



DEGREE PROJECT, IN OPTIMIZATION AND SYSTEMS THEORY , SECOND  
LEVEL

*STOCKHOLM, SWEDEN 2015*

# Integration of Production Scheduling and Energy Management

SOFTWARE DEVELOPMENT

ABDERRAHMAN AIT ALI

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCI SCHOOL OF ENGINEERING SCIENCES





**KTH Engineering Sciences**

# Integration of Production Scheduling and Energy Management

Software Development

A B D E R R A H M A N   A I T - A L I

Master's Thesis in Optimization and Systems Theory (30 ECTS credits)

Master Programme in Computer Science and Engineering (300 credits)

Royal Institute of Technology year 2015

Supervisors at ABB Corporate Research Center, Tyskland,

were Hubert Hadera and Lennart Merkert

Supervisor at KTH was Anders Forsgren

Examiner was Anders Forsgren

TRITA-MAT-E 2015:01

ISRN-KTH/MAT/E--15/01--SE

Royal Institute of Technology  
*School of Engineering Sciences*

**KTH SCI**  
SE-100 44 Stockholm, Sweden

URL: [www.kth.se/sci](http://www.kth.se/sci)



**Abstract:**

Demand-Side Management concepts have the potential to positively impact the financial as well as the environmental aspects of energy-intensive industries. More specifically, they allow reducing the energy cost for the industrial plants by dealing with energy-availability fluctuations.

In this context, efficient frameworks for scheduling with energy awareness have been studied and showed potential to reduce the overall energy bill for energy-intensive industries, for instance stainless steel and paper plants. Those frameworks usually combine scheduling and energy optimization into one monolithic system. This work investigates the possibility of integrating the two systems by specific exchange of signals, while keeping the scheduling model separated from the energy-cost optimization model. Such integration means that the pre-existent schedulers and energy optimizers could be easily modified and reused without re-implementing the whole new system.

Two industrial problems with different scheduling approaches are studied. The first problem is about pulp and paper production which uses the Resource Task Network (RTN) scheduling approach. The second one is about stainless steel production which is based on a bi-level heuristic implementation of an improved energy-aware scheduler. This work presents the decomposition methods that are available in literature and their application to the two industrial problems. Besides an improvement in the RTN approach for handling storages, this thesis describes a prototype implementation of the energy-aware RTN scheduler for paper and pulp production. Furthermore, this work investigates the performance of the application of different decomposition methods on different problem instances.

The numerical case studies show that even though the decomposition decreases the solution quality compared to the monolithic system, it still gives good solutions within an acceptable duration with the advantage of having two separate pre-existent systems which are simply exchanging signals.

**Keywords:** Pulp and paper production, stainless steel production, scheduling, energy-awareness, energy optimization, RTN, MILP, demand-side management, decomposition methods, prototype implementation.



# Table of Contents

List of Figures .....	iv
List of Tables .....	v
1. Introduction .....	1
1.1. Background .....	1
1.2. Problem .....	1
1.3. Aim and Scope .....	2
1.4. Methodology .....	2
2. Problem Description.....	3
2.1. Pulp and Paper Production Process.....	3
2.2. Stainless Steel Production Process.....	4
2.3. Energy-cost Optimization .....	5
3. Literature Review .....	7
3.1. Resource Tasks Network (RTN) for Discrete Scheduling.....	7
3.2. General Monolithic Model.....	9
3.3. Primal Decomposition.....	10
3.4. Dual Decomposition.....	12
3.5. Cross Decomposition .....	15
3.5.1. Mean-Value Cross Decomposition (MVCD) .....	16
3.5.2. Weighted Mean-Value Cross Decomposition (WMVCD).....	16
3.5.3. One-sided Weighted Mean-Value Cross Decomposition (OWMVCD) .....	17
4. Mathematical Modeling .....	18
4.1. Basic Model for Pulp Production.....	18
4.1.1. Pre-existent Model.....	18
4.1.2. Simplified Model .....	19
4.2. Extended Model for Paper Production .....	21
4.2.1. RTN Superstructure and Excess Balance Equation.....	21
4.2.2. Processing Rates .....	21
4.2.3. Machines Start-up and Stop.....	22
4.2.4. Late Orders .....	23
4.2.5. Resource Storage .....	24
4.2.6. Paper Making Change-over .....	24
4.2.7. Energy Consumption .....	26
4.2.8. Objective Function .....	26

4.3. Pre-existent Model for Stainless Steel Production.....	26
5. Energy Integration to Production Planning.....	29
5.1.1. Simple Decomposition Scheme for Pulp and Paper Production .....	29
5.1.2. Complex Decomposition Scheme – Stainless Steel Model.....	30
6. Prototype Software Development .....	32
6.1. Architecture Overview .....	32
6.2. Input Structure.....	33
6.3. Objects Representation.....	33
6.4. Graphical User Interface (GUI) .....	34
7. Case Study.....	35
7.1. Setup and Input Data – Pulp and Paper Production .....	35
7.2. Results - Pulp and Paper Production .....	36
7.3. Setup and Input Data – Stainless Steel Production .....	38
7.4. Results – Stainless Steel Production .....	39
8. Concluding Remarks .....	42
8.1. Conclusions .....	42
8.2. Limitations .....	42
8.3. Future Work .....	43
Bibliography .....	44
Appendix 1 – RTN Superstructure of Paper Production .....	46
Appendix 2 – Bi-level Heuristic for Stainless Steel Problem.....	47
Appendix 3 – Decomposition Algorithms on Bi-level Heuristic.....	48
Appendix 4 – Prototype Graphical User Interface.....	49



## List of Figures

Figure 1 - Basic process of pulp production .....	3
Figure 2 - Extended process of paper and pulp production .....	4
Figure 3 - Stainless steel production process, from (Hadera, et al., 2014a) .....	5
Figure 4 - Energy sources and sinks, based on (Hadera, et al., 2014a) .....	6
Figure 5 – Consumption and production of resources by a processing task.....	7
Figure 6 – Continuous and discrete interactions in RTN superstructure .....	7
Figure 7- Primal block-angular structure of the monolithic problem .....	10
Figure 8 - Primal decomposition algorithm, based on (Hadera, 2015).....	12
Figure 9 - Dual block-angular structure of the monolithic problem.....	12
Figure 10 - Dual decomposition algorithm, based on (Hadera, 2015).....	14
Figure 11- Overview of the primal and dual decomposition methods.....	15
Figure 12 - Cross decomposition algorithm, based on (Hadera, 2015) .....	15
Figure 13 – Improved cross decomposition algorithms, based on (Hadera, 2015) .....	17
Figure 14 – RTN Superstructure of the basic pulp production model (Hadera, et al., 2015)..	18
Figure 15 – RTN Superstructure of the simplified basic model for pulp production .....	20
Figure 16 - Simple task-machine interaction with starting and stopping task.....	23
Figure 17 - Use of structural parameters for starting and stopping discrete tasks .....	23
Figure 18 - Change-over illustration in a production machine .....	24
Figure 19 - Capturing machine change-over using additional variables .....	25
Figure 20 - Example of RTN superstructure to capture change-overs .....	26
Figure 21 - Stainless production planning problem, based on (Hadera, et al., 2014a) .....	27
Figure 22 - Structure overview of the demonstrator software .....	32
Figure 23 - UML class diagram showing the overall objects representation .....	34
Figure 24 - Spot market hourly prices for energy cost (NordPoolSpot, 2014).....	36
Figure 25 - Hourly day-ahead electricity prices for 24 hours (EPEX, 2013) .....	38
Figure 26 - Iteration history of the first decomposition approach on instance 2 .....	41
Figure 27 - Iteration history of the second decomposition approach on instance 2 .....	41

## List of Tables

Table 1 - Basic real scenario (S0) for pulp and paper production .....	35
Table 2 - Case study scenarios for pulp and paper production .....	35
Table 3 - Numerical results of the monolithic model for different instances .....	37
Table 4 - Numerical results of the cross decomposition iterative method.....	37
Table 5 - Investigated problem instances for cross decomposition .....	38
Table 6 - Solution algorithms implemented for stainless steel production case study .....	39
Table 7 - Numerical results of the bi-level heuristic algorithm on the different instances.....	39
Table 8 - Numerical results of the decomposition for all levels of the bi-level heuristic with two different signal strategies .....	40
Table 9 - Numerical results of the decomposition for the upper-level of the bi-level heuristic with two different signal strategies .....	40

# **1. Introduction**

This document is the final report for the Master Degree in Computer Science and Engineering at The Royal Institute of Technology in Stockholm (KTH) as well as for the Engineer Diploma in Applied Mathematics and Computer Science at The Grenoble Institute of Technology in Grenoble (INPG).

This introductory chapter describes the professional and scientific context surrounding the degree project. The second chapter gives a description in more details of the industrial problems which are studied in this project. The literature review of the scientific researches and the theoretical tools related to the project are exposed and summarized in the third chapter. In the fourth chapter, the mathematical models are presented and explained whereas the prototype development is discussed in the following chapter. Finally, the case study is conducted to check the mathematical models in the sixth chapter before drawing several concluding remarks in the last chapter.

## **1.1. Background**

The project is performed at ABB Corporate Research Center in Ladenburg, Germany. ABB is among the leading companies in power and automation technologies. It strives to increase the industrial productivity through its efficient products and breakthroughs in the domain of electrical power and automation combined with the care of lowering the environmental impact. The company is operating in power products and systems, discrete automation and motion, low voltage products and process automation. In addition to the research center of Ladenburg in Germany, ABB has six other research centers all over the world. Those research centers and their research teams are of paramount importance for the company to stay competitive as well as innovative against its main competitors (ABB, 2014).

The project was carried out between July 2014 and February 2015 in an ongoing research project within the research group of Process and Production Optimization, one of the eight research groups in ABB research center in Ladenburg.

## **1.2. Problem**

In the last few years, world's electric energy consumption (measured in watt-hours) has increased, especially in industry. Meanwhile, electricity volatility has increased due to the deployment of renewable energy generation, environmental concerns and shrinking of the conventional energy sources (Merkert, et al., 2014). Energy markets became liberalized and energy prices became subject to high fluctuations for the final consumers. In this case, production planning without awareness of these considerations may lead to expensive electricity bills at the user side. For this reason and within the concept of Demand-Side Management, integrating energy awareness into production planning for energy-intensive industries may have a noticeable potential in the future (Hadera, et al., 2014a).

Energy-cost optimizers and production planning solutions are available for energy-intensive industry and there are very efficient implementations of those systems that are the result of years of engineering and research. Thus, the idea is to integrate energy management and

production planning in a way that uses such pre-existent optimizers. Hence, the problem is to find an efficient iterative scheme between the energy optimizer and the production planner in order to get an integrated optimizer that could be used for real problem instances.

### **1.3. Aim and Scope**

Decomposition methods are widely used to decompose optimization problems with special structure into sub-problems, either to increase model performance (lower computation times or better solution quality) or to represent desired problem structures. There are several methods that are proven to perform well for specific optimization problems. The aim of this study will be:

- To model and expand pulp and paper production planning problem using the Resource Task Network (RTN) technique.
- To explore different decomposition methods in literature: primal, dual and cross decomposition methods.
- To apply cross decomposition methods to two different production planning problems (pulp and paper production and stainless steel production) using different iterative schemes for energy awareness.
- To investigate the behavior of the decomposition algorithms on real world instances.
- To draw useful conclusions for further work.
- To implement a prototype software demonstrator for pulp and paper production.

The study here will not aim to prove the theoretical properties of the decomposition methods. On the contrary, it will be restricted to a more algorithmic and numerical study of the decomposition methods and the iterative schemes on the specific industrial problems.

### **1.4. Methodology**

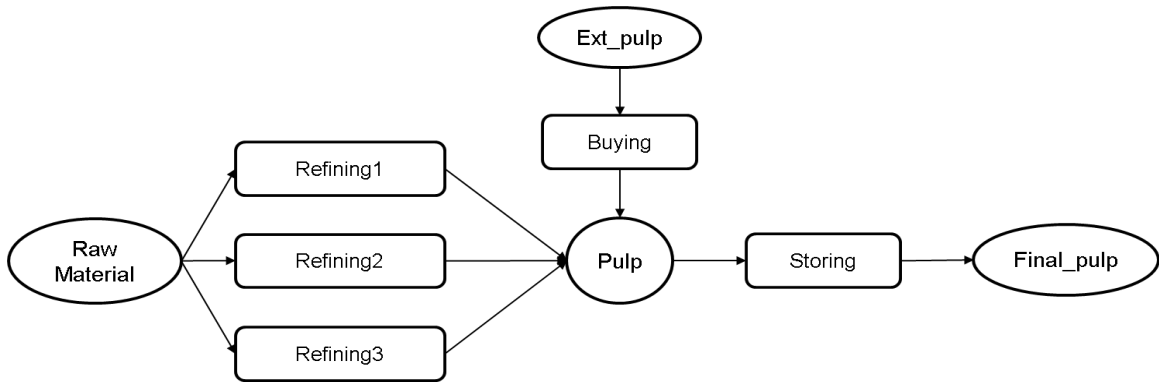
The two industrial problems which are studied here are based on mathematical programming and more precisely on large-scale Mixed Integer Linear Programming (MILP). One of the industrial problems (i.e. pulp and paper production) will be modeled and extended using the RTN techniques which lead to MILP problems. The decomposition of the energy-aware production planning will use Primal-Dual decomposition methods called cross decomposition methods. In order to study the cross decomposition methods applied to the industrial problems considered here, the starting point is the monolithic model of each production problem which is simply the full problem in one mathematical optimization program. Setting the monolithic model will help to decompose the full problem into sub-problems and therefore leads to the problem decomposition. The literature review will help to explore the way to construct efficient iterative schemes for the two production planning problems. To assess the implemented models, numerical case studies will be performed and will be discussed and analyzed in order to conclude with regard to the performance of the investigated cross decomposition methods.

## 2. Problem Description

Two industrial problems are explored in this project. One is about pulp and paper production and the other is about stainless-steel production. The main concern is to find an optimal (or close to optimal) production schedule by considering an effective scheduling of the production processes and an optimal energy purchase strategy. The production plan has to take into account several production and energy-related constraints. Whereas the energy purchase has to consider different energy contracts as discussed later. In this chapter, the two production processes will be presented as well as the energy management component which is common between the two problems.

### 2.1. Pulp and Paper Production Process

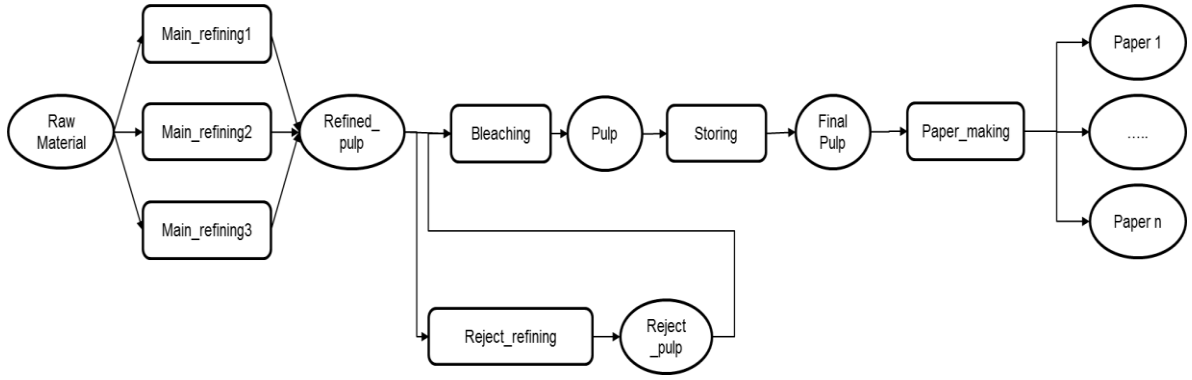
The production process of paper starts with the preparation of raw material (wood, recycled paper, agricultural residues, etc.). The pulp is produced after several stages of mechanical or chemical pulping, pulp washing and screening. In thermo-mechanical pulping (TMP) which is considered here, the pulping technique uses the application of heat alongside the mechanical refiners. Once the pulp is produced, it is fed to the storage tanks for additional chemical processing before being used in the paper machines. In order to add flexibility to the model, pulp can be bought from external sources. The basic process of pulp production is given in Figure 1.



*Figure 1 - Basic process of pulp production*

In the extended model, the basic stages have more details of the production. Certain amount of the main processed pulp is refined once again in the reject refiners and then used by the paper machines after the bleaching stage in order to produce rolls of paper. Finally, those paper rolls are cut into smaller rolls which form the final paper product (Bajpai, 2012). This cutting stage is not considered in the study, for interested readers this topic was treated by Harjunoski (Harjunoski, et al., 1996).

Figure 2 shows an overview of the paper production process which is considered in this study.



*Figure 2 - Extended process of paper and pulp production*

The extended production process has several stages, each of which contains parallel and identical machines. The process has many functional constraints which have to be satisfied. For instance, both refiners and paper machines have certain production rates which cannot be exceeded. Besides, there are other production details which are not represented in Figure 2. For example, steam is produced in the main refining stage and consumed in the paper making stage. Other properties and requirements will be discussed in more details in the next chapters. To add more flexibility to the production process, steam can be bought from some external sources with given constant prices and with maximal buying capacity. Besides, all the necessary resources, such as woodchips, electricity, external pulp and steam are considered unlimited.

It is important to mention that the extended process for paper production could be used as well for pulp production by considering the first stages which correspond to pulp production process. Practically, this could be done by fixing the allocations tasks and quantities of paper according to the demand of pulp and the production rates of the paper machines.

