MATHEMATICAL OPTIMIZATION TECHNIQUES FOR MANAGING SELECTIVE CATALYTIC REDUCTION FOR A FLEET OF COAL-FIRED POWER PLANTS

by

ANTONIO ALEJANDRO ALANIS PENA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2016

To my mother, Blanca Hilda Peña Rojas, and to the loving memory of my father, Luis Gerardo Alanis Salinas.

Acknowledgements

I would like to express my deepest gratitude to my advisor Dr. Jay M. Rosenberger. I really appreciate all his guidance, support, patience, motivation, and help during my doctoral studies; without his help, I would never have finished this project. My deep gratitude also goes to my committee members Dr. Victoria Chen, Dr. Melanie Sattler, and Dr. Bill Corley for their help, suggestions, and advice.

Additionally, I am very grateful to Anoop Ade for his help in my dissertation. In addition, I am thankful to all my COSMOS friends, especially to Dr. Aera LeBoulluec, Dr. Diana Martinez Cepeda, Dr. Nadia Martinez Cepeda, Dr. Kamil Alotaibi, Dr. Na Wang, Alireza Noroziroshan, Gazi Iqbal, Amin Gharehyakheh, and Nilabh Ohol for their help, advice and motivation.

Many thanks to the Mexican taxpayers and CONACYT (the Mexican National Council for Science and Technology), which granted me a scholarship to cover some of my educational expenses. Additionally, I want to be grateful to the company Viakable and to Sergio Peña-Rojas for their support and help.

This journey started with the great motivation I obtained from Dr. Francisco Vargas and Lily Treviño. I am very grateful for their encouragement. Many thanks to Ramses Romero, Victor Calderon, and Monica Macias for their help to improve my computational abilities and their support.

Here, I want to be grateful to my uncle Robert, my aunt Myrtha, and Pablo Paez for their constant help and motivation. In addition, I want to be thankful to all the professors that helped me in my life, especially, all the professors of the department of Industrial, Manufacturing, & System Engineering at the UTA.

Additionally, I am very thankful to my extremely good mother, Blanca Hilda Peña Rojas, my sister, Blanca Hilda Alanis Peña, and my brothers Arturo Javier Alanis Peña, Jose Alberto Alanis Peña, and Enrique Adrian Alanis Peña.

Neither Patricia Rodriguez nor Paula Umbarila will be outside of this acknowledges, I really appreciate their friendship and help. In addition, I want to be grateful to Julian Lizarazu for his support and advice.

Kindness is not easy to find, but I really want to be thankful to Julie Estill for her kindness and help during my studies.

From the bottom of my heart, I want to express my deepest gratitude to my wonderful wife and "mi propia amorcito" Gladys Orieth Rojas Lozano for her help, support, and motivation. It is a privilege to be your husband.

Until now I did not thank Dr. Jose Manuel Sanchez, Dr. Gerardo Mejia, and Dr. Jose Guadalupe Rios, but I really want to thank you to all of them for their help to start this journey.

Last but not least, I want to thank God for blessing me during my life and helping me to finish this project.

April 20, 2016

Abstract

MATHEMATICAL OPTIMIZATION TECHNIQUES FOR MANAGING

SELECTIVE CATALYTIC REDUCTION FOR A FLEET OF

COAL-FIRED POWER PLANTS

ANTONIO ALEJANDRO ALANIS PENA, PhD

The University of Texas at Arlington, 2016

Supervising Professor: Jay M. Rosenberger

Major commercial electricity generation is done by burning fossil fuels out of which coal-fired power plants produce a substantial quantity of electricity worldwide. The United States has large reserves of coal, and it is cheaply available, making it a good choice for the generation of electricity on a large scale. However, one major problem associated with using coal for combustion is that it produces a group of pollutants known as nitrogen oxides (NO_x). NO_x are strong oxidizers and contribute to ozone formation and respiratory illness. The Environmental Protection Agency (EPA) regulates the quantity of NO_x emitted to the atmosphere in the United States. One technique coal-fired power plants use to reduce NO_x emissions is Selective Catalytic Reduction (SCR). SCR uses layers of catalyst that need to be added or changed to maintain the required performance. Power plants do add or change catalyst layers during temporary shutdowns, but it is expensive. However, many companies do not have only one power plant, but instead they can have a fleet of coal-fired power plants. A fleet of power plants can use EPA cap and trade programs to have an outlet NO_x emission below the allowances for the fleet. For that reason, the main aim of this research is to develop SCR management mathematical optimization methods that, with a given set of scheduled

outages for a fleet of power plants, minimize the total cost of the entire fleet of power plants and also maintain outlet NO_x below the desired target for the entire fleet.

We use a multi-commodity network flow problem (MCFP) that creates edges that represent maintenance decisions of the SCR catalyst layers for each plant. At the beginning, the MCFP is relaxed because it does not consider average daily NO_x constraints, and it is solved by a binary integer program. After that, we add the average daily NO_x constraint to the model with a schedule elimination constraint (MCFPwSEC). The MCFPwSEC eliminates, one by one, the solutions that do not satisfy the average daily NO_x constraint and the worst NH_3 slip until it finds the solution that satisfies that requirement. Multi-cut MCFPwSEC targeting new layer speeds up the MCFPwSEC, because it eliminates many infeasible solutions at once instead of one by one. We introduce an algorithm called heuristic MCFPwSEC (HMCFPwSEC). When HMCFPwSEC algorithm starts, we calculate the cost of the edges estimating the average NH_3 slip level. After we have a schedule that satisfies the average daily NO_x constraint and the worst NH_3 slip, we update the cost of the edges with the average NH_3 slip for this schedule. We repeat this process until we have the solution. Since HMCFPwSEC does not guarantee optimality, we compare its results with SGO, which is optimal, using computational experiments. The solutions for both methods are very similar, but the time to solve each model is significantly different. Then, a fleet HMCFPwSEC (FHMCFPwSEC) uses HMCFPwSEC to create the SCR management plan for each plant of the fleet, with a discrete NO_x emissions value for each plant. FHMCFPwSEC repeats this process with different discrete levels of NO_x emissions, for each plant, in order to create a new problem with schedules with different cost and NO_x emissions for each plant of the fleet. Finally, FHMCFPwSEC solves this new problem with a binary integer program, which satisfies a NO_x emission value for the fleet, minimizes the total cost for the fleet, and uses each plant once. FHMCFPwSEC can work with single cut and also with multi-cut methods. Similar as HMCFPwSEC, FHMCFPwSEC does not guarantee optimality, and we compare its results with fleet SGO (FSGO) using computational experiments. The results for both models are very similar, but in the experiments, FHMCFPwSEC multi-cut targeting new layers always uses less time than FSGO.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

INTRODUCTION

Coal-fired power plants generated 37% of the total electricity consumed in the United States in 2012 [1]. There are various processes involved in generating electricity from coal, but the main process is to burn coal at high temperatures in a boiler to heat water so as to generate steam. Then, this steam is further used to spin one or more generators, or turbines, that produce electricity. The combustion gases from the boiler are sent to the stack in order to be emitted in the atmosphere [2]. One of the major problems associated with this process is that combustion of coal generates many pollutants and harmful gases. One such group of these pollutants is nitrogen oxides (NO_x). NO_x is formed in the boiler when nitrogen and oxygen from the atmospheric air react (thermal NO_x). It is also formed when nitrogen inside the coal reacts with the oxygen present in the atmospheric air (fuel NO_x) [3]. NO_x in atmospheric air can lead to death, serious respiratory illness in living beings, damage to forest ecosystems, and acidification of surface water [4]. In addition, NO_x contributes to regional haze, speeds up weathering of buildings, monuments, stone and metal structures, and contributes to coastal eutrophication [4]. The Environmental Protection Agency (EPA) regulates the emission of NO_x and other pollutants to the atmosphere [5]. In the United States, power plants need to comply with EPA regulations when they emit the combustion gases in the atmosphere. One technology that power plants use to satisfy NO_x EPA regulations is Selective Catalytic Reduction (SCR) [6].

SCR technology has been used by Japan and Germany for more than 25 years to reduce NO_x emissions [6]. SCR is a process that injects ammonia (NH_3) into the boiler flue gas stream that contains NO_x, and then these mixed components react in a catalyst layer where NH_3 and NO_x form harmless nitrogen (N_2) and water (H_2O) [6]. SCR is placed between the boiler and the stack of the power plants. The NO_x vented from the boiler and entering into the SCR is called *inlet NO_x*, and the NH_3 injected into the SCR is called *NH_3 injection*. Not all of the NH_3 injection and inlet NO_x react in the SCR; thus, the remaining NH_3 is called *NH_3 slip* and the persisting NO_x is known as *outlet NO_x*. Since outlet NO_x is then emitted to the atmosphere, it needs to satisfy EPA regulations.

Reactor potential (RP) of the catalyst in the SCR determines the quantity of NH_3 slip and outlet NO_x. At a constant NH_3 injection and inlet NO_x, high RP means less outlet NO_x and less NH_3 slip, whereas low RP means more outlet NO_x and more NH_3 slip. RP, as well as the catalyst itself, is degraded gradually over time. The NH_3 injection is increased in order to maintain the target outlet NO_x, but the consequence is that it also increases the NH_3 slip. The problem with greater NH_3 slip is that NH_3 is hazardous to living beings, changes the pH of water, forms particulate in the air, and can kill aquatic life [7]. In addition, NH_3 damages the SCR and is expensive; for these reasons, it is not convenient to overuse it [8].

Another way to maintain the targeted outlet NO_x is by adding or changing catalyst layers when the RP decays by a considerable amount. The layer addition or changes occur during a temporary shutdown of the power plant. Power plants have a plan of scheduled outages in order to maintain a good condition of the equipment of the power plant. The cost of changing or adding a catalyst is high, but it is even more expensive if the outage was not a scheduled outage. For this reason, companies prefer to add or change catalyst layers on a scheduled power plant outage.

Since there is a cost associated with NH_3 injection and for adding or changing catalyst layers, the minimum total cost of the SCR that satisfies NO_x regulations is a compromise between adding or changing catalyst layers on a scheduled outage, and increasing the NH_3 injection.

EPA cap and trade programs work in a way that, if a source reduces its emissions below the number of allowances (authorizations to emit, in our case NO_x emissions) it holds, then it may trade allowances with other sources in their system, sell them to other sources on the open market or on the EPA auctions, or bank them in order to use in future years [9]. Then, a company that has a fleet of power plants can minimize the cost of reducing pollution if the company did not need to use more than the allowances it has for the entire fleet. Consequently, it is very convenient to find an optimal SCR management plan with minimum cost that satisfies a predetermined upper limit on NO_x emissions for the fleet of plants given a plan of scheduled outages [10].

SCR catalyst vendors not only sell catalyst layers, but also they can help in designing and tuning the SCR. In fact, they also offer software that helps to minimize the cost of the SCR [11]. Phananiramai [10] observed that the software works for an individual plant basis. For that reason, he researched on the cost minimization of an entire fleet of power plants using optimization mathematical techniques.

Phananiramai [10] used two different methodologies to address the problem. He considered a schedule generation and optimization (SGO) algorithm, and he also used a multi commodity flow model (MCFP) algorithm. He showed a computational experiment with exactly the same input and the same result using both methodologies. The motivation to research in the second methodology was to save time, which he did.

A brief explanation of the methodologies used by Phananiramai [10] is as follows. Both methodologies assumed a given scheduled outage plan. The SGO algorithm enumerates all the feasible schedules and then selects the least expensive schedule. On the other hand, MCFP creates edges that symbolize actions on the SCR catalyst layers. The edges flow from the beginning of the time horizon and through all the outages until the end of the time horizon. Then the MCFP is solved using a 0-1 integer programming model. This MCFP is a relaxation because it does not consider the average daily NO_x constraint. After that, the average daily NO_x constraint is added to the MCFP model with a schedule elimination constraint (MCFPwSEC). The MCFPwSEC eliminates, one by one, each solution that does not satisfy the average daily NO_x constraint until it finds a solution that satisfies that requirement.

Phananiramai [10] shows that it is possible to use two different NO_x reduction policies: *fixed NH_3 slip policy* and *fixed NO_x policy*. The *fixed NH_3 slip policy* uses a constant NH_3 slip level, and NO_x percentage reduction can vary. The *fixed NO_x policy* uses a constant NO_x percentage reduction level and NH_3 slip can change. He studied both policies in the SGO computational experiments, but in the MCFP computational experiments, he only considered the fixed NO_x policy. One limitation of both policies is that they only work for a single plant at a time and not for a fleet of power plants.

SGO and MCFP minimize the total cost of a fleet of plants, but one limitation of Phananiramai's [10] study is that he considered a NO_x emission constraint for each single plant instead of a NO_x emission constraint for the entire fleet, which could be used to exchange allowances in a cap-and-trade program. The present research aims to address this limitation.

The contribution of this research is to develop SCR management mathematical optimization methods that, with a given set of scheduled outages for a fleet of power plants, minimize the total cost of the entire fleet of power plants and also maintain outlet NO_x below the desired target for the entire fleet.

The remainder of this dissertation is organized as follows. Chapter 2 presents a literature review. In the first section, we present an overview of SCR management. In the second section, we show a general overview of MCFP. In the third section, we present a very brief description of branch-and-price methodology and constraint programming. In the fourth section, we summarize the Phananiramai [10] dissertation, in particular his MCFP approach. In the fifth section, we present the contribution of this research.

In chapter 3 we explain the multi commodity flow model with schedule elimination constraints (MCFPwSEC) introduced by Phananiramai [10]. After that, we explain that we created a new model called heuristic multi-commodity flow problem with schedule elimination constraints (HMCFPwSEC). Unlike MCFPwSEC, our HMCFPwSEC is a heuristic that uses the average NH_3 slip of the schedule to calculate its cost. HMCFPwSEC does not guarantee optimality, but looks at multiple near optimal solutions. Then, we present computational experiments using eight different discrete percentage NO_x reduction levels. We solve three experiments for each of the discrete percentage NO_x reduction levels: two using HMCFPwSEC and one using SGO. We present some conclusions about the computational experiments.

In chapter 4 we use the HMCFPwSEC for a fleet of power plants (FHMCFPwSEC). To do that, we first explain that we cannot have a percentage NO_x reduction level for the fleet of power plants, but we can solve different discrete percentage NO_x reduction levels for each power plant. Then, the solution is a schedule with a minimum cost and outlet NO_x, corresponding to each of the discrete percentages NO_x reduction levels selected for each power plant of the fleet. Later, we create a problem with the outlet NO_x and costs for the entire fleet, and we minimize the total cost of the fleet satisfying a given maximum outlet NO_x for the fleet of power plants, and each plant used one time. We present eight computational experiments with an outlet NO_x emission constraint for a fleet of power plants. We compare their results to a computational experiment with the same outlet NO_x emission for the fleet, but with a percentage NO_x reduction level constraint pre-defined for each power plant alone. We show that the solution using our FHMCFPwSEC is less expensive than the final cost with the pre-defined percentage NO_x reduction level for each plant. For comparison, we solve the same computational experiments using the SGO used by Phananiramai [10] and fleet SGO (FSGO). We show that the greater difference between both models is the CPU time and wall clock time.

In chapter 5 we present conclusions and some areas of future research.

Chapter 2

LITERATURE REVIEW

The present chapter is divided into five parts. The first section gives a general overview of SCR management. The second section presents a very general overview of MCFP. The third section shows a very brief description of branch-and-price methodology and constraint programming. The fourth section summarizes the research of Phananiramai [10] with a special focus on his MCFP algorithms. The fifth section presents the contribution of the present research.

2.1 Overview of SCR management

Based on Cichanowicz and Muzio [12], the fixed design variables important for an SCR are: initial cost and activity of the catalyst, degradation of reactor potential and control of NO_x as well as NH_3, and catalyst addition or change in a scheduled outage. They show the difference in time, cost, and NO_x reduction between new, regenerated, and cleaned catalyst. A new catalyst layer reduces more NO_x, but it is also more expensive. A regenerated catalyst is less expensive, but it reduces less NO_x. The cheapest option is cleaning catalyst, but it gives the worst performance when reducing NO_x. New and regenerated catalysts need the same time to be added or replaced, but cleaned catalyst needs less time. With an example, they present how the cost changes by accelerating or delaying a catalyst change, and conclude that the best option is to minimize the total cost, not just the catalyst cost.

