Combining Optimization and Simulation for Strategic and Operational Industrial Gas Production and Distribution

Wasu Glankwamdee
Jeff Linderoth
Jierui Shen
Lehigh University

Report No. 07T-004

Combining Optimization and Simulation for Strategic and Operational Industrial Gas Production and Distribution

Wasu Glankwamdee
Jeff Linderoth
Jierui Shen

Lehigh University
Department of Industrial and Systems Engineering
Bethlehem, PA

{wag3,jtl3,jis6}@lehigh.edu

July 5, 2007

Abstract

We describe flexible optimization models for the production and distribution of liquefied atmospheric gases. A simulation engine that interacts with the optimization models is used to test the effectiveness of the models in the presence of uncertainty in customer demand and product availability. The paper concludes with a case study using data from a large industrial gas company.

Keywords: Enterprise-wide Optimization; Simulation Optimization;

1 Introduction

Liquid oxygen (LOX) and liquid nitrogen (LNI) are typically produced in bulk through a cryogenic air separation process. One large industrial gas company plans its production and distribution of LOX and LNI to customers with the help of a linear program. The planning to set rough production levels and customer demand assignments is typically done at a monthly level. In operations, the company must dynamically adjust the rough monthly production plan as the state of the system evolves. With recent market conditions and competitive demands, the schedule adjustment is becoming more frequent and complicated. This leads to a natural question of *why* the schedule adjustment is so often necessary. There are two reasonable explanations for the necessity of frequent rescheduling:

1. Aggregation of customer demands into monthly time buckets does not account for the (known) normal monthly fluctuations of the demands and available supply, or
2. Uncertainty surrounding aspects of the model, e.g. customer demands and plant outages.

The company's course of action to resolve the frequent rescheduling issue is dependent on the reason *why* the rescheduling is necessary. In the first case, a finer-grain production-distribution model should be used, while in the latter case, the uncertainty should be explicitly considered in the model, perhaps through stochastic programming or robust optimization. To help the company study this and related questions involving strategic and operational aspects of its industrial gas business, we built various optimization models for the production and distribution of bulk liquid gases. Instances of the models with real and simulated data and with various time aggregations were solved in an effort to draw meaningful conclusions about the best course of action to reduce the frequency and severity of intra-month schedule adjustments.

Hand-in-hand with flexible optimization models for the production and distribution of industrial gases, a simulation tool was also built so that the quality of the proposed decisions from the optimization model could be assessed independent from the instance data used to create the model. This is an important point—just because the solution to an optimization model indicates that gains can be achieved does *not* necessarily imply that the gains will be achieved in practice. More convincing evidence is obtained by simulating the "improved" policies and measuring key performance indicators. This requires the development of a simulation model alongside the optimization model. In a recent article in a series on the *Great Unsolved Problems in Operations Research*, Fu (2007) argues that current software combining simulation and optimization "offers mainly marriages of convenience rather than well-integrated partnerships," and that attempts at combining the two "usually lead to one partner dominating the other." We agree with this assessment. In fact, we considering ourselves guilty of this marriage of convenience—the optimization models of our system are more sophisticated that the simulation. However, we have built the simulation system from the ground up, and the company will continue to develop the entities built as part of the simulation, hopefully fashioning the combined system into a marriage that truly works. Developing a simulation model has a number of additional benefits. The simulation engine can be used independently from the optimization model. It allows planners to perform what-if analysis, and planners can obtain entire distributions of performance statistics of the system under study.

Simulation has a long and distinguished history of use in the Chemical Engineering community. It has evolved into an indispensable and ubiquitous tool for plant design and analysis (Evans, 1987). More recently, optimization methods have been combined with the simulation models for a variety of problem faced by chemical engineers. The applications are too numerous to mention in full here, but the interested reader is directed to the references to simulation in the survey paper on Enterprise-Wide Optimization by Varma et al. (2007).

To our knowledge, this is the first work demonstrating the combination of simulation and optimization for liquid bulk-gas production distribution problems. A related work of Harper (2006) describes how Air Liquide uses an ant colony optimizer as part of its production and distribution planning for liquid gases. The contributions of this paper are the following:

- Definition of an approximate model for the production and distribution of bulk-industrial gases;
- Extension of the model to account for uncertainty in the system, both via a minimax formulation and a two-stage stochastic program;
- Development of a simulation engine capturing the production and customer demand uncertainties faced by the company,
- Combination of simulation with various optimization models to assess important strategic questions for an industrial partner, using both random and authentic data sets.

The remainder of the paper is divided into five sections. In Section 2, we give an overview of the company's production and distribution process for bulk atmospheric gases, and we describe three different optimization models that plan the production and distribution of gases: a linear program, a minimax version of the linear program, and a stochastic program. Section 3 describes a simulation engine developed to mimic the behavior of real production-distribution systems of atmospheric gases and explains how the simulation engine is used to measure the expected actual performance of various production-distribution

plans produced by the different optimization models. Section 4 explains a number of different experimental studies using the simulation-optimization engine undertaken on behalf of the company to help them understand the impact of various modeling approximations and strategic business decisions. Section 5 describes a case study of the simulation-optimization tool on a real-life system faced by the company. We offer some conclusions and directions of future work in Section 6.

2 The Production Distribution Model

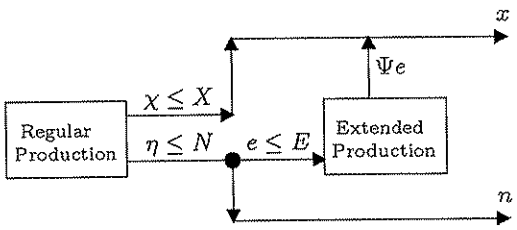
In this section, we describe a multi-period model for the production and distribution of bulk atmospheric gases. The model described here is only an approximation of the production-distribution problem faced by the company, but it does capture many of the important and distinctive features of such a model. Specifically,

- Plants can be modeled as operating in two different modes—a "regular" mode and an "extended" mode in which separated components of air are recycled in order to obtain a higher fraction of LOX than possible in regular separation modes;
- The company enters into "take-or-pay" contracts that allow competitors to take product from certain company sites, and the company can remove an equal amount of product from competitor sites;
- The distribution of products is constrained by the availability of drivers and trucks at each site.

The true model in place at the company is more complicated and more accurate, as it can account for nonlinearities from complications such as electricity contract terms and plant startups and shutdowns. The focus in this work is on the interplay of strategic decisions, operational considerations, and data uncertainty in the production-distribution process. For these purposes, the approximate models we use are sufficient.