## **2.2. Stainless Steel Production Process**

To produce stainless steel, the production starts with a scrap melting phase in an Electric Arc Furnace (EAF) which is a batch process that forms the heats. After a full heat is formed, it is transported to the second stage where the carbon in the molten steel is reduced. This is done in the phase of Argon Oxygen Decarburization (AOD). To adjust the final chemistry and temperature, the heat goes through the Ladle Furnace (LF) stage. The final stage is the Continuous-Casting (CC) stage where the heat is casted into slabs of stainless steel (Hadera, et al., 2014a).

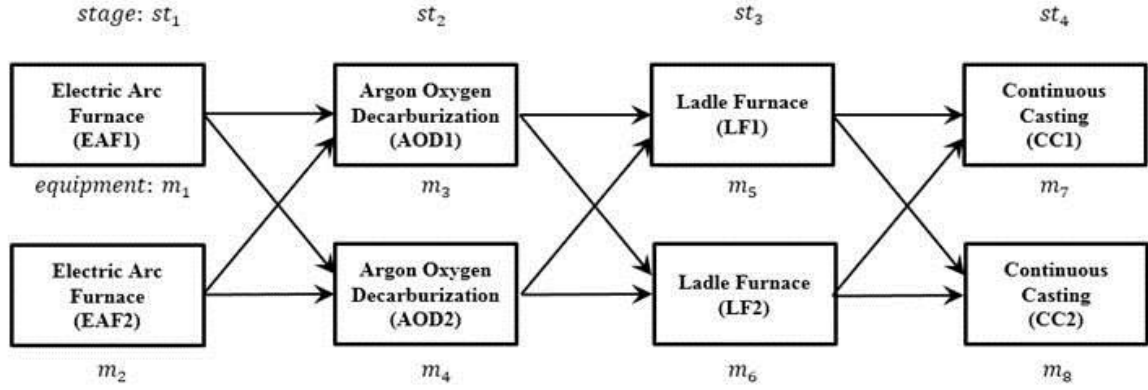


Figure 3 - Stainless steel production process, from (Hadera, et al., 2014a)

The production process has four different stages and two parallel, non-identical machines at each stage. In addition, the industrial process has several functional constraints. For instance, all machines, except those in the CC stage, have machine-specific setup times. Moreover, the heat is subject to minimum transportation time and maximum waiting or hold-up time between two production stages. Heats of the same heat group have to be casted subsequently without hold-up time. More details are described in (Hadera, et al., 2014a).

### 2.3. Energy-cost Optimization

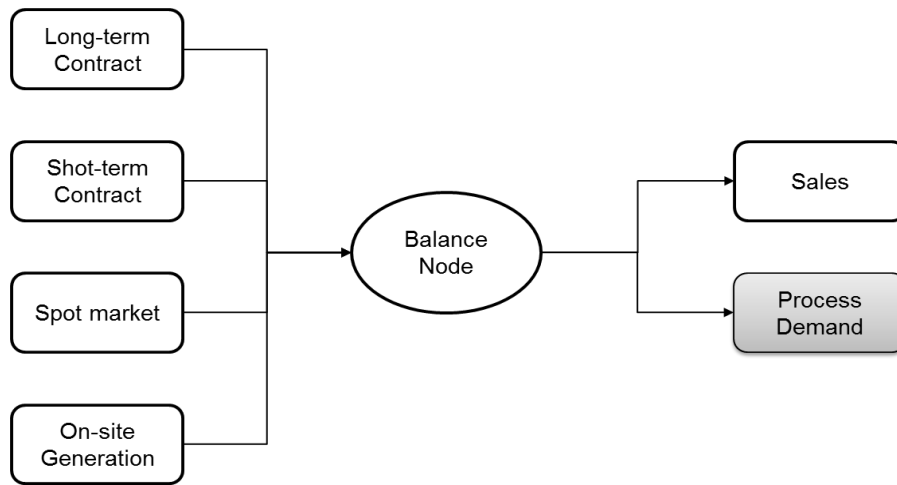
In addition to the production constraints, there are numerous electricity considerations in the energy management side of the problem. Both processes (pulp and paper production or stainless steel production) have high energy consumption. For instance, 45 % of the total energy use in Sweden alone in 2011 was accounted for pulp and paper industry and 18 % for steel industry (Swedish Energy Agency, 2013b). Most of this energy is consumed in the EAF stage in the stainless steel production and by mechanical refiners used in TMP and by electric motors used in paper machines in pulp and paper production.

The energy demand of each stage should always be satisfied. Thus, for more effective energy management, several electricity sources are considered in order to satisfy the energy demand with lower costs. The purchasing contracts are as follows:

- **Long-term or pre-agreed contract:** The price is constant through the time horizon and generally low.
- **Short term contract:** Two price levels (peak and off-peak), one price level is low and the second one is generally higher.
- **Spot market:** Highly fluctuating prices with hourly variations.
- **On-site generation:** The generation cost is constant.

For all the previous contracts, deviation penalties are considered if the energy purchase quantity is different by more than a given tolerance buffer from a given pre-agreed energy consumption. Besides, the purchased electricity is used to satisfy the production process demand and if there is an excess of energy, it has to be sold back to the power grid.

The electricity management process could be summarized in the following figure:



*Figure 4 - Energy sources and sinks, based on (Hadera, et al., 2014a)*

In Figure 4, the different energy contracts can be mathematically seen as sources of a certain flow (of energy) in a flow network problem, whereas the energy demand and the sales are considered as sinks. The balance node ensures the flow conservation between sources and sinks. Besides, each arc has a cost which helps formulate the energy cost objective function to minimize. Hence, the problem of energy management can be considered as a minimum-cost flow network optimization problem (Hadera, et al., 2014a).



### 3. Literature Review

#### 3.1. Resource Tasks Network (RTN) for Discrete Scheduling

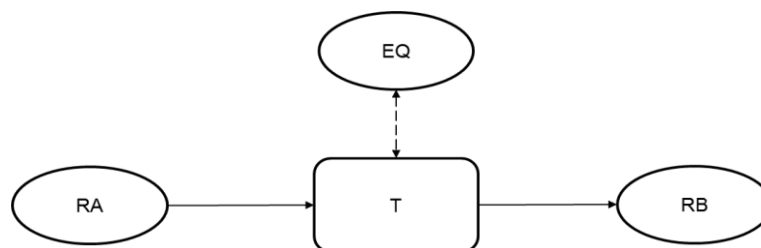
The Resource-Task Network (RTN) approach is a technique which has been recently widely used to model different scheduling problems. The technique was first presented in (Pantelides, 1994) as an extension of the State-Task Network (STN). The logic behind the RTN approach is the use of a general mathematical scheduling formulation to model the consumption and production of resources by processing tasks. The resources quantities propagate through the scheduling time horizon after either consumption or production by some processing tasks. Thus, it is a unified framework for optimal planning and scheduling which regards all processes as bipartite graphs comprising two types of nodes: Resources and Tasks (Pantelides, 1994).

A structural diagram could be drawn to show all the different interactions between resources and tasks in the system, such a diagram is called RTN superstructure of the system. The resources are generally represented with circles and the tasks with rectangles. The following figure shows an example of the superstructure of a simple system in which a task is consuming a resource and producing another resource.



*Figure 5 – Consumption and production of resources by a processing task*

In general RTN models, there are two types of interactions between the resources and the tasks: continuous and discrete interactions. The continuous interactions are represented with continuous arrows and the discrete ones using dashed arrows. The main difference between the two interactions is that the first ones correspond to continuous consumptions or productions of a certain resource, whereas the second is for processes in particular points in the time horizon. These discrete interactions are typically used for assignment of equipment to a processing task as shown in the figure bellow:



*Figure 6 – Continuous and discrete interactions in RTN superstructure*

The RTN scheduling approach can be formulated as a continuous or discrete problem. The latter problem was chosen to be used in the pulp and paper production problem. The main reason is that for large scales industrial problems, the RTN discrete scheduling showed generally better results compared to the continuous one (Castro, et al., 2009).

Once the time horizon was discretized uniformly, the strength of the RTN technique is in the compact Mixed Integer Linear Program (MILP) formulation of the scheduling problem. The most important equation in the mathematical formulation is the excess balance equation. Pantelides presented, in his original paper (Pantelides, 1994), a general form of the RTN excess balance equation as below:

$$\forall r \in Resources \quad \forall t \in \{1 \dots T\}$$

$$\begin{aligned} Re(r, t) = & Re(r, t - 1) + \Pi(r, t) \\ & + \sum_{i \in Tasks} \sum_{\theta \in \tau_i} \lambda(r, i, \theta) \xi(i, t - \theta) + \mu(r, i, \theta) N(i, t - \theta) \end{aligned} \quad (1)$$

Equation (1) describes the variations of the amount  $Re(r, t)$  of each resource through the scheduling time horizon  $t \in \{1 \dots T\}$ . Hence, the amount of resource  $r$  at time slot  $t$  is the quantity of the same resource at time slot  $t - 1$  plus the variations (i.e. consumptions or productions) of the resource during the time slot  $t$ . The variations could be from the interactions with external systems, for instance: energy demand, product orders or material release. This is expressed in the equation with the term  $\Pi(r, t)$ . The variations of the resource quantities could also be related to the processing tasks within the system, this is expressed as a summation of all the variations by each processing task. In this summation the parameters  $\lambda(r, i, \theta)$  and  $\mu(r, i, \theta)$  describe the nature of the interactions between the resource  $r$  and the task  $i$  (discrete or continuous, consumption or production). The integer index  $\theta \in \tau_i$  allows considering the variations over the length of the processing task  $i$ . The variables  $\xi(i, t)$  and  $N(i, t)$  represent, respectively, the amount that task  $i$  processed and the state of the processing task during time slot  $t$ .

The excess balance equation describes how the resources are processed (i.e. consumed or produced) as well as how they propagate through the scheduling time horizon. Yet, this equation is not enough to fully describe the process. To complete the model, initial quantities of some resources are required. To do this, a parameter  $R^0(r)$  for each resource is typically used to set its initial quantity:

$$\forall r \in Resources; \quad R(r, t = 0) = R^0(r) \quad (2)$$

In order to set the processing capacities (minimal and maximal capacity) for certain production stages, the following constraint is generally used:

$$\forall i \in Tasks \quad \forall t \in \{1 \dots T\}; \quad C_{min} N(i, t) \leq \xi(i, t) \leq C_{max} N(i, t) \quad (3)$$

Those equations and constraints may be used in a slightly different form to adapt to certain specific industrial requirements of the scheduling process (Castro, et al., 2009). The objective function (make-span, production cost, etc.) can be mathematically expressed as a linear

combination of the RTN decision variables (i.e.  $\xi(i, t)$  and  $N(i, t)$ ). Thus, the technique yields a compact MILP formulation for large scales production planning and scheduling problems (Pantelides, 1994).

### 3.2. General Monolithic Model

The integration of the scheduling model and the energy management model leads to a monolithic model with two nested optimization problems: production planning problem (PP) and energy flow network problem (EFN). Production planning models with integrated energy management are generally modelled as the minimization of the following objective cost function

$$F = C_{EFN}^T f + C_{PP}^T y \quad (4)$$

With respect to the functional constraints

$$A_{EFN} f + D_{EFN} x - b_{EFN} \leq 0 \quad (5)$$

$$A_{PP} y + D_{PP} q - b_{PP} \leq 0 \quad (6)$$

$$A f - I q = 0 \quad (7)$$

$$f \geq 0; x \in X; y \in Y; q \geq 0 \quad (8)$$

The objective cost function (4) consists of the cost of the production planning – PP (production cost) and the cost of the flow network - EFN (energy purchase).

The inequality constraint (5) represents the functional constraints of the flow network or the energy management whereas (6) describes those of the production planning.

The equality constraint (7) links the production demand of electricity to the flow from the balance node to the process demand node as represented in Figure 4.

The inequalities and inclusions in (8) represent the bounds as well as the type (continuous or discrete) of variables.

Solving this problem means finding the minimal total cost  $F^*$  with the corresponding optimal schedule and energy purchase strategy. It is important to note the linearity of all the constraints and the objective function, thus the problem is linear. Moreover, the model has both continuous and discrete variables. Therefore, the general monolithic model is a Mixed Integer Linear Program (MILP).

MILP problems are a variant of Integer Linear Program (ILP) problems which are NP-hard problems. This means that the industrial models which are treated here do not have any polynomial solving algorithm (unless the famous equation NP=P is proven).

Adding the flow network problem into the scheduling problem makes the monolithic model more complex and even longer to solve (or at least to find a good solution) for real instances. Besides, it is easy to recognize the two different optimization problems that are linked with a simple constraint, i.e. consensus constraint (7). Moreover, there exist very efficient industrial

implementations to solve each of the two problems. Thus, it is desirable here to study the decomposition of this monolithic model into two well-known optimization models. In the next sections, several separation methods will be applied to find out the best scheme from those methods that will serve the industrial purpose of this separation.

### 3.3. Primal Decomposition

Primal decomposition is a method firstly introduced by Benders (Benders, 1962). The method was developed to solve large-scale linear programming problems but it is also used with mixed integer programming problems. Primal decomposition, also called Benders' decomposition, uses the special structure of the linear programs in order to solve them efficiently. This special structure, called primal block-angular structure, is typically when the constraint matrix is in the following block-angular form:

$$\min \{ C_{EFN}^T f + C_{PP}^T y \}$$

*	*
*	0
0	*

↔

$$\begin{pmatrix} \begin{bmatrix} +A & 0 \\ -A & 0 \end{bmatrix} & \begin{bmatrix} -I & 0 \\ +I & 0 \end{bmatrix} \\ \begin{bmatrix} A_{EFN} & D_{EFN} \end{bmatrix} & \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & \begin{bmatrix} D_{PP} & A_{PP} \end{bmatrix} \end{pmatrix} \begin{pmatrix} f \\ x \\ q \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ b_{EFN} \\ b_{PP} \end{pmatrix}$$

Figure 7- Primal block-angular structure of the monolithic problem

This is the case basically when some constraints contain variables from different sub-problems. This is the case with the equality constraint (7) in the general monolithic formulation. The method divides a complex problem into two problems. One is called the primal sub-problem or Benders' sub-problem which gives an upper bound (for minimization problems). The other one is the primal master problem which gives a lower bound. The method iterates between the two problems and reduces the gap between the bounds to find an optimal solution to the complex original monolithic problem.

In the general monolithic problem presented in the previous section, the complicating variable is the energy consumption  $q$  which links the two sub-problems by the equality constraint (7). If  $q$  is fixed, the original problem can be easily separated. After fixing the complicating variable ( $q = \bar{q} \geq 0$ ), we get the primal sub-problem as minimizing

$$C_{EFN}^T f + C_{PP}^T y \tag{9}$$

With respect to the constraints

$$A_{EFN} f + D_{EFN} x - b_{EFN} \leq 0 \tag{10}$$

$$A_{PP} y + D_{PP} \bar{q} - b_{PP} \leq 0 \tag{11}$$

$$Af - I\bar{q} = 0 \quad (12)$$

$$f \geq 0; \quad x \in X; y \in Y \quad (13)$$

This sub-problem is clearly separable into two different optimization problems. The first problem is about finding an energy purchase strategy for the fixed energy consumption  $\bar{q}$  which has a minimal cost  $F_{EFN}(\bar{q})$ , this problem is called here as Primal Energy-cost Flow Network or Primal EFN. The second one is to find a production schedule for the fixed energy consumption ( $\bar{q} \geq 0$ ) which gives a minimal cost  $F_{PP}(\bar{q})$ ; this problem is called as Primal Production Planning or Primal PP.

Solving the primal sub-problem (i.e. summing the two solutions from the two separate optimization problems) gives an upper bound for the optimal solution  $F^*$  of the original general monolithic problem.

In order to get a lower bound, the primal master problem has to be solved. The primal master problem for the general monolithic problem is constructed by relaxing the constraint (6) from the original problem using the Lagrange multiplier  $\mu$ . Thus, the primal master problem at iteration  $k$  is considered for fixed values of the multiplier  $\mu^{(k)}$  as the minimization of

$$w \quad (14)$$

With respect to the constraints

$$\forall k \geq 1; \quad w \geq F_{EFN}(q) + (b_{PP} - D_{PP}q)^T \mu^{(k)} \quad (15)$$

$$q \geq 0 \quad (16)$$

This gives a lower bound for the optimal solution of the original monolithic problem.

$$w^*(\mu^{(k)}) \leq F^* \quad (17)$$

The Primal decomposition algorithm iterates between the sub-problem and the master problem. At iteration  $k$ , a solution from the sub-problem gives the values of the fixed multiplier  $\mu^{(k)}$  in the master problem as the marginal of the constraint (11) and a new cut is added in (15) to the previous ones. A solution from the master problem sets the value of the coupling variable  $\bar{q}$  to the sub-problem. The algorithm stops when the lower and upper bound converge to the optimal solution up to a tolerance.

