Spring 5-1-2008

Optimization of Production, Maintenance, Design and Reliability for Multipurpose Process Plants: an Analysis and Revision of Models

Christopher L. McGann

1 Introduction

1.1 Motivation

The design and operation of an optimal chemical or biochemical process is a priority for chemical engineers. It is the duty of a chemical engineer to build the most efficient process possible and therefore, seek to limit the characteristics of a process that will decrease productivity while augmenting the characteristics that increase productivity. In the development of a chemical or biochemical process the selection of an optimal combination of resources is of extreme importance to the engineer. An appropriate selection of process units combined with an effective production plan allows for high process efficiencies, which translates into greater profits. Therefore, process optimization is a main concern for chemical engineers at the design stage. As a result of this importance, a great deal of research in the chemical engineering field is devoted to developing methods to optimize plant processes.

There are many ways to optimize a chemical process; one could use a cost-benefit analysis or another similar type of qualitative analysis. However, with the advent of computers it has become easier to solve complex mathematical problems. It is now possible to analyze problems quantitatively that could not be analyzed as such in the past. Therefore, mathematical programming has become a useful tool for modeling and optimizing chemical processes. It has the advantage of producing a quantitative picture of a chemical process that can be analyzed in scientific terms thus removing the subjectivity of a qualitative analysis. A set of equations, variables, parameters and constraints provides the means for modeling a chemical process mathematically. This model can then be solved in order to maximize certain components of the process such as production of a product while minimizing others such as the production of a waste by-product.

Multipurpose plants can especially benefit from process optimization. In these process plants different products or different batches of the same product may take different pathways through the plant depending upon the availability of process units. An example of this could be a paint process where different batches of the same color are blended in different mixers due to some production constraint. The resulting paint is the same for each batch, but not all the batches used the same process pathway. The multipurpose process plant is the type of process that is analyzed in this research paper.

1.2 Problem Definition

The problem that this type of research addresses can be defined as: how can a chemical or biochemical process be best modeled and optimized through mathematical programming? However, the scope of that question is quite large for this research paper. Instead, as will be seen, the focus of this research paper can be summarized as: how can initial reliabilities be best allocated through mathematical programming and optimization? To this end, a model from a specific paper is analyzed and then revised.

1.3 Objective

This research analyzes the mathematical modeling and optimization of the production planning, maintenance scheduling, design and reliability of generic multipurpose process plants. The models of the multipurpose process plants are formulated as mixed-integer linear programming (MILP) problems in the General Algebraic Modeling System (or GAMS) developed by the GAMS Development Corporation[1]. The models are then solved using two different types of solvers on the NEOS Servers[2]. Two separate published works and the respective models contained in each are considered in this research. The first work was put forth by E.N. Pistikopoulos et. al.[3] and contains the basic framework from which all the other models discussed in this research are based. However, it does not take into account initial reliability as a potential variable. The second published paper considered in this research was put forth by H.D. Goel et. al.[4] as a revision of the E.N. Pistikopoulos et. al. paper. This paper includes the addition of the initial reliability allocation as a variable that can be optimized. It is this second paper that is of special importance. This research paper introduces a revision of the mathematical formulation that H.D. Goel et. al. uses to describe the initial reliability allocation. The revision introduced in this work is more elegant mathematically and decreases the resources used by the solver. As such it is an improvement on the H.D. Goel et. al. model.

1.4 Outline

This research paper has several goals: first, there is an in-depth analysis of the characteristics as well as the methods associated with the type of process optimization research used for these models. Section 2 is devoted to this goal. Second, there is a detailed description of the mathematics of each of the models in Section 3. This covers the two models introduced in the E.N. Pistikopoulos et. al. paper, the model H.D. Goel et. al. develops and the revision of the H.D. Goel et al. introduced in this paper. Third, in Section 4, there is a detailed description of the results of the reproductions of the three published models as well as the results from the new revision model. Fourth, in Section 5, conclusions are drawn and possible avenues for future research are discussed.

2 Literature and Methods Review

This section contains a review of the theory and the methods that form the framework of the models. The models from the two published works that are analyzed in this paper have common theoretical frameworks that must be described in order to better understand how the models represent a process. There are certain assumptions and methods used by those who are involved in the research of this particular type of process model optimization. A discussion of these assumptions and methods will illuminate the limitations as well as the strengths of the models developed in this vein of research.

2.1 Optimization and Mathematical Modeling

The underlying assumption that forms the foundation of this particular type of research is that processes can be modeled mathematically and optimized through the maximization or minimization of an objective function. In other words, the assumption is that one can represent a chemical process through a system of interrelated equations, variables and parameters, which then can be optimized by solving one of the equations for a maximum or minimum value. As discussed previously it is of extreme importance to the design engineer that a chemical process be an optimal one. An optimal process, by its very nature, is a process that contains high efficiency and therefore, costs less and produces more. In any engineering discipline, whether it is chemical or mechanical or electrical, it is the goal of the engineer to design for efficiency. As a result of this there is a tremendous amount of interest, in terms of research and otherwise, into not only designing optimal chemical processes, but also finding ways to design optimal chemical processes. The latter, of course, is the focus of this research.

In the research discussed in this paper the optimization of chemical processes is accomplished by creating and solving a mathematical model. In general, models are thought of as something tangible such as a model airplane, but models can also be abstract. This is the case of a mathematical model, which is a collection of algebraic symbolism that represents the internal relationships of the entity that is being modeled (for the purposes of this paper the "entity" is the chemical process). The characteristics of a process such as the movement of material or the uptime of a unit are modeled through the use of equations, inequalities and logic[5]. For example, the law of conservation of mass demands that matter be neither created nor destroyed through normal chemical means. To ensure that the mathematical model does not violate this law a material balance equation is used as a constraint. In the mathematical model, the equations that serve as constraints dictate limitations of a particular process and therefore, the limitations of a possible solution. These equations may take the form of equalities or inequalities depending on the process relationship. These equations are composed of scalars and sets of parameters and variables, which are organized through the use of indices. The value of a scalar will always be constant, while the value of a parameter may change based on the index. The value of a variable is free-floating and will be determined by the optimization of the model[5]. Finally, there is the objective function which optimizes the model by being solved for a maximum or minimum value. However, how does finding the maximum value (or minimum value) of the objective function optimize the model? The objective function organizes a portion or all of the variables used in the model into some universal mathematical relationship. By finding the maximum or minimum value of that relationship, different values for the variables are chosen by a solver. These values are considered to be optimal as they provide the largest or smallest value to the objective function[5]. The mathematical process models discussed in this work are all optimized using a similar objective function. A profit equation is maximized in order to determine the optimal production, maintenance, design and reliability characteristics of the process. The total costs associated with the model, such as the fixed costs of the process units, are subtracted from the total revenue of the process. The total revenue is a function of the amount of product produced multiplied by the price of that product. The actual math associated with these models as well as the important characteristics of this type of process optimization such as state-task networks are discussed in further detail in later sections[3].

2.2 Linear Programming, Integer Programming and MILP

There are different types of mathematical models, but there are three main classifications: linear programming models, non-linear programming models and integer programming models.

Linear programming (LP) models are the simplest of all the mathematical programming models since they only contain linear equations. In these models there is no multiplication or division or other non-linear combination of variables. A variable in a linear programming problem can only be either added or subtracted from another variable. Parameters and scalars, however, are not restricted. On the other hand non-linear programming models (NLP) will contain non-linear functions such as X_1X_2 or $X_2/(X_1X_3)$[5].

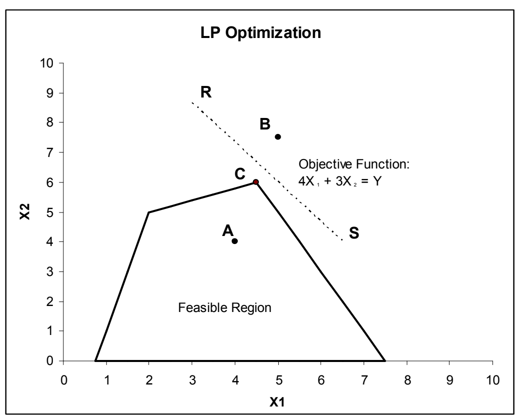

Fig. 1 Graphical Representation of LP Optimization

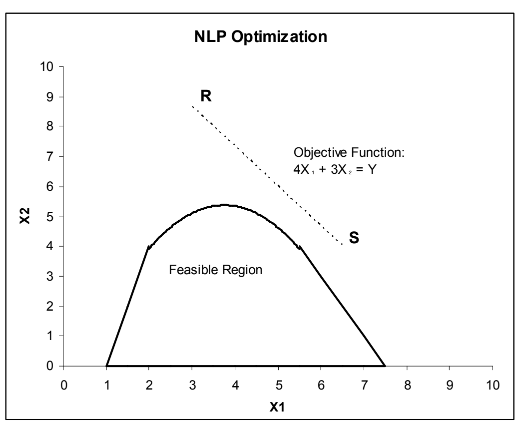

Fig. 2 Graphical Representation of NLP Optimization

The consequences of a linear model can best be described geometrically. Figure 1 shows the axes of two variables and the constraints of a model depicted as boundaries forming the model's feasible region. It is inside the feasible region that a feasible solution for the model can be discovered; the points inside the region satisfy all of the constraints. Figure 2 demonstrates a similar situation, but with a non-linear constraint that results in one of the boundaries becoming curved. The significance of this curvature becomes clear when both models are optimized. If an objective function $4X_1 + 3X_2 = Y$ is to be maximized for Y then the slope RS representing the function will have to come in contact with the feasible region, but only on a boundary. In this way the variables would equal a point in the feasible region, satisfying the constraints, but the solution of the objective function would be at a maximum[6]. Points A, B and C demonstrate different values of the objective function. Point A is inside the feasible region satisfying the constraints but it is not the maximum number. Point B is outside the feasible region and although it is the largest number, it is not a solution. Point C is the only point that is a maximum and satisfies constraints. Point C gives an optimal solution and does so very clearly; it is right on a vertex. The curvature in the non-linear model creates a difficult situation. A vertex is very easy to distinguish, but finding the maximum point on a curve either requires iteration or calculus. Therefore, non-linear problems are more difficult to solve. This should illustrate the simplicity of finding a solution for linear programming problems, which usually means a faster solving time for computerized models. For this reason, it is important solve linear models.

Integer programming (IP) is a very useful form of modeling that uses integer values and linear constraints. IP has a surprisingly wide range of uses; if a particular variable had to represent an indivisible unit such as a person or a car then integer programming could be used to optimize that particular model. The resulting value after optimization for the number of people or cars would be a whole number in an integer programming model. The usefulness of integer programming becomes readily apparent when the integer variables are limited to either a 0 or 1 value. In effect, the integer variable becomes a binary variable, which allows for logical constructions in the form of "yes" or "no" decisions. In fact, these logical constructions are essential to the models discussed in this research. Preventive maintenance, process design and initial reliability allocation would be impossible without these logical underpinnings. The IP models become extremely complex very quickly due to the way in which they are solved[5].

One of the more successful methods, the branch and bound method illustrates the complexity. First, the model is relaxed as if it was a normal LP model and a solution is determined. Then a systematic search begins as the solver attempts to find a suitable integer solution. The solver algorithm creates a tree of submodels of the original model and also introduces new constraints. These new constraints attempt to force integer values for the IP variables. If a feasible solution exists the solver will eventually find an integer solution for the model. Unfortunately this could take a long time, but when the integer variables are binary variables then the process is becomes much easier[6].

The last type of mathematical model is used in the models described later in this paper and is actually a combination of the two models above. The mixed-integer linear programming model combines continuous variables and integer variables. Essentially, this is an integer programming model with added relaxation due to the continuous variables. In fact, IP is divided into two different categories: pure integer programming (PIP) and mixed-integer programming (MIP) [5]. MILP models are extremely useful when optimizing processes because there are certain aspects of a process that can be "selected" using the logical constructions only an IP model can make available. However, there are other characteristics of a process where it is better to have a continuous variable.

2.3 Multipurpose plants and State-Task Networks (STN)

The research that is conducted in this paper deals with multipurpose process plants with time horizons that are divided into periods. These periods serve a specific importance to be discussed in the next section. As mentioned in the Introduction, multipurpose process plants are plants that produce a variety of products through a number of different but interrelated pathways. This is different from a multiproduct plant where a number of different products produced all follow the same sequence or pathway. In multipurpose plants, process units share the burden of producing different products. For example, returning to the paint process mentioned in the introduction, consider a situation where there are four blends of paint and four mixers. However, there is always one mixer that is offline due to maintenance. In this circumstance one or all of the mixers might alternate between the task of blending different paints in such a way that certain levels of demand for all four paints continues to be met. Under these conditions, the task the process unit undertakes becomes more important to understanding the process than the unit itself. Therefore, it no longer makes sense to use the traditional flowsheet representation of a chemical process. Instead a state-task network (STN) is introduced[7].

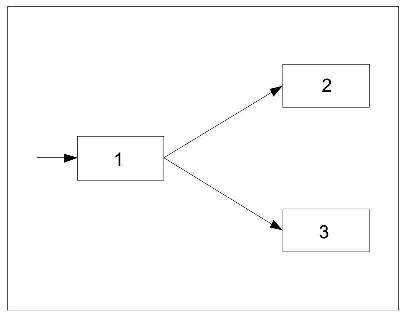

Fig. 3 Normal Recipe Representation of a Chemical Process

E. Kondili et. al. introduces and discusses the benefits of a STN representation when compared to a normal recipe representation of a chemical process. Figure 3 is a slightly modified reproduction of a figure in their work. E. Kondili et. al. points out the difficulty with determining whether Task 1 produces a state that gets divided between Task 2 and 3 or that it alternatively produces two separate states that wholly get sent to Task 2 and 3. However, in an STN representation of the process there is no confusion as depicted by Figures 4 and 5, which are also modifications of figures from the E. Kondili et. al. paper. In Figure 4, Task 1 clearly produces only one state and in Figure 5, Task 1 produces two separate states. The STN is made up of two different nodes: state nodes (circles) and task nodes (squares). A state is any material that exists in the process; it can be a feed or a process intermediate or a final product. A task is any operation that exists in the process.

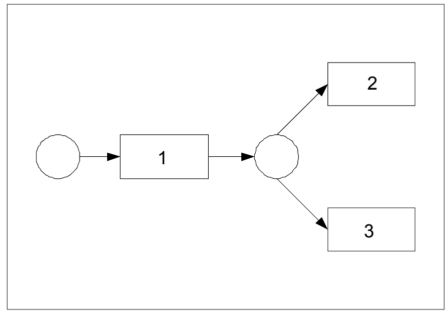

Fig. 4 STN Representation – Version 1

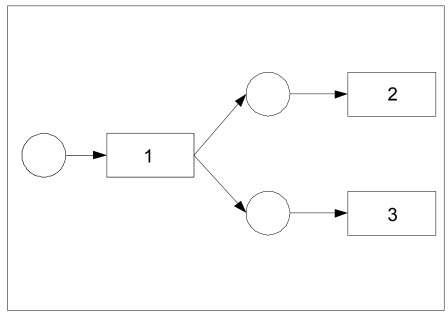

Fig. 4 STN Representation – Version 2

In their paper, E. Kondili mentions that an STM is suitable for any type of operation whether it's continuous, semicontinuous or batch. They also provide these two rules for the construction of a STN:

> *1. A task has as many input (output) states as different types of input (output) material.*
>
> *2. Two or more streams entering the same state are necessarily of the same quality. If mixing of different streams is involved in the process, then this operation should form separate task*[7].

The math involved in a STN will be discussed in greater detail when the models are introduced but it is sufficient to note that the connection between states and tasks are defined by two sets. The first delineates which states serve as input for a given task and the second delineates which states serve as an output for a given task. Also, the models include two sets that are the exact opposite of the two previously stated sets (i.e. set of tasks capable of receiving material from a given state). Process units are still considered in these process models and are described by a set that groups units capable of performing a given task[7].

2.4 Availability, Time Periods, Process Unit Failure Rate and Maintenance

The scheduling of plant production cycles is a major concern for field chemical engineers of multipurpose plants. It is critically important to match demand with supply especially in industries such as the manufacture of foodstuffs or other perishables. However, it is not always possible to predict demand in the long-term. When this is possible plant management normally dedicates all the resources of the plant to a single product so as to minimize costly changeovers. However, when the demand for a particular product fluctuates in the short-term it becomes difficult to run the plant in "campaigns" and therefore, the multipurpose plant must adjust its production planning so that it can produce all necessary products without having too much downtime for the units[7]. This might result in a situation similar to the paint process described in the previous section where a process unit shares the burden of a particular task. Regardless, the plant must maximize the utilization of all its assets. However, the extent to which the plant can maximize asset utilization is directly related to the availability of the process units[3]. Availability is defined as the "fraction of time over a defined period (the mission time) that a component or system is fully operational[8]." Furthermore, the availability of a process unit is directly related to the unit's maintenance schedule as well as its failure rate (i.e. the unit's reliability).