According to Staudt and Engelmeyer [13] and Wicker and Staudt [14], a comprehensive approach to SCR management needs a trade-off between catalyst consumption, frequency and duration of outages, ammonia slip, NO_x reduction, baseline NO_x, and pressure drop. In addition, they defined catalyst activity as the ability to facilitate

NO_x reduction reaction, but arsenic and other impurities in the gas stream decrease this ability. Furthermore, they stated that most SCR systems are designed with four levels of catalyst, and one level normally is empty when the system is new. Later, when the catalyst activity decreases and the ammonia needed is greater than an acceptable level, this empty level is filled. When the SCR system with the four levels filled needs an improvement in catalyst activity, then the catalyst with lower activity is changed. In a down flow reactor, usually the level empty for design is the lowest level, and the first level changed is the upper level.

Staudt et al. [15] showed the results of a survey of 25 different SCR systems, with 23 of them with a NO_x reduction equal or greater than 85%. With a few thousand hours of operation, they achieved the design NO_x reduction, and the overall reliability was satisfactory. The study points out that longer operation may show other issues.

According to Cichanowicz et al. [8], some lessons were learned about SCR in the U.S. since 1995. Some of the most important lessons are: SCR capital cost is higher than expected; NO_x reduction is near design targets; large particles of ash is a big problem in the U.S., but its solution has a modest cost; SO_2 oxidation is a problem and creates new restrictions on design and more operating cost; static mixers help to maintain NH_3/NO_x uniformity below 5% based on coefficient of variation basis; cleaning problems may justify sonic horns (cheap) and sootblowers (expensive); urea is an alternative to ammonia reagent; and SCR provides some mercury oxidation, which may be an important decision factor in the future.

Based on Erickson and Staudt [16] and Staudt and Erickson [17], 90% of NO_x reduction efficiency is achieved by an important portion of coal-fired power plants with SCR. It is possible to improve operations over time in NO_x reduction and in variability. In addition, the best controlled units maintained consistent behavior over time, but the less well controlled systems decreased performance over time.

According to Pritchard [18], effective catalyst management needs a long-term plan in order to maximize performance. A catalyst management strategy relies on the evaluation of different issues such as boiler/SCR system operations, the emissions reduction strategy, fuel management, outage demand, etc. In addition, to optimize the performance of the SCR, it needs the following key parameters: plant performance factors, operating conditions, and system scale-up factors.

Muzio et al. [6] presented the importance of catalyst, velocity distribution, NH_3/NO_x distribution, catalyst deactivation and sneakage in SCR management. The catalyst type, volume, and geometry are selected based on the particular application where the catalyst will be used. The velocity distribution is a key issue in order to maximize NO_x reduction and to minimize ash deposition and erosion. NH_3/NO_x distribution is important in order to obtain NO_x reduction greater than 85%. Catalyst deactivation is a crucial parameter because it helps to determine the SCR performance. Sneakage is a problem because it means that part of the flue gas bypasses the catalyst, and obviously it will degrade the SCR performance.

Kanniche et al. [19] introduced a model for a coal-fired power plant that uses SCR. Their model describes the physicochemical processes inside the reactor and the catalyst deactivation due to poisoning. They test the model with three cases and the NO_x reduction was perfectly modeled in one case but was underestimated in the two other cases. The plant they used as model follows a strategy that installs two catalyst layers. After several thousand hours, it adds a third layer, and after that, it changes the most deactivated layer. Following that strategy they correctly modeled the catalyst deactivation due to the aging of a catalyst.

Liu et al. [20] studied numerical simulations on an SCR system of a 1,000 MW coal-fired power plant. Using numerical simulation, they obtained the structure and size of the SCR system for four different designed schemes and the numerical calculation effectively guiding the design of the SCR. The calculations agree with the experiment, and the results of the simulations provided good reliability.

Xu et al. [21] used a computational fluid dynamics (CFD) simulation in a 300 MW SCR facility in one coal-fired power plant. They used FLUENT 6.3 to solve the partial differential equations. The simulation results were validated with experimental data and showed that velocity distribution and catalyst attrition can be improved.

Fossil Energy Research Corporation (FERCo) [10, 22] sells a software in spreadsheet based format called CatalysTraK-Manage (also known as CatReact). That tool determines better times to change or add the catalyst, provides options based on the outage schedule in the power plant, and calculates the economic consequences of the different options.

Tian and Jin [23] created a mathematical model for SCR NO_x reduction and regenerative heat exchange. Using this model they created a software of four modules written in C# based on Visual Studio 2010 and made some simplifications in the computation procedure. The results show that catalyst has a desirable heat storage property. Additionally, the model does not perfectly match the experiments, but the simulation results follow the same trend as the experimental results.

Phananiramai [10] did not find a previous work that tried to model an SCR management plan for a fleet of plants. The recent literature also did not show research in this area but shows better ways to design and maintain the SCR in good condition. The total cost of the SCR of a fleet of power plants not only depends on the SCR design and condition of the hardware, but also on the management of the SCR. Hence, the present research wants to help in the management part of minimizing the total cost for a fleet of plants while satisfying a constraint on the outlet NO_x emissions for the fleet.

2.2 Overview of MCFP

A problem that determines an optimal path flow between a source node and a sink node is called a network flow problem. A path has a set of arcs (edges) that, after satisfying some restrictions, join a source node and a sink node. In an edge, the source node is called the tail and the sink node is called the head [24]. A directed network only has arcs that allow positive flows only in one direction. [25].

The minimum cost flow problem is a common type of network flow problem. The goal of this problem is to find the lowest cost to send a commodity in a network and satisfy the demand at each node. The multi-commodity network flow problem or multi-commodity flow problem (MCFP) is a problem that moves several commodities simultaneously in the network [26, 27]. MCFP is used to solve many different practical problems; some examples are: public railway transportation [28], energy transportation [29], telecommunication networks [30], the design of supply chain networks [31], and shortest route problems [25].

Some of the earliest research in multi-commodity flow networks were presented in the 1950's and 1960's by Kantorovich [32], Ford and Fulkerson [33], Hu [34], Tang [35] and Tomlin [36]. There are many more recent works in this area, but we only present an overview of some of them.

Goffin et al. [37] showed that the analytic center cutting plane method can solve large non-linear multi-commodity flow problems. They presented the influence of some aggregation and disaggregation techniques and also showed that their method is faster than a Dantzig-Wolfe algorithm. They used several large problems and two small problems as examples to prove the efficiency of their algorithm.

Ben-Ameur and Neto [38] proposed an in-out algorithm that carefully selects the separation point in a cutting plane and column generation algorithm. They assumed that they have an exact separation algorithm, and that the problem is convex. The paper shows the convergence of the algorithm and applies the in-out algorithm to three problems: survivable network design, multi-commodity min-cost flow problem and random linear programs. In the computational experiments of the multi-commodity min-cost flow problem, the value of alpha that minimizes the time depends on three factors: the time consumed by the separation procedure, the time spent on solving the relaxations, and the quality of the cuts generated. They propose more investigation in order to find the best value for alpha.

Nabona [39] used a multi-commodity network to solve the long term hydro-generation problem when the company has thermal units and also hydro units. The problem has a probability density function that he approximated with three blocks. He created a model for total dependence and also a model for partial dependence. He used MINOS to solve numerical examples that showed a good approximation to reality.

Hane et al. [40] presented a fleet assignment model. They use a large multi-commodity flow problem with side constraints defined on a time-expanded network. Their methodology solves a 150-city, 2500-flight, eleven-fleet daily fleet assignment problem routinely in less than one hour.

Pilla et al. [41] presented a fleet assignment model using a two stage stochastic program and the concept of demand driven dispatch. In addition, they reduce computation using design and analysis of computer experiments and multivariate adaptive regression splines. They presented a case study using real airline information and obtained a very good fit.

Sadjady and Davoudpour [42] researched on designing a two-echelon supply chain network. They modeled the problem as mixed integer programming, trying to minimize the total cost of the network. They used a heuristic procedure based on Lagrangian relaxation and showed that their algorithm is effective and efficient for small and large problems.

Except for Phananiramai [10], the recent literature did not show the use of MCFP to model and solve an SCR management plan for a fleet of power plants. We will use the MCFP model of Phananiramai [10] in order to minimize the total cost for a fleet of power plants while satisfying the outlet NO_x emissions for the fleet.

2.3 Branch-and-price and constraint programming.

A general idea to formulate branch and price started with Dantzig and Wolfe [43] when they presented what is known today as Dantzig-Wolfe decomposition, and with Gilmore and Gomory [44] when they solved the cutting-stock problem. Branch and price follows a similar procedure as branch and cut, but instead of row generation, it uses column generation [45].

In branch and price, some sets of columns are not included in the linear programming (LP) relaxation, because there are a large number of columns and most of them will have their respective variable equal to zero in an optimal solution. Later, in order to check the optimality of the LP, a separate problem for the dual LP, known as the sub-problem or *pricing problem*, is solved in order to identify columns, which will enter the basis. If there are such columns present, then the LP is reoptimized. The branching process is done when no columns price out to enter the basis, and the integrality conditions are not satisfied by the LP solution [45].

There are many practical applications of branch and price. Salani and Vacca [46] used branch and price to solve the Discrete Split Delivery Vehicle Routing Problem with Time Windows. Dayarian et al. [47] solved a class of multi-period vehicle routing problem utilizing branch and price. Gunnerud et al. [48] used a branch and price framework to optimize production in an oil and gas field. Brunner and Stolletz [49] utilized branch and price to address the problem of staff scheduling at check-in counters with time varying demand. Belien and Demeulemeester [50] used branch and price to solve to optimality the trainee-scheduling problem in a hospital. Pereira Lopes and Valerio de Carvalho [51] utilized branch and price to solve to optimality the problem of scheduling parallel machines with sequence dependent setup times. Robenek et al. [52] used a branch-and-price framework to solve to optimality the integrated berth allocation and yard assignment problem in bulk ports.

Constraint programming is a relatively new approximation method in the realm of linear programming. Its use started late in the decade of 1960, and it was restricted to very specific applications. In the 1990’s, the programs were much better than before because they were suited for a greater variety of applications [53]. Constraint programming wants to program in a declarative way the constraint satisfaction problems. Constraint satisfaction problems have a set of variables, a domain of values for each variable and constraints among sets of variables. The constraints allow some combinations of value assignments, and a solution is obtained which satisfies all the constraints with an assigned value for each variable. With the help of more powerful solvers, constraint programming will offer more complicated applications in the future [54].

2.4 Summary of Phananiramai [10] research.

Phananiramai [10] studied an SCR management planning method that minimizes the cost of an entire fleet of power plants. He used two different methodologies to address this problem: a schedule generation and optimization (SGO) algorithm, and a multi commodity flow model (MCFP) algorithm. The SGO algorithm was used to minimize cost or NO_x emissions whereas the MCFP was used only to minimize cost. For both methodologies he assumed a given *scheduled outage plan* (calendar with dates when the power plant will have an outage).

The SGO algorithm enumerates all the feasible schedules and then selects the least expensive schedule or the schedule that minimizes NO_x emissions. The selection depends upon whether SGO tries to minimize cost or to minimize NO_x emissions. The SGO algorithm is divided into two main modules: SCR schedule generation and SCR optimization. The first module enumerates a set of possible outage schedules for all plants in the fleet. The second module uses Computational Infrastructure for Operations Research branch and cut (COIN-OR CBC), a 0-1 large scale integer programming solver, to select the least expensive schedule or the schedule that minimizes NO_x emissions.

SCR schedule generation creates possible outage schedules using a fixed NH_3 slip policy or a fixed NO_x policy. The *fixed NH_3 slip policy* uses a constant NH_3 slip level, and NO_x percentage reduction can vary. The *fixed NO_x policy* uses a constant NO_x percentage reduction level, and NH_3 slip can change. One limitation of both policies is that they only work for a single plant at a time and not for a fleet of plants. For the SGO algorithm, Phananiramai [10] uses a fixed NH_3 slip policy in two computational experiments and a fixed NO_x policy for one computational experiment.

On the other hand, MCFP creates edges that symbolize maintenance decisions for the SCR catalyst layers. The edges flow from the beginning of the time horizon and through all the outages until the end of the time horizon. Then the MCFP is solved using a 0-1 integer programming model. This MCFP is relaxed because it does not consider the average daily NO_x constraint. After that, the average daily NO_x constraint is added to the model with a schedule elimination constraint (MCFPwSEC). The MCFPwSEC eliminates, one by one, the solutions that do not satisfy the average daily NO_x constraint until it finds the solution that satisfies this requirement. In order to reduce the time for the MCFPwSEC, Phananiramai [10] introduces a multi-cut MCFPwSEC, which can eliminate many infeasible solutions at once instead of one by one.

For each of the two MCFPwSEC versions, he provides a computational experiment using a fixed NO_x policy and the same information of the SGO computational experiment that uses fixed NO_x policy. For both MCFPwSEC versions, he obtained exactly the same results as with the SGO but in less wall clock time, and the multi-cut MCFPwSEC uses less wall clock time of the three algorithms.

2.5 Contribution

SGO and MCFPwSEC successfully minimize the total cost for a fleet of power plants, but one limitation of Phananiramai's [10] research is that he considered an outlet NO_x emission constraint for each individual plant instead of an outlet NO_x emission constraint for the entire fleet.

For each plant, the fixed NO_x policy uses a constant *NO_x reduction level percentage*, which is the reduction from the inlet NO_x to the SCR to the outlet NO_x of the SCR as a percentage of the inlet NO_x. However, we cannot apply the same for the fleet because the percentage NO_x reduction level only works for each plant. We can use different percentage NO_x reduction levels for each plant of the fleet that combined give us the same outlet NO_x for the entire fleet. To do that, we use a NO_x reduction level in percentage for each plant in order to obtain the cost and the outlet NO_x emission for that particular plant and percentage NO_x reduction level. We repeat this process many times, changing the percentage NO_x reduction level. In the end, we have a new problem with many costs and outlet NO_x emissions for each plant. Solving this new problem, we minimize the cost of the fleet with an outlet NO_x emission constraint for the entire fleet.

Furthermore, the contribution of this research is to develop SCR management mathematical optimization methods that, with a given set of scheduled outages for a fleet of power plants, minimize the total cost of the entire fleet of power plants and also maintain outlet NO_x below the desired target for the entire fleet. In addition, the plants have fixed NO_x at several discrete reduction levels (within regulation), the total outlet NO_x constraint is across all plants, and the cost depends on average NH_3 slip of the entire schedule, not an upper bound on NH_3 slip. Unlike SGO, we propose a fleet SGO (FSGO) that considers an outlet NO_x emissions limit for the entire fleet of power plants not for each power plant. Unlike MCFPwSEC, we propose a heuristic fleet MCFPwSEC (FHMCFPwSEC) that considers an outlet NO_x emissions limit for the entire fleet of power plants not for each power plant. Observe that, not only we can apply the methods of this research for coal-fired power plants, but also we can use them for power plants with different fuels, transportation problems, energy distribution, etc.

Chapter 3

OPTIMIZATION OF A SINGLE COAL-FIRED POWER PLANT

This chapter has been divided into three sections. In section 3.1, we describe the multi-commodity flow problem with schedule elimination constraints (MCFPwSEC) introduced by Phananiramai [10]. The explanation does not pretend to be an exhaustive one, but instead, we want to explain the most important part of it in order to use it for our model. We will use the same formulae used by Phananiramai [10] in order to calculate cost and percentage NO_x reduction in our model. We cannot describe in detail the formulae because they are proprietary, but we will show some generalizations to explain our model.

In section 3.2, we explain our utilization of NH_3 slip in the MCFPwSEC using the fixed NO_x policy. We show that with our utilization we created a new model called heuristic multi-commodity flow problem with schedule elimination constraints (HMCFPwSEC). Unlike MCFPwSEC, our HMCFPwSEC is a heuristic model that uses the average NH_3 slip of the schedule to calculate its cost. HMCFPwSEC does not guarantee optimality, but it looks at multiple near optimal solutions.

