Managing Short-Term Electricity Contracts Under Uncertainty: A Minimax Approach

Samer Takriti* Shabbir Ahmed†

March 1, 2002

Abstract

A common problem facing energy producers is marketing excess capacity in the short-term market. A producer holds an auction asking interested buyers to submit bids reflecting their capacity and price requirements. As a buyer's demand is not known in advance, the supplier constructs several demand forecasts and associates a set of possible probability measures with these forecasts. The supplier then maximizes the expected profit while considering the worst-case probability distribution. The formulation—a minimax mixed-integer program—is solved using a branch-and-cut technique. The proposed technique is quite general and can be used to solve a wide class of minimax two-stage stochastic programs. Numerical testing indicates that the developed method is successful in solving practical models in less than a minute.

*Mathematical Sciences Department, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598, USA, takriti@us.ibm.com

†School of Industrial & Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, Atlanta, Georgia 30332, USA, sahmed@isye.gatech.edu. The work of this author was supported, in part, by the National Science Foundation under grant DMI-0099726.

1 Introduction

The deregulation of the electric-power industry has transformed electricity into a commodity, the price of which follows supply and demand imbalances. Electricity prices, which were tightly controlled prior to deregulation, are now set by market participants who buy and sell electricity as necessary. Due to the non-storable nature of electricity and the real-time aspect of power delivery, electricity has become one of the most volatile traded commodities. An example of this severe volatility is the situation that occurred in the Midwest during the week of June 22, 1998, when the day-ahead electricity price departed from its normal level of \$50 per MWH, and reached an unprecedented level of \$6000 per MWH (FERC 1998).

In order to cope with price volatility, several exchanges, including the New York Mercantile Exchange and the Nordic Power Exchange, began to offer standardized power contracts, such as futures and options. Unfortunately, the electric-power industry has been slow to adopt these standardized instruments, and instead has embraced over-the-counter contracts, such as daily and hourly options. The popularity of over-the-counter contracts may be attributed to transmission constraints and losses, to the contractual flexibility needed to cover the electric load, and to the tight relationship between electricity and other fuels. We refer the reader to Kaminski, Gibner, and Krishnarao (1999) for more details.

Although over-the-counter contracts are negotiated on a case-by-case basis, it is possible to classify them based on duration into long and short term (Ancona 1997). Long-term contracts have a duration of a few weeks to several years, whereas short-term contracts span a period of a few hours to several days. In this paper, we present our methodology as it applies to the evaluation of short-term contracts. Electricity contracts may also be classified based on the type of supply into firm and non-firm. In general, a firm contract obligates the electricity producer to deliver power throughout the contracted term unless there is a catastrophic event obstructing the production or delivery of power. Non-firm contracts provide a higher level of flexibility for the power producer as they do not come with the obligation to deliver. We limit our interest to firm contracts.

Throughout the rest of the paper, we refer to the entity that is interested in selling its short-term capacity in the form of firm contracts as the producer. Entities that bid for the producer's excess capacity are referred to as the bidders or buyers. We first describe the structure of a firm contract. Typically, a firm contract specifies a maximum amount of power C, known as capacity, that can be delivered during a single time period. It reflects the generation capacity that the producer must reserve on its system in order to respond to the buyer's demand. For each time period, the buyer of the contract has the right but not the obligation to consume an amount of power in the range $[0, C]$. In order to issue the contract and reserve the needed capacity, the producer charges an up-front fee which is a function of the reserved capacity: the higher the reserved capacity, the more expensive the initial fee. This fee is known as the capacity charge and is denoted by $\overline{r}$. The second charge associated with a firm contract is a variable charge that reflects the total consumption of power over the contract duration. This charge is known as energy charge. We use the function $f(d_t)$ to represent this charge during time period t and for demand d_t. Price structures that consider both capacity and energy charges are popular as they encourage buyers to accurately estimate and truthfully reveal their peak consumption, while discouraging them from over consuming.

