Master Thesis

Nested Decomposition Algorithm to Solve Multistage Transmission Expansion Planning Problems

Confidentiality Notice

This report is not confidential. It can be published on the Internet.

Author: Ghita KASSARA
Field of Study: Optimization, Operations Research and Command
Master in Parallel: Optimization, Université Paris-Saclay
Scholar Year: 2016-2017

Supervisor at ENSTA :
Pierre CARPENTIER

Supervisors at PSR :
Camila NUNES METELLO
Mario VEIGA PEREIRA

Internship performed from 5 April 2017 to 5 August 2017

PSR – Energy Consulting and Analytics
Centro Empresarial Rio- Praia de Botafogo, 228- Botafogo, Rio de Janeiro - RJ, 55580-906, Brasil

Abstract

Long-term Transmission Expansion Planning is the problem of deciding at each time stage which new transmission lines should be added to an existing transmission network in order to satisfy system objectives efficiently. It is one of the main strategic decisions in power systems and it is also known to come with a computational burden. Indeed, the presence of integer variables in a multistage stochastic setting makes the computation of the optimum difficult.

A recent paper developed a finite and exact Nested Decomposition algorithm able to solve Multi-stage Stochastic Integer Programs efficiently. The aim of the thesis is to apply this algorithm in order to solve Long-term Transmission Expansion Planning problems.

Keywords : Multi-stage Stochastic Integer Programming, Long-term Transmission Expansion Planning, Nested Decomposition Algorithm, Cutting planes, SDDP

Acknowledgments

I would like to express my sincere gratitude and profound regards to my supervisor at PSR, Camila Nunes Metello for the patient guidance, listening and encouragement. I have been lucky to have a supervisor who deeply cared about my work, and who always answered my questions.

My sincere thanks also goes to Dr Mario Veiga Pereira who provided me an opportunity to join the PSR team as an intern, and who gave me such a challenging and interesting internship topic. I am very obliged to him for his advices and directives during the internship.

Besides my supervisors, I am very obliged to Dr Gerson Couto for his knowledge on Transmission Expansion Planning problems, to Dr Sergio Granville for his advices and to Dr Maria de Lujan Latorre for her help on test instances.

I also take this opportunity to express a sense of gratitude to Dr Raphael Chabar for his cordial support on all the logistical matter, that made my four months in Rio de Janeiro a pleasure .

I finally thank all my colleagues at PSR for their sympathy and the very nice working atmosphere.

Table of Contents

Part 1

Introduction

This thesis applies Nested Decomposition algorithms to long-term Transmission Expansion Planning problems. In this section, we first present the problem, discuss related work, present the framework used to solve the problem and provide a summary of our contributions.

1.1 Long term Transmission Expansion Planning problem

The electric transmission grid is the infrastructure transferring the electricity from generation plants to substations. From there, the distribution grid handles the delivery of electricity to consumers. Transmission Expansion Planning problem has been defined in [9] as "deciding which new lines will enable the system to satisfy forthcoming loads with the required degree of reliability". Hence, in the long-term horizon, the goal is to choose, **stage-by-stage**, the new transmission lines that should be added to the existing transmission network, by satisfying the physical constraints and supplying the forecasted demand over the planning horizon.

Arguably, the design of the power transmission network is one of the most strategic questions for policy-makers when power systems are at stake. In [11], the authors shed the light on the new challenges to transmission expansion planning. They point out that recent developments such as regional planning or renewable integration, have increased the complexity and relevance of a problem that has already been addressed in a vast literature.

As a matter of fact, there is currently an increasing number of ambitious projects aiming at installing a vast amount of new generation capacity. These projects extend across the borders of a single country and need important transmission investments in order to support the long distances on which they spread. Hence, the size of the system under optimization **increases from countries to regions**. That is why the resolution method for the problem should be able to deal with even larger problem sizes.

In addition, the penetration of renewable power and its erratic nature bring new challenges. On the one hand, it strengthens the need of reinforcing the network, so as it becomes able to export excess generation or provide a back-up when renewable power is not available. On the other hand, **it extends the uncertainty of the system** since it becomes crucial to deal with several operational scenarios.

Consequently, Transmission Expansion Planning problem is by nature a **multistage stochastic problem with combinatorial nature** since the investment decisions are either to invest or not to invest.

1.2 Prior work

Because of its complexity, the problem has been addressed in its static form, that is to say as a snapshot of the long-term system at one particular moment, in a vast majority of literature ([8], [7], [17], [18]). To be precise, the static form is a subproblem of the original problem, which aims to determine, at a given time, the new transmission facilities that should be installed.

Among all optimization techniques used to solve the static version, Benders decomposition has been a successful resolution method. Applied to the problem in [12], the method consists in subdividing the static Transmission Expansion Planning problem into a *master* and a *slave* subproblems. The *master* subproblem deals with the investment variables and propose a network expansion plan. The *slave* subproblem performs the expansion plan suggested by the *master* and checks its feasibility. The iterations between the two problems are then ensured by Benders' cuts, computed from the *slave*'s solution and added to the *master* subproblem. In [4], the authors introduce a new Benders decomposition approach using a mixed linear (0-1) formulation which ensures the optimality of the solution found.

Tackling the dynamic version of the Transmission Expansion Planning problem, as done in this thesis, comes with the computational burden of integer variables in a stochastic setting. Given the complexity of this option, most research focused on small case studies or used heuristic methods, according to [11].

1.3 Multistage stochastic programming

Multistage stochastic programming is a framework for sequential decision making under uncertainty, where the uncertainty is modeled by general stochastic process. Let us describe a generic nested formulation for a Multistage Stochastic Linear Programming (MSLP) problem **in the mixed-integer linear setting** with T stages, as done in [14].

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}_{\xi_2}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}_{\xi_3}\left[\cdots + \mathbb{E}_{\xi_T}\left[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T\right]\right]\right] \tag{1.1}$$

In the above formulation, the variables are x_t associated to the costs c_t and the set of constraints is formed by $\{x_t \geq 0| \quad B_{t+1} x_t + A_{t+1} x_{t+1} = b_{t+1}\}$.

Since we are in a stochastic setting all (A_t, B_t, c_t, b_t) are random variables. The stochastic data process at time t is $\xi_t = (A_t, B_t, c_t, b_t)$ and the expectation at time t is taken relatively to ξ_t. Without loss of generality, ξ_1 is assumed deterministic.

When the problem involves integer variables, the problem is a Multistage Stochastic Integer Program (MSIP).

Solving approaches for (MSP) involve to approximate the stochastic data process by a finite set of many realizations. Hence, one can model stochastic problems as a scenario tree. We denote by $\mathcal{N}$ the considered scenario tree, by $a(n)$ the parent node of a node n of the tree, with the convention that $x_{a(1)} = 0$ when $n = 1$ is the root node of the tree. We also denote by p_n the probability associated with the node n.

One can approximate (1.1) as the following **extensive form** :

$$\min_{p_n(x_n)_{n\in\mathcal{N}}} \left\{ \sum_{n\in\mathcal{N}} c_n^\top x_n : B_n x_{a(n)} + A_n x_n = b_n, x_n \geq 0, \quad \forall n \in \mathcal{N} \right\} \tag{1.2}$$

A well-known alternative for this extensive form is a Dynamic Programing formulation involving nested cost-to-go functions (or Bellman equations [1]). We introduce the set $\mathcal{C}(n)$ of children nodes of node n and the transition probability from node n to m, q_{nm}.

The **Dynamic Programing equations** are then :

$$\begin{aligned} Q_n(x_{a(n)}) &= \min_{x_n} \left\{ c_n^\top x_n + \mathcal{Q}_n(x_n) : B_n x_{a(n)} + A_n x_n = b_n, x_n \geq 0 \right\} \\ &\text{with} \\ \mathcal{Q}_n(x_n) &= \sum_{m\in\mathcal{C}(n)} q_{nm} Q_m(x_n), \quad \forall n \in \mathcal{N} \end{aligned} \tag{1.3}$$

$\mathcal{Q}_n(\cdot)$ corresponds to the **expected cost-to-go function** for the node n. In a linear setting, these functions are **piece-wise linear and convex**. Hence, it is possible to under-approximate them by linear cuts as in Nested Benders' decomposition methods [2], [5]. The key of the algorithm is to approximate the cost-to-go functions by adding Benders' cuts. It has also been proven that the algorithm converges in finite steps towards the optimal solution.

When the scenario tree representing the stochastic process is large the previous algorithm might not perform well. That is why its stochastic variant has been introduced in [13] and named the Stochastic Dual Dynamic Programming (SDDP) algorithm. SDDP performs then a sampling of scenario by exploiting the stage-wise independence of the process.

However, these decomposition techniques could not be applied to (MSIP), which are the scope of our thesis. Due to presence of integer variables, the expected cost-to-go functions are not piece-wise linear and convex anymore. **Recently, the authors of [19] developed effective decomposition algorithms for (MSIP) problems**. They were able to adapt decomposition algorithms and its stochastic variant SDDP to the binary setting, by introducing, amongst others, a new class of customized cuts .

Hence, the first objective of the thesis is to **apply the Nested Decomposition algorithm to the long-term Transmission Expansion Planning problem**.

Also, we would like to take into advantage the leverage done on the static problem to handle the cases where the scenario tree is large.

1.4 Contributions of the thesis

The key contributions of the thesis are summarized below :

- We implement the application of the Nested Decomposition algorithm developed in [19] to the multistage Transmission Expansion Planning problem.
- We perform computational experiments on two instances of tests widely used in the literature.
- We extend the work already done on the static version of the problem and come with a **a new double decomposition approach** .

The thesis is organized as follows. In Part (2), we model the Transmission Expansion Planning problem, in its static and dynamic versions. In Part (3), we present the Nested Decomposition algorithm we will use to solve our model. The Part (4) is dedicated to computational experiments. In Part (5), we give the new double decomposition approach and test it on our instances. Finally, we provide some concluding remarks in Part (6).

Part 2

Modeling of the Transmission Expansion Planning problem

The following chapter presents the mathematical formulation of the Transmission Expansion Planning problem. As a first step, the notations are presented. Then, the deterministic version of the problem is formalized as an optimization problem. Finally, the stochastic and definitive model is presented.

The electrical network is modeled as a directed graph. The vertices of the graph represent the buses and the edges represent the circuits. At each vertex, there might be generation and load. For instance, the figure below represents a network with 6 buses and 6 existing circuits. This network is actually the Garver system [8] and we will use it as a reference for our tests.