In section 3.3, we present computational experiments using eight different discrete percentage NO_x reduction levels. We solve three experiments for each of the discrete percentage NO_x reduction levels: two using HMCFPwSEC and one using SGO. We present some conclusions about the computational experiments.

3.1 Multi-commodity flow problem with schedule elimination constraints.

In his research, Phananiramai [10] first introduced the relaxed MCFP and later discussed the MCFPwSEC. Then, we follow the same reasoning in our research. An exhaustive explanation of both models is given in [10].

The assumptions of the model of Phananiramai [10] are

1) The NO_x reduction involves catalyst causing the reaction of NH_3 and NO_x to reduce outlet NO_x as well as NH_3 slip.

2) As reactor potential decreases, NO_x reduction decreases and NH_3 slip increases. In general, NO_x reduction is a function of allowance from NH_3 slip and reactor potential from the catalyst. In Phananiramai [10] research, NH_3 slip is kept constant and therefore NO_x reduction is strictly a function of reactor potential. Although the details of this function are proprietary, it is a function in which NO_x reduction increases with reactor potential.

3) Regenerated catalyst layer is less expensive but also has less reactor potential than a new catalyst layer, and cleaned catalyst layer is the least expensive but also has the least reactor potential.

4) The model can correctly predict the values of outlet NO_x, inlet NO_x, NH_3 injection, and NH_3 slip.

5) The cost can be determined *a priori* based upon the actions done on the layers within the outages. In reality, there are some operational costs based upon usage of NH_3 injection and NO_x emissions. However, these costs are relatively small. In the cost calculated for this particular model, Phananiramai [10] estimates NO_x emissions reduction from the maximum average daily NO_x emissions in the model and a constant NH_3 slip value.

6) The layer assets are available and also the data needed for the formulae of the model.

7) The only way to obtain the average daily NO_x reduction of a schedule involves integrating NO_x reduction over the time horizon when the schedule is given in its entirety.

8) Due to time constraints on outages, only one layer may be changed or added in each outage.

9) The cost difference between adding a catalyst layer to an empty SCR slot and changing a catalyst layer is *disposal cost* of the existing layer, which is the same for all layers.

Now, we describe the relaxed MCFP. Relaxed MCFP generates edges that represent all SCR catalyst layers flowing from the start of the time horizon and through outages until the end of the time horizon. The edges only go in a forward direction and represent a layer and an action that can be taken. The edges need to maintain the RP greater than or equal to the RP of the minimum instantaneous NO_x reduction requirement. The variable vector x represents the edges that go from one node to another. The nodes can represent: the start of the time horizon, the outages, and the end of the time horizon. The RP and cost corresponding to each edge is calculated after the edge is generated, where RP_{ija}^{l} is the reactor potential between two consecutive outages i and j where action a is taken on layer l in outage i, and C_{ija}^{l} is the cost incurred between two consecutive outages i and j where action a is taken on layer l in outage i. Then, for each edge a 0-1 decision variable determines whether the edge is used in the solution plan.

The binary multi-commodity network flow model can be constructed as follows:

Nodes:

1) Create a sink node for each slot of the plant at the end of time horizon.

2) Create a source node for each slot of the plant at the start of the time horizon.

3) Create an intermediate node for each slot of the plant at all the possible outages in the time horizon in chronological order.

Arcs:

1) Create an arc from the start node of a slot to the sink node of the same slot.

2) Create an arc from the start node of a slot to each intermediate node of the same slot.

3) Create an arc for each of three actions from each intermediate node of a layer to each intermediate node of the same layer if and only if the tail of the arc starts in a node with a date prior to the date of the node where the head of the arcs arrives.

4) Create an arc for each of the three actions from each intermediate node of a layer to the sink node of the same layer.

If the slot has a layer, we only can change that layer, then, the three possible actions are: Change New layer, Change Regenerated layer, and Change Cleaned layer. If the slot does not have a layer, we only can add a layer, and then the three possible actions are: Add New layer, Add Regenerated layer, and Add Cleaned layer.

List of parameters used:

d = minimum value of average daily NO_x reduction to obtain.

O = Set of all outages, o denotes an outage in the Set O.

A = Set of all actions, where a represent an action in the Set A.

L = Set of all layers, where l is a layer in the Set L.

$El(i)$ = Set of edges from the node outage i in the sub-network layer l.

S = Set of all source nodes.

T = Set of all sink nodes.

RP_{ija}^{l} = reactor potential between two consecutive outages i and j and action a is taken on layer l in outage i.

C_{ija}^{l}= total cost incurred between the two consecutive outages i and j and action a is taken on layer l in outage i.

$f(x)$ = Average daily NO_x reduction.

List of variables used

x_{ija}^{l} = 1 if two consecutive outages i and j are used and action a is taken on layer l in outage i, and 0 otherwise.

The 0-1 integer program to solve the SCR management problem is given by equations (3.1) to (3.8).

$$min \sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{(i,j) \epsilon E} C_{ija}^{l} x_{ija}^{l} \tag{3.1}$$

s.t.

$$\sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{(j,k) \epsilon E_l(i)} RP_{jka}^{l} x_{jka}^{l} \geq min\, RP \qquad \forall i \in O \tag{3.2}$$

$$\sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{j | \exists (i,j) \epsilon E} x_{ija}^{l} \leq 1 \qquad \forall i \in O \tag{3.3}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (i,j) \epsilon E} x_{ija}^{l} = \sum_{a \epsilon A} \sum_{j | \exists (j,i) \epsilon E} x_{jia}^{l} \qquad \forall i \in O, l \in L \tag{3.4}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (s,j) \epsilon E} x_{sja}^{l} = 1 \qquad \forall l \in L,\ s \in S,\ t \in T \tag{3.5}$$

$$\sum_{a \epsilon A} \sum_{j | \exists (j,t) \epsilon E} x_{jta}^{l} = 1 \qquad \forall l \in L,\ s \in S,\ t \in T \tag{3.6}$$

$$x_{ija}^{l} \in \{0,1\} \qquad \forall (i,j) \in E,\ l \in L,\ a \in A \tag{3.7}$$

$$f(x) \geq d \tag{3.8}$$

The problem is to minimize the total costs across all edges in the plant subject to flow constraints. Constraints (3.3) to (3.7) are traditional binary MCFP, where edges flow from sources to sinks in the layer sub-networks. Constraint set (3.2) states that RP must be over a certain RP value, which implies it is an upper limit on peak NO_x emissions.

Constraint (3.8) controls average daily NO_x, where $f(x)$ can only be obtained once we have the schedule.

Constraint (3.8) is needed for the MCFPwSEC, and the reason is as follows. The only way to obtain the average daily NO_x reduction using relaxed MCFP is with the schedule. However, in order to obtain the schedule, we first need to optimize and obtain the solution. Then, with the solution and the schedule, we determine if it violates the average daily minimum NO_x constraint (3.8). If it does, we make that solution infeasible and re-optimize the problem. If it does not, then it is an optimal solution.

The mathematical formulation is as follows. Define F as the set of all feasible flows in MCFP that relax the average daily NO_x constraint (3.8). Define $S = \{x \in F \mid f(x) \geq d\}$ as the set of all feasible schedules. Define $S^c = F \backslash S$ as the set of flows in MCFP that are infeasible schedules. For example, schedule $s \in S^c$ may violate average daily NO_x constraint. Thus, the set of schedule elimination constraints is given by equation (3.8').

$$\sum_{s \epsilon S} x_s \leq |s| - 1 \qquad \forall s \in S^c \qquad (3.8')$$

Since S^c can be large, it is generated dynamically through MCFPwSEC. Then, a cut is added after a schedule is found that violates the minimum average daily NO_x constraint. Cuts are added one by one until the minimum average daily NO_x constraint is met, and then, the optimal solution is obtained.

Using multi-cut MCFPwSEC it is possible to eliminate multiple schedules in a given time. Multi-cut MCFPwSEC makes use of the following two facts: (a) cleaned layers, are the cheapest layer but have the least reactor potential; (b) new layers, are the most expensive layer but have the greatest reactor potential. Multi-cut targeting new layers checks the feasibility of replacing every layer with a new layer in a schedule that does not satisfy NO_x emissions, and if still that schedule does not satisfy the NO_x emissions, it eliminates all combination of layers for that schedule. The reason being that no other combination of layers in that schedule will reduce NO_x emissions more than the schedule with only new layers.

Phananiramai [10] presents a computational experiment solving MCFPwSEC for each power plant of a six fleet example sequentially with a target of outlet NO_x. It is important to note that in the computational experiment the assumptions of Phananiramai [10] are

1) Uses a fixed NO_x policy.
2) Ignores the effects of average NH_3 slip on cost.
3) C_{ija}^{l} is independent of average NH_3 slip.
4) Uses maximum NH_3 slip to calculate C_{ija}^{l} making the cost an upper bound.

3.2 Heuristic MCFPwSEC with fixed NO_x policy (HMCFPwSEC).

After having presented the summary of MCFPwSEC introduced by Phananiramai [10], we will explain our utilization of NH_3 slip in MCFPwSEC using the fixed NO_x policy. To do that, we create a heuristic multi-commodity flow problem with scheduled elimination constraints (HMCFPwSEC). In our HMCFPwSEC, we relax assumptions 2 and 5 of Phananiramai [10]. For assumption 2, HMCFPwSEC does not require that NH_3 slip be kept constant, so NO_x reduction is a function of reactor potential and NH_3 slip. For assumption 5, HMCFPwSEC includes operational costs based upon usage of NH_3 injection and NO_x emissions. Since we relaxed assumptions 2 and 5, our HMCFPwSEC is a nonlinear problem that periodically updates the average NH_3 slip of the edges and associated cost. HMCFPwSEC is a heuristic model that does not guarantee optimality, but, as we show later, it considers multiple near optimal solutions.

As mentioned before, the fixed NO_x policy maintains a constant level of average daily NO_x reduction and permits the NH_3 slip level changes over the time horizon. We have two different measures of NH_3 slip level. One measure is called *maximum NH_3 slip* level, also known as *worst NH_3 slip* level because it is the highest NH_3 slip level in the time horizon. The other measure is called *average NH_3 slip* level, and, as the name indicates, it is the average NH_3 slip level during the time horizon.

The major part of the cost of the edge depends on the action taken, but some part of the cost depends upon the level of the average daily NO_x reduction selected and on the average NH_3 slip level. The average NH_3 slip level has a relationship with the average daily NO_x reduction and with the schedule, in a way that changing any of these affects the others. We cannot explain in detail this relationship because it is a proprietary formula, but in general, we can state that if we maintain a constant daily NO_x reduction, and the average NH_3 slip level increases, the cost of the schedule increases; and if the average NH_3 slip level decreases, the cost of the schedule decreases. If we maintain a constant average NH_3 slip level and the average daily NO_x reduction level increases, the cost of the schedule increases. If the average daily NO_x reduction level decreases, the cost of the schedule decreases.

HMCFPwSEC first calculates the cost of the edges and later the cost of the schedule. In order to calculate the cost of the edges, we need to know the average daily NO_x reduction we want to obtain and its corresponding average NH_3 slip level. But, as well as with the average daily NO_x reduction, we only know the average NH_3 slip level after we have the schedule. We call *revised average NH_3 slip* the average NH_3 slip level obtained after we have the schedule. To solve this problem, we calculate the cost of the edges estimating the average NH_3 slip level, and after we obtain the schedule, we re-calculate the cost of the edges using the revised average NH_3 slip of the schedule. We call *revised cost of a schedule* the cost of a schedule calculated using the revised average NH_3 slip. Note that we only estimate the average NH_3 slip level to calculate the cost of the edges, but the revised average NH_3 slip level still can be different from the estimated level in order to satisfy the fixed NO_x policy. In fact, we need to re-calculate the cost of the edges using the revised average NH_3 slip level of the schedule, because we know that the original value and the revised value may be different.

Utilizing the same proprietary formula used by Phananiramai [10], we know that normally the revised average NH_3 slip level is less than half of the worst NH_3 slip level. This fact is important because we have the advantage that we select the permissible worst NH_3 slip level. In order to select the worst NH_3 slip level, we can use the maximum permissible level by law or, if we want a more restricted level, we can use other technical considerations. Thus, we calculate the cost of the edges using the average NH_3 slip level as half the worst NH_3 slip level selected and, after we have the schedule, we re-calculate the cost of the schedule using the revised average NH_3 slip level of this particular schedule.

HMCFPwSEC only calculates the revised cost of the schedules that satisfy constraints (3.2) to (3.7), the average daily NO_x reduction, and the worst NH_3 slip level (3.8). The reason to do this is that the schedules that do not satisfy the average daily NO_x reduction as well as the worst NH_3 slip level (3.8) are considered infeasible, and we eliminate those schedules.

Thus, we solve a relaxed MCFP problem (3.1) to (3.7). If the schedule representing the solution to the relaxed MCFP problem does not satisfy the average daily NO_x reduction as well as the worst NH_3 slip level (3.8), we add a cut, as in (3.8’), for this schedule and resolve the relaxed MCFP problem. When we have a schedule that satisfies (3.2) to (3.7), and the average daily NO_x reduction as well as the worst NH_3 slip (3.8), we save the information of this schedule. Then, we re-calculate the cost of the edges using the revised average NH_3 slip level of this schedule, and we put the revised cost of this schedule in the information of this schedule. Since this is the first schedule that satisfies (3.2) to (3.7) and the average daily NO_x reduction as well as the worst NH_3 slip (3.8), it becomes the *least cost schedule*. We will denominate least cost schedule each schedule that satisfies (3.2) to (3.7), the average daily NO_x reduction as well as the worst NH_3 slip (3.8), and has a real cost smaller than the previous least cost schedule.

There may exist a schedule that, calculating the cost of the edges using the same revised average NH_3 slip level of the least cost schedule, has the same cost as the least cost schedule. However, this schedule may have a revised average NH_3 slip level smaller than the revised average NH_3 slip level of the least cost schedule. As mentioned before, if we use less average NH_3 slip level to calculate the cost of the edges, the cost of the schedule is a little bit less. Thus, we want to explore if there is a schedule that, using the same revised average NH_3 slip level of the least cost schedule, has the same cost as the least cost schedule, but with less revised average NH_3 slip level. In order to do that, we update the cost of edges using the same revised average NH_3 slip level of the least cost schedule. Then, if we found that schedule, we re-calculate the cost of the edges using the revised average NH_3 slip level of this new schedule in order to obtain the revised cost of this new schedule. The revised cost of this new schedule will be less than the revised cost of the first least cost schedule, because the revised average NH_3 slip of this new schedule is lower than the revised average NH_3 slip of the least cost schedule.

The HMCFPwSEC is shown in Algorithm 1 and Figure 3.1. We use the same notation as in section 3.1, but adding the following

Additional parameter

Let wNH_3 be the worst NH_3 slip.

Let τ be a pre-defined percentage, which is multiplied by the cost of the current least cost schedule.

Additional variables

Let $\overline{NH_3}$ be the average NH_3 slip.

Let $\overline{rNH_3}$ be the revised average NH_3 slip.

Then, consider $C^l_{ija}(\overline{NH_3})$ in which $\overline{NH_3}(x)$. With this consideration, the average NH_3 slip depends on the entire schedule, then, the cost also depends on the entire schedule and $C^l_{ija}(x)$ is nonlinear.

We change (3.1) to be

$$min \sum_{l \epsilon L} \sum_{a \epsilon A} \sum_{(i,j) \epsilon E} C^l_{ija}(x) x^l_{ija} \tag{3.9}$$

Then, with HMCFPwSEC we periodically update $\overline{NH_3}$ and $C^l_{ija}(\overline{NH_3})$. As we are using a heuristic, and we are testing different near optimal solutions, we introduce τ as the stopping criteria to the searching of an optimal schedule. The parameter τ is a pre-defined percentage, which is multiplied by the cost of the current least cost schedule. Then, if this cost is less than the cost of a schedule, we stop the searching for schedules.

Algorithm 1 HMCFPwSEC process.

Let $\bar{X}$ and $\bar{x}$ be the set of schedules and the current schedule, respectively, which satisfy (3.2) to (3.7).

