coming into the broke circuit, with the exception of diluted paper trimmings from the machines and repulped rejected paper rolls. The problem was addressed by adding a virtual broke input stream into the 30T broke handling tank at the average flow rate that satisfied the mass balance. This stream was naturally removed when the simulation was made dynamic, as described in the following section.

2.2.3 Dynamic Simulation

To properly model paper machine breaks, to observe the impact that they have on the process and to simulate the actions taken during and after their occurrence, the process simulation was run in dynamic mode. Three major modifications to the steady-state simulation were made to obtain the dynamic model:

1. Tank and reservoir levels of major importance to the objectives of the work were made dynamic. Their volumes were defined and the outlet flow rates became user-controlled.
2. Controllers were added to the simulation to control the levels of key tanks and the consistency of pulp at various points around the paper machines. PI controller blocks were used in several places, but in most instances, manipulated control variables (such as the required dilution water flow rate) were calculated using balance equations at each time interval.
3. An input-output panel was included in the simulation. Simulation inputs can be changed in real-time, while the observed outputs can be registered dynamically on a spreadsheet. This input-output interface was programmed in WinGEMS script code and is described in more detail in Section 3.5. The actual inputs and outputs are discussed in the next section.

The following main assumptions were considered while constructing the dynamic model:

- All tanks are perfectly mixed and thus, their contents are instantaneously homogenous;
- Flow delays caused by pipe length, valve openings and pump performance are neglected;
- Fluctuations in input values are neglected;
- Performance of separation units, such as split and screen blocks, remains unchanged regardless of the flow rate into these units;

Below is a list of the more important control schemes incorporated into the dynamic model:

1. Level control on the 30T broke handling tank – a proportional controller is used to maintain the level at approximately 33%;
2. Level control on the 50T broke storage tanks – the total flow rate out of the tanks is calculated based on the sum of the flow rates of the broke pulp streams required for the four machines. In case of inventory shortage, the flow rate of repulped rejected paper rolls is increased. When the level is too high, some of the broke pulp is diverted to the DIP section. More details on this strategy are presented in Section 3.4.2.
3. Level control on the broke-cleaner transfer tank – the level is maintained at 80% using PI control strategy in anticipation of an increased demand for broke pulp;
4. Level control in the buffer whitewater tank and control of whitewater dilution system for the machines – described above in Section 2.1.2. Under normal operation, whitewater from the paper machines fills the buffer whitewater tank. During a break, the appropriate amount of whitewater (depending on the machine) is withdrawn from the tank to dilute the broke. In addition, a significant portion of buffer tank whitewater is used in the broke section and the pulp mills. Equation 2.1 in Section 2.1.4 describes the system put in place and implemented in the simulation to prevent the tank from overflowing or emptying;

5. Dilution control schemes on the paper machines – programmed into the split blocks at the exit of the rich whitewater tanks on each paper machine. During normal operation, some of the rich whitewater is used for diluting the rejects from the machine screens and the paper trimmings, while most of it is sent to the buffer whitewater tank. During paper breaks, the stream to the buffer tank is automatically shut down and most of the water is used to dilute the broke pulp.

2.2.4 Selection of Model Inputs and Outputs

The goal of every process model is to provide the user with output variable values that correspond to the simulated operating regime with the defined input variables. The model should be versatile and robust enough to handle the relevant range of inputs and operation without sacrificing the quality of the results. In dynamic mode, inputs, outputs and operating regimes are subject to change with time. With properly chosen input and output variables, the model becomes a useful tool for various types of analysis, such as optimization studies.

The main purpose of this dynamic model was to evaluate the impact of paper breaks and broke ratio changes on the pulp properties at the headboxes of the machines. The main outputs from the simulation are, thus, the flow rate, consistency, temperature, fibre profile and TDS content in the headbox streams of the four machines. These values are registered on a minute-by-minute basis in the simulation spreadsheet through the interface. With this information, the objective function for a given simulated case can be evaluated, as it is described in Section 3.4. Other registered outputs include the flow rates of secondary treatment streams and whitewater streams leaving the papermaking section.

The model's inputs define the simulation case and manage it in real time through the interface that links the simulation to the input spreadsheet. The following input variables can be manipulated dynamically:

- Wet-end and dry-end break times on the four paper machines;
- Ratio of TMP, DIP and broke pulps in the mixed pulp streams;
- Flow rate of the mixed pulp stream to the paper machines;
- Flow rate of fresh water added to the system via paper machine showers as described below;
- Flow rate and consistency of repulped rejected paper rolls entering the broke system;
- Flow rate and consistency of cloudy whitewater entering the whitewater system from the TMP section.

Only the first two variables were user-varied in time in this study. The flow rate of the TMP cloudy water is kept constant at values mentioned in Section 2.2.2 by suggestion of mill personnel. The total flow rate of fresh water entering the system is kept constant at 10000 l/minute based on available average mill data. The lack of precise dynamic data made it unreliable to vary these values in time. The following are the entry points and flow rates of the fresh water entering the mill:

- Machine 6 showers: 2270 l/min entering at 20°C, heated up to 60°C;
- Machine 7 and 8 showers: 5400 l/min entering at 20°C, heated up to 60°C and divided into 2350 l/min for Machine 7 and 3050 l/min for Machine 8;
- Machine 9 showers: 2380 l/min entering at 20°C and heated up to 60°C

Apart from the above entry points, fresh water lines also enter the buffer whitewater reservoir and the very-very lean water tank, but their flow rates are kept at zero in this work.

2.3 Simulation Validation

A dynamic simulation can serve as a powerful process analysis tool provided it is validated to an appropriate degree of accuracy. The simulation created in this work was validated in two ways. Informal validation was initially done with mill personnel during the construction of the simulation. After the simulation was completed, some of the major outputs were compared with dynamic data obtained from the mill's data collection system. Figures 2.4 through 2.10 show the comparison of the following variables: headbox consistencies for all four machines, level and consistency in the broke storage tank, and level, consistency and temperature in the whitewater buffer tank.

At the time that the simulation was constructed, the paper machines were designed to operate with a headbox consistency of roughly 0.95%, as this was the overall average value in the process. However, depending on the grade of paper being produced and other operational factors, this number can vary between 0.8 and 1.1%. During the day for which simulation data was used, the headbox consistency was about 1.08% on machines 6 and 7 and around 0.8% on the other two machines. It has been assumed that the spike observed in the mill data on Machine 7 around 19:00 o'clock is an abnormality, most likely an instrumentation error. The simulated consistency values are generally shifted, but as can be seen from all four graphs in Figure 2.4, the overall trend between real data and the dynamic simulation follows quite well.

Figures 2.5 and 2.7 show a similar pattern for changes in level values in the broke storage tank and the whitewater buffer tank, respectively. The general trend in both cases matches very well, but the simulated value is consistently lower than the mill value. An explanation for this observation can be suggested by examining the consistency figures, 2.6 and 2.8 respectively for the broke storage and the whitewater tanks. In both cases, the simulated consistency is slightly higher than the one recorded at the mill. Thus, the overall mass balance is reasonably satisfied, as can be seen in Figure 2.9 which represents the product of the level and consistency. Finally, Figure 2.10 shows a comparison between the real and simulated values of temperature inside the whitewater buffer tank. That last variable is shown to vary between 60 and 62 °C in both the simulated and the mill data.

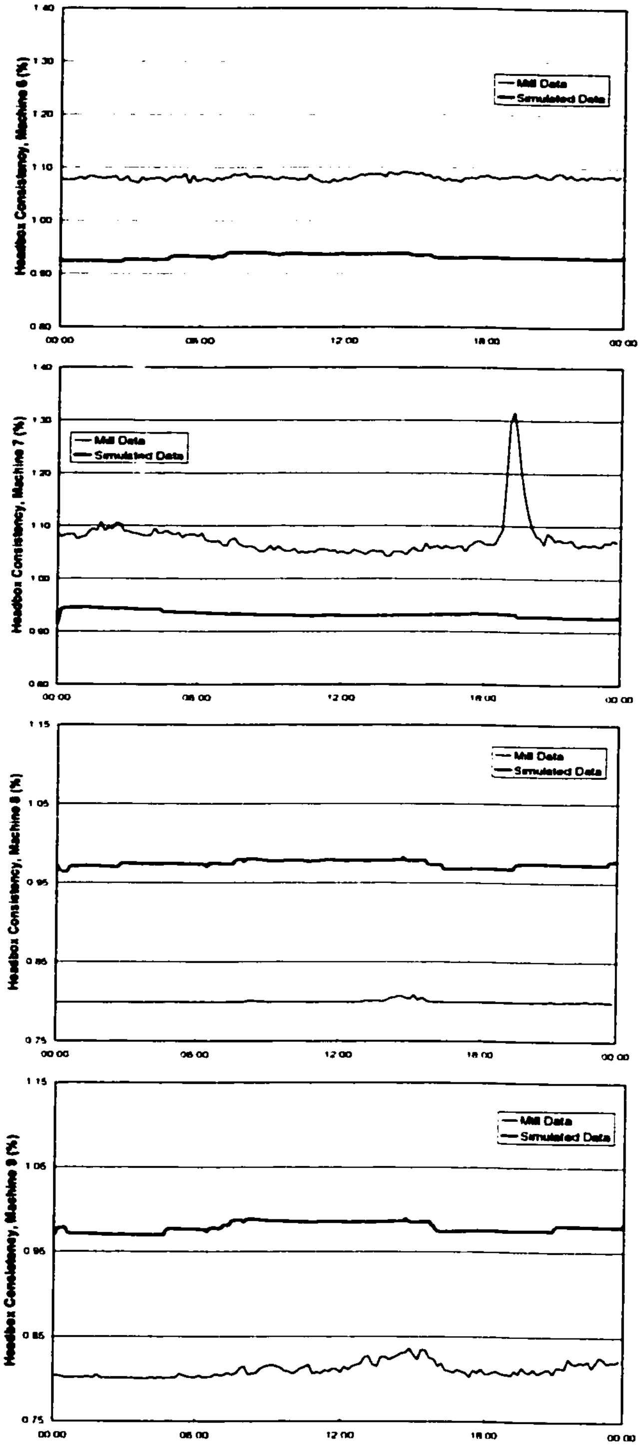


Figure 2.4: Comparison of real and simulated values of the headbox consistencies in each of the four paper machines.

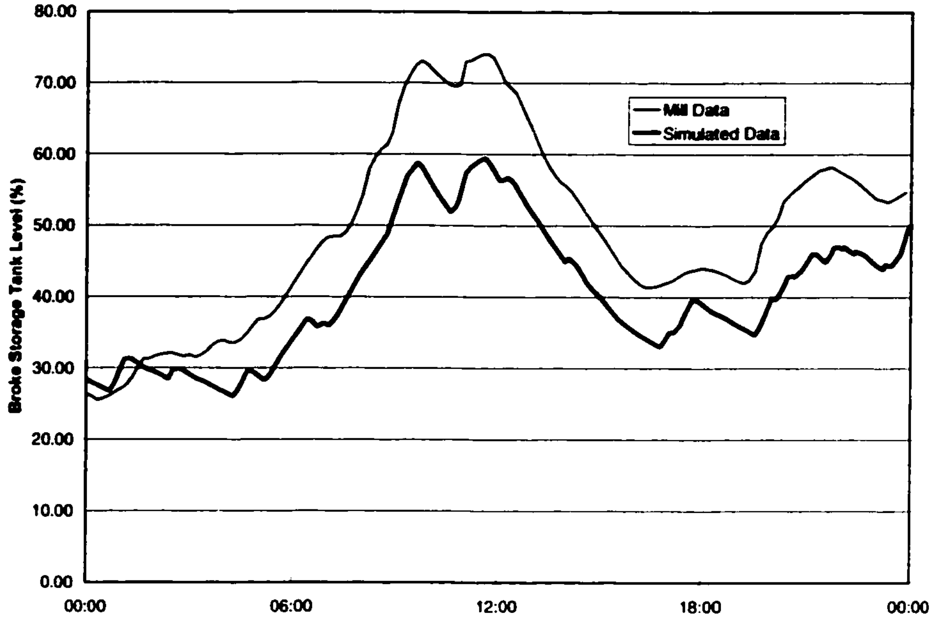


Figure 2.5: Comparison of real and simulated values of the broke storage tank level.

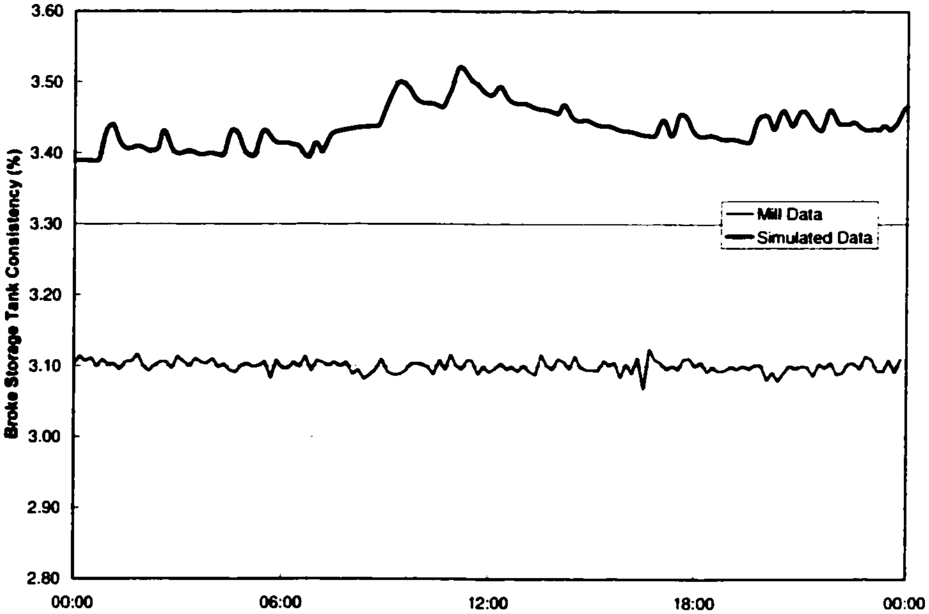


Figure 2.6: Comparison of real and simulated values of the broke storage tank consistency.

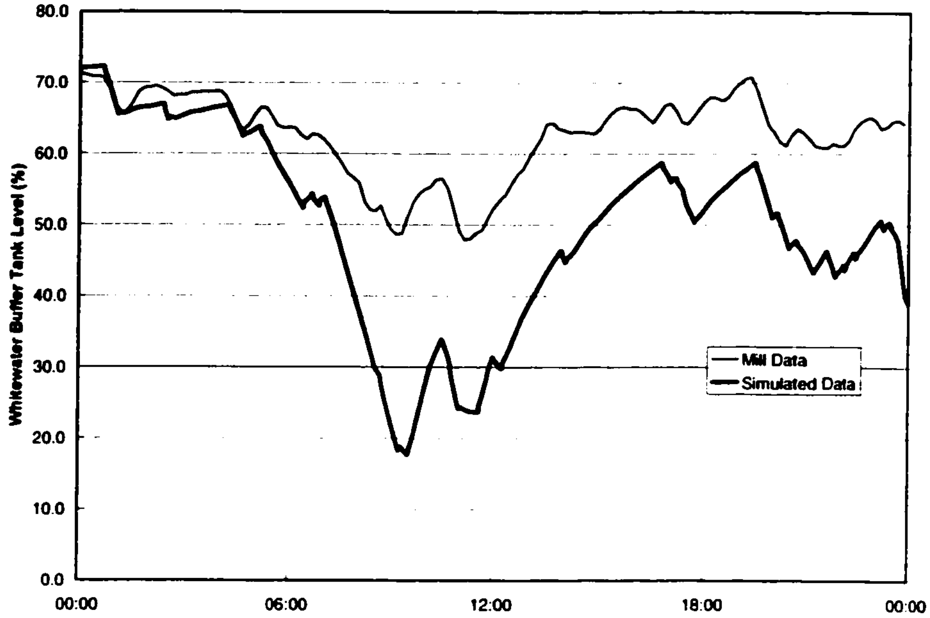


Figure 2.7: Comparison of real and simulated values of the whitewater buffer tank level.

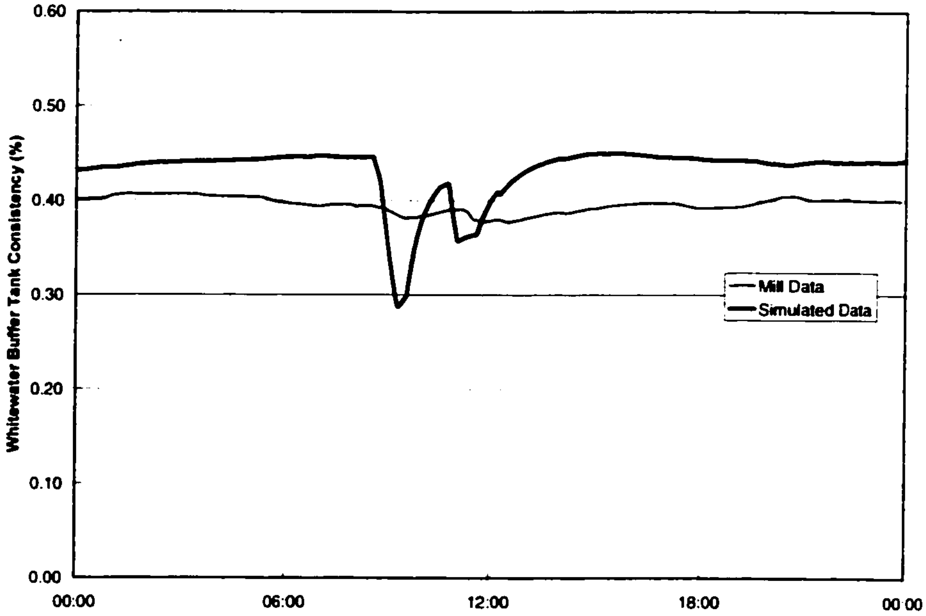
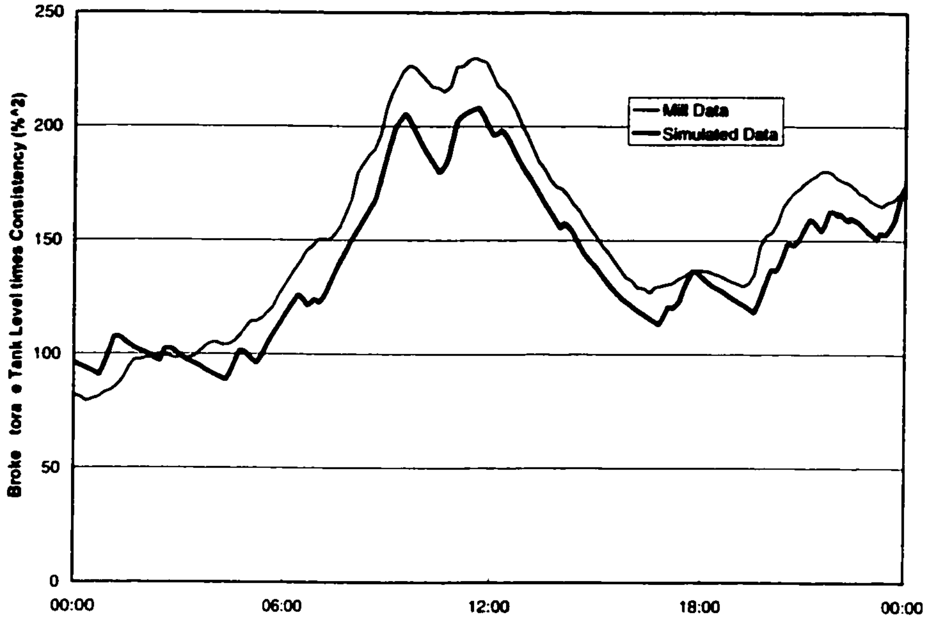


Figure 2.8: Comparison of real and simulated values of the whitewater buffer tank consistency.



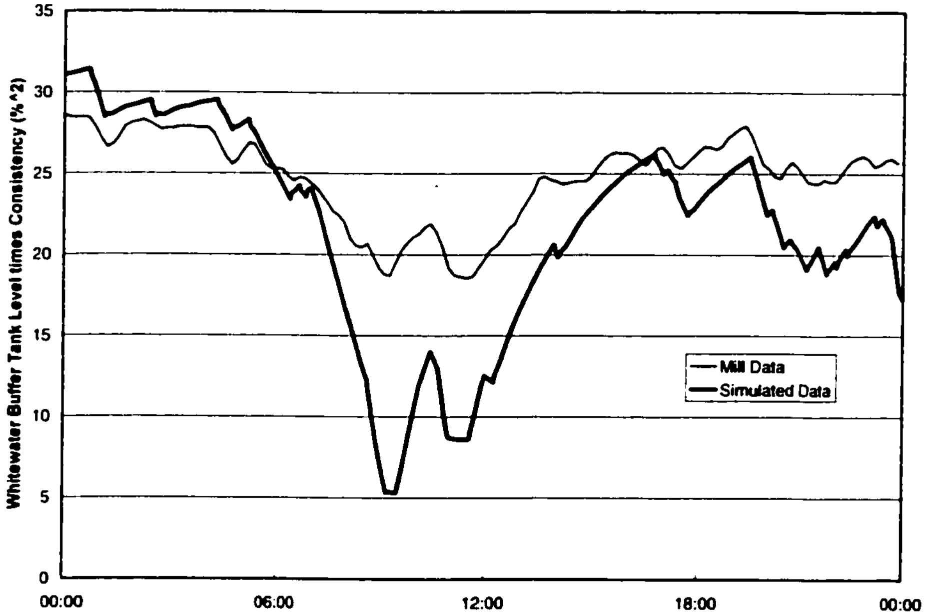


Figure 2.9: Comparison of real and simulated products of level and consistency values for the broke storage and the whitewater buffer tanks.

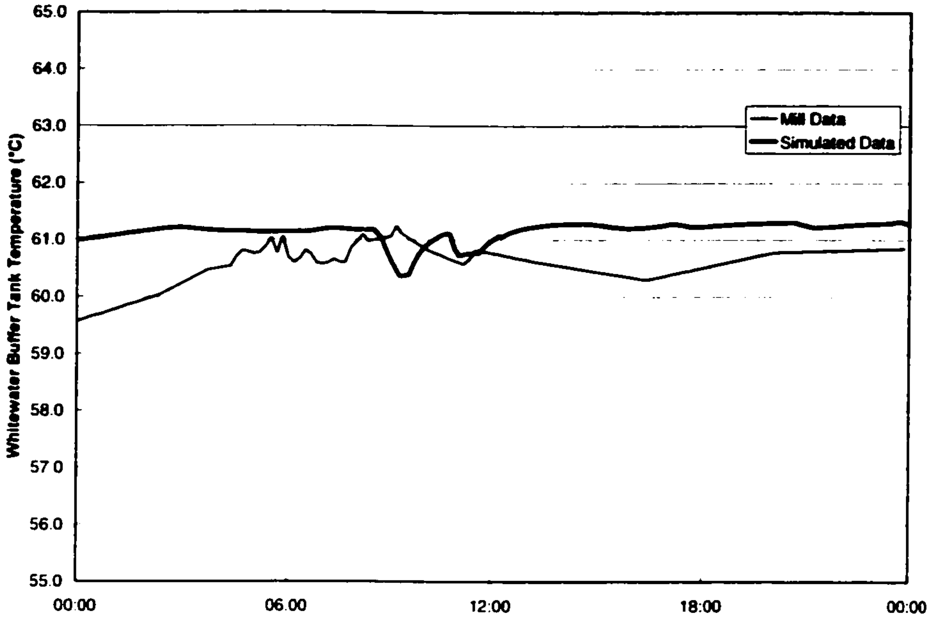


Figure 2.10: Comparison of real and simulated values of the whitewater buffer tank temperature.

CHAPTER III

OPTIMIZATION OF THE BROKE RECIRCULATION STRATEGY

Process optimization is a widely studied area both in research and in industry. Advanced process automation and optimization technologies are essential to maintaining and improving the competitive advantage of manufacturers, including paper makers. Most optimization techniques use an analytical model of a process and classical function minimization methods. For cases where equation-based models are not available, but simulations, based on commercial process simulation software, are available, direct search optimization methods are the alternative.

Following a short introduction, this chapter describes two different direct search methods in Section 3.1. The problem statement and current operating strategies are presented in Sections 3.2 and 3.3, respectively. The following two sections explain the objective function development and the solution approach for the optimization of the broke recirculation strategy at an integrated newsprint mill.

3.1 Introduction and Theory

Simulation optimization can be thought of as a method of finding a combination of input parameters for a given process in a given operating regime and under given constraints, which yields the best attainable response as expressed by an objective function defined in terms of some chosen simulation output variables (Humphrey and Wilson, 1998). The goal of an algorithm is to choose the values of the input variables (among many, possibly infinite, possibilities) that produce a model response that is deemed to be close enough to the optimal performance. The performance is often measured in terms of several objectives, and achieving these objectives simultaneously based on more than one criterion is not always an obvious task.

Standard optimization problems require an analytical objective function, with constraints, that deterministically describes the performance of the system. For example, Monder (2000), who seeks to optimize, in real time, a gasoline blending process with uncertain parameters, uses an objective function that describes the economic benefit of the produced goods under product quality and feedstock availability constraints. Using an analytical model, the performance of the process can be gauged instantaneously as a function of the model inputs. More importantly, the derivatives of the response function are available. For process models that are not equation-based, such as simulations, only the solution values are available and the derivatives of the response surface are not. Without them, or at least their finite-difference approximation, standard function minimization methods are not applicable (Wright, 1998; Hall and Bowden, 1997). Problems of this nature cannot be solved analytically and require an iterative optimization technique. As a solution to such cases, direct search methods surface as the plausible alternative.

Direct search methods were developed in the 1960s and are now becoming increasingly popular in the optimization community (Wright, 1998). Systematic experimentation is done either directly on the process parameters or on a dynamic model's inputs to perturb the process to different operating regions and measure the performance index in these regions (Monder, 2000). Given enough experimentation, the operating regime that yields an acceptable performance index or objective function can be found. The general advantage of these methods is that they are able to find the optimal decision parameters using only the value of the function to be optimized to guide the search, even if the response surface contains discontinuities. Moreover, few measurements are needed to calculate the new movement towards the optimum (Xiong and Jutan, 2000). The main disadvantage of derivative-free methods is the heavy computational load due to the iterative experimentation policy and a typically slow rate of convergence. For large-scale problems, these methods may thus prove difficult, especially if the process itself is used and not its model.