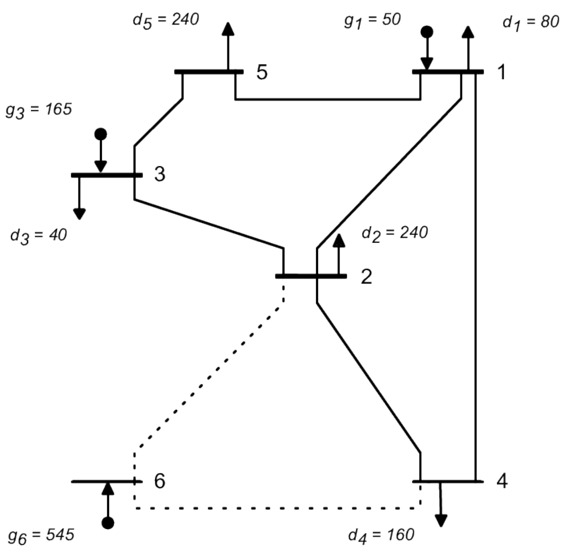

Figure 2.1: Initial confirguation of Garvers' network

2.1 Notations

Sets

J Set of existing circuits

K Set of candidate circuits

I Set of buses

T Set of stages of the planning horizon

Parameters

Bus parameters

g_{ti} Active generation at time $t \in T$ for bus $i \in I$

d_{ti} Load at time $t \in T$ for bus $i \in I$

Circuit parameters

c_k Investment cost to build the candidate circuit $k \in K$

γ_l Susceptance for circuit $l \in K \cup J$

$\bar{f}_l$ Flow limit for circuit $l \in K \cup J$

M_k Disjunctive parameter associated to candidate circuit $k \in K$

A Incidence matrix with dimension $I * (K + J)$, defined as it follows :

$$\forall i \in I, \forall l \in K \cup J, \qquad A_{il} = \begin{cases} -1 & \text{if circuit } l \text{ starts from } i \\ 1 & \text{if circuit } l \text{ ends in } i \\ 0 & \text{otherwise} \end{cases}$$

Variables

Investment variable

x_{tk} Investment decision at time $t \in T$ for circuit $k \in K$. It is a **binary** variable :

$$\forall t \in T, \forall k \in K, \qquad x_{tk} = \begin{cases} 1 & \text{if we invest in candidate circuit } k \\ 0 & \text{otherwise} \end{cases}$$

Operational variables

f_{tl} Algebraic power flow at time $t \in T$ for circuit $l \in K \cup J$. We consider the convention :

$$\forall t \in T, \forall l \in K \cup J, \qquad f_{tl} \begin{cases} \geq 0 & \text{if the flow is directed as the circuit} \\ < 0 & \text{otherwise} \end{cases}$$

$\Delta\theta_{tl}$ Angle at time $t \in T$ for each circuit $l \in K \cup J$. It is defined as the difference between the voltage angles of the extremal buses of the circuit l :

$$\Delta\theta_{tl} = \theta_{t,fr} - \theta_{t,to}$$

with fr and to the starting and ending extremal buses of circuit l, and $\theta_{t,i}$ the voltage angle of each bus $i \in I$.

2.2 Modeling as a Multistage Integer Problem

Let us first consider the static problem, where the planning horizon is limited **to one fixed stage** t. The aim is to determine which candidate circuit should be installed, while respecting the physical constraints of the electric grid and meeting the demand for electricity. We give here the mathematical formulation of the problem as an optimization problem.

Objective

$$\text{Min} \sum_{k \in K} c_k x_{tk} \tag{2.1}$$

The objective function of the problem corresponds to the minimization of all investments costs in building new transmission lines.

Constraints

Binary variables

$$x_{tk} \in \{0, 1\} \quad \forall k \in K \tag{2.2}$$

First Kirchhoff law

$$\sum_{l \in K \cup J} A_{il} f_{tl} = d_{ti} - g_{ti} \quad \forall i \in I \tag{2.3}$$

Second Kirchhoff law

$$f_{tj} = \gamma_j \Delta\theta_{tj} \qquad \forall j \in J \tag{2.4.a}$$

$$\left| f_{tk} - \gamma k \Delta\theta_{tk} \right| \leq M_k (1 - x_{tk}) \quad \forall k \in K \tag{2.4.b}$$

Flow limits

$$\left| f_{tj} \right| \leq \bar{f}_j \qquad \forall j \in J \tag{2.5.a}$$

$$|f_{tk}| \leq \bar{f}_k x_{tk} \quad \forall k \in K \tag{2.5.b}$$

The first constraint (2.2) states the binarity condition.

The two Kirchhoff laws are the linearized power flow equations for the existing and candidate networks. The first Kirchhoff law (2.3) expresses the principle of conservation of electric charge. Hence, at any bus, the sum of currents flowing into that bus is equal to the sum of currents flowing out of that bus. As for the second Kirchhoff law, it states the principle of conservation of energy. Thus, the directed sum of the electrical potential differences (voltage) around any closed network must be zero. One should remark that if the candidate circuit k is build then $x_{tk} = 1$ and (2.4.a) is equivalent to (2.4.b). Otherwise, the parameter M_k must be large enough not to impose an active limit over voltage differences. However, too large values for this constant would induce numerical instabilities in practical investments.

Finally, flow limits constraints illustrate the operational limits on the circuits. Here again, one can notice that when we decide to build a given candidate circuit, (2.5.a) is equivalent to the (2.5.b). Otherwise, the flow equals 0, which simply means that the circuit is not built.

We have then presented a simplified deterministic model for the Transmission Expansion Planning problem.

The static version described above is a subproblem of the dynamic version which aim is to determine the stage-by-stage expansion plan. Hence, the constraints given above must now be verified at each stage of the planning horizon. Furthermore, we also consider the following inter-stage constraint :

Interstage constraint

$$x_{t-1,k} \le x_{tk} \quad \forall t \in T \backslash \{1\}, k \in K \tag{2.6}$$

Hence, investing in a circuit at a given time impose to invest in this same circuit for all the following time stages.

As a synthesis, let us give the MIP of the multi-stage deterministic problem.

$$\begin{aligned}
\min_{x} \quad & \sum_{\substack{t \in T \\ k \in K}} c_k x_{tk} & \\
\text{s.t.} \quad & x_{tk} \in \{0,1\} & \forall t \in T, k \in K \\
& x_{t-1,k} \le x_{t,k} & \forall t \in T \backslash \{1\}, k \in K \\
& f_{tj} = \gamma_j \Delta\theta_{tj} & \forall t \in T, j \in J \\
& \left| f_{tk} - \gamma k \Delta\theta_{tk} \right| \le M_k (1 - x_{tk}) & \forall t \in T, k \in K \\
& \left| f_{tj} \right| \le \bar{f}_j & \forall t \in T, j \in J \\
& |f_{tk}| \le \bar{f}_k x_{tk} & \forall t \in T, k \in K \\
& \sum_{l \in K \cup J} A_{il} f_{tl} = d_{ti} - g_{ti} & \forall t \in T, i \in I
\end{aligned} \tag{2.7}$$

2.3 Introducing Uncertainty

In practice, the knowledge of the demand and generation at each time stage is not deterministic. Indeed, many scenarios of demand and generation may occur.

We model the uncertainty by using a **scenario tree**. Each scenario corresponds to a path of realizations of **demand** and **generation** for the full planning horizon. Hence, there are T levels in the tree that correspond to the T decision-making stages.

Scenario trees are one way **to approximate stochastic problems by solving deterministic optimization problems**. Each leaf of the tree corresponds to a scenario, and each node represents a state of the system considered, at a time t, for a realization of stochastic data. We recall the notations for the tree structure :

Tree structure

$\mathcal{N}$	Set of nodes of the scenario tree
$\{\mathbf{1}\}$	Root node of the scenario tree
S	Set of scenarios leaves
$t(n)$	Time stage $t \in T$ corresponding to the node $n \in \mathcal{N}$
$a(n)$	Parent node at stage $t(n)-1$ of the node $n \in \mathcal{N}$. It is unique
$\mathcal{C}(n)$	Set of children nodes of the node $n \in \mathcal{N}$
p_n	Probability associated with the node $n \in \mathcal{N}$
q_{nm}	Probability of transition from the node $n \in \mathcal{N}$ to $m \in \mathcal{C}(n)$. $p_m = p_n * q_{nm}$

We also extend the previous notations by adding the subscript n when the parameter or variable is stochastic. For instance, $d_{n,i}$ and $g_{n,i}$ respectively denotes for the demand and generation for the bus i, at time $t(n)$ and for a realization of data corresponding to the node n. We also denote by $\xi_n = \{(d_{n,i}, g_{n,i})_{i \in I}\}$, the realization of data associated with the node $n \in \mathcal{N}$. In the same way, $x_{n,k}$ is the investment variable for circuit k, at time $t(n)$ and for a realization of data corresponding to the node n. The scenario tree for $T = 3$ and $S = 9$ is presented in the figure (2.2).

We consider that the decision for a stage are taken **after** observing the data. Consequently, the realization of data at the root node is deterministic. Furthermore, it also implies that the decisions are associated with the nodes of the tree.

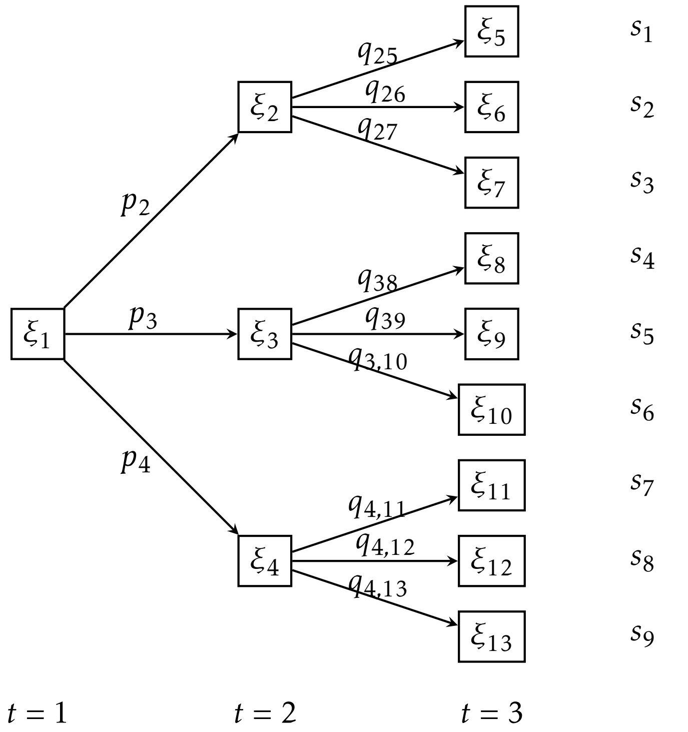

Figure 2.2: Scenario tree of a multistage stochastic program for T=3 and S=9

The MSIP formalizing the Transmission Expansion Planning problem is presented below. The formulation is called the **extensive form**. For the remainder of the thesis, we agree on the convention that : $x_{a(1),k} = 0, \forall k$

$$
\begin{aligned}
\min_{x} \quad & \sum_{\substack{n\in\mathcal{N}\\ k\in K}} p_n c_k x_{n,k} & & \\
\text{s.t.} \quad & x_{n,k} \in \{0,1\} & \forall n \in \mathcal{N}, k \in K & \\
& x_{a(n),k} \le x_{n,k} & \forall n \in \mathcal{N}, k \in K & \\
& f_{n,j} = \gamma_j \Delta\theta_{n,j} & \forall n \in \mathcal{N}, j \in J & \\
& \left| f_{n,k} - \gamma_k \Delta\theta_{n,k} \right| \le M_k(1 - x_{n,k}) & \forall n \in \mathcal{N}, k \in K & \qquad (2.8)\\
& \left| f_{n,j} \right| \le \bar{f}_j & \forall n \in \mathcal{N}, j \in J & \\
& \left| f_{n,k} \right| \le \bar{f}_k x_{n,k} & \forall n \in \mathcal{N}, k \in K & \\
& \sum_{l\in K\cup J} A_{il} f_{n,l} = d_{n,i} - g_{n,i} & \forall n \in \mathcal{N}, i \in I &
\end{aligned}
$$

To conclude this section, we have modeled the power transmission network problem as a MSIP with binary state variables. It has the following crucial properties :

- The objective function in each node is a linear function in the state variables.
- The constraint set is a nonempty compact mixed integer polyhedral set.