A simplified model of production of LOX and LNI is depicted in Figure 1. In the figure, a "regular mode" amount of LOX (χ) and a "regular mode" amount of LNI (η) are produced. Each of these quantities is constrained via physical production limits: $\chi \leq X, \eta \leq N$. Next, depending on the plant setup, a (variable) amount (e) of LNI can be re-processed in order to produce an extra amount (Ψe) of LOX. The amount of LNI re-processed can be at most E. The total LOX that is produced is then $x = \chi + \Psi e$, while the final quantity of LNI that is produced is $n = \eta - e$. There is a limit on the total production capacity at the plant, expressed via the constraint $\chi + \eta \leq M$. For simplification, the variables $\chi = x - \Psi e$ and $\eta = n + e$ may be substituted

Figure 1: LOX-LNI Production



out, yielding the following constraints for production at a site:

$$\begin{aligned} x + n + (1 + \Psi)e &\leq M, \\ x - \Psi e &\leq X, \\ n + e &\leq N, \\ e &\leq E. \end{aligned}$$

2.1 Model Entities

The problem is characterized by a set of two bulk gas products $(\mathcal{P}) = \{\text{LOX}, \text{LNI}\}$ that are produced at various sites $(\mathcal{S})$, picked up or taken out from contract locations $(\mathcal{Q}, \mathcal{R})$ and delivered to customers $(\mathcal{C})$ over a time horizon $(\mathcal{T})$. In Figure 2 we list the sets of the optimization model. Even the simplified version of the model we present here contains a large number of parameters, and these parameters are described in Figure 3. In Figure 4, we list the decision variables of the optimization model.

Figure 2: Sets Over Which Model Entities are Indexed

- $\mathcal{P}$: Set of *Products* $\stackrel{\text{def}}{=} \{\text{LOX}, \text{LNI}\}$;
- $\mathcal{S}$: Set of *Sites*;
- $\mathcal{Q} \subseteq \mathcal{S}$: Competitor sites from which extra product may be picked up.
- $\mathcal{R} \subseteq \mathcal{S}$: Production sites from which competitors may take out product.
- $\mathcal{C}$: Set of *Customers*;
- $\mathcal{T}$: Set of *Time periods*. $\mathcal{T} \stackrel{\text{def}}{=} \{1, 2, \ldots, T\}$.

In reality, the number of driver and truck hours required to deliver one unit of product from site s to customer c is a complicated function of the routings used in the delivery. We approximate these quantities with the formula:

$$d_{sc} = \frac{2\delta_{sc}}{\varpi \Upsilon}, \tag{1}$$

where δ_{sc} is the distance from site s to customer c, ϖ is the estimated average truck speed, and Υ is the truck volume. The dollars per unit delivery cost are then computed as

$$f_{sc} = \zeta \delta sc + \gamma_s d_{sc},$$

where ζ is an estimated mileage cost (combining gas costs and truck depreciation), and γ_s is the (negotiated) driver hourly rate from site s.

2.2 Mathematical Program

Given the definitions in Figures 2—4, the problem of finding a minimum cost production and distribution schedule of the liquid gas products $\mathcal{P}$ meeting forecast customer demands is given by the following linear program:

Figure 3: Parameters in Optimization Model

M_{st}	Maximum total production amount at site s in period t
X_{st}	Maximum total LOX production amount at site s in period t
N_{st}	Maximum total LNI production amount at site s in period t
E_{st}	Maximum extended production amount at site s in period t. $E_{st} = 0\ forall s \in \mathcal{Q}$, as the company can not command extended production of LOX at competitor pick up sites.
K_{pst}	Truck hours available for shipping product p from site s in period t
D_{st}	Driver hours available at site s in period t
d_{sc}	Number of driver and truck hours required to deliver one unit of product from site s to customer c
f_{sc}	Dollars per unit delivery cost from site s to customer c
B_{pct}	Demand for product p from customer c in period t
I_{ps0}	Initial units of inventory for product p at site s
U_{ps}	Upper bound on inventory for product p at site s
α_{pst}	Regular mode per unit cost of producing product p. The production cost for pickup sites is $\alpha_{pst} = 0, \forall s \in \mathcal{Q})$, since by contract competitors will remove an equal amount of product from the company's sites.
β_{st}	Extended mode per unit cost at site s in period t
γ_{ps}	Inventory cost of product p at site s
Γ_{ps}	Inventory change fraction for product p at site s
Φ_p	Contract amount for product p for the planning horizon
z_{pst}	Amount of product p contract partner takes out at site s in period t. These can be estimated, but typically $\sum_{r \in \mathcal{R}} \sum_{t \in T} z_{prt} = \Phi_p$, so equal amounts are taken by competitors and given to competitors. Also, $z_{pst} = 0\ \forall s \in \mathcal{S} \setminus \mathcal{R}$, expressing that competitors may only take product from the designates take-out sites $\mathcal{R}$
Ψ_s	LOX/LNI conversion ratio at site s

Figure 4: Decision Variables in Optimization Model

x_{st}	Units of LOX produced at site s in period t
n_{st}	Units of LNI produced at site s in period t
e_{st}	Extended mode production quantity at site s in period t
I_{pst}	Inventory of product p at site s at the end of period t
y_{psct}	Units of product p shipped from site s to customer c in period t
u_{pct}	Inventory of product p at customer c by the end of period t
v_{pct}	Shortage of product p for customer c by the end of period t

$$\min \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\left(\alpha_{\text{LOX},st}x_{st}+\alpha_{\text{LNI},st}n_{st}+\beta_{st}e_{st}+\sum_{p\in\mathcal{P}}\gamma_{ps}I_{pst}\right) + \sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}(f_{sc}y_{psct}) + \sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}(\delta_{pc}v_{pct})$$

$$\begin{aligned}
\text{s.t} \quad x_{st}+n_{st}+(1+\Psi_s)e_{st} &\leq M_{st}, \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (2)\\
x_{st}-\Psi_s e_{st} &\leq X_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (3)\\
n_{st}+e_{st} &\leq N_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (4)\\
e_{st} &\leq E_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (5)\\
\sum_{s\in\mathcal{S}} y_{psct}+u_{pc,t-1}-v_{pc,t-1}-B_{pct} &= u_{pct}-v_{pct} \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T}\setminus\{1\} && (6)\\
\sum_{s\in\mathcal{S}} y_{psc1}-B_{pc1} &= u_{pc1}-v_{pc1} \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C} && (7)\\
\sum_{c\in\mathcal{C}} y_{\text{LOX},sct}-I_{\text{LOX},s,t-1}-x_{st}+I_{\text{LOX},st} &= -z_{\text{LOX},st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (8)\\
\sum_{c\in\mathcal{C}} y_{\text{LNI},rct}-I_{\text{LNI},s,t-1}-n_{st}+I_{\text{LNI},st} &= -z_{\text{LNI},st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (9)\\
\sum_{q\in\mathcal{Q}}\sum_{t\in\mathcal{T}} x_{qt} &\leq \Phi_{\text{LOX}} && (10)\\
\sum_{q\in\mathcal{Q}}\sum_{t\in\mathcal{T}} n_{qt} &\leq \Phi_{\text{LNI}} && (11)\\
\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}} d_{sc}y_{psct} &\leq D_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (12)\\
\sum_{c\in\mathcal{C}} d_{sc}y_{psct} &\leq K_{pst} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (13)\\
I_{pst} &\leq U_{ps} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (14)\\
I_{psT} &\geq \Gamma_{ps}I_{ps0} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S} && (15)\\
x_{st}, n_{st}, e_{st} &\geq 0 \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (16)\\
I_{pst} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T} && (17)\\
y_{psct} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T} && (18)\\
u_{pct}, v_{pct} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T}. && (19)
\end{aligned}$$