Several popular direct search methods exist, two of which are the Nelder-Mead Simplex method and Genetic Algorithms. The algorithms are generally simple, and the techniques have been implemented in various fields with a reasonable degree of success. Their general popularity and extensive coverage in literature contributed to the choice of these two methods for this work. The following two subsections describe the algorithms and present examples of their implementation.

3.1.1 Nelder-Mead Simplex Method

Conceptually simple and widely applicable, the Nelder-Mead (NM) Simplex method is by far the most popular direct search method used in a variety of fields ranging from chemistry to chemical engineering to medicine (Wright, 1998; Xiong and Jutan, 2000). The technique is based on theory originally developed by Nelder and Mead (1965) to minimize a scalar-valued non-linear function of n real variables. At each step, a *simplex* of n+1 vertices (real n-vectors) is maintained and modified until the defined termination criteria are satisfied. The general algorithm is described below (Wright, 1998) and presented in Figure 3.1.

The major assumptions are that the decision variables are continuous and that the response surface contains a global minimum (Nelder and Mead, 1965). We consider the initial polytope of n+1 vertices x. For minimization studies, the objective function is evaluated at these points, and the responses are ordered to satisfy:

$$f(x_1) \le f(x_2) \le \ldots \le f(x_{n+1}) \qquad (3.1)$$

Furthermore, $\bar{x}$ is the centroid of the n best points, that is of all the vertices except for x_{n+1}:

$$\bar{x} = \sum_{i=1}^{n} x_i / n \qquad (3.2)$$

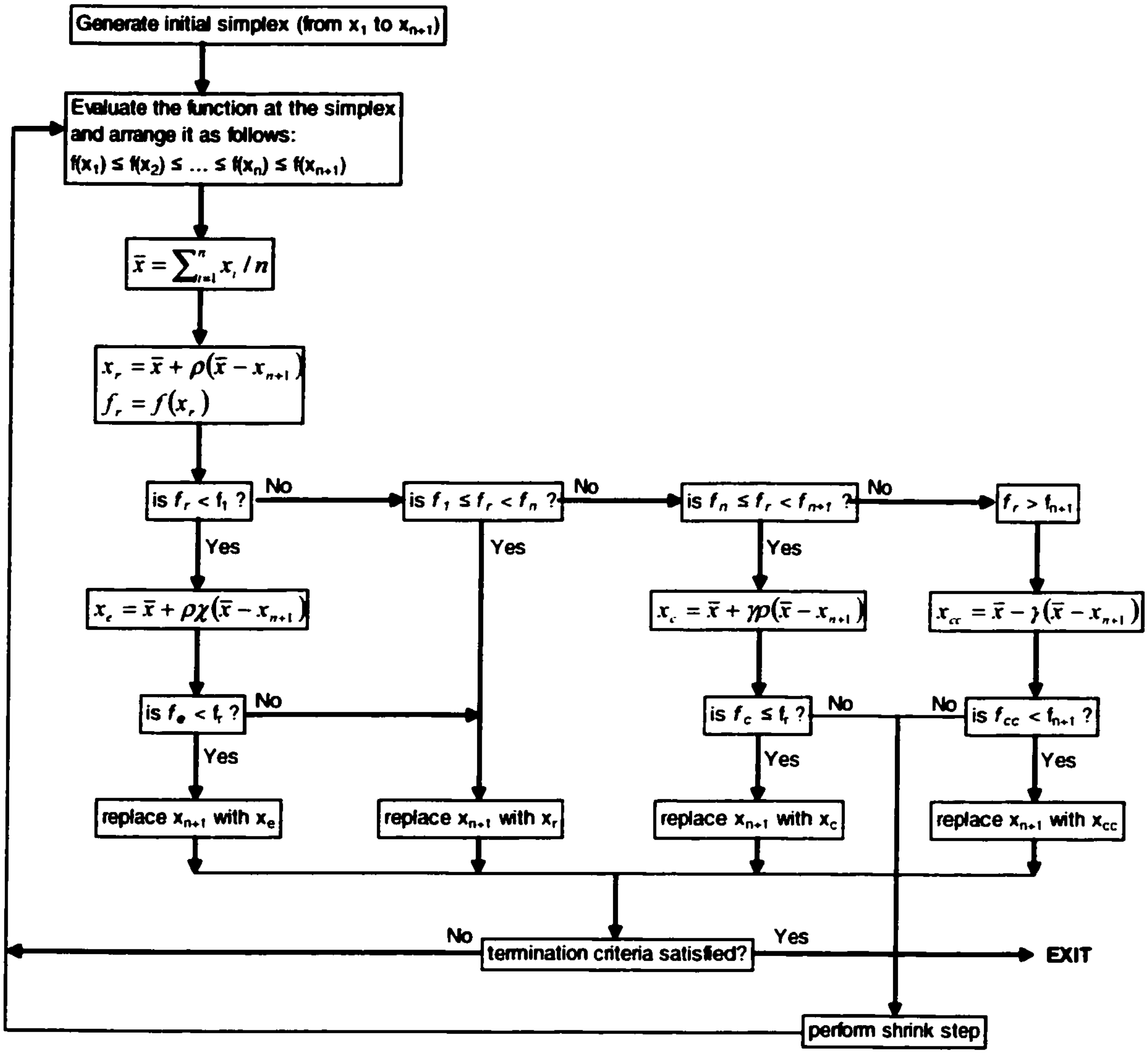


Figure 3.1: The Nelder-Mead Simplex Algorithm (Nelder and Mead, 1965; Wright, 1998)

At each step of the process, x_{n+1} is replaced by a new ("better") point using one of the three operations: reflection, expansion and contraction. The reflection point is first computed:

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1}), \qquad (3.3)$$

where ρ is a positive coefficient, the reflection coefficient. The function is then computed for x_r and written as f_r. If $f_1 \le f_r < f_n$, then the reflected point x_r replaces x_{n+1} in the new simplex and the iteration is terminated.

If $f_r < f_1$, the expansion point is calculated:

$$x_e = \bar{x} + \rho\chi(\bar{x} - x_{n+1}), \qquad (3.4)$$

where χ is the expansion coefficient, greater than unity. The function f_e is evaluated at the expansion point. If $f_e < f_r$, the point x_e replaces x_{n+1} and the iteration is terminated. Otherwise (if $f_e \ge f_r$), the reflection point x_r is accepted as the replacement of x_{n+1} and the iteration terminated.

If $f_r \ge f_n$, a contraction step is performed. For cases where $f_n \le f_r < f_{n+1}$, an *outside* contraction is done, where the point x_c is calculated:

$$x_c = \bar{x} + \gamma\rho(\bar{x} - x_{n+1}), \qquad (3.5)$$

where γ is the contraction coefficient lying between 0 and 1. If $f_c \leq f_r$, the point x_c is accepted as the replacement of x_{n+1} and the iteration is terminated.

If $f_r \geq f_{n+1}$, an *inside* contraction is performed and the point x_{cc} is calculated in the following way:

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) \tag{3.6}$$

The function f_{cc} is evaluated at x_{cc}. If $f_{cc} < f_{n+1}$, the point x_{cc} replaces x_{n+1}, which terminates the iteration.

There is no standard procedure for cases where $f_c > f_r$ in the outside contraction step or where $f_{cc} \geq f_{+1}$ in the inside contraction step. In the original algorithm (Nelder and Mead, 1965), the authors suggest responding to the failed contraction step by replacing all x_i's by $(x_i + x_1)/2$ and restarting the entire process. Wright (1998) suggests performing a *shrink* step, where the function is evaluated at n points $v_i = x_1 + \sigma(x_i - x_1)$ for $i = 2, \ldots, n+1$, and where σ is the shrinkage coefficient (typically between 0 and 1). The (unordered) vertices of the polytope used in the following iteration are then $x_1, v_2, \ldots, v_{n+1}$. Fortunately, failed contractions occur rarely in practice (Nelder and Mead, 1965), (Wright, 1998). Thus, the NM algorithm typically requires only one or two evaluations of the objective function per iteration, which is one of its major advantages over other direct search methods (Wright, 1998).

Termination criteria normally involve an analysis of the function values in the simplex and their variation from one iteration to the next. Nelder and Mead (1965) suggest halting the procedure once the "standard error" of the function values in the form of $\sqrt{\sum (f_i - \bar{f})^2 / n}$ reaches a pre-defined value. A more heuristic way would involve exiting the procedure once the variations between the function values from one iteration to the next become small enough.

The Nelder-Mead Simplex method is capable of handling process constraints. The most effective way of implementing constraints (especially in the case of processes with more than one input x) is the penalty function method, which involves modifying the objective function to take on an artificially high value in response to inputs that violate the constraint (Nelder and Mead, 1965). The attempts of the simplex to "trespass" the constraints will then be handled by a contraction step. However, the penalty function method will not be used in the scope of this work. Rather, implicit constraints will be implemented, as it will be described in Section 3.4.2.

The shortfalls of the method include the lack of a general convergence theory (Wright, 1998). Apart from that, the Nelder-Mead Simplex can be prone to stagnation, failure and convergence at local minima. Standard approaches to overcome this problem include starting with a sufficiently large initial simplex and attempting different starting polytopes. Hall and Bowden (1997) have found that the method is good if the response surface is generally bowl-shaped, even if local optima are present. A global optimum convergence criterion suggested by Nelder and Mead (1965) involves continuing after the first convergence for a certain number of evaluations in order to see if the second solution converges sufficiently close to the first one.

3.1.2 Genetic Algorithms

Inspired by the principles of natural selection, Genetic Algorithms (GA) have recently become another very popular direct search optimization method. A problem to be optimized is encoded into a series of bit strings (*chromosomes*) that are manipulated by the algorithm using three major operations: mutation, crossover and selection. The iterations are repeated according to the principle of survival of the fittest until a near-optimum solution is found (Chambers, 1995). Genetic algorithms are a powerful adaptive search method for practical optimization problems with complex or unpredictable solution spaces. The algorithm is robust and, unlike the Nelder-Mead simplex method, resistant to getting trapped in local optima (Bingül *et al.*, 2000). It is also highly efficient for multi-objective optimization problems. Disadvantages include choosing efficient, problem specific, encoding and implementation strategies – a task that is often far from obvious in heuristic methods such as GA (Chambers, 1995). There is also a lack of standardized techniques for determining chromosomes for crossover, mutations, etc.

The algorithm starts with a fixed population size (pool of candidate solutions). With each iteration, chromosomes are probabilistically selected from the population for reproduction according to survival of the fittest (Chambers, 1995). The performance of each chromosome is then evaluated in terms of the fitness function. Unattractive chromosomes are rejected and new ones are made using the selection, crossover (reproduction), mutation and replacement operations. The mechanics of GA algorithms are very simple (Bingül *et al.*, 2000), which contributes to their popularity. Chambers (1995) presents the following general logic for GA's:

```
procedure GA
begin
  t = 0;
  initialize Population P(t)
  evaluate structures in P(t)
  while termination condition not satisfied do
  begin
    t = t + 1;
    P(t) = select from P(t-1)
    alter structures in P(t)
    evaluate structures in P(t)
  end
end.
```

The central GA operations are based on genetic evolution rules of observed in nature. *Selection* is one of the fundamental operators in genetics. The original population must be sufficiently large in size and varied over the search space. Most commonly, it is generated randomly. The population size is an important parameter that determines the effectiveness of the method. A population size that is too small introduces the risk of not reaching the optimal solution. Choosing a large population, improves the learning rate of the algorithm (finding the neighbourhood of the optimal solution), but can lower the method's efficiency (Bingül, *et al.*, 2000).

Crossover is the operation that is analogous to biological reproduction. It is a very powerful step as it introduces new chromosomes ('offspring') to the population. The general idea behind the crossover operation is the exchange of one or more bits between two chromosomes of the same length to generate two offspring. Below is an example of a crossover between two chromosomes of eight genes:

$$\begin{array}{ccc} 1\,0\,1\,1\,0\,1\,\underline{0\,0} & & 1\,0\,1\,1\,0\,1\,\underline{0\,1} \\ \text{X} & \rightarrow & \\ 1\,1\,0\,1\,0\,0\,\underline{0\,1} & & 1\,1\,0\,1\,0\,0\,\underline{0\,0} \end{array}$$

The rate of crossover can have a significant effect on the efficiency of the method depending on the application. Bingül *et al.* (2000) found that using crossover rate values that are too low (0.1 – 0.3) or too high (0.9) can result in settling at non-optimal fitness function values. In their application, values of between 0.6 and 0.7 were found to be optimal.

The *mutation* operation mimics the sudden changes that occasionally occur in nature. Random alterations are done on chromosomes picked out of the population pool. For example, the chromosome 1 1 1 1 1 1 1 can be replaced by 1 1 0 1 1 1 1. The mutation step is less critical than the crossover operation, but it may be essential to prevent premature convergence (Bingül *et al.*, 2000). As the optimal solution is approached, the chromosomes in the population become all very similar, risking getting trapped in a local optimum. The mutation step acts against that by reviving the population. Mutation can, however, provoke the risk of inefficiency or damaging good genetic material. Bingül *et al.* (2000) found that a mutation rate of 0.02 was a good choice. At low mutation rates (0.001), the algorithm displays a very slow learning rate and is inefficient, whereas at higher rates (such as 0.1), the final solution is reached faster but is not the optimal one.

A generic challenge in implementing genetic algorithms is the representation (coding). Continuous physical solutions are not easy to represent in terms of bit strings. Yet, representation proves to be an important factor influencing the success and robustness of a GA (Bingül *et al.*, 2000). Genetic algorithm coding is very problem specific. A common method for representing a real number is to express it with a four-digit chromosome where each of the digits is multiplied by 1, 2, 3 and 4, respectively. For example, 0111 translates to 0*1 + 1*2 + 1*3 + 1*4 and represents the number 7. For numbers containing several significant digits, a separate chromosome is used for each significant digit. This method was deemed suitable and was used in this work.

Genetic algorithms suffer from the same major disadvantage as all direct search methods: they have no concept of the optimal solution. There is no way of determining with certainty that the obtained solution is the best one available. Compared to the Nelder-Mead simplex, GA are more difficult to implement, but with more efficient exploration capabilities are less prone to converging at a local minimum and more dependable to find the near-optimum solution.

3.2 Problem Statement and Description

Paper breaks are a random phenomenon in the papermaking process and their occurrence is for the large part unpredictable. They are at the same time a major perturbation to the process (Orccotoma, *et al.*, 1997). The broke and white water inventories are affected during paper breaks and need to be controlled. Inventories in the broke system increase and broke pulp may increasingly be reused in the process to prevent potential tank overflows. Meanwhile, whitewater levels decrease as the water is used to dilute the broke that is produced. When the whitewater levels become low, fresh water is typically added. The goal of the operator during paper breaks is to handle this perturbation and restore the production with minimal impact on the stability of the paper machines.

The effects of recirculating broke pulp back into the process following a paper break are widely discussed in the literature. As shown in Table 3.1, broke pulp has different properties than virgin pulp (TMP, DIP) and reusing it perturbs the properties of the machine furnish. Fibre length distribution is different in broke and fresh pulp for two reasons: a) fines are much less retained in the paper sheet than long fibres and b) the dilution waters used for diluting broke and fresh pulp have different fibre distributions (Bonhivers, *et al.*, 2000; Orccotoma, *et al.*, 1997). Broke pulp also contains a lower composition of dissolved solids. Variations in the properties of the mixed pulp can greatly affect the retention of the machines and water drainage. These variations, in turn, affect the stability of the paper machines and, ultimately, can cause subsequent breaks. In fact, stock quality variations have been identified as one of the most frequent causes for wet-end paper breaks (Orccotoma, *et al.*, 1997).

Table 3.1: Comparison of properties of virgin pulp and broke pulp.

Property	TMP	DIP	Broke
Consistency (%)	3.4	4.0	3.2
Temperature (°C)	65	52	58
Fibre Content (%)	79	79	49
Fine Content (%)	21	21	51
Dissolved Solids Content (%)	0.4	0.4	0.08

The most important mixed pulp properties that should be maintained as constant as possible are consistency, fibre length distribution, dissolved solids content and temperature (Bonhivers, *et al.*, 2000). Theoretically, the broke ratio should be kept at a constant level to achieve steady paper machine headbox conditions. In practice, this is impossible as the inventory of the broke collection system needs to be controlled in order to avoid possible overflows. In effect, the rate of broke recycling ("broke ratio") that can be applied without causing additional perturbations is a constraint, as is the inventory of the broke system (Orccotoma, *et al.*, 1997).

At the mill studied, the magnitude of broke ratio varies between 5% and 45% during normal operation, depending on the machine, the type of paper produced and other factors. However, to control the level in the broke storage tank during and following breaks, the broke ratio for one or several machines is often adjusted, typically by 5% or 10% at a time. This action is performed manually according to operator experience. The purpose of this work is to simulate and evaluate current operating conditions using the dynamic model and then to apply direct search methods in an attempt to find better solutions. The focus is on the management of broke recirculation ratios during and following paper machine breaks. The performance of different operating strategies will be measured using an objective function.

3.3 Current Operating Strategies

In response to prolonged or sequential sheet breaks, the mill operator often increases the broke ratio on one or several machines in order to maintain a desirable level in the broke storage tanks. This adjustment is performed in anticipation of an excessive increase in the level of the broke tanks. Conversely, after periods of relatively stable operation, when few breaks have occurred, the broke ratio is lowered in order to avoid draining the broke storage tank. The mill's operating objective to maintain the level of the broke storage (50T) tanks between 30% and 50%. Theoretically, the manipulation of the broke ratio is related to break occurrence. In reality, however, this action is to a large degree arbitrary and depends mainly on operator experience and judgment. The standard broke ratio change of 5% or 10% is accomplished by compensating the ratio of TMP or DIP fed into the machine (depending on fresh pulp inventories, type of paper produced, etc.)

Figure 3.2 shows the pattern of adjustment of the broke ratio for the four machines over a span of 15 hours. The paper machine breaks are also shown, as well as the inventory level in the broke storage tanks. It can be seen that the broke ratios on machines 6, 8 and 9 are repeatedly increased by 5% following a series of simultaneous breaks between 5 o'clock in the morning and noon. Eventually, when all the broke ratios are at 35%, the level in the broke storage tank settles around 70%. In the afternoon there are very few breaks, and the level in the broke storage begins to decrease. The broke ratio values are then gradually brought back down to their original values. Just before 18 o'clock, a few

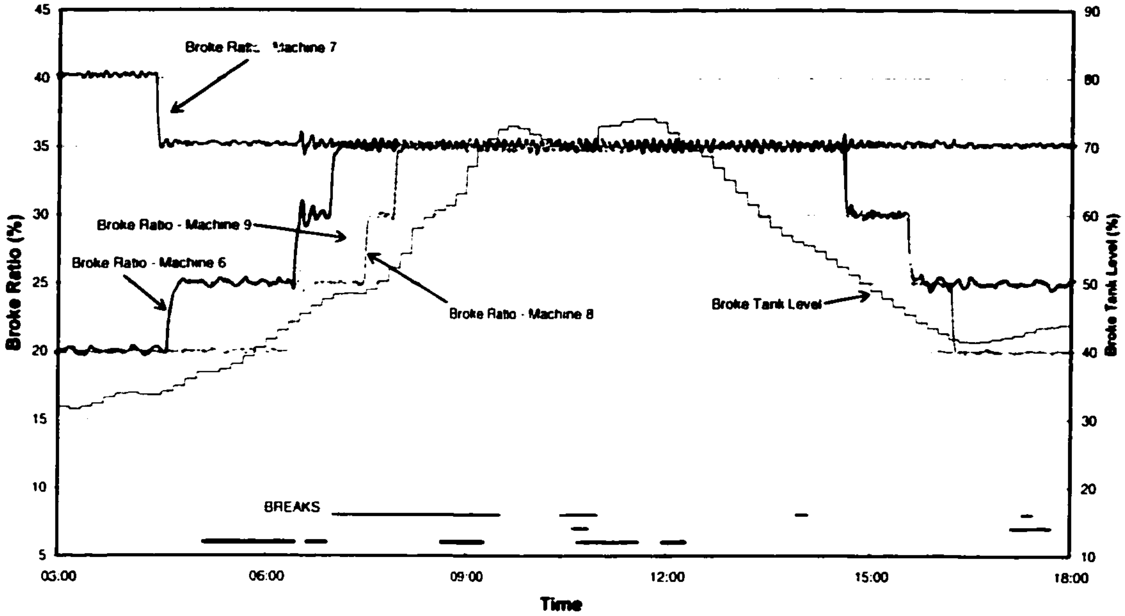


Figure 3.2: Broke Storage Level and Broke Recirculation Ratios in Response to Breaks.

more breaks occur, and as the broke ratios remain fixed, the level in the storage tank begins to rise once again. It can be generalized that broke ratio adjustments are performed in response to break occurrences and broke storage inventories, and that the action remains manual and ultimately depends on operator judgement.

The sudden and significant increases in the broke ratio are expected to have caused increased variability in the pulp properties at the headbox. Figure 3.3 shows the simulated response in some headbox parameters following a change of the broke ratio on Machine 8 from 20% to 25%. To benchmark the level of these variations, an objective function is developed and presented in the following section. The hypothesis of this work is that the headbox variations can be decreased by managing the broke recycle rate more efficiently. For example, the changes can be made in a smoother manner and over a longer period of time. Direct Search methods can be applied to systematically seek out improved broke ratio adjustment profiles. The solution approach is described in Section 3.5.

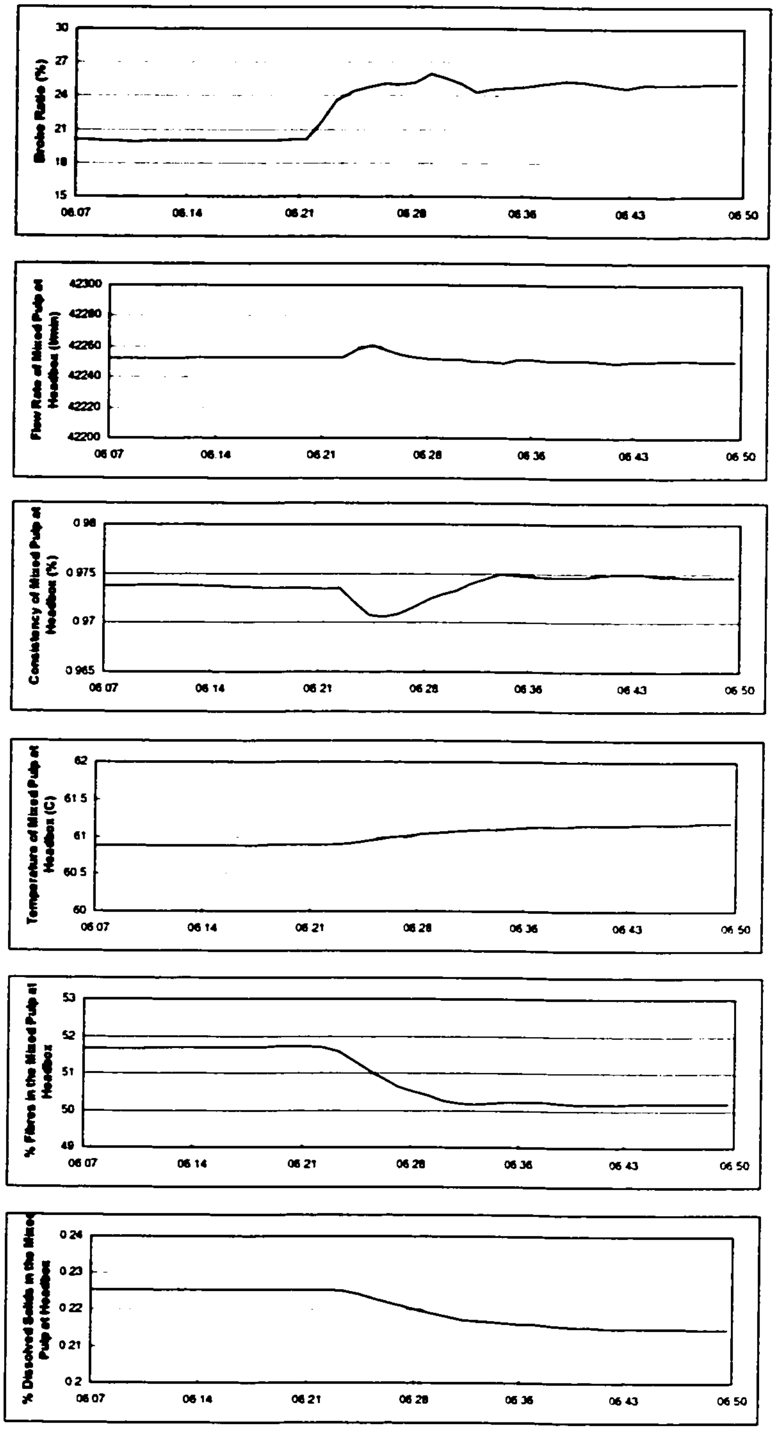


Figure 3.3: Pulp Properties at the Headbox following a change in the Broke Ratio.

3.4 Objective Function Development

The performance of the cases studied is gauged using an objective function with constraints. An objective function is written such that when evaluated, it expresses how close a given solution has come to satisfy the objectives of the experiment. The variables used to manipulate the solution are called manipulated variables, and their field of operation is restricted by constraints. This section will describe the objective function used for this work and the associated process constraints.

3.4.1 Stabilization of Headbox Conditions

As mentioned in the previous section, the main objective of this work was to minimize variations in the properties of the mixed pulp at the headbox of the paper machines that result from changes in the broke ratio during and following sheet breaks. The required objective function needs to express these variations for a given process operation scenario tested with the dynamic model. It should be noted that although discussions in this work revolve around function minimization, they could be easily adapted to maximization problems by multiplying the objective function by -1.

The following objective function (hereafter referred to as Objective Function #1) is proposed for each of the four paper machines and represents the weighted sum of changes in flow rate, consistency, temperature, fibre fraction and dissolved solids content at each time interval:

$$F_{obj} = \sum_{t=1}^{t} \left[a\left|Q_t - Q_{t-1}\right| + b\left|C_t - C_{t-1}\right| + c\left|T_t - T_{t-1}\right| + d\left|F_t - F_{t-1}\right| + e\left|S_t - S_{t-1}\right| \right] \quad (3.7)$$

where:

- Q: flow rate of the pulp stream at the headbox;
- C: consistency of the pulp stream at the headbox;
- T: temperature of the pulp stream at the headbox;
- F: fraction of fibres in the pulp stream at the headbox;
- S: weight fraction of dissolved solids in the pulp stream at the headbox;
- a, b, c, d, e: weighing coefficients;
- t: simulation time.

