A computational model for multi-objective optimization of zero emission power plants

A dissertation presented
by
Xinxin Li
Submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the
Graduate School of Arts and Sciences
Columbia University
2011
Abstract

A computational model for multi-objective optimization of zero emission power plants

Xinxin Li

Choosing among technologies is difficult and requires a means of making comparisons across different technologies. This dissertation proposes a new methodology to make comparisons across different technologies and across different times based on a user supplied set of evaluation criteria.

A simple model is developed to evaluate different technologies and to identify optimal technologies and technology pathways based on a user supplied set of evaluation criteria which allow ranking of different plants, and technology pathways, which represent different time sequences of introducing new power plant designs. This model is applied to a simple set of choices for power plant designs that involve eight basic operation modules and a total of 96 possible power plant designs, of which 18 are physically feasible. The model also considers five unique pathways of transition over time from one type of power plant to another type. These pathways are ranked based on penalties assigned on the module level, plant level and pathway level. This dissertation studies two cases, where CO$_2$ regulation does and does not take effect. The results show that a shorter path is favorable, and multiple changes at the same time is undesirable. The relative ranking of different pathways are different in the two cases.

To find the optimal path among the entire space of solutions, we develop two combinatorial optimization algorithms. The objective function is defined as the minimum of penalties which are imposed for all deviations from an ideal or perfect system.
Abstract

The numerical problem of finding an optimum is solved by means of a branch-and-bound method, and a heuristic based on the label-correcting algorithm for solving the shortest-path problem. The proposed algorithms are applied to the practical examples of finding the optimal sequence of various power plant designs. The computational results show that the performance of the path-dependent shortest path algorithms depends on the structure of the problem. For average problems, the branch-and-bound algorithm is more efficient compared with the brute force search approach. In the worst case, the branch-and-bound algorithm degenerates into the brute-force search approach. Both branch-and-bound and the brute-force search approach are exact methods. For average problems, the heuristic is more efficient than the branch-and-bound algorithm. The heuristic is not an exact method and there is no guarantee that it always finds the optimum. However, it can find a good result in a reasonable time.

We use these algorithms to study technology pathways which consist of power plant designs with CO$_2$ post-combustion capture technologies. We consider a small problem that consists of 6 designs and 14 levels of decisions, a medium problem consisting of 84 designs and 15 levels of decisions, and a big problem consisting of 492 designs and 15 decisions. We use the branch and bound algorithm for the small problem, and the heuristic for the medium and big problems. The results of small, medium and big problems show that, the best technology pathway, or the best sequence of technologies, does not agree with the sequence of best technologies of various times. By choosing a suboptimal design upfront, one can obtain a better technology pathway than the pathway with a sequence of best designs.
Abstract

We develop a flexible software tool that enables process modeling and optimization of complicated energy systems. The software tool models a plant in terms of basic operation modules and streams that connect the modules with material and energy flows. The software represents the beginning of new modeling capability that is useful for studying individual energy systems. It introduces a new concept in comparison to traditional software tools by optimizing over entire technology pathway consisting of a time sequence of plant designs and technology choices.
Table of Contents

Table of Contents ........................................... i
List of Figures .................................................. ii
List of Tables ................................................... iv
Abbreviations .................................................. v
Acknowledgments ............................................... ix
Dedication ...................................................... xii

1 Dissertation Objective and Organization 1
   1.1 Background ............................................... 1
   1.2 Goal & Objectives ....................................... 4

2 A numerical method for multi-objective optimization of zero emission power plants 8
   2.1 Evaluation criteria based on a penalty model .............. 10
   2.2 Problem formulation .................................... 12
   2.3 Penalties .................................................. 13

3 A simple model for power plant pathway optimization 15
   3.1 Model Formulation ....................................... 17
   3.2 Model Assumptions ...................................... 35
   3.3 Results and Discussion .................................. 36

4 Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs 41
   4.1 Problem statement ....................................... 42
   4.2 Mathematical formulation ................................. 45
   4.3 Literature review on shortest-path problem ............... 47
   4.4 Path-dependent shortest-path algorithms .................. 48
   4.5 Exhaustive search ....................................... 50
   4.6 Branch and bound algorithm with bottom-up pruning ....... 50
   4.7 Heuristic based on label-correcting algorithm ............ 56
# Table of Contents

4.8 Computational results ........................................... 59  
4.9 Discussion .................................................. 65  

5 Optimizing CO₂ post-combustion capture technologies  69  
5.1 Motivation .................................................. 70  
5.2 Literature Review ............................................ 72  
5.3 Modeling CO₂ absorber physics ............................... 75  
5.4 Absorber Cost Model ........................................ 83  
5.5 Penalty model ............................................... 85  
5.6 Basic optimization assumptions .............................. 89  
5.7 Results .................................................... 93  
5.8 Discussion ................................................ 103  
5.9 Conclusion ............................................... 106  

6 Designing the software tool for advanced power plant modeling and optimization  108  
6.1 Motivation .................................................. 111  
6.2 A model of modules and pipes ............................... 113  
6.3 Input file .................................................. 116  
6.4 Reconcile: solving a system of equations ................. 118  
6.5 Penalty .................................................... 122  
6.6 Parameters ............................................... 122  
6.7 Optimization ............................................... 123  
6.8 Output ..................................................... 124  
6.9 Test routines ............................................... 124  
6.10 Limitation ............................................... 124  

7 Conclusion .................................................. 126  

Bibliography .................................................. 130  

A Appendix .................................................. 139  
A.1 Introduction to TéX ........................................ 139  
A.2 Appendix: Performance data calculated using IECM Model 140  
A.3 The basic assumptions in the module library .............. 142  
A.4 Testing model ............................................ 142  
A.5 Module library ............................................ 144  
A.6 Post-combustion capture technology .......................... 148
List of Figures

3.1 Environmental Penalty ........................................ 22
3.2 Five pathways of different numbers of sequence elements .......... 35
3.3 Case 1: Pathways end in an environment with CO\(_2\) regulations .... 36
3.4 Case 2: Pathways end in an environment without CO\(_2\) regulations .. 36
3.5 Case 1: Plant level penalties with CO\(_2\) regulations ................ 37
3.6 Case 2: Plant level penalties without CO\(_2\) regulations ............. 37
3.7 Case 1: Environmental penalties with CO\(_2\) regulations ............. 38
3.8 Case 2: Environmental penalties without CO\(_2\) regulations .......... 39

4.1 An instance of a rooted tree .................................... 43
4.2 An instance of a directed graph .................................. 46
4.3 An instance of depth first search strategy in branch and bound .... 52
4.4 An instance of a graph, in which an edge has multiple values as a result of multiple path histories .................................... 57
4.5 Visited nodes in branch-and-bound(BB) algorithm .................... 62
4.6 The optimal results given by BB and the brute-force approach ....... 62
4.7 The visited nodes comparison between BB and the heuristic .......... 63
4.8 The results given by the heuristic compared to the brute-force approach 63
4.9 The results of the heuristic as x increases .......................... 64
4.10 Visited nodes in the branch-and-bound and the hybrid algorithm .... 67

5.1 An instance of the cross-section of a packing tower .................. 76
5.2 Basic assumptions of the three sorbents ........................... 90
5.3 Power Generation Unit Assumptions ................................ 91
5.4 Forecast CO\(_2\) capture percentage standards for small, medium and big problems ............................................. 92
5.5 Plant performance data of the 6 designs ............................ 94
5.6 Single plant and pathway optimization, 6 design choices for 14 decision points .................................................. 95
5.7 6 design results. Plant configuration comparisons .................... 96
5.8 84 design configurations ............................................ 97
List of Figures

5.9  84 design results. The optimal cost of a single plant and a sequence of plants ........................................... 98
5.10 84 design results. Plant configuration comparisons ...................... 98
5.11 492 design results. The optimal cost of a single plant and a sequence of plants ........................................... 101
5.12 492 design results. Plant configuration comparisons ...................... 102
5.13 84 design results. The optimal cost of a single plant and a sequence of plants. $k_{CO2} = 30$ ........................................... 103
5.14 84 design results. Plant configuration comparisons. $k_{CO2} = 30$ ........ 104
5.15 492 design results. The optimal cost of a single plant and a sequence of plants ........................................... 105
5.16 492 design results. Plant configuration comparisons ...................... 105

6.1  The concept of a generic module ........................................ 115
6.2  The concept of a generic pipe ........................................ 115
6.3  An instance of a flowsheet input file with one Furnace .................... 117

A.1  Power plant performance data from IECM model I ...................... 141
A.2  Power plant performance data from IECM model II ...................... 141
A.3  Power plant performance data from IECM model III ...................... 141
List of Tables

3.1 Summary of power plant designs and pathways designs . . . . . . . . . 33
3.2 Module level, plant level and pathway level penalties . . . . . . . . . 34
6.1 Input File (Flowsheet) Definition (Format) . . . . . . . . . . . . . 116
A.1 Modules and the parameters that measure the module sizes . . . . . 142
A.2 Module Operating and Maintenance Cost breakdown . . . . . . . . . 142
Abbreviations

Chapter 2

$m = \text{No. of decision points in a sequence of decisions}$

$n = \text{No. of options to choose from at each decision time.}$

$t = \text{time}$

$z = \text{a representation of a sequence of power plants -}$

$y = \text{a specific power plant configuration}$

$x = \text{a module, usually an operational unit, like a pump, a furnace, or an absorber}$

$N = \text{the number of modules in a power plant design}$

Chapter 3

$\text{TPI} = \text{Total plant investment $}$

$\text{OAM} = \text{Operation and maintenance cost $/yr}$

$w = \text{Weight}$

$w_{\text{TPI}} = \text{total plant investment weight}$

$w_{\text{OAM}} = \text{operation and maintenance cost weight}$

$w_{\text{reg}} = \text{regulatory factor}$

$w_{\text{comp}} = \text{compliance factor}$

$w_{\text{quan}} = \text{quantitative factor}$

$w_j^k = \text{plant weight;}$

$w_{\text{new}} = \text{new technology weight;}$

$w_{\text{delete}} = \text{obsolete technology weight;}$
Abbreviations

M = criteria pollutants, M = SO₂, particulates, NO, NO₂, Hg, CO₂.

P = Penalty Number – unit

\( P_{E_m} \) = Environmental penalty of criteria pollutants

\( P_{rec} \) = reconcile penalty

\( P_\eta \) = efficiency penalty

\( D_t \) = diffusion coefficient – m² s⁻¹

\( D_l \) = diffusivity in the liquid

\( D_{co2,mea} \) = diffusion coefficient of CO₂ in MEA – m² s⁻¹

\( D_{mea,mea} \) = diffusion coefficient of MEA in MEA – m² s⁻¹

Chapter 4

S = root of the tree

t = targe or leave in a tree

PDSP = path-depdent shortest path

T = N-ary rooted tree

c = cost function

Chapter 5

\( \tau \) = momentum – kg.m/s

\( \mu \) = dynamic viscosity – Pa.s

\( v_z \) = velocity on z direction – m/s

\( r \) = radius – m

\( J_{CO2,g} \) = CO₂ flux in bulk gas – mol/s/m²

\( \rho \) = density – kg/m³

D = diffusivity – m²/s
Abbreviations

\[ D_{\text{CO}_2,\text{AM}} = \text{CO}_2 \text{ diffusivity in amine solution} \]
\[ D_{\text{MEA,AM}} = \text{MEA diffusivity in amine solution} \]
\[ \text{Sc} = \nu/D_i - \text{Schmidt number} \]
\[ \nu = \text{kinetic viscosity} - m^2/s \]
\[ k_l = \text{liquid side mass transfer coefficient} - m/s \]
\[ \bar{P} = \text{pressure} \]
\[ \bar{P}_{\text{CO}_2,W} = \text{CO}_2 \text{ pressure on the wall} \]
\[ \bar{P}^*_{\text{CO}_2} = \text{CO}_2 \text{ equilibrium pressure} \]
\[ h = \text{tower height} \]
\[ \text{Re}_l = \text{Reynold’s number in the liquid. } Re = \rho \times V \times L/\mu \]
\[ \text{Ga} = \text{Galilei number. } Ga = g \times L^3/\nu^2 \]
\[ \text{Ha} = \text{Hatta number.} \]
\[ k_l^0 = \text{the physical mass transfer coefficient} \]
\[ E = \text{the enhancement factor} \]
\[ X = \text{mol fraction} \]
\[ X_{\text{amine,0}} = \text{initial amine as a fraction the sum of amine and } \text{H}_2\text{O} \]
\[ X_{\text{CO}_2} = \text{mol fraction of chemically bound } \text{CO}_2 \text{ in the solution.} \]
\[ K_{\text{CO}_2} = \text{CO}_2 \text{ vapor-liquid equilibrium partial pressure constant} \]
\[ \theta_{\text{avg}} = \text{the average loading (mo of } \text{CO}_2 /\text{mol of amine)} \]

- $C_{\text{CO}_2,i} = \text{CO}_2 \text{ concentration at the interface}$
Abbreviations

\[ C_M E A \] = MEA concentration

\[ T \] = Temperature

\[ E_{\text{inf}} \] = enhancement factor at infinity

\[ \gamma \] = stoichiometric ratio in the reaction

\[ R \] = absorber tower radius — m

\[ r_{\text{CO2}} \] = CO₂ uptake rate

\[ r_t \] = packing tube radius — m

\[ A_p \] = packing surface area — \( m^2 \)

\[ Q \] = volumetric flow rate — \( m^3/s \)

\[ \text{CP} \] = CO₂ capture percentage

\[ \text{PEC} \] = Purchase equipment cost—\$ 

\[ \text{EC} \] = Erection cost—\$

\[ \text{TCI} \] = total capital investment—\$

\[ \text{Capcost} \] = annual capital cost — (\$/yr)

\[ n_i \] = the number of unit production in the history

\[ \beta \] = the rate at which cost decreases as the unit production increases

\[ \alpha \] = the rate at which cost decreases as the unit size increases

\[ \eta_{\text{gross}} \] is the gross efficiency of the power plant

\[ \eta_{\text{gen}} \] is the generation efficiency, or net efficiency of the power plant
Acknowledgments

Completing this dissertation has been a fantastic experience. What began as an education became an unforgettable scientific adventure, and self discovery journey.

The first person I want to thank is my advisor Klaus S. Lackner, who is undoubtedly the most intuitive and supportive advisor that anyone could ask for. Klaus is not only the thought-leader in sustainable energy, but he’s also extraordinarily gentle and kind to everyone he works with. He has fostered certainly the most open, friendly, and collaborative research group in this department. He has also been there for me with patience and generous financial support during my program extension.

Throughout my seven years, I was supported for many semesters by the New York State Energy Research and Authority (NYSERDA). During my first two years, I thank the department for providing me with the teaching assistantship positions.

Klaus’ other students and post-docs, both past and present, comprise a superb research group. The opportunity to bounce ideas off so many excellent minds has been priceless. My most intense collaboration has been with Ian Katz, without whose help I would not have been able to implement the path-dependence optimization algorithms, much less produce any results.

Tim Rappold, whose clarity, persistence, cheerful personality taught me a lot and motivated me during some of the most difficult times. Federico Barrai has similarly influenced me and became not only collaborator, but a great friends through this experience. Tao Wang, whose invaluable insights and conversation on carbon sequestration and air capture was the most stimulating snack during the coffee-breaks. In my first couple of years in the Lackner group, Frank Zeman and I swapped many thoughts on carbon -sequestration. Jennifer Wade shared many good times and in-
introduced me to a world of membrane and solid oxide fuel cell. Joe Di Dio, for being my goto person in the first two years when I was trying hard to adjust to the life in America and graduate school. Jonathan Levine was my buddy through thesis writing. He has also helped me out many a time during those pre-presentation panics. Sam Krevor, Christopher Grave and Tom Socci have been companions since the year we arrived together.

Our group’s secretary Carey Russell is surely the kindest, coolest and most witty person one could possibly hope to spend a lunch-break gossiping with. This dissertation is not possible without Carey’s help to secure numerous meetings for me with Klaus among his busy calendar.

I must also thank Sarah Brennan for her total organization and help. She will be sorely missed in this group.

There are countless others who have been there for me throughout my time as a student. My superb committee including professor Cliff Stein, Tuncel Yegulalp, Paul Duby, and Alan West gave me numerous invaluable suggestions to make my work better. My dear friend Jianing Shi stands out as both a caring friend and a brilliant ballroom partner.

My parents and brother have always been there for me, encouraging me and reminding me how lucky I am that I was given the opportunity to pursue and live my dream. This dissertation is dedicated to them for their years of support and unconditional love.

Finally, my husband Christopher Ngan has been the most important reason that I continued my work over these last three years. I am truly grateful for his love and
Acknowledgments

friendship, I can not imagine finishing this journey without him. And as he rightfully pointed out, I must also thank Winnie the Pooh, without whose hugs and honey I would not have finished with a smiley face.
Dedicated to my father Shulin,
my mother Meihua,
my brother Xiaoming,
and my husband Christopher.
Chapter 1

Dissertation Objective and Organization

1.1 Background

Traditional power plant modeling tools only measure the thermodynamic performance and economic cost of a power plant. Optimization is often carried out to reach a compromise between plant efficiency and generation costs of the individual plants. Today, increasingly stringent environmental constraints, especially in response to climate change, require a new modeling tool to evaluate different power plant designs under various energetic, economical, environmental and infrastructural constraints. Choosing among different low-emission power generation technologies has a profound impact, not only on the cost-efficiencies of individual power plants, but more importantly on the pathways connecting current power generation technologies to future technologies.
The need to develop low-emission power generation technologies became clear because of two observations. The first observation is the rapidly growing consensus that excess carbon dioxide in the atmosphere will cause a significant change in climate with repercussions on a wide variety of human activities [Yegulalp 2001]. The second observation is that power plants burning fossil fuel are a major source of carbon dioxide emissions, accounting for roughly one third of the global emissions. In order to mitigate the emission of carbon dioxide from fossil fuel-fired power plants, a means is required to obtain a concentrated form of carbon dioxide with little impurities, before the gas can be disposed of safely and permanently. The capture and storage of carbon dioxide (CCS) is one of the greatest challenges not only to power generation infrastructure, but more importantly to sustainable access to cheap and abundant fossil fuels [Yegulalp 2001].

A range of low-emission power plant designs has been studied to address the environmental impacts of impurities and undesirable combustion products from fossil fuel for some time. At one end of the spectrum are power plants with environmental control units for desulphurization, particulate removal and mercury emission control, etc. At the other end of the spectrum is a special class of technologies for zero emission power plants. Such plant designs produce power without allowing any atmospheric emissions, in some designs even without a smoke stack. For example, a technology proposed by the zero emission coal alliance (ZECA) generates electricity with hydrogen fuel cells producing only water, where clean hydrogen is derived from coal with a hydrogasifier and a decarbonation reactor, followed by a calciner which separates out carbon dioxide together with impurities and undesirable combustion by-products for
sequestration [Lackner 2002].

Zero emission power plants have yet to be built. Today, small pilot scale low-emission power plants have shown how to minimize emissions of carbon dioxide to the atmosphere through separation and capture of carbon dioxide by one of three general techniques: pre-combustion, oxy-fuel, and post-combustion de-carbonization [IPCC, 2005]. Governments and firms that are eager to deploy low-emission thermal power plants must make their decisions among these three distinct types of power plant designs: the integrated gasification combined cycle design (IGCC), which relies on pre-combustion separation of carbon dioxide in a gasifier [IEA GHG, 2003], the oxy-fuel design which uses pure oxygen to yield a nearly pure stream of carbon dioxide after combustion [Croiset and Thambimuthu, 2000], and the post-combustion design which relies on chemical absorption processes for flue gas treatment [Hendriks, 1994]. Currently, there is no clear technological winner [MIT, 2007]. For both the IGCC and the oxy-fuel processes, the chief obstacle to broad deployment is the high capital and operational cost associated with each technique. Opportunities exist both for incremental modifications leading to near-term cost efficiencies, and for major redesigns leading to advanced next-generation power plant designs.

The IGCC and oxy-fuel processes are not the only possible implementations of low-emission power plants. There are a broad range of potential designs which incorporate a variety of components and approaches. For example, the advanced zero emission power plant (AZEP) addresses the development of a specific, zero emissions, gas turbine-based, power generation process, which relies on a mixed-oxide membrane for oxygen production [Griffin 2005]. A mixed-oxide membrane device is
a critical component shared by a variety of oxygen-enhanced combustion and gasification processes.

A wide range of different choices of designs for the next-stage designs, a far cry from the existing low efficiency, and polluting fossil fuel power plants, will lead to different technological pathways, which constitute a sequence of plant designs that build on each other, connecting to the efficient, economical and clean future technologies. The overall objectives of this dissertation are to create conceptual plant designs and conduct engineering assessments of the component modules of an advanced coal fired power plant; evaluate different power plant designs under various energetic, economic, environmental and infrastructural constraints; develop a software tool for modeling and optimizing complicated processes; and perform optimization, not just of the individual power plants alone, but of the pathways connecting current power generation technologies to future technologies.

1.2 Goal & Objectives

The goal of this dissertation is to develop a decision making tool, through comparing across different technologies, and selecting the best technology and the best technology pathway among the universe of all feasible solutions.

The dissertation is presented in phases. In Phase I, we define the modules that can be used to build an advanced coal-fired power plant and the range of operational criteria for their recombination to function in various modes by establishing the performance criteria, identifying key parameters, and creating computational models for each module. In Phase II, we focus on developing computational capability and
databases to model various power plant designs based on these models. Different plant process configurations are established, heat and mass balances for various plant designs are developed, and different power plant designs are assessed based on efficiency, emissions, economic and technical feasibility.

Specifically, this dissertation is divided into the following chapters. A novel numerical method is presented (Chapter 2) to evaluate different technologies in order to identify optimal technologies, based on a user-supplied set of evaluation criteria. The numerical method used here for advanced power plant designs evaluation and comparison, has taken its inspiration from the typesetting system \TeX, in which Donald Knuth demonstrated the power of these methods for trading off between text layouts with very different properties and with properties that are very difficult to quantify.

In Chapter 3, a simple example is presented to show how the model can be used to select appropriate power plant modules and a wide range of technologies, to arrive at a sequence of plant designs that provides an advantageous technology pathway from today’s power plant designs to a future design via a number of intermediate steps. Eight basic modules are chosen to form 96 possible plant designs, of which 18 are physically valid. The enumeration of the 18 power plants in a pathway consisting of five different power plants implemented in sequence, gives $10^6$ variations. As a preliminary study, we do not explore the entire space of all solutions in this Chapter. Instead, five unique pathways are chosen to show how various aspects of penalties affect their rankings. For example, a shorter path is favorable, but multiple changes at the same time is undesirable. The modeling results are subjective because they depend highly on the user-supplied set of weights for evaluation criteria. However,
the bias input are exogenous to the model, which reflects its flexibility to produce
customized solutions. By giving users the freedom to put their own preference to the
model, the model allows users to obtain their customized solutions.

To explore the entire space of solutions, we develop two combinatorial optimization
algorithms (Chapter 4). The objective function is defined as the minimum of
penalties. The problem is solved by means of a branch-and-bound method, and a
heuristic based on the label-correcting algorithm for solving shortest-path problem.
The proposed algorithms are applied for practical problems on finding the optimal
sequence of various power plant designs.

In Chapter 5, we study various post-combustion capture technologies using the
ranking algorithms introduced in Chapter 4. In the first part of our work, we develop
a simple model for the absorber system. We find the optimum design for a simple
model system given a specific sorbent strength by varying the packing tube radius and
the absorber tower height. In the second part of our work, we study various power
plant designs with absorbers in a sequence and find the optimum sequence using the
algorithm.

As a part of the dissertation, a computational software tool is developed to model
the mass-energy balance of a module, and to implement the ranking method (Ap-
pendix). This input-output model consists of modules and pipes, where each module
represents a fundamental operation unit (i.e, a expander, an heat exchanger, a CO₂
absorber etc), and a pipe can be considered as a material or energy flow. Each mod-
ule is characterized by a set of parameters, which satisfy a system of equations. The
program begins with a set of complete but inconsistent parameters, and finds a con-
sistent set of parameters through iterations. Upon finding such a reconciled system, the user may relax one (or a few) parameter(s) to a set of discrete values bounded by an upper bound and a lower bound. One can then call the optimization routine, such that the software can find the optimum choices of these parameters. For pathway optimization, the algorithms described in Chapter 4 are implemented. A library of five modules are built, and the manuals on coding a new module are included in the Appendix.
Chapter 2

A numerical method for multi-objective optimization of zero emission power plants

In this chapter, a novel numerical method to optimize (rank) advanced power plant designs and technology pathways is introduced. This work is motivated by the desire to develop a decision making tool, which compares across different technologies, and chooses the best technology and the best technology pathway among the universe of all feasible solutions. However, various technologies are characterized by very different properties, that are very difficult to quantify. For example, power plant owners are faced with not only economic and energetic constraints, but also increasingly stringent environmental, infrastructural, and social constraints. The former are relatively easy to quantify, the latter are very difficult. The need to compare different technologies and technology pathways, calls for a new evaluation metric based on very different
design constraints and optimization criteria. The need to rank different technologies and technology pathways, calls for a new method which is efficient, flexible and can deal with very complex systems.