The objective function minimizes a combination of the production costs, inventory costs, distribution costs, and penalty terms for not meeting a customer's entire demand. Constraints 2—5 are the constraints to approximately model the production process at each site and each time period. Constraints 6 and 7 ensure that customer demand for each product will be met in each period or a shortage will be tracked. Constraints 8 and 9 are the inventory balance constraints at each site in each period for LOX and LNI respectively. Constraints 10 and 11 limit the total amount of each product picked up from contract sites over the planning horizon. Constraints 12 and 13 enforce that number of driver/truck hours required to deliver the products from each site are not greater than the available driver/truck hours, respectively. Note that the truck hours are constrained for each product, as the trucks are product specific. Constraint 14

enforces an upper bound on the inventory of each product at each site and in each time period, and Constraint 15 states that the ending inventory must be at a certain percentage of the initial inventory.

The take-or-pay parameters z_{pst} pose a particular modeling challenge. They are not entirely variables, in the sense that the company cannot determine when, where, and how much its competitor can remove from the system. Nor are they parameters that can be specified *a priori* with great certainty. In reality, there is a negotiation that takes place during the month to determine reasonable amounts and locations. Likewise, the amounts taken out from competitor sites $y_{psct}, s \in \mathcal{Q}$ is not *entirely* a variable in this model, as these amounts need to be negotiated. It is treated as a variable in our system, however. One potential use of the simulation-optimization system is to examine the impact of take-or-pay contracts, and we give preliminary results along this direction in Section 4.3.

2.3 Modeling Uncertainty

There are many significant sources of uncertainty in atmospheric gas production and distribution. For example, customer demands B_{pct} can only be forecast and may not be known with reasonable accuracy, the competitor product-removal amounts z_{pst} cannot be directly controlled, and the production capacities M_{st} may unexpectedly drop to zero should there be a plant failure.

In order to address the uncertainty explicitly via optimization, new models were built. Uncertainty in our models is represented with a set of *outcomes* $\mathcal{O}$. The outcomes are possible realizations of demands for the customers B_{pcto} and possible competitor removal amounts z_{psto}. A future use of the model and simulation engine will be to hedge against unplanned plant outages, in which case the outcomes can be augmented to force plant production maximum quantities $M_{sto} = 0$.

There are many ways to create optimization models that attempt to hedge against uncertainty. One simple computational model that fairly accurately reflects the production reality is to ensure that the production quantities of LOX and LNI are the same regardless of the outcome, while the deliveries and inventories are allowed to vary based on the outcome. Gas production is a unique process in that the production levels can be changed with relatively little lead-time or without the necessity of incurring great additional costs, so it may seem overly conservative to not allow these decision variables to depend on the outcome. However, in the simulation used to *truly* assess the performance of a particular optimization model, the production rates are allowed to vary on a daily basis.

2.3.1 Minimax Model

The first extension of the model presented in Section 2.2 is a *minimax model* whose aim to to find a schedule that will minimize the maximum production and distribution cost over all the outcomes. To model this extension, we introduce (and minimize) a new variable η representing the value of the maximum cost outcome, and simply enforce that η is at least as large as the (production-distribution) cost in each outcome. Williams (1999) gives an explanation of this well-known modeling artifice.

$$\min \eta$$

$$\text{s.t} \qquad \eta \geq \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \left(\alpha_{\text{LOX},st} x_{st} + \alpha_{\text{LNI},st} n_{st} + \beta_{st} e_{st} + \sum_{p \in \mathcal{P}} \gamma_{ps} I_{psto} \right)$$

$$+\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}(f_{sc}y_{pscto})+\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}(\delta_{pc}v_{pcto}),\quad \forall o\in\mathcal{O} \tag{20}$$

$$\begin{aligned}
x_{st}+n_{st}+(1+\Psi_s)e_{st} &\leq M_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T}, && (21)\\
x_{st}-\Psi_s e_{st} &\leq X_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T}, && (22)\\
n_{st}+e_{st} &\leq N_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T}, && (23)\\
e_{st} &\leq E_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T}, && (24)\\
\sum_{s\in\mathcal{S}} y_{pscto}+u_{pc,t-1,o}-v_{pc,t-1,o}-B_{pcto} &= u_{pcto}-v_{pcto} \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T}\setminus\{1\},\ \forall o\in\mathcal{O}, && (25)\\
\sum_{s\in\mathcal{S}} y_{psc1o}-B_{pc1o} &= u_{pc1o}-v_{pc1o} \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C},\ \forall o\in\mathcal{O} && (26)\\
\sum_{c\in\mathcal{C}} y_{\mathrm{LOX},scto}-I_{\mathrm{LOX},s,t-1,o}-x_{st}+I_{\mathrm{LOX},sto} &= -z_{\mathrm{LOX},sto} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (27)\\
\sum_{c\in\mathcal{C}} y_{\mathrm{LNI},rcto}-I_{\mathrm{LNI},s,t-1,o}-n_{sto}+I_{\mathrm{LNI},sto} &= -z_{\mathrm{LNI},sto} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (28)\\
\sum_{q\in\mathcal{Q}}\sum_{t\in\mathcal{T}} x_{qt} &\leq \Phi_{\mathrm{LOX}} && (29)\\
\sum_{q\in\mathcal{Q}}\sum_{t\in\mathcal{T}} n_{qt} &\leq \Phi_{\mathrm{LNI}} && (30)\\
\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}} d_{sc}y_{pscto} &\leq D_{st} \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (31)\\
\sum_{c\in\mathcal{C}} d_{sc}y_{pscto} &\leq K_{pst} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (32)\\
I_{psto} &\leq U_{ps} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (33)\\
I_{psTo} &\leq \Gamma_{ps}I_{ps0o} \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall o\in\mathcal{O} && (34)\\
x_{st},n_{st},e_{st} &\geq 0 \quad \forall s\in\mathcal{S},\ \forall t\in\mathcal{T}, && (35)\\
I_{psto} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (36)\\
y_{pscto} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall s\in\mathcal{S},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (37)\\
u_{pcto},v_{pcto} &\geq 0 \quad \forall p\in\mathcal{P},\ \forall c\in\mathcal{C},\ \forall t\in\mathcal{T},\ \forall o\in\mathcal{O} && (38)
\end{aligned}$$

