ADVANCED AND NOVEL MODELING TECHNIQUES FOR SIMULATION, OPTIMIZATION AND MONITORING CHEMICAL ENGINEERING TASKS WITH REFINERY AND PETROCHEMICAL UNIT APPLICATIONS

A Dissertation

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Doctor of Philosophy in The Department of Chemical Engineering

by
Gregory Robertson
B.S. Louisiana State University, 2007
May 2014
This work is dedicated to those who supported me during my research, my wife Jennifer Robertson and the LA taxpayers, along with those who benefit from research, especially my born and unborn children.

Through my experience during my graduate years, I have learned that research is peering into the future at the expense of tasks at hand.

Therefore, researchers are reliant on those who nurture them during long spells of trials and errors. I have been supported part through scholarships provided by LA and part my wife. It has often felt selfish to choose this road, thank you for your graceful and unwavering support.

However, peering into the future is most often a selfless task considering the brevity of life. I hope the main benefactors, the future generations, continue to look up after climbing on top the shoulders of those before them.
ACKNOWLEDGEMENTS

I want to begin by thanking my research advisor, Professor Jose A. Romagnoli, for his positive influence during my graduate years. In my doctoral work, he gave enough room for me to find things that interest me while guiding me throughout. This careful balance of motivation and boundaries channeled my energy to grow into the researcher I am today, with which I am very happy.

“Jose, thank you for teaching how independent research is performed, using balance and facilitating creativity.”

I also want to acknowledge my family and friends, and especially my mother (Susan Murphy), my father (Barnaby Robertson), my best friend and brother (William Robertson) and my love and wife, (Jennifer Robertson). They have been the support structure that has helped me achieve throughout my life. I could not have asked for a better support system around me. Your belief in me is the source of my inspiration and the driving force of my perseverance.

“Mom, Dad, Billy, and Jen; I am better as a piece of what you all form around me than the entirety of what I am by myself.”

Finally, I want to thank my research group peers, who have always been there to help me get grip of the daily gyrations of research work. In part, this research work has come to fruition through their efforts, especially Michael Thomas and Aryan Gerali with whom I worked personally.

“Guys, no one stand alone. This dissertation is aided by your contributions. Thank you for all your help!”
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS .............................................................................. iii

ABSTRACT ................................................................................................. vii

CHAPTER 1: INTRODUCTION .................................................................... 1
  1.1 Thesis Focus ..................................................................................... 1
  1.2 Mathematical Modeling Introduction ........................................... 1
    1.2.1 Overview of Model Structures ................................................. 2
    1.2.2 Model Properties ...................................................................... 3
  1.3 Chemical Engineering Modeling Tasks ......................................... 4
  1.4 Contributions of Dissertation ...................................................... 5
  1.5 Organization of Dissertation ....................................................... 6

CHAPTER 2: INDUSTRIAL PROCESS MODELING/SIMULATION .......... 7
  2.1 Generic Structured Modeling Methodology .................................... 7
    2.1.1 Identifying Control Volumes ..................................................... 7
    2.1.2 Material Balance Equations ..................................................... 8
    2.1.3 Energy Balances ..................................................................... 9
    2.1.4 Kinetic Equations ................................................................... 10
    2.1.5 Thermodynamic Relationships .............................................. 11
    2.1.6 Logistical Equations ............................................................... 12
    2.1.7 Solution Methods .................................................................... 14
  2.2 Improving Convergence ............................................................... 15
    2.2.1 Computation Saving Techniques - Specification Selection ........ 15
  2.3 Refinery Case Study Application ................................................. 16
    2.3.1 Process Description ............................................................... 16
    2.3.2 Process Simulation ............................................................... 22
    2.3.3 Repair Strategy Implementation ............................................ 24
    2.3.4 Results ............................................................................... 25
  2.4 Crude Oil Loading/Unloading Model ............................................. 25
    2.4.1 Supply Chain Background ..................................................... 26
    2.4.2 Process Description - PetroChem Supply Chain .................... 26
    2.4.3 Novel Crude Loading/Unloading Scheduling Model ............... 28
  2.5 Conclusions ............................................................................... 32

CHAPTER 3: INDUSTRIAL PROCESS OPTIMIZATION MODELING .......... 33
  3.1 Optimization Methodology ............................................................. 33
    3.1.1 Traditional Techniques .......................................................... 34
    3.1.2 Stochastic Optimization ........................................................ 36
  3.2 VNS/TA Metaheuristic with Repair Strategy ................................. 37
    3.2.1 VNS/TA Hybrid Metaheuristic .............................................. 37
    3.2.2 Stochastic Optimization Integration with Repair Strategy ........ 40
  3.3 Stochastic Optimization of Distillation Utilizing Repair Strategy ... 42
    3.3.1 Introduction ....................................................................... 42
    3.3.2 Problem Motivation and Description ..................................... 44
ABSTRACT

Engineers predict, optimize, and monitor processes to improve safety and profitability. Models automate these tasks and determine precise solutions. This research studies and applies advanced and novel modeling techniques to automate and aid engineering decision-making.

Advancements in computational ability have improved modeling software’s ability to mimic industrial problems. Simulations are increasingly used to explore new operating regimes and design new processes. In this work, we present a methodology for creating structured mathematical models, useful tips to simplify models, and a novel repair method to improve convergence by populating quality initial conditions for the simulation’s solver. A crude oil refinery application is presented including simulation, simplification tips, and the repair strategy implementation. A crude oil scheduling problem is also presented which can be integrated with production unit models.

Recently, stochastic global optimization (SGO) has shown to have success of finding global optima to complex nonlinear processes. When performing SGO on simulations, model convergence can become an issue. The computational load can be decreased by 1) simplifying the model and 2) finding a synergy between the model solver repair strategy and optimization routine by using the initial conditions formulated as points to perturb the neighborhood being searched. Here, a simplifying technique to merging the crude oil scheduling problem and the vertically integrated online refinery production optimization is demonstrated. To optimize the refinery production a stochastic global optimization technique is employed.

Process monitoring has been vastly enhanced through a data-driven modeling technique Principle Component Analysis. As opposed to first-principle models, which make assumptions about the structure of the model describing the process, data-driven techniques make no assumptions about the underlying relationships. Data-driven techniques search for a projection that displays data into a space easier to analyze. Feature extraction techniques, commonly dimensionality reduction techniques, have been explored fervidly to better capture nonlinear relationships. These techniques can extend data-driven modeling’s process-monitoring use to nonlinear processes. Here, we employ a novel nonlinear process-monitoring scheme, which utilizes Self-Organizing Maps. The novel techniques and implementation methodology are applied and implemented to a publically studied Tennessee Eastman Process and an industrial polymerization unit.
CHAPTER 1: INTRODUCTION

This thesis organization is designed to give the reader a refresher on the modeling terminology, principles, and classifications. Three different Chemical Engineering tasks which utilize modeling are discussed in independent chapters. Each chapter begins with the state of research of the related modeling subjects and discusses the novel contributions of the work done by the author of the thesis.

The organizational structure of the Introduction Section begins by outlining the focus of the thesis. It then proceeds by briefly describing some aims, structures and properties of models utilized throughout this thesis. Then, the Chemical Engineering tasks aided by mathematical models are stated. Finally, a list of the contributions to the chemical engineering tasks through the use of modeling techniques this research has made is given before giving a more detailed organization of the thesis.

1.1 THESIS FOCUS

In an effort to automate and improve decision-making, chemical engineers create tools to aid their goals of predictability, profitability, and safety. Process control engineers specifically perform predicting, optimizing, and process monitoring tasks. Complex mathematical models are solved through a computationally expensive iterative procedure. With advancements of computational ability, modeling software’s ability to mimic industrial problems has improved significantly. This has allowed engineers to push the boundaries of the size, complexity and detail of the engineering problems they model to make process decisions with a vastly improved level of accuracy. The purpose of this research is to study and apply advanced and novel mathematical modeling techniques to aid in solving chemical engineering problems. A large amount of the creativity of this work has been not only in the designing methodology, but also in overcoming practical implementation difficulties.

1.2 MATHEMATICAL MODELING INTRODUCTION

The goals of modeling are to describe processes using mathematical equations in the simplest form to represent processes to a certain degree of accuracy. Since the goal of modeling is balancing a trade-off between simplicity and accuracy, a model can be judged not by being “correct,” but by the simplicity of the model given a degree of acceptable accuracy. It is common that many models can be created to describe the same process. Following an Occam’s razor approach, among models with relatively the same accuracy, the model with simplest ease of analysis and fewest assumptions should be chosen. The idea of model simplicity is important and will be echoed throughout this thesis. At some point, a modeler can add variables or equations to more realistically describe a process, but it is not worth the computational expense and the higher difficulty of comprehension.
1.2.1 Overview of Model Structures

A mathematical model is a symbolic object that tries to describe or predict a phenomenon. Mathematical models can be formulated to solve many different types of problems giving them different attributes. The structure (or lack thereof) of the model chosen is dependent on the problems for which the model is being developed.

Models that assume a structure of the variable relationships are composed of one or more equations. Mathematical modeling equations are composed of elements that can vary or are constant. Constant values are parameters determined outside of the model, known as exogenous, while variables change. The degree of freedom (DOF) is an important characteristic, which is the difference of the number of equations and the number of variables. The DOF gives the number of variables which when specified (given value to) all other variables are defined. Variables in models can be classified as input and output depending on their practical role to the engineer. The input variables are the variables we set the values of, such as manipulated or decision variables. The output variables of the model are the variables the engineer determines with the model, such as control or objective-related variables. The DOF also determines the use of the model for different problems. Models with more equations than variables have either redundant equation or are infeasible. Models with equal or more equations than variables are used differently in engineering problems.

Models with an equal number of equations and variables have zero DOF. These models consist of a set of equations describing the phenomenon, which is equal to the size of the observed variables describing them. These models define feasible spaces where observable variables are related to one another. In other words, these models possess a unique solution for the unknown output variables when the input variables are specified. A feasible subspace of the observed space can be defined by moving input variables along their physical limits and plotting the output variables. These models are useful for predicting new operating regimes, model predictive control, and process design.

When a model has more variables than equations, the output variables of a set of specified input variables form a solution space. This means when a set of variables the size of the number of equations is defined, a set of solutions is possible for the remaining variables. These equations can consist of equality and inequality constraints. These models typically lend themselves well to optimization problems. Optimization models are used to aid in maximizing or minimizing. Optimization problems are structured in a fashion where the inequality and equality equations are known as constraints and an objective function is added. The objective function quantifies the item being maximized or minimized. Each objective function uses a degree of freedom. In optimization problems, the independent variables being manipulated to maximize the objective, the input variables, are known as decision variables. The dependent variables, which the decision variables define, are the output or state variables.

There are also data-driven modeling techniques that capture variable relationships without assuming any structure at all. They instead extract patterns from historical data.
Common approaches involve intermediate outputs in an abstract space that better relate to the target outputs. Data-driven modeling techniques are commonly referred to as features extraction. This parallels modeling goals of extracting the important information, while leaving less important details behind. These types of models are useful in process monitoring tasks such as fault detection. Pattern recognition covers a class of problems where data are organized into classes and the class is the output. When classes are not known a priori, then it is a pattern classification problem and clustering algorithms are used. When historical data is labeled with known classes, supervised techniques are used.

### 1.2.2 Model Properties

Model properties largely are a function of the types of variables and equations within the model. Some common model properties are described here that are used in this thesis here below.

An important distinction in a model is its linearity. Linear is derived from a Latin word meaning resembling a line. Linear functions have two properties, additively \((f(x + y) = f(x) + f(y))\) and homogeneity \((f(\alpha x) = \alpha f(x)\) for all \(\alpha\)). Typically, a model is defined as linear if it possesses all linear operators. Nonlinear models contain at least one term identifying a nonlinear relationship. There are exceptions in statistical linear models and differential equation models. More broadly, nonlinearity involves a sort of irreversibility. Nonlinear problems tend to be more difficulty to solve. Linearization using Taylor series expansion is a common approach to solving nonlinear problems.

Models with variables that are uniquely determined by the parameters of the model are deemed deterministic. Deterministic models produce the same outcome for a given input variable state. If at least one variable is random, the model is considered stochastic. Stochastic models where a probability distribution function can regenerate the data are considered statistical models. A statistical model is said to be linear if it is linear in its parameters, but not necessarily predictor variables. Statistical tests are typically formed into statistical models; i.e., student t-test for comparing means of two groups can be formulated as seeing if an estimated parameter of the model is different from 0. An important distinction of a distribution is parametric versus nonparametric distributions. Distributions that can be described by a set of parameters, such as mean and standard deviation, are parametric distributions.

Static models are composed of variables that are independent of time. Those with parameters changing in time are dynamic. Dynamic models try to predict how the process state is changing with time depending on the current state and input variables. The set of state variables comprises a geometric manifold. The manifold limits future states depending on the current state. It is modeled as a set of differential equations. Differential models are termed linear if the order of the derivatives is one. Differential models can sometimes be reduced to static or steady state using quasi steady state assumptions.

White-box models are derived from known information that can be available. White-box models. They can be deductive models, a logical structure based on theory or inductive
which arise from empirical findings. Black-box model is a system that no a-priori information is available. Since no information is available about the structure of the problem, black-box modeling depends on experimental information. The more a priori information available about the model, the more accurate many knowledge-based and first principle models are. Often the a priori information comes in forms of knowing the type of functions relating different variables. In black-box modeling, both the functional form of the variables relationships and the parameters must be estimated. An often-used approach is neural networks.

Continuous models contain all continuous variables. Continuous variables represent continuous objects, such as flows rates, temperatures, pressures, product qualities and other state variables. Continuous models are typical to predict and analyze unit which are composed of state variables. A discrete model possesses variables which are discrete such integers or binary. These variables typically correspond to decisions such as “how many” or “yes no”. Discrete models are used in scheduling and capacity expansion problems.

1.3 CHEMICAL ENGINEERING MODELING TASKS

Simulations of processes are mathematical models that can predict outputs for a certain set of inputs. Their prediction capabilities make them useful in many engineering tasks such as design, operation, monitoring, and control. When designing or altering a process, simulations help visualize the process which ensures proper sizing specifications and can foresee integration challenges. They also help evaluate the financial benefit of the project by predicting the key economic indicators. Simulations prediction capabilities are also useful in process optimization. The inputs of the simulation are typically the decision variables and the outputs are the variables pertaining to the objective being satisfied. Simulations can also be used in model predictive control. The input variables are the manipulated variables and current state of the control problem while the output variables are the target variables. Model predictive control predicts target variables over a horizon of a set of input variables a different time steps. The numerous applications of accurate modeling create large benefits in accuracy improvements.

Optimization is another critical Chemical Engineering task that can have significant financial benefit. Optimization of processes can be done if there are degrees of freedom. Inputs to the model the size of the degrees of freedom can be manipulated to optimize an output. A typical unit optimization manipulates the controlled inputs of a process to obtain the best product quality/cost of production targets. Some examples are maximization of yield with resource constraints or minimization of costs with quality constraints. In addition to unit optimization, chemical engineers frequently solve problems such as efficient allocation of resources and optimal scheduling. These problems can large considering the size of plant operations. Financial consequences can vary widely within a large solution space. Using heuristics to make decisions is efficient, but can miss more profitable operating conditions. Creating precise optimization models to search for global optima across large optimization problems is a critical task.
Process monitoring is important for safety and economy of a plant. Process monitoring involves the identification of abnormal operations. It has a goal of early detection of equipment failure that may lead to hazardous working conditions or off spec product. The abnormalities experienced in process plants consist of many types of malfunctions including unit failures, parameter drifts and process unit degradation. In large industrial plants, sensor data for thousands of process variables are measured every second. Often measurements of these variables are incomplete, insufficient or unreliable. This arises from a number of reasons including sensor failure, noise or presence of outliers. It can easily be envisaged that under such conditions operators may take erroneous decisions and actions that can worsen the situation. The decisions taken by an operator may depend on additional factors including differences in plant operation between shifts and individual response behavior. Skills, mental condition, time available for the action to be implemented also affect an individual’s response. It is therefore important to develop methodologies and technologies to assist operators, process engineers and managers in handling abnormal situations.

1.4 Contributions of Dissertation

During this research, I have:

- Simulated the main units of a refinery, the distillation columns
- Formulated a novel strategy to improve the robustness of the model’s convergence
- Optimized the production unit’s optimization using a stochastic technique that incorporates an interaction with the model’s solver repair strategy.
- Integrated the production optimization problem with a higher level Crude Scheduling problem by creating a novel model.
- Created a novel nonlinear process monitoring scheme utilizing Self Organizing Maps
- Implemented this scheme on a polymerization unit, saving the company ~100000k per year

Publications from this thesis:


1.5 Organization of Dissertation

Chapter 1 begins discussing the structure and types of mathematical modeling. Then, the purposes, goals, and Chemical Engineering uses of mathematical modeling are emphasized.

In the second chapter, a generic methodology for first principle modeling of chemical processes is presented. This methodology is implemented on a crude scheduling problem. A complex refinery unit simulation process demonstrates improving convergence with model simplification while focusing on maintaining model accuracy. Chapter 2 ends with the methodology applied to the Crude Oil Scheduling problem.

The third chapter discusses modeling effects on optimization problems. Techniques to integrate the repair strategy outlined in Chapter 2 and Stochastic Global Optimization algorithms are demonstrated. Chapter 3 ends exemplifying simplification techniques to integrate the CSP crude scheduling and Production unit models.

Chapter 4 gives a methodology for implementing data-driven models. The fourth chapter also details a methodology of creating an advanced process monitoring system is included that combines the mathematical techniques and the engineer’s intimate process knowledge. This describes recent control strategy advances such as dimensionality reduction applied to a refinery unit, a polymerization unit as well as Tennessee Eastman Process.
CHAPTER 2: INDUSTRIAL PROCESS MODELING/SIMULATION

A process model, or simulation, of a plant is the set of equations that describe the operations of a plant. The model typically consists of material, momentum, and energy balances, thermodynamic equations, kinetic relationships, financial equations and logistical constraints. Simulations of processes can predict process outputs for a defined set of process inputs. Their prediction capabilities make them useful in many engineering tasks such as design, operation, monitoring, and control. When designing or altering a processes, simulations help visualize the process which ensures proper sizing specifications and can foresee integration challenges. They also help evaluate the financial benefit of the project by predicting the key economic indicators. Simulations prediction capabilities also improve operations by being the model in process optimization. The inputs of the simulation include the decision variables and the outputs include the variables necessary to calculate the objective. Simulations can also be used in model predictive control. Model predictive control predicts target variables over a horizon of a set of input variables at different time steps in an effort to control the target variable’s trajectory. The input variables are the manipulated variables and current state of the control problem while the output variables are the target variables. These are a few examples of simulations helping to perform Chemical Engineering tasks.

In Section 2.1, a general methodology for formulating models is introduced. Section 2.2 introduces a novel technique for improving the convergence of models with a repair strategy. The method is applied to a refinery distillation in Section 2.3. Section 2.4 the theme of model simplification is reinforced with the development of a supply chain application, crude oil loading and unloading scheduling problem.

2.1 GENERIC STRUCTURED MODELING METHODOLOGY

Section 2.1 gives a brief overview of the general methodology for formulating models with the intent to remind the reader of certain terms and refresh the reader’s understanding of mathematical models. The types of equations used in the thesis work are discussed such as balances, thermodynamic equations, and logistical constraints. A brief overview of kinetics is included to do its importance in Chemical Engineering. Some important types of equations not used in this work, such as financial equations, are not mentioned.

2.1.1 Identifying Control Volumes

Breaking a process down by identifying units on a process, or supply chain, and identifying nodes, or control volumes, is the first step of the modeling process. A control volume is any closed region in space. Selecting the control volumes is an important step in the modeling process. Any control volume is in principle correct, but selecting control volumes that are more convenient than others can greatly simplify constructing the model equations. A control volume, or node, is not necessarily fixed in space, although it commonly should be. Common control volumes are reacting and separating units, mixing and splitting points, and storage and blending tanks. More abstract control volumes
include an age domain for batch processes and moving control volumes referred to as a basis in processes.

2.1.2 Material Balance Equations

After control volumes in the process are identified, the modeler then makes use of balance equations \((\text{Rate in} - \text{rate out} = \text{accumulation} - \text{rate of production} + \text{rate of consumption})\). Balance equations are derived by integrating a differential balance (the continuity equation) throughout the control volume. For distributed systems, this is necessary. Throughout this subsection, common forms of the terms in the balance equation are illustrated for lumped parameter systems. Balance equations are applied to extensive variables that are subject to a law of conservation such as moles, thermal energy, and the number of something. Often, measureable extensive variables are mapped to conserved extensive variables using intensive variables. \(\text{Rate in}\) quantifies the amount of a substance entering the control volume while \(\text{rate out}\) quantifies the amount leaving. The rate terms must account for all types of transport across the control volume. For open systems, the rate terms correspond to flow across the boundary. For closed systems, an age domain is typically considered a part of the control volume and the rate terms correspond to initial and final values. The accumulation is zero for steady state processes. Otherwise the problem is dynamic and a boundary or initial condition must be specified to solve. In the balancing equations for a process, the rate of production and consumption are caused by chemical reactions.

Material, such as moles, is the amount of something and is conserved. Elemental, compound, and total mole balances are types of material balances. Material balances are necessary for modeling many processes. Extensive measurable variables such as weight and volume are converted to a conserved amount form, such as moles, through intrinsic conversions such as density, species fractions, and molecular weights.

To compute the rate terms, one must consider convective transports. Convective transport includes flow from bulk fluid movement (advection) and diffusion. Advection is normally accounted for by measuring total flow \((F)\) across the control volume along with intrinsic conversions to a conserved substance like density \((\rho)\) converts to the flows overall mass, concentration \((C_i)\) converts to a particular species’ mass of a liquid mixture, or partial pressure \((p_i)\) for a particular species’ mass of a gas mixture. Rate terms for overall mass balances, liquid component balances, or gas component balances take the form \(F\rho, FC_i\) or \(Fp_i/P\) respectively. Other commonly used useful intrinsic values include weight fraction \((w_i)\), mole fraction \((x_i)\), and molecular weight \((MW_i)\). An often useful equation for a model is that many of these intrinsic species fraction conversions sum to one.

The diffusive mass transportation representation in the rate terms is formulated using boundary layer theory. Boundary layer theory assumes the diffusion is occurring at the thin layer at the boundary of the control volume. Fick’s law of diffusivity describes the diffusion as the multiplication of an intrinsic flux and the extensive area, \(A\), of the control volume where the diffusion is taking place. For a lumped parameter system, \textit{diffusive}
rate = AkmΔC, where km is a mass transfer coefficient and ΔC is the driving force. The driving force for a distributed system is the interface gradient and for a lumped parameter system, ΔC = C_{bulk} - C_{bl} where C_{bulk} is the concentration of the bulk phase system and C_{bl} is the concentration at the boundary. If a phase change occurs at the boundary, the driving is the difference between the concentrations is the bulk phase and the equilibrium concentration at the phase boundary and a diffusivity coefficient, for a 2-component system rate = AD_{AB} (C_{bulk} - C_{eq}). The equilibrium concentration, C_{eq}, is determined by the thermodynamic relationship between the phases discussed in Section 2.1.4.

Generation/consumption terms come about when a species is formed or used during a chemical reaction. The rate of reaction, r, is expressed as an algebraic relationship where the rate is a function of the concentrations (C_i) involved in the reaction. r = f(C_1, C_2, ...). The generation/consumption terms are a multiple of the rate of the reactions and the relatively quantity of the species in the reaction (the stoichiometric coefficient). Typically an extent of conversion one reactant, the limiting reactant, or product yield is defined. Generation or consumption amounts of other species are determined through their stoichiometric relationship with the limiting reactant.

The accumulation term for material balances is the total rate of change of the material in the control volume (d(ρV)/dt or d(C_iV)/dt are examples of overall or component balances). Common assumptions include constant density (ρd(V)/dt), constant volume (Vd(C_i)/dt), or constant cross sectional area (ρAdH/dt).

### 2.1.3 Energy Balances

The conservation of energy is another law that formulates a balance equation used in process modeling. Thermal energy is the portion of total energy that results in system temperature. It is important to a process model since energy strongly affects financial and pollutions goals. The thermal energy balance equation has many similarities to the material balance equation. An important intrinsic conversion in energy balances is the heat capacity, c_p, which is the amount of energy required to elevate a unit mass of something a temperature degree.

Thermal energy transports by convective and non-convective phenomenon. Thermal rate terms are defined relative to a reference temperature, T_{ref}. Advection is modeled by converting a stream’s flow, F, and temperature, T_{flow}, to the relative total energy flow, Fρcp(T_{flow} - T_{ref}). If the bulk flow in to the control volume is equal to the bulk flow out, the reference temperatures are typically the same and the rate terms are combined, rate in − rate out = FρcpΔT. ΔT is difference between the inflow and outflow for open systems and initial and final values for closed systems.

Convective transport of energy can occur analogously to diffusivity of mass transport. If the boundary of the control volume is not a fluid substance, energy flows according to conductive and to a lesser extent radiation methods. Fourier’s law describes conductive energy flow is usually described as the multiplication of an intrinsic flux and the extensive area, A, of the control volume where the diffusion is taking place. For a
lumped parameter system \( rate = -\frac{Ak\Delta T}{l} \), where \( k \) is the mass transfer coefficient and \( \Delta T \) is the driving force which is the difference between the fluid in the bulk phase and at the surface. If the boundary of the control volume is a fluid, convective heat-transfer is assumed that the heat transfers through a thin film and the equation takes the form \( rate = Ah\Delta T \) where \( h \) is the heat transfer coefficient. Convective and conduction heat transfer relations can be combined described into an overall transfer coefficient \( (U = 1/h_1 + l/k + 1/h_2) \) where \( l \) is the thinness of the material and \( h_1 \) and \( h_2 \) are the heat transfer coefficients of the two fluids respectively. The overall becomes useful in heat exchanger situations where a rate is represented as \( (UA_{\text{trans}}(T_{\text{in}} - T_{\text{out}})) \).

Generation/consumption of energy is proportional to the reaction rate by a scaling factor equal to the heat of the reaction; \( \Delta H_{\text{rxn}} \). \( \Delta H_{\text{rxn}} \) represents the change in enthalpy of the reaction. It simultaneous considers the change in heat and work when internal energy is too difficult to calculate. The heat of reaction can be determined by summing the heats of formation of the products and subtracting the sum of the heats of formation of the reactants. As with material balances, rate is in the generation terms. The generation and consumption terms is made a function of dependent variables using the kinetics of the reaction discussed in the next section.

The accumulation corresponds to a temperature increase of a substance with a particular heat capacity. Accumulation terms take the form \( Vpc_pdT/dt \) for constant volume constant density system.

### 2.1.4 Kinetic Equations

Kinetic equations determine the rate of reactions and are therefore are critical to determining the generation and consumption terms of the general balances. It is knowledge of these equations that distinguishes chemical engineers from other engineers. Modeling kinetics is important because the system that operates in the most efficient manner can be critical to the economic success of the process. For example: a process, which creates a large amount of undesired products, can cause subsequent purification expenses to unacceptably rise.

The rate of reaction is the rate at which one chemical species is being consumed to form another. The rate equation is a function of reacting materials and conditions such as species concentrations, temperature, pressure, or catalyst type (if any). The rate equation (or rate law) is some form of an algebraic equation usually determined experimentally. The rate may vary within the control volume, therefore a general expression for the generation or consumption term is \( \int r_i \, dV \). The expression simplifies by the type of reactor (batch or continuous). The component rate \( r_i \) is related to the overall rate, \( r \), by a multiplicative factor equal to the stoichiometric coefficient, \( \gamma_i \). It is also a useful relationship that \( r_i/\gamma_i = r_j/\gamma_j \) for all \( i \) and \( j \) species in a reaction since it correlates generation and consumption of all species only having to measure one.

The rate law form has a temperature and composition component: \( r = [k(T)][f(C_{A_i}C_{B_i} ...)] \) where \( k \) is a rate constant which is multiplied by some function of the concentrations.
Details of reaction rate law formulations are beyond the scope of this work, however some examples of reaction rate laws are given in Table 1.

<table>
<thead>
<tr>
<th>Rate Law Type</th>
<th>Reaction Expression</th>
<th>Rate Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-Order Rate Laws</td>
<td>A $\rightarrow$ B + C</td>
<td>$r = kC_A$</td>
</tr>
<tr>
<td>First Order Reversible</td>
<td>A $\leftarrow$ B</td>
<td>$r = k[C_A-C_B/K]$</td>
</tr>
<tr>
<td>Second Order Rate Laws</td>
<td>2A $\rightarrow$ B</td>
<td>$r = kC_A^2$</td>
</tr>
<tr>
<td>Nonelementary Rate Laws</td>
<td>A$\rightarrow$ B+C</td>
<td>$r = kC_A^{3/2}$</td>
</tr>
<tr>
<td>Heterogeneous Gas Reaction</td>
<td>A+B+Cat $\rightarrow$ C+D+Cat</td>
<td>$r = kP_A/(1+K_CP_C+K_AP_A)$</td>
</tr>
<tr>
<td>Enzymatic Reactions</td>
<td>A+B+Enz$\rightarrow$D+E+Enz</td>
<td>$r = kC_A/(K_M+C_A)$</td>
</tr>
<tr>
<td>Biomass Reactions</td>
<td>Substrate(S)+Cells(C) $\rightarrow$ C+Product</td>
<td>$r = kC_SC_C/(K_S+C_S)$</td>
</tr>
</tbody>
</table>

The rate constant is not truly a constant; rather it is dependent on temperature rather than concentrations. Its temperature dependence follows the Arrhenius equations, $k = Ae^{-E/RT}$ where $A$ is a frequency factor, $E$ is the activation energy, $R$ is the gas constant, and $T$ is the temperature.

### 2.1.5 Thermodynamic Relationships

Most processes deal with materials that consist of a variety of chemical substances. In these processes, we are concerned with the transfer of substances from one phase to another. Phases in contact exchange constituents until their compositions are static. The phases are then in equilibrium. Chemical processes take advantage of the fact that compositions of the two phases at equilibrium can be very different in order to separate mixtures in operations such as extraction, adsorption, distillation, leaching and absorption. Most chemical processes consist of removing undesired reactants from a raw material, reacting, then purifying the product and recycling any unreacted reactants. The essential separation processes are often financially substantial.