The weighting coefficients are chosen arbitrarily based on the relative importance of the terms as well as on their contribution to the objective function. Alternatively, the following variation of the objective function (hereafter referred to as Objective Function #2) can be used:

$$F_{obj} = \sum_{t=1}^{t} \left[a(Q_t - Q_{t-1})^2 + b(C_t - C_{t-1})^2 + c(T_t - T_{t-1})^2 + d(F_t - F_{t-1})^2 + e(S_t - S_{t-1})^2 \right]$$

(3.8)

In this case, the individual parameters contribute more to the overall value of the function making the response more sensitive to spikes. The squared terms are also less sensitive to noise in the parameters.

3.4.2 Constraints

The following are the two major process constraints that are considered in this work:

1. The level in the broke storage tank must be maintained between minimum and maximum limits, and
2. The broke ratio (control input) should be allowed to vary within a specified range only.

The original reason for adjusting the broke ratio in response to paper breaks is to maintain the level of the broke tanks at a desirable level and to prevent the tank from overflowing or drying up. It is therefore clear that this goal cannot be compromised by the optimization scheme. The first constraint can be expressed as:

$$L(50T)_{\min} \leq L(50T)_t \leq L(50T)_{\max} \tag{3.9}$$

where $L(50T)_t$ is the level of the broke storage (50T) tank at time t and $L(50T)_{min}$ and $L(50T)_{max}$ are the lower and upper limits, respectively. This implicit constraint is handled directly in the simulation in a manner similar to the method used at the mill. When the level in the broke reservoir drops below 20%, an additional amount of rejected paper rolls is repulped and added to the broke handling tank. When the level is excessively high (above 80%), some of the broke pulp is sent back to the DIP storage tank in the pulp preparation section.

Restricting the movement of the broke ratio is aimed at preventing these inputs from assuming potential infeasible values that may be dictated to them by the optimization algorithm. The second constraint can be expressed as follows:

$$BR_{\min} \leq BR_t \leq BR_{\max} \tag{3.10}$$

where BR_t is the value of the broke ratio at time t and $BR_{\min}$ and $BR_{\max}$ are the minimum and maximum allowable values. The minimum broke ratio value is 0%, as the machines can be operated using fresh pulp only. The upper limit was set at 50% because, as was indicated by mill personnel, the TMP composition of the mixed pulp needs to be at least 50% to ensure machine stability. These limits were also implemented directly into the process simulation and can be regarded as an implicit constraint.

3.5 Solution Approach

This section explains the approach taken to achieve the objectives of this work using the dynamic process model and the direct search optimization algorithms. First, the Nelder-Mead Simplex and Genetic Algorithms codes will be presented. A description of the interface linking the model with the algorithm code and the overall implementation of the optimization scheme follows.

3.5.1 Algorithm Coding

The two direct search algorithms presented in Section 3.1 were programmed using the Visual Basic Application coding in Microsoft Excel. This program is sufficiently versatile for the needs of the algorithms and has the advantage of working directly in Excel with the simulation's input and output spreadsheet. Both algorithms were programmed using macro modules for each of their operations. The code for the macro modules is included in Appendix III.

3.5.2 Model – Algorithm Interface and Implementation

The interface linking the dynamic simulation with the algorithm coding in VBA was created in the WinGEMS source code. The coding capabilities in this software are generally primitive and not well adapted for custom modifications, but two features are very useful for this application: the ability of dynamically reading inputs and writing outputs and the command for calling and executing macros in Excel at the appropriate time during the simulation.

The general implementation approach used during each simulation run was as follows:

1. Start simulation, converge at steady state and switch to dynamic mode;
2. Read inputs from the spreadsheet in real-time;
3. Call algorithm macros at appropriate times during the simulation;
4. Write outputs to the spreadsheet in real-time.

The same dynamic model was used for the two optimization algorithms. The user has the option of first simulating the original case to evaluate the mill original operating scenario. The desired optimization algorithm and number of iterations can then be specified within WinGEMS. Other data, such as which machine is to be optimized and the initial values for the simulation are read from Excel through the dynamic interface. More details on the case study setup and execution are given in Section 4.1. The complete code for the model-spreadsheet interface is included in Appendix III.

CHAPTER IV

CASE STUDY

The process model described in Chapter 2 was used to evaluate the current operating strategies and benchmark them against modified operating schemes developed with the two direct search optimization algorithms that were presented in Chapter 3. As the same model is used to simulate the actual mill operation and the optimized strategies, there is no structural mismatch. The case study has two major purposes: to confirm whether direct search optimization methods are successful at finding an improved method of adjusting the broke ratio so as to minimize the impact on pulp properties at the headbox of the paper machines; and to compare the results of two different direct search algorithms.

The general setup of the case study is presented in Section 4.1. The data collected at the newsprint mill and used in the study is described, followed by the assumptions that were adopted and the approach that was used to carry out the simulations. Finally, the results and discussions of the study are presented for the two optimization algorithms

4.1 Case Study Setup

The original benchmark problem is based on real operation data collected for the 3rd of May, 2001. Initially, data for the entire first week of May 2001 was collected and examined, as it was confirmed by plant personnel that plant operations during that week could be considered trivial. One day was chosen out of that particular week for the following criteria:

- No machine was down for reasons other than breaks;
- Breaks occurred on each machine;
- Broke ratio changes were done for each machine.

Using data collected at the mill, plant operations during that day were first simulated under the assumptions listed in Section 4.1.2. The same day was then simulated for each iteration of the two optimization algorithms, this time with modified profiles of the broke ratio adjustments.

4.1.1 Summary of Trial Period Data

The values of the Broke and TMP ratios in the mixed pulp streams fed into the paper machines were collected on a minute-to-minute basis for the applicable time period. The ratio of DIP in the mixed pulp was then derived by subtraction. The flow rates of the mixed pulp streams were also measured at each minute, but for the purpose of the simulation, these flow rates were assumed constant. Finally, the exact break times for the wet-end and the dry-end of each machine were recorded. Figure 4.1 shows the values of the broke ratio for the four machines during the time period considered. Figure 4.2 shows the break occurrences for the paper machines, and Table 4.1 outlines the locations of and the reasons for the breaks, as noted in the Operations log book.

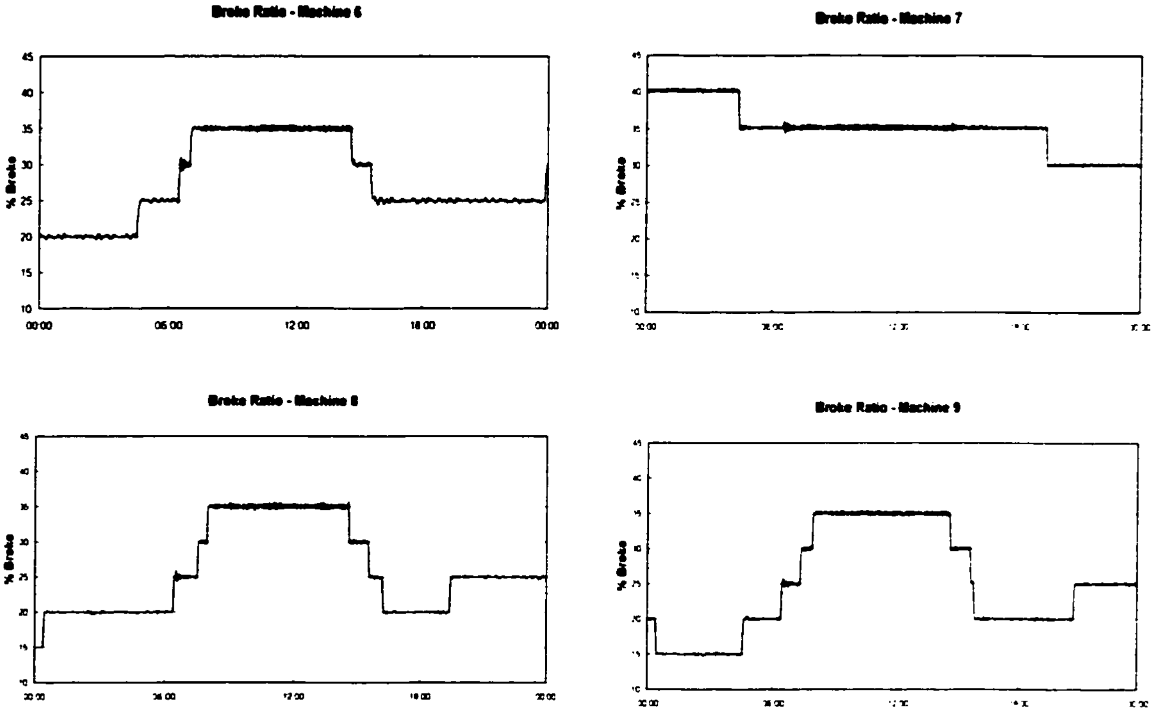


Figure 4.1: Broke ratio profiles for the paper machines during day of the case study.

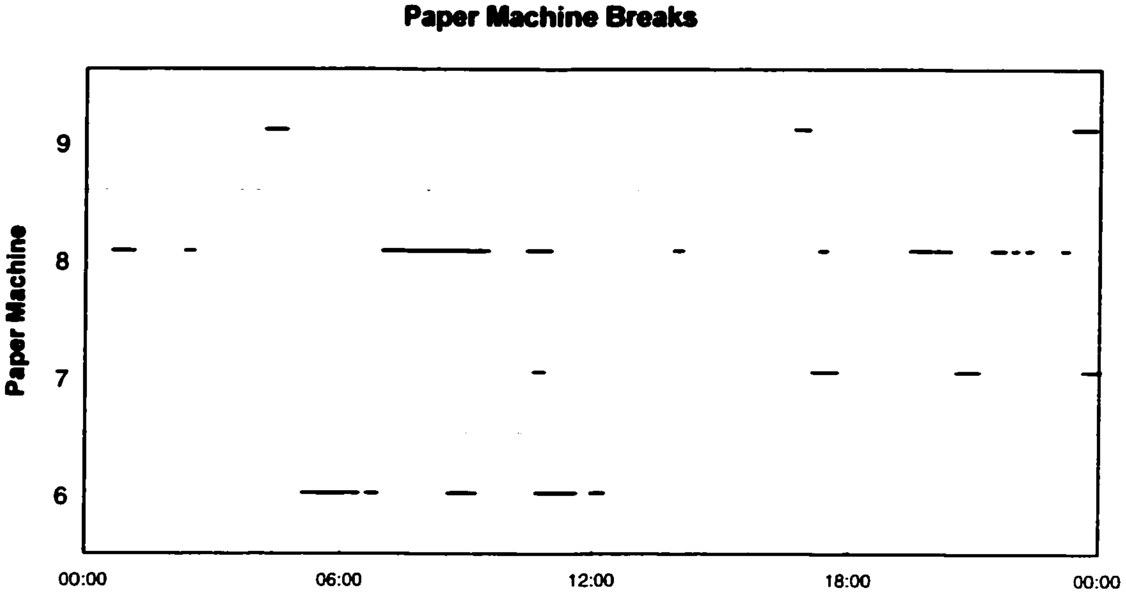


Figure 4.2: Break occurrences on the paper machines during the day of the case study.

Table 4.1: Break reasons and location of occurrence during the day of the case study.

Time	Machine	Break Type / Reason	Section	Time Lost, min
05:09	6	Lumps / Dirt	1st Press	75
06:40		Unknown	2nd Press	12
08:40		Lumps / Dirt	1st Press	35
10:45		Operations	Dry end	40
11:25		Mechanical	Dry end	10
12:00		Lumps / Dirt	1st Press	15

10:41	7	Lumps / Dirt	1st Press	11
17:12		Sheet Pull	Couch Roller	32
20:38		Unknown	Dryer	30
23:40		Lumps / Dirt	1st Press	70
00:40	8	Lumps / Dirt	Dryer	25
02:22		Measurex Breakdown	Winding	8
07:02		Measurex Breakdown	Dryer	60
08:02		Instrumentation Breakdown	Dry end	90
10:31		Cables	Dryer	15
10:46		Instrumentation Breakdown	Dry end	14
13:56		Other Operations	Winding	7
17:20		Unknown	Winding	6
19:28		Lumps / Dirt	Dryer	29
20:06		Sheet Cut	Dryer	20
21:30		Holes in Sheet	Winding	15
22:00		Other Operations	Winding	3
22:20		Unknown	Winding	3
23:10		Other Operations	Winding	5
04:15	9	Lumps / Dirt	Dryer	25
16:45		Unknown	Dry end	16
22:55		Unknown	2nd Press	28

In addition, the following data were collected at 10-minute intervals, for the purposes of simulation construction and validation. They are included in Appendix II.

- Flow rate and consistency of the mixed pulp stream
- Consistency of the pulp stream at the headbox
- Level and consistency of the broke transfer tank (30T)
- Level and consistency of the broke storage tanks (50T)
- Flow rate and consistency of the repulped rejected paper rolls
- Level, consistency and temperature of the whitewater buffer tank
- Dissolved solids content in the TMP and DIP pulp streams and in the whitewater buffer tank
- Dissolved solids content in the pulp stream at the paper machines
- Consistency of the cloudy whitewater stream entering the papermaking section from the TMP shop
- Flow rate of lean whitewater to secondary treatment

4.1.2 Assumptions

The following assumptions were adopted for the case study:

- The dynamic simulation was considered to be the perfect process model and was used to simulate and compare the results of both the original mill procedures and the optimized operation strategies;
- The magnitude and time of the changes to the broke ratio were kept the same as in the original mill data (only the profile of the adjustment was modified);
- Changes to the TMP and DIP ratios alone (not accompanied by broke ratio changes) were left unchanged from the original mill data;
- The trajectory of a shift in the broke ratio assumed a 5-minute logarithmic response as described in the next sub-section.

In addition, the following parameters were kept constant in the dynamic model and are summarized in Appendix II:

- Constant flow rate of the total mixed pulp stream (TMP, DIP and Broke pulp) to each machine;
- Constant consistency, temperature, fibre content and dissolved solids content of TMP and DIP;
- Constant flow rate, constancy, temperature, fibre content and dissolved solids content of the repulped rejected paper rolls;

- Constant flow rate, consistency, temperature, fibre content and dissolved solids content of the cloudy whitewater entering the papermaking section from the TMP plant;
- Constant flow rate and temperature of fresh water added to the paper machines.

4.1.3 Case Study Execution

The optimization case study was run for twenty iterations of each algorithm for each of the two objective functions. The simulations were done on Pentium 4 – 1.6 GHz processor using WinGEMS 5.0 and Excel XP with VBA. The results of the runs as well as the simulation times were recorded and compared.

The same one-day simulation was run to simulate the original operation and subsequently for each iteration of the two optimization algorithms. The algorithms were run on one paper machine at a time, while the profiles of the broke ratio changes were kept constant (at the best input) for the other three. The time increment used for the simulation was one minute. Identical initial values were loaded at the beginning of each run to ensure valid comparable results. The initial values were based on the operational data collected at the mill for the beginning of the day that was simulated.

Four changes were considered for the optimized change profiles, one every five minutes. The total 20-minute adjustment was programmed into the input data to start 10 minutes before and finish 10 minutes after the original time of change in order to maintain the global mass balance in the broke storage inventories. The magnitudes of each of the four changes (expressed as fractions of the total change) were left as the degree of freedom for the optimization algorithms.

Each shift was modeled using a logarithmic function to represent the process response at the mill. It was noted in the original data and confirmed by mill personnel, that the response time to a change in the broke ratio is around five minutes and assumes a logarithmic shape. The following general logarithmic function was used to model this behaviour:

$$C_t = C_{ti} + (C_{tf} - C_{ti})\frac{\ln(t+1)}{\ln(6)}, \qquad (4.1)$$

where C_t is the value of the broke ratio at time t (ranging from 0 to 5 minutes), and C_{ti} and C_{tf} are the values of the broke ratio at the beginning and end of the five-minute change period, respectively.

Figure 4.3 illustrates a comparison between the broke ratio change trajectory originally performed on Machine 8 at 7:30 o'clock, and the near-optimum change profile calculated by the Nelder-Mead Simplex for Objective Function #2.

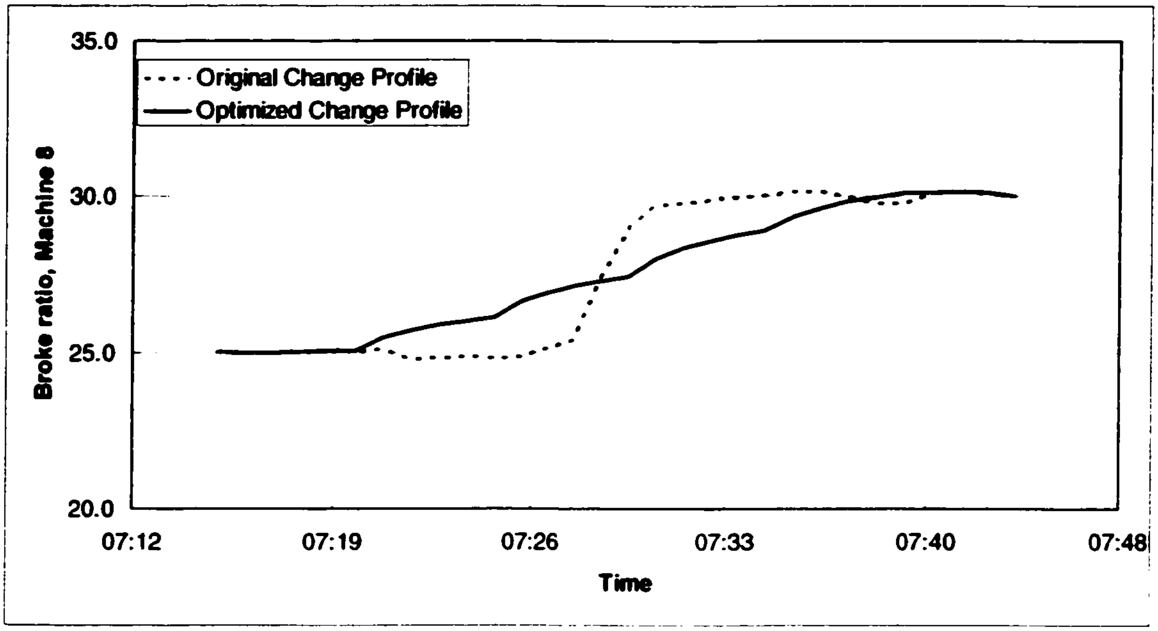


Figure 4.3: Example of original and optimized broke ratio change profiles.

One of the goals of the case study was to compare the performance of the Nelder-Mead Simplex method and Genetic Algorithms. At first, ten iterations were run for each objective function, as it was noticed that the two algorithms converge to similar results after ten iterations. Subsequently, ten more iterations were done for both algorithms to investigate whether the improvement in the objective function values warranted additional iterations.

A five-member simplex and a population size of 20 were used in this study for the Nelder-Mead Simplex and the Genetic Algorithms, respectively. The original simplex for the NMS method and the original generation for the GA were chosen using a random number generator. The coefficients for the two algorithms were taken from literature.

For the Nelder-Mead Simplex, the following standard coefficients were used (Wright, 1998):

$$\rho = 1, \chi = 2, \gamma = \tfrac{1}{2}, \sigma = \tfrac{1}{2}$$

The mutation and crossover rates for Genetic Algorithms were taken from two sources. Bingül *et al.* (2000) suggests using 0.1 for the rate of mutation, and 0.6 for the crossover probability. Goldberg (1989) confirms that a crossover rate of 0.6 gives good results for most applications. Thus, the above values were used for this study. Three best chromosomes were conserved during the crossover operation and the best chromosome was made immune to mutation.

The following coefficients were used for both objective functions:

- flow rate: a = 0.1;
- consistency: b = 1;
- temperature: c = 0.6;
- % fibres: d = 1;
- dissolved solids: e = 1.

4.2 Results

Two major results were observed during the case study. First, the time required to complete one iteration and the entire run were recorded for both algorithms (Section 4.2.1). Then, the values of the objective function for each paper machine, obtained with both algorithms after 10 and 20 iterations, were compared to the values of the objective function corresponding to the original mill scenario for broke recirculation (Section 4.2.2).

4.2.1 Execution Time

The time required to simulate one day of operation at one-minute increments using the computer described in the previous section is in the order of three minutes. The NMS method first runs five simulations for the initial simplex and after that, each iteration requires only two simulation runs. Thus, a typical iteration takes about six minutes. The GA algorithm, due to its structure, runs a simulation for each member entire generation at each iteration. Thus, a typical iteration for a 20-member generation takes significantly longer: about one hour. Table 4.2 summarizes the average run times for the two algorithms.

The total execution time of 20 iterations for two objective functions was thus in the order of 40 hours for the NMS algorithm and just under one week for GA.

Table 4.2: Simulation times for the Nelder-Mead Simplex and Genetic Algorithms.

Algorithm	One 24-hour Simulation	One iteration	One Machine (10 iterations)	Four Machines (10 iterations)
Nelder-Mead Simplex	3 minutes	6 minutes	1 hr 15 minutes	5 hours
Genetic Algorithms	3 minutes	1 hour	10 hours	40 hours

4.2.2 Objective Function Results

The solutions of the algorithms are expressed as the fraction of the total broke ratio adjustment after the first, second and third 5-minute interval. After the fourth change, the broke ratio reaches the final value. Table 4.3 summarizes the results found by the two algorithms for Objective Function #1 after twenty iterations. Table 4.4 shows the same results for Objective Function #2.

Table 4.3: Near-optimum change profiles that minimize Objective Function #1.

Objective Function #1	Machine	Initial Value	% Total Change after 1st Adjustment	% Total Change after 2nd Adjustment	% Total Change after 3rd Adjustment	Final Value after 4th Adjustment
Nelder-Mead Simplex	6	0	16.9%	33.1%	67.4%	1
	7	0	29.7%	48.1%	75.3%	1
	8	0	14.7%	55.8%	81.4%	1
	9	0	19.1%	39.6%	68.0%	1
Genetic Algorithms	6	0	30.0%	52.3%	67.6%	1
	7	0	20.2%	54.9%	74.5%	1
	8	0	21.7%	59.3%	81.7%	1
	9	0	28.6%	49.7%	72.8%	1

Table 4.4: Near-optimum change profiles that minimize Objective Function #2.

Objective Function #2	Machine	Initial Value	% Total Change after 1st Adjustment	% Total Change after 2nd Adjustment	% Total Change after 3rd Adjustment	Final Value after 4th Adjustment
Nelder-Mead Simplex	6	0	23.6%	43.6%	73.7%	100%
	7	0	26.2%	48.2%	77.6%	100%
	8	0	20.8%	46.8%	76.3%	100%
	9	0	30.0%	51.9%	73.9%	100%
Genetic Algorithms	6	0	30.4%	52.7%	74.3%	100%
	7	0	23.1%	52.9%	80.5%	100%
	8	0	24.1%	53.7%	79.3%	100%
	9	0	25.0%	50.0%	78.6%	100%

Figure 4.4 shows the objective function values for Objective Function #1 for actual mill operation and for the two algorithms after ten and twenty iterations. Figure 4.5 shows the corresponding results for Objective Function #2.

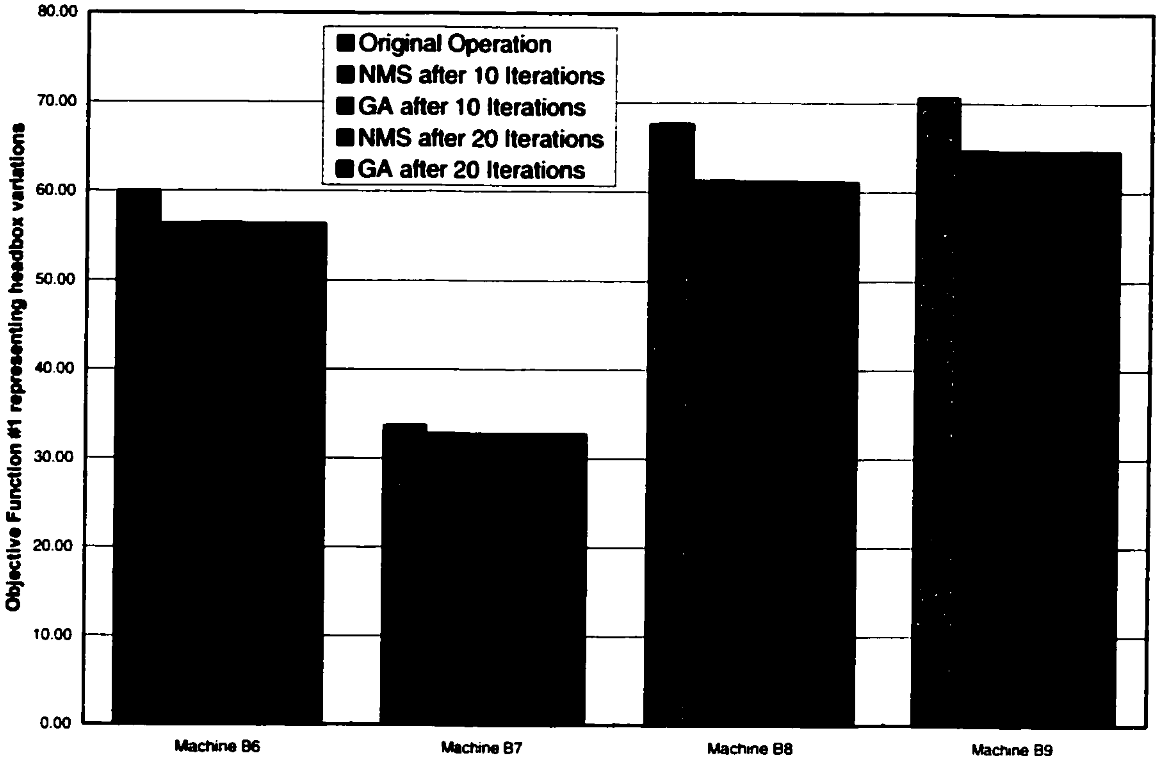


Figure 4.4: Values of Objective Function #1 before and after optimization.