Thus, the primal decomposition algorithm performs as follows:

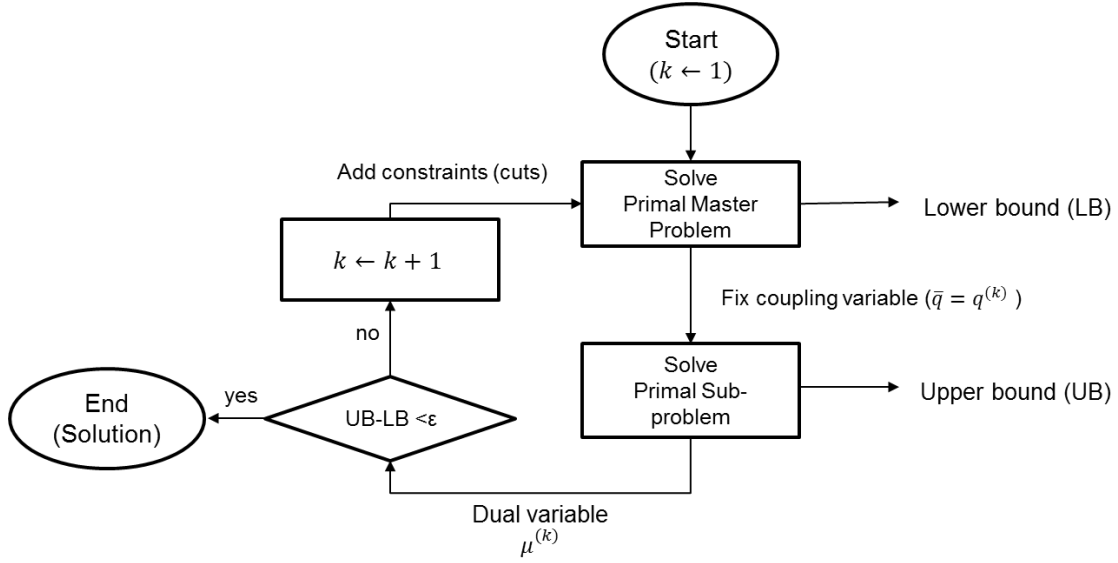


Figure 8 - Primal decomposition algorithm, based on (Hadera, 2015)

The primal decomposition algorithm has the advantage of decomposing the monolithic problem while solving the primal sub-problem. However, the main disadvantage is the obligation to solve the master problem which can be even more complex than the original monolithic problem.

### 3.4. Dual Decomposition

Dual decomposition, also known as Dantzig-Wolfe decomposition, is another decomposition method which was developed in 1960 by Wolfe and Dantzig (Wolfe & Dantzig, 1960). As with the primal decomposition, the dual decomposition uses the dual block-angular structure of the constraint matrix to solve large scale optimization problems. This is when the constraint matrix is in the following form

$$\min \{ C_{EFN}^T f + C_{PP}^T y \}$$

$$\begin{bmatrix} * & * & 0 \\ * & 0 & * \end{bmatrix} \Leftrightarrow \begin{pmatrix} [A_{EFN}] & [D_{EFN}] & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ +A \\ -A \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ D_{PP} & A_{PP} \\ -I & 0 \\ +I & 0 \end{bmatrix} \end{pmatrix} \begin{pmatrix} f \\ x \\ q \\ y \end{pmatrix} \leq \begin{pmatrix} b_{EFN} \\ b_{PP} \\ 0 \\ 0 \end{pmatrix}$$

Figure 9 - Dual block-angular structure of the monolithic problem

This structure is when some variables are present in the different sub-problems. This is the case with the complicating variable  $q$  which is present in EFN and PP sub-problems.

In contrast with the primal decomposition, this method makes use of Lagrangean relaxation to eliminate the complicating constraints. The Lagrangean multipliers in this case are to be interpreted as shadow prices for the objective cost function.

As with the primal decomposition, the dual decomposition iterates between one problem which is called the dual sub-problem and another problem called the dual master problem.

To eliminate the complicating constraint, the equality constraint (7) in the general monolithic problem is relaxed with the corresponding free Lagrangean multiplier  $\lambda$ . The dual sub-problem is constructed by setting the multiplier to a fixed value  $\bar{\lambda}$ . Hence, the dual sub-problem is to minimize

$$(C_{EFN}^T - \bar{\lambda}^T A)f + C_{PP}^T y + \bar{\lambda}^T q \quad (18)$$

With respect to the constraints

$$A_{EFN}f + D_{EFN}x - b_{EFN} \leq 0 \quad (19)$$

$$A_{PP}y + D_{PP}q - b_{PP} \leq 0 \quad (20)$$

$$f \geq 0; x \in X; y \in Y; q \geq 0 \quad (21)$$

Once the complicating constraint is relaxed and the multiplier is fixed, the problem becomes clearly separable. Therefore, the dual sub-problem can be solved by minimizing the first term in (18) with respect to the constraint (19) independently from minimizing the last two terms in (18) with respect to (20). The first optimization problem is called Dual EFN and the second is called Dual PP. The dual sub-problem gives a lower bound for the optimal solution of the original monolithic problem.

$$F_{PP}(\bar{\lambda}) + F_{EFN}(\bar{\lambda}) \leq F^* \quad (22)$$

Similarly to the primal decomposition, the dual master problem has to be solved in order to get an upper bound. The master problem is formed by considering the convex combination of the solutions (extreme points) from the dual sub-problem. This combination is applied to the objective function and the complicating constraint in the original problem to form the dual master problem. Thus, at iteration  $k$  and for fixed values of  $f^{(k)}$ ,  $y^{(k)}$  and  $q^{(k)}$ , the dual master problem is to minimize

$$F(f^{(k)}, y^{(k)}, q^{(k)}) = \sum_{k \geq 1} (c_{EFN}^T f^{(k)} + c_{PP}^T y^{(k)}) \lambda^{(k)} \quad (23)$$

With respect to the constraints

$$\sum_{k \geq 1} (A f^{(k)} - I q^{(k)}) \lambda^{(k)} = 0 \quad (24)$$

$$\sum_{k \geq 1} \lambda^{(k)} = 1 \quad (25)$$

$$\forall k \geq 1; \quad \lambda^{(k)} \geq 0 \quad (26)$$

The dual master problem gives an upper bound for the optimal solution of the original monolithic problem.

$$F^* \leq F^*(f^{(k)}, y^{(k)}, q^{(k)}) \quad (27)$$

As for the primal decomposition, the dual decomposition algorithm iterates between the dual sub-problem and the dual master problem. The dual sub-problem sets the values of  $f^{(k)}$ ,  $y^{(k)}$  and  $q^{(k)}$  for the master problem, whereas the master problem sets the value of  $\bar{\lambda}$  for the sub-problem from the marginal value of the constraint (24). The algorithm stops, as for the primal algorithm, when the lower and upper bound converge to the optimal solution up to a tolerance. Thus, the dual decomposition algorithm performs as follows:

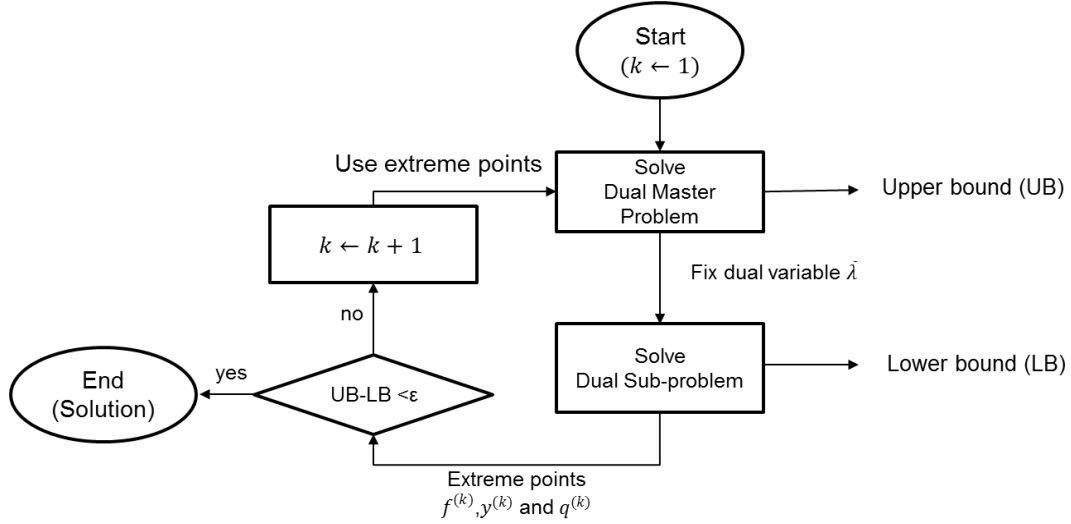


Figure 10 - Dual decomposition algorithm, based on (Hadera, 2015)

As with the primal decomposition algorithm, the dual algorithm has the advantage of decomposing the monolithic problem. Yet, the main disadvantage is once again the need to solve the dual master problem which can be more difficult to solve than the original monolithic problem. One may notice that the PP dual sub-problem in this decomposition is a scheduling problem with a modified objective function.

From the two decomposition methods previously described, one can draw the general overview of the methods as in the following figure:



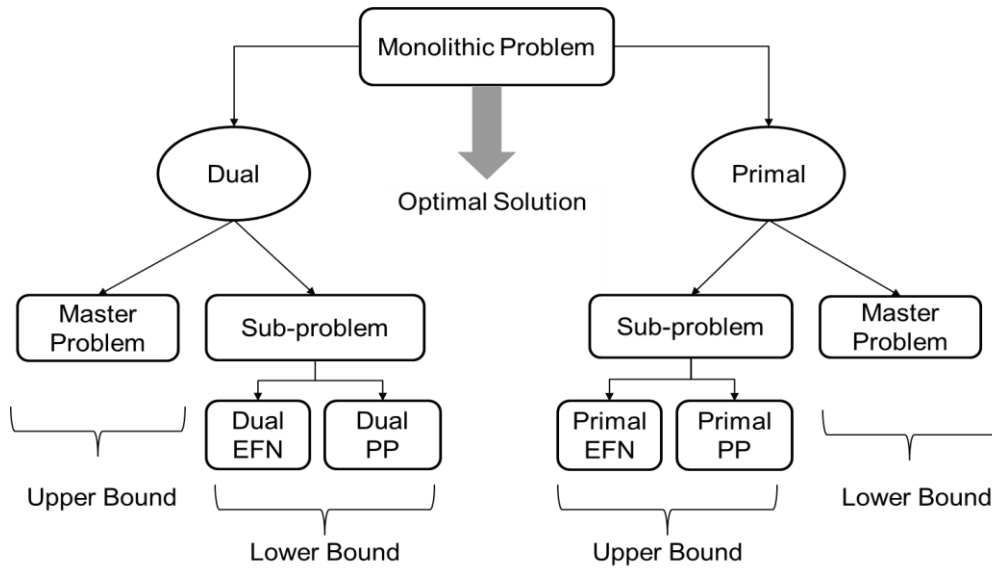


Figure 11- Overview of the primal and dual decomposition methods

### 3.5. Cross Decomposition

Van Roy in (Roy , 1983) published an interesting method called cross decomposition method that was originally applied to solve MILP-type problems. The method iterates between the primal sub-problem from the primal or Benders' decomposition and the dual sub-problem from the dual or Dantzig-Wolfe's decomposition. In each iteration, a convergence test is performed and if the test fails, the primal or dual master problem has to be solved to get the solution iterate for the next iteration. As discussed previously, the primal and the dual sub-problems give, respectively, an upper and a lower bound for the solution of the original minimization problem. Thus, the method converges when the bound gap is small enough. The following figure shows a generic scheme of the cross decomposition method:

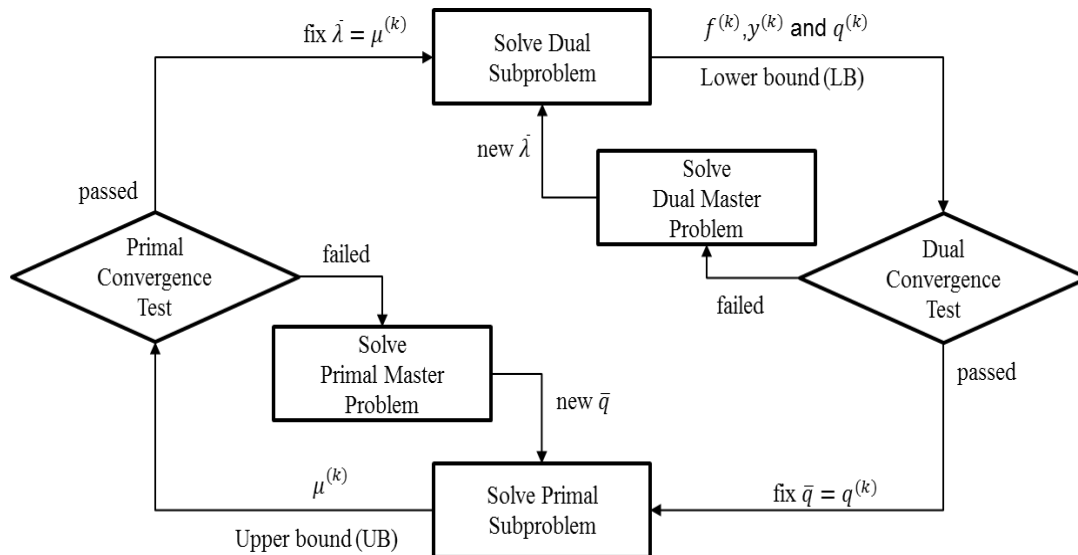


Figure 12 - Cross decomposition algorithm, based on (Hadera, 2015)

It is important to note that for the sub-problems, one part of the solution is used as an input signal in the other sub-problem. This means that the complete sub-problem (EFN and PP) does not need to be solved. Thus, instead of solving the complete dual sub-problem, only dual PP is solved to get the value for  $\bar{q}$ . Similarly, instead of solving the complete primal sub-problem, only the Primal EFN is solved to get the value for  $\bar{\lambda}$ .

As mentioned previously, dual PP sub-problem is a scheduling problem with a modified objective function. Similarly, primal EFN is an energy flow network problem with a fixed flow between two nodes. Hence, to solve those problems, it is possible to use the pre-existent optimizers with slight modifications.

Cross decomposition method has the advantage of decomposing the original monolithic problem by solving the primal and dual sub-problems. It uses the master problems only to steer the algorithm toward convergence if a convergence test fails. However, with this scheme it is still necessary to implement the model of the master problems which as shown previously does not correspond to any of the pre-existent models.

### 3.5.1. Mean-Value Cross Decomposition (MVCD)

A new development from the cross decomposition method was the so called Mean-Value Cross Decomposition method. The method was first published by Holmberg in 1992 and was initially used to solve large Linear Program (LP) problems (Holmberg, 1992). It was proven to converge for LP problems in 1994 (Holmberg, 1994). In contrast with the cross decomposition method, MVCD iterates between the primal and dual sub-problem without having to solve any master problem when the convergence test fails. Instead of using the current solution as an input signal to each sub-problem, the method uses the mean value of all the solutions to construct the input signal. In MVCD, the input signal at iteration  $k$  is as follows:

$$\mu^k = \frac{1}{k} \tilde{\mu}^{k-1} + \frac{k-1}{k} \mu^{k-1} \quad (28)$$

Where the signal input in the previous iteration is  $\mu^{k-1}$  and  $\tilde{\mu}^{k-1}$  is the previous solution from the dual or primal sub-problem and  $\mu^k$  is the signal input in the current iteration.

Note that in (28), the previous solution  $\tilde{\mu}^{k-1}$  is used instead of the current solution  $\tilde{\mu}^k$ . This is used in order to allow solving the primal and dual sub-problems in parallel. Thus, a second option would be to consider the current solution which gives the following input signal:

$$\mu^k = \frac{1}{k} \tilde{\mu}^k + \frac{k-1}{k} \mu^{k-1} \quad (29)$$

### 3.5.2. Weighted Mean-Value Cross Decomposition (WMVCD)

Another way of constructing the input signal was reported by Holmberg and Kiwiel in 2006 and yields a method that was proven to converge for nonlinear convex problems (Holmberg & Kiwiel, 2006). The method is called Weighted Mean-Value Cross Decomposition. WMVCD uses a weighted mean to construct the signal as in the following equation:

$$\mu^k = \delta_k \tilde{\mu}^{k-1} + (1 - \delta_k) \mu^{k-1} \quad (30)$$

Where  $\delta_i = \frac{\beta+\gamma}{\beta i+\gamma}$ , Holmberg and Kiwiel showed that appropriate values for  $\beta$  and  $\gamma$  are, respectively, 1 and 3 (Holmberg & Kiwiel, 2006).

### 3.5.3. One-sided Weighted Mean-Value Cross Decomposition (OWMVCD)

OWMVCD is a special case of WMVCD. In this method, the weighted mean-value is used to construct one of the two input signal whereas the second input signal is constructed directly from the current solution of the sub-problem. This method is practically used when the mean value of an input signal may lead to an infeasible solution to the original problem. This is the case with the input signal  $q$  in the industrial problem being treated here. An important disadvantage of this method is the lack of convergence proof.

