POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Méthode de décomposition pour l'optimisation de la contrôlabilité du procédé de lavage de pâte à papier
Title:

Auteur: Benoit Archambault
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Archambault, B. (2003). Méthode de décomposition pour l'optimisation de la contrôlabilité du procédé de lavage de pâte à papier [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7111/
Citation:

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: https://publications.polymtl.ca/7111/
PolyPublie URL:

Directeurs de recherche: Brigitte Jaumard, Michel Perrier, & Gilles Savard
Advisors:

Programme: Non spécifié
Program:

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

UNIVERSITÉ DE MONTRÉAL

MÉTHODE DE DÉCOMPOSITION POUR L'OPTIMISATION DE LA CONTRÔLABILITÉ DU PROCÉDÉ DE LAVAGE DE PÂTE À PAPIER

BENOIT ARCHAMBAULT
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
SEPTEMBRE 2003

© Benoit Archambault, 2003.

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisisitons et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-86383-2
Our file Notre référence
ISBN: 0-612-86383-2

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MÉTHODE DE DÉCOMPOSITION POUR L'OPTIMISATION DE LA CONTRÔLABILITÉ DU PROCÉDÉ DE LAVAGE DE PÂTE À PAPIER

présenté par: ARCHAMBAULT Benoit

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. AUDET Charles, Ph. D., président
Mme. JAUMARD Brigitte, T. Doct., T. Hab., membre et directrice de recherche
M. PERRIER Michel, Ph. D., membre et codirecteur de recherche
M. SAVARD Gilles, Ph. D., membre et codirecteur de recherche
M. STUART Paul , Ph. D., membre

REMERCIEMENTS

Je tiens tout d'abord à remercier Anne-Marie, la femme de ma vie, pour son amour et son support tout au long de ce travail.

Je remercie aussi ma famille, mes parents pour m'avoir donné la vie et de m'avoir laissé choisir mon chemin dans la vie, ma soeur Annie pour sa complicité incomparable.

Je tiens à remercier ma directrice, Brigitte Jaumard et mes codirecteurs Michel Perrier et Gilles Savard pour leur support financier, les discutions lors des nombreuses réunions et leurs encouragements.

Je remercie aussi tous les collègues du GERAD pour leur amitié, spécialement Sébastien qui m'a permis d'utiliser sa librairie INTERPLEX, Pascal pour ses nombreux conseils techniques et Têkogan pour sa patience et sa bonne humeur.

RÉSUMÉ

Le sujet traité dans ce mémoire est l'optimisation de la contrôlabilité des procédés, en particulier le procédé de lavage de la pâte Kraft. Généralement, l'étude de la contrôlabilité du procédé est une étape ultérieure à l'optimisation des coûts d'opérations. L'objectif de ce mémoire est l'élaboration et l'application d'une méthode qui permet d'intégrer l'optimisation du procédé et l'analyse de contrôlabilité dans une même procédure.

En utilisant une mesure de la contrôlabilité appelée la matrice des gains relatifs (RGA), nous avons développé un algorithme d'optimisation de la contrôlabilité (AOC). Dans le cadre de ce projet, nous avons traité deux procédés MIMO, chacun avec deux variables d'entrées et deux variables de sorties. Ainsi, l'algorithme AOC a été appliqué tout d'abord à un problème test et ensuite à un modèle du procédé de lavage de la pâte à papier Kraft. Ce dernier a été exprimé comme un problème d'optimisation sous contraintes bilinéaires. Le mémoire contient deux modèles du procédé de lavage, un premier modèle de base et un modèle plus adapté à l'optimisation. Pour résoudre ce problème, une méthode de décomposition de Benders généralisée a été implantée.

Nous avons obtenu des résultats de calcul satisfaisant, en particulier, ces résul-

sultats montrent que l'AOC est un algorithme prometteur pour l'optimisation de la contrôlabilité.

ABSTRACT

The main subject of this thesis is the optimization of the process controllability and it's application to the kraft pulp washing process. Generally, the controllability analysis is done after the process has been optimized in terms of the operation costs.

The main objective of this thesis is to develop and experiment a method which integrates the process optimization and the controllability analysis into the same procedure. Using the Relative Gain Array (RGA) for the measurement of the controllability we developed a controllability optimization algorithm (AOC). We studied two multi-input multi-output (MIMO) processes, each with two input variables and two output variables. We first apply the AOC on a test problem then on a kraft pulp washing process model. The thesis contains two models of the kraft pulp washing process. Both models has been expressed as a bilinear optimization problem. The first model is a basic one, the other is more adapted to optimization. To solve these problems, a generalized Benders decomposition has been developed and implemented.

We have obtained satisfactory results in the computational experience. In particular, they show that the AOC is a promising algorithm for controllability analysis.

TABLE DES MATIÈRES

REMERCIEMENTS . . . . . **iv**

RÉSUMÉ . . . . . **v**

ABSTRACT . . . . . **vii**

TABLE DES MATIÈRES . . . . . **viii**

LISTE DES TABLEAUX . . . . . **x**

LISTE DES FIGURES . . . . . **xi**

LISTE DES ANNEXES . . . . . **xiii**

INTRODUCTION **1**

CHAPITRE 1 : REVUE DE LA LITTÉRATURE **3**

1.1 Contrôle et contrôlabilité 3

1.2 Le papier et le procédé de lavage de la pâte kraft 8

1.3 Optimisation non-linéaire non-convexe 15

CHAPITRE 2 : MODÉLISATION DU PROCÉDÉ DE LAVAGE . . **18**

2.1 Le modèle de base 19

CHAPITRE 3 : UNE MÉTHODE DE DÉCOMPOSITION EN OPTIMISATION GLOBALE **32**

3.1 Décomposition de Benders 33

3.2 Décomposition de Benders généralisée 41

3.3 GOP 45

3.4 Implantation 72

3.5 Améliorations du GOP 72

CHAPITRE 4 : ALGORITHME D'OPTIMISATION DE LA CONTRÔLABILITÉ **74**

4.1 Description de l'algorithme 75

4.2 Validation de l'algorithme 79

CHAPITRE 5 : RÉSULTATS DE CALCULS **85**

5.1 Validation et performance du GOP 85

5.2 Optimisation du procédé de lavage 87

5.3 Optimisation de la contrôlabilité du procédé de lavage (AOC) 89

5.3.1 Couplage des variables contrôlées et manipulées 89

5.3.2 Approximation par différences finies. 90

5.3.3 Implantation 93

5.3.4 Paramètres de l'AOC 93

5.3.5 Résultats de l'AOC 97

CHAPITRE 6 : DISCUSSION **104**

CONCLUSION **107**

ANNEXES **113**

LISTE DES TABLEAUX

5.1 Résultats du GOP sur des problèmes tests de la littérature 86

5.2 Sensibilité du GOP par rapport au point de départ sur le problème soland. 86

5.3 Sensibilité du GOP par rapport au point de départ sur le problème de laveur. 87

5.4 Cas de base 88

5.5 Sensibilité du modèle par rapport aux paramètres α et DR. 88

5.6 Différents points de départs pour l'AOC 94

5.7 Calibration des pas h et δ 95

LISTE DES FIGURES

1.1 Schéma du mélange eau chaude/eau froide 4
1.2 Schéma de la contrôlabilité 5
1.3 Schéma des boucles de contrôle 5
1.4 Schéma du procédé de fabrication de la pâte kraft 9
1.5 Principe de déplacement 10
1.6 Schéma d'un module de lavage 10
1.7 Variables impliquées dans un module de lavage 12
1.8 Mélangeurs à contre-courant idéaux 13

2.1 Un module de lavage 19
2.2 Schéma du procédé de lavage 19

3.1 Itération 1 55
3.2 Itération 2 56
3.3 Itération 3 59
3.4 Itération 3 60

4.1 Schéma du procédé de mélange 79
4.2 Résultats du procédé de mélange 84

5.1 Couplage des variables manipulées et contrôlées 90

5.2 Variation de la valeur de λ. 98

5.3 Résultats de l'AOC, point de départ 1. 99

5.4 Résultats de l'AOC, point de départ 2. 100

5.5 Résultats de l'AOC, point de départ 3. 101

5.6 Résultats de l'AOC, point de départ 4. 102

LISTE DES ANNEXES

A Reformulation du modèle **113**

B Lemme de Farkas **119**

C x-connectées **121**

INTRODUCTION

Une partie importante de l'ingénierie des procédés est le contrôle des procédés. Généralement, un procédé et ses conditions d'opération sont judicieusement choisis pour optimiser les coûts d'opération, la productivité ou la quantité de rejets. Une fois que tout est fixé, il faut ensuite choisir une stratégie de contrôle pour s'assurer que ce procédé conservera les mêmes conditions d'opération tout au long de son fonctionnement. Mais que ce passe-t-il si ce procédé est difficilement contrôlable, c'est-à-dire si la contrôlabilité est mauvaise ? Il sera peut-être très difficile de conserver les variables de sorties du procédé à leurs valeurs de consigne.

Le but de ce projet est de développer et d'appliquer une méthode qui permet d'intégrer l'optimisation du procédé et l'analyse de contrôlabilité en une même procédure. Cette procédure pourrait permettre de trouver des conditions d'opération qui combinent une bonne contrôlabilité et de faibles coûts d'opération. L'avantage d'une telle approche est que ces conditions d'opération seront beaucoup plus faciles à conserver lors du fonctionnement du procédé.

Pour atteindre ces objectif, nous avons développé une méthode pour évaluer un critère de contrôlabilité appelé la matrice des gains relatifs (RGA) en un point donné. En utilisant cette méthode, nous avons développé un algorithme d'optimisation de la contrôlabilité (AOC).

Cet algorithme a été appliqué avec succès à un problème test ainsi qu'à un modèle

du procédé de lavage de la pâte à papier kraft.

Ce mémoire contient tout d'abord une brève revue de la littérature des domaines impliqués dans ce projet, c'est-à-dire le contrôle, et la contrôlabilité, la fabrication du papier et le procédé de lavage et finalement l'optimisation de problèmes non-linéaires non-convexes. Par la suite, deux modèles mathématiques du procédé de lavage de la pâte kraft seront présentés. La description détaillé d'une méthode d'optimisation globale suivra cette partie. Ensuite, l'algorithme d'optimisation de la contrôlabilité (AOC) sera présenté ainsi qu'un exemple d'application. Finalement, Les résultats de calculs seront présentés et analysés.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Contrôle et contrôlabilité

La conception d'un système de contrôle de procédés constitue une discipline inhérente à l'ingénierie des procédés. Tous les procédés chimiques sont soumis à des perturbations. Ces perturbations peuvent venir de facteurs externes comme les conditions météorologiques ou des autres procédés qui interagissent entre eux. Certaines de ces perturbations peuvent devenir dangereuses si elles ne sont pas contrôlées rapidement. Par exemple, une réaction exothermique qui n'est pas contrôlée peut s'emballer et causer un accident grave. Le principal objectif d'un système de contrôle consiste à atténuer l'effet des perturbations.

Pour contrôler un procédé, l'industrie utilise plusieurs types de contrôleurs généralement basé sur le principe suivant : identifier le changement de la valeur d'une variable contrôlée, transmettre cette information à une composante physique qui appliquera la variation appropriée à une variable manipulée. Supposons, par exemple, que le procédé à contrôler est un mélange d'eau chaude et d'eau froide (la douche matinale) et qu'un contrôleur est chargé de conserver la température de la douche constante.

Ce contrôleur est associé à une vanne qui règle le débit d'eau chaude. Lorsque la température de l'eau diminue subitement, le contrôleur envoie un signal à la vanne pour augmenter le débit d'eau chaude afin d'ajuster la température de la douche à sa valeur de consigne.

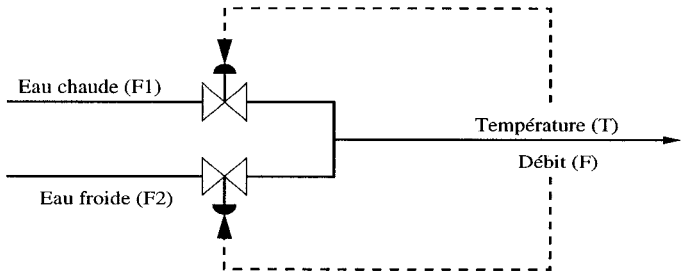


Figure 1.1 – Schéma du mélange eau chaude/eau froide

Certains procédés sont plus faciles à contrôler que d'autres. Mais comment savoir si un procédé est plus facilement contrôlable qu'un autre ? Comment savoir si la configuration d'un procédé le rendra facile à contrôler ? C'est ici que le concept de contrôlabilité entre en jeu. On appelle contrôlabilité d'un procédé sa capacité à atteindre de bonnes conditions de contrôle. Plus précisément, pour un procédé P avec les variables d'entrées (u) et les variables de sorties (y), la contrôlabilité de ce procédé est sa capacité à conserver les variables de sorties à leur point de consigne (r) malgré les perturbations (d) en utilisant les variables d'entrées (u) et des mesures appropriées. Il est bien important de comprendre que la contrôlabilité est indépendante de la stratégie de contrôle. Elle ne dépend que du procédé et de ses conditions d'opération.

Dans l'exemple de la douche, on peut se poser la question suivante : si l'eau chaude était à une température plus élevée, le procédé serait-il plus facilement contrôlable ? Pour répondre adéquatement à cette question, une mesure de la contrôlabilité est nécessaire. La matrice des gains relatifs (RGA) est l'outil qui a été utilisé dans le

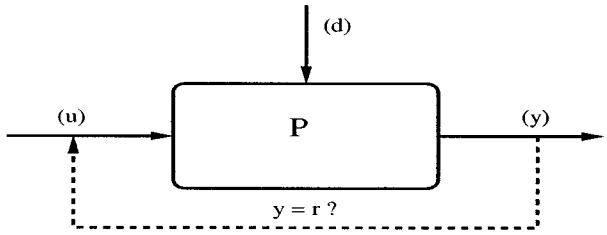


Figure 1.2 – Schéma de la contrôlabilité

présent projet pour caractériser la contrôlabilité. Cette mesure a été introduite par Bristol [7]. Plusieurs concepts sont nécessaires pour assimiler cette mesure, le plus important est la notion de boucle ouverte et de boucle fermée.

Considérons un procédé ayant n variables contrôlées et n variables manipulées. Il n'est pas nécessaire que ces deux ensembles de variable soit de même cardinalité. Pour simplifier le problème, cette supposition sera adoptée pour la suite du document. La matrice des fonctions de transfert de ce procédé est notée $G(s)$, les éléments de cette matrice seront notés $g_{ij}(s)$ où s représente la variable d'état. Pour les informations du procédé en régime permanent, les notations sont les mêmes, mais en fixant $s = 0$, c'est-à-dire $G(0) = G$ et $g_{ij}(0) = g_{ij}$. Les variables manipulées seront notées par $m_i, i = 1...n$ et les variables contrôlées par $c_j, j = 1...n$.

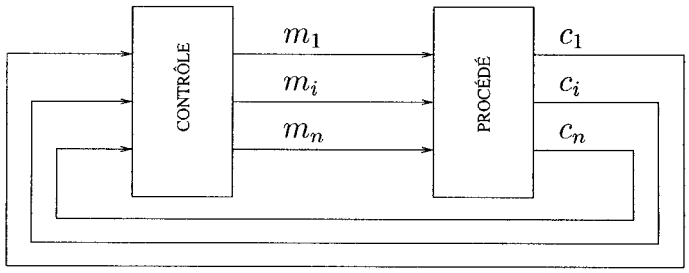


Figure 1.3 – Schéma des boucles de contrôle

Pour bien saisir le comportement d'un procédé, il faut connaître les variations

des variables contrôlées par rapport aux variables manipulées.

Ces variations sont appelées gains, le gain de la variable contrôlée c_j par rapport à la variable manipulée m_i est $\frac{\partial c_j}{\partial m_i}$. Bien entendu, les autres variables de sortie vont aussi subir des variations si elles ne sont pas contrôlées. C'est pourquoi il faut préciser si les contrôleurs associés à ces variables sont actifs. Dans ce cas, on dit que les boucles de contrôle sont fermées. Si les contrôleurs sont inactifs, les boucles de contrôle sont ouvertes. Le gain de la variable contrôlée c_j par rapport à la variable manipulée m_i en boucle ouverte est défini comme : $\left(\frac{\partial c_j}{\partial m_i}|_{\partial m_k=0,\forall k\neq i}\right) = g_{ij}$. D'un autre côté, le gain de la variable contrôlée c_j par rapport à la variable manipulée m_i en boucle fermée est $\left(\frac{\partial c_j}{\partial m_i}|_{\partial c_k=0,\forall k\neq j}\right) = \hat{g}_{ji}$ où $\hat{g}_{ji}$ représente ici le terme en position j, i de la matrice G^{-1}. La relation entre ces deux gains a été décrite entre autre par Grosdidier et Morari [15].

À l'aide de ces deux gains, on peut définir un indice de gains relatifs :

$$\lambda_{ij} = g_{ij}\hat{g}_{ji}. \tag{1.1}$$

L'ensemble de ces gains relatifs forme la matrice des gains relatifs ou **R**elative **G**ain **A**rray :

$$RGA = G \otimes G^{-1^T}. \tag{1.2}$$

Où $\otimes$ représente ici le produit termes à termes de deux matrices. Le RGA possède plusieurs propriétés intéressantes [15] :

- le RGA décrit l'interaction entre les variables contrôlées et les variables manipulées ;
- le RGA ne dépend pas de la mise à l'échelle des variables ;
- les permutations de lignes ou de colonnes dans la matrice G se répercutent sur

les mêmes lignes ou colonnes du RGA ;

- plus un procédé est facilement contrôlable, plus son RGA se rapproche de la matrice identité I ;
- la somme sur les lignes ou sur les colonnes du RGA est égale à un.

Le RGA permet de mesurer la contrôlabilité d'un procédé. Le RGA, tel que défini, utilise seulement des informations du procédé en régime permanent, mais il permet quand même de décrire certaines propriétés dynamiques du procédé. Zhu et Jutan [35] ont montré que les signes des éléments du RGA permettent de tirer une condition nécessaire à la stabilité du système. De plus, un lien clair a été établi entre la positivité des éléments du RGA, la positivité de l'index de Niederlinski [23] et l'intégrité du système [15, 35]. Le RGA est un outil très riche en information aussi bien en régime permanent que dynamique.

Pour un procédé simple, il est possible d'obtenir l'expression analytique de la matrice de transfert G. Le calcul du RGA peut alors se faire de façon analytique. Par contre, si on ne possède pas l'expression analytique de la matrice de transfert, on doit estimer les gains relatifs de façon numérique à l'aide des dérivées partielles des fonctions reliant les variables contrôlées et manipulées.

1.2 Le papier et le procédé de lavage de la pâte kraft

Les premiers hommes gravaient des signes sur la roche et les os, puis sur l'argile et le bois. Ce sont les égyptiens qui les premiers ont mis de côté ces supports encombrants pour utiliser des rouleaux de papyrus. Ces rouleaux étaient faits avec des roseaux provenant des marécages du Nil appelé *Cyperus papyrus*. C'est de ce nom que provient le mot papier. Mais ce sont les Chinois qui ont découvert le papier. Cette découverte semble remonter à l'an 105 après J.C., sous le règne de l'empereur Hoti, et est généralement attribuée au ministre de l'agriculture Tsaï-Lun. À cette époque, on fabriquait le papier à partir de vieux chiffons et d'écorces réduits en bouillies. De nos jours, le papier est fabriqué à grande échelle avec d'immenses machines à partir du bois. Les machines à papier modernes peuvent mesurer jusqu'à 100 mètres de longueur, et fabriquer des papiers de grandes largeurs à une vitesse de l'ordre de plusieurs milliers de mètres à la minute pour obtenir au delà de 1000 tonnes de papier en bobines par jour [8].

Avant de passer dans ces machines modernes, le bois doit tout d'abord être transformé en pâte à papier, c'est-à-dire que l'on doit séparer les fibres de cellulose qui formeront le papier de la lignine qui retient les fibres ensembles. Deux méthodes sont utilisées pour effectuer cette transformation. La méthode thermomécanique et la méthode chimique. La méthode thermomécanique consiste à séparer les fibres en brisant physiquement les liens qui les unissent. Cette méthode est relativement simple, mais elle est très gourmande en énergie et de plus, le procédé endommage les fibres. Les fibres endommagées sont plus courtes, ce qui rend le papier moins résistant. La méthode chimique quant à elle, utilise des produits ayant la capacité de dissoudre la

lignine du bois sans endommager les fibres de cellulose.

Le procédé kraft est un procédé chimique de fabrication de la pâte et est schématisé à la figure 1.4. Dans le procédé kraft, les billes de bois sont d'abord déchiquetées en copeaux. Par la suite, ces copeaux sont déversés dans un grand bassin de cuisson avec la liqueur de cuisson (*cooking liquor*) constituée d'eau et de produits chimiques. Généralement on ajoute à l'eau de l'hydroxyde de sodium ($NaOH$) et du sulfure de sodium (Na_2S). Il est important de ne dissoudre que la lignine et d'épargner les fibres de cellulose qui formeront le papier. Une fois la lignine dissoute, on obtient une pâte relativement homogène qui contient de l'eau, des fibres de cellulose, des résidus de lignine et des produits chimiques. On doit séparer les fibres de cellulose des autres constituants de la pâte. Tous ces autres constituants sont regroupés sous l'appellation solides dissous. Cette opération s'appelle le lavage de la pâte. C'est à cette étape du procédé de fabrication de la pâte kraft que se situe le présent projet.

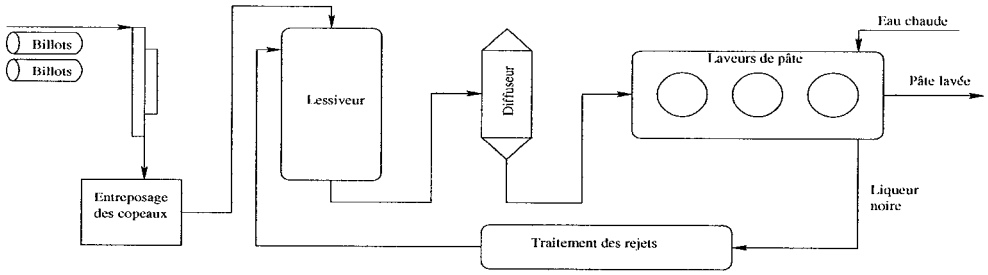


Figure 1.4 – Schéma du procédé de fabrication de la pâte kraft

Le retrait des solides dissous (résidus de lignine et produits chimiques) est effectué par le principe de déplacement. L'eau propre provenant des douches traverse une couche de pâte et retire une certaine portion des solides dissous. Un système de pompe à vide force l'eau à traverser la couche de pâte (Figure 1.5).

La structure la plus commune pour appliquer ce principe est une série de tam-

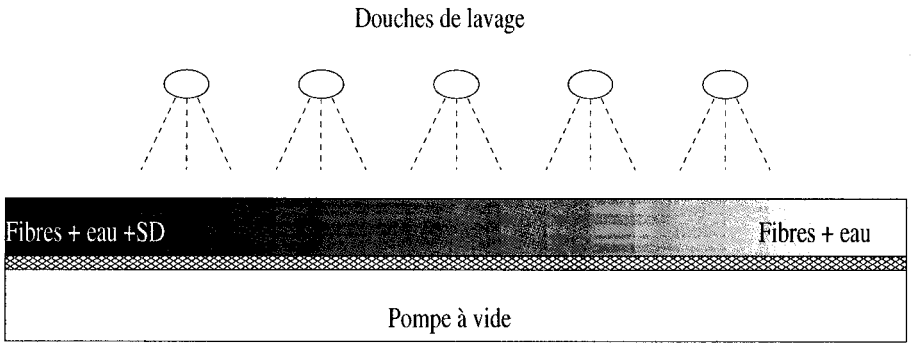


Figure 1.5 – Principe de déplacement

bours rotatifs. Une pompe crée un vide à l'intérieur du tambour qui tourne dans une cuve pleine de pâte qui a été préalablement diluée (Figure 1.6). La couche de pâte qui colle au tambour est arrosée avec l'eau de lavage, puis cette couche est raclée et envoyée au mélangeur, puis au tambour suivant. Le bassin de mélange et le tambour rotatif forment un module de lavage.

Généralement, une série de deux à quatre modules de lavages est utilisée pour

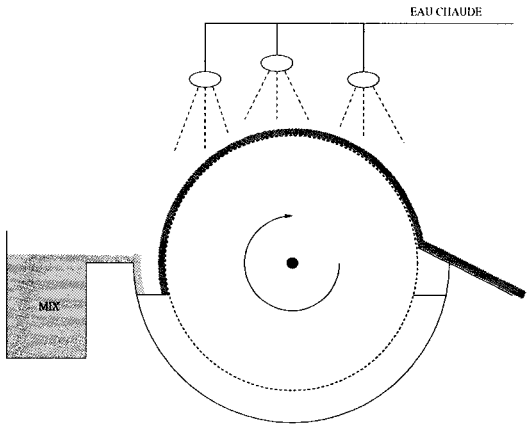


Figure 1.6 – Schéma d'un module de lavage

former le système complet. Ces modules sont disposés à contre-courant, c'est-à-dire que la solution d'eau et de solides dissous qui traverse le tambour est réutilisée comme eau de lavage pour les modules de lavage précédents, elle est aussi utilisée pour diluer

la pâte. De cette façon, on utilise de l'eau fraîche seulement à la dernière étape, là où la pâte est la plus propre. La solution de solides dissous du premier module de lavage est envoyée vers une série de procédés pour récupérer les produits chimiques. Cette solution sera tout d'abord concentrée à l'aide d'évaporateur, puis brûlée pour utiliser l'énergie des molécules organiques de la solution. Finalement, les produits chimiques seront récupérés des cendres, puis acheminés de nouveau vers le bassin de cuisson.

L'étape du lavage de la pâte kraft est une étape clé du procédé de fabrication du papier. La qualité de la pâte lavée influencera les étapes suivantes dont le blanchiment qui est sensible aux différents produits contenus dans la pâte. En assurant une pâte décontaminée, on aura un blanchiment plus facile et plus économique. On souhaite donc une pâte aussi propre que possible. D'autant plus que les produits chimiques qui ne sont pas récupérés dans le procédé de lavage sont perdus et devront être renouvelés, ce qui représente des coûts importants. Ceci est sans compter que pour avoir une pâte plus propre, on doit augmenter la consommation d'eau fraîche. La consommation en eau fraîche est coûteuse car elle influencera directement la concentration de la liqueur noire qui est envoyée aux évaporateurs. Plus cette liqueur noire est diluée, plus l'évaporation sera gourmande énergétiquement et donc coûteuse. Il y a ici un compromis à faire entre la propreté de la pâte lavée et la consommation d'eau fraîche.

Mesures d'efficacité

Depuis longtemps déjà, les chercheurs ont tenté de décrire le fonctionnement des systèmes de lavage de pâte kraft à l'aide de différents facteurs. Nous présenterons ici trois mesures ainsi que leurs avantages et leurs inconvénients.