We borrow our inspiration from the typesetting system \TeX, which is very powerful in ranking different typesetting layouts as more or less favorable. \TeX ranks different layouts via a penalty number, which represents the aesthetic value of the system. The penalty number is calculated by comparing the actual system with a perfect but unattainable state. This state is called the anchor point, which has a zero penalty value. The differences between the actual system and the anchor point, are captured and summarized into one penalty number. Since the anchor point is not achievable, the penalty number is always positive. The ranking of each aspect of the layout is evaluated with a penalty number. Each penalty number is multiplied with a penalty weight, which is chosen by the user. Different users with different preferences and goals may give different penalty weights. The summation of all weighted penalty numbers is the total penalty of the layout, which characterizes its aesthetic value. The smaller the penalty number is, the more favorable the layout is.

When applied to power plant modeling, a perfect and unattainable power plant design is chosen as the anchor point. The actual power plant design is compared to the anchor point design via a penalty number, which captures the differences in various aspects between the actual plant design and the anchor point. We can rank all possible designs to obtain the best design. In addition, we can rank various sequences of power plant designs, in which a new design can either build upon the previous ones, or can be entirely new. We can obtain the best technology pathway, by ranking all
possible technology pathways. The best technology pathway, or the best sequence of technologies, may not agree with the sequence of best technologies for various times.

This chapter introduces the new evaluation approach via the penalty number inspired by TeX, and describes the evaluation criteria for ranking various plant designs and various sequences of plant designs. It further introduces the penalty functions for power plant design evaluation and pathway evaluation. The ranking methods to find the best power plant designs and technology pathways are introduced in Chapter 4.

\section*{2.1 Evaluation criteria based on a penalty model}

Power plant development leading to a zero emission plant design can move through a set of new plants, each designed to be the best available, given the knowledge at the time and with little regard for the long term goal, or for the basic knowledge that is embedded in previous designs. In such a strategy one may introduce technologies even though it is clear from the outset that they do not lend themselves to further advances, essentially locking-into a wrong path. For example, post combustion technology, may well be in this category, as any R&D investment into flue gas scrubbing is most likely be made obsolete by the next generation of power plants. Alternatively, the goal could be achieved by a set of incremental improvements that are introduced in each new plant or in each upgraded plant, where changes are designed to build upon each other. In this example, by suffering perhaps from a little extra cost during early adaptation of an upgraded plant, the pathway may prevent itself from locking technology advances into a wrong direction, which will incur a much bigger cost to break away from at a later time. Oxyfuel combustion designs are likely to fit into this
A consideration of the intermediate plant designs can reduce the long-term cost of power plant designs. However, a rational implementation of such an approach requires the means of making comparisons across different technologies and across different times. We propose a methodology by which we can make such an assessment. The method introduces a penalty function that can be applied to modules, plants, and sequences of plant designs. In optimizing the design, one varies design parameters so as to minimize the penalty function. The penalty function is zero for some perfect state of the system which is typically not attainable, and the penalty function is optimized by varying all the available design parameters. The penalty can be thought of as a sum of penalties for specific aspects of the plant, for example its efficiency, its cost or its environmental impact. Individual modules may have component penalties. Some aspect of the penalty will depend on properties that can only be defined for the entire plant, or even for a sequence of plants.

The relative weights of these penalties can be chosen appropriately by a user, who has specific goals. For example, the penalties that one may associate with having to build a new plant on a new site, may vary for users in different countries. Building new plants in China is likely to introduce a relatively small penalty for greenfield plants. The same decision in the West is likely to introduce a much bigger weight, because the political difficulties in opening up new sites are much larger. The different weights may result in alternative development pathways.

It is also recognized that the availability and maturity of novel technologies, as well as environmental thresholds for existing and potential criteria pollutants are likely to
change over time, thus the weights for these penalties may not be static. The nature of these penalty specifications, allow users to choose, instead of the best possible plant at a specific time, but the best possible pathway connecting technologies from the current time to a future time. Pursuing the optimal pathways on the basis of finding the minimum total pathway penalties helps users lower the cost of achieving the specific goals, even if it results in seemingly sub-optimal outcomes for individual plants.

Much of the work we present is in defining an appropriate set of penalties on which the optimization rests. The evaluation approach outlined here has taken its inspiration from the typesetting software TeX. The ranking algorithms to find the best solution among a universe of all feasible solutions are presented in Chapter 4. We will show how the ranking algorithms can be used to select appropriate modules, and power plants to arrive at a sequence of plant designs that provides an advantageous technology pathway from today’s power plant designs to a future design that has far higher efficiency, avoids all emissions to the air, and provides the CO\textsubscript{2} produced in a concentrated stream ready for disposal (Chapter 5).

2.2 Problem formulation

We can represent a module with symbol $x$. A module is an operation unit, for example, a furnace, a boiler, or a steam turbine. We can represent a power plant configuration at any time of choice with a network of $N$ modules.

$$y(t_j) = \{x_1, x_2, \ldots x_i, \ldots x_N\} \quad (2.1)$$
where $i \in [1, N]$.

If one plans to build $m$ new power plants at a rate of one new power plant per year for $m$ years, at each time of choice $t_j$, one can choose to build either the same plant as before, or a new configuration. We can further represent this sequence of $m$ power plant configurations as follows.

$$z = \{y(t_1), y(t_2), \ldots, y(t_j), \ldots, y(t_m)\}$$  \hspace{1cm} (2.2)

where $z$ represents a sequence of $m$ power plants, $t_j$ represents time, $y(t_j)$ represents the power plant configuration at time $t_j$, $j \in [1, m]$.

### 2.3 Penalties

The penalty number is the summation of all sub-penalties, where each penalty evaluates a unique aspect of a system. In this case, a system can be a single power plant design, or a sequence of power plant designs. The penalty number quantifies the "badness" of a system. A physically impossible or an illegal system that violates regulatory constraints has an infinite penalty number. Penalty functions are methods to calculate the penalty number of a design or a sequence of designs. Hierarchically, we break down the penalty into three levels. At the bottom level, the module penalty functions capture the impact of design decisions that only have effects at the module level. The penalty functions do not know the rest of the plant. At the intermediate level, the plant penalty functions capture the impact of design decisions on a plant from economic, environmental, infrastructural, and technological aspects respectively. In addition it contains the sum of all module level penalties. In designs in
which modules can be combined to sub-systems, sub-system penalties, which are like system penalties, become the module penalties of the next higher level. At the top level, pathway penalty captures the transitional costs between subsequent designs, in addition to the summation of individual plant penalties of all plants in a sequence.

Different users can customize their own penalty functions at various levels to account for particular considerations in their decision making. In Chapter 3, we develop a simple example to illustrate the general methods of making comparisons across different technologies and across different times. In that example, we design a rather complex hierarchical penalty model, to show the powerful concept that one can penalize various aspects that were not accounted for at the module level. In Chapter 5, we use the method to study the problem of optimizing various power plant designs with post-combustion capture technologies.
Chapter 3

A simple model for power plant pathway optimization

To demonstrate the application of the new evaluation method (Chapter 2) on power plant modeling and ranking, a simple model is developed to introduce the concept.

This simple model considers eight basic power plant modules. Three generation modules and five environmental modules can be combined to form three major plant designs in 32 different ways, hence 96 different power plant configurations. The generation modules include a subcritical boiler island, a supercritical boiler island, and an ultrasupercritical boiler island. (The generation modules could again be divided into a network of water cycle modules and a sequence of fuel processing modules, which could again be combined differently. However, this chapter only considers the generation unit as a whole.) The environmental modules are flue gas cleaning devices for SO$_2$, NO$_x$, fine particulates, mercury, and CO$_2$ emissions respectively. Each envi-
Chapter 3: *A simple model for power plant pathway optimization*

Environmental module can be considered as an optional add-on to the main plant designs in a binary manner. The environmental module is either added or not. The overall number of power plant configurations is $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$. However, it’s important to note that many of these configurations are not feasible in practice. For example, a selective catalytic reduction (SCR) can not be attached directly to a supercritical boiler island without adding a heat exchanger to lower the incoming flue gas temperature. In another instance, the amine CO$_2$ scrubber can not be directly added to the back of the boiler island, without pretreating the flue gas with a desulphurization unit to avoid sulfur poisoning. Infeasible configurations will be characterized with an infinite penalty. Out of the 96 possible configurations, only 18 are physically valid. For simplicity, we limit the input to our model only to the 18 power plant configurations that are physically valid.

A sequence of power plants can now be organized into a technology pathway, which moves from one type of power plant to the next. The enumeration of the 18 power plant designs in a pathway consisting of five power plants in the sequence, gives $10^6$ variations. In this Chapter, we do not explore the entire solution space. Rather, five unique pathways with different numbers of elements in the sequence, are chosen to show how various aspects of penalties affect their rankings. For example, a shorter path is favorable, but multiple changes at the same time is undesirable. For the purpose of this simple model evaluation, the initial power plant for all possible pathways is a subcritical power plant without environmental modules attached. It represents the current starting point for a technology pathway to a more advanced set of technologies. This simple model aims to rank the different pathways and power
Chapter 3: A simple model for power plant pathway optimization

plant designs, based on assigning penalty functions that scale with the deviation from a perfect state, which sets the anchor points against which various aspects in various levels of a pathway compare. The following sections describe a hierarchy of different levels of penalties, namely module level penalty, plant level penalty and pathway level penalty. Different levels of penalties characterize various performances of the corresponding level. Together, the penalties at various levels allow for the rankings of different pathways according to the accumulated penalties.

In the following sections, we identify individual penalty variables (or aspects of penalties at different penalty level), the anchor points for each penalty variable, and functions describing the behaviors of each penalty variable.

3.1 Model Formulation

3.1.1 Module level penalties

Module level penalties characterize individual module performances independently from the rest of the plant. For example, cost and size of a module are strictly independent from the rest of the modules in a plant design, thus they belong to the module level in the penalty hierarchy. In comparison, cost per unit capacity ($/kW, M$/kW) or size per unit capacity (m²/kW, m³/kW) are not module level penalty variable, because the total power output of the whole plant (MW) measures the plant level performance, rather than module level performance.

The module level penalties, their anchor points, and penalty functions for this simple model are summarized in Table 3.2. The sum of all module penalties for
the $N$ modules comprising a specific plant design $j$ represent one term in the plant level penalty sum. One can calculate this plant level penalty by adding each penalty specifications accounting for all modules in a plant design:

$$\sum_{i=1}^{N} P_i = \sum_{i=1}^{N} P_{TMI,i} + \sum_{i=1}^{N} P_{OAM,i}$$

where

$N$ is the number of all modules in a power plant design.

$P_{TMI,i} = w_{TMI} \times TMI,i$

$TMI,i$ is the Total Module Investment for module $i$;

$w_{TMI}$ is the penalty weight for TMI

$P_{OAM,i} = w_{OAM} \times OAM,i$

$OAM,i$ is the annual Operational and Maintenance cost for module $i$;

$w_{OAM}$ is the penalty weight for OAM

For the purpose of this model, we assume $w_{TMI} = w_{OAM} = 20$. One can change the penalty weight to suit one’s own preference.

There are many other penalties one could introduce on the module level. For example, the size of a module could add its own penalty. All else being equal, a smaller module appears preferable, hence penalizing size represents a strategy that would favor more compact plant designs. However, for this model, due to the lack of data on physical size of modules examined in this example, the penalty calculation for size is not included in the simple model. Data on costs and emissions as a result of mass and energy balance calculations is produced using the integrated environmental control model (IECM), the plant performance data is attached in Appendix A.2.
It is worth noting that, in this example, the cost of a module does not change as its production capacity grows. In other words, we do not consider the impact of learning on costs in this Chapter. However, later in Chapter 4 and 5, we change the assumption to account for the impact of learning on costs. This change creates a critical difference in the choice of ranking algorithms, since it significantly increases the complexity of the problem. The details of the discussion are given in Chapter 4.

3.1.2 Plant level penalty

In addition to the sum of module level penalties as described above, plant level penalties also characterize energetic and environmental performance at plant level.

Reconcile penalty

A collection of modules does not necessarily add up to a working power plant. The set of modules might be incomplete, the various modules may be incompatible in size, or in the requirements for their input and output streams. As a result a first review of a power plant design is to assure that the parameter choices that make up all the modules results in a viable plant. This process of consistency checking, and if possible parameter selection, we refer to as a reconciliation step. If the modules of a power plant cannot be reconciled into a viable power plant design, a reconcile penalty is added. This $P_{\text{rec}}$ is added to plant configurations that do not reconcile. As stated earlier, many of the 96 configurations in the model are not feasible in practice. For example, a selective catalytic reduction (SCR) can not be attached directly to a supercritical boiler island without adding a heat exchanger to lower the
incoming flue gas temperature. In another instance, the amine CO$_2$ scrubber can not be directly added to the back of the boiler island, without pretreating the flue gas with desulphurization unit to avoid sulfur poisoning. They involve contradictory input conditions. Infeasible configurations will be characterized with an infinite penalty. A fully reconciled plant configuration has a zero reconcile penalty. Since this Chapter only considers feasible configurations, all 18 feasible designs in this simple example have zero reconcile penalties. The reconcile penalty is of practical interest in large numerical searches or optimization strategies where many different plant designs are considered and where one iterates over different parameter choices for the different modules. Examples, of such searches will be discussed in subsequent chapters.

**Energy efficiency penalty**

For energy efficiencies, we penalize gross plant efficiency and generation efficiency separately. The former penalty measures gross power output as a fraction of total energy input (represented by the total potential energy embedded in the fuel); the latter penalty measures the net power output (which is gross output less internal power consumption) as a fraction of gross power output, indicating the amount of power consumed by the system itself. The need to separate the two specifications is to distinguish two kinds of plant designs with the same net plant efficiency: one design with a high gross efficiency but many energy-intensive additional units (i.e. auxiliary units and environmental control units), the other one with a low gross efficiency to start with but few energy-intensive auxiliary units. It may come out to be the same overall efficiency, but the system’s inefficiency is captured separately, thus identifying
the opportunities and incentives to reduce the inefficiency respectively. When making an one-time decision, these two configurations are the same from net efficiency point of view. It is also possible to penalize inefficiencies that occur on a single module level. However, these only refer to inputs and outputs of the individual module. The current implementations do not consider module level inefficiencies as some modules can not be characterized with efficiency. Therefore, we only consider efficiency penalties on the plant level. It is worth noting that, when making a sequence of decisions, the one with a higher gross efficiency to start with, may prove to be advantageous over the long run.

\[
P_{\eta_{\text{gross},j}} = w_{\text{gross}} \times \frac{(1 - \eta_{\text{gross},j})}{\eta_{\text{gross},j}} \tag{3.1}
\]

\[
P_{\eta_{\text{gen},j}} = w_{\text{gen}} \times \frac{(1 - \eta_{\text{gen},j})}{\eta_{\text{gen},j}} \tag{3.2}
\]

where

\( \eta_{\text{gross}} \) is the gross efficiency of the power plant

\( \eta_{\text{gen}} \) is the generation efficiency, or net efficiency of the power plant

We assume \( w_{\text{gross}} = 5000 \), and \( w_{\text{gen}} = 50000 \). The choice of relative penalty weights reflect the user’s strong preference for higher generation efficiencies over gross efficiencies. One can change the penalty weights for one’s own preferences and goals.

**Environmental penalty**

Environmental impact is penalized through a three-level environmental penalty hierarchy (Figure 3.1). The environmental penalties are evaluated as a product of regulatory factor, compliance factor and quantitative factor. We consider environmental impacts at the plant level, because emissions are a measure of plant performance.
Chapter 3: A simple model for power plant pathway optimization

For now, we assume all emissions are emitted at the smoke stack. One exception to this rule is when users are concerned with dangerous (i.e. flammable, or poisonous) chemicals flowing around a plant, or with any dangerous chemical leakage from a single operation unit in a plant. In that case, an additional environmental penalty will be placed at the module level, in an attempt to capture the cost associated with its operation, particularly the cost related environmental health and safety. There is no such module level penalty in this simple model, but it could be easily included for future optimization when required. Also, we penalize environmental impact only to the extent that it is due to atmospheric emissions for simplicity, of the model. At present, we do not penalize other environmental impacts such as solid waste discharge and water consumption, but they can be easily incorporated when needed.

![Figure 3.1: Environmental Penalty](image)

**Regulatory factor** The regulatory factor accounts for various sets of regulatory states at each position along the pathway. If there is a regulation, the factor is one, if there is none, the factor is zero. It is also possible to assign a value that is between zero and one to capture a situation where the introduction of a regulation is either
considered likely or desirable. The regulatory factor can be calculated according to the following equation.

\[
\begin{align*}
\text{reg} & = 1 & \text{if there is a regulation on a specific kind of emissions at a specific time} \\
\text{reg} & = 0 & \text{if there is no regulation on a specific kind of emissions at a specific time} \\
\text{reg} & \neq (0 \land 1) & \text{if there is no regulation on a specific kind of emissions, but the regulation is expected to materialize at a future time}
\end{align*}
\]

where the subscript M stands for various chemicals, i.e. SO$_2$, fine particulate, NO$_x$, Hg, and CO$_2$.

In this example, the regulatory factors are not uniform for all pathways, since the five pathways that we study, have different numbers of elements in the sequence. Therefore, we assume the environmental control modules are introduced only when corresponding regulations take effect. In other words, we assume there is no motivation to introduce the environmental control modules, before the emissions are regulated. The historic development of technologies in power industry agrees with this assumption. However, it’s worth noting that later in Chapter 5, we relax this assumption to characterize regulatory states differently. Our findings in Chapter 5 show that, sometimes it is beneficial to introduce the environmental control modules before the emissions are regulated. The following equation describes the regulatory
Chapter 3: A simple model for power plant pathway optimization

factors for designs with corresponding modules.

\[ w_{\text{reg}, M} \begin{cases} 
  = 1 & \text{if a specific design include an environmental control unit for } M \\
  = 0 & \text{if a specific design does not include an environmental control unit for } M 
\end{cases} \]

where M stands for pollutants, i.e. SO\(_2\), fine particulate, NO\(_x\), Hg, and CO\(_2\).

The necessity to have a top level regulatory factor is to capture differences in regulated emissions and unregulated discharge, and to ensure the corresponding technological and infrastructural change economically viable in a historical context. For example, the installation of flue gas desulphurization unit (FGD) on the existing pulverized coal power plants in US was driven by the regulation of sulfur emissions in 1990s as a result of amendments to clean air act. However the use of limestone to control SO\(_2\) emissions incurs unintended consequences of generating additional CO\(_2\) emissions\(^1\), which was considered not harmful at the time. In this situation, since no regulations on CO\(_2\) emissions were in place, the regulatory factor is zero. Since the environmental penalty is a product of regulatory factor, compliance factor and quantitative factor, the overall environmental penalty of CO\(_2\) emissions is zero for all plant configurations at that time.

Among the five unique pathways that we study, most pathways include designs that introduce new environmental control modules one at a time, which reflects that various environmental regulations come in effect one at a time. This assumption agrees with the regulatory situation in US, where the process of introducing a new module:

\[ \text{SO}_2 + \frac{1}{2} \text{O}_2 \rightarrow \text{SO}_3 \]  
\[ \text{SO}_3 + \text{CaCO}_3 \rightarrow \text{CaSO}_4 + \text{CO}_2 \]  

\(^1\)
environmental regulation, or introducing a new criteria pollutant, is long and difficult.

**Compliance factor** Compliance factor evaluates whether a specific kind of emissions are compliant with the legal limit or not. Compliance factor hinges upon the fact that there is a regulatory factor for a specific kind of emissions. The legal limit against which a compliance factor is measured, is used to normalize an output-based emission rate in this study. The limit, which for example, could be expressed as lb/kWh, differs from the traditional input-based standard that reflects an allowed amount of emissions per unit of fuel burned (lb/Btu). Output-based standards are preferable to traditional input-based standards, as they capture differences in efficiency among sources in converting input energy (e.g., heat) to useful output (e.g., electricity) and therefore reward efficient use of energy [Roy, E. 1998]. In addition, output-based standards capture emissions generated not only from the fuel source, but also from other sources that are converted to various emissions via both chemical process (i.e. reaction 3.4) and physical process (i.e. the injection of active carbon in order to control mercury emissions will introduce additional particulates emissions), and therefore penalize the overall plant emissions, instead of only part of the emissions in a plant.

If a specific kind of emissions is compliant with the regulation, then it receives a compliance factor of one, and the model proceeds to the quantitative factor calculation. If it is not compliant with the regulation, it receives an infinite overall environmental penalty. In this simple model, we penalize SO\(_2\), fine particulates, NO, NO\(_2\), Hg, and CO\(_2\) emissions gradually for different designs, assuming the corresponding regulations materialize over time. The compliance factor can be calculated
according to the following equation.

\[
\begin{align*}
    w_{\text{comp}} = & \begin{cases} 
    1 & \text{If a specific kind of emissions is compliant with the} \\
    & \text{regulation at a specific time} \\
    \infty & \text{If a specific kind of emissions is not compliant with the} \\
    & \text{regulation at a specific time} \\
    & \text{In the model, we use } 1e^{300} \text{ to represent infinity.}
    \end{cases}
\end{align*}
\]

For simplicity, we assume technologies with environmental control for certain criteria pollutants, regardless of the level of control, comply with the regulations. Technologies without environmental control for certain criteria pollutants receive an infinite penalty.

**Quantitative factor** Unlike the regulatory factor and compliance factor that examine "qualitatively" the environmental impact of various emissions, this factor examines the absolute emission quantitatively. Quantitative factors are applied if a specific emission is regulated and is compliant with the regulation, the necessity to have a quantitative factor is to penalize the absolute emissions from a plant design, thus in essence rewarding plant designs with less emissions. For quantitative factors, we design different functions to describe (penalize) the undesirability of a specific kind of emissions mathematically by scaling the emissions per unit output (lb/kWh) with a set of user-supplied scaling factors (penalty weights).

The functions are carefully designed to characterize the environmental impact of various emissions. The function for CO$_2$ is a linear function of the mass flow rate CO$_2$ at the smoke stack of a power plant (equation 3.5), because its environmental impact is approximately linear to the amount of CO$_2$ emitted. In comparison, the function for
SO$_2$ has several components as SO$_2$ emissions have several aspects of environmental impact, both locally and regionally. At local level, aerosols of sulfuric acid and other sulfates contribute significantly in the reduction of visibility and damage to material, it has also been recognized that there are some localized areas where asthmatics may be repeatedly exposed to short-term SO$_2$ concentrations [Wark, K., 1997]. These local environmental impacts are not linear in the SO$_2$ concentration, rather they show a much greater impact at higher concentrations. For simplicity, we approximate penalties attributed to the local impact as quadratic function to the concentration of SO$_2$ emissions at the smoke stack of a plant. We choose the quadratic power, rather than cubic or any other power, as a first approximation. The functions might change as we gain better scientific understanding, and collect more data for the environmental impact of the local SO$_2$ emissions. At regional level, SO$_2$ can be transported a long distance by air masses and then be precipitated in the form of acid rain somewhere else. The function attributed to the regional impact of SO$_2$ emissions is chosen to be linear in the SO$_2$ concentration at the smoke stack, because the amount of acid rain is linear to the amount of SO$_2$ emissions. The overall quantitative factor for SO$_2$ emissions can be calculated according to equation 3.6.

\[
\text{w}_{\text{quan,CO}_2} = k_{\text{CO}_2} \times (m_{\text{CO}_2}) \quad \text{where} \quad m_{\text{CO}_2} = \frac{\text{lbCO}_2}{\text{kWh}} \quad (3.5)
\]

\[
\text{w}_{\text{quan,SO}_2} = k_{\text{SO}_2,g} \times (m_{\text{SO}_2}) + k_{\text{SO}_2,l} \times (m_{\text{SO}_2})^2 \quad \text{where} \quad m_{\text{SO}_2} = \frac{\text{lbSO}_2}{\text{kWh}} \quad (3.6)
\]

where the subscript $g$ stands for global, and $l$ stands for local.