2.3.2 Stochastic Model

A simple two-stage stochastic model can be made from the minimax model simply by removing the variable η that is used to measure the worst-case cost and instead replace the objective function with the average cost incurred over all the outcomes. Specifically, the constraints 20 are removed, and the objective is replaced with

$$\min \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\left(\alpha_{\mathrm{LOX},st}x_{st}+\alpha_{\mathrm{LNI},st}n_{st}+\beta_{st}e_{st}+|\mathcal{O}|^{-1}\sum_{o\in\mathcal{O}}\sum_{p\in\mathcal{P}}\gamma_{ps}I_{psto}\right)$$
$$+\sum_{p\in\mathcal{P}}\sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{o\in\mathcal{O}}|\mathcal{O}|^{-1}(f_{sc}y_{pscto})+\sum_{p\in\mathcal{P}}\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}\sum_{o\in\mathcal{O}}|\mathcal{O}|^{-1}(\delta_{pc}v_{pcto}).$$

More sophisticated (multi-stage) models that capture the decision-event process more accurately are being considered as a future extension of this work.

2.3.3 Outcome Generation

The outcomes (or scenarios) are generated by sampling from the customer demand distributions and the competitor removal distributions. More details of how this is implemented is given in subsequent sections. The outcomes could also be chosen specifically by the company as coming from a worst-case set of scenarios. The models on which we experiment in Sections 4 and 5 contain only a small number of outcomes. There is some theoretical and empirical evidence that for two-stage problems involving uncertainty, a very small number of outcomes is necessary to ensure that a good solution is achieved with high probability (Shapiro and Homem-de-Mello, 2000; Linderoth et al., 2006). Limiting the number of outcomes minimizes the increased computational overhead incurred by these more complicated optimization models.

3 Simulation

A fundamental issue that limits the broad acceptance of optimization as a tool for decision making is the ability to convince a user that a solution optimized for an abstract model is a useful solution in reality. Moreover, the optimization models for production and distribution of industrial gases introduced in Section 2 are by themselves of little value to the company, as the models alone do not help the company understand the impact of uncertainty or the impact of planning production and distribution in a time-aggregated fashion. Having a reliable simulation engine can help bridge the gap between theory and practice and can give evidence as to the practical quality of solutions obtained from the optimization models. In addition, with a simulation, entire *distributions* of important statistics about the system can be produced, yielding valuable insights that could not otherwise be obtained. In the case we study here, distributions of the estimated production cost, the delivery cost, the number of customer outages, and the average site and customer inventories can be collected and displayed in a useful graphical format. Managers making decisions about production and distribution can compare different solutions with regards to the distributions of these key performance indicators before deciding which policy to follow.

3.1 Simulation Objects

The engine built to simulate the production-distribution process in place at the company consists of over 4000 lines of C++ code. The program is constructed of objects representing physical entities or attributes in the company's planning problem. The objected-oriented features of C++ used for abstracting certain objects. In this section, we briefly describe features and methods of the most important objects comprising the simulation.

`Site`: The `Site` class contains information about a site. A site may be of one of thee types: A competitor pick-up site, indexed in the optimization model by the set $\mathcal{Q}$, a take-out site, indexed in the optimization model by the set $\mathcal{R}$, or neither of these ($\mathcal{S} \setminus \mathcal{Q} \setminus \mathcal{R}$). The site object contains the location of the site, the number of drivers and trucks available at the site, and all operating parameters for production at the site: $(M_{st}, X_{st}, N_{st}, E_{st}, U_{ps})$.

`Customer`: The `Customer` class contains attributes for the customer's location, the capacity of its tank, a safety-stock inventory level, and the inventory level at the beginning of the planning horizon. Each customer has a demand for one type of product, either LOX or LNI. For information about a customer's usage pattern for this product, the `Customer` class also contains an abstract object known as a `DemandDistribution`. The `DemandDistribution` class is subsequently explained in more detail.

`InstanceFamily` A design principle of the optimization-simulation engine was to allow for the creation, solution, and simulation of different optimization instances all representing the same physical system. To allow for different optimization instances to be created from the same production-distribution problem and the resulting policies simulated, the entities in the problem are collected into a class known as an `InstanceFamily`. The `InstanceFamily` class is an object to which the simulation can be applied, so the class has attributes that record the state of the production-distribution system when it is being simulated. For example, the current day of the simulation and the production and delivery costs incurred to date are attributes in the `InstanceFamily` class.

`DemandDistribution`: The `DemandDistribution` is an abstract class via which a variety of different customer product usage patterns can be modeled. In order for the results of the experimental simulations to be of use to the company, it is vital that the uncertain parameters of instances behave as they do in practice. To that end, a customer's usage pattern can be classified into one of four main categories:

1. Normally-Distributed: The customer's daily demand B_c is modeled as a normally distributed random variable with mean μ_c and variance σ_c^2, $B_c \sim \mathcal{N}(\mu_c, \sigma_c^2)$.

2. Specific-Needs: Customer c has a typical daily demand that is normally distributed with mean μ_c and variance σ_c^2. However, for certain periods during the planning horizon (perhaps when a new production process is being tested), the customer requires a significantly larger daily volume: which can be modeled as a normally distributed random variable with mean μ_c' and variance $\sigma_c^{2'}$.

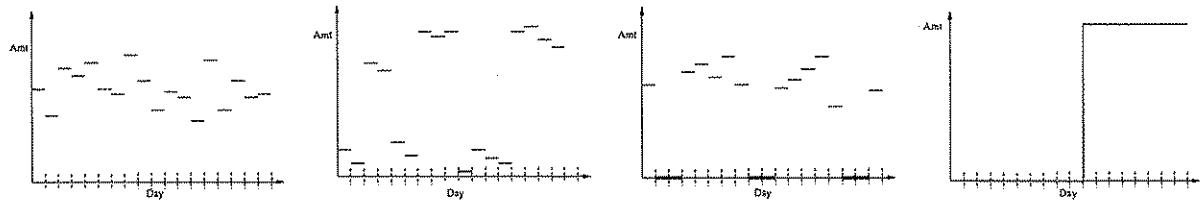
3. On-Off: For a majority of days (e.g. Monday-Friday), this class of customer has a daily demand that can accurately be modeled with a normally distributed random variable, $B_c \sim \mathcal{N}(\mu_c, \sigma_c^2)$. However, on the other days, the customer does not use any product.