The essence of the thermodynamic problem is figuring out the composition value at which two contacting phases will obtain at equilibrium. As with converting observed variables to relevant variables in mass and energy balances, thermodynamic problems are solved by projecting the physical problem into an different space, solving the problem, and projecting back in physically significant terms such as temperatures, pressures and phase compositions. Gibb’s work performs the first two steps by defining a value known as the chemical potential for each phase and setting them equal at equilibrium.
Chemical potential is more abstract than some terms such as fugacity and activity. For a liquid vapor mixture, the Gibbs equation can be written for each component \( i \) as \( \varphi_i y_i P = y_i x_i f_i^0 \).

where, in the vapor phase, \( \varphi_i \) is the fugacity coefficient, \( y_i \) is the component composition, and \( P \) is the pressure and in the liquid phase \( y_i \) is the activity coefficient, \( x_i \) is the mole fraction, and \( f_i^0 \) is the fugacity at a standard state.

Activity coefficients of liquid mixtures and fugacity coefficients of vapor mixtures are functions of temperature, pressure and compositions. These functions are based off of concepts of ideality, where an ideal liquid has an \( \gamma_i = 1 \) and an ideal vapor has a \( \varphi_i = 1 \). These assumptions, with some additional, reduce the gas-vapor equilibrium equation to Rault’s Law, \( y_i P = x_i P f_i^0 \).

The functions describing the activity and fugacity coefficients to determine the equilibrium concentrations characterize non-ideality utilizing equations of state. The function describing the activity and fugacity functions are not discussed further, but these functions are used to determine the equilibrium values of process mixtures.

2.1.6 Logistical Equations

At the heart of using mathematical models to aid in decision making is including the decisions in the model itself. When a decision must be made in a yes or no fashion, it is mathematically quantified as an integer variable, either binary variable, \( y = \{0,1\} \). Decisions can also be a selection \( y = \{1,2,3,4,...\} \). Logistical equations usually arise from constraints on decision-making and decisions relationships to one another. They are used in scheduling and real options analysis. In this subsection we give some useful equations for common decision relationships.

A common dilemma in modeling during decision making is the changing of constraints and equations depending on which decision is chosen. A common approach to overcoming this issue is known as the Big M method which utilizes a very large number \( M \gg 0 \). Consider three examples of constraints that depend on decisions: 1) greater than or equal to \( (x \geq A) \), 2) less than or equal to \( (x \leq B) \), and 3) equal to constraints \( (x = C) \). If the constraint is only applicable if the decision, \( y = \{0,1\} \), is made, \( y = 1 \), then the Big M method can make the constraint unrestrictive. To loosen constraints 1, 2, and 3, model constraints as

\[
x + M(1-y) \geq A, \quad x \leq B + M(1-y), \quad x + M(1-y) \geq C \quad \& \quad x \leq C + My
\]

A common example of using the Big M method is the Fixed-Charge problem. If one is faced with a decision \( y = \{0,1\} \) of purchasing a good with a fixed charge, \( b \), for buying along with a variable charge, the unit price of the good \( a \), which depends on the amount of the good you buy, \( x \). The fixed cost \( b \) is commonly a transaction costs such as shipping
or establishing a connection. In general mathematical terms, if \( x \leq 0 \), \( \text{cost} = 0 \). If \( x > 0 \), then \( \text{cost} = b + ax \). The modeling equations typically take the form:

\[
\text{cost} = by + ax \quad \text{and} \quad x \leq My. \tag{2}
\]

The latter constraint ensures that if the item is not purchased, \( y = 0 \), then \( x \) must be 0. Otherwise, \( x \) is free to range under the very large number \( M \).

The Big M method is useful in either-or scenarios. For example, suppose you must choose a vendor A or B for a reactant. Vendors A and B’s reactant may have different quantity constraints. Either-or scenarios arise when there are mutually exclusive decisions, \( y_1 \) and \( y_2 \), each which bring their own sets of constraints,

\[
f(x_1, x_2, ..., x_n) \leq b_1 \tag{3}
\]

and

\[
g(x_1, x_2, ..., x_n) \leq b_2 \tag{4}
\]

Including the equation

\[
y_1 + y_2 = 1 \tag{5}
\]

Converting the constraints to

\[
f(x_1, x_2, ..., x_n) \leq b_1 + M(1 - y_1) \tag{6}
\]

and

\[
g(x_1, x_2, ..., x_n) \leq b_2 + M(1 - y_2) \tag{7}
\]

Here \( y_1 \) and \( y_2 \) are the decisions to go with Vendors A or B and \( f \) and \( g \) are their respective constraints. If \( y_1 \) is chosen, the equation 5 prevents \( y_2 \) from being chosen. \( y_1 \) being chosen eliminates the large number from \( f \) in Eq. 6 activating that constraint while \( y_2 \) equaling zero renders the \( g \) in Eq. 7 constraints unrestrictive.

In the previous demonstration, a simpler set of equations, and therefore more preferable, could have been chosen, but the technique is more generalizable. The set of constraints below perform the same tasks while reducing the number of variables and equations. The modeling equations

\[
f(x_1, x_2, ..., x_n) \leq b_1 + My_2 \tag{8}
\]

and

\[
g(x_1, x_2, ..., x_n) \leq b_2 + M(1 - y_2) \tag{9}
\]
Eliminating the need for the variable $y_1$ and the equation $y_1 + y_2 = 1$. However, one can see the generalization of the technique can be seen in a K-of-Three-Constraints example. Suppose three sets of constraints

\[ f(x_1, x_2, ..., x_n) \leq b_1 \]  
\[ g(x_1, x_2, ..., x_n) \leq b_2 \]  
\[ h(x_1, x_2, ..., x_n) \leq b_3 \]

For decisions $y_1$, $y_2$, and $y_3$. The modeling equations

\[ f(x_1, x_2, ..., x_n) \leq b_1 + M(1-y_1) \]  
\[ g(x_1, x_2, ..., x_n) \leq b_2 + M(1-y_2) \]  
\[ h(x_1, x_2, ..., x_n) \leq b_3 + M(1-y_3) \]

Along with

\[ y_1 + y_2 + y_3 = k \]

Effectively allow one to select any $k$ of the three decisions. One can see how this can be extrapolated any K-of-O-Constraints problem.

If-then constraints are another common occurrence when modeling decision types. Imagine the evaluation of several proposed related project for a fiscal time frame. Whether a project $i$ is selected, binary variable $y_i = 1$, it may have strict ramifications on the selection of the other projects due to resource or other logistical constraints. Some illustrative examples include if project 1 is done, project 2 must be done is represented with constraint $y_1 \leq y_2$. Conversely, if project 1 is not done, project 2 cannot be done is represented $y_1 \geq y_2$. If project 1 is done, project 2 cannot be done: $y_1 + y_2 \leq 1$. If project 1 is not done, project 2 must be done: $y_1 + y_2 \geq 1$. There are many versions of if-then constraints that arise in logistical modeling.

### 2.1.7 Solution Methods

Modeling involves solving sets of equations. If a model is composed of all linear relationships (See Chapter 1) and continuous variables, it is a linear problem. Linear model can be put into the form $Ax = b$. There are many methods to solving. Substitution solves for variables and plugs them into other equations. If $A$ is an invertible matrix, the solution can be found by inverting the $A$ matrix, $A^{-1}x = b$. 
Nonlinear and integer problems models are solved a bit differently. One method for nonlinear continuous variables is to linearize using a Taylor-series expansion and use the methods discussed for linear solutions. Commonly, the nonlinear model is put into the form of an optimization problem where the modelling error is the objective function being minimized. Optimization solution methods are discussed in more detail in Chapter 3. Many solution methods, sometimes referred to as solvers, are iterative procedures. A few examples are fixed point, secant, and most commonly Newton-Rhapson. Many variations of these procedures have been researched in order to improve the algorithms computational performance and reduce the sensitivity to initial conditions (such as the Inside-Out algorithm). Although there have been improvements in the sensitivity to techniques which solve sets of nonlinear equations, sensitivity to initial conditions are still an issue (Biegler, Grossman, & Weseterberg, 1985).

2.2 Improving Convergence

This section introduces a method to aid in convergence of nonlinear equations by relocating the solver. The repair algorithm has the ability to relocate the solver for non-converged models caused by poor initial conditions. To show the effectiveness of the proposed repair strategy, a large-scale industrial process, the primary units of a crude oil refinery is simulated and tested. The repair strategy is integrated into a higher-level optimization procedure in the Chapter 3.

2.2.1 Computation Saving Techniques - Specification Selection

In many cases the nonlinear set of equations has more variables than equations and the modeller is interested in the model conditions for a specific set of defined variables. These are the inputs of the model. The modeling equations are solved by setting (giving values to) variables known as active specifications, the inputs. The set of active specifications are the same size as the degrees of freedom of the problem. The active specification of interest to the modeler, $X$, may not be the easiest way of solving the set of equations. For example, calculating fluid properties at particular temperatures usually takes one calculation; however, calculating the temperature at a vapor pressure is an iterative process (Antoine’s equation). There may be multiple different sets of active specifications, $Y$, which are easier to solve than $X$.

Our approach will formulate new I.C.s, which can help attain convergence at non-converged specified values of set $X$, by using values obtained at solutions of the easier-to-converge sets $Y$. The variables of all sets $Y_i$'s which are not included in set $X$ are denoted by set $Z$. $Z$ can be a set of state variables of the problem whose values can be predicted. Deactivating the problem inputs and activating intelligently chosen state variables at predicted values efficiently formulates initial conditions. Figure 1 illustrates the algorithm block diagram for the proposed repair strategy.

The repair strategy tries to reach a solution at the specified values of $X$. If the model does not converge, an easier-to-solve set $Y_i$ out of a total of $n$, is activated at values from a number of prediction methods. If the model convergence at set $Y_i$’s values, this converged point is used as the I.C. to the original set of D.V.’s by switching back to the set $X$. If it
then converges, the set of D.V.s, $X$, was solved by changing the I.C. Once the algorithm has exhausted the set of easier-to-converge sets, $Y_i$'s, then the point is concluded to be infeasible.

Figure 1: Nonlinear Simulation Repair Strategy

**2.3 Refinery Case Study Application**

This section provides a summary of the mathematical simulation of the refinery distillation column heat integration model. It illustrates the application of the applies the convergence method in described in Section 2.2.

**2.3.1 Process Description**

The crude/vacuum distillation is the foremost step in the petroleum refining process. In these main refinery units, crude is separated by vapor pressures into fluids with differing properties. The separation train considered in this study consists of a pre-fractionator, atmospheric, and vacuum distillation columns (PDU, ADU, VDU) each preceded by a preheat train which exchanges heat of the atmospheric/vacuum column products and pump rounds and ending with a furnace to elevate the feed temperature of the column.

Various crude types wait in storage tanks before they are blended and pumped into the refinery. In our case study, Masila crude is blended with lighter Dubai crude for refining. In the refinery in our analysis, the crude is first heated between 135°C to 140°C by exchanging heat with hot streams including Heavy Naphtha, Kerosene Circulating Reflux,
Kerosene-1, and Diesel product streams. In the first reacting unit, the electrostatic desalter, salts are removed in a water phase. Inorganic salts are the main impurities separated in the de-salter (mostly Chlorides also Sulphides of Sodium, Potassium, Magnesium etc.) These contaminants can cause corrosion of downstream units, fouling of catalyst, and a high-pressure drop in the downstream circuit. Five to six percent water is used as a solvent to dissolve the ionic-bonded salts. The emulsion created is fed to the desalter. The brine solution is then separated and removed from the de-salter by providing adequate time for settling. If not removed, these contaminants can cause corrosion/fouling problems as well as high-pressure drop in the downstream circuit. The desalted effluent water is routed to the Effluent Treatment plant (ETP) after heat recovery.

The desalted crude’s pressure is increased through a booster pump and it is sent through a second preheat train. As in the preheat scheme upstream of the de-salter, hot streams from downstream distillation units are utilized as heating mediums for the crude oil, in our case Diesel and Diesel Circulating Reflux. Preheat Section II end with a furnace with a target outlet temperature ranging from 225°C – 250°C depending on the crude oil composition. The preheated crude then enters the Pre-fractionator. Around 5-6 wt% of the crude is flashed in the Pre-fractionator.

Pre-fractionator column is being provided to recover IBP -95°C cut Naphtha as the overhead product. The operating conditions are fixed so as to achieve condensation of the C3/C4 components in the overhead condenser. The recovery of IBP-95°C cut in this column also helps to reduce the vapor load in the crude column. The pre-topped crude from the Pre-fractionator column bottom is sent to crude column after being heated in crude preheat train and in the crude charge heater. The pre-fractionator is a column provided with 30 valve trays. This consists of 12 trays in stripping section and 18 trays in the rectification section. The pre-fractionator column operating conditions are summarized in Table 2.

<table>
<thead>
<tr>
<th>Table 2: Operation of Pre-fractionator for Refinery Case Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed Details</td>
</tr>
<tr>
<td>Feed rate</td>
</tr>
<tr>
<td>Feed temperature</td>
</tr>
<tr>
<td>Stripping steam rate</td>
</tr>
<tr>
<td>Column Details</td>
</tr>
<tr>
<td>Top reflux</td>
</tr>
<tr>
<td>Column top/bottom pressure</td>
</tr>
<tr>
<td>Column top / bottom temperature</td>
</tr>
<tr>
<td>Un stabilized naphtha flow rate</td>
</tr>
<tr>
<td>Reflux Ratio (L/D)</td>
</tr>
</tbody>
</table>

The pre-topped crude from the column bottom is routed through a third preheat train where it is heated by the HVGO product stream, HVGO circulating reflux, and VR. The stream is then heated in the furnace to a temperature of around 360-365°C for lighter
crude (370-375°C for heavy crude) and then fed to the main atmospheric distillation unit where side draws Heavy Naphtha, two Kerosene grades (1,2), and Diesel side draws are removed. The crude distillation column is a typical fractionation column with an overhead condenser and side strippers. It consists of 26 trays (5 stripping, 20 rectifying, 1 feed) and packing for vapor liquid contact. The cold reflux for condensing the products is provided by the overhead reflux and the pump arounds at different sections. The heat from the pump around and the product streams is recovered in the crude preheat trains. The unstabilized overhead liquid product from the condenser is routed to the stabilizer section for further treatment. The un-condensed gas (if any) is routed to the refinery fuel gas system or fired in the crude heater. The distillate products are drawn from the trays above the flash zone according to their boiling ranges and are steam stripped in the side strippers with the stripped vapors being routed to the main column.

This partly vaporized crude feed exiting the furnace enters the flash zone of the main crude column. Light hydrocarbons flash as the feed mixes with vapor coming from the stripping section of the column. The vapor consists of light hydrocarbons and medium pressure (310 kPa) steam having some degree of superheat that is introduced at the bottom of the column at approximately 980.7 kPa (g) and 245° C which entered the stripping zone to strip the lightest hydrocarbons from the topped crude product. The steam entering the bottom means that there is no need for an external heat reboiler at the bottom of the column. The steam also lowers the partial pressure of the light hydrocarbons allowing the column to operate at lower temperatures necessary to distil the overhead product. Non-flashed liquid moves down the stripping section which is largely bottom product called Reduced Crude Oil (RCO). The vapors from the flash zone rise and are contacted with cooler reflux liquid (overflash) flowing down the column in the rectifying section condensing heavier hydrocarbons in the rising vapor, which causes the liquid side draws. The overflash is controlled at around 3-5 volume percent. Reflux is produced both by the condenser and pump around circuits in which side draws are taken from the column and cooled in the preheat sections. Pump around flow rates are high in order to heat the crude entering the column at high rates.

Pump around circuits serve three purposes. First, they remove latent heat from hot flash zones vapors and help condense the side products. Secondly, they improve the efficiency of the preheat train by enabling heat recovery at higher temperature levels than the overhead condenser. Finally, they reduce the vapor load of the column allowing for smaller column sizes.

The distillate products are drawn from the trays above the flash zone according to their boiling range. All are stripped in side strippers to control their composition. In the column configuration, the HN, Kero-1, Kero-2 and Diesel products are drawn through side strippers. Kero-1 and Diesel products are routed to the preheat train. Reduced Crude Oil (RCO) is pumped as feed to Vacuum distillation Unit.

Since Heavy Naphtha and Kero-2 products have a low heat potential they are routed via product coolers before being sent to storage. The other product streams from CDU (Kero-1 & Diesel) are allowed to exchange heat with crude in the preheat train and are cooled in product coolers before being sent to storage. Operating conditions of the Atmospheric Distillation unit are summarized in Tables 3 and 4.
Table 3: Operating Condition of the ATM Column for Refinery Case Study

<table>
<thead>
<tr>
<th>Feed Details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Crude column feed rate</td>
<td>558.5 MT/hr</td>
</tr>
<tr>
<td>Feed temperature</td>
<td>383°C</td>
</tr>
<tr>
<td>Stripping Steam rate</td>
<td>5.9 MT/hr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column Details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Temperature</td>
<td>100°C</td>
</tr>
<tr>
<td>Bottom Temperature</td>
<td>364°C</td>
</tr>
<tr>
<td>Top Pressure</td>
<td></td>
</tr>
<tr>
<td>Bottom Pressure</td>
<td>85kPa</td>
</tr>
<tr>
<td>Flash zone temperature</td>
<td>294.2kPa</td>
</tr>
<tr>
<td>HN Circulating flow rate</td>
<td>271 MT/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>32°C</td>
</tr>
<tr>
<td>K Circulating flow rate</td>
<td>300 MT/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>73°C</td>
</tr>
<tr>
<td>D Circulating flow rate</td>
<td>356 MT/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>80.4°C</td>
</tr>
</tbody>
</table>

Table 4: ATM Distillation Product Specifications

<table>
<thead>
<tr>
<th>Overhead Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow rate</td>
<td>9.0 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>45°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heavy Naphtha Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut Point</td>
<td>169°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>47.4 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>245°C</td>
</tr>
<tr>
<td>Stripping steam</td>
<td>0.678 MT/hr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kero-1 Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut Point</td>
<td>268°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>88.4 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>168.4°C</td>
</tr>
<tr>
<td>Stripping steam</td>
<td>3.56 MT/hr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Kero-2 Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut Point</td>
<td>312°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>51.5 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>235°C</td>
</tr>
<tr>
<td>Stripping steam</td>
<td>1.0 MT/hr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diesel Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut Point</td>
<td>362°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>65.3 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>282.9°C</td>
</tr>
<tr>
<td>Stripping steam</td>
<td>2.24 MT/hr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RCO Product</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow rate</td>
<td>294.2 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>364°C</td>
</tr>
</tbody>
</table>
The topped crude from the column bottom is routed to the vacuum unit furnace. The transfer line temperature at the furnace outlet is maintained at around 400°C (varies with crude properties) to avoid excessive cracking. The hot oil from the furnace is transferred to the flash zone of the vacuum distillation column maintained below atmospheric pressure by the steam ejectors. The purpose of this unit is to make feed of required quality to be processed in Fluid catalytic cracking unit (FCCU). The topped crude is distilled under vacuum into four different cuts namely Vacuum Diesel Oil (VDO), Light Vacuum Gas Oil (LVGO), Heavy Vacuum Gas Oil (HVGO) and Slop Distillate (SD). The flash zone liquid, called Vacuum Residue (VR), is routed to storage as LSHS/FO or to a Bitumen unit. The process flow diagram of the crude distillation unit is shown in the Figure 2.

VDO is drawn and cooled in an air fin cooler. A part of this stream is returned to the column as pump, and the remaining is further cooled in a trim cooler and is routed to storage for Diesel blending. LVGO is drawn off and is cooled in steam generator, generating LP steam. A part of this stream is routed as pump around stream and is returned to the column. LVGO product stream is further cooled in the air fin cooler and routed to VGO storage along with HVGO. HVGO is drawn off, and one portion is routed as wash oil to the lower bed. The major HVGO flow is utilized for heating up crude in Crude Preheat Train Downstream of Pre-fractionator as discussed in the crude distillation section. An outlet from these exchangers is split into two streams. One portion of HVGO is returned to the column as HVGO pump around flow. The other portion of HVGO stream passes through a BFW/HVGO exchanger, which preheats the BFW to the utility Boilers. The outlet of this exchanger is cooled in Air Fin Exchangers and is routed to the storage. Slop distillate is withdrawn and is cooled to 235°C in the slop distillate product cooler and further cooled to 80°C using tempered water and is routed to storage. Vacuum residue is utilized to produce bitumen from the bitumen unit and furnace oil from the visbreaker unit during crude processing. Vacuum bottom has very high heat potential and is used to preheat the crude. One stream is cooled in crude/vacuum residue exchangers and taken to Bitumen unit. The rest of the vacuum residue is directed to the visbreaker. The operating conditions of Vacuum Heater & Vacuum Distillation unit are summarized in Tables 5, 6 & 7. Some separated products are finished while others must be further treated, but each can be assigned a value.

Table 5: VDU Feed Composition Details for Refinery Case Study

<table>
<thead>
<tr>
<th>Feed Conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RCO Feed Rate</td>
<td>295 Mt/hr</td>
</tr>
<tr>
<td>Inlet temperature</td>
<td>398.3</td>
</tr>
<tr>
<td>Inlet Pressure</td>
<td>26.7 kPa</td>
</tr>
</tbody>
</table>
### Table 6: VDU Feed and Product Specifications for Case Study

<table>
<thead>
<tr>
<th>Product Details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vacuum Diesel Oil Product</td>
<td></td>
</tr>
<tr>
<td>Cut Point</td>
<td>379°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>24.1 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>192°C</td>
</tr>
<tr>
<td>LVGO Product</td>
<td></td>
</tr>
<tr>
<td>Cut Point</td>
<td>441.5°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>0.05 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>268.4°C</td>
</tr>
<tr>
<td>HVGGO Product</td>
<td></td>
</tr>
<tr>
<td>Cut Point</td>
<td>485.1°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>79.5 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>308°C</td>
</tr>
<tr>
<td>Slop Distillate Product</td>
<td></td>
</tr>
<tr>
<td>Cut Point</td>
<td>555°C</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>40.9 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>342.7°C</td>
</tr>
<tr>
<td>Vacuum Residual</td>
<td></td>
</tr>
<tr>
<td>Flow rate</td>
<td>149.2 MT/hr</td>
</tr>
<tr>
<td>Temperature</td>
<td>371.9°C</td>
</tr>
</tbody>
</table>

### Table 7: Operating Conditions of the VDU for Refinery Case Study

<table>
<thead>
<tr>
<th>Column Conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Temperature</td>
<td>114.5°C</td>
</tr>
<tr>
<td>Flash Zone temperature</td>
<td>381.7°C</td>
</tr>
<tr>
<td>Top Pressure</td>
<td>2kPa</td>
</tr>
<tr>
<td>Flash Zone Pressure</td>
<td>13.33kPa</td>
</tr>
<tr>
<td>Pump Around Details</td>
<td></td>
</tr>
<tr>
<td>Vacuum diesel</td>
<td></td>
</tr>
<tr>
<td>Circulating flow rate</td>
<td>82.9 Mt/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>114</td>
</tr>
<tr>
<td>Return temperature</td>
<td>78.3°C</td>
</tr>
<tr>
<td>Light vacuum Gas Oil</td>
<td></td>
</tr>
<tr>
<td>Circulating flow rate</td>
<td>122 Mt/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>83.4</td>
</tr>
<tr>
<td>Draw Off temperature</td>
<td>185°C</td>
</tr>
<tr>
<td>Heavy Vacuum Gas Oil</td>
<td></td>
</tr>
<tr>
<td>Circulating flow rate</td>
<td>150Mt/hr</td>
</tr>
<tr>
<td>ΔT</td>
<td>66.5</td>
</tr>
<tr>
<td>Draw Off temperature</td>
<td>242°C</td>
</tr>
</tbody>
</table>
2.3.2 Process Simulation

The refinery process simulation is developed using Aspen HYSYS. The simulation of petroleum processes is unique and challenging due to the complex and dynamic nature of these processes such as the complex feedstocks, highly coupled and integrated processes, stringent product specifications, and environmental regulations. Figure 2 depicts the Process Flow Diagram (PFD) simulation of the refinery process.

Figure 2: PFD Simulation of Process

For this study a crude oil blend 42 wt% - Masila & 58 wt% - Dubai crude is selected. The blending of different stocks is normally done to obtain the required product yields and to meet the process constraints. The crude assay data is presented in Table A6. The thermodynamic fluid package selected is Peng Robinson, an equation of state, which is recommended for the petroleum components. The equation of state is important as stated in Section 1 it is used to find equilibrium concentrations. Since the exact composition of the crude is unknown and is defined in terms of distillation temperatures, the feed developed is a combination of pure library components (lighter components) and pseudo components. The lighter components (methane, propane, i-butane, n-butane, i-pentane, n-pentane and hexane) are added to the pure component library. The data from the crude assay is used to define the petroleum pseudo-components. The pseudo components are the theoretical components that are not readily available in the component library and have to be defined. The data from the pure component library are used to represent the defined light components in the crude oil. It is required to input the laboratory distillation curve (TBP or ASTM data) and any bulk property such as Molecular Weight, Density, or Watson K Factor. It should be noted that the more the information is provided to the simulation, the accuracy of the property prediction is improved. In this study, the light
end composition, TBP distillation curve, density, and viscosity @ 10 & 50 deg are used in characterizing the oil. Each crude type is characterized separately and the required crude oil blend is defined and installed into the flow sheet. The calculated TBP data by HYSYS for the given crude is compared to the input data to identify any inaccuracies. The data is summarized in the Table 8.

Table 8: Assay Data for Dubai and Masila Crude

<table>
<thead>
<tr>
<th>Properties</th>
<th>Masila Crude</th>
<th>Light End Analysis</th>
<th>TBP distillation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density 15 °C, kg/m³</td>
<td>874</td>
<td>Component wt %</td>
<td>°C   wt % vol %</td>
</tr>
<tr>
<td>° API</td>
<td>30</td>
<td>Ethane 0.02 0.05</td>
<td>15   1.4 1.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Propane 0.29 0.5</td>
<td>15</td>
</tr>
<tr>
<td>Viscosity, cSt at 10°C</td>
<td>20</td>
<td>iso-Butane 0.23 0.36</td>
<td>149 15.6 19.2</td>
</tr>
<tr>
<td>Viscosity, cSt at 50°C</td>
<td>5.9</td>
<td>n-Butane 0.86 1.29</td>
<td>232 28.9 33.8</td>
</tr>
<tr>
<td>Pour Point, °C</td>
<td>-30</td>
<td></td>
<td>342 48.6 53.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>362 53.4 58.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>509 74.4 78.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>550 79.3 82.7</td>
</tr>
<tr>
<td>Dubai Crude</td>
<td>868</td>
<td>Component wt %</td>
<td>°C   wt % vol %</td>
</tr>
<tr>
<td>° API</td>
<td>31</td>
<td>Ethane 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Propane 0.05 0.09</td>
<td>15   0.39 0.3</td>
</tr>
<tr>
<td>Viscosity, cSt at 10°C</td>
<td>22</td>
<td>iso-Butane 0.1 0.22</td>
<td>32 1.09 1.28</td>
</tr>
<tr>
<td>Viscosity, cSt at 50°C</td>
<td>7.3</td>
<td>n-Butane 0.2 0.3</td>
<td>93   4.45 5.53</td>
</tr>
<tr>
<td>Pour Point, °C</td>
<td>-9</td>
<td></td>
<td>149 12.4 14.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>182 17.7 20.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>260 30.8 34.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>371 52.8 56.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>427 59.9 63.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>482 70.1 73.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>538 78.1 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>550 80.4 83.2</td>
</tr>
</tbody>
</table>

Some simulation techniques were utilized in order to gain higher convergence. Variables to ease in convergence were selected. This includes temperatures of the condenser, top stage, and feed. Recycles were reduced. For the vacuum circulating flows, the draws in the column were both connected to coolers and exchangers in the preheat streams. The
return temperature of the circulation flow rate was then defined using the exchanger outgoing streams. The exported stream was used to feed the heat exchanger. The stream exiting the cooler at the column had a temperature set by the stream exiting the heat exchanger used in the preheat train. This technique reduced the number of circulating flow rates.

2.3.3 Repair Strategy Implementation

In the refinery case study, modeling calculations involve solving the distillation unit models (a nonlinear set of equalities). The Newton-Rhapson technique was first used to solve the non-linear set describing the distillation columns, which minimizes error. A description of this technique can be found in (Goldstien & Standfield, 1970). The Inside-Out method for a solution of distillation columns was devised by Boston (Boston, A new Class of Quasi-Newton Solution Methods for Multicomponent, Multistage Separation Processes, 1970) (Boston & Sullivan, A new Class of Solution Methods for Multicomponent, Multistage Separation Processes, 1974). The Inside-Out algorithm offers significant advantages over Newton-Rhapson version in calculation speed and storage requirements (Russel, 1983). Although relatively insensitive to initial conditions, it has been noted that the Inside-Out algorithm can be subject to recognition of different optima, convergence to the wrong optima, and divergence from the optima (Biegler, Grossman, & Weseterberg, 1985) thus indicating that non-converged distillation simulations can occur due to poor initial conditions.

The specifications required are equal to the degrees of freedom. In our case, some of the active specs used are pump around flow rates, temperatures, and product flow rates. Easier-to-converge sets constructing a set Z composed of variables that can be interchanged with the decision-variable-inclusive set X as active specifications creates Y’s. Manipulating the heat removed with the pump around flow rates at constant return temperatures can be infeasible by conflicting with the heat added by steam and furnace duties at the cut point constraints, causing the distillation columns in the model to not converge. However, there is a set of active specifications that will more easily converge; the product flow rates corresponding to the cuts will commonly cause the solver to return to a converging solution. This is because the product flow rates are stronger functions of the composition of the feed rather than the degree of separation or unit conditions. The product flow rates can then be used as new variables, Z, to create easier-to-converge sets Y. When a point does not converge, the specification on the heat removal is relaxed by removing a PA stream flow rate. A product flow rate (PF) corresponding to one of the cut point constraints is then activated. If switching a PA to a PF brings the solver back into the converging region, then an I.C. is discovered where only one variable value differs. This is the strategy to create repairing-potential new initial conditions equal to the product of the number of pump around and the number of product flow rates.