NO$_x$ emissions like SO$_2$ emissions, also contribute to acid rain formation, in addition to its local impact, thus it is penalized both locally and regionally. Regard-
ing particulates, we exclusively focused on their local impact. Mercury emissions have a more complex environmental impact, as elemental mercury emissions can be transported and deposited in the watershed, and form methylmercury. Harris and Rudd predicted that mercury emissions reductions will yield repaid reductions in fish methylmercury concentrations and will yield concomitant reductions in risk [Harris, R. C., Rudd, J.W.M. 2007]. Hence, we approximate the environmental impact of mercury emissions as a linear function of the mass flow rate of the mercury emissions from the power plants. Functions to calculate quantitative factors for NO\(_x\) and Hg are listed as follows.

\[
    w_{\text{quan}, \text{NO}_2} = k_{\text{NO}_2,g} \cdot (m_{\text{NO}_2}) + k_{\text{NO}_2,l} \cdot (m_{\text{NO}_2})^2 \quad \text{where} \quad m_{\text{NO}_2} = \frac{\text{lb NO}_2}{\text{kWh}} \quad (3.7)
\]

\[
    w_{\text{quan}, \text{NO}} = k_{\text{NO},g} \cdot (m_{\text{NO}}) + k_{\text{NO},l} \cdot (m_{\text{NO}})^2 \quad \text{where} \quad m_{\text{NO}} = \frac{\text{lb NO}}{\text{kWh}} \quad (3.8)
\]

\[
    w_{\text{quan}, \text{Hg}} = k_{\text{Hg}} \cdot (m_{\text{Hg}}) \quad \text{where} \quad m_{\text{Hg}} = \frac{\text{lb Hg}}{\text{kWh}} \quad (3.9)
\]

Regarding fine particulate matters (PM), current research show that increasing levels of fine particulates in the air are linked to health hazards such as heart disease, altered lung function and lung cancer. For simplicity, we do not classify the fine particulates with respect to size (i.e. PM\(_{10}\), PM\(_{2.5}\), etc), rather we treat all particulate matters as one big category. We approximate penalties attributed to fine particulate matters as quadratic function to the concentration of PM emissions at the smoke stack of a plant. We choose the quadratic power, rather than cubic or any other power, as a first approximation to account for its dangerous impact on human health. The functions might change as we gain better scientific understanding, and collect
more data for the environmental impact of the particulate emissions. The function
to calculate quantitative factors for PM is listed as follows.

\[ w_{\text{quan,PM}} = k_{\text{PM}} \times (m_{\text{PM}})^2 \]

where \( m_{\text{PM}} = \frac{\text{lbPM}}{\text{kWh}} \) (3.10)

The relative weights \( k_{\text{CO2}}, k_{\text{SO2,l}}, ..., k_{\text{PM}} \) are user-supplied penalty weights for
different emissions. Different penalty functions for different emissions show different
sensitivities to user-input weights. The relative weights depend on what users expect
as consequences. For example, in the case of \( \text{SO}_2 \) emissions, if users are mainly
cconcerned about local problems, then the penalty is dominated by the quadratic
term, if the users are mainly concern about the long-distance problems, it’s driven
by the linear term. In essence these penalties are transferable, users that are less
concerned about the long-distance problems may build extremely tall smokestacks
hence transferring local sulfur pollution problems into regional sulfur problems. A
good example is the famous Inco Superstack in Sudbury, Ontario. With a height of
380 meters, it is the tallest chimney in Canada and the Western hemisphere, and was
built to disperse sulphur gases and other by-products of the smelting process away
from the city itself to improve local environment.

In this simple example, the values of penalty weights are given as follows.

\( k_{\text{CO2}} = 1 \times 10^5 \),
\( k_{\text{SO2,l}} = k_{\text{SO2,g}} = 1 \times 10^3 \),
\( k_{\text{NO,g}} = k_{\text{NO,l}} = 1 \times 10^4 \),
\( k_{\text{NO2,g}} = k_{\text{NO2,l}} = 2 \times 10^3 \),
\( k_{\text{Hg}} = 1 \times 10^9 \),
\( k_{\text{PM}} = 1 \times 10^5 \).
This set of penalty weights reflect a user who mainly concerns about mercury emissions, CO$_2$ emissions, and particulates emissions, and cares equally about local impact and regional impact of emissions for NO$_x$ and SO$_2$. Given a different user with different preferences, the values of the penalty weights may be different.

The sum of environmental penalty for a specific emission can be calculated by the following equation.

$$ P_{E_M} = w_{regM} \times w_{compM} \times w_{quanM} $$  \hspace{1cm} (3.11)

where

where $M$ stands for pollutants, i.e. SO$_2$, fine particulate, NO$_x$, Hg, and CO$_2$.

The sum of all plant level penalty specifications for a plant design adds up to the total penalty for that plant at time $t$, where $t$ refers to the time in the sequence, $t = 0, 1, 2...m$. $m$ is number of elements in the sequence. Note that the five different pathways we study have different numbers of elements in the sequence.

$$ P_{tot,t} = \sum_{i=1}^{m} P_{i,t} 
+ P_{rec,t} 
+ P_{\eta(gross,t)} + P_{\eta(gen,t)} 
+ P_{E(CO_2,t)} + P_{E(SO_2,t)} + P_{E(NO_1,t)} + P_{E(NO_2,t)} + P_{E(Hg,t)} + P_{E(PM,t)} $$  \hspace{1cm} (3.12)

### 3.1.3 Pathway level penalty

Mathematically, there are $N^m$ possible pathways, assuming there are $m$ elements in the sequence, and each element can be chosen from $N$ power plant configurations.
Chapter 3: A simple model for power plant pathway optimization

For a sequence of 5 decisions where each decision has 96 choices, the total number of all possible pathways is $96^5 = 8.15e9$. We can implement ranking algorithms such that the computer will examine all possible pathways and choose the best one. The details of the computational algorithms are given in Chapter 4. In Chapter 5, we use such algorithms to solve a real life problem of optimizing post-combustion capture systems. In this chapter, instead of exploring the entire solution space, we focus on five unique pathways with different numbers of elements in the sequence. This section introduces various aspects of pathway level penalties.

To compare and calculate penalties for different pathways, we consider four aspects of penalties. Firstly, when a power plant introduces a new module (or modules) that has not been used in previous designs along the same pathway, it has to be penalized by a factor of $w_{\text{new}} \times N_{\text{new}}^2$, where $N_{\text{new}}$ is the number of new modules, and $w_{\text{new}}$ is the penalty weight. Secondly, when a power plant removes an existing technology (or technologies) from an existing design, it has to be penalized by a factor of $w_{\text{delete}} \times N_{\text{delete}}$, where $N_{\text{delete}}$ is the number of technologies removed despite the initial cost to introduce it at a previous time, and $w_{\text{delete}}$ is a constant. Thirdly, when the new module introduced is a new technology (or technologies) that has not been widely commercialized, it has to be penalized by a factor of $w_{\text{invent}} \times N_{\text{invent}}^2$, where $N_{\text{invent}}$ is the number of new technologies. Specific to the simple model, and $w_{\text{invent}}$ is a constant. In this example, we assume the ultra-supercritical boiler island and the CO$_2$ scrubber are the only new technologies among the nine modules considered. We design the penalty factors for the introduction of new technologies (or new inventions) and the introduction of new modules with quadratic functions ($w_{\text{invent}} \times N_{\text{invent}}^2$ and
Chapter 3: A simple model for power plant pathway optimization

For the penalty factor for technology removal, we use a linear function ($w_{\text{delete}} \times N_{\text{delete}}$). We do so to discourage multiple changes at once, and we assume it is easier to remove an existing technology, than to introduce a new unit or new technologies. The summation of these factors multiplied with their corresponding weights (assigned by users) form the total pathway factor $w^k_j$ for a plant design $j$ and pathway $k$.

$$w^k_j = 0.0001 \times (w_0 + w_{\text{new}} \times (N_{\text{new},j}^k)^2 + w_{\text{delete}} \times (N_{\text{delete},j}^k) + w_{\text{invent}} \times (N_{\text{invent},j}^k)^2) \quad (3.13)$$

where

- $w_0 = 100$,
- $w_{\text{new}} = 50$,
- $w_{\text{delete}} = 20$,
- $w_{\text{invent}} = 10$.

One can change the penalty weights for one’s own preferences and goals. The total penalty for each pathway is calculated according to the following equation:

$$P_{\text{tot},k} = \sum_{j=1}^{n} (w^k_j \times P_{\text{tot},j}) \quad (3.14)$$

Pursuing the best pathways on the basis of the minimum total pathway penalties helps users lower the cost of achieving the specific goals, even if it results in seemingly sub-optimal outcomes for individual plants.

In Table 3.2, we summarize the categories of penalties at module level, plant level and pathway level. We also give the penalty variables for each category, penalty functions, and anchor points against which penalty variables are measured.
Table 3.1: Summary of power plant designs and pathways designs

<table>
<thead>
<tr>
<th>$y_0$</th>
<th>${x_1, x_9}$</th>
<th>subcritical pulverized coal plant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>${x_2, x_9}$</td>
<td>supercritical pulverized coal plant</td>
</tr>
<tr>
<td>$y_2$</td>
<td>${x_3, x_9}$</td>
<td>ultra-supercritical pulverized coal plant</td>
</tr>
<tr>
<td>$y_3$</td>
<td>${x_1, x_4, x_9}$</td>
<td>subcritical plant with SOx</td>
</tr>
<tr>
<td>$y_4$</td>
<td>${x_1, x_4, x_5, x_9}$</td>
<td>subcritical plant with SOx and particulate control</td>
</tr>
<tr>
<td>$y_5$</td>
<td>${x_1, x_4, x_5, x_6, x_9}$</td>
<td>subcritical power plant with SOx, particulate and NOx control</td>
</tr>
<tr>
<td>$y_6$</td>
<td>${x_1, x_4, x_5, x_6, x_7, x_9}$</td>
<td>subcritical power plant with SOx, particulate, NOx and mercury control</td>
</tr>
<tr>
<td>$y_7$</td>
<td>${x_1, x_4, x_5, x_6, x_7, x_8, x_9}$</td>
<td>supercritical power plant with SOx, particulate, NOx, mercury and CO$_2$ control</td>
</tr>
<tr>
<td>$y_8$</td>
<td>${x_2, x_4, x_9}$</td>
<td>supercritical plant with SOx control</td>
</tr>
<tr>
<td>$y_9$</td>
<td>${x_2, x_4, x_5, x_9}$</td>
<td>supercritical plant with SOx and particulate control</td>
</tr>
<tr>
<td>$y_{10}$</td>
<td>${x_2, x_4, x_5, x_6, x_9}$</td>
<td>supercritical power plant with SOx, particulate and NOx control</td>
</tr>
<tr>
<td>$y_{11}$</td>
<td>${x_2, x_4, x_5, x_6, x_7, x_9}$</td>
<td>supercritical power plant with SOx, particulate, NOx and mercury control</td>
</tr>
<tr>
<td>$y_{12}$</td>
<td>${x_2, x_4, x_5, x_6, x_7, x_8, x_9}$</td>
<td>supercritical power plant with SOx, particulate, NOx, mercury and CO$_2$ control</td>
</tr>
<tr>
<td>$y_{13}$</td>
<td>${x_3, x_4, x_9}$</td>
<td>USC power plant with SOx control</td>
</tr>
<tr>
<td>$y_{14}$</td>
<td>${x_3, x_4, x_5, x_9}$</td>
<td>USC with SOx and particulate control</td>
</tr>
<tr>
<td>$y_{15}$</td>
<td>${x_3, x_4, x_5, x_6, x_9}$</td>
<td>USC power plant with SOx, particulate and NOx control</td>
</tr>
<tr>
<td>$y_{16}$</td>
<td>${x_3, x_4, x_5, x_6, x_7, x_9}$</td>
<td>USC power plant with SOx, particulate, NOx and mercury control</td>
</tr>
<tr>
<td>$y_{17}$</td>
<td>${x_3, x_4, x_5, x_6, x_7, x_8, x_9}$</td>
<td>USC power plant with SOx, particulate, NOx, mercury and CO$_2$ control</td>
</tr>
<tr>
<td>$z_1$</td>
<td>${y_0, y_3, y_4, y_5, y_6, y_7}$</td>
<td>pathway 1</td>
</tr>
<tr>
<td>$z_2$</td>
<td>${y_0, y_1, y_8, y_9, y_{10}, y_{11}, y_{12}}$</td>
<td>pathway 2</td>
</tr>
<tr>
<td>$z_3$</td>
<td>${y_0, y_2, y_{13}, y_{14}, y_{15}, y_{16}, y_{17}}$</td>
<td>pathway 3</td>
</tr>
<tr>
<td>$z_4$</td>
<td>${y_0, y_{17}}$</td>
<td>pathway 4</td>
</tr>
<tr>
<td>$z_5$</td>
<td>${y_0, y_2, y_{13}, y_{14}, y_{15}, y_{16}, y_{(16,tn)}}$</td>
<td>pathway 5</td>
</tr>
</tbody>
</table>

$x_1$ = subcritical boiler island; $x_2$ = supercritical boiler island; $x_3$ = ultra-supercritical boiler island; $x_4$ = wet fluid gas desulphurization unit (FGD); $x_5$ = electrostatic precipitator (ESP); $x_6$ = selective catalytic reduction unit (SCR); $x_7$ = mercury removal unit (with active carbon injection ACI); $x_8$ = amine scrubber (MEA); $x_9$ = smoke stack.

Notes: USC stands for ultra-supercritical
Table 3.2: Module level, plant level and pathway level penalties

<table>
<thead>
<tr>
<th>Penalties</th>
<th>Penalty Variable</th>
<th>Anchor</th>
<th>Penalty functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>module level</td>
<td>TPI(M$/yr)</td>
<td>TPI\textsubscript{0} = 0</td>
<td>P_{\text{TPI},i} = w_{\text{TPI}} \times TPI_{i}</td>
</tr>
<tr>
<td>OAM(M$/yr)</td>
<td>OAM\textsubscript{0} = 0</td>
<td>P_{\text{OAM},i} = w_{\text{OAM}} \times OAM_{i}</td>
<td></td>
</tr>
<tr>
<td>plant level</td>
<td>$\eta_{\text{gross},i} = \frac{\text{MW}_{\text{gross},i}}{\text{HHV}}$</td>
<td>$\eta_{\text{gross},0} = 1$</td>
<td>$P_{\eta_{\text{gross},i}} = w_{\text{gross}} \times \frac{(1 - \eta_{\text{gross},i})}{\eta_{\text{gross},i}}$</td>
</tr>
<tr>
<td></td>
<td>$\eta_{\text{gen},i} = \frac{\text{MW}<em>{\text{net},i}}{\text{MW}</em>{\text{gross},i}}$</td>
<td>$\eta_{\text{gen},0} = 1$</td>
<td>$P_{\eta_{\text{gen},i}} = w_{\text{gen}} \times \frac{(1 - \eta_{\text{gen},i})}{\eta_{\text{gen},i}}$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{CO}_2,i} = \frac{\text{CO}_2 \text{ emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}_{\text{CO}<em>2,i}} = k</em>{\text{CO}<em>2} \times m</em>{\text{CO}_2,i}$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{SO}_2,i} = \frac{\text{SO}_2 \text{ emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}_{\text{SO}<em>2,i}} = k</em>{\text{SO}<em>2,g} \times m</em>{\text{SO}<em>2,i} + k</em>{\text{SO}<em>2,l} \times m</em>{\text{SO}_2,i}^2$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{NO},i} = \frac{\text{NO emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}<em>{\text{NO},i}} = k</em>{\text{NO}<em>2,g} \times m</em>{\text{NO},i} + k_{\text{NO}<em>2,l} \times m</em>{\text{NO},i}^2$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{NO}_2,i} = \frac{\text{NO}_2 \text{ emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}_{\text{NO}<em>2,i}} = k</em>{\text{NO}<em>2,g} \times m</em>{\text{NO}<em>2,i} + k</em>{\text{NO}<em>2,l} \times m</em>{\text{NO}_2,i}^2$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{Hg},i} = \frac{\text{Hg emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}<em>{\text{Hg},i}} = k</em>{\text{Hg}} \times m_{\text{Hg},i}$</td>
</tr>
<tr>
<td></td>
<td>$m_{\text{PM},i} = \frac{\text{PM emissions from smokestack}}{\text{Net Power Output kWh}}$</td>
<td></td>
<td>$P_{\text{qu}<em>{\text{PM},i}} = k</em>{\text{PM}} \times m_{\text{PM},i}^2$</td>
</tr>
<tr>
<td></td>
<td>$M = \text{SO}_2, \text{PM}, \text{NO}, \text{NO}_2, \text{Hg}, \text{CO}_2$</td>
<td></td>
<td>$P_{E,i} = P_{\text{reg}<em>{E,i}} \times P</em>{\text{comp}<em>{E,i}} \times P</em>{\text{qu}_{E,i}}$</td>
</tr>
<tr>
<td></td>
<td>$P_{E,i} = 0$</td>
<td></td>
<td>$P_{E,i} = \sum P_{E,i}$</td>
</tr>
</tbody>
</table>

Notes:
1. all penalty weights to be supplied by users. One can change the penalty weights for their own preferences and goals
2. In this model, the author makes assumptions of the weights as below.
   
   $w_{\text{TPI}} = 20, w_{\text{OAM}} = 20, w_{\text{gross}} = 5000, w_{\text{gen}} = 50000, k_{\text{CO}_2} = 1e5 ; k_{\text{SO}_2,1} = k_{\text{SO}_2,g} = 1e3 ;
   k_{\text{NO}_2,g} = k_{\text{NO}_2,l} = 1e4; k_{\text{NO}_2,2,g} = k_{\text{NO}_2,2,l} = 2e3; k_{\text{Hg}} = 1e9; k_{\text{PM}} = 1e5;
   w_0 = 100, w_{\text{new}} = 50, w_{\text{delete}} = 20, w_{\text{invent}} = 10$
3. The calculation of material and energy balance and costs are carried out using the IECM model.
4. i represent the number of modules, j represents power plants, k represent pathways.
3.2 Model Assumptions

Despite of different numbers of elements in the sequences, we consider that the regulatory states for all pathways on the first and the last elements are uniform for all pathways. The first element of all pathways is the same (Fig 3.2), and it faces no environmental regulation. The last elements of all pathways are different, and we consider two cases.

- Case 1, we consider that $\text{SO}_2$, PM, $\text{NO}_x$, mercury, and $\text{CO}_2$ emissions are all regulated on the last elements.

- Case 2, $\text{SO}_2$, PM, $\text{NO}_x$ and mercury emissions are regulated, but $\text{CO}_2$ emissions is not.

The five pathways are illustrated in Figure 3.2. In pathway $z_5$, we include design Figure 3.2: Five pathways of different numbers of sequence elements

$y_{16}$ twice, we denote the second one as $y_{16,tn}$. In case 1, we expect the initial regulated $\text{CO}_2$ standards to be very low for practical purpose. By adapting a higher efficiency unit, which is ultra-supercritical generation unit, design $y_{16,tn}$ complies with the $\text{CO}_2$ regulation, despite of the lack of $\text{CO}_2$ emission control unit.
Chapter 3: A simple model for power plant pathway optimization

It is worth to note that, pathway z3 and z5 are very similar, the only difference is the last element in the sequence. In z3, the design choice on the last element consists of all environmental control modules, including the CO₂ emission control unit. In z5, the design choice on the last element does not include the CO₂ emission control unit, but does include all other environmental control modules. We expect to observe the opposite relative penalties of these two pathways in the two cases.

3.3 Results and Discussion

Figure 3.3 and Figure 3.4 show the total pathway penalties of case 1 and case 2 respectively.

In case 1, all emissions are regulated on the last elements of the sequence, including CO₂ emissions. Under this situation, pathway z₃ compared to z₅ results in a lower penalty. This is due to the fact that, despite that the CO₂ control module is expensive, the quantitative penalty to emit extra amount of CO₂ is even higher. Hence the cost of the new expensive CO₂ control unit is paid off in pathway z₃. In comparison, by
avoiding the cost in introducing the new expensive module, pathway $z_5$ still have to pay for the draconian cost in emitting extra amount of CO$_2$, hence it has a higher penalty.

In case 2, all emissions are regulated on the last elements of the sequence, except for CO$_2$ emissions. Under this situation, pathway $z_3$ compared to $z_5$ results in a higher penalty. In this case, the cost of introducing new technology in $z_3$ is not paid off, due to the lack of regulations in CO$_2$ emissions.

Figure 3.5: Case 1: Plant level penalties with CO$_2$ regulations

Figure 3.6: Case 2: Plant level penalties without CO$_2$ regulations
Figure 3.5 and 3.6 show the breakdown of plant level penalty in Case 1 and Case 2, reflecting author’s concerns on environmental impact more than on the cost and efficiency.

Figure 3.7: Case 1: Environmental penalties with CO$_2$ regulations

Figure 3.7 and 3.8 show the breakdown of environmental penalties in Case 1 and Case 2 respectively. The legend P$_E$ _M stands for the environmental penalty of specific emissions, where M = CO$_2$, SO$_2$, NO$_x$, PM, Hg.

Our first observation is that, in Case 1(Fig 3.7), the environmental penalty of design $y_{16,tn}$ is far greater than other designs. This is because, $y_{16,tn}$ complies with the environmental regulations of CO$_2$ by merely adopting to a higher efficiency design, such that less fuel is burned and less CO$_2$ is produced by kWh of electricity. However, since we penalize the absolute amount of CO$_2$ emissions, uncaptured CO$_2$ with this design is significant, hence the draconian environmental penalty.

Secondly, note $y_{16}$ and $y_{16,tn}$ are exactly the same designs built in different times. In Case 1 (Fig 3.7), when there is a CO$_2$ regulation, $y_{16,tn}$ receives a significant
environmental penalty on CO\textsubscript{2} emissions compared to y\textsubscript{16}. However, in Case 2 (Fig 3.8), when there’s no CO\textsubscript{2} regulation, y\textsubscript{16,tn} has the same environmental penalties as y\textsubscript{16}.

Figure 3.8: Case 2: Environmental penalties without CO\textsubscript{2} regulations

Another interesting observation can be found both in the environmental penalty results illustrated in Fig 3.7 and in Fig 3.8. When power plants introduce CO\textsubscript{2} scrubber, the overall environmental penalties are smaller than previous designs on the same pathways, for example y\textsubscript{7}, y\textsubscript{12}, y\textsubscript{17}. This is because MEA sorbents are easily poisoned by impurities in the flue gas, the inlet concentration of SO\textsubscript{2} emissions is required to be controlled at a very low level when CO\textsubscript{2} scrubber is installed, hence the overall environmental penalty for plant designs with CO\textsubscript{2} scrubber is much lower than other designs. However, this model does not take into account the environmental impact of MEA leakage from the reboiler, if the users consider the environmental impact of MEA leakage, the overall environmental penalties of plant designs with CO\textsubscript{2} scrubber may not have a lower value.

In this Chapter, we demonstrated the application of the new evaluation method
inspired by TeX (Chapter 2) on a simple power plant model. We manually selected five unique technology pathways among a large solution space, and applied penalty functions to evaluate their performances on module level, plant level, and pathway level. This model was not intended to show the best possible pathway, since we were not looking at the entire solution space. Rather, it was developed to show how various properties that characterize the system, can impact the penalties on different levels, as well as their relative rankings.

To find the optimal pathway, we need to explore the entire solution space explicitly or implicitly. It is worth to note that in this chapter, the cost of a module does not change as the number of units produced increases. In other words, we do not consider the impact of learning on costs in this simple model. However, we change this assumption to account for the impact of learning in future chapters. This change creates a critical difference in the choice of ranking algorithms, since it significantly increases the complexity of the problem. In the following chapters, we develop two algorithms which find the optimal solution in reasonable time (Chapter 4), apply the algorithms to solve a real life problem (Chapter 5), and develop a software which enables the modeling and optimization exercise (Chapter 6).
Chapter 4

Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

Once we consider the cost of a power plant, or a power plant module to depend on the amount of experience the manufacturer and operator has with this particular module, the objective function which needs to be minimized for a technology will depend not only on the individual plants in the chosen pathway, but on the entire history of the pathway. Without such a path dependence one could map the problem on that of searching for a shortest path (the one with the minimum penalty), now the problem is more difficult as in the equivalent shortest path problem, the length of the path between two points will depend on the choice of the path taken at earlier steps. We propose two combinatorial optimization algorithms for multi-variate technology designs with path dependent objective functions. The objective function is defined
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

as the minimum of a sum of penalties. The problem is solved exactly by means of a branch-and-bound method, and approximated via a heuristic, which is based on the label-correcting algorithm for solving a shortest-path problem. The proposed algorithms are applied for practical problems on finding the optimal sequence of various power plant designs.

4.1 Problem statement

Let us consider each power plant design in a temporal sequence as a vertex in a path. A sequence of \( m \) power plant designs with a given initial design, can be represented by a path of \( m \) vertexes with a given source vertex \( S \). Each pair of connected vertexes is labeled with a distance number, which measures the distance between the two vertexes. The path distance is the summation of all distance numbers along the path. Thus, the problem of finding the optimal power plant pathway with the lowest penalty can be considered the same as the problem of finding the optimal path with the minimum path distance.

Given a source vertex \( S \) and a sequence of \( m \) decision points, there is a pool of \( n \) available power plant designs from which only one design is chosen at each decision point. The same design can be chosen more than once in a path. All possible paths start with a shared source vertex \( S \), and all paths have exactly \( m \) vertexes. The enumeration of all possible paths forms a rooted tree. The first level for all paths is the root \( S \). Except for the root, each vertex in the path has a parent, which is the vertex immediately before it on the same path. The vertexes on the last level of the tree are called leaves. A vertex that is neither the root nor a leave, can be viewed as
a subroot. A subtree is a smaller tree originating from a subroot, but with the same tree structure.

Each path can be viewed as a branch of the tree, a path from the root to a subroot is named as a subbranch. Note in this discussion, a subbranch always begins at the root. The whole tree has $n^m$ branches. On each level $i \ (1 \leq i \leq m)$, there are $n^i$ vertexes. As an example, Fig 4.1 illustrates an instance of a rooted tree of four levels, the root is $S$, three designs $d_1, d_2, d_3$ are chosen repeatedly, forming a total of twenty-seven possible branches. To distinguish the same design choices on different levels, each level of the tree is indexed with a time, i.e, $t_1, t_2, t_3,$ and $t_4$.