Let $X*$ and $x*$ be the set of schedules and the current schedule, respectively, which satisfy (3.2) to (3.7) as well as $f(x)$ and wNH_3 (3.8).

Let start with C ($x*$) and $C\ \bar{x}$ = \$100,000,000.

Set $C^l_{ija} \leftarrow C^l_{ija}(wNH_3/2)$

while $C\bar{x}$ < $(1+\tau)C(x^*)$ **do**

Find $\bar{x} \in \bar{X}$ using MCFPwSEC.

If $\bar{x} \in X*$ **then**

Calculate $\overline{rNH_3}(\bar{x})$

Calculate $C(\bar{x})$ with $\overline{rNH_3}(\bar{x})$

If $C(\bar{x})$ < $C(x^*)$ **then**

$x^* \leftarrow \bar{x}$

$C^l_{ija} \leftarrow C^l_{ija}(\overline{NH_3}(\bar{x}))$

end if

end if

Add cut (3.8') for $\bar{x}$

end while

Return $x*$ as the solution.

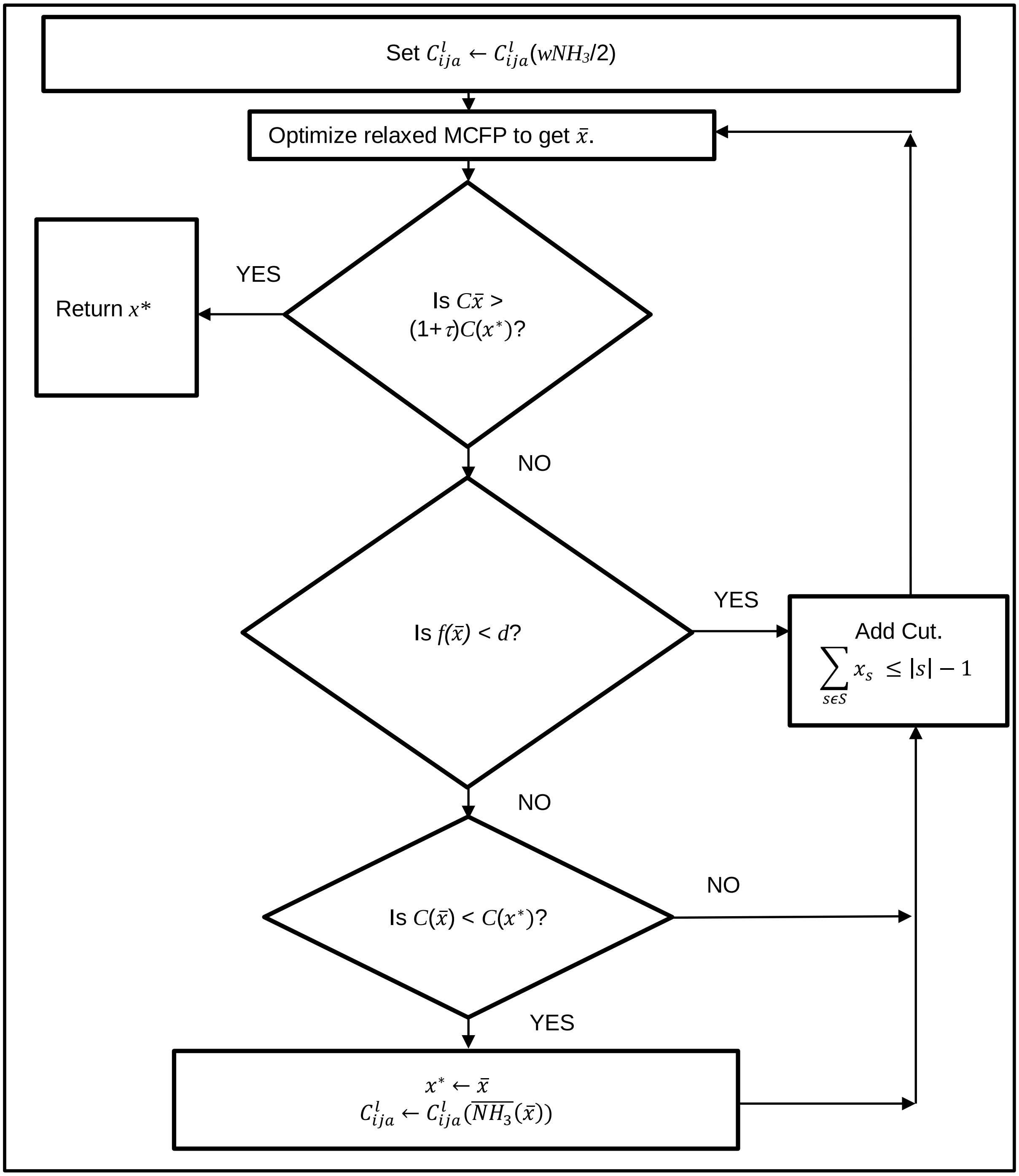

Figure 3.1 HMCFPwSEC algorithm.

HMCFPwSEC can be summarized as follows. First, to calculate the cost of the edges we estimate the average NH_3 slip level as half the worst NH_3 slip level. After that, we solve a relaxed MCFP problem (3.1) to (3.7). If the schedule representing the solution to the relaxed MCFP problem does not satisfy the average daily NO_x reduction as well as the worst NH_3 slip level (3.8), we add a cut, as in (3.8'), for this schedule and resolve the relaxed MCFP problem. If we have a schedule that satisfies the average daily NO_x reduction as well as the worst NH_3 slip (3.8), we save the information of this schedule. Additionally, we re-calculate the cost of the edges using the revised average NH_3 slip level of this schedule in order to save the revised cost of this schedule. We named this first saved schedule as least cost schedule. Then, we update the cost of the edges calculated with the revised average NH_3 slip of the least cost schedule and we add a cut. After that, we solve again a relaxed MCFP problem (3.1) to (3.7). If we found another schedule that satisfies the average daily NO_x reduction as well as the worst NH_3 slip level (3.8), we save the information of this schedule. Then, we re-calculate the cost of the edges using the revised average NH_3 slip level of this schedule in order to put the revised cost of this schedule in the information of this schedule. If the cost of this schedule is less or equal than the previous least cost schedule, then this new schedule becomes the least cost schedule. After that, we update the cost of the edges calculated with the revised average NH_3 slip of the least cost schedule and we add a cut. However, if the cost of this new schedule is greater than the previous least cost schedule, then this new schedule does not become the least cost schedule. After that, we do not update the cost of the edges calculated with the revised average NH_3 slip of this new schedule, but we only add a cut. We will continue with this process until we find a schedule with a greater cost than the least cost schedule. When we find a schedule with greater cost than the least cost schedule we stop the process. Thus, the last least cost schedule becomes the solution.

Now, remember that the major part of the cost of the edge depends on the action taken, but some part of the cost depends on the level of the average daily NO_x reduction selected and on the average NH_3 slip level. HMCFPwSEC with fixed NO_x policy does not change the average daily NO_x reduction we want to obtain, but changes the average NH_3 slip level. A concern with this approach is that, there may exist a schedule that, calculating the cost of the edges with the revised average NH_3 slip level of the least cost schedule, has higher cost than the least cost schedule. However, this schedule may has a revised average NH_3 slip level lower than the least cost schedule. Then, it may be a better solution than our least cost schedule if in this new schedule, the difference in cost by the lower average NH_3 slip level is enough to surpass the increase in cost by the actions and outages selected.

If we want to solve this concern, then, we still can use HMCFPwSEC, but with some changes. Instead of stopping the search of schedules when a schedule has a higher cost than the least cost schedule, we will stop when the cost of the schedule increases τ percentage above the least cost schedule. Since the actions and outages are the major part of the cost and we are very close to the lowest cost solution, then, a good value for τ might be 5% or less.

To illustrate this process, also described in Algorithm 1 and Figure 3.1, consider the following example. Suppose we have a single power plant, and we select 4.00 parts per million (ppm) as the worst NH_3 slip level, and the selected τ is 3%. Then, calculating the cost of edges using half of the worst NH_3 slip level (2.00 ppm) as the average NH_3 slip level, we generate 200 cuts.

Schedule 201 satisfies (3.2) to (3.7) and the average daily NO_x reduction as well as the worst NH_3 slip level (3.8). We obtain that for this schedule the revised average NH_3 slip level is 1.15 ppm. Then, we re-calculate the cost of the edges using 1.15 ppm instead of 2.00 ppm and we obtain the revised cost of schedule 201. We save the information for schedule 201 including the revised cost of schedule 201 (cost using the average NH_3 slip as 1.15 ppm). Now, our least cost schedule is schedule 201. After that, we update the cost of the edges using the average NH_3 slip as 1.15 ppm (because it is the revised average NH_3 slip of the least cost schedule), we cut schedule 201, and we continue searching for a new schedule. In table 3.1 we can see the advance of the example.

Table 3.1 Advance of the example until schedule 201.

Number of Schedule	$C\overline{x}$ **<** **(1+**τ**)**$C(x^*)$**?**	**Is it feasible?**	**Estimated** $\overline{NH_3}$	$\overline{rNH_3}(\overline{x})$	**Is the cost better than previous least cost schedule?**	**Do we update cost of edges?**
1 to 200	Yes	No	2.00			
201	Yes	Yes	2.00	1.15	Yes	Yes

Schedules 202, 203, and 204 have exactly the same cost as schedule 201; but only schedule 204 satisfies (3.2) to (3.7) and the average daily NO_x reduction as well as the worst NH_3 slip level (3.8). We obtain that for schedule 204 the revised average NH_3 slip level is 1.05 ppm. Then, we re-calculate the cost of the edges using 1.05 ppm instead of 1.15 ppm, and we obtain the revised cost of schedule 204. We note that the cost of schedule 204 using average NH_3 slip as 1.05 ppm is smaller than the cost of schedule 201 using the average NH_3 slip as 1.15 ppm. Since in both cases we are using the revised average NH_3 slip corresponding to each schedule, schedule 204 is a better schedule compared to schedule 201. Now, our least cost schedule is schedule 204. After that, we update the cost of the edges using the average NH_3 slip as 1.05 ppm (because it is the revised average NH_3 slip of the least cost schedule), we create a cut for schedule 204, and we continue searching for a new schedule. In table 3.2 we can see the advance of the example.

Table 3.2 Advance of the example until schedule 204.

Number of Schedule	$C\bar{x}$ **< (1+**τ**)**$C(x^*)$**?**	**Is it feasible?**	**Estimated** $\overline{NH_3}$	$\overline{rNH_3}(\bar{x})$	**Is the cost better than previous least cost schedule?**	**Do we update cost of edges?**
1 to 200	Yes	No	2.00			
201	Yes	Yes	2.00	1.15	Yes	Yes
202 & 203	Yes	No	1.15			
204	Yes	Yes	1.15	1.05	Yes	Yes

Schedule 205 is 1.5% more expensive than schedule 204, but neither satisfies the average daily NO_x reduction nor the worst NH_3 slip level. If in our example τ is smaller than 1.5%, we stop here and the least cost schedule is schedule 204. However, because τ is 3%, we continue our searching. Schedule 206 has the same cost of schedule 205, satisfies (3.2) to (3.7), and the average daily NO_x reduction as well as the worst NH_3 slip level (3.8). We obtain that for schedule 206 the revised average NH_3 slip level is 1.02 ppm. Then, we re-calculate the cost of the edges using 1.02 ppm instead of 1.05 ppm, and we obtain the real cost of schedule 206. The cost of schedule 206 using the average NH_3 slip as 1.02 ppm is greater than the cost of schedule 204 using the average NH_3 slip as 1.05 ppm. Since in both cases we are using the revised average NH_3 slip corresponding to each schedule, schedule 204 is a better schedule compared to schedule 206. Then, our least cost schedule is still schedule 204. After that, we create a cut for schedule 206, and we continue searching for a new schedule with the cost of edges using average NH_3 slip as 1.05 ppm. We do not update the cost of the edges because schedule 206 is worse than schedule 204. In table 3.3 we can see the advance of the example.

Table 3.3 Advance of the example until schedule 206.

Number of Schedule	$C\bar{x}$ **<** **(1+**τ**)**$C(x^*)$**?**	**Is it feasible?**	**Estimated** $\overline{NH_3}$	$\overline{rNH_3}(\bar{x})$	**Is the cost better than previous least cost schedule?**	**Do we update cost of edges?**
1 to 200	Yes	No	2.00			
201	Yes	Yes	2.00	1.15	Yes	Yes
202 & 203	Yes	No	1.15			
204	Yes	Yes	1.15	1.05	Yes	Yes
205	Yes	No	1.05			
206	Yes	Yes	1.05	1.02	No	No

Schedule 207 still is 1.5% more expensive than schedule 204, satisfies (3.2) to (3.7), and the average daily NO_x reduction as well as the worst NH_3 slip level (3.8). We obtain that for schedule 207 the revised average NH_3 slip level is 0.65 ppm. Note that the difference in the average NH_3 slip level between schedules 204 and 207 is 0.40 instead of 0.10, which is the difference between schedules 201 and 204, and instead of 0.03, which is the difference between schedules 204 and 206. Then, we calculate the cost of the edges using the average NH_3 slip level as 0.65 ppm instead of 1.05 ppm, and we obtain the revised cost of schedule 207. The cost of schedule 207 using average NH_3 slip as 0.65 ppm is smaller than the cost of schedule 204 using average NH_3 slip as 1.05 ppm. Since in both cases we are using the revised average NH_3 slip corresponding to each schedule, schedule 207 is better schedule than schedule 204. Now, our least cost schedule is schedule 207. After that, we update the cost of the edges using the average NH_3 slip as 0.65 ppm. We create a cut for schedule 207, and we continue searching for a new schedule. In table 3.4 we can see the advance of the example.

Table 3.4 Advance of the example until schedule 207.

Number of Schedule	$C\overline{x}$ < (1+τ)$C(x^*)$?	**Is it feasible?**	**Estimated** $\overline{NH_3}$	$\overline{rNH_3}(\overline{x})$	**Is the cost better than previous least cost schedule?**	**Do we update cost of edges?**
1 to 200	Yes	No	2.00			
201	Yes	Yes	2.00	1.15	Yes	Yes
202 & 203	Yes	No	1.15			
204	Yes	Yes	1.15	1.05	Yes	Yes
205	Yes	No	1.05			
206	Yes	Yes	1.05	1.02	No	No
207	Yes	Yes	1.05	0.65	Yes	Yes

Schedule 208 has a cost 3.5% greater than schedule 207. Since our selected τ is 3%, we stop here, and we obtain that schedule 207 is the least cost schedule. Then, schedule 207 is the solution. In table 3.5 we can see the example until the final solution.

After we presented HMCFPwSEC algorithm and an example to illustrate it, in the next section we will discuss a computational experiment.

Table 3.5 Solution of the example.

Number of Schedule	$C\overline{x}$ **<** **(1+**τ**)**$C(x^{*})$**?**	**Is it feasible?**	**Estimated** $\overline{NH_3}$	$\overline{rNH_3}(\overline{x})$	**Is the cost better than previous least cost schedule?**	**Do we update cost of edges?**
1 to 200	Yes	No	2.00			
201	Yes	Yes	2.00	1.15	Yes	Yes
202 & 203	Yes	No	1.15			
204	Yes	Yes	1.15	1.05	Yes	Yes
205	Yes	No	1.05			
206	Yes	Yes	1.05	1.02	No	No
207	Yes	Yes	1.05	0.65	Yes	Yes
208	No					

3.3 Computational Experiments

In this section, consider a single power plant with five scheduled outages in a time horizon of five years. The plant has one scheduled outage per year (not on the same date each year), and the plant cannot work more than 750 days without adding or changing a layer in an outage. The power plant has two filled slots with a catalyst layer and two empty slots at the start of the time horizon. Layers are indexed 1, 2, 3, and 4 where the layer closest to the inlet is layer 4. We use 4.00 parts per million (ppm) as the worst NH_3 slip level for the plant. We want to know the followings percentage NO_x reduction levels for the plant: 60%, 70%, 79%, 80%, 90%, 92%, 93% and 95%. We solve three experiments for each of the previous percentage NO_x reduction levels: two using HMCFPwSEC and τ as 0%, and 2%, and one using SGO. We use SGO because we want to compare the CPU time and the cost of our model with the schedule generation and optimization (SGO) algorithm used by Phananiramai [10]. The reason to compare models is that SGO is an optimal model, but HMCFPwSEC is a heuristic model that does not guarantee optimality. We conduct the experiments using the C++ programming language with CPLEX version 12.5.1 callable library [55] on a workstation with UNIX and also with Intel(r) Xeon(r) X3450 2.67GHz processor and 16323884 kB of memory.