Apart from the difference in the construction strategy of the input signal, all the three cross decomposition methods (MVCD, WMVCD and OWMVCD) yield the same optimization algorithm as shown below:

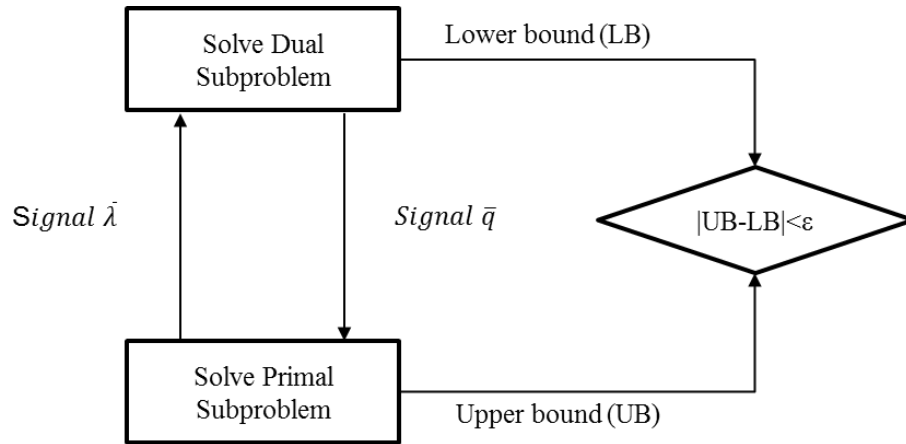


Figure 13 – Improved cross decomposition algorithms, based on (Hadera, 2015)

## 4. Mathematical Modeling

Once the theoretical and technical tools have been exposed and discussed in the literature review, those tools are applied to build the mathematical models corresponding to the two industrial problems which are treated here. Firstly, the RTN technique is applied to revise and simplify a pre-existent basic model of pulp production. Secondly, the model is extended to consider paper production and additional extensions. Thirdly, we integrate the production planning with energy management into a monolithic model for both problems.

### 4.1. Basic Model for Pulp Production

#### 4.1.1. Pre-existent Model

As presented in Figure 1, the basic model for pulp production represents the first stage of paper production in which the final product is pulp. This pre-existent basic model was developed by Hadera and co-authors in (Hadera, et al., 2015) . The RTN superstructure of the model is below in Figure 14.

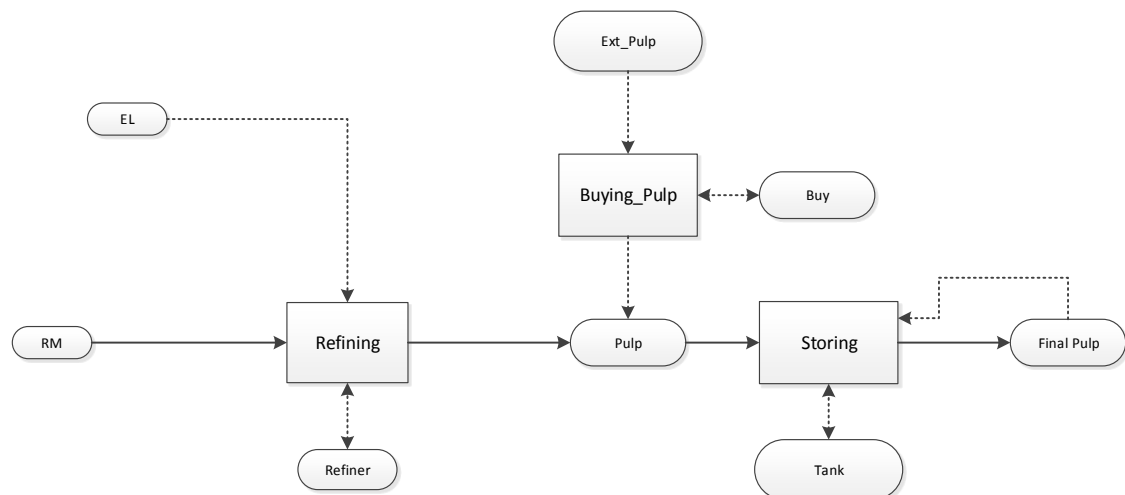


Figure 14 – RTN Superstructure of the basic pulp production model (Hadera, et al., 2015)

This basic model describes essentially the first stage in the paper production process. The final product in this model is the final pulp that will be fed to the paper machines. There is a possibility to store the final pulp before leaving the system. The excess balance equation used in this basic model was inspired by the formulation in the literature with some adaptations in order to handle the pulp storing possibility.

$$\forall r \in R, \forall t \in T$$

$$\begin{aligned} R_{r,t} = & R_{r,t}^0|_{t=1} + R_{r,t-1}^{end}|_{r \in R^{RM} \cup R^{pulp} \cup R^{FinalPulp}} \\ & + R_{r,t-1}|_{r \in R^{Refiner} \cup R^{Buy} \cup R^{Tank} \cup R^{Extpulp}} + \sum_{i \in I} (\mu_{r,i} N_{i,t} + v_{r,i} \xi_{i,t} \\ & + \bar{\mu}_{r,i} N_{i,t-1}) + \sum_{i \in I^{Storing}} (\bar{\mu}_{r,i} N_{i,t}|_{t=1}) + \Pi_{r,t}^{in}|_{r \in R^{EL}} \\ & - \Pi_{r,t}^{out}|_{r \in R^{FinalPulp}} \end{aligned} \quad (31)$$

$$\forall r \in r \in R^{RM} \cup R^{pulp} \cup R^{FinalPulp}, \forall t \in T$$

$$R_{r,t}^{end} = R_{r,t} + \sum_{i \in I^{Refining}} \lambda_{r,i} \xi_{i,t} + \sum_{i \in I^{Storing}} (\bar{v}_{r,i} \xi_{i,t} + \lambda_{r,i} \xi_{i,t}^*) + \sum_{i \in I^{Buying}} (\bar{v}_{r,i} \xi_{i,t}) \quad (32)$$

The previous expression of the excess balance equation is slightly different from the general form exposed in the literature review. Indeed, an additional variable ( $R_{r,t}^{end}$ ) has been added to handle the continuous resource quantities a part from the discrete ones. Besides, additional variables and structural parameters have been used to express the hybrid interaction between the storing tasks and the final pulp resource. However, as for the general excess balance equation, the quantities of each resource are either increasing (production) or decreasing (consumption).

The previous excess balance equation has been presented to show how the basic model looks like before the simplifications and then the extension. Hence, this model was not discussed here in details as will be the case with the simplified and extended model.

#### 4.1.2. Simplified Model

In order to simplify the basic model, the solution to the storing requirement has been rethought again. Indeed, the current way of handling the storing process adds more complexity to the modeling step as well as the solving step. Another problem with this model is that it does not make use of the propagation property of resource quantities in the RTN technique. This property of the RTN approach could be used to express the storing of the produced pulp in the system. In this way, the model could be simplified and the new RTN superstructure could be presented as in the Figure 15 below.

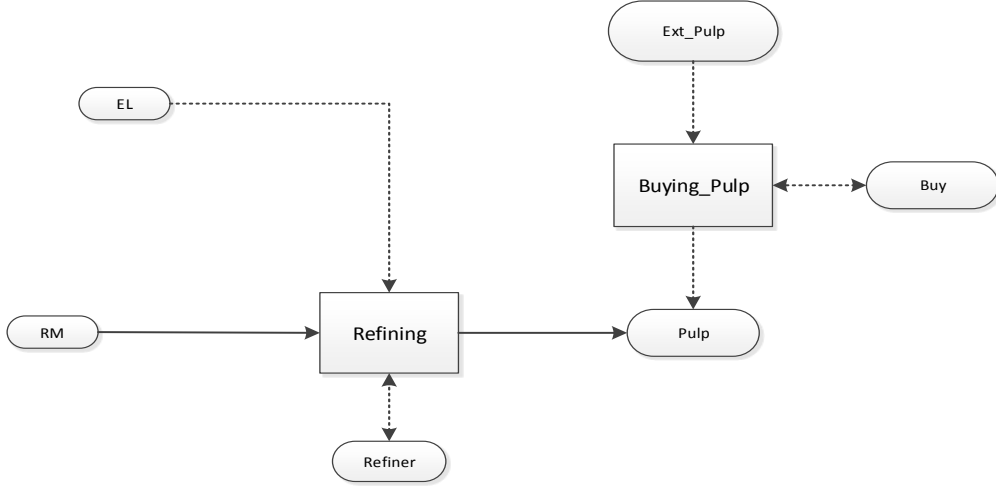


Figure 15 – RTN Superstructure of the simplified basic model for pulp production

Thus, instead of handling explicitly the storing requirement with an explicit processing task (i.e. storing), the new structure makes use of the pulp quantity propagation in the RTN excess balance equation to model the pulp storing process in a more efficient way. Therefore, there will be no need to consider additional variables and structural parameters and the excess balance equation will be more compact and expressive compared to the one in the pre-existent basic model:

$$\forall r \in R, \forall t \in T$$

$$Re(r, t) = Re(r, t - 1)_{r \neq EL} + \sum_{i \in Task} (\mu(r, i)N(i, t) + \lambda(r, i)\xi(i, t)) + \Pi^{in}(r, t) - \Pi^{out}(r, t) \quad (33)$$

To complete the modeling of the interactions between tasks and resources, the values of the structural parameters for continuous  $\lambda(r, i)$  and discrete interactions  $\mu(r, i)$  are set according to the RTN superstructure. For production interactions between a resource and a task, the value is set to +1 whereas for consumption interaction, it is set to -1. The value is set to 0 if there is no interaction. For instance, the refining task consumes continuously the raw material, therefore  $\lambda(raw\_material, refining)$  is set to -1.

Another parameter which has to be specified for the RTN model is the initial quantities of each resource. This could be given as follows:

$$\left[ \begin{array}{cccccc} & r = EL & r = RM & r = Refiner & r = Buy & r = Pulp & r = ext\_Pulp \\ Re(r, t = 0) & 0 & inf & 3 & 1 & init\_vol & inf \end{array} \right]$$

For unlimited resources such as raw material and external pulp, the initial quantities are expressed using a large number, whereas the limited amount, such as the stored pulp, can be initialized with an initial volume in the tank. Since the process includes 3 refiners and 1 source for buying, the initial quantities of refiner and buy resources are, respectively, 3 and 1.

All the resources propagate except electricity (i.e. EL); this is expressed in the term  $Re(r, t - 1)_{r \neq EL}$ . The main reason is that electricity cannot be stored but it is only provided from external sources (i.e. energy grid) at the beginning of each time slot. This supply of electricity is expressed in  $\Pi^{in}(r = EL, t)$ .

The supply of raw material and external pulp is considered as unlimited. However, there is a limited amount of external pulp that could be bought at the end of each time slot which can be expressed using the following constraint

$$\xi(i = Buying_{Pulp}, t) \leq Max_{pulp_{ext}} N(Buying_{Pulp}, t) \quad (34)$$

The demand of pulp is given in the parameter  $\Pi^{out}(r = Pulp, t)$ . Thus, this amount of pulp demand will be subtracted from the amount of pulp in the system at the end of time slot  $t$ . One may notice that the index  $\theta$  is considered neither for the continuous interactions  $\xi(i, t - \theta)$  nor for the discrete ones  $N(i, t - \theta)$ . Indeed, the processing duration of the continuous tasks are not known *a priori*. So, the duration is considered always one time slot (i.e.  $\theta \in \{0\}$ ). For the discrete interactions, which are mainly between tasks and equipment resources, the semantic of the variable  $N$  is the state of the processing task since there is only one equipment resource per task.

Thus, the basic model was significantly simplified. However, the model is still basic and does not include many of the production details. It will essentially serve as a basis towards a more general and extended model of paper production process.

## 4.2. Extended Model for Paper Production

Real world models for paper production consider certain level of details compared to the basic model for pulp production. Thus, as represented in Figure 2, instead of considering that the TMP is modeled with one refining stage, the real world models consider two different refining stages. The pulp produced in the first stage is used with a certain percentage directly to form the final pulp for bleaching. The remaining percentage is refined again using the refiners in the second stage before being bleached.

### 4.2.1. RTN Superstructure and Excess Balance Equation

The RTN superstructure can be built easily from the overview of the paper process production process in Figure 2. Using RTN conventions, Appendix 1 shows the superstructure of the developed extended model. In order to handle the storing functionalities, the same idea as in the simplified basic model could be used. Hence, the RTN excess balance equation will remain the same. The structural parameters alongside the initial quantities are set accordingly as described in the basic model.

### 4.2.2. Processing Rates

The scheduling time grid is discretized uniformly and the time slot length is noted as *TimePeriod* (given in hours). The time horizon is set by giving the total number of time slots. Continuous processing tasks have a limit on the amount they can process per time slot.

Those limits are given by minimal and maximal production rates (*MinRate* and *MaxRate*), the constraint could be formulated in the following form:

$$\forall i \in Production_{Tasks} \quad \forall t$$

$$TimePeriod.MinRate.N(i, t) \leq \xi(i, t) \leq TimePeriod.MaxRate.N(i, t) \quad (35)$$

Some material resources could be bought within a certain maximal rate *MaxBuy*. This limit could be set to zero to avoid any external purchase for certain material resources. The corresponding constraint is:

$$\forall i \in Buying_{Tasks} \quad \forall t$$

$$\xi(i, t) \leq TimePeriod.MaxBuy.N(i, t) \quad (36)$$

It is important to note that those rates depend on the time period which is a global parameter of the model. This allows more flexibility in considering the discretization and the time horizon.

#### 4.2.3. Machines Start-up and Stop

In order to consider some functional requirements in start-ups and stops of machines (refiners, bleachers and paper machines), some modifications need to be brought to the structural parameters and the excess balance equation. For instance, non-active refiners, when switched on, need some time to start processing the resources. Besides, there is a start-up cost that needs to be accounted in the total cost. There are two different ways of doing this:

- Using additional functional variables and constraints

As mentioned before, the state of each production task  $i$  at time  $t$  is indicated by the RTN integer variable  $N(i, t)$ . Additional positive variables  $st$  and  $end$  are used to capture, respectively, the number of machines that start and the ones that stop. The additional equation is therefore as follows:

$$\forall i \in Production_{Tasks} \quad \forall t$$

$$st(i, t) - end(i, t) = N(i, t) - N(i, t - 1) \quad (37)$$

Accounting the cost of start-ups and stops is now easy once the variables and the constraints are added. However, this does not allow forcing easily the start-up and stopping time for the tasks.

- Using additional RTN tasks and resources

In order to force the start-up and stop time for the tasks as well as to capture the number of starting and stopping machines, some RTN tasks and resources need to be added. For each production task  $i$ , two additional tasks will be added: one for starting idle machines and the other for stopping active machines. The RTN superstructure of a simple task-machine interaction as in Figure 6 becomes in this case as represented in Figure 16 below.



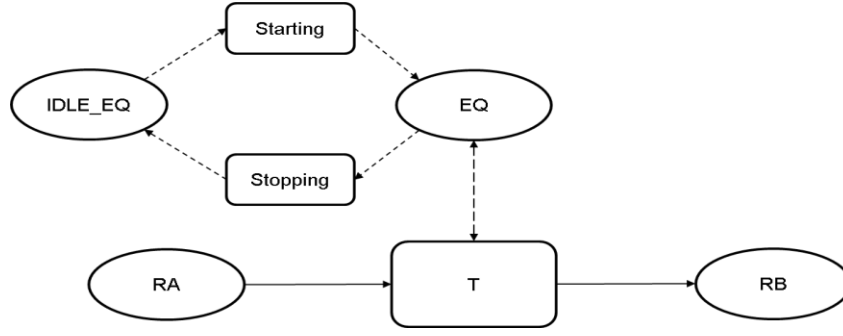


Figure 16 - Simple task-machine interaction with starting and stopping task

With the new model of task-machine interaction in mind, the RTN variable  $N(i, t)$  for the new tasks is exactly the same as  $st$  and  $end$  in the previous method. However, the power of this new way of modeling this problem is the ability to control the duration of the starting and stopping tasks. Indeed, instead of using the structural parameter for discrete interaction as  $\mu(r, i)$ , it is now more convenient to use  $\mu(r, i, t)$  with the additional time parameter. The excess balance equation will now include an additional summation over time for the discrete interaction term. The following figure illustrates the scenario of an example of starting task with 3 time slots as starting duration.