Figure 4.1: An instance of a rooted tree

The arc distance is characterized by a numerical value. The arc distance measures the direct distance between the two ends, i.e., the distance between the two without intervening intermediate nodes. The arc distance between two plant designs can depend on the location of the two members in the tree. For example in the hypothetical example laid out in Fig. 4.1, the distance of arc $S(t_1)\rightarrow d_1(t_2)$ is 20, the
distance of arc $d_1(t_2) \rightarrow d_1(t_3)$ is 15. Each arc distance is non-negative, and more importantly, it is path-dependent. It depends not only on the pair of vertexes on both ends of the arc, but also on its location in the rooted tree. However, this dependence is not as general as it could be. In the model we consider here, the arc distance between two designs depends on all members of the path that are prior to this arc. The arc distance does not depend on choices further down in the branch, nor does it depend on other branches. For example, consider path $S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3) \rightarrow d_1(t_4)$ and path $S(t_1) \rightarrow d_2(t_2) \rightarrow d_1(t_3) \rightarrow d_1(t_4)$ in Figure 4.1. Both paths include arc $d_1(t_3) \rightarrow d_1(t_4)$ on the last level, however the arc distances are different, one being 2, the other being 14. This is a result of different path histories.

In the case of power plant technology development, two different technology pathways may coincide in some choices, but the chosen technologies may have different costs in different paths. The cost depends on previous choices, but not future choices. In other words, the cost of current decisions depends on the history of past decisions, but past decisions are sunk cost. Once a dominant technology is chosen, it is natural that one will choose new technologies which are compatible with the existing one. For example, the owner of an Apple computer is more likely to choose compatible Apple products when new electronic devices emerge (i.e. iphone). Additionally, the cost of technology goes down as we gain experience in building it [Arrow, Kenneth J. 1962]. For example, the cost of personal computers lowered significantly in the last 30 years, as the production increased. As a result, technologies and components of technologies, that have been chosen in the path over and over again, are less costly than the same technologies in a path proceeded by various unrelated choices.
The total path distance is the summation of all arc distances in a path. For example, the total distance of path $S(t1) \rightarrow d1(t2) \rightarrow d1(t3) \rightarrow d1(t4)$ is $20 + 15 + 2 = 37$.

The goal of this work is to find the path with the minimum total distance number.

## 4.2 Mathematical formulation

We define our problem with the same mathematical formulation as the problem defined by Tan and Leong [Tan, J. 2004].

Let $G=(N, A)$ be a simple directed graph, where $N = \{v_1, v_2 \ldots v_n\}$ is a finite set of nodes and $A = \{a_1, a_2 \ldots a_m\} \subseteq N \times N$ is a finite set of arcs. Let $s$ and $t$ ($s \neq t$) be any two nodes of $G$. A sequence from $s$ to $t$ in $G$ of alternating nodes and arcs, of the form $(s = v_1, a_1, v_2, a_2, \ldots, a_{k-1}, v_k = t)$ where $v_k \in N, a_{k-1} \in A$ and $a_{k-1} = (v_{k-1}, v_k)$, is called a path from $s$ to $t$. An instance of a directed graph is illustrated in Figure 4.2. On each level of the graph, there are exactly $n$ nodes, and each node represents a unique design choice. Different paths are represented by connecting the nodes on each level with directed arc. Each path has $m$ arcs. An instance of a directed graph is shown in Figure 4.2.

We use $P_{st}$ to denote a particular path from $s$ to $t$ and write $P_{st} = v_1 - v_2 - \ldots - v_k$ or $P_{st} = a_1 - a_2 - \ldots - a_{k-1}$ interchangeably. If the path $P_{st}$ contains a particular arc $(i,j)$, we denote it as $(i, j) \in P_{st}$; a path $P$ is said to be a subpath of $P_{st}$ if $P_{st}$ can be written of the form $P_{st} = P_1 - P - P_2$, where $P_1$, $P_2$ can be empty paths; we denote it as $P \in P_{st}$. The length of path $P_{st}$, $|P_{st}|$ is the number of arcs contained in it. Furthermore, we use $Q$ to denote the set of paths in $G$, $Q_{st}$ from $s$ to $t$.

Consider a graph $G=(N, A)$, for any arc $(i, j)$ and any path $P$, let $c : A \times Q \rightarrow R$.
be a function which assigns a value \( c((i, j), P) \) to an (arc, path) pair. We also define \( \text{cost}: Q \rightarrow \mathbb{R} \) to be a function that associates a cost \( \text{cost}(P) \) to each path \( P \).

\[
\min \text{cost}(P_{st})
\]  

(4.1)

\[
\text{cost}(P_{st}) = \sum_{(i,j) \in P_{st}} c((i, j), P_{si})
\]  

(4.2)

where \( P_{st} \) is a full directed path in the tree, \( s \) is the root of the tree, \( t \) is a leaf on the tree. \( P_{si} \) is the path from \( s \) to \( i \). For any arc \( (i, j) \) where \( i < j \), and any path \( P \), let \( c \) be a function which assigns a value \( c((i, j), P) \) to an (arc, path) pair. \( c((i, j), P) \) is non-negative. In the case of the power plant pathway optimization, the cost of the path refers to the pathway penalty of a sequence of power plant designs.

The problem of finding the shortest path in \( G \) is known as path-dependent shortest-path search. The decision version of the problem is defined as \( \text{PDSP} = \{ <G, c, s, t, k> \}: \) there exists a path \( P_{st} \) in \( G \), s.t. \( \text{cost}(P_{st}) \leq k \).

In the examples in Section 4.8 and Chapter 5, the function \( c \) refers to the penalty functions. Note the domain of the function \( c \) is exponential in the size of \( G \), therefore
it may need exponential space even to describe the input function.

4.3 Literature review on shortest-path problem

Although the problem of finding the shortest path in a graph has been studied extensively, the effort of most research has been focusing on non-path-dependent problems [Ahuja, R. K. 1993]. Mathematical procedures to solve this set of problems often take advantage of the principle of dynamic programming, which states if a path is the shortest, then its subpath is also the shortest [Bellman, Richard 1957].

There are two general procedures to solve the non-path-dependent shortest-path problems. Both algorithms maintain a distance label of a node, and iteratively update the distance label, until finding the shortest path. One procedure is the label-setting algorithm, which designates one label as permanent at each iteration (Dijkstra’s algorithm [Dijkstra, E. W. 1959]). The other is label-correcting algorithm, which considers all labels temporary until the final step, when they all become permanent [Bellman, Richard 1957]. Both algorithms require that the distance label of a current node depends only on the previous node, rather than the entire history.

When taking path-dependency into account, the problem becomes much more complicated to solve. Tan and Leong showed that the path-dependent shortest path in a graph, in general is NP-complete\(^1\), whereas its special case can be solved by any

---

\(^1\)NP-complete represent the complexity of a problem in computer science. The non-deterministic Turning machine is an imaginary machine that is like no machine anyone has ever seen. It is allowed to make guesses, and take branches based on those guesses. Given a sequence of guesses, the machine may halt after some number of steps. The running time of the non-deterministic Turning machine is the minimum number of steps executed for any sequence of guesses. The set NP is the set of all problems that can be solved in polynomial time on a non-deterministic Turning machine. The theory of NP-complete was established in the late 1960s when S. Cook [Cook, S.A, 1971] showed that the
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

shortest path procedure in polynomial time. The special case assumes that the cost does not depend on the entire path, but only the last suffix-k paths, therefore it can take advantage of the last k nodes [Tan, J. 2004].

Unlike the partial path-dependent problem that Tan and Leong described, our problem is fully path-dependent, which cannot be transformed to a special case, therefore it cannot be solved by the shortest path procedure.

4.4 Path-dependent shortest-path algorithms

To find the true optimum, we can use a brute-force search approach to examine every single path, and compare the distances of all possible paths. In total, a tree that enumerates all the paths in a graph has $n^m$ branches, the runtime of the brute-force search approach is $O(n^m)$. This is very computationally expensive, especially for a big tree.

To improve on this algorithm, we can design efficient pruning procedures by removing the subbranches that incur a penalty that exceeds a bound, which is set by the best known path. Furthermore, we can tighten the bound by continuously updating the current best path to improve the efficiency even more. When the cur-

---

satisfiability problem for propositional calculus is least as hard as any problem in NP, in the sense that if there existed an efficient algorithm for the satisfiability problem, then any problem in NP would be solvable by an efficient algorithm. A problem that shares this property with satisfiability is called NP-hard, and if it is also in NP, it is called NP-complete. Since no efficient way of solving the satisfiability problem, or any NP-complete problem, has ever been found, an NP-complete problem is normally considered to be intractable.[Plass, M. F., 1981]

runtime refers to the time during which a program is running (executing). The symbol $O()$ means "on the order of". It suggests that the overall effort required is proportional to the argument given. But two different algorithms of the same order could differ substantially, in their total time requirement.
rent best has an infinite penalty, we do not gain anything from pruning, as we need to explore essentially the entire space of solutions. When the current best path is the true optimum, one would prune the tree most efficiently. This method is called branch-and-bound method. It is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization [Ahuja, R. K. 1993]. The original branch and bound algorithm is proposed by A. H. Land and A. G. Doig in 1960 for linear programming [Land, A.H. 1960]. The term "branch-and-bound" was first proposed by J.D.C Little in 1963 for traveling salesman problems [Little, J.D.C. 1963]. It is the most widely used tool for solving large scale optimization problems that have a finite but usually very large number of feasible solutions, for example vehicle routing, crew scheduling, and production planning [Clausen, J., 1999].

We developed a specific branch and bound algorithm, with a strategy that prunes the tree from bottom to top using a best-first search strategy through an iterative procedure. The detail of the procedure is given in Section 4.6. This method is an exact approach for finding the optimal solution, and it is very efficient in detecting bad choices on the upper level of the tree. But if the bad choices are made on the lower level of the tree, or even on the leaves, the algorithm will not detect these bad choices in order to prune them off, until the full tree is examined. In this special case, the branch and bound algorithm becomes essentially the brute-force search approach, and all branches are explored and compared. As mentioned before, it is computationally challenging to use brute-force search approach for very big trees.

To gain a reasonable computational performance for solving the extreme case
problems described above, we develop a second approach which is a modification to the label-correcting algorithm. This approach can solve the problem in polynomial time, rather than exponential time. The details of this approach are given in Section 4.7. The second approach however, it's not an exact approach that always gives the true optimum solution, rather it gives a good solution which is relatively close to the true optimal. To examine the accuracy of the second approach, we run the first approach (branch and bound algorithm) and compare the solutions of the two approaches for a special case. The results are average runtime results and problem specific, that can not be generalized.

4.5 Exhaustive search

In exhaustive search approach, we enumerate all possible paths in the graph, and consider all possible paths from a n-ary rooted tree as illustrated in Figure 4.1. We examine each branch in the tree independently and individually. In total, a tree that enumerates all the paths in a graph has \( n^m \) branches, the run time of the brute-force search approach is \( O(n^m) \). We are guaranteed to find the exact solution using this approach, however it is computationally very expensive, especially for big trees.

4.6 Branch and bound algorithm with bottom-up pruning

Instead of exploring the entire solution space as with the exhaustive search approach, branch and bound algorithm only examines parts of the solution space, hence
increasing the algorithm efficiency. By using a bound for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of the solution space only implicitly.

Assuming we are given a bound and a corresponding path. The distance of the bound is the distance of the full path. If a subbranch distance already exceeds the bound, one can immediately throw away the subbranch and all paths that contain it without having to look further into the $n$ subtrees below it. This procedure is known as eager node evaluation [Clausen, J., 1999].

Taking advantage of the eager node evaluation approach, we first find a bound, and sort the tree such that we always keep the bound as the leftmost branch. Then we re-examine the bound from its leaf to its root. We can do so because the subbranch distance is not influenced by the choice of plant designs further down in the branch. In other words, future decisions do not change the cost of decisions in the past (i.e. past decisions are sunk cost). This nature of the problem makes the algorithm special. In the more general case, where future costs do change the cost of decisions in the past, we can not use a bound to prune out the bad branches early on, since the seemingly bad branches may turn out to be a good one later on.

Regarding search strategy, we employ the depth first search (DFS) approach. Figure 4.3 shows a small search tree, the number in each node corresponds to the sequence, in which the nodes are processed when DFS is used [Clausen, J., 1999]. The search strategy usually reflects a trade off between keeping the number of explored nodes in the search tree low, and staying within the memory capacity of the computer used [Knuth, 1997].
Figure 4.3: An instance of depth first search strategy in branch and bound

Let's look at Figure 4.1 and walk over the steps in this algorithm. We assume the initial bound is infinity, and there is no corresponding path. First, we branch off from the root to a full path by always choosing the local optimum at each step until we finish. As illustrated in this example, this path is \( S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3) \rightarrow d_1(t_4) \) with a distance of 37. Since 37 is less than infinity, we update the bound as our current best and keep it on the leftmost of all branches. Now that we have a bound and a corresponding path, we start pruning the tree from the leaf of the current bound towards the root in an iterative manner. For example, in this case, we back off from the leaf. First we back off one step to \( d_1(t_3) \), since from \( d_1(t_3) \) to the last step, we only need to look at the local optimum, which we did when we chose this path in the first place, there cannot be a better path. Therefore, we back up one more level and look at \( d_2(t_3) \), while maintaining the subbranch above as \( S(t_1) \rightarrow d_1(t_2) \). We notice the subbranch distance \( S(t_1) \rightarrow d_1(t_2) \rightarrow d_2(t_3) \) is 20 + 30 = 50, and this is already greater than the 37. Immediately, we can throw away the whole subbranch, without having to look any further. Next we look at \( d_3(t_3) \), while maintaining the subbranch above as \( S(t_1) \rightarrow d_1(t_2) \). The subbranch distance \( S(t_1) \rightarrow d_1(t_2) \rightarrow d_3(t_3) \) is 20 + 20 = 40. It is greater than 37, therefore, we can throw away the whole subbranch. Now we are
done on the t3 level, and we back off two steps of the bound at d1(t2), and prune the tree in the same manner describe above. Whenever we find a full branch shorter than the current bound, we update the leftmost branch to ensure it is always the shortest. This way, we obtain a tighter bound. We continuously tighten the bound as we prune the tree, which makes the algorithm even more efficient. In this case, when we look at path \( S(t1) \rightarrow d3(t2) \rightarrow d1(t3) \rightarrow d2(t4) \), we find the path distance to be \( 10 + 13 + 2 = 25 \). Since 25 is smaller than 37, we update our bound as \( S(t1) \rightarrow d3(t2) \rightarrow d1(t3) \rightarrow d2(t4) \).

When we finish pruning the tree, we find the current bound as the best path. In this case, the best path is \( S(t1) \rightarrow d3(t2) \rightarrow d1(t3) \rightarrow d2(t4) \).

This method is especially useful for problems in which very bad choices are made on the upper level of the tree. For problems in which very bad choices are made on the upper level of the tree, this method is very useful as one can prune the bad branches pretty quickly. But if the bad choices are on the lower level of the tree, or even on the leaves, the algorithm is not able to detect them until the full tree is thoroughly examined. For example, if on each level except for the last level, the arc distance to each node are exactly the same, then one would not be able to throw away any subbranches, because all subbranches have the same length. If the differences only begin to appear on the last level, one needs to calculate and compare all branches. In this case, the branch and bound algorithm becomes essentially the brute-force search approach, and one ends up doing an exhaustive search of the full tree.

In summary, this algorithm can find the optimum solution, which has the lowest distance number for a path-dependent problems. It is especially efficient with problems where bad choices are made on the upper level of the tree, and less efficient when
bad choices are made on the lower level of the tree. The worst case of this algorithm is a case of the brute-force search approach.

The pseudocode of this algorithm is given below.

- *path* is the path we are looking at (its lower level parts need to be defined. The part from level on up, need not be defined
- *level* is the current working level, all parts earlier in the path are defined
- *bound* is the current best bound, it can be "infinite"
- *bestpath* is the current best path, it may be undefined at the outset, it may be defined
- *PowerPlant* is the list of all design choices
- *allnodes* is the list of all nodes in the graph, each node represents one design at a specific level on the path
- *Function P(level, path, PowerPlant, allnodes)* calculates the penalty number of current path. Since *path* is an index array of integers, the function need to convert the array of integers to an array of actual nodes, where each node represents a unique power plant design object. Therefore the input of the function includes *PowerPlant* and *allnodes*. The function calculates the penalty number of the path, by calling the penalty functions located inside the design objects.
Algorithm FindBestPath(path, level, bound, bestPath)

1  **Input:** path, level, bound, bestPath
2  ▷ if it’s a full path, calculates the penalty and wrap up
3  if (level ≡ n_{step})
4      then
5      Penalty = P(level, path, PowerPlant, allnodes);
6      if (Penalty < bound)
7          then
8          bestPath ← path;
9          bound ← Penalty;
10  ▷ if it’s not a full path, calculate all next levels and then sort
11  for i ← 1 to n_{step}
12      do
13         path[level] = i;
14         P[i] = P(level, path, PowerPlant, allnodes);
15         P[s[i]] = sort(P[i]) ▷ s[i] is the secondary index array
16  ▷ find all paths from here, quit when this level exceeds the bound
17  for i ← 1 to n_{step}
18      do
19         if (P[s[i]] > bound)
20             then
21                 break;
22         FindBestPath(path, level + 1, bound, bestPath);
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

4.7 Heuristic based on label-correcting algorithm

To solve problems in which the distances of all paths are very similar to each other, it may be too expensive to use the branch and bound algorithm. In order to reach a reasonable computational cost, we seek alternative methods that would find a solution which is close to the best possible answer, i.e. a heuristic. The outcome of the heuristic may not be the optimum path, but it is a good path that is close to the optimum. The results can be compared with the results of the exhaustive search approach to verify the accuracy of the heuristic.

Instead of working on a tree with an exhaustive search approach, the heuristic works on a graph (Fig 4.2) as originally defined in section 4.2. This heuristic is based on the label-correcting algorithm, which is very efficient in solving the shortest-path problem in polynomial time.

4.7.1 The shortest-path problem

The shortest-path problem is not path-dependent, each arc only has one unique value, regardless of its past history. Therefore we can associate a numerical value or a distance label with each node, representing the subbranch distance from the root to that node. Different paths give different distance labels.

Since it is not path-dependent, one can calculate the distance label of a node by considering only local information, namely the length of single arc. Therefore, instead of having to remember the full tree, one needs only remember the best path reaching each node on that level. Since there are exactly $n$ nodes on every level, one needs to keep $n$ paths, each being the shortest for each node.
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

One always works on two neighboring levels, each level has exactly $n$ nodes, therefore, the total number of enumeration for every pair of levels is $n^2$ paths. On each level for each node, one iteratively reduces the distance label until the best path to reach that node is found.

The procedure described above is essentially a label-correcting algorithm, which is a general procedure for successively updating the distance label until they satisfy the shortest path optimality condition. The total runtime of this algorithm is $O(nm)$. This approach solves the problem in polynomial time.

4.7.2 Path-dependent shortest-path

The critical difference between our problem and the shortest-path problem lies in the difference in path-dependency. Figure 4.4 illustrates an instance of graph that is path-dependent.

Figure 4.4: An instance of a graph, in which an edge has multiple values as a result of multiple path histories

As illustrated, the same arc may have multiple values, instead of just one value.
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

For example, arc \(d_1(t_3) \rightarrow d_1(t_4)\) has three values, and each one corresponds to a unique path histories. Specifically, the distance of \(d_1(t_3) \rightarrow d_1(t_4)\) is 5 after subpath \(S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3)\), 14 after subpath \(S(t_1) \rightarrow d_2(t_2) \rightarrow d_1(t_3)\), and 31 after subpath \(S(t_1) \rightarrow d_3(t_2) \rightarrow d_1(t_3)\). What is more, the principal of dynamic programming no longer holds. For example, path \(S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3) \rightarrow d_1(t_4)\) is the shortest path in graph, however, subpath \(S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3)\) is suboptimal compared to subpath \(S(t_1) \rightarrow d_3(t_2) \rightarrow d_1(t_3)\).

If we use the label-correcting algorithm to solve Figure 4.4, subpath \(S(t_1) \rightarrow d_1(t_2) \rightarrow d_1(t_3)\) will be pruned out before reaching the last level. To avoid pruning out a good path early on, we modify the label-correcting algorithm, such that on every level, we keep more than \(n\) paths. In addition, we keep another \(x\) good paths, such that on every level we keep a total of \(x+n\) paths. The choice of \(x\) depends on one’s preference of runtime versus accuracy \((0 \leq x \leq (n^m-n))\). On one end of the spectrum, when \(x\) equals zero, in total we are keeping \(n\) paths on each level, this is essentially the non-path-dependent shortest-path algorithm. On the other end, when \(x\) equals \(n^m-n\), in total we are keeping \(n^m\) paths, this is essentially the brute-force search approach. For simplicity, we assume \(x=2n\) in this work.

We further improve the heuristic by introducing the concept of a bound. In this case, we first get a bound by running the shortest-path algorithm \((x=0)\). Then we run the heuristic, we keep a maximum total of \(x+n\) subbranches on every level, such that all subbranches we keep are smaller than the bound. By using a bound, we prune out the bad subbranches early on.

One can further improve the heuristic by gradually increasing \(x\). One can update
the bound whenever a smaller optimal path is found for a given $x$. The price to pay in this scenario, is that one needs to prune the tree once for every value of $x$. Further, the optimal path does not change monotonically as $x$ increases. In other words, by increasing the value $x$, one is not guaranteed to find a better path. A detailed discussion is given in the following section.

The limitation of this heuristic is that it may not be able to find the true optimum path. To examine the accuracy of the heuristic, we can compare results of the heuristic with the brute-force search approach. The detail of the comparison is given in the following section.

### 4.8 Computational results

We summarize the algorithms described in this chapter as following.

- **exhaustive search** or brute-force search, explicitly enumerates the complete space of solutions, calculates and compares all possible solutions for optimality.

- **branch-and-bound algorithm** By using a bound for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of the solution space only implicitly.

- **heuristic** improves the label-correcting algorithm for the shortest path problem, by remembering a few additional good subpaths ($x$ subpaths) on each level, to avoid pruning out a good path early on. For simplicity, we assume $x=2n$.

In this section, we present computational experiments conducted to evaluate the quality of the three approaches described above, namely the exhaustive search ap-
proach, the branch-and-bound algorithm and the heuristic. We tested the approaches on a sequence of examples with a feasible solution space ranging from thousands to trillions of paths. These particular example problems form 6-ary search trees of different levels. On a small problem, a 5-level tree gives 1296 possible paths ($6^4$), on a big problem, a 15-level tree gives 7.84E10 possible paths ($6^{14}$). Problems forming search trees with all levels between 5 to 15 are also explored. All results are average runtime results.

The nodes on the trees in this example are all power plant designs, and each full branch on the tree is a unique technology pathway. The functions to calculate the path distance are penalty functions that calculate the penalties of technology pathways. The nodes are not random nodes, and the penalty numbers are not random numbers. Rather, they are real objects with real meanings. If we use random numbers, we are taking advantage of certain structures in the numbers. Specifically, they tend to be large in variation, and there is certain percentage of extremely large numbers. The performance of the algorithms will depend on the statistical property of the numbers. Therefore, by giving a realistic example, we’ll get a different distribution than for example using Gaussian distribution around zero. In essence, the results we presented here not only demonstrate the efficiency of the algorithm, but also solve a real engineering problem. Since this chapter focuses on the algorithm aspect, and because it is formulated as a test model without the intention to explore the engineering implication, we do not go into details in the science and engineering aspects of the results. The details of problem assumptions are given in Appendix A.4, for interested readers who want to reproduce our results following our methods.
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

We prune the trees with the branch-and-bound algorithm and the heuristic respectively. We also provide the brute-force search approach to calibrate ourselves.

The algorithms are coded in C++ and tests are carried out on a PC with AMD64 architecture under Linux system Ubuntu. All results we present here are average-case results, rather than worst-case results.

4.8.1 Verifying branch-and-bound algorithm

First, we compare the the branch-and-bound algorithm with the brute-force search approach over a set of 6-ary search trees with a height from 5 levels to 15 levels. The branch-and-bound algorithm can degenerate into the brute-force search approach for a tree with relatively uniform branches. In that case, every node on the tree has to be visited exactly once. Further, whenever a node is visited, the program triggers a function call which calculates a distance number. From a computational point of view, this calculation is the most expensive task for each iteration. Therefore, comparing the actual nodes visited using the branch-and-bound algorithm against all nodes in the tree, is a good measurement of the algorithm efficiency.

Figure 4.5 illustrates the number of visited nodes against all nodes for all levels considered. First of all, we were successful in discarding bad branches and nodes using the algorithm over all level of problems considered. Secondly, none of the problems degenerate into the brute-force search approach. Lastly, as the tree level increases, the fraction of total nodes visited decreases considerably. Therefore, for large trees, the branch-and-bound algorithm is much more efficient than the brute-force search approach.
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

4.8.2 Verifying Heuristic

Secondly, we compared the visited nodes of the heuristic with the branch-and-bound algorithm to show that heuristic is actually more efficient. The heuristic assumes $x=2n$, therefore it remembers a maximum total of $3n$ paths on each level of the tree.

Figure 4.7 illustrates the visited nodes of both the branch-and-bound algorithm and the heuristic, over a set of 6-ary search trees from 5 levels to 15 levels. It is worth noting that these results are average runtime results, not the worst case runtime results. Given different inputs, the results may be different.

As illustrated, the visited nodes for both branch-and-bound algorithm and the
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

Figure 4.7: The visited nodes comparison between BB and the heuristic

Figure 4.8: The results given by the heuristic compared to the brute-force approach

The heuristic increases as the tree levels increases. Further, the heuristic is more efficient compared with the branch-and-bound algorithm over all levels considered. Lastly, as the number of tree level increases, the fraction of visited nodes decreases considerably. The fraction of total visited nodes of the heuristic compared to the branch and bound is 47 percent at 5 levels, and only 2 percent at 15 levels.