Part 3

Nested decomposition algorithm

In this chapter, we present the algorithm that we will use to solve the MSIP. First, we present the dynamic formulation of the extensive form (2.8). Then, we present the Nested Decomposition algorithm as done in [19] and we summarize the main results of the paper relevant for our work. Finally, we present the suitable cuts families.

3.1 Dynamic formulation

The extensive formulation given in (2.8) is a problem which grows exponentially with the number of scenarios S and the number of stages T. We use decomposition techniques to overcome the problem, thanks to Bellman equations of Dynamic Programming that enable us to rewrite (2.8).

For lighter notations, let us introduce the following vectorial notations. From now on, we abusively use the operations between scalars as vectorial operations. Furthermore, the operator $\cdot$ denotes for the term-to-term multiplication between vectors.

Vectorial notations

$$x_n = (x_{n,k})_{k\in K}$$

$$c = (c_k)_{k\in K}$$

$$f_n = (f_{n,l})_{l\in K\cup J}$$

$$\Delta\theta_n = (\Delta\theta_{n,l})_{l\in K\cup J}$$

Besides, for sake of clarity, for all $n \in \mathcal{N}$, we denote by X_n the vector containing all the variables $(x_n, f_n, \Delta\theta_n)$. We can hence denote in an abbreviate form the last five constraints of (2.8). Indeed, we introduce the constraint matrix B_n and the right-hand side b_n defined as it follows.

$\forall n \in \mathcal{N}$,

$$B_n X_n \leq b_n \Leftrightarrow \begin{cases} f_{n,j} = \gamma \cdot \Delta\theta_n \\ \left|f_n - \gamma \cdot \Delta\theta_n\right| \leq M \cdot (1 - x_n) \\ |f_n| \ \leq \bar{f} \\ |f_n| \ \leq \bar{f} x_n \\ A f_n = d_n - g_n & \forall i \in I \end{cases}$$

We leave to the reader the exact formulation of the matrices B_n and vector b_n by reminding that a constraint containing absolute values ($|z| \leq c$) is immediately linearized ($z \leq c$, $-z \leq c$).

The extensive formulation is then :

$$\begin{aligned} \underset{x}{\text{Min}} \quad & \sum_{n \in \mathcal{N}} p_n c^T x_n \\ \text{s.t.} \quad & x_n \in \{0,1\}^K \qquad \forall n \in \mathcal{N} \\ & x_{a(n)} \ \leq x_n \qquad \forall n \in \mathcal{N} \backslash \{\mathbf{1}\} \\ & B_n X_n \leq b_n \qquad \forall n \in \mathcal{N} \end{aligned} \tag{3.1}$$

Which is equivalent to the following DP equation :
For each node $n \in \mathcal{N}$,

$$\begin{aligned} Q_n(x_{a(n)}) = \underset{x_n}{\text{Min}} \ & c^T x_n + \sum_{m \in \mathcal{C}(n)} q_{nm} Q_m(x_n) \\ \text{s.t.} \quad & x_n \in \{0,1\}^K \\ & x_{a(n)} \ \leq x_n \\ & B_n X_n \leq b_n \end{aligned} \tag{3.2}$$

$Q_n(\cdot)$ is the optimal value function at node n. It maps to each possible state $x_{a(n)}$ of the parent node the optimal cost. We also denote the function $\mathcal{Q}_n(\cdot) = \sum_{m \in \mathcal{C}(n)} q_{nm} Q_m(\cdot)$ as the **expected cost-to-go** function at node n.

The dynamic formulation of the problem drastically reduces the size of the initial MSIP. Nonetheless, the exact computation of the expected cost-to-go function is difficult and requires a discretization of the space of states that induces the curse of dimensionality. That is why we introduce the Nested Decomposition algorithm (ND) based on [19].

3.2 Presentation of the ND algorithm

3.2.1 Idea of the algorithm

In the reference paper [19], the authors propose to solve the DP recursive equations and (3.2) by iteratively strengthening a convex piece-wise lower approximation of the expected cost-to-go function $\mathcal{Q}_n(\cdot)$. The key of success of their algorithm lies in two notions :

(i) making a local copy of the state variables

(ii) strengthening the tightness of the lower approximation of the value function

The first point is achieved by introducing an immediate reformulation of (3.2) that consists in making a local copy of the state variables. Hence, for each node $n \in \mathcal{N}$, we introduce an **auxiliary variable** $z_{n,k}$ that is equal to the parent node state variable $x_{a(n),k}$. The interest of such a mathematical trick is the fact that $z_{n,k}$ is not necessarily an integer variable. It will enable to develop valid and tight inequalities to approximate the value function, achieving then our second point (ii).

Let us introduce the local copies of the state variable and write the DP resulting equations that are considered for the remainder of the thesis. For each node $n \in \mathcal{N}$,

$$\begin{aligned} Q_n(x_{a(n)}) = \underset{x_n}{\text{Min}} \;\; & c^T x_n + \sum_{m \in \mathcal{C}(n)} q_{nm} Q_m(x_n) \\ \text{s.t.} \;\; & x_n \in \{0,1\}^K \\ & z_n \in [0,1]^K \\ & z_n \leq x_n \\ & z_n = x_{a(n)} \\ & B_n X_n \leq b_n \end{aligned} \tag{3.3}$$

3.2.2 Outline

Each iteration p of the algorithm splits into a forward and a backward steps. In the forward step, we proceed stage-wise from $t = 1$ to T by solving, at each node, a DP equation with an approximate expected cost-to-go function. At the end of this step, we store the investment decision variables and compute a feasible upper-bound of the problem as the weighted average overall scenarios of the total investment cost. In the backward step, we proceed stage-wise from the last stage T and solve at each node a suitable auxiliary problem of the forward problem. We deduce then an inequality that lower approximate the true value function. This enables to strengthen the approximation we use in the forward phase. Finally, the problem solved at the root node provides a lower-bound of the problem. The algorithm stops when the upper and lower bound are close enough, according to a convergence criteria. Let us detail these steps.

Forward step

At iteration i, we go through the scenario-tree stage-wise, from $t = 1$ to T, and we solve for each node of the t-th level, the following approximation of (3.3).

For each node $n \in \mathcal{N}$,

$$\begin{aligned} (P_n^i(x_{a(n)}^i, \psi_n^i)): \quad \underline{Q}_n(x_{a(n)}^i) := \underset{x_n}{\text{Min}} \;\; & c^T x_n + \psi_n^i(x_n) \\ \text{s.t.} \;\; & x_n \in \{0,1\}^K \\ & z_n \in [0,1]^K \\ & z_n \leq x_n \\ & z_n = x_{a(n)}^i \\ & B_n X_n \leq b_n \end{aligned} \tag{3.4}$$

Let us define the approximation of the expected cost-to-go function $\psi_n^i(x_n)$. It is actually approximated by a set of lower inequalities corresponding to cutting-planes. Each cut is hence defined by its cuts coefficients $\{v_m^i, \pi_m^i\}$. One can consider v_m^i as the intercept of the cut while π_m^i is its slope. Hence, $\psi_n^i(x_n)$ is computed as :

$\forall n \in \mathcal{N}$:

$$\psi_n^i(x_n) := \text{Min } \{\alpha_n : \alpha_n \geq \sum_{m \in \mathcal{C}(n)} q_{nm}(v_m^l + (\pi_m^l)^T x_n), \forall l = 1, \ldots, i-1\} \tag{3.5}$$

The forward problem for the node n is hence characterized by the investment decision $x_{a(n)}^i$ obtained by solving the parent node problem $(P_{a(n)}^i)$ at the stage $t(n)-1$. It is also characterized by the approximation of the expected cost-to-go function that is embodied in the cuts (3.5). The computation of the cuts coefficients $\{v_m^i, \pi_m^i\}$ is done in the backward step.

Once we solve (3.4)-(3.5), the optimal solution x_n^i is stored and passed on to the forward problem $(P_m^i(x_n^i, \psi_m^i))_{m \in \mathcal{C}(n)}$. Hence, the output of the forward phase is a state variable solution x_n^i for each node $n \in \mathcal{N}$.

Backward step

After solving all the forward problems, we proceed stage-wise through the scenario tree, from $t = T$ to 2. At node $n \in \mathcal{N}$, we solve an auxiliary problem (R_n^i) to the updated problem $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$. This enables to collect the coefficients $\{v_n, \pi_n\}$ for the cuts that we add to the strengthen the approximation of $\psi_{a(n)}^{i+1}$. We proceed then iteratively until we reach $t = 2$.

The exact formulation of the auxiliary problem (R_n^i) is still to precise. It will depend on the **nature of the cut** that we want to compute and will be specified in the following section.

Convergence criterion

The forward phase enables to compute a feasible upper-bound at iteration i, denoted UB^i and equals to $\sum_{n \in \mathcal{N}} p_n c^T x_n^i$. On the other hand, the optimal value of the problem $(P_1^i(\psi_1^i))$ provides the lower bound LB^i of the true optimal value.
The algorithm stops when the gap $\frac{|UB^i - LB^i|}{UB^i}$ is lower than a convergence threshold ϵ.

3.2.3 Summary

As a synthesis, let us summarize the main steps of ND algorithm applied to our problem.

Algorithm 1: Nested Decomposition algorithm

Data: convergence threshold ϵ

Initialization : $i = 0$, $UB^0 = 10^6$, $LB^0 = -10^6$

Result: $(x_n)_{n\in\mathcal{N}}$ investment optimal solution for each node of the tree

while $\frac{|UB^i - LB^i|}{UB^i} > \epsilon$ **do**

  $i \leftarrow i + 1$

  /* Forward step :

  forall $t = 1 \dots T$ **do**

    forall $n \in \mathcal{N} : t(n) == t$ **do**

      Solve $(P_n^i(x_{a(n)}^i, \psi_n^i))$

      collect investment solutions x_n^i

      collect ψ_1^i, approximation of the cost-to-go function at the root node

    end

  end

  /* Backward step :

  forall $t = T \dots 2$ **do**

    forall $n \in \mathcal{N} : t(n) == t$ **do**

      Solve (R_n^i) auxiliary to $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$

      collect cuts coefficients $\{v_n^i, \pi_n^i\}$ to compute $\psi_{a(n)}^{i+1}$

    end

  end

  /* Lower and Upper bounds update :

  $LB^i = c^T x_1^i + \psi_1^i$

  $UB^i = \sum_{n\in\mathcal{N}} p_n c^T x_n^i$

end

return $(x_n)_{n\in\mathcal{N}}$

3.3 Cut families

The algorithm presented above is identical to the standard Nested Benders Decomposition algorithm that we use for solving MSLP, as in [6], at the difference that in the latter case, benders cuts enable to reach convergence for the algorithm. It is not the case anymore when MSIP are considered. The main contribution of [19] is to introduce sufficient cut conditions by introducing the notion of **valid, tight and finite cuts**. Besides, the authors have proven that the ND algorithm **admits finite convergence to an optimal solution when these three conditions are met**. One can report to [19] for the detail and proof of this statement.

However, the authors considered four cut families, namely Benders' cuts, Integer Optimality cuts, Lagrangian cuts and Strengthened Benders cuts. Let us present each of the cutting plane families by giving the explicit form of the auxiliary problem (R_n^i) and the cuts coefficients $\{v_n^i, \pi_n^i\}$.

3.3.1 Benders' cuts

This well-known family of cuts has first been introduced in [2]. At iteration i of the algorithm, (R_n^i) corresponds to the LP relaxation of $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$. Then, the coefficient π_n^i is equal to the dual multiplier associated to the constraint $z_n = x_{a(n)}^i$.