In the researched process models the reliability of a process unit is described by an increasing failure rate that affects the availability in two ways. First, the unit could fail and have to be fixed; this is corrective maintenance. Secondly, the failure rate plays a role in how the unit is maintained so that it doesn't fail; this is preventive maintenance. Both forms of maintenance cut into the amount of time a particular process unit is available to complete a task[3].

The failure rate used in the models of the research report is regarded as the number of failures for a given time period. The mathematical representation of this is:

$$\overline{N}_{jT} = \int_0^T \lambda_j(s)\, ds$$

where $\overline{N}_{jT}$ stands for the expected number of failures and $\lambda_j(s)$ is the number of failures per unit time. The expected number of failures is simply the failure rate integrated across the time period T [3]. If the failure rate is constant as it is in process models then the equation simplifies down to the failure rate multiplied by the time period. This makes perfect sense:

$$\overline{N}_{jT} = \lambda_j(T - 0) = \lambda_j T$$

However, it has to be noted that this does not always result in a whole number of failures. In fact, it is quite possible for there to be only a fraction of a failure for a given time period. This can be confusing: how can there only be a fraction of a failure? A process unit either fails or it functions normally. While this is true it is not easily replicated mathematically. The very nature of failure is that its occurrence is uncertain. One cannot predict exactly when a failure will occur. However, the likelihood or rate at which a unit fails can be ascertained through statistical analysis. A good way to think about the failure rate is that it is how the model accounts for that future failure; it does so a little portion per period. This concept will become clear when the relationship between the failure rate and other parts of the model is discussed.

In the process models researched in this work it is assumed that the failure rates increase at constant rate. A constant rate of increase was assumed because it simplified the mathematical model. The time horizon considered in these models is broken up into equal length time periods. For example, if the time horizon were two years then the time period could potentially be 104 weeks or 24 months. The time period is important to the scheduling of production as well as the scheduling of maintenance. However, for now, it is sufficient to recognize that the failure rate increases for each time period, but that the failure rate is also constant for that time period. Figure 6 demonstrates this graphically[3]. The γ_{j1} represents the failure rate for unit j during time period 1. The horizontal line just beneath the symbol represents that failure rate. This demonstrates how the failure rate stays constant until the maximum allowed periods before maintenance represent by τ_j.

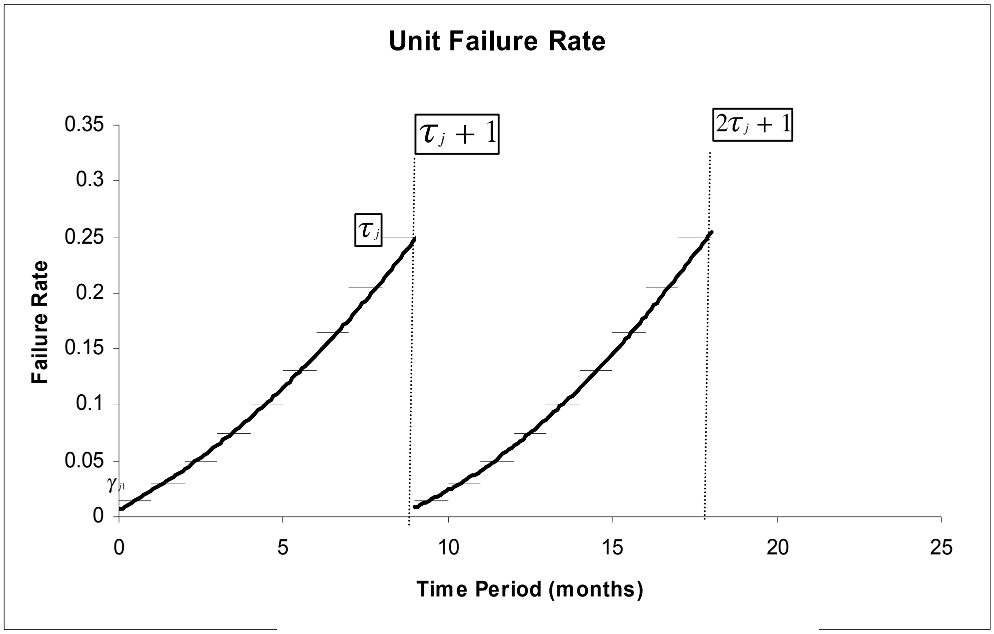

Fig. 6 Graphical Representation of Unit Failure Rate

Maintenance actions play an important role in the failure rates of a given process unit. When a preventive maintenance action is taken the failure rate is returned to its original value or the value it has when it is "new." This is a very important assumption made in the model; a preventive maintenance action will return the model to as-good-as-new conditions (AGAN). The graph repeats itself to represent how the failure rate is AGAN after maintenance. If failure occurs then only corrective maintenance action is taken and the unit is returned to (AGAO) conditions[3]. However, this idea of corrective maintenance is not as explicit mathematically as preventive maintenance because of the ideas describe earlier in this section. Finally, there is a maximum number of periods that can pass between preventive maintenance periods before another action must return the unit to AGAN conditions[3].

2.5 Design and Initial Reliability

The mathematical process models considered in this work have two remaining components: design characteristics and the initial reliability characteristics. A crucial aspect of process optimization in multipurpose plants is determining the number of units that are necessary to carry out all of the tasks in a way that is profitable. Sometimes it might make sense to include duplicate units if maintenance scheduling puts a particular unit out of commission. It is a common practice for process engineers to include back-up units or design the process to have several duplicate units working below full capacity so that even if one malfunctions there is still enough capacity in the remaining units for the process to continue normally. Since the selection of units plays an important role in process design it is also included in these process optimization models[3]. Furthermore, the size of a particular process unit is also variable in these models. The inclusion of the initial reliability allocation feature stems from the fact that some process units may have lower failure rates than others[4]. This may be due to better design, superior materials or some other advantage not present in the generic process unit. However, in the models discussed in this work the increased initial reliability comes at a higher initial capital cost. The method of initial reliability allocation is the main focus of the experimentation aspect of this report.

3 Process Models Mathematics

In the following sections the mathematics of the models from the two published works are presented and discussed. Four models are discussed in this section, but each model builds on the previous either by increased complexity or revision. The four models are the Multiperiod Production Planning and Maintenance Model and the Simultaneous Design, Production, and Maintenance Planning Model proposed by E.N. Pistikopoulos et. al., the Design, Reliability, Production and Maintenance Planning model proposed by H.D. Goel et. al. and the Revision of the Design, Reliability, Production and Maintenance Planning model, which is introduced in this project.

3.1 Multiperiod Production Planning and Maintenance Planning Model

This model encompasses two components: the first is a multiperiod production component and the second is the maintenance scheduling component. The purpose of the first is to actually describe the process: the flow of materials, the number of batches a unit produces, the capacity of the process units and the demand for products. The second part of the model describes the maintenance schedule and also determines how the failure rate affects the uptime of the process units.

Appendix A summarizes the mathematics that is used in all four models that are to be discussed and therefore, only a description of the importance of each equation will be discussed here:

3.1.1 Model Structure

The fundamental aspect of the math behind the process models is the index. The index decreases the complexity of the model by allowing a few general variables or parameters to represent all the different aspects of the process model. For example, let the index j represents all of the different units that can exist in the model and let the variable V represents the size of a unit. If there are several different units then there has to be several different V variables. Instead of more letters a subscript is used to denote the different unit size variables: V_j. This decreases the complexity of the model and neatens the mathematics. The use of indexes becomes increasingly important when a certain variable has 3 or 4 indexes.

The next building block in the model framework is the set. The set serves to organize the model as well as limit its size. For example, there may be ten different process unit sizes that exist in the model, but only four maybe appropriate for a particular unit. A set could be formed that contains only the four available sizes for the given process unit. The addition of an index would make this notation even simpler as a different symbol for each different process unit would not be needed just a subscript j denoting the unit: ψ_j. The result is a set of all possible sizes for unit j.

There are six indices used in the first model representing the process tasks i, the process units j, the time periods t, the states of material s, the utilities u and the number of periods since the last maintenance action θ. There are also 6 sets representing the set of states that are consumed or produced by task i (S_i, $\overline{S}_i$), the set of tasks consume or produce state s (T_s, $\overline{T}_s$), the set of tasks for which unit j is suitable (I_j) and the set of units for which task i is suitable (K_i).

3.1.2 Resource Utilization Constraint

$$\sum_{i \in I_j} p_i N_{ijt} \leq U_{jt} \qquad \forall j, t$$

This inequality restricts the number of batches N_{ijt} of a particular task produced by a unit in a given time period. The inequality states that the total time spent completing tasks can not exceed the uptime U_{jt} of the unit for a given period. The total time spent completing tasks is found by summing the number of batches for a given task multiplied by the processing time p_i of the batch for a given task.

3.1.3 Capacity Constraints

$$\phi_{ij}^{\min} V_j N_{ijt} \le B_{ijt} \le \phi_{ij}^{\max} V_j N_{ijt} \quad \forall i, j \in K_i, t$$

This inequality limits the amount of material B_{ijt} that can be processed by a particular unit in a given time period for a given task. The simplest function of this equation states that the amount of material processed for a given task-unit combination has to be less than the size V_j of the unit multiplied by the number of batches for a given task-unit combination. The utilization factors, $\phi_{ij}^{\min}$ and $\phi_{ij}^{\max}$, simply complicate the capacity constraint by further limiting the feasible region for B_{ijt}.

3.1.4 Material Balance Constraints

$$S_{st} = S_{s,t-1} + \sum_{i \in T_s} \sum_{j \in K_j} \bar{\rho}_{is} B_{ijt} - \sum_{i \in T_s} \sum_{j \in K_j} \rho_{is} B_{ijt} - D_{st} \quad \forall s, t$$

The material balance constraint ensures that amount of material present in the process remains constant (i.e. mass is neither created nor destroyed). This equation can be broken down into three different components the input, the output and the remainder. The remainder is depicted by the left hand side of the equation and is simply the amount of state that is stored in a given time period (S_{st}). This is equal to the input minus the output. The input is the amount of material stored in a given state for the previous time period ($S_{s,t-1}$) and the amount of material that is processed into that state in the current time period ($\sum_{i \in T_s} \sum_{j \in Kj} \bar{\rho}_{is} B_{ijt}$). The output is the amount of the given state that is processed into a different state ($\sum_{i \in T_s} \sum_{j \in Kj} \rho_{is} B_{ijt}$) and it is the amount of the given state that is delivered to the customers (D_{st}). At certain points in the analysis of these models a sixth term is added to this equation to account for the initial amount of state in time period 1 as will be seen later. The summation terms may be difficult to understand. However, $Bijt$ is only the amount of material processed for a given task-unit-time period combination, but the given state may be produced or processed in other task-unit-time period combinations. Therefore, the model must "find" and sum up all the other places that the state can be produced or processed.

3.1.5 Demand Constraints

$$D_{st}^{\min} \le D_{st} \le D_{st}^{\max} \quad \forall s, t$$

The demand constraints simply state only so much or so little of a given state can be delivered to the customers in a given time period.

3.1.6 Utility Constraints

$$\sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i - 1} \beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt} \le A_{ut}^{\max} H \quad \forall u, t$$

The utility constraints are actually not used in this research, but they simply limit the amount of batches that can be produced and the amount of material that can be produced due to plant restrictions such as the availability of process water or heat.

3.1.7 Failure Rate Constraints

The failure rate constraints are the one of the more complicated parts of the process models, but they are essential in the determination of the maintenance scheduling as well as the uptime of the process unit. As was stated previously, there is maximum number of periods τ_j that may pass before a preventive maintenance action is taken. Therefore, the index θ, which denotes the number of periods since the last maintenance period, has to be less than or equal to τ_j. However, it is assumed that maintenance always takes place at the beginning of a period and therefore θ never equals zero. As θ increases the failure rate increases until the unit is preventively maintained or θ equals τ_j and preventive maintenance is forced. At this point the $\gamma_{j\theta}$ parameter must be discussed. This parameter is the failure rate value for the unit θ periods since the last preventive maintenance. It is this parameter that determines how the failure rate increases from period to period. For example, a process unit could have a failure rate of 0.002 when θ is equal to 2 and then when θ is equal to 3 the failure rate could be 0.004. It must be noted that this

$\gamma_{j\theta}$ is different from the continuous variable λ_{jt} which is the actual failure rate of the unit during a given time period. The last two important components of the failure rate constraints are the binary variables: X_{jt} and $Z_{jt\theta}$. The X_{jt} is equal to 1 when preventive maintenance occurs during the time period and equal to 0 when it does not occur. The $Z_{jt\theta}$ is equal to 1 during a given period when maintenance was last performed θ periods ago. When one of these variables is equal to 1 it can be thought of as "yes" decision while 0 would be a "no" decision. The following are the failure rate constraints:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} \qquad \forall j, t$$

$$Z_{jt\theta} \leq X_{j,t-\theta} \qquad \forall j, t, \theta = 1, \ldots, \tau_j$$

$$\sum_{\theta=1}^{\tau_j} Z_{jt\theta} = 1 \qquad \forall j, t$$

The first equation is responsible for actually reporting the failure rate while the next two equations are responsible for making sure that only one $\gamma_{j\theta}$ is chosen. The first equation is the summation of each potential failure rate multiplied by the $Z_{jt\theta}$ decision variable. Only one of these $Z_{jt\theta}$ will be equal to 1 and therefore only one $\gamma_{j\theta}$ will be chosen. To prevent there from being more than one "yes" for $Z_{jt\theta}$ in a given unit-time period combination the third equation stipulates the summation of the $Z_{jt\theta}$s must be equal to 1. Therefore, only one $Z_{jt\theta}$ for a given unit-time period (*jt*) can be equal to 1. The second constraint is how the model ensures that when any maintenance action (X_{jt}) was last taken it is accounted for in future $Z_{jt\theta}$ variables because only the $Z_{jt\theta}$ with the correct θ will be equal to 1.

3.1.8 Uptime Definitions and Constraints

The uptime constraints further confuse and complicate matters, especially as there are multiple definitions of process unit uptimes. The uptime of a unit is the period of time that the unit is expected to be perform its normal functions. Therefore, the expected number of failures as well as the preventive maintenance schedule play an important role in determining how much uptime there will be for a given process unit. The time devoted to corrective and preventive maintenance must be subtracted from the total period time. However, this can be done in a three different ways:

Case 1: Failure can occur at all times: during both minimal repair and preventive maintenance.

Case 2: Failure cannot occur during preventive maintenance.

Case 3: Failure can not occur during minimal repair or preventive maintenance.

The equations are listed in order according to its respective case:

$$U_{jt} = H(1 - \Delta_j^c \lambda_{jt}) - \Delta_j^p X_{jt} \qquad \forall j, t$$

$$U_{jt} = (H - \Delta_j^p X_{jt})(1 - \Delta_j^c \lambda_{jt}) \qquad \forall j, t$$

$$U_{jt} = (H - \Delta_j^p X_{jt}) / (1 + \Delta_j^c \lambda_{jt}) \qquad \forall j, t$$ [3]

In Case 1, the total uptime U_{jt} is equal to the total duration of the time period H subtracted by the time devoted to corrective maintenance $H\Delta_j^c \lambda_{jt}$ minus the time devoted to preventive maintenance $\Delta_j^p X_{jt}$ if it occurs. The significance of the binary variable is that if the preventive maintenance does not occur then that term is equal to zero. In the second term the $H\lambda_{jt}$ component provide the number of failures for the time period and the Δ_j^c component is simply the amount of repair time per repair. Similarly the Δ_j^p parameter is the time it takes for preventive maintenance to take place.

However, Case 1 is only accurate if failure can occur during maintenance. Case 2 is needed if failure can not occur during preventive maintenance. The math in this situation is very similar but there is an extra term : $\Delta_j^p X_{jt} \Delta_j^c \lambda_{jt}$. This term adds back any repair time that would have taken place during preventive maintenance. This is simply done by multiplying the preventive maintenance time by the failure rate/repair time combination and then adding it back to the equation. This might be confusing, but if the $\Delta_j^p X_{jt}$ component acts in the same way that H does when multiplied against the $\Delta_j^c \lambda_{jt}$[10].