It is worth noting that, the shape in Fig 4.7 is convex, compared to the concave shape in Fig 4.5. The reasons are as follows. First, the concave slope in Fig 4.5 shows that, as the tree levels increases, without increasing the variation in the designs, the relative advantage of using branch and bound compared to the brute-force search approach increases. This result reflects the structure of the problem, rather than the general properties of the algorithm. Secondly, the convex slope in Fig 4.7 shows that, as the tree levels increase, without increasing the variation in the designs, the relative advantage of using heuristic compared to branch-and-bound(not the brute-force search) decreases. Again, this result reflects the structure of the problem, rather than a general feature of the algorithm.

We compare the optimal results given by the heuristic and the brute-force ap-
Chapter 4: Path-dependent shortest-path algorithms for optimizing a sequence of power plant designs

proach to calibrate ourselves. Note in this comparison, we ran the heuristic assuming \( x=2n \). Figure 4.8 illustrates that for these specific sets of problems, the optimal results found using the heuristic agrees with brute-force approach at lower levels, specifically from 5 levels to 11 levels. At higher levels, namely from 12 level to 15 levels, the heuristic gives suboptimal solutions compared to the results given by the brute force search approach.

As mentioned earlier, heuristic is not an exact method, therefore it does not guarantee always to give the optimal results. Figure 4.9 illustrates the optimal results given by heuristic compared with the brute-force search, by increasing the number of additional subpath \( x \) for a problem with 6-ary trees of 15 levels. Firstly, when \( x=0 \), this heuristic is essentially the shortest path algorithm. Since it ignores the path-dependent nature in the problem, the results given are suboptimal. Secondly, when \( x=2n \), the results agree with the optimal path on level 15 in Fig 4.8. Thirdly, the
results changes as \( x \) increases, reflecting that heuristic does not always guarantee to find the optimum. Lastly, the changes of the results are not monotonic as \( x \) increases. The detail interpretations of the results are given in the discussion section.

4.9 Discussion

We have shown that the branch-and-bound algorithm is indeed more efficient than the brute-force search approach for the test model, as illustrated in Figure 4.5. For these specific sets of problems, the efficiency of the branch-and-bound algorithm increases as the number of levels in the tree increases. Further, more Figure 4.6 illustrates that the results given by the branch-and-bound algorithm agree with the brute-force search approach over all levels considered.

Despite the improved efficiency, the branch-and-bound algorithm can still take up to nine minutes in solving the test problem of 6-ary tree of 15 levels. For a large tree with a large number of steps and a great number of choices for each step, the runtime can increase considerably. The significance of the increment in runtime depends on the structure of the input. In the best case, if the additional choices are all bad choices, then the increase in runtime is minimum. However, in the worst case, the branch-and-bound method degenerates into a brute-force search approach. To improve the algorithm even more, we proposed a heuristic to obtain a good solution in a reasonable time. We have shown in Figure 4.7 that the heuristic is indeed more efficient than the branch-and-bound algorithm. The fraction of total visited nodes compared to the branch-and-bound algorithm decreases as the tree level increases, indicating an increased efficiency gain with larger trees. We compare the results
given by the brute-force approach and the heuristic assuming \(x=2n\). We have shown in Figure 4.8 that the heuristic agrees with the brute-force approach over lower levels of the tree, but disagrees at higher levels.

To calibrate ourselves, we run the heuristic for a 6-ary tree of 15 levels, by slowly increasing \(x\) from zero to nine times the number of design options. As illustrated in Figure 4.9, given different \(x\), the results may not agree with the optimal results given by the exact method. Further, the differences between the optimal and suboptimal results are very small. This is problem-specific. Given a different tree, the difference may be more dramatic. Lastly, the change in the optimal results is not monotonic with the increase of \(x\). This is because though some of the \(x\) partial paths look promising on the upper level, they can later become terrible choices. An extreme case is when the partial path is locked-in to a wrong path, which later becomes too expensive to escape from. Additionally, keeping \(x\) additional subpaths on early levels, can crowd out the true optimal subpath, leading to suboptimal results. In fact, this is the case when \(x\) is between \(2n\) and \(8n\), as illustrated in Figure 4.9. When \(x=n\), the results given by the heuristic agree with the exact method. This is because keeping only \(n\) additional subpaths did not crowd out the true optimum. It is worth to noting, the crowding out does not need to happen. It is just what happened here. Finally, from Figure 4.9, we observe that the results of the heuristic is not necessarily monotonic in \(x\). Given a different tree, the range of \(x\) values, in which optimal solution is crowded out may be different.

In addition to the two approaches discussed above, we can combine the two approaches for a hybrid algorithm. Specifically, we can first run the heuristic to get a
good bound, then we can run the branch-and-bound algorithm with the bound obtained from the heuristic. Theoretically, the hybrid algorithm should be more efficient than the branch-and-bound because it starts with a tighter bound, therefore making the pruning more efficient. We implemented the hybrid algorithm. Figure 4.10 illustrates the number of discarded nodes using the branch-and-bound and the hybrid algorithm over a set of 6-ary trees of 5 levels to 15 levels, the differences are illustrated as dots. The results show that, first there are differences in the number of discarded nodes over all levels considered. Secondly, the differences are very small. We believe this reflects the nature of this specific set of problems, rather than generic features of the algorithm. Given a different set of inputs, the outcome may be different.

Figure 4.10: Visited nodes in the branch-and-bound and the hybrid algorithm

So far we have presented the results of deterministic discrete optimization. Firstly, we have assumed that the decision maker is a rational economic person, and is willing to invest the cost upfront, and receive the benefit much later. However, if the decision
maker wants immediate gratification, rather than the true global optimal, he/she may prefer a suboptimal solution, which gives the optimal benefit in the short run. Secondly, although the cost of technology and the environmental penalty are expected to change over time, we do not intend to forecast the technology development. Rather, we view these changes as exogenous to our model. Right now, it is hard-coded, but we can modify the interface for user-supplied inputs in the future. We aim to give users the freedom to obtain their own results, by allowing them to provide their own forecast in technology and policy changes. Finally, the uncertainties in the technology development, and potentially emerging new technology choices over time are beyond the scope of this research.
Chapter 5

Optimizing CO$_2$ post-combustion capture technologies

In this chapter, we study various post-combustion capture technologies using the ranking algorithms described in Chapter 4. In the first part of our work, we developed a simple model for the absorber system and the penalty model for the whole plant.

In the second part of our work, we model various power plant designs with different CO$_2$ absorbers and various power plant technology pathways. We find the optimum pathway by exploring the entire solution space of all possible pathways, via the algorithms described in the Chapter 4. We compare the results of the best sequence of technologies and the sequence of best technologies at various times.
5.1 Motivation

Post-combustion capture technology captures carbon dioxide at point sources (i.e. coal-fired power plants or cement plants), by separating carbon dioxide from the flue gas via chemical or physical absorption. This procedure can be used for capturing carbon dioxide from power plants as an "end-of-the-pipe" approach, allowing the existing technology designs and infrastructures to remain relatively unchanged. Therefore, it is especially attractive for governments and companies who are eager to deploy low-emission power plants without abandoning existing plants.

The technology of absorbing carbon dioxide from a mixed gas stream is well studied and widely used in the chemical industry for gas treatment. However, it has not been used in coal-fired power plant for CO2 separation. Traditionally, gas treatment systems are designed to absorb as much CO2 as possible while meeting the economic constraints. Drawing from the experience in the chemical industry, power plant designs with CO2 capture using post-combustion capture technologies often target 90% capture percentage or even higher (i.e. [NETL, 2007], [Abu-Zahra, M., 2007a]). Furthermore, techno-economic studies of the power plant system with CO2 absorber, often takes the absorber as an optimized module. While the absorber itself is optimized locally, it may not be the optimal design when evaluating the entire power plant, due to the fact that the regeneration of sorbents in the absorption system requires energy from the power generation island. The optimal absorption system fully integrated into the power plant island, may be different from the design by merely putting an optimized absorption system and an optimized power plant island together.
It is our view that, the capture percentage range below 90% may offer economic opportunities for re-optimizing the absorber system integrated with the power plant, by taking advantage of the weaker and cheaper sorbent.

On one end of the spectrum, as one attempts to capture the last fraction of CO$_2$ from the flue stack, it is conceivable that the capture cost would rise significantly. This is due to the fact that as the partial pressure of CO$_2$ approaches zero, the driving force required for absorption increases significantly. This necessary increase in the driving force for absorption results in diminishing returns which make it difficult to collect the last bit. Theoretically, to capture one hundred percent of CO$_2$, one needs to build an infinitely tall absorber tower. Although one can capture virtually all the CO$_2$ produced, the cost per ton of CO$_2$ captured is getting too large.

On the opposite extreme, if one were to capture an infinitely small amount of CO$_2$, the cost would likely be small. If one only requires a small amount of CO$_2$, then even a weak sorbent can collect a little amount of CO$_2$. For example, by licking one's finger tip and simply holding it in the air, one can capture an infinitely small percentage CO$_2$ from the air on the wet finger tip surface. The capital cost of doing so is nil. Despite the low capital cost, the unit cost per ton of CO$_2$ captured can be significant as one only captures very small amount of CO$_2$.

Between the two ends of the spectrum, there is an optimal capture cost ($/ton CO$_2$), which is associated with an optimum absorber design using an optimum sorbent. In such a design, design parameters such as the sorbent strength, packing geometry, flow pattern and flow rates, are reoptimized for the lowest penalty at each targeted capture percentage.
One can further optimize the sequence of power plant designs. As the world move from ignoring CO\(_2\) emissions to forcing significant reductions in CO\(_2\) emissions or alternatively charging a high price for CO\(_2\) emissions, the type of power plants built will change to account for the change in economics. Such a sequence can be optimized on various designs with various CO\(_2\) absorber configurations. The optimal sequence of power plants may be different from a sequence of designs, all of which are optimized to capture 90\% of the CO\(_2\) emissions. It is possible to build an advantageous technology pathway from today’s power plant designs to a future low emissions design via a number of intermediate steps.

This chapter studies the subject of optimizing post-combustion capture power plants. Optimization is accomplished by redesigning absorption systems for various capture percentages using various sorbents. In the first part of this chapter, we optimize a single plant design at various times. In the second part of this chapter, we optimize a sequence of plant designs using the algorithms introduced in Chapter 4.

A post-combustion capture power plant using monoethanolamine (MEA) is modeled with the software discussed in detail in the Chapter 6. The absorber modeling procedure strictly follows Perrys’ chemical engineering handbook and agrees with the procedure described in Oexemanne 2008 [Oexemanne, J., 2008]. The cost model is described in section 5.5. Optimization parameters include packing tube radius, column height, capture percentage, choice of sorbent, etc. Results are presented in section 5.7.
Chapter 5: Optimizing CO₂ post-combustion capture technologies

5.2 Literature Review

Many researchers have studied optimizing post-combustion capture technologies (i.e. [NETL, 2007], [Abu-Zahra, M., 2007a], etc). In the DOE/NETL study, a 300 MW subcritical power plant with 90 percent CO₂ captured using MEA sorbent is considered as the reference case. To simulate a lower capture percentage in the full-size plant, the method proposed in the DOE/NETL study bypasses fluegas such that lower capture percentage in the full-size plant can be considered equivalent (in terms of CO₂ captured per hour) to a 90 percent capture rate in a much smaller plant. The optimization of CO₂ absorber design takes advantage of the economy of scale, and therefore does better with a larger unit. Thus, the optimal result is a capture system with the largest possible absorber column. However, this study did not account for various capture system designs, it only reduced the size of the stream subject to capture. For various CO₂ capture percentages in power plants, one can choose to pack the absorption tower loosely or densely. One can also choose to use a weaker or a stronger sorbent.

Different sorbents have different reaction kinetics, costs, and regeneration heat requirements, which have a non-trivial impact on the cost per ton of CO₂ capture. For simplicity, we consider sorbents with very fast reaction kinetics, i.e., strong sorbents, and the opposite case, sorbents with very slow reaction kinetics, i.e., weak sorbents. For a perfectly strong sorbent in the laminar flow regime, when the Schmidt number is close to 1, the CO₂ absorption as a fraction of total is approximately the same as the momentum loss as the fraction of total momentum. If loose packing in a taller absorber gives the same pressure drop as a dense packing in a shorter absorber, a
perfectly strong sorbent in the laminar flow is insensitive to the size and packing of the column as long as the pressure drop is the same. However, for imperfect sorbents, the design of the column matters.

The DOE/NETL study also failed to take advantage of various sorbents with various binding energies. For example, when a weaker sorbent is used (i.e. slightly alkaline sorbent, seawater, etc), less energy per ton of carbon dioxide is needed for CO$_2$ regeneration.

In addition, when studying optimization with various sorbents, it is important to fully integrate the capture system into the power plant. In the post-combustion capture system, substantial amount of steam is extracted from the steam turbine for CO$_2$ regeneration in the stripper. Therefore, a globally optimized power plant can operate with the lowest cost and the lowest energy penalty of the entire system. However, due to the lack of software tools suited for this purpose, many studies have shown that, researchers use one software for power plant modeling, and an other software for capture system modeling. Since it is usually difficult to fully integrate the two softwares, optimization is often carried out locally in the capture system based on the pre-determined power plant operating condition (i.e. [Oexemanne, J., 2008]). With the help of the newly developed software tool (described in the Chapter 4), it is possible to fully integrate the capture system into the power plant designs, and carry out global optimization.

The DOE/NETL study [NETL, 2007] found a decreasing CO2 avoidance cost ($/ton CO2 capture) with increase in CO2 capture percentage, implying that the more one scrubs the cheaper/easier it becomes. This result is counter-intuitive, but a
closer examination of this report revealed that the assumption of bypassing a portion of the flue gas, is essentially equivalent to scrubbing a smaller power plant. The cost reduction of more severe scrubbing is caused by the economy of scale with larger absorption columns at a higher capture percentage. Applying absorber designs for high capture percentage ranges (90 percent or above) to all capture ranges, will likely to result in suboptimality.

Unlike the DOE/NETL approach, we aim to study the CO2 avoidance cost ($/ton CO2 capture) as a function of capture percentage without bypassing any flue gas for a hypothetical sorbent in a hypothetical column design. Further, we study the optimal sequence of power plant designs, choosing from a set power plant designs with various CO2 absorption systems.

5.3 Modeling CO2 absorber physics

A post-combustion capture power plant using monoethanolamine is modeled with the software introduced in the Chapter 6. For the purpose of this discussion, the modeling details of the power plant island are included in the Appendix A.5.4 and A.5, while this chapter focuses on the modeling details of CO2 absorber.

Let us consider a hypothetical packed bed absorber tower for CO2 absorption using a hypothetical sorbent. The uptake rate of the absorption tower is characterized by a surface area and a mass transfer coefficient. Specifically, the surface area is determined by the packing geometry, whereas the mass transfer coefficient is determined by the packing geometry, the sorbent strength, and the hydrodynamic condition of the absorber.
For the simplicity of the discussion, the packing structure inside the absorber can be considered as bundles of evenly divided tubes, the length of which equals to the absorber height. This type of structured packing is rare in reality, however this treatment greatly simplifies the packing geometry, hence allowing us to understand the physics of packed bed column with a rather simple model.

Let us consider the packing arrangements such that all tubes are identical, and are parallel to each other. An example of the cross sectional view of the absorber column is shown in Figure 5.1, where 33 small tubes are packed in a big column [Specht, E., 2010].

![Figure 5.1: An instance of the cross-section of a packing tower](image)

The tube wall is fully coated with sorbent which is continuously refreshed. The overall tower diameter and volumetric flue gas flow rate are known, hence we can determine the average gas velocity. For practical purposes, whether the sorbent combines with CO₂ chemically or physically cannot be determined.
5.3.1 Diffusion process

In the laminar or somewhat turbulent flow, we can make the following argument about the absorption process. During the CO₂ absorption, momentum transfer to the wall, follows a similar diffusion equation (equation 5.2), as that of the CO₂ transfer to the wall (equation 5.1). Momentum transfer coefficient and mass transfer coefficient are of the same order of magnitude, their ratio is measured by the Schmidt number, as given in equation 5.3. When the Schmidt number approaches unity, the value of momentum transfer coefficient equals the value of mass transfer coefficient.

\[
\tau = -\mu \times \frac{\partial v_z}{\partial r} \quad (5.1)
\]

\[
J_{CO_2,g} = -\rho D \nabla CO_2 = -\rho D \frac{\partial y_{CO_2}}{\partial r} \quad (5.2)
\]

\[
Sc = \frac{\nu}{D} = \frac{\mu}{\rho D} \quad (5.3)
\]

In the center of the tube, both momentum and CO₂ are at their maximum. On the wall, momentum always go to zero. CO₂ will also go to zero if a perfect sorbent is coated on the wall, and is continuously refreshed (i.e. no CO₂ loading in the sorbent). Therefore, given the shared boundary conditions on the wall and at the tube center, we argue that the fractional total of CO₂ and the fractional total of momentum across the tube radius are approximately the same. In other words, if one percent of momentum is lost, then one percent of CO₂ is lost, too.

Nevertheless, momentum does not get depleted as CO₂ does, because the momentum loss is continuously replenished by pressure drop. If one percent of momentum
is replenished, then one percent of pressure is lost, therefore we can establish that, momentum loss and pressure drop are proportional. Since we have established in the previous paragraph, that momentum lost is also proportional to CO$_2$ loss, we can further establish that CO$_2$ loss is proportional to the pressure drop. Therefore, if one wants to take out a certain fraction of the CO$_2$, one needs to take out the same fraction of the flue gas pressure.

For a sorbent which meets the boundary condition of zero CO$_2$ concentration on the wall, as long as the pressure drop is kept constant, the CO$_2$ capture percentage is also kept constant. Therefore, the packing structure inside the bed can be either long tubes with big openings, or short tubes with narrow openings. Since the latter is more favorable for economic reasons, one can reduce the tube opening, to the point where the boundary condition does not hold. The boundary condition of a small partial pressure of CO$_2$ at the wall will break down for very narrow tubes, because the resistance to flow in the gas is getting smaller until it eventually does not dominate the impedance in CO$_2$ transport. The tube radius at the point, where the gas side impedance ceases to dominate, is defined as critical radius $r_c$. Each sorbent with a unique binding strength, has a unique critical radius. The value of $r_c$ is relatively small for a strong sorbent, and large for a weak sorbent.

When the tube radius is smaller than $r_c$, it is undesirable to continue reducing the tube radius for higher mass transfer coefficient. This is because with a small opening, the CO$_2$ uptake rate is small due the limitation on the wall, but one still need to pay for the big pressure drop. Therefore, the proportion of the momentum taken out to the amount of CO$_2$ taken out is suddenly unfavorable.
Since the uptake rate and the mass transfer are greatly influenced by the packing geometry, there exists huge opportunities to redesign the packing structure for various sorbents. It is our view that one can redesign the packing structure for each sorbent with a unique binding strength and a unique critical radius, for the lowest cost per ton of CO\textsubscript{2} capture. This optimum design corresponds to a capture percentage for the chosen sorbent. Therefore, for a given capture percentage target, one can select the optimum sorbent with the optimum design for the lowest cost of per ton of CO\textsubscript{2} capture.

5.3.2 Interfacial partial pressure and concentration

CO\textsubscript{2} transfer on the wall side is given by the following equation

\[
J_{CO2,w} = k_l \times (\bar{P}_{CO2,w} - \bar{P}^*_C)
\]  

(5.4)

For the simplicity of the problem, we assume that the mass transfer profile across the tube is linear, and the boundary layer in the gas side is the tube radius.

\[
J_{CO2,g} = -\rho D \frac{\bar{P}_{CO2, in} - \bar{P}_{CO2,w}}{r_t}
\]  

(5.5)

Consider steady state, the CO\textsubscript{2} flux on the wall is always in equilibrium

\[
J_{CO2,g} = J_{CO2,w}
\]  

(5.6)

One can solve for the boundary condition as

\[
P_{CO2,w} = \frac{k_l \times \bar{P}^*_C + \rho D}{k_l + \frac{\rho D}{r_t}} \times \bar{P}_{CO2, in}
\]  

(5.7)

\[
\bar{P}_{CO2, in} > \bar{P}_{CO2,w} > \bar{P}^*_C \text{ is the condition for absorption, } \bar{P}_{CO2, in} < \bar{P}_{CO2,w} < \bar{P}^*_C
\]

is the condition for desorption.
The CO₂ flux on the wall can be expressed as

\[ J_{CO_2} = \frac{\rho D k_l (P_{CO_2, in} - \bar{P}_{CO_2})}{\rho D + k_l r_t} \tag{5.8} \]

The interfacial liquid side CO₂ concentration can be found with Henry’s law.

\[ C_{co_2,w} = H P_{co_2,w} \tag{5.9} \]

### 5.3.3 Vapor Liquid Equilibrium

The correlation for vapor liquid equilibrium is taken from [Gabrielsen, 2005]

\[ P_{CO_2}^* = K_{CO_2} 10^3 X_{CO_2} \frac{X_{amine,0} \theta_{avg}}{(X_{amine,0} * (1 - 2 * \theta_{avg}))^2} \tag{5.10} \]

where

\( \theta_{avg} \) = the average loading (mo of CO₂ / mol of amine)

\( X_{amine,0} \) = initial concentration of amine (mol of amine/ (mo of amine+ mol of H₂O))

\( X_{CO_2} \) = mol fraction of chemically bound CO₂ in the solution.

\( K_{CO_2} \) = CO₂ vapor-liquid equilibrium partial pressure constant

Combined Henry’s law and chemical equilibrium constant for CO₂ partial pressure (kPa), it is calculated using equation 5.10

\[ \ln K_{CO_2} = A + \frac{B}{T} + C X_{amine,0} \theta_{avg} + D \sqrt{X_{amine,0} \theta_{avg}} \tag{5.11} \]

where

\( A = 30.96 \pm 1.86 \)

\( B = -10584 \pm 670 \)
5.3.4 liquid side mass transfer coefficient

Liquid size mass transfer coefficient is determined by the packing geometry, the hydrodynamic conditions, and the sorbent strength.

\[ k_l = k_{l0} \times E \]  

(5.12)

where \( k_{l0} \) is the physical mass transfer coefficient, \( E \) is the enhancement factor.

**physical mass transfer coefficient**

\[ k_{l0} = \frac{0.4185 \times D_L}{h} \times Re_i^{0.333} \times Ga^{0.18} \times Sc^{0.5}; \]  

(5.13)

This correlation is from Shetty and Cerro [Shetty, S; Cerro, R.L 1997].

**Enhancement factor**

Enhancement factor for the sorbents are:

\[ E = \frac{Ha \times \sqrt{(E_\infty - E)}}{tanh \times [Ha \times \sqrt{(E_\infty - E)}]} \]  

(5.14)

where

\[ Ha = \sqrt{D_{CO_2,AM} \times k_2 \times C_{MEA}} \]  

(5.15)

Ha is the Hatta number [Abu-Zahra, M., 2007a].

\[ k_2 = 4.4 \times 10^8 exp\left[-\frac{5400}{T}\right] \]  

(5.16)
k₂ is the reaction rate, assuming the reaction between CO₂ and MEA is simply second order [Versteeg, G., 1996].

\[ E_\infty = [1 + \frac{D_{MEA,am}C_{MEA}}{\gamma D_{CO₂,am} \times C_{CO₂,i}}] \]  

(5.17)

### 5.3.5 CO₂ uptake rate

\[ r_{CO₂} = A_p J_{CO₂} \]  

(5.18)

\[ A_p = 2\pi h R^2 r_t \]  

(5.19)

### 5.3.6 Pressure Drop

Pressure drop is given by Hagen-Poiseuille equation.

\[ Q = \frac{\Delta P \pi r_t^4}{8\mu h} \]  

(5.20)

By rearranging the equation, we give the pressure drop as a function of height and radius.

\[ \Delta P = \frac{8\mu h Q}{\pi r_t^4} \]  

(5.21)

To maintain the same pressure drop, one needs to maintain a constant \( \frac{h}{r_t^4} \). Hence, the tube height (also the absorber height) can be expressed as a following equation

\[ h = \frac{\Delta P \pi}{8\mu Q} r_t^4 \]  

(5.22)
5.3.7 Mass conservation

Two mass conservation equations are postulated in this model. Firstly, the difference between CO$_2$ flow rate (kg/s) on both sides of the absorber equals the total CO$_2$ absorption.

\[ \phi_{\text{CO}_2} = m_{\text{CO}_2,\text{in}} - m_{\text{CO}_2,\text{out}} \]  

(5.23)

Secondly, the difference between flue gas flow rate (kg/s) on both sides of the absorber are only caused by CO$_2$ absorption. This is a gross assumption, assuming zero SO$_x$ or H2O reacts with the sorbent. In reality, this equation needs to take into account the effect of SO$_x$ and H2O. For the simplicity of the model, we use the following equation.

\[ m_{\text{FlueGas, in}} - m_{\text{FlueGas, out}} = m_{\text{CO}_2,\text{in}} - m_{\text{CO}_2,\text{out}} \]  

(5.24)

5.3.8 Capture Percentage

Capture percentage (CP) refers to CO$_2$ capture percentage of the plant.

\[ CP = \frac{(m_{\text{CO}_2,\text{in}} - m_{\text{CO}_2,\text{out}})}{m_{\text{CO}_2,\text{in}}} \]  

(5.25)

5.4 Absorber Cost Model

In this section, we focus on the absorber cost model alone. The details of other modules are listed in the Appendix.