Un de ces facteurs est appelé le facteur de dilution (*dilution factor*) ou DF. Il représente le surplus d'eau de lavage, soit l'eau provenant des douches notée F_d moins l'eau qui quitte le module avec la pâte qui va au prochain module notée F_c :

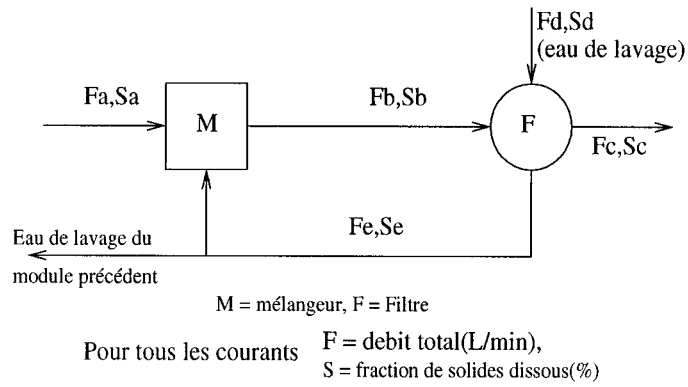


Figure 1.7 – Variables impliquées dans un module de lavage

$$DF = F_d - F_c. \tag{1.3}$$

Le facteur de dilution est très facile à calculer et à intégrer dans des modèles mathématiques, par contre il dépend de l'unité et de la mise à l'échelle des variables F_d et F_c. De plus, cette information est insuffisante pour bien caractériser l'efficacité d'un module de lavage et encore moins du système au complet. Par la suite, d'autres facteurs plus détaillés ont été proposés [30]. Le rapport de déplacement (*Displacement Ratio*) ou DR représente la diminution réelle des solides dissous par rapport à la diminution maximale théorique :

$$DR = \frac{S_a - S_c}{S_a - S_d} \tag{1.4}$$

Où S_a, S_c et S_d représentent respectivement les fractions de solides dissous des courant a, c et d. Le rapport de déplacement est un facteur sans unité qui se situe toujours entre zéro et un. Le pourcentage de solides dissous dans le courant c, sera toujours plus grand que celui dans le courant d. Contrairement au facteur de dilution, le rapport de déplacement ne dépend pas de la mise à l'échelle des variables. Le rapport de déplacement est très utile, car il donne vraiment une mesure de l'efficacité de lavage

d'un module. Il est aussi très simple à calculer à partir des données brutes provenant du procédé.

Le troisième Le facteur d'efficacité Norden (NEF) est un autre facteur plus complet [25]. Le NEF représente le nombre de mélangeurs à contre-courant idéaux nécessaires pour obtenir la même efficacité de lavage que le module étudiée (Figure 1.8).

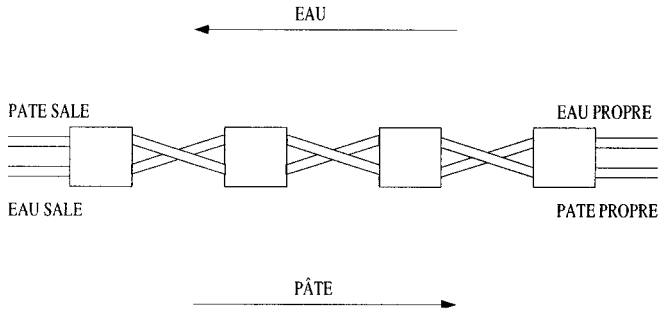


Figure 1.8 – Mélangeurs à contre-courant idéaux

Le NEF est définit comme :

$$NEF_i = \frac{\ln\left(\frac{F_A(S_A - S_E)}{F_C(S_C - S_D)}\right)}{\ln\left(\frac{F_D}{F_C}\right)} \tag{1.5}$$

Cet outil est très pratique car il est moins sensible aux changements du procédé que les précédents. Il est très utilisé en industrie et en simulation. Il est possible de simplifier la formulation du NEF en définissant WR, le ratio de liqueur de lavage comme le rapport du débit de liqueur entrant dans le module et du débit de liqueur sortant avec la pâte :

$$WR_i = \frac{F_D}{F_C}. \tag{1.6}$$

Une fois que le NEF a été calculé pour chacun des modules de lavage, il est possible de calculer un facteur global qui permet de caractériser l'efficacité du système au

complet :

$$E = \frac{\sum_{i=1}^{N} NEF_i \ln WR_i}{\ln WR_N}. \tag{1.7}$$

La formule du NEF est toutefois beaucoup plus complexe que les mesures précédentes, ce qui rend cette mesure difficile à utiliser.

Les systèmes de lavage de plusieurs stages à contre-courant sont des procédés complexes et difficiles à contrôler. Des simulations dynamiques ont été utilisées pour bien comprendre leurs comportements [32]. Ces simulations ont permis de proposer différentes stratégies de contrôle. Certaines de ces stratégies ont été implantées dans des moulins à papier existants et ont démontré leur efficacité [24]. Ces études ont démontré que la dynamique des systèmes de lavage de pâte est principalement affectée par le contrôle des bassins d'entreposage. L'effet des modifications des niveaux de ces bassins se fait sentir seulement après plusieurs minutes contrairement aux modifications sur les autres paramètres du procédé.

1.3 Optimisation non-linéaire non-convexe

La résolution de problèmes d'optimisation forme une part importante de l'ingénierie et des sciences appliquées. Plusieurs algorithmes ont été développés pour trouver des optima locaux à ces problèmes faute de pouvoir trouver les optima globaux à cause de la complexité de ces problèmes. Toutefois, si les problèmes possèdent plusieurs minima ou maxima locaux, ces algorithmes sont la plupart du temps très sensibles au choix du point de départ. Ce type de problème est très difficile à résoudre de façon globale. L'optimisation globale s'intéresse à ces problèmes et aux stratégies pour caractériser ces optima, pour trouver des bornes valides et trouver des solutions exactes ou approchées [18].

Il a d'ailleurs été montré que l'optimisation globale de problèmes non-linéaires non-convexes avec contraintes était un problème NP-difficile [22]. De même, la résolution des problèmes quadratiques à contraintes quadratiques est aussi NP-difficile [29]. La puissance des ordinateurs actuels permet de développer des algorithmes complexes d'optimisation globale. Toutefois, dans le cas général des problèmes non-linéaires non-convexes sans structures particulières, peu d'algorithmes permettent de trouver la solution globale du problème. De plus, ces algorithmes sont limités à des problèmes de petite taille.

L'étude des problèmes quadratiques et bilinéaires a permis le développement

d'algorithmes relativement efficace. Un problème quadratique se formule comme suit :

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_ix_j + \sum_{i=1}^{n} q_ix_i + q_0 \\
& \sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij}^k x_ix_j + \sum_{i=1}^{n} r_i^k x_i + r_0^k \leq 0, \quad k = 1, 2, ..., m \\
& x_i \in \mathbb{R}, \quad i = 1, 2, ..., n
\end{aligned}$$

et un problème bilinéaire ainsi :

$$\begin{aligned}
\min_{x,y} \quad & F(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{p} c_{ij}x_iy_j + \sum_{i=1}^{n} c_i'x_i + \sum_{j=1}^{p} c_j''y_j + c_0 \\
& \sum_{i=1}^{n}\sum_{j=1}^{p} a_{ij}^k x_iy_j + \sum_{i=1}^{n} a_i'^k x_i + \sum_{j=1}^{p} a_j''^k y_j + a_0^k, \quad k = 1, 2, ..., m \\
& x_i \in \mathbb{R}, \quad i = 1, 2, ..., n \\
& y_j \in \mathbb{R}, \quad j = 1, 2, ..., p.
\end{aligned}$$

Il a été démontré que les problèmes quadratiques et les problèmes bilinéaires sont équivalent du point de vue de la complexité de calcul [26]. Plusieurs algorithmes exacts ont été développés pour résoudre les problèmes de cette classe. Parmi les différentes stratégies de résolution, on peut identifier plusieurs catégories : les méthodes de branchements ou de partition [33], les méthodes de plan coupant [18], les méthodes de décomposition primal-dual [6] et les méthodes de reformulation-linéarisations [5].

Les méthodes heuristiques sont généralement formées d'une méthode de recherche locale couplée avec une stratégie pour aiguiller la recherche locale vers un meilleur optimum.

La catégorie des méthodes heuristiques regroupe principalement les méta-heuristiques évolutives : recherche tabou [13], recherche à voisinage variable [27], algorithmes gé-

nétiques [34] et autres.

La méthode d'optimisation globale utilisée dans ce projet est une méthode exacte du type décomposition primal-dual. Cette méthode sera détaillée à la section ??.

CHAPITRE 2

MODÉLISATION DU PROCÉDÉ DE LAVAGE DE LA PÂTE KRAFT

Dans ce chapitre, deux modèles du procédé de lavage de la pâte kraft seront présentés. Tout d'abord, un modèle plus intuitif qui est issu des principes physiques en jeu et qui est souvent utilisé dans les travaux de simulation ; ensuite une réécriture équivalente de ce modèle qui est mieux adapté à la méthode d'optimisation utilisée. Ces deux versions du modèle ont aussi été étudiés dans les travaux de Abdelmoula [1].

Comme il a été décrit dans le chapitre précédent, le procédé de lavage kraft est constitué de plusieurs modules de lavage fonctionnant en série et/ou en parallèle. Dans le modèle, chaque module sera constitué de deux parties, un mélangeur et un filtre, voir la figure (2.1). Le procédé qui a été modélisé ici contient trois modules de lavage en série, voir la figure (2.2).

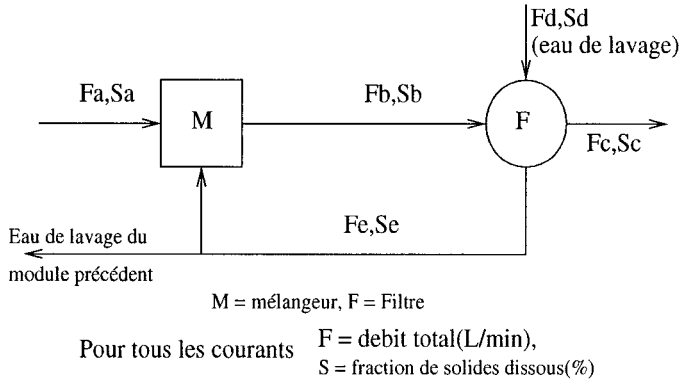


Figure 2.1 – Un module de lavage

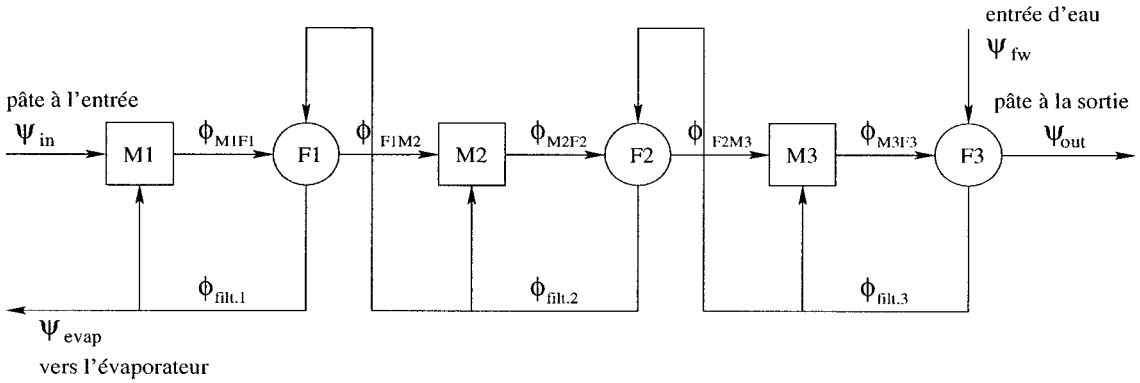


Figure 2.2 – Schéma du procédé de lavage

2.1 Le modèle de base

Le modèle de base considère les débits des différents courants et la composition de chaque courant. Les bilans totaux, les bilan partiels et les contraintes d'efficacité constituent la grande partie des équations du modèle. Le modèle intuitif est simple et facile à comprendre, mais il est difficile à résoudre efficacement avec les outils d'optimisation.

Les variables du modèle

Les variables en jeux sont les débits des courants, notés ϕ_{ij} et les proportions des différentes composantes dans ces courants notées $f^s\phi_{ij}$, $f^w\phi_{ij}$ et $f^f\phi_{ij}$ respectivement pour les solides dissous, l'eau et les fibres. Les courants d'entrée et de sortie du procédé sont notés différemment des courants internes. Soient Ψ_{in} le débit de pâte à l'entrée, Ψ_{out} le débit de pâte à la sortie, Ψ_{evap} le filtrat qui se dirige vers le système d'évaporation et finalement Ψ_{fw} le débit d'eau de lavage en entrée.

Les filtrats de chaque module de lavage sont notés $\phi_{filt.i}(i = 1, 2, 3)$, ils sont divisés en deux selon un rapport à déterminé $\alpha_i(i = 1, 2, 3)$, une partie étant utilisée pour diluer la pâte provenant du précédent module et l'autre partie étant utilisée comme eau de lavage pour le module précédent.

Toutes ces variables sont positives et les variables de proportion sont inférieures à un par définition.

Les paramètres du modèle

Plusieurs paramètres permettent d'ajuster la performance du procédé. Le choix de la valeur de ces paramètres est important car ils donnent une sens au modèle. Voici la liste des paramètres ainsi que leurs signification.

- $DR_i(i = 1, 2, 3)$ un paramètre d'efficacité. Le critère d'efficacité utilisé ici est le ratio de déplacement (*Displacement Ratio*) ou DR, voir la formule (1.4). Ces paramètres notés se situent toujours entre zéro et un et ils représentent la diminution réelle des solides dissous par rapport au maximum théorique. En pratique, ces paramètres se situent autour de 0.80.
- $\alpha_i(i = 1, 2, 3)$ représentent la proportion de filtrat utilisé pour la dilution dans chacun des modules. Dans un système de lavage standard, ces proportions se

situent entre 0.80 et 0.90.

- r représente la proportion des solides dissous qui doit être récupérée par le système de lavage. Généralement, on demande au procédé de récupérer environ 95% des solides utilisés à l'entrée.
- $\underline{f^s_{out}}$ représente la propreté minimum de la pâte à la sortie du système de lavage.

Les contraintes du modèle

Les équations du modèle découlent directement des principes physiques qui sont en jeu dans ce procédé. Pour commencer, les bilans matières doivent être respectés pour toutes les composantes (mélangeurs et filtres) du procédé. Ce qui nous donne une série d'équations de bilan totaux. Pour les bilans liés aux des mélangeurs on obtient :

bilan matière M1 $$\Psi_{in} + \alpha_1\phi_{filt.1} - \phi_{M1F1} = 0 \tag{2.1}$$

bilan matière M2 $$\phi_{F1M2} + \alpha_2\phi_{filt.2} - \phi_{M2F2} = 0 \tag{2.2}$$

bilan matière M3 $$\phi_{F2M3} + \alpha_3\phi_{filt.3} - \phi_{M3F3} = 0. \tag{2.3}$$

Pour les bilans liés aux filtres on obtient

bilan matière F1 $$\phi_{M1F1} + (1-\alpha_2)\phi_{filt.2} - \phi_{F1M2} - \phi_{filt.1} = 0 \tag{2.4}$$

bilan matière F2 $$\phi_{M2F2} + (1-\alpha_3)\phi_{filt.3} - \phi_{F2M3} - \phi_{filt.2} = 0 \tag{2.5}$$

bilan matière F3 $$\phi_{M3F3} + \Psi_{fw} - \Psi_{out} - \phi_{filt.3} = 0. \tag{2.6}$$

En plus de respecter les bilans totaux, les variables doivent respecter les contraintes de bilans pour chaque composante des courants. Ce qui nous donne une série d'équations

de bilan partiel. Pour les bilans partiels concernant les fibres on obtient :

$$f^f\Psi_{in}\cdot\Psi_{in}+f^f\phi_{filt.1}\cdot\alpha_1\cdot\phi_{filt.1}-f^f\phi_{M1F1}\cdot\phi_{M1F1}=0 \tag{2.7}$$

$$f^f\phi_{F1M2}\cdot\phi_{F1M2}+f^f\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^f\phi_{M2F2}\cdot\phi_{M2F2}=0 \tag{2.8}$$

$$f^f\phi_{F2M3}\cdot\phi_{F2M3}+f^f\phi_{filt.3}\cdot\alpha_3\cdot\phi_{filt.3}-f^f\phi_{M3F3}\cdot\phi_{M3F3}=0 \tag{2.9}$$

$$f^f\phi_{M1F1}\cdot\phi_{M1F1}+f^f\phi_{filt.2}\cdot(1-\alpha_2)\cdot\phi_{filt.2}=f^f\phi_{F1M2}\cdot\phi_{F1M2}+f^f\phi_{filt.1}\cdot\phi_{filt.1} \tag{2.10}$$

$$f^f\phi_{M2F2}\cdot\phi_{M2F2}+f^f\phi_{filt.3}\cdot(1-\alpha_3)\cdot\phi_{filt.3}=f^f\phi_{F2M3}\cdot\phi_{F2M3}+f^f\phi_{filt.2}\cdot\phi_{filt.2} \tag{2.11}$$

$$f^f\phi_{M3F3}\cdot\phi_{M3F3}+f^f\Psi_{fw}\cdot\Psi_{fw}=f^f\Psi_{out}\cdot\Psi_{out}+f^f\phi_{filt.3}\cdot\phi_{filt.3}. \tag{2.12}$$

Pour les bilans partiels concernant les solides dissous on obtient :

$$f^s\Psi_{in}\cdot\Psi_{in}+f^s\phi_{filt.1}\cdot\alpha_1\cdot\phi_{filt.1}-f^s\phi_{M1F1}\cdot\phi_{M1F1}=0 \tag{2.13}$$

$$f^s\phi_{F1M2}\cdot\phi_{F1M2}+f^s\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^s\phi_{M2F2}\cdot\phi_{M2F2}=0 \tag{2.14}$$

$$f^s\phi_{F2M3}\cdot\phi_{F2M3}+f^s\phi_{filt.3}\cdot\alpha_3\cdot\phi_{filt.3}-f^s\phi_{M3F3}\cdot\phi_{M3F3}=0 \tag{2.15}$$

$$f^s\phi_{M1F1}\cdot\phi_{M1F1}+f^s\phi_{filt.2}\cdot(1-\alpha_2)\cdot\phi_{filt.2}=f^s\phi_{F1M2}\cdot\phi_{F1M2}+f^s\phi_{filt.1}\cdot\phi_{filt.1} \tag{2.16}$$

$$f^s\phi_{M2F2}\cdot\phi_{M2F2}+f^s\phi_{filt.3}\cdot(1-\alpha_3)\cdot\phi_{filt.3}=f^s\phi_{F2M3}\cdot\phi_{F2M3}+f^s\phi_{filt.2}\cdot\phi_{filt.2} \tag{2.17}$$

$$f^s\phi_{M3F3}\cdot\phi_{M3F3}+f^s\Psi_{fw}\cdot\Psi_{fw}=f^s\Psi_{out}\cdot\Psi_{out}+f^s\phi_{filt.3}\cdot\phi_{filt.3}. \tag{2.18}$$

Finalement, pour les bilans partiels concernant l'eau on obtient :

$$f^w\Psi_{in}\cdot\Psi_{in}+f^w\phi_{filt.1}\cdot\alpha_1\cdot\phi_{filt.1}-f^w\phi_{M1F1}\cdot\phi_{M1F1}=0 \tag{2.19}$$

$$f^w\phi_{F1M2}\cdot\phi_{F1M2}+f^w\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^w\phi_{M2F2}\cdot\phi_{M2F2}=0 \tag{2.20}$$

$$f^w\phi_{F2M3}\cdot\phi_{F2M3}+f^w\phi_{filt.3}\cdot\alpha_3\cdot\phi_{filt.3}-f^w\phi_{M3F3}\cdot\phi_{M3F3}=0 \tag{2.21}$$

$$f^w\phi_{M1F1}\cdot\phi_{M1F1}+f^w\phi_{filt.2}\cdot(1-\alpha_2)\cdot\phi_{filt.2}=f^w\phi_{F1M2}\cdot\phi_{F1M2}+f^w\phi_{filt.1}\cdot\phi_{filt.1} \tag{2.22}$$

$$f^w\phi_{M2F2}\cdot\phi_{M2F2}+f^w\phi_{filt.3}\cdot(1-\alpha_3)\cdot\phi_{filt.3}=f^w\phi_{F2M3}\cdot\phi_{F2M3}+f^w\phi_{filt.2}\cdot\phi_{filt.2} \tag{2.23}$$

$$f^w\phi_{M3F3}\cdot\phi_{M3F3}+f^w\Psi_{fw}\cdot\Psi_{fw}=f^w\Psi_{out}\cdot\Psi_{out}+f^w\phi_{filt.3}\cdot\phi_{filt.3}. \tag{2.24}$$

Chaque courant n'est formé que d'eau, de fibres et de solides dissous, on doit

donc inclure des équations qui garantissent la complémentarité de ces fractions.

$$f^s\phi_{ij} + f^w\phi_{ij} + f^f\phi_{ij} = 1 \quad \forall\phi_{ij} \text{ dans le modèle.} \tag{2.25}$$

Il reste ensuite les inégalités qui règlent la performance du système de lavage. Pour commencer, un seuil de propreté minimum de la pâte à la sortie doit être garanti. Ce qui donne une borne sur la fraction de solides dissous à la sortie :

$$f^s\Psi_{out} \leq \overline{f^s\Psi_{out}}. \tag{2.26}$$

Il y a aussi le fait qu'une certaine proportion des produits chimiques doit être récupérée et renvoyée vers l'évaporateur. Ce qui donne une borne sur la fraction de solides dissous envoyée vers le système d'évaporateurs :

$$f^s\Psi_{évap} \geq \underline{f^s\Psi_{évap}}. \tag{2.27}$$

L'efficacité des modules de lavage est représentée par les valeurs des *Displacement ratio* ou DR. L'efficacité est bornée inférieurement et supérieurement comme suit :

$$\underline{DR_i} \leq DR_i \leq \overline{DR_i}, \quad i = 1, 2, 3. \tag{2.28}$$

Les bornes sur les variables

Au-delà des équations qui interviennent dans le procédé, certaines variables doivent être bornées pour refléter le fonctionnement réel du procédé. Les variables de débit doivent être bornées inférieurement et supérieurement, car la dimension des différents conduits est fixée et donc le débit dans ces conduits est limité. Les bornes inférieures

sur les débits sont présentes pour assurer des résultats réalistes. La consistance de la pâte ou la fraction de fibres doit aussi être bornée supérieurement, car si la pâte est trop consistante, elle ne s'écoulera plus dans les conduits. La fraction de fibres dans les débits de filtrats est bornée inférieurement pour modéliser le fait qu'en pratique, les filtres laissent passer une petite fraction de fibres.

Le problème d'optimisation

Comme il a été expliqué au chapitre 1, le coût d'opération du procédé de lavage de la pâte kraft est surtout influencé par la quantité d'eau fraîche utilisée et par la propreté de la pâte lavée. La minimisation des coûts d'opération passe donc par la minimisation des variables Ψ_{fw} et $f^s\Psi_{out}$.
L'importance accordée à ces variables est reflétée dans les poids ω_{fw} et ω_{out}. Ce qui donne la fonction objectif du modèle de base :

$$\omega_{fw} \cdot \Psi_{fw} + \omega_{out} \cdot f^s\Psi_{out}. \tag{2.29}$$

Ceci termine la description du modèle de base qui se base directement sur les principes physiques en jeu. Ce modèle est très facile à comprendre, mais il possède toutefois plusieurs inconvénients. D'abord, plusieurs équations sont redondantes. En effet les équations de bilans totaux (2.1) à (2.6) et de bilans partiels sur les fibres et sur les solides dissous (2.7) à (2.18) avec les équations de complémentarité (2.25) suffisent pour générer les équations de bilans partiels sur l'eau (2.19) à (2.24).
Ensuite, la majorité des équations de ce modèle font intervenir des termes bilinéaires, ce qui rend le problème difficile à résoudre. De plus, presque toutes les variables sont impliquées dans ces termes bilinéaires.

2.2 Le modèle adapté

Il est possible de simplifier la représentation mathématique du modèle. D'abord, les contraintes de bilans partiels sur l'eau sont redondantes et seront donc retirées du problème. Ensuite, un simple changement de variables permettra de réduire énormément la difficulté du problème. Au lieu de considérer les débits et les fractions de chaque courant, considérons maintenant les débits totaux et les débits partiels, et effectuons le changement de variables suivant :

$$\left.\begin{aligned} f^f\phi_{ij}\cdot\phi_{ij} &= \phi^f_{ij} \\ f^s\phi_{ij}\cdot\phi_{ij} &= \phi^s_{ij} \\ f^w\phi_{ij}\cdot\phi_{ij} &= \phi^w_{ij} \end{aligned}\right\} \quad \forall\phi_{ij} \text{ dans le modèle.} \tag{2.30}$$

Cela permet de convertir presque toutes les contraintes bilinéaires en contraintes linéaires. Par exemple la contrainte de bilan partiel du premier mélangeur sur les fibres :

$$f^f\Psi_{in}\cdot\Psi_{in} + f^f\phi_{filt.1}\cdot\alpha_1\cdot\phi_{filt.1} - f^f\phi_{M1F1}\cdot\phi_{M1F1} = 0,$$

devient linéaire :

$$\Psi^f_{in} + \alpha_1\cdot\phi^f_{F1M1} - \phi^f_{M1F1} = 0.$$

Les contraintes de complémentarité seront aussi quelque peu modifiées. La contrainte

$$f^f\phi_{ij} + f^s\phi_{ij} + f^w\phi_{ij} = 1 \qquad \forall\phi_{ij} \text{ dans le modèle,}$$

est remplacé par

$$\phi_{ij}^{f} + \phi_{ij}^{s} + \phi_{ij}^{w} = \phi_{ij} \qquad \forall \phi_{ij} \text{ dans le modèle.}$$

Malheureusement, on ne peut pas éliminer la bilinéarité du problème par un simple changement de variable. Les contraintes de bornes sur les DR qui étaient auparavant linéaires devient plus complexes. Prenons par exemple l'efficacité du second module avant le changement de variable :

$$\underline{DR_2} \leq DR_2 = \frac{f^s\phi_{F1M2} - f^s\phi_{F2M3}}{f^s\phi_{F1M2} - f^s\phi_{filt.3}}.$$

Après le changement de variables décrit dans (2.30), cette inéquation devient :

$$\underline{DR_2} \leq DR_2 = \frac{\frac{\phi_{F1M2}^s}{\phi_{F1M2}} - \frac{\phi_{F2M3}^s}{\phi_{F2M3}}}{\frac{\phi_{F1M2}^s}{\phi_{F1M2}} - \frac{\phi_{filt.3}^s}{\phi_{filt.3}}}.$$

Sous cette forme, cette contrainte est beaucoup trop complexe, il est préférable de la conserver sous sa forme initiale. Si on conserve cette équation sous sa forme initiale, il faut conserver les variables $f^s\phi_{F1M2}, f^s\phi_{F2M3}$ et $f^s\phi_{filt.3}$ qui interviennent dans l'expression de DR_2 dans le modèle. De plus, pour garantir la cohérence du modèle, il faut inclure pour chaque variable $f^s\phi_{ij}$ conservée, une équation du type :

$$f^s\phi_{ij} \cdot \phi_{ij} = \phi_{ij}^s.$$

Ces équations seront dorénavant les seules équations bilinéaires du modèle.