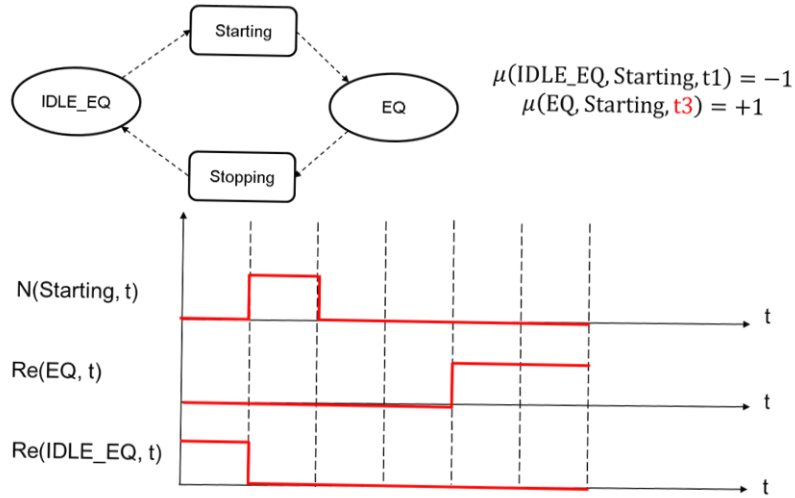


Figure 17 - Use of structural parameters for starting and stopping discrete tasks

#### 4.2.4. Late Orders

In order to add more flexibility to the demand satisfaction, late orders are considered in the model. However, penalties are applied whenever an order is not satisfied before its due date. To model this, one variable will be added to the excess balance equation. The variable  $LateOrder(r, t)$  will account the overdue quantity of resource  $r$  at time  $t$ . The new balance equation with the new structural parameters for discrete interaction as well as late orders is as follows:

$$\forall r \in R, \forall t \in T$$

$$Re(r, t) = Re(r, t-1)_{r \neq EL} + \sum_{i \in Task} \lambda(r, i) \xi(i, t) + \sum_{\tau} \mu(r, i, \tau) N(i, t - \tau + 1) + \Pi^{in}(r, t) - \Pi^{out}(r, t) + LateOrder(r, t) - LateOrder(r, t-1) \quad (38)$$

In case the scheduling starts initially with initial late orders, the variable  $LateOrder(r, t)$  for  $t = 0$  can be set to the initial late order of product  $r$ . As mentioned, those overdue quantities are penalized with a fixed cost  $LOCost$  in the objective function; therefore the cost can be easily accounted as in the following equation:

$$LateOrderCost(r) = \sum_{t \in T} LOCost \cdot LateOrder(r, t) \quad (39)$$

#### 4.2.5. Resource Storage

Resources can be stored in different storage tanks. Those tanks are characterized with minimal and maximal volume that can be stored. This is easily incorporated in the RTN model by adding the following constraint on each resource  $r$  at each time  $t$

$$MinVolume(r) \leq Re(r, t) \leq MaxVolume(r) \quad (40)$$

Storage cost can be accounted by associating a fixed cost  $StCost$  to each excess of a resource  $r$  in the storage tank as in the following equation

$$StorageCost(r) = \sum_{t \in T} StCost \cdot Re(r, t) \quad (41)$$

#### 4.2.6. Paper Making Change-over

The changeover in a machine is needed when two different products can be produced. This is for instance what happens for the paper machines with the different paper grades. The change-over in a machine  $R$ , performing two different tasks, could be illustrated as in the following figure:

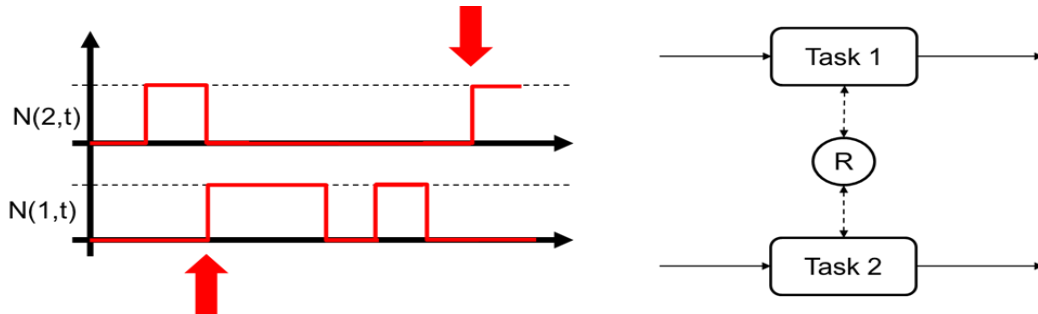


Figure 18 - Change-over illustration in a production machine

One may notice that the change-over occurs when the machine is shifting from performing one task to a different one, this is indicated in the figure using the red arrows. In order to

handle the changeover requirements, for instance: changeover costs and time delay, there are two different ways to do this:

- Using additional variables and constraints

This first solution requires some thinking effort to set the correct equations. It is instructive to discuss the equations based on a simple example. The following figure shows a scenario of the simple process presented in Figure 18. The variables  $N(i, t)$  represent the usual RTN state variables of task  $i$  at time  $t$ , we add an additional Boolean variable  $R(i, t)$  which indicates if the machine is assigned to perform task  $i$  at time  $t$ .

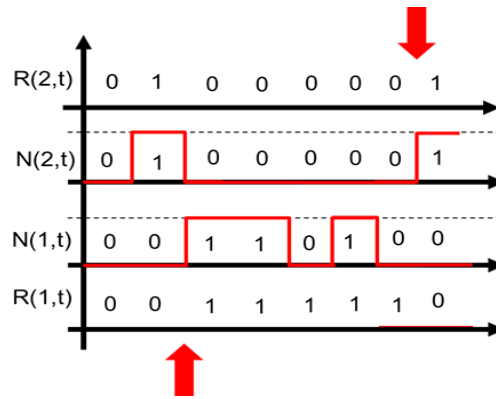


Figure 19 - Capturing machine change-over using additional variables

The following constraints give the relation between the variables  $R(1, t)$  and  $N$  for the example. Similar constraints could be written for  $R(2, t)$ :

$$R(1, t) \geq R(1, t - 1) - N(2, t) \quad (42)$$

$$R(1, t) \geq N(1, t) \quad (43)$$

One more important constraint has to be added to ensure that initially the machine is assigned to perform at most one task:

$$R(1, t = 0) + R(2, t = 0) \leq 1 \quad (44)$$

The change-over can now be easily captured using an additional equation and two constraint. The first constraint captures the change-over from task 1 to 2 and the second from 2 to 1:

$$IsChg(t) \geq R(1, t) - R(1, t - 1) \quad (45)$$

$$IsChg(t) \geq R(2, t) - R(2, t - 1) \quad (46)$$

The constraints can be easily generalized to the case where a machine can perform more than two tasks. Thus, the cost is accounted for a given change-over cost  $ChgOverCost$  as:

$$Chg_{cost} = ChgOverCost * \sum_t IsChg(t) \quad (47)$$

- Using additional RTN tasks and resources

The second way is by using additional resources and tasks in the RTN superstructure. The idea is to use the following change-over interactions instead of the ones used in Figure 18.

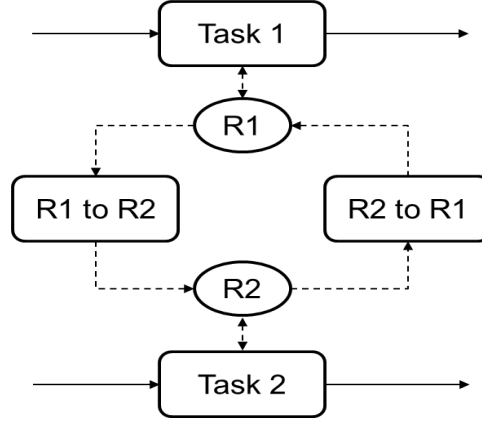


Figure 20 - Example of RTN superstructure to capture change-overs

The structural parameters are afterwards set accordingly. In this way, the constraints to capture the changeover are implicitly added to the model via the excess balance equation since the corresponding structural parameters were set. Once again, the method can be easily used for more than two tasks. Thus, the change-over cost can be accounted using the RTN state variables  $N(Ri \text{ to } Rj, t)$ .

#### 4.2.7. Energy Consumption

As mentioned in 2.3 about energy management model, the energy consumption is accounted as an optimal purchase strategy of the energy demand in each time slot of the production. The model is basically a flow network optimization problem. Since this energy optimization model is not the main topic of this project, we simply refer to (Hadera, et al., 2014a) for interested readers.

#### 4.2.8. Objective Function

The objective function expresses the different production and energy costs. This includes electricity consumption cost, material resource buying costs and storing costs as well as penalties on late orders. Electricity consumption cost is of course accounted from the optimal electricity purchase strategy using the energy optimizer. The overall cost of the process is given in the following equation:

$$Obj = ElCost + \sum_r (BuyingCost(r) + StorageCost(r) + LateOrderCost(r)) \quad (48)$$

### 4.3. Pre-existent Model for Stainless Steel Production

So far only pulp and paper production planning model was revised and extended, stainless steel production planning model will be shortly discussed in this section. Unlike pulp and paper production model which uses discrete scheduling, stainless steel production model uses

continuous-time production scheduling model. The starting time of a machine  $m$  which is processing a heat or a product  $p$  is noted as  $t_{m,p}^s$  and can take any real value in the scheduling time horizon. Like the pulp and paper case, the energy management is modeled as an Energy Flow Network - EFN for all the time intervals, i.e. for each time slot  $s$ . The EFN is modeled with one balancing node between the nodes which correspond to the sources of energy and the ones which represents  $q_s$  the demand of energy for production during the time slot  $s$  as well as the sales (Figure 4) (Hadera, et al., 2014a). The deviation penalties are modeled as the cost of the deviation of the load curve  $q = (q_s)$  from a pre-agreed curve. Thus, the overall cost of the steel production problem considered in this application is as follows.

$$Cost = c \sum_{m,p} t_{m,p}^s + \mu(q) + \delta(f) \quad (49)$$

An overview of the model corresponding to the steel production problem being studied is given below:

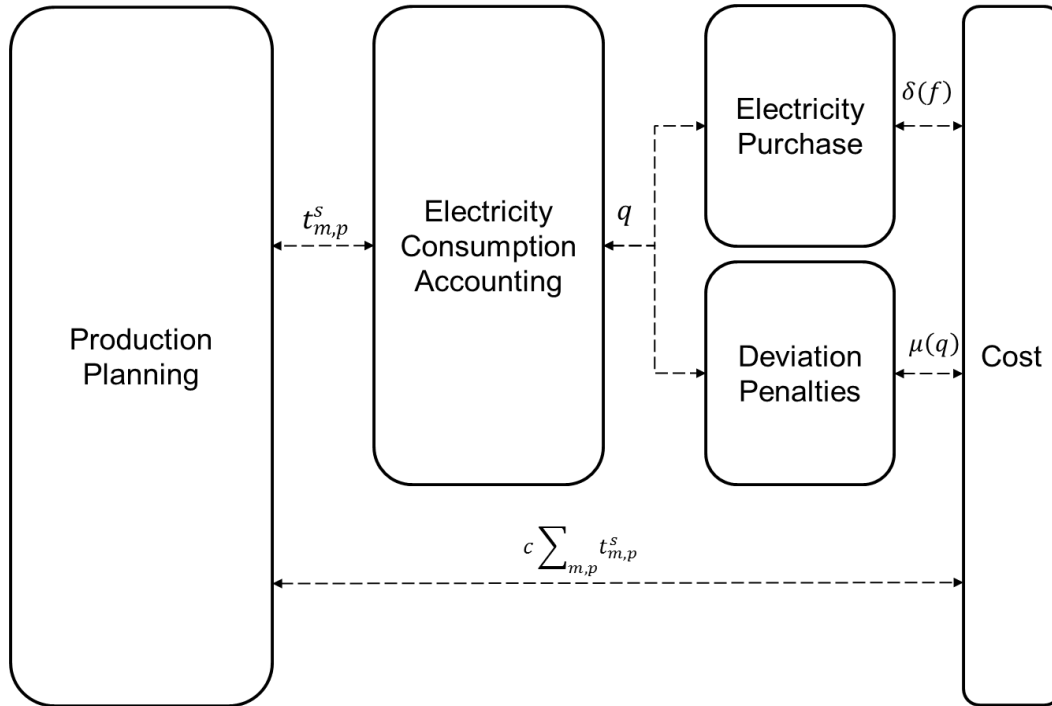


Figure 21 - Stainless production planning problem, based on (Hadera, et al., 2014a)

Hadera and co-authors have developed a detailed model for the monolithic problem as a MILP problem in (Hadera, et al., 2014a). The mathematical model was later on improved in (Hadera, et al., 2014b) with a more concise time accounting in the scheduling model.

The pre-existent developed MILP model can be formulated in a general compact form as minimizing the overall cost

$$c \sum_{m,p} t_{m,p}^s + \mu(q) + \delta(f) \quad (50)$$

With respect to the production planning constraints

$$A_{PP}^{(1)} y + D_{PP}^{(1)} t^s - b_{PP}^{(1)} \leq 0 \quad (51)$$

$$A_{PP}^{(2)} w + D_{PP}^{(2)} t^s - b_{PP}^{(2)} \leq 0 \quad (52)$$

$$Hw - Iq = 0 \quad (53)$$

And the energy flow network constraints

$$A_{EFN} f + D_{EFN} g - b_{EFN} \leq 0 \quad (54)$$

$$Af - Iq = 0 \quad (55)$$

And the variable type and bounds requirements

$$g \in G; y \in Y; q, f, w, t^s \geq 0 \quad (56)$$

The objective function (50) represents all the cost of the production process. The production planning constraints regarding the scheduling are given by the constraint (51). Whereas (52) allows the accounting of the production time which is translated to the energy consumption  $q$  with the constraint (53). The consumption  $q$  is linked to the flow from the balance node to the demand node using (55) and the flow network constraints are represented using (54). For more detailed equations of the models see (Hadera, et al., 2014a) and (Hadera, et al., 2014b).

In order to get a good feasible solution to this optimization problem for real industrial instances within a reasonable time, this monolithic formulation has to be changed and some heuristic has to be implemented. For this, Hadera and co-authors in (Hadera, et al., 2014a) developed a bi-level heuristic that decomposes the monolithic problem into two main levels. The basic idea is to use some variable fixings between the different levels and iterate after adding cuts. Fixing some variables decreases the complexity of the problem in each level. This allows iterating until a CPU time limit is reached to get a good feasible solution which is not necessarily optimal since this is a heuristic method.

Appendix 2 shows the optimization algorithm using the bi-level heuristic to solve the monolithic problem for stainless steel production process.

## 5. Energy Integration to Production Planning

After revising and developing an extended version of pulp and paper production planning model as well as revising an already existing stainless steel production planning model, we get two of the most interesting mathematical models for energy integration using different decomposition schemes.

Through the literature review, different methods were described which led at the end to the so-called cross decomposition methods. These decomposition methods do not only meet the industrial requirements of using the pre-existent optimizers, but are also considered among the best decomposition methods in the literature dealing with this decomposition problem. The study here could be considered as the first to investigate those methods on real industrial problems. In this chapter, energy management for pulp and paper production is first integrated using simple decomposition schemes. Second, stainless steel case requires complex schemes. Along the way, the mathematical models will be used in a compact form.

### 5.1.1. Simple Decomposition Scheme for Pulp and Paper Production

The pulp and paper production planning model is a MILP model which is based on the RTN scheduling technique. To serve the purpose of energy integration to production planning using cross decomposition methods, we first start by writing the two main sub-problems: the primal sub-problem which is the Energy Flow Network - EFN model with fixed flow on one arc and the dual sub-problem which is the Production Planning– PP model with a marginal cost for pricing energy in the objective function.

On the one hand, the primal EFN sub-problem is easily constructed by fixing  $q$  to a certain value  $\bar{q} \geq 0$  and considering the energy flow network sub-problem as minimizing

$$C_{EFN}^T f \quad (57)$$

With respect to the energy flow network constraints

$$A_{EFN} f + D_{EFN} g \leq b_{EFN} \quad (58)$$

$$A f - I \bar{q} = 0 \quad (59)$$

$$g \in G; f \geq 0 \quad (60)$$

On the other hand, the dual sub-problem is constructed by fixing the Lagrangean multiplier  $\lambda$  to the marginal cost  $\bar{\lambda}$  of the complicating constraint in the previous primal EFN sub-problem. The dual PP sub-problem corresponds to the following production planning problem as minimizing

$$Obj_{PP}(x) + \bar{\lambda} q \quad (61)$$

With respect to the production planning constraints

$$A_{PP} y + D_{PP} x \leq b_{PP} \quad (62)$$

$$H x - I q = 0 \quad (63)$$

$$y \in Y; x, q \geq 0 \quad (64)$$

As presented in section 3.5, iterating between the two sub-problems leads to the cross decomposition methods. The algorithm iteration is done as described in Figure 13 which does not need the resolution of any master problem. The construction of signals defines the decomposition method or signal strategy of the algorithm. As discussed in the literature review, One-sided Weighted Mean Value strategy, which will be used here, leads to better results for simple iterative schemes (Holmberg & Kiwiel, 2006).

### 5.1.2. Complex Decomposition Scheme – Stainless Steel Model

The monolithic formulation, given previously, is almost similar to the general monolithic formulation used in Section 4.3 about the pre-existent stainless steel production planning model. Hence, we can easily formulate the primal EFN and the dual PP sub-problems of the monolithic stainless steel problem according to the results from the literature review.

On the one hand, the EFN primal sub-problem is similar to the one considered previously for pulp and paper case since both industrial problems use the same energy management model. On the other hand, the PP dual sub-problem is constructed by fixing the Lagrangean multiplier  $\lambda$  to the marginal cost  $\bar{\lambda}$  of the complicating constraint in the EFN primal sub-problem. Thus, the PP dual sub-problem corresponds to the following production planning problem as the minimization of

$$c \sum_{m,p} t_{m,p}^s + \mu(q) + \bar{\lambda}q \quad (65)$$

With respect to the production planning constraints

$$A_{PP}^{(1)}y + D_{PP}^{(1)}t^s \leq b_{PP}^{(1)} \quad (66)$$

$$A_{PP}^{(2)}w + D_{PP}^{(2)}t^s \leq b_{PP}^{(2)} \quad (67)$$

$$Hw - lq = 0 \quad (68)$$

$$y \in Y; q, w, t^s \geq 0 \quad (69)$$

As mentioned before, cross decomposition methods iterate by exchanging input signals between the primal and dual sub-problem. More precisely, dual PP sub-problem sends a load curve  $q = (q_s)$  to the primal EFN sub-problem. The latter sends the marginal cost  $\bar{\lambda}$  of the complicating constraint to set the shadow prices in the dual PP sub-problem.

To apply the decomposition scheme with the signal exchange to the bi-level heuristic, which are used to solve the problem, two different approaches were investigated.

- **Bi-level heuristic with Benders' sub-problem at all levels**

A first approach could be to use the bi-level heuristic on the primal PP sub-problem. That is to say that the original problem is decomposed to an energy flow network problem which is easily solved to optimality as well as the production planning which is solved using the bi-



level heuristic until a CPU time is reached. The cross decomposition algorithm iterates between the two decomposed problems by exchanging the signals until another CPU time is reached. For the detailed algorithm, the first figure in Appendix 3 is presenting the overall architecture of the algorithm iteration.