For the absorber, there are two parts of the cost in capital cost, the total tower cost and the packing cost. The absorber cost estimation procedure follows OAQPS.
control cost manual [Barbour, W., 1996].

### 5.4.1 CO₂ absorber capital cost

\[
\text{Erection cost (EC)} = \text{Total Tower Cost} + \text{Packing Cost} \tag{5.26}
\]

\[
\text{Total Tower Cost} = 115 \times 1.61 \times A_p \times \frac{h}{h_0} \tag{5.27}
\]

where

- \(A_p\) = the absorber surface area. \(h\) is the tower height.
- \(h_0\) = the reference tower height, we assume \(h_0\) is 30 meter.

The total tower cost factor is 115, the correlation is from taken from [Barbour, W., 1996]. 1.61 is the conversion rate of dollar in 1991 and 2010.

\[
\text{Packing Cost} = \text{unit cost} \times V_{\text{tower}} \tag{5.28}
\]

where unit cost is assumed $2/m³. \(V_{\text{tower}}\) is the volume of the tower.

\[
V_{\text{tower}} = \pi \times R_{\text{tower}}^2 \times h \tag{5.29}
\]

where \(R_{\text{tower}}\) is the radius of the absorber tower.

PEC (Purchase equipment cost) includes equipment cost, instrumentation, sales tax and freight.

\[
\text{PEC} = 1.18 \times EC \tag{5.30}
\]

TCI (total capital investment) of the absorber tower includes, is obtained by multiplying the purchasing equipment cost, PEC, by the total installation factor. The
total installation factor is 2.2, and is confirmed by the gas absorber vendor survey [Barbour, W., 1996].

\[ TCI = 2.2 \times PEC \quad (5.31) \]

Capcost is the annual capital cost ($/yr). We assume the discount rate is 0.138.

\[ Capcost = 0.138 \times TCI \times \left( \frac{1}{n_i} \right)^\beta \quad (5.32) \]

### 5.4.2 CO₂ absorber operation and Maintenance (OAM) cost

For the absorber, operation and maintenance cost here includes fixed oam cost and variable oam cost. Fixed oam costs is estimated to be 2 percent of the absorber annual capital cost.

\[ FixedOAM = 0.02 \times Capcost \quad (5.33) \]

Variable oam costs include sorbent makeup cost, and pump cost.

\[ VariableOAM = 0.0001 \times m_{sorbent,in} + PumpCost \quad (5.34) \]

Assume 0.01% MEA loss, and the cost of MEA is $1250/ton.

The pump cost is the annual electricity cost of the pump. This cost is associated with operating a gas absorber derived from fan requirements to overcome pressure drop in the column, ductwork, and other parts of the control system, and pump requirement to recirculate the solvent. [Barbour, W., 1996].

\[ PumpCost = \frac{1.17 \times Q \times \delta P}{\eta_p} \quad (5.35) \]

where \( Q \) is the volumetric flowrate of flue gas, \( \eta_p \) is the pump efficiency, we assume the efficiency is 70%. 


5.5 Penalty model

In this section, we describe the penalty model which is modified based on the model introduced in Chapter 2. Critically different from the penalty model in Chapter 2, we introduce the concept of learning on the module level. Therefore, the capital cost of a module decreases as its number of production increases. It is worth noting that, the capital cost of a module at a given time is a function of the entire history of the same module built before, thus resulting in the special characteristics of path-dependency of the problem. The plant penalty and pathway level penalty are simplified compared to the model in Chapter 2. The details are given as follows.

5.5.1 Module penalty

The penalty of a module is essentially the capital cost of the module. We assume the cost a module benefit from both the economies of scale and learning.

Economy of scale

The concept of economy of scale in chemical processes is introduce by [Manne, A. 1967], and can be described in the equation as follows.

\[
Cost(\$) = Cost^0(\$) \times \left( \frac{S}{S_0} \right)^\alpha
\]  \hspace{1cm} (5.36)

Size refers to the output of the module, for example, total heat production of a furnace. We assume \( \alpha=0.75 \) for all modules in the system [EPRI TAG 1991].
Experience curve

The concept of technological learning was first introduced in 1936 at Wright-Patterson Air Force Base in the United States, where it was determined that every time total aircraft production doubled, the required labor time decreased by 10 to 15 percent [Wright, T.P. 1936].

A typical experience curve has the form $Y = ax^\beta$, where $Y$ is the estimated average cost per unit for the $x$th unit of product; $a$ is the unit cost of the first unit; and $b(b<0)$ is a parametric constant called learning elasticity. The learning rate (LR) is defined as the fractional reduction in unit cost for every doubling of cumulative output, and is thus equal to $(1-2^{-\beta})$. The progress ratio, PR, is defined as the fraction of initial cost after a doubling of output, which equals $(1-LR)$, that is $2^{-\beta}$.

Yeh and Rubin studied the learning curves for pulverized coal-fired utility boilers based on historic data [Yeh, S., Rubin, E. 2007], Broek studied the effects of learning on future cost of power plants with CO$_2$ capture [Broek, M. et 2009]. In both studies, learning curve is based on installed capacity (MW). Since we introduce learning on module level, we cannot use capacity as it is a description of the whole plant. Therefore, we account for the learning based on the number of modules produced.

Due to the lack of data, the choice of learning elasticity in this model is quite arbitrary. Given more understanding of the learning curve for individual modules, and more data for learning elasticity, we can improve our assumption in the future.
Module penalty

The penalty of a module is a function of the size of the module, and the number of same modules previously built.

\[ P_{TPL,i} = Cap_i^0 \left( \frac{S_i}{S_0} \right)^\alpha \left( \frac{1}{n_i} \right)^\beta \]  \hspace{1cm} (5.37)

\[ \sum_{i=1}^{m} P_i = \sum_{i=1}^{m} P_{TPL,i} + \sum_{i=1}^{m} P_{OAM,i} \]

where

\( S_i \) = the size of module i

\( S_0 \) = the size of a reference module

\( Cap_i^0 \) is the capital cost of the reference module.

\( \alpha \) is the factor of how fast the cost reduces as one increases the size. We assume \( \alpha \) is 0.75.

\( n_i \) is the number of the same modules previously built.

\( \beta \) is the learning elasticity, we assume \( \beta \) is 0.29.

5.5.2 Plant penalty

The penalty of a plant is the summation of three sub penalties. The first sub penalty is the summation of all module penalties, essentially the annual cost of the plant. The second sub penalty is the reconcile penalty. Certain plant configurations may not be made to work physically, in which case the mass and energy balance model of the plant will not reconcile. The penalty of an irreconcilable configuration is a very large number. We use \( 1e300 \) to represent the penalty of an irreconcilable configuration in the code. The third sub penalty is the environmental cost. Environmental cost is
a function of plant size, and time. As the regulation changes over time, the penalty associated with environmental cost also changes over time. For simplicity, we only account for CO₂ environmental cost in this section. Later in section 5.8, we change this assumption to see how penalty weights affect the pathway optimization results.

\[ P_{\text{tot},t} = \sum_{i=1}^{m} P_{i,t} + P_{\text{rec},t} + c_{\text{co2}} \times P_{E(CO_2,t)} \]

where \( c_{\text{co2}} \) is the cost of CO₂ ($/ton). By default we assume \( c_{\text{co2}} \) is zero, implying that firms will have no motivation to scrub harder than the forecast regulation level.

### 5.5.3 Pathway Penalty

The pathway penalty consist of three parts. The first part describes the summation of all plant penalties, the second and the third part describe the transitional cost between two immediately adjacent plant configurations.

The total penalty for each pathway is calculated according to the following equation:

\[ P_{\text{tot},k} = \sum_{j=1}^{n} \{ P_{\text{tot},j}^{k} \} \times w_{j}^{k} + w_{\text{new}} \times (N_{\text{new},j}^{k}) + w_{\text{delete}} \times (N_{\text{delete},j}^{k}) \]

where

- \( w_{j}^{k} = 1; \)
- \( w_{\text{new}} = 1e4; \)
- \( w_{\text{delete}} = 2e5; \)
5.6 Basic optimization assumptions

5.6.1 Vary the choice of sorbents

The absorber model given above uses monoethanolamine (MEA) as sorbent. We intend to vary the choice of sorbents as an optimization parameter, for the optimal power plant design, and the optimal sequence of power plant designs.

We assume the choice of sorbent defines the absorber technology, mainly because many experiments and development work are required before a new sorbent is used in the absorber. In contrast, changing packing geometry (tube radius) and the absorber tower height are considered modification on the existing absorber technology.

Here, we use MEA as the default sorbent (Sb1), we include two other hypothetical sorbents, Sb2 and Sb3 respectively. Since each unique sorbent associates with an absorber technology, Sb1 associates with Absorber, Sb2 associates with Weak Absorber, and Sb3 associates with Strong Absorber. The main assumptions of the three sorbents are listed in Figure 5.2

<table>
<thead>
<tr>
<th>Sorbent</th>
<th>k2 (reaction constant)</th>
<th>Regen (Kcal/mol)</th>
<th>Cost ($/ton)</th>
<th>Associated absorber</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sb1</td>
<td>$k_2 = 4.4 \times 10^8 \times \exp(-5400/T)$</td>
<td>70.98</td>
<td>1250</td>
<td>Absorber</td>
</tr>
<tr>
<td>Sb2</td>
<td>$k_2 = 0.02 \times 4.4 \times 10^8 \times \exp(-5400/T)$</td>
<td>59.15</td>
<td>1000</td>
<td>Weak Absorber</td>
</tr>
<tr>
<td>Sb3</td>
<td>$k_2 = 50 \times 4.4 \times 10^8 \times \exp(-5400/T)$</td>
<td>94.64</td>
<td>1500</td>
<td>Strong Absorber</td>
</tr>
</tbody>
</table>
5.6.2 Vary the packing tube radius

By varying packing tube radius, we change the transport of \( \text{CO}_2 \) in the absorber tower, and the uptake rate. We assume the default tube radius is 32mm. We can vary the tube radius in each absorber type from 17mm to 34mm at the interval of 1mm.

5.6.3 Vary the absorber tower height

By varying absorber tower height, we change the uptake rate of \( \text{CO}_2 \) in the absorber, by changing the packing surface area. Various absorber tower heights also give different flue gas pressure drops in the absorber tower, hence change the kinetics. We assume the default absorber height is 30m. A tall absorber tower is 35m, a short absorber is 25m.

5.6.4 Vary the power generation technology

We consider three types of power generation technology, the subcritical power generation technology, the supercritical generation technology, and the ultra-supercritical power generation technology. The more efficient the power generation technology is, the more expensive it becomes. By default we use subcritical power generation technology. The basic assumptions of the power generation technologies are tabulated in Figure 5.3.

where

\[
\text{HHV} = \text{Higher heating value of coal} \\
\text{coal} = \text{coal consumption}, \text{this number changes as the configuration of the power plant}
\]
changes

Gross Size = gross power output of the plant

Size0 = reference plant size

Capcost0 = reference plant cost. The reference plant costs of supercritical and ultra-supercritical power generation unit are from DOE Parson report. [DOE, 2002].

If = \( \left( \frac{1}{n_i} \right)^\beta \), sf = \( \left( \frac{\text{size}}{\text{size}_0} \right)^\alpha \)

5.6.5 Three examples: small, medium and big problems

We study three different problems. In a small problem, we give an input of 6 designs for a sequence of 14 decisions. In a medium problem, we give an input of 87 designs for a sequence of 15 decisions. In a big problem, we give 492 designs for a sequence of 15 decisions. We use the branch and bound algorithm for the small problem. The forecast CO\textsubscript{2} regulation over time for the small, medium and big problems are tabulated in Figure 5.4. Capture percentage is the fraction of CO\textsubscript{2} captured as a total of CO\textsubscript{2} mass flow rate at the absorber inlet.
Figure 5.4: Forecast CO₂ capture percentage standards for small, medium and big problems

<table>
<thead>
<tr>
<th>Time</th>
<th>Small problem</th>
<th>Medium/Big problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5.00%</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10.00%</td>
<td>5.00%</td>
</tr>
<tr>
<td>4</td>
<td>15.00%</td>
<td>10.00%</td>
</tr>
<tr>
<td>5</td>
<td>18.00%</td>
<td>15.00%</td>
</tr>
<tr>
<td>6</td>
<td>20.00%</td>
<td>18.00%</td>
</tr>
<tr>
<td>7</td>
<td>23.00%</td>
<td>20.00%</td>
</tr>
<tr>
<td>8</td>
<td>25.00%</td>
<td>23.00%</td>
</tr>
<tr>
<td>9</td>
<td>30.00%</td>
<td>25.00%</td>
</tr>
<tr>
<td>10</td>
<td>33.00%</td>
<td>30.00%</td>
</tr>
<tr>
<td>11</td>
<td>35.00%</td>
<td>33.00%</td>
</tr>
<tr>
<td>12</td>
<td>38.00%</td>
<td>35.00%</td>
</tr>
<tr>
<td>13</td>
<td>40.00%</td>
<td>38.00%</td>
</tr>
<tr>
<td>14</td>
<td>43.00%</td>
<td>40.00%</td>
</tr>
<tr>
<td>15</td>
<td>45.00%</td>
<td>43.00%</td>
</tr>
</tbody>
</table>

We use the branch and bound algorithm for the small problem, and the heuristic for the medium problem and the big problem. For the heuristic, we assume x=2n, such that on every level, we remember a maximum total of 3n subpaths.

5.7 Results

In this section, we present the results of sequence optimization, and the results of single plant optimization in each decision point of the sequence.

- **Single plant optimization** gives the optimal single plant at time t (t = t₁, t₂, ... tₙ). It is a discrete optimization, in which the penalty of all designs at the corresponding decision time are calculated. The one with the lowest penalty is the optimal design. At each decision time t, one can find an optimal design. Since each decision is viewed as independent, the benefit of learning and cost of technology transitions are ignored. Therefore, the cost of the same design does not change when it is chosen repeatedly.
• **Local optimization** calculates the pathway penalty by putting the optimal single plant designs in a sequence. In this case, the cost of current decision depends on the entire history of the past decisions. The benefit of learning and costs of technology transitions are taken into account.

• **Global optimization** finds the optimal sequence of designs by comparing all possible sequences of designs. In this results section, all global optimal results are calculated using the heuristic described in section 4.6. As a result of the optimization, an earlier additional cost (a suboptimal single plant design) may prevent the sequence from locking in to the wrong path, which incurs a much bigger costs later.

### 5.7.1 The small problem: 6 design choices with 14 decision times

In this section, we present the results of a small problem, given 6 designs for a sequence of 14 decisions. The configuration and plant performance of the 6 designs are illustrated in figure 5.5. We assume that learning elasticity $\beta = -0.29$. The CO₂ emission weight is zero.

<table>
<thead>
<tr>
<th>Plant description</th>
<th>CP</th>
<th>Plant efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub_Absorber(TubeRadius 0.022)</td>
<td>34.59%</td>
<td>39.07%</td>
</tr>
<tr>
<td>Sub_StrongAbsorber(TubeRadius 0.017)</td>
<td>46.24%</td>
<td>39.05%</td>
</tr>
<tr>
<td>Sub_NoScrubber</td>
<td>0.00%</td>
<td>34.54%</td>
</tr>
<tr>
<td>Sub_Absorber(TubeRadius 0.032)</td>
<td>18.24%</td>
<td>39.07%</td>
</tr>
<tr>
<td>Sub_StrongAbsorber(TubeRadius 0.032)</td>
<td>18.26%</td>
<td>39.05%</td>
</tr>
<tr>
<td>Sub_WeakAbsorber(TubeRadius 0.036)</td>
<td>14.51%</td>
<td>39.09%</td>
</tr>
</tbody>
</table>

Figure 5.5: Plant performance data of the 6 designs
We use the branch and bound algorithm to find the optimal pathway. The optimization results are illustrated in Figure 5.6.

Figure 5.6: Single plant and pathway optimization, 6 design choices for 14 decision points

As illustrated in Figure 5.6, we can make following observations. Firstly, single plant optimization results do not account for learning and pathway penalties, therefore, the same designs chosen repeatedly have the same cost (i.e. \( t_6, t_7, t_8, t_9, t_{10} \)). In contrast, when the same sequence of designs take into account of learning and pathway penalties, as in local optimization, the cost of the same design decreases it is built repeatedly.

Secondly, the results of local optimization favors near-term benefit in comparison to global optimum. For example, at \( t_2 \) and \( t_3 \), the design penalties of the local optimum is smaller than global optimum.

Lastly, we observe an drastic spike in the penalty at \( t_{11} \) in the local optimum results. Whereas in the results for the global optimum, by choosing suboptimum
designs at \( t_2 \) and \( t_6 \), the global optimum path avoids the cost of lock-in at \( t_{11} \).

Plant configuration comparisons for global optimum and local optimum calculations are illustrated in Table 5.7. As illustrated in Fig 5.7, we can make following observations. In a global optimization, most designs share the same modules with variation in design parameters. For example, from \( t_2 \) to \( t_{14} \), all modules are configured with a subcritical power plant island and a strong absorber. In comparisons, in local optimization, the choice of absorber types changes frequently, hence resulting in less learning and more transitional costs, and therefore resulting with a higher global pathway penalty.

In summary, the implication of the results of the small problems are threefold. First, if one only chooses the best designs at each time, one may gain short-term cost efficiency. Secondly, if one fails to look ahead, one may be unintentionally locked into the wrong technology pathway which results in a much bigger cost later on. Thirdly,
Chapter 5: Optimizing CO$_2$ post-combustion capture technologies

by suffering a little extra cost upfront, one may gain a long-term benefit.

5.7.2 The medium problem: 84 design choices with 15 decision times

In this section, we expand the design pool to a total number of 84 designs. The configurations of a power plant can be chosen from three power generation units with three different efficiencies, three sorbent strengths, three absorber tower heights, and three absorber packing tube radius. In total, they form a pool of 81 plant configurations ($81=3\times3\times3\times3$). In addition, we include six power plant designs without CO$_2$ capture, in which three are power plants with one train with different efficiencies. We also include three power plant configurations without CO$_2$ capture at different efficiencies, all three designs assume two trains such that they are ready to integrate with CO$_2$ capture units in the future. We call such designs CO$_2$ capture ready designs. It is worth noting that, the root of all paths starts with a design without CO$_2$ capture units, and is CO$_2$ capture-ready. We use NoScrubber to represent the initial design.

In total, we have 84 designs at each decision time. The details of each category is listed in table 5.8.

<table>
<thead>
<tr>
<th>Power generation unit</th>
<th>Sorbent choice/absorber types</th>
<th>Absorber Tower Height</th>
<th>Absorber tube radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subcritical (Sub)</td>
<td>StrongAbsorber</td>
<td>T(35m)</td>
<td>0.017m</td>
</tr>
<tr>
<td>Supercritical (SC)</td>
<td>Absorber</td>
<td>M(30m)</td>
<td>0.022m</td>
</tr>
<tr>
<td>Ultrasupercritical (USC)</td>
<td>WeakAbsorber</td>
<td>S(25m)</td>
<td>0.034m</td>
</tr>
</tbody>
</table>

Figure 5.8: 84 design configurations
The enumeration of all possible pathways with 84 designs and 15 levels is 1e26, which is a factor of 1e14 compared to the small problem. We choose the heuristic, rather than branch and bound algorithm to find the good pathway in a reasonable time. We assume x=2n, such that on every level, a maximum total of 3n subpaths are remembered.

The optimization results are illustrated in Figure 5.9. Plant configuration comparisons for global optimum and local optimum are illustrate in Figure 5.10

As illustrated in Figure 5.9 and Figure 5.10, we can make following observations.

First, we observe four spikes in the local optimal pathway, specifically at t_5, t_10, t_13 and t_15. We also observe that there are technology transitions by examining the plant configurations at the corresponding times (Fig 5.10). For example, the configurations at t_5 is SC_M_WeakAbsorber(TubeRadius0.034), compared to the
previous design Sub_S_WakeAbsorber(TubeRadius0.034), two changes are made. For one, subcritical power generation technologies are abandoned. For another, new supercritical power generation technologies are introduced. It is worth noting that there are two power generation units in the designs with CO₂ absorber, one unit is mainly responsible for power production, and other operates at a different condition to supply heat for sorbent regeneration in the absorption system. In this case, design NoScrubber contains two power generation units, such that it is CO₂ capture-ready. When there are technology transitions to change the power generation unit, four units instead of two units are changed. In addition, at $t_5$, we observe that the absorber size increases from a small absorber tower to a medium absorber tower, this however is not a change in technologies, but rather a change of configurations of the existing technology.

Secondly, we find the evidence of strong learning in our model. For example, in the local optimum results, the designs at $t_2$ is a subcritical power plant without CO₂
absorber, the designs at $t_3$ is a subcritical power plant with a CO$_2$ absorber. By examining the single plant optimization curve, we find that the design at $t_3$ has a higher penalty than the design at $t_2$. However, on the local optimization curve, the design at $t_3$ has a lower penalty than the design at $t_2$. The reasons are two folds. First, the cumulative number of subcritical power generation unit increases quickly as there are two units in each design. At time $t_3$ the production history of subcritical power generation units is six units. Secondly, due to the strong learning, the cost savings of the power generation units is so significant that it dwarfs the cost of a new absorber unit. The same observation is found at $t_6$ and $t_{11}$, in which the benefit of learning overcomes the cost of technology transitions.

Thirdly, the global optimum favors long-term benefit by paying extra cost upfront. In the global optimum results, we observe the introduction of a CO$_2$ control unit as early as $t_2$, despite the fact that the CO$_2$ regulation does not take effect starting at $t_3$ (Fig 5.4). In doing so, more learning can be obtained for the rest of the pathway. In comparison, for the locally optimized results, the plant at $t_2$ is a subcritical power plant without a CO$_2$ control unit. The local optimum path gains a short-term cost efficiency by choosing the best design at the time, but since it fails to look ahead and prepare for future constraints, it suffers additional costs later in the pathway.

Lastly, the plant configurations of all elements in the global optimum are more uniform compared to the plant configurations of all elements in the local optimum. For example, in a global optimum calculation, there are only two technology transitions, once at time $t_2$, and once at time $t_{15}$. In comparison, in local optimum, there are six technology transitions, specifically at time $t_3$, $t_5$, $t_6$, $t_{10}$, $t_{11}$, $t_{15}$.
5.7.3 The big problem: 492 design choices with 15 decision times

In this section, we expand the design pool to a total number of 492 designs. The configurations of a power plant can be chosen from three power generation units with three different efficiencies, three sorbent strengths, three absorber tower heights, 15 absorber packing tube radii, which range from 17m to 34m at an interval of 1mm. In total, they form a pool of 486 plant designs (486 = 3X3X3X17). In addition, we include six power plant designs without CO₂ capture, in which three are power plants with one train with different efficiencies, three are power plants with two trains such that they are CO₂ capture-ready. It is worth noting that the root of all paths starts with a design without CO₂ capture units, and is not CO₂ capture-ready. We use NoScrubberNew to represent the initial design. In total, we have 492 designs at each decision time.

The enumeration of all possible pathway for 492 designs and 15 levels is 1e37, which is a factor of 1e27 larger when compared to the small problem. We choose the heuristic, rather than branch and bound algorithm to find the good pathway in a reasonable time. We assume x = 2n, such that on every level, a maximum total of 3n subpaths are remembered.

The optimization results are illustrated in Figure 5.11. Plant configuration comparisons for global optimum and local optimum are illustrated in Figure 5.12.

As illustrated in Figure 5.11, we can make following observations. First, the global optimum favors early costs for future benefit. By paying a little extra at time t₂ and t₃, the optimization gains the benefit of learning for the rest of the sequence. Further, as
Chapter 5: Optimizing CO₂ post-combustion capture technologies

Figure 5.11: 492 design results. The optimal cost of a single plant and a sequence of plants.

Figure 5.12: 492 design results. Plant configuration comparisons.