Also, let us denote by $Q_n^{LP}(x_{a(n)}^i)$ the optimal value of (R_n^i). The i-th Bender's cut added to the parent node $a(n)$ is then :

$$\alpha_{a(n)} \geq \sum_{m \in \mathcal{C}(a(n))} q_{a(n)m} \; (\; Q_m^{LP}(x_{a(n)}^i) + (\pi_m^i)^T (x_{a(n)} - x_{a(n)}^i)$$

To recover the formalism (3.5), we hence have $v_n^i = Q_n^{LP}(x_{a(n)}^i) - \; \pi_n^i x_{a(n)}^i$

Benders' cut are valid, finite but they are not tight when applied to MSIP. Hence, using this cut's family is not guaranteed to lead to an optimal solution.

3.3.2 Integer Optimality cuts

Integer Optimality cuts have first been introduced in [10] to solve two-stage stochastic programs with binary first stage variables. They have been adapted to ND algorithm in [19]. For this cut family, (R_n^i) corresponds to the problem $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$ itself. Let v_n^i be the optimal objective value of (R_n^i). Then, the i-th integer optimality cut added to the parent node $a(n)$ is :

$$\alpha_{a(n)} \geq \sum_{m \in \mathcal{C}(a(n))} q_{a(n)m} v_m^i \; (1 + \sum_{k \in K} (x_{a(n),k}^i - 1) x_{a(n),k} + \sum_{k \in K} (x_{a(n),k} - 1) x_{a(n),k}^i)$$

These cuts are valid, finite and tight since they are an exact approach for solving MSIP with binary state variables. Nonetheless, they can be very loose at other solutions, and only tight at the considered solution. That is why they are not computationally efficient.

3.3.3 Lagrangian cuts

The authors of [19] have introduced Lagrangian cut family and have proven it to be valid, finite and tight. Their work was leveraged on the PhD thesis of Fernanda Thomé [15], which showed how to get the most binding convex cut by a customized Lagrangian relaxation.

The main idea of the Lagrangian cut is to consider to nodal problem at iteration i of the algorithm, $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$ and solve its Lagrangian dual (R_n^i) by relaxing the constraint $z_n = x_{a(n)}^i$.

More precisely,

$$(R_n^i) : \underset{\pi_n}{\text{Max}} \;\; \mathcal{L}_n^i(\pi_n) + \pi_n^T x_{a(n)}^i \qquad (3.6)$$

Where the Lagrangian $\mathcal{L}_n^i(\pi_n)$ is defined as it follows :

$$\begin{aligned} \mathcal{L}_n^i(\pi_n) = \underset{x_n}{\text{Min}} \quad & c^T x_n + \alpha_n - \pi_n^T z_n \\ \text{s.t.} \quad & x_n \in \{0,1\}^K \\ & z_n \in [0,1]^K \\ & z_n \leq x_n \\ & B_n X_n \leq b_n \\ & \alpha_n \geq \sum_{m \in \mathcal{C}(n)} q_{nm}(v_m^l + (\pi_m^l)^T x_n), \forall l = 1, \ldots, i \end{aligned} \tag{3.7}$$

The coefficients $\{v_n^i, \pi_n^i\}$ are then respectively computed as the optimal objective cost of the Lagrangian problem (3.7) and optimal dual solution of (3.6). In [16], the authors propose an iterative scheme to Lagrange multipliers optimization.

The collection of Lagrangian cuts $\{v_n^i, \pi_n^i\}_{n \in \mathcal{N}}$ is valid, thigh and finite (one can refer to [19] for the proof). Hence, they theoretically ensure convergence. However, the computation of the Lagrangian might be difficult.

3.3.4 Strengthened Benders' cuts

This cuts family has been deduced from Benders' and Lagrangian cuts. First, one can remark that for any fixed π_n, solving the Lagrangian (3.7) leads to a **valid cut**. Obviously, if we would have an idea of the optimal π_n, we could hence avoid the solve (3.6). The key fact is that we can access to π_n by making the same reasoning as for Benders' cut. The idea is then to strengthen Benders' cuts by solving an additive nodal problem corresponding to the the Lagrangian relaxation (3.7).

Concretely, in the i-th backward phase, at the node n, one should proceed as follows :

- Solve a LP relaxation of $(P_n^i(x_{a(n)}^i, \psi_n^{i+1}))$.
- Store the coefficient π_n^i as the dual multiplier associated to the constraint $z_n = x_{a(n)}^i$.
- Solve the Lagrangian relaxation (3.7) with fixing π_n to π_n^i.
- Store the coefficient v_n^i as the optimal objective cost of the previous problem : $v_n^i = \mathcal{L}_n^i(\pi_n)$.

We can then add the i-th cut to the $a(n)$ nodal problem :

$$\alpha_{a(n)} \geq \sum_{m \in \mathcal{C}(a(n))} q_{a(n)m} \ (v_m^i + (\pi_m^i)^T x_{a(n)})$$

One can easily remark that this cut is as least as tight as Benders cut. Indeed, the slopes of both Benders and strengthened Benders (SB) cuts are equal. Furthermore,

$$\forall m \in \mathcal{C}(a(n)), \quad v_m^i = \mathcal{L}_m^i(\pi_m) \geq Q_m^{LP}(x_{a(n)}^i) - (\pi_m^i)^T x_{a(n)}^i$$

And we recall that $Q_m^{LP}(x_{a(n)}^i) - (\pi_m^i)^T x_{a(n)}^i$ is the intercept of benders' cut. Q.E.D.

Nonetheless, even though strength Benders cuts strictly improves Benders cut, they are not tight in the sense of the [19] and hence, do no ensure the convergence of the ND algorithm. However, in the computation experiments conducted in [19], the authors point out that the (SB) cuts perform very well in practice, even though they do not ensure theoretical convergence.

In the following table, we summarize the convergence properties of the four cuts considered. In all the remainder of the thesis, (B), (I), (L) and (SB) respectively denotes for Benders', Integer Optimality, Lagrangian and strength Benders' cuts.

Cut family	(B)	(I)	(L)	(SB)
Valid ?	Yes	Yes	Yes	Yes
Finite?	Yes	Yes	Yes	Yes
Tight?	**No**	Yes	Yes	**No**

Table 3.1: Convergence properties of the cut families

Part 4

Computational experiments

We have implemented the ND algorithm applied to the Network Transmission Planning problem in Julia with GLPK solver for MIP and LP. All computations are conduced on a Windows 7 Professional desktop with four 1.60GHz processors and 8GB RAM.

Let us present first the test cases we have considered. Both of them are well-known in the literature. Then, we will precise the methodology we have followed for the tests. Finally, we will give our results and conclude.

4.1 Presentation of the test cases

4.1.1 Garver 6-bus system

This system was originally used in [8], and since then has become the most popular test system in transmission expansion planning. This system has 6 buses, 6 existing circuits and 60 candidate circuits. In the literature, the system is one stage and there is one scenario of demand for the first and only stage, that is equal to 760 MW. Since we study multistage problems, we have added 2 other stages, by increasing the demand to 815 MW for $t = 2$ and 845 MW for $t = 3$.

The relevant data are given in Annex (6.1). The initial topology is shown in Fig. (2.1) at the beginning of the thesis.

4.1.2 IEEE 24-bus system

The IEEE 24-bus reliability test system was developed by the IEEE reliability subcommittee and published in 1979 as a benchmark for testing various reliability analysis methods. This system has 24 buses, 38 existent circuits and 78 candidate circuits. In the literature the system is one stage and there are 4 scenarios of demand. For all the scenarios, the overall demand for the first time stage is equal to 8550 MW. Here again, we have added

2 other stages by increasing the demand to 8746 MW for $t = 2$ and to 8846 MW for $t = 3$. The relevant data are given in Annex (6.3).

4.2 Methodology

The tests aimed at validating the ND decomposition algorithm for solving the multistage transmission network problem. First, we consider **the deterministic case where the tree is reduced to one single path**. This enables to evaluate the ND algorithm by **comparing it to the extensive form**. In effect, we recall that one can also solve the MIP without decomposition, as expressed in the (2.8). The important issue of this method is the fact that it is not scalable. Nonetheless, testing one single path lead to relatively small MIPs that can still be solved to optimality by our solver.

Besides, the second Kirchhoff law can bring about computational difficulties because of the big M parameter. That is why we first validate the ND algorithm **without the 2nd Kirchhoff law**.

In a second phase, we re-introduce the 2nd Kirchhoff law and discuss the computation of the big M parameters.

Finally, we test the ND algorithm on the tree and compare, when it is possible, to the extensive form.

Concerning the parameters of the ND algorithm, we have set the optimality gap $\epsilon = 10^{-5}$ and the maximal number of iterations $I_{max} = 500$. We have chosen to restrict ourselves to Benders', Integer Optimality and Strength Benders' cuts. In effect, the numerical experiments conducted by the authors of [19] have shown that (SB) cuts were computationally efficient in practice, despite of not ensuring theoretical convergence. That is why we decided to consider (L) cuts only if the other cuts would not be sufficient.

As a summary, we will follow these steps:

- Compare the ND algorithm to the extensive form, by running Garver and IEEE systems on one single scenario for a multistage planning horizon, and without the 2nd Kirchhoff law. It is the deterministic case. We discuss the difference between cut families.
- Introduce the 2nd Kirchhoff law. It is the deterministic case with 2nd Kirchhoff law. We confirm the difference between cut families.
- Solve on a tree of 4 and 9 leaves. It is the stochastic case.

4.3 Deterministic case without 2nd Kirchhoff law

4.3.1 Garver system

We remove the 2nd Kirchhoff law and consider the Garver system over a planning horizon of 2 and 3 stages. On the one hand, we solve the model without decomposition (i.e the MIP of the extensive form (2.8)). This enables to have the reference optimal values and computational times presented in Table (4.1).

	Optimal cost	Time (in s)
T = 2	400	18
T = 3	630	17

Table 4.1: Garver system without 2nd Kirchhoff law, in the deterministic case, in the extensive form.

On the other hand, we run the ND algorithm on the same instances, with a single class of cutting planes ((B),(I) or (SB)). We obtain the results in Table (4.2). For each of the two cases, we tell whether the algorithm converged or not, and if yes, we give the optimal cost. We also give the number of iterations before convergence (500 being the maximal number of iterations), the solving time in seconds and the relative gap between the upper and lower bounds.

		(B)	(I)	(SB)
T = 2	Convergence?	**Yes**	No	**Yes**
	Optimal value	400	-	400
	Nb Iterations	3	500	2
	Time (in s)	20	1358	17
	Relative Gap	0%	33%	0%
T = 3	Convergence?	No	No	**Yes**
	Optimal value	-	-	630
	Nb Iterations	500	500	3
	Time (in s)	137	1147	19
	Relative Gap	3%	57%	0%

Table 4.2: Garver system without 2nd Kirchhoff law, in the deterministic case, with ND algorithm.

We can make the following remarks :

- With (SB) cuts, the ND algorithm converges for the two tests in a few number of iterations towards the optimal value given when solving the extensive form.
- With (B) cuts, the ND algorithm converges for $T = 2$ towards the optimal solution. Nonetheless, it does not converges for $T = 3$ and the relative gap reaches 3% at the 5th iteration and stays constant.
- With (I) cuts, the ND algorithm does not converge within 500 iterations for the two tests. However, the gap keeps decreasing relatively slowly.