Table 3.6 Percentage NO_x reduction level, τ, CPU time, number of cuts, and cost with HMCFPwSEC and SGO.

Percentage NO_x reduction level	τ **(%)**	**CPU time (sec)**	**Number of cuts**	**Cost ($)**	**CPU time (sec) SGO**	**Cost ($) SGO**
60	0	1.7	23	11,124,307.95	13.05	11,124,500.00
60	2	2.95	31	11,124,307.95		
70	0	1.92	23	11,856,354.80	13.8	11,856,500.00
70	2	2.69	31	11,856,354.80		
79	0	1.72	23	12,582,603.61	13.82	12,582,800.00
79	2	3.3	33	12,582,603.61		
80	0	1.66	23	12,671,181.84	13.86	12,668,900.00
80	2	3.24	34	12,668,774.50		
90	0	2.88	36	13,746,042.50	13.97	13,746,200.00
90	2	3.49	36	13,746,042.50		
92	0	2.78	36	14,087,709.77	14.03	14,087,900.00
92	2	3.51	36	14,087,709.77		
93	0	11.32	115	14,586,322.32	14.1	14,586,500.00
93	2	17.3	125	14,586,322.32		
95	0	11.62	115	15,110,389.94	14.25	15,110,600.00
95	2	17.34	126	15,110,389.94		

For comparison, we present in table 3.6 the percentage NO_x reduction level, the τ, the CPU time, and the cost for the plant using HMCFPwSEC and SGO. Observe that the two columns on the right are the CPU time and cost of SGO. Since SGO does not use τ, we include the CPU time and cost using the rows of τ at 0%.

In table 3.6, we can see that, the cost using HMCFPwSEC with τ at 0% and at 2% is exactly the same, using the same percentage NO_x reduction level, except for 80% NO_x reduction level. At that level, the use of τ at 2% gives us a better cost than with τ at 0%. On the other hand, comparing the cost at same percentage NO_x reduction level between HMCFPwSEC and SGO, we observe that the only cost that is higher using HMCFPwSEC than using SGO is exactly at 80% NO_x reduction level and τ at 0%. We realize that at any percentage NO_x reduction level with τ at 2%, HMCFPwSEC give us a slightly lower cost than SGO (around $200), but this is due to rounding errors. The same is true for τ at 0%, except at 80% NO_x reduction level. Additionally, note that in table 3.6 the cost always increases if the percentage NO_x reduction level increases.

We observe that the selected schedule is not always the same using different %NO_x reduction levels. We have the same schedule with 60%, 70%, and 79% NO_x reduction levels regardless of using τ at 0%, at 2%, or using SGO. For 80% NO_x reduction level, we still have the same schedule as before with τ at 0%, but not with τ at 2% and SGO. In fact, we observe that for 80% NO_x reduction level with τ at 0%, we have a higher cost than with τ at 2% and SGO. The reason is that SGO and τ at 2% change the selected schedule at 80% NO_x reduction level, but τ at 0% still has the same schedule as with the previous percentage NO_x reduction levels. However, with 90% and 92% NO_x reduction levels, τ at 0%, τ at 2%, and SGO give us exactly the same schedule as τ at 2% and SGO give us with 80% NO_x reduction level. Furthermore, we have a different selected schedule with 93% and 95% NO_x reduction levels. For these percentage NO_x reduction levels we have exactly the same selected schedule for τ at 0%, τ at 2%, and SGO.

We note that, regardless of the selected percentage NO_x reduction level, if we maintain the percentage NO_x reduction level constant, then, τ at 0% always uses less CPU time than τ at 2%. We expected this result because if we increase τ, then we need to check more schedules than with smaller τ. On the other hand, if we maintain the percentage NO_x reduction level constant, we observe that τ at 0% always uses less CPU time than SGO. With percentage NO_x reduction levels smaller than 93%, τ at 2% uses less CPU time than SGO. In fact, we note that between 92% and 93% NO_x reduction levels there is a big difference in CPU time, regardless using τ at 0% or τ at 2%. The explanation of this behavior is as follows, at 92% or less NO_x reduction levels the selected schedules only add two cleaned layers and change one cleaned layer, but at 93% NO_x reduction level the selected schedule add one regenerated layer, one cleaned layer, and change one cleaned layer. The need to use a regenerated layer causes us to check many more possible schedules than on the previous percentage NO_x reduction levels, and that process uses more CPU time. In fact, as shown in table 3.6, the increase is 79 more cuts for τ at 0% and 89 more cuts for τ at 2% at 93% NO_x reduction level than at 92% NO_x reduction level.

For this experiment, we can conclude that the cost increases if the percentage NO_x reduction level increases. In addition, if we maintain constant the percentage NO_x reduction level, τ at 0% always uses the least CPU time. SGO and τ at 2% always give us the same schedule and cost (except for rounding errors), but τ at 0% only miss the correct schedule at 80% NO_x reduction level. The selection of a schedule that uses a regenerated layer increases greatly the CPU time for HMCFPwSEC versus the selection of a schedule that only uses cleaned layers.

Chapter 4

OPTIMIZATION OF A FLEET OF COAL-FIRED POWER PLANTS.

This chapter has been divided into three sections. In section 4.1, we summarize the formulation of the optimization module of SGO of Phananiramai [10]. In section 4.2, we explain the use of HMCFPwSEC to reduce the total cost for a fleet of power plants, maintaining the outlet NO_x for the fleet below a targeted limit. To do that, we create problems with different discrete percentage NO_x reduction levels for each power plant. After solving these problems, we have a schedule with a minimum cost and outlet NO_x, corresponding to each one of the different percentage NO_x reduction levels selected for each power plant of the fleet. Later, we create a problem with the outlet NO_x and costs for each one of the power plants of the fleet, and then, we minimize the total cost of the fleet satisfying a given maximum outlet NO_x for the fleet of power plants.

In section 4.3, we present eight computational experiments using 41 different discrete percentage NO_x reduction levels for each one of the six power plants of the fleet. We use an outlet NO_x emission constraint for the fleet of power plants. We solve each computational experiment using HMCFPwSEC for a fleet of power plants (fleet HMCFPwSEC). We compare them to a computational experiment using the same six power plants, but with a percentage NO_x reduction level constraint pre-defined for each one of the six power plants. The sum of the six outlet NO_x emission for the plants is exactly the same as the outlet NO_x emission for the fleet. Then, we show that, the solution using an outlet NO_x emission constraint for the fleet of power plants is less expensive than the final cost when we have the same outlet NO_x emission for the fleet, but with a percentage NO_x reduction level constraint pre-defined for each power plant alone. After that, we solve again the same computational experiment, but now, we are using the SGO used by Phananiramai [10]. We create a fleet SGO modifying the SGO using the same logic as we use to create our fleet HMCFPwSEC (FHMCFPwSEC). Then, we solve the same computational experiments as we do with our FHMCFPwSEC. We show that there is little difference in cost and savings between both models, but there is an important difference in CPU time and wall clock time.

4.1 Schedule Generation and Optimization algorithm.

As mentioned before, Phananiramai [10] research include a schedule generation and optimization (SGO) algorithm. In this section, we present the algorithm of the optimization module in a summarized way, for an exhaustive explanation of SGO model refers to Phananiramai [10].

The SGO algorithm enumerates all the feasible schedules and then selects the least expensive schedule or the schedule that minimizes NO_x emissions. The selection depends upon whether the SGO tries to minimize cost or to minimize NO_x emissions. The SGO algorithm is divided in two main modules: SCR schedule generation and SCR optimization. The first module enumerates a set of possible outage schedules for all plants in the fleet. The second module uses Computational Infrastructure for Operations Research branch and cut (COIN-OR CBC), a 0-1 large scale integer program, to select the least expensive schedule or the schedule that minimizes NO_x emissions. This SCR optimization module finds a set of schedules that maximizes NO_x emissions reduction subject to a total operating cost and power generation plan. In this formulation each plant is assigned to exactly one schedule in the plan, each outage is included in at most one schedule in the plan, and uses fixed NH_3 slip policy.

List of parameters:

Let S be the set of all schedules from the SCR schedule generation module.

Let P be the set of all plants.

Let O be the set of all outages.

For each schedule $s \in S$

Let $DNOX_{ps}$ be the NO_x reduction of schedule s in plant p,

let c_{ps} be the operating costs,

let g_{ps} be the power generation.

For each outage $o \in O$, let $S(o)$ be the set of schedules that include outage o.

For each plant $p \in P$, let $S(p)$ be the set of schedules that can be assigned to plant p.

Let C be the maximum operating costs of the fleet.

Let G be the minimum power generation.

List of variables

For each schedule $s \in S$

let x_{ps} =1 if schedule s in plant p is selected for the outage, and 0 otherwise.

The integer linear programming problem is given by the following

$$\max \sum_{p \epsilon P} \sum_{s \epsilon S_{(p)}} DNOX_{ps} x_{ps} \qquad (4.1)$$

s.t.

$$\sum_{p \epsilon P} \sum_{s \epsilon S_{(p)}} c_{ps} x_{ps} \leq C \qquad (4.2)$$

$$\sum_{p \epsilon P} \sum_{s \epsilon S_{(p)}} g_{ps} x_{ps} \geq G \qquad (4.3)$$

$$\sum_{s \epsilon S_{(p)}} x_{ps} = 1 \qquad \forall p \epsilon P \qquad (4.4)$$

$$\sum_{s \epsilon S_{(o)}} x_{ps} \leq 1 \qquad \forall p \epsilon P, \quad \forall o \epsilon O \qquad (4.5)$$

$$x_{ps} \in \{0,1\} \qquad \forall p \epsilon P, \quad \forall s \epsilon S_{(p)} \qquad (4.6)$$

The problem is to maximize the average daily NO_x reduction over all power plants subject to a total operating cost and power generation plan. Constraint (4.2) states that the total operating cost of the plan is less than or equal to a predetermined budget. Constraint (4.3) states that the total power production is greater than or equal to a predetermined minimum production. Constraint (4.4) states that each plant is assigned to exactly one schedule in the plan, and constraint (4.5) states that each outage is included in at most one schedule in the plan.

4.2 Use of HMCFPwSEC to reduce total cost of a fleet with maximum outlet NO_x for the fleet.

As mentioned by Phananiramai [10], the selection between using the fixed NH_3 slip policy or the fixed NO_x policy will depend upon the emphasis of whether we would like to control either NH_3 slip or NO_x emissions. Since in our case we want to control NO_x emissions, as Phananiramai [10] did in the computational experiments using the MCFPwSEC, we will use the fixed NO_x policy.

The fixed NO_x policy uses a percentage NO_x reduction level, for example, 75% NO_x reduction. Depending on the inlet NO_x, this reduction level means a value of outlet NO_x. For example, if the inlet NO_x is 100 pounds per hour (lb/hr) and the reduction NO_x level is 75%, then we reduce 75 lb/hr, and our outlet NO_x is 25 lb/hr. We can select different percentage NO_x reduction level, and then the solution will be a schedule with different cost and different outlet NO_x. Then, HMCFPwSEC with fixed NO_x policy uses a percentage NO_x reduction level, and the solution gives us a schedule, with the minimum cost and the corresponding outlet NO_x, to the level selected using that particular plant.

We can use the fixed NO_x policy with one power plant, but for a fleet of power plants we need more research. The reason is that each power plant has its own outage plan and its own SCR. Following that reasoning, we cannot optimize a fleet of power plants using the fixed NO_x policy for all the fleet. However, we can use different percentage NO_x reduction levels for each power plant, and the solution is a schedule, with a minimum cost and outlet NO_x, corresponding to each of the percentage NO_x reduction levels selected for each plant. Then, we can create a problem with different costs and outlet NO_x for each one of the power plants of the fleet. If we put together the outlet NO_x and costs for each one of the plants of the entire fleet, then we can minimize the total cost of the fleet satisfying a given maximum outlet NO_x for the fleet of power plants.

For example, we assume that we have four power plants A, B, C, and D. For plant A we have an inlet NO_x of 100 lb/hr, for plants B and C we have an inlet NO_x of 120 lb/hr, and for plant D we have an inlet NO_x of 140 lb/hr. If in the current plan for plants A and B obtain 75% NO_x reduction, and for power plants C and D we were to obtain 80% NO_x reduction, then, the outlet NO_x obtained for plant A is 25 lb/hr, for plant B is 30 lb/hr, for plant C is 24 lb/hr, and for plant D is 28 lb/hr. The total outlet NO_x is 107 lb/hr. But we realize that in order to obtain 107 lb/hr there are many possible solutions. For example, another possible solution is that plant A obtains 72% NO_x reduction (outlet NO_x is 28 lb/hr), plant B obtains 74% NO_x reduction (outlet NO_x is 31.2 lb/hr), plant C obtains 80% NO_x reduction (outlet NO_x is 24 lb/hr), and plant D obtains 83% NO_x reduction (outlet NO_x is 23.8 lb/hr). The solution that minimizes cost for the fleet depends on the cost of each one of the four plants, given that the four power plants need to satisfy the outlet NO_x requirement and each plant needs to be used one time.

This type of problem is known in the literature as the knapsack problem. The knapsack problem takes its name from the problem of a hiker, who needs to decide what to put in a knapsack given a weight limitation on how much he can carry [56]. In our case, we need to decide how much cost to accommodate in each power plant, given how much outlet NO_x we can emit to the atmosphere. The requirement that each plant needs to be used one time is an assignment constraint [56].

Therefore, in order to minimize the cost of an entire fleet of power plants, we need to do the following. We need to decide the quantity of percentage NO_x reduction levels we will find for each plant. Then, we solve the first percentage NO_x reduction level using HMCFPwSEC for each plant, and we obtain the schedule with minimum cost for each plant for this first percentage NO_x reduction level. After that, we update the information for the knapsack problem (optimal schedule with its cost and outlet NO_x emissions for each plant), but we do not solve the knapsack problem at this moment. Then, we solve the second percentage NO_x reduction level using HMCFPwSEC for each plant, and we obtain the schedule with minimum cost for each plant for this second percentage NO_x reduction level. Again, we update the information for the knapsack problem (optimal schedule with its cost and outlet NO_x emissions for each plant), but we do not solve the knapsack problem at this moment. We continue with this process until we finish with the entire quantity of percentage NO_x reduction levels we want to search. When the search is complete, we have the complete knapsack problem. The knapsack problem has different optimal schedules with their costs and outlet NO_x emissions for all the plants in the fleet. Later, we solve the knapsack problem minimizing the cost for all the plants with an outlet NO_x emissions less or equal to the value selected for the fleet, and with each plant used one time (assignment constraint). In the end, we obtain the solution of the knapsack and assignment problem.

The overview of the algorithm used for that problem is showed in Figure 4.1. Observe that in Figure 4.1 we can use HMCFPwSEC or SGO to solve a discrete NO_x reduction level.