Variables in the Z set, which are used to create Y sets for I.C.’s, must be assigned predicted values. Each incoming crude has about the same cut fraction, $x_{cut}$, depending on its composition; however it can change slightly due to operating conditions. The cuts are a strong function of feed composition, but product flow rates can be adjusted slightly and within the constraints; therefore, the product-cut flow rate values, $PF_{cut}$, are calculated
each time a neighborhood is constructed. At the operating condition of the center point of the neighborhood, the flow rate of each incoming crude oil, $FR_{Crude}$, is perturbed. Regressed values for the cut fractions are obtained from using GRG method. Predicted values for each cut are then formulated by summing each incoming crude oil by the formulated cut fraction,

$$PF_{cut} = \sum_{CrudeFeeds} FR_{Crude} * x_{cut}$$

(17)

2.3.4 Results

The results in Table 9 are experiments with points randomly selected within neighborhoods centered on a base case. Beginning with the base case as the I.C., points that did not converge initially underwent the repair strategy proposed in Section 2.2. Ten random points from five neighborhoods were chosen. Neighborhoods were defined as a proportion above and below the value of the centered point. The window sizes were .05, 0.1, 0.15, 0.2, and 0.25 for neighborhoods 1-5 respectively.

Table 9: Repair of Base Case Results

<table>
<thead>
<tr>
<th>Neighborhood</th>
<th>Number of Points Tested</th>
<th>Non-Converged runs</th>
<th>Repaired</th>
<th>Converged at alternative I.C.</th>
<th>Determined Infeasible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>21</td>
<td>3</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Percent of Total</td>
<td>100%</td>
<td>42%</td>
<td>6%</td>
<td>22%</td>
<td>14%</td>
</tr>
</tbody>
</table>

The repair algorithm helps gain convergence of nonlinear sets of equations. As can be seen from Table 1 Column 3, 42% of all points tested did not converge. Column 4 shows that 6% of points did not initially converge before using the repair strategy, indicating that they are feasible points that the base case was a poor I.C. During the repair process, 22% of points were able to converge at some $Y$ sets. Only 14% of points had no chance of converging.

2.4 Crude Oil Loading/Unloading Application

In this section the general modeling methodology is applied to a crude oil loading and unloading problem. The model’s intention is to be used as the process model of a supply chain optimization problem. It has novelty in that it can be integrated with the production unit model described in the previous section by using crude oil flows rates as terms.
2.4.1 Supply Chain Background

Supply chain management (SCM) is the oversight of materials, information, and finances as they move in a process from suppliers to manufacturers to wholesalers to retailers and then ultimately to consumers (Figure 3). Global solutions for supply chain problems can be computationally burdensome due to the size of the problem and the number of alternatives. Therefore, both within and among companies, objectives are naturally separated horizontally along the supply chain, with entities along the supply chain having their own local objective. The decision-making can also be separated vertically into layers by the time horizons they consider.

Figure 3: Supply Chain Overview

Process industry supply chains are striving to improve efficiency and profitability (Shah, 2005). Integrating different levels within a supply chain can improve profitability. Many planning problems have been addressed using managerial judgments where complex interactions between different decision making levels were disregarded. Recent developments in mixed integer process optimization provide new tools to help solve more complex problems of a company’s hierarchy (Puigjaner & Heyen, 2006). The decisions made by planning, scheduling, and control functions have a substantial economic impact on process industry operations - estimated to be as high as US $10 increased margin per ton of feed for many plants (Puigjaner & Heyen, 2006; Prausnitz, Lichtenthaler, & Gomez de Azevedo, 1999; Fogler, 2006; Stowell, 2010). Hence decision support tools for scheduling and planning can have a profound effect on the profitability of an enterprise.

Modern petroleum refining has become an extremely competitive business due to the deteriorating quality of crude oil coupled with tighter product specifications and more stringent environmental regulations. Furthermore, refineries today receive shipments of crude from a variety of sources. These crude oils are of different quality and composition and usually blending can improve the economics of the refinery. Therefore, refineries deal with a dynamic schedule of incoming crude which cause them to frequently change unit’s operating conditions to reduce operating expenses including environmental impact. Final products in a petroleum refinery are created by combining feed stocks emerging from distillation units, reformers, and catalytic cracker. The input streams which have varying chemical compositions and physical properties, are sent into common tanks or pools to be mixed into final products.

2.4.2 Process Description - PetroChem Supply Chain

Our analysis focuses on nodes involved in unloading, storing, blending, and processing. A general description of the oil production system considered in our analysis is shown in Figure 4. It consists of vessels, docks, storage tanks, and a separation train.
A refinery is a system composed of docks, pipelines, a series of tanks to store crude oil (and prepare different blends), CDUs, production units (such as reforming, cracking, alkylation, and hydrotreating), blenders and tanks to store intermediates and final products (Saharidis, 2009). In a typical refinery system, crude oil is shipped directly from overseas sources by tankers to docking terminals, which are connected to refineries through a pipeline network. Crude oil is stored in tanks, blended in additional tanks or a manifold, and then separated CDUs/VDUs. Separated crude cuts are converted to various intermediate products in production units and blended into final products. Final products are sent to distribution centers, and then transported by pipelines, trucks, or rail cars to customers.

Two local objectives along the supply chain are identified, minimizing the logistical cost of unloading crude oil (CSP) and maximizing the profit of the main refinery unit’s (distillation units) operating conditions. A model is sought to consider these problems simultaneously. In order to combine the models into one work, a model that includes both the CSP logistical costs and crude oil production unit is sought. In order to integrate the production unit model, the crude oil flow rates must be the linear terms or more generally decision variables) of the scheduling model because they are inputs to the production model.

(Lee et al. 1996) demonstrated a tradeoff between inventory and sea waiting costs. This paper demonstrated a tradeoff between unit operation and production level events by adapting Saharidis model. A natural extension of this work is to create model combining the advantage of the two aforementioned models. A model which 1) uses crude oil types for linear terms 2) considers the sea waiting, unloading, inventory, and refining costs, and 3) can account for multiple tank types. These characteristics will allow the scheduler to account for CDU blend changes, are compatible with both refining and feedstock selection problems, and will identify inventory and sea waiting tradeoffs in the schedule.
The loading and unloading CSP receives the shipping vessel’s schedule, arrival times and amounts, initial inventory quantities, and the CDU’s demand and blend range. The scheduler of the crude unloading must decide how to store the incoming crude oil and feed the refinery distillation units a proper blend of crude. The scheduler then gives then tells vessel when to hook up to the dock, tells personnel which network connection to ready, and provides inventory profiles of entities for visualizations. The scheduler does so in a fashion that minimizes production level costs such as inventory holding, unloading, sea waiting and setup. The scheduler achieves his/her goals by manipulating the unloading amounts of crude from vessels to storage tanks, transfer amounts from storage tanks to blending tanks for blending, and mixed crude oil amounts from blending tanks to CDUs. Before crude can be transferred from one entity to another, the scheduler must establish a connection between the entities.

2.4.3 Novel Crude Loading/Unloading Scheduling Model

Here, a novel model combining the advantage of the previous CSP models is proposed. A brief overview of the previous models is given in Section 3.4. For now, it is sufficient to say we would to consider all logistical costs and the unit operational costs. The model will use crude oil types for linear terms in order to be compatible with horizontal and vertical problems such as the production problem simulated in Section 2.3. The model takes into account the sea waiting costs, inventory costs, can account for multiple tank types. Equations within the model include operating rules for arrival and departure of vessels, material balances for vessels, material balances for storage tanks, material balances for blending tanks, and operating rules for the tanks. We discuss the problem formulation by breaking the equations and variables into four section, the vessel unloading schedule, inventories for the entities, network topology, and blending and capacity constraints.

Let us first consider the operating rules for arrival and departure of vessels. The incoming vessel time, \( TV\_V \), is an input to the model as the loading and unloading scheduler receives the incoming crude oil schedule. The scheduler can decide to initiate a vessel’s unloading and when a vessel leaves. The strategy is to divide the time horizon into time periods and binary variables are used to determine whether an action starts are terminates in during that spot. Let us first define these binary decision variables, weather a connection is establish between entities along the supply chain and how much is transferred between the entities. The scheduler must decide at what time \( t \) each incoming vessel \( v \) gets unloaded and leaves so Unloading Initiated Binary Variables (UIBV) and Vessels Leaves Binary Variables (VLBV) are created. With the defined decision variables, we can now define some important logistical constraints. For each vessel, unloading and leaving can only occur once. The decisions of when they come or leave are either–or constraints. So the constraints take the form:

\[
\sum_t \text{UIBV}_{v,t} = 1, \forall v \text{ and } \sum_t \text{VLBV}_{v,t} = 1, \forall v.
\]  

The outputs of the model include the vessel loading and unloading schedule. The vessel schedule is composed of the times each vessel \( v \) unloading initiation begins, \( TUIB_v \), and the times each vessel \( v \) leaves, \( TVL_v \). These times are defined from the input variables by...
summing the product of the time and the respective binary variable that represents the event:

\[ TUIB_v = \sum t \cdot UIBV_{v,t}, \forall v \quad \text{and} \quad L_v = \sum t \cdot VLBV_{v,t}, \forall v \]  

(18)

With the definition of the vessel unloading input, decision, and state variables, vessel unloading constraints can now be considered. For the model to make physical sense, the vessel must arrive at the dock before unloading can begin: \( TVA_v \leq TUIB_v, \forall v \). For the dock, vessels cannot begin unloading until the previous one is finished: \( B_{v+1} \geq TVL_v + 1, \forall v \). Each vessel takes particular time to unload represented by a number of periods to unload \((NPtoUnload)\).

\[ TUIB_{v+1} - TVL_v \geq NPtoUnload, \forall v \]  

(19)

Next, the model is extended to consider the connection scheduling. The scheduler must make a schedule for establishing connections between entities. Binary variables are created to establish connections between vessels and storage tanks, storage and blending tanks, and blending and distillation columns during the time periods \((t)\). The variables to represent these are defined as a vessel \((v)\) to storage tank \((i)\) connection established binary variable \((VSCBV_{v,i,t})\), storage tank \((i)\) to blending tank \((n)\) connection established binary variable \((SBCBV_{i,n,t})\), and a blending tank \((n)\) to distillation column \((k)\) connection establishment variable \((BDCBV_{n,k,t})\).

After establishing some decision variables, variables which describe the state of the process can be create. First variables that describe the whether entities along the supply chain are established. Vessels and storage tanks are connected binary variables \((VSCBV_{v,i,t})\), storage tanks to blending tanks are connected binary variables \((SBCBV_{i,n,t})\), and blending tanks to distillation columns are connected binary variables \((BDCBV_{n,k,t})\) are created to describe whether entities along the supply chain are currently connected.

These state variables allow us to represent certain are operational rules for the entities. Both storage and blending tanks cannot be emptied and filled at the same time:

\[ \sum_v VSCBV_{v,i,t} + \sum_n SBCBV_{i,n,t} \leq 1 \forall i, t \quad \text{and} \quad \sum_i SBCBV_{i,n,t} + \sum_k BDCBV_{n,k,t} \leq 1 \forall n, t \]  

(20)

Equation 20 represents these constraints respectively. Also, each distillation column \(k\) must be connected to one and only one blending tank:

\[ \sum_n BDCBV_{n,k,t} = 1 \forall k, t \]  

(21)

Similar to other entities, the vessel can be connected to no more than one storage tank, but only after unloading initialization has begun. A vessel cannot be connected to any storage tank before unloading initiation has begun. Both of these constraints can be represented by one equation:

\[ \sum_l VSCBV_{v,l,t} \leq \sum_{t' \leq t} UIBV_{v,t'}, \forall v, t \]  

(22)
Summing the vessel storage connection binary variables over all storage tanks equated to how many storage tanks are connected to the vessel. Summing the unloading initiation binary variable over all previous time periods determines whether the unloading initiation had not occurred. If unloading initiation had not occurred, the right hand side of the constraint would be zero and the vessel would not be allowed to connect to any tank. Otherwise, the right hand side would achieve its maximum value of 1 and the vessel could connect to one tank. In a similar fashion, unloading occurs before leaving occurs. This is performed by setting the sum of all vessel tank connection to be less than the sum of all vessels leaving decision for the rest of the time horizon:

$$\sum_{i} VSCBV_{v,i,t} \leq \sum_{t'\leq t} VLBV_{v,t'}, \forall v,t$$ (23)

The relationship between the network topology decision variables and state variables are formulated:

$$VSCBV_{v,i,t-1} + VSCEBV_{v,i,t} \geq VSCBV_{v,i,t} \forall v,i,t, SCBVi,n,t-1 + SCEBV_{i,n,t} \geq SCBVi,n,t \forall i,n,t, \text{and } BDCBV_{n,k,t-1} + BDEBV_{n,k,t} = BDCBV_{n,k,t} \forall n,k,t$$ (24)

In each equation for each time period, if the connection established binary variable is zero, then nothing occurs and previous time period’s state variables is equal to the current time period’s state variable. If a connection established is made, the previous connection variable must have been zero and the current variable must be one. The equation does allow for the connection to be established, but the previous and current connection variables to be zero. This is costly and an unnecessary case to consider due to the optimizer minimizing cost parameters.

Another important output of the model are the inventory profiles of the vessels, storage tanks, and blending tanks of each crude type $j$ at each time period $t$ ($VV_{v,j,t}$, $VS_{i,j,t}$ and $VB_{n,j,t}$ respectively). The inputs of the model are the initial volumes of the vessels, storage, and blending tanks ($VV_{v,j,t=1}$, $VS_{i,j,t=1}$, and $VB_{n,j,t=1}$). The decisions variables are the flows of the crude types from vessels to storage tanks, storage to blending tanks, and blending tanks to distillation units at each time period ($FVS_{v,i,j,t}$, $FSB_{i,n,j,t}$ and $FBD_{n,k,j,t}$).

We relate the input and output variables are related through inventory balances. We draw a control volume is not only the entity, but also the time since unloading has begun. The amount at the beginning of the time horizon (entering the control volume) is an input to the model. The amount exiting the control volume is the flow from vessel to all the storage tanks $i$ at all previous times, $FVS_{v,i,t}$. The inventory balance of vessel is written as:

$$VV_{v,j,t} = VV_{v,j,t=1} - \sum_{t'\leq t} \sum_{i} FVS_{v,i,j,t} \forall v,j,t, VS_{i,j,t} = VS_{i,j,t=1} + \sum_{t'\leq t} \sum_{v} FVS_{v,j,i,t} - \sum_{t'\leq t} \sum_{n} FSB_{n,i,j,t} \forall i,j,t$$ (25)

and

$$VC_{n,j,t} = VC_{n,j,t=1} + \sum_{t'\leq t} \sum_{i} FSB_{i,n,j,t} - \sum_{t'\leq t} \sum_{k} FBD_{n,k,j,t} \forall n,j,t$$ (26)
The overall balance on the distillation unit satisfies schedule must satisfy the demand of the CDU:

$$\sum_n \sum_j FBD_{n,k,j,t} = S_{k,t} \quad \forall \, k, t.$$  \hspace{1cm} (27)

As a necessary constraint, in order to ensure all the inventory of the vessel is unloaded. An overall balance is performed. The control volume does not include an age domain, only that for all time periods all the crude type \( j \) in vessels \( v \) must be unloaded.

$$\sum_t FVS_{v,i,j,t} = VV_{v,j,t=1} \quad \forall \, v, j$$  \hspace{1cm} (28)

Inventory decision and state variables allow the physically limiting constraints such as capacity. The capacity of the flow is either the maximum amount the pipe can contain \((FVS_{\text{max}})\). It is also constrained by either the vessel is hooked up:

$$\sum_j FVS_{v,i,j,t} \leq VSCBV_{v,i,t} \cdot FVS_{\text{max}} \quad \forall \, v, i, t, \sum_j FSB_{i,n,j,t} \leq FSB_{\text{max}} \cdot SBCBV_{i,n,t} \quad \forall \, i, n, t$$  \hspace{1cm} (29)

And

$$\sum_j FBD_{n,k,j,t} \leq BDCBV_{n,k,t} \cdot FBD_{\text{max}} \quad \forall \, n, k, t.$$  \hspace{1cm} (30)

This is analogous to using a Big M method, however the M is actual capacity and the constraint is always relevant.

We also now consider tank capacity and blending constraints. There is no blending in storage tanks and particular blending for the blending tanks. To model no blending in storage tanks, an additional state binary variable is created, whether storage tank \( i \) contains crude type \( j \) at time \( t \), \( STTBV_{i,j,t} \). An equation that does not allow multiple tank types simultaneously is imposed:

$$\sum_j STTBV_{i,j,t} \leq 1 \quad \forall \, i, t.$$  \hspace{1cm} (31)

Now that it is ensured only one crude type is in each storage tank, we can divide that capacity to crude types. For the crude type elected, it is less than the capacity. For all crudes not selected for the tank, the capacity is zero:

$$VS_{i,j,t} \leq STTBV_{i,j,t} \cdot \text{Cap} \quad \forall \, i, j, t$$  \hspace{1cm} (32)

The blending tanks capacities are modeled simply:

$$\sum_j VB_{n,j,t} \leq \text{cap} \quad \forall \, n, t.$$  \hspace{1cm} (33)

However, blending tanks have additional constraints of the amount of each crude type that can enter the distillation column.
\[ \forall n, j, t \quad VB_{n,j,t} - \sum_{i'} VB_{n,j',t} \leq M_j \cdot (1 - \sum_k BDCBV_{n,k,t}) \]
\[ \forall n, j, t \quad VB_{n,j,t} - \sum_{i'} VB_{n,j',t} \geq -M_j \cdot (1 - \sum_k BDCBV_{n,k,t}) \]

Using the Big M method, these constraints are only imposed if the blending tank is connected to a distillation column. Each distillation column has a minimum \((amin_j)\) and maximum \((amax_j)\) amount of crude \(j\) it can receive. These can be done by comparing the crude \(j\) in each tank and comparing to the amount allowed. The amounts allowed are the total crude amount in each tank multiplied by the upper and lower percentages. The difference between the crude in the tank and the maximum amount allowed should always be less than or equal to zero. The crude \(j\) in the tank the minimum amount of crude allowed in the tank should always be greater than zero.

### 2.5 Conclusions

In this chapter, an overview of the general modeling methodology was presented, which included a description of the important equations used in this thesis work. The methodology was applied to a crude oil loading and unloading problem. The model was done with the intention it can be integrated with the lower level production unit by including crude oil flow rates as the linear terms. Crude oil amounts were the inputs of a simulation that modeled the crude oil refinery units. A methodology for improving the robustness of convergence was proposed applied to the production unit. This was performed with the intention of the production unit simulation automatically repairs itself. The repairing allows the model to be combined with the crude scheduling problem. An area around the operating point was created and random point was selected to evaluate. 12.5\% of the points that did not converge were able to be repairs.
CHAPTER 3: INDUSTRIAL PROCESS OPTIMIZATION MODELING

Optimization is another critical Chemical Engineering task that can have significant financial benefit. Optimization of processes can be done if there are degrees of freedom. Inputs to the model the size of the degrees of freedom can be manipulated to optimize an output. A typical unit optimization manipulates the controlled inputs of a process to obtain the best product quality/cost of production targets. Some examples are maximization of yield with resource constraints or minimization of costs with quality constraints. In addition to unit optimization, chemical engineers frequently solve problems such as efficient allocation of resources and optimal scheduling. These problems can be large considering the size of plant operations. Financial consequences can vary widely within a large solution space. Using heuristics to make decisions is efficient, but can miss more profitable operating conditions. Creating precise optimization models to search for global optima across large optimization problems is a critical task.

Chapter 3 is organized as follows: Section 3.1 introduces optimization techniques and solutions. Section 3.2 discusses methods to improve robustness including integration with the repair strategy introduced in Chapter 2. Section 3.3 defines optimization of the process simulation introduced in Section 2.3. The optimization method used is stochastic optimization and the technique is integrated with the repair strategy. Section 3.4 discusses simplification techniques to combine a scheduling and production unit optimization problem. It utilizes the novel model for the crude oil scheduling loading and unloading problem developed in Section 2.4. It then describes decision-making differences when considering the multilevel integration.

3.1 OPTIMIZATION METHODOLOGY

Three things must be necessary to solve an optimization problem. First, a mathematical model of the process must be available along with the process variables that can be manipulated and controlled known. Secondly, an economic model is needed which models profit from sale of products and costs associated with their production such as raw materials, operating costs, fixed costs, taxes, etc… Finally, the optimum procedure must be selected to locate the optimal profit or costs defined by economic model subjected to constraints in material, process equipment, and the process model.

An optimization problem is modeled as an objective function along with variable relationships and constraints. In an optimization problem, the values of decision variables are manipulated to optimize (minimize or maximize) an objective function subject to a set of constraints

\[ \min(y(x)) \text{ or } \max(y(x)) \text{ st } g(x) = 0, \; h(x) \leq 0 \]  \hspace{1cm} (35)

Among the constraints are physically limiting constraints as well as the model constraints violated by a certain combination of decision variables that can be more difficult to define. In practical large-scale applications, models in simulation environments are used to mimic complex process behavior. In this case, model equations are embedded into the
optimization problem. Input variables are further defined where uncontrollable measurable variables are parameters. Uncontrollable unmeasured variables are random variables. Controlled variables are the independent input variables and the output variables are the objective function costs, profit, etc…

3.1.1 Traditional Techniques

The objective of optimization is to select the best possible decisions for a given set of circumstances without having to enumerate all possibilities (Pike, 1986). Optimization methods take advantage of the structure of the problem to find the best values efficiently. A general approach to solving optimization problems is presented. First, unconstrained twice-differentiable optimization problems are solved using traditional mathematical techniques. If constraints are present, we use Lagrangian techniques to transform the problem to an unconstrained optimization problem where traditional methods can be used. This is not always feasible. If the objective and constraints are linear, linear programming efficiently solves the problem. For polynomials, or posynomials, geometrics methods can be used. If the problem fits a stage structure, dynamic programming is useful. Otherwise, multivariable search techniques must be explored.

Classical theory of maxima and minima suggest checking necessary and sufficient conditions. There are three places to check for optima: stationary points, boundaries, and at discontinuities in the first derivative. For continuous unconstrained optimization, optima must be stationary points. Stationary points are points where the first derivative of the objective function is equal to zero with respect to all decision variables. Stationary points are classified as minima, maxima, or saddle points dependent on the Hessian matrix composed of the second partial derivatives.

Constrained optimization problems may contain optima at the constraint borders. They are typically converted to unconstrained optimization and solved using classical techniques. Constraints commonly exist even if only to exist to say variables are positive. Inequality constraints are typically handled by adding slack and surplus variables to form equality constraints. Quadratic slack and surplus terms must be used so that second derivatives exist. Once the problem is formulated as optimize \( f(x) \) subject to \( g(x) = 0 \), three methods are used to solve the problem: Direct Substitution, Constrained Variation, and the Lagrangian Method. Direct Substitution attempts to solve constraints for the independent variables and substitute them into the objective. Afterwards, classical methods are used. These substitutions are not always possible. Constrained variation utilizes an important principle that the changes in the dependent variables proportions are equal for the constraints and objective functions. This technique is not used much of practical reasons, but its theory is the foundation for multivariable optimization such as generalized reduced gradient.

The Lagrangian technique converts the constrained problem to an unconstrained problem. The Lagrangian function, \( L = f + \lambda g \), combines the objective function and constraints. It can be shown that the set of equalities, \( dL/dx = 0 \) and \( dL/d\lambda = 0 \), has the same solution as the optimization problem and a degree of freedom of 0. Solving the set of nonlinear equations is performed by minimizing the equations’ error. The problem has been
transforms into an unconstrained optimization problem where classical methods can be used. The sensitivity of the objective function to changes in the constraints is given by $\lambda$. The slack variable equations,

$$\frac{dL}{dx_s} = \frac{d(f + \lambda[g + x_s])}{dx_s} = \lambda x_s = 0$$ (36)

give an interesting result for inequality constraints. Either $\lambda$ or $x_s$ is 0, indicating if the optimal is on the border (known as the constraint being tight). Kuhn-Tucker conditions list all necessary conditions for $\min(y(x))$ subject to $f(x) \leq 0$. The conditions include the original constraints ($f(x) \leq 0$), the boundary constraints ($\lambda x_s = 0$), the Lagrangian conditions ($\frac{dL}{dx} = 0$ and $\frac{dL}{d\lambda} = 0$), and sign restrictions on the Lagrangian coefficients.

Classical techniques may not be feasible if the second derivatives do not exist. Particular techniques are available for the case that objective and constraints equations are all linear and all variables are positive. Linear programming is the most widely used technique and takes advantage of the model’s geometric features. For linear problems, the optimal must lie on an intersection of the constraints. The Simplex method is used to check the corners of the problem. Inequality constraints are converted to equalities by adding or subtracting first order slack or surplus variables. An initial solution is found by giving a set of variables the size of the DOF, which is known as the basis, a value of zero. The simplex method formulates a methodology by removing and adding variables from the basis which have positive effects on the objective. This is done until the objective can no longer be improved. This powerful method has made optimization problems that were once too difficult to solve are now commonplace. The ability to solve large sets of linear equations with the computation has improvement. Linear programming problems have been applied to industrial problems such as petroleum refining. In chapter 1, the desire of using simplest model was stated in the context of number of equations. It is worth noting that a model expressed in a linear form is a simpler model for the ease of solving. For example: $\max(x^2y^3)$ can be transformed to a linear problem by posing as $\max(2ln(x) + 3ln(y))$.

For cases where the process models have been done in simulations software or the process is too complex, the constraints may not be available. Search techniques can be used for complex problems where the economic model may even be the process itself. The most effective single variable search techniques are types of elimination procedures where experiments are placed in the search space, objective values are considered, and regions are excluded from having the objective function. These search techniques use search problems, search plans, unimodality, and minimax. Stochastic search problems tend to take longer to find the optima than deterministic. Search problems can be classified according to the number of independent variables whether they need single variable search techniques or multivariable search techniques. Search plans can be simultaneous experiment placing or sequential experiments. Simultaneous can include putting reactor temperature sensors along a bed to get the max temp in the bed. Sequential are typically preferred as experiments are expensive and sequential can place experiment optimally by considering the results of previous experiments. Search plans
take advantage of unimodality of the problem, a property of a function that has only one optima within an interval. Unimodality implies the function is always approaching the optima as one departs from the boundaries, but is not restricted to continuous functions. Regions can be excluded by comparing the results of the experiments. The Minimax principle ties to find a search plan with the minimum final interval of uncertainty. The Fibonacci Search plan is the most effective at reducing the final level of uncertainty, but requires the number of experiments to be an input. Golden Search method is effective and can be put forward for an indefinite number of experiments.

Mixed-integer problems contain one or more variables that are integers. If the problem is linear, the problem is termed a Mixed Integer Linear Problem (MILP). There are three general types of solutions for MILPs: LP-relaxation, cutting planes, and branch and bound. LP-relaxation relaxes the integer variables and allows them to vary continuously. The set of solutions at each decision rounded up and down are then compared with one another. This can be computationally burdensome. Cutting Plane algorithms successively tries to solve tighter LP-relaxation problems eliminating sections of the feasible space thought not to contain the objective function. This typically requires many cutting planes. Branch and bound (BAB) is the most widely used. BAB divides the entire problem into subsets and eliminates entire set of decisions. The problem is branched by considering two distinct values of a binary decision variable. The objective is bounded where the optimal objective is found by relaxing all other integer variables, and comparing optimal bounds eliminates entire sets of decision variables.

The field of multivariable search techniques is the most dynamic area of growth in optimization research. Linear problems are solved efficiently with Simplex, but nonlinear research is more dynamic. It typically involves beginning at an initial condition, improving the objective, and ending at some point. The middle step varies the most. For unconstrained multivariate optimization, Quasi-Newton method chooses the next point to solve as a function of the partial derivate at the current point and a step size. The Quadratic technique fits the function to a parabola at the current points and selects the next point as the first derivative of the quadratic function. For constrained multivariate problems, Successive linear programming linearizes the problem, performs the simplex algorithm, and then iterates. Successive Quadratic fits the problem to a quadratic form, the Lagrangian technique is performed, and then the process is iterated.

3.1.2 Stochastic Optimization

Stochastic Global Optimization (SGO) is a method of multivariable search techniques for nonlinear optimization which places experiments (values of D.V.s) randomly in the feasible region (Pike, 1986). The algorithm makes no assumption of the differentiability of the objective function as opposed to many analytical methods. (Luus & Jaakola, 1973) first proposed a direct search method in which randomly chosen points within a range are simultaneously evaluated, the constraints are checked to be proper, and the objective is calculated for the values meeting the constraints. (Gaines & Gaddy, 1976) demonstrated the effectiveness of Random Search Algorithm on chemical process optimization that he noted as flow sheet optimization. (Heuckroth & Gaddy, 1976) gave an examination of the
efficiency of the Random Search technique previously solved by other techniques. It proved to be robust although computationally burdensome. (Goulcher & Casares Long, 1978) used uniform distributions centered around the current optimum point to select new points to evaluate. The importance of reducing the range of the variables searched after each of the iterations was illustrated by (Spaans & Luus, 1992). An effective way of choosing region size which the random points are chosen to improve convergence rate was presented by (Luus, 1998). Increasing the efficiency of the optimization procedure and making the LJ optimization procedure viable for high-dimensionality problems using a multi-pass procedure was shown by (Luus, Hartig, & Keil, 1995). This technique involves morphing the optimization problem so that the equality constraints are absorbed by the objective function. (Salcedo, Goncalves, & Feyo de Azevedo, 1990) demonstrated the effectiveness of an improved Random Search Algorithm that did not allow back moves in 24 well known test functions. This algorithm also allowed the search region updating to be specific to the performance of the algorithms current performance, helping the algorithm to escape and to be less sensitive to local optima. Limitations of the Random Search are the troubles with the initial search region and the ability to escape local optima. Random Search fits into a more recent and growing class of stochastic global optimization techniques being coined as metaheuristics.

Metaheuristics is an emerging field of SGO. Metaheuristic methods create a class of SGO techniques which has been successfully applied to many chemical processes and found to be superior (Rangaiah, 2010). Metaheuristics are solution methods of optimization problems that orchestrate an interaction between local improvement procedures and higher level strategies to create a process of escaping local optima and performing robust searches of solution spaces. Hybrid metaheuristics take advantage of the strengths of their individual metaheuristic components to better explore solution spaces. On the front of applications, metaheuristics are used to find high-quality solutions to an ever-growing number of complex, ill-defined real world problems. Metaheuristic applications have greatly improved in the past two decades due to: 1) progress in mathematical theory and design, 2) rapid improvement in computer performances, 3) better communication of new ideas and integration in widely used complex software.

### 3.2 VNS/TA Metaheuristic with Repair Strategy

Here the formulation of a hybrid metaheuristic approach is proposed. Convergence issues arising in Stochastic Global Optimization (SGO), such as model failing, are discussed. A strategy to improve stochastic optimization of simulations robustness by integrating with the repair strategy is given. The repair strategy is then integrated into the VNS/TA metaheuristic.