In Case 3, there is no failure during repair or maintenance duties. This scenario requires a little more sophisticated math. According to the simplified failure rate equation given in section 2.4 ($\overline{N}_{jT} = \lambda_j(T - 0) = \lambda_j T$), the number of failure is equal to the failure rate multiplied by the length of time. Now if the T is just the time period minus the repair and maintenance time then the following equation can be derived:

$$T = H - \varDelta_j^c \overline{N}_{jT} - \varDelta_j^p X_{jt}.$$

Upon the substitution of $\overline{N}_{jT}$ the following equation is derived:

$$T = H - \varDelta_j^c \lambda_{jt} T - \varDelta_j^p X_{jt}.$$

The variable T is simply the uptime of the unit for a given period:

$$U_{jt} = H - \varDelta_j^c \lambda_{jt} U_{jt} - \varDelta_j^p X_{jt}$$

and when the equation is solved for the uptime then the Case 3 equation can be derived:

$$U_{jt} = \frac{H - \varDelta_j^p X_{jt}}{1 + \varDelta_j^c \lambda_{jt}}$$

This final equation clearly shows that the uptime will increase in value [10].

Unfortunately, there is one more complication with the model and that is that the failure rate λ_{jt} is a variable as well as the decision variable X_{jt} and the Case 2 and Case 3 uptime equations are made non-linear. As discussed earlier, non-linear models are harder to solve. However, the following constraints and changes to the Case 2 and 3 uptime equations can linearize the model:

$$\overline{XZ}_{jt\theta} \leq Z_{jt\theta} \qquad \forall j, t, \theta = 1, \ldots, \tau_j$$

$$X_{jt} = \sum_{\theta = 1}^{\tau_j} \overline{XZ}_{jt\theta} \qquad \forall j, t$$

Case 2

$$U_{jt} = H - H\varDelta_j^c \sum_{\theta = 1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - \varDelta_j^p X_{jt} + \varDelta_j^p \varDelta_j^c \sum_{\theta = 1}^{\tau_j} \gamma_{j\theta} \overline{XZ}_{jt\theta} \qquad \forall j, t$$

Case 3

$$U_{jt} = \sum_{\theta = 1}^{\tau_j} \frac{HZ_{jt\theta} - \varDelta_j^p \overline{XZ}_{jt\theta}}{1 + \varDelta_j^c \gamma_{j\theta}} \qquad \forall j, t.$$

3.1.9 Objective Function

The objective function is the most important part of the model. The maximization or minimization of this feature of the model is what leads to the optimization of the process. In the process models that are researched in this paper the objective function is a profit equation that is maximized. The following is the mathematical representation of the profit equation:

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$
$$\sum_{jt} C_{jt}^p X_{jt} - \sum_{jt} C_{jt}^c (H - U_{jt} - \varDelta_j^p X_{jt}) / \varDelta_j^p$$

The profit equation will change as the models increase in complexity, but the basic idea remains the same. The left hand side of the equation shows the maximization of the profit variable $\boldsymbol{\Phi}$. The first term on the right hand side of the equation is the total revenue from all of the deliveries made to the customer. The revenue found from a single delivery is the price η_{st} multiplied by the amount D_{st}. From this, the costs of the process are subtracted. The second term can actually be ignored since it is not used in these models, but it represents the costs associated with the utilities. The third term is the cost associated with the preventive maintenance. It simply multiplies a cost factor C_{jt}^{p} against the decision variable and then summates all of these up for all time periods and units. The final term is the corrective maintenance cost. The terms inside the parenthesis find the time spent on corrective maintenance by subtracting the uptime of the unit and the time spent on preventive maintenance from the total time. This is divided by the corrective maintenance time which would give the total number of corrective repairs, which is then multiplied by a cost factor C_{jt}^{c}. Then all the costs for each period and unit are summated and subtracted from the revenue. The result is an equation for the profit that can be maximized in order to optimize all the variables in process model.

3.2 Simultaneous Design, Production, and Maintenance Planning

This model increases the complexity of the preceding model by allowing the design of the process model to be optimized. This means that there are some revisions of the previous equations but also the addition of new equations and variables. The Simultaneous Design, Production, and Maintenance Planning Model has the ability to choose whether or not units should be included in the process and also the size of these units. However, this also means that the profit equation has to be adjusted to include design cost factors.

3.2.1 Model Structure Additions

First, another index k is added to differentiate between possible unit sizes. A new set $\boldsymbol{\Psi}_j$ of potential unit sizes for a given unit is included and several new parameters are introduced. The $\overline{V}_{jk}$ parameter provides a unit size for a given unit while $N_{ijt}^{\max}$ is used to limit the number of batches for a task-unit-time period. The cost factors K_j^0, the fixed cost for a unit, and K_j^1, the variable size cost factor, are included in the model. Two new binary variables are used as decision variables to choose units, E_j and their size, E_{jk}. Finally, a continuous variable V_j for the unit size is also included.

3.2.2 Design Constraints

The two following equations are the design constraints that respectively choose the units to be included in the process and choose the size of the units:

$$E_j = \sum_{k \in \psi_j} E_{jk} \quad \forall j$$

$$V_j = \sum_{k \in \psi_j} \overline{V}_{jk} E_{jk} \quad \forall j$$

The first equation creates a situation where if a particular unit is chosen (E_j=1) then a size, but only one size, must be chosen from a set of different sizes suitable for the unit. The second equation demonstrates how the chosen size decision variable can be translated into a number. The result from this equation provides a numerical value for the volume of the E_{jk} decision variable. Due to the first equation only one E_{jk} can equal 1 for a given unit and size. As a result, in the second equation only one term on the right hand side of the equation will there be a E_{jk} equal to 1 multiplied by the size (or volume) parameter. The volume parameter will correspond to specific decision variable E_{jk}. As a result of the interaction of these two equations, a particular unit will be designated a particular size (or volume).

3.2.3 Revisions

The addition of a design component to the model requires the revision of some of the constraints. For instance, the capacity constraints are changed to the following due to the addition of the second design constraint:

$$\phi_{ijt}^{\min} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{E}_{jk} N_{ijt} \leq B_{ijt} \leq \phi_{ijt}^{\max} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{E}_{jk} N_{ijt} \quad \forall i, j \in K_i, t.$$

However, this revision creates its new problems as the $\overline{E}_{jk} N_{ijt}$ is non-linear and therefore, it must be linearized in order to keep the model in a MILP form. First, a new variable can be introduced:

$$\overline{EN}_{ijkt} \equiv \overline{E}_{jk} N_{ijt} \quad \forall i, j \in K_i, k \in \psi_j, t$$

along with two constraints to integrate the new variable into the model's math:

$$\overline{EN}_{ijkt} \le N_{ij}^{\max} E_{jk} \qquad \forall i, j \in K_i, k \in \psi_j, t$$

$$N_{ijt} = \sum_{k \in \psi_j} \overline{EN}_{ijkt} \qquad \forall i, j \in K_i, t$$

The capacity constraint then becomes:

$$\phi_{ijt}^{\min} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{EN}_{ijkt} \le B_{ijt} \le \phi_{ijt}^{\max} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{EN}_{ijkt} \qquad \forall i, j \in K_i, t$$

The maximum number of batches $N_{ij}^{\max}$ can be found by:

$$N_{ijt}^{\max} = \frac{U_j^{\max}}{p_i} \qquad \forall i, j \in K_i$$

and $U_j^{\max}$ can be found using the following equations with the first for Cases 1 and 2:

$$U_j^{\max} = H(1 - \Delta_j^c \lambda_{j1}) \qquad \forall j$$

and the second for case 3:

$$U_j^{\max} = \frac{H}{(1 + \Delta_j^c \lambda_{j1})} \qquad \forall j$$

The uptime constraints for Case 1 and 2 are also adjusted:

Case 1

$$U_{jt} = H(E_j - \Delta_j^c \lambda_{jt}) - \Delta_j^p X_{jt} \qquad \forall j, t$$

Case 2

$$U_{jt} = HE_j - H\Delta_j^c \sum_{\theta = 1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - \Delta_j^p X_{jt} + \Delta_j^p \Delta_j^c \sum_{\theta = 1}^{\tau_j} \gamma_{j\theta} \overline{XZ}_{jt\theta} \qquad \forall j, t$$

3.2.4 Maintenance Constraints

Two maintenance constraints are added in order to make sure that preventive maintenance actions only occur if the unit exists:

$$X_{jt} \le E_j \qquad \forall j, t$$

$$\sum_{\theta = 1}^{\tau j} Z_{jt\theta} = E_j \qquad \forall j, t$$

The second equation is an adjustment of the previous equation shown here:

$$\sum_{\theta = 1}^{\tau_j} Z_{jt\theta} = 1 \qquad \forall j, t$$

3.2.5 Objective Function

Finally, the profit equation is adjusted so that the fixed cost and the variable size cost are included. This simply requires the following term:

$$\sum_j (K_j^0 E_j + K_j^1 \sum_{k \in \Psi_j} \overline{V}_{jk} E_j)$$

where the cost factors are included only if the decision variables equal 1. The profit equation now looks like this:

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) - \sum_{jt} C_{jt}^{p} X_{jt} - \sum_{jt} C_{jt}^{c} (H - U_{jt} - \Delta_j^{p} X_{jt}) / \Delta_j^{p} - \sum_{j} (K_j^{0} E_j + K_j^{1} \sum_{k \in \Psi_j} \overline{V}_{jk} E_j)$$

3.3 Design, Reliability, Production and Maintenance Planning Model

This model was introduced by H.D. Goel, et. al. and it increases the complexity of the previous model. The theory behind this new model is not difficult to understand, but it uses a very complex and somewhat cumbersome mathematical framework. This model introduces the concept that the initial reliability of a process unit is a variable that can be chosen by introducing a new decision variable. In the same way a large size process unit has a greater cost; a more reliable process unit will also have a greater cost. The optimization of this model in effect determines whether or not the lower reliability is in fact worth the extra cost. There is a number of different reliability optimization methods present in the literature. These formulations can either focus on the reliability of the process units or on the redundancy of the process units. In this research, an initial failure rate for a unit is selected for a given process unit; a more reliable process unit will have lower initial failure rates.

As with the previous model the introduction of new complexities results in the revision of some of the equations; the reliability model is not an exception. The revisions as well as the new math are discussed in the following sections.

3.3.1 Model Structure Additions

First, a new index l is added to describe the initial failure rates and a new set is introduced ζ_j which denotes the set of possible initial failure rates for a given unit. Several new parameters are added: $\overline{\lambda}_{jl}$ describes an initial failure rate for a given unit, K_{jl}^{2} describes the cost factor for a unit given a specific initial failure rate and α_j describes a failure rate increment given a specific unit. New variables include a decision variable E_j for choosing an initial failure rate given a unit and a continuous variable $\gamma_{j\theta}$ for the failure rate value for a unit given a maintenance action that took place periods ago.

3.3.2 Reliability Allocation Constraints

The following equations are the three new reliability allocation constraints:

$$\gamma_{j1} = \sum_{\theta=1}^{\zeta_l} \bar{\lambda}_{jl} E_{jl} \quad \forall j$$

$$E_j = \sum_{l \in \zeta_l} E_{jl} \quad \forall j$$

$$\gamma_{j\theta} = \gamma_{j,\theta-1} + \alpha_j \quad \forall j, 2 \leq \theta \leq \tau_j$$

The first equation chooses the initial reliability just as the size of a unit is chosen. The second equation makes sure that only one failure rate is chosen and that the unit must exist for that failure rate to be chosen. If the left hand side of the equation is equal to 1 then only one of the initial failure rate decision variables may be equal to 1 as well. Finally, the last equation provides means for increasing the failure rate. It simply states that the failure rate given that maintenance occurred θ periods ago is equal to the previous failure rate plus the incremental change α_j in failure rate for the given unit.

3.3.3 Failure Rate Constraint Revisions

All of the failure rate equations remain the same as they had been in the previous model with one very important exception:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} \quad \forall j, t$$

This equation needs to be revised since it is no longer a linear equation. The $\gamma_{j\theta}$ is now a variable and as a result $\gamma_{j\theta} Z_{jt\theta}$ is a nonlinearity which must be made linear. A continuous variable $h_{jt\theta}$ is now introduced and defined as $h_{jt\theta} \equiv \gamma_{j\theta} Z_{jt\theta}$. The following constraints are introduced:

$$\gamma_{j\theta} - \gamma_{j\theta}^{\max}(1 - Z_{jt\theta}) \leq h_{jt\theta} \leq \gamma_{j\theta} - \gamma_{j\theta}^{\min}(1 - Z_{jt\theta}) \quad \forall j, t, \theta = 1 \ldots \tau_j$$

$$\gamma_{j\theta}^{\min} Z_{jt\theta} \leq h_{jt\theta} \leq \gamma_{j\theta}^{\max} Z_{jt\theta} \quad \forall j, t, \theta = 1 \ldots \tau_j$$

The new parameters $\gamma_{j\theta}^{\max}$ and $\gamma_{j\theta}^{\min}$ are given by the following equations:

$$\gamma_{j\theta}^{\max} = \max_{l \in \zeta_l}(\bar{\lambda}_{jl}) + \tau_j \alpha_j \quad \forall j$$

$$\gamma_{j\theta}^{\min} = \min_{l \in \zeta_l}(\bar{\lambda}_{jl}) \quad \forall j$$

The first equation takes the greatest initial failure rate for a given unit and adds $\tau_j \alpha_j$ which will provide the greatest failure rate that can be seen in the model for a given unit. The $\tau_j \alpha_j$ term collects all the possible increases in the failure rate. The second equation simply takes the lowest possible initial failure rate which will be equal to the smallest possible failure rate that could be seen in the model. These two parameters serve as bounds for the constraints above.

This method of linearization was demonstrated by C. Floudas[9] (1995) in Nonlinear and Mixed-Integer Optimization and introduced by C.C. Petersen (1971). The goal behind this math is that $h_{jt\theta}$ has to equal $\gamma_{j\theta} Z_{jt\theta}$ when the decision variable is equal to 1 while the $h_{jt\theta}$ has to equal 0 when the decision variable equals 0. Therefore, the $h_{jt\theta}$ equals the failure rate only when there should be a failure rate to report. Two different cases are possible; the first is that the decision variable equals 1. The first equation simplifies to the following:

$$\gamma_{j\theta} \leq h_{jt\theta} \leq \gamma_{j\theta} \quad \forall j, t, \theta = 1 \ldots \tau_j$$

and the second to:

$$\gamma_{j\theta}^{\min} \leq h_{jt\theta} \leq \gamma_{j\theta}^{\max} \quad \forall j, t, \theta = 1 \ldots \tau_j.$$

In this case the $h_{jt\theta}$ variable takes on the value of the failure rate as it should and the second constraint is satisfied as the failure rate value falls within the two parameters. However, if the decision variable is equal to 0 then the following results:

$$\gamma_{j\theta} - \gamma_{j\theta}^{\max} \leq h_{jt\theta} \leq \gamma_{j\theta} - \gamma_{j\theta}^{\min} \quad \forall j, t, \theta = 1 \ldots \tau_j$$

$$0 \leq h_{jt\theta} \leq 0 \quad \forall j, t, \theta = 1 \ldots \tau_j$$

This time the $h_{jt\theta}$ is forced to equal zero, but it will still satisfy the first constraint because the left hand side of $h_{jt\theta}$ is a negative number and the right hand side is a positive number (or either could be equal to zero) (Floudas 245). Finally, the $\gamma_{j\theta} Z_{jt\theta}$ is substituted by the $h_{jt\theta}$ to form:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} h_{jt\theta} \quad \forall j, t$$

and the model becomes linear again.

3.3.4 Other Revisions

Only two revisions are necessary; one is made to the objective function and the second is made to the maximum number of batches equation. In order to account for the initial failure rate cost factor the term $\sum_{l \in \zeta_j} K^2_{jl} E_{jl}$ is attached to the profit equation to give the following:

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$
$$\sum_{jt} C^p_{jt} X_{jt} - \sum_{jt} C^c_{jt} (H - U_{jt} - \Delta^p_j X_{jt}) / \Delta^p_j - \sum_{j} (K^0_j E_j$$
$$+ K^1_j \sum_{k \in \Psi_j} \overline{V}_{jk} E_j + \sum_{l \in \zeta_j} K^2_{jl} E_{jl})$$

The $N^{\max}_{ij}$ equation present in the previous model also has to be adjusted in order to account for the change in failure rate formulations:

$$N^{\max}_{ij} = \frac{H(1 - \Delta^c_j \min_{l \in \zeta_l} (\overline{\lambda}_{jl}))}{p_i} \qquad \forall i, j \in K^*_i$$

The purpose of this function is to find the maximum number of batches that can occur in for a given unit-task. The maximum number of batches can be found when the failure rate is the smallest since this would provide the longest uptime for that unit. This equation is similar to the one in the previous model except that the failure rate term is adjusted. In the previous model the failure rate term was $\Delta^c_j \lambda_{j1}$ since in the first time period the failure rate was guaranteed to be the lowest. However, with the new ability to decide the initial failure rate there are several different failure rates the model must choose from. The greatest number of batches will exist when the failure rate is the smallest. Therefore, the model takes the smallest failure rate out of the set of failure rates for a given process unit using the min function.