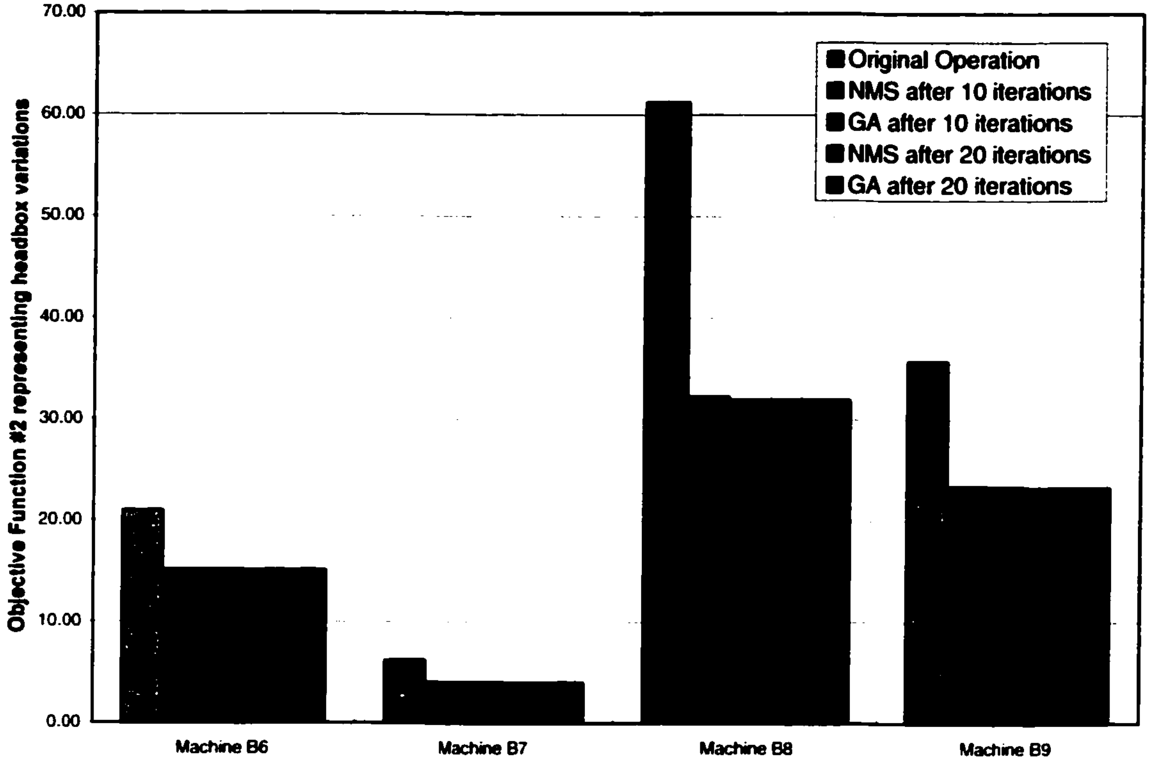


Figure 4.5: Values of Objective Function #2 before and after optimization.

4.3 Discussions

The broke recirculation strategies derived by the two algorithms successfully reduced the value of the objective function after ten iterations. Additional ten iterations yielded only slightly better results, thus, not warranting the necessity to attempt more. Both algorithms, though completely different in structure, successfully produced similar final results in all cases. Simulation time proved to be inefficient in the case of Genetic Algorithms as many more simulation runs are required for each iteration than for the

Nelder-Mead Simplex method. The results obtained with this work could serve as a proof of concept study to quantify the benefits of improving the management of broke recirculation.

This section will concentrate on the evaluation of the case study, as well as the extensions of the work and implementation possibilities.

4.3.1 Case Study Evaluation

The work that has been done and the case study presented in this chapter are only a small piece of the pool of possible solutions to the problem of headbox variations caused by broke recirculation. The model and case study assumptions, objective function coefficients and algorithm parameters that were considered for this work are subject to revision and sensitivity studies. The entire broke management strategy is at the present nearly completely dependent on operator judgment, and is a good candidate for automation and thorough optimization.

The time interval used in the dynamic model was of one minute. Several reasons contributed to this choice. Dynamic historical data is available from the mill's data acquisition system at every six seconds. Decreasing the time interval down to less than one minute could possibly further boost the sampling precision, but would also increase the simulation time, which in the case of Genetic Algorithms would make the method even less efficient. At longer time intervals, the simulation time would be shortened, but some trends in the dynamic responses would be lost due to less frequent sampling. The effects of the broke ratio changes on the pulp properties at the headbox are amplified in the minutes following the adjustment; therefore, by-the-minute sampling seemed to be appropriate.

The flow ratios of the virgin pulps (TMP and DIP) were kept at their original values. This study only looked at optimizing the profiles of the broke ratio adjustments because, as it was discussed in Section 3.2, broke pulp has different properties than virgin pulp and reusing it has the largest effect on the machine pulp furnish. It can therefore be expected that varying the ratio of the broke recirculation stream has the greatest effect on pulp property variations at the headbox. However, abrupt changes in the TMP and DIP ratios most likely also have an effect on headbox stability, especially given the differences in temperature and consistency between the different types of pulp. The rates of adjustment to the virgin pulp ratios are thus equally good candidates for analysis and improvement.

For the purpose of this study, four changes over twenty minutes were used for the following reasons:

- if the scheme were to be tested manually at the mill prior to implementation, performing four changes over twenty minutes to adjust the broke ratio would be feasible;

- if the scheme were to be implemented automatically, the control computer (DCS) could be programmed to execute four small changes or one gradual change over the twenty- minute span;
- increasing the span of the changes beyond twenty minutes could interfere with neighbouring adjustments, as the frequency of changes to the broke ratio is often less than half an hour (Figure 4.1).

One of the major assumptions considered in this study was that the changes originally done by the operators were justified and necessary. Thus, the magnitude and time of the changes to the broke ratio were kept the same as in the original mill data and only the profile of the adjustment was modified. An endeavour to improve or optimize the entire broke management strategy would involve introducing many more variables into the model: more exact and reliable mass balances in the broke and white water system, factors accounting for the grade of paper being produced, possibly differences between summer and winter production and so on. It could then be worthwhile to consider the work currently done by Xiong and Jutan (2000) on Dynamic Simplex Methods, where the Nelder-Mead Simplex method is extended to problems with moving optimum.

The "optimized" adjustment profiles minimize the objective function for the day of the study, but it remains to be confirmed whether the same profiles would be optimum (or near-optimum) on a different day when, for example, different paper were produced or a machine were down. More case studies would have to be done at different operating conditions to establish how case-dependent the near-optimum profiles are.

The purpose of the objective functions was to express and minimize the variations in the various properties of the pulp at the headbox following changes in the broke ratio. Two functions were used in order to double the number of approaches and observe the differences in the results. Objective Function #1 simply considers the sum of the weighted absolute differences of all the terms. Objective Function #2 is more sensitive to spikes in individual terms, as the difference for each term is squared. The results of the case study show that the solutions and the general conclusions were very similar for both functions.

A sensitivity study was performed to test the choice of objective function coefficients. The goal of this test was to confirm that there is no competition between the different terms in the objective function and that they all contribute in a similar way to the results. Objective Function #2 was used with one active term at a time (and the others set to zero) to run 20 iterations of the Nelder-Mead Simplex on Machine 8. As can be seen in Figure 4.6, the results follow a fairly similar path for all the coefficients, indicating that all the terms in the functions contribute with a similar trend to the outcome of the optimization. A separate study into the net contribution of the individual terms could be conducted to find the optimum magnitude of the coefficients.

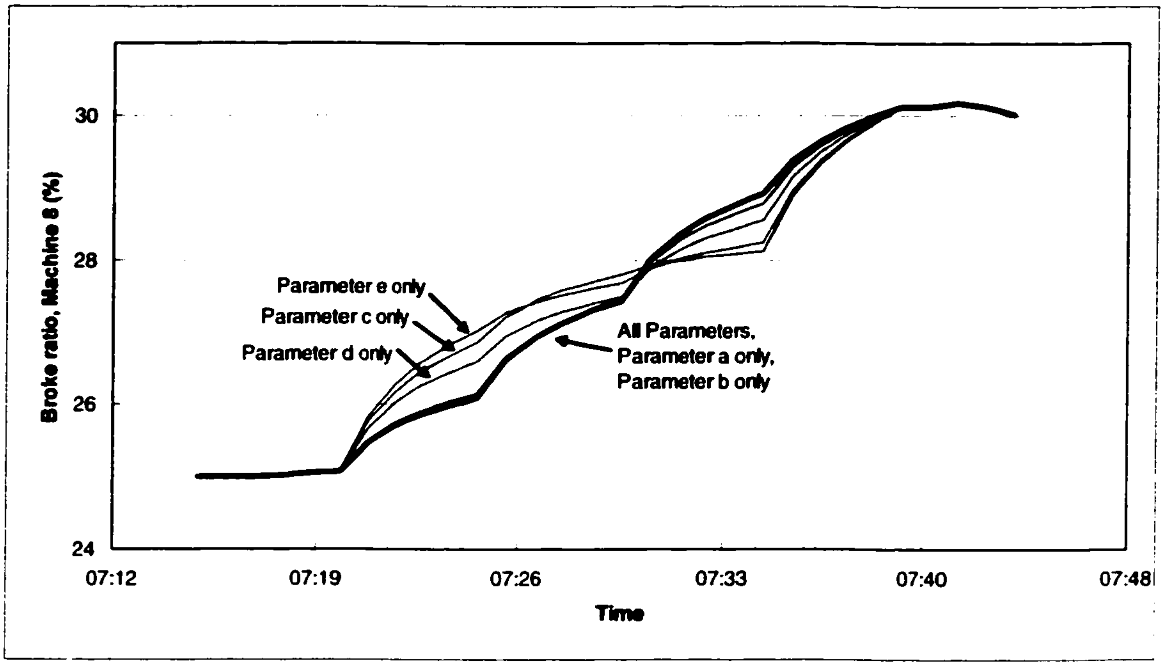


Figure 4.6: Optimized broke ratio change profiles using single terms in Objective Function #2.

The sizes of the simplex and genetic population, as well as the parameters used in both algorithms, were fixed for this study and represent another area of potential sensitivity study. It was shown that with a 5-member simplex and a 20-chromosome population, the algorithms settled near the optimal solution after ten iterations. Decreasing the population size could, however, reduce the run time for Genetic Algorithms without sacrificing the quality of the results. The accordance in results between the two algorithms validates that the solutions likely converged at the global minima rather than at local ones. The number of necessary iterations could, however, be decreased by optimizing the values of the algorithm parameters.

4.3.2 Implementation Issues

The results and implementation possibilities of this work were discussed with mill personnel towards the end of the project. Smoothing the profile of the broke ratio adjustments was intuitively guaranteed to result in more stable pulp conditions at the headbox. This study confirmed and quantified that hypothesis by providing the near-optimum magnitude of the four ratio changes under the assumptions specified in Section 4.1.2.

It is interesting to note that the solutions are fairly similar for all four machines. This suggests that a general profile could be implemented for broke ratio changes regardless of the machine for which the adjustment was made. Manual implementation of the change profile seems infeasible considering the 20-minute duration due to the operator attention that would be required. It was, however, confirmed with mill personnel that a change profile could be programmed into the DCS.

An implementation analysis is necessary to test the solution of the case study in the real process. Evaluating the outcome for the actual process would not be easy, as measuring pulp properties at the headbox in real-time is far from obvious. Currently, consistency is the only parameter measured at the headbox and that with questionable accuracy and precision. Consistency is a difficult property to measure in real-time, especially considering the elevated flow rates of the mixed pulp at the headbox. Thus, the measurement serves more as a general indicator of the magnitude of the consistency, rather than precise dynamic data.

A more conclusive and practical way of evaluating the advantage of optimizing the profile of broke ratio changes would be to observe whether the average number of paper machine breaks diminishes as a result of more stable machine furnish properties. However, drawing a correlation between headbox conditions and machine breaks is a difficult task, as determining the cause of a break is often impossible. As it was documented in Table 4.1, six breaks happened for unknown reasons. Of these six events, at least two could be suspected of having been caused by the sudden changes in the broke ratio:

1. Wet-end break on Machine 6 at 6:40 – broke ratio had been changed on Machines 6, 7 and 8 quarter of an hour prior at 6:25;
2. Dry-end break on Machine 9 at 16:45 – broke ratio had been changed on Machine 8 half an hour prior at 16:15.

It can be safely hypothesized, that smaller fluctuations in the pulp properties at the headbox contributes to machines stability and has the potential of reducing the frequency of breaks. This hypothesis could be verified statistically at the mill following the implementation of the smoother broke ratio change profiles if all other factors that could contribute to a break frequency reduction were accounted for.

Large solution times seen in the case of Genetic Algorithms may discourage the use of this method for further studies at the mill. The execution time could be shortened by using a smaller population or modifying the code so it does not re-run the simulation for chromosomes that are conserved from one iteration to the next. However, since the case study showed that very similar results can be obtained with the much faster Nelder-Mead Simplex method, it may be more practical to use that algorithm for implementation purposes.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

In today's competitive industry, paper manufacturers are increasingly focusing on process analysis and improvement technologies. Model-based optimization is a particularly attractive technique as it avoids plant experimentation and offers increased speed of testing (Fraleigh, 1999; Hall and Bowden, 1997). Direct search methods are gaining popularity for these applications, as they are structurally simple and do not require an analytical process model.

This study examined the broke management strategy at a newsprint mill and model-based direct search optimization to decrease its impact on paper machine headbox stability. The following sections summarize the results of the work and present some conclusions and recommendations.

5.1 Summary

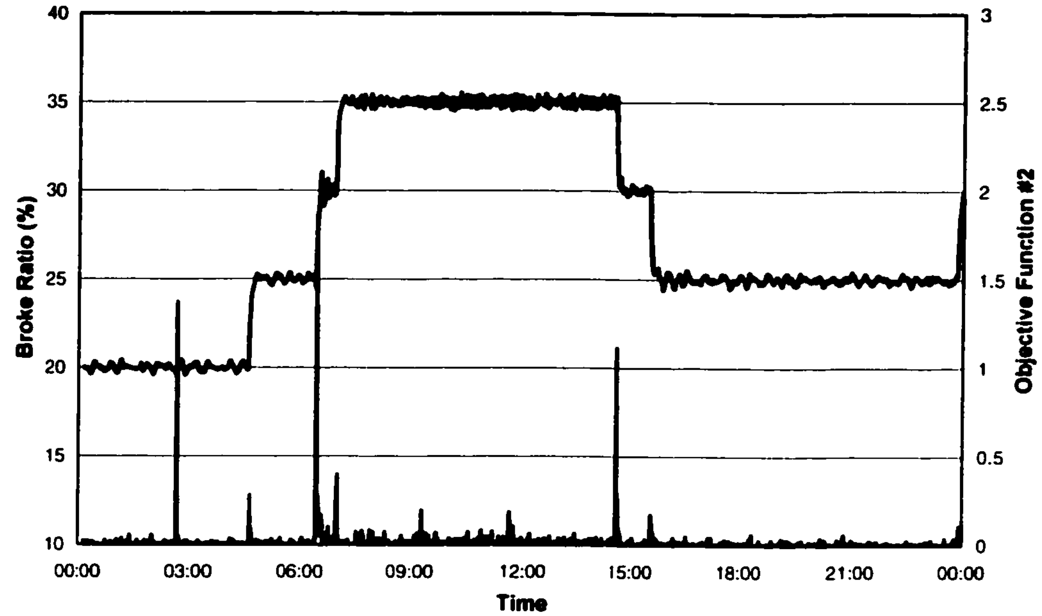
Bonhivers, *et al.* (2000) investigated broke recirculation strategies using model predictive control, and was able to reduce the variations in the intensive properties of the mixed pulp caused by broke recycling. This thesis has focused on the development of a dynamic simulation of the papermaking section of an integrated newsprint mill and applying direct search optimization methods to find improved broke recycling strategies. The profile of the broke ratio changes was manipulated in order to find one that has reduced impact on the properties of mixed pulp stream at the paper machine headbox.

Chapter 2 addressed the issue of dynamically modeling the papermaking process. Process flow diagrams, historical and real-time data, as well as input from mill personnel were used to ensure the general validity of the simulation. The simulation was linked through a dynamic exchange interface with the simulation worksheet. The worksheet contained the input and output data of the simulation, as well as the algorithm codes for the Nelder-Mead Simplex and Genetic Algorithms. An objective function was developed to reflect pulp property fluctuations at the headboxes of the paper machines following operational changes such as broke ratio adjustments. The theory of the search algorithms, the problem statement, objective function and solution approach were all described in Chapter 3.

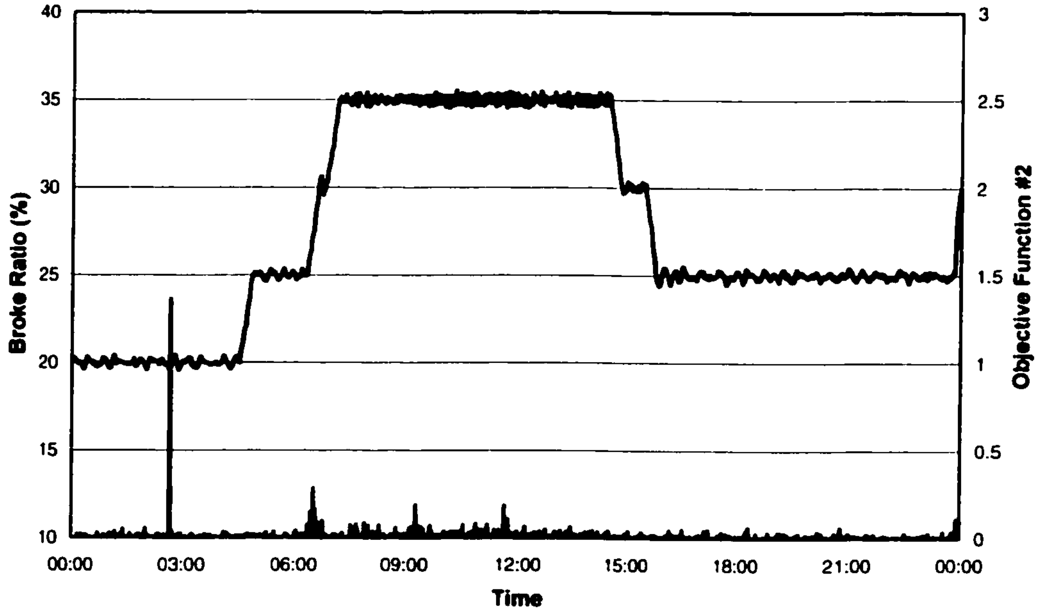
Chapter 4 was dedicated to the case study treated in this work. The study provides a proof-of-concept insight into the possibility of optimization in the broke management strategies. Based on historical data, process operation was simulated for a 24 hour period. Focus was placed on observing and quantifying the fluctuations in pulp properties at the headbox following the adjustments to the broke ratio that were done by the process operators. It was hypothesized that these fluctuations could be significantly decreased if the adjustments had been made more gradually. The two direct search algorithms were used under the set of assumptions described in Section 4.1 to find the near-optimal change profiles. The performance of the optimized operation was gauged using the objective function and compared to the results obtained after simulating the original case.

5.2 Conclusions

This work is the first step towards a complete optimization of the broke system management. The results of the case study in Chapter 4 show that changing the broke ratio more gradually and over a longer period of time considerably reduces variations in pulp properties at the headbox. Figure 5.1 shows a comparison of the objective function response for Paper Machine 6 between the original operation and the case optimized by the Nelder-Mead simplex. The spikes in the objective function values are clearly smaller in the optimized case, confirming that smoothening out the broke ratio adjustment profile significantly dampens headbox variability.



Original Operation, Machine 6



Optimized Operation, Machine 6

Figure 5.1: Comparison of objective function responses between original operation and the optimized case.

The dynamic model developed and described in Chapter 2 is basic and can only serve as a tool for proof-of-concept studies, such as this one. Much more work is needed to achieve appropriate levels of robustness and dynamic validity. The process is very complex and often problematic. Moreover, additional data is needed from the plant to improve the accuracy of the simulation. Ultimately, the quality of the model will be restricted by plant-model mismatch and by the limitations of sequential simulators like the one used in this study.

The two direct search algorithms that were used provided nearly the same final results for the near-optimum change profile. The results of the case study were very promising, but the experiment was limited in scope. The possibilities for sensitivity studies into the performance of the algorithms are considerable. Furthermore, the work only addressed one out of an extensive pool of possible solutions and optimization candidates. For example, analyzing the spike observed in the objective function response in Figure 5.1 just before 3:00 o'clock (when no changes were made to the broke ratio) revealed that the disturbance at the headbox was actually caused by an abrupt change in the TMP ratio.

Besides the numerous possible extensions of this work presented in Section 4.3, a study into the potential of implementing the new broke recycling strategy is needed. Though difficult to quantify, the frequency of paper breaks could most certainly be decreased with more stable conditions at the headbox.

5.3 Recommendations and Potential for Implementation

The results of the work offer a new approach to broke recycle strategies at the mill and offer an insight into the possible benefits that could be derived from its implementation. The major recommendation stemming from this work involves revising the current method of changing pulp proportions (not only broke, but also TMP and DIP). The adjustment profiles found by the direct search algorithms could be implemented into the control computers at the plant. Then, an implementation study would be needed to verify whether the headbox variations can actually be decreased in the real process. Ultimately, a decrease in break frequency is expected.

Lastly, it is recommended that the entire broke management policy be examined for potential improvements. The dynamic model developed in Chapter 2 could serve as a tool to attempt more efficient ways of recycling broke pulp, for example, an automatic scheme of managing broke ratios based on broke and whitewater inventories as well as machine operation.

BIBLIOGRAPHY

BAGÜL, Z., SEKMEN, A.Ş., PALANIAPPAN, S. and ZEIN-SABATTO, S. (2000). Genetic algorithms applied to real time multiobjective optimization problems. Proc. IEEE SOUTHEASTCON, Nashville, TN, USA, 95-103.

BONHIVERS, J.C. (1999). Simulation dynamique d'un atelier de fabrication de papier journal et commande prédictive appliquée à la gestion des cassés et des réservoirs d'eaux blanches. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

BONHIVERS, J.C., PERRIER, M. and PARIS, J. (2000). Management of broke recirculation in an integrated paper mill. Control Systems 2000, Victoria, Canada, 359-367.

CHAMBERS, L. (1995). Practical handbook of genetic algorithms. Boca Raton, Florida, USA.

CROTEAU, A.P. and ROCHE, A.A. (1987). Study of broke handling and whitewater management using a dynamic simulation. Pulp & Paper Canada, 88(11), T420-T423.

FRALEIGH, L.M. (1999). Optimal Sensor Selection and Parameter Estimation for Real-time Optimization. Master's thesis, University of Alberta, Edmonton, Canada.

GOLDBERG, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learining. Addison Wesley Longman, Inc.

HALL, J.D. and BOWDEN, R.O. (1997). Simulation optimization by direct search – a comparative study. Proc. 6th Annual Industrial Engineering Research Conference, IERC, Miami Beach, FL, USA, 298-303.

HUMPHREY, D.G. and WILSON J.R. (1998). Revised Simplex Search procedure for stochastic simulation response-surface optimization. Proc, Winter Simulation Conference, IEEE, Washington, DC, USA, Part 1 (of 2), 751-759.

LIUQUET, G. (1995). Production stability on paper machines and pulp and whitewater circuits. Tappi Journal, 78(10), 256-258.

MONDER, D.S. (2000). Real-Time Optimization of Gas Blending with Uncertain Parameters. Master's thesis, University of Alberta, Edmonton, Canada.

NELDER, J.A. and MEAD R. (1965). A simplex method for function minimization. Computer Journal, 7 308-313.

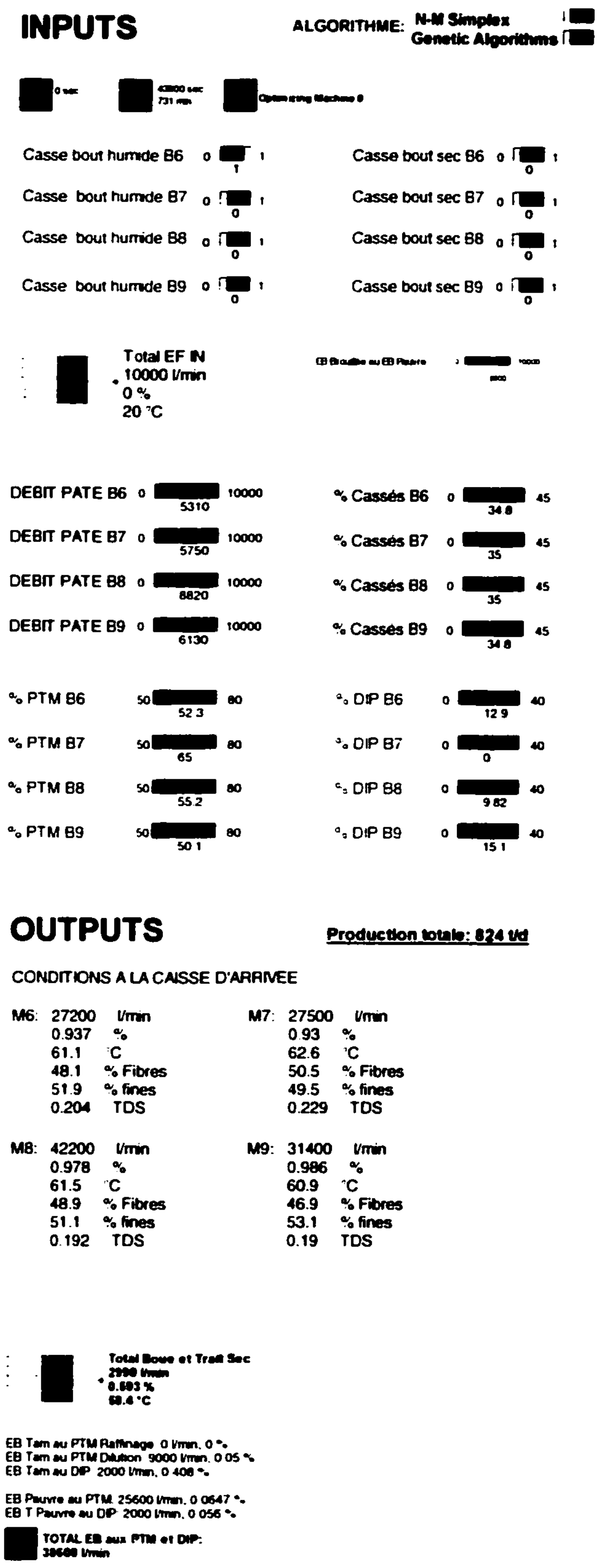
ORCCOTOMA, J.A., PARIS, J., PERRIER, M. and ROCHE, A.A. (1997). Dynamics of whitewater networks during web breaks. Tappi Journal, 80(12), 101-110.

WRIGHT, M.H. (1998). Optimization methods for base station placement in wireless applications. Proc. 48th IEEE Vehicular Technology Conference, Ottawa, Canada, Part 1 (of 3), 387-391.