#### 3.2.1 VNS/TA Hybrid Metaheuristic

Threshold accepting (TA) is a metaheuristic derived from simulated annealing (SA). Simulated annealing is an algorithm with a key feature of allowing objective worsening moves in hopes to escape local optima. Entire books have been devoted to its origins and applications (Aarts & Korst, 1989; Aarts & Lenstra, 1997). The original criterion for allowing moves that worsen the objective function is probabilistic and analogous to a process of physical annealing in which a crystalline solid is heated and allowed to cool.
slowly until it achieves its most regular possible crystal lattice, minimizing free energy. Threshold accepting challenges the need for a probabilistic criterion and simply uses a constant value for allowing moves which worsen the objective function (Dueck & Scheuer, 1990; Moscato, 1990). TA is a computationally efficient version of SA. Methods such as threshold accepting can search over larger solution spaces due to their ability to escape local optima, which increases our chances of returning non-convergent solutions.

Variable Neighborhood Search (VNS) was proposed by (Mladenovic & Hasen, 1997). The basic idea of Variable Neighborhood Search is a systematic change of neighborhood both within a descent phase to find a local optima and a perturbation phase to get out of the local optima. It has proven to be successful in generating good feasible solutions to continuous non-linear programs. VNS is based upon three facts: 1) a local minima with respect to one neighborhood structure is not necessarily one for another; 2) a global minima is a local minimum with respect to all possible neighborhood structures; 3) for many problems, local minima with respect to one or several neighborhoods are relatively close to each other.

Intelligently choosing the neighborhoods searched can allow the solver to search smaller spaces, reducing the non-converged model runs from poor initial conditions and avoids combinations that violate constraints and therefore are infeasible. The optimizer is separate than the model so cannot explicitly consider the constraints. To do so, the large solution space comprised of the DV’s physical ranges, which the optimizer considers, is narrowed by searching smaller sub-regions denoted as the neighborhoods of the VNS. To search smaller subsections of the space for a complex objective function threshold accepting’s ability to escape local optima is utilized. After TA exploits a neighborhood, new subsections or neighborhoods are chosen to search in line with the reasoning of VNS in order to explore the domain. In order to escape local optima, the search region is expanded after iterations where the same optimum is found twice. Once the same optimum is found for sufficiently large neighborhood, the algorithm ends. In the context of random search, objective worsening moves are allowed in the first step, vary the neighborhood structure which experiments are performed within, and have a neighborhood count as a termination criterion.

Figure 5 illustrates the layout of the algorithms block diagram. The parameters of the algorithm are the number of runs to limit CPU time (maxruns), iterations per a run (itr/run), threshold accepting value (TA), window size (size of the neighborhood searched), number of functions evaluated for repair purposes (nfez), max number of searches for the neighborhood (rmax) and initial point for searching (xinit). The algorithm stores important quantities in variables such as the best point found in the algorithm (xbest), the optimal point found in a run (xopt), the next point to be evaluated (xnew), and the current point being searched (xcurr), the previous loop’s best value (xprevbest), along with some point’s associated objective value (objbest, objopt, objnew, and objcur respectively). Algorithm counters and performance criterion (trackers) are also stored. In our formulation, they are: the total functional evaluations (nfe), the exploration loop counter (i), the exploitation loop counter (k), and the number of times the current neighborhood has been searched (r).
Initially in the algorithm, zero values are given to the xbest point, objbest value, and objcur value and set counters nfe and i. Xprevbest is given a value of one simply to ensure it is different than xbest for the first outer loop. Initially and after each explorative run, the index i, is increased by one to indicate the run number. If i is not greater than maxruns, then a run is performed by first checking if the xbest is the same as xprevbest (note they are initially set to be different).

Neighborhoods are constructed around points. For the first run, a neighborhood around xinit is created. For each successive explorative loop, xbest and xprevbest are compared. When xbest is the same point as xprevbest, then a different neighborhood is constructed around the point xbest. If rmax neighborhoods around the same point are searched, the algorithm ends in accordance with VNS presumption that the global optimum is the local optima for all neighborhoods around the point (rmax has been reached commonly denoted as neighborhood decent). If the maximum number of times to search around a point has not been reached, a new neighborhood around xbest is created. The values of set Z, which is used to repair non-converged models, are determined each time the point at which a neighborhood is centered about is changed. Typically predicting values for the set Z involves running the model and therefore each time a new set Z is created, the model is evaluated according to the nfez specified.

Once a neighborhood is constructed and the repair strategy initialized with set Z, xprevbest and xcur are stored as xbest and xopt respectively. This neighborhood is then searched using TA. As long as the number of iterations k is less than the maximum number specified, itr\text{run}, the algorithm finds a new neighboring point xnew of xcur. Xnew is sent from the optimizer to the model to be solved and to calculate the objective value. If the model fails to converge, the repair strategy is called. Otherwise, the objective
function value from the model is evaluated and compared to the current best objective function related to the $x_{\text{best}}$. If the point is within the TA tolerance, it is selected as our current point and objective point. Otherwise, the algorithm skips this step and goes directly to the next iteration increasing $k$ by one.

If $k$ is less than the $itr/\text{run}$ allowed, the algorithm will continue to create and evaluate new points in the neighborhood of $x_{\text{cur}}$. When $k$ is equal to the $itr/\text{run}$, the algorithm will evaluate the runs performance by comparing the $\text{objopt}$ of the TA run to the $\text{objbest}$ found so far. If the $\text{objopt}$ is better than previous $\text{objbest}$, then $x_{\text{best}}$ and $\text{objbest}$ will updated to be $x_{\text{opt}}$ and $\text{objopt}$ respectively. After each run, the algorithm starts over from the best $x$ it can find and continues on to the next run until all runs are satisfied or a specified number of neighborhoods have been searched around the same point.

### 3.2.2 Stochastic Optimization Integration with Repair Strategy

Simulations are increasingly important in the field of optimization as software becomes more specialized at accurately describing particular processes. Engineers assigned to particular units are the most adept at modeling their units and can choose from a variety of software to do so. In a competitive and evolving modeling software landscape, software can have superior benefits in different areas. For example, Software A may have a better feed characterization to aid in modeling a complex unit, where Software B is designed to construct models of large MILPs encountered in scheduling problems. This makes optimization of simulations better suitable for an environment where multiple software types are specialized and communicate with one another. Tasks are specialized as well. The unit engineers can update simulations to incorporate process changes and long-term dynamics using the same software configuration.

Stochastic optimization is suitable for a real-world environment where modeling and optimization tasks are specialized and multiple software types are integrated. When a process simulator is used as part of the optimization strategy, as in our case, the optimizer is a separate entity that treats the model as a black box. The optimizer sends a set of decision variables to the model and the model returns values necessary to solve the objective function. For optimization problems to be solved analytically constraints must be included in the objective function and an associated set of nonlinear equations solved. In many real-world situations, the modeling equations are embedded in simulation software and cannot easily be extracted. Therefore, stochastic optimization is selected for problems in which the modeling constraints are distributed among software types and are not expressed explicitly.

Stochastic optimization can exacerbate simulation convergence issues. A converged solution is a solution in which all equations in the model are satisfied. A set of decision variables generated by an optimizer can violate a constraint within the model when chosen stochastically since the optimizer does not consider modeling constraints. If the values of a set of decision variables are outside the convergence space, the model’s solver will not converge. Also, the solver may not converge if the optimizer makes large jumps in the solution space. For a nonlinear set of equations to be solved, modeling equations are typically solved through an iterative procedure. Despite advances of non-linear
equation solvers, the solver’s initial condition strongly affects convergence. Typically, the modeling solver’s initial condition is the value of the previous point solved. Therefore, increasing the solution space can help find global optima; however, taking larger steps increases occurrences of non-converging solutions by worsening the solver’s initial condition.

In the case of optimization, the variables of interest, set $X$, discussed in Section 2.2 must include the set of decision variables. The remaining DOF are chosen to meet constraints and ease of convergence criterion. The different sets of active specifications $Y$ which are easier to solve than $X$ may not be composed of the original decision variables of the optimization problem. These sets can be used to bring the non-converged points in $X$ back to the solution space at the highest objective function value found during active specification switching. However when evaluating a point to use as new I.C., we may serendipitously find that it has a higher objective function.

Figure 6 illustrates an integration of a general stochastic optimization procedure and the repair strategy for optimizing a nonlinear set of equality constraints. The method begins by defining an inner and outer loop counter, $I$ and $R$. We generate new random points, $X_{\text{new}}$, similar to the metaheuristic strategy and then send them to the model to solve for values necessary to calculate the objective function. The method differentiates from the other optimization methodologies where the models do not converge. Instead of ignoring the point or simply only repairing it and returning to the counter, the repair strategy is initiated. The easier-to-converge sets ($Y_i$’s) are evaluated, and if a set $Y_i$ has a better optimal value than the current optimal, then the solver immediately moves to the new point (serendipitous outcome). This outcome is the perturbation phase of the optimization algorithm. A new neighborhood is immediately chosen around this point.

![Figure 6: Stochastic Optimization of Distillation Utilizing Repair Strategy](image-url)
3.3 Stochastic Optimization of Distillation Utilizing Repair Strategy

Here, stochastic optimization of the refinery simulated in Chapter 2 is performed. Convergence issues arise during the population of values given to D.V.s so we employ the repair strategy to reduce its effect. The repair strategy is also intelligently integrated into an optimization procedure improving the procedure’s robustness. Many stochastic optimization methods would suggest ignoring non-converged simulation models as infeasible. The repair algorithm has the ability to (1) relocate the solver for non-converged models caused by poor initial conditions and (2) utilize points formulated in relocating non-converged models caused by infeasible sets of decision variables as a perturbation phase of a stochastic optimization algorithm. To show the effectiveness of the proposed repair strategy, the hybrid metaheuristic of a Variable Neighborhood Search (VNS) and Threshold Accepting (TA) discussed in Section 3.2.1 is tested on the optimization of a large-scale industrial process, the primary units of a crude oil refinery. The specification selection of the distillation model variables is illustrated. The performance of the hybrid VNS/TA Metaheuristic with repair is compared to the algorithm without the repair technique. Results indicate that the repair strategy is able to improve the convergence of the distillation models. Furthermore, embedding the strategy into the VNS/TA Algorithm improves VNS’s drawback of sensitivity to initial search region and allows for a larger range of solution space to be explored.

3.3.1 Introduction

Modern petroleum refining has become a competitive business due to the deteriorating quality of crude oil coupled with tighter product specifications and more stringent environmental regulations. Refineries today receive shipments of crude from a variety of sources. These crude oils are of different quality and composition and usually blending can improve the economics of the refinery. Refineries must deal with a dynamic schedule of incoming crude which cause them to frequently change unit operating conditions to reduce expenses.

In a previous publication (Robertson et al., 2011), the integration of the production-layer scheduling problem, the crude oil unloading scheduling problem, and the operational-layer process unit optimization problem of the main refinery units, the heat integration of the distillation units, was considered. Integrating tactical-layer decisions with the operational optimization is a dynamic optimization problem and was performed using modeling techniques. The need to create more robust optimization techniques of crude oil refining operations is apparent.

Distillation is an energy intensive process, and improvement can have substantial benefits. The heat integration of a distillation train is such an example of a problem that is difficult to formulate precise feasibility constraints. Simulations of distillation processes are used for optimizing the energy integration problem. A simulation of a distillation column solves the heat, mass, and thermodynamic equations at a particular set of active specifications. During the optimization algorithm, the active variables must include the
decision variables of the optimization problem such as pump around flow rates, steam flow rates, and constraints of the problem such as cut point temperature values. It is assumed in our problem that cut points are constant constraints. The robustness of temperature variables leads us to choose as many temperatures as possible for the active specs such as overhead, boiler, and PA return temperatures for rapid convergence and model robustness.

Distillation optimization problems have been studied in the past using analytical methods. In (Dennis & Schnabel, 1979), mathematical conditions such as null space constraints and secant condition of quasi-Newton methods are satisfied to perform distillation optimization. (Lucia & Kumar, 1988) demonstrate the space of secant matrices and those that satisfy constraints like homogeneity or the Gibbs-Duhem equation do not intersect so cannot be simultaneously satisfied. (Lucia & Kumar, 1988) show that it is more important to satisfy thermodynamic constraints than purely mathematical constraints. (Lucia & Xu, 1990) showed trust regions can avoid reliability issues common with line searching methods. Limitations of these analytical techniques include the assumption of well-defined objective function and constraints which are twice continuous differentiable. For analytical solution technics of optimization problems such as Karish-Kuhn-Tucker and Lagrangean methods, a specific optimization problem is defined and solved by creating and solving its unique associated set of nonlinear equations. Objective functions and models are becoming increasingly more complex and when using simulations as models, this is a difficult task.

Previous works of (Lucia & Kumar, 1988) demonstrate information about the physics (e.g., satisfying the Gibbs-Duhem equation) take precedence over purely mathematical conditions (e.g., satisfying secant condition of quasi-Newton methods). This result is in line with our reasoning that in some practical applications it is best to use specialized software and personnel to model the physics of the problem, and implement stochastic optimization algorithms on the most physically accurate simulations containing the embedded modelling equations. This is why stochastic optimization can work well with the distillation optimization.

A method to aid in convergence of nonlinear equations by relocating the solver is introduced. Initial Conditions are efficiently formulated by deactivating known values and activating intelligently-chosen unknown variables at predicted values. This repair strategy is integrated into an optimization procedure improving the procedure’s robustness. Many stochastic optimization methods would suggest ignoring non-converged simulation models as infeasible. The repair algorithm has the ability to (1) relocate the solver for non-converged models caused by poor initial conditions and (2) utilize points formulated in relocating non-converged models caused by infeasible sets of decision variables as a perturbation phase of a stochastic optimization algorithm. To show the effectiveness of the proposed repair strategy, a hybrid metaheuristic of a Variable Neighborhood Search (VNS) and Threshold Acceptance (TA) is tested on the optimization of a large-scale industrial process, the primary units of a crude oil refinery. The specification selection of the distillation model variables is illustrated. The performance of the hybrid VNS/TA Metaheuristic with repair is compared to the algorithm without the repair technique. Results indicate that the repair strategy is able to
improve the convergence of the distillation models by 6% in a case scenario. Furthermore, embedding the strategy into the VNS\TA Algorithm improves VNS’s drawback of sensitivity to initial search region and allows for a larger range of solution space to be explored.

### 3.3.2 Problem Motivation and Description

In a previous section, integration of the production layer scheduling problem, the crude oil unloading scheduling problem, and the operational layer process unit optimization problem of the main refinery units, the heat integration of the distillation units was considered. Integrating tactical-layer decisions with the operational optimization was performed using modeling techniques. The need to create more robust optimization techniques of crude oil refining operations is apparent.

Optimizing iteratively can exacerbate convergence issues. A converged solution is a solution in which all equations in the model are satisfied. A set of decision variables generated by an optimizer can violate a constraint within the model when chosen stochastically since the optimizer does not consider modeling constraints. If the values of a set of decision variables are outside the convergence space, the model’s solver will not converge. Also, the solver may not converge if the optimizer makes large jumps in the solution space. For a nonlinear set of equations to be solved, modeling equations are typically solved through an iterative procedure.

Despite advances of non-linear equation solvers, the solver’s initial condition strongly affects convergence. Typically, the modeling solver’s initial condition is the value of the previous point solved. Therefore, increasing the solution space can help find global optima; however, taking larger steps increases occurrences of non-converging solutions by worsening the solver’s initial condition.

### 3.3.3 Optimization Problem Description

On the operational level, personnel must maximize the total value of product from the Crude Distillation; less the environmental impact costs of the refinery units by manipulating the steam side stripper, pump-around, product stream flow rates, and unit operating conditions. These variables have a complex impact on the performance of the refinery units. The crude oil pre-heat furnaces and stripping steam production have a significant environmental impact due to generation of flue gases during fuel combustion process. The heat integration strategy recovers as much heat as possible from the distillation process; therefore, it recovers heat from final products and column pump-around streams. The product streams are valued differently and must meet certain specifications such as density and compositions, but side cuts of the distillation columns can be slightly manipulated to produce each amount within these specification ranges.

The complex heat integration schemes and the interactive nature of the process due to the presence of pump-around and side-stripper distillation features make it difficult to operate at the optimal conditions and consequently create a difficult optimization
problem. The huge capital expenditure involved in the refining operations creates good opportunities for optimization. It is estimated that crude oil cost account for about 85-90% of the total operating cost and; therefore, a wide variety of crude blends are processed depending on the cost and demand of the various products. This change in feed composition often results in inferior crude unit performance and reduces the unit’s run length. Therefore, the optimal conditions vary depending on the crude selected and optimizing the operation of the crude unit is essential to maximize a refiner’s economics. In addition, recent crude oil price fluctuations and increased economic pressure further emphasize the importance of optimizing crude unit performance.

The decision variables of the operational level optimization problem are the stripping steam mass flow rates, product flow rates, pump around flow rates, and overhead column flow rates. The constraints are the quality parameters such as the ASTM D86 95% temperature of the product flows detailed in Section 3 along with bounds on the decision variables.

Objective Function: The goal in the operational planning level is to maximize operational revenue. Equation (1) is the typical profit function, \( PF \), which is equal to the revenue from the products, \( RP \), minus the utility costs to produce them, \( UC \), and the raw material costs, \( RM \). In refineries, the energy demand is met by burning side products. Due to rising global warming concerns and with implementation of emissions trading programs (“cap and trade”), the triple bottom line (TBL) objective function given in the equation below was used in our approach. The objective function used accounts for costs associated with the feed, products, utilities, and environmental effects.

\[
PF = RP - UC - RM \tag{37}
\]

\[
TBL = PF - SD - EC - SC \tag{38}
\]

In Eq. (2), \( EC \) is the cost required to comply with environmental regulations such as permits, monitoring emissions, fines, etc. \( SC \) represents the sustainable credits given to the processes that consume pollutants. \( SD \) represents the sustainable debits that penalize processes for producing pollutants (poll). In this study, sulphur dioxide (SO\(_2\)), carbon dioxide (CO\(_2\)), and nitrogen oxides (NO\(_x\)) are chosen as the environmental load. The plant requires electricity for the condensers, steam for stripping, and heat for elevating feeds generates releases to the environment that are considered substantial debits. A portion of the net energy required is obtained by using the overhead gas of the PDU as the fuel in the furnace and the balance is met from fuel oil. From the environmental loads analysis, it is evident that the use of fuel gas in the furnace reduces the emissions to a greater extent but at the same time reduces the quantity of the useful product having a negative impact on the column economics (Yela et. al, 2008).

\[
UC = \sum_r FD_r * C_h + \sum_c CD_c * C_c + \sum_s Q_s * C_s \tag{39}
\]

\[
RP = \sum_p PP_p * Q_p \tag{40}
\]

\[
TEFO = (FHR+SHR) \eta^{ce} \tag{41}
\]
\[
FHR = \frac{(\sum FD_f - FGEF)}{\eta_f} \tag{42}
\]

\[
SHR = \frac{(\sum Q_s \cdot \text{SpecificHeat})}{\eta_b} \tag{43}
\]

\[
SD = \sum_{poll} \left( FGEF \ast R_{poll/duty} + TEFO \ast R_{poll/duty} \right) \ast \text{Penalty}_f \tag{44}
\]

The utility costs is calculated as the sum of the total furnace heating duties, \(FD_f\) over all furnaces \(f\) multiplied by the cost of heating, \(C_h\), the total condenser cooling duties, \(CD_c\), of all condensers \(c\) multiplied by the cooling costs, \(C_c\), and the steam flow rates, \(Q_s\), multiplied by the cost to produce steam, \(C_s\), the revenue of products, \(RP\), is calculated as the multiplication of unit prices of final products \((PP_p)\) and product flow rates \((Q_p)\), summed over all products \(p\).

Table 10, pollution ratios are given as pollution amount proportional to the electricity produced if the fuel had been used to create electricity in a combustion engine. The available correlations relate the amount of pollutants released by the fuel burned to the electricity generated in combustion engines.

**Table 10: Pollutions Ratios**

<table>
<thead>
<tr>
<th>Environmental Loads</th>
<th>Fuel oil</th>
<th>Fuel gas</th>
</tr>
</thead>
<tbody>
<tr>
<td>C02, Ton/GWH</td>
<td>657</td>
<td>439</td>
</tr>
<tr>
<td>SO2 Kg/GWH</td>
<td>1030</td>
<td>1</td>
</tr>
<tr>
<td>NOX Kg/GWH</td>
<td>988</td>
<td>1400</td>
</tr>
</tbody>
</table>

The theoretical electricity of fuel oil \((TEFO)\) is the electricity produced if the fuel oil used in the process is converted to electricity. Equation (41) calculates \(TEFO\) by multiplying the fuel oil total heat requirement by the combustion engine efficiency, \(\eta^{ce}\). The fuel oil heat requirement is equal to the sum of the furnace heat requirement, \(FHR\), and the heat necessary to produce the steam, \(SHR\) (Eq. 42). The \(FHR\) for the fuel oil is the total heater duty minus fuel gas enthalpy flow, \(FGEF\), divided by furnace efficiency, \(\eta_f\). Steam heat requirements, \(SHR\), is equal to the total mass flow rate of the steam streams, \(Q_s\), over all steam streams \(s\) multiplied by the steam’s specific heat divided by the boiler efficiency, \(\eta_b\), to produce the heat (Eq. 43). This method of calculating the pollution loads by finding a theoretical electricity production is due to the availability of pollution amount/energy produced ratio data \((R_{poll/duty})\) and can be replaced by simply inserting a ratio of pollution amount to fuel oil required where that information is needed. The sustainable debits, \(SD\), are then calculated as the pollutant amount multiplied by a pollution penalty, calculated in Eq. (44).

In the optimization problem considered in this experiment, temperature profiles are kept relatively constant. For example, feed temperatures to the column, condenser temperatures and top stage temperatures are kept constant. It can be stated that our problem is using the minimum energy required to achieve an approximate temperature profile; therefore, column temperatures are variables to aid in convergence. Since the
amount of heat given to the feeds is varied, the furnace duty is set to achieve the feed temperature. Although this is the optimization problem considered here, the tools and techniques discussed in this paper can be applied in other scenarios. Feed costs are ignored since it is assumed that the incoming oil has already been purchased and raw water necessary is proportional to feed costs. $SC$ is equal to zero as no processes consume pollutants. $EC$ are ignored because they are a function of feed flow rate and do not affect the optimization problem. A 100% conversion of the fuel gas enthalpy to electricity is assumed, but this accounts for less than 2% of the total pollutants. The bulk of the pollutants come from the theoretical electricity of fuel oil multiplied by the pollutant penalty per duty.

Assumptions in the objective calculation include: The sources of emissions are steam generated from utility boilers, furnaces flue gas, and electricity; the net equivalent electricity (fuel had been used to create electricity in a combustion engine) was determined. A heat to power ratio of 1.25 to account for fouling was used. The steam is $	ext{Mp steam}$ @ 245 °C with an enthalpy of 13.5 MMKJ/hr per ton of steam. Efficiency of the cogeneration plant is 75%.

The modeling equations are thermodynamic relationships, mass balances, and energy balances obtained using HYYS®. The optimization model is a NLP solved with a VBA based code for the Metahueristic. The solver uses an improved generalized gradient method capable of solving large-scale nonlinear problems used for predicting the values of the easier-to-converge set variables. A bridge code is programmed in Visual Basic Application (VBA). The bridge code allows the user to import and export any selected variables between the HYYS model and Excel worksheet. Interaction with HYYS uses object linking and embedding (OLE) automation.

The optimization process first takes the objects that are the variables of the triple bottom line objective function from the HYSYS library. The Visual Basic for Applications (VBA) bridge code then embeds them into an Excel spreadsheet where the metahueristic optimizer chooses the next set of variables to insert into the HYSYS model. In each of the iterations, the total cost of the refinery operations is embedded into the Excel spreadsheet.

**3.3.4 TA/VNS Hybrid Implementation**

Since the thermodynamic, energy, and mass balances of the column are embedded into the simulation. There are multiple nonlinearities, especially in the thermodynamic equations. Randomly selected points without considering the modelling equations are not guaranteed to be feasible. Therefore, distillation optimization with a stochastic algorithm can benefit greatly from an embedded repair strategy.

For the aforementioned benefits of metaheuristic algorithms stated in Section 2, stochastic optimization routine can work well with the distillation optimization. Optimizing a simulation of a process containing distillation units by stochastic algorithms
leads to issues of convergence. The thermodynamic, energy, and mass balances of the column are embedded into the simulation. There are multiple nonlinearities, especially in the thermodynamic equations. Randomly selecting new feasible points without considering the modelling equations is difficult. Therefore, distillation optimization with a stochastic algorithm can benefit greatly from an embedded repair strategy.

The specifications required are equal to the degrees of freedom. In our case, some of the active specs used are pump around flow rates, temperatures, and product flow rates. Easier-to-converge sets Y’s are created by constructing a set Z composed of variables which can be interchanged with the decision-variable-inclusive set X as active specifications. Manipulating the heat removed with the pump around flow rates at constant return temperatures can be infeasible by conflicting with the heat added by steam and furnace duties at the cut point constraints, causing the distillation columns in the model to not converge. However, there is a set of active specifications which will more easily converge, the product flow rates corresponding to the cuts will commonly cause the solver to return to a converging solution. This is because the product flow rates are stronger functions of the composition of the feed rather than the degree of separation or unit conditions. The product flow rates can then be used as new variables, Z, to create easier-to-converge sets Y. The process is depicted in Figure 7.

Figure 7: Refinery Repair Strategy Integrated with VNS/TA
The repair strategy exits in three basic ways. The first two are desirable and terminate the repair strategy immediately. If the model is solved at a different set of active specifications and upon switching back to the DV specification converges, then the repair strategy is terminated immediately (1). This is an example of when an appropriate initial condition was not given to the solver causing the model to not converge. Second, if the model converges on a new set of specifications with a higher objective function (or one within the current objectives $TA_i$ value) the repair strategy will update deactivated decision variable’s values and create a neighborhood around this new point. In this case we have serendipitously found a good solution to search around (2). In a situation where the algorithm evaluates an infeasible point and no set $Y$ converges, the algorithm refers to the optimization optimal solution (3).

It should be noted there is a chance to be stuck within an infinite loop in the rare case $x_{opt}$ does not converge. This is because there is a scenario that exists where the model can converge on an unstable solution that cannot be found again. This can be fixed by a number of methods. Replacing $x_{opt}$ if it does not converge, placing a counter on the number of times the solver try to return to $x_{opt}$, and increasing the threshold value are a few methods. Here, the threshold value was increased to a large number making any point found during the repair strategy selected. This corrected infinite loop problems in our case. However a chance of an infinite loop remains. If the new point does not converge, the solver cannot return to $x_{opt}$, and no point during the repair strategy converges, you will be stuck in an infinite loop. This is an extremely rare circumstance.

### 3.3.5 Model and Optimization Results

From the results in Section 2.3, these results indicate that 6% of potentially valuable points will be ignored by a typical stochastic optimization routine that ignores non-converged points. This represents 14% of all points that did not converge. Also, 22% of points can enter the convergence region at a higher objective value than the optimizer has thus far or these points were able to return to the convergence region at a relatively similar point.

Parameter tuning was performed to obtain the algorithm’s parameters including threshold-accepting value. At different window sizes, VNS|TA Metaheuristic was implemented with and without utilizing the repair strategy. A total of 15 runs were performed at each parameter setting. The objective value was used to measure the performance of the technique. The objective value’s standard deviation of the runs was used to measure consistency. Maximum objective value was used to measure ability. Table 11 contains the results of the parameter tuning process of the window size. Manipulating the window size had an interesting effect; the problem-specific tradeoff between computation of having to repair infeasibilities and better objectives from more methodical exploration of the solution space was demonstrated during tuning the window size.
Table 11: VNS/TA Optimization Results

<table>
<thead>
<tr>
<th></th>
<th>0.1 Repair</th>
<th>0.1 No Repair</th>
<th>0.15 Repair</th>
<th>0.15 No Repair</th>
<th>0.2 Repair</th>
<th>0.2 No Repair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Objective Value (k$/hr)</td>
<td>146.96</td>
<td>146.78</td>
<td>147.59</td>
<td>147.26</td>
<td>147.87</td>
<td>N/A</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>.46</td>
<td>.09</td>
<td>.28</td>
<td>.45</td>
<td>.7</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>147.59</td>
<td>146.87</td>
<td>148.29</td>
<td>148.29</td>
<td>149.06</td>
<td>N/A</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>146.29</td>
<td>146.69</td>
<td>147.32</td>
<td>146.67</td>
<td>147.22</td>
<td>N/A</td>
</tr>
<tr>
<td>Average Number of Function Evaluations</td>
<td>44</td>
<td>26</td>
<td>58</td>
<td>26</td>
<td>43</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Results indicate that the repair strategy improved the consistency and amplitude of SGO method. A t-test on the means of the two sample populations indicates that the objective values obtained with the VNS\|TA Metaheuristic with repair are statistically higher than the values obtained using the routine without repair. The data points of metaheuristic without repair had a lower average value at all three window sizes. At the .15 window size, the minimum value obtained with Repair (147.31) was higher than the average objective value obtained without repair (147.26). Additionally, results indicate that the smaller window sizes explore less solution space during optimization and more non-converged solutions are encountered as the larger window sizes are chosen due to poor initial conditions. One theory explaining the improved performance of the VNS\|TA metaheuristic with repair is that it is more useful to repair points that do not converge rather than ignoring them. These ignored points in the VNS\|TA metaheuristic without repair may be feasible points with higher objective values.

Another reason explaining the improved performance of the VNS\|TA metaheuristic with repair is that the repair strategy may help overcome getting caught in local optima. VNS\|TA metaheuristic with repair obtained a maximum objective value at the 0.1, 0.15, and 0.2 window sizes that was greater than or equal to the maximum objective value obtained by metaheuristic without repair. This could lead us to believe it consistently was searching a different neighborhood. Perhaps the repair strategy served a purpose other than aiding in convergence; it helped exiting local optima during the repair strategy by using the I.C.’s formulated to aid in convergence as points used to perturb the neighborhood. If they have an objective value within TA range, they will move the neighborhood thus increasing the area searched.

SGO techniques with repair are more robust than SGOs without repair. As the window size increased, the average and maximum objective values obtained by VNS\|TA metaheuristic with repair increased. This is because searching a larger portion of the solution space helps attain global optima; however, searching a larger space can worsen convergence issues. In a larger window size, the specific set of values given to the
decision variables are dispersed throughout a larger area. Since the I.C. of points being evaluated is the previous converged point, the I.C. in larger window sizes is typically farther away. By repairing infeasible points, the metaheuristic algorithm’s drawbacks due to poor initial search region are improved by giving the algorithm the ability to evaluate points outside the search region.