4. Call-In: The class of customer is used to model the customer whose daily demand is typically zero. But then at certain points during the planning horizon, for a specified number of days, the customer will require a (known, deterministic) daily demand of m_c.

Figures 5(a)—5(d) show a sample demand distribution for each type of customer. For normally-distributed customers, the daily usage amounts may accurately be treated as independent random variables. However, for the other classes of customers, there is a correlation between the usage amounts on consecutive days, thus, when simulating the customer usage patterns over the course of a month, it is required to keep track of what day it is. Again, the current date of the simulation is stored in the `InstanceFamily` object.

The `DemandDistribution` class has a virtual method

```
vector<double> getDemandProfile(vector<int>& daysInEachPeriod)
```



(a) Normally Distributed (b) Specific Needs (c) On-Off (d) Call-In

Figure 5: Sample Realizations of Customer Demands

that must be implemented. The argument to the method is a container of integers representing the number of days in each planning period. The method returns a sample realization of the customer's demand for each of the period. Concrete implementations of the `DemandDisribution` class were created to implement the distributions shown in Figures 5(a)—5(d): `NormalDemandDistribution`, `ZeroOrNormalDemandDistribution`, `SpecificNeedsDemandDistribution`, and `CallInDemandDistribution`.

These classes are used when a random production-distribution system is to be simulated, but in case the simulation engine is to be instantiated with real customer data, an additional implementation is required. The company's database does not categorize the customers into the classes described above, but it does contain forecast and historical demands for the customers from their ERP system. An interesting question we faced in this work was *how* to (easily) represent distribution information for the uncertain entities in the model. In fact, Greenberg (2007) posed the representation of uncertainty in OR models as a *Great Unsolved Problem in OR*. For our purposes, we chose to model each customer's demand as containing both a daily part that is normally distributed with specified mean and variance and also a temporal part that captures the portion of the customer demand that is correlated in time. Mathematically, a customer's demand on day d of the planning horizon is computed as

$$B_{cd} = \max\{0, \xi_c + I_c(d)\},$$

where $\xi \sim \mathcal{N}(\mu_c, \sigma_c^2)$ and $I_c(d)$ is a random variable for specifying the amount of customer c's demand that is dependent on day d. For input purposes, the random variable $I_c(d)$ is given as a list of possible *scenarios* for each customer. For each scenario, the probability of that scenario occurring is given, as is the customer's daily demand for each day in the planning horizon in that scenario. These scenarios are sampled to create realizations of the demand. As implemented in the simulation engine, these customer demands are encapsulated using the class `NormalWithImpulseDemandDistribution`.

The quantity of product that a competitor removes from a take-out site z_{pst} is also simulated via a `DemandDistribution` object.

`Instance`: The `Instance` class is an object representing a mathematical program than can be solved. The `Instance` class is an abstract class, as it can be used to represent the nominal production-distribution model of Section 2.2, the minimax model of Section 2.3.1, or the stochastic program of Section 2.3.2. In each case, the optimization models are created using the Mosel modeling language from Dash Optimization (Colombani and Heipcke, 2002). In particular, the engine makes use of the Mosel Run Time Library to execute models that need to be solved as part of the simulation process and to read the results of the solution back from the linear programming solver.

In order to create a concrete `Instance` object, the user must implement the method `void writeMosel(string &fileName)`, which writes the data describing the instance to a data file that can be read by XPRESS. The second method that must be implemented is the method `int setSolution(XPRMmodel &ewoMod)`, which takes the optimization instance object from XPRESS, reads the solution, and fills appropriate solution containers with objects corresponding to the decisions variables that are greater than zero. Most important are the `RegProductionAmount` class for the decision variables relating to production (x_{st}, n_{st}) and the `Delivery` class for the solution variables (y_{psct}). The `writeMosel` and `setSolution` methods were written for each of the types of instances we wish to simulate, resulting in the classes `NominalInstance`, `MinimaxInstance`, and `StochasticProgram`. Each instance also has an associate Mosel model file describing the model.

Instances of varying time-grain can be created with the method

```
void create(const InstanceFamily &instanceFamily, const vector<int>&daysInEachPeriod).
```

The user must specify the number of period in the planning model and the days in each of the planning periods. The information of sites and customers contained in `InstanceFamily` object are then aggregated into periodic time buckets accordingly.

3.2 Simulation Execution

The production and distribution process is simulated over a period of length δ referred to as the *simulation increment*. Typically δ is one day. The process repeats until the end of the planning horizon. Statistics about the total production cost, distribution cost, and number of customer outages are collected and reported at the end of the simulation. An outline of the execution of one loop of the simulation is given in Figure 3.2. In this section, we also give more details of the specific steps of the simulation.

Figure 6: Simulating Production-Distribution

Step 1. An optimization instance is created given the initial operating conditions (site inventories and customer inventories) and estimated customer demands and competitor product removal amounts.

Step 2. The optimization instance is solved, yielding production quantities for each sites the `RegProductionAmount` objects and sourcing decisions for each customer demand, the `Delivery` objects, as created by the method `Instance::setSolution()`.

Step 3. The (simulated) site inventories are increased by a quantity related to the suggested by the solution to the optimization instance

Step 4. The resources hours (for both drivers and trucks) are incremented by an appropriate quantity

Step 5. Competitor quantities are taken out from the sites

Step 6. Deliveries are made

Step 7. Customer inventories are updated, and customer outages are counted. Return to Step 1.

Steps 3&4. Let τ be the number of days in the first period of the optimization model producing suggested production amounts $\hat{x}$ and $\hat{n}$. The site inventories are increased by an amount $\hat{x}_{s1}\delta/\tau$ and $\hat{n}_{s1}\delta/\tau$ for each site s. This is equivalent to having the simulation use the suggested production amounts from the optimization model, but pro-rated to run at this rate only for δ days. Similarly, at each loop of the simulation, the site is "renewed" with an driver and truck hours with which it can feasibly make deliveries.

Step 5. Random quantities resampled from the same distribution that is used to derive the z_{pst} parameters in the optimization model are taken from each of the sites designated as competitor pickup sites ($s \in \mathcal{Q}$). The `getDemandProfile` method of the `DemandDistribution` object is used to implement this step of the algorithm, by asking for a demand profile for δ days.