In order to validate that (I) cuts ensure convergence, we have implemented a hot start : we use (B) cuts for the 2 first iterations, then we use (I) cuts. The algorithm converged in that case in 11 iterations for T=2. This has enabled to validate Integer Cuts.

Figure (4.1) illustrates the comparison of (B) and (SB) cuts. Hence, it depicts the

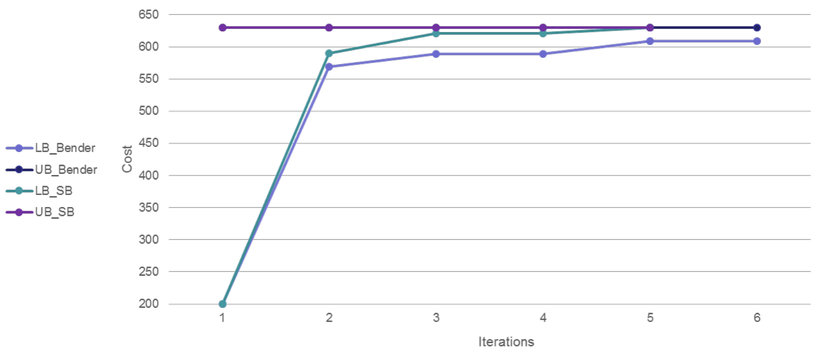

Figure 4.1: Comparison of (B) and (SB) cuts on Garver system with $T = 3$

evolution of the lower and upper bounds for the system when T =3. With (SB) cuts, the algorithm converges in 5 iterations. With (B), it does not converge and the lower bound stays constant from the 5th iteration.

From these first results, **we can already confirm one of the conclusions of [19]**. In effect, (SB) cuts turned out to be computationally efficient in contrast to (B) and (I) cuts. Also, (I) cuts perform poorly in practice, even thought they ensure convergence.

4.3.2 IEEE system

We consider now the IEEE 24-bus system. Since the data set contained 4 scenarios of demand and generation, we tested the ND algorithm on each of 4 instances, with $T = 2$ and $T = 3$. For each of the 8 runs, we have also solved the MIP of the extensive form, in order to give the reference optimal cost.

		Optimal cost	Time (in s)
Scenario 1	T = 2	512	17
	T = 3	768	30
Scenario 2	T = 2	588	29
	T = 3	882	23
Scenario 3	T = 2	404	17
	T = 3	606	15
Scenario 4	T = 2	468	30
	T = 3	714	72

Table 4.3: IEEE system without 2nd Kirchhoff law, in the deterministic case, in the extensive form for 4 independent scenarios.

			(B)	(I)	(SB)
Scenario 1	T = 2	Convergence ?	**Yes**	No	**Yes**
		Optimal value	512	-	512
		Nb Iterations	13	500	13
		Time	103s	1h12	104s
		Relative Gap	0%	41%	0%
	T = 3	Convergence ?	**Yes**	No	**Yes**
		Optimal value	768	-	768
		Nb Iterations	27	500	27
		Time	6min	1h05	7min
		Relative Gap	0%	61%	0%
Scenario 2	T = 2	Convergence ?	**Yes**	No	**Yes**
		Optimal value	588	-	588
		Nb Iterations	22	500	22
		Time	48min	7h	1h
		Relative Gap	0%	42%	0%
	T = 3	Convergence ?	**Yes**	No	**Yes**
		Optimal value	882	-	882
		Nb Iterations	38	500	39
		Time	5h	6h30	5h11
		Relative Gap	0%	61%	0%
Scenario 3	T = 2	Convergence ?	**Yes**	No	**Yes**
		Optimal value	404	-	404
		Nb Iterations	20	500	20
		Time	63s	1h15	63s
		Relative Gap	0%	39%	0%
	T = 3	Convergence ?	**Yes**	No	**Yes**
		Optimal value	606	-	606
		Nb Iterations	33	500	33
		Time	95s	58min	137s
		Relative Gap	0%	60%	0%
Scenario 4	T = 2	Convergence ?	No	No	**Yes**
		Optimal value	-	-	468
		Nb Iterations	500	500	63
		Time	54min	1h30	5min20s
		Relative Gap	3%	46%	0%
	T = 3	Convergence ?	No	No	**Yes**
		Optimal value	-	-	714
		Nb Iterations	500	500	77
		Time	1h11	2h	8min30
		Relative Gap	4%	65%	0%

Table 4.4: IEEE system without 2nd Kirchhoff law, in the deterministic case, with ND algorithm for 4 independent scenarios.

The same conclusions as for Garver instance are valid:

- With (SB) cuts, the ND algorithm always converges towards the optimal cost given by the extensive form solving.
- With (B) cuts, it does not always converge.
- As for (I) cuts, it never converges within the 500 iterations.

Figure (4.2) below represents the evolution of upper and lower bounds for the scenario 4 and for T = 2. It is a case where the ND algorithm does not converge with (B) cuts and converges in 63 iterations with (SB) cuts.

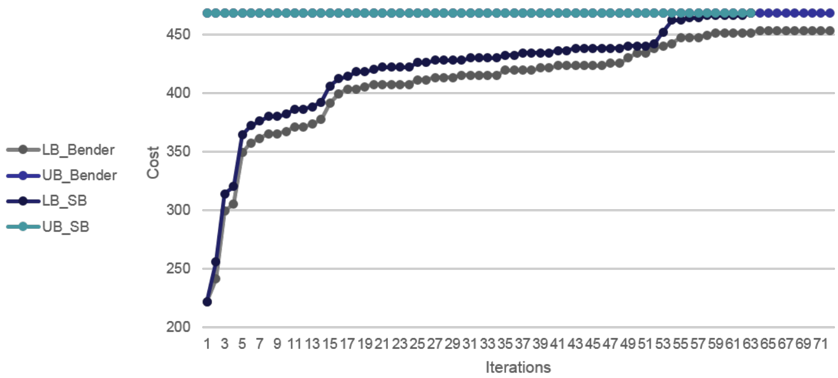

Figure 4.2: Comparison of (B) and (SB) cuts on IEEE system with $T = 2$

4.3.3 Combination of cuts

Since (I) cuts perform very poorly and ensure theoretical convergence while (SB) cuts perform well without ensuring convergence, a good track was to combine both of these cuts. The idea is then to use (SB) cuts as a hot start : we apply the ND algorithm with (SB) cuts, and then we **activate (I)** cuts only when the gap between upper and lower bound is relatively small. Note that (SB) are active throughout all the optimization phase.

Adding (I) cuts for our instances is not relevant since ND already converge with (SB) cuts. Nonetheless, we have tried different iterations for which we activate (I) cuts. For instance, we consider the scenario 3 of the IEEE case with $T = 3$. In this example, the (SB) cuts make the ND algorithm converge in 33 iterations.

The table below summarizes the convergence, computation time and iterations, with (SB) cuts when we try several iterations to activate the (I) cuts. The columns represent the iterations for which we activate the integer cuts. We compare them to the case where we have only (SB) cuts (last column).

	0	10	15	20	25	30	Only (SB)
Convergence ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Nb Iterations	34	35	34	34	33	33	33
Time (in s)	243	206	176	186	172	178	137
Relative Gap	0%	0%	0%	0%	0%	0%	0%

Table 4.5: IEEE system without 2nd Kirchhoff law for scenario 3 and $T = 3$, with ND algorithm, by combining (SB) and (I) cuts.

Hence, activating the (I) cuts too soon (at 0th, 10th, 15th or 20th iteration) does not improve the convergence. When it is done at iteration 25 and 30, the convergence takes 33 iterations but more time, since adding a cut is time consuming (one additive MIP to solve at each iteration). It is impossible to draw the pattern of a general rule from this test, however, one can argue that this combination of cuts might be useful to give us the warranty of convergence.

To conclude this section, we have been able to confirm the results of [19] applied to Network Transmission planning problem, by testing our implementation of ND algorithm on several instances. (SB) cuts perform well and enable to reach optimality, (B) cuts do not ensure convergence and (I) cuts have a very slow convergence rate. We have finally implemented the combination of (SB) and (I) cuts and made sensitivity analysis about it.

4.4 Introduction of 2nd Kirchhoff law

4.4.1 Discussion on the big M

The difficulty of 2nd Kirchhoff law lies in the computation of the parameters $(M_k)_{k\in K}$. A balance must be struck to compute large values but tight as possible (too large values would induce numerical instabilities while too low values would introduce explicit bounds over operational variables). In [3], Binato derived the minimum values associated with each circuit.

Let us consider a circuit k that links the bus i_{fr} to the bus i_{to}.

- If there is already an existing circuit j that links i_{fr} to i_{to}, the 2nd Kirchhoff law constraint of circuit k can be written when it is active (i.e when $x_k = 0$) :

$$|\gamma_k \Delta\theta_{n,k}| \leq M_k \tag{4.1}$$

  Also, for the j-th existing circuit, we have : $f_{n,j} = \gamma_j \Delta\theta_{n,j} = \gamma_j \Delta\theta_{n,k}$.
  Since $f_{n,j} \leq \bar{f}_j$, we have : $\gamma_j \Delta\theta_{n,k} \leq \bar{f}_j$ or $\Delta\theta_{n,k} \leq \frac{\bar{f}_j}{\gamma_j}$.

Hence, based on (4.1), we set M_k as it follows :

$$M_k = \gamma_k \frac{\bar{f}_j}{\gamma_j} \tag{4.2}$$

We extend the results to the case where there is more than one existing circuit that links the bus i_{fr} to the bus i_{to}. If J_k is the set formed by these circuits, then we logically compute M_k as :

$$M_k = \text{Min } \{\gamma_k \frac{\bar{f}_j}{\gamma_j} : j \in J_k\} \tag{4.3}$$

- If there is no existing circuit linking the buses i_{fr} and i_{to}, the parameter M_k can be determined by solving the shortest path problem that computes the shortest distance D_k^{min} that links i_{fr} to i_{to} by using only the existent network. This problem can be solved using Dijkstra algorithm. Then, we have $M_k = \gamma_k D_k^{min}$.

Since it is not the core issue of our work, we have preferred to restrict ourselves to the first case. In order to avoid the solving of shortest-path problems, we have then **modified Garver and IEEE existing networks**. Hence, all the candidate circuits have at least one existing equivalent circuit. By doing so, we have augmented the installed capacity of the two systems, which led us to also augment the total demand and generation. Relevant data of the modified network are given in Annexes (6.2) and (6.4).

4.4.2 Garver system

We consider the modified Garver system over a planning horizon of 2 and 3 stages, with activating the 2nd Kirchhoff law. On the one hand, we solve the extensive form :

	Optimal cost	Time (in s)
T = 2	151	25s
T = 3	272	27s

Table 4.6: Garver system with 2nd Kirchhoff law, in the deterministic case, in the extensive form.

Then, we run ND with (B) and (SB) cuts, and with the combination (SB) + (I)

		(B)	(SB)	(SB)+(I)
T = 2	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	151	151
	Nb Iterations	500	5	5
	Time (in s)	306	23	24
	Relative Gap	13%	0%	0%
T = 3	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	272	272

Nb Iterations	500	12	12
Time (in s)	499	29	31
Relative Gap	30%	0%	0%

Table 4.7: Garver system with 2nd Kirchhoff law, in the deterministic case, with ND algorithm.

4.4.3 IEEE system

We apply the same tests to the modified IEEE system. Since we modified the data, we have chosen to focus on only one scenario, namely scenario 3. We also give computation times in order to be able to compare to the extensive form.