The largest window size, 0.2, demonstrates where the robustness is no more apparent. We have seen that as the window size increases the objective value becomes better with or without including the repair strategy. However, at the 0.2 window size the optimization routine cannot properly function on its own. This is because if we ignore non-converged simulations, these erroneous values are used as initial conditions for the next point being evaluated. Non-converged points are particularly poor initial conditions, decreasing the chances of the next point to converge. Eventually, the routine crashes. Conversely, the robust integrated approach was able to function at the 0.2 and find high objective values since these non-converges points are repaired and initial conditions for new evaluated points are within the convergence space.

Although the integrated approach proved to be more effective in producing higher objective and maximum values, it was more computationally expensive in comparison to metaheuristic without repair. As the window size increased, the benefit of the repair strategy became larger, growing from an average of 180$/hr difference, to 330 $/hr difference, until the largest window size where the model did not work without repair. With respect to window size, the computational load is flat. This leads one to believe that there may be a tradeoff between the repair strategy’s benefit of searching a larger region and drawback of computation. Although the repair strategy was computationally burdensome, the expense of these function evaluations was worth it at the 0.2 window size since the repair strategy was able to aid in convergence (149 k$).

3.4 OPERATIONAL OPTIMIZATION INTEGRATION WITH PRODUCTION SCHEDULING

This section demonstrates the results of the integrated supply chain problem formulated in Section 2.4. The model is the integrated crude oil loading and unloading scheduling problem and the production unit optimization. It gives the history of modeling the CSP and integrates the production optimization discussed in Section 3.3. The model developed in Section 2.2 is used as the optimization model and is altered slightly and a objective function is added.

3.4.1 Problem Description - Multilevel Modeling Decision Types

The longer the time horizon a decision considers, the less frequent the decision is made and the greater its ramifications. In Figure 8, we have broken down decision making into four layers (strategic, tactical, production, and operational.) Strategic level problems are long term, around 5 years, and can include building or expanding a facility and contract negotiations. Within the time period of the strategic layer problem, many medium term
tactical layer decisions are made such as resource allocation and transportation problems. Production layer problems include the scheduling of production units and blending problems. Operational layer problems are online problems including plant diagnosis, fault detection, and process unit optimization. Decomposition of problems both vertically and horizontally leads to a feasible but not necessarily an optimal solution.

![Decision Problem Types]

Figure 8: Decision Problem Types

Many resource planning problems have been addressed using managerial judgments. In these cases, complex interactions between the different decision making levels were often ignored. Recent developments in mixed integer optimization have made possible tools to help solve the problems of the operations hierarchy.

New environmental regulations and fluctuating prices and demands have increased the need for quantitative process planning techniques along the crude oil supply chains. Some features of the refinery operations process-planning problem are convergent multicomponent material flows. Companies can gain a competitive advantage in the creation of a schedule outlining their distribution, production, and inventory profiles. This optimal schedule can meet business objectives as well as production, logistical, and customer constraints.

The crude oil unloading loading scheduling problem (CSP) described in Section 2.4 was one of the first applications where the advances in mixed integer optimization were utilized. The traditional approach to the CSP for a refinery is a discrete time optimization
formulation where the scheduling horizon is split into time intervals of equal size and binary variables are used to indicate if an action starts or terminates in the beginning of the associated time interval (Saharidis, 2009). (Lee H. P., 1996) developed a MILP model for short term refinery scheduling with inventory management decisions. The model’s output considers the logistical costs of the crude oil loading and unloading schedule including inventory, changeover, and sea-waiting costs. (Lee H. P., 1996) demonstrated a tradeoff between sea waiting and inventory costs. This model was flexible in the consideration of different network topologies as it included combinations of storage and blending tanks. Including blending tanks improves the solution by having a means to account for feed-blend changes to the Crude Distillation Unit (CDU). To overcome the bilinear term which arises from mixing, individual key component flows are used as linear terms rather than crude oil stream types. Using the crude flow types as the linear terms of the equation has a benefit in that it has the features of being integrated with unit operational problem (Robertson, Gerailli, & Romagnoli, 2014; Chen, Grossman, & Zheng, A comparative study of continuous-time modelings for scheduling of crude oil operations) and the larger level transportation problem where crude oil types are purchased and delivered to refineries with certain demands. (Saharidis, 2009) used crude oil types as the linear terms within the model, but their model does not take into account sea-waiting, unloading, and inventory costs. Also, that model is not adaptable to all network topologies. The operational purpose of a tank is assigned as either storage or blending exclusively and there is no possibility for a combination of both.

(Pinto, 2000) and (Joly, 2002) also studied the refinery operations in detail. These studies focused on the planning and scheduling models for a refinery with only one type of tank which is used for both the blending and the storage of crude oil. (Wu, 2008) considered the short-term refinery-scheduling problem for crude oil operations and obtained a target-refining schedule from production planning as a constraint to create an executable schedule. The refinery system was modeled by a hybrid Petri net and the proposed model was based on an efficient heuristic algorithm. (Mendez, 2006) presented a novel MILP-based method to simultaneously optimize the off-line blending and the short-term scheduling problem in oil-refinery applications. (Abraham & Rao, 2009; Deventer, Moolman, & Aldrich, 1996; Downs & Vogel, 1993; Garcia & Gonzales, 2004; Gu, Zhao, & Wahg, 2004; Heikkinen, Poutiainen, Liukkonen, Heikkinen, & Hiltunen, 2011; Heskes, 2001; Himberg; Jamsa-Jounela, Vermasvuori, Enden, & Haavisto, 2003; Kaski, 1997) developed an optimization model for production scheduling in an oil refinery. In their study, for production scheduling operations of lube oil section in the refinery, a binary ILP model was introduced by integrating the operations of the plant. A continuous approach to the scheduling problem has been addressed where the binary variables that are used to indicate if an action starts or terminates are allowed at any point within the scheduling horizon (Jia, 2004). Global optimization for scheduling refinery crude oil operations was implemented by (Karuppiah, 2008). (Yuzgec, 2010) demonstrated the usefulness of scheduling solution approaches applied with a model predictive control strategy.
3.4.2 Novel Model

The objective functions of the various mathematical models of the CSP include cost incurred for waiting sea vessels, unloading cost, inventory cost, and setup costs. Yet the blend of the crude affects the optimized refining cost even when within the operational range allowed by the various models. The unit operation and crude scheduling problems are currently solved independently. Our approach will mathematically model and integrate different nodes along the petroleum refining supply chains. In this Section, we present the development, implementation, and testing of an integration strategy for optimization models of docks, tanks, and processing nodes along the petroleum supply chain. Lee’s model did use crude oil flow rate as the linear terms and therefore cannot be integrated with the production unit. Saharidis used crude oil types as the linear terms within the model, but his model does not take into account sea waiting and inventory costs. Also, tanks in his model are either storage or blending and there is not a possibility for a combination of both. The model proposed in section 2.4.3 takes into account the sea waiting costs, inventory costs, and can account for multiple tank types.

An integrated approach for refinery production scheduling and unit operation optimization problems is presented. Each problem is at a different decision making layer and has an independent objective function and model. The objective function at the operational level is an on-line maximization of the difference between the product revenue and the energy and environmental costs of the main refinery units. It is modeled as an NLP and is constrained by ranges on the unit’s operating condition as well as product quality constraints. Its description and optimization is described in Section 3.3.

The production scheduling layer is modeled as an MILP with the objective of minimizing the logistical costs of unloading the crude oil over a day-to-week time horizon. The objective function is a linear sum of the unloading, sea waiting, inventory, and setup costs. The nonlinear simulation model for the process units is used to find optimized refining costs and revenue for a blend of two crudes. Multiple linear regression of the individual crude oil flow rates within the crude oil percentage range allowed by the facility is then used to derive linear refining cost and revenue functions. Along with logistics costs, the refining costs or revenue are considered in the MILP scheduling objective function. Results show that this integrated approach can lead to a decrease of production and logistics costs or increased profit, provide a more intelligent crude schedule, and identify production level scheduling decisions which have a tradeoff benefit with the operational mode of the refinery.

The objective formulation must add cost as an output ti the model formulated in 2.4.3. The unloading costs can be calculated by multiplying the cost to unload by the time period the vessel is unloading.

\[
UnloadingCosts = C_u \sum_v (TVL_v - TVA_v)
\]  \hspace{1cm} (45)

Before the vessel unloads, there is a cost to unloading the vessel; this is the sea waiting costs:

\[
SeaWaitingCosts = C_{sw} \sum_v (TUIB_v - TVA_v)
\]  \hspace{1cm} (46)
Inventory Costs are the cost of keeping oil in the storage and blending tanks:

\[ \text{InventoryCosts} = C_{\text{invst}} \sum_{i} \sum_{t} \left( \frac{\sum_{j} V_{i,k,t} + \sum_{j} V_{i,k,t-1}}{2} \right) + C_{\text{invet}} \sum_{n} \sum_{t} \left( \frac{\sum_{j} V_{i,k,t} + \sum_{j} V_{i,k,t-1}}{2} \right) \]

(47)

Lastly we consider the changeover costs of establishing connections between entities along the supply chain.

\[ \text{SetupCosts} = C_{\text{setup}} \left( \sum_{n} \sum_{k} \sum_{t} BDCEBV_{n,k,t} + \sum_{n} \sum_{k} SBCEBV_{n,t} + \sum_{v} \sum_{i} VSCEBV_{v,i,t} \right) \]

(48)

The total logistical costs function is

\[ \text{LogisticalCosts} = \text{Unloading Costs} + \text{SeaWaitingCosts} + \text{InventoryCosts} + \text{SetupCosts} \ldots \]

(49)

The logistical cost is now an output of the scheduling model. To account for the refining financial consequence of the blend entering the crude distillation unit, the operational revenue and costs are tabulated for different blends within the range of feed blend. The revenue and costs tabulated have been optimized to produce the maximum operating profit according to the method in Section 3.1. After the optimized revenue and associated costs are tabulated, each is regressed with individual crude feed flow rates, \( Y_{z,k,j,t} \), and expressed as a linear function.

\[ \Delta \text{OperationCost} = \sum_{t} N_{P} \sum_{k} N_{CDU} \sum_{z} N_{T} \prod_{j} c_{j,k} Y_{z,k,j,t} - A \]

(50)

\[ \Delta \text{OperationRevenue} = \sum_{t} N_{P} \sum_{k} N_{CDU} \sum_{z} N_{T} \prod_{j} p_{j,k} Y_{z,k,j,t} - B. \]

(51)

The coefficients, \( c_{j,k} \), \( p_{j,k} \) in the objective equations are determined by multiple linear regressions around the range allowable for the production facility. Parameters \( A \) and \( B \) are determined to ensure the operating costs are of the same magnitude of order as the logistical costs. Therefore, these equations represent the financial difference across the blend range. It is only necessary to regress around the total number of crude flows minus one. This method is capable of expanding to any number of crude types. This refining cost function is embedded into the scheduling and planning MILP objective function. This leads to two possible improved objective functions:

\[ \max(\text{OperationalProfit} - \text{LogisticalCosts}) \text{ or } \min(\text{Total Costs}) \]

(52)

3.4.3 Results and Discussion

Variables determined in our case study are: flow rates from vessel to storage tank for each vessel, flow rates from storage tank to CDU’s manifold for each storage tank; crude
oil inventory levels in storage tanks for each time interval; series of crude oil blends to be fed to the CDU, and periods where connections (or setups) are established. First, the operational layer was optimized (see Table 12)

Table 12: Data from Operational Layer Optimization

<table>
<thead>
<tr>
<th>Type 0 Flow (m³/hr)</th>
<th>Type 1 Flow (m³/hr)</th>
<th>Vol. Fraction Dubai</th>
<th>Total Flow Rate (m³/hr)</th>
<th>Operational Costs ($/hr)</th>
<th>Operational Revenue ($/hr)</th>
<th>Operational Profit ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>298.10</td>
<td>467.1</td>
<td>0.3896</td>
<td>765.20</td>
<td>35651.6</td>
<td>169753</td>
<td>134101</td>
</tr>
<tr>
<td>292.40</td>
<td>472.8</td>
<td>0.3821</td>
<td>765.20</td>
<td>35688.5</td>
<td>169705</td>
<td>134016</td>
</tr>
<tr>
<td>286.60</td>
<td>478.6</td>
<td>0.3745</td>
<td>765.20</td>
<td>35730.1</td>
<td>169659</td>
<td>133929</td>
</tr>
<tr>
<td>280.90</td>
<td>484.4</td>
<td>0.367</td>
<td>765.30</td>
<td>35770.5</td>
<td>169611</td>
<td>133840</td>
</tr>
<tr>
<td>275.20</td>
<td>490.1</td>
<td>0.3596</td>
<td>765.30</td>
<td>35817.2</td>
<td>169558</td>
<td>133741</td>
</tr>
<tr>
<td>269.40</td>
<td>495.9</td>
<td>0.352</td>
<td>765.30</td>
<td>35858.1</td>
<td>169512</td>
<td>133654</td>
</tr>
<tr>
<td>263.70</td>
<td>501.7</td>
<td>0.3445</td>
<td>765.40</td>
<td>35909.5</td>
<td>169472</td>
<td>133562</td>
</tr>
<tr>
<td>258.00</td>
<td>507.1</td>
<td>0.3372</td>
<td>765.10</td>
<td>35950.3</td>
<td>169425</td>
<td>133475</td>
</tr>
<tr>
<td>252.20</td>
<td>513.2</td>
<td>0.3295</td>
<td>765.40</td>
<td>35997.9</td>
<td>169379</td>
<td>133381</td>
</tr>
</tbody>
</table>

The coefficients for the operational cost function were obtained using multiple linear regressions of columns 5 and 6 with column 1. Both revenue and cost are positively related to the flow of Type 0. The more expensive crude blend to refine cost gave less revenue. Therefore, there is an inexpensive profitable end of the blend range and a more expensive less profitable end of the blend range. The cost equations obtained from the regression of the above data around one product flow are given below. The approach can be extended for multiple crude types. The intercepts are readjusted (weighted) to the same magnitude of order of the logistical costs. Equations below represent the difference in operational cost and revenue.

\[
\Delta \text{OperationProfit} = 15.7 \ \frac{s}{m^3} \ Y_{Type0} - 350 \ k$\] \hspace{1cm} (53)

\[
\Delta \text{OperationalCosts} = -7.7 \ \frac{s}{m^3} \ Y_{Type0} + 400 \ k$\] \hspace{1cm} (54)

Figure 9 illustrates the residual plots (goodness of fit) for the operating profit function for the CDU feed composition (0.335-0.39 Type 0.) As we can appreciate, the linear approximation works quite well within the specified range. However, when the range is expanded more general approximations, such as piecewise linear approximation, may be needed, for better representation.

56
The optimized financial parameters remain moderately linear. As the amount of the lighter Type 0 crude is increased, the cuts are changed proportionally. This is due to the fact that the cuts are much stronger function of composition than the degree of separation. It is most evident in the Vacuum Residue and Light Naphtha profiles the most, while others remained rather stable. This causes the operational revenue function to be linear function of the feed composition. The costs remain more or less linear as well. The furnace duties decrease linearly as the lighter crude is added as the specific heat of the mixture is a reasonably linear function of composition. This causes an equally linear decrease of sustainable costs as the amount of fuel oil required decreases with the furnace duties. The condenser duties are proportional to the vapor load in the columns, which is proportional to the composition. Despite the increases in the vapor load of the PDU, the ADU and VDU condenser duties decrease. An overall decrease in the condensing costs occurs due to the fact that the condenser duties on the ADU and VDU are larger.

Table 13: Initial Conditions and Process Parameters of Base Case

<table>
<thead>
<tr>
<th>Initial Conditions/ Process Parameters</th>
<th>Base Case Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Horizon</td>
<td>110 hours (4.5 days)</td>
</tr>
<tr>
<td>Time interval</td>
<td>1 hour</td>
</tr>
<tr>
<td>Arrival and Departure times of Vessels</td>
<td>Vessel1: 3rd Time interval</td>
</tr>
<tr>
<td></td>
<td>Vessel2: 51st Time interval</td>
</tr>
<tr>
<td>Set up Cost (Includes more than just unloading cost but a</td>
<td>$25000</td>
</tr>
<tr>
<td>penalty for an erratic schedule)</td>
<td></td>
</tr>
<tr>
<td>Crude Type</td>
<td>Vessel 1: Type 0</td>
</tr>
<tr>
<td></td>
<td>Vessel 2: Type 1</td>
</tr>
<tr>
<td>Crude Amount in each Vessel</td>
<td>90,000 m³</td>
</tr>
<tr>
<td>Number of Docking Stations/Storage</td>
<td>1/4/1</td>
</tr>
<tr>
<td>Tanks/Distillation Units</td>
<td></td>
</tr>
<tr>
<td>Storage Tanks/ Initial Amounts/ Initial Types/ Capacities</td>
<td>T1/9.4 km³/Type 0/100 km³</td>
</tr>
<tr>
<td></td>
<td>T2/42 km³/Type 1/100 km³</td>
</tr>
<tr>
<td></td>
<td>T3/0 m³/no type/100 km³</td>
</tr>
</tbody>
</table>
In our case study, one dock is used to feed one refinery CDU with four tanks as intermediate storage for incoming crude and a blending manifold connecting the storage tanks and CDU. A list of initial conditions and process parameters is presented in Table.

The following assumptions can be made for the case considered. There is only one vessel docking station for unloading of crude oil, crude oil is unloaded in a predetermined schedule and takes 36 hours to unload (i.e., no sea waiting cost and constant unloading costs), the amount of crude oil remaining in the pipeline is neglected, changeover times are neglected due to having small values in comparison with the scheduling horizon. It is assumed that there is perfect mixing in the blending manifold. The inventory costs are equal for the crudes and will not play a role in the optimization, therefore, they are set to zero.

Scenarios exist where the total costs are higher when only the logistical costs are optimized. This scenario arises if a logistical event can affect the feed composition for a sufficient amount of time; the economic consequences of the unit operating costs will outweigh the logistical event’s penalty. Our base case is an example of a particular combination of the penalty for production level event outweighing operational cost or revenue difference across the blend range.

In the base case, the initial tank Type 0 inventory can last for as short as 31.5 hrs or as long as 36.6 hrs depending on which end of the blend range the distillation unit is operating (high Type 0 or low Type 0). The Type 1 can last as short as 82.5 hrs or as long as 90 hrs. The penalty for a set up establishment event on the production level is $25,000. The operation cost savings across the blend range is $324. By dividing the penalty of the production level event by the operational benefit across the blend range a minimum number of hours to justify a change in production cost for a change in operation cost is 77 hrs and for revenue profit difference to affect a changeover is 38 hrs.

The base case study demonstrates a scenario when it is beneficial to incur another logistical penalty of setting up a connection in order to save operational costs. Figure 10 is divided into three columns, each corresponding to a different objective function (Eq. 22, 26, and 27.) Each column contains graphs of the feed composition entering the CDU and inventory profiles of the four tanks. The first row is composed of the CDU compositions during the time horizon. The next two rows are inventories of Type 0 and Type 1 respectively. In all three cases, one of the inventory profiles of each crude type is always decreasing. This is the one feeding the CDU the corresponding crude type. During hours 3-38, a shipment of Type 0 is to be loaded. During hours 50-85 an incoming vessel of Type 1 is to be loaded. During these times, one of the inventory profiles increases to accommodate the storage of these loads.

The different objective functions have a different effect on the CDU composition. The logistical costs changed the composition between the extremes as needed to minimize the logistical setup costs. However, the logistical costs objective does not consider the operational costs, so the blend is indiscriminate as it does not affect the objective function. The objective functions of maximizing profit or minimizing total costs are inclined to keep the CDU feed composition on the high Type 0 end (the inexpensive –
profitable end) of the blend range. Table 14: Base case results of the scheduler using each objective function.

Table 14: Base Case Objective Values Results

<table>
<thead>
<tr>
<th>Objective</th>
<th>Logistical Cost (k$)</th>
<th>Total Costs (k$)</th>
<th>Profit (k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min (Log Costs)</td>
<td>150</td>
<td>320</td>
<td>119</td>
</tr>
<tr>
<td>Min (Total Costs)</td>
<td>150</td>
<td>312</td>
<td>135</td>
</tr>
<tr>
<td>Max Profit</td>
<td>175</td>
<td>327</td>
<td>156</td>
</tr>
</tbody>
</table>

When considering only the logistical cost (column one in Figure 10, Type 0 inventory in Tank 1 is completely consumed while the incoming Type 0 crude is stored in an empty tank (Tank 4). Accordingly, the CDU is fed on the low end of Type 0 (the expensive less profitable end of the blend range) to make Tank 1’s (initial Type 0) inventory stretch until the incoming crude is completely unloaded. The tank has more than enough to accomplish this and the blend feed is changed to the high end for time period 23-27. This change could have occurred at the end without changing the objective (during time periods 35-39). This would have reduced the number of times the blend was switched back and forth. After time 39, the CDU feed operates at the high end of the feed range while in the end (at time period 86) the CDU feed switches back to the low end for no apparent reason. This is a manifestation of the fact that blending changes are not considered when blending in a manifold occurs. This is due to blending changes being marked by connection establishment binary variables of the tanks to the CDU. The scheduler must note this shortcoming of the model and smooth it over himself/herself by ensuring blend changes are not occurring unnecessarily.

When considering the total costs, the second column, the optimizer tries to keep the CDU on the high end of Type 0 to save operational costs. This schedule does not include the unnecessary switch back to the low end at the end of the schedule. Before the 39th period, the CDU composition does spend the same time in the expensive region (periods 23-27). This is the minimum time before a connection would have been necessary to have enough Type 1 available to satisfy the CDU demand up to period 86. Once again this time could have been during 35-39, avoiding two operational mode changes. This is an example where operating at less than favorable conditions on the operational level can save total costs.

When the profit is considered, the optimizer places an additional connection to operate at the more profitable end of the blend range. The two empty tanks share the incoming Type 0 while Tank 1 is emptied quickly. The empty Tank 1 is then converted into a Type 1 Tank in order to store the incoming Type 1 crude. This is an example where adding an extra connection is necessary to keep the operational mode of the tank in the profitable region.
In this case study, a slight deviation of the base case is also explored (Figure 11). The initial crude inventory in Tank 2 is 40,200 m³. In this case, there is not enough Type 1 to both operate on the low Type 0 end initially to ensure that Tank 1’s (Type 0) initial inventory can last until the incoming crude is stored and Tank 2’s (Type 1) initial inventory can last until Type 1 is stored. In this case, three different schedules arise with the same logistical costs. Table 15 depicts the results of the scheduler using the different objective functions. The returned total cost and profit over the entire horizon are included in the table. The first row entries show the results when the scheduler minimizes only the logistical cost, which is dominated by the setup cost in our scenario (number of setups is tabulated), while still meeting the production level blend specification. The second row entries demonstrate the results when total logistical and refining costs are considered. In most cases, it was noted that a different total cost could arise when compared to min logistical cost refining cost even if the logistical costs were the same. This is true because, since the flow rates were decision variables to the CDU, they were allowed to move freely within the blend range without affecting the objective function if they did not cause a change in connections. There was more than one arrangement possible. In other words, there exists a solution space for the logistical objective function that was reduced when considering total costs.
Table 15: Base Case Deviation Optimization Results

<table>
<thead>
<tr>
<th>Objective</th>
<th>Logistical Cost (k$)</th>
<th>Total Costs (k$)</th>
<th>Profit (k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min (Log Costs)</td>
<td>175</td>
<td>356</td>
<td>97</td>
</tr>
<tr>
<td>Min (Total Costs)</td>
<td>175</td>
<td>327</td>
<td>156</td>
</tr>
<tr>
<td>Max Profit</td>
<td>175</td>
<td>327</td>
<td>156</td>
</tr>
</tbody>
</table>

This is an example where giving the blend range a preferable side shrinks the solution space of the objective function. In the case where the setup cost was minimized, the solver found no difference in the objective function to operate somewhere on either end of the blend range, leading to a more fluid schedule than necessary (randomness increases as schedule complexity increases). The optimizer does not consider blend changes within the CDU feed composition when minimizing setup costs and since the base case was selected to have preferable cost at either end of the blend range, large improvements can be expected when considering total cost or profit.

As shown in the objective table, minimizing total costs or maximizing total profit yields the same financial parameters, however, they have different schedules. In keeping Type 0 flow rate high to stay on the profitable inexpensive side of the blend range, the incoming Type 0 crude has to be shared among two tanks. Maximum profit formulation achieves this by using the two empty tanks, then converting Tank 1 into a Type 1 tank for the incoming Type 1. Minimum costs formulation achieves this by emptying the tank and then refilling it, which is a more desirable schedule as converting tank types requires a cleaning process. This scenario clearly shows that using different objectives could result in superior scheduling.

In Figure 11, the feed composition of the CDU was at either the high end or the low end despite the model allowing it move freely within the blend range. Due to the objective’s linearity, operating somewhere in the middle of the blend range can only be as good or worse than operating at one of the ends of the blend range. Therefore, the linear solver will only check against the constraints at either end of the blend range. From this perspective, it does not matter how well the linear approximation describes the middle of the blend range, only the ends of the constraints. This remains true for multiple crude types.
Since the solver operates against a constraint, the scheduler is advised to begin with the operating point he/she is at and choose a second operating point that he/she deems has potential. Process knowledgeable personnel could perceive potential in various ways. If inventory of a particular type is high and that crude is pooling, inventory costs can be easily be incorporated in our objective function. A potentially better operational point would consume more of the pooling crude. If inventory of a particular product is scarce, then there is a potential to cause many unwanted connections as in the base case. A second operational point would use less of this crude. If data or simulation shows a new feed blend could improve the unit’s operational costs at the expense of having to change the unit’s operational condition, the more optimal condition could be the next point of interest. In any case, the scheduler should set the ends of the blend ranges to describe the point he operating, and the other end to be in the direction of interest. This will allow the scheduler to evaluate the tradeoff potential of changing the unit’s operating condition.

Crude oil scheduling control is a natural extension of the work presented. The mathematical model does not account for all time constraints, personnel opinions, as well as other factors are present in a real world application of scheduling. Therefore, controlling the crude oil scheduling is necessary. Parameters which are used to control the schedule include the shifting factors (A, B), operational costs coefficients, (cj,k, pj,k), and penalties for production layer events such as penalty for setups. Based on the above
described strategy, a crude oil scheduling policy can be envisaged to take into account practical considerations.

- Time constraints could be violated when the schedule is formulated. For example, establishing a connection is a time consuming process including:
  - Configuring the pipeline network (e.g., opening valves, configuration of pumps, etc.)
  - Filling pipelines with crude oil is a lengthy and dangerous procedure,
  - Sampling crude for chemical analysis, measuring crude oil stock in tank before loading/unloading,
  - Starting loading/unloading,
  - Stopping the loading/unloading.

In the event that there are too many logistical events so that a time constraints is violated, one has to first check cost shifters. If the production layer is not on the same order of magnitude as the operational layer, then the operational layer may drown out the production layer events leading to a very erratic schedule. Increasing the setup costs will decrease the number of times establishing connections occur. If there is only one undesirable production layer event, increasing that event’s penalty reduces the frequency with which it occurs. In our base case, a penalty of $31,500 for establishing connections will prevent the optimizer from establishing another connection to maximize profit (i.e., the profit schedule will look like the min cost schedule).

If there are many unwanted production layer events, reducing the benefit across the operational blend range using the operational cost parameters can reduce the likelihood of a scenario where a production layer event can affect enough time intervals to outweigh the operational benefit. For example, the scheduler may tell the operator to change a unit from optimal feed conditions in order to save logistical costs of having that feed. The operator may be hesitant and resist. The operator can then either increase the financial consequence of changing to the less desirable feed or decrease the penalty incurred on the logistical level. If the scheduler still gets the same schedule, he/she can assert the operator change the operating conditions. If the scheduler finds that an insignificant change in the parameters leads to a schedule that pleases the operator, he/she may decide to allow the unit to remain at its current condition.

### 3.5 CONCLUSIONS

In Section 3.3, we first noted that some active specifications were more robust at solving a particular set of nonlinear equations than other, perhaps more useful, active specifications. When the easy-to-converge active specifications were not the ones of interest, we used easier-to-solve sets Y’s to formulate I.C.s to aid in the convergence of the desired set of active specifications. We then embedded this into an optimization routine where the interesting set consisted of decision variables and constraints of an optimization problem. Upon conclusion of testing, at various parameters, a stochastic algorithm embedded with the repair strategy and comparing it to the results of the algorithm that lacked the repair strategy, we discovered that the repair strategy enhanced
the search algorithm in many different ways. The repair strategy improved the robustness of the algorithm and expanded the solution space we could explore. We found an interesting result in that the points used to aid in convergence of the model helped escape local optima. Not only did this easier-to-solve set $Y$'s allow for a limitation of getting caught in local optima, but it also allowed for a higher average of objective value obtained. Improved robustness and an improved performance in obtaining higher objective values are two ways in which stochastic optimization benefitted from the addition of the repair algorithm.

In Section 3.4, we found that in all cases it is important to consider operational cost in the production layer scheduling. In most cases the linear operation cost function narrowed the solution space of the previous models to the economically beneficial side of the blend range. This led to a more intelligent schedule. A scenario was identified where the overall costs were reduced by increasing production layer costs. The operational costs function is able evaluate tradeoffs of production level events. Scenarios were identified where operating the unit at unfavorable conditions and implementing a schedule that did not minimize the local logistical costs could reduce the costs across the entire supply chain. We also found shortcomings in the model that could be overcome by trying different objective functions.
CHAPTER 4: DATA-DRIVEN MODELING FOR PROCESS MONITORING

In a highly integrated and automated process within a typical industrial environment, a Distributed Control System (DCS) mainly handles the normal operations and some of the disturbances and the operator acts as a supervisor. However, many complex abnormal situations and faulty conditions cannot be controlled by the DCS. In these conditions, the operator is required to rapidly assess the situation, determine the appropriate cause(s) and take necessary and appropriate corrective actions. Failure to act under these circumstances can have tremendous economic, safety and environmental impacts. Improving process monitoring can minimize downtown, increase safety, reduce manufacturing costs, and improve performance.