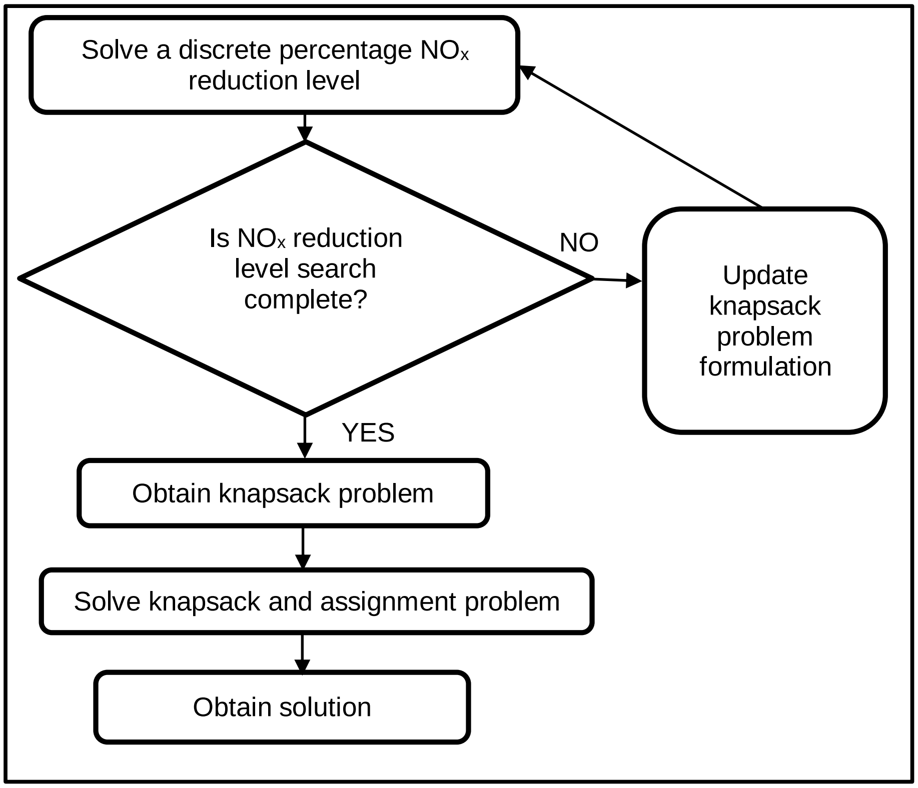

Figure 4.1 Overview of knapsack and assignment algorithm for the entire fleet.

The formulation is as follows.

Let $c_{s(p,\ \delta)}$ be the cost for the schedule with minimum cost for plant p with outlet NO_x level δ.

Let $ot_{s(p,\delta)}$ be the outlet NO_x emissions for the schedule with minimum cost for plant p with outlet NO_x level δ.

Let mot be the maximum outlet NO_x emissions for the fleet. It is a limit level to satisfy regulations.

Let $x_{s(p,\delta)}$ be 1 if the schedule with minimum cost for plant p with outlet NO_x level δ is selected, and 0 otherwise.

$$\min \sum_{p \epsilon P} \sum_{\delta \epsilon D_{(p)}} c_{s(p,\delta)} x_{s(p,\delta)} \tag{4.7}$$

s.t.

$$\sum_{p \epsilon P} \sum_{\delta \epsilon D_{(p)}} ot_{s(p,\delta)} x_{s(p,\delta)} \leq mot \tag{4.8}$$

$$\sum_{\delta \epsilon D_{(p)}} x_{s(p,\delta)} = 1 \qquad \forall p \,\epsilon\, P \tag{4.9}$$

$$x_{s(p,\delta)} \in \{0,1\} \qquad \forall p \,\epsilon\, P, \quad \delta \,\epsilon\, D_{(p)} \tag{4.10}$$

The problem is to minimize the total cost across the fleet of power plants, with an outlet NO_x emissions less or equal to the maximum outlet NO_x emissions for the fleet, and with each power plant used one time.

As mentioned previously, observe that it is possible to apply this formulation to the schedule generation and optimization (SGO) algorithm. For SGO, we follow the same procedure outlined in figure 4.1 by using SGO to solve a discrete NO_x reduction level. In the end, we optimize selecting the schedules that minimize the cost for the fleet and also satisfy the maximum outlet NO_x constraint for the fleet, given that each plant needs to be used once. Note that for SGO we do not need to find the optimal schedule immediately after solving each percentage NO_x reduction level. The reason being that we have the cost and outlet NO_x for many schedules that satisfy each percentage NO_x reduction level. Remember that the outlet NO_x is fixed for each percentage NO_x reduction level. Then, for each plant the optimization in the end selects the schedule with cost and outlet NO_x that minimizes the total cost of the fleet while satisfying the maximum outlet NO_x constraint, using each plant once.

As we mentioned in section 3.1, Phananiramai [10] used a multi-cut method targeting new layer in order to save time. We can follow the same procedure outlined in figure 4.1 for HMCFPwSEC multi-cut targeting new layer. As explained in section 3.1, multi-cut targeting new layer checks the feasibility of replacing every layer with a new layer in a schedule that does not satisfy NO_x emissions, and, if still that schedule does not satisfy the NO_x emissions, it eliminates all combination of layers for that schedule. The reason being that no other combination of layers in that schedule will reduce NO_x emissions more than the schedule with only new layers. On the other hand, we propose the use of a multi-cut method targeting regenerated layer. Multi-cut targeting regenerated layer checks the feasibility of replacing every layer with a regenerated layer in a schedule that does not satisfy NO_x emissions, and if still that schedule does not satisfy the NO_x emissions, we only eliminate the possible combinations of layers that include cleaned or regenerated layers for that schedule. Targeting regenerated layer does not eliminate schedules with new layer. The reason being that using new layer is more expensive than cleaned or regenerated layer, then, many schedules using new layer probably never are checked before we have the solution.

After the knapsack and assignment algorithm has been formulated, in the next section we will show computational experiments.

4.3 Computational Experiments

In this section, we present two different fleet of six power plants. For each fleet, we compare the results using HMCFPwSEC, fleet HMCFPwSEC (FHMCFPwSEC), SGO, and fleet SGO (FSGO). We do that because the HMCFPwSEC is a heuristic model, but SGO is an optimal model. We conduct the experiments using the C++ programming language with CPLEX version 12.5.1 callable library [55] on a workstation with UNIX and also with Intel(r) Xeon(r) X3450 2.67GHz processor and 16323884 kB of memory.

4.3.1 Experiments A, B, C, and D.

Consider a fleet of six power plants. The original NO_x reduction level for plants 1, 2, 4, and 5 is 75%, and for plants 3 and 6, 80%. For plants 1, 2, 4, and 5 we cannot have more than 720 days without adding or changing a layer in an outage, and for plants 3 and 6 that time is 1080 days. For all the plants, at the start of the time horizon, we have 2 filled slots with a catalyst layer and 2 empty slots. The inlet NO_x for plants 1, 2, 4, and 5 is 132 lb/hr, and for plants 3 and 6 it is 145 lb/hr. We solve each one of the six plants using HMCFPwSEC with τ at 0%, and we obtain the schedules that give us the minimum cost with the selected percentage NO_x reduction level for each one of the six plants. We add the cost of these schedules, and we found that the total cost for the six plants is $70,700,113.70. The addition of the outlet NO_x for the six plants is 190 lb/hr. With this result, we know that for the fleet formulation the maximum outlet NO_x for the six plants is 190 lb/hr, and that we want to reduce the cost of $70,700,113.69 for the six power plants.

After using HMCFPwSEC, we solve the same problem with SGO, and we obtain the schedules that give us the minimum cost with the selected percentage NO_x reduction level for each one of the six plants. We add the cost of these schedules, and we found that the total cost for the six plants is $70,701,560.00. The addition of the outlet NO_x for the six plants is again 190 lb/hr. Note that this cost is $1,446.30 more than the cost using HMCFPwSEC. The difference in cost is attributed to rounding errors.

Then, we created an experiment for the fleet of plants, named experiment A, using 41 different discrete percentage NO_x reduction levels for each one of the power plants. The percentage NO_x reduction levels for each one of the six plants are 55%, 56%, 57% and so on until 95%. We solve the fleet of power plants using FHMCFPwSEC with τ at 0%. For the fleet of plants, the selected percentage NO_x reduction level for each plant is presented in table 4.1, and the results are presented in table 4.2.

In table 4.1, note that each plant has a different selected percentage NO_x reduction level versus the original solution. In table 4.2, observe that experiment A gives us savings of more than 46 thousand dollars versus the original solution, and the outlet NO_x is maintained below the 190 lb/hr of the original solution. Furthermore, the time to solve the knapsack and assignment problem is less than 0.4% because the more time is used to solve the 41 different discrete percentage NO_x reduction levels.

Table 4.1 Selected percentage NO_x reduction level for each plant of experiment A.

Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
71	73	85	71	73	86

Table 4.2 Results of experiment A.

Cost ($)	Outlet NO_x (lb/hr)	Savings ($)	CPU time (sec)	Wall clock (sec)	Percentage time to solve knapsack + assignment	Percentage time to solve 41 percentage NO_x levels
70,654,000.12	189.89	46,113.58	5,966.01	2,363.84	0.006369	99.993631

After we obtained the solution for the first experiment, we want to search in more detail around the selected percentage NO_x reduction level of each plant, in order to know if it is possible to minimize the cost further. Then, we created three more experiments named experiments B, C, and D. For each experiment, we use 41 different discrete percentage NO_x reduction levels for each one of the power plants. We solve each experiment using FHMCFPwSEC with τ at 0%. Each experiment will use the following logic: for an experiment we will search around the selected percentage NO_x reduction level for each power plant in the previous experiment. Observe that in each experiment the different discrete percentage NO_x reduction levels are not always the same for each power plant. For example, for experiment B the percentage NO_x reduction levels for plants 1, 2, 4, and 5 are 65%, 65.5%, 66%, and so on until 85%; but for plants 3 and 6, the percentage NO_x reduction levels are 70%, 70.5%, 71%, and so on until 90%. The discrete percentage NO_x reduction levels are presented in table 4.3, where $Dl_{(p)}$ indicates the number of power plant p that uses that percentage NO_x reduction level.

Table 4.3 Discrete percentage NO_x reduction levels for each plant of experiments A, B, C, and D using FHMCFPwSEC.

A: $Dl_{(1,2,3,4,5,6)}$ = {55, 56, 57, …, 95}
B: $Dl_{(1,2,4,5)}$ = {65, 65.5, 66, …, 85}
B: $Dl_{(3,6)}$ = {70, 70.5, 71, …, 90}
C: $Dl_{(1,4)}$ = {70, 70.05, 70.1, …, 72}
C: $Dl_{(2,5)}$ = {72, 72.05, 72.1, …, 74}
C: $Dl_{(3,6)}$ = {84, 84.05, 84.1, …, 86}
D: $Dl_{(1,4)}$ = {71.4, 71.41, 71.42, …, 71.8}
D: $Dl_{(2,5)}$ = {72.9, 72.91, 72.92, …, 73.3}
D: $Dl_{(3,6)}$ = {84.62, 84.63, 84.64, …, 85.02}

On the other hand, we also want to solve experiments A, B, C, and D with SGO. Using SGO we can solve a fleet of power plants, but with only one pre-defined outlet NO_x

emission for each plant. In order to work with different NO_x emissions for each plant, we create the SGO for a fleet of power plants (fleet SGO). To do that, we use the same formulation of the knapsack and assignment algorithm we presented in Figure 4.1 for a fleet of power plants. Recall that, solving the fleet of six power plants with SGO and NO_x reduction level for plants 1, 2, 4, and 5 at 75%, and for plants 3 and 6, 80%, we do not obtain the same cost as with HMCFPwSEC. Then, because the cost is not exactly the same, we will compare the savings of the experiments A, B, C, and D using FSGO versus the cost using SGO.

The range and selected percentage NO_x reduction level for each experiment using FHMCFPwSEC are presented in table 4.4. Observe that in the upper part is the range, and in the lower part is the selected percentage NO_x reduction level. On the other hand, the range and selected percentage NO_x reduction level for each experiment using FSGO are presented in table 4.5.

Table 4.4 Range and selected percentage NO_x reduction level for each plant of experiments A, B, C, and D using FHMCFPwSEC.

	Percentage NO_x reduction level. Range, selected level					
	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
A	55-95, 71	55-95, 73	55-95, 85	55-95, 71	55-95, 73	55-95, 86
B	65-85, 71	65-85, 73	70-90, 85	65-85, 71.5	65-85, 73	70-90, 85.5
C	70-72, 71.6	72-74, 73.1	84-86, 84.85	70-72, 71.6	72-74, 73.1	84-86, 84.8
D	71.4-71.8, 71.55	72.9-73.3, 73.04	84.62-85.02, 84.92	71.4-71.8, 71.55	72.9-73.3, 73.04	84.62-85.02, 84.93

Table 4.5 Range and selected percentage NO_x reduction level for each plant for experiments A, B, C, and D using FSGO

	Percentage NO_x reduction level. Range, selected level					
	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
A	55-95, 71	55-95, 73	55-95, 85	55-95, 71	55-95, 73	55-95, 86
B	65-85, 71	65-85, 73	70-90, 85	65-85, 71	65-85, 73.5	70-90, 85.5
C	70-72, 71.55	72.3-74.3, 73.05	84.3-86.3, 84.9	70-72, 71.55	72.3-74.3, 73.25	84.3-86.3, 84.75
D	71.35-71.75, 71.4	72.95-73.35, 73.11	84.65-85.05, 84.87	71.35-71.75, 71.67	72.95-73.35, 73.11	84.65-85.05, 84.88

The cost, outlet NO_x and savings of each one of the experiments using FHMCFPwSEC and FSGO are presented in table 4.6.

Table 4.6 Cost, outlet NO_x and savings of experiments A, B, C, and D using FHMCFPwSEC and FSGO.

	Cost ($)	Outlet NO_x (lb/hr)	Savings ($)	Cost ($) FSGO	Outlet NO_x (lb/hr) FSGO	Savings ($) FSGO
A	70,654,000.12	189.89	46,113.58	70,655,600.00	189.89	45,960.00
B	70,649,248.58	189.955	50,865.11	70,650,930.00	189.955	50,630.00
C	70,646,309.96	189.9995	53,803.73	70,647,820.00	189.9995	53,740.00
D	70,646,257.36	189.9999	53,856.33	70,647,760.00	189.9999	53,800.00

For experiment A we obtain exactly the same selected level on tables 4.4 and 4.5. For experiment B we have a small difference on plants 4 and 5; in fact, the difference is that in table 4.5 for plant 4 we obtained 0.5 percentage NO_x reduction level less than in table 4.4, but for plant 5 we obtained 0.5 percentage NO_x reduction level greater than in table 4.4. For experiment C observe that the six plants each have a different selected level, but the levels between both tables are very similar; in fact in table 4.5 for plants 3 and 5 the selected level is greater than in table 4.4, but for plants 1, 2, 4, and 6 the selected level is less than in table 4.4. For experiment D note that again the selected levels are different but very similar. In table 4.5 for plants 2, 4, and 5 the selected level is greater than in table 4.4, and for plants 1, 3, and 6 the selected level is less than in table 4.4. Observe that, the percentage NO_x reduction levels using FHMCFPwSEC and FSGO, are different compared to the pre-defined levels we have in the experiment solved using HMCFPwSEC and SGO.

Since in the original solution we observe a small difference between HMCFPwSEC and SGO by rounding errors, we also think that some rounding errors slightly change the selected levels for experiments B, C, and D. One possible explanation for obtaining exactly the same solution for experiment A is because for experiment A the difference between the different discrete 41 percentage NO_x reduction levels (55%, 56%, 57%, …, 95%) is double or more than in the other experiments. For example, in plants, 1, 2, 4, and 5 of experiment B we have 65%, 65.5%, 66%, …, 85%. Then, because the experiments B, C, and D have closer different discrete percentage NO_x reduction levels, the rounding errors slightly change the final selected percentage NO_x reduction levels.

In table 4.6 observe that neither the cost nor the savings are the same for each experiment using FHMCFPwSEC and FSGO, but both results are very close. In fact, the greatest difference is obtained in experiment B because in it the cost for FSGO is

$1,700.00 greater compared to FHMCFPwSEC and the savings are nearly $200.00 smaller compared to FHMCFPwSEC. Observe that the savings are higher on experiment D than on the other experiments. In addition, note that the difference between the savings on experiments C and D is very small. On the other hand, observe that the outlet NO_x is exactly the same regardless of using FHMCFPwSEC or FSGO.

In order to save time, we observe that experiment A gives us enough information to later use experiment C or experiment D without the use of experiment B. Experiments C and D increase the savings by nearly seven thousand dollars versus experiment A, but the difference between experiments C and D is only around 60 dollars. Then, we can omit experiments B and C or B and D, and we still obtain a solution that gives us a significant increase in savings using half of the time compared with using all the four experiments.