3.4 Revision of the Design, Reliability, Production and Maintenance Planning Model

The model developed by H.D. Goel et. al. uses very complicated mathematics in their model; by introducing new variables and constraints the model becomes quite cumbersome. The time spent by the solver increases dramatically and in short, the model becomes significantly harder to solve. The model presented in this section accomplishes the same goal as the previous model but with far fewer variables and mathematical complexities.

3.4.1 Reliability Allocation and Failure Rate Constraint Revisions

Aside from the reliability and failure rate constraints of the previous model all other aspects of the math are unchanged. The major modification to the previous model is to where and how the initial failure rate is chosen. In the previous model the reliability allocation constraints were separate from the failure rate constraints. However, in this model the failure rate constraints are merged with the reliability constraints using a new failure rate parameter: $\delta\gamma_{jl}$. The $\delta\gamma_{jl}$ parameter is the initial failure rate differential. This differential is used in the following failure rate constraint:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta}$$

The differential is combined with the initial failure rate decision variable E_{jl} in the following way:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - \sum_{l=1}^{\zeta_i} \delta\gamma_{jl} E_{jl} \qquad \forall j, t$$

The differential, if chosen, subtracts itself from the failure rate. Instead of actually choosing an initial failure rate as in the previous model, a nominal failure rate exists from which all possible initial failure rates can be utilized by subtracting different initial failure rate differentials. For example, if the nominal failure rate begins at a value of 0.002 for Unit A and this is deemed to be the most cost effective then the E_{jl} decision variable will equal zero for all available initial failure rate differentials. Therefore, for Unit A the equation would resemble the following for all potential time periods:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - 0 \qquad \forall j, t$$

However, if the optimal failure rate is actually 0.0015 for a given unit then the E_{jl} decision variable will choose the initial failure rate differential that equals 0.0005 (provided that it exists). The result would be that the equation would now resemble the following for all potential time periods:

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - 0.0005 \qquad \forall j, t$$

The second term acts as a constant value that will adjust the failure rate for every time period and has the same effect on the failures as if the initial failure rate had just been "chosen" like it was in the previous model. This methodology keeps the $\gamma_{j\theta}$ a parameter and therefore, the entire equation remains linear. There is no need for the linearization constraints and only one further reliability allocation constraint is needed:

$$E_j = \sum_{l \in \zeta_l} E_{jl} \qquad \forall j$$

This equation simply makes sure that an initial failure rate is chosen only for units that exist. It also makes sure that only one differential is chosen for any one unit. All of the other constraints in the previous model pertaining to reliability allocation or the linearization of $\gamma_{j\theta} Z_{jt\theta}$ can be deleted. The $\overline{\lambda}_{jl}$ is replaced with $\delta\gamma_{jl}$, the α_j is no longer needed and $\gamma_{j\theta}$ becomes a parameter again.

4 Model Experimentation: Results and Analysis

The following section provides the results and analysis of the experimentation of the previously discussed mathematical process models. A single problem representing a simple process was used for each mathematical formulation discussed in the previous section. This was done so that the results could be compared with each other. For each different mathematical formulation, this process was modeled using the GAMS modeling environment[1] and then they were solved using the NEOS solver network[2]. The solver used in the experimentation was either the XPRESS solver or the CPLEX 10.1.0 solver.

4.1 The Problem and Important Definitions

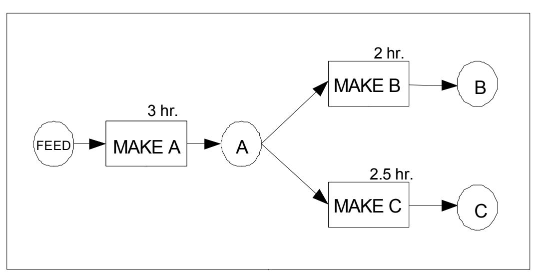

Fig. 7 STN under analysis for this research project

Table 1: Process Unit Details

PROCESS UNIT	SUITABILITY
Unit 1	MAKE A
Unit 2	MAKE B, MAKE C
Unit 3	MAKE B, MAKE C

Table 1 Process Unit Details for STN (same for all models)

Table 2: Storage Unit Details

PROCESS UNIT	CAPACITY	SUITABILITY
Feed Tank	∞	FEED
B Tank	∞	B
C Tank	∞	C

Table 2 Storage Unit Details for STN (same for all models)

The process that is under experimentation is represented by the STN in Figure 7 with the process unit relationships represented by Table 1[7]. In this process there are three process units and there are also three storage tanks represented by Table 2. Table 1 demonstrates which units are suitable for the tasks shown in the STN. It should be noted that in the mathematics of model the storage tanks are not considered to be process units. Although in a physical sense the storage tanks are units in a process they are not considered to be a process unit here. The reason for this is that the tank doesn't actually perform any tasks; however, a tank's capacity serves as an upper bound for the storage of materials variable S_{st}. Since the capacities of the three tanks are considered infinite there is no need for an upper bound on S_{st} for these materials. However, there is no tank for State A and therefore, any State A that is produced in a given period must immediately be used by Unit 2 or Unit 3. An upper bound of 0 exists for the S_{st} variable of state A.

Each model was analyzed as part of a two year time horizon split into periods of one month. Therefore the number of time period t will be equal to 24.

The results of each experiment are reported in a table of the different components of the model. The definitions of these components are important to know. The first section of any table will be the statistical results of the optimization of a given model. The following will be present in this section:

Solver Type – This will either be the XPRESS or CPLEX solver

OPTCR – This is the value of the relative optimality tolerance[1]

Binary Variables – The number of binary variables

Continuous Variables – The number of continuous variables

Constraints – The number of constraining equations

Relative Gap – The relative difference between the optimal solution and the LP fully relaxed solution (best possible solution)

CPU Time – The number of seconds the CPU takes to compute final solution[1]

Iterations - The number of iterations the solver uses[1]

The next section provides a summary of the finances of the process:

VD (Value of Deliveries)

Total revenue from all the deliveries:

$$\sum_{st} \eta_{st} D_{st}$$

TCM (Total Corrective Maintenance)

Total cost of repairs (corrective maintenance):

$$\sum_{jt} C^{c}_{jt}(H - U_{jt} - \Delta^{p}_{j} X_{jt}) / \Delta^{p}_{j}$$

TPM (Total Preventive Maintenance)

Total cost of preventive maintenance:

$$\sum_{jt} C^{p}_{jt} X_{jt}$$

Design (Total Cost of Design)

Total fixed, variable and reliability design costs:

$$\sum_{j} (K^{0}_{j} E_{j} + K^{1}_{j} \sum_{k \in \Psi_{j}} \overline{V}_{jk} E_{j} + \sum_{l \in \zeta_{j}} K^{2}_{jl} E_{jl})$$

Reliability (Total Cost of Reliability)

The cost of Reliability

$$\sum_{j} \sum_{l \in \zeta_{j}} K^{2}_{jl} E_{jl}$$

Objective (Solution)

The solution to the Profit (objective) function

Production and Maintenance Planning Model

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_{i}} \sum_{\omega = 0}^{p_{i} = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$

$$\sum_{jt} C^{p}_{jt} X_{jt} - \sum_{jt} C^{c}_{jt}(H - U_{jt} - \Delta^{p}_{j} X_{jt}) / \Delta^{p}_{j}$$

Design Model

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) - \sum_{jt} C^{p}_{jt} X_{jt} - \sum_{jt} C^{c}_{jt} (H - U_{jt} - \Delta^{p}_{j} X_{jt}) / \Delta^{p}_{j} - \sum_{j} (K^{0}_{j} E_{j} + K^{1}_{j} \sum_{k \in \Psi_j} \overline{V}_{jk} E_{j})$$

Reliability Models

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) - \sum_{jt} C^{p}_{jt} X_{jt} - \sum_{jt} C^{c}_{jt} (H - U_{jt} - \Delta^{p}_{j} X_{jt}) / \Delta^{p}_{j} - \sum_{j} (K^{0}_{j} E_{j} + K^{1}_{j} \sum_{k \in \Psi_j} \overline{V}_{jk} E_{j} + \sum_{l \in \zeta_j} K^{2}_{jl} E_{jl})$$

The existence of some of these values depends on which model is being analyzed.

4.2 Multiperiod Production and Maintenance Planning Model (PMP Model)

The first model that was analyzed was the Multiperiod Production Planning and Maintenance Planning Model proposed by E.N. Pistikopolous et. al[3]. The mathematics of this model was discussed in Section 3.1: the production and maintenance scheduling are the only two components considered here. The model that was optimized was simply a reproduction of the model used in the published paper as the same parameters were used. Therefore, the results of this model were expected to be the same as the results in the published paper. A summary of the parameter values is shown by Table 3. The time per period *H* will always be equal to 720 hours. The demand constraints on this first problem set bounds at 5000 and 20000 for states B and C for every time period. The processing times for each task are presented with the STN and remain constant for all models researched in this paper. The capacities are fixed for this model at 200, 50, 40 for Unit 1, Unit 2, and Unit 3 respectively.

Table 3	
Parameters for Multiperiod Production and Planning Model	
τ_j	9 periods
γ_{j1}	0.002 h^{-1}
$\gamma_{j\theta}$	$\gamma_{j\theta-1} + 0.001$ h^{-1}
Δ_j^c	24 h
Δ_j^p	6 h
C_j^c	50
C_j^p	1000
η_{st} (B and C only)	0.5

Since the effects of the utility constraints are not considered in these process models the parameters associated with them are assumed to be equal to zero. Therefore, only the parameters listed in the table and in the previous paragraph have any effect on the models results.

The value of τ_j for all three units is equal to 9. All three different uptime definitions were analyzed as separate cases. The number of each case listed in the results corresponds to the same case that was defined earlier. A summary of the results for each case is reported in Table 4.

Table 4 - Summary of Multiperiod Production and Maintenance Planning Results

Case 1		**Case 2**		**Case 3**	
APPmodelP1C1(pist).gms		APPmodelP1C2(pist).gms		APPmodelP1C3(pist).gms	
Solver	Xpress	Solver	Xpress	Solver	Xpress
Statistics		Statistics		Statistics	
Binary Variables	612	Binary Variables	612	Binary Variables	612
Continuous Variables	457	Continuous Variables	997	Continuous Variables	997
Constraints	925	Constraints	1537	Constraints	1537
Relative Gap	0	Relative Gap	0	Relative Gap	0
CPU time	0.07	CPU time	0.12	CPU time	0.12
Iterations	493	Iterations	734	Iterations	655
Values		Values		Values	
VD	334326.24	VD	335179.104	VD	334631.1
TCM	14000	TCM	10130.1	TCM	9643.28
TPM	10368	TPM	15000	TPM	12000
Objective	309958.24	Objective	310049.004	Objective	312987.8

The results of each case match exactly the results reported in the published paper[3]. The notable exceptions to this are the values of the variables. However, this can be explained. The Feed state in the model has the option of being included in the mathematical formulation. Since the feed is infinite it does not necessarily need to be included in the mathematical formulation. However, in this particular part of the research project the feed is considered. This resulted in two changes made to the model. First, an initial S_{s1} parameter must be added to the right hand side of the material balance equation to account for the initial amount of feed material. The result is:

$$S_{st} = S_{s1} + S_{s,t-1} + \sum_{i \in T_S} \sum_{j \in K_j} \bar{\rho}_{is} B_{ijt} - \sum_{i \in T_S} \sum_{j \in K_j} \rho_{is} B_{ijt} - D_{st}$$

Second, the additional parameter needs to be defined in the problem statement. For this particular case a very large number was given for the Feed state while for the other three states S_{s1} was equal to zero. Since the feed state is considered there is an extra value for the s index and therefore, more variables. The changes made to the material balance equation and the inclusion of the Feed state had no effect on the other reported values and therefore, was considered to be inconsequential. However, the Feed state could be disregarded, as it will be in future models, in order to simplify the model.

Figures 8-10 represent the failure rate profiles for the different units of each case. These values match the reported values in E.N. Pistikopoulos, et. al.[3]

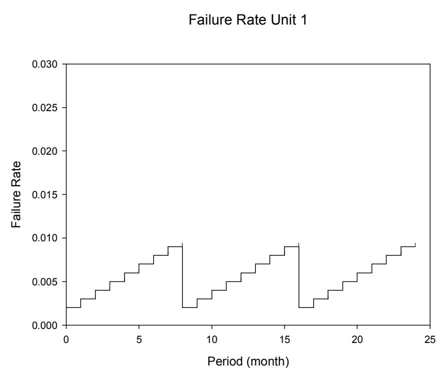

Fig. 8-1 Case 1 Unit 1 Failure Rate Profile

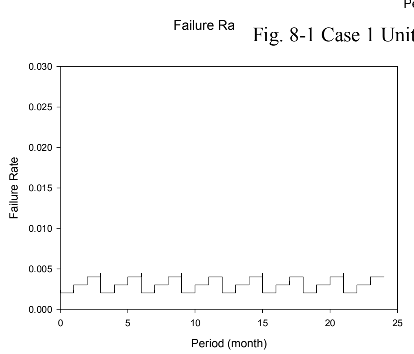

Fig. 8-2 Case 1 Unit 2 Failure Rate Profile

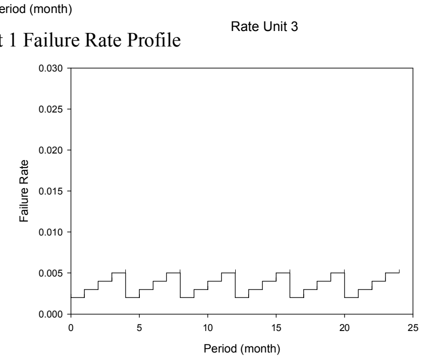

Fig. 8-3 Case 1 Unit 3 Failure Rate Profile

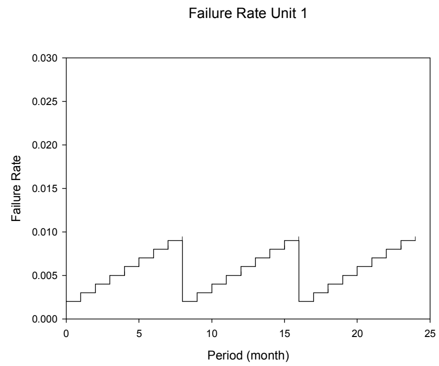

Fig. 9-1 Case 2 Unit 1 Failure Rate Profile

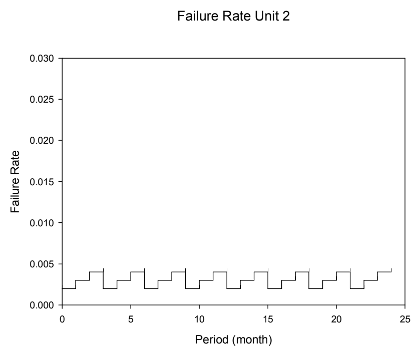

Fig. 9-2 Case 2 Unit 2 Failure Rate Profile

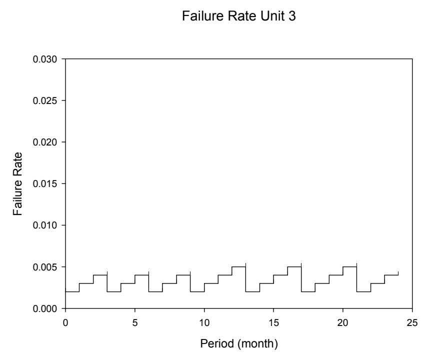

Fig. 9-3 Case 2 Unit 3 Failure Rate Profile

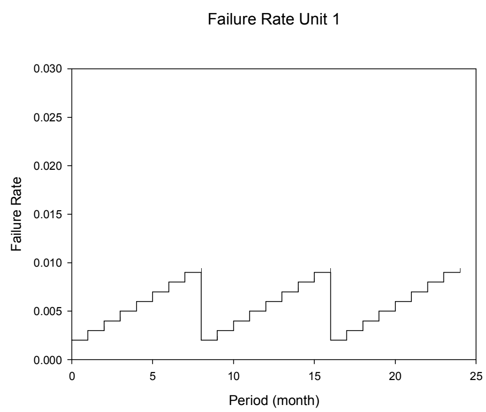

Fig. 10-1 Case 3 Unit 1 Failure Rate Profile

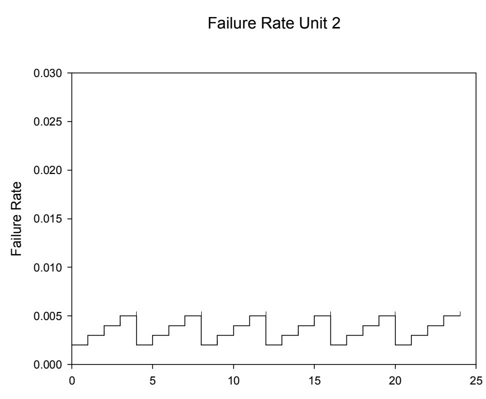

Fig. 10-2 Case 3 Unit 2 Failure Rate Profile

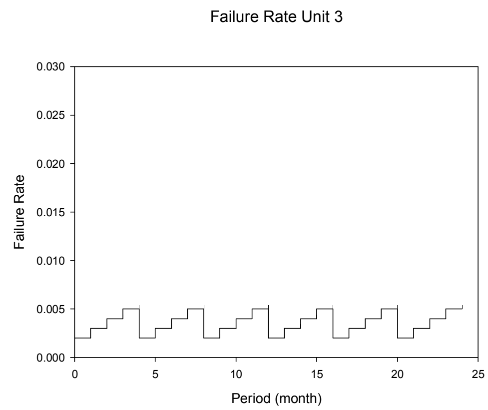

Fig. 10-3 Case 3 Unit 3 Failure Rate Profile

4.3 Simultaneous Design, Production, and Maintenance Planning Model

This model was an extension of the previous model made in the same published paper.[3]

As mentioned previously, this model introduces the ability for the optimization of its own design. The result of this, however, is that the costs related to design are not factored into the profit equation (see above profit equations). There are six cases for this part of the research project; the feed was either included or it was not and then all three different uptime definitions tested. Again the same parameters were used with the intention of achieving the same results. The CPLEX 10.1.0 solver was used for all cases. A summary of the parameter values is shown by Table 5. The failure rate now uses the following equation:

$$\gamma_{j\theta} = \gamma_{j\theta-1} + \alpha_j$$

Table 5 Design, Failure Rate, and Maintenance Data

	K_j^0	K_j^1	γ_{j1}	α_j	Δ_j^c	Δ_j^p	C_{jt}^c	C_{jt}^p
Unit 1	5000	100	0.002	0.001	24	6	50	1000
Unit 2	20000	300	0.004	0.001	40	9	100	2000
Unit 3	20000	350	0.002	0.001	30	7	75	2000

Table 6 Design Alternatives

	Available Unit Sizes			
Unit 1	150	175	200	250
Unit 2	50	80	150	200
Unit 3	60	100	125	200

The value of τ_j for all three units is now equal to 6. This will be the case for the rest of the research. Table 6 provides a list of all the different design alternatives. A summary of the results for each case is reported in Table 7A for the FEED variation; 7B contains the results for the non-FEED.