<table>
<thead>
<tr>
<th>Local optimal</th>
<th>Global Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Sub_NoScrubber_New</td>
<td>Sub_NoScrubber_New</td>
</tr>
<tr>
<td>2 Sub_NoScrubber_New_1</td>
<td>SC_NoScrubber_New</td>
</tr>
<tr>
<td>3 Sub_S_WeakAbsorber(TubeRadius 0.034)_13</td>
<td>SC_S_WendAbsorber(TubeRadius 0.032)_13</td>
</tr>
<tr>
<td>4 Sub_S_WeakAbsorber(TubeRadius 0.034)_14</td>
<td>SC_S_WendAbsorber(TubeRadius 0.034)_14</td>
</tr>
<tr>
<td>5 Sub_S_WeakAbsorber(TubeRadius 0.031)_15</td>
<td>SC_S_WendAbsorber(TubeRadius 0.034)_15</td>
</tr>
<tr>
<td>6 Sub_S_WeakAbsorber(TubeRadius 0.028)_16</td>
<td>SC_S_WendAbsorber(TubeRadius 0.030)_16</td>
</tr>
<tr>
<td>7 Sub_S_WeakAbsorber(TubeRadius 0.026)_17</td>
<td>SC_S_WendAbsorber(TubeRadius 0.028)_17</td>
</tr>
<tr>
<td>8 Sub_S_WeakAbsorber(TubeRadius 0.024)_18</td>
<td>SC_S_WendAbsorber(TubeRadius 0.026)_18</td>
</tr>
<tr>
<td>9 Sub_S_WeakAbsorber(TubeRadius 0.023)_19</td>
<td>SC_S_WendAbsorber(TubeRadius 0.025)_19</td>
</tr>
<tr>
<td>10 Sub_S_WeakAbsorber(TubeRadius 0.020)_20</td>
<td>SC_S_WendAbsorber(TubeRadius 0.022)_20</td>
</tr>
<tr>
<td>11 Sub_S_WeakAbsorber(TubeRadius 0.019)_21</td>
<td>SC_S_WendAbsorber(TubeRadius 0.020)_21</td>
</tr>
<tr>
<td>12 Sub_S_WeakAbsorber(TubeRadius 0.018)_22</td>
<td>SC_S_WendAbsorber(TubeRadius 0.020)_22</td>
</tr>
<tr>
<td>13 SC_S_WeakAbsorber(TubeRadius 0.018)_23</td>
<td>SC_S_WendAbsorber(TubeRadius 0.018)_23</td>
</tr>
<tr>
<td>14 SC_S_WeakAbsorber(TubeRadius 0.018)_24</td>
<td>SC_S_WendAbsorber(TubeRadius 0.018)_24</td>
</tr>
<tr>
<td>15 USC_S_WeakAbsorber(TubeRadius 0.017)_25</td>
<td>SC_M_WeakAbsorber(TubeRadius 0.018)_25</td>
</tr>
</tbody>
</table>
illustrated in Figure 5.12, the global optimum calculation chooses a supercritical unit as early as t2, and repeatedly chooses this module to gain the benefit of learning. In comparison, the local optimum calculation always chooses the current best technology. For the first 12 decisions, it repeatedly chooses the design with subcritical unit. At t13, the chosen design in the local optimum has a supercritical module, and at at t15, the chosen design has an ultrasupercritical module. Frequent changes of technologies increase the transitional cost, more importantly, one looses the benefit of learning in the first 12 years, when one has to choose a brand new technology starting from t13.

Secondly, the benefit of learning is greater than the cumulative cost difference between the supercritical units and the subcritical units. This may change, given a different learning rate with a different learning elasticity. The same problem under a lower learning elasticity is modeled and discussed in Section 5.8.2.

5.8 Discussion

5.8.1 Increased CO₂ emission penalty weight favors higher plant efficiency

In the medium problem presented in section 5.7.2, we assume the CO₂ quantitative penalty is zero. In this case, we increase the CO₂ quantitative penalty \( k_{CO2} \) to 30. The CO₂ quantitative penalty function is defined as Equation 3.5.

The optimization results are illustrated in Figure 5.13. Plant configuration comparisons for global optimum and local optimum are illustrated in Figure 5.14.

As illustrated in Figure 5.13 and Figure 5.14, we can make following observations.
Chapter 5: Optimizing CO$_2$ post-combustion capture technologies

Figure 5.13: 84 design results. The optimal cost of a single plant and a sequence of plants. $k_{CO2} = 30$

Figure 5.14: 84 design results. Plant configuration comparisons. $k_{CO2} = 30$

<table>
<thead>
<tr>
<th>Local optimal</th>
<th>Global Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Sub NoScrubber</td>
<td>Sub NoScrubber</td>
</tr>
<tr>
<td>2 Sub NoScrubber t2</td>
<td>USC NoScrubber t2</td>
</tr>
<tr>
<td>3 SC_S_Absorber Weak(TubeRadius 0.017) t3</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t3</td>
</tr>
<tr>
<td>4 SC_S_Absorber Weak(TubeRadius 0.017) t4</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t4</td>
</tr>
<tr>
<td>5 SC_S_Absorber Weak(TubeRadius 0.017) t5</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t5</td>
</tr>
<tr>
<td>6 SC_S_Absorber Weak(TubeRadius 0.017) t6</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t6</td>
</tr>
<tr>
<td>7 SC_S_Absorber Weak(TubeRadius 0.017) t7</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t7</td>
</tr>
<tr>
<td>8 SC_S_Absorber Weak(TubeRadius 0.017) t8</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t8</td>
</tr>
<tr>
<td>9 SC_S_Absorber Weak(TubeRadius 0.017) t9</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t9</td>
</tr>
<tr>
<td>10 SC_S_Absorber Weak(TubeRadius 0.017) t10</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t10</td>
</tr>
<tr>
<td>11 SC_S_Absorber Weak(TubeRadius 0.017) t11</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t11</td>
</tr>
<tr>
<td>12 SC_S_Absorber Weak(TubeRadius 0.017) t12</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t12</td>
</tr>
<tr>
<td>13 SC_S_Absorber Weak(TubeRadius 0.017) t13</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t13</td>
</tr>
<tr>
<td>14 SC_S_Absorber Weak(TubeRadius 0.017) t14</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t14</td>
</tr>
<tr>
<td>15 USC_S_Absorber Weak(TubeRadius 0.017) t15</td>
<td>USC_S_Absorber Weak(TubeRadius 0.017) t15</td>
</tr>
</tbody>
</table>
First, with a higher CO$_2$ quantitative penalty, there are motivations for utilities to upgrade to more efficient and more advanced power generation units. We observe that both local optimum and global optimum favor power generation units with higher efficiency compared to the results in Fig 5.10.

Secondly, from Figure 5.13, we can see by paying a little extra at time t2, the global optimum gains the benefit of learning over the rest of the decision time. At time t15, we observe a jump in the cost in the local optimum, reflecting its lock-in cost.

5.8.2 Decreasing learning elasticity

In this subsection, we run the optimization in the large problem, with a lower learning elasticity at $\beta = -0.19$. The results are presented below.

The optimization results are illustrated in Figure 5.15. Plant configuration comparisons for global optimum and local optimum are illustrated in Figure 5.16

As illustrated in Figure 5.15 and 5.16, we can make following observations.

First, a decrease in learning, changes the optimization results, compared with section 5.7.3. The local optimum plant configurations are still the same, with different path penalty as a result of a lower learning rate. However, the global optimum plant configurations are different as illustrated in Figure 5.16.

Secondly, a lower learning elasticity implies smaller benefit of learning. We observe that the benefit of learning is smaller than the cumulative cost difference between the supercritical units and the subcritical units. A lower learning rate gives less benefit in cost reduction from increased production, hence lowers the motivation to pay extra
Figure 5.15: 492 design results. The optimal cost of a single plant and a sequence of plants.

Figure 5.16: 492 design results. Plant configuration comparisons.

<table>
<thead>
<tr>
<th>Local optimal</th>
<th>Global Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Sub_NoScrubber_New</td>
<td>Sub_NoScrubber_New</td>
</tr>
<tr>
<td>2Sub_NoScrubber_New_t12</td>
<td>Sub_NoScrubber_New_t12</td>
</tr>
<tr>
<td>3Sub_S_WeakAbsorber(TubeRadius 0.034) t3</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.034) t3</td>
</tr>
<tr>
<td>4Sub_S_WeakAbsorber(TubeRadius 0.034) t4</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.034) t4</td>
</tr>
<tr>
<td>5Sub_S_WeakAbsorber(TubeRadius 0.031) t5</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.031) t5</td>
</tr>
<tr>
<td>6Sub_S_WeakAbsorber(TubeRadius 0.028) t6</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.028) t6</td>
</tr>
<tr>
<td>7Sub_S_WeakAbsorber(TubeRadius 0.026) t7</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.026) t7</td>
</tr>
<tr>
<td>8Sub_S_WeakAbsorber(TubeRadius 0.024) t8</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.024) t8</td>
</tr>
<tr>
<td>9Sub_S_WeakAbsorber(TubeRadius 0.023) t9</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.023) t9</td>
</tr>
<tr>
<td>10Sub_S_WeakAbsorber(TubeRadius 0.020) t10</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.020) t10</td>
</tr>
<tr>
<td>11Sub_S_WeakAbsorber(TubeRadius 0.019) t11</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.019) t11</td>
</tr>
<tr>
<td>12Sub_S_WeakAbsorber(TubeRadius 0.018) t12</td>
<td>Sub_S_WeakAbsorber(TubeRadius 0.018) t12</td>
</tr>
<tr>
<td>13SC_S_WeakAbsorber(TubeRadius 0.018) t13</td>
<td>Sub_M_WeakAbsorber(TubeRadius 0.018) t13</td>
</tr>
<tr>
<td>14SC_S_WeakAbsorber(TubeRadius 0.018) t14</td>
<td>Sub_M_WeakAbsorber(TubeRadius 0.017) t14</td>
</tr>
<tr>
<td>15USC_S_WeakAbsorber(TubeRadius 0.017) t15</td>
<td>SC_M_WeakAbsorber(TubeRadius 0.018) t15</td>
</tr>
</tbody>
</table>
cost upfront for more expensive (and more efficient) power generation technologies.

Lastly, by comparing the plant configurations in the global optimum in Figure 5.12 and Figure 5.16, we find that a lower learning favors subcritical power plant unit, whereas, a higher learning favors supercritical power plant unit.

5.9 Conclusion

We have the following findings based on the results presented in this chapter.

First, for all problems considered (i.e. the small, medium and big problems), the best technology pathways are different from and better than the pathway consisting of the best designs at all times. This finding suggest that, with an alternative perspective on technology development, we can provide new insights that facilitates more informed decision making in technology development.

Secondly, we observe that a \(\text{CO}_2\) quantitative penalty that penalizes residual emissions below the regulatory limit has a strong impact on the choice of the technologies. An increased \(\text{CO}_2\) quantitative penalty, encourages utilities to adapt more efficient power generation technologies. The sensitivity of \(\text{CO}_2\) quantitative penalty weights on the optimization is beyond the scope of this dissertation, due to the large number of parameters involved in the model. To study the sensitivity of \(\text{CO}_2\) quantitative penalty weights on the optimization, one should define a small problem with fewer parameters.

Lastly, we observe that learning has a strong impact on the choice of technologies, and the outcome of pathway optimization. Due to the lack of data for experience curves based on the number of unit productions, we test the model using two ar-
arbitrary values to show the impact of the learning elasticity. Given further research on the historic data of various modules, one can find the experience curve based on the number of unit productions, with the new learning elasticity, one can find more realistic optimization results.
Chapter 6

Designing the software tool for advanced power plant modeling and optimization

In this chapter, a software tool for advance power plant modeling and optimization is designed and developed. We are motivated to develop a new software, because the specifics of our problem cannot be solved readily with existing software and that the time sequence aspect of the problem is novel and not implemented in existing software. This is confirmed by the limitation observed in Chapter 5, where the existing software can not meet the requirement of the optimization exercise. This chapter will explain in detail the design and the development of the software tool which was used for the calculations presented in Chapter 5.

The software is very general and starts from describing a design in terms of a flowsheet. A flowsheet consists of an arrangement of modules and pipes. Here a
module is a basic operation unit (i.e. pressure pump, absorber, boiler, etc), and a pipe is the material and energy flow that connects two modules. Each pipe is connected on both ends. Some modules are very simple and may connect to an input device. For example, a furnace will be connected to an air intake. The air intake is a special module in that it only has an output, reflecting the fact that some modules are in effect open. A module can also be a derived module, which internally is described by another flowsheet, i.e., the derived module itself is built from modules and pipes. Some of the component modules in a derived module can themselves can be derived modules. A module is characterized by a set of parameters, some of the parameters are properties and flow rates that need to be matched to the flow rates in the connecting pipes, other parameters are known only to the module itself. They are internal parameters, which may be fixed, or could be varied in order to optimize a plant design.

The program starts with a user-provided input file which describes a power plant of modules and pipes characterized by a complete but usually inconsistent set of parameters. It is not expected that the user can correctly guess all the internal flows, but by making an educated guess, the user can achieve more rapid convergence in the iterations that the program runs to obtain an internally consistent set of parameters. In general, some of the flow fields are free to be set, while others may have to satisfy various conservation laws. The code is designed to accept certain parameters as fixed and others that can be varied until the system converges to an internally consistent answer. While the current implementation does not do this yet, the code is designed so that in future implementations the code can ascertain whether the system is over-
constrained or under-constrained and it creates an error state in either case.

Given a set of input parameters, the program then reconciles the inconsistencies in the over-determined parameters through an iterative process, until it finds the solution. The penalty associated with the reconciled power plant is then calculated. Up to this point, the program has calculated the penalty number for a physically consistent (coherent) power plant which operates in one specific condition and of one specific size.

To study the impact of parameters of interest on the power plant penalty, instead of specifying a fixed value for the parameter, one can relax the parameter and allow it to take on a set of discrete values bounded by an upper bound and a lower bound. The program calls the reconcile routine to find each possible physically consistent system provided that the parameters fall into the range specified above. Each reconciled system has one unique penalty number. The optimization routine compares the penalty numbers of all reconciled systems and finds the optimized system with the lowest penalty number. In this way, one can find the optimized plant design and the optimized operating conditions for systems of interest.

Similarly, one can use this method to find the optimum pathway for a sequence of power plant designs. First, reconcile will find the physically consistent set of parameters for each plant design. Then the penalty of the sequence will be calculated, where the pathway penalty is the sum of all plant level penalties in addition to pathway level penalties. We set a set of power plant designs as a discrete parameter, where each plant design is an element in the set. Optimization will run through each possible combination of elements in the set, calculate the pathway penalty for each
combination, and find the optimized pathway with the lowest penalty.

This chapter describes in detail the program design, and introduces its main functionalities. Currently seven modules and six pipes are developed. The module library includes a boiler, a steam turbine, a condenser, a pressure pump, a CO$_2$ absorber, a gas splitter, a generic source and a generic sink. The last two modules allow the plant to be open to the outside, by allowing input and output streams to connect to the outside world. The pipe library includes heat flow, work flow, flue gas flow, water flow, coal flow, and sorbent flow. It is anticipated that in future versions of this code, additional units are added to the system library.

6.1 Motivation

There is a lack of process modeling and simulation, in particular to evaluate the potential of CO$_2$ capture by various sorbents for various capture percentage, in comparison with solvents such as MEA, which is often aiming for 90-95% recovery of the CO$_2$.

When considering the integration of the capture and compression sub-processes into the overall process of a coal-fired power plant, the simulation of the CO$_2$ capture process and the power plant modeling is often carried out in two independent software systems respectively (i.e. Oexmann, 2008). Therefore, it becomes very difficult (unstable) to optimize an integrated design. Finding the optimum process parameters for the overall system (instead of only part of the system, i.e. the power plant, or the CO$_2$ capture process) that shows the lowest penalty is difficult. The existing software is often too brittle, and too inflexible to allow for a higher level optimization. Our
goal has been to develop this ability conceptually. Further work is necessary, to use the new algorithms for more complex systems with a larger library of modules.

Since the optimization does not consider the entire power plant, the optimization in flue gas scrubbing for CO$_2$ is often minimizing energy loss in CO$_2$ post-combustion capture process (due to the nature of such modeling tools). The capital cost of the overall integrated system is subsequently calculated based on the result of the optimization. In reality, business decisions are made often with the goal of minimizing cost, where the capital cost of the entire plant is not only a function of power loss, but also a number of other factors, for example the operating temperature of the furnace, the number of stages in steam turbine, etc.

Power plants today are faced with multiple competing objectives, in addition to the objective of minimizing the cost. To solve the problem that combines environmental constraints, infrastructural constraints, and energy constraints, in addition to economic constraints, and to make trade-offs between these various aspects of the problem, demand for a tool that allows users (firms and policy maker) to make informed decisions to choose the optimum power plant design, and the optimum pathways for building a sequence of power plants.

To satisfy the need and requirement stated above, a software tool which allows users to conduct multi-objective optimization on very complex and highly flexible system is developed.

We uses this software to find the optimum sorbent in post-combustion process for various capture percentage which would give the lowest penalty (environmental impact, infrastructural impact, energetic impact, and economic impact).
6.2 A model of modules and pipes

The purpose of "design" is to create a clean and relatively simple internal structure, sometimes also called architecture, for a program [Stroustrup, B., 2000]. This program is designed with a modules-and-pipes structure which I will explain in greater details below.

A module is a generic concept as illustrated in (Figure 6.1), the module can be at some level (for example it could be a single unit of operation, but it can also be a power plant). A module has a number of inputs and it has a number of outputs. A model (flowchart) of a power plant has only two fundamental components: they are modules and pipes, a fully developed plant can be thought of as combination (a network) of modules and pipes. For a flowchart of a power plant design, the blocks of the flowcharts are the modules, the streams and pipes are the lines (or streams) connecting them.

A module has to have the feature that it is connected to another by a number of pipes. For example, a furnace is a module that takes coal and air, and puts out heat and flue gas. In other words, it has two input streams and two output streams. The flue gas is the combustion product of coal and air, air is air, and for coal we can specify some ten parameters that are specific to coal, thus we define the furnace. The pipes connecting to the furnace are the coal coming in, the air coming in, the flue gas going to the stack, and the heat going out the other end.

We decide as a structural decision, that we have a handful of modules which only have pipes that going out, or pipes coming in. These are the sources and the sinks. For example, the atmosphere is a source module that is available to you as much as
you like. Since the model is developed this way, all pipes in a complete flow chart are attached to modules. Some pipes are connected to the end members of the system, which simply characterize the inputs and outputs of the plant.

A pipe can be thought of as a stream carrying material and/or energy, every pipe has to have the feature that both ends of a pipe is connected to a module. A pipe stores three copies of datasets, a copy of a dataset at the inlet of the pipe, a copy of a dataset at the outlet of the pipe, and an iterative copy of a dataset, with which the pipe reinitialize (writes to) both the inlet copy and the outlet copy after each iteration (Figure 6.2). Note the inlet of the pipe is the outlet of the block that the pipe connects to at the front end, the outlet of the pipe is the inlet of the block that the pipe connects to at the back end. For a flowchart to be internally consistent, the inputs and outputs to a pipe need to agree. Furthermore, the modules must satisfy a set of conservation laws. The reconcile operation on the flowsheet is an iterative procedure which assures that the system arrives in an internally consistent state.

We can describe any network (or flowcharts) as a set of pipes and modules, so every module has a name, every pipe has a name, and both ends of every pipe are connected to a module. We describe any power plant as a network with the format in the input file (Table 6.1). Once all modules and pipes are connected, we make sure that it is connected properly, by iteratively recalculating the inputs and outputs until every connection is converged (internally consistent). Convergence is achieved if the input and the output of a pipe agree. In the current implementation, the pipe’s input and output, which are based on the initial guesses provided by the module definitions, are replaced by some averaging procedure. This in turn changes
the values inside the modules, which need to recalculate their own behavior in order to make their relationship between inputs and outputs internally consistent. These numbers in turn will overwrite the values in the pipe, which typically results in a mismatch between the inflow and the outflow of the pipe. In general discrepancies will get smaller with every iteration, and the reconcile step will be repeated until the differences have become smaller than a tolerated error threshold, or alternatively, if the system fails to converge after a prescribed number of iterations. The pipe can also count and store the number of iterations before they are internally consistent.

The system is hierarchical, a module can be a network of modules and pipes itself. As a result, the structure of the program allows us to build very complicated things out of very simple things, hence making it easy to extend the complexity of a power plant design.
6.3 Input file