XIONG Q. and JUTAN A. (2000). Continuous Optimization using a Dynamic Simplex Method. Submitted to Chemical Engineering Science.

APPENDICES

APPENDIX I: SIMULATION



INPUTS

ALGORITHME: N-M Simplex
Genetic Algorithms

0 sec
43800 sec
731 min
Optimizing Machine 8

Casse bout humide B6 0 1 — 1
Casse bout humide B7 0 1 — 0
Casse bout humide B8 0 1 — 0
Casse bout humide B9 0 1 — 0

Casse bout sec B6 0 1 — 0
Casse bout sec B7 0 1 — 0
Casse bout sec B8 0 1 — 0
Casse bout sec B9 0 1 — 0

Total EF IN
10000 l/min
0 %
20 °C

EB Brouillée au EB Pauvre 0 10000

DEBIT PATE B6 0 10000 — 5310
DEBIT PATE B7 0 10000 — 5750
DEBIT PATE B8 0 10000 — 8820
DEBIT PATE B9 0 10000 — 6130

% Cassés B6 0 45 — 34.8
% Cassés B7 0 45 — 35
% Cassés B8 0 45 — 35
% Cassés B9 0 45 — 34.8

% PTM B6 50 80 — 52.3
% PTM B7 50 80 — 65
% PTM B8 50 80 — 55.2
% PTM B9 50 80 — 50.1

% DIP B6 0 40 — 12.9
% DIP B7 0 40 — 0
% DIP B8 0 40 — 9.82
% DIP B9 0 40 — 15.1

OUTPUTS

Production totale: 824 t/d

CONDITIONS A LA CAISSE D'ARRIVEE

M6: 27200 l/min
0.937 %
61.1 °C
48.1 % Fibres
51.9 % fines
0.204 TDS

M7: 27500 l/min
0.93 %
62.6 °C
50.5 % Fibres
49.5 % fines
0.229 TDS

M8: 42200 l/min
0.978 %
61.5 °C
48.9 % Fibres
51.1 % fines
0.192 TDS

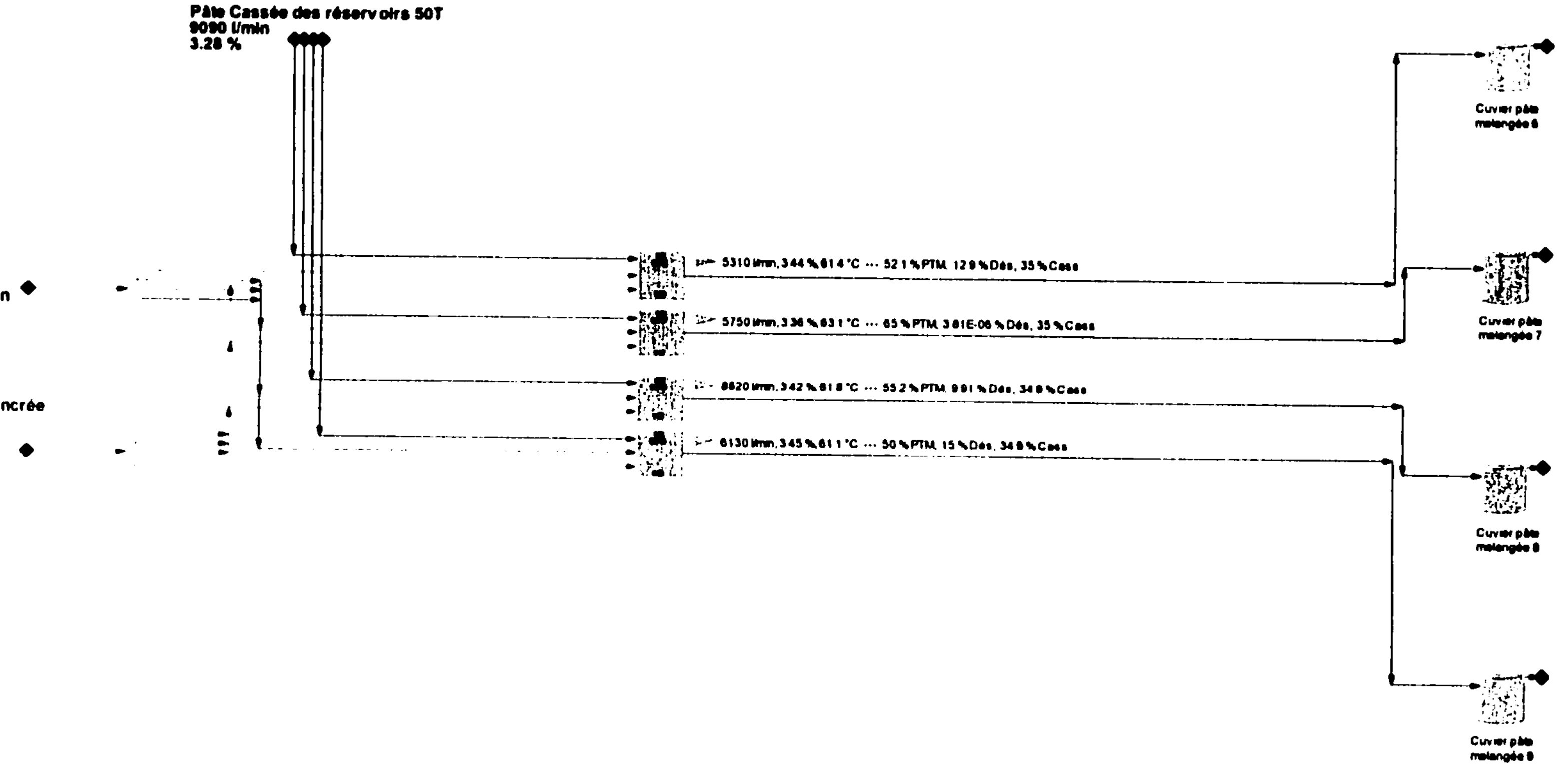
M9: 31400 l/min
0.986 %
60.9 °C
46.9 % Fibres
53.1 % fines
0.19 TDS

Total Boue et Trait Sec
2990 l/min
0.893 %
60.4 °C

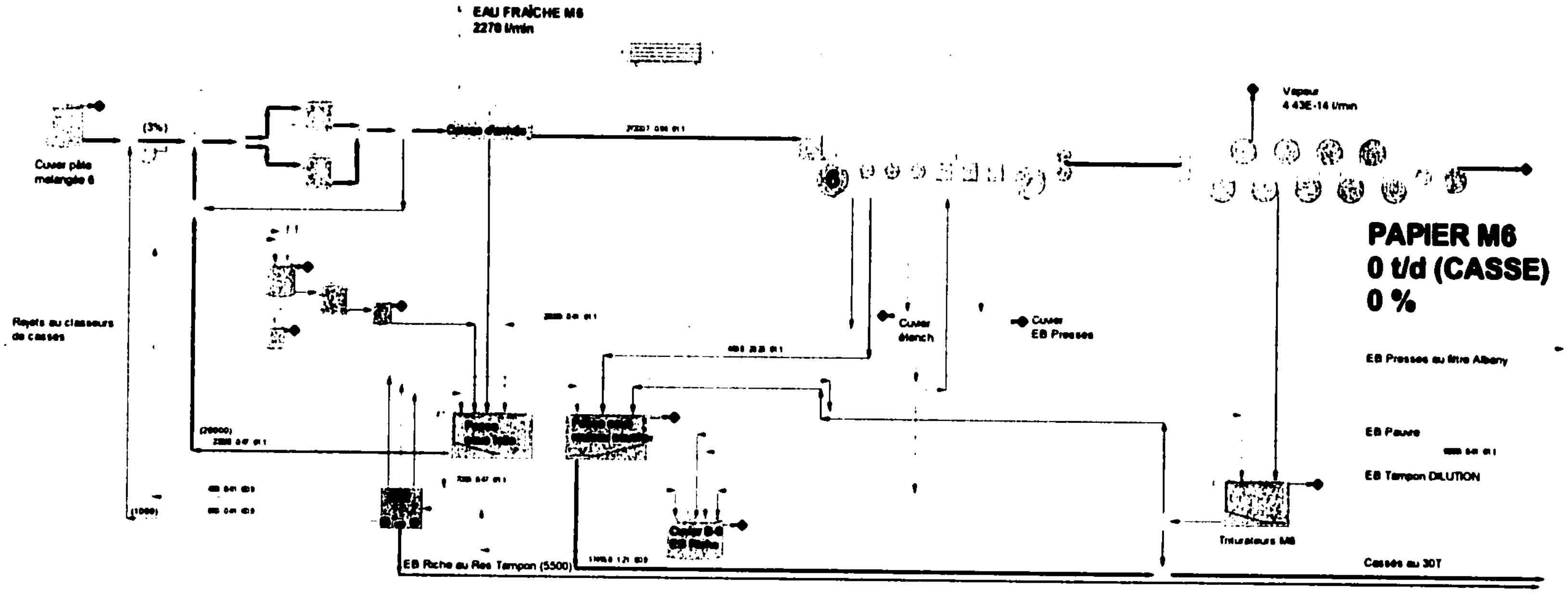
EB Tam au PTM Raffinage 0 l/min, 0 %
EB Tam au PTM Dilution 9000 l/min, 0.05 %
EB Tam au DIP 2000 l/min, 0.408 %

EB Pauvre au PTM 25600 l/min, 0.0647 %
EB T Pauvre au DIP 2000 l/min, 0.056 %

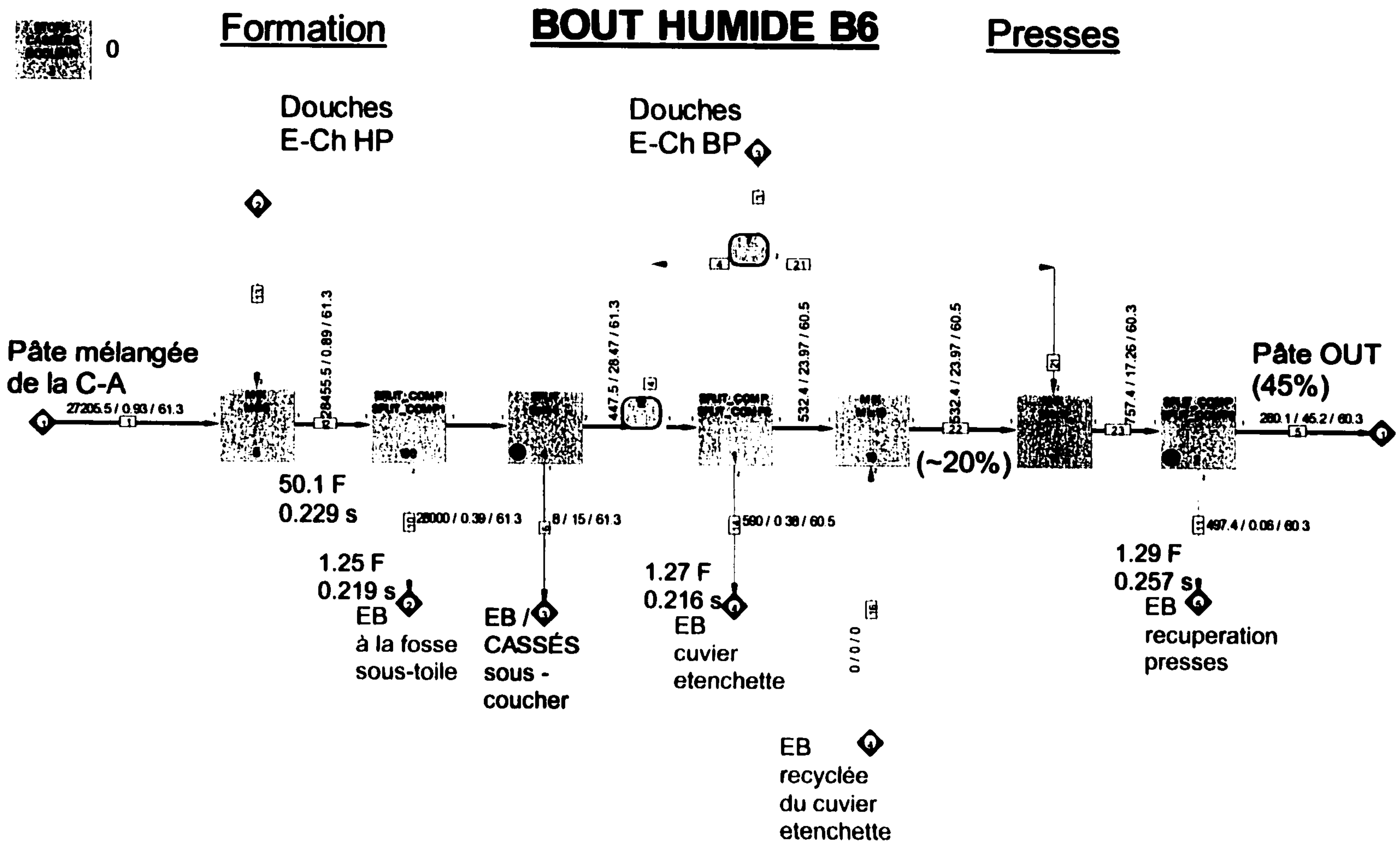
TOTAL EB aux PTM et DIP:
38600 l/min

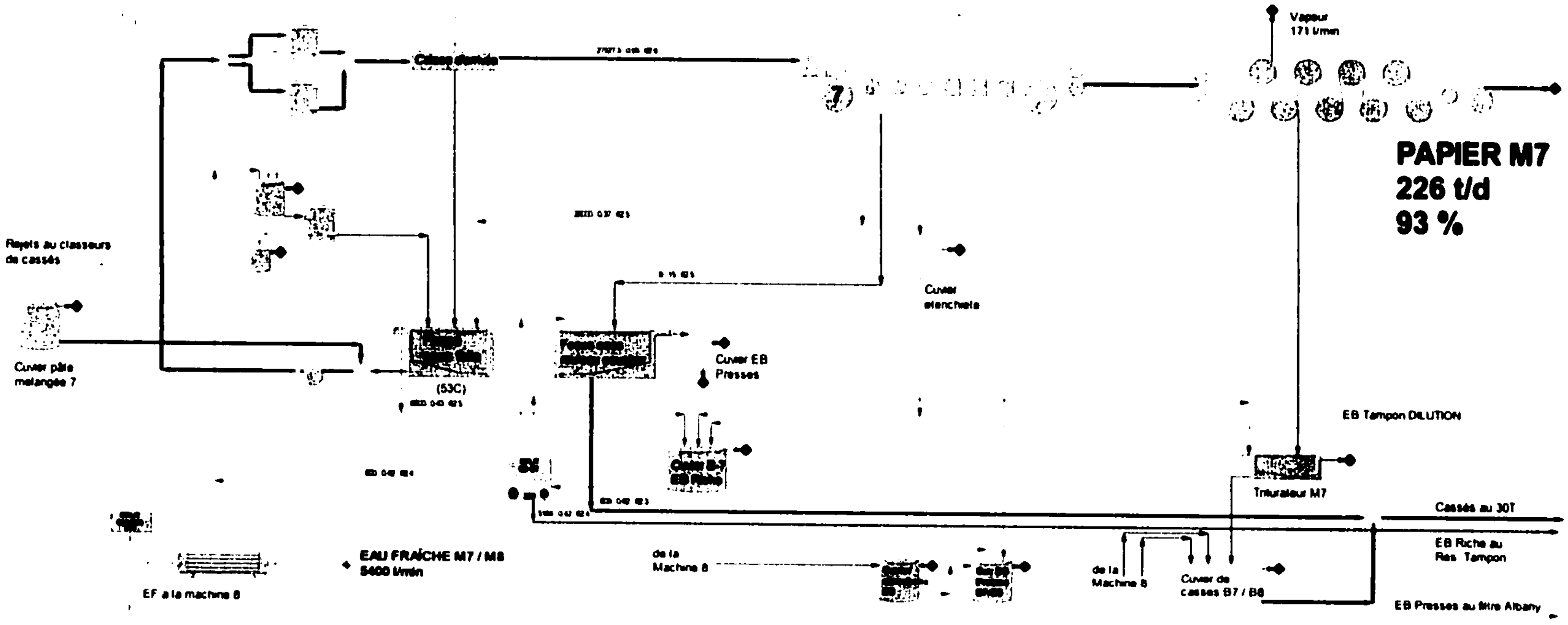


Mélange des pâtes



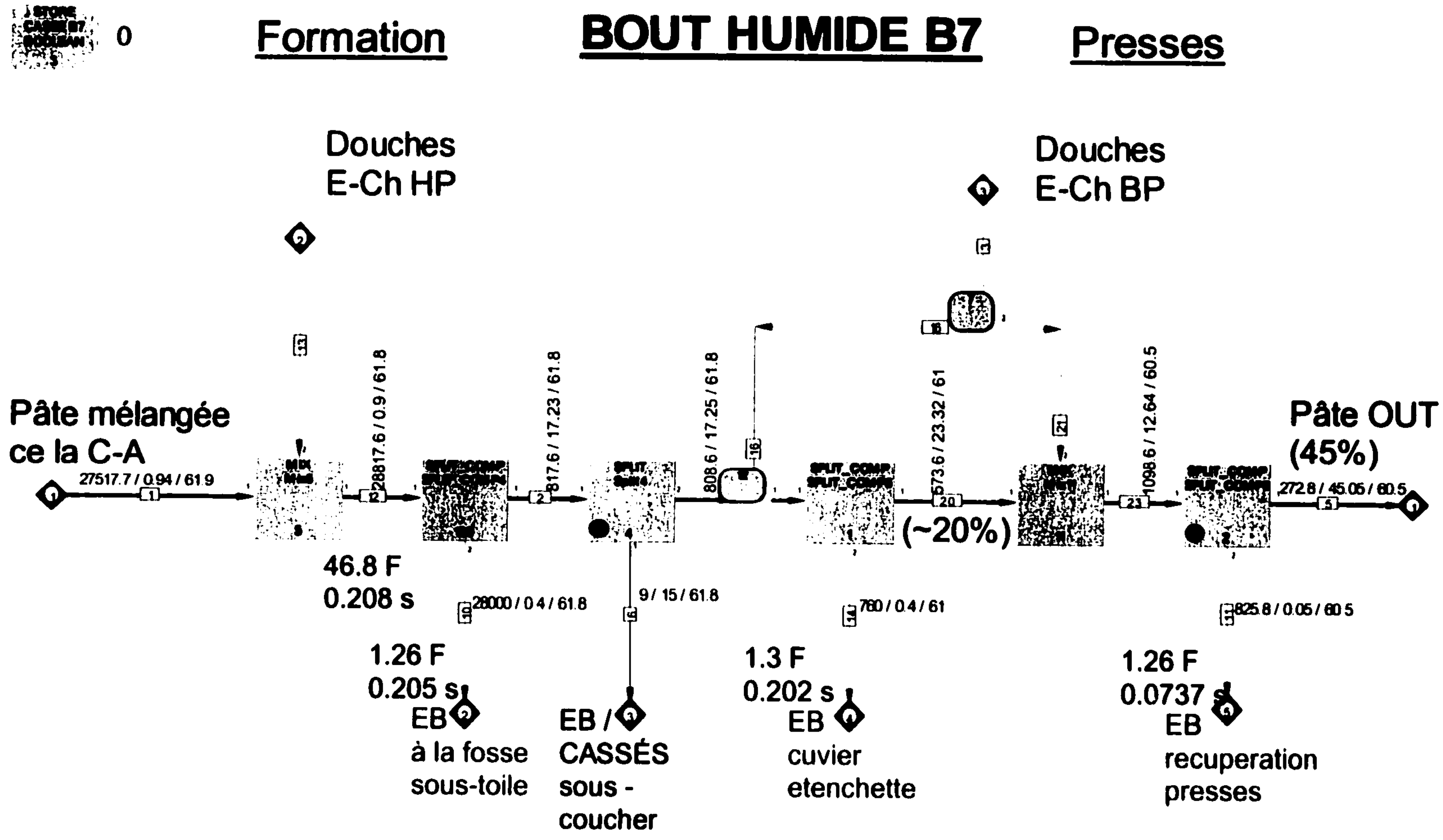
Machine à papier 6

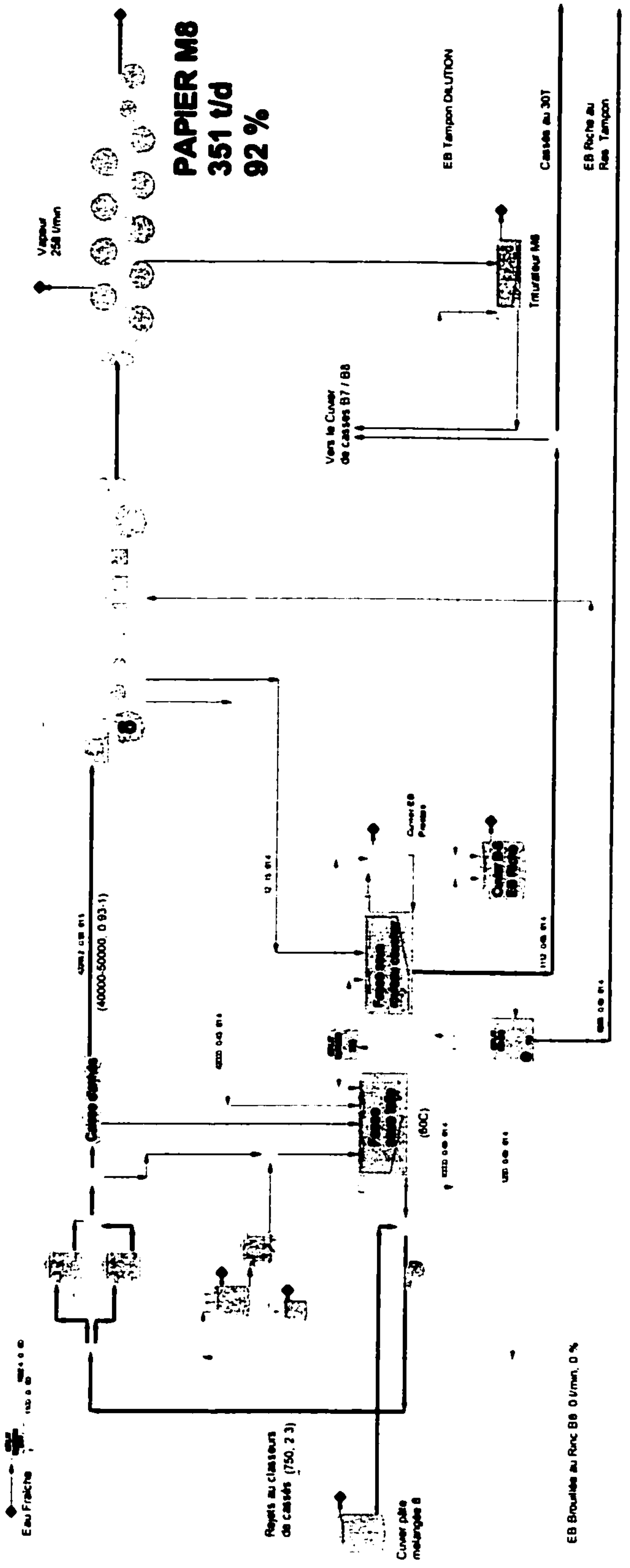

0
Formation
BOUT HUMIDE B6
Presses
Douches
E-Ch HP
Douches
E-Ch BP
Pâte mélangée
de la C-A
27205.5 / 0.93 / 61.3
28455.5 / 0.89 / 61.3
447.5 / 28.47 / 61.3
532.4 / 23.97 / 60.5
532.4 / 23.97 / 60.5
757.4 / 17.26 / 60.3
Pâte OUT
(45%)
280.1 / 45.2 / 60.3
50.1 F
0.229 s
28000 / 0.39 / 61.3
8 / 15 / 61.3
590 / 0.38 / 60.5
(~20%)
497.4 / 0.06 / 60.3
1.25 F
0.219 s
EB
à la fosse
sous-toile
EB /
CASSÉS
sous -
coucher
1.27 F
0.216 s
EB
cuvier
etenchette
0 / 0 / 0
1.29 F
0.257 s
EB
recuperation
presses
EB
recyclée
du cuvier
etenchette