	Optimal cost	Time (in s)
T = 2	272	38
T = 3	432	436

Table 4.8: IEEE system with 2nd Kirchhoff law, in the deterministic case, in the extensive form.

		(B)	(SB)	(SB)+(I)
T = 2	Convergence?	**Yes**	**Yes**	**Yes**
	Optimal value	272	272	272
	Nb Iterations	5	5	5
	Time (in s)	44	44	74
	Relative Gap	0%	0%	0%
T = 3	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	432	432
	Nb Iterations	500	7	7
	Time (in s)	2100	63	101
	Relative Gap	5%	0%	0%

Table 4.9: IEEE system with 2nd Kirchhoff law, in the deterministic case, with ND algorithm.

Hence adding second Kirchhoff law has not changed the precedent conclusions. For the last test, we have also presented the computation time for solving the extensive form and compared it to the computation time of the ND algorithm. On this instance, we can observe that the time taken for the instance $T = 3$ considerably decreases when we use the decomposition algorithm (436 s without decomposition VS 63 s with ND and (SB) cuts) and is equivalent between the two solving methods for instance $T = 2$

4.5 Solving on a tree

Since the ND algorithm has been validated on the deterministic case, we consider now the scenario trees.

4.5.1 Garver system

Since the original data set contained one scenario of demand and generation, we had to create other scenarios for demand and generation. We consider the case where $T = 3$ and create two instances of tests, with $S = 4$ (2 scenarios by stage) and $S = 9$ (3 scenarios by stage).

Here again, we would like to compare the values obtained by the ND algorithm to the results obtained by solving the extensive form. For the tree with $S = 4$, the extensive form corresponds to a MIP of 720 binary variables, which presented a problem to the solver. In effect, **the instance has not been solved**. We verified that this claim did not depend on the capacity of the computer by testing on a more powerful computer with height 2.40 GHz processors and 32 GB RAM.

Nonetheless, the solver gave results for the case $S = 4$, when removing the second Kirchhoff law. We will then use this case as validation case.

4.5.1.1 Without 2nd Kirchhoff law

As presented in the tables below, when we consider a tree of 4 scenarios, the ND algorithm converges towards the optimal solution and is very fast in comparison to the model without decomposition.

	Optimal cost	Time (in s)
S = 4, T = 3	150	1437

Table 4.10: Garver system without 2nd Kirchhoff law, for a tree of 4 scenarios, in the extensive form.

	Optimal cost	Time (in s)	Nb Iterations	Relative gap
S = 4, T = 3	150	23	6	0 %

Table 4.11: Garver system without 2nd Kirchhoff law, for a tree of 4 scenarios, with ND algorithm with (SB) cuts.

4.5.1.2 With 2nd Kirchhoff law

We extend our tests by reintroducing the 2nd Kirchhoff law. Since the extensive form do not work anymore, we present here the performance of the ND algorithm with (B) and

(SB) cuts, and with the combination (SB) +(I).

Here again the results are encouraging, the ND algorithm converges few iterations towards the optimal solution with (SB) cuts, for cases that **are not computable without decomposition**.

		(B)	(SB)	(SB)+(I)
S = 4	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	259	259
	Nb Iterations	500	10	10
	Time (in s)	1380	45	51
	Relative Gap	23%	0%	0%
S = 9	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	276	276
	Nb Iterations	500	11	11
	Time (in s)	4800	101	109
	Relative Gap	26%	0%	0%

Table 4.12: Garver system with 2nd Kirchhoff law, in the scenario tree case, with ND algorithm.

4.5.2 IEEE system

We create two instances of tests for IEEE system, with $S = 4$ and $S = 9$ and present the results in the following tables :

		(B)	(SB)	(SB)+(I)
S = 4	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	420	420
	Nb Iterations	500	7	7
	Time (in s)	8358	75	112
	Relative Gap	2.9%	0%	0%
S = 9	Convergence?	No	**Yes**	**Yes**
	Optimal value	-	476	476
	Nb Iterations	500	8	8
	Time (in s)	4157	151	145
	Relative Gap	13%	0%	0%

Table 4.13: IEEE system with 2nd Kirchhoff law, in the scenario tree case, with ND algorithm.

Part 5

Stochastic setting : Introducing uncertainty at each node

So far, we have assumed that the knowledge of the uncertain data was deterministic at each node. In other words, when we are at the node $n \in \mathcal{N}$, we know exactly the demand and generation for the current time $t(n)$. Adding more scenarios for this demand and generation only requires to add nodes to the tree. Nonetheless, when the scenario tree is large, it might be computationally difficult to solve the problem using ND algorithm.

In this section, we use prior literature on the static Transmission Expansion Planning problem to model **uncertainty at each node of the tree**. We consider that each node is associated with a set of many realizations for the current time stage. In the first section, we introduce some notations and formalize the problem to solve at each node $n \in \mathcal{N}$. Then, we present the master-slave decomposition algorithm and embed it the ND algorithm with strength benders cuts. Finally, we present our computational experiments on Garver and IEEE systems.

5.1 Formulation of the problem at node n

Let us introduce some additional notations :

$\tilde{S}_n$ Set of sub-scenarios, or possible realizations of the stochastic process for the node $n \in \mathcal{N}$.

$\tilde{d}_{n,s}$ Realization of the demand for $n \in \mathcal{N}$ and sub-scenario $s \in \tilde{S}_n$.

$\tilde{g}_{n,s}$ Realization of the generation for $n \in \mathcal{N}$ and sub-scenario $s \in \tilde{S}_n$.

$\tilde{f}_{n,s}$ Algebraic power flow by circuit for $n \in \mathcal{N}$ and sub-scenario $s \in \tilde{S}_n$.

$\tilde{\Delta}\theta_{n,s}$ Difference between voltage angles by bus, for $n \in \mathcal{N}$ and sub-scenario $s \in \tilde{S}_n$.

Also, let us denote by $s = \{1\} \in \tilde{S}_n$, the base scenario corresponding to the data $\{d_n, g_n\}$.

Finally, we introduce for a given $n \in \mathcal{N}$ and $s \in \tilde{S}_n$ the aggregated vector of variables :

$$\tilde{X}_{n,s} = \begin{bmatrix} x_n \\ \tilde{f}_{n,s} \\ \Delta\tilde{\theta}_{n,s} \end{bmatrix}$$

Hence, at node n, one must solve the investment problem (3.4) **with verifying that the investment plan is feasible for the sub-scenarios** $s \in \tilde{S}_n$. The MIP at the node n and at iteration i of the ND algorithm is then :

$$\begin{aligned}
(P_n^i(x_{a(n)}^i, \psi_n^i)) : \quad \underline{Q}_n(x_{a(n)}^i) := \underset{x_n}{\text{Min}} \quad & c^T x_n + \psi_n^i(x_n) \\
\text{s.t.} \quad & x_n \in \{0,1\}^K \\
& z_n \in [0,1]^K \\
& z_n \leq x_n \\
& z_n = x_{a(n)}^i \\
& \psi_n^i(x_n) \geq \sum_{m \in \mathcal{C}(n)} q_{nm}(v_m^l + (\pi_m^l)^T x_n), \forall l < i \\
& B_n \tilde{X}_{n,s} \leq \tilde{b}_{n,s}, \qquad \forall s \in \tilde{S}_n
\end{aligned} \tag{5.1}$$

Let us detail the last constraint :

$$B_n \tilde{X}_{n,s} \leq \tilde{b}_{n,s},\ \forall s \in \tilde{S}_n \Leftrightarrow \begin{cases} \tilde{f}_{n,s} = \gamma \cdot \Delta\tilde{\theta}_{n,j} & \forall s \in \tilde{S}_n \\ \left|\tilde{f}_{n,s} - \gamma \cdot \Delta\tilde{\theta}_n\right| \leq M \cdot (1 - x_n) & \forall s \in \tilde{S}_n \\ \left|\tilde{f}_{n,s}\right| \quad \leq \bar{f} & \forall s \in \tilde{S}_n \\ \left|\tilde{f}_{n,s}\right| \quad \leq \bar{f} \cdot x_n & \forall s \in \tilde{S}_n \\ A\tilde{f}_{n,s} = \tilde{d}_{n,s} - \tilde{g}_{n,s} & \forall s \in \tilde{S}_n \end{cases} \tag{5.2}$$

5.2 Master-Slave decomposition algorithm

The method consists in decomposing (5.1)-(5.2) into a **master** and many **slaves** subproblems. The **master** subproblem deals with the investment problem at node n. The **slaves** subproblems perform the expansion plan suggested by the *master* and check its feasibility on the sub-scenarios $s \in \tilde{S}_n$. The iterations between the master and the $\tilde{S}_n 1$ slaves problems is then ensured by feasibility cuts, computed from the **slaves**' solutions and added to the **master** subproblem. We remind that the interest of the decomposition is to transform a problem in many sub-problems of lower sizes that can be parallelized.

One can remark that we have chosen to include to implement as many slave subproblems as sub-scenarios. We could have done differently, by including one scenario or more in the master. Hence, it is a heuristics issue that we do not detail here. Nonetheless, in the tests that we have performed, we remarked that including the base test scenario in the master problem enabled to improve the convergence.

To summarize, the idea is to combine the ND algorithm that iterates **over the nodes** of the tree, with the master-slave decomposition algorithm, that iterates **within the nodes** of the tree.

At iteration i of the ND algorithm, we solve a master-slave decomposition algorithm within each node. At iteration j, we solve then the master problem at node n:

$$\begin{aligned}
(P_n^{i,j}(x_{a(n)}^i, \psi_n^i)) : \quad \underline{Q}_n^j(x_{a(n)}^i) := \min_{x_n} \ & c^T x_n + \psi_n^i(x_n) \\
\text{s.t.} \ & x_n \in \{0,1\}^K \\
& z_n \in [0,1]^K \\
& z_n \leq x_n \\
& z_n = x_{a(n)}^i \\
& \psi_n^i(x_n) \geq \sum_{m \in \mathcal{C}(n)} q_{nm}(v_m^l + (\pi_m^l)^T x_n), \forall l < i \\
& 0 \geq w_s^l + {\beta_s^l}^T (x_n - x_n^l) \qquad , \forall l < j, s \in \tilde{S}_n
\end{aligned} \tag{5.3}$$

The constraint $0 \geq w_s^l + {\beta_s^l}^T (x_n - x_n^l)$ is the feasibility cut associated with the sub-scenario s and iteration $l < j$ of the master-slave algorithm. The coefficients $\{w_s^l, \beta_s^l\}$ are obtained by solving the s-th ($\in \tilde{S}_n$) slave problem at iteration l. Also, x_n^l is the solution of the master problem solved at iteration l.

The input of the slave problem is the investment plan x_n^j deduced from the resolution of the master problem at iteration j. The idea is to check if the $\tilde{S}_n$ scenarios are feasible when implementing the investment plan. In order to do so, we relax the first Kirchhoff law constraint by introducing a slack variable $\tilde{r}_{n,s}$ that represents load curtailment for node n and scenario $s \in \tilde{S}_n$. The objective of the slave problem is then to minimize the load curtailment that would induce the investment plan x_n^j.