Les contraintes du modèle

En résumé, le modèle adapté contiendra comme contraintes les équations de bilans totaux autour des mélangeurs et autour des filtres :

$$\Psi_{in} + \alpha_1 \phi_{filt.1} - \phi_{M1F1} = 0 \tag{2.31}$$

$$\phi_{F1M2} + \alpha_2 \phi_{filt.2} - \phi_{M2F2} = 0 \tag{2.32}$$

$$\phi_{F2M3} + \alpha_3 \phi_{filt.3} - \phi_{M3F3} = 0 \tag{2.33}$$

$$\phi_{M1F1} + (1 - \alpha_2)\phi_{filt.2} - \phi_{F1M2} - \phi_{filt.1} = 0 \tag{2.34}$$

$$\phi_{M2F2} + (1 - \alpha_3)\phi_{filt.3} - \phi_{F2M3} - \phi_{filt.2} = 0 \tag{2.35}$$

$$\phi_{M3F3} + \Psi_{fw} - \Psi_{out} - \phi_{filt.3} = 0. \tag{2.36}$$

Il contiendra aussi les équations de bilans partiels sur les solides dissous et les fibres :

$$\Psi^s_{in} + \alpha_1 \phi^s_{filt.1} - \phi^s_{M1F1} = 0 \tag{2.37}$$

$$\phi^s_{F1M2} + \alpha_2 \phi^s_{filt.2} - \phi^s_{M2F2} = 0 \tag{2.38}$$

$$\phi^s_{F2M3} + \alpha_3 \phi^s_{filt.3} - \phi^s_{M3F3} = 0 \tag{2.39}$$

$$\phi^s_{M1F1} + (1 - \alpha_2)\phi^s_{filt.2} - \phi^s_{F1M2} - \phi^s_{filt.1} = 0 \tag{2.40}$$

$$\phi^s_{M2F2} + (1 - \alpha_3)\phi^s_{filt.3} - \phi^s_{F2M3} - \phi^s_{filt.2} = 0 \tag{2.41}$$

$$\phi^s_{M3F3} + \Psi^s_{fw} - \Psi^s_{out} - \phi^s_{filt.3} = 0 \tag{2.42}$$

$$\Psi^f_{in} + \alpha_1 \phi^f_{filt.1} - \phi^f_{M1F1} = 0 \tag{2.43}$$

$$\phi^f_{F1M2} + \alpha_2 \phi^f_{filt.2} - \phi^f_{M2F2} = 0 \tag{2.44}$$

$$\phi^f_{F2M3} + \alpha_3 \phi^f_{filt.3} - \phi^f_{M3F3} = 0 \tag{2.45}$$

$$\phi^f_{M1F1} + (1-\alpha_2)\phi^f_{filt.2} - \phi^f_{F1M2} - \phi^f_{filt.1} = 0 \tag{2.46}$$

$$\phi^f_{M2F2} + (1-\alpha_3)\phi^f_{filt.3} - \phi^f_{F2M3} - \phi^f_{filt.2} = 0 \tag{2.47}$$

$$\phi^f_{M3F3} + \Psi^f_{fw} - \Psi^f_{out} - \phi^f_{filt.3} = 0. \tag{2.48}$$

Le modèle adapté contiendra également les équations de complémentarité modifiées :

$$\phi^f_{ij} + \phi^s_{ij} + \phi^w_{ij} = \phi_{ij} \qquad \forall \phi_{ij} \text{ dans le modèle.} \tag{2.49}$$

Les variables de fractions de solides dissous $f^s\phi_{F1M2}, f^s\phi_{F2M3}, f^s\Psi_{out}, f^s\phi_{filt.3}$ et $f^s\phi_{filt.2}$ qui interviennent dans l'expression des DR devront aussi faire partie du modèle. La variable $f^s\Psi_{in}$ n'est pas incluse dans ces variables malgré le fait qu'elle intervient dans l'expression de DR_1 car le courant Ψ_{in} et sa composition font partie des données du problème. Une contrainte bilinéaire par variable de fractions de solides dissous doit être incluse, ce qui nous donne les cinq contraintes bilinéaires du modèle :

$$f^s\phi_{F1M2} \cdot \phi_{F1M2} = \phi^s_{F1M2}$$

$$f^s\phi_{F2M3} \cdot \phi_{F2M3} = \phi^s_{F2M3}$$

$$f^s\phi_{out} \cdot \phi_{out} = \phi^s_{out}$$

$$f^s\phi_{filt.3} \cdot \phi_{filt.3} = \phi^s_{filt.3}$$

$$f^s\phi_{filt.2} \cdot \phi_{filt.2} = \phi^s_{filt.2}.$$

Les contraintes concernant les DR, décrites dans (2.28), demeurent inchangées dans

le nouveau modèle.

Les bornes sur les variables $f^f\phi_{ij}$ qui n'apparaissent plus dans le modèle adapté deviennent :

$$\begin{aligned} & lb^f_{ij} \leq f^f\phi_{ij} \leq ub^f_{ij} \\ \Longrightarrow \quad & lb^f_{ij} \leq \frac{\phi^f_{ij}}{\phi_{ij}} \leq ub^f_{ij} \quad \text{Si} \quad \phi_{ij} \neq 0 \\ \Longrightarrow \quad & \phi_{ij} \cdot lb^f_{ij} \leq \phi^f_{ij} \leq \phi_{ij} \cdot ub^f_{ij}. \end{aligned} \tag{2.50}$$

Les contraintes (2.50) seront incluses dans le modèle adapté pour chaque variable $f^f\phi_{ij}$ du modèle de base.

Les bornes sur les variables

Les variables qui sont conservées d'un modèle à l'autre conservent aussi les mêmes bornes. Soient lb_{ij} et ub_{ij} les bornes inférieures et supérieures sur les débits ϕ_{ij}, lb^f_{ij} et ub^f_{ij} les bornes inférieures et supérieures sur les variables de fraction ϕ^f_{ij}. On définit de la même façon les bornes de ϕ^s_{ij} et ϕ^w_{ij}. Les bornes sur les nouvelles variables ϕ^f_{ij}, ϕ^s_{ij} et ϕ^w_{ij} sont calculées ainsi :

$$\begin{aligned} & lb^f_{ij} \leq f^f\phi_{ij} \leq ub^f_{ij}, \\ \Longrightarrow \quad & lb^f_{ij} \cdot \phi_{ij} \leq f^f\phi_{ij} \cdot \phi_{ij} \leq ub^f_{ij} \cdot \phi_{ij} \quad \text{puisque } \phi_{ij} \geq 0, \\ \Longrightarrow \quad & lb^f_{ij} \cdot lb_{ij} \leq lb^f_{ij} \cdot \phi_{ij} \leq \phi^f_{ij} \leq ub^f_{ij} \cdot \phi_{ij} \leq ub^f_{ij} \cdot ub_{ij}, \\ \Longrightarrow \quad & lb^f_{ij} \cdot lb_{ij} \leq \phi^f_{ij} \leq ub^f_{ij} \cdot ub_{ij}. \end{aligned} \tag{2.51}$$

De façon similaire, on calcule les bornes sur ϕ_{ij}^{s} et ϕ_{ij}^{w}.

La fonction objectif

Le modèle adapté conserve la même fonction objectif que le premier modèle, soit la somme pondérée de la consommation d'eau fraîche et de la propreté de la pâte lavé.

Reformulation

Soient deux classes de problèmes P_A et P_B. Une *réduction de turing en temps polynomial* de P_A à P_B est définit par un algorithme en trois phases. Une première phase reformule un problème A de P_A en un problème $B(A)$ de façon à ce que $B(A)$ soit un problème de P_B. Une seconde étape résout le problème $B(A)$. La dernière étape transforme la solution de $B(A)$ en une solution de A.

Nous nous intéressons ici à la première étape, c'est-à-dire la reformulation telle que définit par Audet [4]

Définition 1 (Reformulation) *Soient P_A et P_B deux classes de problème d'optimisation. Une reformulation $B(\cdot)$ de P_A en P_B est une transformation de P_A à P_B telle que pour tout problème A de P_A ainsi qu'une solution optimale de $B(A)$, on peut obtenir une solution optimale de A en un temps polynomial.*

Preuve : voir annexe A.

Proposition 1 (Reformulation du modèle) *Le modèle adapté est une reformulation du modèle de base.*

Preuve : voir annexe A.

En résumé, le modèle de base peut être reformulé comme un autre problème plus simple, un changement de variables permet de passer de l'un à l'autre. Le modèle adapté est plus facile à résoudre que le modèle de base car il contient beaucoup moins de contraintes avec termes bilinéaires et ces contraintes impliquent beaucoup moins de variables.

CHAPITRE 3

UNE MÉTHODE DE DÉCOMPOSITION EN OPTIMISATION GLOBALE

Ce chapitre présente l'algorithme d'optimisation globale GOP qui a été utilisé dans le cadre de ce projet. La première section présente l'algorithme de décomposition de Benders qui est nécessaire à la compréhension du GOP. Cette description sera suivie d'un bref aperçu de la décomposition de Benders généralisée développée par Geoffrion [12]. Ensuite, l'algorithme GOP sera présenté en détail en suivant le formalisme de Floudas et Visweswaran [11]. Finalement, les dernières sections contiennent les détails d'implantation et les améliorations apportées à l'algorithme.

3.1 Décomposition de Benders

La méthode de décomposition de Benders [6] est une méthode pour résoudre des problèmes de la forme suivante :

$$(\mathcal{P}) \begin{cases} \min\limits_{x,y} & c^t x + f(y) \\ & Ax + F(y) \geq b \\ & x \geq 0, \quad y \in S \end{cases}$$

où la matrice A possède m lignes et n colonnes, c est un vecteur de dimension n, $f : \mathbb{R}^p \longrightarrow \mathbb{R}$ une fonction scalaire de y, $F : \mathbb{R}^p \longrightarrow \mathbb{R}^m$ une fonction vectorielle de y, b un vecteur de dimension m et $S \subseteq \mathbb{R}^p$ un sous-ensemble de $\mathbb{R}^p$. Dans le cas où S représente les points entiers ou binaires de $\mathbb{R}^p$ alors le problème est un problème mixte. Certaines conditions sont nécessaires pour assurer la convergence de la méthode. Tout d'abord, la fonction f et les composantes de F doivent êtres continues et différentiables. L'ensemble S doit être fermé et borné, de plus si il ne représente pas les points entiers ou binaires de $\mathbb{R}^p$, alors il doit être déterminé par un ensemble d'inéquations linéaires et non-linéaires.

$$S = \{y | g_i(y) \leq 0, i = 1, ..., r, a_j' y \leq b_j, j = 1, ..., s\} \tag{3.1}$$

Les fonctions $g_i(y)$ de cette définition étant continues et différentiables.

Les variables y représentent ici les variables compliquantes. On les appellent compliquantes car le problème $\mathcal{P}$ est linéaire en x pour une valeur de y fixe. Considérant cela, il est normal de vouloir résoudre le problème pour une valeur de y fixe afin d'obtenir des informations qui permettront d'obtenir un meilleur y. Toute la difficulté

réside dans cette expression du meilleur y. Comment fait-on pour décider si un y est meilleur qu'un autre ? Comment passer d'un mauvais y à un autre meilleur ? Pour commencer, il faut considérer seulement les valeurs de y pour lesquelles il existe un x réalisable, c'est-à-dire :

$$y \in R = \{y \in S | \exists x \geq 0, Ax \geq b - F(y)\}. \tag{3.2}$$

Les vecteurs $y \in R$ ainsi obtenus seront appelés réalisables. Cette définition de l'ensemble R montre qu'il est difficile d'expliciter les contraintes le décrivant. Pour changer cette formulation, le lemme de Farkas est utilisé [21].

Lemme 1 (Lemme de Farkas) *Soit* $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ *alors*
$\exists x \geq 0, \quad Ax \geq \boldsymbol{b} \quad \Longleftrightarrow \quad (\forall u \geq 0, \quad u^t A \leq 0 \Longrightarrow u^t \boldsymbol{b} \leq 0)$

Voir preuve à l'annexe B.

Considérons le problème $(\mathcal{P})$ pour une valeur de y fixe, soit $y = \tilde{y}$. Définissons $(\mathcal{P}_{\tilde{y}})$ comme suit :

$$(\mathcal{P}_{\tilde{y}}) \qquad \begin{aligned} \min_x \quad & c^t x + f(\tilde{y}) \\ & Ax \geq b - F(\tilde{y}) \\ & x \geq 0. \end{aligned} \tag{3.3}$$

Par définition de R, on a :

$$\begin{aligned} y \in R &\iff (\mathcal{P}_{\tilde{y}}) \quad \text{est réalisable} \\ &\iff \exists x \geq 0, \quad Ax \geq b - F(\tilde{y}) \\ &\iff \forall u \geq 0, \quad (u^t A \leq 0 \quad \Longrightarrow \quad u^t(b - F(\tilde{y})) \leq 0) \\ &\iff \forall u \geq 0, \quad (A^t u \leq 0 \quad \Longrightarrow \quad (b - F(\tilde{y}))^t u \leq 0). \end{aligned}$$

Considérons maintenant le cône C :

$$C = \{u | A^t u \leq 0, u \geq 0\}. \tag{3.4}$$

Le cône C possède un nombre fini de générateurs $u_i^r, i = 1, ..., n_r$. Par la définition de C, chaque élément $u \in C$ peut être exprimé comme une combinaison conique des générateurs de C :

$$\forall u \in C \quad \exists \lambda_i \geq 0, i = 1, ..., n_r \quad t.q. \quad u = \sum_{i=1}^{n_r} \lambda_i u_i^r, \tag{3.5}$$

où n_r représente le nombre de générateurs de C. La définition des valeurs de y réalisable, peut être réécrite en fonction de ces générateurs :

$$\begin{aligned} y \in R &\iff \forall u \in C \quad (b - F(\tilde{y}))^t u \leq 0 \\ &\iff \forall \lambda \geq 0 \quad (b - F(\tilde{y}))^t \sum_{i=1}^{n_r} \lambda_i u_i^r \leq 0 \\ &\iff \forall \lambda \geq 0 \quad \sum_{i=1}^{n_r} \lambda_i (b - F(\tilde{y}))^t u_i^r \leq 0. \end{aligned}$$

Étant donné que cette inégalité est vraie pour toute valeur de λ, elle est également vraie pour chacun des λ_i, ce qui force chaque élément de la sommation à être non-négatif. On obtient donc :

$$y \in R \iff (b - F(y))^t u_i^r \leq 0, \quad \forall i = 1, ..., n_r.$$

Ces équivalences nous permettent de donner une nouvelle définition de R :

$$R = \{y \in S | \quad (b - F(y))^t u_i^r \leq 0, \quad i = 1, ..., n_r\}. \tag{3.6}$$

Pour la suite de l'algorithme, l'ensemble R est supposé non-vide, car si R est vide, alors le problème $(\mathcal{P})$ est irréalisable.

Le problème $(\mathcal{P})$ étant supposé réalisable, on peut le reformuler comme suit :

$$\min_{y \in R} \left\{ f(y) + \quad \min_{x \geq 0} \{c^t x | Ax \geq b - F(y)\} \right\}. \tag{3.7}$$

La séparation du problème en une minimisation interne et une minimisation externe le rend mieux adapté pour l'application du principe initial de fixer y et de passer à une meilleure valeur de y et ainsi de suite. Si $y \in R$, le problème de minimisation interne est réalisable. Par le théorème de dualité de la programmation linéaire, la valeur optimale du problème est la même que celle de son dual :

$$(\mathcal{P}_{int}) \quad \begin{array}{ll} \min\limits_{x} & c^t x \\ & Ax \geq b - F(y) \\ & x \geq 0 \end{array} \qquad (\mathcal{D}_{int}) \quad \begin{array}{ll} \max\limits_{u} & (b - F(y))^t u \\ & A^t u \leq c^t \\ & u \geq 0. \end{array}$$

En considérant que la solution optimale de $\mathcal{D}_{int}$ est $-\infty$ s'il est irréalisable, on peut

substituer le problème de minimisation interne dans (3.7) par son dual.

$$\min_{y \in R} \left\{ f(y) + \quad \max_{u \geq 0} \{(b - F(y))^t u | A^t u \leq c\} \right\}. \tag{3.8}$$

Considérons maintenant le polyèdre P des valeurs réalisables du problème de maximisation interne $\mathcal{D}_{int}$:

$$P = \{A^t u \leq c, u \geq 0\}. \tag{3.9}$$

On suppose que P est non-vide. Si P est vide alors le problème de maximisation interne $\mathcal{D}_{int}$ est irréalisable et donc le problème de minimisation interne $\mathcal{P}_{int}$ est non borné tout comme le problème de départ. Ce cas n'étant pas intéressant pour notre étude, P est considéré non-vide. Les rayons extrêmes de P sont les générateurs u_i^r du cône C.

Si le problème de maximisation interne $\mathcal{D}_{int}$ est borné, une solution est atteinte en un point extrême de P ; si le problème est non-borné, la solution tend vers ∞ le long d'un rayon extrême. Si le problème $\mathcal{D}_{int}$ est non borné, alors le problème $\mathcal{P}_{int}$ est irréalisable, c'est-à-dire que $y \notin R$. Pour la suite, supposons que $y \in R$. Dans ce cas, la solution optimale de $\mathcal{D}_{int}$ est un point extrême de P. Soient $u_i^p, i = 1, ..., n_p$, les points extrêmes de P. Le problème $\mathcal{P}$ peut être reformulé :

$$\min_{y \in R} \left\{ f(y) + \quad \max_{1 \leq i \leq n_p} \{ \quad (b - F(y))^t u_i^p \quad \} \right\} \tag{3.10}$$

ou encore

$$\min_{y} \quad z \tag{3.11}$$

$$z \geq f(y) + (b - F(y))^t u_i^p, \quad i = 1, ..., n_p \tag{3.12}$$

$$y \in R. \tag{3.13}$$

En développant la contrainte (3.13) à l'aide de la définition de R donnée en (3.6), on obtient

$$(\mathcal{P}2) \begin{cases} \min\limits_{y} \quad z \\ \qquad z \geq f(y) + (b - F(y))^t u_i^p, & i = 1, ..., n_p \\ \qquad 0 \geq (b - F(y))^t u_i^r, & i = 1, ..., n_r \\ \qquad y \in S. \end{cases} \tag{3.14}$$

Ce problème est équivalent au problème initial $(\mathcal{P})$, voir [20] pour plus de détails.

Remarque : si on dispose d'un ensemble restreint I_1 de points extrêmes et d'un ensemble restreint I_2 de rayons extrêmes on peut construire le problème $(\mathcal{R}2)$:

$$(\mathcal{R}2) \begin{cases} \min\limits_{y} \quad z \\ \qquad z \geq f(y) + (b - F(y))^t u_i^p, & i \in I_1 \subsetneq \{1, ..., n_p\} \\ \qquad 0 \geq (b - F(y))^t u_i^r, & i \in I_2 \subsetneq \{1, ..., n_r\} \\ \qquad y \in S. \end{cases} \tag{3.15}$$

Ce problème est une relaxation de $(\mathcal{P}2)$ et donc de $(\mathcal{P})$, ce qui implique que la valeur optimale de $(\mathcal{R}2)$ constitue une borne inférieure au problème initial.

Il est aussi possible d'obtenir une borne supérieure au problème initial. Pour $y = \tilde{y} \in R$

fixé, le problème $\mathcal{P}_{\tilde{y}}$ étant réalisable, sa valeur optimale constitue une borne supérieure au problème initial.

Résolution

Nous résumons ci-dessous le schéma général de la résolution d'un problème $(\mathcal{P})$ à l'aide de la décomposition de Benders. Cette façon de résoudre le problème consiste à générer les points extrêmes et les rayons extrêmes de P de façon itérative. À chaque itération, la résolution du problème $(\mathcal{P}_{\tilde{y}})$ appelé le problème auxiliaire permet de générer une coupe qui sera intégré dans le problème $(\mathcal{R}2)$ appelé le problème maître. Ces coupes réduisent le domaine réalisable de $(\mathcal{R}2)$. La borne inférieure au problème initial est donc resserrée à chaque itération. La méthode converge vers l'optimum global du problème $(\mathcal{P})$ en un temps fini [20].

Décomposition de Benders :

1. **Initialisation**

   $k \leftarrow 0$, $y_k \leftarrow y_0 \in S$, $I_1 = I_2 \leftarrow \emptyset$, $LB \leftarrow -\infty$, $UB \leftarrow +\infty$

2. **Résolution du problème auxiliaire**

   Résoudre le problème $\mathcal{D}_{int}$, puis choisir une de ces deux alternatives.

   (a) *Le problème est borné :* la solution u_k^p se trouve sur un point extrême de P.

   Dans ce cas, on génère une coupe d'optimalité

   $$z \geq f(y_k) + (b - F(y_k))^t u_k^p. \tag{3.16}$$

   On ajoute u_k^p dans le sous-ensemble des points extrêmes visités et inclure (3.16) dans ($\mathcal{R}2$). On met à jour la borne supérieure UB si nécessaire : si $\mathcal{D}_{int}$ possède une solution u_i^p alors $\mathcal{P}_{int}$ possède une solution x_i.

   Si $c^t x_i + f(y_i) \leq UB$ alors $UB \leftarrow c^t x_i + f(y_i)$.

   (b) *Le problème est non borné :* la solution tend vers ∞ le long d'un rayon extrême u_i^r de P.

   Dans ce cas, on génère une coupe de réalisabilité

   $$0 \geq (b - F(y_k))^t u_i^r. \tag{3.17}$$

   On ajoute u_k^p dans le sous ensemble des rayons extrêmes visités et inclure (3.17) dans ($\mathcal{R}2$).

3. **Résolution du problème maître**

   Résoudre le problème ($\mathcal{R}2$), puis mettre à jour la borne inférieure.

4. **Test de convergence** Si $LB = UB$ terminé, sinon on retourne à l'étape 2.

3.2 Décomposition de Benders généralisée

Le talon d'Achille de la méthode de Benders standard réside dans le fait que les problèmes résolus doivent être linéaires lorsque les variables compliquantes sont fixées. Geoffrion [12] atténue cette restriction de linéarité en généralisant le travail de Benders à une classe de problème beaucoup plus large. Les problèmes traités par cette nouvelle méthode doivent quand même posséder certaines caractéristiques.
Voici la catégorie de problèmes qui sont concernés :

$$(\mathcal{P}) \quad \begin{cases} \min\limits_{x,y} & f(x,y) \\ & G(x,y) \leq 0 \\ & x \in X, \quad y \in Y. \end{cases} \tag{3.18}$$

Cette fois, les variables y sont appelées les variables compliquantes car lorsque ces variables sont fixées, le problème $(\mathcal{P}_{\tilde{y}})$ est beaucoup plus facile à résoudre que le problème initial :

$$(\mathcal{P}_{\tilde{y}}) \quad \begin{cases} \min\limits_{x} & f(x,\tilde{y}) \\ & G(x,\tilde{y}) \leq 0 \\ & x \in X. \end{cases} \tag{3.19}$$

On peut penser par exemple à l'une des situations suivantes :

1. Pour y fixé, $(\mathcal{P}_{\tilde{y}})$ est un problème séparable en une série de sous-problèmes indépendants.

2. Pour y fixé, $(\mathcal{P}_{\tilde{y}})$ possède une structure connue comme le problème de transport classique, le problème de flot à coût minimum ou le problème du sac à dos pour lesquels des méthodes de résolution efficaces sont connues.

3. Pour y fixé, $(\mathcal{P}_{\tilde{y}})$ est un problème convexe qui peut être résolu à l'aide d'une méthode locale.

Tout comme dans l'algorithme de Benders de base, l'idée maîtresse de la méthode est le concept de projection ou de partitionnement. La projection du problème $\mathcal{P}$ sur y est :

$$\min_y \{v(y) | y \in Y \cap R\}, \tag{3.20}$$

où la fonction $v(y)$ est définie comme :

$$v(y) \equiv \inf_x \{f(x,y) | G(x,y) \leq 0, \quad x \in X\}. \tag{3.21}$$

L'ensemble des points réalisables R est défini comme :

$$R \equiv \{y | \exists x \in X, G(x,y) \leq 0\}. \tag{3.22}$$

Cette projection mène au problème suivant appelé problème maître

$$(\mathcal{PM}) \quad \begin{cases} \min\limits_y & z \\ & z \geq L_o(y, u_i) \qquad i = 1, ..., p \\ & 0 \geq L_r(y, \lambda_i) \qquad i = 1, ..., q \\ & y \in Y \end{cases} \tag{3.23}$$

Dans lequel les second membres des inégalités correspond aux coupes d'optimalité et de réalisabilité définies comme suit :

$$L_o(y, u_i) = \inf_{x \in X} \left\{ f(x, y) + {u_i}^t G(x, y) \right\}, \tag{3.24}$$

$$L_r(y, \lambda_i) = \inf_{x \in X} \left\{ {\lambda_i}^t G(x, y) \right\}. \tag{3.25}$$

Ici u_i est le vecteur multiplicateur retourné par la résolution du problème $(\mathcal{P}_{\tilde{y}})$ dans le cas où il est réalisable. De même, λ_i est le vecteur multiplicateur retourné par un algorithme dual dans le cas où $(\mathcal{P}_{\tilde{y}})$ est irréalisable.

Les conditions générales qui assurent la convergences de l'algorithme sont détaillées dans les travaux de Geoffrion [12]. L'algorithme d'optimisation correspondant à cette décomposition de Benders généralisé se décrit ainsi :

Décomposition de Benders Généralisée :

1. **Initialisation**

   $k \leftarrow 0$, $y_k \leftarrow y_0 \in Y$, $LB \leftarrow -\infty$, $UB \leftarrow +\infty$, $p = q \leftarrow 0$.

2. **Résolution du problème auxiliaire**

   Résoudre le problème $(\mathcal{P}_{\tilde{y}})$, puis choisir une de ces deux alternatives.

   (a) *Le problème est réalisable* : la résolution retourne alors le vecteur dual u_k.

   Si $f(x_k, y_k) \leq UB$ alors $UB \leftarrow f(x_k, y_k)$ et on conserve x_k et y_k comme meilleures valeurs réalisables.

   On pose $p \leftarrow p + 1$, $u_p \leftarrow u_k$ et on génère une coupe d'optimalité

   $$z \geq L_o(y, u_p) = \inf_{x \in X} \left\{ f(x, y) + {u_p}^t G(x, y) \right\} \tag{3.26}$$

   que l'on insère dans $(\mathcal{PM})$.

   (b) *Le problème est irréalisable* : l'algorithme dual retourne le vecteur multiplicateur λ_k.

   Dans ce cas, On pose $q \leftarrow q + 1$, $\lambda_q \leftarrow \lambda_k$ et on génère une coupe de réalisabilité

   $$0 \geq L_r(y, \lambda_q) = \inf_{x \in X} \left\{ {\lambda_q}^t G(x, y) \right\} \tag{3.27}$$

   que l'on insère dans $(\mathcal{PM})$.

3. **Résolution du problème maître**

   Résoudre le problème $(\mathcal{PM})$, récupérer la valeur optimale z_k et le vecteur optimal y_{k+1}, mettre à jour la borne inférieure $LB \leftarrow z_k$.

4. **Test de convergence**

   Si $\frac{(UB-LB)}{|LB|} \leq \epsilon$ alors terminé. Sinon on retourne à l'étape 2.

3.3 GOP

Floudas et Wiswerwaran [11] ont développé une version de la méthode de décomposition de Benders généralisée. Cette méthode est adapté a divers problème de design de procédé particulièrement en génie chimique. Floudas et Wiswerwaran ils aussi développé dans le cadre de cet algorithme d'importantes améliorations au niveau des tests de réductions. Cette méthode est connue sous le nom de "**G**lobal **Op**timization Algorithm" ou plus simplement "GOP". Les problèmes concernés par le GOP sont du type :

$$(\mathcal{P}) \quad \begin{cases} \min\limits_{x,y} & f(x,y) \\ & G(x,y) \leq 0 \\ & H(x,y) = 0 \\ & x \in X \subseteq \mathbb{R}^n, \quad y \in Y \subseteq \mathbb{R}^m \end{cases} , \tag{3.28}$$

où les ensemble X et Y sont des ensembles non-vides, compacts et convexes. Les bornes sur les variables sont incluses dans ces ensembles. $f(x,y) : X \times Y \longrightarrow \mathbb{R}$ une fonction scalaire, $G(x,y) : X \times Y \longrightarrow \mathbb{R}^p$ et $H(x,y) : X \times Y \longrightarrow \mathbb{R}^q$ des fonctions vectorielles. Les fonctions f, G et H sont continues, différentiables par morceaux et possèdent une définition analytique sur le domaine concerné. La partition des variables dans les ensembles X et Y doit respecter les conditions suivantes :

- $f(x,y)$ et $G(x,y)$ sont biconvexes, c'est-à-dire qu'elles sont convexes en x pour tout y fixé et convexes en y pour tout x fixé.
- $H(x,y)$ est affine en x pour tout y fixé et affine en y pour tout x fixé.

En fait, plusieurs problèmes sont inclus dans cette classe. Notamment, les problèmes quadratiques, et donc les problèmes bilinéaires.