In this paper, we consider a power producer that is interested in marketing its short-term excess capacity. Excess capacity is often a result of the conservative planning of electric-power generators in anticipation of existing demand obligations, also known as the native load. As more information becomes available, a producer frequently finds itself in the position of having excess generating capacity at its disposal. The capacity is marketed in the form of firm power by soliciting bids—capacity, capacity charge, and energy charge—from interested buyers. Given the uncertain nature of the native load as well as that of demand of the received bids, the producer's problem consists of selecting bids to maximize its expected profit.

In order to model demand uncertainty, we construct several sample paths—scenarios—

that represent possible future demand patterns. Then, an optimal production schedule is calculated by minimizing the average cost of operating the system while remaining feasible for each of the scenarios. However to construct the scenario tree and assign the appropriate probabilities with its nodes, we need to have access to sufficient historical information. More often than not, decision makers are faced with a lack of historical data, leading them to rely on subjective probabilities that may differ greatly from one expert opinion to another.

In the case of marketing short-term power, estimating the probabilities is a particularly difficult task. The producer has little knowledge, if any, of the goal behind the purchased power. Occasionally, a contract is purchased as a backup to enhance system security, in which case there is a little probability that it will be used. In other cases, a contract is purchased in order to compensate for supply shortage, which means that the buyer will most likely consume most of the contracted capacity. Therefore, it is unlikely that one can associate accurate probabilities with the scenario tree.

To overcome the difficulty of constructing an accurate probability measure on the demand scenarios, we assume the presence of several experts who are capable of providing views regarding the probabilities. That is, given a scenario tree that represents the future demand for each contract as well as that of the native load, each expert provides his/her views regarding the probability of each scenario. For example, assume that our problem is to maximize the expected profit over three future scenarios. Furthermore, assume that we have access to two experts. The first feels that scenario 1 is more likely than scenario 2, which in turn is more likely than scenario 3. Hence, we can write $p_1 \geq p_2 \geq p_3$, where p_1, p_2, and p_3 are the probabilities associated with scenarios 1, 2, and 3, respectively. The second expert believes that scenario 2 is the most likely one; i.e., $p_2 \geq p_1$ and $p_2 \geq p_3$. Then, the set of feasible probabilities can be identified as $p_1 = p_2 = \pi$ and $p_3 = 1 - 2\pi$, where π is a parameter satisfying $1/3 \leq \pi \leq 1/2$. Alternatively, we write $p \in \mathbb{P}$, where $\mathbb{P} = \{(p_1, p_2, p_3) : p_1 = p_2 = \pi,\ p_3 = 1 - 2\pi,\ 1/3 \leq \pi \leq 1/2\}$. The question then is which set of probabilities should we choose in our stochastic program?

From a risk viewpoint, it is prudent to consider the worst set of probabilities; i.e., probabilities that minimize the expected profit or maximize the expected cost. That is, given the differing views of our experts, one needs to consider the most bleak outlook—probabilities assigned to the scenario tree—of the future. In the case of selecting bids, the optimization problem consists of choosing contracts so as to maximize the expected profit of delivering power under the most pessimistic probability distribution; i.e., the probability distribution that causes the expected profit to be at its minimum. These types of models are referred to as minimax stochastic programs.

In the case of power delivery and production, it seems that the minimax formulation is of special interest as it reflects the risk-aversion attitude of this industry. Social and financial consequences of blackouts are costly and never tolerated. Therefore, generating companies often plan for the worst-case scenario. As a matter of fact, stochastic models that maximize expected profit, such as the one suggested in Takriti, Krasenbrink, and Wu (2000), are often criticized as having strong appetite for risk. As a result, it seems that the minimax formulation provides a reasonable compromise that captures stochasticity while accounting for forecast errors that may cause severe financial damages.