Hence, the slave problem for scenario s at iteration j of the master-slave decomposition algorithm and iteration i of the ND algorithm is :

$$\begin{aligned}
(\tilde{P}_s^{i,j}(x_n^j)) : \quad R_s(x_n^j) := \min_{\tilde{r}_{n,s}} \ & \tilde{r}_{n,s} \\
\text{s.t.} \ & \tilde{f}_{n,s} = \gamma \cdot \Delta\tilde{\theta}_n \\
& \left|\tilde{f}_{n,s} - \gamma \cdot \Delta\tilde{\theta}_n\right| \leq M \cdot (1 - x_n^j) \quad \leftarrow (\Pi_M) \\
& \left|\tilde{f}_{n,s}\right| \leq \bar{f} \\
& \left|\tilde{f}_{n,s}\right| \leq \bar{f} \cdot x_n \qquad \leftarrow (\Pi_f) \\
& A^T \tilde{f}_{n,s} + \tilde{r}_{n,s} = \tilde{d}_{n,s} - \tilde{g}_{n,s} \\
& |\tilde{r}_{n,s}| \leq \min(\tilde{d}_{n,s}, \tilde{g}_{n,s})
\end{aligned} \tag{5.4}$$

Thanks to duality theory, as computed in [4], we use the dual multipliers associated to the constraints to compute the feasibility cut coefficient β_s^l. Let us denote by Π_M the dual multiplier associated with the second constraint of (5.4) and by Π_f the dual multiplier associated with the fourth constraint. Then, we have :

$$\beta_s^l = M \cdot \Pi_M - \bar{f} \cdot \Pi_f$$

And we also have : $w_s^l = R_s(x_n^j)$.

We can then add the feasibility cut obtained to the master problem. The master-slave decomposition algorithm stops when the curtailment of all sub-scenarios $s \in \tilde{S}_n$ equals to the null vector.

Let us summarize the master-slave decomposition algorithm done at iteration i of the ND algorithm, at the node $n \in \mathcal{N}$.

Algorithm 2: Master-slave Decomposition algorithm

Data: convergence threshold $\tilde{\epsilon}$
Initialization : $j = 0$
Result: x_n^i investment optimal solution for iteration i of ND algorithm

while $\sum_{s \in \tilde{S}_n} |\tilde{r}_{n,s}| > \tilde{\epsilon}$ **do**
  /*While there is at least one sub-scenario with non-null curtailment
  $j \leftarrow j + 1$
  /* Solve master problem :
  Solve $(P_n^{i,j}(x_{a(n)}^i, \psi_n^i))$
  Collect investment plan x_n^j
  /* Solve slave problems :
  forall $s = 1 \ldots \tilde{S}_n$ **do**
    Solve $(\tilde{P}_s^{i,j}(x_n^j))$
    Collect cuts coefficients $\{w_n^j, \beta_n^j\}$
    Add a new feasibility cut to the master problem $(P_n^{i,j+1}(x_{a(n)}^i, \psi_n^i))$
    Store the curtailment $\tilde{r}_{n,s}$
  end
end
return x_n^i

5.3 Computational experiments

This section presents the numerical results obtained by the combination of the two decomposition approaches.

5.3.1 Garver system

For each node of the tree (either $S = 4$ or $S = 9$), we create the set $\tilde{S}_n$. For a given, we vary the number of sub-scenarios, by testing with 2, 3 and 5 sub-scenarios. We create these scenarios by applying random perturbations to the basis scenario.

5.3.1.1 Comparison to the extensive form

In order to validate the algorithm, we compare it to the extensive form. As noted before, the solver is not able to solve the problem with 2nd Kirchhoff law, on a tree of 4 leaves

$(S = 4)$. That is why we remove the 2nd Kirchhoff law for the moment. We also consider one sub-scenario by node $(\tilde{S}_n = 2, \forall n \in \mathcal{N})$.

As presented in the tables above, when we consider a tree of 4 scenarios and 2 sub-scenario by node, the ND algorithm converges towards the optimal solution and is extremely fast in comparison to the model without decomposition.

	Optimal cost	Time
S = 4, $\tilde{S}_n$ = 2, T = 3	190	6h36min

Table 5.1: Garver system without 2nd Kirchhoff law, for a tree of 4 scenarios, and 2 sub-scenario by node, in the extensive form.

	Optimal cost	Time (in s)	Nb Iterations	Relative gap
S = 4, $\tilde{S}_n$ = 2, T = 3	190	21s	6	0 %

Table 5.2: Garver system without 2nd Kirchhoff law, for a tree of 4 scenarios, and 2 sub-scenario by node, with ND algorithm with (SB) cuts.

5.3.1.2 With 2nd Kirchhoff law

We reintroduce the 2nd Kirchhoff law and increase the leaves of the tree as well as the number of sub-scenarios by node.

	Optimal cost	Time (in s)	Nb Iterations	Relative gap
S = 4, $\tilde{S}_n$ = 3, T = 3	370	187s	4	0 %
S = 9, $\tilde{S}_n$ = 5, T = 3	416.7	887s	8	0 %

Table 5.3: Garver system with 2nd Kirchhoff law, by varying scenarios and sub-scenarios by node, with ND algorithm and (SB) cuts.

Hence, for cases that cannot be handled in the extensive form, our algorithm is able to solve the problem to optimality in a reasonable time.

5.3.2 IEEE system

Finally, we test our double decomposition approach on IEEE instance, for a tree of 4 leaves (resp 9) and 2 or 3 sub-scenarios by node. We create the sub-scenarios by imposing a perturbation of :

- 1% to all buses' generation and demand of the base-scenario when there is two sub-scenario by node.
- More or less 5% to all buses' generation and demand of the base-scenario when there is three sub-scenario by node.

Here again, we observe the convergence of algorithm when (SB) cuts are used.

	Optimal cost	Time (in s)	Nb Iterations	Relative gap
S = 4, $\tilde{S}_n = 2$, T = 3	432	190s	6	0 %
S = 4, $\tilde{S}_n = 3$, T = 3	564	592s	20	0%
S = 9, $\tilde{S}_n = 2$, T = 3	481.3	314s	8	0%
S = 9, $\tilde{S}_n = 3$, T = 3	608.7	922s	20	0%

Table 5.4: IEEE system with 2nd Kirchhoff law, by varying scenarios and sub-scenarios by node, with ND algorithm and (SB) cuts.

Part 6

Conclusion and future work

To conclude, we consider the well-known problem of long-term Transmission Expansion Planning. Due to the presence of binary variables in a stochastic setting, these problems are known to come with a computational burden. In a recent work [19], Ahmed proposed a finite and exact Nested Decomposition algorithm and its stochastic variant (SDDIP) for solving MSIP. Extensive numerical studies gave encouraging results on real-life examples, and led us to implement the algorithm for solving Transmission Expansion Planning problems.

We were able to confirm the conclusions of [19] and to obtain encouraging results for many test-cases. By comparing to a solving method without decomposition we validated our implementations. **Besides, in all the cases we have studied, we verified the convergence of the algorithm towards optimality in a relatively fast time, when strength benders' cuts were used**. We even obtained convergence in cases that were intractable when solving without decomposition.

We can then conclude on the relevance of the ND algorithm for solving to optimality the long-term Transmission Expansion Planning.

In order to circumvent the computational issue of large trees, we came with a new approach, combining Nested Decomposition to a master-slave decomposition algorithm, previously used in the literature on static problems. This approach also gave encouraging results.

There are several interesting directions worth investigating for future research :

- We have transformed the original Garver and IEEE instances in order to deal with the computation of the big M parameter of the 2nd Kirchhoff Law. We could use the literature on the big M and compute the corresponding parameter without modifying our network.
- It would be interesting to test our approach on real-life instances.
- It can be valuable to implement a sampling procedure as in SDDIP. It would enable to tackle a very large number of scenarios.

Annexes

We give here the data for the instances we have tested. We use the following notations.

Nomenclature

Bfr	Original bus of the circuit
Bto	Ending bus of the circuit
Nc	Possible number of candidate circuits with same extremal buses to be installed
R	Reactance of the circuit
Cap	Maximal capacity of the circuit
G*t*	Generation at time *t* by bus
D*t*	Demand at time *t* by bus
G*t*S*s*	Generation by bus at time *t* for scenario *s*

6.1 Data for Garver system

Bus	Bus name
1	BARRA---1
2	BARRA---2
3	BARRA---3
4	BARRA---4
5	BARRA---5
6	BARRA---6

Table 6.1: Bus data for Garver system

Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	1	40.	100.	0.
1	4	1	60.	80.	0.
1	5	1	20.	100.	0.
2	3	1	20.	100.	0.
2	4	1	40.	100.	0.
3	5	1	20.	100.	0.

Table 6.2: Existing circuits for Garver system

Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	5	40.	100.	40.
1	3	5	38.	100.	38.
1	4	5	60.	80.	60.
1	5	5	20.	100.	20.
1	6	4	68.	70.	68.
2	3	5	20.	100.	20.
2	4	5	40.	100.	40.
2	5	4	31.	100.	31.
2	6	4	30.	100.	30.
3	4	4	59.	82.	59.
3	5	5	20.	100.	20.
3	6	4	48.	100.	48.
4	5	4	63.	75.	63.
4	6	4	30.	100.	30.
5	6	4	61.	78.	61.

Table 6.3: Candidate circuits for Garver system

Bus	G1	G2	G3	D1	D2	D3
1	50	50	50	80	135	135
2	0	0	0	240	240	270
3	165	165	165	40	40	40
4	0	0	0	160	160	160
5	0	0	0	240	240	240
6	545	600	630	0	0	0

Table 6.4: Demand and Generation by bus for Garver system

6.2 Data for the modified Garver system

We augment the network by connecting the buses and by adding demand and generation as it follows. We present only the modified tables.

Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	1	40	100	0
1	3	1	38	100	0
1	4	1	60	80	0
1	5	1	20	100	0
1	6	1	68	70	0
2	3	1	20	100	0
2	4	1	40	100	0
2	5	1	31	100	0
2	6	1	30	100	0
3	4	1	59	82	0
3	5	1	20	100	0
3	6	1	48	100	0
4	5	1	63	75	0
4	6	1	30	100	0
5	6	1	61	78	0

Table 6.5: Existing circuits for modified Garver system

Bus	G1	G2	G3	D1	D2	D3
1	50	50	50	80	135	135
2	360	360	360	240	240	270
3	165	165	165	400	400	400
4	0	0	0	160	160	160
5	0	0	0	240	240	240
6	545	600	630	0	0	0

Table 6.6: Demand and Generation by bus for modified Garver system

6.3 Data for the IEEE system

Bus	Bus name	-	Bus	Bus name
1	BARRA---1	-	13	BARRA---13
2	BARRA---2	-	14	BARRA---14
3	BARRA---3	-	15	BARRA---15
4	BARRA---4	-	16	BARRA---16

Bus	Bus name	-	Bus	Bus name
5	BARRA---5	-	17	BARRA---17
6	BARRA---6	-	18	BARRA---18
7	BARRA---7	-	19	BARRA---19
8	BARRA---8	-	20	BARRA---20
9	BARRA---9	-	21	BARRA---21
10	BARRA---10	-	22	BARRA---22
11	BARRA---11	-	23	BARRA---23
12	BARRA---12	-	24	BARRA---24