Table 7A Simultaneous Design, Production, and Maintenance Planning Model - Research Results

	Case 1	Case 2	Case 3
File	APPDmodelP2C1(pist).gms	APPDmodelP2C2(pist).gms	APPDmodelP2C3(pist).gms
Solver	CPLEX 10.1.0	CPLEX10.1.0	CPLEX10.1.0
OPTCR	0.03	0.03	0.03
Statistics			
Binary Variables	474	474	474
Continuous Variables	1060	1447	1447
Constraints	1569	2028	2028
Relative Gap	0.02657	0.013265	0.029187
CPU time	9.17	8.77	9.81
Iterations	7799	5424	7679
Values			
VD	674050	674254	673440.425
TCM	16632	17419.125	17775.225
TPM	31000	27000	27000
Design	136000	137750	137750
Objective	490418	492084.875	490915.2

Table 7B Simultaneous Design, Production, and Maintenance Planning Model - Research Results

	Case 1	Case 2	Case 3
File	APPDmodelP2C1(pist)nofeed.gms	APPDmodelP2C2(pist)nofeed.gms	APPDmodelP2C3(pist)nofeed.gms
Solver	CPLEX 10.1.0	CPLEX 10.1.0	CPLEX10.1.0
OPTCR	0.03	0.03	0.03
Statistics			
Binary Variables	474	474	474
Continuous Variables	1012	1398	1398
Constraints	1546	2005	2005
Relative Gap	0.02657	0.005311	0.027258
CPU time	8.92	11.06	21.56
Iterations	7799	6135	6894
Values			
VD	674050	674080.8	673741
TCM	16632	17554.5	17007.975
TPM	31000	27000	29000
Design	136000	137750	136000
Objective	490418	491776.3	491733.025

The optimal sizes for each case are reported in Table 8.

Table 8A Optimal Unit Sizes for Simultaneous Design, Production, and Maintenance Planning Model - Published Results (w/ FEED)			
	case 1	case 2	case 3
Unit 1	250	250	250
Unit 2	150	80	80
Unit 3	60	125	125

Table 8B Optimal Unit Sizes for Simultaneous Design, Production, and Maintenance Planning Model - Published Results (NO FEED)			
	case 1	case 2	case 3
Unit 1	250	250	250
Unit 2	150	80	150
Unit 3	60	125	60

The figures depicting the optimal maintenance schedule are located in Figure 11.

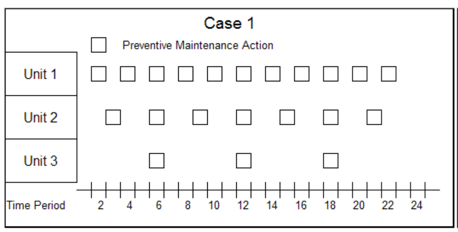

Fig. 11-1 Preventive Maintenance Schedule for Researched Design Model

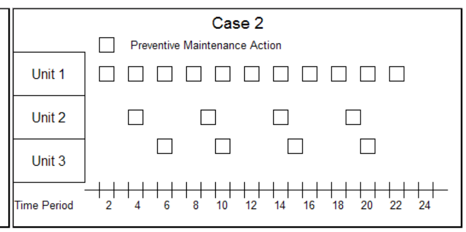

Fig. 11-2 Preventive Maintenance Schedule for Researched Design Model

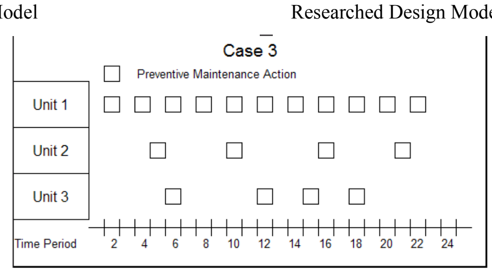

Fig. 11-3 Preventive Maintenance Schedule for Researched Design Model

The results of this model for all the cases differ from the results provided in the published work. The reason for this inconsistency could not be explained. However, the number of variables for each case corresponds directly to the statistics provided in the model or there are more variables because of the inclusion of the Feed state. This means there is most likely no problem with the mathematical structure of the model and potentially something wrong with the parameters. However, upon inspection, the parameters in the models equal those that are reported in the paper. The inconsistency in the results becomes difficult to explain. Even the optimal sizes found by the models in this research do not match the published results. Table 9[3] demonstrates the optimal sizes found in the published work.

Table 9 Optimal Unit Sizes for Simultaneous Design, Production, and Maintenance Planning Model - Published Results

	case 1	case 2	case 3
Unit 1	250	250	250
Unit 2	80	200	60
Unit 3	125	50	200

It is apparent that the value of the deliveries is accurate for some of the cases but the design and the maintenance costs are not equal to the reported values. For example, in the feed and non-feed Case 1 models the value of the deliveries is accurate but the costs and thus the objective function are inconsistent.

Table 10 Simultaneous Design, Production, and Maintenance Planning Model - Published Results

Case 1		**Case 2**		**Case 3**	
OPTCR	0.03	OPTCR	0.03	OPTCR	0.04
Statistics		Statistics		Statistics	
Binary Variables	474	Binary Variables	474	Binary Variables	474
Continuous Variables	1012	Continuous Variables	1398	Continuous Variables	1398
Constraints	1545	Constraints	2004	Constraints	2004
CPU time	71	CPU time	162.6	CPU time	138.7
Values		Values		Values	
VD	674050	VD	674253	VD	678229
TCM	17208	TCM	19114	TCM	15662
TPM	24000	TPM	20000	TPM	25000
Design	137750	Design	151000	Design	155000
Objective	495092	Objective	484139	Objective	482567

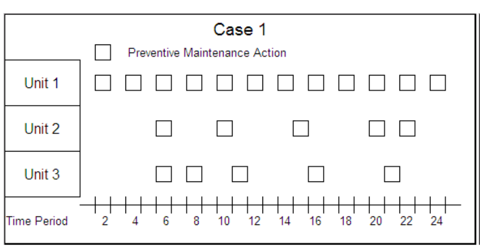

Fig. 12-1 Preventive Maintenance Schedule for Design Model Published Results

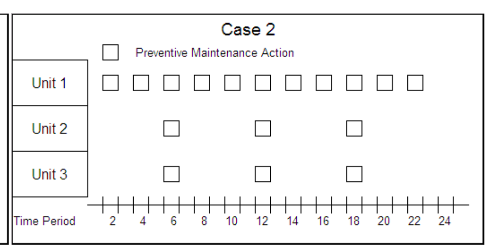

Fig. 12-2 Preventive Maintenance Schedule for Design Model Published Results

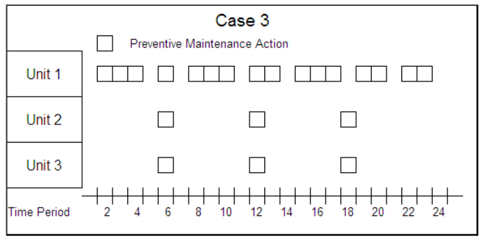

Fig. 12-3 Preventive Maintenance Schedule for Design Model Published Results

Table 10 lists the reported results from the published paper[3] and Figure 12 represents the reported preventive maintenance schedule for all three cases. In the table the total preventive maintenance for case 1 is equal to 24000. The value retrieved by this research project is 31000. However, by counting the number of maintenance actions Figure 12, the published results, it becomes apparent that there is an inconsistency in the published paper. There are 11 maintenance actions performed on Unit 1, five performed on Unit 2 and five performed on Unit 3. The Unit 1 maintenance actions cost 1000 units a piece while the others cost 2000 a piece for a grand total of 31000 not 24000 as is reported in the table. Furthermore, this is the same as the value found in this research project. As a result, only part of the published work agrees with the researched results. Unfortunately, the actual schedule for preventive maintenance presented by the published work still disagrees with the researched work as can be seen upon comparison between Figure 11 and Figure 12. In the other cases similar inconsistencies exist.

Due to the inconsistency within the published work it becomes difficult to attest to the accuracy of the model created for this research report. Various attempts were made to try to determine if there was anything incorrect with the model constructed for this research. However, despite fixing variables and fixing the maintenance schedule the published results were never matched.

4.4 Design, Reliability, Production and Maintenance Planning Model

The analysis and experimentation of the next model is the focus of this research report. This model was proposed by H.D. Goel et. al. as a revision and expansion of the previous Simultaneous Design, Production, and Maintenance Planning Model.

As mentioned previously, it includes the ability to optimize the initial reliability allocation. This model serves as the focus of this research report because it does not appear to efficiently introduce initial reliability allocation. As a result, an alternative is proposed in the next section. It appears that the model proposed by H.D. Goel et. al. unnecessarily increases the size of the model through complicated mathematics. The next section presents a simpler alternative that reduces the size of the model.

The analysis of the H.D. Goel et. al. model uses the same process and the same parameters established in the previous model. This is done so that there is a basis of comparability. Table 11 summarizes new initial failure rate cost factor and the initial reliabilities that is introduced in this model; the rest of the parameters equal the values given in Table 5 and 6.

Table 11 – New Parameters for H.D. Goel et. al.

	Available Initial Failure Rates $\bar{\lambda}_{jl}$			Failure Rate Cost Factor K^2_{jl}		
L	1	2	3	1	2	3
Unit 1	0.002	0.0015	0.001	0	2200	6000
Unit 2	0.004	0.003	0.002	0	2200	6000
Unit 3	0.002	0.0015	0.001	0	2200	6000

Table 12 Design, Production, Reliability and Maintenance Planning Model - Research Results

Feed		**No Feed**	
goelmodel.gms		goelmodelnofeed.gms	
Solver	CPLEX 10.1.0	Solver	CPLEX 10.1.0
OPTCR	0.03	OPTCR	0.03
Statistics		Statistics	
Binary Variables	483	Binary Variables	483
Continuous	1465	Continuous	1417
Constraints	3258	Constraints	3114
Relative Gap	0.029585	Relative Gap	0.029409
CPU time	262.35	CPU time	122.86
Iterations	242388	Iterations	91763
Values		Values	
VD	690565.2	VD	690565.2
TCM	15048	TCM	15048
TPM	27000	TPM	27000
Design	145950	Design	145950
Reliability	8200	Reliability	8200
Objective	502567.2	Objective	502567.2

FEED Optimal Reliability and Unit Size			**NO FEED Optimal Reliability and Unit Size**	
	Size	Initial Failure Rate	Size	Initial Failure Rate
Unit 1	250	0.001	250	0.001
Unit 2	80	0.003	80	0.003
Unit 3	125	0.002	125	0.002

Table 12 represents the results from the FEED/no-FEED state variations.

Note: The following results are those from the published paper.

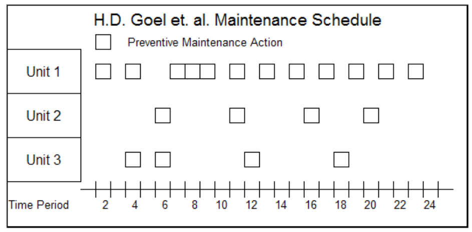

Fig. 13A Preventive Maintenance Schedule
Published Results and Failure Rate Profiles

Fig. 13B Preventive Maintenance Schedule
Published Results and Failure Rate Profiles

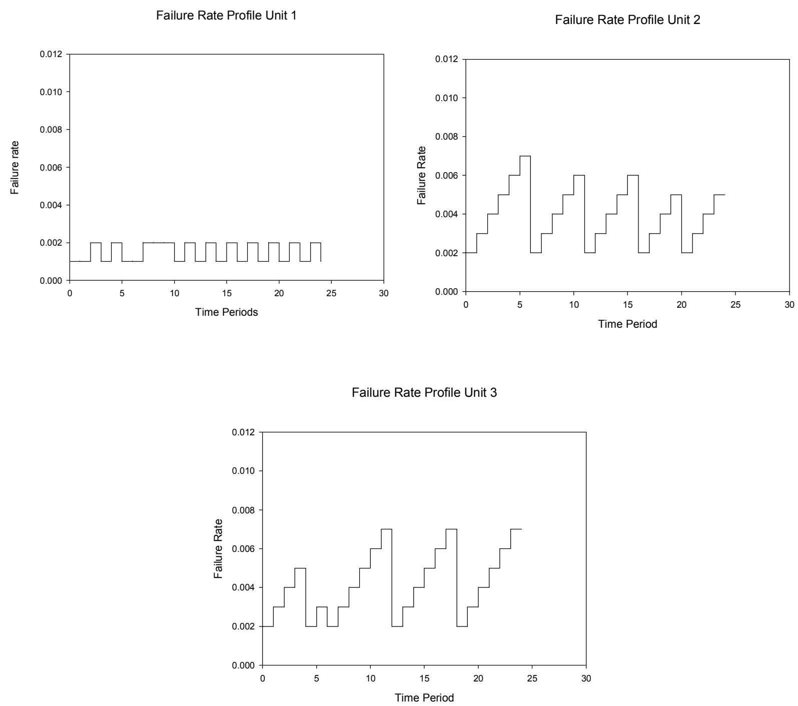

Table 13 summarizes the results published in the paper and it can be clearly seen that the researched results do not match the published results.

Table 13 Design, Production, Reliability and Maintenance Planning Model - Published Results	
OPTCR	0.035
Statistics	
Binary Variables	480
Continuous	2042
Constraints	3247
CPU time	0.33
Values	
VD	690905
TCM	13536
TPM	28000
Design	148000
Objective	501370

cont'd from previous page

Optimal Reliability and Unit Size		
	Size	Initial Failure Rate
Unit 1	250	0.001
Unit 2	150	0.002
Unit 3	60	0.002

As in the published results of the previous model there is another inconsistency in the results of the H.D. Goel et. al. paper. Figure 13A/B depicts a graphical copy of the maintenance scheduling reported in the published work as well as the failure rate profiles. In these diagrams if the failure rate profile and the preventive maintenance block diagram are compared then it becomes apparent that for Unit 1 the two graphs do not correlate. The failure rate profile shows that there is no change in the failure rate for period 6 which suggests a maintenance action for period 5. However, this is not shown in the block diagram. Furthermore, the block diagram depicts a failure rate action takes place in period 7 and yet there is no change in the failure rate according to the profile. Realistically, the profile should have returned to the AGAN failure rate. Unfortunately, this is not enough to account for the discrepancy between the published and researched results.