Exemple illustratif 1 :[3]

$$\begin{aligned} \min_{x,y} \quad & -x + xy - y \\ -6x + 8y & \leq 3, \\ 3x - y & \leq 3, \\ x, y & \geq 0. \end{aligned}$$

Tout comme dans les deux précédentes méthodes, le principe clé du GOP est la projection du problème sur l'espace des y. La projection du problème $(\mathcal{P})$ sur l'espace des y est :

$$\begin{aligned} \min_{y} \quad & v(y) \\ & y \in Y \cap R \end{aligned}$$

où la fonction $v(y)$ est définie comme :

$$(\mathcal{P}_y) \begin{cases} v(y) \equiv \inf_{x} \quad f(x,y) \\ \qquad G(x,y) \leq 0, \\ \qquad H(x,y) = 0, \\ \qquad x \in X. \end{cases}$$

L'ensemble des y réalisables R est défini comme :

$$R \equiv \{y | \exists x \in X, G(x,y) \leq 0, H(x,y) = 0\}$$

Remarques

- Le problème de minimisation interne est un problème paramétrique en y. Ce problème sera le problème auxiliaire $(\mathcal{P}_{y^k})$ pour une valeur de y fixé $y = y^k$.
- La solution du problème $(\mathcal{P}_{y^k})$ permet de calculer une solution réalisable du problème initial et donne donc une borne supérieure valide. Cette borne sera réajustée de façon itérative pour raffiner la meilleure solution connue.
- La fonction $v(y)$ et l'ensemble R sont seulement connus implicitement. Le problème est donc difficile à résoudre sous cette forme.

Pour surmonter la difficulté causée par l'expression implicite de la fonction $v(y)$ et l'ensemble R, le problème $(\mathcal{P}_{y^k})$ est représenté dans sa forme duale.

Supposons que le problème $(\mathcal{P}_{y^k})$ respecte les conditions suivantes :

- $f(x, y)$ et $G(x, y)$ sont convexes en x pour tout y fixé,
- $H(x, y)$ est affine en x pour tout y fixé,
- $\forall y^k, \exists \hat{x} \in X$ tel que $G(\hat{x}, y^k) < 0, H(\hat{x}, y^k) = 0$ et $0 \in intH(X)$ où $H(X) \equiv \{H(x) : x \in X\}$.

Alors, par le théorème de dualité forte, sa solution est la même que celle de son dual :

$$\left\{ \begin{array}{ll} \min\limits_{x} & f(x, y^k) \\ & G(x, y^k) \leq 0 \\ & H(x, y^k) = 0 \\ & x \in X \subseteq \mathbb{R}^n \end{array} \right\} = \left\{ \sup_{\mu \geq 0, \lambda} \inf_{x \in X} L(x, y^k, \lambda, \mu) \right\} \quad \forall y^k \in Y \cap R$$

où λ et μ sont respectivement les multiplicateurs de Lagrange des contraintes d'inégalité et d'égalité du problème $(\mathcal{P}_{y^k})$ et

$$L(x, y, \lambda, \mu) = f(x, y) + \lambda^t G(x, y) + \mu^t H(x, y) \tag{3.29}$$

est le lagrangien du problème auxiliaire.

Il est maintenant possible de substituer $(\mathcal{P}_{y^k})$ par sa forme duale dans la définition de $v(y)$:

$$v(y) = \sup_{\mu \geq 0, \lambda} \inf_{x \in X} L(x, y, \lambda, \mu), \qquad \forall y \in Y \cap R. \tag{3.30}$$

En utilisant la définition d'un supremum, on peut déduire que

$$v(y) \geq \inf_{x \in X} L(x, y, \lambda, \mu), \qquad \forall \lambda, \mu \geq 0, y \in Y \cap R. \tag{3.31}$$

On substitue ensuite cette relaxation de $v(y)$ dans la définition du problème initial :

$$\begin{aligned} \min_{y} \quad & v(y) \\ & v(y) \geq \min_{x \in X} L(x, y, \lambda, \mu), \qquad \forall \lambda, \mu \geq 0 \\ & y \in Y \cap R, & (3.32) \\ & R \equiv \{y | \exists x \in X \quad H(x, y) = 0, G(x, y) \leq 0\}. & (3.33) \end{aligned}$$

Les contraintes (3.32) et (3.33) sont des fonctions implicites de y, ce qui rend le problème difficile à résoudre. Pour faciliter la résolution, il suffit de relaxer le problème en laissant de coté ces deux contraintes. Ce qui nous donne l'expression du problème dual relaxé $(\mathcal{RD})$:

$$(\mathcal{RD}) \quad \begin{cases} \min_{y \in Y} \quad z \\ \qquad z \geq \min_{x \in X} L(x, y, \lambda, \mu), \qquad \forall \lambda, \mu \geq 0. \end{cases} \tag{3.34}$$

La solution du problème $(\mathcal{RD})$ représente une borne inférieure valide au problème initial, cette borne sera utilisée avec la borne supérieure provenant du problème $(\mathcal{P}_{y^k})$ comme critère d'arrêt de l'algorithme. Le problème $(\mathcal{RD})$ demeure toutefois difficile

à résoudre pour deux raisons. D'abord, la contrainte (3.34) contient le problème de minimisation interne ($\mathcal{IRD}$) paramétrique en y.

$$(\mathcal{IRD}) \qquad \min_{x \in X} L(x, y, \lambda^k, \mu^k) \tag{3.35}$$

Ensuite, la contrainte $z \geq \min_{x \in X} L(x, y, \lambda, \mu)$ doit être respectée pour toutes valeurs de λ et de $\mu \geq 0$, ce qui représente en fait une infinité de contraintes. Il est donc nécessaire de le reformuler de façon à ce qu'il ne contienne qu'un nombre fini de contraintes tout en fournissant encore une borne inférieure valide.

Reformulation de ($\mathcal{RD}$)

Pour contourner la seconde difficulté, les multiplicateurs de Lagrange seront générés de façon itérative. À chaque itération, le problème ($\mathcal{P}_{y^k}$) étant un problème convexe, il peut donc être résolu globalement sans trop de difficultés à l'aide d'un algorithme d'optimisation convexe (algorithme de descente). Les valeurs des multiplicateurs de Lagrange λ^k et de $\mu^k \geq 0$ générées par cet algorithme seront conservées.

Cette propriété est facilement observable sur l'exemple illustratif

Suite de l'exemple 1 :

$$\begin{aligned} \min_{x} \quad & (y^k - 1)x - y^k, \\ & -6x \leq 3 - 8y^k \\ & 3x \leq 3 + y^k \\ & 0 \leq x \leq 3. \end{aligned}$$

est un problème linéaire, il est donc très facile de trouver sa solution optimale et de conserver les valeurs des multiplicateurs de Lagrange.

Pour contourner la première difficulté, la minimisation sur l'ensemble X sera remplacée par une énumération sur l'ensemble des combinaisons des bornes de certaines variables x. Cette modification changera le problème $(\mathcal{RD})$ par une série de problèmes convexes qui ne font intervenir que les variables y. Pour arriver à ce résultat, il faudra tout d'abord utiliser la linéarisation du lagrangien au lieu du lagrangien lui-même. Puis l'espace des y sera divisé successivement pour cerner la valeur optimale des variables compliquantes.

Linéarisation du lagrangien

la linéarisation du lagrangien $L(x, y, \lambda^k, \mu^k)$ autour du point x^k est :

$$\begin{aligned} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} &= L(x^k, y, \lambda^k, \mu^k) + \nabla_x L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} (x - x^k) \\ &= L(x^k, y, \lambda^k, \mu^k) + \sum_{i=1}^{n} \nabla_{x_i} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} (x_i - x_i^k) \end{aligned}$$

Par convexité, on obtient :

$$\min_{x \in X} L(x, y^d, \lambda^k, \mu^k)\big|_{x^k}^{lin} \leq \min_{x \in X} L(x, y^d, \lambda^k, \mu^k) \quad \forall y = y^d \tag{3.36}$$

Preuve : Le lagrangien est une fonction convexe en x et donc sa linéarisation est une approximation qui sous-estime la valeur de la fonction pour toutes les valeurs de x. Et donc, pour chaque point de linéarisation x^k, on a l'inégalité suivante :

$$L(x, y^d, \lambda^k, \mu^k)\big|_{x^k}^{lin} \leq L(x, y^d, \lambda^k, \mu^k) \quad \forall y = y^d \tag{3.37}$$

Cette inégalité étant vraie pour tout valeur de x, l'inégalité (3.36) découle.

Définition des variables *x-connectées*

Les variables x_i qui sont présentes dans la linéarisation du lagrangien et telle que $\nabla_{x_i} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin}$ est une fonction de y sont appelées les *x-connectées* car ils interviennent de façon bilinéaire dans la linéarisation du lagrangien. À l'aide de cette définition, la linéarisation du lagrangien peut être reformulée ainsi :

$$L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} = L_0^k(y, \lambda^k, \mu^k) + \sum_{i=1}^{|I_c^k|} x_i g_i^k(y), \tag{3.38}$$

où I_c^k est l'ensemble des *x-connectées* pour l'itération k, $L_0^k(y, \lambda^k, \mu^k)$ représente l'ensemble des termes de $L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin}$ qui ne dépendent que de y et $g_i^k(y) = \nabla_{x_i} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin}$ est le gradient du lagrangien par rapport à x_i.

Théorème 1 (x-connectées) *La solution du problème de minimisation interne ($\mathcal{IRD}$) avec le lagrangien remplacé par sa linéarisation ne dépend que des variables x-connectées.*

Preuve : voir annexe C.

On peut illustrer cette propriété à l'aide de l'exemple suivant :

Exemple 2 :

$$(\mathcal{P}) \quad \begin{cases} \min\limits_{x_1, x_2, y_1} & x_1 y_1 + x_2, \\ -x_1 + 4y_1 & \leq 2 \\ 3x_1 - x_2 & \leq 3 \end{cases} \tag{3.39}$$

Soit $(\mathcal{P}(y_1))$ le problème $(\mathcal{P})$ pour une valeur de y_1 fixé, alors la solution de $(\mathcal{P}(1))$ donne les valeurs suivantes :

$x_1 = 2$, $x_2 = 3$, $\mu_1 = 4$, $\mu_2 = 1$.

L'expression du lagrangien pour ce problème donne :

$$\begin{aligned} L(x, y, \lambda^k, \mu^k) &= -x_1 y_1 + x_2 + 4(-x_1 + 4y_1 - 2) + 1(3x_1 - x_2 - 3) \\ &= x_1(y_1 - 1) + 16y_1 - 11 \end{aligned}$$

La variable x_2 n'étant pas une variable *x-connectées* n'apparaît plus dans l'expression du lagrangien. Cependant, toutes les variables qui ne sont pas des *x-connectées* ne disparaîtront pas nécessairement toutes de l'expression du lagrangien. Dans le cas où une variable apparaît "de façon convexe" dans le lagrangien mais n'est pas *connectée*, cette variable ne disparaîtra pas dans le lagrangien lui-même, mais dans sa linéarisation. C'est pourquoi le lagrangien dans le problème de minimisation interne ($\mathcal{IRD}$) est remplacé par sa linéarisation.

$$\left\{ \begin{array}{ll} \min\limits_{y \in Y} & z \\ & z \geq \min\limits_{x \in X} L(x, y, \lambda, \mu), \\ & \forall \lambda, \mu \geq 0. \end{array} \right\} \geq \left\{ \begin{array}{ll} \min\limits_{y \in Y} & z \\ & z \geq \min\limits_{x \in I_c^k} L(x, y, \lambda, \mu)|_{x^k}^{lin}, \\ & \forall \lambda, \mu \geq 0. \end{array} \right\}$$

Le théorème 1 est surtout important d'un point de vue algorithmique. En accord avec ce théorème, on peut remplacer le problème de minimisation interne par la minimisation du lagrangien linéarisé. De plus, cette minimisation se fait sur l'ensemble des *x-connectées* seulement et non sur l'ensemble des x au complet. Le problème est donc beaucoup plus petit et plus facile à résoudre.

Combinaison des bornes sur les *x-connectées*.

Soit B_j une combinaison des bornes supérieures et inférieures de ces variables et finalement CB l'ensemble de toutes les combinaisons de bornes. Par la définition même

de ces ensembles, on a que $|CB| = 2^{|I_c^k|}$.

Théorème 2 (Combinaison des bornes) *Si le lagrangien est linéaire en x, $\forall y = y^d$ alors :*

$$\min_{x \in X} L(x, y, \lambda^k, \mu^k) \geq \min_{B_j \in CB} L(x^{B_j}, y, \lambda^k, \mu^k) \quad \forall y. \tag{3.40}$$

Preuve

Supposons le contraire. Dans ce cas, il existe une valeur $y^* \in Y$ telle que :

$$\min_{x \in X} L(x, y^*, \lambda^k, \mu^k) < \min_{B_j \in CB} L(x^{B_j}, y^*, \lambda^k, \mu^k). \tag{3.41}$$

La fonction $L(x, y, \lambda^k, \mu^k)$ étant linéaire en x pour toute valeur de y. Le minimum sera atteint sur la frontière de l'ensemble X. Comme les bornes sur les variables x font partie de la définition de l'ensemble X, il est impossible que l'inégalité (3.41) soit valide.

Cette propriété peut être observée à l'aide de l'exemple illustratif.

Suite de l'exemple 1 :

$$L(x, y, \mu^k) = -x + xy - y + \mu_1^k(-6x + 8y - 3) + \mu_2^k(3x - y - 3) + \mu_3^k(0 - x) + \mu_4^k(x - 4).$$

Le lagrangien de l'exemple 1 est linéaire en x pour tout y fixé, et donc la minimisation sur x aboutira toujours sur l'une des bornes de x. En conséquence l'optimum sera atteint en $x = 0$ ou en $x = 4$.

Partitionnement de l'espace des y

La positivité ou la négativité des fonctions $g_i(y)$ définit un ensemble d'inéquations qui partitionne l'espace des y en différentes parties. Ces différentes parties seront très

importantes dans le déroulement de l'algorithme. Dans chacune de ces parties, la minimisation sur les combinaisons des bornes donnera toujours le même résultat. Prenons le problème de minimisation interne avec le lagrangien linéarisé,

$$\begin{aligned}\min_{x\in X} L(x,y,\lambda,\mu)|_{x^k}^{lin} &= \min_{x\in X}\left\{L_0^k(y,\lambda^k,\mu^k)+\sum_{i=1}^{|I_c^k|} x_i g_i^k(y)\right\}\\ &= L_0^k(y,\lambda^k,\mu^k)+\sum_{i=1}^{|I_c^k|}\min_{x_i}\left\{x_i g_i^k(y)\right\}, \quad \forall y\in Y.\end{aligned}$$

La composante i dans le dernier terme est linéaire en x_i et donc la minimisation sur x_i aboutira à l'une des bornes. Deux cas sont possibles :

1. si $g_i^k(y)\geq 0$ alors $\min_{x_i}\left\{x_i g_i^k(y)\right\}\geq x_i^L g_i^k(y)$;
2. si $g_i^k(y)\leq 0$ alors $\min_{x_i}\left\{x_i g_i^k(y)\right\}\geq x_i^U g_i^k(y)$;

où x_i^L et x_i^U sont respectivement la borne inférieure et supérieure de la variable x_i. Ce résultat permet d'énoncer une généralisation du théorème 2.

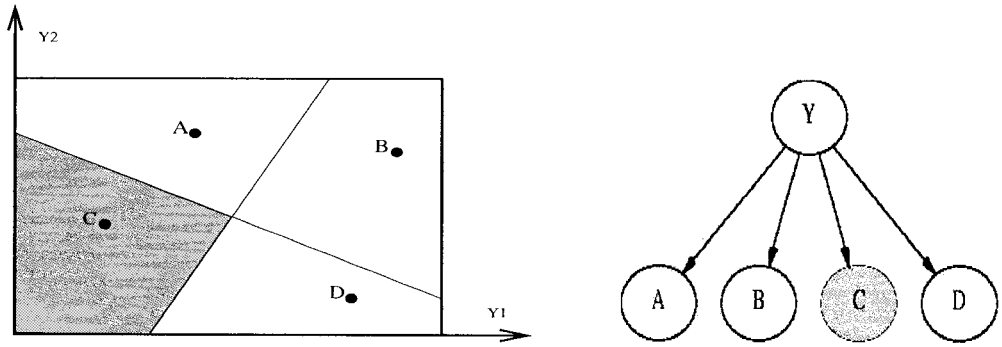
$$\min_{x\in X} L(x,y,\lambda,\mu)|_{x^k}^{lin} \geq \min_{B_j\in CB}\left\{\begin{array}{l} L(x^{B_j},y,\lambda,\mu)|_{x^k}^{lin}\\ \quad g_i(y)\geq 0 \quad \forall x_i^{B_j}=x_i^L\\ \quad g_i(y)\leq 0 \quad \forall x_i^{B_j}=x_i^U\end{array}\right\}\forall y\in Y \tag{3.42}$$

Les contraintes de positivité et de négativité des fonctions $g_i(y)$ sont appelées les "qualifying constraints" ou contraintes de partitionnement. Ces contraintes seront linéaires si le problème de base est bilinéaire ou quadratique. Dans ce cas, elles partitionneront l'espace des y en $2^{|I_c^k|}$ parties. Dans chacune de ces parties, une relaxation du lagrangien qui borne inférieurement la solution globale sera générée en fixant les variables *x-connectées* à leurs bornes appropriées. Le choix des bornes sera fait en utilisant la propriété (3.42).

Considérons maintenant la première itération du GOP. La région initiale est l'espace des y au complet. Le point de départ y^1 est supposé appartenir à l'ensemble $Y \cap R$. La résolution du problème $(\mathcal{P}_{y^1})$ permettra d'obtenir les valeurs des multiplicateurs de Lagrange μ^1 et λ^1. On doit ensuite identifier les variables *x-connectées* et générer les contraintes de partitionnement.

Supposons que l'ensemble des y du problème initial contient deux variables, y_1 et y_2 et que l'ensemble des *x-connectées* à la première itération contient deux variables. Dans ce cas, le partitionnement de la région est illustré à la figure (3.1).

Dans chacune des $2^{|I_c^1|}$ sous-régions, un sous-problème de cette forme sera résolu :



(a) Partitionnement de l'espace des y à l'itération 1

(b) Branchement du GOP à l'itération 1 : la meilleure borne inférieure est obtenue dans la sous-région C

Figure 3.1 – Itération 1

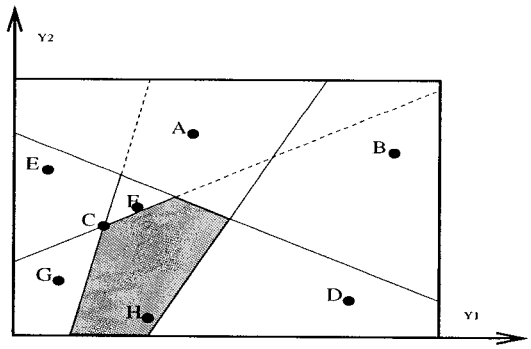
$$\min_{y \in Y, z} \quad z$$

$$z \geq L(x^{B_j}, y, \lambda^1, \mu^1)|_{x^1}^{lin}$$

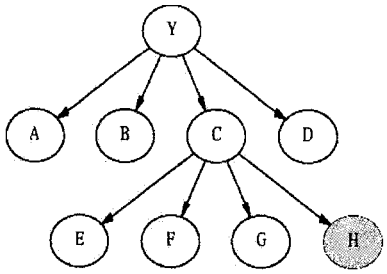
$$\left.\begin{array}{lll} g_i^1(y) \geq 0 & \text{si} & x_i^{B_j} = x_i^L \\ g_i^1(y) \leq 0 & \text{si} & x_i^{B_j} = x_i^U \end{array}\right\} \forall i \in I_c^1.$$

Si la valeur optimale z^* de ce problème est prometteuse, c'est-à-dire qu'elle est inférieure à la borne supérieure obtenue à l'aide du problème auxiliaire $(\mathcal{P}_{y^1})$, cette valeur est conservée en vue d'un raffinement futur. Dans le cas contraire, la valeur est rejetée et la sous-région correspondante est déclarée sondée et ne sera plus considérée dans les raffinements futurs.

La seconde itération utilisera la solution la plus prometteuse de la première itération. Soit $z^C = min\{z^A, z^B, z^C, z^D\}$, alors $y^2 = y^C$ et l'algorithme devra construire et résoudre le problème $(\mathcal{P}_{y^2})$. Si la solution optimale de $(\mathcal{P}_{y^2})$ est inférieure à la borne supérieure obtenue à la première itération, la valeur de la meilleure borne supérieure devra être réajustée. Supposons que l'ensemble I_c^1 contienne deux variables, dans ce cas la sous-région C devra être partitionnée en quatre sous-régions. Comme le montre la figure (3.2).



(a) Partitionnement de l'espace des y à l'itération 2



(b) Branchement du GOP à l'itération 2 : la meilleure borne inférieure est obtenue dans la sous-région H (tous les noeuds ouverts sont considérés).

Figure 3.2 – Itération 2

Dans chacune des $2^{|I_c^2|}$ sous-régions, un sous-problème de la forme suivante est résolu :

$$
\begin{aligned}
\min_{y \in Y, z} \quad & z \\
& z \geq L(x^{B_j}, y, \lambda^1, \mu^1)|_{x^1}^{lin} \\
& \left.\begin{array}{l} g_i^1(y) \geq 0 \quad \text{si} \quad g_i^1(y^2) \geq 0 \\ g_i^1(y) \leq 0 \quad \text{si} \quad g_i^1(y^2) \leq 0 \end{array}\right\} \forall i \in I_c^1 \\
& z \geq L(x^{B_j}, y, \lambda^2, \mu^2)|_{x^2}^{lin} \\
& \left.\begin{array}{l} g_i^2(y) \geq 0 \quad \text{si} \quad x_i^{B_j} = x_i^L \\ g_i^2(y) \leq 0 \quad \text{si} \quad x_i^{B_j} = x_i^U \end{array}\right\} \forall i \in I_c^2.
\end{aligned}
$$

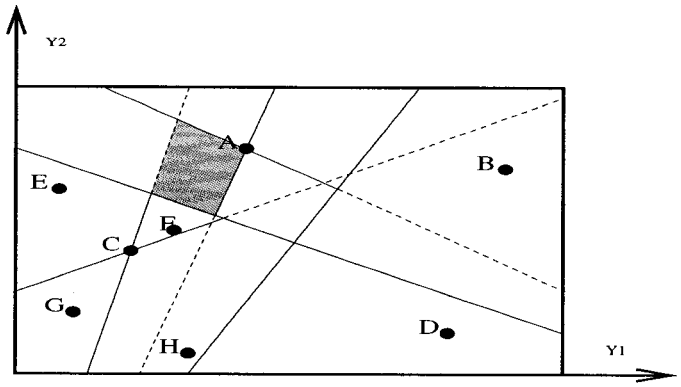
Le premier ensemble de contraintes de partitionnement représente la définition de la sous-région C.

Apportons maintenant quelques précisions à l'algorithme GOP tel que décrit dans les travaux de Floudas et Visweswaran [11]. À partir de la troisième itération, il faut faire bien attention au choix des contraintes de partitionnement à inclure dans le sous-problème. En se basant sur les sous-problèmes des itérations un et deux, on pourrait

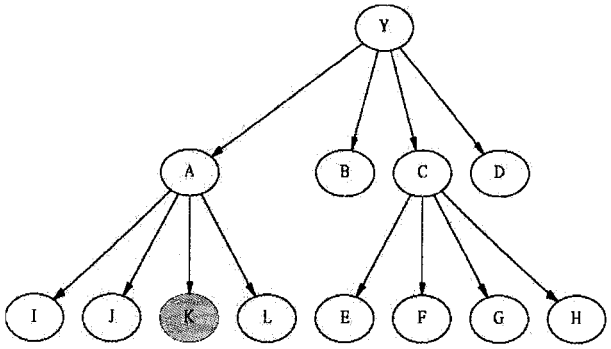
être tenté de formuler ainsi le sous-problème de l'itération trois :

$$
\begin{aligned}
\min_{y \in Y, z} \quad & z \\
& z \geq L(x^{B_j}, y, \lambda^1, \mu^1)|_{x^1}^{lin} \\
& \left.\begin{array}{l} g_i^1(y) \geq 0 \quad \text{si} \quad g_i^1(y^3) \geq 0 \\ g_i^1(y) \leq 0 \quad \text{si} \quad g_i^1(y^3) \leq 0 \end{array}\right\} \forall i \in I_c^1 \\
& z \geq L(x^{B_j}, y, \lambda^2, \mu^2)|_{x^2}^{lin} \\
& \left.\begin{array}{l} g_i^2(y) \geq 0 \quad \text{si} \quad g_i^2(y^3) \geq 0 \\ g_i^2(y) \leq 0 \quad \text{si} \quad g_i^2(y^3) \leq 0 \end{array}\right\} \forall i \in I_c^2 \\
& z \geq L(x^{B_j}, y, \lambda^2, \mu^2)|_{x^2}^{lin} \\
& \left.\begin{array}{l} g_i^3(y) \geq 0 \quad \text{si} \quad x_i^{B_j} = x_i^L \\ g_i^3(y) \leq 0 \quad \text{si} \quad x_i^{B_j} = x_i^U \end{array}\right\} \forall i \in I_c^3.
\end{aligned}
$$

Mais cette formulation est fausse car elle peut mener à l'élimination de régions non-sondées. En effet, le choix de la solution la plus prometteuse doit se faire non seulement sur les $2^{|I_c^2|}$ solutions de l'itération précédente, mais en plus sur les autres solutions non-sondées des autres itérations précédentes. Dans notre exemple la solution est : $y^3 = \min\{z^A, z^B, z^D, z^E, z^F, z^G, z^H\}$. Supposons que le minimum est z^A, dans ce cas, on doit générer des contraintes pour partitionner la sous-région A et inclure les contraintes de partitionnement de la première itération qui définissent la sous-région A. Mais on ne doit pas inclure celles de la seconde itération, car elles ne concernent que la sous-région C.



(a) Mauvais partitionnement de l'espace des y à l'itération 3 : les coupes de la subdivision de la sous-région C interviennent dans la sous-région A.

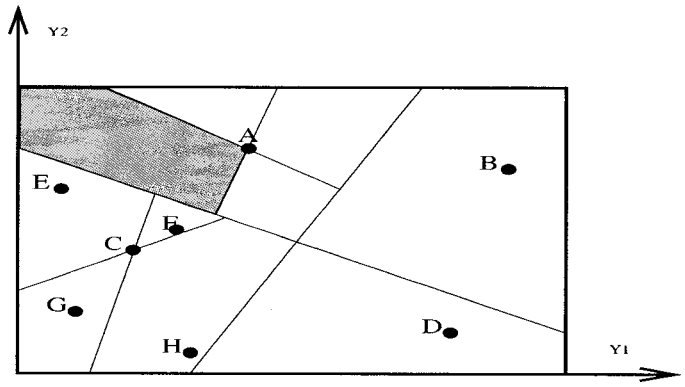


(b) Mauvais branchement du GOP à l'itération 3

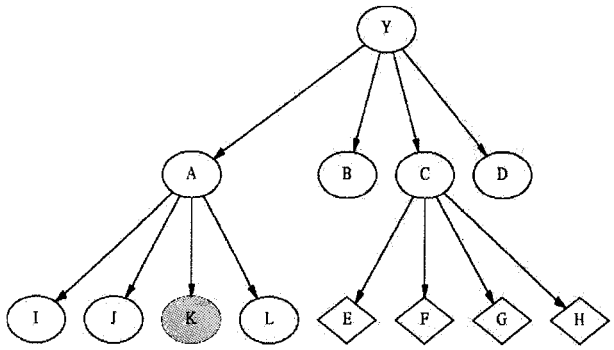
Figure 3.3 – Itération 3

Voici donc la formulation correcte du sous-problème de l'itération k :

$$
\begin{aligned}
\min_{y \in Y, z} \quad & z \\
& z \geq L(x^{B_j}, y, \lambda^k, \mu^k)|_{x^k}^{lin} \\
& \left.\begin{array}{ll} g_i^k(y) \geq 0 & \text{si} \quad x_i^{B_j} = x_i^L \\ g_i^k(y) \leq 0 & \text{si} \quad x_i^{B_j} = x_i^U \end{array}\right\} \forall i \in I_c^k \\
& (y, z) \in C_R,
\end{aligned}
$$



(a) Bon partitionnement de l'espace des y à l'itération 3 : les coupes de la subdivision de chaque sous-régions n'interviennent que localement.