Machine à papier 7

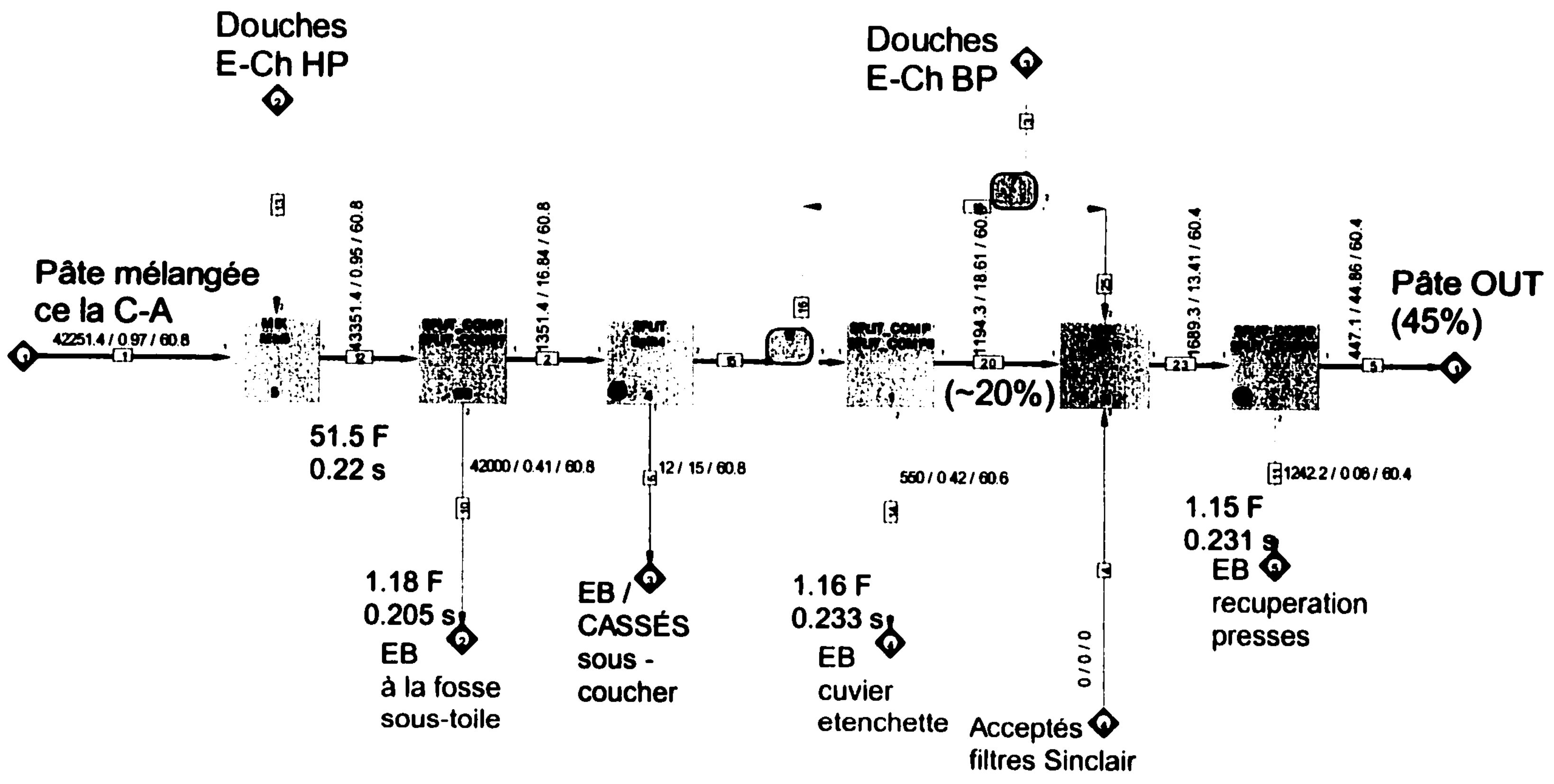
BOUT HUMIDE B7





Machine à papier 8

0 **Formation** **BOUT HUMIDE B8** **Presses**



Douches
E-Ch HP

Douches
E-Ch BP

Pâte mélangée
ce la C-A

51.5 F
0.22 s

1.18 F
0.205 s
EB
à la fosse
sous-toile

EB /
CASSÉS
sous -
coucher

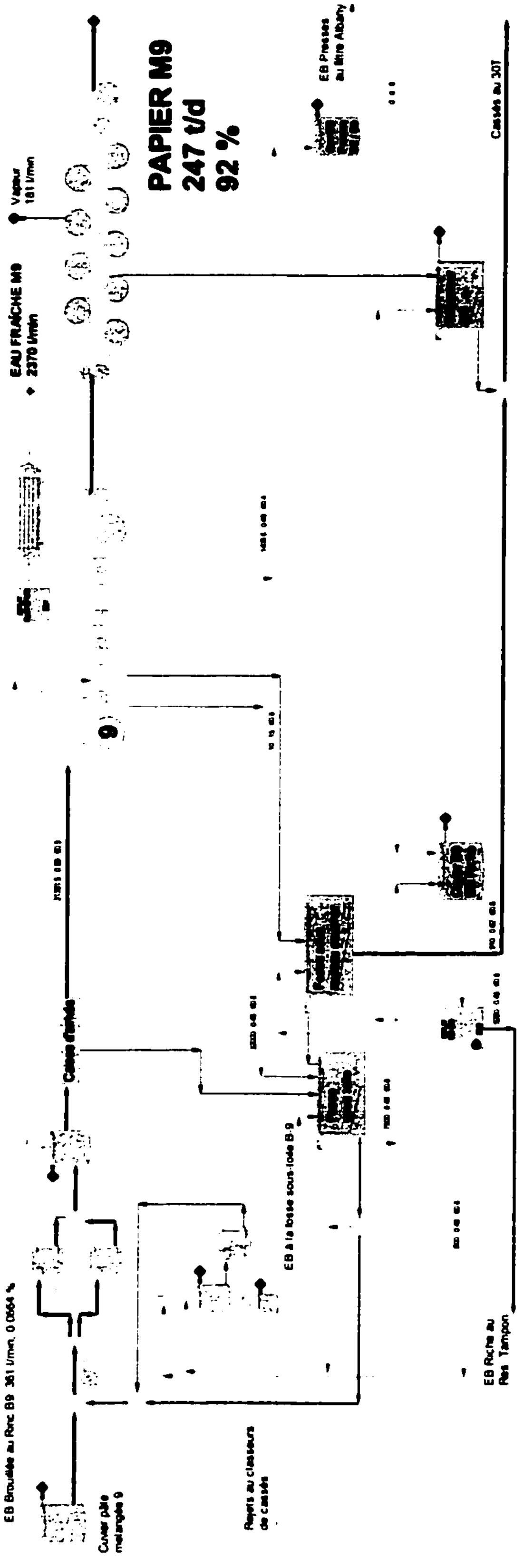
1.16 F
0.233 s
EB
cuvier
etenchette

(~20%)

Acceptés
filtres Sinclair

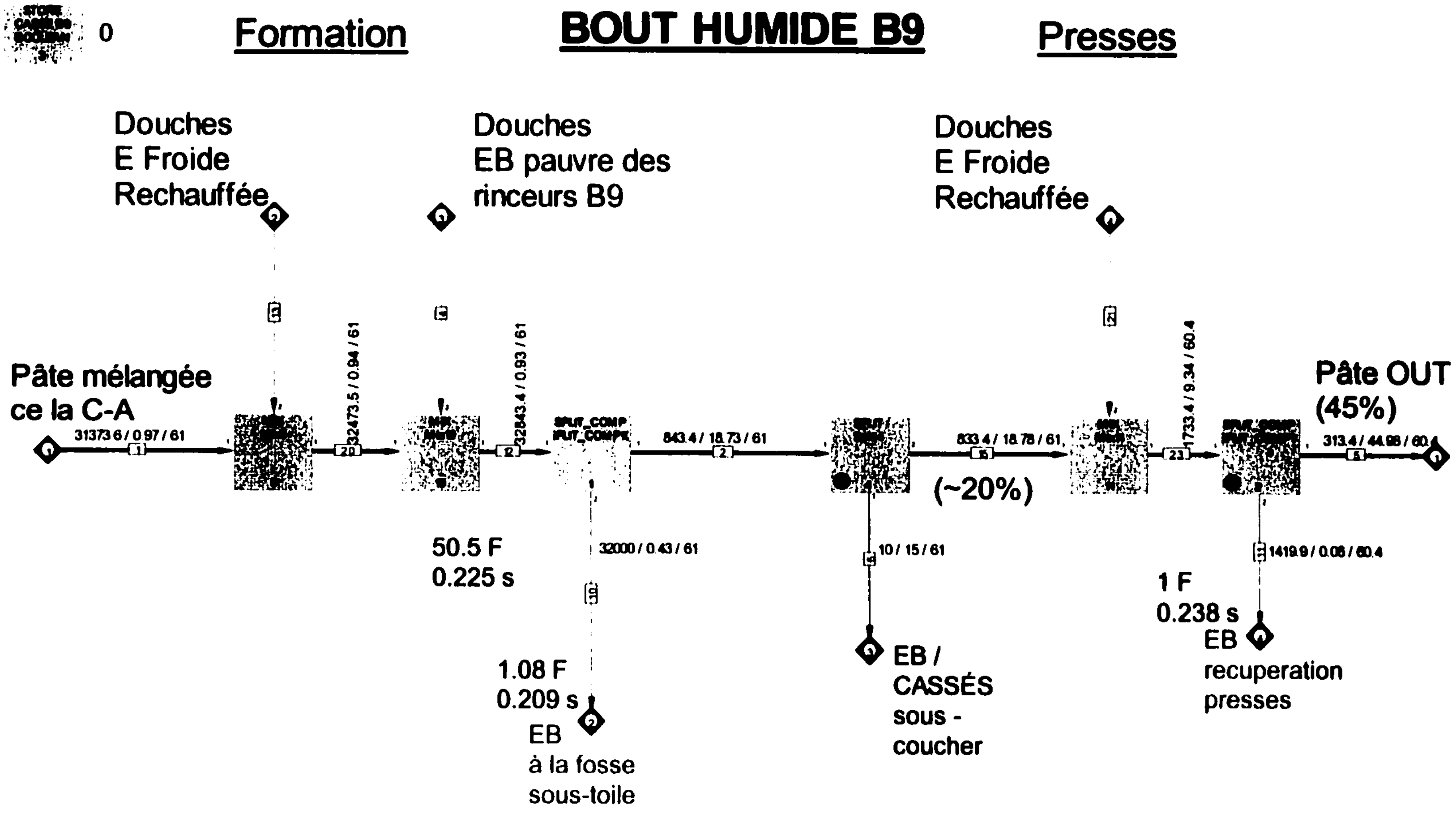
1.15 F
0.231 s
EB
recuperation
presses

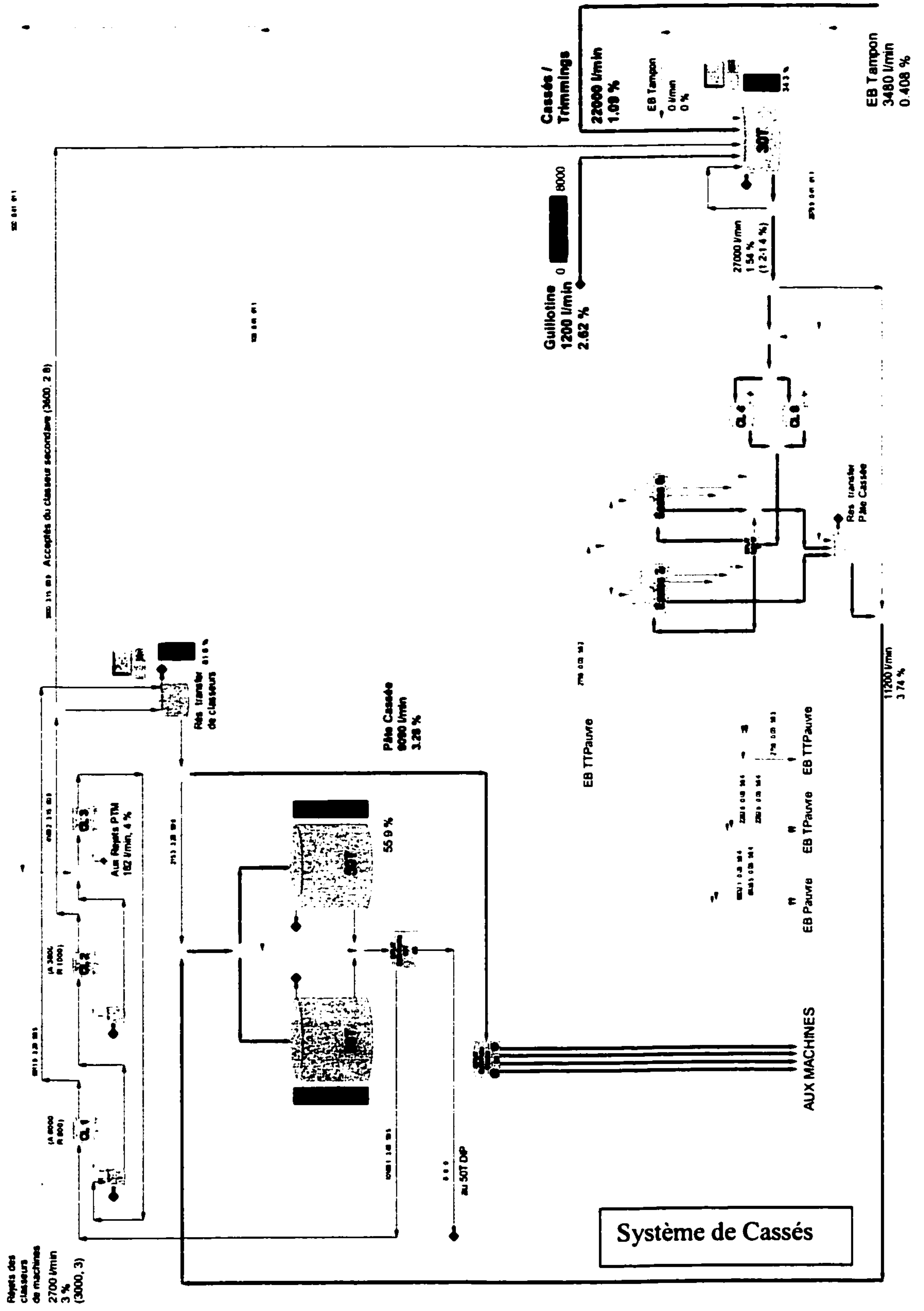
Pâte OUT
(45%)



Machine à papier 9

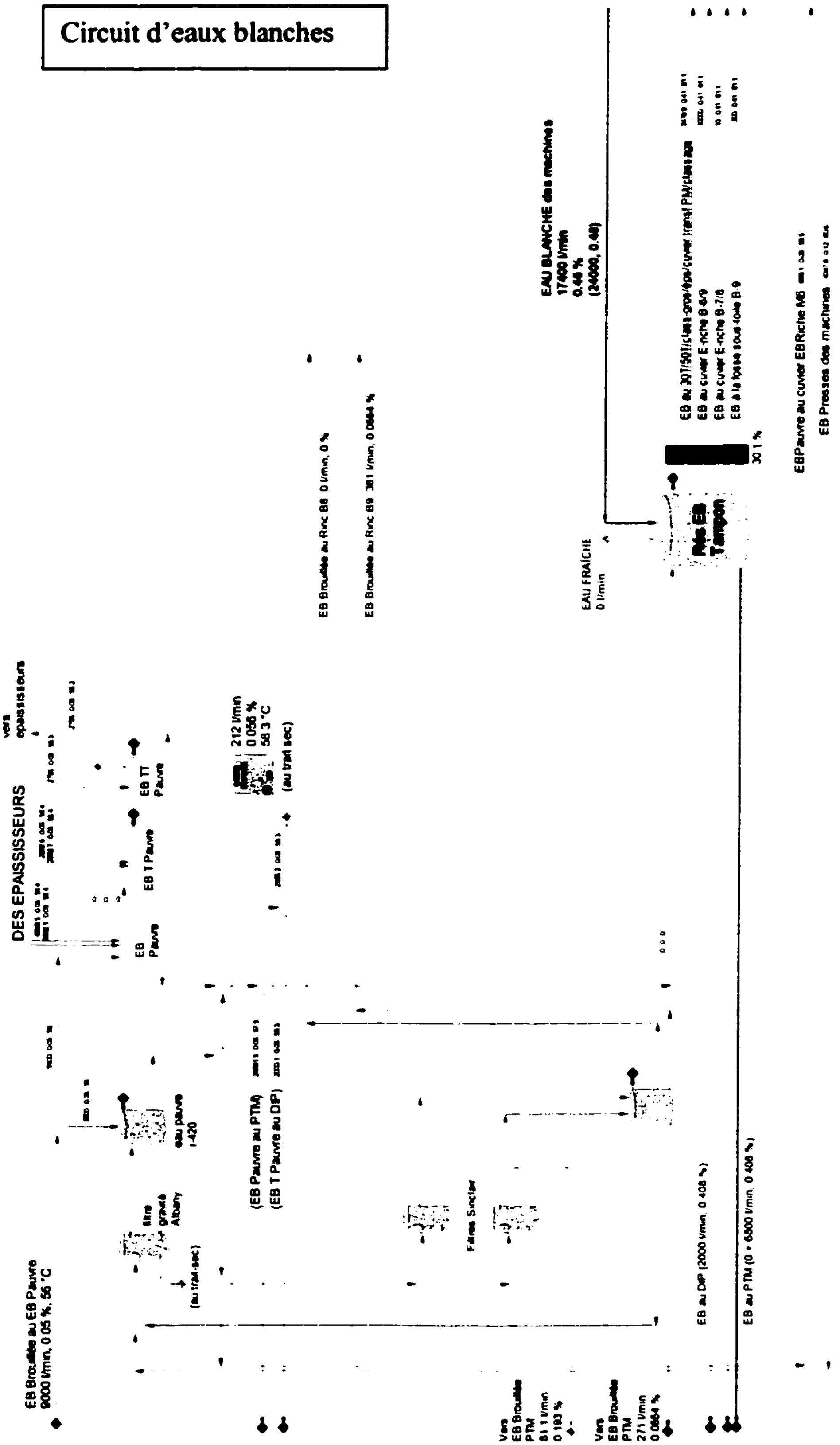
BOUT HUMIDE B9





Système de Cassés

Circuit d'eaux blanches



APPENDIX II: DATA

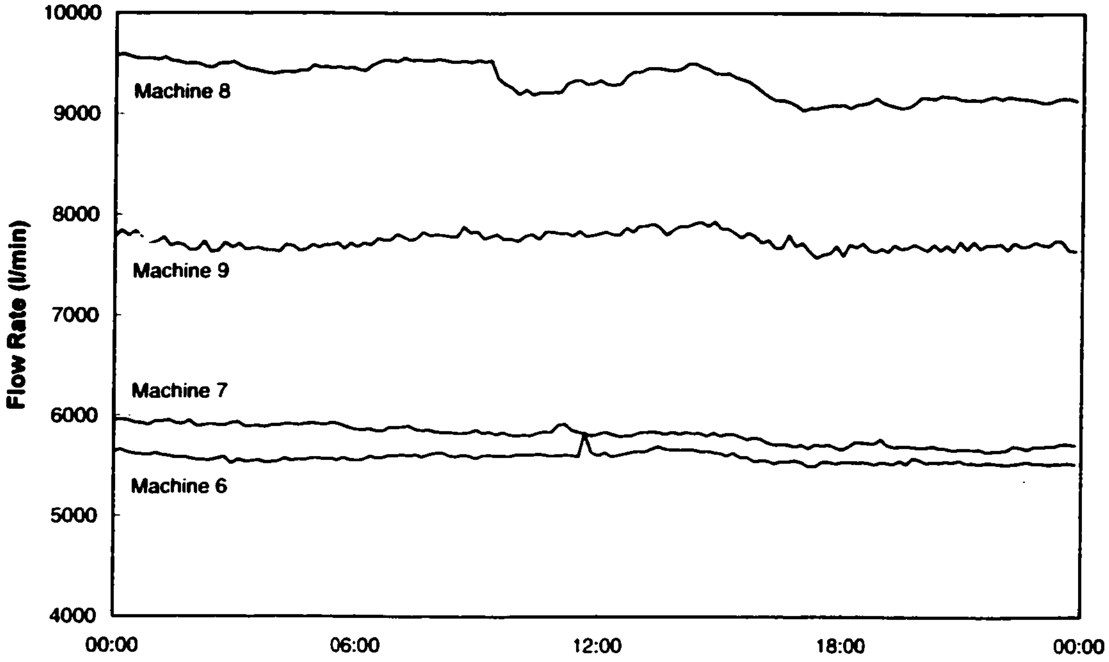
GENERAL OVERALL STEADY-STATE BALANCES FOR NORMAL OPERATION

INPUTS	Flow Rate (l/min)	Consistency (%)	**Flow Rate (T/d)**
TMP	16900	3.4	24000
DIP	4500	4.0	6500
Fresh Water	10000	0	14400
Cloudy WW from TMP	9000	0.05	12800
Repulped Paper Rejects	1200	2.62	1740
TOTAL IN			**59440**

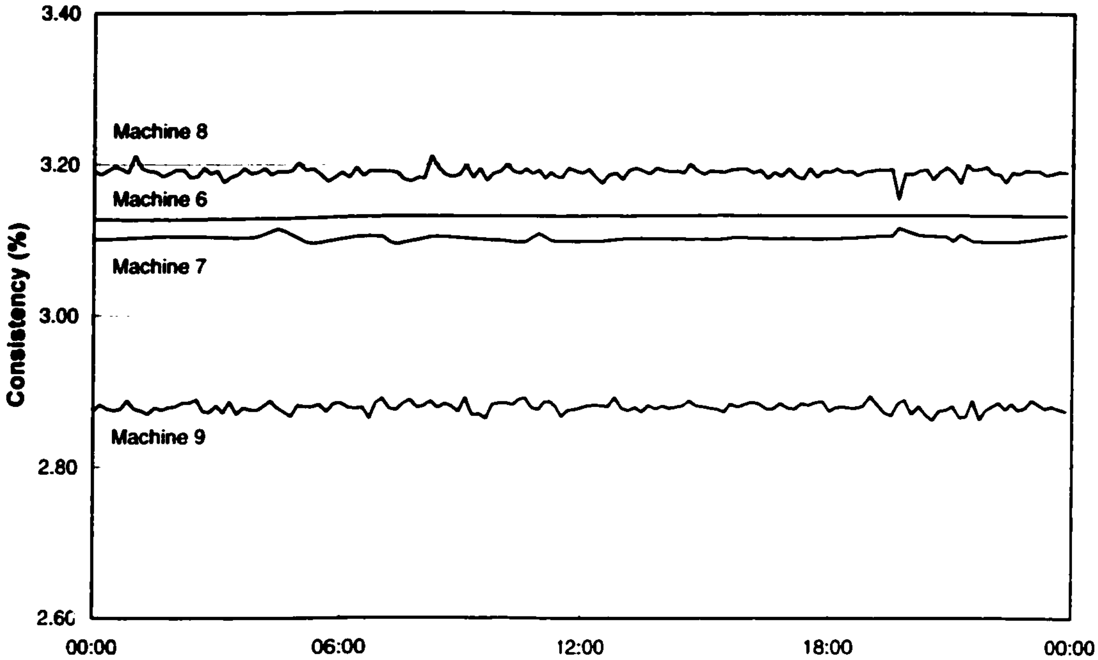
OUTPUTS	Flow Rate (l/min)	Consistency (%)	**Flow Rate (T/d)**
Paper B6		92	210
Paper B7		93	240
Paper B8		92	360
Paper B9		92	255
Steam (approx. acc. to simulation)		0	1100
WW to TMP	18600	0.43	26500
WW to DIP	2000	0.43	2850
Lean WW to TMP	9200	0.07	13000
Very-lean WW to DIP	2300	0.06	3200
Lean WW to Secondary Treatment	8000	0.06	11400
Other Streams to Secondary Treatment and Pulp Shops (approx.)	250	0.18 to 4	400
TOTAL OUT			**59515**

ADDITIONAL DATA COLLECTED

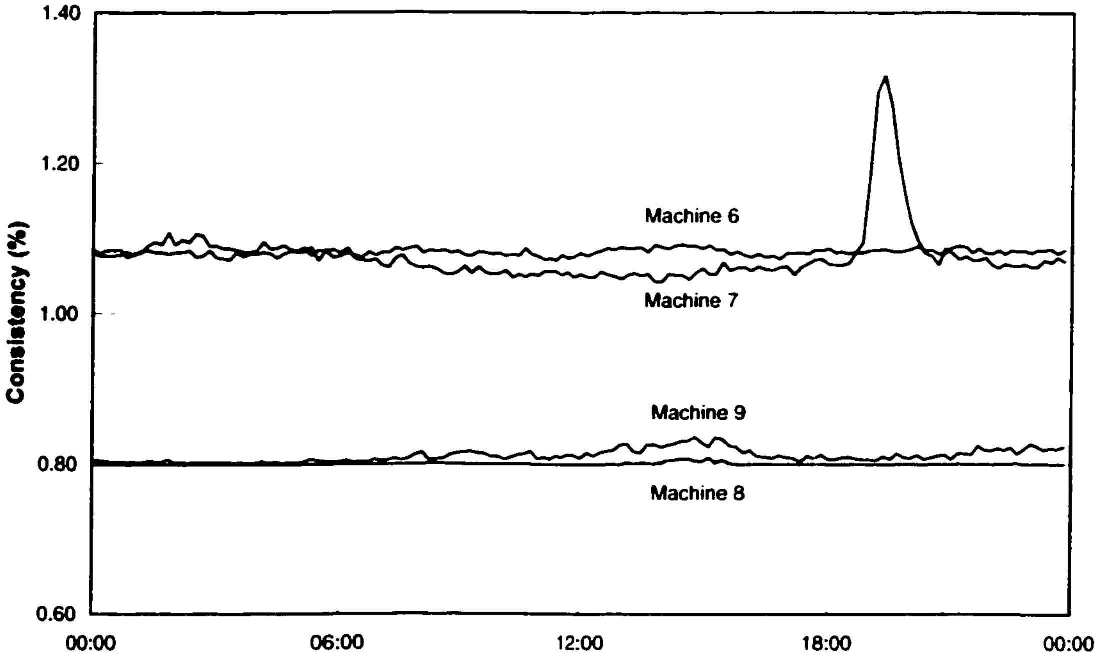
Flow Rate of Mixed Pulp Streams to Machines



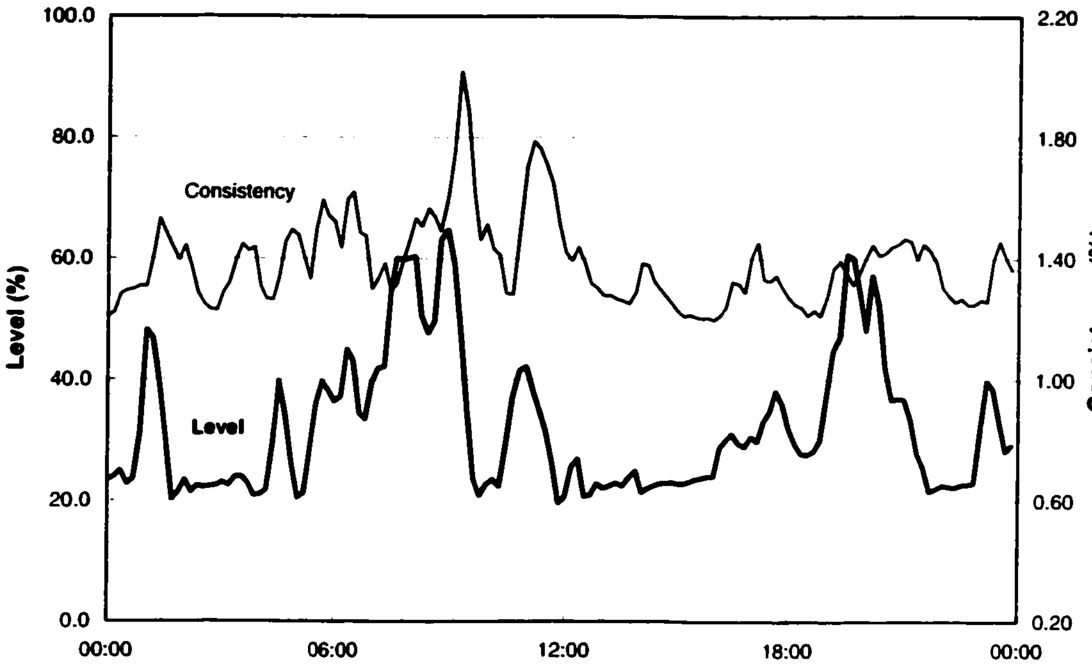
Consistency of Mixed Pulp Streams to Machines