Step 6. The suggested deliveries from the model $\hat{y}_{psct}$, as collected in the container of `Delivery` objects, are used to suggest the sourcing decisions. In reality, the company does not deliver exact amounts $\hat{y}_{psct}$ to its customers, but rather the scheduling of trips and deliveries is a complicated process that takes into account delivery rules and routing costs. We approximate the delivery process in the simulation as follows.

First, the deliveries $\hat{y}_{psct}$ are priority-sorted such that deliveries for customers that will have the lowest expected inventory if the delivery is not made are attempted first. Deliveries for which it is not expected that the customer inventory will fall below the customer safety-stock volume are not made. Deliveries to customers are made in increments of 1/2 truckloads in order to meet the predicted demand for the delivery period. The hours used to make the delivery are estimated in a similar fashion to equation 1, and the product is removed from the site and the resources hours for that site are reduced. If a site does not have enough product or resource in order to make the delivery suggested by the solution to the optimization instance, an auxiliary delivery will be made from the next closest site that has the resources necessary to make the delivery.

Step 7. After the deliveries are made, the true customer demands are revealed and the inventories are updated. At this point the simulation checks to see if a customer's inventory has fallen below a minimum "run out" volume. A customer may be run out in the simulation for one of two reasons. The first reason is because there was not enough product produced anywhere in the system for a delivery to be made. This rarely happens. The more likely reason for a customer outage in the simulation is that the true customer demand is quite different than the predicted demand that was used to make the delivery, and for which the optimization instance was created.

4 Computational Experiments

In this section, we described the results of three experiments designed to both demonstrate the flexibility and utility of the simulation-optimization engine and to address open questions about the planning process at the company. First, we attempt to provide insight as to the impact of planning production-distribution using a more finely-aggregated model. Next, we assess the impact of demand variability and show that the simple minimax and stochastic programming models we developed can significantly help

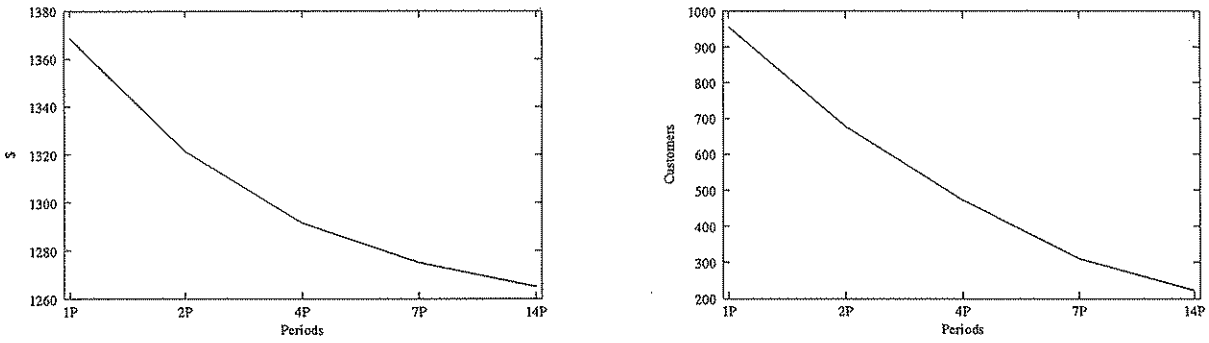
in this regard. Finally, we demonstrate how the simulation-optimization engine could be used as a tool for contract pricing for take-or-pay arrangements.

4.1 Time Aggregation

The first question we attempt to answer for the company with the simulation-optimization engine is the following: "Does a finer time-grain model lead to less manual intervention in the scheduling during the planning process?" We consider two useful statistics from the simulation in order to measure the amount of manual intervention that would be required in reality. The first is the number of customer outages reported by the simulation, and the second is the average cost per delivery. In reality, in these cases, production planners would need to manually adjust production and deliveries to ensure that customers did not run out of product, and if the average cost of delivery increases in the simulation, it typically indicates that the product is not being delivered from the "primary" site suggested by the optimization.

For this experiment, 10 random `InstanceFamilies` were created, each consisting of 10 sites and 200 customers for which we will simulate behavior over a planning horizon of 28 days. The simulated site s had a 20% and 30% chance of being designated as a 'Pick up' location ($s \in \mathcal{Q}$) or a 'Take out' location ($s \in \mathcal{R}$), respectively. The simulated customer had a 75% probability of being a LOX customer and a 25% chance of being a LNI customer. Customer demand distributions were chosen to be of each type described in Section 3.1 with the following probabilities: 33% Normally Distributed, 12% Specific Needs, 50% On-Off, and 5% Call-In. The mean customer demand μ_c are chosen uniformly and randomly from an interval that is typical of the company's customers, and the customer demand variance (for those distributions that require it) was (randomly) set to be between 5% and 50% of the mean. For each of these random systems, the demand and production data were aggregated so as to make different instances of the model (for the same problem) of varying time fidelity. Specifically, multi-period models consisting of 1 period, 2 periods, 4 periods, 7 periods, and 14 periods were created and simulated.

Figures 7(a) and 7(b) show the average cost per delivery and average number of customer outages for the 10 simulated systems using each fidelity of planning model for the optimization.



(a) Total Cost Per Delivery

(b) Number of Customer Outages

Figure 7: Effectiveness of Finer-Grain Model

The results of this experiment are quite striking and clearly show that there is potential benefit for the company in considering additional periods in the planning model. This is not a trivial undertaking at the company, as the true models used for planning are larger and more complex. Thus, adding many

additional periods might lead to unacceptable computational times. However, based on the results of this study, the company is currently considering making a change to more multi-period models.

4.2 Dealing with Uncertainty

The second experiment is aimed at quantifying the impact of uncertainty on the production-distribution process, and studying whether it can be reasonably controlled with the simple models we suggest in Sections 2.3.1 and 2.3.2. Specifically, we demonstrate how the total simulated cost and number of customer outages change as a function of variance of customer demand.

In this experiment, we compare the effectiveness of three models, the nominal model of Section 2.2, the minimax model of Section 2.3.1, and the stochastic program of Section 2.3.2. For each model, four-periods were used ($|\mathcal{T}| = 4$), and five outcomes were used ($|\mathcal{O}| = 5$) for both the minimax model and stochastic program. Again, 10 random `InstanceFamilies` objects were created as in the experiment in Section 4.1. All parameters used for creation of the objects were the same, save for the variance of customer demand. In this case, customer demand variance was allowed to increase by setting the upper bound of each customer's variance to be 25%, 50%, 75%, and 100% of the mean.

Figure 8 and 9 show the average total simulated production-distribution cost and the average number of customer outages for different customer demand variability levels. In these figures, the 95% confidence intervals on the values are plotted as vertical bars.