- **Bi-level heuristic with Benders' sub-problem at the upper level**

In contrast with the previous approach, the second approach uses the cross decomposition scheme in the upper level of the PP primal sub-problem. The choice of using the decomposition only in the upper level is justified by the fact that this level serves in the bi-level heuristic basically as an indicator of the direction of where it is worth looking for the optimal solutions. The remaining problems at the lower level  $LL$  are only evaluating the proposed guesses provided by the upper level problem. Following this idea another structure of the bi-level heuristic integration with Cross Decomposition concept can be formulated. For more details about the algorithm, the second figure in Appendix 3 is showing an overview of the iterative scheme.

## 6. Prototype Software Development

After the development of the functional models, it is very important to build a demonstrator to show the different functionalities of the developed models and to do the case studies. In this chapter, we describe the development of a prototype demonstrator for pulp and paper production models. The development is done in C# object oriented programming language using Microsoft Visual Studio 2013 as an Integrated Development Environment (IDE).

First, the architecture of the software is discussed. Second, a presentation of the conventional structure of the input data file is given. Third, an overview of the object representation and data structure that is used in the demonstrator is exposed. Finally, the graphical user interface is shown and described.

### 6.1. Architecture Overview

The demonstrator consists basically of several modules, each of which is used to perform specific task. The input data file is an Excel file that contains a description of the problem. Before getting the final results of the optimization, there are several steps to go through starting from reading the input data from the Excel file. Once the tables are stored, the data is organized in a specific data structure as C# objects. The objects data is translated into GAMS file that describes the problem data as well as the different RTN parameters. This GAMS file is later on used alongside the generic GAMS model for optimization. The solution is finally saved as a GDX file in the temporary directory and as an Excel file in the local directory for results visualization. The following figure gives an overview of the architecture:

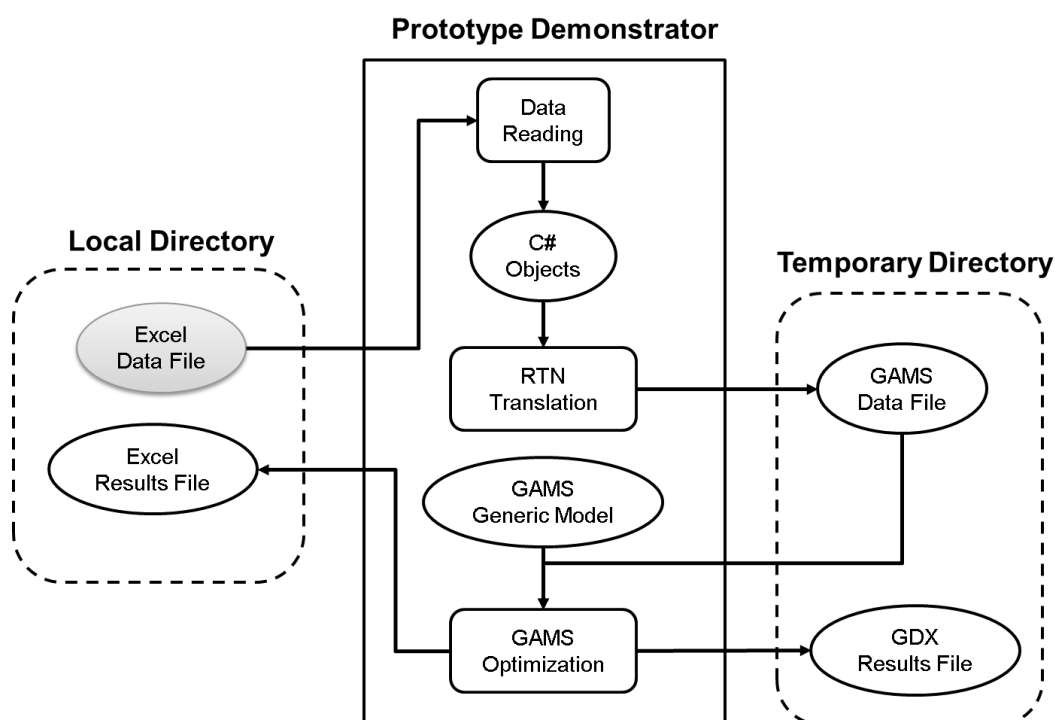


Figure 22 - Structure overview of the demonstrator software

It is important to mention that GAMS optimization module is basically performed using CPLEX solver for MILPs which is one of the solvers available in GAMS.

## 6.2. Input Structure

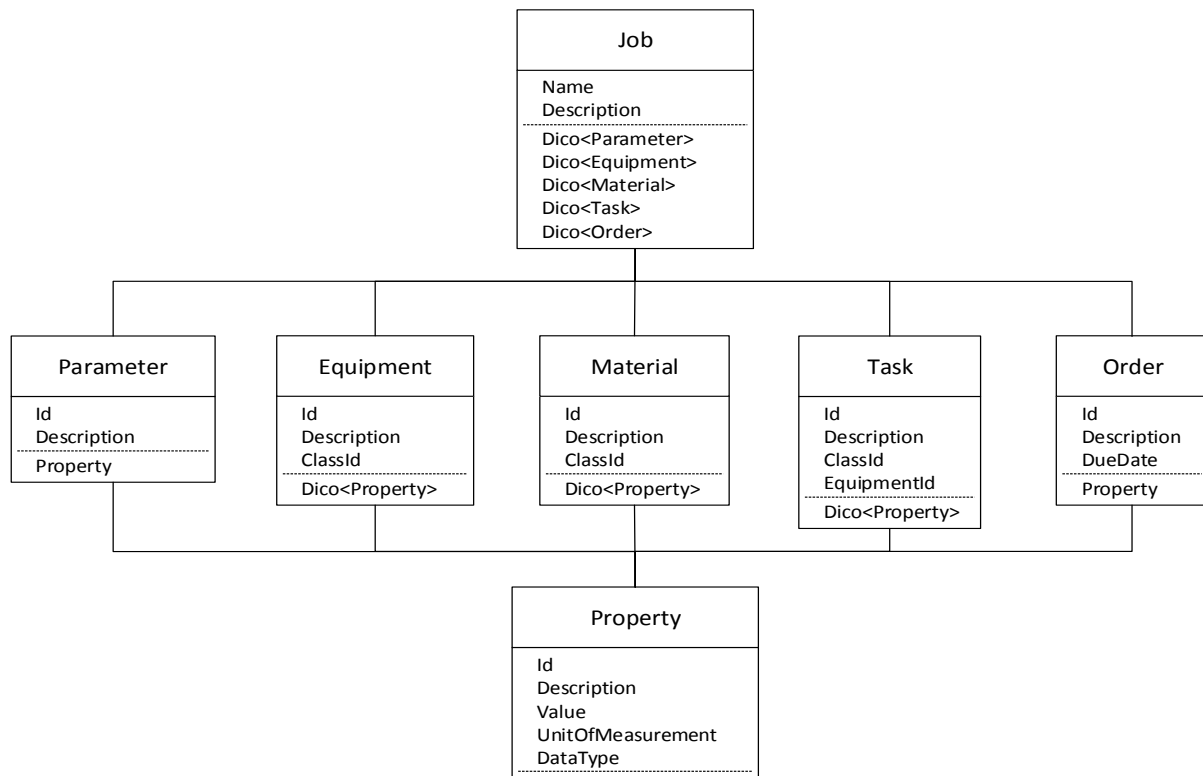
As mentioned previously, the input files are Excel files with a conventional structure which describes the problem to be solved. A valid data Excel file contains different sheets that contain a table describing different elements of the problem in tables. The sheets are:

- **Parameters:** Describes the different parameters of the problems such as the time period, the number of slots, reference date for scheduling and some Boolean parameters to consider or not different production costs.
- **Equipment Information:** This sheet contains information about the physical equipment present in the production process such as machines (paper machines, refiners, etc.) and storage tanks. For instance, one can specify the volume capacity of the storage tanks, products to store and the corresponding storage cost. One can also specify the number of equipment of one type, the products that can be produced as well as the initial state of the equipment.
- **Material Information:** This sheet contains information about the physical material used in the process such as raw material, energy, intermediate products and final products. One can specify the purchase capacity and prices as well as the delay cost for late orders of products.
- **Task Information:** This sheet is very important because it describes the production process. One has to mention all the tasks that take place in the process by specifying the equipment which performs the task, consumed and produced material, the main product and the maximal and minimal rate of production. For starting, stopping and change-over tasks, the duration of the shifting as well as the consumed and produced materials has to be mentioned.
- **Order Information:** This last sheet describes the ordered products, quantities as well as their due date.

The input files are translated in an intermediate step into a GAMS data file with descriptions of the RTN interactions. Thus, it is very important to mention that the Excel input file does not require any prior knowledge of the RTN technique or GAMS syntax. It is enough to check the sample example for any issue with the conventional structure and the key words.

## 6.3. Objects Representation

C# is an object oriented programming language which uses the concept of objects as the basic data structure. This concept is therefore used in the architecture of the software to store and organize the components of the input data for later use and even for later modifications or extensions. As mentioned previously, any job and its corresponding input data is composed of parameters, equipment, materials, tasks and orders. Those components are implemented as C# classes which are used to instantiate their corresponding objects. Each component is using a data unit called property to represent useful information of the component data. The overall object representation is presented in the following UML class diagram:



*Figure 23 - UML class diagram showing the overall objects representation*

It is important to mention that the data structure which is used to group data units is the in-built C# dictionary. A C# dictionary is basically a collection of data values and keys. Microsoft .NET Framework includes useful methods which allow easy manipulation of the data type. Once the objects representation is established, one may easily access and use the data stored in the program. This could be done by implementing useful methods inside the appropriate classes.

#### **6.4. Graphical User Interface (GUI)**

To facilitate the demonstrator utilization, a basic graphical user interface was developed. As represented in Figure 22, the demonstrator is used to read an Excel input data to generate an RTN model in the form of a GAMS file before using the generic GAMS model to create another Excel file which contains the optimization results. The graphical interface is presented and described in Appendix 4.

## 7. Case Study

In this chapter, the developed models will be used and checked with real industrial scenarios. Those scenarios are presented first for each industrial problem and then used to show the behavior of the models. The results are discussed in forms of concluding remarks in the last chapter. Pulp and paper production planning problem is studied first before moving to the second problem about stainless steel production planning. All the case study has been conducted using GAMS/CPLEX 24.2.3 on a personal computer with Intel® Core™ i5-2400 CPU @ 3.10 GHz with 4.0 GB RAM.

### 7.1. Setup and Input Data – Pulp and Paper Production

As discussed in the previous chapters, an extended model has been developed as well as the energy integration for pulp and paper production problem. Therefore, the scenarios will serve to show the new capabilities of the model as well as to check the performance of the energy integration methods. Thus, seven scenarios are considered out of the basic real scenario (i.e. scenario S0). The seven scenarios are used to show different aspects of the extended model. The basic real scenario is as follows:

*Table 1 - Basic real scenario (S0) for pulp and paper production*

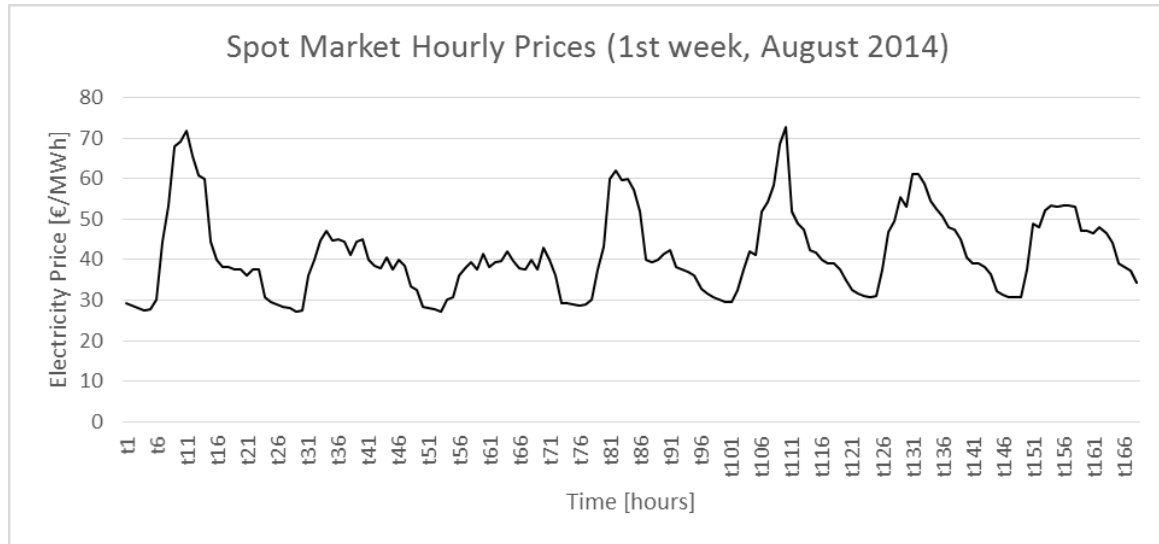
Scenario	Discretization	Costs	Machines	Storages	Orders
<b>S0</b>	168 slots of 1 hour (1 week)	change-over, start-up & stop	3 main refiners, 1 paper machine	(40-60) % for pulp tanks	4

Based on the real scenario (S0), seven other scenarios will be generated. The following table shows the different scenarios and the investigated aspect as well as the modifications brought to the basic scenario:

*Table 2 - Case study scenarios for pulp and paper production*

Scenario	Subject	Modification to Scenario (S0)
<b>S1</b>	energy cost	add energy cost
<b>S2</b>	flexible storages	add energy cost, (20 - 80)% of pulp tanks
<b>S3</b>	very flexible storages	add energy cost, , (5 - 95)% of pulp tanks
<b>S4</b>	larger storages	add energy cost , (5 – 1,5)% of pulp tanks
<b>S5</b>	1 more refiner	add energy cost, add 1 refiner
<b>S6</b>	1 more refiner and flexible storage	add energy cost, add 1 refiner, (5 - 95)% of pulp tanks
<b>S7</b>	delayed orders	add energy cost and delay cost

In order to account energy cost, we considered on-site generation and hourly spot market. For the spot market, we considered the hourly prices in the first week of August 2014. The following figure shows the price curve:



*Figure 24 - Spot market hourly prices for energy cost (NordPoolSpot, 2014)*

Once the scenarios were built from the basic case excel file, they are solved using two different models: Monolithic and Iterative. The monolithic model integrates the energy management in one model in order to get the best solution, whereas the iterative model implements the cross decomposition method with one-sided weighted mean value as the signal exchange strategy.

## **7.2. Results - Pulp and Paper Production**

The instances were first solved using the monolithic model in order to find the optimal or near optimal feasible solution if the CPU time exceeds the limit of 10 minutes. The instances were later on solved using the iterative scheme. The numerical results were as follows:

Table 3 - Numerical results of the monolithic model for different instances

Scenario	Model statistics			Model results			
	Bin vars	Cont. vars	Equations	EFN Cost	PP Cost	Overall Cost	Integration Profit
<b>S0</b>	8568	15580	11979	122.212 €	25.358 €	147.571 €	(*)
<b>S1</b>	8568	15580	11979	113.348 €	25.622 €	138.970 €	6%
<b>S2</b>	8568	15580	11979	110.258 €	25.612 €	135.870 €	8%
<b>S3</b>	8568	15580	11979	108.078 €	25.567 €	133.645 €	9%
<b>S4</b>	8568	15580	11979	106.635 €	26.150 €	132.785 €	10%
<b>S5</b>	8568	15580	11979	113.348 €	25.622 €	138.970 €	6%
<b>S6</b>	8568	15580	11979	106.170 €	26.182 €	132.352 €	10%
<b>S7</b>	8400	15580	11979	113.348 €	25.622 €	138.970 €	6%

(\*) There is no profit of energy integration because the basic scenario is the reference.