The abnormalities experienced in process plants consist of many types of malfunctions including unit failures, parameter drifts and process unit degradation. The complexity of the diagnostic tasks increases with the size of the plant. In this information technology (IT) era, the number of measured process variables in a plant can be high. In large industrial plants, sensor data for thousands of process variables are measured every second. Often measurements of these variables are incomplete, insufficient or unreliable. This arises from a number of reasons including sensor failure, noise or presence of outliers. It can easily be envisaged that under such conditions operators may take erroneous decisions and actions that can worsen the situation. The decisions taken by an operator may depend on additional factors including differences in plant operation between shifts and individual response behavior. Skills, mental condition, time available for the action to be implemented also affect an individual’s response. It is therefore important to develop methodologies and technologies to assist operators, process engineers and managers in handling abnormal situations.

This chapter is organized as follows. Section 4.1 first introduces process monitoring tasks, data-driven modeling techniques, and then gives a methodology for their offline development and online implementation. Section 4.2 details the data-driven modeling process monitoring based on principal component analysis (PCA) and self-organizing maps (SOM). The SOM is a novel approach for a data-driven topology preservation scheme. Furthermore, this section describes novel extensions of SOM proposed in this thesis expanding the capabilities of SOM to process monitoring tasks. Section 4.3 validates the tools developed within an industrial application for a pump-around system of an Exxon Mobile polymerization unit. Finally, Section 4.4 discusses the offline development and online implementation methodology to the industrial challenge problem of the Tennessee Eastman Process.

4.1 DATA-DRIVEN MODELING INTRODUCTION

Process monitoring tasks include fault detection, fault identification, fault diagnosis, and process recovery (Figure 12). Fault detection recognizes a deviation from the normal operating regime from process measurements. Fault identification can help personnel identify the fault by finding the measured variables most related to the fault. Fault
diagnosis determines the root causes of the fault and process recovery is the manual correction for the effect of the fault. Fault detection and identification can be automated processes. For fault diagnosis to be an automated procedure, we assume it is a known fault with available data. Process recovery typically involves the process supervisor’s intervention so the supervisor closes the loop.

![Figure 12: Process Monitoring Tasks](image)

The essence of process monitoring is to quantify the process with measures that are sensitive and robust to faults. Measures to aid in process monitoring have been divided into three categories: knowledge-based, analytical, and data-driven methods (Chiang et al. 2001). Analytical methods create a prediction of the process with a model often derived from first principles. Knowledge-based methods are mostly based on causality and expert systems. Most applications of these systems are for small inputs-outputs systems. For large-scale systems, these techniques require a detailed model or a large amount of knowledge. Data-driven methods are derived directly from process data without using any underlying laws. Process monitoring using data-driven methods to build class prediction models is a concept that continues to develop within the broad tapestry of control engineering.

Process Monitoring using data-driven methods aims to build a model that predicts the class of a particular data point. Pattern recognition is the statistical process of assigning an output to an input vector. In our case the output is the class of the observation vector. When data labels are not available to the data set, the prediction of categorical labels is unsupervised and involves clustering algorithm including k-means, hierarchical clustering, or kernel PCA. Pattern classification algorithms predict categorical labels in a supervised manner where the historical data is labelled. The process can be divided into 3 steps as shown in Figure 13; feature extraction, discriminant analysis and maximum selection (Chiang et al., 2001). The feature extraction vector is usually attained through a dimensionality reduction technique that aims at capturing the most important information at determining class and leaving useless information behind. The feature extraction vector is then compared to particular known class information using a discriminate measurement. The discriminate function is a measure of the degree of belongingness to a class. A fault is detected when the degree of belongingness to the normal class is low. Identification can be aided when the feature vector deviations are mapped back into the observed space. This can capture relationship abnormalities among variables. If a faulty class has a significant degree of belongingness, the fault can be automatically diagnosed.
Figure 13: Pattern Recognition Process

Mathematical tools used in the process can be categorized into feature extraction and discriminate analysis. Feature extraction (FE) techniques manipulate the data so that it can be more easily analyzed by reducing random univariate error (noise), reducing high dimensionally issues, and reorienting so that the interdependence of the variance is accentuated. This capturing of the multivariance makes the pattern recognition process more accurate.

The feature extraction category is composed of dimensionality reduction methods that can be categorized on some basic features. Distance preservation methods try to preserve the distance between each point in the upper and lower space where topological preservation try to preserve the order of points. Linear vs. nonlinear reduction categorizes techniques on whether they assume linear relationships between the input variables before reducing the dimensionality. Linear techniques typically perform a matrix decomposition step that assumes a linear relationship among the variables. Concepts using nonlinear feature extraction techniques have begun to form a sub-discipline that could add new interpretations or outperform linear techniques on particular applications.

Distance preservation techniques reduce the dimensionality of the data by attempting to map the data into a lower space where the pair-wise distance between points are preserved. Examples are PCA, Fisher Discriminant Analysis (FDA), Component Analysis (CCA), and Sammons Nonlinear Mapping (SNLM). Topological preservation techniques preserve the order of the data and the general feature of the data cloud rather than the distances between them. Examples include SOM, ISOTOP, and Local Lineal Embedding (LLE).

There are some feature extraction tools which need information about the classification of data points to create a map; this is known as supervised learning. Supervised learned mappings can do a better job at forming data clouds with features that correlate plant events or states than unsupervised feature extraction techniques. These techniques are discussed in more detail in Appendix A. There have been many successful applications of
data-driven techniques on industrial processes. In this chapter we will focus on a linear distance preservation technique (PCA), and a nonlinear topological preservation technique (SOM).

Once data has been labeled, discriminate functions give incoming points a degree of belongingness to the particular clusters. Thresholds are formulated to decide if a point has the enough of a belongingness to be classified with the cluster. Graphically this is drawn as a boundary. Discriminate functions fall into two main categories, parametric and nonparametric. Parametric functions assume a previously defined distribution such as $T^2$ and $Q$ statistics. Non-parametric functions cannot be generated by a set of parameters, such as density estimation. Parametric functions have the benefit of having a threshold that can be looked up from standard tables. Nonparametric functions’ thresholds must be manually tuned.

The data-driven techniques utilized in this work are introduced next. After a brief overview of the typical linear approach such as PCA, it continuous by introducing a topological preservation technique (SOM) including their visualization benefits.

4.1.1 Feature Extraction Methods

PCA is a linear distance-preservation technique which determines a set of orthogonal vectors which optimally capture the variability of the data in order of the variance explained in the loading vector directions (Chiang et al., 2001). Given a set of $n$ observations and $m$ process variables in the $n \times m$ matrix $X$ with covariance $S$, the loading vectors are determined from an eigenvalue decomposition of $S$:

$$S = \frac{1}{n-1} X^T X = VA^T$$

(55)

Where $A$ is the diagonal matrix containing the non-negative real eigenvalues of the covariance matrix in order of decreasing magnitude, and $V$ holds their corresponding eigenvectors. In order to reduce the misclassification rate, it is often desirable to remove directions that may contain little useful information or simple statistical noise. In PCA this is achieved by selecting the columns of the loading matrix which correspond to the $a$ largest eigenvalues $P \in R^{m \times a}$. The projects of the observation in $X$ into the lower dimensional space, also known as the scores, can be found from:

$$T = XP$$

(56)

In Section 4.2.1 the projections of the process variables into the score space will be later used to generate statics for process monitoring.

Self-organizing maps (SOMs), also known as Kohonen Network, are a type of neural network used to visualize complicated, high-dimensional data. SOM may be described formally as nonlinear, ordered, smooth mappings of high dimensional input data manifolds onto the elements of a regular, low dimensional array (Kohonen, 2001). It simultaneous performs vector quantization (data compression) and topographical preservation (dimensionality reduction).
PCA fits a hyperplane into a data cloud and points are encoded as coordinates in that hyperplane, in this regard, SOM can be interpreted as a nonlinear PCA. If the data cloud is curved, so should be that plane. SOM conforms to the shape of the data by replacing the hyperplane with a discrete representation consisting of connected nodes which are updated to warp to the data cloud’s shape.

A map is a graph composed of an array of nodes connected together in a rectangular or hexagonal grid. Each node, called prototype vector, models the observation vectors in the data space. These prototypes have a representation in the input (observation) and output (latent) space. The closest prototype vector to a data point in the observed space, known as the Best Matching Unit (BMU), can represent that data point on the map. The BMU of \( i \)th data point \( v_i \) is found by finding the closest prototype vector \( m_k \) according to:

\[
BMU_i = \arg\min_k \left( \partial (m_k, v_i) \right), \forall k
\]  

(57)

To implement the SOM procedure, first the map shape is selected. Then the prototypes vectors’ positions are initially embedded in the data space along the PCA hyperplane. Next, the map is trained to capture curves in the manifold (Figure 14). The prototype vectors’ positions are updated using each data point from a training set according to the formula:

\[
m_k(t + i) = m_k(t) + \alpha(t) h_{k,BMU_i}(v_i(t) - m_k(t))
\]  

(58)

Where \( t \) is the discrete time coordinate of the mapping steps and \( \alpha \) is the monotonically decreasing learning rate. The scalar \( h_{k,BMU_i} \) denotes a neighborhood kernel function centered at the BMU. Data vectors are matched to the prototype vectors with the smallest Euclidean distance from the data vector of interest. A neighborhood kernel \( h_{k,BMU_i} \) centered at \( m_k(t) \) is usually chosen in the Gaussian form:

\[
h_{m_k,BMU_i} = \exp\left( -\frac{||m_k - BMU_i||^2}{2\sigma^2} \right)
\]  

(59)

\( \sigma(t) \) denotes the monotonically decreasing width of the kernel that allows for a regular smoothing of the prototypes. The algorithm continues to make passes over the data updating the locations of the prototype vectors and terminates after some predefined number of time steps have passed or prototype updating becomes negligible.'

![Figure 14: Demonstration of SOM Learning (Stowell, D (2010))](image-url)
Figure 14 shows the node nearest to the training data point (yellow) is moved towards the training datum (white), as (to a lesser extent) are its neighbors on the grid. The learning algorithm runs through the data finding the BMU and updating all prototype vectors. Each time the algorithm completes a pass in the data, it is called an epoch. The algorithm terminates after some predefined number of epochs have passed or prototype updating becomes negligible.

A well trained SOM can be used as a model for data. As explored by Holmstrom, (1993) and Heskes, (2001), SOM can be used to create kernel density estimates from the data. The advantage of using a Gaussian mixture model (GMM) within the SOM for discrimination comes from its ability to estimate the Kernel Density Estimation (KDE) using a predefined number of kernels.

Data of different classes may vary in size. In order for class identification to be accurate, large classes must be prevented from dominating the space. Therefore, it is advantageous to compress large data sets to fewer points. Rather than eliminating points in some fashion, SOM captures density and topological properties of the data cloud. In our process monitoring framework, a point’s probability on a given map was used to evaluate its degree of belongingness to the various classes in the training data.

A 2-D SOM offers excellent visualization tools for data exploration. The approach to high dimensional data analysis with the SOM consists of a set of 2-D visualization displays. The set of visualization displays are constructed by projecting data onto the map. For instance, the displays allow identifying the shape of the data distribution, cluster borders, projection directions and possible dependencies between the variables. Common visualization discussed here are component places, distance matrices, and projections as proposed by Kaski, (1997), Vesanto, (1999, 2002) and Himberg, (2001).

To illustrate the SOM visualization, consider this toy example (Figure 15). It consists of four variables demonstrated here through a combination of scatter and histogram plots. It is also composed of two clusters. One can see that variables 3 and 4 define the clusters.

![Figure 15: Toy Data Set Scatter Plots and Histograms](image)
A distance matrix (Figure 16) visualizes on the SOM array the distances between each prototype vector and its closest neighbors. In a distance matrix, distances are encoded into gray levels or colors and each unit on the array is colored according to the distances between neighbors. The most widely used distance matrix for the SOM is the Unified Distance Matrix, U-matrix (Ultsch, 1993) where all pairwise distances are visualized. In a U-matrix, the dominant clustering structure can be seen as areas with small distances within and separated by large distances. As in the basic U-matrix, visualization of the clusters can be greatly improved by augmenting the distance matrix with an additional entry between each prototype vector and each of its neighbors. Alternatives to the basic U-matrix and other distance matrices are reported in the literature (Kraaijveld et al., 1995; Ultsch and Moutarde, 2005; Ultsch, 2005).

Figure 16: Distances Matrices for Toy Data Set

Figure 16 shows the map trained on the toy example and two distance matrices. The far left contains the output space, the two-dimension grid. The next plot demonstrates a grid with nodes representing the distances between each node, which are the arcs in the first figure, in the XY plane. The height is the distance’s value. It can easily be demonstrated that there are two clusters and the map represents them. The far right demonstrates the U-matrix. The blue colors represent distance that is very close together and the redder nodes represent distances that are very far apart. It can be interpreted that there are two clusters by the large distances in the prototype vectors.

A component plane displays the value of individual variables at each prototype vector in the map. Each component plane is associated to one variable in the observation space. The value of the variable at all map nodes is visualized using different colors which allows the user to visually identify possible dependencies between variables (Vesanto and Ahola, 1999; Lampinen and Kostiainen, 2000). The dependencies between variables can be seen as similar patterns in identical locations on the component planes. In that sense, the SOM reduces the effect of noise and outliers in the observations and, therefore, may actually make any existing dependency clearer than in the original data.

The component planes (Figure 17) are useful in identifying variables that distinguish clusters. Comparing component planes with the U matrix gives insight to the variables related to the clustering (operational modes). Comparing component planes to each other can give insight to variables relationships.
Each component plane is associated to one variable and there are as many planes as directions in the original data space (four variables in our example). The dependencies between variables can be seen as similar patterns in identical locations. Variables 3 and 4 distinguish the classes as can be seen from similar patterns as the U-matrix.

Finally, projections onto the map are very important visualization tools. Highlighting the BMU of data points represents them on the map. Two important figures derived from projections are hit diagrams and trajectories. Hit diagrams project a collection of data points to show a region belonging to a class. Trajectories take a series of time and project them on to the map, connecting them with lines. The trajectory makes it possible to visually indicate the current state of the process and observe how that state has been reached.

### 4.1.2 Offline Development and Online Implementation Methodology

This section explores a methodology to aid others in designing and implementing data-driven process-monitoring schemes. Process monitoring is broken down into two different series of tasks: offline development and online implementation.

The offline development process consists of the following major steps:

1. Gather process insight and data collection
2. Variable selection
3. Historical data labeling
4. Feature extraction and discriminate function evaluation

Gathering process data and insight is the intuitive first step. Data collection involves taking data from the historical database. General features to aid preprocessing include characterizing the distribution. Pre-processing steps include outlier removal, noise reduction, and corrupt or missing data removal. Data-driven methods cannot be fully exploited without process knowledge. Process observation, personnel interviewing, and expert analysis are helpful if not critical in many steps of development.
The second step, variable selection, is a critical step to describing the process and can be greatly enhanced with process insight. Variables are selected according to their proximity to the unit and temporal relations to events. Variables can be transformed from their original form to reduce input size, relationship complexity, eliminate dynamics. Variables showing irrelevant information can also distort the data analysis and method accuracy and should be removed. This step can be greatly enhanced with process insight. New variables can be created from original variables in a variety of ways. Time derivatives can help describe transition periods and using their values can help detect abnormal deviations from steady state. For batch processes, taking the differences of variables’ values at a particular point in the batch process between the batches is a useful technique. Variables showing irrelevant information can distort the data analysis and method accuracy.

Typically historical data is unlabeled. Supervised feature extraction techniques like MSOM require labeled data in their learning phase. Data-driven techniques can benefit from labeling historical data in two ways: 1) through the superior performance of supervised techniques that require class information and 2) labelled data can evaluate different monitoring techniques by comparing predicted classes with the labeled data. Typically, when the large amount of historical data is gathered, it is unlabeled. That is to say, events are not categorized. It is common practice to take all data and label it normal. The Figure 18 demonstrates the workflow for labeling historical data. Labeling historical data is a pattern recognition problem with unlabeled data so therefore involves a clustering algorithm. To label the data using SOM, first a map is trained to the data. The map prototype vectors are then clustered using an unsupervised method, in this work k-means. Then process knowledge is used to label a sample of points, which are then projected onto the map. The most common class of the points projected onto a cluster’s map units determines the cluster’s label. It is common for one class to be separated into multiple clusters. Finally, the rest of the historical data are labeled according to its BMU label. Ideally, the data clusters would correspond to important process events. If not, the engineer would return to step 2 and reevaluate the variable selection.

Figure 18: Historical Data Labelling (Step 3)
To fine-tune the variables selected (Figure 19), the clustered data is analyzed against known process events. If the clusters found in Step 3 do not correspond well to process events, the variables selected in Step 2 should be reevaluated. The Figure 19 demonstrates Steps 2-4 as a whole.

[Diagram]

Figure 19: Fine-tuning Step (Step 4)

Poor clustering due to variable selection could occur from a number of ways. Since clustering algorithms utilize a distance metric, not including the critical variables in differentiating operating modes can make it impossible to discern the operating regimes. Similarly, counting too many variables that contain the same information can make other information differentiating regimes seem inconsequential. Counting irrelevant variables, such as the Operator ID, can differentiate operations that are not useful clusters. Variables that allow the data to cluster well, also improve the data driven schemes performance.

The final step to designing a data driven process-monitoring scheme is to select the feature extraction and discriminant function methods (Figure 20). Feature extraction and discriminant functions are evaluated on their efficacy in classifying a previously labeled set of data. A priori knowledge is not used in the prediction by the diagnosis algorithm. For each technique, the predicted class of every data point is then compared to the known class and a correct classification rate is tabulated. Methods are then selected depending on the classification rates obtained. The best technique to detect a certain event may not be the best technique for detecting all events.

[Diagram]

Figure 20: Technique Selection (Step 5)
Online implementation must have several important features. It must have the ability to incorporate new faults as they occur into the methods used in the data driven model. It must also increase the integrity of the data used to train the feature extraction and discriminate algorithms because data labeled during the development process is susceptible to mislabeling and data that is old may no longer represent the process due to long-term dynamics. Immediately following development, there is initially a low confidence in the data labeling and the model trained with it, whereas data manually labeled by the engineer has the highest level of confidence. Data classified by fault detection system while online has a mid-level of accuracy as the engineer has the ability to manage this data in a more day-by-day approach. Updating training data-bases to keep confidently labeled data helps stay abreast of long-term process shifts and initial incorrect labeling.

Fault detection tries to detect when an operating point is abnormal and therefore should consider normal data. If a fault is not detected for an incoming observation, the point is labeled as normal and the normal database is updated. The incoming labeled point replaces the oldest lower confident point labeled during clustering techniques by the scheme in Figure 21. The fault detection model is then ready to be trained with this new set of data. This process outlined gives an adaptive feature to the monitoring scheme. If a fault is detected, fault diagnosis tasks are then employed.

![Figure 21: Adaptive Fault Detection](image)

Similar to fault detection, abnormal data bases are updated to give the adaptive feature to the fault diagnosis models, as shown in Figure 22. New points are evaluated using each discriminate function of each class. The fault diagnosis should automatically detect points from new operating regimes and points that are difficult to classify. Points difficult to classify may lie on the border of classes and are important to correctly identify. They are indicated by having similar degree of belongingness to multiple classes. Points that are new faults may have a low degree of belongingness to all classes. The engineer must classify these points manually, and can use information from fault identification tasks for guidance.
The engineer cannot be removed from the process when difficult to classify points and new faults must be identified (Figure 23). In these circumstances, tools are provided to aid the engineer in assessing the situation and determining the root cause. Fault identification tools aim at giving the engineer the variables most related to the fault. These tools are referred to as, sometimes deceivingly, contribution plots. The actual contributing factor may be unmeasured, where only variables affected by the fault would stand out.

4.2 DATA-DRIVEN METHODS IN PROCESS MONITORING

This explains how the methods described in Section 4.2 can be utilized in the process-monitoring tasks. One of the main advantages of PCA is its simplicity and a series of tools that can be used for implementation. Our intention is to basically develop a nonlinear approach that will mimic PCA by defining similar measures for process monitoring using SOM. Consequently, section 4.3.1 briefly overviews PCA tools and Section 4.3.2 expand these tools within the SOM framework.

4.2.1 PCA in Process Monitoring

Here, the application of PCA to the three process-monitoring tasks, fault detection, identification, and diagnosis, is explained. In addition to the techniques presented here, PCA has been applied to process monitoring other forms which more specifically take
into account dynamics and non-linearity. A comprehensive review of the application of these standard techniques to process monitoring can be found in Qin, (2012) and Venkatasubramanian, (2002).

Using the projection of the process data found in Section 2.1, we use the Hotelling’s $T^2$ statistic to detect faults for multivariate process data. Given the $1 \times a$ projection of a $1 \times m$ observation row vector into the score space $t$, the $T^2$ statistic can be calculated:

$$T^2 = t^T \Lambda_a^{-1} t$$ (61)

Where $\Lambda_a$ is the diagonal matrix containing the first $a$ largest eigenvalues in order of decreasing magnitude. The $T^2$ statistic threshold for normal operation can be represented as an ellipsoid with a $1 - \alpha$ statistical significance using the $T^2_{\alpha}$ threshold and a level:

$$t^T \Lambda_a^{-1} t \leq T^2_{\alpha}$$ (62)

When the actual covariance matrix is estimated from the sample covariance matrix, the $T^2_{\alpha}$ threshold is found from the $F$ distribution by the equation:

$$T^2_{\alpha} = \frac{a(n-1)(n+1)}{n(n-a)} F_{\alpha}(a, n - a)$$ (63)

Where $n$ is the number of data points and $a$ is the number of principal components. The $T^2$ statistic is highly sensitive to inaccuracies in the PCA space corresponding to the smaller eigenvalues (Chiang et al 2001) because it directly measures scores corresponding to the smaller singular values. The $m-a$ smallest principal components can be monitored using a supplement to fault detection based on the $T^2$ statistic, the Q statistic. Also known as the squared prediction error (SPE), the Q statistic can be computed to measure how well each sample conforms to the PCA model and the amount of variation not captured by the principal components retained in the model (Wise 1996). The part of the observation space corresponding to the $m-a$ smallest singular values, which correspond to the $m-a$ ignored by the reduction in dimension, can be monitored using the Q statistic:

$$r = (I - PP^T)x \quad Q = r^T r$$ (64) & (65)

Where, $r$ is the residual vector, a projection of observation $x$ into the residual space. The threshold for the Q statistic is given by the following equations from Wise and Gallagher, (1996).

$$Q_{\alpha} = \Theta_1 \left[ \frac{c_\alpha \sqrt{2 \theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + 1 \right]^{\frac{1}{h_0}}$$ (66)

where $c_\alpha$ is the standard normal deviate corresponding to the upper $(1-\alpha)$ percentile and

$$h_0 = 1 - \frac{2 \theta_1 \theta_3}{3 \theta_2^2}$$ (67)
and

\[ \Theta_i = \sum_{j=a+1}^{N} \lambda_j^i \quad \text{for} \quad i = 1 - 3 \quad (68) \]

where \( \lambda_i \) are the eigenvalues of the covariance matrix of \( X \) and \( N \) is the total number of principal components, which is either the number of variables or samples in \( X \), whichever is smaller. The thresholds defined for the \( T^2 \) and \( Q \) statistics set the boundaries for normal variations and random noise, and a violation of the threshold indicates that the random noise has changed (Chiang et al. 2001) and a statistically separate operating status reached.

Contribution plots are a PCA approach to fault identification based on determining the process variable responsible for the faulty condition. For a \( T^2 \) violation, follow the approach in Chiang et al. (2001) in which the contribution of each variable is variable is calculated according to:

\[ \text{cont}_{i,j} = \frac{t_i}{\lambda_i} p_{i,j} (x_j - \mu_j) \quad (69) \]

Where \( p_{i,j} \) is the \( i^{th} \) element of the loading matrix \( P \). The total contribution of the \( j^{th} \) process variable is determined from:

\[ \text{CONT}_j = \sum_{i=1}^{i'} \text{cont}_{i,j} \quad (70) \]

The \( \text{CONT}_j \) for all process variables are then plotted on a single graph in order to decide which variables are responsible for the faulty condition.

To determine the root cause of a given fault, data previously collected from out of control operations can be grouped and used by a pattern classification scheme to diagnose future faults. Pattern classification proceeds via three steps: feature extraction, discriminant analysis, and maximum selection. Feature extraction is the process of isolating the important trends in the data and removing noise which could disrupt classification. PCA performs feature extraction in the projection into the score space using \( P \) in the fault detection step.

The pattern classification system assigns an observation \( x \) to class \( i \) if

\[ g(x)_i > g(x)_j \quad \forall j \neq i \quad (71) \]

Where, \( g_j \) is the discriminant function for class \( j \). \( g(x)_i \) can be approximated using the \( -T^2 \) statistic assuming the probability of each class is the same and the total amount of variability in each class is the same.

PCA, and similar techniques that use matrix decomposition, assume linearity and preserve distances. Preserving topologies with non-linear techniques preserves the order of data, which may aid our classification goals. In the next section, SOM as a topological preservation method is discussed.
4.2.2 SOM in Process Monitoring

SOM has been applied for chemical process data analysis with some success. Deventer 1996 used SOM in tandem with textural analysis for monitoring of a mineral flotation process. Jamsa-Jounela utilizes SOM to detect several faults in a smelter and an online tool to assist in its implementation. Garcia et. al (2004) use SOM and k-means clustering for system state estimation and monitoring and their strategy to a wastewater treatment plant. Ng 1&2 2008 create an effective training strategy using SOM for multistate operations in addition to a sequence comparison algorithm with applications to a lab scale distillation unit, a refinery hydrocracker, and the Tennessee Eastman process. They also resample the training data in order to achieve an equal representation of the different operating regimes. Corona et. al (2010) applied SOM to the classification of different operating regimes of an industrial deethanizer and include a method to consider quality specifications. Throughout all these applications, the use of SOM’s visualization tools was the key advantage, and not its superior ability to detect, identify and diagnose the faults.

Figure 24 illustrates the overall approach followed. The approach first trains maps to the historical data. The maps then serve one of two general purposes, data analysis or points classification. Data analysis includes using visualizations, such as component planes, hit diagrams, as well as unsupervised clustered maps to identify variables important to clusters in the data. Data analysis is used to select variables and label clusters so that SOM maps can be trained and used for the classification of new data points.

![Figure 24: Schematic of Overall Strategy](image-url)
SOMs can serve a powerful role in variable selection. Since SOM extraction data features into two dimensions, component planes can help identify related variables. First, one trains a map with all possible variables (Figure 25). Then the map is clustered and the clusters and component planes are compared to one another. Variables that partition the data into different clusters will be very similar to the clustered-data map. Variables that correspond well to one another contain similar information. The variable selected list is initially populated by variables that divide the data well, with only one variables selected among variables containing similar information.

Figure 25: SOM for Variable Selection

SOM can also help in data labeling during the cluster analysis step. Experience with the process may be able to highlight a hand full of instances. Even though manually labeling all data is impossible, a hand full of points is possible. One is to observe samples of the cluster and another is to plot some known process events onto a hit histogram. Important process regions are identified by a hit diagrams due to the size of the node and density of nodes increasing.

For fault detection, a SOM is trained to data from normal operation of the process to create a non-parametric model proportional to the probability distribution of the data. SOM’s vector quantization abilities are used to provide a compressed representation for the normal operating region of the process in order to form the basis for a Gaussian mixture model.

For an SOM with M map nodes, the Gaussian mixture model is the weighted sum of M component Gaussian densities as given by the equation:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i),$$  \hspace{1cm} (72)

Where $x$ is a $D$-dimensional continuous-valued data vector (i.e. measurement or features), the $w_i$ are the mixture weights, and $g(x|\mu_i, \Sigma_i)$ are the component Gaussian densities. Each component density is a $D$-variate Gaussian function of the form,

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_i)'\Sigma_i^{-1}(x - \mu_i) \right\}$$  \hspace{1cm} (73)

With mean vector $\mu_i$ and covariance matrix $\Sigma_i$. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. The complete model is parameterized by the mean vectors, covariance,
and mixture weights from all component densities. The model is collectively represented in $\lambda$ (Reynolds, 2008):

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \ i = 1, \ldots, M$$  \hspace{1cm} (74)

Here, we use SOM map vectors as the centers of the Gaussian kernels instead of the data, a feature useful for very large sets of multivariate data. A Gaussian mixture model is formed by summing kernels centered on the map nodes along with a regularization term (Heskes, 2001). Here, the discriminate function used is:

$$g(x)_i = \log(p(x|\lambda))$$  \hspace{1cm} (75)

Therefore, a normal point will have a small positive value and a faulty point will have a larger positive value. The main advantage of a density estimation approach over parametric approaches lies in density estimation’s ability to describe irregular data shapes caused by nonlinearities, as opposed to the hyper ellipsoid shape assumed by the Hotelling’s $T^2$ statistic or linear assumptions of the Q statistic.

Projections of the probabilities on the map are included in Figure 26.

![Figure 26: Degree of Belongings Projected onto PCA](image)

Since nonparametric points do not assume a distribution, the thresholds for the normal region must be determined. Below this threshold, the process is considered at normal operation and above the bound the process is under abnormal operation. To set the threshold, values of the density estimation function were calculated for the normal and faulty training data sets. The threshold for normal operation is initialized to three standard deviations above average discriminant function on the training data. This initial threshold is then adjusted based on the type I or type II errors of the initial threshold. If the type I error rate is higher, the threshold would be decreased to capture faulty data with a lower probability. If the type II error rate is higher, the threshold is raised to reduce the number of false alarms. This optimization converges once the type 1 error $= 10$*type 2 error. Type 2 error is weighted heavily to avoid a large number of false faulty alarms.
The variables most responsible for the fault are determined by analyzing the residual of the faulty point and the SOM model of the normal region used in fault detection. The data points are projected on the model by locating the BMU of the point. The residual vector is the squared difference of the point and its BMU. For variable $j$, the residual is calculated by:

$$r_j(t) = (x_j(t) - m_{c_j})^2$$  \hspace{1cm} (76)

Figure 27 illustrates the effectiveness of analyzing the residuals for the SOM model for fault identification. Suppose a data set exists with related dimensions $x_1$ and $x_2$ and other independent variables. A detected faulty data point, shown as a large blue circle, inconsistent with the general trend of the data caused by an alteration in an unmeasured variable that affects $x_1$ and $x_2$, creating deviations in both directions $x_1$ and $x_2$ directions.

In Figure 27, three residuals are plotted using the black braces: the centroid, the nearest data point, and the BMU in the SOM. In this case, the centroid of the nonlinearly related data does not lie within the data cloud itself. Therefore, the deviations from the data point are inconsequential and do not register as indicated by a small residual. It should be noted that density estimation must be used to detect this point. Using raw uncompressed data causes the faulty point to be matched with a noisy outlier. Not only is the residual vector small, it would erroneously capture $x_1$ as the deviated variable which ignores $x_2$’s contribution. The residual is an approximation of an orthogonal projection onto the topological structure of the data cloud. The SOM residual accurately captures both $x_1$ and $x_2$’s contribution. This is a simple yet effective way of capturing variable contributions.