The program starts with a user-provided input file which describes a power plant of modules and pipes characterized by a complete but inconsistent set of parameters. The general format of an input file is given in Table 6.1. Figure 6.3 illustrates an instance of a flowsheet input file with one Furnace. As illustrated, there are four parts in the input file. The %FlowsheetParam section gives the user-provided preference of the model. The Block section describes the name and properties of the operational modules. The Stream section gives the name and the guessed value of the pipes. And finally, the Connect section gives the connecting instruction of the modules and pipes.
Figure 6.3: An instance of a flowsheet input file with one Furnace

```plaintext
%Flowsheet:Furnace
{
  %FlowsheetParams
  {
    iteration = 5000
    epsilon = 0.0001
  }
  %Blocks:6
  {
    GenericSource:CoalSource:
    {
      SetProperty
      {
        HighHeatingValue = 2.9e7
        OxygenDemand = 3.1
      }
    }
    GenericSource:AirSource:{SetProperty}
    Furnace:F1
    {
      PressureOut = 99000 #pascal
    }
    GenericSink:HeatSink:{SetFlow:3.0e8}
    GenericSink:AshSink
    GenericSink:FGS:
  }
  %Streams:5
  {
    Air:air:300
    Coal:coal:100
    FlueGas:fluegas:400
    Heat:heat:3.2e8
    Ash:ash:10
  }
  %Connect
  {
    # Connect input line format should be:
    # BlockName:Port:StreamName
    CoalSource:out:coal
    AirSource:out:air
    F1:Fuel:coal
    F1:Oxidant:air
    F1:Exhaust:fluegas
    F1:Heat:heat
    F1:Ash:ash
    FGS:in:fluegas
    HeatSink:in:heat
    AshSink:in:ash
  }
}```
6.4 Reconcile: solving a system of equations

6.4.1 Reconcile algorithm: numerical routine for solving a system of equations

A set of \( m \) parameters \( x_1, x_2, \ldots, x_m \) satisfies a set of equations as below.

\[
F_i(x_1, x_2, \ldots, x_m) = 0
\]

where \( i \) is the equation number, \( i \in [1, k] \), \( k \geq m \).

This can be simplified to

\[
F_i(x_j) = 0
\]

where \( j \in [1, m] \).

For a complete system, with the initial guess of all the parameters, we can obtain the equation as follows.

\[
F_i(x_j^{(0)}) = A_i^{(0)}
\]

After \( n \)’th iteration, we can obtain the equation as follows.

\[
F_i(x_j^{(n)}) = A_i^{(n)} \quad (6.1)
\]

Take \( F_i(x_j^{(n)} + \Delta x_j^{(n)}) \) for Taylor series expansion including the first derivatives, we can obtain the equation as follows.

\[
F_i(x_j^{(n)} + \Delta x_j^{(n)}) = F_i(x_j^{(n)}) + \sum_{j=1}^{m} \left( \frac{\delta F_i(x_j^{(n)})}{\delta x_j} \right) \times \Delta x_j^{(n)} \quad (6.2)
\]

Assume at \( n \)’th iteration the system converges, we can obtain the equation as follows.

\[
F_i(x_j^{(n)} + \Delta x_j^{(n)}) = 0 \quad (6.3)
\]
Given Equation 6.1, 6.2 and 6.3, we can obtain

$$
\sum_{i=1}^{m} \left( \frac{\delta F_i(x_j^{(n)})}{\delta x_j^{(n)}} \times \Delta x_j^{(n)} \right) = -A_i^{(n)}
$$

Equation(4) can be simplified to a system of linear equations of $\Delta x_j^{(n)}$:

$$
G_i(\Delta x_j^{(n)}) = 0
$$

For each iteration of $n$, one can use a linear solver (i.e matrix solve) to find $\Delta x_j^{(n)}$. Subsequently, one can solve for $x_j^{(n+1)}$ based on the following equation

$$
x_j^{(n+1)} = x_j^{(n)} + \Delta x_j^{(n)}
$$

The iteration continues, until $\sum_{j=1}^{m} |\Delta x_j^{(n)}| < \epsilon$. The choice of $\epsilon$ is somewhat arbitrary, and it reflects a compromise between accuracy and convergence time. Here, we assume $\epsilon = 0.1\%$. 0.1 is a generically conservative assumption.

### 6.4.2 Finding the optimized step for the true optimum

The steps of the optimization matter because they affect the rate of convergence. If the steps are too large, then the optimization might be unstable, and unable to converge. If the steps are too small, then the optimization might take too long to converge. One can modify the steps by introducing $\lambda_i$, and modify Equation (4) to

$$
\sum_{i=1}^{m} \left( \frac{\delta F_i(x_j^{(n)})}{\delta x_j^{(n)}} \times \Delta x_j^{(n)} \right) = -\lambda_i \times A_i^{(n)}
$$

Assuming $\lambda_i = 1$ is equivalent to directly solving Equation (4), but the system may be unstable (or fail to converge). $\lambda_i = \frac{1}{3}$ is a generically conservative assumption, it is more stable, but it needs more iterations.
One can further vary $\lambda_i$ such that

$$\min \| A_i^{(n+1)}(\lambda_i) \|$$

or

$$\min \sum_k \{ A_i^{(n+1)}(\lambda_i) \cdot A_i^{(n+1)}(\lambda_i) \}$$

where $\lambda_i \in [0, 1]$. When $\lambda_i = 0$, $A_i^{(n+1)}(\lambda_i) = A_i^{(0)}$.

Equation (6) is modified to

$$x_j^{(n+1)}(\lambda_i) = x_j^{(n)} + \lambda_i \times \Delta x_j^{(n)}$$  \hspace{1cm} (6.8)

If one plots $\| A_i^{(n+1)}(\lambda_i) \|$ against $\lambda_i$, the slope of the plot is negative because if it’s all differentiable, it starts with a negative slope. Typically the lowest point is neither $\frac{1}{3}$ or 1. The more non-linear it is, the further it is away from 1. If the problem is strictly linear, then the optimum is $\lambda_i = 1$.

### 6.4.3 Reconcile implementation

Let $M_{ij}^{(n)}$ be a k-by-k Jacobian matrix, where

$$M_{ij}^{(n)} = \frac{\delta F_i(x_j^{(n)})}{\delta x_j^{(n)}}$$

One need a linear solver to solve equation

$$C_i \times M_{ij}^{(n)} = B_j$$  \hspace{1cm} (6.9)

where

$$C_i = \Delta x_j^{(n)}$$

$$B_j = -\lambda_i \times A_i^{(n)}$$
\[ C_i = B_j \times (M_{ij}^{(n)})^{-1} \]

Once this is solved, one can solve for \( x_j \) iteratively.

### 6.4.4 Discussion of Newton-Raphson method

The method described above is essentially Newton-Raphson for finding successively better approximations to zeros (or roots) of a real-value function. In general, its convergence rate is quadratic (the error is essentially squared at each step), which means that the number of accurate digits roughly doubles in each step.

Although it’s a powerful technique, there are some limitations with the method. For example, the method may fail to converge if the derivative of the function is not continuous, or if the derivative is zero (where the tangent line overshoots the desired root), or if the initial guess is too far from true zero.

To increase the convergence stability, a number of studies (Press et al. (1992)) presented various improvements of Newton-Raphson method. However, the implementation of the improved method is beyond the scope of this work.

Another limitation on Newton’s method, comes from the requirement that the derivative need to be calculated directly. This is difficult in most practical problems, where functions may be given by a long and complicated formula, and hence an analytical expression for the derivative may be difficult to obtain. In these situations, it may be appropriate to approximate the derivative by using the slope of a line through two points on the function, or the Secant method. This has slightly slower
convergence than Newton’s method but does not require the existence of derivatives. This work chooses Newton-Raphson method because the current development is still relatively small scale, Secant method may be a consideration for future development.

### 6.5 Penalty

From software design point of view, penalty is rather simple. It consists of a hierarchy of penalty functions, which locate inside various objects (i.e. the flowsheets, modules and pipes), and return a penalty number. The penalty routine for a sequence of flowsheets is very similar. It is worth to point out two special routines for pathway penalty. The first routine updates the history of all modules regarding the module production in every flowsheet of the flowsheet list. The second routine compares the modules of any two neighboring flowsheets for number of the obsolete and new technologies, in order to calculate the transitional cost.

### 6.6 Parameters

Up to this point, the program has calculated the penalty number for a physically consistent (coherent) power plant which operates in one specific condition and of one specific size. One can choose to study one or a number of parameters of the design (i.e. plant size, furnace temperature, pressure, etc.), such that by varying the value of the parameters, one can find the optimal design or operation condition.

To study the impact of parameters of interest on the power plant penalty, instead of specifying a fixed value for parameters, one can relax the parameter to a set of
discrete values bounded by an upper bound and a lower bound.

6.7 Optimization

Upon setting the range for the chosen parameters, the program calls the reconcile routine to find each possible physically consistent system provided the parameters fall into the range specified above. Each reconciled system has one unique penalty number. The optimization routine compares the penalty numbers of all reconciled systems and finds the optimized system with the lowest penalty number. In this way, one can find the optimized plant design and the optimized operating condition for systems of interest.

In a flowsheet optimization, one always remembers the current optimum, and the corresponding parameter value. There is no need to create multiple copies for the same flowsheet. We can do the same for Flowsheetlist optimization, but the trade off is the efficiency, since this is an exhaustive search, for a large problem, this may take a long time. Currently, we use the algorithms introduced in Chapter 4, both algorithms require memory allocations for multiple copies of each flowsheet, they are efficient for problems with small and medium size options. However, when there are a large number of options at each decision time, this can be challenging, and one may need to trade off the efficiency for better memory management.
6.8 Output

The output of the software is designed to have the same format as the input file, but now with a consistent set of parameters describing the power plant configuration.

6.9 Test routines

To make sure the program runs as expected, we developed a set of test routines. For example, we can take the output of a run as a new input file, the new output compared with the first output should be the same, otherwise the test routine fails. To test all modules, we developed a set of basic input files, each describing a single module. In doing so, we can test all modules in the library before designing more complicated plant configurations.

6.10 Limitation

The advantage of optimizing over the entire integrated system is the most significant in greenfield plants. When it comes to retrofit, it may or may not be more advantageous than using two separate softwares, due to the limitation in how much improvement can be done on the existing power plants, especially if the existing plant is old. If the existing power plant is relatively new, changing operating conditions from the design operation conditions may be allowed, for older generation power plants, this may be too difficult hence one can get the same result using two separate softwares. In addition, retrofit is highly site-specific. It is yet another question (and still an open question) whether one is better off to retrofit the existing fleet or build
new power plants. From engineering point of view, it is likely advantageous to retire the existing fleet and build new plants, however, this will be subject to a range of factors like policies, public opinion, and the economics.
Chapter 7

Conclusion

In this dissertation, we propose and design a new methodology to make comparisons across different technologies and across different times, develop two optimization algorithms for solving path-dependent shortest path problem, apply the algorithm in the decision making of choosing various advanced power plant technologies, and devise a flexible software tool for modeling and optimizing complicated processes, by building a network of modular units.

We summarize our findings as follows.

• The simple example presented in Chapter 3 shows that using the novel evaluation method, we can select appropriate power plant modules and a wide range of technologies, to arrive at a sequence of plant designs that provide an advantageous technology pathway from today’s power plant designs to future designs via a number of intermediate steps.

• The computational results of the algorithms (Chapter 4) show that the performance of the path-dependent shortest path algorithms depends on the structure...
of the problem.

- For the typical power plant technology pathway, the branch-and-bound algorithm is more efficient compared with the brute-force search approach. In the worst case, the branch-and-bound algorithm degenerates into the brute-force search approach.

- Both branch-and-bound and the brute-force search are exact methods.

- For average problems, the heuristic is much more efficient than the branch-and-bound algorithm. However, the heuristic is not an exact method and does not guarantee that one finds the optimum, but it can find a good result in a reasonable time.

- The results of the heuristic are not monotonic with increasing memory allocation for a subpath on each level. This is due to the fact that the optimal results can be crowded out in early iterations.

- We apply these algorithms to study technology pathways which consist of power plant designs with CO$_2$ post-combustion capture technologies. We consider a small problem consisting of 6 designs and 14 levels, a medium problem consisting of 84 designs and 15 levels, and a big problem consisting of 492 designs and 15 levels. We use the branch and bound algorithm for the small problem, and the heuristic for the medium and big problems. The results of the small, medium and big problems show that, the best technology pathway, or the best sequence of technologies, do not agree with the sequence of best technologies of various times.
Firstly, for all problems considered (i.e. the small, medium and big problems), the best technology pathways are different from and better than the pathway consisting of the best designs at all times. This finding suggest that, with an alternative perspective on technology development, we can provide new insights that facilitates more informed decision making in technology development.

Secondly, we observe that a quantitative CO$_2$ penalty that applies to CO$_2$ emissions that are below the maximum permitted level has a strong impact on the choice of the technologies. An increased CO$_2$ quantitative penalty that could be driven by a price on carbon, encourages utilities to adapt more efficient power generation technologies. The sensitivity of a quantitative CO$_2$ penalty weights on the optimization is beyond the scope of this dissertation. Because of the large number of parameters involved in the model, it does not lend itself to such a study. To study the sensitivity of CO$_2$ quantitative penalty weights on the optimization, one should define a smaller problem with fewer parameters.

Lastly, we observe that learning has a strong impact on the choice of technologies, and the outcome of pathway optimization. Due to the lack of data for experience curves based on the number of unit produced, we test the model using two arbitrary values to show the impact of the learning elasticity. Given further research on the historic cost data of various modules, one could find the experience curve, with the new learning elasticity thus obtained, one could find more realistic optimization results.
– By paying a little extra cost (or choosing an suboptimal design) upfront, one can obtain a better technology pathway than the pathway with a sequence of best designs at various times.

– Sometimes, in doing so, one can prevent the technology from locking into a wrong technology pathway that incurs much bigger cost later on.

• We began the development of a flexible software tool which enables process modeling and optimization of complicated energy systems. The software is useful for modeling novel energy systems that cannot be done with traditional software tools.

In the future, we propose to study the sensitivity of all penalty weights on the optimization results. Better understanding of the learning curve of various modules is needed. We propose to develop better algorithms for solving problems with a very large number of design options, develop more modules for a zero emission power plant, apply the method in the decision making of other technology choices, for example renewable energy technologies. We can also use the model to study technology roadmap for the global energy demand-supply, and help policy makers make informed decisions.
Bibliography


NC: U.S. Environmental Protection Agency, Office of Air Quality Planning & Standards.


[Clausen, J., 1999] Clausen, J. *Branch and Bound Algorithms: Principles and Examples*, Class Notes Department of Computer Science, University of Copenhagen, Denmark. 1999


[Cormen, T. H., et.al. 2000] Cormen, Thomas H. and Leiserson, Charles E. and
Rivest, Ronald L. and Stein, Clifford *Introduction to algorithms* The MIT Press, 2000


[Harris, R. C., Rudd, J.W.M. 2007] Harris R. C., Rudd J.W.M. Whole-ecosystem study shows rapid fish-mercury response to changes in mercury deposition PNAS, 104, 2007


[Li, X. 2007] Li, X., Internal report on technoeconomic analysis of post-combustion capture technologies. Babcock & Wilcox


[Möller, B. et al, 2006] B. Möller, M. Assadi, I. Potts CO₂-free power generation in combined cycles - Integration of post-combustion separation of carbon dioxide in the steam cycle Energy, 31, 1520-1532


During the Past 200 years reconstructed by Deconvolution of Ice Core Data
Tellus 1987, 398, 140-154


[Tan, J. 2004] Tan, Jinsong and Leong, Hon Wai *Least-cost path in public transportation systems with fare rebates that are path- and time-dependent* Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC’04), Washington D.C., USA. 2004


Appendix A

Appendix

This dissertation is written in \LaTeX{} a typesetting program using \TeX{} language.

A.1 Introduction to \TeX{}

\TeX{} is a typesetting language invented by Donald Knuth. It is famous for a well designed and extremely efficient strategy and computational algorithm for ranking different typesetting layouts as more or less optimal [Knuth, 1981]. In this approach, an ideal typesetting layout is considered as the anchor point, the differences between the actual typesetting layout and the anchor point is penalized with a numeric penalty. Different aspects of the layout incur different penalties. The relative weights of these penalties can be chosen appropriately by a user, who has specific goals, as in the typesetting example, a specific aesthetic approach. The algorithm calculates the penalties and determines an optimal layout with the least penalty.

\TeX{} penalizes an actual layout compared to an ideal layout, in several aspects.
For example, if the word spacing in a paragraph is too dense, there is a penalty, if it is too sparse, there’s another penalty. If there is hyphenation, there is additional penalty. Users can choose their own weights on these penalties. For example, a user with weak vision may choose to put a relatively big weight on dense penalty, while preferring a sparse layout with large font. On the other hand, a user who’s motivated to save space, might put a relatively big weight on the sparse penalty, while show little concerns with a dense layout. An optimal layout for a user with weak vision might be a disaster for a user motivated to save space. T\TeX provides a method to allow users choose optimal layout with their own taste in a flexible and efficient manner. In this thesis, the T\TeX algorithm has served as motivation and inspiration for the algorithms implemented in finding an optimal power plant design or an optimal technology pathway, which can be viewed as a sequence of power plants.

A.2 Appendix: Performance data calculated using IECM Model

Plant performance data are calculated using the IECM model, all 18 plant designs are plants of 500MW gross output.
Appendix A: Appendix

Figure A.1: Power plant performance data from IECM model I

<table>
<thead>
<tr>
<th>Cycles</th>
<th>PC</th>
<th>SC</th>
<th>USC</th>
<th>PC+FGD+PC+SCR+ESP+FGD</th>
<th>ESP+FGD+ESP+FGD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>y0</td>
<td>y1</td>
<td>y2</td>
<td>y3</td>
<td>y4</td>
</tr>
<tr>
<td>Total O&amp;M M$/yr</td>
<td>49.55</td>
<td>47.17</td>
<td>44.88</td>
<td>58.24</td>
<td>59.97</td>
</tr>
<tr>
<td>Total Capex M$</td>
<td>570.1</td>
<td>607.3</td>
<td>609.5</td>
<td>642.6</td>
<td>666.6</td>
</tr>
<tr>
<td>Gross Eff LHV</td>
<td>38.60%</td>
<td>42.85%</td>
<td>47.10%</td>
<td>38.61%</td>
<td>38.61%</td>
</tr>
<tr>
<td>Generation Eff</td>
<td>84.16%</td>
<td>84.16%</td>
<td>84.16%</td>
<td>82.28%</td>
<td>92.10%</td>
</tr>
<tr>
<td>co2_out_gross lb/MWh</td>
<td>1.6020</td>
<td>1.824</td>
<td>1.4776</td>
<td>1.818</td>
<td>1.818</td>
</tr>
<tr>
<td>so2_out_gross lb/MWh</td>
<td>27.436</td>
<td>24.712</td>
<td>22.468</td>
<td>5.28</td>
<td>5.28</td>
</tr>
<tr>
<td>no_out_gross lb/MWh</td>
<td>3.0976</td>
<td>3.19</td>
<td>3.5934</td>
<td>3.5934</td>
<td>3.5934</td>
</tr>
<tr>
<td>no2_out_gross lb/MWh</td>
<td>0.25</td>
<td>0.1950</td>
<td>0.2094</td>
<td>0.2094</td>
<td>0.2094</td>
</tr>
<tr>
<td>h2_out_gross lb/MWh</td>
<td>7.0E-06</td>
<td>6.36E-05</td>
<td>5.79E-05</td>
<td>2.37E-05</td>
<td>1.83E-05</td>
</tr>
<tr>
<td>PM_out_gross lb/MWh</td>
<td>38.888</td>
<td>35.028</td>
<td>31.972</td>
<td>19.444</td>
<td>0.28512</td>
</tr>
</tbody>
</table>

Figure A.2: Power plant performance data from IECM model II

<table>
<thead>
<tr>
<th>PC+SCR+ESP+FGD+SCR+ESP+FGD+Hg</th>
<th>SC+SCR+ESP+FGD+SCR+ESP+FGD+Hg</th>
</tr>
</thead>
<tbody>
<tr>
<td>y5</td>
<td>y10</td>
</tr>
<tr>
<td>Total O&amp;M M$/yr</td>
<td>67.94</td>
</tr>
<tr>
<td>Total Capex M$</td>
<td>700.0</td>
</tr>
<tr>
<td>Gross Eff LHV</td>
<td>38.61%</td>
</tr>
<tr>
<td>Generation Eff</td>
<td>91.50%</td>
</tr>
<tr>
<td>co2_out_gross lb/MWh</td>
<td>1.8176</td>
</tr>
<tr>
<td>so2_out_gross lb/MWh</td>
<td>5.284</td>
</tr>
<tr>
<td>no_out_gross lb/MWh</td>
<td>0.8216</td>
</tr>
<tr>
<td>no2_out_gross lb/MWh</td>
<td>0.00550</td>
</tr>
<tr>
<td>h2_out_gross lb/MWh</td>
<td>4.88E-06</td>
</tr>
<tr>
<td>PM_out_gross lb/MWh</td>
<td>0.28512</td>
</tr>
</tbody>
</table>

Figure A.3: Power plant performance data from IECM model III

<table>
<thead>
<tr>
<th>SC+SCR+ESP+FGD+SCR+ESP+FGD+Hg</th>
<th>SC+SCR+ESP+FGD+SCR+ESP+FGD+Hg</th>
</tr>
</thead>
<tbody>
<tr>
<td>y12</td>
<td>y17</td>
</tr>
<tr>
<td>Total O&amp;M M$/yr</td>
<td>107.3</td>
</tr>
<tr>
<td>Total Capex M$</td>
<td>936.6</td>
</tr>
<tr>
<td>Gross Eff LHV</td>
<td>42.86%</td>
</tr>
<tr>
<td>Generation Eff</td>
<td>90.30%</td>
</tr>
<tr>
<td>co2_out_gross lb/MWh</td>
<td>0.18404</td>
</tr>
<tr>
<td>so2_out_gross lb/MWh</td>
<td>0.00540</td>
</tr>
<tr>
<td>no_out_gross lb/MWh</td>
<td>0.74</td>
</tr>
<tr>
<td>no2_out_gross lb/MWh</td>
<td>0.0448</td>
</tr>
<tr>
<td>h2_out_gross lb/MWh</td>
<td>4.49E-06</td>
</tr>
<tr>
<td>PM_out_gross lb/MWh</td>
<td>0.1194</td>
</tr>
</tbody>
</table>
Table A.1: Modules and the parameters that measure the module sizes

<table>
<thead>
<tr>
<th>Modules</th>
<th>Size Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>boiler</td>
<td>steam flow rate</td>
</tr>
<tr>
<td>steam turbine</td>
<td>steam flow rate</td>
</tr>
<tr>
<td>furnace</td>
<td>heat flow</td>
</tr>
<tr>
<td>condenser</td>
<td>steam flow rate</td>
</tr>
<tr>
<td>pressure pump</td>
<td>steam flow rate</td>
</tr>
<tr>
<td>CO₂ absorber</td>
<td>absorber column size</td>
</tr>
<tr>
<td>CO₂ stripper</td>
<td>stripper column size</td>
</tr>
</tbody>
</table>

Table A.2: Module Operating and Maintenance Cost breakdown

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Cost</td>
<td>( FixedCost = 0.2 \times \text{Capcost} )</td>
</tr>
<tr>
<td>Variable Cost</td>
<td>Consumables,FuelCost,etc</td>
</tr>
</tbody>
</table>

A.3 The basic assumptions in the module library

A.4 Testing model

This section describes the test model for Chapter 4. The six input designs in this test model are the same as the designs in Chapter 5 section 5.7.1.

NoScrubber

Absorber(TubeRadius 0.032)

Absorber(TubeRadius 0.022)

StrongAbsorber(TubeRadius 0.017)

StrongAbsorber(TubeRadius 0.032)

WeakAbsorber((TubeRadius 0.04).

The default heights of all absorbers are 30 meters. The default choice of power generation technology in all six designs are subcritical power generation technologies.
The penalty weights are given as follows:

Pathway levels = 14
Tree root = NoScrubber
EmissionWeight = 0
CostWeight = 1
PathWeight = 1
RemoveOldTechFactor = 10000
IntroNewTechFactor = 2e5
iteration = 5000
epsilon = 0.00001
CostWeight = 1
LearningFactor = -0.29

The CO$_2$ Absorber modules used in the test model does not account for steam consumption for sorbent regeneration. The details of these three modules are given in Section A.5.6.

Other modules (GenericSource, GenericSink, Splitter, PowerGeneration, Mixer) that are used in this example can be found in Appendix A.5. The pipes (streams) being used in this test model include: air stream, coal stream, fluegas stream, heat stream, ash stream, electricity stream, and sorbent stream.

The penalty functions are the same as the one described in Chapter 5. All costs assumptions are the same.
A.5 Module library

A.5.1 Generic Source and Generic Sink

Generic Source and Sink are the type of modules, which only have pipes that going out, or pipes coming in. They accept any types of streams without having to know the property of the stream. In the test module of Chapter 4 and the example in Chapter 5, we have use the following Generic Source and Sink units. AirSource, CoalSource, SorbentSource, SorbentSink, AshSink, FluegasSink, ElectricitySink, HeatSink.

A.5.2 Boiler

Boiler is a type that is connected with two input streams and one output stream. The two inputs are Heat and Water, the one output is Steam. The default setting of the boiler are as follows.

Temp = 873K
Pressure = 200e5 Pascal
Scaling factor = 0.75

The reconcile algorithm by which I implemented for the Boiler are as follows. Here we are applying a set of correction factors $(1-\alpha), (1+\alpha), (1-\beta), (1+\beta)$, etc, to the physical quantities, where these correction factors have been chosen such that the new values are internally consistent, i.e., satisfy mass and energy conservation laws. Then $ret$ is a measure of the size of the correction applied. We iterate until $ret$ is approaching zero.
\[ \Delta H = (\text{Steam Enthalpy}() - \text{Water Enthalpy}()) \]  
\[ \alpha = \frac{\text{Water}() - \text{Steam}()}{(\text{Water}() + \text{Steam}())}; \]  
\[ \beta = (\text{Heat Flux}()) - \frac{\text{Steam}() \ast \Delta H}{\text{Heat Flux}() + \text{Steam}() \ast \Delta H}; \]  
\[ \text{Heat Flux}() = \text{Heat Flux}() \times (1.0 - \beta); \]  
\[ \text{Water}() = \text{Water}() \times (1.0 + \beta); \]  
\[ \text{Steam}() = \text{Steam}() \times (1.0 + \beta); \]  
\[ a0 = (1 - \alpha) \ast (1 + \beta) \]  
\[ a1 = (1 + \alpha) \ast (1 + \beta) \]  
\[ a2 = (1 - \beta) \]  
\[ \text{ret} = \sqrt{(a0 - 1.0) \ast (a0 - 1.0) + (a1 - 1.0) \ast (a1 - 1.0) + (a2 - 1.0) \ast (a2 - 1.0))} \]  

where \( \text{ret} \) stands for return of the Boiler. The flowsheet summarize the return of all modules and streams, and iteratively finds the set of properties that describes a physically feasible design, by minimizing the total flowsheet return.

### A.5.3 Furnace

Furnace is a type that is connected with two input streams and three output streams. The two inputs are Fuel and Oxidant, the output streams are Exhaust, Heat, and Ash. The default setting of the Furnace are as follows.

\( \text{Temp} = 873\text{K} \)
PressureDrop = 1000.0 Pascal
Scaling factor= 0.75

The reconcile of Furnace modules considers five material and energy balance equations: oxygen conservation, energy conservation, nitrogen conservation, ash conservation, and normalization.

\[ A_0 \cdot \text{CoalFeed} \cdot \text{Oxygen} = A_1 \cdot \text{AirFlow} \cdot \text{AirOxygen} - A_2 \cdot \text{FlueGas} \cdot \text{FlueOxygen} \]  
(A.11)

\[ A_0 \cdot \text{CoalFeed} \cdot \text{HighHeatingValue} = A_3 \cdot \text{HeatFlow} \]  
(A.12)

\[ A_1 \cdot \text{AirFlow} \cdot \text{AirNitrogen} = A_2 \cdot \text{FlueGas} \cdot \text{FlueNitrogen} \]  
(A.13)

\[ A_0 \cdot \text{CoalFeed} \cdot \text{AshContent} = A_4 \cdot \text{Ash} \]  
(A.14)

\[ A_0 + A_1 + A_2 + A_3 + A_4 = 5 \]  
(A.15)

\[ m_{jk} \times A_k = B_j \]  
(A.16)

\[ ret = \sqrt{(A(0,0) - 1.0)^2 + (A(1,0) - 1.0)^2 + (A(2,0) - 1.0)^2 + (A(3,0) - 1.0)^2 + (A(4,0) - 1.0)^2} \]  
(A.17)

### A.5.4 PowerGeneration

PowerPlant is a type that is connected with one input stream and two output streams. The input is HeatIn, the output streams are Electricity, HeatOut.

The reconcile of PowerGeneration modules are as follows.

\[ A_0 \cdot \text{HeatFlux1()} \cdot \text{Eff} - A_2 \cdot \text{Elec()} = 0 \]  
(A.18)
\[ A_0 \times HeatFlux1() - A_1 \times HeatFlux2() - A_3 \times Elec() = 0 \]  
(A.19)

\[ A_0 + A_1 + A_2 = 3 \]  
(A.20)

Additional equation for mass balance:

\[ A_0 \times CoalFeed() + A_1 \times AirFeed() - A_2 \times FlueGas() - A_4 \times Ash() = 0 \]  
(A.21)

\[-m_{jk} \times A_k = B_j\]  
(A.22)

\[ ret = \sqrt{(A(0,0) - 1.0)^2 + (A(1,0) - 1.0)^2 + (A(2,0) - 1.0)^2} \]  
(A.23)

\[ HeatFlux1() = HeatFlux1() \times A(0,0); \]  
(A.24)

\[ HeatFlux2() = HeatFlux2() \times A(1,0); \]  
(A.25)

\[ Elec() = Elec() \times A(2,0); \]  
(A.26)

**Subcritical PowerGeneration**

\[ Eff = 0.6 \times (\text{HeatFlux1Temp()} - \text{HeatFlux2Temp()} )/\text{HeatFlux1Temp()} \]

**Supercritical PowerGeneration**

\[ Eff = 0.7 \times (\text{HeatFlux1Temp()} - \text{HeatFlux2Temp()} )/\text{HeatFlux1Temp()} \]

**Ultra-supercritical PowerGeneration**

\[ Eff = 0.8 \times (\text{HeatFlux1Temp()} - \text{HeatFlux2Temp()} )/\text{HeatFlux1Temp()} \]
A.5.5 Mixer, Splitter

Mixer and Splitter are generic modules that can split or mix any two number of streams. They accept any types of streams without having to know the property of the stream.

A.5.6 Absorber Modules

Absorber is a module that is connected with three input streams and two output streams. The inputs are Heat, FlueGas, CO2LeanSorbent, the output streams are Exhaust and CO2RichSorbent. The default setting of the PowerGeneration are as follows. The mass and energy functions that reconciles the absorber are given in Chapter 5.

A.6 Post-combustion capture technology

Post-combustion capture technology captures CO₂ in the flue gas by combining CO₂ in the flue gas medium with sorbent chemically or physically in an absorber, the resulting compound (or intermediate compound) then undergoes a process where the compound is broken down via applying pressure, heat, or conditioning the humidity, therefore allowing CO₂ to be captured in concentrated form.