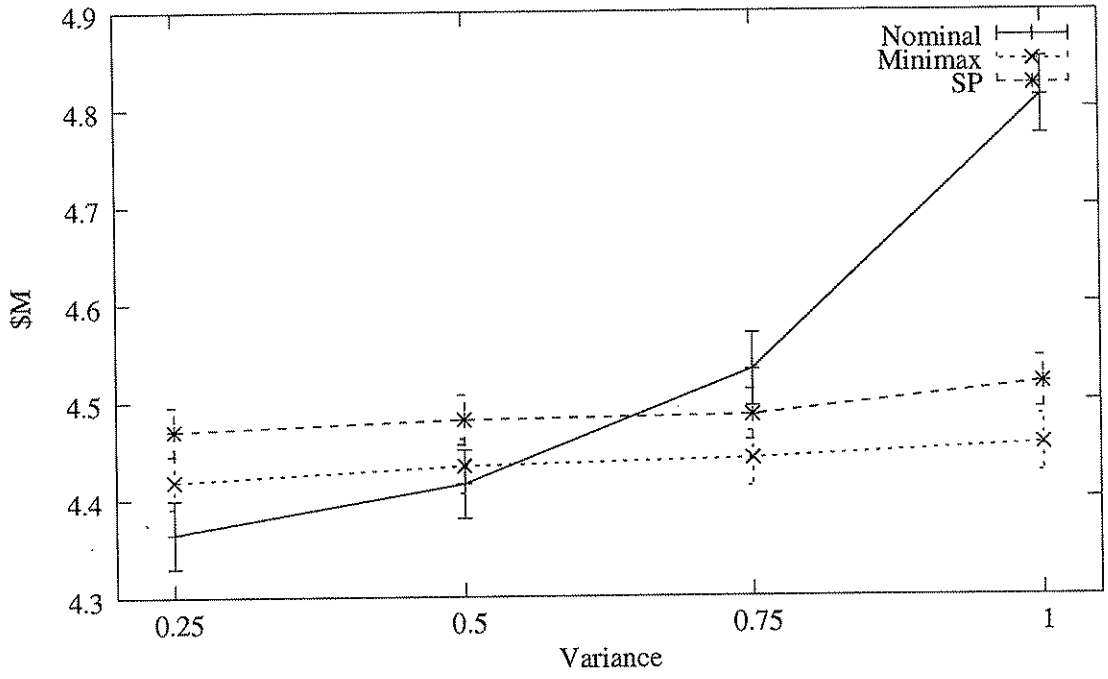


Figure 8: Total Cost

Two results of the experiment are worthy of note. First, it was striking that both the minimax model and stochastic program were able to mitigate the high uncertainty remarkably well, even considering an extremely low number of outcomes in the optimization model. The nominal model has great difficulty

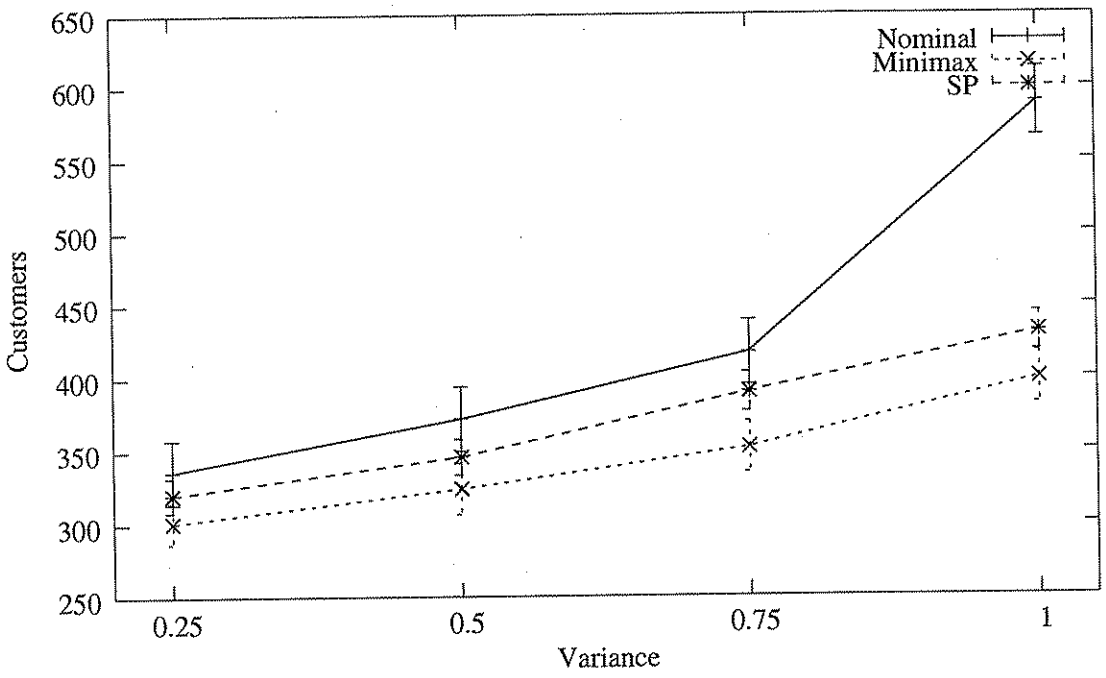


Figure 9: Number of Customer Outages

planning effectively as the variance increases. Second, the "conservative" minimax model outperformed the stochastic program when the resulting policies were simulated, even in terms of total cost.

4.3 Contract Pricing

The third experiment is aimed to demonstrate the utility of the optimization-simulation engine. Specifically, the engine can be used to estimate the value of entering into "take-or-pay" contract agreements with competitors.

To estimate the value of a take-or-pay contract, we simply simulate the same `InstanceFamily` object with and without a contract arrangement in place. The "take-or-pay" contracts can be withdrawn by setting the Φ_p and z_{prt} parameters to zero.

In this case, we used a four-period nominal model for planning on a random `InstanceFamily` object created using the parameters described in Section 4.1. Table 4.3 compares the average total cost and the average number of customer outages with and without the contract agreements.