Table 14: Breakdown of Binary Variables

E_j - 3 units	3 variables
E_{jl} - 3 units 3 reliabilities	9 variables
E_{jk} - 3 units 4 sizes	12 variables
X_{jt} - 3 units 24 periods	72 variables
$Z_{jt\theta}$ -	387 variables
	483 variables

However, the model statistics reported by H.D. Goel et. al. are a little suspicious. The number of binary variables that H.D. Goel et. al. reports is 480. However, a breakdown of the number of binary variables represented in Table 14 demonstrates the existence of 483 binary variables. The researched models uniformly have 483 binary variables. How is H.D. Goel et. al. able to have only 480? It is possible that their model did away with the E_j variable, but this would be very difficult due to the number of uses it has in the model. Furthermore, H.D. Goel et. al. reports using 2042 continuous variables and 3247 constraints. The researched models have only 1417 or 1465 continuous variables and 3258 or 3114 constraints. The numbers of constraining equations are very close to the published work, but the number of continuous variables is wildly different. However, this could be due to confusion over how GAMS reports the number of continuous variables. Instead of separating reporting the number of continuous variables a report called "Single Variables" is provided in the Solve Summary. Therefore, the number of binary variables must be subtracted away from the number of "Single Variables" in order to find the number of continuous variables. This could account for the extra continuous variables. Regardless, the main concern with the inconsistency in the number of variables is that it represents structural inconsistencies between the researched models and the published work. The most troubling statistic reported in the H.D. Goel et. al. paper is the CPU time. It reports that the model was solved in 0.33 seconds. This is less than half a second! The reproductions of the H.D. Goel et. al. model do not come near to this value. Furthermore, the H.D. Goel et al. research reports that the calculations were run on a simple AMD athlon processor[4] while the reproductions were computed on the sophisticated NEOS Solvers Servers[2]. The fastest reproduction of the Simultaneous Design, Production, and Maintenance Planning model of the E.N. Pistikopoulos paper was solved in 9.17 seconds and that model does not have nearly as many variables or equations. This must mean that there is either something very wrong with the H.D. Goel et. al. model or that the published CPU time is simply incorrect. These types of inconsistencies within the H.D. Goel et. al. paper make it very difficult to draw conclusions about the accuracy of the reproductions.

Note: The rest of the results from this section are from the H.D. Goel et. al. model reproduction (research results).

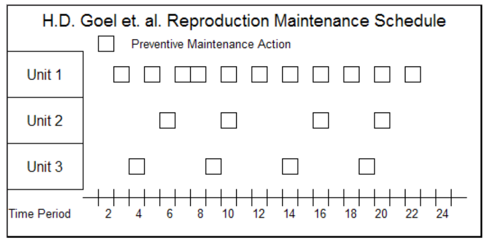

Fig. 14 Preventive Maintenance Schedule Reproduction (Researched) Results

Therefore, the reproductions of the H.D. Goel et. al. model were analyzed in depth by using different combinations of solvers and OPTCR values. Furthermore, the binary variables of the model were either fixed to duplicate the H.D. Goel et. al. model's maintenance schedule, design and initial reliability or the binary variables were left to be optimized. Figure 14 depicts the results found for the H.D. Goel et. al. Maintenance Schedule by this research report (these are the results found when the H.D. Goel et. al. was reproduced and then solved). Table 15 is a summary of these results. The most important result is that from the cases where the OPTCR is set to 0.01 and the binary variables are fixed. In these cases, the model actually equals the results reported in the published paper. It should also be noted in that in these cases the CPU time is very small, but this is because the binary variables are all fixed. It is a simple LP model. Nevertheless, the fact that the profit values of these models match the published value suggests that the model may not be as inaccurate as previously thought. Unfortunately, this only occurs for a very low OPTCR and when all the binary variables are fixed. An unfixed solution with the same OPTCR of 0.01 actually finds a better objective value.

Table 15 – Detailed Analysis of H.D. Goel, et. al. Model Reproduction

	Fixed Binary Variables					
Solver	XPRESS			CPLEX		
OPTCR	0.035*	0.035	0.01	0.035*	0.035	0.01
Statistics						
Resource Usage	0.11	4.71	5.3	0.10	3.16	4.680
Iteration Count	212	6065	6065	680	4019	4642
Best Solution	501369.6	501754.81	505391.78	503637.6	512701.92	503788.01
Relative Gap	0.00	0.008640	0.00	0.012155	0.033651	0.004824
Single Equations	3259	3259	3259	3259	3259	3259
Single Variables	1948	1948	1948	1948	1948	1948
Binary Variables	387	411	411	387	411	411
Values						
Objective Value	501369.6	497419.7	497419.7	497589.6	504480.50	501369.6
Deliveries Value	690905.6	686435.70	691011.60	687305.6	690656.50	690905.60
Preventive Main.	28000.00	28000.00	28000.00	28000.00	27000.00	28000.00
Corrective Main.	13536.00	14616.00	14616.00	13716.00	14976.00	13536.00
Reliability Cost	12000.00	10400.00	10400.00	12000.00	8200.00	12000.00
Design Cost	148000.00	146400.00	146400.00	148000.00	144200.00	148000.00

	Free Binary Variables			
Solver	XPRESS		CPLEX	
OPTCR	0.035	0.01	0.035	0.01
Statistics				
Resource Usage	82.62	721.40	282.51	1000.14
Iteration Count	54860.00	1000000.00	231899.00	522218.00
Best Solution	517113.94	514100.53	520131.09	512374.13
Relative Gap	0.03	0.021576	0.03	0.02
Single Equations	3259	3259	3259	3259
Single Variables	1948	1948	1948	1948
Binary Variables	483	483	483	483
Values				
Objective Value	499689.00	503008.10	502567.20	502645.30
Deliveries Value	691423.00	691228.00	690565.00	691281.30
Preventive Main.	30000.00	26000.00	27000.00	30000.00
Corrective Main.	13734.00	14220.00	15048.00	14436.00
Reliability Cost	12000.00	12000.00	8200.00	8200.00
Design Cost	148000.00	148000.00	145950.00	144200.00

* all decision variables apart from the $Z_{jt\theta}$ is fixed

Regardless, the values of the objective function (the profit) for all reproductions are well within 7000 units of the value reported by the published paper. For this reason, the researched reproductions can be considered to be fairly accurate despite the absence of an exact match. As a result they are used as a basis for comparison in the following section.

Finally, although there are clearly problems with the published results the H.D. Goel et. al. model is not a complete failure. The initial reliability allocation feature is demonstrated to be successful. Table 16 presents a comparison of initial failure rates between the two published papers: H.D. Goel et. al. and E.N. Pistikopoulos et. al.

Table 16 – Initial Failure Rates for the Design Model and the Reliability Model		
	H.D. Goel et. al.	E.N. Pistikopoulos et. al.
Unit 1	0.001	0.002
Unit 2	0.002	0.004
Unit 3	0.002	0.002

The H.D. Goel et. al. initial failure rates differ from the original Design Model from E.N. Pistikopoulos et. al. and therefore, the optimization of initial reliability is a worthwhile addition to these models. Note that the initial reliability allocation and the design choices for the research version of the H.D. Goel et. al. model is presented at the bottom of Table 12.

4.5 Revision of the Design, Reliability, Production and Maintenance Planning Model

The mathematical model that is proposed in the H.D. Goel et. al. paper includes very complex math, which leads to difficulties when the model is solved. As a result of the extra linearization equations the model takes a very long time to be solved. Therefore, a revision of this model is proposed and compared to the original. The changes in the mathematics are explained in Section 3.4.1, but it would be sufficient to note that the revised model introduces an initial failure rate differential, chosen by a decision variable, that becomes a constant subtracted from the failure rate equation. Thus, an initial failure rate is never chosen outright as it is in the H.D. Goel et. al. model, but the same values of an initial failure rate can be attained if the appropriate differential is selected. The method results in the model retaining its linearity and therefore, it by-passes the need for extra constraints and variables. This reduces the model solution time. The parameters for this model remain the same as the parameters for the previous model except for the new parameters describing the initial failure rate differential; these are described in Table 17.

Table 17 – New Parameters for Revised Reliability Model						
	Initial Failure Rates Differentials $\delta\gamma_{jl}$			Failure Rate Cost Factor K^2_{jl}		
L	1	2	3	1	2	3
Unit 1	0	0.0005	0.001	0	2200	6000
Unit 2	0	0.001	0.002	0	2200	6000
Unit 3	0	0.0005	0.001	0	2200	6000

Figure 15 represents the preventive maintenance scheduling.

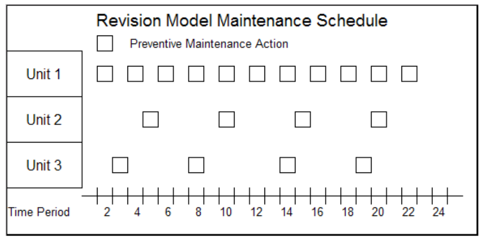

Fig. 15 Revision Model Preventive Maintenance Schedule Results

The revised model is analyzed on a number of levels. Table 18 represents the two solutions based on the existence of the Feed state.

Table 18 - Revised Design, Production, Reliability and Maintenance Planning Model - Research Results

Feed		**No Feed**	
newmodel.gms		newmodelnofeed.gms	
Solver	CPLEX 10.1.0	Solver	CPLEX 10.1.0
OPTCR	0.03	OPTCR	0.03
Statistics		Statistics	
Binary Variables	483	Binary Variables	483
Continuous	1060	Continuous	1012
Constraints	1572	Constraints	1548
Relative Gap	0.028817	Relative Gap	0.028817
CPU time	19	CPU time	18.5
Iterations	23039	Iterations	23039
Values		Values	
VD	691012.2	VD	691012.2
TCM	14976	TCM	14976
TPM	27000	TPM	27000
Design	145950	Design	145950
Reliability	8200	Reliability	8200
Objective	503086.2	Objective	503086.2

It is quite clear from this table that the existence of a Feed state does not affect the solution. From these results it is quite clear that the model is solved faster than their equivalents represented by Table 12. The number of iterations is lower and the CPU time is shorter. Furthermore, the objective functions provide higher values. Right away it is quite clear that the revised model is substantially better. Table 19s goes into a more detailed comparison of the two models.

Table 19 - Comparison of Methods: Revised Model vs. H.D. Goel, et. al. Model

	Revised Model					
Solver	XPRESS			CPLEX		
OPTCR	0.05	0.035	0.01	0.05	0.035	0.01
Statistics						
Resource Usage	10.81	11.31	12.00	15.86	16.02	14.99
Iteration Count	31978.00	31978.00	35256.00	9487.00	9674.00	12927.00
Best Solution	505418.31	505418.31	505391.78	505796.29	505796.29	505320.51
Relative Gap	0.00	0.00	0.00	0.00	0.00	0.00
Single Equations	1573	1573	1573	1573	1573	1573
Single Variables	1543	1543	1543	1543	1543	1543
Binary Variables	483	483	483	483	483	483
Values						
Objective Value	504907.60	504907.60	504907.60	503512.00	504480.50	504907.60
Deliveries Value	691011.60	691011.60	691011.60	691330.00	690656.50	691011.60
Preventive Main.	27000.00	27000.00	27000.00	27000.00	27000.00	27000.00
Corrective Main.	14904.00	14904.00	14904.00	14868.00	14976.00	14904.00
Reliability Cost	8200.00	8200.00	8200.00	8200.00	8200.00	8200.00
Design Cost	144200.00	144200.00	144200.00	145950.00	144200.00	144200.00
	H.D. Goel, et. al. Model					
Solver	XPRESS			CPLEX		
OPTCR	0.05	0.035	0.01	0.05	0.035	0.01
Statistics						
Resource Usage	65.65	82.62	721.40	133.95	282.51	1000.14
Iteration Count	54572.00	54860.00	1000000.00	108848.00	231899.00	522218.00
Best Solution	517113.94	517113.94	514100.53	527545.46	520131.09	512374.13
Relative Gap	0.04	0.03	0.02	0.05	0.03	0.02
Single Equations	3259	3259	3259	3259	3259	3259
Single Variables	1948	1948	1948	1948	1948	1948
Binary Variables	483	483	483	483	483	483
Values						
Objective Value	497767.00	499689.00	503008.10	502567.20	502567.20	502645.30
Deliveries Value	688709.00	691423.00	691228.00	690565.00	690565.00	691281.30
Preventive Main.	32000.00	30000.00	26000.00	27000.00	27000.00	30000.00
Corrective Main.	14742.00	13734.00	14220.00	15048.00	15048.00	14436.00
Reliability Cost	8200.00	12000.00	12000.00	8200.00	8200.00	8200.00
Design Cost	144200.00	148000.00	148000.00	145950.00	145950.00	144200.00

In this analysis the revised model and the H.D. Goel et. al. model are solved under 6 different conditions. Each model is solved for both the CPLEX solver and the XPRESS solver for 3 different OPTCRs. From this analysis, it becomes quite clear on several levels that the revised model is more robust. Again the iteration counts and CPU time are lower while the profits are greater. However, another important observation is that the OPTCR for the revised model is very easily attained as represented by the value of the Relative Gap, but the H.D. Goel et. al. model struggles to reach the required

OPTCR. In summary, the revised model appears to do a better job of initial reliability allocation than the model proposed in H.D. Goel et. al. paper.

5 Conclusion

5.1 Research

Overall the results of this project demonstrate that it was reasonably successful research. There were a lot of difficulties regarding the correlation of researched results with the published results. However, the reproductions of the first model, the Multiperiod Production and Maitenance Planning model provided results that perfectly matched the results provided in the published paper. The real difficulty with finding results that correlated was with the next two models. The Simulatenous Design, Production and Maintenance planning model and the Design, Production, Reliability and Maintenance planning model proved to be very difficult to duplicate. The only time that one of the models' published results were matched exactly was when one of the Design, Production, Reliability and Maintenance planning models reproductions was fixed so that all the decision variables chose the published results. Only in this particular case was the objective function found to be equal to the published value of the objective function. However, in both cases there is some question as to the published results. First, in the Simultaneous Design, Production and Maintenance Planning model there were inconsistencies with in the published paper. These inconsistencies resulted in two different answers to the same problem existing in the same published document. As a result, it was difficult to draw conclusions. The Design, Production, Reliability and Maintenance planning model also had some interestingly and unexplainable results. For instance, the solution time was a fraction of the normal solution time for solving most of these models. Also, there was an inconsistency in the number of binary variables. There ought to have been 483, but the published work reported three less. If this is true than somehow H.D. Goel et. al. managed to reduce the number of binary variables, but does not explain how. Nevertheless, all the results from the reproductions were very close to the published values and based on this it can be assumed that the models were indeed accurate. The reproduction of the Design, Production, Reliability and Maintenance planning model was close enough that it was assumed comparable to the Revision model that was introduced in this paper. The

Revision model proved to be tremendously effective in lowering the number of variables used in a model that optimizes design, production, maintenance and reliability. Compared to the H.D. Goel et. al. model reproductions the revision model was solved faster, easier and provided better solutions. The short solution times and the lower number of interactions for the revision model provide the evidence for its superiority.

In this research report, the first model created was a basic one that considered the production and maintenance planning of a process. Subsequent models added new components to the process and new complexities to the mathematics. This building up of the complexity was important to the methodology of the research because it provided a systematic and logical path to replicating the models purposed in the literature. Furthermore, each new model that was introduced provided additional realism to the model. In a real chemical process, an engineer has the ability to vary many different process variables; however, in mathematical programming there is a limitation on the number of variables due to model size constraints. The more variables a model contains, the “larger” the model; larger models are harder for the computer to solve. Therefore, the best models are those that closely approximate reality while remaining relatively small and easy to solve. This is another reason as to why the revisionist model proposed in this research is an improvement on the original model proposed by H.D. Goel et. al. The numbers of binary variables in each model are equal, but there are far more continuous variables in the H.D. Goel et. al. model compared to the revisionist model. The cumbersome mathematical formulation in the H.D. Goel et. al. model is the cause for these extra variables. On the other hand, the revisionist model, with its simpler mathematical formulation, is just as powerful (it accomplishes the same task) but it does it quicker and with greater accuracy.

5.2 Future Directions

If the investigation into initial reliability allocation was to be continued, one direction that could be taken would be into using one decision variable to choose both the initial failure rate and the unit size. The logic behind this idea is that when a process unit size is chosen a specific type of unit with a given reliability would be chosen as well. This may be a little bit closer to reality when only a given number of possible process units are available. In other words, if there are only a few units that may be available to accomplish a specific task, one may not have the ability to pick a particular size and then separately choose reliability. The two variables may be combined into a single unit with a given size and reliability. The benefit to this approach is that there would be a lower number of binary variables. As demonstrated in Section 2.2, it is the branching and bounding of these variables that dramatically increase the solution time and the complexity of the model. However, this approach would have its limitations. For example, consider the case were the market contains a huge variety of process units where one could choose size and reliability separately. In this case, it would make more sense to use the models presented in this paper.