We use FHMCFPwSEC multi-cut targeting new layer, with τ at 0%, and multi-cut FHMCFPwSEC targeting regenerated layer, with τ at 0%, with experiments A, B, C, and D, and we obtain exactly the same results as with FHMCFPwSEC (single cut); the only change is with CPU time and wall clock. In table 4.7 we show the CPU time used by FHMCFPwSEC single cut (τ at 0%), FHMCFPwSEC multi-cut targeting new layer (τ at 0%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 0%), and FSGO for experiments A, B, C, and D. In table 4.8 we show the wall clock used by FHMCFPwSEC single cut (τ at 0%), FHMCFPwSEC multi-cut targeting new layer (τ at 0%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 0%), and FSGO for experiments A, B, C, and D.

In table 4.7, it is clear that FHMCFPwSEC multi-cut targeting new layer uses the least CPU time. Furthermore, targeting new layer uses around 5 times less CPU time than single cut, 1.8 times less CPU time than targeting regenerated layer, and 1.4 times less CPU time than FSGO

Table 4.7 CPU time used by FHMCFPwSEC single cut (τ at 0%), FHMCFPwSEC multi-cut targeting new layer (τ at 0%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 0%), and FSGO for experiments A, B, C, and D.

	CPU time (sec) FHMCFPwSEC single cut	CPU time (sec) FHMCFPwSEC targeting new layer	CPU time (sec) FHMCFPwSEC targeting regenerated layer	CPU time (sec) FSGO
A	5,966.01	1,262.80	2,314.04	1,746.02
B	6,169.82	1,249.55	2,286.39	1,764.66
C	6,342.82	1,247.77	2,333.17	1,759.21
D	6,766.13	1,260.87	2,364.42	1,764.08

Table 4.8 Wall clock used by FHMCFPwSEC single cut (τ at 0%), FHMCFPwSEC multi-cut targeting new layer (τ at 0%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 0%), and FSGO for experiments A, B, C, and D. .

	Wall clock (sec) FHMCFPwSEC single cut	Wall clock (sec) FHMCFPwSEC targeting new layer	Wall clock (sec) FHMCFPwSEC targeting regenerated layer	Wall clock (sec) FSGO
A	2,363.84	927.72	1,508.44	1,753.53
B	2,427.24	933.00	1,511.20	1,769.89
C	2,455.05	936.00	1,515.02	1,764.82
D	2,533.75	929.59	1,525.23	1,766.33

In table 4.8, FHMCFPwSEC multi-cut targeting new layer uses the least wall clock time. In this case, targeting new layer uses around 2.6 times less wall clock time than the single cut, 1.6 times less wall clock time than targeting regenerated layer, and 1.9 times less wall clock time than FSGO. Note that wall clock time using any of the two multi-cut methods is smaller than the wall clock time using FSGO.

We also test τ at 2% using FHMCFPwSEC single cut and multi-cut methods in experiments A, B, C, and D to know if there is any change versus the solution we obtained before. We realize that the results do not change with τ at 2% compared with τ at 0%, but the CPU time and wall clock increases. Then, to save time it is recommended to use τ at 0%.

Table 4.9 CPU time used by FHMCFPwSEC single cut (τ at 2%), FHMCFPwSEC multi-cut targeting new layer (τ at 2%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 2%), and FSGO for experiments A, B, C, and D.

	CPU time (sec) FHMCFPwSEC single cut	CPU time (sec) FHMCFPwSEC targeting new layer	CPU time (sec) FHMCFPwSEC targeting regenerated layer	CPU time (sec) FSGO
A	6,816.90	1,654.31	3,347.37	1,746.02
B	7,006.48	1,582.03	3,235.62	1,764.66
C	6,976.14	1,534.98	3,244.47	1,759.21
D	7,462.87	1,551.70	3,417.59	1,764.08

In table 4.9, we show the CPU time used by FHMCFPwSEC single cut (τ at 2%), FHMCFPwSEC multi-cut targeting new layer (τ at 2%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 2%), and FSGO for experiments A, B, C, and D. In table 4.10 we show the wall clock time used by FHMCFPwSEC single cut (τ at 2%), FHMCFPwSEC multi-cut targeting new layer (τ at 2%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 2%), and FSGO for experiments A, B, C, and D.

Table 4.10 Wall clock used by FHMCFPwSEC single cut (τ at 2%), FHMCFPwSEC multi-cut targeting new layer (τ at 2%), FHMCFPwSEC multi-cut targeting regenerated layer (τ at 2%), and FSGO for experiments A, B, C, and D.

	Wall clock (sec) FHMCFPwSEC single cut	Wall clock (sec) FHMCFPwSEC targeting new layer	Wall clock (sec) FHMCFPwSEC targeting regenerated layer	Wall clock (sec) FSGO
A	2,731.72	1,161.97	2,028.42	1,753.53
B	2,769.13	1,140.41	2,009.09	1,769.89
C	2,747.38	1,120.88	1,968.00	1,764.82
D	2,848.00	1,103.18	2,014.60	1,766.33

In table 4.9, it is clear that FHMCFPwSEC multi-cut targeting new layer uses the least CPU time. Furthermore, targeting new layer uses around 4.5 times less CPU time than single cut, 2.1 times less CPU time than targeting regenerated layer, and 1.1 times less CPU time than FSGO. In table 4.10, again FHMCFPwSEC multi-cut targeting new layer uses the least wall clock time. In this case, targeting new layer uses around 2.5 times less wall clock time than the single cut, 1.8 times less wall clock time than targeting regenerated layer, and 1.6 times less wall clock time than FSGO.

4.3.2 Experiments AA, BB, CC, and DD.

We modify the original fleet of six plants in a way that for plants 2 and 5, instead of using 720 days without an outage, we increase that time to 750 days without an outage. With this unique modification the total cost, using the previously pre-defined percentage NO_x reduction levels for each plant alone, with HMCFPwSEC is $65,683,793.57 and using SGO is $65,685,360.00.

In tables 4.11 and 4.12, we present the range and selected percentage NO_x reduction level for each plant of experiments AA, BB, CC, and DD using FHMCFPwSEC and FSGO, respectively. There is no difference in the selected percentage NO_x reduction level using FHMCFPwSEC single cut and both multi-cut methods, and the same is true using τ at 0% and τ at 2%. For experiments AA and BB we have exactly the same selected percentage NO_x reduction level, regardless of using FHMCFPwSEC or FSGO. In addition, for experiments CC and DD the difference in the selected percentage NO_x reduction level is very small between both models. Observe that in this case plants 1 and 4 use a very similar (for experiment AA exactly the same) percentage NO_x reduction level than the pre-defined level (75%), but the other plants use a different percentage NO_x reduction level.

The cost, outlet NO_x and savings of each one of the experiments using FHMCFPwSEC are presented in table 4.13; and the cost, outlet NO_x, and savings of each one of the experiments using FSGO are presented in table 4.14.

Table 4.11 Range and selected percentage NO_x reduction level for each plant of experiments AA, BB, CC, and DD using FHMCFPwSEC.

	Percentage NO_x reduction level. Range, selected level					
	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
AA	55-95, 75	55-95, 69	55-95, 85	55-95, 75	55-95, 69	55-95, 86
BB	65-85, 74.5	65-85, 69.5	70-90, 85.5	65-85, 75	65-85, 69.5	70-90, 85
CC	73.75-75.75, 74.35	68.5-70.5, 69.2	84.5-86.5, 85.85	73.75-75.75, 74.35	68.5-70.5, 69.25	84.5-86.5, 85.85
DD	74.15-74.55, 74.36	69-69.4, 69.23	84.65-86.05, 85.84	74.15-74.55, 74.36	69-69.4, 69.22	84.65-86.05, 85.84

Table 4.12 Range and selected percentage NO_x reduction level for each plant of experiments AA, BB, CC, and DD using FSGO.

	Percentage NO_x reduction level. Range, selected level					
	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
AA	55-95, 75	55-95, 69	55-95, 85	55-95, 75	55-95, 69	55-95, 86
BB	65-85, 74.5	65-85, 69.5	70-90, 85.5	65-85, 75	65-85, 69.5	70-90, 85
CC	73.5-75.5, 74.4	68.5-70.5, 69.1	84.5-86.5, 85.9	73.5-75.5, 74.55	68.5-70.5, 69.1	84.5-86.5, 85.8
DD	74.3-74.7, 74.46	68.9-69.3, 69.23	85.7-86.1, 85.79	74.3-74.7, 74.35	68.9-69.3, 69.24	84.7-86.1, 85.79

Table 4.13 Cost, outlet NO_x, and savings of experiments AA, BB, CC, and DD using FHMCFPwSEC.

	Cost ($)	Outlet NO_x (lb/hr)	Savings ($)
AA	65,596,511.96	189.89	87,281.61
BB	65,592,697.96	189.955	91,095.61
CC	65,589,123.54	189.997	94,670.03
DD	65,588,971.45	189.9996	94,822.12

Table 4.14 Cost, outlet NO_x, and savings of experiments AA, BB, CC, and DD using FSGO.

	Cost ($)	Outlet NO_x (lb/hr)	Savings ($)
AA	65,598,000.00	189.89	87,360.00
BB	65,594,230.00	189.955	91,130.00
CC	65,590,620.00	189.997	94,740.00
DD	65,590,460.00	189.9994	94,900.00

In tables 4.13 and 4.14 the savings are versus HMCFPwSEC and versus SGO, with pre-defined outlet NO_x level for each plant, respectively. Observe that the savings are around $90,000. In table 4.13, we do not divide the cost, outlet NO_x, and savings between τ at 0%, τ at 2%, FHMCFPwSEC single cut, and FHMCFPwSEC multi-cut methods because they are exactly the same. Again, we see a little difference in cost and savings between FHMCFPwSEC and FSGO, but still these values are very close. Observe that the outlet NO_x using FHMCFPwSEC and using FSGO are exactly the same, but for experiment DD. On experiment DD the difference of outlet NO_x between both models is only 0.0002 lb/hr.

In table 4.15, we present the CPU time and τ of experiments AA, BB, CC, and DD using FHMCFPwSEC and FSGO. In table 4.16, we show the wall clock time and τ of experiments AA, BB, CC, and DD using FHMCFPwSEC and FSGO.

Same as with experiments A, B, C, and D, the greater difference between models is the CPU time and wall clock time. However, in this case FHMCFPwSEC (single cut and multi-cut) uses less CPU time and wall clock time as compared to the FSGO. In table 4.15, it is clear that FHMCFPwSEC single-cut uses the least CPU time, regardless of τ at 0% or at 2%. Furthermore, with τ at 0%, single cut uses around 1.7 times less CPU time than targeting new layer, 1.6 times less CPU time than targeting regenerated layer, and 6.1 times less CPU time than FSGO. On the other hand, with τ at 2%, single cut uses around 1.8 times less CPU time than targeting new layer, 1.6 times less CPU time than targeting regenerated layer, and 4 times less CPU time than FSGO.

In table 4.16, it is clear that FHMCFPwSEC single-cut uses the least wall clock time, regardless of τ at 0% or at 2%. Furthermore, with τ at 0%, single cut uses around 2.3 times less wall clock time than targeting new layer, 2.1 times less wall clock time than targeting regenerated layer, and 10.9 times less wall clock time than FSGO. On the other hand, with τ at 2%, single cut uses around 2.3 times less wall clock time than targeting new layer, 2.1 times less wall clock time than targeting regenerated layer, and 7.4 times less wall clock time than FSGO. Since the results using τ at 0% and τ at 2% are exactly the same, in order to save time, it is recommended to use τ at 0%.

Table 4.15 CPU time and τ of experiments AA, BB, CC, and DD using FHMCFPwSEC and FSGO.

	CPU time (sec) τ = 0%, single	CPU time (sec) τ = 0%, new	CPU time (sec) τ = 0%, regenerated	CPU time (sec) FSGO	CPU time (sec) τ = 2%, single	CPU time (sec) τ = 2%, new	CPU time (sec) τ = 2%, regenerated
AA	331.27	613.42	535.85	2,033.36	482.52	898.84	732.59
BB	245.92	516.01	471.81	2,054.21	375.24	856.55	704.15
CC	277.34	440.25	445.02	2,051.06	490.46	862.72	783.48
DD	790.29	1,001.90	959.00	2,081.02	1,002.94	1,367.03	1,309.42

Table 4.16 Wall clock and τ of experiments AA, BB, CC, and DD using FHMCFPwSEC and FSGO.

	Wall clock (sec) τ = 0%, single	Wall clock (sec) τ = 0%, new	Wall clock (sec) τ = 0%, regenerated	Wall clock (sec) FSGO	Wall clock (sec) τ = 2%, single	Wall clock (sec) τ = 2%, new	Wall clock (sec) τ = 2%, regenerated
AA	210.88	486.86	442.47	2,039.77	285.91	662.33	575.80
BB	161.82	420.94	391.18	2,060.57	233.22	631.69	560.68
CC	157.52	370.60	343.80	2,052.19	267.21	620.57	569.66
DD	247.47	468.79	432.57	2,065.44	355.14	704.98	669.16

Observe that, in tables 4.15 and 4.16, FHMCFPwSEC single cut uses less CPU time and wall clock time than targeting regenerated layer or targeting new layer. The reason being that, in this experiment single cut, targeting regenerated layer, and targeting new layer use the same number of cuts, but targeting regenerated layer and targeting new layer add more infeasible schedules. None of these added infeasible schedules helps to reduce the number of cuts, but their creation uses more time.

Observe that always for FSGO the wall clock time is higher than the CPU time, but for FHMCFPwSEC the wall clock time is lower than the CPU time. Then, FHMCFPwSEC takes advantage of parallel execution on multiple cores/CPUs inside the computer, but FSGO cannot do the same.

Remember that, the only difference between experiments A, B, C, and D, compared to AA, BB, CC, and DD, is that we changed the days without an outage for plants 2 and 5, but the time using FHMCFPwSEC changes by a significant amount. We see that for FHMCFPwSEC this modification reduces the time, but for FSGO the time increases. The increase in time by FSGO is smaller than the reduction in time for FHMCFPwSEC. However, we can say that the constraint in days without an outage is an important constraint for both models.

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

This research develops SCR management mathematical optimization methods that, with a given set of scheduled outages for a fleet of power plants, minimize the total cost of the entire fleet of power plants and also maintain outlet NO_x below a desired target for the entire fleet. The plants have fixed NO_x at several discrete reduction levels (within regulation), the total outlet NO_x constraint is across all plants, and the cost depends on average NH_3 slip of the entire schedule, not an upper bound on NH_3 slip. Unlike SGO, we propose a fleet SGO (FSGO) that considers an outlet NO_x emissions limit for the entire fleet of power plants not for each power plant. Unlike MCFPwSEC, we propose a heuristic fleet MCFPwSEC (FHMCFPwSEC) that considers an outlet NO_x emissions limit for the entire fleet of power plants not for each power plant.

We explain the work of Phananiramai [10] in order to introduce the MCFPwSEC in our work. After mentioning that we use fixed NO_x policy, we present the difference between worst NH_3 slip and average NH_3 slip. We explain that, in order to calculate the cost of the edges, we want to estimate the average NH_3 slip as half the worst NH_3 slip. After we have a schedule that satisfies the percentage NO_x reduction level and also the worst NH_3 slip, we re-calculate the cost of the edges using the revised average NH_3 slip of the schedule, in order to obtain the updated cost of the schedule. We call this schedule the least cost schedule. We update the cost of edges with the average NH_3 slip of the least cost schedule, we cut the least cost schedule, and we continue our search of schedules. If we find another schedule that satisfies the percentage NO_x reduction level and also the worst NH_3 slip, we re-calculate the cost of the edges again. If this new schedule has a better cost than the cost of the least cost schedule, we denominate this new schedule as least cost schedule, we update the cost of edges with the revised average NH_3 slip of the least cost schedule, we cut the least cost schedule and we continue our search of schedules. We continue with this process until the cost increases some selected τ percentage versus the least cost schedule. Then, the last least cost schedule is the solution. We call the algorithm to solve this process HMCFPwSEC, and we present an example to illustrate this process. After that, we show computational experiments with two different levels of τ and we conclude that regardless of the selected τ, the cost increases if the percentage NO_x reduction level increases. In addition, if we maintain the percentage NO_x reduction level constant, then τ at 0% always uses the least CPU time. We compare the results using HMCFPwSEC with SGO, and we observe that τ at 2% always give us the same results as SGO, but τ at 0% only misses the same results one time, suggesting that the first feasible schedule found is almost always optimal.