Table 6.7: Bus data for IEEE system

Bfr	Bto	Nc	R(%)	Cap.	Cost	-	Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	1	1.39	175	0	-	12	23	1	9.66	500	0
1	3	1	21.12	175	0	-	13	23	1	8.65	500	0
1	5	1	8.45	175	0	-	14	16	1	3.89	500	0
2	4	1	12.67	175	0	-	15	16	1	1.73	500	0
2	6	1	19.2	175	0	-	15	21	1	4.9	500	0
3	9	1	11.9	175	0	-	15	21	2	4.9	500	0
3	24	1	8.39	400	0	-	15	24	1	5.19	500	0
4	9	1	10.37	175	0	-	16	17	1	2.59	500	0
5	10	1	8.83	175	0	-	16	19	1	2.31	500	0
6	10	1	6.05	175	0	-	17	18	1	1.44	500	0
7	8	1	6.14	175	0	-	17	22	1	10.53	500	0
8	9	1	16.51	175	0	-	18	21	1	2.59	500	0
8	10	1	16.51	175	0	-	18	21	2	2.59	500	0
9	11	1	8.39	400	0	-	19	20	1	3.96	500	0
9	12	1	8.39	400	0	-	19	20	2	3.96	500	0
10	11	1	8.39	400	0	-	20	23	1	2.16	500	0
10	12	1	8.39	400	0	-	20	23	2	2.16	500	0
11	13	1	4.76	500	0	-	21	22	1	6.78	500	0
11	14	1	4.18	500	0	-	12	13	1	4.76	500	0

Table 6.8: Existing circuits for IEEE system

Bfr	Bto	Nc	R(%)	Cap.	Cost	-	Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	3	1.39	175	3	-	11	14	3	4.18	500	58
1	3	3	21.12	175	55	-	12	13	3	4.76	500	66
1	5	3	8.45	175	22	-	12	23	3	9.66	500	134
1	8	3	13.44	500	35	-	13	14	3	4.47	500	62
2	4	3	12.67	175	33	-	13	23	3	8.65	500	120
2	6	3	19.2	175	50	-	14	16	3	3.89	500	54
2	8	3	12.67	500	33	-	14	23	3	6.2	500	86
3	9	3	11.9	175	31	-	15	21	3	4.9	500	68

3	24	3	8.39	400	50	-	15	21	3	4.9	500	68
4	9	3	10.37	175	27	-	15	24	3	5.19	500	72
5	10	3	8.83	175	23	-	16	17	3	2.59	500	36
6	7	3	19.2	500	50	-	16	19	3	2.31	500	32
6	10	3	6.05	175	16	-	16	23	3	8.22	500	114
7	8	3	6.14	175	16	-	17	18	3	1.44	500	20
8	9	3	16.51	175	43	-	17	22	3	10.53	500	146
8	10	3	16.51	175	43	-	18	21	3	2.59	500	36
9	11	3	8.39	400	50	-	19	20	3	3.96	500	55
9	12	3	8.39	400	50	-	19	23	3	6.06	500	84
10	11	3	8.39	400	50	-	20	23	3	2.16	500	30
10	12	3	8.39	400	50	-	21	22	3	6.78	500	94
11	13	3	4.76	500	66	-						

Table 6.9: Candidate circuits for IEEE system

Bus	G1S1	G2S1	G3S1	G1S2	G2S2	G3S2	G1S3	G2S3	G3S3	G1S4	G2S4	G3S4
1	576	576	576	465	661	661	576	576	576	520	520	520
2	576	576	576	576	576	576	576	576	576	520	520	520
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	196	196	0	0	0	0	0	0	0	0	0
7	900	900	900	722	722	722	900	900	900	812	812	812
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	100	0	0	0	0	0	0	0	0	0
13	1424	1424	1524	1457	1457	1457	1599	1599	1599	0	196	196
14	0	0	0	0	0	0	0	0	0	795	795	795
15	645	645	645	325	521	521	581	581	581	582	582	582
16	465	465	465	465	465	465	282	282	282	419	419	419
17	0	0	0	0	0	0	0	0	100	0	0	0
18	1200	1200	1200	1200	1200	1200	603	603	603	718	718	718
19	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
21	1200	1200	1200	1200	1200	1200	951	951	951	1077	1077	1077
22	900	900	900	900	900	900	900	900	900	900	900	1000
23	315	315	315	953	953	953	1980	1980	1980	1404	1404	1404
24	0	0	0	0	0	0	0	0	0	0	196	196

Table 6.10: Generation by bus for IEEE system

Bus	D1	D2	D3	-	Bus	D1	D2	D3
1	324	324	324	-	13	0	196	196
2	291	291	291	-	14	795	795	795
3	540	540	540	-	15	582	582	582
4	222	222	222	-	16	951	951	951
5	213	213	213	-	17	300	300	300
6	408	408	408	-	18	999	999	999
7	375	375	375	-	19	543	543	543
8	513	513	513	-	20	384	384	384
9	525	525	525	-	21	0	0	0
10	585	585	585	-	22	0	0	0
11	0	0	0	-	23	0	0	0
12	0	0	100	-	24	0	0	0

Table 6.11: Demand by bus for IEEE system

6.4 Data for the modified IEEE system

We augment the network by connecting the buses and by adding demand and generation as it follows. We present only the modified tables.

Bfr	Bto	Nc	R(%)	Cap.	Cost	Bfr	Bto	Nc	R(%)	Cap.	Cost
1	2	1	1.39	175	0	11	14	1	4.18	500	0
1	3	1	21.12	175	0	12	13	1	4.76	500	0
1	5	1	8.45	175	0	12	23	1	9.66	500	0
1	8	1	13.44	500	0	13	14	1	4.47	500	0
2	4	1	12.67	175	0	13	23	1	8.65	500	0
2	6	1	19.2	175	0	14	16	1	3.89	500	0
2	8	1	12.67	500	0	14	23	1	6.2	500	0
3	9	1	11.9	175	0	15	16	1	1.73	500	0
3	24	1	8.39	400	0	15	21	1	4.9	500	0
4	9	1	10.37	175	0	15	21	2	4.9	500	0
5	10	1	8.83	175	0	15	24	1	5.19	500	0
6	7	1	19.2	500	0	16	17	1	2.59	500	0
6	10	1	6.05	175	0	16	19	1	2.31	500	0
7	8	1	6.14	175	0	16	23	1	8.22	500	0
8	9	1	16.51	175	0	17	18	1	1.44	500	0
8	10	1	16.51	175	0	17	22	1	10.53	500	0
9	11	1	8.39	400	0	18	21	1	2.59	500	0
9	12	1	8.39	400	0	18	21	2	2.59	500	0
10	11	1	8.39	400	0	19	20	1	3.96	500	0
10	12	1	8.39	400	0	19	20	2	3.96	500	0
11	13	1	4.76	500	0	19	23	1	6.06	500	0
20	23	2	2.16	500	0	20	23	1	2.16	500	0

Bfr	Bto	Nc	R(%)	Cap.	Cost	Bfr	Bto	Nc	R(%)	Cap.	Cost
21	22	1	6.78	500	0						

Table 6.12: Existing circuits for modified IEEE system

Bus	G1S1	G2S1	G3S1	G1S2	G2S2	G3S2	G1S3	G2S3	G3S3	G1S4	G2S4	G3S4
1	1076	1076	1076	965	1161	1161	1076	1076	1076	1020	1020	1020
2	1076	1076	1076	1076	1076	1076	1076	1076	1076	1020	1020	1020
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	500	500	500	500	500	500	500	500	500	500	500	500
7	900	900	900	722	722	722	900	900	900	812	812	812
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	100	0	0	0	0	0	0	0	0	0
13	2273	2273	2273	1924	1924	2024	1957	1957	1957	2099	2099	2099
14	500	696	696	500	500	500	500	500	500	500	500	500
15	645	645	645	645	645	645	325	521	521	581	581	581
16	965	965	965	965	965	965	782	782	782	919	919	919
17	0	0	0	0	0	0	0	0	100	0	0	0
18	1200	1200	1200	1200	1200	1200	603	603	603	718	718	718
19	500	500	500	500	500	500	500	500	500	500	500	500
20	0	0	0	0	0	0	0	0	0	0	0	0
21	1200	1200	1200	1200	1200	1200	951	951	951	1077	1077	1077
22	900	900	900	900	900	900	900	900	900	900	900	1000
23	315	315	315	953	953	953	1980	1980	1980	1404	1404	1404
24	0	0	0	0	0	0	0	0	0	0	196	196

Table 6.13: Generation by bus for modified IEEE system

Bus	D1	D2	D3	-	Bus	D1	D2	D3
1	324	324	324	-	13	0	196	196
2	291	291	291	-	14	1295	1295	1295
3	540	540	540	-	15	582	582	582
4	222	222	222	-	16	951	951	951
5	213	213	213	-	17	300	300	300
6	408	408	408	-	18	999	999	999
7	875	875	875	-	19	543	543	543
8	1513	1513	1513	-	20	384	384	384
9	525	525	525	-	21	0	0	0
10	585	585	585	-	22	0	0	0

Bus	D1	D2	D3	-	Bus	D1	D2	D3
11	0	0	0	-	23	1500	1500	1500
12	0	0	100	-	24	0	0	0

Table 6.14: Load by bus for modified IEEE system

Bibliography

[1] Richard Bellman. Dynamic programming princeton university press. *Princeton, NJ*, 1957.

[2] J. F. Benders. *Partitioning methods for solving mixed-variables programming problems*, volume 4. Numerische mathematik, 1962.

[3] S. Binato. *Optimal power transmission expansion planning by Benders decomposition and cutting planes techniques. D.Sc. thesis.* COPPE-Federal University of Rio de Janeiro, 2000.

[4] S Binato, M. V. F. Pereira, and S. Granville. A new benders decomposition approach to solve power transmission network design problems. *Transactions on Power Systems*, 16, 2001.

[5] J.R. Birge. *Decomposition and partitioning methods for multistage stochastic linear programs.*, volume 33. Operations Research, 1985.

[6] J.R. Birge and F. Louveaux. *Introduction to Stochastic programming.* Springer, 2 edition, 2011.

[7] Pereira M. V. F. and S. Granville. Analysis of the linearized power flow model in benders decomposition. 1985.

[8] L.L. Garver. Transmission network estimation using linear programming. *IEEE Trans. Power Appar. Syst.*, 89, 1970.

[9] J.C. Kaltenbach, J. Peschon, and E.H. Gehrig. A mathematical optimization technique for the expansion of electric power transmission systems. pages 113—119, 1970.

[10] G. Laporte and F. V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 2016.

[11] S. Lumbreras and A. Ramos. The new challenges to transmission expansion planning. survey of recent practice and literature review. *Electric Power Systems Research*, 2015.

[12] M. V. F. Pereira. *Application of sensitivity analysis on generation-transmission system expansion planning, D.Sc. dissertation (in Portuguese).* COPPE-Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil.

[13] M. V. F. Pereira and L. M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1-3):359–375, 1991.

[14] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[15] F.S. Thomé. *Application of Decomposition Technique with Evaluation of Implicit Multipliers in Electrical Systems Generation and Network Expansion Planning. Dissertação de Mestrado, (in portuguese).* COPPE/UFRJ, Rio de Janeiro, Brazil, 2008.

[16] F.S. Thomé, M. V. F. Pereira, S. Granville, and M.H.C.F Fampa. Non-convexities representation on hydrothermal operation planning using sddp. *IEEE PES Transactions on Power Systems*, 2013.

[17] P. Tsamasphysrou, A. Renaud, and P. Carpentier. *Transmission network planning: An efficient Benders decomposition scheme.* Proceedings of the 13th PSCC, 1999.

[18] R. Villanasa. *Transmission network planning using linear and mixed linear integer programming,Ph.D. dissertation.* Ressenlaer Polythechnic Institute, 1984.

[19] Jikai. Zou, S. Ahmed, and X.A Sun. Nested decomposition of multistage stochastic integer programs with binary state variables. 2016.

List of Figures

List of Tables