	Total Cost ($M)	Production Cost ($M)	Delivery Cost ($M)	Number of Customer Outages
With Contract	4.42	2.47	1.95	372
No Contract	4.51	2.48	2.03	366

Table 1: Effectiveness of Contract Agreements

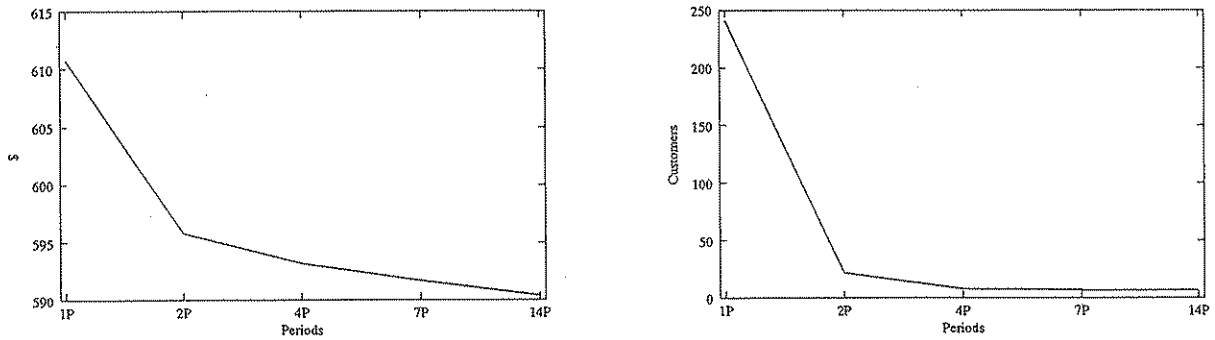
Without contract, we are still able to satisfy customer demand equally well. However, the average

total cost increases mostly from the fact that some of the customers are served by another site further away incurring higher delivery cost. In this case, the simulation-optimization engine has estimated the value of the contract at 4.51M$ - 4.42M$ = .09M$/month.

5 A Case Study

In this section, we present results of a case study conducted for the company. The company provided us with data from a small region that would be typical of the planning models they face worldwide. The region consists of 7 sites ($|\mathcal{S}| = 7$), 3 of which are competitor sites, ($|\mathcal{Q}| = 3$), and there are 823 customers in the region ($|\mathcal{C}| = 823$). A majority of the customers are of the normally-distributed variety, but 10 customers have demand distributions that could be accurately modeled as being independent each day. As discussed in Section 3.1, these customer demand distributions were simulated using `NormalWithImpulseDemandDistribution` objects, and the company also provided us with between two and five different scenarios for possible customer demands over the course of the planning horizon.

On this real data set, we performed the same three experiments as for the random data sets described in Section 4. The first experiment tested the effectiveness of multi-period versions of the nominal planning model of Section 2.2. Figures 10(a) and 10(b) demonstrate that the company benefits from using the finer-grain model to save on the average total cost per delivery and better serve the customers. However, in the case of real data, it seems that the marginal benefit of increasing the model to more than four periods is small.



(a) Total Cost Per Delivery

(b) Number of Customer Outages

Figure 10: Effectiveness of Finer-Grain Model

The next experiment tested the effectiveness of minimax and stochastic programming models. Like in Section 4.2, four-period versions of the nominal model, the minimax model (with five outcomes), and the stochastic program (with five outcomes) were simulated. Table 5 summarizes the results for this experiment. Like in the random instances, it seems that use of the conservative minimax model for planning production and distribution could lead to lower costs.

The final experiment measured the value of the contract agreement for the region by simulating the cost of the four-period nominal model both with and without the contract agreement in place. Table 5 shows that the cost of the agreement in this case is roughly $.03M/month.

All results in the case study coincide with the simulated instances. The company can reduce the

Model	Total Cost ($M)	Number of Customer Outages
Nominal	2.03	7
Minimax	1.94	5
SP	2.02	7

Table 2: Effectiveness of Different Models

	Total Cost ($M)	Production Cost ($M)	Delivery Cost ($M)	Number of Customer Outages
With Contract	2.03	1.45	0.58	7
No Contract	2.06	1.45	0.61	7

Table 3: Effectiveness of Contract Agreements

average total cost per delivery and the number of customer outages by considering a finer-grain planning model. The minimax model and stochastic program deal with customer demand variability better than the nominal model—the company manages to satisfy more customers with less total cost. The contract agreements help the company save on delivery cost by removing products from competitor sites that is closer to its customers.

6 Conclusions

This work describes a mechanism for combining optimization and simulation in the production and distribution of atmospheric gases. Three optimization models were presented, each capable of being connected to a simulation engine that was able to assess practical performance of the models at time time-aggregation levels.

Using the tool has demonstrated convincingly that production-distribution decisions made using a multi-period model of up to four periods are likely to be significantly better than those from a one period model. Further, optimization models that specifically account for uncertainty via outcomes or scenarios can be significantly better than those that rely on point-estimates for unknown parameters. Finally, it was demonstrated how to use the simulation-optimization tool to perform contract pricing. These conclusions were drawn as a result of runs both on randomly generated instances and on real data from a large industrial gas company.

Many of the conclusions of this work are indeed intuitive, but the concept of combining simulation and optimization for decision-making is broadly applicable. Many planning problems suffer from a disconnect between planning and operations, and "Enterprise-Wide Optimization" initiatives are underway that seek to close this gap (Varma et al., 2007). However, an important question that any Enterprise-Wide Optimization initiative must address is to measure the *real* impact one can expect in operations if planning is done differently. Simulation can be a very useful tool to help answer this question. Further, having a simulation model of the enterprise-wide system allows the production of an an entire distribution of performance statistics.

Continuing work includes incorporating the prototype simulation and optimization model more broadly in the company, and in identifying additional novel uses for the simulation tool besides in contract pricing. More complicated decision planning models, such as robust optimization models and multi-stage stochastic programs are also being considered for future extensions.

Acknowledgements

The authors would like to acknowledge the state of Pennsylvania, through the Pennsylvania Infrastructure Technology Alliance (PITA) for support of this work. The availability of the XPRESS-MP and Mosel software, through the Academic Partner Program of Dash Optimization, is gratefully acknowledged.

References

Y Colombani and S. Heipcke. Mosel: An extensible environment for modeling and programming solutions. In N. Jussien and F. Laburthe, editors, *Proceedings of the Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR'02)*, pages 277–290, 2002.

L. Evans. Computer-aided design: Advances in process flowsheeting systems. In *Recent Developments in Chemical Process and Plant Design*, pages 261–287. Wiley, 1987.

M. C. Fu. Are we there yet? The marriage between simulation & optimization. *OR/MS Today*, 34(3): 16–17, June 2007.

H. Greenberg. Representing uncertainty in decision support. *OR/MS Today*, 34(3):14–15, June 2007.

C. N. Harper. Evolutionary computation at American Air Liquide. *ACM SIGEVOlution*, 1:2–4, 2006.

J. T. Linderoth, A. Shapiro, and S. J. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142:219–245, 2006.

A. Shapiro and T. Homem-de-Mello. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11:70–86, 2000.

V. A. Varma, G. V. Reklaitis, G. E. Blau, and J. F. Pekny. Enterprise-wide modeling & optimization—An opportunity of emerging research challanges and opportunities. *Computers and Chemical Engineering*, 31:692–711, 2007.

H. P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, 1999.