(b) Bon branchement du GOP à l'itération 3

Figure 3.4 – Itération 3

où l'ensemble C_R représente la sous-région de l'itération courante qui doit être partitionnée. Il est construit ainsi :

Soient m l'indice d'un noeud dans l'arbre et P_m le noeud parent de m.

1. si $k = 1$, $C_R = Y$;
2. si $k \geq 2$, $C_R = \mathrm{Y}$, $m = c$, le noeud correspondant à la sous-région de l'itération courante dans l'arbre de branchement.
   (a) ajouter à C_R la coupe de Lagrange $z \geq L(x^{B_m}, y, \lambda^m, \mu^m)|_{x^m}^{lin}$ du noeud m,
   (b) ajouter à C_R les contraintes de partitionnement du noeud m qui sont sa-

tisfaites par la solution y^k,

(c) fixer $m = P_m$ le noeud parent de m dans l'arbre de branchement. Si m est le noeud racine, la construction est terminée.

(d) répéter les étapes 2a à 2c.

Problème auxiliaire irréalisable

Depuis le début de la description du GOP, les vecteurs optimaux des sous-problèmes y^k sont considérés réalisables, c'est-à-dire que $y^k \in R$. Cette supposition permet de garantir que le problème auxiliaire $(\mathcal{P}_{y^k})$ sera réalisable et qu'il sera possible de récupérer les multiplicateurs de Lagrange correspondants. En fait, il est possible que $(\mathcal{P}_{y^k})$ soit irréalisable. Dans ce cas, il faut résoudre un autre problème auxiliaire pour générer des multiplicateurs quasi-optimaux.

$$(\mathcal{PI}_{y^k}) \quad \begin{cases} \min\limits_{\beta^+,\beta^-,\alpha\geq 0,x} & \delta = \sum_{i=1}^{p} \alpha_i + \sum_{i=1}^{q} (\beta_i^+ + \beta_i^-), \\ & H(x, y^k) + \beta^+ - \beta^- = 0, \\ & G(x, y^k) - \alpha \leq 0, \\ & x \in X. \end{cases}$$

Étant donné que les fonctions $H(x, y^k)$, $G(x, y^k)$ et la fonction objectif sont convexes, le théorème de dualité forte permet d'affirmer que la solution du problème $(\mathcal{PI}_{y^k})$ est la même que celle de son dual.

$$\left\{ \begin{array}{l} \min\limits_{\beta^+,\beta^-,\alpha\geq 0,x} \quad \delta, \\ H(x, y^k) + \beta^+ - \beta^- = 0, \\ G(x, y^k) - \alpha \leq 0, \\ x \in X. \end{array} \right\} = \left\{ \max_{\mu_I \geq 0, \lambda_I} \min_{x \in X} \left\{ \begin{array}{l} \delta \\ +\lambda_I^t (H(x, y^k) + \beta^+ - \beta^-) \\ +\mu_I^t (G(x, y^k) - \alpha) \end{array} \right\} \right\},$$

où λ_I et μ_I sont les multiplicateurs de Lagrange des contraintes $H(x, y^k)$ et $G(x, y^k)$. Les conditions d'optimalité du problème $(\mathcal{PI}_{y^k})$ permet d'affirmer que si δ est la solution optimale de $(\mathcal{PI}_{y^k})$, alors

$$\delta = \max_{\mu_I \geq 0, \lambda_I} \min_{x \in X} \left\{ \lambda_I^t H(x, y) + \mu_I^t G(x, y) \right\}. \tag{3.43}$$

La valeur de δ représente l'irréalisabilité et on peut utiliser (3.43) comme une coupe de réalisabilité dans le sous-problème $(\mathcal{RD})$

$$\max_{\mu_I \geq 0, \lambda_I} \min_{x \in X} \left\{ \lambda_I^t H(x, y) + \mu_I^t G(x, y) \right\} = 0. \tag{3.44}$$

En fait, c'est une relaxation de cette contrainte qui sera incluse dans le problème :

$$\min_{x \in X} \left\{ L_I(x, y, \lambda_I, \mu_I) = \lambda_I^t H(x, y) + \mu_I^t G(x, y) \right\} \leq 0. \tag{3.45}$$

La situation où $(\mathcal{PI}_{y^k})$ est irréalisable, n'est donc pas si catastrophique, car elle permet de générer une coupe de réalisabilité 3.45 qui sera ajoutée dans les sous-problèmes des itérations suivantes.

Resserrement des bornes sur les variables

La qualité des bornes sur les variables dans le problème de base influence beaucoup la rapidité de la résolution. Les bornes les plus importantes sont celles sur les *x-connectées* car ces variables sont fixées à leurs bornes supérieures et inférieures pour construire les coupes d'optimalités et de réalisabilités. Il est donc important d'obtenir les bornes les plus précises possibles pour ces variables.

La méthode utilisée pour le resserrement des bornes est emprunté à Audet *et al.*[5].

Pour commencer, le problème bilinéaire doit être considéré comme un problème linéaire. Ceci peut être fait en substituant à tous les produits $x_i y_j$ une nouvelle variable w_{ij}. Ces nouvelles variables doivent être bornées en utilisant les bornes sur les variables initiales :

$$\begin{array}{c} 0 \leq x_i^L \leq x_i \leq x_i^U \text{ et} \\ 0 \leq y_i^L \leq y_i \leq y_i^U \end{array} \quad \Longrightarrow \quad x_i^L y_i^L \leq w_{ij} \leq x_i^U y_i^U.$$

De façon similaire, on calcule les bornes dans le cas où les variables ne sont pas positives.

Pour relier les variables w_{ij} aux variables initiales du problème, on utilise une série de contraintes d'inégalitées

$$\begin{aligned}
(x_i - x_i^L)(y_i - y_i^L) &= w_{ij} - y_i^L x_i - x_i^L y_i + x_i^L y_i^L \geq 0, \\
(x_i - x_i^L)(y_i - y_i^U) &= w_{ij} - y_i^U x_i - x_i^L y_i + x_i^L y_i^U \leq 0, \\
(x_i - x_i^U)(y_i - y_i^L) &= w_{ij} - y_i^L x_i - x_i^U y_i + x_i^U y_i^L \leq 0, \\
(x_i - x_i^U)(y_i - y_i^U) &= w_{ij} - y_i^U x_i - x_i^U y_i + x_i^U y_i^U \geq 0.
\end{aligned}$$

Une fois que les contraintes sont ajoutées pour chaque couple $x_i y_j$, on commence le resserrement des bornes. Pour chaque variable x_i, y_j et w_{ij}, on résout deux problèmes linéaires, minimiser et maximiser la variable sous les contraintes de base et les contraintes d'inégalité décrites ci-haut. Si la valeur du problème est plus serrée que la borne correspondante, on doit réajuster la borne et recommencer le processus à partir de la première variable. Une fois que toutes les variables ont été resserrées, il suffit d'insérer ces nouvelles bornes dans le problème de base pour accélérer la résolution.

Algorithme GOP

Voici une description sommaire des principales étapes de l'algorithme. En plus des notations utilisées précédemment, on définit F comme l'ensemble des itérations où le problème $(\mathcal{PI}_{y^K})$ est réalisable et I comme l'ensemble des itérations où il est irréalisable.

1. ***Initialisation***

   (a) Lecture des données du problèmes, dont le critère de convergence ϵ et le point de départ y_1.

   (b) Définition des bornes supérieure et inférieure globale LB et UB.

   (c) Resserrement des bornes du problème initial.

   (d) Fixation des valeurs $k \leftarrow 1$,$C_R \leftarrow Y$,$m \leftarrow$ noeud racine.

2. ***Sélection de la région à explorer***

   Si $k \geq 2$, $C_R \leftarrow Y$, $m \leftarrow c$ le noeud correspondant à la sous-région de l'itération courante dans l'arbre de branchement.

   (a) Ajouter à C_R la coupe de Lagrange $z \geq L(x^{B_m}, y, \lambda^m, \mu^m)|_{x^m}^{lin}$ du noeud m, ou la coupe de réalisabilité selon le cas.

   (b) Ajouter à C_R les contraintes de partitionnement du noeud m qui sont satisfaites par la solution y^k.

   (c) Fixer $m \leftarrow P_m$. Si m est le noeud racine, la construction est terminée.

   (d) Répéter les étapes 2a à 2c.

3. ***Problème auxiliaire***

   (a) résoudre le problème $(\mathcal{P}_{y^k})$

i. S'il est réalisable, $F = F \cup \{k\}$, mettre à jour la borne supérieure $UB \leftarrow \min\{UB, f(x^*, y^k)\}$.

ii. Sinon, résoudre $(\mathcal{PI}_{y^k})$, $I \leftarrow I \cup \{k\}$.

(b) Mémoriser les valeurs de y^k, μ^k et λ^k.

4. ***Partitionnement***

(a) Générer le lagrangien de l'itération courante,

$L^k(x, y, \lambda^k, \mu^k)$ si $(\mathcal{P}_{y^k})$ est réalisable, $L_I^k(x, y, \lambda^k, \mu^k)$ sinon.

(b) Identifier l'ensemble I_c^k des variables *x-connectées* et les fonctions $g_i(y)$ correspondantes.

(c) Si possible resserrer les bornes des *x-connectées* sur C_R.

5. ***Résolution des sous-problèmes***

(a) Sélectionner une combinaison B_j des bornes sur les *x-connectées*.

(b) Résoudre le sous-problème suivant :

$$\begin{aligned}
\min_{y \in Y, z} \quad & z \\
& z \geq L(x^{B_j}, y, \lambda^k, \mu^k)|_{x^k}^{lin} \\
& \left.\begin{array}{l} g_i^k(y) \geq 0 \quad \text{si} \quad x_i^{B_j} = x_i^L \\ g_i^k(y) \leq 0 \quad \text{si} \quad x_i^{B_j} = x_i^U \end{array}\right\} \forall i \in I_c^k \\
& (y, z) \in C_R
\end{aligned}$$

i. Si $z^* < UB - \epsilon$, conserver les solutions (y^*, z^*).

ii. Sinon, déclarer la région sondée.

(c) Choisir une nouvelle combinaison B_j des bornes sur les *x-connectées*.

(d) Répéter les étapes (5b) et (5c) jusqu'à ce que toutes les combinaisons des bornes aient été choisies.

6. ***Sélection de la borne inférieure***

   Trouver le minimum de toutes les valeurs de z^k soit z^p, fixer $y^{k+1} \leftarrow y^p$, $LB \leftarrow z^p$.

7. ***Test de convergence***

   Soit ϵ le critère de convergence. Si $\left|\frac{UB-LB}{UB}\right| \leq \epsilon$, ARRÊT. Sinon, $k \leftarrow k+1$, et retourner à l'étape (2)

Déroulement du GOP sur un exemple

Considérons l'application du GOP sur le problème suivant proposé par Falk et Al-Khayyal [3].

$$\begin{aligned} \min_{x,y} \quad & -x + xy - y \\ -6x + 8y & \leq 3, \\ 3x - y & \leq 3, \\ x, y & \geq 0. \end{aligned}$$

Le point de départ est le point $y^1 = 1$.

Tout d'abord, il est possible de resserrer les bornes sur les variables, en utilisant la technique décrite dans la section GOP. Cette étape permet d'affirmer que $0 \leq x, y \leq 1.5$.

Itération 1 :

En partant du point $y^1 = 1$ on construit le problème auxiliaire $(\mathcal{P}_{y^1})$.

$$\begin{aligned} \min_x \quad & -1 \\ -6x \quad & \leq \quad -5, \\ 3x \quad & \leq \quad 4, \\ 0 \quad & \leq x \leq \quad 1.5. \end{aligned}$$

La solution est $x = 0, \mu_1^1 = \mu_2^1 = \mu_3^1 = 0$. Et la valeur objectif de ce problème est -1 ce qui nous donne la borne supérieure de l'algorithme. Le problème est réalisable et une coupe d'optimalité est générée pour être introduite dans les problèmes maîtres :

$$L^1(x, y, \mu^1) = -x + xy - y = xg_1^1(y) - y$$

où $g_1^1(y) = y - 1$ est la contrainte de partitionnement. Étant donné qu'il n'y a qu'une contrainte de partitionnement, il faut créer deux noeuds et résoudre deux sous-problèmes pour cette itération.

Noeud 1

$$\begin{aligned} \min_{y,\mu} \quad & \mu \\ g_1^1(y) = \quad & y - 1 \geq 0, \\ L^1(x^L, y, \mu^1) = \quad & -y \leq \mu, \\ & 0 \leq y \leq 1.5. \end{aligned}$$

et

Noeud 2

$$\begin{aligned} \min_{y,\mu} \quad & \mu \\ g_1^1(y) = \quad & y - 1 \leq 0, \\ L^1(x^U, y, \mu^1) = \quad & 0.5y - 1.5 \leq \mu, \\ & 0 \leq y \leq 1.5. \end{aligned}$$

Le premier problème donne la solution $\mu = -1.5, y = 1.5$ et le deuxième problème donne la solution $\mu = -1.5, y = 0$. La nouvelle borne inférieure doit être fixée à -1.5. Les valeurs optimales étant identiques, on peut continuer la recherche à partir de l'un ou l'autre des noeuds. Prenons par exemple le premier noeud pour exploration future.

Itération 2 :

Pour cette itération, la valeur courante est $y = 1.5$, le problème auxiliaire pour cette itération est réalisable et donne la solution $x = 1.5, \mu_1^2 = 0.083333, \mu_2^2 = \mu_3^2 = 0$ avec la valeur objectif -0.75. Comme la valeur objectif est supérieure à la valeur de la borne supérieure globale, cette solution ne permet pas de resserrer les bornes globales. Le problème étant réalisable, on génère une coupe d'optimalité :

$$L^2(x, y, \mu^2) = -x + xy - y + \frac{1}{12}(-6x + 8y - 3) = xg_1^2(y) - \frac{y}{3} - \frac{1}{4}$$

où $g_1^2(y) = y - 1.5$ est la contrainte de partitionnement de cette itération. il faudra créer deux noeuds et résoudre deux sous-problèmes pour cette itération.

Noeud 3

$$\begin{aligned} \min_{y,\mu} \quad & \mu \\ g_1^1(y) = \quad & y - 1 \geq 0, \\ L^1(x^L, y, \mu^1) = \quad & -y \leq \mu, \\ g_1^2(y) = \quad & y - 1.5 \geq 0, \\ L^2(x^L, y, \mu^1) = \quad & 0.5y1 - 1.5 \leq \mu, \\ & 0 \leq y \leq 1.5. \end{aligned}$$

et

Noeud 4

$$\begin{aligned} \min_{y,\mu} \quad & \mu \\ g_1^1(y) = \quad & y - 1 \geq 0, \\ L^1(x^L, y, \mu^1) = \quad & -y \leq \mu, \\ g_1^2(y) = \quad & y - 1.5 \leq 0, \\ L^2(x^U, y, \mu^1) = \quad & \tfrac{7}{6}y1 - 2.5 \leq \mu, \\ & 0 \leq y \leq 1.5. \end{aligned}$$

La solution du noeud 3 est $y = 1.5$ avec une valeur objectif de -0.75, supérieure à la borne supérieure globale. La solution optimale ne se trouve donc pas dans la sous-région 3 et le noeud est déclaré sondé. La solution du noeud 4 est $y = 1.153846$ avec une valeur objectif de -1.153846.

Remarque :

Le noeud 3 est inutile dans la résolution du problème car la contrainte $g_1^2(y) = y - 1.5 \geq 0$ fixe la variable y à la valeur 1.5. Du coté du noeud 4, la contrainte $g_1^2(y) =$

$y - 1.5 \geq 0$ est redondante car la borne supérieure de y spécifie déjà qu'elle doit être plus petite que 1.5. Le même phénomène se produira à chaque fois que la contrainte de partitionnement ne coupe pas la sous-section dans laquelle elle est générée, c'est-à-dire si l'une des contraintes $g_i^k(y) \geq 0$ ou $g_i^k(y) \leq 0$ est satisfaite sur toute la sous-section étudiée. Pour éviter de générer ces noeuds inutiles, il suffit de vérifier *a priori* que les contraintes de partitionnement ne sont pas des contraintes redondantes. Ce test s'appelle le *test de monotonicité.*

À la fin de cette itération, il y a deux sous-région potentielles, la sous-région correspondant au noeud 2 : $0 \leq y \leq 1$ avec une borne inférieure de -1.5 et la sous-région correspondant au noeud 4 : $1 \leq y \leq 1.5$ avec une borne inférieure de -1.153846. Le GOP sélectionne la sous-région de valeur minimale de façon à améliorer la sous-estimation de la fonction objectif du problème initial. L'algorithme sélectionne donc la sous-région correspondant au noeud 2 et fixe la valeur courante à la solution optimale de ce noeud ($y = 0$).

Itération 3 :

La valeur courante est $y = 0$, le problème auxiliaire pour cette itération donne la solution $x = 1, \mu_2^3 = 0.3333, \mu_1^3 = \mu_3^3 = 0$ avec la valeur objectif -1. Il est à noter qu'il est possible de resserrer les bornes sur la variable x en résolvant ces deux problèmes :

$$\begin{aligned} \min_{x,y} \quad & \pm x, \\ -6x + 8y \quad & \leq \quad 3, \\ 3x - y \quad & \leq \quad 3, \\ & 0 \leq y \leq 1, \\ & 0 \leq x \leq 1.5. \end{aligned}$$

Ceci permet de générer les nouvelles bornes sur x pour ce noeud, soit $0 \leq x \leq \frac{4}{3}$. Le problème étant réalisable, on génère une coupe d'optimalité :

$$L^3(x, y, \mu^3) = -x + xy - y + \frac{1}{3}(3x - y - 3) = xg_1^3(y) - \frac{4}{3}y - 1,$$

où $g_1^3(y) = y$ est la contrainte de partitionnement de cette itération. En appliquant le test de monotonicité, on réalise que $g_1^3(y) \geq 0$ pour toute la région étudiée, un seul noeud est nécessaire pour cette itération. Dans ce sous-problème, il ne faut pas inclure les contraintes générées à l'itération précédente car elles ne sont pas valides dans cette sous-section.

Noeud 5

$$\begin{aligned}
\min_{y,\mu} \quad & \mu \\
g_1^1(y) = \quad & y - 1 \geq 0, \\
L^1(x^L, y, \mu^1) = \quad & -y \leq \mu, \\
g_1^3(y) = \quad & y \geq 0, \\
L^3(x^L, y, \mu^1) = \quad & -\tfrac{4}{3}y - 1 \leq \mu, \\
& 0 \leq y \leq 1.5.
\end{aligned}$$

La solution du noeud 5 est $y = 0.2727$ avec une valeur objectif de -1.3636. À la fin de cette itération, il y a deux sous-régions potentielles, la sous-région relative au noeud 4 et celle du noeud 5. L'algorithme poursuit l'exploration du noeud 5 puisque sa valeur optimale est inférieure à celle du noeud 4.

L'algorithme poursuit ainsi l'exploration des noeuds pendant 18 itérations, avant de converger vers la solution optimale -1.0833 au point $x = 1.16530, y = 0.4959$ avec un critère de tolérance ϵ de 10^{-3}. Rappelons que la tolérance ϵ représente la différence

entre la borne inférieure et la borne supérieure de l'algorithme. On remarque que les tests de monotonicité et le resserrement des bornes est très efficace, car sans ces stratégies de calcul, l'algorithme de base prend 76 itérations pour atteindre la même solution.

Recherche d'un point de départ réalisable

Jusqu'à maintenant, y^1 le point de départ de l'algorithme est considéré comme réalisable. Cette supposition garantit que dès la première itération, les sous-problèmes posséderont une coupe d'optimalité. Le fait de posséder au moins une de ces coupes est important car c'est le seul type de contrainte qui fait intervenir la variable z. Si le point de départ n'est pas réalisable, les sous-problèmes de la première itération ne contiendront pas de coupe d'optimalité et ces problème seront non-bornés ou irréalisables. Les noeuds irréalisables seront sondés et les noeuds non-bornés généreront des solutions optimales ayant peu de signification, car toutes les solutions réalisables sont optimales.

Pour éviter de se retrouver avec des solutions non-bornées, la variable z est bornée artificiellement dans l'implantation de l'algorithme. La valeur de cette borne inférieure artificielle sur z est calibré empiriquement de façon à ce qu'elle soit suffisamment petite pour ne jamais intervenir dans les sous-problèmes contenant des coupes d'optimalités. Ces sous-problèmes non-bornées des premières itérations n'empêcheront pas l'algorithme de trouver la solution optimale. Par contre, en pratique quelques itérations seront nécessaires pour trouver une première solution réalisable. Cette solution réalisable permettra de générer une coupe d'optimalité qui sera valide dans tout le sous-arbre de ce noeud. L'algorithme ayant comme directive de toujours choisir le noeud de valeur minimale, il sera forcé de visiter tous les noeuds où le sous-problème est non borné avant de poursuivre l'exploration des sous-arbres où la solution est

bornée.

Pour éviter de visiter tous ces noeuds, une petite modification à été proposée au GOP initial. Si la solution de départ est irréalisable, le GOP sera démarré une première fois jusqu'à ce qu'il trouve une solution réalisable. Cette solution réalisable sera utilisée comme point de départ pour un second appel du GOP.

3.4 Implantation

L'algorithme GOP tel que présenté précédemment à été implanté dans le langage C++ sous une plateforme Linux. Le logiciel CPLEX 8.1 est utilisé pour résoudre les sous-problèmes linéaires. Le code source est constitué d'environ 6000 lignes réparties dans une dizaine de fichiers. Les commentaires représentent près de 25% du code.

3.5 Améliorations du GOP

La version du GOP présentée ici est celle qui a été programmée dans le cadre de ce projet. C'est une version de base de cette méthode, en effet, plusieurs améliorations ont été apportées à la version de base. Gourdin & Jaumard [14] ont intégré la structure de branchement dans un problème mixte à variables binaires. Floudas et Visweswaran ont aussi proposé des améliorations à cette méthode [10]. L'algorithme αbb basé sur les travaux de Floudas *et al* [2] hérite du GOP. Il permet d'étendre la classe de problème résolus des problèmes quadratiques ou biconvexes aux problèmes non-linéaires deux fois différentiables [2].

αBB permet de résoudre les problèmes de type :

$$\begin{aligned} \min_x \quad & f(x) \\ & g(x) \leq 0 \\ & h(x) = 0 \\ & x \in X \subset \mathbb{R}^n \end{aligned}$$

où x est de taille n et les fonctions f, g et h sont éléments de $\mathcal{C}^2$, l'ensemble des fonctions deux fois diférentiables. Tout comme les améliorations précédente du GOP, l'algorithme αBB utilise une stratégie de branchement. La recherche d'une "bonne" borne inférieure représente le principal effort de cet algorithme. Pour avoir une borne inférieure la plus précise possible, chacun des termes des fonctions f, g et h sera étudié séparément selon sa structure mathématique. Par exemple, la fonction objectif peut se décomposer en cinq parties : linéaire, convexe, bilinéaire, concave à une variable et non linéaire quelconque. La décomposition de la fonction donne le résultat suivant :

$$f(x) = L(x) + C(x) + \sum_{i=1}^{bt} b_i x^{i,1} x^{i,2} + \sum_{i=1}^{ut} U_i(x^i) + \sum_{i=1}^{nt} N_i(x)$$

où $L(x)$ représente les termes linéaires, $C(x)$ les termes convexes, bt le nombre de termes bilinéaires, ut le nombre de termes concave à une variable, $U_i(x^i)$ chacun de ces termes, nt le nombre de termes non-linéaires quelconque et $N_i(x)$ chacun de ces termes. αBB propose une façon de calculer une approximation convexe qui sous-estime la fonction $f(x)$ ce qui rend possible l'application du GOP.

Dans notre modèle du procédé de lavage Kraft, la fonction objectif est très simple, ces améliorations ne sont donc pas utiles dans notre cas.

CHAPITRE 4

ALGORITHME D'OPTIMISATION DE LA CONTRÔLABILITÉ

Ce chapitre présente le point charnière de ce projet, c'est-à-dire le développement et l'application d'une méthode pour permettre d'optimiser la contrôlabilité d'un procédé. Cet algorithme d'optimisation de la contrôlabilité (AOC) se veut assez général pour pouvoir être appliqué à différents procédés.

L'optimisation des coûts d'opération d'un procédé mène bien souvent à un optimum où la contrôlabilité est mauvaise. Tenter de maintenir ce procédé à ce point d'opération sera donc difficile et la stabilité en souffrira. Globalement, il serait peut-être plus efficace de maintenir le procédé à un point voisin du point optimal qui aura une meilleure contrôlabilité et une meilleure stabilité que de tenter de le maintenir au point d'opération optimal.

C'est dans cette optique que nous avons développé l'AOC en collaboration avec Abdelmoula [1]. La méthode utilisée est une stratégie d'optimisation locale de type gradient, qui nécessite donc un point de départ. L'application la plus intéressante de l'AOC est de prendre comme point de départ le point optimal au sens des conditions d'opération et de tenter d'améliorer la contrôlabilité près de ce point optimal. Cependant la méthode proposée peut aussi être utilisée avec un point de départ arbitraire.

4.1 Description de l'algorithme

Dans le chapitre consacré à la contrôlabilité, le RGA a été présenté comme l'indice le plus utilisé pour décrire la contrôlabilité d'un procédé. Le RGA sera donc utilisé ici comme critère dans l'AOC. Comme il a été dit, plus la contrôlabilité d'un procédé est bonne, plus son RGA se rapproche de la matrice identité $I_{n\times n}$. L'AOC aura donc comme objectif de forcer le RGA à se rapprocher de $I_{n\times n}$. Supposons tout d'abord que le jumelage optimal des variables contrôlées et manipulées est connu. Dans ce cas, la fonction objectif de l'AOC sera la somme des carrés des différences entre un et les valeurs de la diagonale du RGA :

$$\min_{\mu}\left\{F(\mu,y,x)=\sum_{i=1}^{n}(1-\lambda_{ii}(\mu,y,x))^2\right\} \tag{4.1}$$

où λ_{ij} représente l'élément ij de du RGA, μ les variables manipulées, y les variables contrôlées et x les autres variables du modèle.

Dans le cas où on ne connaît pas le jumelage optimal, on ne peut pas obtenir l'optimum en une étape. Il faudra pour chaque jumelage trouver la contrôlabilité optimale, puis choisir le meilleur jumelage. Ce qui donne un objectif à deux niveaux. Soit P l'ensemble des jumelages possibles et soit $P_K \in P$ un jumelage quelconque tel que

$(i,j) \in P$ si et seulement si les variables μ_i et y_j sont jumelées dans P_k. Dans ce cas, la fonction objectif de l'AOC sera :

$$\min_{P_K \in P} \left\{ \min_{\mu} \sum_{i=1}^{n} (1 - \lambda_{ij}(\mu, y, x))^2 \right\}. \tag{4.2}$$

Pour la suite de la description de l'algorithme, le jumelage optimal des variables contrôlées et manipulées est supposé connu et la notation λ_{ij} sera utilisé pour désigner la fonction $\lambda_{ij}(\mu, y, x)$.

Pour effectuer la recherche du minimum, une méthode de type gradient a été développée. Rappelons tout d'abord ce qu'est une méthode du gradient.

Méthode du gradient :

Soit $f(x)$ une fonction à valeur réelle, on cherche $\min_x f(x)$.

Soit x^0 le point de départ de l'algorithme, $k = 0$ l'indice de l'itération courante, $\epsilon > 0$ le critère d'arrêt.