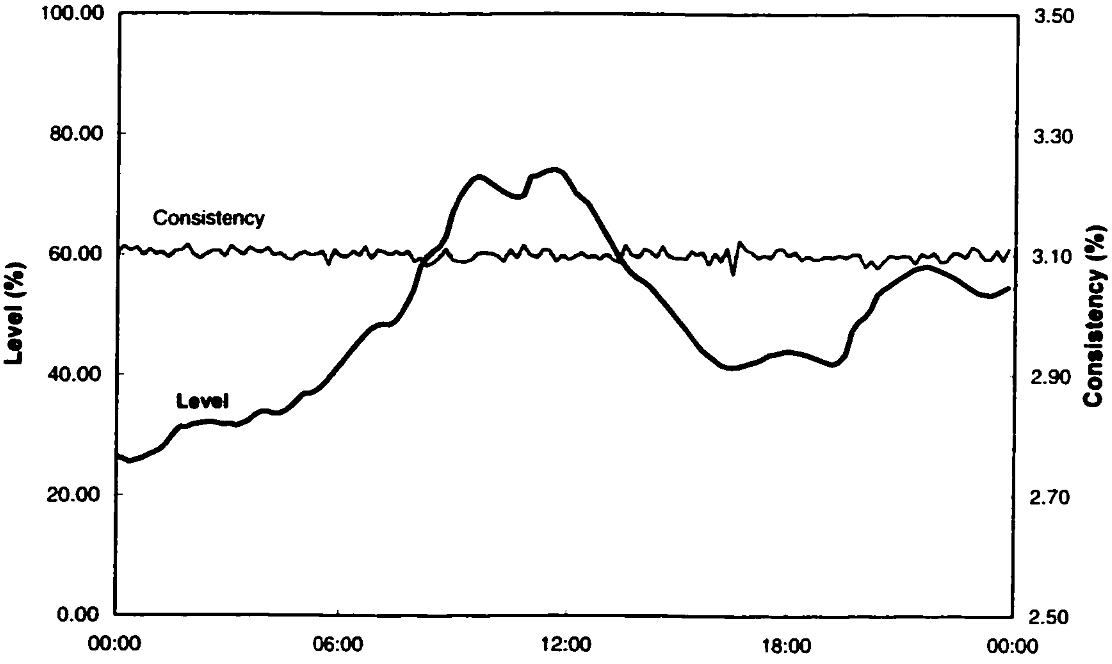
Consistency of Pulp Streams at Headbox

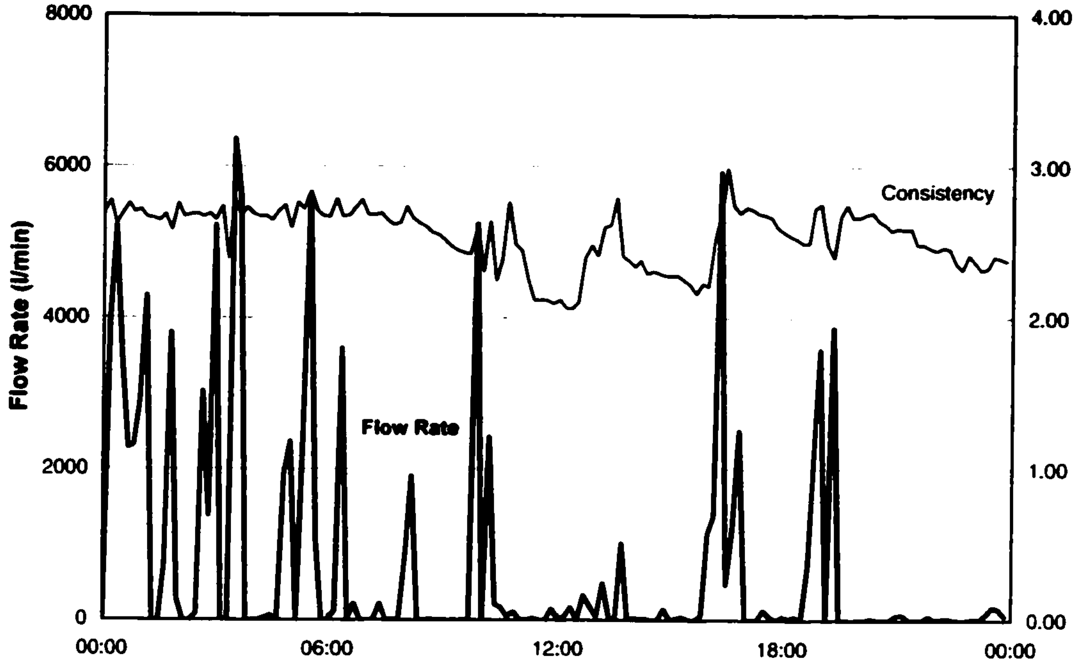


Level and Consistency of 30T Broke Transfer Tank

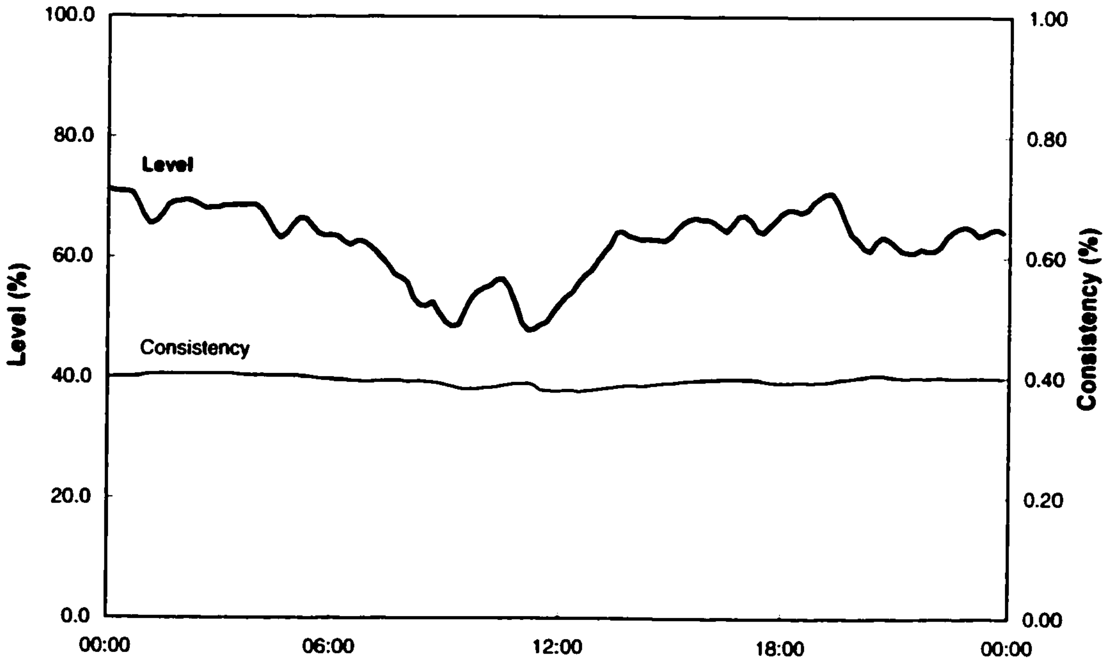


Level and Consistency of 50T Broke Storage Tank



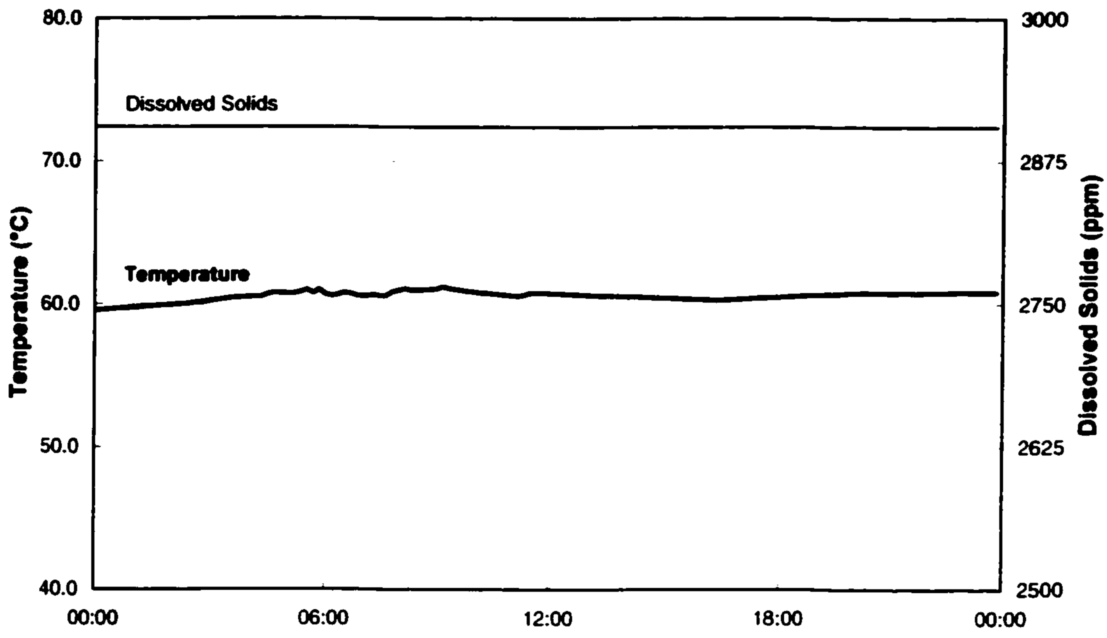


Flow Rate and Consistency of Rejected Paper Rolls

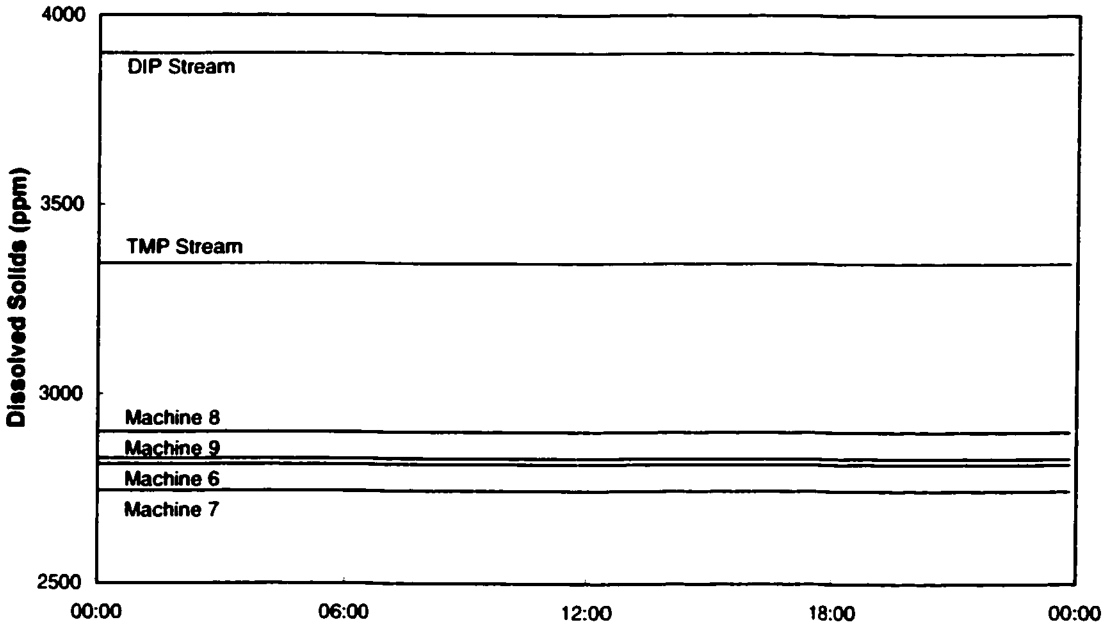


Level and Consistency of Whitewater Buffer Tank

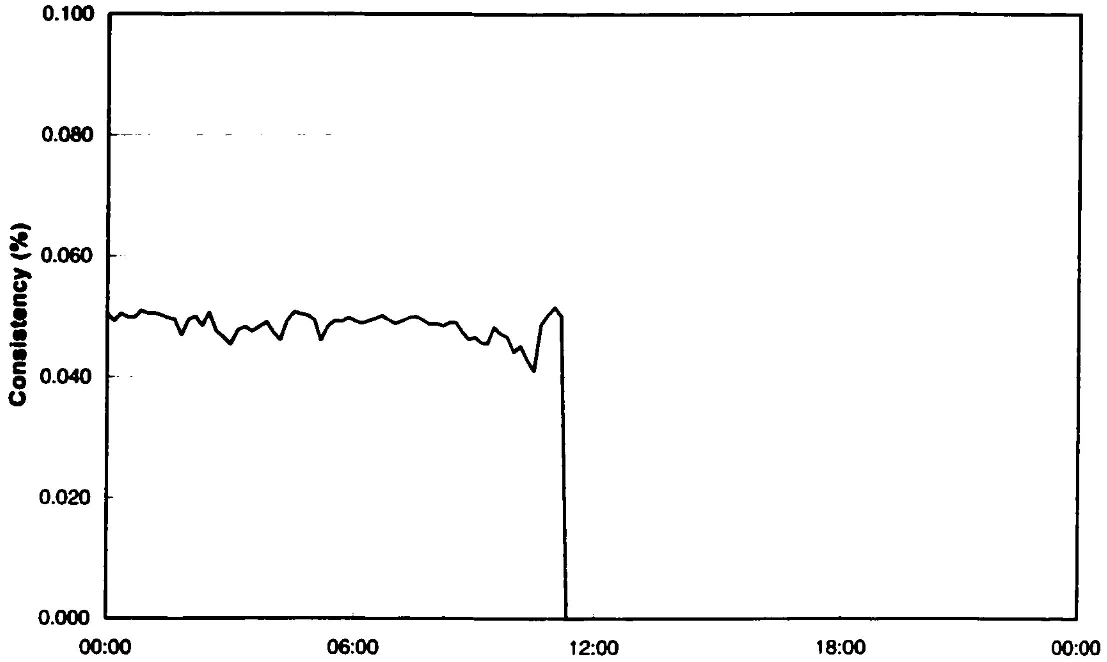
Temperature and Dissolved Solids Content in Whitewater Buffer Tank



Dissolved Solids Content in the Inlet Streams and at the Paper Machines



Consistency of Cloudy Whitewater from TMP Shop



Flow Rate of Lean Whitewater to Secondary Treatment

CONSTANT MODEL PARAMETERS

Flow rate of mixed pulp stream to Machine 6	5310 l/min
Flow rate of mixed pulp stream to Machine 7	5750 l/min
Flow rate of mixed pulp stream to Machine 8	8820 l/min
Flow rate of mixed pulp stream to Machine 9	6130 l/min

Consistency of TMP	3.4%
Temperature of TMP	65 °C
Fibre content in TMP	79%
Dissolved solids content in TMP	0.4%

Consistency of DIP	4.0%
Temperature of DIP	52 °C
Fibre content in DIP	79%
Dissolved solids content of DIP	0.4%

Flow rate of repulped rejected paper rolls	1200 l/min
Consistency of repulped rejected paper rolls	2.62%
Temperature of repulped rejected paper rolls	20 °C
Fibre content in repulped rejected paper rolls	70%
Dissolved solids content in repulped rejected paper rolls	0.4%

Flow rate of cloudy whitewater entering the papermaking section from the TMP shop	9000 l/min
Consistency of cloudy whitewater entering the papermaking section from the TMP shop	0.05%
Temperature of cloudy whitewater entering the papermaking section from the TMP shop	56 °C
Fibre content of cloudy whitewater entering the papermaking section from the TMP shop	10%
Dissolved solids content of cloudy whitewater entering the papermaking section from the TMP shop	0.325%

Flow rate of fresh water added to Machine 6	2300 l/min
Flow rate of fresh water added to Machine 7	2400 l/min
Flow rate of fresh water added to Machine 8	3100 l/min
Flow rate of fresh water added to Machine 9	2400 l/min

Temperature of fresh water added to Machine 6	60 °C
Temperature of fresh water added to Machine 7	60 °C
Temperature of fresh water added to Machine 8	60 °C
Temperature of fresh water added to Machine 9	60 °C

APPENDIX III: CODE

CODE FOR THE NELDER-MEAD SIMPLEX ALGORITHM

```
Sub Sort_xr()
'
' Sort_xr Macro
' Macro écrit le 28/02/02 par Michal Dabros
' Sorts the simplex according to objective function values and assigns the
' input xr for the following simulation run

    Application.Run "check"

    Range("B2:H6").Select
    Selection.sort Key1:=Range("H2"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
    Range("C7").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"
    Range("D7").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"
    Range("E7").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"

End Sub
```

```
Sub xcc()
'
' xcc Macro
' Macro écrit le 28/02/02 par Michal Dabros
' Assigns the input xcc for cases where f(r) >= f(n+1)

    Application.Run "check"

    Range("C8").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C[11]"
    Range("D8").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C[11]"
    Range("E8").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C[11]"

End Sub
```

```
Sub xc()
```

```
'
' xc Macro
' Macro écrit le 28/02/02 par Michal Dabros
' Assigns the input xc for cases where f(n) <= f(r) < f(n+1)

    Application.Run "check"

    Range("C8").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C[11]"
    Range("D8").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C[11]"
    Range("E8").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C[11]"

End Sub
```

```
Sub xe()
'
' xe Macro
' Macro écrit le 28/02/02 par Michal Dabros
' Assigns the input xc for cases where f(r) < f(1)

    Application.Run "check"

    Range("C8").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"
    Range("D8").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"
    Range("E8").Select
    ActiveCell.FormulaR1C1 = "=R[-4]C[11]"

End Sub
```

```
Sub xr()
'
' xe Macro
' Macro enregistrée le 28/02/02 par Michal Dabros
' Assigns the input xc for cases where f(1) <= f(r) < f(n)

    Application.Run "check"

    Range("C8").Select
    ActiveCell.FormulaR1C1 = "=R[-5]C[11]"
    Range("D8").Select
```

```
    ActiveCell.FormulaR1C1 = "=R[-5]C[11]"
    Range("E8").Select
    ActiveCell.FormulaR1C1 = "=R[-5]C[11]"

End Sub
```

```
Sub compareandact()
'
'compareandact Macro
' Macro écrit le 28-02-2002 par Michal Dabros
' Compares the value of f(r) to the objective function values of the
' main simplex and accepts xr if f(1) <= f(r) < f(n) or assigns xe,
' xc or xcc as the next input, depending on the ranking of f(r)

    Application.Run "check"

    If Range("H7") >= Range("H2") And Range("H7") < Range("H5") Then
       xr 'assign xr (trivial case)
    End If

    If Range("H7") < Range("H2") Then
       xe 'assign xe
    End If

    If Range("H7") < Range("H6") And Range("H7") >= Range("H5") Then
       xc 'assign xc
    End If

    If Range("H7") >= Range("H6") Then
       xcc 'assign xcc
    End If

End Sub
```

```
Sub replace()
'
' replace Macro
' Macro écrit le 01-03-2002 par Michal Dabros
' Replaces the worst input in the mains simplex (xn+1) with the appropriate
' new input (either xr or one of xe, xc or xcc, depending on the case)

    Application.Run "check"

' If the input that was tried second gave a lower function value than f(r)
```

```
If Range("H8") < Range("H7") Then
   Range("C8:E8").Select
   Selection.Copy
   Range("C6:E6").Select
   Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
      False, Transpose:=False
   Range("H8").Select
   Application.CutCopyMode = False
   Selection.Copy
   Range("H6").Select
   Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
      False, Transpose:=False

   Range("H7:H8").Select
   Selection.ClearContents
   Range("C7:E8").Select
   Selection.ClearContents
   Range("H10").Select

' If the input that was tried second gave a higher function value than f(r)
Else
   Range("C7:E7").Select
   Selection.Copy
   Range("C6:E6").Select
   Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
      False, Transpose:=False
   Range("H7").Select
   Application.CutCopyMode = False
   Selection.Copy
   Range("H6").Select
   Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
      False, Transpose:=False

   Range("H7:H8").Select
   Selection.ClearContents
   Range("C7:E8").Select
   Selection.ClearContents
   Range("H10").Select

End If

End Sub
```

```
Sub sort()
```

```
'
' sort Macro
' Macro écrit le 10 mai 2002 par Michal Dabros
' Sorts the final simplex after the last algorithm iteration

   Application.Run "check"

   Range("B2:H6").Select
   Selection.sort Key1:=Range("H2"), Order1:=xlAscending, Header:=xlGuess, _
      OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
   Range("H10").Select

   Range("I14") = 7 'Set Current Input Number

   Range("I15") = Range("I15").Value + 1 'Augment Iteration Number

End Sub
```

```
Sub movecurrentinput()
'
' Macro écrit le 24 mai 2002 par Michal Dabros
' Moves the values of the currently simulated input to the appropriate
' place in the input spreadsheet, from where it is read dynamically by
' the simulation

Application.Run "check"

i = Range("I14").Value ' Current Input Number

Range("B12:F12").FormulaArray = Range("B" & (i) & ":F" & (i) & "").Value
' (move the input values)

End Sub
```

```
Sub movecurrentobjectivefunction()
'
' Macro écrit le 24 mai 2002 par Michal Dabros
' Moves the calculated value of the objective function from the output
' spreadsheet to the appropriate place in the simplex table after the
' simulation is terminated

Application.Run "check"
'
i = Range("I14").Value 'Current Input Number
```

```
Range("H" & (i) & "").FormulaR1C1 = Range("H12").Value
'(move the objective funciton value)

Range("I14") = Range("I14").Value + 1 'Augment Current Input Number

End Sub
```

```
Sub initiate()
'
'initiate Macro
' Macro écrit le 28-02-2002 par Michal Dabros
' Randomly creates an initial simplex and resets the worksheet for a given machine
'


    Range("C2").Select
    ActiveCell.FormulaR1C1 = Range("b2") + (Range("f2") - Range("b2")) * Rnd
    Range("D2").Select
    ActiveCell.FormulaR1C1 = Range("c2") + (Range("f2") - Range("c2")) * Rnd
    Range("E2").Select
    ActiveCell.FormulaR1C1 = Range("d2") + (Range("f2") - Range("d2")) * Rnd
    Range("C3").Select
    ActiveCell.FormulaR1C1 = Range("b3") + (Range("f3") - Range("b3")) * Rnd
    Range("D3").Select
    ActiveCell.FormulaR1C1 = Range("c3") + (Range("f3") - Range("c3")) * Rnd
    Range("E3").Select
    ActiveCell.FormulaR1C1 = Range("d3") + (Range("f3") - Range("d3")) * Rnd
    Range("C4").Select
    ActiveCell.FormulaR1C1 = Range("b4") + (Range("f4") - Range("b4")) * Rnd
    Range("D4").Select
    ActiveCell.FormulaR1C1 = Range("c4") + (Range("f4") - Range("c4")) * Rnd
    Range("E4").Select
    ActiveCell.FormulaR1C1 = Range("d4") + (Range("d4") - Range("d4")) * Rnd
    Range("C5").Select
    ActiveCell.FormulaR1C1 = Range("b5") + (Range("f5") - Range("b5")) * Rnd
    Range("D5").Select
    ActiveCell.FormulaR1C1 = Range("c5") + (Range("f5") - Range("c5")) * Rnd
    Range("E5").Select
    ActiveCell.FormulaR1C1 = Range("d5") + (Range("f5") - Range("d5")) * Rnd
    Range("C6").Select
    ActiveCell.FormulaR1C1 = Range("b6") + (Range("f6") - Range("b6")) * Rnd
    Range("D6").Select
    ActiveCell.FormulaR1C1 = Range("c6") + (Range("f6") - Range("c6")) * Rnd
    Range("E6").Select
```

```
    ActiveCell.FormulaR1C1 = Range("d6") + (Range("f6") - Range("d6")) * Rnd

'Clear Objective Function result table and simulation output table
    Range("C7:E8").Select
    Selection.ClearContents
    Range("H2:H8").Select
    Selection.ClearContents
    Range("G20:L1459").Select
    Selection.ClearContents

    Range("I14") = 2 'Initial Input Number
    Range("I15") = 0 'Initial Iteration Number

    Range("A1").Select

End Sub
```

```
Sub Initiate_All()
'
' Initiate_All Macro
' Macro written 07/05/2002 by Michal Dabros
' Runs the 'initiate' macro for all four machines

    Sheets("CA6").Select
    Application.Run "Interface_R0.xls!initiate"
    Sheets("CA7").Select
    Application.Run "Interface_R0.xls!initiate"
    Sheets("CA8").Select
    Application.Run "Interface_R0.xls!initiate"
    Sheets("CA9").Select
    Application.Run "Interface_R0.xls!initiate"
    Sheets("IO").Select

End Sub
```

```
Sub clearoriginalcase()
'
' clearoriginalcase Macro
' Macro written 17/06/2002 by Michal Dabros
' Resets the original case worksheet
'

    Range("W10:AB2889").Select
    Selection.ClearContents
```

```
    Range("AJ10:AO2889").Select
    Selection.ClearContents

    Range("AW10:BB2889").Select
    Selection.ClearContents

    Range("BJ10:BO2889").Select
    Selection.ClearContents

    Range("A1").Select

End Sub
```

```
Sub check()
'
' Macro écrit le 24 mai 2002 par Michal Dabros
' Verifies which machine is being optimized and opens up the appropriate
' spreadsheet before an operative macro (such as sort, xr, xc etc.) is called

   If Sheets("IO").Range("b7") = 6 Then
      Sheets("CA6").Select
   ElseIf Sheets("IO").Range("b7") = 7 Then
      Sheets("CA7").Select
   ElseIf Sheets("IO").Range("b7") = 8 Then
      Sheets("CA8").Select
   ElseIf Sheets("IO").Range("b7") = 9 Then
      Sheets("CA9").Select
   End If

End Sub
```

```
Sub optimize6()
' Assigns Machine 6 to be optimized
Sheets("IO").Select
Range("b7") = 6
End Sub
```

```
Sub optimize7()
' Assigns Machine 7 to be optimized
Sheets("IO").Select
Range("b7") = 7
End Sub
```

```
Sub optimize8()
'Assigns Machine 8 to be optimized
Sheets("IO").Select
Range("b7") = 8
End Sub
```

```
Sub optimize9()
'Assigns Machine 9 to be optimized
Sheets("IO").Select
Range("b7") = 9
End Sub
```

```
Sub gotoio()
'Opens the IO (input/output) spreadsheet in the Excel interface during
'the simulation in order to speed it up. This spreadsheet contains no
'graphs, and therefore, consumes less computer memory

Sheets("IO").Select

End Sub
```

```
Sub resetinputnumber()
'Reset the initial input number after the iteration series is completed
'This macro also assigns the best input to be used by the simulation for
'the particular machine after the iteration series is completed and while
'another machine is being optimized

Range("I14") = 2 'Initial Input Number

Range("B2:F2").Select
Selection.Copy
Range("B12").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
   :=False, Transpose:=False
Range("A1").Select
Application.CutCopyMode = False

End Sub
```

```
Sub moreafter10iter()
'
'moreafter10iter Macro
```

```
'Macro written 2002-07-06 by Michal Dabros
'Saves the results after 10 iterations before continuing with more iterations

Sheets("OriginalCase").Select
   Range("C4:C7").Select
   Selection.Copy
   Range("D4").Select
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False

Sheets("CA6").Select
   Range("B2:F6").Select
   Selection.Copy
   Range("R2").Select
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Range("H2:H6").Select
   Application.CutCopyMode = False
   Selection.Copy
   Range("W2").Select
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Application.CutCopyMode = False
   Range("A1").Select

Sheets("CA7").Select
   Range("B2:F6").Select
   Selection.Copy
   Range("R2").Select
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Range("H2:H6").Select
   Application.CutCopyMode = False
   Selection.Copy
   Range("W2").Select
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Application.CutCopyMode = False
   Range("A1").Select

Sheets("CA8").Select
   Range("B2:F6").Select
   Selection.Copy
   Range("R2").Select
```

```
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Range("H2:H6").Select
    Application.CutCopyMode = False
    Selection.Copy
    Range("W2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Range("A1").Select

Sheets("CA9").Select
    Range("B2:F6").Select
    Selection.Copy
    Range("R2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Range("H2:H6").Select
    Application.CutCopyMode = False
    Selection.Copy
    Range("W2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Range("A1").Select

End Sub
```