In chapter 4, we use FHMCFPwSEC to reduce the cost and NO_x emissions for a fleet of power plants. We explain that the HMCFPwSEC cannot work directly to solve a fleet of power plants, because it is not possible to use a percentage NO_x reduction level for the fleet. Instead, we propose an FHMCFPwSEC that uses an outlet NO_x emission for the fleet. Then, we present a way to use different percentage NO_x reduction levels for each plant of the fleet. After solving them, we have the least cost schedule and the outlet NO_x for each different percentage NO_x reduction level for each plant. With this information, we create a problem and minimize it with a constraint on maximum outlet NO_x emission and with each plant used one time. With four different experiments, we show that we reduce the cost more $50,000 using an outlet NO_x emission limit for the fleet of power plants, instead of having the same outlet NO_x emission for the fleet, but with a percentage NO_x reduction level constraint pre-defined for each power plant alone. We realize that the first experiment gives us enough information to use only one more experiment to increase the savings instead of using the four experiments. For comparison, we use the same experiments with the SGO. We create the FSGO using the same logic used to create the FHMCFPwSEC. We solve the same experiments, and we note that the results in cost and savings are nearly the same, and the greatest difference is CPU time and wall clock time. Multi-cut FHMCFPwSEC targeting new layer uses less time than FSGO. We also show that τ at 2% obtains the same results as τ at 0%, but with τ at 2% we need more CPU time and wall clock time compared with τ at 0%. After that, we create four more computational experiments modifying two plants of our fleet, and we solve these new computational experiments with FHMCFPwSEC and with FSGO. We note that the results in cost and savings are nearly the same using both methods. The savings using FHMCFPwSEC and FSGO are around $90,000, compared to HMCFPwSEC and SGO with pre-defined outlet NO_x for each plant of the fleet. However, now our FHMCFPwSEC uses less CPU time than the FSGO, regardless of using FHMCFPwSEC single cut or any of the two FHMCFPwSEC multi-cut methods. We note that the constraint in days without an outage is a very important constraint for FSGO and FHMCFPwSEC.

For future research, we want to explore with more detail the relationship between days without an outage and CPU time for FHMCFPwSEC and for FSGO. In addition, we want to explore more cases when FHMCFPwSEC is faster than FSGO and vice versa. On the other hand, recall that we have the assumption that only one layer can be added or changed in an outage. Then, we want to explore if the relaxation of this assumption can reduce the SCR maintenance cost of the fleet. In addition, we want to explore the application of the methods of this research for power plants that use fuels different from coal.

Appendix A

Glossary

Average NH_3 slip is the average NH_3 slip level during the time horizon.

Disposal cost is the cost to dispose the used catalyst layer within regulations.

Fixed NH_3 slip policy uses a constant NH_3 slip level, and NO_x percentage reduction can vary.

Fixed NO_x policy uses a constant NO_x percentage reduction level, and NH_3 slip can change.

Inlet NO_x is the NO_x vented from the boiler and entering into the SCR.

Least cost schedule is each schedule that satisfies (3.2) to (3.7), the average daily NO_x reduction as well as the worst NH_3 slip (3.8), and has a real cost smaller than the previous least cost schedule.

Maximum NH_3 slip, also known as worst NH_3 slip, is the highest NH_3 slip level in the time horizon.

NH_3 injection is the NH_3 injected into the SCR.

NH_3 slip is the NH_3 that does not react in the SCR.

NO_x reduction level percentage is the reduction from the inlet NO_x to the SCR to the outlet NO_x of the SCR as a percentage of the inlet NO_x.

Outlet NO_x is the NO_x that does not react in the SCR.

Reactor potential of the catalyst in the SCR determines the quantity of NH_3 slip and Outlet NO_x emitted to the atmosphere.

Revised average NH_3 slip is the average NH_3 slip level obtained after we have the schedule.

Revised cost of a schedule is the cost of a schedule calculated using the revised average NH_3 slip.

Scheduled outage plan is a calendar with dates when the power plant will have an outage.

References

[1] L. Fleischman, R. Cleetus, J. Deyette, S. Clemmer, and S. Frenkel. "Ripe for Retirement: An Economic Analysis of the U.S. Coal Fleet". The Electricity Journal, Volume 26, Issue 10, Pages 51–63. December 2013.

[2] Tennessee Valley Authority. "How a Coal Plant Works". https://www.tva.gov/Energy/Our-Power-System/Fossil-Fuel-Generation/How-a-Coal-Plant-Works

[3] The U.S. Department of Energy and Southern Company Services, Inc. (1997, July). Topical report number 9, control of nitrogen oxide emissions: Selective catalyst reduction (SCR). http://www.netl.doe.gov/File%20Library/Research/Coal/major%20demonstrations/cctdp/Round2/topical9.pdf

[4] Environmental Protection Agency (EPA). "Overview of the Human Health and Environmental Effects of Power Generation: Focus on Sulfur Dioxide (SO2), Nitrogen Oxides (NOX) and Mercury (Hg)". 2002. http://www.epa.gov/clearskies/pdfs/overview.pdf

[5] United States Environmental Protection Agency. "Air Resources". April 2016. https://www.epa.gov/learn-issues/air-resources

[6] L. J. Muzio, R. A. Smith, and T. D. Martz. "New Tools for Diagnosing SCR Performance Issues". Meeting of the WSS/CI in Irvine, CA. Fossil Energy Research Corporation. Fall 2009. http://www.ferco.com/Files/P134R3.pdf

[7] United States Environmental Protection Agency. "Aquatic Life Criteria - Ammonia". March 2016. https://www.epa.gov/wqc/aquatic-life-criteria-ammonia

[8] J. E. Cichanowicz, L. J. Muzio, and M. C. Hein. "The First 100 GW of SCR in the U.S. - What Have We Learned?". Paper # 129. Mega Symposium, Baltimore, Maryland, 08/06. http://www.ferco.com/Files/Paper129.pdf

[9] United States Environmental Protection Agency. "Progress Report, Market Activity". February 2016. https://www3.epa.gov/airmarkets/progress/reports/market_activity.html

[10] P. Phananiramai. "Mathematical optimization techniques for managing selective catalytic reduction for coal-fired power plants". Ph.D. dissertation, University of Texas at Arlington, 2011.

[11] Fossil Energy Research Corporation (FERCo). "SCR Catalyst Management". http://www.ferco.com/SCR-Catalyst-Management.html

[12] J. Cichanowicz and L. Muzio, "Factors affecting selection of a catalyst management strategy," in Proceedings of the DOE-EPRI-EPA-AWMA Power Plant Air Pollutant Control Mega Symposium: DOE/EPRI/EPA/AWMA, Washington, DC, May 2003.

[13] J. E. Staudt and A. J. Engelmeyer, "SCR catalyst management strategies-modeling and experience," Presented at the EPA-DOE-EPRI Combined Power Plant Air Pollution Control Symposium, The Mega Symposium, May 19-22, 2003. http://www.andovertechnology.com/DOCUMENTS/Cat_Management.pdf

[14] K. Wicker and J. Staudt. "SCR Maintenance Fundamentals". Power Magazine, June 2004. http://www.andovertechnology.com/DOCUMENTS/Power_Jun04_Emissions.pdf

[15] J. E. Staudt, S. R. Khan, and M. J. Oliva. "Reliability of Selective Catalytic Reduction (SCR) and Flue Gas Desulfurization (FGD) Systems for High Pollutant Removal Efficiencies on Coal Fired Utility Boilers". The MEGA Symposium, Paper # 04-A-56-AWMA. 2004. http://www.andovertechnology.com/DOCUMENTS/MEGA_Paper_56.pdf

[16] C. Erickson and J. Staudt. "Selective Catalytic Reduction System Performance and Reliability Review", The 2006 Power Plant Air Pollution Control "MEGA" Symposium, Paper # 121, Baltimore, MD. August 28-31, 2006. http://www.andovertechnology.com/DOCUMENTS/Mega_2006_SCR_Paper.pdf

[17] J. E. Staudt and C. A. Erickson. “Selective Catalytic Reduction System Performance and Reliability Review”. Presented at Power-Gen International, Orlando. December 2-4, 2008.http://www.andovertechnology.com/images/Staudt%20Power%20Gen%202008%20paper.pdf

[18] S. Pritchard. “Long-term catalyst health care”. Power Magazine, January/February 2006.

[19] M. Kanniche, M. Mourigal and S. Gross. “Modeling of NOx selective catalytic reduction in pulverized coal power plants”. Chemical Engineering Transactions, 25, 671-676. 2011. DOI: 10.3303/CET1125112

[20] H. Liu, T. Guo, Y. Yang, and G. Lu. “Optimization and Numerical Simulation of the Flow Characteristics in SCR System”. Energy Procedia 17, 801-812. 2012.

[21] Y. Xua, Y. Zhanga, J. Wang, and J. Yuan. “Application of CFD in the optimal design of a SCR–DeNOx system for a 300 MW coal-fired power plant” Computers and Chemical Engineering 49, 50-60. 2013.

[22] Fossil Energy Research Corporation (FERCo). “CatalysTraK® - Manage: SCR Catalyst Management Software”. http://www.ferco.com/SCR-Catalyst-Management-CatalysTraK-Manage.html

[23] Z. Tian and S. Jin. “A mathematical model and the corresponding software for combining selective catalytic reduction with regenerative heat exchange”. Journal of Chemical and Pharmaceutical Research, 6(2):307-315. 2014.

[24] V. Chvatal, Linear Programming, W.H. Freeman and Company, 1983.

[25] H. A. Taha, Operations research: An introduction, 9th ed. Prentice-Hall, Inc., 2011.

[26] M. Manfren. “Multi-commodity network flow models for dynamic energy management – Mathematical formulation”. Energy Procedia 14, 1380-1385. 2012.

[27] K. L. Jones, I. J. Lusting, J. M. Farvolden, and W. B. Powell. “Multiocommodity network flow: The impact of formulation on decomposition”. Mathematical Programming 62, 95-117. 1993.

[28] M. E. Schmidt, Integrating Routing Decisions in Public Transportation Problems, Ebooks Corporation, New York: Springer. 2014. DOI: 10.1007/978-1-4614-9566-6

[29] P. C. Fusaro, “Energy Convergence: The Beginning of the Multi-Commodity Market”, New York: Wiley; 2002.

[30] P. Chardaire and A. Lisser, “Simplex and interior point specialized algorithms for solving nonoriented multicommodity flow problems”. Operations Research Vol. 50, No. 2, pp. 260–276. March–April 2002.

[31] M.T. Meloa, S. Nickela, and F. Saldanha da Gama. “Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning”, Computers & Operations Research 33, 181–208. 2005.

[32] L. V. Kantorovich, “Mathematical methods in the organization and planning of production. Publication House of the Leningrad University,” Translated in Management Science, vol. 6, pp. 366–422, 1960.

[33] L. R. Ford and D. R. Fulkerson, “A suggested computation for maximal multicommodity network,” Management Science, vol. 5, no. 1, pp. 97–101, 1958.

[34] T. Hu, “Multi-commodity network flows,” Operations Research, vol. 11, no. 3, pp. 344–360, 1963.

[35] D. T. Tang, “Bi-path networks and multicommodity flows,” IEEE Transactions, vol. 11, no. 4, pp. 468–473, 1964.

[36] J. A. Tomlin, “Minimum-cost multicommodity network flows,” Operations Research, vol. 14, no. 1, pp. 45–51, 1966.

[37] J.-L. Goffin, J. Gondzio, R. Sarkissian, and J.-L. Vial. “Solving nonlinear multicommodity flow problems by the analytic center cutting plane method”. Mathematical Programming, 76, 131-154. 1996

[38] W. Ben-Ameur, and J. Neto. “Acceleration of Cutting-Plane and Column Generation Algorithms: Applications to Network Design”. Networks, 49(1), 3-17. 2007.

[39] N. Nabona, “Multicommodity network flow model for long-term hydro- generation optimization”. IEEE Transactions on Power Systems, Vol. 8, No. 2, May 1993

[40] C. A. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. C. Sigismondi. “The fleet assignment problem: Solving a large integer program”. Mathematical Programming, 70, pp. 211–232. 1995.

[41] V. L. Pilla, J. M. Rosenberger, V. C.P. Chen, and B. Smith, “A statistical computer experiments approach to airline fleet assignment,” IIE Transactions, vol. 40, no. 5, pp. 524–537, 2008.

[42] H. Sadjady and H. Davoudpour. “Two-echelon, multi-commodity supply chain network design with mode selection, lead-times and inventory costs”. Computers & Operations Research 39, 1345–1354. 2012

[43] G. B. Dantzig and P. Wolfe. “Decomposition Principle for Linear Programs”. Operations Research, Vol. 8, No. 1, pp. 101-111. Jan. - Feb. 1960.

[44] P. C. Gilmore and R. E. Gomory. “A Linear Programming Approach to the Cutting-Stock Problem”. Operations Research, Vol. 9, No. 6, pp. 849-859. Nov. - Dec. 1961.

[45] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance “Branch-and-Price: Column Generation for Solving Huge Integer Programs”. Operations Research, Vol. 46, No. 3, pp. 316-329. May - Jun. 1998.

[46] M. Salani and I. Vacca. “Branch and price for the vehicle routing problem with discrete split deliveries and time windows”. European Journal of Operational Research 213, 470–477. 2011.

[47] I. Dayarian, T. G. Crainic, M. Gendreau, and W. Rei. “A branch-and-price approach for a multi-period vehicle routing problem” Computers & Operations Research 55, 167–184. 2015.

[48] V. Gunnerud, B.A.Foss, K.I.M. McKinnon, and B. Nygreen. “Oil production optimization solved by piecewise linearization in a Branch & Price framework”. Computers & Operations Research 39, 2469–2477, 2012.

[49] J. O. Brunner and R. Stolletz. “Stabilized branch and price with dynamic parameter updating for discontinuous tour scheduling”. Computers & Operations Research 44, 137–145. 2014.

[50] J. Belien and E. Demeulemeester. “Scheduling trainees at a hospital department using a branch-and-price approach”. European Journal of Operational Research 175, 258–278. 2006.

[51] M. J. Pereira Lopes and J.M. Valerio de Carvalho. “A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times”. European Journal of Operational Research 176, 1508–1527. 2007.

[52] T. Robenek, N. Umang, M. Bierlaire, and S. Ropke. “A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports”. European Journal of Operational Research 235, 399–411. 2014.

[53] K. Marriott and P. J. Stuckey. Programming with constraints: An introduction. Cambridge, Mass: MIT Press. 1998.

[54] E. C. Freuder and M. Wallace. Constraint programming. (pp. 239-272). Boston, MA: Springer US. 2005. doi:10.1007/0-387-28356-0_9

[55] IBM (2013). CPLEX callable library version 12.5.1. Retrieved August1, 2014, CPLEX > User's Manual for CPLEX > Languages and APIs: http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r5/index.jsp

[56] G. L. Nemhauser and L. A. Wolsey, Integer and combinatorial optimization. Wiley-Interscience, 1999.

Biographical Information

Antonio Alejandro Alanis Pena received his B.S. in Chemical Engineering from Tecnologico de Monterrey in 2000 and his Master in Environmental Planning and Management from Tecnologico de Monterrey in 2002. He obtained his Master in Engineering with Specialization in Quality Systems and Productivity from Tecnologico de Monterrey in 2008 and the Certification Black Belt in Six Sigma by Arizona State University in 2008. He worked in Tecnologico de Monterrey as online Instructor at Master Level for 10 years. He worked as an Environmental Science/Lab Summer Instructor on the Upward Bound Math & Science Center, a TRiO Pre-College Program at The University of Texas at Arlington. Additionally, he worked as a Graduate Teacher Assistant in the department of Industrial, Manufacturing, & System Engineering at The University of Texas at Arlington. His research interest are in the areas of optimization and environmental management.