1. $k \leftarrow 0$.
2. Calculer $\nabla f(x)|_{x^k}$. Si $|\nabla f(x)|_{x^k}| < \epsilon$, STOP. Sinon passer à l'étape suivante.
3. Trouver α^k tel que $f(x^k) \geq f(x^k - \alpha^k \nabla f(x)|_{x^k})$.
4. Itérer $x^{k+1} \leftarrow x^k - \alpha^k \nabla f(x)|_{x^k}$, $k \leftarrow k+1$ et retourner à l'étape 2.

Le principe de base d'une méthode du gradient est d'améliorer l'objectif petit à petit en se dirigeant dans la direction de plus forte diminution à chaque itération. Un peu comme ferait un ballon qui dévale une montagne. Cette méthode ne trouve pas nécessairement l'optimum global de la fonction. Si la fonction n'est pas convexe, il est possible qu'elle possède des minima locaux. La méthode du gradient va converger

vers un de ces minima locaux sauf dans de rares exceptions.
L'avantage de la méthode du gradient est sa facilité d'implantation et sa rapidité. De plus, des améliorations comme le gradient conjugué permettent à cette méthode d'être encore plus rapide.

Dans notre proposition de l'AOC, le point de départ est la solution de l'optimisation du procédé au sens des coûts d'opérations (μ^0, y^0, x^0). À partir de ce point, l'AOC doit calculer le gradient de F par rapport à μ, les variables manipulées :

$$\nabla F^k = \nabla_\mu F(\mu^k, y^k, x^k). \tag{4.3}$$

A l'aide de ce vecteur, on peut connaître la direction vers laquelle on doit modifier les variables manipulées pour améliorer la contrôlabilité du procédé. Le gradient ∇F^k donne la direction de plus forte croissance de la fonction F. Pour diminuer la fonction, il suffit de se diriger dans la direction inverse du gradient.

Remarque

Dans le cas où le modèle du procédé contient des équations d'égalité linéaires qui concernent les variables manipulées, ces équations forment un sous-espace dans l'espace des variables. Il faut projeter le gradient sur ce sous-espace avant de choisir la direction pour s'assurer d'obtenir une solution réalisable.

Maintenant que le choix de la direction est fait, il faut trouver la bonne valeur du pas de déplacement α^k tel que le procédé sera plus contrôlable si on change les valeurs de μ par $\mu^k - \alpha^k \nabla F^k$. Plus le pas est petit, plus la diminution de la valeur de F sera petite. Plus le pas sera grand, plus le risque de dépasser le minimum est grand. Le choix du pas se fait par une méthode de recherche linéaire (*line search*) ou

par ajustement empirique.

Les variables x et y dépendent de la valeur de μ et on ne peut donc pas passer directement au point suivant en définissant $(\mu^{k+1}, y^{k+1}, x^{k+1})$ par $(\mu^k - \alpha^k \nabla F^k, y^k, x^k)$, ce point peut ne pas être réalisable. Les variables x et y sont donc déterminées en optimisant le procédé où les variables μ ont été fixées à $\mu^k - \alpha^k \nabla F^k$. Cette optimisation se fait au sens des coûts d'opérations et non de la contrôlabilité. Donc si $cout(\mu, y, x)$ représente la fonction coût du procédé en question on aura :

$$(\mu^{k+1}, y^{k+1}, x^{k+1}) \in arg \min_{x,y} \left\{ \begin{array}{ll} cout(\mu, y, x) & \text{sous les contraintes du procédé,} \\ & \text{avec } \mu = \mu^k - \alpha^k \nabla F^k. \end{array} \right\}$$

En résumé l'AOC se décrit comme :

Algorithme d'Optimisation de la Contrôlabilité :

Soient $F(\mu, y, x)$ une fonction à valeur réelle, exprimant la contrôlabilité du procédé, μ les variables manipulées, y les variables contrôlées et x les autres variables du système. On cherche $\min_{\mu,y,x} F(\mu, y, x)$.
Soit k l'indice de l'itération courante, $\epsilon > 0$ le critère de convergence.

1. Poser $k \leftarrow 0$; $\mu^k \leftarrow \mu^0$ le point de départ de l'algorithme.
2. Optimiser le modèle au sens des coûts d'opérations, conserver les valeurs de y^k et x^k, calculer la valeur de la contrôlabilité $F(\mu^k, y^k, x^k)$.
3. Calculer ∇F^k. Si $|\nabla F^k| < \epsilon$, STOP. Sinon passer a l'étape suivante.
4. Calculer la projection du gradient ∇F^k_{proj} sur le sous-espace des contraintes d'égalités.
5. Trouver α^k le pas de déplacement.
6. Itérer : $\mu^{k+1} \leftarrow \mu^k - \alpha^k \nabla F^k_{proj}$, $k \leftarrow k + 1$ et retourner à l'étape 2.

4.2 Validation de l'algorithme

Pour une première validation de l'algorithme AOC, un procédé beaucoup plus simple que le procédé de lavage sera utilisé. Dans un procédé simple où la contrôlabilité est définie analytiquement, la contrôlabilité optimale est connue. Il sera donc possible de prévoir la solution que devrait trouver l'AOC sur cet exemple.

Le procédé qui a été choisi pour valider l'algorithme est un mélange homogène. Soit deux courants, l'un avec une concentration forte et l'autre avec une concentration faible d'alcool qui sont mélangés dans le but d'obtenir une concentration moyenne.

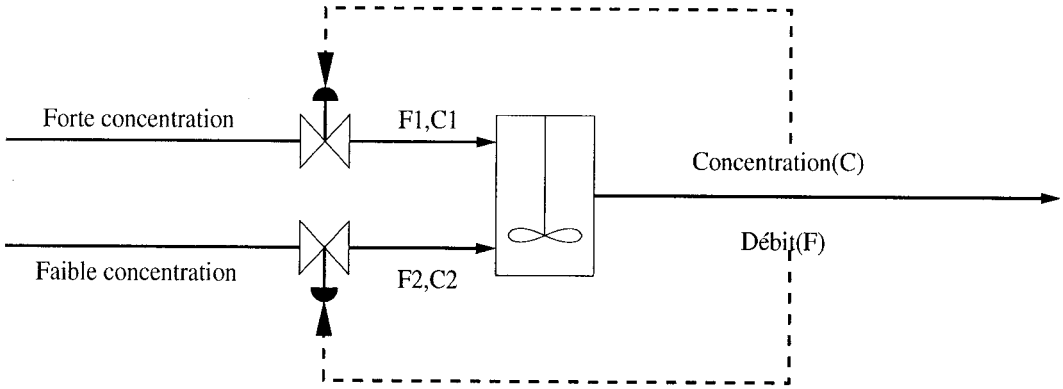


Figure 4.1 – Schéma du procédé de mélange

Les équations qui régissent ce procédé sont :

$$F = F_1 + F_2,$$
$$C = C_1 \left(\frac{F_1}{F_1 + F_2} \right) + C_2 \left(\frac{F_2}{F_1 + F_2} \right).$$

F_1 est le débit du courant fort et C_1 sa concentration, F_2 est le débit du courant faible et C_2 sa concentration. Enfin, F est le débit total et C la concentration du mélange. Pour ce procédé, les variables contrôlées sont les variables du mélange F et C. Les variables manipulées sont les débits des deux courants initiaux F_1 et F_2. Les variables

de concentration C_1 et C_2 représentent les autres variables du système. Étant donné ces équations on peut calculer la matrice des gains relatifs (RGA) :

$$RGA = \begin{pmatrix} \frac{F_1}{F_1+F_2} & \frac{F_2}{F_1+F_2} \\ \frac{F_2}{F_1+F_2} & \frac{F_1}{F_1+F_2} \end{pmatrix}. \tag{4.4}$$

Remarque

Dans cet exemple, le RGA est une matrice 2×2. Sachant que la somme sur les lignes et les colonnes de la matrice donne toujours un, il est suffisant de calculer un seul élément pour connaître la matrice au complet, c'est-à-dire :

$$RGA = \begin{pmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{pmatrix}.$$

Dans ce cas, il suffit de chercher λ_{11} le plus près possible de un pour améliorer la contrôlabilité du procédé. L'objectif peut donc se résumer à :

$$\min_{\mu} F(\mu, y, x) = (1-\lambda_{11})^2.$$

Supposons maintenant que le liquide à forte concentration d'alcool est très coûteux par rapport au liquide à faible concentration. L'optimisation du procédé au sens des coûts d'opérations tentera de diluer au maximum ce courant avec de l'eau. On aura

le problème d'optimisation des coûts d'opérations suivant :

$$\begin{aligned}
\min_{C_1,C_2} \quad & C_1, \\
& F = F_1 + F_2, \\
& C = C_1\left(\frac{F_1}{F_1+F_2}\right) + C_2\left(\frac{F_2}{F_1+F_2}\right), \\
& \underline{C_1} \leq C_1 \leq \overline{C_1}, \\
& \underline{C_2} \leq C_2 \leq \overline{C_2}.
\end{aligned}$$

Remarque

La contrainte $F = F_1 + F_2$ ne concerne que les variables manipulées. Il faudra donc projeter le gradient sur cette droite avant de calculer le pas de déplacement. Si on ne fait pas cette projection, les valeurs de F_1 et F_2 de l'itération suivante ne seront pas nécessairement réalisables

L'AOC a été appliqué à l'instance suivant de ce problème de mélange homogène.

On veut produire 10 L/min d'un mélange d'alcool à 65% de concentration en utilisant un courant fort et un courant faible. Plus précisément, la concentration C_1 du courant fort est entre 70% et 90% selon le besoin et celle du courant faible C_2 est entre 5% et 15%. Plus la concentration du courant fort est forte, plus il est coûteux. Pour optimiser le procédé au sens des coûts d'opérations, on tentera de minimiser C_1, la concentration du courant fort. Les concentrations des courants sont au départ de 80% et de 10%. Les deux équations du modèle permettent de calculer les valeurs des débits initiaux soit $F_1 = 7.8571$L/min et $F_2 = 2.1428$L/min. Maintenant que l'on connaît les valeurs des variables manipulées, il est possible de détailler la première

itération de l'AOC.

1. *Optimiser le modèle au sens des coûts d'opérations, conserver les valeurs de y^k et x^k, calculer la valeur de la contrôlabilité $F(\mu^k, y^k, x^k)$.*

   Résoudre le problème linéaire suivant :

$$\begin{aligned} \min_{C_1,C_2} \quad & C_1, \\ & C_1\left(\frac{F_1}{F_1+F_2}\right) + C_2\left(\frac{F_2}{F_1+F_2}\right) = 10, \\ & F_1 = 7.8571, \\ & F_2 = 2.1428. \end{aligned}$$

   La solution optimale de ce problème est $C_1 = 78.63\%$ et $C_2 = 14.99\%$. Conserver ses valeurs ainsi que $F = 10$ et $C = 65\%$ le débit et la concentration du mélange final.

   Comme le RGA est connu de façon analytique, on peut calculer facilement la valeur de la fonction objectif :

$$(1-\lambda_{11})^2 = (1 - \frac{F_1}{F_1+F_2})^2 = 0.0459.$$

2. *Calculer ∇F^k. Si $|\nabla F^k| < \epsilon$, STOP. Sinon passer a l'étape suivante.*

   Pour cet exemple, le calcul de ∇F^k, peut se faire de façon analytique ou peut être estimeé par différences finies. Pour le bien de l'exemple, le calcul a été fait par différences finies et a donné $\nabla F^k = (-0.0092, 0.0337)$.

3. *Projection du gradient sur le sous-espace des contraintes d'égalités.*

   Dans ce modèle, la contrainte $F = F_1 + F_2$ ne concerne que les variables manipulées. Il faut projeter ∇F^k sur cette droite avant de passer au prochain point.

$\nabla F^k_{proj} = (-0.0214, 0.0214)$.

4. *Trouver α^k le pas de déplacement.*

   Nous avons choisi un pas constant $\alpha^k = 2$

5. *Itérer : $\mu^{k+1} \leftarrow \mu^k - \alpha^k \nabla F^k_{proj}$, $k \leftarrow k+1$ et retourner à l'étape 1.*

   $F_1 = 7.8571 - 2 * (-0.0214) = 7.8999$,

   $F_2 = 2.1428 - 2 * (0.0214) = 2.1001$.

Après la première itération, le débit du courant fort a augmenté et sa concentration a diminué, passant de 80% à 78.63%. Le débit du courant faible a diminué et sa concentration a augmenté, passant de 10% à 15%, sa valeur maximale. Les résultats des itérations suivantes sont illustrés dans le graphique 4.2.

L'optimisation de la contrôlabilité du procédé de mélange a poussé les variables de concentration à leur borne inférieure soit 70% pour le courant fort et 5% pour le courant faible. La valeur de la fonction de contrôlabilité est passée de 0.0459 à 0.0059 ce qui confirme une nette amélioration de la contrôlabilité. Ces résultats étaient prévisibles étant donné la simplicité du modèle.

L'AOC a permit d'améliorer la contrôlabilité du procédé de mélange. En contrepartie, le débit du courant fort a augmenté et les coûts d'opérations du procédé ont augmenté. Ce qui montre bien qu'un gain en contrôlabilité s'accompagne généralement par une perte au sens des coûts d'opérations. Reste maintenant à l'opérateur du procédé à établir ses priorités et à faire un compromis.

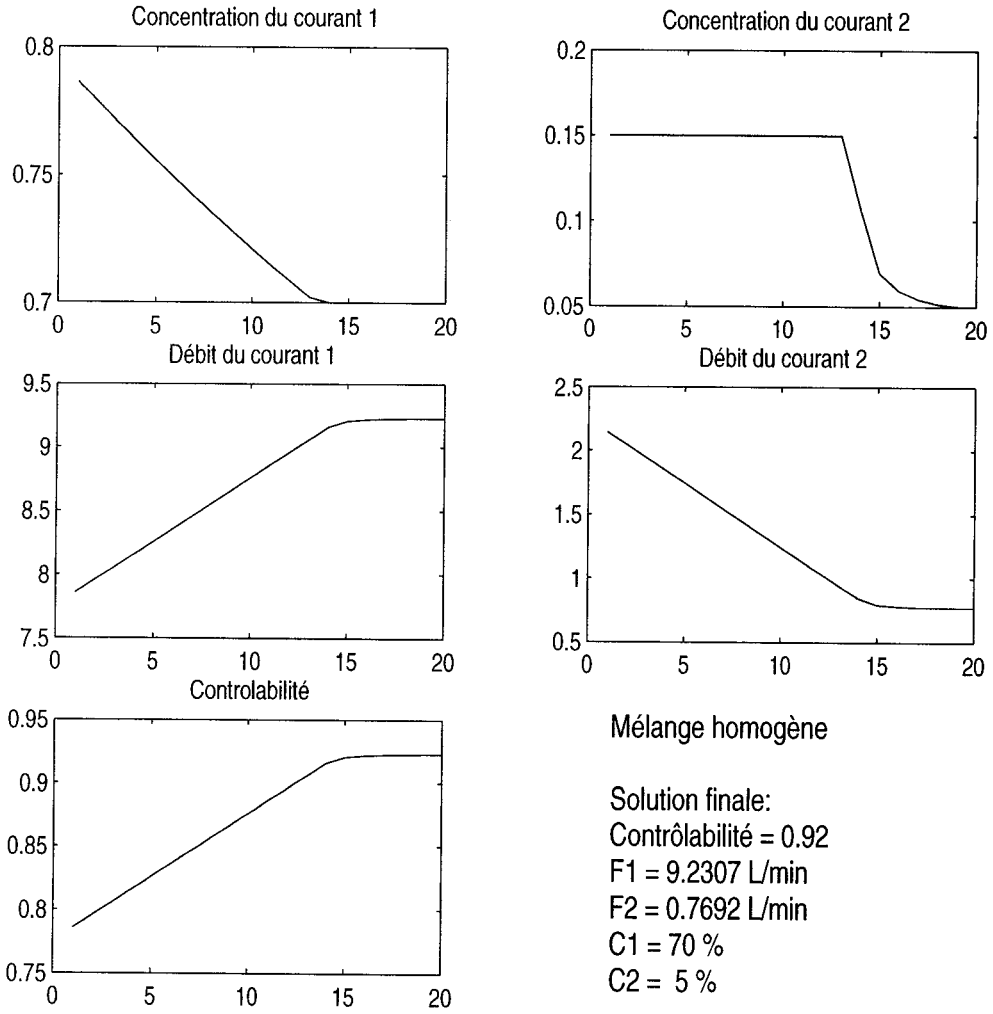


Figure 4.2 – Résultats du procédé de mélange

CHAPITRE 5

RÉSULTATS DE CALCULS

Ce chapitre présente les résultats numériques du projet. Une première partie contient la validation du GOP sur quelques problèmes types, l'évaluation des performances du GOP et le choix de certains paramètres. La seconde partie présente les résultats de l'application du GOP sur le modèle de lavage de pâte kraft pour deux modules de lavage. La troisième partie contient la calibration et les résultats de l'application de l'algorithme d'optimisation de la contrôlabilité sur le problème de lavage.

5.1 Validation et performance du GOP

Le GOP à été testé sur quelques problèmes tirés de la littérature, le tableau 5.1 donne un aperçu de la résolution de ces problèmes.

Pour qualifier la taille du problème, le nombre de variables et le nombre de contraintes sont indiqués dans la colonne taille. La précision ϵ est aussi indiquée dans le tableau ainsi que le nombre d'itérations nécessaires pour atteindre la solution. La dernière colonne représente le temps de calcul de l'algorithme GOP et de l'algorithme QP de

Audet et al.[5] utilisé dans le travail de Abdelmoula [1]. Les problèmes ont été testés sur un Sunfire 4800 tournant à 900 Megahertz et possédant 2 gigaoctet de mémoire vive. D'après le tableau 5.1, le GOP est capable de résoudre à optimalité, en un temps

Tableau 5.1 – Résultats du GOP sur des problèmes tests de la littérature

Problème	Taille	optimum	ϵ	itérations	GOP	QP
Falk [3]	2v,2c	-1.083	1e-3	18	0.45 sec	0.19 sec
Konno [19]	4v,6c	-13	1e-3	18	0.46 sec	0.07 sec
Hesse [17]	12v,13c	-174	1e-3	26	2.99 sec	0.16 sec
Haverly [16]	9v,7c	-400	1e-3	12	0.52 sec	0.10 sec
Sherali Shetty [31]	4v,7c	9	1e-2	32	1.10 sec	12.92 sec

raisonnable, plusieurs problèmes tirés de la littérature. Il est toutefois précipité de dire que le GOP tel que programmé est un algorithme efficace et performant. D'ailleurs, les deux dernières colonnes permettent d'illustrer que l'algorithme QP est généralement plus rapide que le GOP.

Point de départ

Contrairement à QP, le GOP nécessite un point de départ de l'algorithme. Pour certains problèmes, le choix du point de départ peut avoir un impact important sur le temps de calcul. Le tableau 5.2 présente

Tableau 5.2 – Sensibilité du GOP par rapport au point de départ sur le problème soland.

Point de départ			Réalisable	CPU	Itérations
Point	y_1	y_2			
optimum	0.7081	1.4970	-	-	-
1	0.75	1.3671	×	49.81 s	327
2	0.6	1.8	×	3.97 s	52
3	0.1	1	×	2.58 s	38
4	0.9	2	×	45.13 s	334
5	2	3	×	31.85 s	275

Tableau 5.3 – Sensibilité du GOP par rapport au point de départ sur le problème de laveur.

Point de départ				Réalisable	CPU	Itérations
Point	y_1	y_2	y_3			
optimum	1.5713	0.4391	0.2309	-	-	-
1	1.7714	0.3492	0.67754	×	357.4 s	185
2	2.4	0.2	0.6	×	24.7 s	24
3	1.5	0.6	0.7	×	6668.8 s	2660
4	4	0	0.9	×	7596.7 s	2980
5	9	6	0.9	×	7521.3 s	2980
solution réalisable	1.5050	0.3685	0.4208	√	20.9 s	9

5.2 Optimisation du procédé de lavage

Le modèle du procédé de lavage tel que décrit dans la section 2 comporte plusieurs paramètres. Pour permettre au modèle de se rapprocher le plus possible de la réalité, les valeurs des paramètres sont tirées d'un exemple de la littérature : le modèle de Felder [9].

Le cas de base

Plusieurs paramètres interviennent dans les modèles décrits précédemment. Le choix de ces paramètres est important, un bon calibrage de ces paramètres permet au modèle de représenter fidèlement la réalité. Un mauvais choix des paramètres mène à des résultats non-significatifs peu importe la qualité du modèle.

Un cas d'étude d'un système de lavage tiré de la littérature à été choisi pour paramétrer les modèles décrit précédemment. Le cas de Felder [9] est très réaliste : les valeurs des différents débits, fractions et taux de recirculation sont du même ordre que ce qui existe en industrie. Felder [9] présente dans son ouvrage un modèle de trois laveurs en série. Les valeurs des paramètres sont données dans le tableau 5.4.

Le cas de base présenté ici, tout comme le modèle du chapitre 2, contient trois

Tableau 5.4 – Cas de base

Courants					Modules		
Nom	Débit	% pâte	% solides dissous	% eau	# module	DR	α
ψ_{in}	10050 L/min	13.93%	19.01%	67.06%	1	0.9396	0.8453
ϕ_{M1F1}	99857.35 L/min	1.40%	12.32%	86.28%	2	0.8530	0.8857
$\phi_{filt.1}$	106240.45 L/min	0%	11.57%	88.43%	3	0.7703	0.8735
ϕ_{F1M2}	9491.97 L/min	14.75%	2.46%	82.79%			
ϕ_{M2F2}	132575.76 L/min	1.05%	1.47%	97.48%			
$\phi_{filt.2}$	138958.85 L/min	0%	1.39%	98.61%			
ϕ_{F2M3}	9314.06 L/min	15.03%	0.59%	84.38%			
ϕ_{M3F3}	117746.01 L/min	1.19%	0.29%	98.52%			
$\phi_{filt.3}$	124129.10 L/min	0%	0.27%	99.73%			
ψ_{fw}	15655 L/min	0%	0%	100%			
ψ_{out}	9271.52 L/min	15.10%	0.014%	84.886%			

modules de lavage. Abdelmoula [1] a montré que le comportement du modèle était similaire pour deux ou trois modules de lavages. Nous avons utiliser un modèle à deux modules pour effectuer les expérimentations. Ce choix est motivé par la lente convergence du GOP sur le modèle et sa sensibilité au point initial.

Le tableau 5.5 donne un aperçu de l'impact sur la fonction objectif de la modifications des principaux paramètres

Tableau 5.5 – Sensibilité du modèle par rapport aux paramètres α et DR.

Paramètres		Objectif	CPU	Itérations
α_1	α_2			
0.790	0.81	27820.54	9.23	16
0.845	0.885	20005.70	4398	1784
0.86	0.90	18824.89	36.1	84
DR_1	DR_2			
0.934	0.848	20726.14	29.29	40
0.939	0.853	20005.70	4398	1784
0.944	0.858	19761.81	3392	4351

D'après le tableau 5.5, les coûts d'opérations du procédé diminuent lorsque les taux de recirculations (α_i) augmentent ou lorsque l'efficacité des laveurs (DR_i) augmente. Pour la suite des expériences numériques, les paramètres α_1, α_2, DR_1 et DR_2 seront fixés respectivement au valeurs suivantes : 0.845, 0.885, 0.939 et 0.853. Par contre, pour ce qui est du temps de résolution, il n'y a pas de liens direct entre les

paramètres et le temps de calcul. On peut remarquer d'énorme différence dans les temps de résolutions des différents cas.

5.3 Optimisation de la contrôlabilité du procédé de lavage (AOC)

Dans la section 4.2, nous avons démontré que l'AOC donnait des résultats satisfaisants pour un petit exemple pratique, soit un exemple de mélange homogène. L'AOC sera maintenant appliqué sur un procédé beaucoup plus complexe, soit sur le procédé de lavage kraft tel que modélisé à la section 2.

Cette section présente tout d'abord les variables en jeux, puis les adaptations nécessaires pour appliquer l'AOC sur le procédé de lavage et les détails d'implantation. Par la suite, les différents paramètres de l'AOC seront calibrés pour pouvoir présenter enfin les résultats de l'optimisation de la contrôlabilité.

5.3.1 Couplage des variables contrôlées et manipulées

Pour calculer la contrôlabilité du procédé de lavage, il faut tout d'abord choisir parmi les variables du modèle quelles seront les variables manipulées et les variables contrôlées. Nous avons choisi de contrôler le débit de production (Ψ_{out}) ainsi que la qualité de la pâte lavée (f^s_{out}), c'est-à-dire la fraction de solides dissous à la sortie. Du coté des variables manipulées, deux débits ont été choisis : le débit de pâte à l'entrée (Ψ_{in}) et le débit d'eau fraîche (Ψ_{fw}).

Le choix de ces variables a été motivé par le fait que le débit de production ainsi que la qualité de la pâte lavée doivent rester stables pour ne pas perturber les procédés qui suivent le lavage de la pâte, par exemple le procédé de blanchiment.

Le débit d'eau fraîche peut être ajusté facilement à l'aide de vannes de même que le débit de pâte à l'entrée. Ces deux débits sont donc des candidats idéaux pour jouer le rôle des variables manipulées. Pour le couplage des variables, deux situations sont possibles, contrôler le débit de pâte à la sortie à l'aide du débit de pâte à l'entrée et contrôler la fraction de solides à la sortie à l'aide du débit d'eau fraîche, ou faire le contraire. La figure 5.1 illustre ces deux choix. Intuitivement, il est plus naturel d'opter pour la première option. Les expériences numériques qui suivent confirmeront ce choix de couplage des variables.

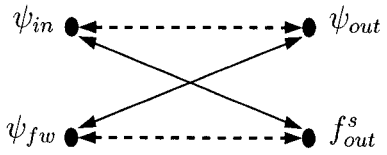


Figure 5.1 – Couplage des variables manipulées et contrôlées

Tout comme dans l'exemple du mélange homogène, le RGA du procédé de lavage est une matrice 2×2. Dans ce cas, l'élément λ_{11} du RGA est suffisant pour décrire la contrôlabilité. La fonction de contrôlabilité sera donc réduite à $F = (1 - \lambda_{11})^2$.

5.3.2 Approximation par différences finies.

Dans la section 4.2, le nombre d'équations et de variables du modèle était réduit et le RGA a pu être calculé de façon analytique. L'expression analytique du RGA a été utilisée pour calculer ∇F^k, le gradient de la fonction de contrôlabilité. Par contre, dans le cas du procédé de lavage de la pâte kraft, le RGA ne possède pas d'expression analytique. Cette difficulté provient du fait que le jacobien des variables contrôlées par rapport aux variables manipulées ne possède pas non plus d'expression analytique. Il faut calculer d'abord une approximation du jacobien avant de calculer une

approximation du gradient de la fonction de contrôlabilité. Nous avons donc à faire face à deux niveaux d'approximation par différences finies.

Les modifications nécessaires pour permettre d'appliquer l'algorithme AOC au procédé de lavage se situent principalement à l'intérieur de l'étape (3) de l'algorithme AOC, celle qui calcule ∇F^k, le gradient de la fonction de contrôlabilité.

Pour le procédé de lavage, nous devrons évaluer

$$\bar{\nabla} F^k \approx \nabla F^k = \left(\left. \frac{\partial F(\mu, y, x)}{\partial \mu_1} \right|_{\mu^k}, \left. \frac{\partial F(\mu, y, x)}{\partial \mu_2} \right|_{\mu^k} \right)$$

une approximation de la valeur du gradient au point μ^k. Cette évaluation devra être séparée en plusieurs phases.

Calcul de $\bar{\nabla} F^k$

1. **Approximation de $\left.\frac{\partial F}{\partial \mu_i}\right|_{\mu^k}$ pour $i = 1, 2$:**

   $$\left.\frac{\partial F}{\partial \mu_i}\right|_{\mu^k} = 2(\lambda_{11}(\mu^k) - 1) \left(\left.\frac{\partial \lambda_{11}(\mu)}{\partial \mu_i}\right|_{\mu^k} \right)$$
   $$\approx 2(\lambda_{11}(\mu^k) - 1) \left(\frac{-\lambda_{11}(\mu^k+2\delta_i)+4\lambda_{11}(\mu^k+\delta_i)-3\lambda_{11}(\mu^k)}{|\delta_i|} \right).$$

   δ_i est choisi suffisamment petit pour garantir une bonne approximation.