The monolithic will serve to determine the quality of the developed iterative scheme. The following table presents the results for the iterative scheme and the solution quality for each scenario:

Table 4 - Numerical results of the cross decomposition iterative method

Scenario	Model statistics (EFN)		Model statistics (PP)			Model results		
	Cont. vars	Equations	Bin. vars	Cont. vars	Equations	No. Iterates (Best)	Overall Cost	Relative Gap
<b>S1</b>	1177	841	8568	14404	11139	4 (1)	138.963 €	0,01%
<b>S2</b>	1177	841	8568	14404	11139	5 (3)	135.930 €	0,04%
<b>S3</b>	1177	841	8568	14404	11139	4 (1)	133.713 €	0,05%
<b>S4</b>	1177	841	8568	14404	11139	4 (1)	132.797 €	0,01%
<b>S5</b>	1177	841	8568	14404	11139	2 (1)	138.956 €	0,01%
<b>S6</b>	1177	841	8568	14404	11139	4 (3)	132.324 €	0,02%
<b>S7</b>	1177	841	8400	14404	11139	2 (1)	138.963 €	0,01%

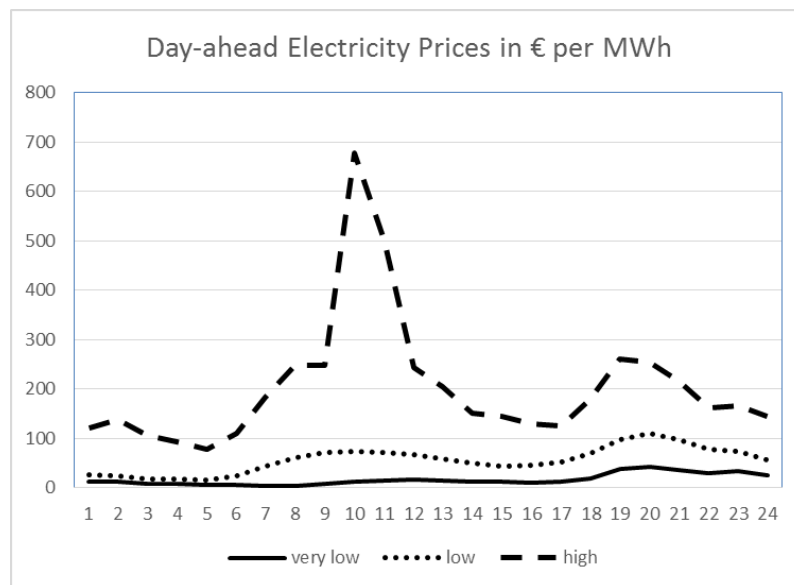
### 7.3. Setup and Input Data – Stainless Steel Production

The input data and the problem instances which are used in this study are the same as in (Hadera, et al., 2014a). Six different scenarios are considered here as presented in the following table:

*Table 5 - Investigated problem instances for cross decomposition*

<i>Scenario</i>	<i>Horizon</i>	<i>Products</i>	<i>day-ahead electricity prices</i>
<b>1</b>	24h	20	High
<b>2</b>	24h	20	Low
<b>3</b>	24h	16	high
<b>4</b>	18h	12	High
<b>5</b>	24h	16	Low
<b>6</b>	24h	20	very low

In addition to four scenarios from (Hadera, et al., 2014a), this study has included two additional instances (scenario 5 and 6). Scenario 5 is similar to 2 except that the number of products is 16 instead of 20, whereas scenario 6 is scenario 1 with very low day-ahead electricity prices as shown in the following figure:



*Figure 25 - Hourly day-ahead electricity prices for 24 hours (EPEX, 2013)*

Several models based on different decomposition solution approaches are implemented and tested. The following table gives a description of the models used in this case study.



Table 6 - Solution algorithms implemented for stainless steel production case study

<i>Name</i>	<i>Model type</i>
<b>BR</b>	<b>Bi-level heuristic</b> using improved event binaries model (cf. (Hadera, et al., 2014b))
<b>DH</b>	Cross decomposition with <b>direct signal</b> on <b>all levels</b> in the bi-level heuristic.
<b>DO</b>	<b>One-sided weighted mean value</b> cross decomposition on <b>all levels</b> in the bi-level heuristic.
<b>WH</b>	Cross decomposition with <b>direct signal</b> on <b>the upper level</b> in the bi-level heuristic.
<b>WO</b>	<b>One-sided weighted mean value</b> cross decomposition on <b>the upper level</b> in the bi-level heuristic.

#### 7.4. Results – Stainless Steel Production

First, the problem instances are solved using the heuristic bi-level algorithm. This allows finding good quality solutions to the monolithic problem after a CPU time limit. The solutions are used later to evaluate the quality solution of the decomposed models. The following table shows the best solutions found after 10 minutes of CPU time:

Table 7 - Numerical results of the bi-level heuristic algorithm on the different instances

Scenario	Model statistics (LL2)					Iteration results		Best Possible Solution
	Binary variables	Total variables	Equations	MIP solution	Relative gap	No. of iterations	Best Iteration	
<b>BR1</b>	1458	10308	70735	193844	9,86%	5	2	176452
<b>BR2</b>	1458	10308	70735	165197	9,09%	5	5	151434,3
<b>BR3</b>	1276	8068	55248	174017	9,31%	5	1	159197,4
<b>BR4</b>	724	4708	28709	194082	3,51%	18	5	187493,1
<b>BR5</b>	1276	8068	55248	136998	13,10%	5	5	121128,8
<b>BR6</b>	1458	10308	70735	124292	14,28%	5	5	108764

Cross decomposition method is applied to the six instances. Two different decomposition approaches are studied and each one with two different signal strategies. The first approach is to use the decomposition for all the levels of the bi-level heuristic. The corresponding results are given below:

Table 8 - Numerical results of the decomposition for all levels of the bi-level heuristic with two different signal strategies

Scenario	Model Statistics						Iteration results		
	PP LL2			EFN			MIP solution	Relative gap	No. iterates (best)
	Binary variables	Total variables	Equations	Binary variables	Total variables	Equations			
DH1	1434	10044	70431	24	265	305	193599	9,72%	3 (1)
DO1	1434	10044	70431	24	265	305	193599	9,72%	3 (1)
DH2	1434	10044	70431	24	265	305	165425	9,24%	3 (1)
DO2	1434	10044	70431	24	265	305	165425	9,24%	3 (1)
DH3	1252	7804	54944	24	265	305	178235	11,96%	3 (1)
DO3	1252	7804	54944	24	265	305	178288	11,99%	3 (1)
DH4	706	4510	28483	18	199	227	200274	6,82%	3 (2)
DO4	706	4510	28483	18	199	227	200274	6,82%	3 (2)
DH5	1252	7804	54944	24	265	305	143221	18,24%	3 (2)
DO5	1252	7804	54944	24	265	305	143221	18,24%	3 (2)
DH6	1434	10044	70431	24	265	305	129968	19,50%	3 (1)
DO6	1434	10044	70431	24	265	305	129968	19,50%	3 (1)

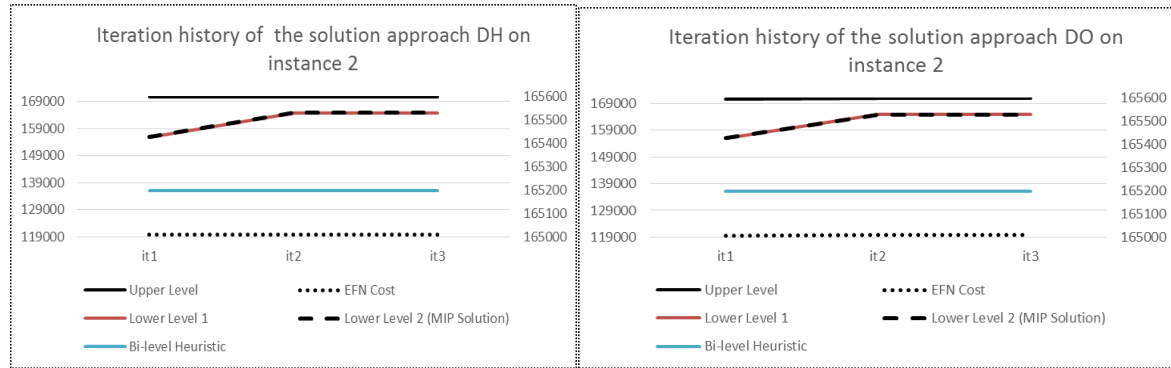
The instances are later on tested using the second decomposition approach, which differs from the previous approach by considering the cross decomposition only on the upper level instead of all the levels in the bi-level heuristic. The numerical results are as follows

Table 9 - Numerical results of the decomposition for the upper-level of the bi-level heuristic with two different signal strategies

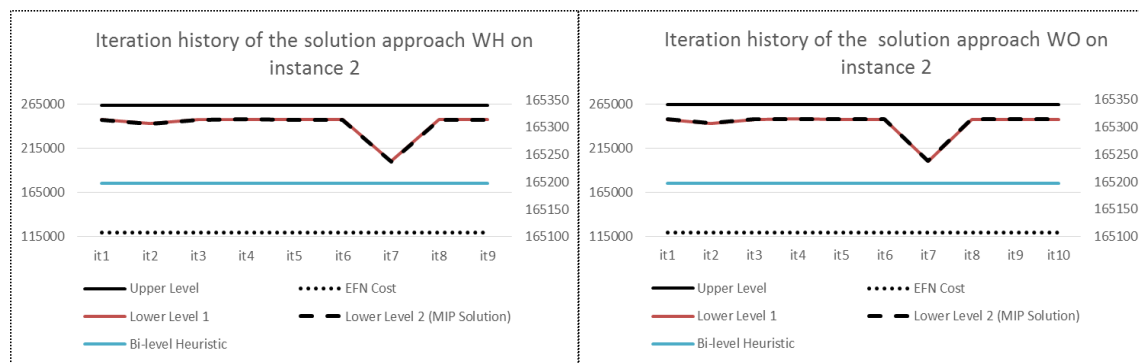
Scenario	Model Statistics						Iteration results		
	PP LL2			EFN			MIP solution	Relative gap	No. of iterates (Best)
	Binary variables	Total variables	Equations	Binary variables	Total variables	Equations			
WH1	1434	10044	70431	24	265	305	197337	11,84%	9 (4)
WO1	1434	10044	70431	24	265	305	197337	11,84%	9 (4)
WH2	1434	10044	70431	24	265	305	165237	9,12%	9 (7)
WO2	1434	10044	70431	24	265	305	165237	9,12%	10 (7)
WH3	1252	7804	54944	24	265	305	176335	10,77%	3 (1)
WO3	1252	7804	54944	24	265	305	176335	10,77%	3 (1)
WH4	706	4510	28483	18	199	227	194650	3,82%	4 (3)
WO4	706	4510	28483	18	199	227	194650	3,82%	4 (3)
WH5	1252	7804	54944	24	265	305	142546	17,68%	3 (3)
WO5	1252	7804	54944	24	265	305	142265	17,45%	3 (3)
WH6	1434	10044	70431	24	265	305	129754	19,30%	10 (2)
WO6	1434	10044	70431	24	265	305	129754	19,30%	10 (2)

Overall, the decomposition approaches tested give slightly worse results than the heuristic bi-level. Yet, the quality of the solution is still good, knowing that the relative gap does not exceed 20% of the best possible optimal solution which means that the relative gap to the optimal solution is actually better.

In order to see the behavior of the two approaches with the two signal strategies, the iteration history of the algorithm is presented below for one of the problem instances. The second instance (i.e. scenario 2) is chosen here since it has more realistic input data.



*Figure 26 - Iteration history of the first decomposition approach on instance 2*



*Figure 27 - Iteration history of the second decomposition approach on instance 2*

To be able to show the variations of the iterative solution in one figure, the curves of the upper level and EFN in the figures above are scaled according to the right vertical axis, whereas the other curves are scaled according to the left axis. The iteration history shows that the two signal strategies give slightly the same behavior for the two approaches and that the second algorithm performs more iterations which allows to better approach the best solution from the bi-level heuristic (around the 7<sup>th</sup> iteration). This could be explained by the fact that the more iteration the decomposition algorithm performs the more signal exchange happens and thus the optimization is done with better knowledge of the two decomposed components of the monolithic problem.

## 8. Concluding Remarks

### 8.1. Conclusions

To sum up the results from the two studies, the following conclusions can be drawn concerning the different cross decomposition approaches and signal strategies:

- For pulp and paper production case, a simple cross decomposition method gives very good results. Thus, this decomposition method will serve the purpose of the integration as intended. However, in the stainless steel production case, there is no approach which is generally better. One may prefer to use the second approach (decomposition only in the upper level) which showed better results than the first one. This could be explained by the fact that this second approach makes less errors since the decomposition is applied only in the upper level of the bi-level heuristic.
- When it comes to the signal strategy, in pulp and paper production case, one-sided weighted mean value was used and yields very good results. Yet, for the stainless steel case, there is no big difference between the two strategies which have been investigated (direct and one-sided weighted mean value signal). This is due to the fact that the two signal strategies behave in the same way in the first iterations of the algorithm because the mathematical sequences are initially similarly constructed.
- The algorithms developed here have no controllability. That is to say that the iteration solutions are not expected to systematically improve after each iteration. The main reason is that the cross decomposition method, with the signal strategies used here, was not proven to converge for MILP problems. This study could be used as an example to show that the method for the investigated signal exchange strategies does not converge for MILP problems. Moreover, in the stainless steel production case, the decomposition algorithms were developed upon the bi-level heuristic.
- Since there is no obvious convergence test, the algorithm stops iterating after a fixed time limit. This is a very convenient property for industrial use, since the main purpose is to find a good quality solution within a time limit.
- The quality of the solution is determined using the best possible solution provided by the solver when solving the monolithic original problem. This bound corresponds to the best lower bound. Since the problem is a MILP, this means that the real relative gap to optimality is probably even better (less) than the ones reported here.

### 8.2. Limitations

The decomposition algorithms investigated here have several limitations that could be summarized as follows:

- In the stainless steel production case, the algorithms are based on a heuristic algorithm (i.e. bi-level heuristic). This means that there may be an accumulated error from this heuristic. Therefore, the error made by the algorithm is as good as the one from the bi-level heuristic.
- The algorithms use the Lagrangean relaxation. This generally yields a duality gap which is not necessarily equal to zero for MILP problems as is the case with the

problem studied here. This means that the decomposition method does not generally give an optimal solution.

- There is no best way of initializing the algorithm. Here and for academic purposes, some input data were used for the initialization. As noted in (Holmberg, 1994), it is important to provide a good initialization strategy, which could give a reasonable guess of one of the signals.
- In a study by Holmberg (Holmberg, 1997), it is pointed out that the decomposition method may not show the best performance on unbalanced separable problems. The study here could be a practical proof that this is true. Indeed, the decomposition method performs in the paper production case better than in the stainless steel production case. The latter case gives unbalanced separable problems (i.e. PP is a much more complex problem than EFN). This would be a limitation of the application of the decomposition method on such industrial problems.
- The second signal strategy (i.e. one-sided weighted mean value), that was investigated in the stainless steel case, allows more concurrency in the resolution of the sub-problems (i.e. EFN and PP). That is to say that the sub-problems could be solved at the same time in a parallel environment. The implemented algorithms do not use this property and are fully sequential which doubles the resolution time.

### **8.3. Future Work**

There are several areas for improvement and future work to further enhance the performance of the decomposition algorithms which were developed in this project. As mentioned in the previous section, this preliminary study opens the door for new theoretical and practical future work. On the one hand, the methods could be theoretically studied to clarify the convergence properties of the method for different signal strategies on MILP problems. Besides, the developed models could be used to investigate new signal exchange strategies. On the other hand, the iterative algorithms could be implemented in a more efficient way so that it is possible to parallelize the resolution especially in a parallel environment. Such parallelization of the decomposition algorithms will decrease the resolution time and therefore increase the quality of the solutions.

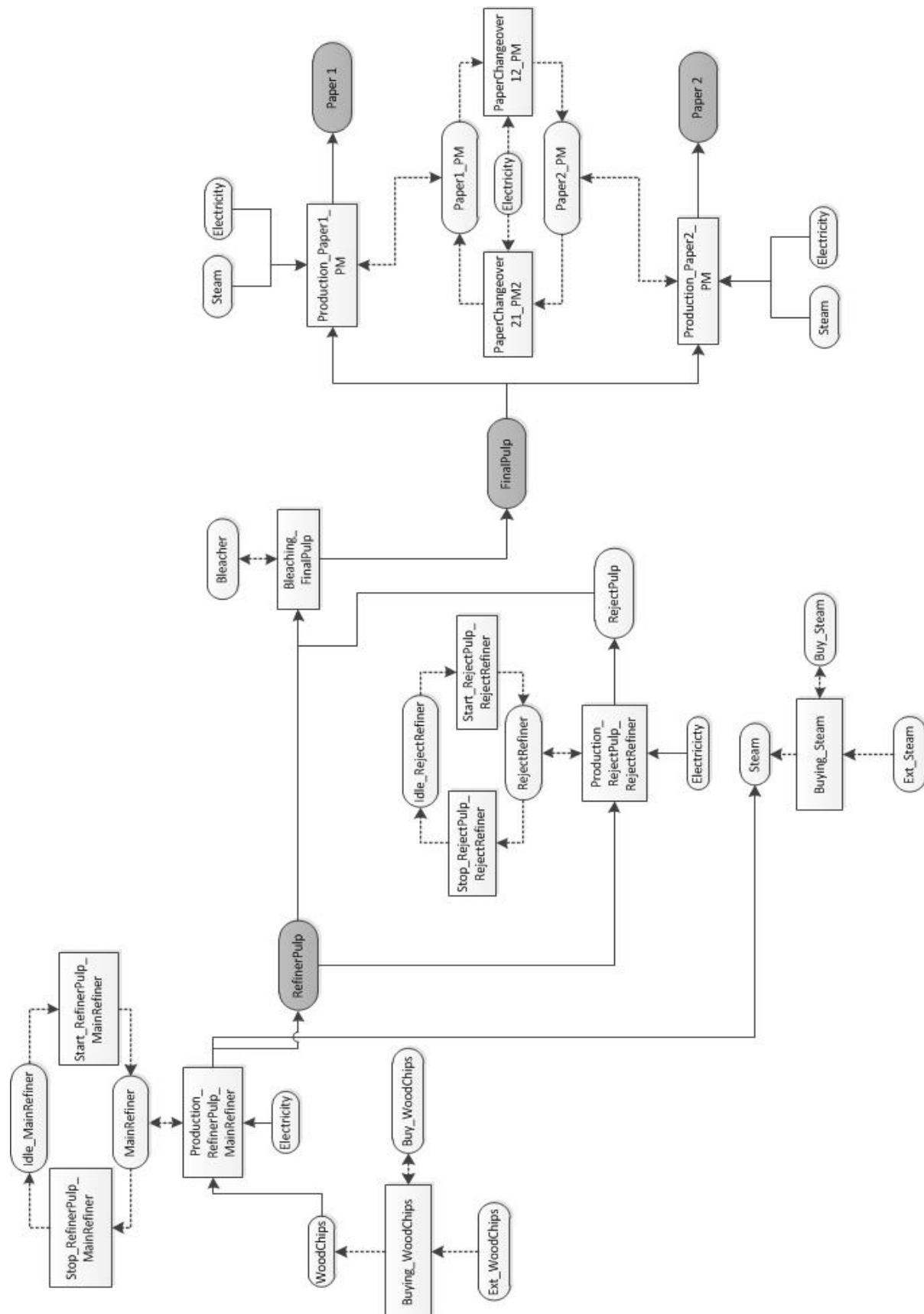
When it comes to the prototype software, there is still room for improvements. First, the graphical user interface could be enhanced to contain more settings and parameters. Second, one could implement a nice graphical visualization of the solution variables for an easy and quick analysis of the optimization results. Finally, since the prototype is based on a generic RTN formulation, then it could be easily generalized for other industrial problems, for instance stainless steel production problem.