However, the most important direction one could take this research is into an investigation of the $N_{ij}^{\max}$ parameter. This parameter plays a very important role in the amount of production accomplished by a given process. It appears to be more than a simple upper bound for the capacity constraints. Often, the value of $N_{ij}^{\max}$ is what is used for the number of batches for a given process and therefore, it plays an immediate role in the production. However, very little attention is paid to the parameter; it ought to have an investigation into how it is defined, the values the definition produces and how those values affect the rest of the process.

References

[1] Brooke, A.; Kendrick, D.; Meeraus, A. (1988) *GAMS: A User's Guide*; The Scientific Press: Redwood City, CA.

[2] NEOS: Server for Optimization. http://www-neos.mcs.anl.gov/.

[3] Pistikopoulos, E. N., Vassiliadis, C. G., Arvela, J., & Papageorgiou, L. G. (2001). Interactions of maintenance and production planning for multipurpose process plants – a system effectiveness approach. *Industrial Engineering and Chemical Research 40*, 3195-3207.

[4] Goel, H.D., Grievink, J., & Weijnen, M.P.C. (2003). Integrated optimal reliable design, production, and maintenance planning for multipurpose process plants. *Computers and Chemical Engineering 27,* 1543-1555.

[5] Williams, H.P., (1990) *Model Building in Mathematical Programming*. John Wiley & Sons: New York.

[6] Williams, H.P. (1993) *Model Solving in Mathematical Programming*. John Wiley & Sons: New York.

[7] Kondili, E., Pantelides, C. C. & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations- I. MILP formulation. *Computers and Chemical Engineering 17,* 211-227.

[8] Koolen, J. L. A. (2001). *Design of Simple and Robust Process Plants*. Weinheim, Germany: Wiley-VCH GmbH.

[9] Floudas, C. A. (1995). *Nonlinear and Mixed-Integer Optimization.* New York: Oxford University Press.

[10] Dedopoulos, I. T. & Shah, N. (1995). Long-term maintenance policy optimization in multipurpose process plants. *Chemical Engineering Research and Design 74,* 307-320.

Appendix A – Mathematics

Indices

i	process tasks
j	equipment units
s	states of material
t	time periods
u	utilities
l	unit initial failure rate
k	unit sizes
θ	number of periods elapsed since unit j was last maintained

Sets

$S_i / \overline{S}_i$	sets of states consumed/produced by task i
$T_s / \overline{T}_s$	sets of tasks receiving/producing materials in state s
I_j	set of tasks for which unit j is suitable
ψ_j	set of unit sizes available for unit j
K_i	set of units suitable for task i
ζ_j	set of possible initial failure rates for unit j

Parameters

V_j	capacity of unit j
$\overline{V}_{jk}$	size k for unit j
$\overline{\lambda}_{jl}$	initial failure rate l for unit j
$\rho_{is} / \overline{\rho}_{is}$	is proportion of input/output of task i from state $s \in S_i / \overline{S}_i$
p_i	set-up and processing time of task i
$\beta_{uij\omega} / \delta_{uij\omega}$	fixed/variable demand factor for utility u by task i in unit j at the time ω relative to the start of the task
$\phi_{ij}^{\min} / \phi_{ij}^{\max}$	minimum/maximum utilization factor
$A_{ut}^{\max}$	maximum availability level of utility u during time period t
$N_{ujt}^{\max}$	maximum number of batches when task i is performed in unit j during time period t duration of each period
Δ_j^c	corrective maintenance (repair) duration of unit j
Δ_j^p	preventive maintenance duration of unit j
τ_j	maximum number of consecutive elapsed time periods without maintenance of unit j
$\gamma_{j\theta}$	failure rate value for unit j when the last maintenance action took
place u	periods ago
K_j^0	fixed cost for unit j over considered time horizon of planning
K_j^1	variable size factor for unit j over considered time horizon of planning
K_{jl}^2	cost factor for unit j with failure rate l over considered time horizon of planning
η_{st}	unit price of state s during period t
C_{ut}	unit cost of utility u during period t
C_{jt}^p	preventive maintenance cost of unit j during period t
C_{jt}^c	corrective maintenance cost of unit j during period t
α_j	constant increment in failure rate
$\delta\gamma_{jl}$	initial failure rate differential

$\gamma_{j\theta}^{\max}/\gamma_{j\theta}^{\min}$ (defined below)

Variables
Binary Variables

E_j	1 if unit j is chosen; 0 otherwise
E_{jk}	1 if size k is chosen for unit j ; 0 otherwise
E_{jl}	1 if failure rate l is chosen for unit j; 0 otherwise
X_{jt}	1 if preventive maintenance is performed on unit j during period t ; 0 otherwise
$Z_{jt\theta}$	1 during period t if unit j was maintained for the last time u periods ago; 0 otherwise

Continuous Variables

N_{ijt}	number of batches of task i processed in unit j over time period t
S_{st}	amount of material in state s in storage at the end of period t
D_{st}	amount of material delivered to external customers from state s over period t
V_j	size of unit j
B_{ijt}	amount of material undergoing task i in unit j during period t
U_{jt}	expected uptime of unit j during period t
λ_{jt}	failure rate of unit j during period t
$\gamma_{j\theta}$	failure rate value for unit j when the last maintenance action took place u periods ago
$\overline{EN}_{ijkt}$	linearization variable used for Capacity Constratins (not needed in Multiperiod Production and Maintenance Planning Model)
$h_{jt\theta}$	linearization variable for failure rate (used in H.D. Goel et. al. Model only)

Resource Utilization

$$\sum_{i \in I_j} p_i N_{ijt} \leq U_{jt} \quad \forall j,t$$

Capacity Constraints

for multiperiod production and maintenance planning model only:

$$\phi_{ij}^{\min} V_j N_{ijt} \leq B_{ijt} \leq \phi_{ij}^{\max} V_j N_{ijt} \quad \forall i,j \in K_i, t$$

for all other models:

$$\phi_{ijt}^{\min} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{EN}_{ijkt} \leq B_{ijt} \leq \phi_{ijt}^{\max} \sum_{k \in \psi_j} \overline{V}_{jk} \overline{EN}_{ijkt} \quad \forall i,j \in K_i, t$$

Linearization:

$$\overline{EN}_{ijkt} \leq N_{ij}^{\max} E_{jk} \quad \forall i,j \in K_i, k \in \psi_j, t$$

$$N_{ijt} = \sum_{k \in \psi_j} \overline{EN}_{ijkt} \quad \forall i,j \in K_i, t$$

E.N. Pistikopoulos et. al. Design Model

$$N_{ijt}^{\max} = \frac{U_j^{\max}}{p_i} \quad \forall i, j \in K_i \quad K_j^1 \quad V_j \quad E_{jk}$$

$$U_j^{\max} = \frac{H}{(1 + \Delta_j^c \lambda_{j1})} \quad \forall j \quad or \quad U_j^{\max} = H(1 - \Delta_j^c \lambda_{j1}) \quad \forall j$$

H.D. Goel et. al Model

$$N_{ij}^{\max} = \frac{H(1 - \Delta_j^c \min_{l \in \zeta_l} (\overline{\lambda}_{jl}))}{p_i} \quad \forall i,j \in K_i$$

Revised Model

$$N_{ij}^{\max} = \frac{H(1 - \Delta_j^c(\gamma_{j1} - \max_{l \in \zeta_l} (\delta l_{jl})))}{p_i} \quad \forall i,j \in K_i$$

Material Balance Constraints

$$S_{st} = S_{s,t-1} + \sum_{i \in T_S} \sum_{j \in K_j} \overline{\rho}_{is} B_{ijt} - \sum_{i \in T_S} \sum_{j \in K_j} \rho_{is} B_{ijt} - D_{st} \quad \forall s,t$$

or

$$S_{st} = S_{sI} + S_{s,t-1} + \sum_{i \in T_S} \sum_{j \in K_j} \overline{\rho}_{is} B_{ijt} - \sum_{i \in T_S} \sum_{j \in K_j} \rho_{is} B_{ijt} - D_{st}$$

Demand Constraints

$$D_{st}^{\min} \leq D_{st} \leq D_{st}^{\max} \quad \forall s,t$$

Utility Constraints (not used)

$$\sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} \beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt} \leq A_{ut}^{\max} H \quad \forall u,t$$

Failure Rate Constraints

E.N. Pistikopoulos et. al. Models

$$\lambda_{jt} = \sum_{\theta = 1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} \quad \forall j,t$$

$$Z_{jt\theta} \leq X_{j,t-\theta} \quad \forall j,t,\theta = 1,\ldots,\tau_j$$

$\sum_{\theta = 1}^{\tau_j} Z_{jt\theta} = E_j \quad \forall j,t$ (for Multiperiod Production and Maintenance Model $\sum_{\theta = 1}^{\tau_j} Z_{jt\theta} = 1 \quad \forall j,t$
)

$X_{jt} \leq E_j \quad \forall j,t$ (not needed for Multiperiod Production and Maintenance Model)

$$\gamma_{j\theta} = \gamma_{j,\theta-1} + \alpha_j \quad \forall j, 2 \leq \theta \leq \tau_j$$

Linearizations for Case 2 and 3 Uptime Definitions:

$$\overline{XZ}_{jt\theta} \leq Z_{jt\theta} \quad \forall j,t,\theta = 1,...,\tau_j$$

$$X_{jt} = \sum_{\theta = 1}^{\tau_j} \overline{XZ}_{jt\theta} \quad \forall j,t$$

H.D. Goel et. al. Model

$$\lambda_{jt} = \sum_{\theta = 1}^{\tau_j} h_{jt\theta} \quad \forall j,t$$

$$Z_{jt\theta} \le X_{j,t-\theta} \qquad \forall j,t,\theta = 1,\ldots,\tau_j$$
$$\sum_{\theta=1}^{\tau j} Z_{jt\theta} = E_j \qquad \forall j,t$$
$$X_{jt} \le E_j \qquad \forall j,t$$

Linearization

$$\gamma_{j\theta} - \gamma_{j\theta}^{\max}(1 - Z_{jt\theta}) \le h_{jt\theta} \le \gamma_{j\theta} - \gamma_{j\theta}^{\min}(1 - Z_{jt\theta}) \qquad \forall j,t,\theta = 1\ldots\tau_j$$
$$\gamma_{j\theta}^{\min} Z_{jt\theta} \le h_{jt\theta} \le \gamma_{j\theta}^{\max} Z_{jt\theta} \qquad \forall j,t,\theta = 1\ldots\tau_j$$
$$\gamma_{j\theta}^{\max} = \max_{l \in \zeta_l}(\bar{\lambda}_{jl}) + \tau_j \alpha_j \qquad \forall j \qquad \text{and} \qquad \gamma_{j\theta}^{\min} = \min_{l \in \zeta_l}(\bar{\lambda}_{jl}) \qquad \forall j$$

Revised Model

$$\lambda_{jt} = \sum_{\theta=1}^{\tau_j} \gamma_{j\theta} Z_{jt\theta} - \sum_{l=1}^{\zeta_l} \delta\gamma_{jl} E_{jl} \qquad \forall j,t$$
$$Z_{jt\theta} \le X_{j,t-\theta} \qquad \forall j,t,\theta = 1,\ldots,\tau_j$$
$$\sum_{\theta=1}^{\tau j} Z_{jt\theta} = E_j \qquad \forall j,t$$
$$X_{jt} \le E_j \qquad \forall j,t$$

Design Constraints

H.D. Goel et. al. Model and Revised Model Only

$$E_j = \sum_{k \in \psi_j} E_{jk} \qquad \forall j$$
$$V_j = \sum_{k \in \psi_j} \bar{V}_{jk} E_{jk} \qquad \forall j$$

Reliability Allocation Constraints

H.D. Goel et. al. Model Only

$$\gamma_{j1} = \sum_{\theta=1}^{\zeta_l} \bar{\lambda}_{jl} E_{jl} \qquad \forall j$$
$$E_j = \sum_{l \in \zeta_l} E_{jl} \qquad \forall j$$
$$\gamma_{j\theta} = \gamma_{j,\theta-1} + \alpha_j \qquad \forall j, 2 \le \theta \le \tau_j$$

Objective Function

E.N. Pistikopoulos et. al. Multiperiod Production and Maintenance Planning

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_i \sum_{j \in K_i} \sum_{\omega=0}^{p_i=1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$
$$\sum_{jt} C_{jt}^p X_{jt} - \sum_{jt} C_{jt}^c (H - U_{jt} - \Delta_j^p X_{jt}) / \Delta_j^p$$

E.N. Pistikopoulos et. al. Simultaneous Design, Production and Maintenance Planning

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_i \sum_{j \in K_i} \sum_{\omega=0}^{p_i=1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$
$$\sum_{jt} C_{jt}^p X_{jt} - \sum_{jt} C_{jt}^c (H - U_{jt} - \Delta_j^p X_{jt}) / \Delta_j^p - \sum_j (K_j^0 E_j$$
$$+ K_j^1 \sum_{k \in \Psi_j} \bar{V}_{jk} E_j)$$

H.D. Goel et. al., Design, Reliability, Production and Maintenance Planning Model and Revised Model

$$\max \Phi = \sum_{st} \eta_{st} D_{st} - \sum_{ut} C_{ut} \sum_{i} \sum_{j \in K_i} \sum_{\omega = 0}^{p_i = 1} (\beta_{uij\omega} N_{ijt} + \delta_{uij\omega} B_{ijt}) -$$
$$\sum_{jt} C^{p}_{jt} X_{jt} - \sum_{jt} C^{c}_{jt} (H - U_{jt} - \Delta^{p}_{j} X_{jt}) / \Delta^{p}_{j} - \sum_{j} (K^{0}_{j} E_j$$
$$+ K^{1}_{j} \sum_{k \in \Psi_j} \overline{V}_{jk} E_j + \sum_{l \in \zeta_j} K^{2}_{jl} E_{jl})$$

Appendix B – Example of GAMS File - Revised Design, Reliability, Production and Maintenance Planning Model