2. **Calcul de $\lambda_{11}(\mu^k)$:**

   $\lambda_{11}(\mu^k) = \frac{J_{11}(\mu^k)J_{22}(\mu^k)}{J_{11}(\mu^k)J_{22}(\mu^k)-J_{12}(\mu^k)J_{21}(\mu^k)}$ où $J_{ij}(\mu^k)$ est l'élément en position i, j du Jacobien des variables contrôlées par rapport aux variables manipulées $\left.\frac{\partial y_i}{\partial \mu_j}\right|_{\mu^k}$. C'est-à-dire la dérivée partielle de y_i par rapport à μ_j au point μ^k.

   Le calcul de $\lambda_{11}(\mu^k + \delta_i)$ et de $\lambda_{11}(\mu^k + 2\delta_i)$ se fait de façon similaire.

3. **Approximation de $J_{ij}(\mu^k)$ et $J_{ij}(\mu^k + \delta_i)$:**

   $J_{ij}(\mu^k) = \frac{\partial y_i}{\partial \mu_j}|_{\mu^k} \approx \frac{-y_i(\mu^k+h_i)+4y_i(\mu^k+h_i)-3y_i(\mu^k)}{|h_i|}$

   Les valeurs de $y_i(\mu^k)$, $y_i(\mu^k + h_i)$ et $y_i(\mu^k + 2h_i)$ sont obtenues en optimisant le modèle du procédé de lavage pour μ fixé.

   Remarque : Le pas utilisé dans l'approximation de λ ne doit pas être plus précis que celui utilisé pour J_{ij}, c'est-à-dire $h < \delta$.

Dans la procédure proposer pour le calcul de $\bar{\nabla} F^k$, les dérivées partielles ont été estimées par des différences finies décentrées d'ordre deux. Il aurait aussi été possible d'utiliser des différences finies décentrées d'ordre un, mais ces approximations donnent une précision numérique relativement faible. Des différences finies centrées d'ordre deux sont aussi envisageables.

Pour notre projet, nous avons utilisé une approximation décentrée d'ordre 2 pour chacun des deux niveaux de différences finies. Nous avons fait ce choix à cause du

statut particulier de la variable Ψ_{fw} qui représente la consommation d'eau fraîche. Cette variable est une variable manipulée et elle est en même temps dans la fonction objectif du modèle. Comme la valeur initiale des variables manipulées est donnée par l'optimisation du modèle, il est très probable que le modèle sera irréalisable pour une valeur de Ψ_{fw} inférieure à celle retournée par l'optimisation. De la même façon, il est probable que le modèle sera irréalisable pour une valeur de ψ_{in} supérieure à la valeur optimale. Le choix de la direction du sens de la décentralisation est fait en fonction du principe précédemment décrit.

5.3.3 Implantation

L'algorithme d'optimisation de la contrôlabilité à été programmé à l'aide du langage C++ sur une plateforme Linux-Unix. La conception de L'AOC à été réalisé en collaboration avec Abdelmoula [1] alors l'implantation de L'AOC est le fruit de son travail. Le code initial de l'AOC utilisait un algorithme d'optimisation quadratique à contraintes quadratiques. Nous l'avons adapté pour permettre l'utilisation du GOP comme outil d'optimisation.

5.3.4 Paramètres de l'AOC

Comme il a été dit précédemment, l'application de l'AOC sur le procédé de lavage utilise deux niveaux d'approximation par différences finies. Le premier niveau de différences finies est l'approximation du calcul du jacobien des variables contrôlées par rapport aux variables manipulées. Le deuxième niveau est l'approximation du calcul du gradient de la fonction de contrôlabilité. Le choix des pas pour ces deux niveaux de calculs est primordial pour avoir une bonne qualité numérique du résultats.

Tableau 5.6 – Différents points de départs pour l'AOC

Point de départ μ^0	Ψ_{in}	Ψ_{fw}	coûts d'opération	
			1 module	2 modules
1	8624	15650	27240	18550
2	6525	13540	25130	16200
3	10050	22630	34220	25180
4	10050	24800	36390	27300

Un autre paramètre important de l'AOC est le pas de déplacement. Les méthodes de types gradient utilisent généralement un pas pour modifier l'amplitude du déplacement dans la direction optimale. Le choix judicieux de ce paramètre a surtout un impact sur le nombre d'itérations nécessaires avant d'atteindre la solution optimale.

Choix des pas de différence finie

Pour calibrer ces deux paramètres, nous avons testé l'algorithme avec quatre points de départ réalisables. Le tableau 5.6 présente les points de départs ainsi que leurs coûts d'opérations respectifs et ce, pour un et deux modules.

Le tableau 5.7 présente les modifications de la contrôlabilité et de la direction de descente par rapport aux différentes combinaisons de pas. D'après le tableau 5.7, le choix des pas h et δ n'influence pas ou très peu l'évaluation de la contrôlabilité. Comme on peut le constater, pour un même point de départ, les valeurs de λ_{11} et $F(\lambda_{11})$ sont pratiquement les mêmes peu importe les choix de pas. Par contre, la direction de déplacement est plus influencée par la modification des pas. Dans le cas où le point de départ assure une bonne contrôlabilité, les directions de descente sont toutes assez petites et confirment donc que la solution est près de l'optimum. Dans les cas où la contrôlabilité du point de départ est mauvaise, les directions sont plus sensibles aux choix des pas. Après analyse des résultats numériques, les valeurs de h et δ équivalent à 10 et 100 respectivement paraissent donner les résultats les plus

Tableau 5.7 – Calibration des pas h et δ

			2 modules			
Départ	h	δ	λ_{11}	$F(\lambda_{11})$	∇_1	∇_2
	5	30	1.0132	1.731e-4	-3.45e-5	2.05e-5
	5	50	1.0132	1.731e-4	-9.59e-5	5.71e-5
	10	80	1.0132	1.731e-4	-2.48e-4	1.46e-4
1	10	100	1.0132	1.731e-4	-2.61e-4	8.36e-4
	20	200	1.0132	1.731e-4	-1.98e-3	-3.16e-3
	50	500	1.0132	1.731e-4	-5.58e-2	1.26e-2
	5	30	1.0169	2.859e-4	3.31e-3	-3.16e-3
	5	50	1.0169	2.859e-4	5.46e-3	-1.79e-2
2	10	80	1.0114	1.291e-4	-1.36e-2	9.61e-3
	10	100	1.0114	1.291e-4	-1.12e-2	4.20e-2
	20	200	1.0130	1.699e-4	-1.94e-2	-9.27e-2
	50	500	1.0157	2.452e-4	0.224	-6.46e-2
	5	30	0.1469	0.7276	-9.8e-6	1.92e-6
	5	50	0.1469	0.7276	-0.274	5.91e-6
3	10	80	0.1469	0.7276	-2.48	2.34e-4
	10	100	0.1469	0.7276	-7.13	-1.03
	20	200	0.1469	0.7276	-3.94	1.61
	50	500	0.1469	0.7276	33.5	-0.968
	5	30	0.1470	0.7276	-4.36e-5	1.02e-07
	5	50	0.1470	0.7276	-7.25e-5	2.81e-07
4	10	80	0.1470	0.7276	-1.16e-4	7.31e-07
	10	100	0.1470	0.7276	-1.45e-4	1.14e-06
	20	200	0.1467	0.7280	0.13	-0.13
	50	500	0.1473	0.7270	-0.164	0.391

stables. Ces valeurs sont donc adoptées pour la suite des tests numériques.

Commentaire

Les résultats du modèle pour un module ne sont pas représentatifs de la réalité. Ceci découle de la dégénérescence du problème bilinéaire généré pour le cas d'un module de lavage. Pour ce cas, le problème se réduit à un système d'équation contenant 11 équations et 13 variables qui possède donc une infinité de solutions équivalentes. Pour forcer le modèle à donner une solution unique, une solution envisageable est d'ajouter une perturbation dans l'objectif.

$$\omega_{fw} \cdot \Psi_{fw} + \omega_{out} \cdot f^s \Psi_{out} + \epsilon \cdot x_j,$$

Où x_j est une autre variable du modèle. Mais ceci n'est pas suffisant pour garantir la validité du modèle dans le cas où il n'y a qu'un module de lavage.

Choix du pas de déplacement

Après avoir calculé la direction de descente (d_1, d_2), on doit calculer le pas de déplacement α. Les méthodes de type recherche linéaire (*line search*) permettent de trouver de bons pas de déplacement. Toutefois, ces méthodes demandent des évaluations successives de la fonction objectif. Dans notre cas, l'évaluation de la fonction objectif est très coûteuse en temps de calcul, c'est pourquoi ces méthodes ont étés rejetées au profits de méthodes qui demandent peu d'évaluations de la fonction. Une bonne méthode de remplacement est un pas de déplacement décroissant qui permet de stabiliser le processus. Par exemple un pas du type $\alpha_k \approx \frac{\alpha_0}{k^p}$ où k représente l'indice des itérations, α_0 le pas initial et p un paramètre de puissance généralement entre 0.5 et 1.

Une particularité de notre problème est que nous connaissons à l'avance l'optimum désiré, idéalement le RGA devrait se rapprocher de l'identité et donc $\lambda_{11}^* = 1$. Cette particularité peut être utilisée pour pondérer le pas de déplacement décroissant. Cette pondération permet d'accélérer l'approche de l'optimum et de minimiser les déplacements à proximité de cet optimum. Le choix de la méthode pour calculer le pas de déplacement s'est finalement arrêté sur une méthode mixte qui combine le pas décroissant et une pondération proportionnelle à la distance par rapport à l'optimum :

$$\alpha_k = |1 - \lambda_{11}| \frac{\alpha_0}{k^p}.$$

Pour les expériences de calcul, le paramètre de puissance est fixé à 0.5 et le pas initial est fixé à 1000. Pour éviter que le point suivant ne se retrouve dans une région irréalisable, les variables du débit de pâte et du débit d'eau fraîche sont bornées. Si le point suivant atteint ou dépasse les bornes, le pas de déplacement sera réduit jusqu'à ce qu'il soit à l'intérieur des bornes.

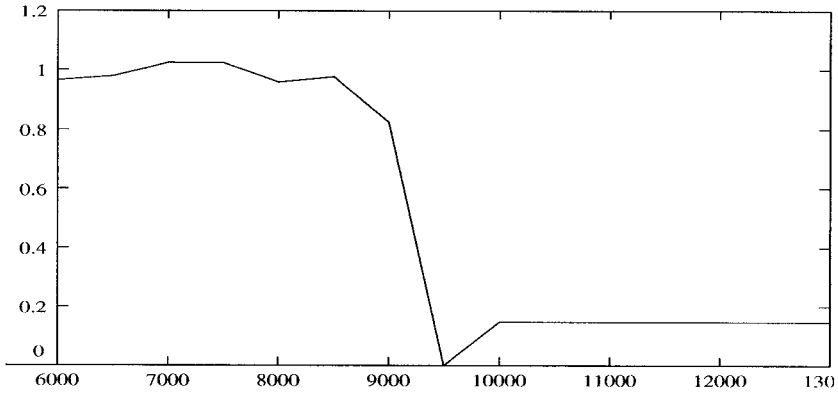
5.3.5 Résultats de l'AOC

Pour étudier son comportement, l'AOC a été testé sur plusieurs instances du problème de lavage. Les expériences de calcul ont été faites sur le modèle pour 2 modules de lavage. En variant le point de départ, l'AOC montre deux types de comportement :

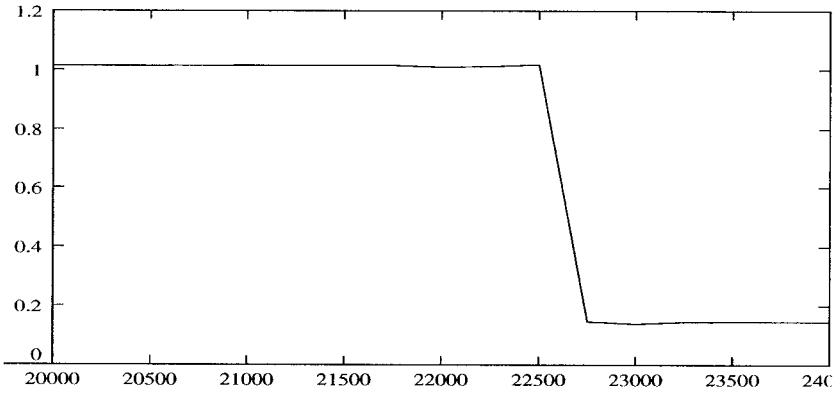
1. Le point de départ possède une bonne contrôlabilité, c'est-à-dire entre 1 et 1.1. Dans ce cas, l'algorithme piétine autour de ce point et finit par converger près du point initial ayant lui aussi une bonne contrôlabilité. Les graphiques 5.3 et 5.4 illustrent l'évolution de l'algorithme pour ce premier comportement.

2. La contrôlabilité du point de départ est mauvaise, c'est-à-dire entre 0.1 et 0.2. Dans ce cas, si l'AOC réussit à sortir de la région de mauvaise contrôlabilité, il y aura un saut dans les valeurs de λ_{11}, ψ_{out} et ψ_{fw}. Les courbes 5.5 et 5.6 illustrent le comportement de l'algorithme dans cette situation.

La figure 5.2 contient deux graphiques qui illustrent la variation de l'indice de contrôlabilité par rapport au débit de pâte et par rapport au débit d'eau. Ces deux graphiques montrent que l'indice de contrôlabilité se situe principalement sur deux paliers, l'un autour de 1 et l'autre autour de 0.1. Dans les deux graphiques la situation est semblable, c'est l'augmentation du paramètre libre qui fait chuter dramatiquement la valeur de l'indice. Une analyse approfondie à permis de découvrir que la valeur de l'indice se situe autour de 0.1 lorsque l'une des bornes sur les débits des courants est actives.

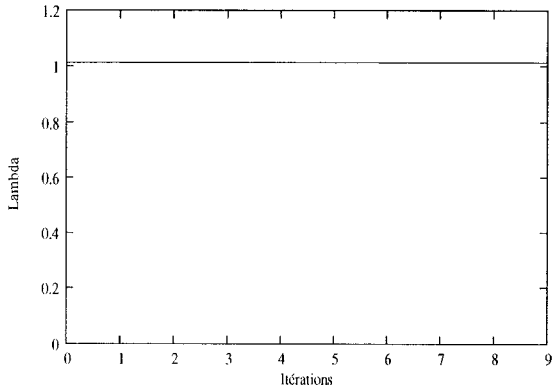


(a) Variation de l'indice de contrôlabilité en fonction du débit de pâte pour un débit d'eau = 22630 L/min

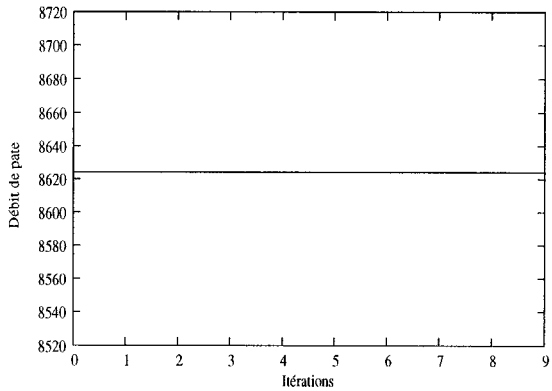


(b) Variation de l'indice de contrôlabilité en fonction du débit d'eau pour un débit de pâte = 10000 L/min

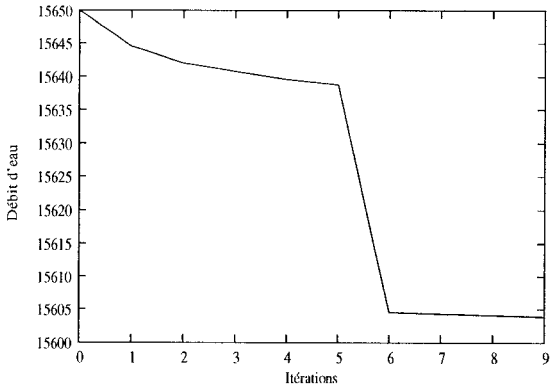
Figure 5.2 – Variation de la valeur de λ.



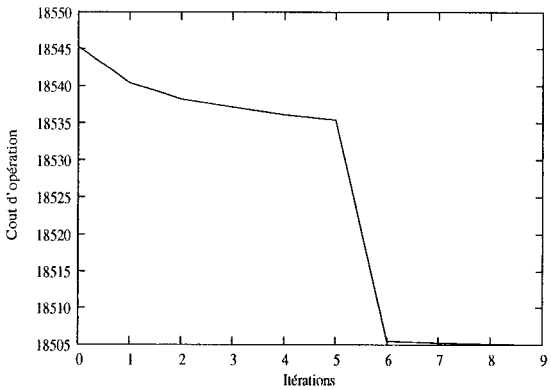
(a) Lambda



(b) Débit de pâte

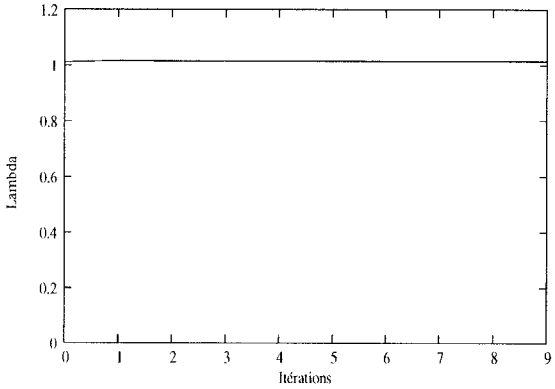


(c) Débit d'eau fraîche

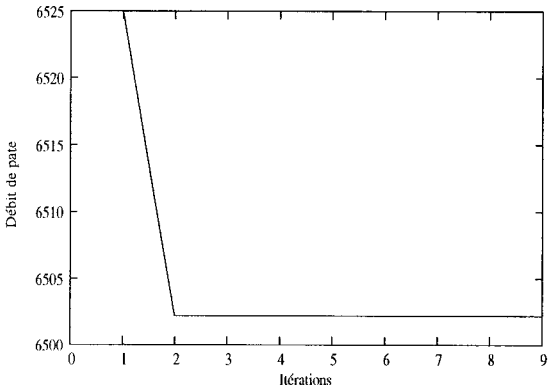


(d) Coût d'opération

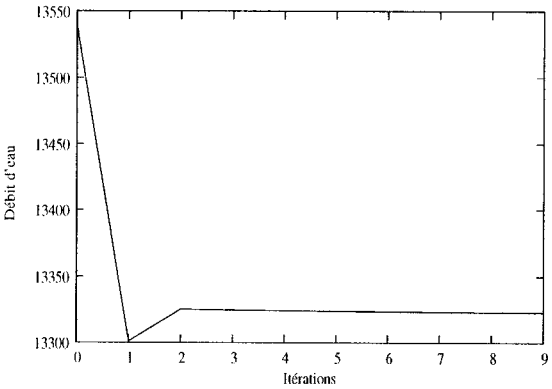
Figure 5.3 – Résultats de l'AOC, point de départ 1.



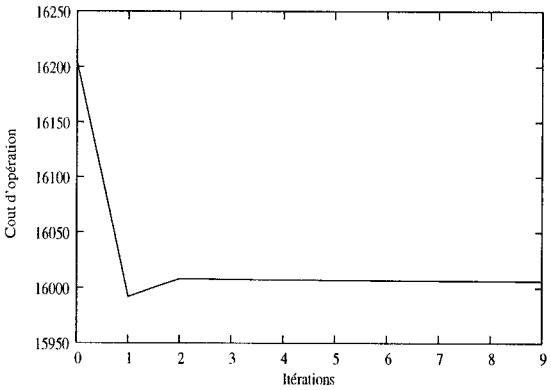
(a) Lambda



(b) Débit de pâte

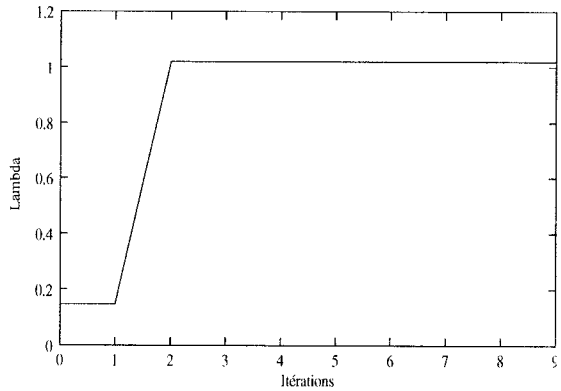


(c) Débit d'eau fraîche

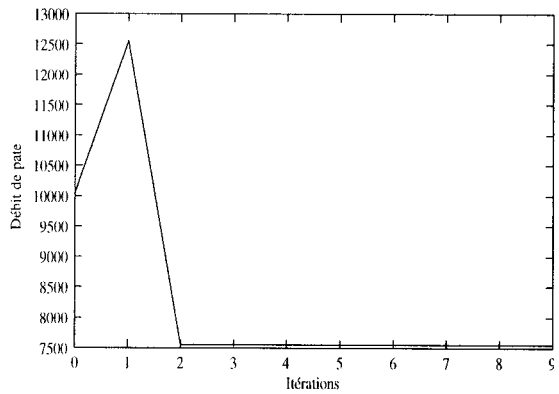


(d) Coût d'opération

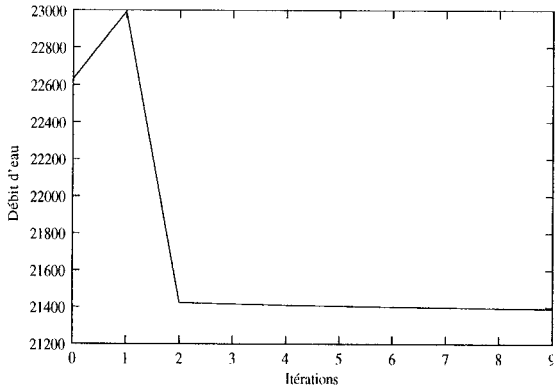
Figure 5.4 – Résultats de l'AOC, point de départ 2.



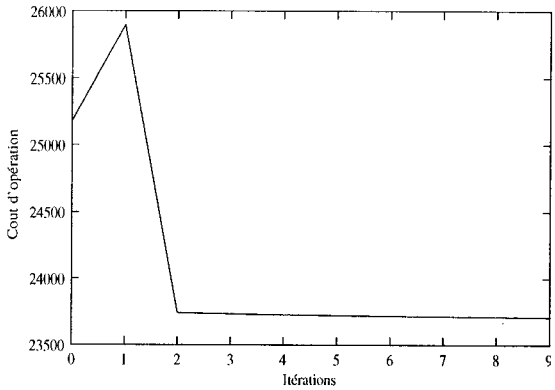
(a) Lambda



(b) Débit de pâte

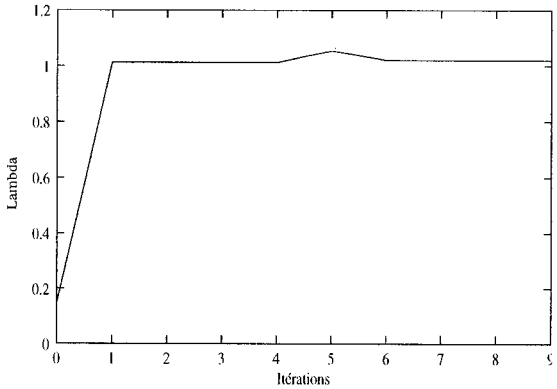


(c) Débit d'eau fraîche

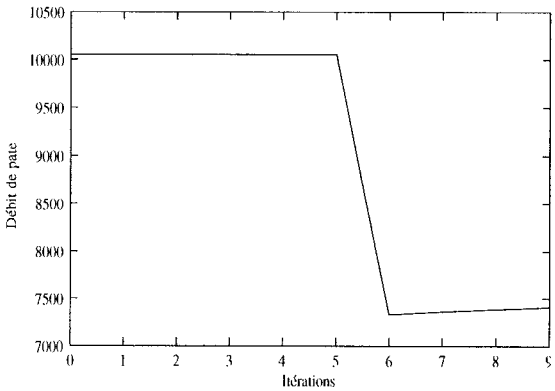


(d) Coût d'opération

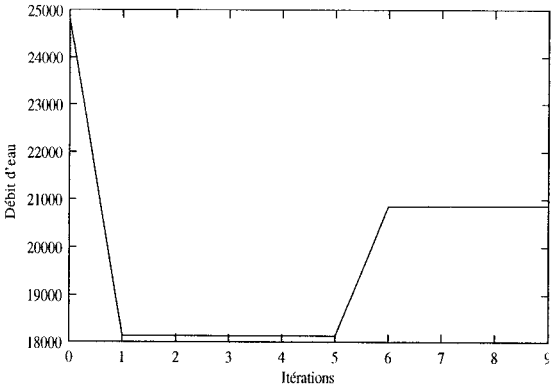
Figure 5.5 – Résultats de l'AOC, point de départ 3.



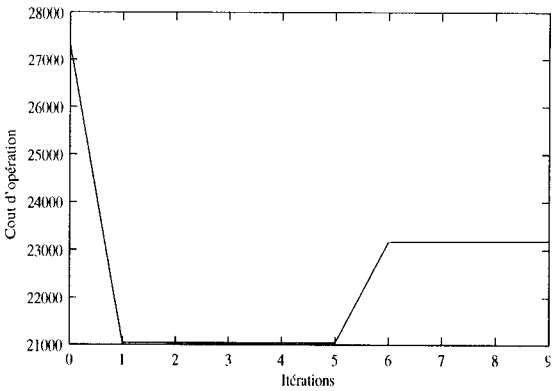
(a) Lambda



(b) Débit de pâte



(c) Débit d'eau fraîche



(d) Coût d'opération

Figure 5.6 – Résultats de l'AOC, point de départ 4.

Analyse des résultats :

Comme le montrent les graphiques 5.3 à 5.6, l'AOC a réussi à améliorer la contrôlabilité du procédé de lavage ou à la conserver dans le cas où elle était déjà excellente. Dans les deux derniers cas, l'indice de la contrôlabilité λ_{11} effectue un saut avant d'atteindre une valeur près de la valeur recherchée. Dans tous les cas, le coût d'opération du procédé diminue au fil des itérations. Dans les quatre solutions présentées, les variables contrôlées et manipulées sont inférieures ou égales à leur valeurs de départ. Ceci concorde avec le graphique 5.2 qui illustre que lorsque les variables manipulées atteignent de grande valeurs, la contrôlabilité chute et atteint un plancher autour de 0.1.

Selon l'AOC, il existe plusieurs configurations du système de lavage de la pâte kraft ayant une très bonne contrôlabilité. Parmi ces configurations, le choix de la meilleure peut donc se faire du point de vue des coûts d'opérations du procédé.

CHAPITRE 6

DISCUSSION

Le but principal de ce projet est d'intégrer l'étude des critères de contrôlabilité à même l'étape de conception du procédé. Ceci constitue un objectif ambitieux, l'unification de ces deux étapes généralement isolées ne peut se faire qu'à petit pas. Ce projet multidisciplinaire représente un de ces pas.

Dans ce projet, des méthodes d'optimisation globales ont été utilisées pour résoudre un problème de contrôlabilité appliqué à un problème particulier de l'industrie des pâtes et papiers. Le côté pluridisciplinaire de ce projet est en soit une réalisation.

Pour situer les contributions de ce projet, il faut l'évaluer par rapport à chacun des champs de connaissance auquel il est rattaché.

Tout d'abord, dans le domaine des pâtes et papiers, Felder [9] a été un des premiers à réaliser une bonne modélisation du procédé de lavage Kraft. Il a proposé une modélisation dont nous nous sommes inspirés pour construire le modèle mathématique simplifié du procédé de lavage de pâte Kraft, c'est-à-dire sans les bassins d'entreposage. De plus un second modèle équivalent au premier a été développé pour faciliter la résolution à l'aide des algorithmes d'optimisation. Ce modèle adapté pré-

sente l'avantage de contenir beaucoup moins de variables et de contraintes bilinéaires que le modèle inspiré par Felder.