![Figure 27: Illustration of MSOM Fault Identification](image)

Data of different classes may vary in size. In order for class identification to be accurate, large classes must be prevented from dominating the space. Therefore, it is advantageous to compress large data sets to fewer points. Rather than eliminating points in some fashion, SOM captures density and topological properties of the data cloud.
In traditional applications of SOM in process monitoring, one map (1-SOM) represents the entire observed space by training with data from all observed states of the process. Fault detection and diagnosis are performed in one step: new observations are classified according to the label of their BMU. The discriminate function is the distance from a point to the map units. This has major drawbacks in practical applications. If the new observation belongs to a fault that has not previously occurred, 1-SOM will erroneously classify it to an existing condition. Additionally, some map vectors are located between multiple classes, particularly during the transition states of a process, increasing the chances for misleading classifications.

MSOM takes a different approach in that each operating regime has its own map expressing the topology of the data under those conditions, but poorly represents any data under other conditions. Multiple SOM maps have been previously applied to classification problems by (Wan 1999), who found it to perform better than standard SOM for an image analysis application. For our application, to determine how to classify a new data vector detected as faulty, the vector’s probability is evaluated for each of the Gaussian mixture models created from the SOM of each regime under consideration. Normal operation was also included to lower the chances of a false alarm. Once the probability of the faulty point occurring on each map is calculated, the maximum discriminant function assigns a point to an operating regime associated to a class:

$$g(x)_i > g(x)_j \quad \forall j \neq i$$

Where, $g(x)_i$ is the discriminant function for class $j$.

When practically implementing a data-driven process-monitoring scheme, engineers must balance a tradeoff between time to develop the model and the unknown potential benefits. Long periods of developing faulty databases can be an arduous task. We instead envision putting a limited data set consisting of only normal operating regime and include suboptimal incidents as they occur. This would involve the process adapting as new conditions are met. This allows less upfront time input. For single map representations, adding new process conditions affects the model for all data representations. MSOM allows the process monitoring scheme to identify new points, allow the process supervisor to classify them, and incorporate them into the data-driven scheme without affecting the representations of other operating regimes.

### 4.3 POLYMERIZATION UNIT APPLICATION

The objective of the Exxon/LSU research collaboration was to create a process-monitoring scheme for a nonlinear dynamic polymerization unit where previous techniques yielded unacceptable results. Previous techniques include Principle Component Analysis, which makes an assumption of linearity. The process-monitoring scheme should be efficient, easy to implement, understandable, and adaptive. This application gives a complete demonstration of the offline development steps of implementation. The steps to offline development are: data collection, process insight collection, variable selection, data labeling, and data-driven method selection. The application chooses single Self-Organizing Map as a feature extraction technique.
Section 4.3.1 gives a process description of the unit. Section 4.3.2 discusses the variable selection process. Section 4.3.3 demonstrates the technique selection process. Section 4.3.4 demonstrates some visualizations of the map capturing faults while implemented. 4.3.5 is a conclusion.

### 4.3.1 Process Description

The first step of developing the process-monitoring scheme is process to study the unit, observe its operation, and general features of its data. The thermal polymerization unit has the overall goal of converting monomer mixtures to an intermediate with desired product qualities. The different batches are distinguished according to their grade. Grades differ by being aromatic grades or cyclic grades. Monomers and other agents are first blended in feed tanks. The blended feed is then sent into two reactors that work in tandem to produce the product. The blended feed is mixed with a pump around stream from the reactor. The mixture is then heated by hot oil in a heat exchanger.

Each reactor has six main states of operation (Figure 28): Process Wait, Charge, Heat-up, React, Transfer, and Stand-by. The reactor runs through phases by first charging a specified amount of feed which drops the temperature. Then, the contents are heated to a desired temperature followed by the reaction occurring for a set amount of time. Finally, a transfer period occurs which discharges the reaction product to a stripper before the cycle repeats itself. The reactor will occasionally initialize and has the ability to be put in the Stand-by phase when not in use.

![Polymerization Phase Diagram](image)

During observational period, abnormal operations in the pump around section were observed. Figure 29 gives the properties of the faults. An abnormal event occurs at data point 2000. The (pump-around) PA flow rate, shown in blue, is operating abnormally for 6 batches. It is clear the problem is related to pump around flow valve operation shown in cyan. Personnel correct this occurrence by putting the valve in manual and “stroking” the valve, opening and closing. A sticking or clogged valve causes this type of problem.
Figure 29: Abnormal Operation

Currently, these problems persist for 5-20 batches. This leads to a reduced number of non-ideal performing batches. Each batch that under performs causes reduced mixing during batch, an extended heat up period, and safety concerns. Many of the suboptimal operation can affect product quality such as a softening point. Lowering the product softening point reduces the yield of the desired product. The extended heat up period is small for a single batch, but the accumulation over time is significant. It reduces the total number of batches made per year. The reduced yield not only has the associated opportunity costs of making product, but the unreacted product must be stripped in a finishing plant downstream. The energy cost of stripping the final product is increased.

Figure 30 contains a sample of the pump-around related variables demonstrate the short-term and long-term dynamics. The plot on the left consists of the phase number of the reactor, bottom reactor temperature, flow rate in the pump around, pressure drop across a filter, and temperature entering the heat exchanger. The selection of variables in the preheat section illustrates the point that these variables are highly correlated to the phase number, shown in cyan on the left plot. The plot on the right demonstrates the pump around flow over several years. One can see the flow rate goes through phases where it slowly deteriorates as the pump fails, and the spikes upon its replacement. These short-term and long-term dynamics make characterizing the process difficult.
4.3.2 Variable Selection and Historical Data Labeling

After the process is studied, abnormal operations are identified, and basic variables features are determined, the variable selection process begins. Variable selection tries to detect the type of faults observed. First, observe the variables that are close in proximity to the fault. Starting with the reactor and working around the unit, these variables included are the reactor temperature, reactor level, bottom valve position of the reactor, pressure drop across the filter, valve before splitter, temperature before heat exchanger, temperature exiting heat exchanger, the heat exchanger inlet temperature, the heat exchanger outlet temperature, and the heat exchanger flow rate. Included in variables selection is identifying variable transformations which may be useful in better describing the process. The general workflow is to perform a variable transformation, plot a histogram, and analyze. It is challenging to detect variables closely related to reactor phase being out of normal operating regimes as they vary over a large range during the batch phases. This is demonstrated in the Figure 31 on the left where a month’s worth of data is plotted.
Our aim is to use variable transformations to overcome the short-term dynamics. Since the variable values at a particular time are strongly dependent on the phase number, an intuitive transformation is to sample points at the phase transitions rather than over the entire run. The histogram above on the right shows the reactor temperature at the transition from charge to heat up phases. Despite many variables being in transition at this time rather than quasi-steady state, a normal distribution is formed. This variable transformation has reduced the short-term effects and helped characterize the process state. An important consideration to note is this transformation can be put online.

Even though the short-term dynamics we removed for the sample of data above, the distribution of the transitional plot shifts due to long-term dynamics. The inspiration of the variable transformation to overcome long-term dynamics is from the ease of noticing the fault in Figure 29 demonstrating the abnormal operation. The fault is easily identified because the observer can compare to the previous batch. It is therefore likely that taking the differences between variables in the current batch and last batch is a transformation robust against long-term faults: Relevant Variable = Variable Current Batch - Variable Previous Batch. This is illustrated in Figure 32.

![Figure 32: Overcoming Long-term Dynamics, Pump Around Flow Drop Over 2 Years](image)

In Figure 32, the data is colored according to the upper 75% over data is represented in blue while the lower 25% is represented in red. It is known that the flow rate should drop about every two days, as the filter is prone to being clogged when solids enter the pump around system. However, the lower 25% of data is not evenly dispersed. It persists for long periods and the filter-clogging event cannot be detected. After the variable transformation on the right, the lower 25% of the data is evenly dispersed and more capable of detecting a problematic pump-around system.

For the initial selection of state variables with close proximity to the PA section, the transition from heat up to heat-up to reaction phase is taken along with their change from the previous batch. The historical data-labeling phase then ensues. After selecting state variables and the differences with the last batch, an unsupervised map is trained and clustered in Figure 33 below. The unsupervised map clustered is on the left. These clusters are analyzed and do not corresponds well to process events. Some manually labeled faults are projected onto the map using a hit diagram on the right. For this type of hit diagram, a hexagonal shape is inserted over the map node where its size is a function
of the number of hits it received. These clusters are analyzed and do not correspond well to process events.

Figure 33: Unsupervised Clustered Map and Hit Diagram of Faulty Events

4.3.3 Fine-Tuning

With the unsatisfactory results from Step 3, the fine-tuning procedure ensues. Variables are tuned using component planes as a guide. Figure 34 depicts the component planes and clustered maps used to aid in variable selection. Solid green component planes represent all the phase number, Bottom reactor Valve PV and Op, PA Flow Valve OP and their difference variables. The homogenous green color indicates they are constant at this transition and further evaluation does not consider the variables. The two reactor temperature measurements along with the PA’s inlet and outlet temperatures are similar to the strong brown cluster at the top of the clustered map. Analysis shows this cluster is a particular grade with a high temperature in the recipe, which is not important information, so they are removed as well as their differences. Both the hot oil (HO) flow rate and valve position create a sharp region in the upper right hand corner. This is due to the flow being on if heat is needed, or off if it is not. These are unimportant events and their differences offer no additional information. The hot oil inlet and outlet temperatures show a gradient across the map. These variables’ dependence on time causes this gradient. The hot oil temperatures change with the outside, weather it is day or night. Their diff variables offer no useful information. The Batch ID counts and is then reset. It therefore is related to time, but has alternating red and blue bands pertaining to its high and low values. The direction of alternating bands on Batch ID is parallel to time. The hot oil temperatures gradient direction is perpendicular with time as they are high and low each day.
Figure 34: Component Planes and Clustered Map for Fine-tuning Analysis PA Section

The type of observed faults obstruct pump around flow due to mechanical difficulties. The PA flow has regions that correspond largely to its long-term dynamics, which offers no valuable insight. The PA Flow difference however has distinct regions that correlate to PA flow events. This could be due to known issues or non-detected problem. Our goal now is to isolate regions of the difference map highlighting immeasurable faults. The heat exchanger (HE) pressure drop and its difference have a distinct region that corresponds to an important PA event, clogged filter. This variable, along with its difference eliminate measured flow drops and help accentuate the cluster we wish to identify. The level variable has a region that corresponds well sudden drops in the flow rate. The variable also has other
nonsensical variation, but the level difference contains the same information without the noise without the noise.

After tuning the variables, the flow rate change, filter pressure drop change, level changes, and filter pressure drop state value are the final variables. The map is clustered and depicted as the far right map of Figure 35. The second map on Figure 35 is the HIT diagram of faults we are trying to identify. They are located in the same region of the map and are clustered together as an orange cluster. The cyan and bright red clusters correspond to normal. The flow change variable demonstrates that all not normal area demonstrate a process event. Filter being clogged region is well defined by the Pressure drop across the filter. The filter drop difference also identifies regions where the filter was emptied. PA function also changes at the end and beginning of grade runs, signaling a level drop and increase. The remaining abnormal region of the flow difference map corresponds to the undetected events.

![Clustered Map](image)

Figure 35: SOM results for variables selection

Once the variables corresponded well to data clusters on the umatrix, the data was categorized and labeled.

### 4.3.4 Technique Selection

Once data has been labeled, technique is evaluated by predicting the labeled classification against the known prediction rates. The first technique evaluated is PCA with $t^2$. A fault detection process using PCA to reduce the process space and $t^2$ ellipsoid to identify data points was tested. The Figure 36 shows the projected space with the labeled data the red and blue colored point opposite one another are the clogging and cleaning of the filter, the yellow and green across one another re the level dropping and increasing during grade switches and the black dot above the normal region in cyan is the other abnormal faulty conditions. Figure 36 on the right shows the boundary of the faulty region using $T^2$ ellipsoid.
This method leaves some faulty points outside of the faulty region boundary. This region contained an 85% classification rate before many false positives began occurring. If the boundary were increased, misclassifications where the normal points are deemed faulty would increase.

This method is compared to SOM with density estimation. Since data on the fault is limited, we do consider classification of the training data. On the training set, the PCA method gives an 85% detection rate of misclassifications. The SOM classifies them 100% correctly. For a historical data, the SOM method detects 22 previously occurring faults while the PCA method detects only 15. Single map SOM is chosen over PCA.

Below, PCA contribution plots with the SOM contribution plots proposed are compared. In each row, a particular event and its contribution using traditional PCA techniques and our developed SOM technique is compared. Level increases and decreases both signal level as the major contributing variable. Along with clogged filters and the other faults, the flow change alone is signaled as the highest contributing factor. This result is consistent with intuition that the fault contribution consists of the flow rate, and nothing else. For emptied filter events however, the newly developed technique puts more emphasis on pressure drop change rather than the flow change. This is a more useful analysis since flow rate change covers the more general suite of PA problems.

After the SOM is implemented, the SOM visualization features are utilized. Figure 38 of the process trajectory projected onto a colored SOM map. The orange region pertains to a No Event status, the dark blue corresponds to the faulty region of the map, the light blue and cyan are clogged filter and emptied filter respectively and the brown and yellow are level drops and level increases. Each black dot is a batch and they are connected in the order they occur.
Figure 37: PCA/SOM Contribution Plot Comparison

Figure 38: Online Trajectory Projection for Different Events
When the process is running normally, the trajectory simply bounces around the normal region. When a clogged filter occurs, the trajectory moves into the clogged filter event. The operator then cleans the filter and process moves to the cyan cleaned filter event. The trajectory also moves from brown to yellow during grade switches.

**4.4 Tennessee Eastman Process Application**

This work proposes a methodology for the offline development and online implementation of data-driven process monitoring schemes as well as a novel use of SOM for process-monitoring tasks. Previous topological preservation applications in PM have been restricted to the fault diagnosis task. These applications use a single SOM (1-SOM) for all process operating regimes. This presents challenges when a new state is encountered because the map must be trained again to monitor for the new condition. Here, SOMs are extended to aiding all aspects of process monitoring and the fault diagnosis is performed in a more flexible way. Specifically, fault detection is performed by using a Gaussian mixture model of the normal region. For fault identification, a new method is introduced using residuals of the deviation from the normal region. Fault diagnosis is done by creating a map for each fault, known as MSOM. This allows the ability to include new faults without directly affecting previously characterized faults. The proposed methodology is applied to the Tennessee Eastman Process (TEP). Previous studies of the TEP have considered particular step-change faults where the root causes of the disturbance are generally limited to one variable. Here, in order to fully utilize SOM's nonlinear topology preservation features, a focus on analyzing more challenging faults such as random variations, sticky valves and slow drift in kinetics is included to effectively illustrate the advantage afforded by SOM. Implementing the proposed methodology indicates that MSOM is able to improve upon linear distance preservation techniques such as PCA and the more standard SOM based approach in process monitoring tasks.

Distance-preservation fault-detection techniques, including PCA, have previously been applied to the Tennessee Eastman process with great success. The ease with which training data can be generated using the Tennessee Eastman Process has allowed it to become the benchmark for any new process monitoring scheme. Raich and Cinar, (1996) demonstrated the use of standard PCA for fault detection and diagnosis on the Tennessee Eastman process. (Zhu 2011) applied an ensemble clustering method based on a dynamic PCA-ICA model to label process transitions to the TEP and achieve transition process monitoring using PCA based dimensionality reduction. A thorough treatment of PCA and other statistical techniques to the Tennessee Eastman process is presented in Chiang, (2001). A comparison of these standard techniques applied to TEP can be found in Yin et al (2012). Previous applications of linear data-driven techniques to the TEP have included detection, identification, and diagnosis rates. These previous works all may be limited by the use of distance preservation techniques and are more applicable to the step changes faults they are applied. Some researchers have applied SOMs nonlinear topological-preservation features to the TEP. Chen (2012; 2013) improved the performance of their SOM algorithm using two linear dimensionality reduction techniques, CCA and FDA. Gu, (2004) presented the visualization of several step faults in the Tennessee Eastman process. Ng, (2008) applied SOM to a different set of
measurements generated by the TEP. These works include multiple step faults on a single map. They have not included a method of fault identification and were not applied to faults with any time dependency.

Distance-preservation fault-detection techniques, including PCA, have previously been applied to the Tennessee Eastman process with great success. The ease with which training data can be generated using the Tennessee Eastman Process has allowed it to become the benchmark for any new process monitoring scheme. Raich and Cinar, (1996) demonstrated the use of standard PCA for fault detection and diagnosis on the Tennessee Eastman process. (Zhu 2011) applied an ensemble clustering method based on a dynamic PCA-ICA model to label process transitions to the TEP and achieve transition process monitoring using PCA based dimensionality reduction. A thorough treatment of PCA and other statistical techniques to the Tennessee Eastman process is presented in Chiang, (2001). A comparison of these standard techniques applied to TEP can be found in Yin et al (2012). Previous applications of linear data-driven techniques to the TEP have included detection, identification, and diagnosis rates. These previous works all may be limited by the use of distance preservation techniques and are more applicable to the step changes faults they are applied. Some researchers have applied SOMs nonlinear topological-preservation features to the TEP. Chen (2012; 2013) improved the performance of their SOM algorithm using two linear dimensionality reduction techniques, CCA and FDA. Gu, (2004) presented the visualization of several step faults in the Tennessee Eastman process. Ng, (2008) applied SOM to a different set of measurements generated by the TEP. These works include multiple step faults on a single map. They have not included a method of fault identification and were not applied to faults with any time dependency.

4.4.1 Process Description

The process uses four gaseous reactants (A, C, D, and E) to produce two liquid products (G and H) in competition with an unwanted by product (F) and in the presence of an inert (B). There are four competing reactions related to temperature through Arrhenius laws, two producing desired products, the others producing the byproduct. The entire simulation includes five unit operations: a reactor, condenser, separator, compressor, and a stripper and has 41 measured variables along with 12 manipulated variables for a total of 53 process variables. Downs and Vogel (1998) also defined 20 disturbances or faults for the challenge process.

The data was generated using the control scheme and Simulink code from Ricker, who applied a decentralized control strategy to the process, which involved partitioning the plant in to subunits and the creation of a controller for each. The process flow sheet and controller scheme (from Ricker, 2000) is given in Figure 39. Because online application is the focus of this work, composition measurements with a time delay are not considered, leaving 22 of the original 41 measurements for consideration. Additionally, of the 12 manipulated variables, three are held constant under Ricker’s control scheme (compressor recycle valve, stripper steam valve, and agitator speed) and not considered in the analysis presented.
The process simulations of the TEP process contain 20 preprogrammed faults. Most of these faults are particular feed-step-change faults. In this study, in-depth analysis to characterize other more difficult faults such as random variations, sticky valves and slow drift in kinetics is addressed. Consequently, the specific faults considered in this work are:

- Fault 1 is a step change in the composition of stream 4.
- Fault 8 is a random variation in the feed composition of stream 4.
- Fault 11 imposes a random variation in the cooling water to the reactor, which requires the control system to constantly adjust the cooling water flow to compensate for the reduced or increased cooling capacity of the water used.
- Fault 13 is a slow drift in the kinetics of the reaction which requires many adjustments by the control system to accommodate the changing composition of the reactor output.
- Fault 14 is a sticky valve in the reactor cooling water. Its effect on the rest of the process is similar to the earlier random variation faults where it creates variations that would not be observed under normal operation.
- Faults 14 and 1 (14+1) is the combination of faults 1 and 14, where the state is disturbed by a step change in the composition of stream 4 while coping with a sticky valve.

Step faults are well-studied using traditional SOM methods (Gu 2004, Chen 2012, 2013). In this section Fault 1 is studied, but the focus is on the dynamic Faults 8, 11, 13, 14 and a combination of faults 14 and 1 (Fault 14+1). First, faults are understood using SOM
analysis and PCA/clustering projections to understand their effect on the TEP. Various faults were run and tested against PCA and SOM detection and identification methods. Finally, the fault diagnosis rates of PCA, standard SOM, and MSOM are compared.

This study performs the offline development of a data-driven scheme for the TEP and analyzes the results of their implementation.

### 4.4.2 Data Analysis

For Step 1, gathering process insight and data collection, data from 48 hours of normal operation were collected before introducing each fault under consideration and generating 48 hours of data during faulty operation. The data from each of these runs were projected into three dimensions using PCA and plotted in each row of Figure 40. Data in the left column are colored according to the known process status input into the simulation and the data in the right column are colored according to the historical data labeling procedure outlined in Section 4.2.

Fault 1 and Fault 1+14 include a step change that brings the process to a new operating condition in the plant. For these faults, the k-means approach does a good job at differentiating the normal and faulty operating regime. For the other dynamic faults, the faulty data surrounds the normal regime, making it more difficult to differentiate them from normal. Fault 1 has been studied in previous works (Gu 2004, Chen 2012, 2013) and is not covered in-depth here.

Faults 11 and 14 are localized faults in the cooling water and the control system quickly adjusts the cooling water rate. Fault 11 appears as random noise around normal as the control action compensates for the noisy cooling water temperature. Since Fault 14 is a sticky valve fault, there is a lag time between the controller action and the temperature deviation from normal. As a result, either the reactor temperature or the cooling water valve is always deviated from normal, which prevents all variables from entering the normal operating region simultaneously, which is easy to distinguish from normal operation. An interesting scenario is if multiple faults are occurring simultaneously. When Fault 1 and Fault 14 occur simultaneously, a similar shell is created around Fault 1 operating point similar to what Fault 14 created around normal operation. For these faults, the clustering algorithm isolates normal and spherical sections around normal, which is due to the k-means weighting function assuming a distribution of a spherical or hyper-spherical shape. Fault 14 data are differentiated from normal very well. For Fault 11, faulty points in the interior of normal operation are misdiagnosed as normal.

Faults 8 and 13 are plant wide and affect many process variables and units. The action of multiple controllers compensating for the propagating error give these faults long trajectories. Each fault affects the system differently, and the trajectories of the dynamics each contain unique patterns that can be recognized by an effective fault diagnosis algorithm. Since Fault 13 map’s prototype vectors will be more concentrated along the trajectory, points running along the trajectory are more likely to be classified as Fault 13. Density estimation could identify areas where the transition trajectory is likely to pass through during the evolution of the fault. The hyperellipsoids created from parametric
statistical methods have difficulty differentiating the overlapping but uniquely patterned
dynamic faults, which will be a problem in diagnosis.

Figure 40: Data projected into the PCA space colored according to (left) true class and
(right) unsupervised clustering results from k-means
4.4.3 Historical Data Labeling

The TEP simulation does not offer any key insight to variable selection as only pertinent variables to the process were included in the simulation. However, variables with time delays such as composition, are eliminated due to being temporally distant from the event. The simulation offers a definite database where at any point in the process the operating regime is known with certainty, which can be exploited to evaluate the historical data labeling process described above.

A data set consisting of all faults was classified using the method outlined in Section 4.1 with a k-means clustering algorithm. The k-means index minimizes an error function using a predefined number of centroids. Since the number of clusters is an input to the algorithm, a performance metric is used to quantify its effectiveness. The Davies-Bouldin index is proportional to the ratio of the error within the classes to the error between the classes. The minimum Davies-Bouldin index was five, however only a few transitional parts were separated; therefore, the next minimum, 29, was selected.

Figures 41 below contain the data projected into the PCA space. In Figure 41a, the data colors refer to the operation condition. The normal operating regime, as well as fault 14 cannot be seen in the cloud. In Figure 41b, an enlarged picture of the cloud core is shown. Figure 41c shows the data labeled by the clusters.

![Figure 41: a) Data Colored according Fault, b) Expansion Data Core, c) Data labeled according to cluster](image)

A process ensues where clusters are identified based on the position of manually labeled data on the map. In Figure 49c the clustered map, the clusters are projected onto the map. Not all 29 clusters are clearly separated due to the number of colors required. It can be seen however that region boundaries around normal, Fault 8, Fault 13 and more are captured, even if by multiple clusters. Fault 14+1 surrounds the region of Fault 1 similar
to how Fault 14 surrounds the normal region. This process led to 87% accuracy of labeling the data.

![Figure 42: MSOM in Fault Detection of TEP](image)

### 4.4.4 MSOM in Fault Detection of TEP

Fault detection was tested by comparing PCA based fault detection methods with SOM. Data was collected for the six faulty operating regimes and run through mPCA or MSOM fault detection algorithms. The correct classification rates of the two methods are given in Table 16. All variables without a time delay were used except the compressor recycle valve, stripper steam valve, and agitator speed as these numbers were held constant by Ricker’s control system. For fault detection, 480 data points were collected from normal operation, corresponding to 48 hours of operation and used the data generated as a training set for an SOM map. After training, we used a GMM to represent the probability distribution and characteristics the normal operating region. The bound for normal operation was found using the optimization described in Section 4.2.

<table>
<thead>
<tr>
<th>TEP Fault</th>
<th>mPCA</th>
<th>MSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>86.0</td>
<td>99.6</td>
</tr>
<tr>
<td>1</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td>8</td>
<td>99.4</td>
<td>99.3</td>
</tr>
<tr>
<td>11</td>
<td>99.7</td>
<td>99.0</td>
</tr>
<tr>
<td>13</td>
<td>99.6</td>
<td>99.5</td>
</tr>
<tr>
<td>14</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td>14+1</td>
<td>99.9</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Figure 43 gives the map representation of the normal region. Each map has its own color coding according to the probability of a data point belonging to the each map node. Figure 43a gives the PDF of the normal point on the map. It can be seen that the data point with the reddest color indicates the highest probability of being associated with that map node and corresponds to the map node that best represents the data point. Figure 43b
however is the same illustration point at the beginning of Fault 1’s deviation. The state of the process is leaving the normal region and approaching the boundaries of normal operation. Finally, in Figure 43c, we see the representation of the faulty point long after Fault 1 has occurred. The solid green color indicates a zero probability of the data point being associated with any map node, meaning that a fault has been detected. Other faults follow the same behavior.

![Figure 43](image)

**Figure 43**: Plots of the PDF of the state of the TEP as it progresses through Fault 1.

### 4.4.5 SOM in Fault Identification of TEP

Fault identification was performed on data sets from the Tennessee Eastman Process according to the PCA method described in Section 3.1 and the MSOM method proposed in Section 3.2. Contribution plots of faulty data points from each fault were analyzed and compared with contribution plots calculated by a PCA method and our process knowledge to evaluate the novel method’s ability to isolate the causes of the faulty condition. Sometimes, the root causes of the problem are not in variables we have data for, so the goals of fault identification are to isolate a subsection of the plant where the root causes of the issue are likely located and to give engineers useful clues on how to identify them. In Figure 44, a data vector roughly 10 hours after the imposition of the faulty condition was used to create contribution plots as illustrative examples of the method proposed.

In Figure 44a, contribution plots generated from Fault 1 data are compared. In response to the changes in the composition of the feed, the composition controller on the reactor feed creates changes in Streams 1 and 4 in response to the disturbance. Both PCA and MSOM point to a disturbance in the composition of the feeds to the process since from the plots it can be seen that variables associated with the feeds (Stream 1 through 4) have significant contributions. In both cases, the variable with the largest contribution is the controller output to Stream 1, which, based on the knowledge that Streams 1 and 4 are in the same control loop, points to a feed composition issue.

Applying a similar analysis to Fault 13, it can be seen that MSOM successfully reproduces the results of PCA in isolating the fault to changes in the pressure of the reactor and in the separation units following the reactor. Dramatic changes in all of these variables at the same time could point to a change in the composition of the outlet of the reactor. In fact, Fault 13 creates a “slow drift” in the kinetics of the reactor. The result is
wild variations spread over many process variables, but both methods successfully isolate it to a subset of variables related to the reactor and its composition.

Faults 11 and 14 are illustrative of the value of using MSOM. Both faults relate to the cooling water of the reactor, which may not be very important to the PCA model, as under normal operation reactor cooling water faults remain relatively constant. The result in Figure Fault 11 and Fault 14 is that when a fault related to the cooling water occurs, cooling water variables are sidelined in favor of flow variables. For fault identification, MSOM uses all online measured variables and weights them equally. The result is that while PCA appears to point to more changes in feed streams, MSOM’s largest contribution variables clearly point to a problem with the reactor and its cooling water.

The plots in Figure 44 illustrate MSOM’s superior ability in identifying variables responsible for a deviation from normal operation. The data compression abilities of SOM help identification by reducing the noise in individual data, without the loss of information that comes with performing a dimensionality reduction using PCA. The next section will illustrate the effectiveness of MSOM in diagnosing the faulty condition of the process.

![Figure 44: PCA/SOM Contribution Plot Comparison](image)
4.4.6 MSOM for Fault Diagnosis of TEP

In this section, the MSOM method proposed was used to classify data sets from the faults of interest and compared to two standard techniques: mPCA, a standard fault detection and diagnosis technique, and the standard form of SOM which uses a single map for classification. In each case, a series of data from an operating regime of the Tennessee Eastman process is given to each set of models, and every data point is classified into the different fault categories using each technique’s discriminant function. PCA uses Hotelling’s $T^2$ statistic, 1-SOM uses a given data point’s BMU on the map of the process, and MSOM uses the probability estimated from the GMM fit to each class’s map.

From the results in Table 17, both 1-SOM and MSOM techniques represent a dramatic improvement over conventional PCA fault detection. 1-SOM gave fewer false fault classifications, but MSOM had a better correct classification rate, particularly for the random variation faults.

<table>
<thead>
<tr>
<th>Table 17: Fault Diagnosis TEP Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEP Fault</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>14+1</td>
</tr>
</tbody>
</table>

Figure 45 shows how each method classified different portions of the process data. Visualizing the results in this way allows us analyze possible reasons for low classification rates in the results. For example, MPCA’s successful diagnosis of Fault 13 suggests that this fault has been characterized well by its PCA model, while Figure 23 shows that this high detection rate for Fault 13 came at the expense of a large number of points from Faults 8, 11, and 14 being incorrectly classified into Fault 13 due to Fault 13’s large $T^2$ bounds.