CODE FOR GENETIC ALGORITHMS

```
Sub sort()
'
'sort Macro
' Macro écrit le 05/03/02 par Michal Dabros
' Sorts the generation after an iteration, and augments the iteration number

Application.Run "check"

   Range("A2:AS21").Select
   Selection.sort Key1:=Range("AS2"), Order1:=xlAscending, Header:=xlGuess, _
      OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
   ActiveWindow.LargeScroll ToRight:=1
   Range("AL23").Select

   Range("AY18") = Range("AY18").Value + 1 'Augment Iteration Number
   Range("AY19") = 2 'Reset Chromosome # Treated to zero

End Sub
```

```
Sub crossover()
'
' crossover Macro
' Macro écrit le 17 mai 2002 par Michal Dabros
' Simulates the algorithm's Crossover operation

Application.Run "check"

Let keep = Range("AY10").Value 'Number of rows to be conserved
For k = 2 To 2 + keep - 1
   Range("A" & (k) & "") = "conserve"
Next k

Let probc = Range("AY11").Value 'Probability of crossover

For i = 2 + keep To 21 'For each chromosome
   p = Rnd
   If p <= probc Then
      Range("A" & (i) & "") = "cross" 'Label as crossed
      Let rand1 = 1 + Round(35 * Rnd) 'Random1 - column from 1 to 36
```

```
      r = Rnd
      If r < (i - 2) / (21 - 2) Then
        Let rand2 = 2 + Round((i - 3) * Rnd) 'Random2 - row to crossover with
      Else
        Let rand2 = i + Round(1 + (20 - i) * Rnd) 'Random2 - row to crossover with
      End If

      'Define the Parents
      Range("A" & (i) & "").Select
      ActiveCell.Offset(0, rand1).Select
      Let p1 = ActiveCell.AddressLocal 'Parent1
      ActiveCell.Offset(rand2 - i, 0).Select
      Let p2 = ActiveCell.AddressLocal 'Parent2

      Let Order = Rnd 'Define the order of Parents
      If Order < 0.5 Then
        Range("" & (p1) & ":AK" & (i) & "").FormulaArray = Range("" & (p2) & ":AK"
& (rand2) & "").Value
      Else
        Range("B" & (i) & ":" & (p1) & "").FormulaArray = Range("B" & (rand2) & ":"
& (p2) & "").Value
      End If
  Else
    Range("A" & (i) & "") = "keep" 'Label as kept (not crossed)
  End If
Next i

End Sub
```

```
Sub mutation()
'
'mutation Macro
' Macro écrit le 16 mai 2002 par Michal Dabros
' Simulates the algorithm's Mutation operation

Application.Run "check"

Let keep = Range("AY14").Value 'Number of rows to be conserved
For k = 2 To 2 + keep - 1
  Range("A" & (k) & "") = "conserve"
Next k

Let probm = Range("AY15").Value 'Probability of mutation
```

```
For i = 2 + keep To 21 'For each chromosome
   p = Rnd
   If p <= probm Then
      Range("A" & (i) & "") = "mutate" 'Label as mutated
      Range("A" & (i) & "").Select
      ActiveCell.Offset(0, 1 + Rnd * 35).Select
      If ActiveCell = 0 Then
         ActiveCell = 1
      Else
         ActiveCell = 0
      End If
   Else
      Range("A" & (i) & "") = "keep" 'Label as kept (not-mutated)
   End If
Next i

End Sub
```

```
Sub movecurrentinput()
'
' Macro écrit le 24 mai 2002 par Michal Dabros
' Moves the values of the currently simulated input to the appropriate
' place in the input spreadsheet, from where it is read dynamically by
' the simulation

Application.Run "check"
'

i = Range("AY19").Value 'Current Chromosome Number

Range("AM26:AQ26").FormulaArray = Range("AM" & (i) & ":AQ" & (i) & "").Value
' (move the input values)

End Sub
```

```
Sub movecurrentobjectivefunction()
'
' Macro écrit le 24 mai 2002 par Michal Dabros
' Moves the calculated value of the objective function from the output
' spreadsheet to the appropriate place in the simplex table after the
' simulation is terminated

Application.Run "check"

i = Range("AY19").Value 'Current Chromosome Number
```

```
Range("AS" & (i) & "").FormulaR1C1 = Range("AS26").Value
'(move the objective funciton value)

Range("AY19") = Range("AY19").Value + 1 'Augment Chromosome Number

End Sub
```

```
Sub initiate()
'
'initiate Macro
'Macro written on 28-02-2002 by Michal Dabros
'Simulates the algorithm's Selection operation by randomly choosing
'the initial generation and rests the worksheet for a given machine

Range("a1").Select
For i = 2 To 21
ActiveCell.Offset(1, 0).Select
For j = 1 To 36
   ActiveCell.Offset(0, 1).Select
   ActiveCell.Value = Round(Rnd) 'Range("b" & (i) & "")
Next j
Range("a" & (i) & "").Select
Next i

'Clear Objective Function result table and simulation output table
Range("AS2:AS21").ClearContents
Range("AR32:AW1471").ClearContents

Range("AY18") = 0 'Iteration Number
Range("AY19") = 2 'Chromosome # Treated

End Sub
```

```
Sub Initiate_All()
'
'Initiate_All Macro
'Macro written 07/05/2002 by Michal Dabros
'Runs the 'initiate' macro for all four machines

   Sheets("CA6").Select
   Application.Run "initiate"
   Sheets("CA7").Select
   Application.Run "initiate"
```

```
    Sheets("CA8").Select
    Application.Run "initiate"
    Sheets("CA9").Select
    Application.Run "initiate"
    Sheets("IO").Select

End Sub
```

```
Sub clearoriginalcase()
'
' clearoriginalcase Macro
' Macro written 17/06/2002 by Michal Dabros
' Resets the original case worksheet

    Range("W10:AB2889").Select
    Selection.ClearContents

    Range("AJ10:AO2889").Select
    Selection.ClearContents

    Range("AW10:BB2889").Select
    Selection.ClearContents

    Range("BJ10:BO2889").Select
    Selection.ClearContents

    Range("A1").Select

End Sub
```

```
Sub check()
'
' check Macro
' Macro écrit le 24 mai 2002 par Michal Dabros
' Verifies which machine is being optimized and opens up the appropriate
' spreadsheet before an operative macro (such as sort, crossover,
' mutation etc.) is called

    If Sheets("IO").Range("b7") = 6 Then
        Sheets("CA6").Select
    ElseIf Sheets("IO").Range("b7") = 7 Then
        Sheets("CA7").Select
    ElseIf Sheets("IO").Range("b7") = 8 Then
        Sheets("CA8").Select
```

```
    ElseIf Sheets("IO").Range("b7") = 9 Then
        Sheets("CA9").Select
    End If

End Sub
```

```
Sub optimize6()
'Assigns Machine 6 to be optimized
Sheets("IO").Select
Range("b7") = 6
End Sub
```

```
Sub optimize7()
'Assigns Machine 7 to be optimized
Sheets("IO").Select
Range("b7") = 7
End Sub
```

```
Sub optimize8()
'Assigns Machine 8 to be optimized
Sheets("IO").Select
Range("b7") = 8
End Sub
```

```
Sub optimize9()
'Assigns Machine 9 to be optimized
Sheets("IO").Select
Range("b7") = 9
End Sub
```

```
Sub gotoio()
'Opens the IO (input/output) spreadsheet in the Excel interface during the
'simulation in order to speed it up. This spreadsheet contains no graphs,
'and therefore, consumes less computer memory

Sheets("IO").Select

End Sub
```

```
Sub moreafter10iter()
'
'moreafter10iter Macro
'Macro written 2002-07-06 by Michal Dabros
```

```
'Saves the results after 10 iterations before continuing with more iterations

Sheets("OriginalCase").Select

    Range("C4:C7").Select
    Selection.Copy
    Range("D4").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

Sheets("CA6").Select
    Range("AM2:AS21").Select
    Selection.Copy
    ActiveWindow.SmallScroll ToRight:=16
    Range("BB2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    ActiveWindow.LargeScroll ToRight:=-1

Sheets("CA7").Select
    Range("AM2:AS21").Select
    Selection.Copy
    ActiveWindow.SmallScroll ToRight:=16
    Range("BB2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    ActiveWindow.LargeScroll ToRight:=-1

Sheets("CA8").Select
    Range("AM2:AS21").Select
    Selection.Copy
    ActiveWindow.SmallScroll ToRight:=16
    Range("BB2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    ActiveWindow.LargeScroll ToRight:=-1

Sheets("CA9").Select
    Range("AM2:AS21").Select
    Selection.Copy
    ActiveWindow.SmallScroll ToRight:=16
    Range("BB2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
```

```
  ActiveWindow.LargeScroll ToRight:=-1
End Sub
```

CODE FOR THE WINGEMS-EXCEL INTERFACE

Script : Execution

```
‘ Basic execution script
‘ Created Thursday, May 03, 2001
Read CurrentValues from FILE::CurrentValues.CV
Call script {TOPLEVEL}:Run Steady State
‘ Run a test simulation in dynamic mode
Repeat with time = 0 to 1:00:00:00 step 01:00
  Call script {TOPLEVEL}:Calculation order
End repeat
‘ Check which algorithm to use
If (B500"NMS":4 > 0) Then
  Call script {TOPLEVEL}:Execution NMS
  Send [Run("moreafter10iter")] to XL::
  Call script {TOPLEVEL}:Execution NMS
End If
If (B501"GA":4 > 0) Then
  Call script {TOPLEVEL}:Execution GA
  Send [Run("moreafter10iter")] to XL::
  Call script {TOPLEVEL}:Execution GA
End If
```

Script : Run Steady State

```
Repeat until converged...
  Call script {TOPLEVEL}:Calculation order
End repeat
```

Script : Execution NMS

```
‘ Basic execution script
‘ Created Thursday, May 03, 2001
‘ NELDER-MEAD SIMPLEX
Call script {TOPLEVEL}:Run Original Case NMS
Call script {TOPLEVEL}:XL-IO NMS
‘ Run the Optimization Algorithm
Send [Run("optimize6")] to XL::
Call script {TOPLEVEL}:Optimize NMS
Send [Run("optimize7")] to XL::
```

```
Call script {TOPLEVEL}:Optimize NMS
Send [Run("optimize8")] to XL::
Call script {TOPLEVEL}:Optimize NMS
Send [Run("optimize9")] to XL::
Call script {TOPLEVEL}:Optimize NMS
```

```
Script : Run Original Case NMS

Read CurrentValues from FILE::CurrentValues.CV
Call script {TOPLEVEL}:XL-IO NMS
Call script {TOPLEVEL}:Run Steady State
' RUN in Dynamic Mode
Repeat with time = 0 to 1:00:00:00 step 01:00
 Call script {TOPLEVEL}:XL-IO NMS
 Call script {TOPLEVEL}:Calculation order
 ' Read Wet-end Breaks
 Read B306:4 offset from XL::[Interface_R0.xls]OriginalCase!R10
 Read B307:4 offset from XL::[Interface_R0.xls]OriginalCase!AE10
 Read B308:4 offset from XL::[Interface_R0.xls]OriginalCase!AR10
 Read B309:4 offset from XL::[Interface_R0.xls]OriginalCase!BE10
 ' Read Dry-end Breaks
 Read B316:4 offset from XL::[Interface_R0.xls]OriginalCase!S10
 Read B317:4 offset from XL::[Interface_R0.xls]OriginalCase!AF10
 Read B318:4 offset from XL::[Interface_R0.xls]OriginalCase!AS10
 Read B319:4 offset from XL::[Interface_R0.xls]OriginalCase!BF10
 ' Read Broke Ratio Values
 Read B336:4 offset from XL::[Interface_R0.xls]OriginalCase!U10
 Read B337:4 offset from XL::[Interface_R0.xls]OriginalCase!AH10
 Read B338:4 offset from XL::[Interface_R0.xls]OriginalCase!AU10
 Read B339:4 offset from XL::[Interface_R0.xls]OriginalCase!BH10
 Read PTM ratio values
 Read B326:4 offset from XL::[Interface_R0.xls]OriginalCase!V10
 Read B327:4 offset from XL::[Interface_R0.xls]OriginalCase!AI10
 Read B328:4 offset from XL::[Interface_R0.xls]OriginalCase!AV10
 Read B329:4 offset from XL::[Interface_R0.xls]OriginalCase!BI10
 ' Register Time Stamp
 Write b26:2 to XL::[Interface_R0.xls]IO!B20
 ' Register Headbox Conditions
 Write s241:1, s241:2, s241:3, s241:4, s241:5, s241:7, offset to
XL::[Interface_R0.xls]OriginalCase!W10:AB10
 Write s210:1, s210:2, s210:3, s210:4, s210:5, s210:7, offset to
XL::[Interface_R0.xls]OriginalCase!AJ10:AO10
 Write s473:1, s473:2, s473:3, s473:4, s473:5, s473:7, offset to
XL::[Interface_R0.xls]OriginalCase!AW10:BB10
```

```
 Write s492:1, s492:2, s492:3, s492:4, s492:5, s492:7, offset to
XL::[Interface_R0.xls]OriginalCase!BJ10:BO10
End repeat
```

Script : XL-IO NMS

```
' Read which machine is to be optimized
Read B101:1 from XL::[Interface_R0.xls]IO!B7
' Read the total flow of pulp to each machine (constant)
Read b406:4 from XL::[Interface_R0.xls]IO!B11
Read b407:4 from XL::[Interface_R0.xls]IO!C11
Read b408:4 from XL::[Interface_R0.xls]IO!D11
Read b409:4 from XL::[Interface_R0.xls]IO!E11
' Read the flow rate of TMP Cloudy WW to Lean WW tank (constant)
Read b110:4 from XL::[Interface_R0.xls]IO!F12
' Read the flow rate of repulped paper rejects (constant)
Read b56:4 from XL::[Interface_R0.xls]IO!F13
' Write the total flow to Secondary Treatment
Write s600:1 to XL::[Interface_R0.xls]IO!B16
' Write the total flow of WW to TMP and DIP
Write b23:1 to XL::[Interface_R0.xls]IO!B17
```

Script : Optimize NMS

```
Call script {TOPLEVEL}:Initial Simplex NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Call script {TOPLEVEL}:Algorithm NMS
Send [Run("resetinputnumber")] to XL::
```

Script : Initial Simplex NMS

```
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
```

```
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
```

Script : Algorithm NMS

```
Send [Run("sort_xr")] to XL::[Interface_R0.xls]CA6
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
Send [Run("compareandact")] to XL::
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic NMS
Send [Run("movecurrentobjectivefunction")] to XL::
Send [Run("replace")] to XL::
Send [Run("sort")] to XL::
```

Script : Run Dynamic NMS

```
Read CurrentValues from FILE::CurrentValues.CV
Call script {TOPLEVEL}:XL-IO NMS
Call script {TOPLEVEL}:Run Steady State
' RUN in Dynamic Mode
Repeat with time = 0 to 1:00:00:00 step 01:00
 Call script {TOPLEVEL}:XL-IO NMS
 Call script {TOPLEVEL}:Calculation order
 ' Read Wet-end Breaks
 Read B306:4 offset from XL::[Interface_R0.xls]CA6!B20
 Read B307:4 offset from XL::[Interface_R0.xls]CA7!B20
```

```
  Read B308:4 offset from XL::[Interface_R0.xls]CA8!B20
  Read B309:4 offset from XL::[Interface_R0.xls]CA9!B20
  ' Read Dry-end Breaks
  Read B316:4 offset from XL::[Interface_R0.xls]CA6!C20
  Read B317:4 offset from XL::[Interface_R0.xls]CA7!C20
  Read B318:4 offset from XL::[Interface_R0.xls]CA8!C20
  Read B319:4 offset from XL::[Interface_R0.xls]CA9!C20
  ' Read Broke Ratio changes
  Read B336:4 offset from XL::[Interface_R0.xls]CA6!E20
  Read B337:4 offset from XL::[Interface_R0.xls]CA7!E20
  Read B338:4 offset from XL::[Interface_R0.xls]CA8!E20
  Read B339:4 offset from XL::[Interface_R0.xls]CA9!E20
  ' Read PTM changes
  Read B326:4 offset from XL::[Interface_R0.xls]CA6!F20
  Read B327:4 offset from XL::[Interface_R0.xls]CA7!F20
  Read B328:4 offset from XL::[Interface_R0.xls]CA8!F20
  Read B329:4 offset from XL::[Interface_R0.xls]CA9!F20
  ' Register Time Stamp
  Write b26:2 to XL::[Interface_R0.xls]IO!B20
  ' Register Headbox Conditions
  Write s241:1, s241:2, s241:3, s241:4, s241:5, s241:7, offset to
XL::[Interface_R0.xls]CA6!G20:L20
  Write s210:1, s210:2, s210:3, s210:4, s210:5, s210:7, offset to
XL::[Interface_R0.xls]CA7!G20:L20
  Write s473:1, s473:2, s473:3, s473:4, s473:5, s473:7, offset to
XL::[Interface_R0.xls]CA8!G20:L20
  Write s492:1, s492:2, s492:3, s492:4, s492:5, s492:7, offset to
XL::[Interface_R0.xls]CA9!G20:L20
End repeat
```

Script : Execution GA

```
' Basic execution script
' Created Thursday, May 03, 2001
' GENETIC ALGORITHMS

' Run Original Case first
Call script {TOPLEVEL}:Run Original Case GA
' Call the desired numer of iterations
Send [Run("optimize6")] to XL::
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
```

```
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Last Iteration GA
Send [Run("optimize7")] to XL::
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Last Iteration GA
Send [Run("optimize8")] to XL::
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Last Iteration GA
Send [Run("optimize9")] to XL::
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Iteration GA
Call script {TOPLEVEL}:Last Iteration GA
```

Script : Run Original Case GA

```
Read CurrentValues from FILE::CurrentValues.CV
Call script {TOPLEVEL}:XL-IO GA
```

```
 Call script {TOPLEVEL}:Run Steady State
 ' RUN in Dynamic Mode
 Repeat with time = 0 to 1:00:00:00 step 01:00
  Call script {TOPLEVEL}:XL-IO GA
  Call script {TOPLEVEL}:Calculation order
  ' Read Wet-end Breaks
  Read B306:4 offset from XL::[Interface_GA.xls]OriginalCase!R10
  Read B307:4 offset from XL::[Interface_GA.xls]OriginalCase!AE10
  Read B308:4 offset from XL::[Interface_GA.xls]OriginalCase!AR10
  Read B309:4 offset from XL::[Interface_GA.xls]OriginalCase!BE10
  ' Read Dry-end Breaks
  Read B316:4 offset from XL::[Interface_GA.xls]OriginalCase!S10
  Read B317:4 offset from XL::[Interface_GA.xls]OriginalCase!AF10
  Read B318:4 offset from XL::[Interface_GA.xls]OriginalCase!AS10
  Read B319:4 offset from XL::[Interface_GA.xls]OriginalCase!BF10
  ' Read Total Pulp Flow Rates
  Read B406:4 offset from XL::[Interface_GA.xls]OriginalCase!T10
  Read B407:4 offset from XL::[Interface_GA.xls]OriginalCase!AG10
  Read B408:4 offset from XL::[Interface_GA.xls]OriginalCase!AT10
  Read B409:4 offset from XL::[Interface_GA.xls]OriginalCase!BG10
  ' Read Broke Ratio Values
  Read B336:4 offset from XL::[Interface_GA.xls]OriginalCase!U10
  Read B337:4 offset from XL::[Interface_GA.xls]OriginalCase!AH10
  Read B338:4 offset from XL::[Interface_GA.xls]OriginalCase!AU10
  Read B339:4 offset from XL::[Interface_GA.xls]OriginalCase!BH10
  ' Read PTM Ratio Values
  Read B326:4 offset from XL::[Interface_GA.xls]OriginalCase!V10
  Read B327:4 offset from XL::[Interface_GA.xls]OriginalCase!AI10
  Read B328:4 offset from XL::[Interface_GA.xls]OriginalCase!AV10
  Read B329:4 offset from XL::[Interface_GA.xls]OriginalCase!BI10
  ' Register Time Stamp
  Write b26:2 to XL::[Interface_GA.xls]IO!B20
  ' Register Headbox Conditions
  Write s241:1, s241:2, s241:3, s241:4, s241:5, s241:7, offset to
XL::[Interface_GA.xls]OriginalCase!W10:AB10
  Write s210:1, s210:2, s210:3, s210:4, s210:5, s210:7, offset to
XL::[Interface_GA.xls]OriginalCase!AJ10:AO10
  Write s473:1, s473:2, s473:3, s473:4, s473:5, s473:7, offset to
XL::[Interface_GA.xls]OriginalCase!AW10:BB10
  Write s492:1, s492:2, s492:3, s492:4, s492:5, s492:7, offset to
XL::[Interface_GA.xls]OriginalCase!BJ10:BO10
End repeat
```

Script : Iteration GA

```
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Send [Run("sort")] to XL::
Send [Run("mutation")] to XL::
Send [Run("crossover")] to XL::
```

Script : Last Iteration GA

```
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
```

```
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Call script {TOPLEVEL}:Chromosome GA
Send [Run("sort")] to XL::
Send [Run("movecurrentinput")] to XL::
```

Script : XL-IO GA

```
‘ Read which machine is to be optimized
Read B101:1 from XL::[Interface_GA.xls]IO!B7
‘ Read the total flow of pulp to each machine (constant)
Read b406:4 from XL::[Interface_GA.xls]IO!B11
Read b407:4 from XL::[Interface_GA.xls]IO!C11
Read b408:4 from XL::[Interface_GA.xls]IO!D11
Read b409:4 from XL::[Interface_GA.xls]IO!E11
‘ Read the flow rate of TMP Cloudy WW to Lean WW tank (constant)
Read b110:4 from XL::[Interface_GA.xls]IO!F12
‘ Read the flow rate of repulped paper rejects (constant)
Read b56:4 from XL::[Interface_GA.xls]IO!F13
‘ Write the total flow to Secondary Treatment
Write s600:1 to XL::[Interface_GA.xls]IO!B16
‘ Write the total flow of WW to TMP and DIP
Write b23:1 to XL::[Interface_GA.xls]IO!B17
```

Script : Chromosome GA

```
Send [Run("movecurrentinput")] to XL::
Send [Run("gotoio")] to XL::
Call script {TOPLEVEL}:Run Dynamic GA
Send [Run("movecurrentobjectivefunction")] to XL::
```

Script : Run Dynamic GA

```
Read CurrentValues from FILE::CurrentValues.CV
Call script {TOPLEVEL}:XL-IO GA
Call script {TOPLEVEL}:Run Steady State
‘ RUN in Dynamic Mode
Repeat with time = 0 to 1:00:00:00 step 01:00
 Call script {TOPLEVEL}:XL-IO GA
 Call script {TOPLEVEL}:Calculation order
 ‘ Read Wet-end Breaks
 Read B306:4 offset from XL::[Interface_GA.xls]CA6!AM32
 Read B307:4 offset from XL::[Interface_GA.xls]CA7!AM32
```

```
 Read B308:4 offset from XL::[Interface_GA.xls]CA8!AM32
 Read B309:4 offset from XL::[Interface_GA.xls]CA9!AM32
 ‘ Read Dry-end Breaks
 Read B316:4 offset from XL::[Interface_GA.xls]CA6!AN32
 Read B317:4 offset from XL::[Interface_GA.xls]CA7!AN32
 Read B318:4 offset from XL::[Interface_GA.xls]CA8!AN32
 Read B319:4 offset from XL::[Interface_GA.xls]CA9!AN32
 ‘ Read Broke Ratio Values
 Read B336:4 offset from XL::[Interface_GA.xls]CA6!ap32
 Read B337:4 offset from XL::[Interface_GA.xls]CA7!ap32
 Read B338:4 offset from XL::[Interface_GA.xls]CA8!ap32
 Read B339:4 offset from XL::[Interface_GA.xls]CA9!ap32
 ‘ Read PTM Ratio Values
 Read B326:4 offset from XL::[Interface_GA.xls]CA6!aq32
 Read B327:4 offset from XL::[Interface_GA.xls]CA7!aq32
 Read B328:4 offset from XL::[Interface_GA.xls]CA8!aq32
 Read B329:4 offset from XL::[Interface_GA.xls]CA9!aq32
 ‘ Register Time Stamp
 Write b26:2 to XL::[Interface_GA.xls]IO!B20
 ‘ Register Headbox Conditions
 Write s241:1, s241:2, s241:3, s241:4, s241:5, s241:7, offset to
XL::[Interface_GA.xls]CA6!AR32:AW32
 Write s210:1, s210:2, s210:3, s210:4, s210:5, s210:7, offset to
XL::[Interface_GA.xls]CA7!AR32:AW32
 Write s473:1, s473:2, s473:3, s473:4, s473:5, s473:7, offset to
XL::[Interface_GA.xls]CA8!AR32:AW32
 Write s492:1, s492:2, s492:3, s492:4, s492:5, s492:7, offset to
XL::[Interface_GA.xls]CA9!AR32:AW32
End repeat
```