Plusieurs travaux ont exploré le contrôle du procédé de lavage de pâte Kraft, soit sur des exemples théoriques [32, 28], ou même des implantations industrielles [24]. Ces travaux ont mis en lumière le fait que les boucles de recirculation et les bassins sont les principales causes des problèmes de contrôle. Par contre, l'étude de la contrôlabilité de ces procédés demeurait une zone très obscure. L'estimation de la contrôlabilité du procédé de lavage de pâte Kraft représente une avancée de ce projet.

Les bassins ont une grande importance du point de vue du contrôle comme l'a démontré Roche *et al.* [32]. Par contre, ces bassins sont vides de sens lorsqu'on effectue une étude statique du procédé. L'élimination des bassins d'entreposage dans le modèle simplifié, s'explique par le fait que l'étude du procédé statique est toujours préalable à l'étude dynamique. Il était donc préférable de compléter le projet avec un modèle sans bassins avant de passer au contexte dynamique.

Du point de vue de la recherche opérationnelle, ce projet utilise deux algorithmes d'optimisation combinés. Un algorithme d'optimisation locale est utilisé pour optimiser la contrôlabilité qui doit être calculée à l'aide d'un algorithme d'optimisation globale. Le jumelage de ces outils forme le côté original du projet.

L'implantation en C++ de l'algorithme d'optimisation globale GOP représente une part majeure de l'effort apporté à la production de ce mémoire. Une stratégie de resserrement des bornes sur les variables inspiré des travaux de Audet *et al.*[5] a été intégrée au GOP pour diminuer le nombre d'itérations nécessaires à la résolution des problèmes. La stratégie de recherche de point de départ réalisable ajouté dans le pré-traitement du GOP permet quant à elle de garantir que le GOP ne piétinera pas au départ dans une région non-réalisable. L'algorithme GOP tel que présenté dans la section 3.3 représente une version de base. Plusieurs progrès théoriques dont certains

sont décrits dans la section 3.5 ont été apportés à cet algorithme depuis sa création. La majorité de ces modifications permettant de diminuer le temps de résolution et il pourrait être avantageux d'inclure ces modifications dans une version future de l'algorithme.

La contribution principale de ce projet est le développement de l'AOC. Cet algorithme permet d'optimiser la contrôlabilité d'un procédé en variant les conditions d'opérations.

L'algorithme AOC tel que développé permet d'optimiser la contrôlabilité d'un procédé ayant deux variables contrôlées et deux variables manipulées (2×2). Cette particularité limite l'étude à une gamme de procédés relativement simples. Pour étendre la gamme des procédés possibles, l'AOC devra considérer plus de variables contrôlées et manipulées. Les base théoriques de cette généralisation entamé dans la section 4.1 doivent être concrétisées avant de généraliser l'AOC à des procédés $n \times n$ ou même $n \times m$.

Ce projet se veut un lien entre les trois champs de compétence sur lesquels il s'est appuyé. Un lien qui sera solidifié par les études futures entre ces domaines. Bien entendu, des améliorations sont nécessaires avant de pouvoir qualifier ce lien de véritable pont.

CONCLUSION

En conclusion, ce mémoire est principalement consacré à l'élaboration et l'application d'une méthode pour optimiser la contrôlabilité d'un procédé. Le procédé de lavage de la pâte à papier kraft a été choisi pour évaluer les performances de la méthode. Le modèle bilinéaire décrivant le comportement d'un système de lavage de pâte à papier kraft a été construit conjointement avec Abdelmoula [1]. L'algorithme d'optimisation de la contrôlabilité (AOC) a été appliqué sur un exemple test ainsi que sur le modèle du système de lavage. Dans les deux cas, l'algorithme a réussit à améliorer la contrôlabilité du procédé.

Pour appliquer l'AOC sur le procédé de lavage, nous avons dû implanter et adapter un algorithme d'optimisation bilinéaire.

En ce qui concerne les travaux futurs, plusieurs améliorations sont possibles.

- Il serait très intéressant d'appliquer l'AOC sur un autre procédé plus complexe du domaine du génie chimique. Les colonnes à distiller sont par exemple des procédés très intéressants qui ont déjà été modélisés.
- La modélisation du procédé de lavage est incomplète dans le cas particulier où il n'y a qu'un seul module de lavage. Une étude plus approfondie du procédé serait nécessaire pour identifier les améliorations à apporter à ce modèle.
- La généralisation de l'AOC à des procédés possédant un plus grand nombre de variables manipulées et contrôlées est nécessaire. Les procédés 2×2 sont

intéressants mais ne regroupent qu'une partie des procédés du génie chimique.

Bibliographie

[1] M. ABDELMOULA. Méthode de linéarisation pour l'optimisation sous contraintes de contrôlabilité du procédé de lavage de pâte à papier. Mémoire de maîtrise, École Polytechnique de Montréal, 2003.

[2] C.S. ADJIMAN, I.P. ANDROULAKIS, C.D. MARANAS, et C.A. FLOUDAS. A global optimization method, αBB, for process design. *Computer and chemical engineering*, 20, 1996.

[3] F.A. AL-KHAYYAL et J.E. FALK. Jointly Constrained Biconvex Programming. *Mathematics of Operations Research*, 8(2) :273–286, 1983.

[4] C. AUDET. *Optimisation globale structurée : propriétés, équivalences et résolution.* Thèse de doctorat, École Polytechnique de Montréal, 1997.

[5] C. AUDET, B. JAUMARD, P. HANSEN, et G. SAVARD. A Branch and Cut Algorithm for Nonconvex Quadratically Constrained Quadratic Programming. *Mathematical Programming*, 87(1), 2000.

[6] J.F. BENDERS. Partitioning Procedure for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4, 1962.

[7] E. H. BRISTOL. New Measure of Interaction for Multivariable Process Control. *IEEE Transactions on Automatic Control*, AC-11(1) :133–134, 1966.

[8] Cartons et Celluloses COPACEL : Confédération Francaise de l'Industrie des Papiers. Histoire du papier-carton. *http ://www.copacel.fr*, (Page consultée le 17 juillet 2003).

[9] R.M. FELDER et R.W. ROUSSEAU. *Interactive Chemical Process Principles.* Wiley, Toronto, second edition, 1986.

[10] C.A. FLOUDAS. *Deterministic Global Optimization Theory, Methods and Applications.* Kluwer Academic Publishers, Boston, 2000.

[11] C.A. FLOUDAS et V VISWESWARAN. A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs–II. Theory. *Mathematics of Operations Research*, 8(2) :273–286, 1983.

[12] A.M. GEOFFRION. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10 :237, 1972.

[13] F. GLOVER et M. LAGUNA. *Tabu Search.* Kluwer Academic Publishers, Toronto, July 1997.

[14] E. GOURDIN. *Optimisation Globale des Problèmes Bilinéaires et Lipschitziens.* Thèse de doctorat, École Polytechnique de Montréal, 1994.

[15] P. GROSDIDIER et M. MORARI. Closed-Loop Properties from Steady-State Gain Information. *Ind. Eng. Chem. Fundam.*, 24(2), 1985.

[16] C.A. HAVERLY. Studies of the Behavior of Recursion for the Pooling Problem. *Computer and Operations Research*, 24 :1097–1102, 1997.

[17] R.A. HESSE. A Heuristic Search Procedure for Estimating a Global Solution of Nonconvex Programming Problems. *Operations Research*, 21 :1267–1280, 1973.

[18] R. HORST et H. TUY. *Global Optimization : Deterministic Approaches.* Springer-Verlag, Berlin, second edition, 1992.

[19] H. KONNO. A Cutting Plane Algorithm for Solving Bilinear Programs. *Mathematical programming*, 11 :14–27, 1976.

[20] Leon S. LASDON. *Optimization Theory for Large Systems.* Macmillan Series in Operations Research, New York, 1970.

[21] O.L. MANGASARIAN. *Nonlinear Programming.* McGraw–Hill, New York, 1969.

[22] K.G. MURTY et S.N. KABADI. Some NP-Complete Problems in Quadratic and Non-Linear Programming. *Mathematical programming*, 39 :117–129, 1987.

[23] A. NIEDERLINSKI. A Heuristic Approach to the Design of Linear Multivariable Interacting Control Systems. *Automatica*, 7 :691–701, 1971.

[24] A. NOEL, M. SAVOIE, H. BUDMAN, et L. LAFONTAINE. Advance Brownstock Washer Control :Succesful Industrial Implementation at James Maclaren. *Tappi Journal*, 79(1) :121–127, january 1996.

[25] H. V. NORDEN et M. JARVELAINEN. *Kemian Teollisuus*, 23(4) :344, 1996.

[26] HANSEN P. et JAUMARD B. Reduction of Indefinite Quadratic Programs to Bilinear Programs. *Journal of Global Optimization*, 2 :41–60, 1992.

[27] HANSEN P. et N. MLADENOVIC. Variable Neighbourhood Search. *Computer and Operations Research*, 24 :1097–1102, 1997.

[28] TURNER P.A., B.J. ALLISON, et J.K. OEI. Brownstock Washer Filtrate Tank Level Control. *Pulp & Paper Canada*, 94(11) :45–48, 1993.

[29] P.M. PARDALOS et G. SCHNITGER. Checking Local Optima for Constrained Quadratic Programming is NP-Hard. *Operations Research Letters*, 7 :33–35, 1988.

[30] J.K. PERKIN. *Tappi Journal*, 37(3) :83–89, 1954.

[31] H.D. SHERALI et C.M. SHETTY. A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Faces Cuts. *Mathematical programming*, 19 :14–31, 1980.

[32] P.A. TURNER, A.A. ROCHE, J.D. MCDONALD, et A.R.P. VAN HEININGEN. Dynamic Behavior of a Brownstock Washing System. *Pulp & Paper Canada*, 94(9) :37–42, 1993.

[33] H. TUY. *Convex Analysis and Global Optimization.* Kluwer academic publishers, London, 1998.

[34] D. WHITLEY. A Genetic Algorithm Tutorial. *Statistics and Computing*, 4 :65–85, 1994.

[35] Z.X. ZHU et A. JUTAN. RGA as a Measure for Integrity for Decentralised Control Systems. *Trans IChemE*, 74, January 1996.

Annexe A

Reformulation du modèle

Définition 2 (Reformulation) *Soient P_A et P_B deux classes de problème d'optimisation. Une reformulation $B(\cdot)$ de P_A en P_B est une transformation de P_A à P_B telle que pour tout problème A de P_A ainsi qu'une solution optimale de $B(A)$, on peut obtenir une solution optimale de A en un temps polynomial.*

Soit A une instance du problème de base et $B(\cdot)$ la reformulation du modèle de base vers le modèle adapté décrite au chapitre 2. Soit x_b une solution optimale du modèle adapté $B(A)$ alors le changement de variable suivant permet de créer une solution optimale x_a du modèle de base A.

$$\left.\begin{aligned} f^f\phi_{ij} \cdot \phi_{ij} &= \phi_{ij}^f \\ f^s\phi_{ij} \cdot \phi_{ij} &= \phi_{ij}^s \\ f^w\phi_{ij} \cdot \phi_{ij} &= \phi_{ij}^w \end{aligned}\right\} \quad \forall \phi_{ij} \text{ dans le modèle.}$$

Montrons tout d'abord que x_a est une solution réalisable du modèle de base A.

Les variables de débits totaux sont les mêmes dans les deux modèles, les contraintes

de bilans matières sont aussi les mêmes dans les deux modèles. Si elles sont respectées dans le modèle adapté $B(A)$ elles le sont aussi dans le modèle de base A. Les équations sur les bilans partiels de A sont respectés pour les mélangeurs, par exemple le mélangeur 2.

$$f^f\phi_{F1M2}\cdot\phi_{F1M2}+f^f\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^f\phi_{M2F2}\cdot\phi_{M2F2}=0$$
$$f^s\phi_{F1M2}\cdot\phi_{F1M2}+f^s\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^s\phi_{M2F2}\cdot\phi_{M2F2}=0$$
$$f^w\phi_{F1M2}\cdot\phi_{F1M2}+f^w\phi_{filt.2}\cdot\alpha_2\cdot\phi_{filt.2}-f^w\phi_{M2F2}\cdot\phi_{M2F2}=0$$

Ces bilans sont respectés grâce au changement de variable et à ces équations du modèle adapté $B(A)$:

$$\phi^f_{F1M2}+\alpha_2\cdot\phi^f_{filt.2}-\phi^f_{M2F2}=0$$
$$\phi^s_{F1M2}+\alpha_2\cdot\phi^s_{filt.2}-\phi^s_{M2F2}=0$$
$$\phi^w_{F1M2}+\alpha_2\cdot\phi^w_{filt.2}-\phi^w_{M2F2}=0$$

De même, on démontre que les équations de bilans partiels des autres mélangeurs et des filtres sont respectées.

Les contraintes de complémentarité sont aussi respectées, puisque :

$$\phi^f_{ij}+\phi^s_{ij}+\phi^w_{ij}=\phi_{ij}$$
$$\Longrightarrow\frac{\phi^f_{ij}}{\phi_{ij}}+\frac{\phi^s_{ij}}{\phi_{ij}}+\frac{\phi^w_{ij}}{\phi_{ij}}=1\quad\text{si }\phi_{ij}\neq 0$$
$$\Longrightarrow f^f\phi_{ij}+f^f\phi_{ij}+f^w\phi_{ij}=1$$

et ce, pour toutes les variables ϕ_{ij}.

Les contraintes concernant les DR sont respectées car elles sont déjà incluses dans $B(A)$.

Les bornes sur les variables :

Toutes les variables de A qui sont déjà dans $B(A)$ respectent implicitement leurs bornes. Les variables de fractions de A qui ne sont pas dans $B(A)$ respectent aussi leurs bornes grâce aux contraintes de bornes incluses dans $B(A)$.

$$\begin{aligned}
&\phi_{ij} \cdot lb_{ij}^f \leq \phi_{ij}^f \leq \phi_{ij} \cdot ub_{ij}^f \\
\Longrightarrow &lb_{ij}^f \leq \frac{\phi_{ij}^f}{\phi_{ij}} \leq ub_{ij}^f \quad \text{Si} \quad \phi_{ij} \neq 0 \\
\Longrightarrow &lb_{ij}^f \leq f^f \phi_{ij} \leq ub_{ij}^f.
\end{aligned}$$

Nous avons donc démontré qu'a partir d'une solution optimale x_b du modèle adapté $B(A)$ il est possible de construire en un temps polynomial une solution qui respecte toute les contraintes du modèle de base A et donc une solution réalisable de A.

Montrons maintenant que la solution x_a est une solution optimale de A.

Si x_a n'est pas une solution optimale de A alors il existe une solution x'_a de valeur moindre. À l'aide de cette solution x'_a, il est possible de construire une solution x'_b de B en utilisant le changement de variable.

démontrons tout d'abord que x'_b est une solution réalisable de $B(A)$ car cette solution respecte toutes les contraintes de $B(A)$.

Les équations sur les bilans totaux sont respectées car elles sont déjà incluses dans

le problème A. Les équations sur les bilans partiels sont respectés pour les mélangeurs, prenons par exemple le mélangeur 2.

$$\phi^f_{F1M2} + \alpha_2 \cdot \phi^f_{filt.2} - \phi^f_{M2F2} = 0$$
$$\phi^s_{F1M2} + \alpha_2 \cdot \phi^s_{filt.2} - \phi^s_{M2F2} = 0$$
$$\phi^w_{F1M2} + \alpha_2 \cdot \phi^w_{filt.2} - \phi^w_{M2F2} = 0$$

Ces bilans sont respectés grâce à ces équations du modèle de base A

$$f^f \phi_{F1M2} \cdot \phi_{F1M2} + f^f \phi_{filt.2} \cdot \alpha_2 \cdot \phi_{filt.2} - f^f \phi_{M2F2} \cdot \phi_{M2F2} = 0$$
$$f^s \phi_{F1M2} \cdot \phi_{F1M2} + f^s \phi_{filt.2} \cdot \alpha_2 \cdot \phi_{filt.2} - f^s \phi_{M2F2} \cdot \phi_{M2F2} = 0$$
$$f^w \phi_{F1M2} \cdot \phi_{F1M2} + f^w \phi_{filt.2} \cdot \alpha_2 \cdot \phi_{filt.2} - f^w \phi_{M2F2} \cdot \phi_{M2F2} = 0$$

et au changement de variables.

De même, on démontre que les équations de bilans partiels des autres mélangeurs et des filtres sont respectées.

Les contraintes de complémentarité sont aussi respectées, puisque :

$$f^s \phi_{ij} + f^w \phi_{ij} + f^f \phi_{ij} = 1$$
$$\Longrightarrow \phi_{ij} \cdot (f^s \phi_{ij} + f^w \phi_{ij} + f^f \phi_{ij}) = \phi_{ij}$$
$$\Longrightarrow \phi^f_{ij} + \phi^s_{ij} + \phi^w_{ij} = \phi_{ij}$$

et ce, pour toutes les variables ϕ_{ij}.

Les contraintes concernant les DR sont respectées car elles sont déjà incluses dans le problème A.

Les contraintes bilineaires

$$f^s\phi_{ij} \cdot \phi_{ij} = \phi^s_{ij} \quad \forall f^s\phi_{ij} \quad \text{qui interviennent dans les contraintes des } DR.$$

sont implicitement respectées.

Les contraintes de bornes sur les variables $f^f\phi_{ij}$ sont respectées étant donné leurs constructions.

$$\begin{aligned} &\phi_{ij} \cdot lb^f_{ij} \leq \phi^f_{ij} \leq \phi_{ij} \cdot ub^f_{ij} \\ \Longrightarrow &lb^f_{ij} \leq \frac{\phi^f_{ij}}{\phi_{ij}} \leq ub^f_{ij} \quad \text{Si} \quad \phi_{ij} \neq 0 \\ \Longrightarrow &lb^f_{ij} \leq f^f\phi_{ij} \leq ub^f_{ij}. \end{aligned}$$

Les bornes sur les variables :

Toutes les variables qui proviennent du modèle A respectent leurs bornes puisqu'elles sont les mêmes que dans le modèle A. Les bornes sur les nouvelles variables ϕ^f_{ij}, ϕ^s_{ij} et ϕ^w_{ij} sont respectées :

$$lb^f_{ij} \cdot lb_{ij} \leq f^f\phi_{ij} \cdot \phi_{ij} = \phi^f_{ij} \leq ub^f_{ij} \cdot ub_{ij}$$

puisque tous ces nombres sont positifs.

x'_b est donc une solution réalisable de $B(A)$

Si on note la fonction objectif de A comme f_a et la fonction objectif de $B(A)$ comme f_b alors la définition de ces fonctions nous permet d'affirmer que $f_a(x_a) = f_b(x_b)$ et donc que $f_a(x'_a) = f_b(x'_b)$. Si x_a n'est pas une solution optimale de A alors il existe x'_a tel que $f_a(x_a) > f_a(x'_a)$. Dans ce cas $f_b(x_b) > f_b(x'_b)$, ce qui revient à dire

que x_b n'est pas la solution optimale de B tel que supposé au début de la preuve.

Nous avons donc démontré qu'a partir d'une solution optimale x_b du modèle adapté $B(A)$ il est possible de construire en un temps polynomial une solution optimale du modèle de base A.

Annexe B

Lemme de Farkas

Nous présentons ici une des nombreuses version du lemme de Farkas ainsi que sa démonstration.

Lemme 2 (Lemme de Farkas) *soit* $A \in \mathbb{R}^{mxn}$, $\boldsymbol{b} \in \mathbb{R}^m$ *alors*

$$\exists x \geq 0, \quad Ax \geq \boldsymbol{b} \quad \Longleftrightarrow \quad (\forall u \geq 0, \quad u^t A \leq 0 \Longrightarrow u^t \boldsymbol{b} \leq 0)$$

Considérons le problème d'optimisation $(\mathcal{P})$ et son dual $(\mathcal{D})$:

$$(\mathcal{P}) \quad \begin{array}{ll} \min\limits_{x} & \mathbf{0}^t x \\ & Ax \geq \boldsymbol{b} \\ & x \geq 0 \end{array} \qquad (\mathcal{D}) \quad \begin{array}{ll} \max\limits_{u} & u^t \boldsymbol{b} \\ & u^t A \leq \mathbf{0}^t \\ & u \geq 0 \end{array}$$

Supposons la partie gauche de la proposition vraie, alors

$$\begin{aligned}
&\exists x \geq 0, \quad Ax \geq \boldsymbol{b} \\
\Longrightarrow &\min\{\mathbf{0}^t x | Ax \geq \boldsymbol{b}, x \geq 0\} = 0 \\
\Longrightarrow &\max\{u^t b | u^t \geq 0, u^t A \leq 0\} \leq 0 \\
\Longrightarrow &\forall u \geq 0, \quad u^t A \leq 0 \Longrightarrow u^t \boldsymbol{b} \leq 0.
\end{aligned}$$

Démontrons maintenant la réciproque

$$\begin{aligned}
&\forall u \geq 0, \quad u^t A \leq 0 \Longrightarrow u^t \boldsymbol{b} \leq 0 \\
\Longrightarrow &\max\{u^t b | u^t \geq 0, u^t A \leq 0\} \leq 0 \\
\Longrightarrow &(\mathcal{D}) \text{ est réalisable borné} \\
\Longrightarrow &(\mathcal{P}) \text{ est réalisable borné} \\
\Longrightarrow &\exists x \geq 0, \quad Ax \geq \boldsymbol{b}.
\end{aligned}$$

Annexe C

x-connectées

Nous présentons ici la démonstration de la pertinence des variables *x-connectées* dans le formalisme de Floudas et Visweswaran [11]

Théorème 3 (x-connectées) *La solution du problème de minimisation interne* ($\mathcal{IRD}$) *avec le lagrangien remplacé par sa linéarisation ne dépend que des variables x-connectées.*

Reformulation de la linéarisation du lagrangien

La linéarisation du lagrangien autour de x^k est définie comme suit :

$$\begin{aligned} L(x,y,\lambda^k,\mu^k)\big|_{x^k}^{lin} &= L(x^k,y,\lambda^k,\mu^k) + \nabla_x L(x,y,\lambda^k,\mu^k)\big|_{x^k}^{lin}(x-x^k) \\ &= L(x^k,y,\lambda^k,\mu^k) + \sum_{i=1}^{n} \nabla_{x_i} L(x,y,\lambda^k,\mu^k)\big|_{x^k}^{lin}(x_i-x_i^k) \end{aligned}$$

Les variables x qui sont indépendantes des y gagnent à être regroupées dans une fonction $\Psi_2(x,\lambda^k,\mu^k)$. Ce qui donne l'expression de la linéarisation du lagrangien suivante :

$$L(x,y,\lambda^k,\mu^k)\big|_{x^k}^{lin} = \Phi_1(y,\lambda^k,\mu^k)\Psi_1(x) + \Psi_2(x,\lambda^k,\mu^k) + \Phi_2(y,\lambda^k,\mu^k).$$

Dans le cas où le lagrangien est bilinéaire, la linéarisation autour de x^k est le lagrangien lui même. La reformulation précédente s'applique donc directement au lagrangien :

$$L(x, y, \lambda^k, \mu^k) = \Phi_1(y, \lambda^k, \mu^k)\Psi_1(x) + \Psi_2(x, \lambda^k, \mu^k) + \Phi_2(y, \lambda^k, \mu^k). \tag{C.1}$$

Définition des *x-connectées*

Les variables x qui sont actives dans la fonction $\Psi_1(x)$ sont appelées les *x-connectées* car elles interviennent de façon bilinéaire dans l'expression du lagrangien, ou du lagrangien linéarisé dans le cas biconvexe. Ces variables sont très importantes du point d'un point de vue algorithmique. L'expression du lagrangien linéarisé donné par C.1 donne :

$$\begin{aligned} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} &= L(x^k, y, \lambda^k, \mu^k) \\ &\quad + \left(\Phi_1(y, \lambda^k, \mu^k) \cdot \nabla_x \Psi_1(x)\big|_{x^k} + \nabla_x \Psi_2(x, \lambda^k, \mu^k)\big|_{x^k}\right) \cdot (x - x^k). \end{aligned}$$

En considérant les conditions KKT du problème auxiliaire $(\mathcal{P}_{y^k})$ on a :

$$\begin{aligned} &\nabla_x L(x^k, y^k, \lambda^k, \mu^k) = \Phi_1(y^k, \lambda^k, \mu^k) \cdot \nabla_x \Psi_1(x)\big|_{x^k} + \nabla_x \Psi_2(x, \lambda^k, \mu^k)\big|_{x^k} = 0 \\ \Longrightarrow &\nabla_x \Psi_2(x, \lambda^k, \mu^k)\big|_{x^k} = -\Phi_1(y^k, \lambda^k, \mu^k) \cdot \nabla_x \Psi_1(x)\big|_{x^k}. \end{aligned}$$

En substituant la valeur de $\nabla_x \Psi_2(x, \lambda^k, \mu^k)\big|_{x^k}$ dans la première équation on obtient :

$$\begin{aligned} L(x, y, \lambda^k, \mu^k)\big|_{x^k}^{lin} &= L(x^k, y, \lambda^k, \mu^k) \\ &\quad + \left(\left(\Phi_1(y, \lambda^k, \mu^k) - \Phi_1(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_x \Psi_1(x)\big|_{x^k}\right) \cdot (x - x^k) \end{aligned}$$

A l'aide de cette dernière équation et de la propriété 3.36, le problème de minimisation interne ($\mathcal{IRD}$) peut être reformulé ainsi :

$$
\begin{aligned}
&L^*(\tilde{x}, y, \lambda^k, \mu^k) = \min_x L(x, y, \lambda^k, \mu^k) \\
&\geq \min_{x \in X} \left\{ L(x^k, y, \lambda^k, \mu^k) + \left(\Phi_1(y, \lambda^k, \mu^k) - \Phi_1(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_x \, \Psi_1(x)|_{x^k} \cdot (x - x^k) \right\} \\
&\geq L(x^k, y, \lambda^k, \mu^k) + \min_{x \in X} \left\{ \left(\Phi_1(y, \lambda^k, \mu^k) - \Phi_1(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_x \, \Psi_1(x)|_{x^k} \cdot (x - x^k) \right\}.
\end{aligned}
$$

Le dernier terme est séparable en x et on peut donc le reformuler comme une somme de termes, ce qui nous conduit à :

$$
\begin{aligned}
&\min_{x \in X} \left\{ \left(\Phi_1(y, \lambda^k, \mu^k) - \Phi_1(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_x \, \Psi_1(x)|_{x^k} \cdot (x - x^k) \right\} \\
&= \min_{x \in X} \left\{ \sum_{i=1}^{n} \left\{ \left(\Phi_{1_i}(y, \lambda^k, \mu^k) - \Phi_{1_i}(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_{x_i} \, \Psi_1(x)|_{x^k} \cdot (x_i - x_i^k) \right\} \right\} \\
&= \sum_{i=1}^{n} \left\{ \min_{x \in X} \left\{ \left(\Phi_{1_i}(y, \lambda^k, \mu^k) - \Phi_{1_i}(y^k, \lambda^k, \mu^k)\right) \cdot \nabla_{x_i} \, \Psi_1(x)|_{x^k} \cdot (x_i - x_i^k) \right\} \right\}.
\end{aligned}
$$

À partir de cette séparation si il existe certaines variables x_i qui n'apparaissent pas dans la fonction $\Psi_1(x)$, leur gradient $\nabla_{x_i}\Psi_1(x)$ sera nul et donc le terme en x_i disparaîtra de la sommation. Dans un tel cas, la minimisation par rapport à ce x_i n'influence donc nullement le résultat. On peut donc affirmer que la résolution du problème de minimisation interne avec le lagrangien remplacé par sa linéarisation doit se faire seulement sur les variables qui apparaissent dans la fonction $\Psi_1(x)$, c'est-à-dire les *x-connectées.*