## Bibliography

- ABB, 2014. *Who we are - ABB in brief*. [Online]  
Available at: <http://new.abb.com/about/abb-in-brief>  
[Accessed 01 08 2014].
- Bajpai, P., 2012. Brief Description of the Pulp and Paper Making Process. In: *Biotechnology for pulp and paper processing*. s.l.:Springer.
- Benders, J. F., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, Volume 4, pp. 238-252.
- Castro, P. M., 2012. *Tutorial on Resource-Task Network based mathematical formulations*, internal: ABB Corporate Research Center .
- Castro, P. M., Harjunkoski, I. & Grossmann, I. E., 2009. New Continuous-Time Scheduling Formulation for Continuous Plants under Variable Electricity Cost. *Industrial and Engineering Chemistry Research*, 48(14), pp. 6701-6714.
- EPEX, 2013. *EPEX Spot - European Power Exchange*. [Online]  
Available at: [www.epexspot.com](http://www.epexspot.com)  
[Accessed 11 11 2013].
- Hadera, H., 2015. *Electricity demand-side management in process plants*, Dortmund: Technical Univeristy of Dortmund.
- Hadera, H. et al., 2014a. *Continuous-time Steel Production Scheduling Under Complex Time-sensitive Electricity Optimization*, Ladenburg: ABB Corporate Research.
- Hadera, H. et al., 2014b. Improved Energy-awareness Formulation for Mixed Integer Linear Programing Continuous-time Scheduling Models.
- Hadera, H. et al., 2015. *A Cross Decomposition Strategy for Industrial Demand-Side Management of Pulping Process*. s.l., s.n.
- Harjunkoski, I. et al., 2014. Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, Volume 62, pp. 161-193.
- Harjunkoski, I., Westerlund, T., Isaksson, J. & Skrifvars, H., 1996. Different formulations for solving trim loss problems in a paper-converting mill with ILP. *Computers & Chemical Engineering*, Volume 20, pp. 121-126.
- Holmberg, K., 1992. Linear mean value cross decomposition: A generalization of the Kornai-Liptak method. *European Journal of Operational Research*, Volume 62, pp. 55-73.
- Holmberg, K., 1994. A Convergence Proof for Linear Mean Value Cross Decomposition. *ZOR - Methods and Models of Operations Research*, Volume 39, pp. 157-186.

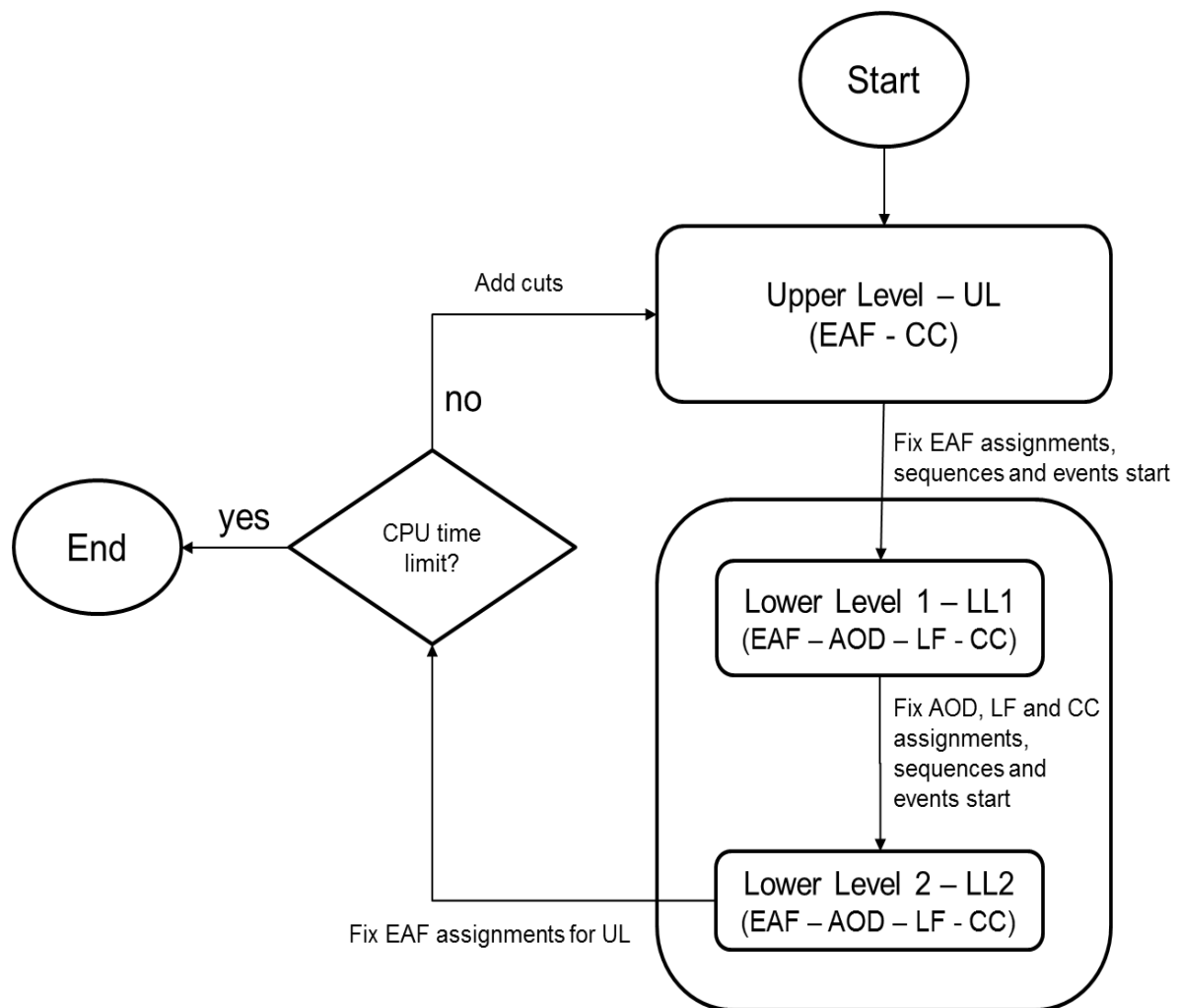
- Holmberg, K., 1997. Mean value cross decomposition applied to integer programming problems. *European Journal of Operational Research*, pp. 124- 138.
- Holmberg, K. & Kiwiel, K. C., 2006. Mean value cross decomposition for nonlinear convex problems. *Optimization Methods and Software*, Volume 21:3, pp. 401-417.
- Merkert, L. et al., 2014. Scheduling and energy - Challenges and Opportunities. *Computers & Chemical Engineering*, 72(2 January 2015), pp. 183 - 198.
- NordPoolSpot, 2014. *Nord Pool Spot*. [Online]  
Available at: <http://www.nordpoolspot.com/#/nordic/table>  
[Accessed 10 12 2014].
- Pantelides, C. C., 1994. *Unified Frameworks for the Optimal Process Planning and Scheduling*. New York, s.n.
- Pulkkinen, P. & Ritala, R., 2008. TMP production scheduling under uncertainty: Methodology and case studies. *Elsevier - Chemical Engineering and Processing*, Volume 47, pp. 1492-1503.
- Roy , V. T. J., 1983. Cross Decomposition for Mixed Integer Programming. *Mathematical Programming*, Volume 25, pp. 46-63.
- Swedish Energy Agency, 2013b. *Slutlig statistik för industrins energianvändning 2011*. [Online]  
Available at: <http://www.energimyndigheten.se/Press/Pressmeddelanden/Slutlig-statistik-for-industrins-energianvandning-2011/>  
[Accessed 3 February 2014].
- Wolfe, P. & Dantzig, G. B., 1960. Decomposition Principle for Linear Programs. *Operations Research*, p. 101–111.

## Appendix 1 – RTN Superstructure of Paper Production





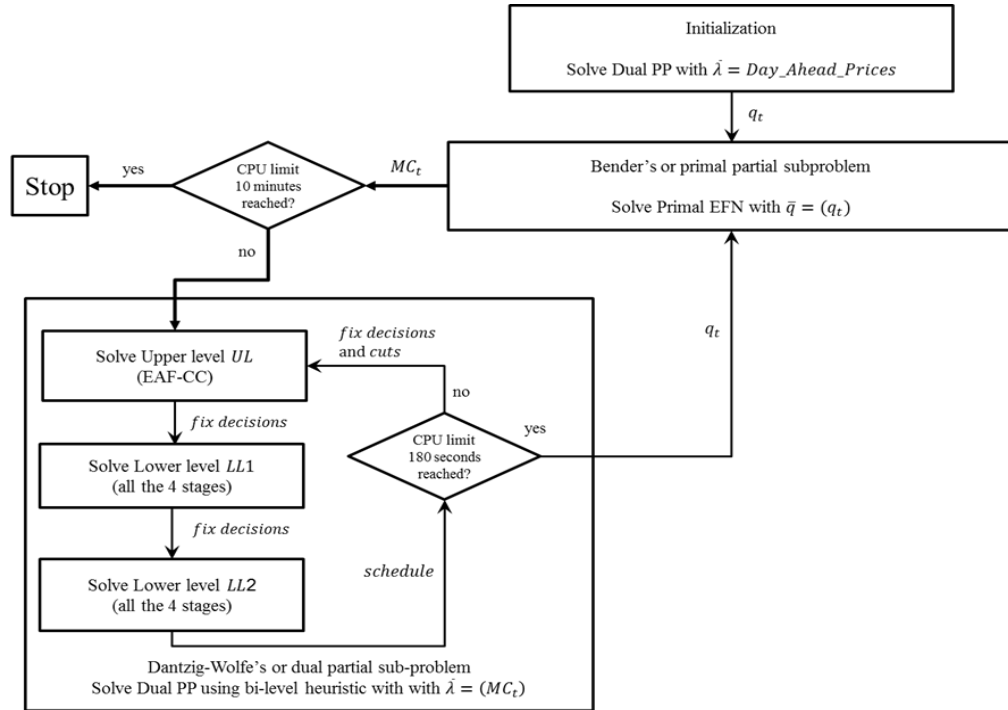
## Appendix 2 – Bi-level Heuristic for Stainless Steel Problem



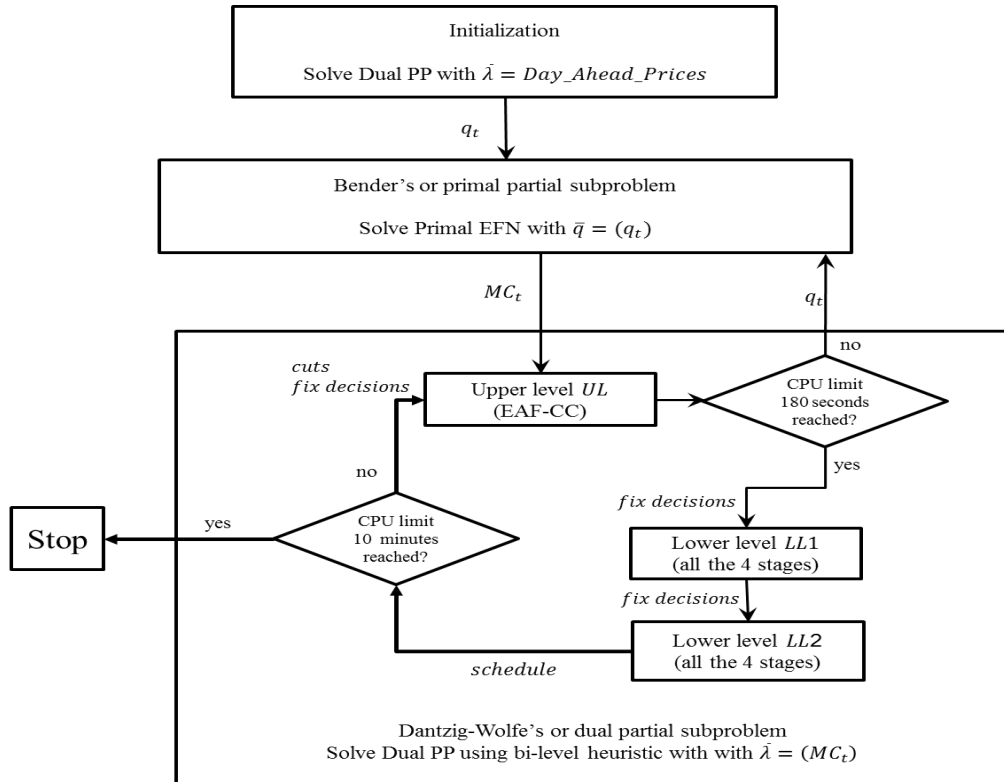
The figure is based on (Hadera, et al., 2014b).

## Appendix 3 – Decomposition Algorithms on Bi-level Heuristic

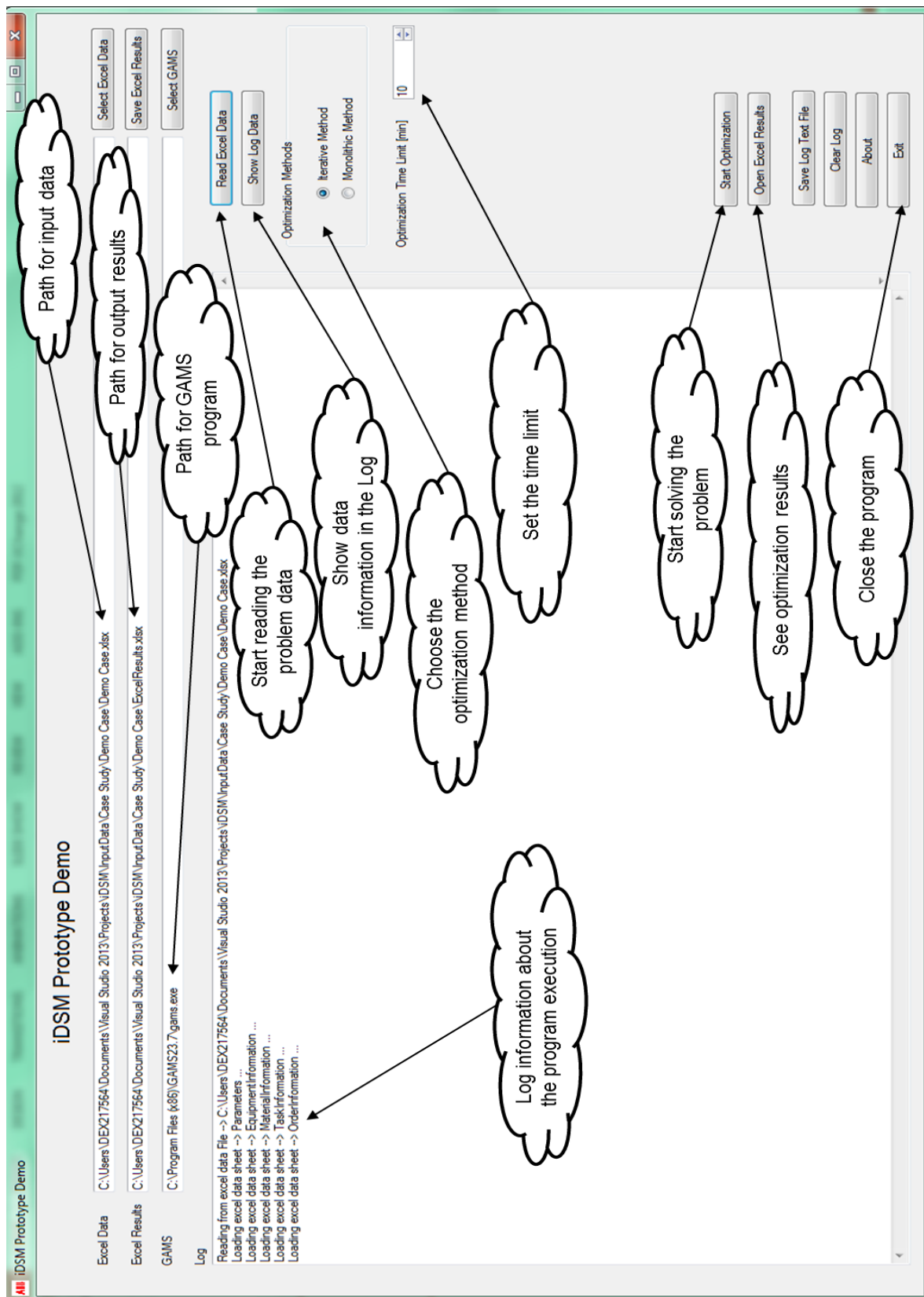
- Approach 1: Cross decomposition algorithm on the two levels of the bi-level heuristic, based on (Hadera, 2015)



- Approach 2: Cross decomposition algorithm on the upper level of the bi-level heuristic, based on (Hadera, 2015)



## Appendix 4 – Prototype Graphical User Interface







TRITA -Mat-E 2015:01  
ISRN -KTH/MAT/E-15/01-SE