only one case is shown – no-FEED

```
*Model based of Problem 2 (P2) of Pistikopoulos
*Simultaneous Design, Production, and Maintenance Planning
*CASE 1 Uptime Constraint - Equipment can fail during both minimal repair and preventive maintenance.
*No Utility Constraints

*AGGREGATE MULTIPERIOD PRODUCTION
Sets
i      processing tasks                        /MAKEA, MAKEB, MAKEC/
j      equipment units                         /UNIT1, UNIT2, UNIT3/
t      time periods (months)                   /1*24/
s      states of material                      /A, B, C/
Ts(i,s)  set of tasks receiving material from state s    /MAKEB.A, MAKEC.A/
Tbs(i,s) set of tasks producing material in state s      /MAKEA.A, MAKEB.B, MAKEC.C/
Ij(i,j)  set of tasks for which unit j is suitable       /MAKEA.UNIT1, (MAKEB, MAKEC).(UNIT2, UNIT3)/
Ki(j,i)  set of units for which task i is suitable       /UNIT1.MAKEA, (UNIT2, UNIT3).(MAKEB, MAKEC)/;

*MAINTENANCE MODEL
Sets
THT      number of periods elasped last maintenance      /1*6/;


*DESIGN MODEL
Set
k      unit size                               /1*4/
l      initial failure rate                    /1*3/
PSIj(j,k) set of unit sizes available for unit j       /UNIT1.(1*4), UNIT2.(1*4), UNIT3.(1*4)/
IjPSIj(i,j,k) set of unit sizes                        /MAKEA.UNIT1.(1*4), (MAKEB, MAKEC).(UNIT2.(1*4),
UNIT3.(1*4))/
ZETA(j,l) set of possible initial failure rates for unit j/UNIT1.(1*3), UNIT2.(1*3), UNIT3.(1*3)/;


*AGGREGATE MULTIPERIOD PRODUCTION

Parameter p(i) setup and processing time (hours)
        /MAKEA      3
         MAKEB      2
         MAKEC      2.5/;

Scalar    H    duration of period  /720/;
Scalar  ALPHA  increase f. rate per period /0.001/;

Parameter ETA(s) unit price of s
       /A     -100
        B     0.5
        C     0.5/;

TABLE c(i,j)          capacity utilisation factors

                 UNIT1   UNIT2   UNIT3

      MAKEA          1
      MAKEB                 1       1
      MAKEC                 1       1;

*MAINTENANCE MODEL
Parameter DTAc(j) corrective maintenance (repair) duration of unit j
       /UNIT1      24
        UNIT2      40
        UNIT3      30/;

Parameter DTAp(j) preventative maintenance duration of unit j
       /UNIT1      6
        UNIT2      9
        UNIT3      7/;

Parameter TAU(j) maximum number of consecutive elaspsed time periods w\o maintenance of unit j
       /UNIT1      6
        UNIT2      6
        UNIT3      6/;
```

```
Parameter dL(j,l) initial failure rate options
      /UNIT1.1    0
       UNIT1.2    0.0005
       UNIT1.3    0.001
       UNIT2.1    0
       UNIT2.2    0.001
       UNIT2.3    0.002
       UNIT3.1    0
       UNIT3.2    0.0005
       UNIT3.3    0.001/;

Parameter GAM(j,THT)
     /  UNIT1.1       0.002
        UNIT1.2       0.003
        UNIT1.3       0.004
        UNIT1.4       0.005
        UNIT1.5       0.006
        UNIT1.6       0.007
        UNIT2.1       0.004
        UNIT2.2       0.005
        UNIT2.3       0.006
        UNIT2.4       0.007
        UNIT2.5       0.008
        UNIT2.6       0.009
        UNIT3.1       0.002
        UNIT3.2       0.003
        UNIT3.3       0.004
        UNIT3.4       0.005
        UNIT3.5       0.006
        UNIT3.6       0.007/;

Parameter Cp(j)    preventive maintenance cost of unit j during all periods
       /UNIT1         1000
        UNIT2         2000
        UNIT3         2000/;

Parameter Cc(j)    corrective maintenance cost of unit j during all periods
       /UNIT1         50
        UNIT2         100
        UNIT3         75/;

*DESIGN MODEL
Table Vcap(j,k) size k for unit j
             1       2      3      4
        UNIT1 150      175     200     250
        UNIT2 50       80      150     200
        UNIT3 60       100     125     200           ;

Parameter Nmax(i,j) maximum number of batches when task i is performed on unit j during period t

Parameter K0(j) fixed cost for unit j
      /UNIT1      5000
       UNIT2      20000
       UNIT3      20000/;

Parameter K1(j)  variable size factor
      /UNIT1      100
       UNIT2      300
       UNIT3      350/;

Table K2(j,l) cost factor for unit j with failure rate l over considered time horizon of planning
                 1       2      3
         UNIT1     0      2200    6000
         UNIT2     0      2200    6000
         UNIT3     0      2200    6000 ;

Nmax(i,j)$Ij(i,j)=(H*(1-(DTAc(j)*(GAM(j,'1')-smax(l,dL(j,l))))))/p(i);

Variables
   MU          objective function maximize
*AGGERGATE MULTIPERIOD PRODUCTION
   N(i,j,t)    number of batches of task i processed in unit j over time period t
   STR(s,t)    amount of material in state s in storage at the end of period t
   D(s,t)      amount of material delivered to external customers from state s over period t
   B(i,j,t)    amount of material undergoing task i in unit j during period t
```

```
*MAINTENANCE MODEL
   LAM(j,t)    actual failure rate of unit j during period t
   Z(j,t,THT)  1 during period t if unit j was maintained for the last time THT periods ago
   X(j,t)      1 if preventive maintenance is performed on j during period t
   U(j,t)      expected uptime of unit j during period t
*DESIGN MODEL
   E(j)        1 if unit j is chosen 0 otherwise
   Ecap(j,k)   1 if size k is chosen for unit j 0 otherwise
   Ebar(j,l)   1 if initial failure rate is chosen 0 otherwise
   V(j)        size of unit j
   EN(i,j,k,t) linearization factor;

Binary Variable   Z(j,t,THT);
Binary Variable   X(j,t);
Binary Variable   E(j);
Binary Variable   Ecap(j,k);
Binary Variable   Ebar(j,l);
Positive Variable V(j);
Positive Variable U(j,t);
Positive Variable N(i,j,t);
Positive Variable B(i,j,t);
Positive Variable STR(s,t);
Positive Variable D(s,t);
Positive Variable EN(i,j,k,t);

Equations
   PROFIT        Objective Function
*AGGERGATE MULTIIPERIOD PRODUCTION
   CapC1(i,j,t)   Capacity Constraints (Upper Bound)
*   CapC2(i,j,t)   Capactiy Constraints (Lower Bound)
   MB(s,t)        Material Balances
   RUC(j,t)       Resource Utilization Constraint
*MAINTENANCE MODEL
   FRA(j,t)       Actual Failure Rate Time Period t
   CON1(j,t,THT)  Failure Rate Constraint 1
   CON2(j,t)      Failure Rate Constraint 2
   CON3(j,t)      Failure Rate Constraint 3
   CON4(j)        Failure Rate Constraint 4
   UPC(j,t)       Uptime Definition Constraint
*DESIGN MODEL
   SELU(j)        Select Unit Constraint
   SELUS(j)       Select Unit Size Constraint
   LIN1(i,j,k,t)  Linearization Constraint 1
   LIN2(i,j,t)    Linearization Constraint 2;
*OBJECTIVE FUNCTION

PROFIT ..
       MU =e= SUM((s,t), ETA(s)*D(s,t)) - SUM((j,t), Cp(j)*X(j,t)) -
            SUM((j,t), Cc(j)*((H-U(j,t)-DTAp(j)*X(j,t))/DTAc(j))) -
            SUM(j,(K0(j)*E(j) + K1(j)*SUM(k$PSIj(j,k), Vcap(j,k)*Ecap(j,k)) + SUM(ZETA(j,l), K2(j,l)*Ebar(j,l))));

*AGGERGATE MULTIIPERIOD PRODUCTION

CapC1(i,j,t)$Ij(i,j) ..
       B(i,j,t) =l= c(i,j)*SUM(k$IjPSIj(i,j,k), Vcap(j,k)*EN(i,j,k,t));

D.up("A",t)=0;
D.up("B",t)=50000;
D.lo("B",t)=20000;
D.up("C",t)=50000;
D.lo("C",t)=20000;

RUC(j,t) ..
       SUM(i$Ij(i,j), p(i)*N(i,j,t)) =l= U(j,t);

MB(s,t) ..
       STR(s,t) =e= STR(s,t-1) + sum((i,j)$(ord(s) eq ord(i) AND Ij(i,j)), B(i,j,t))
       - sum((i,j)$(ord(s) eq 1 AND ORD(i) gt 1 AND Ij(i,j)), B(i,j,t)) - D(s,t);

STR.up("A",t)=0;


*MAINTENANCE MODEL

FRA(j,t) ..
```

```
      LAM(j,t) =e= SUM(THT$(ord(THT) le ord(t)), GAM(j,THT)*Z(j,t,THT)) - SUM(l, dL(j,l)*Ebar(j,l));

CON1(j,t,THT)$(ord(THT) le ord(t))..
      Z(j,t,THT) =l= 1 $ (ord(t) eq ord(THT)) + X(j,t-ord(THT))$(ord(t) gt ord(THT));

CON2(j,t)..
      SUM(THT$(ord(THT) le ord(t)), Z(j,t,THT)) =e= E(j);

CON3(j,t)..
     X(j,t) =l= E(j);

CON4(j)..
     SUM(l, Ebar(j,l))=e= E(j);

*Case 1 uptime constraint
UPC(j,t)..
      U(j,t) =e= H*(E(j)-DTAc(j)*LAM(j,t))-DTAp(j)*X(j,t);

*DESIGN MODEL
SELU(j)..
     E(j) =e= SUM(k$PSIj(j,k), Ecap(j,k));

SELUS(j)..
     V(j) =e= SUM(k$PSIj(j,k), Vcap(j,k)*Ecap(j,k));

LIN1(i,j,k,t)$IjPSIj(i,j,k)..
     EN(i,j,k,t) =l= Nmax(i,j)*Ecap(j,k);

LIN2(i,j,t)$Ij(i,j)..
      N(i,j,t) =e= SUM(k$IjPSIj(i,j,k), EN(i,j,k,t));

Model P1 /all/;

OPTION OPTCR = 0.03;
OPTION ITERLIM = 1000000;
OPTION MIP = CPLEX;

Solve P1 using MIP maximizing MU;

Parameter
VDEL    value of deliveries
TPM     total preventive maintenance  (maintenance costs)
DES     design costs
TCM     total corrective maintenance (repair costs)
KCOST   COST OF K2;

VDEL = SUM((s,t), ETA(s)*D.l(s,t));
DISPLAY VDEL;

TPM = SUM((j,t), Cp(j)*X.l(j,t));
DISPLAY TPM;

DES = SUM(j,(K0(j)*E.l(j) + K1(j)*SUM(k$PSIj(j,k), Vcap(j,k)*Ecap.l(j,k)) + SUM(ZETA(j,l),
K2(j,l)*Ebar.l(j,l))));
DISPLAY DES;

KCOST =  SUM(ZETA(j,l), K2(j,l)*Ebar.l(j,l));
DISPLAY KCOST;

TCM = SUM((j,t), Cc(j)*((H-U.l(j,t)-DTAp(j)*X.l(j,t))/DTAc(j)));
DISPLAY TCM;

DISPLAY Nmax;
```

Summary of Capstone Project

An effective chemical engineer strives to design and operate a chemical process that is efficient, safe, and environmentally-friendly. A well designed chemical process is one that is cost-effective and profitable; minimizes unnecessary risks to operators; and limits negative environmental impact. In order to maximize profitability a chemical engineer must maximize production of the desired product. This is accomplished by adjusting different aspects of a chemical process in order to find the best or *optimal* process. An optimal process limits negative aspects that lead to waste while augmenting positive aspects that increase productivity. These adjustable "aspects" are commonly referred to as *process variables*, due to the fact that their values can be changed. A chemical process that has been *optimized* will have the best possible combination of values for the different process variables. An example of a process variable could be the temperature or quantity of a reactant, or it could be the number of different reactors in the process. Process variables are an important aspect of this research project and are explained in detail in the report. For this summary, it is enough to understand that when an engineer optimizes a chemical process he or she is adjusting the process variables to find the best possible combination of values. An optimal chemical process will be highly efficient and will generate greater profits. Therefore, process optimization is a major concern for chemical engineers at the design stage and as a result, a great deal of research in the chemical engineering field is devoted to developing methods to optimize plant processes. This particular research project is concerned with creating an optimal chemical process based on different production schedules, maintenance schedules, unit reliabilities (how often a unit fails) and overall design considerations.

How does one go about optimizing a chemical process? There are a variety of methods available to develop an optimal chemical process: simply weighing the pros and cons of a particular process would be one course of analysis. However, qualitative analysis is relatively imprecise. A *quantitative* analysis is a more appropriate method, and computers make it feasible. In fact, computers are inherent to the research described in this paper, as well as to other complex optimization problems. How does one conduct a quantitative analysis of a chemical process and how are computers involved? A chemical process can be described algebraically through the use of a mathematical model: a set of interrelated equations, variables (process variables), parameters and constraints. For example, the material that flows through a particular process can be represented mathematically based on the law of the conservation of mass:

INPUT – OUTPUT + GENERATION = ACCUMULATION.

This is a simplified version of the equation used in this research project and it is just one of many equations that taken together can abstractly represent a complete chemical process. The abstract nature of a mathematical model is what defines it from other "normal" models. A mathematical model is nothing tangible, but as will be demonstrated, it still retains the same testability that is normally associated with a model rocket or model airplane.

In this research project, the mathematical model serves as input for a computer program that automatically adjusts the process variables of that model in order to maximize or minimize (optimize) an *objective value*. The objective value is a variable that relates all other components of the model into a single overarching relationship. This is done through the use of an *objective function*. The objective function is the equation that describes the relationship between all the different component parts of the model and the objective value. A profit equation is used in this research project. This makes sense because an optimal chemical process will maximize profit. A simplified equation is shown here:

$$\text{PROFIT} = (\text{PRICE} \times \text{PRODUCT}) - \text{COSTS}.$$

It should be noted that the "COSTS" term is actually the sum of several different costs associated with the chemical process. For the model used in this report, the "COSTS" term includes the cost of purchasing the reactants, the cost of performing maintenance, the cost of purchasing the different process units and the costs associated with the failure of the process units.

By adjusting the values of the different process variables in the mathematical model the computer program will find a maximum value for the PROFIT equation. Essentially, the computer program solves the objective function for different combinations of the values of the process variables and the optimal combination is the solution. A model is *fully optimized* when the computer program finds the best values for the process variables so that the profit reaches a maximum value.

The computer program that is used in this report is called General Algebraic Modeling System (GAMS). The reason that GAMS is chosen as opposed to other computer programs is due to the relative ease of inputting the mathematical model. The input for the model very closely replicates the way a mathematician would represent the mathematical model with pen and paper.

Mathematical programming is the branch of knowledge that deals with the optimization of mathematical models. In mathematical programming, the model can take on a variety of different forms: linear, non-linear, mixed-integer and mixed integer linear. The focus of this research project is on mixed integer linear problems (MILP). However, in order to understand the MILP form the others need to be discussed briefly.

A linear mathematical programming problem is one where there are no non-linear combinations of variables in the equations. In other words, variables can only be added or subtracted from each other. An example is provided below:

$$f(x) = X + Y + Z.$$

There is no multiplication or division of variables and there are no power functions (A^x); all of which are non-linear. One should note however the constants or scalars a, b and c can multiply or divide the variables without the equation losing its linearity:

$$f(x) = aX + bY + cZ.$$

Therefore, a linear model, or LP model is a mathematical model where all the equations are like the ones above and therefore, a non linear or NLP model would contain a scenario where any one of the equations is of the following form

$$f(x) = XY + Z.$$

where XY are variables that form a non-linear combination.

A mixed integer programming problem or MIP is a special case type of programming problem where the variables can only take on integer values: -2, -1, 0, 1, etc. The mixed integer linear programming problem or MILP is a combination of the MIP and LP problems; some variables are restricted to integer values while others can take on continuous values. This research project uses MILP models. Why does this research project use MILP models? Why not NLP or just LP models? As it turns out, LP problems are much easier to solve then NLP.

The easiest way to grasp this concept is to think of a rubber bouncy ball and a toy block. It is very hard to estimate an exact point on the bouncy ball because it's curved while it is relatively easy to determine an exact point on the toy block because it has points. The same goes for linear and non-linear models: the LP models have points while the non-linear model is curved like the ball. Optimization is similar to the act of finding an exact point on either the ball or the toy block. A detailed explanation of this idea that includes figures is covered in the paper.

The reason this research paper includes MIP components is a little complicated, but not beyond explanation. If we limit the MIP variables (integer variables) so that that they can only be a 0 or 1 then we can use those variables to indicate "Yes" or "No" for a particular part of the chemical process. For example, if during optimization the computer decides to include a process unit, such as a reactor, then the variable for that reactor would be set equal to 1 indicating "Yes" or "True." The computer program has automatically brought that reactor into existence; it has become part of the process. The use of this "Yes" or "No" aspect of mathematical modeling is extremely important to this research project. The use of MIP variables is expanded to other parts of the model including whether or not maintenance action is taken on a unit during a specific time period.

Finally, the last important piece of background information is the type of chemical process modeled in this research. This report analyzes *multipurpose process plants* and the distinguishing characteristic for this type of plant is that the individual process units can perform more than one task.

For example, a reactor may be able to produce two different types of product; though not necessarily at the same time.

As mentioned earlier, the focus on this research paper was on optimizing a chemical process by adjusting the production schedule, the maintenance schedule, the design of the process and the reliability of the design. Specifically, the project attempted to find a mathematical model that could better optimize all of the above. A model is "better" if it reduces the time it takes to solve the problem and if the accuracy of the model improves.

The methodology of the project was to first replicate the mathematical models presented in the literature. This was easily accomplished as the models presented in the literature successively built on the prior models. As such, the first model only included the production and maintenance schedule aspects. The second added design aspects and the third incorporated all four. Each model was replicated one at a time and the third model was improved upon to create a fourth and revisionist version.

The revision of the third model was a mathematical adjustment based on how the model "chose" the reliability or initial failure rates for a given process unit. The math used in the revised model was far simpler and therefore, the model was optimized quicker and with greater accuracy. The results are summarized in the fourth section of this project, but the revised model provided higher profit values, with less "resource usage" (a measure of time spent solving) and with a lower relative gap (a measure of accuracy). A quick note on the meaning of the "accuracy" of the optimized/solved model: as the models that are created are often incredibly complex the computer program estimates a "Best Solution" and then proceed to solve the model for that value. However, it is very difficult for the computer program to reach the estimate and therefore, the program will solve the model with an acceptable amount of error (or relative gap). A better model will be solved for a lower value of error, which is exactly what the revised model accomplishes.

At the beginning of this summary the importance of this project was briefly discussed: optimal chemical processes are more efficient and therefore, generate higher profits and create less waste. Indeed, these are very important results of optimization. However, the ability to apply these techniques to real-world scenarios is what gives the project material benefit. The optimization of real chemical plants could introduce new efficiencies that not only increases profit, but also reduces waste. The reduction of waste produced by chemical plants could have huge benefits for our environment. Furthermore, the abstract nature of the mathematical modeling means that the same procedures and methods used in this report can be used for any process. In fact, mathematical modeling and optimization is used in a variety of different disciplines including economics and operations management. Optimization is a very powerful tool: it allows for experimentation of abstract concepts while saving the time, money and labor required in other forms of analysis.