Another challenge comes from regions where multiple fault operating regimes overlap. For example, Fault 14’s classification has been spread by mPCA between Fault 8, 11, 13, and 14 due to overlap of the four models. A similar case is the high detection rate for the combination Fault 14+1 where a high correct diagnosis rate for Fault 14+1 was achieved at the cost of many points from Fault 1 being incorrectly diagnosed due to overlapping regions for Faults 1 and 14+1.
4.5 CONCLUSIONS

The method explored here differs from other SOM-based process monitoring schemes due to the use of multiple maps to characterize the nonlinear topology of each class of faulty data individually. Results of the offline development analysis indicate that labeling historical data labeling using unsupervised clustering techniques is useful, but gives inaccuracies. Due to these methods sensitivity to accurate data, the adaptive features of the online implementation approach proposed are critical. After training a map to each fault class, a GMM is created for each map in the state space from its constituent vectors and used in detection and diagnosis as a discriminant function to classify new data vectors. This process monitoring architecture was compared to PCA and a more conventional 1-SOM on data generated from the Tennessee Eastman Process, with a focus on faulty conditions containing non-linear variations with time that have not been previously analyzed with SOM. The results indicate that MSOM outperforms MPCA and improves upon SOM classifications.

The results indicate that short-term dynamics can be reduced by characterizing the process at values during phase transitions. Long-term dynamics can be reduced by taking the difference between batches. These variables transformations were able to capture dynamics and reduce dynamics so that unsupervised pattern recognition algorithm clustered similar events together. A particular cluster identified in the pump around system of a polymerization unit that is currently a non-detected fault. SOM was used as a model to monitor the process and identify faults. Clogging, pump issues, and sticky valves slow the heat-up period phase and lengthen the batch time. This scheme has an estimated 100$k/year.
CHAPTER 5: CONCLUSIONS AND FUTURE WORKS

Due to prevalent advanced control systems and instrumentation, engineers can now utilize the wealth of information available and improved computational ability of processors to enhance process control, optimization, and process monitoring. In this thesis work, these improvements were utilized to build novel models for performing chemical engineering tasks. Simulations are able to perform more accurate analysis of complex processes. Modeling solution methods were discussed to aid initial conditions to solving these models.

The increasing computational ability allows for model size to increase and optimization technique to perform more exhaustive searches of the solution space. With the constraint on computational ability loosening, the size of optimization problems being undertaken becomes a limitation of the time and energy of personnel to model the process. We propose specialization of tasks which the model is constructed by specialized individuals on specialized software to break the modeling tasks into smaller pieces. This requires the use of stochastic optimization since modeling equations will not readily be available. In an effort to perform multilevel optimization, we integrated a crude oil loading and loading problem with a production unit optimization problem. This involved formulating a unique model to consider production level costs. The production unit problem was optimized using a stochastic algorithm, which exacerbated convergence issues of the model. In this work, a method to formulate initial conditions to repair non-converged simulations was proposed and validated. The repair method was then integrates with the stochastic method which served an additional perturbation phase of the algorithm.

When modeling phenomenon with first-principles is too arduous of a task, engineers are beginning to utilize a wealth of information accumulated in by DCSs. Historical based modeling can be used to predict variables and process states. Previous applications of PCA have been used with success. PCA has function extract important variations from data and analyze it in a way useful for engineer detecting, diagnosis, and identifying faults. In the work here, data-driven techniques were implemented on processes where PCA failed to give adequate results. A dimensionality reduction technique SOM was reformulated to provide necessary tools to perform each of the PM tasks. The technique was implemented on a highly nonlinear unit with positive results. A currently non-detected fault has been shown to be captured by the novel method. The steps to implementing the routine were clearly identified to aid others in implementing data-driven techniques. The steps were applied to the Tennessee Eastman Process. The results strongly validated the efficacy of the routine at identifying and diagnosing highly nonlinear faults.
REFERENCES


APPENDIX A: DIMENSIONALITY REDUCTION TECHNIQUES REVIEW

In this Appendix, we give a detailed summary of Nonlinear Dimensionality Reduction techniques. Data-driven techniques are originating in the mathematical domain faster than they are being applied to process monitoring schemes. It is therefore useful to learn the different techniques and how they could be utilized in a process-monitoring framework. The dimensionality reduction techniques description and images from their implementations are the work of (Lee & Verleysen, 2006). Here we add thoughts on their application to Process Monitoring.

Feature extraction is performed through dimensionality reduction. The reason dimensionality reduction (DR) helps accentuates a data cloud’s features by being a solution to the “curse of dimensionality” (Bellman, 1961). The number of dimensions of a vector space increasing leads to many odd effects including: the volume of a sphere vanishes as dimensionality increases; all the volume of a sphere is concentrated in its shell; Gaussian distributions behave strangely because the confidence radius is a function of the dimension; the variance of the vectors’ norms becomes relatively small compared to their mean. These odd effects make it hard to define regions in higher space.

A.1 REDUCTION PROPERTIES

DR techniques are classified by properties important in the characteristics of the data being reduced, assumptions made during their derivation, or features of their implementation. In this section, some commonly mentioned features are described some observations of their relevance to process-monitoring field are stated.

a) Hard vs. Soft
Whether a problem is hard or soft depends on the ratio of input to output variables. Hard DR can include up to thousands of initial variables. PCA typically finds itself useful. For very soft reduction, initial size less than ten, the goal is typically to get good visualizations as well as classification improvement.

Fault detection problems are typically midrange. They can encounter large scaled problems demanding many variables to be analyzed. They benefit from visualizations to aid engineer and operator decision makers absorb data readily.

b) Traditional vs. generative Models
Traditional vs. generative refers to the relationship between the latent and observed variables, known as the model of the method. Some models try to predict real data from latent variables. These are encountered less. I do not see a clear fit for a generative model in our research.

c) Linear vs. Nonlinear Model
This property name is a straight forward description of the assumptions made by the technique. Many techniques perform some sort of spectral decomposition of a data matrix (Gram or Covariance), which assumes a linearly independent
relationship among the rows and columns of the matrix. Nonlinear techniques are more powerful because the relationship between the latent and observed space may be richer than a matrix multiplication. However, nonlinear techniques require more data to fit a larger number of parameters. Many processes which fault detection can be applied have sufficient historical process data to take advantage of non-linear techniques advantages.

d) Continuous vs. Discrete
Continuous techniques such as PCA map spaces in such a way that is easier to map new points. Discrete methods such as Self Organized Maps (SOMs) mapping new points being embedded into the data set require some sort of interpolation.

In order to take advantage of SOM’s ability to capture nonlinearities, I look gather enough data to ensure we have a large enough data set to accommodate a map which encompasses the data and are close enough to remove interpolations estimation errors.

e) Layered vs. Standalone
Refers to whether the coordinates of target dimension change if another latent dimension is added. PCA is layered, SOM is standalone. All standalone coordinates must be calculated each time a new dimension is added. We anticipate having an engineer choose an optimal target dimensions and the model run autonomously. This means reevaluating coordinates at lower different target dimensions would be rare.

f) Single vs. Multiple Coordinates Systems
Sometimes sections of the data can be localized and there are multiple coordinate systems. An example is localized PCA. We anticipate multiple coordinate systems for particular parts of large units. Furthermore, coordinates could be problem specific.

g) Optional vs. Mandatory Vector Quantization.
Vector quantization (VQ) is performed when there is a large data set; observations can be replaced by a smaller set of sometimes called prototype vectors or centroids. This has the benefit of data compressions and can be more computationally efficient. This is a necessary step in SOM, however the benefits in fault detection are not fully understood.

h) Batch vs. Online Algorithm
This property refers to how the data need to be available. Batch training algorithms provide that the data all be available to train, while online require one data point at a time. We expect models initially be trained in any fashion. Then online algorithms can be used to improve mapping with new data being described by the engineer.
i) Exact vs. Approximate Optimization
Many DR techniques try to minimize a cost or stress function. Some find global minima while others try to iterate toward a nadir. Property closely related to batch or online training. Batch trained algorithms like PCA find exact solutions to an objective function whereas adaptive methods with on-line ability like SOM can fall into local optima. Most exact global optimization techniques lack certain flexibility.

j) The type of criteria being optimized
When reducing dimensionalities, descriptions of distances can be qualitative or quantitative. Quantitative are measurable amounts such as Euclidean distances. Qualitative relationships such as rankings can be used. For example: a & b closer than a & c. I expect to use quantitative uses for most process monitoring applications. However, I will keep in mind qualitative abilities which may help formulate complex objectives such as incorporating safety considerations.

A.2 Feature Extraction Techniques
Feature extraction technique usually try to map data from their observed space, y, into a different space x, typically at a lower dimension, in order to better understand the data. An error function to compare methods can be

\[ E_{\text{codec}} = E_y \{ ||y - \text{dec}(\text{cod}(y))||_2^2 \} \] (78)

Where, cod: \( \mathbb{R}^D \rightarrow \mathbb{R}^P \), \( y \rightarrow x = \text{cod}(y) \) and dec: \( \mathbb{R}^P \rightarrow \mathbb{R}^D \), \( x \rightarrow y = \text{dec}(x) \). The most well-known and studied DR technique is PCA. It has many applications in multivariate analysis and data mining.

A.2.1 Principal Component Analysis (PCA)
PCA assumes a linear relationship between latent and observed variables, the variables are centered at the origin, and latent variables follow a Gaussian distribution. PCA creates a mapping \( W \) which \( D \) observed variables are explained from \( P \) unknown latent variables,

\[ y = Wx \] (79)

\( W \) is an axis change meaning its columns are orthogonal and unit vectors, thus \( W^T W = I \). So the first step in implementing PCA is centering by subtracting the expected value. It should be noted that constant values will produce an error due to dividing by standard deviation. Also, noise can be amplified by normalization of variables that have small standard deviations. PCA will consider this noise an important independent variance.

PCA can be performed two different mathematically equivalent ways. PCA’s error can be defined using functions

\[ \text{cod}(y) = W^T y \] (80)
and
\[ \text{dec}(x) = Wx \text{ as } E_{\text{codec}} = E_{y} \| y - WW^T y \|_2^2 \]  

(81)

If the linear assumptions are accurate and the data is not noisy,

\[ WW^T y = WW^T Wx = WI_{P} x = y \]  

(82)

and the reconstruction error is zero. In many situations, data noise and nonlinearities case the reconstruction error to be greater than zero. By expanding the error function, it can be shown minimizing the error is the same as maximizing the term \( E_{y} \| y^T WW^T y \|_2^2 \), which can be estimated from data as \((1/N)\text{tr}(Y^T WWY^T)\). By inserting the singular value decomposition of the data,

\[ Y = VEU^T \]  

(83)

The optimization problem can be stated with solution as

\[ \text{argmax}(W) \text{tr}(UE^T V^T WW^T VEU) \]  

(84)

Since \( V \) and \( U \) are orthonormal vectors, the solution is at \( W = VI_{D,P} \). A mathematically equivalent approach perform eigenvalue decomposition on the covariance matrix of the data, \( C_{yy} = VΛV^T \). The eigenvectors, \( Λ \), can be related to the covariance of the latent variables since the covariance of the data in the latent space and observed space are related,

\[ C_{XX} = W^T C_{yy} W = W^T Λ W \]  

(85)

The latent variables will be most independent at

\[ C_{XX} = I_{P,D} Λ I_{D,P} \text{ when } W = VI_{D,P} \]  

(86)

Latent variables can be estimated as

\[ X_{\text{pred}} = I_{P,D} \# V^T Y \]  

(87)

PCA has been successfully implemented in many applications that spurred more interest in dimensionality reduction techniques. Dimensionality reduction methods have tried to better compensate for the PCA linear assumption. Nonlinear DR techniques are classified according to distance preservation and topologic preservation techniques.

Let us consider nonlinear distance preservation techniques. Ideally, if the distances between points in the latent space is the same as the observe space, the main geometry properties are the same. Well-behaved distances have certain properties such as non-degeneracy, triangle inequality, non-negativity, and symmetry. These techniques can be defined as an optimization problem where a distance is defined along with a cost function to be minimized. Initially techniques would try to preserve spatial distance preservation.
Spatial Distance Preservation Techniques

Multidimensional scaling (MDS) provides equivalent results as PCA although it performs operations on the Gram matrix. MDS also looks for a linear transformation where Y = WX. The Gram matrix, S, is a matrix of products,

\[ S = Y^T Y = X^T X \]  \hspace{1cm} (88)

Eigenvalue decomposition of the Gram matrix yields:

\[ S = U \Lambda U^T = (A^{1/2} U^T)^T (A^{1/2} U) \]  \hspace{1cm} (89)

If you let

\[ W = A^{1/2} U^T, \quad X_{\text{pred}} = I_{P,N} A^{1/2} U^T \]  \hspace{1cm} (90)

This can be shown to be equivalent to PCA as

\[ X_{\text{pred,PCA}} = I_{P,D} V^T Y = I_{P,D} V^T V E U^T = I_{P,D} (E^T E)^{1/2} U^T = I_{P,N} A^{1/2} U^T = X_{\text{pred,MDS}} \]  \hspace{1cm} (91)

The error function being minimized can be written as

\[ E_{\text{MDS}} = \sum_{i,j} (\langle y(i) y(j) \rangle - \langle x(i) x(j) \rangle)^2 \]  \hspace{1cm} (92)

where dot product are preserved. There is a computational load can be minimized depending on weather \( Y Y^T \) to \( Y^T Y \). If the dimension is much less than the number of data points; PCA is more computationally efficient since \( Y Y^T \) is smaller than \( Y^T Y \). If the data points are much larger than dimension, MDS is more efficient. Sammons Nonlinear Mapping (SNLM) tries to put more emphasis on preserving distances of nearby points by having an error function:

\[ E_{\text{SNLM}} = \sum_{i,j} \left( \frac{(dy(i,j) - dx(i,j))^2}{dy(i,j)} \right) \]  \hspace{1cm} (93)

where \( dy(i,j) \) is the distance between points in the observed space and \( dx(i,j) \) is the distance between points in the latent space. SNLM is performed by initializing data points in the latent space using MDS or PCA, then updating through an iterative method until there is no improvement in the error function. Use more sophisticated optimization routines which allows for a more complex objective function as opposed to MDS which uses algebraic techniques. Curvilinear component analysis (CCA) also puts an emphasis on preserving local distances with a stress function

\[ E_{\text{CCA}} = \frac{1}{2} \sum_{i,j} (d_y(i,j) - d_x(i,j))^2 F_\lambda(d_x(i,j)) \]  \hspace{1cm} (94)

\( F_\lambda(d_x(i,j)) \) is more generic than \( 1/d_x(i,j) \). \( F_\lambda(d_x(i,j)) \) is a positive decreasing function and \( F_\lambda(d_x(i,j)) = H(\lambda - d_x) \) where \( H(u) = \{ 0 \text{ if } u \leq 0, \ 1 \text{ if } u > 0 \} \). Points are initialized by some method, for example using PCA, and then are iteratively updated to minimize the error.

Graphical Distance Preservation Techniques

A variant on preserving spatial distances is preserving geodesic distances. Geodesic distances are the distance between points along the embedded manifold. Geodesic
Distances can be estimated with graphical distances. Data sets are fit within graphs and nearby points are connected. A simple and common way is the K-rule, where points are connected to their K closest neighbors. Another is the $\varepsilon$-rule where points within a distance $\varepsilon$ are connected. Each arc is given a weight, commonly the Euclidean distance between the points. The distance between each pair of points becomes the minimum sum of weights of all paths connecting the two points. Isomap, Geodesic NLM (GNLM), and Curvilinear distance analysis (CDA) define distances using graphical techniques and then perform the same operations as MDS, NLM, and CCA respectively.

Other distance preserving methods warp spatial distances in a less intuitive manner, Kernel PCA and semi-definite embedding (SDE). Kernel PCA tries to linearize local properties by applying a kernel function to the data before performing eigenvalue decomposition. Kernel PCA has not produced many good applications. SDE, also known as multivariate unfolding, maximizes distances if they are not in the same neighborhood,

$$\sigma = \frac{1}{2} \sum_{i,j} d_{*(i,j)^2}$$

(95)

So there is a constraint that if $i$ and $j$ are in the same neighborhood $||x_i-x_j||^2 = ||y_i-y_j||^2$. To prevent an unbounded solution, a constraint

$$\sigma \leq \sum_{i,j} \delta_{y^2(i,j)}$$

(96)

Where, $\delta_{y(i,j)}$ is the graph distance between the points is added. The distances can be represented as dot products

$$d(i,j) = s(i,i) - 2s(i,j) + s(j,j)$$

(97)

Where, $s(i,j) = <y_i*y_j>$. If a matrix $K$ is initialized as matrix of the dot products. It is then iteratively updated to maximize the trace subject to the constraints: it remains positive semidefinite, the sum of all entries is 0, and $s_{x(i,j)} = s_{y(i,j)}$ for nearby points. MDS is then performed on the $K$ matrix.

A.2.4 Topological Preservation Techniques

The topology of a manifold refers to the neighborhood relationships between subregions in the manifold. Preserving distances can be too restrictive when relative region distances are the important, i.e. Region A is close to Region B but far from Region C. Methods attempt to capture the topology by fitting a graph called a lattice to the data. Topology preservation techniques can be divided into two categories, predefined lattices and data-driven lattices.

Predefined lattice topology methods include Kohonen’s Self Organized Method (SOM) and Generative Topological Mapping (GTM). Along with multi-layer perceptron, SOM is one of the most widely known methods of artificial neural networks. SOM is able to perform Vector Quantization and dimensionality reduction simultaneously. SOM first initializes a graph into the data typically along the principle components. Vertices of the graph are typically referred to as prototype vectors. The graph is then bent around the data by running through the data for several iterations, each referred to as an epoch. For each data point, a best matching unit is found as the nearest graph prototype,
\[ r = \text{argmin}_s (d(y(i), c(s))) \]  \hspace{1cm} (98)

Where, \( r \) is the nearest prototype of all prototypes \( s \) to data point \( i \). Then, all map units are updated

\[ c(s) \leftarrow c(s) + \alpha v_l(r,s)(y(i) - c(s)) \]  \hspace{1cm} (99)

Where, \( \alpha \) is a learning rate between \([0,1]\) and decreases as epochs continue. \( v_l(r,s) \) is a neighborhood function. Epochs are repeated until prototype updates become negligible. Generative Topographic Mapping assigns probability distributions to input variables and trains mapping using likelihood functions. The algorithm is a generative model and can predict probabilities of the embedded data point being in regions of the latent space. Unfortunately, GTM cannot be embedded to latent space greater than 2 dimensions.

Another interesting group of topological preservation techniques are data-driven lattice techniques such as Isotop, Laplacian Eigenmaps (LE), and Local Linear Embedding (LLE). Instead of inserting a predetermined graph initialized with PCA, these techniques typically initialize a graph from the data by creating a lattice using the K-rule or e-rule. LLE tries to preserve local angles rather than relative proximities. Preserving local distances and local angles both attempt to preserve local scalar products. The reconstruction of the data is done by only the points neighbors: \( y(i) = \sum_{j \in N(i)} \omega_{i,j} y(j) \) along each row \( \sum_{j} \omega_{i,j} = 1 \). Instead of defining an error in the observed space \( y(i) = \sum_{j \in N(i)} \omega_{i,j} y(j) \), LLE fixes the weights and minimizes an error in the embedded space, \( x(i) = \sum_{j \in N(i)} \omega_{i,j} x(j) \), by manipulating the embedded point positions. The algorithm defines a local gram matrix with entries defined as

\[ g_{r,s}(i) = (y_i - v(r))^T (y_i - v(s)) \]  \hspace{1cm} (100)

Where, \( v(r) \) is the \( r \)th neighborhood of \( y(i) \). Weights estimated by local gram matrix using the formula

\[ w_{r}(i) = \sum_u (G^{-1}(i))_{r,u} / \sum_u (G^{-1}(i))_{r,u} \]  \hspace{1cm} (101)

The weights are used to create a sparse matrix,

\[ M = (I-W)^T (I-W) \]  \hspace{1cm} (102)

The EVD of \( M \) results in a mapping by taking the second to last eigenvectors, finding the minimal covariance in the embedded space. Laplacian Eigenmaps are similar to LLE in that it tries to capture non-linearites on the local level; however, it tries to preserve local distances. It defines an error function as

\[ E_{LE} = (1/2) \sum_{i,j} \| x(i) - x(j) \|_2^2 \omega_{i,j} \]  \hspace{1cm} (103)

Where, \( \omega_{i,j} = \exp(-\| y(i) - y(j) \|_2^2 / 2T^2) \) and \( T \) is a parameter. The error being minimized can be written as

\[ E_{LE} = \text{tr}(XLX^T) \]  \hspace{1cm} (104)
Where, $L=W-D$ and $D$ is a graph distance matrix. The optimization can be shown to minimize via EVD of $L = D^{-1/2}LD^{1/2}$ and taking the smallest eigenvectors. Isotop is the last Data-driven techniques we will discuss. Isotop is sort of a variant of SOM which separates vector quantization and dimensionality reduction. If data points need to be reduced, VQ is optional. Afterwards, Isotop creates a graph using the data as the initial lattice. The graph then is updated according to the a similar rule

$$x(i) \leftarrow x(i) + \alpha \nu_l (r-x(i))$$  \hspace{1cm} (105)$$

Where, $r=\arg\min_i d(r,x(i))$ \hspace{1cm} (106)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS</td>
<td>Y</td>
<td>Batch</td>
<td>Exact</td>
<td>Continuous</td>
<td>Linear</td>
<td>Implicit</td>
<td>Y*</td>
<td>Y</td>
</tr>
<tr>
<td>SNLM</td>
<td>N</td>
<td>Batch</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>N</td>
</tr>
<tr>
<td>CCA</td>
<td>N</td>
<td>Batch</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>N</td>
</tr>
<tr>
<td>Isomap</td>
<td>Y</td>
<td>Batch</td>
<td>Exact</td>
<td>Hybrid</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>Y</td>
</tr>
<tr>
<td>Geodesic NLM</td>
<td>N</td>
<td>Batch</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>N</td>
</tr>
<tr>
<td>CDA</td>
<td>N</td>
<td>Batch</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>N</td>
</tr>
<tr>
<td>Kernel PCA</td>
<td>Y</td>
<td>Batch</td>
<td>-</td>
<td>Continuous</td>
<td>Nonlinear</td>
<td>Implicit</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>SDE</td>
<td>Y</td>
<td>Batch</td>
<td>Exact</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>SOM</td>
<td>N</td>
<td>Either</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Implicit</td>
<td>Y*</td>
<td>N</td>
</tr>
<tr>
<td>GTM</td>
<td>N</td>
<td>Batch</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Implicit</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>LLE</td>
<td>Y</td>
<td>Batch</td>
<td>Exact</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LE</td>
<td>Y</td>
<td>Batch</td>
<td>Exact</td>
<td>Continuous</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Isotop</td>
<td>N</td>
<td>Either</td>
<td>Approx.</td>
<td>Discrete</td>
<td>Nonlinear</td>
<td>Explicit</td>
<td>Y*</td>
<td>N</td>
</tr>
</tbody>
</table>

*Uses an approximation formula
A.3 **Toy Examples**

In the section, the performance of the feature extraction techniques is analyzed. Embedded toy examples are unfolded to view the different feature’s unfolding ability and ability to handle non-convexities.

**A.3.1 Swiss Roll**

A Swiss Roll is a cake, common in Switzerland, where jelly is spread over a light pastry and rolled up. Data mimicking the Swiss Roll is like the jam with the pastry removed. It is an example of a 2D manifold that has been embedded into a 3D space. The Swiss Roll data set has a nice mathematical property of being a developable surface, which means it can be flattened out on to a surface with stretching or compressing the data. There are variants of the Swiss Roll that also help compare dimensionality reduction techniques are the thin Swiss Roll and the non-developable heated Swiss Roll.

*Figure 46: Swiss Roll Graph - Regular, Thin, Heated*

Euclidean distance preserving techniques do a very poor job at unrolling the Swiss Roll Data and its variants. CCA is the best and tears the Swiss roll as it unfolds it, see Figure 47.

*Figure 47: CCA Tearing Swiss Roll*
Graphical distances show their benefit (Figure 48). Isomap, CDA, and GNLM all outperform their spatial distance counterparts and successfully unroll the data. In the creation of the data set 5000 points were made available. The graphs on the left were formulated using 8000 randomly selected points. The graphs on the right were mapped using 600 prototype points attained with vector quantization. Techniques that use graphical distances to estimate geodesic distances have a tendency to make denser data regions denser and sparse areas sparser. Looks like foaming or bubbling of the data. This is due to errors in using graphical distances to estimating geodesic distances worsen when data are farther apart. Enough data to perform VQ is preferable.

Figure 48: (left) Graphical Preservation Techniques with Swiss Roll Data (800 Randomly Selected Points), (right) Non-Euclidean Distance Preservation Techniques Swiss roll data (600 VQ Prototype Vectors)

Topological preservation methods do not do a great job at unfolding the Swiss roll (Figure 49 & 50). SOM and GTM tore the Swiss roll when unfolding it. LLE preserves local topology very well. Isotop performs the best, preserving local topology and keeping the general shape.
Figure 49: Topological Preservation Techniques Swiss Roll Data Set

Figure 50: Spectral or Layered Method Embedding of the Swiss Roll Data Set into 3 Dimensions
Drawbacks of using graphical techniques become more apparent with the thin Swiss Roll data set (Figures 51 & 52). Points in the thin direction are more accurately represented while points on the long distance are poorly represented as the graph tends to zigzag more in their estimation. This causes the graphs to stretch. Since CDA and GNLM tend to focus on preserving close distances, their embedding’s twist.

Figure 51: Graphical Distance Preservation Techniques of Thin Swiss Roll Data Set

Topological preserving techniques do an overall poor job. As can be seen from the SOM, it contains discontinuities. No graph gives a convincing job.

Figure 52: Topological Preservation Techniques on the Thin Swiss Data Set
The Heated Swiss Data set is of particular interest to us since it is a non-developable manifold. We expect to encounter non-developable manifolds in practical applications. CDA performs remarkably well (Figure 53).

![Graphical Distance Preservation Techniques of Heated Swiss Roll Data Set](image1)

Figure 53: Graphical Distance Preservation Techniques of Heated Swiss Roll Data Set

Topological preserving techniques do an overall poor job (Figure 54). As can be seen from the SOM, it contains discontinuities. No graph gives a convincing job. As for the thin roll, Isotop performs the best. A theme begins to emerge. The data-driven methods (LLE, IT, LE) outperform the predefined lattice techniques (SOM and GTM).

![Topological preservation techniques of Heated Swiss Data Set](image2)

Figure 54: Topological preservation techniques of Heated Swiss Data Set
A.3.2 Japanese Flag

The Japanese flag manifold contains hole in the center of map (Figure 55). Graphical distance methods are necessary to unfolding the non-convexity. The Japanese Flag Data set is interesting as it contains a non-convex region. Issues of graphical distances are exacerbated around the nonconvex region. CDA performs well. SDE Once again performs the best.

Figure 55: (left) Japanese Data Set, (right) Euclidean and Geodesic Distance Differences

Topological Preserving techniques do an overall poor job (Figure 56 & 57). As can be seen from the GTM and SOM, it is contains discontinuities. LLE partially unfolds the flag. As for the Japanese Flag, LLE and Isotop perform the best. Laplacian Eigenmaps has similar shape as Isomap due to the fact that commute time distances have the same overestimation error as graph distances.

Figure 56: Graphical Distance Preservation Techniques of Japanese Flag Data Set
A.4 Conclusions

Euclidian distance preservation techniques such as MDS, GNLM, and CCA do a poor job at handling nonlinearities. However, CCA is the best. On the other hand, preserving geodesic distances significantly improve spatial distance preservation techniques; see Isomap, GNLM, and CDA. CDA is particularly robust having the ability to tear unwanted vertices. However geodesic distances are estimated by graph distance which introduces an error. Graphical distances approximations may fail in real situation if there are not enough available points. Also, an error is added when new data points are embedded. Non-spectral methods such as CDA and SOM allow for more flexibility in creating the costs function being minimized because there is not the goal of reaching an EVD problem such as GNLM and LLE or LE.

Topological preservation techniques might be the most useful. They do not rely on graphical estimations of geodesic distances, but are flexible enough to capture nonlinearities. Data driven lattices such as LLE and Isotop outperform predefined lattices like SOM and GNLM. Isotop is of particular interest since it does not rely on graphical distances, is a data-driven lattice, non-spectral method for a flexible objective function, and has been shown to capture nonlinearities well.
APPENDIX B: PERMISSION TO PUBLISH

The works (Robertson, 2011) and (Robertson, 2014) were published in Computers and Chemical Engineering. A response was given to an email pertaining to the rights as:

Dear Greg,

Permission is covered by the rights you retain as an Elsevier journal author as outlined at http://www.elsevier.com/journal-authors/author-rights-and-responsibilities, which include inclusion in a thesis or dissertation, provided that proper acknowledgement is given to the original source of publication. Should you require any further clarification, please let me know.

Best of luck with your thesis.

Thank you,
Laura

Laura Stingelin
Permissions Helpdesk Associate
Global Rights Department
Elsevier
1600 John F. Kennedy Boulevard
Suite 1800
Philadelphia, PA 19103-2899
T: (215) 239-3807
F: (215) 239-3805
E: Lstingelin@elsevier.com

Questions about obtaining permission: whom to contact? What rights to request? When is permission required? Contact the Permissions Helpdesk at:
+1-800-523-4069 x 3808 (US)  permissions.helpdesk@elsevier.com

The permission to use this material in this thesis can be found in Elsevier’s Author Rights and Responsibilities (http://www.elsevier.com/journal-authors/author-rights-and-responsibilities). Here it states:

<table>
<thead>
<tr>
<th>Authors can use either their accepted author manuscript or final published article for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use at a conference, meeting or for teaching purposes</td>
</tr>
<tr>
<td>Internal training by their company</td>
</tr>
<tr>
<td>Sharing individual articles with colleagues for their research use* (also known as 'scholarly sharing')</td>
</tr>
<tr>
<td>Use in a subsequent compilation of the author’s works</td>
</tr>
<tr>
<td>Inclusion in a thesis or dissertation</td>
</tr>
<tr>
<td>Reuse of portions or extracts from the article in other works</td>
</tr>
<tr>
<td>Preparation of derivative works (other than for commercial purposes)</td>
</tr>
</tbody>
</table>
VITA

Gregory Robertson is a Louisiana native. Pre-collegiate interest in broad range of science and mathematics subjects was aided by a bevy of committed underappreciated teachers. Chemical engineering was a natural fit for more develop as it spans so many scientific areas and utilizes them for the better use of mankind.

After receiving an undergraduate in December 2008, he began doctoral research. His graduate research performed has been an effort to understanding complex mathematical concepts and applying them to practical problems.

His next step is to take the research performed and apply them to industry. He plans to look for opportunities to apply the depth of understanding to problems encountered by engineers.