Minimax stochastic programming has received considerable attention in the context of bounding and approximating stochastic programs (Birge and Wets 1987, Gassman and Ziemba 1986, Kall 1991). For the case when the underlying problem is linear or convex, the theoretical properties of the problem of maximization over a probability space have been studied by Žáčková (1966) and by Shapiro and Kleywegt (2000). In contrast, algorithmic techniques for solving the proposed minimax stochastic program are limited. Dupacova (Dupačová 1980, Dupačová 1987, Žáčková 1966) considers minimax stochastic programs with separable and simple recourse structures and develops equivalent deterministic non-linear programming formulations for these. Under suitable convexity assumptions, the equivalent problem can, in principle, be solved using convex optimization techniques. Several general purpose saddle-point type algorithms for this class of minimax problems have also been suggested. For example, Ermoliev et al. (1985) describes a stochastic search method when the problem is convex with respect to the probability measures, and non-convex with respect to the problem variables. However, no numerical experience is reported with this technique. Breton and El Hachem (1995) develop a bundle methods based scheme for solving convex multi-stage minimax stochastic programs. The method requires solving a sequence of linear and quadratic programs. Limited computational results involving a test problem with 9 scenarios, and 18 variables and 12 constraints per scenario are reported. Note that both Ermoliev et al. (1985) and Breton and El Hachem (1995) require convexity of the set of probability measures and are inapplicable when, for example, the space of the probability measures is the union of several discrete distributions.

In this paper, we develop a minimax stochastic-programming model for the problem of selecting short-term power contracts. As the demand profile of a buyer is unknown, we assume the knowledge of several views, each of which represents a potential probability distribution for various demand scenarios. The objective is to maximize expected profit subject to the worst-case probability distribution on future load scenarios. The combinatorial nature of selecting a subset from a set of contracts makes the problem a difficult minimax stochastic mixed-integer program. To the best of our knowledge, algorithms for solving this class of problems have not been developed earlier. We propose a decomposition-based branch-and-cut strategy for this problem. This method extends the standard mixed-integer programming branch-and-bound algorithm by using a cutting-plane scheme to approximate the minimax objective function in each iteration. The proposed method can easily be extended to a wide variety of minimax stochastic programs. In particular, in the absence of the integer variables, the method reduces to a variant of the popular L-shaped decomposition algorithm for stochastic linear programs. We use the proposed branch-and-cut method to solve a set of test problems arising in power contract management. Our computational results indicate that the developed method significantly outperforms standard branch-and-bound techniques when applied to problems with a finite set of probability distributions, with a speed-up ratio of 20–2000 times.

The rest of this paper is organized as follows. Section 2 describes the problem at hand and formulates it as a mixed-integer program. In Section 3, we present a decomposition approach for solving the model. Finally, Section 4 provides a detailed description regarding our computer implementation and presents numerical results.

2 Model Development

We assume that the power producer is faced with J contracts, each of which has a capacity requirement of C_j, $j = 1, \ldots, J$. The capacity indicates the maximum amount of power that a bidder is entitled, but not obligated, to consume in a single time period. By accepting a contract, the producer receives a one-time payment of $\overline{r}_j$ for reserving the capacity. Furthermore, when quantity d_{jt} of the commodity is delivered during time period t, the producer receives a payment of $f_j(d_{jt})$. In order to accommodate demand fluctuations, we assume that the consumption of contract j is stochastic. Exceeding the capacity results in severe financial penalties. Therefore, we assume that the true consumption is bounded above by capacity; i.e., $d_{jt} \leq C_j$.

To model future uncertainties, we associate a set of sample paths—scenarios—with each of the contracts. We assume without loss of generality that the number of scenarios is the same for all contracts and is equal to K. We denote the sampled demand of contract j at time t by d^k_{jt}, where $k = 1, \ldots, K$, is the scenario index. When contract j is accepted, our producer expects to receive a payment of r_j, which is the sum of the present revenue of $\overline{r}_j$ and the expected future revenue of $\sum_{k=1}^{K} p_k \sum_{t=1}^{T} f_j(d^k_{jt})$.

In order to generate power, the producer has I generating units which are assumed to be committed in advance in response to the native load forecasts. The producer may alter the generation levels, but not the commitments, as to keep its cost at a minimum. That is, when bids are accepted, the producer makes the necessary adjustments to the production level without affecting the on-off status of each generator. For a given scenario k, we denote the production of generator i at time period t by y^k_{it}, $i = 1, \ldots, I$, $t = 1, \ldots, T$. In order to balance supply and demand, we enforce the constraint

$$\sum_{i=1}^{I} y_{it}^{k} = d_{0t}^{k} + \sum_{j=1}^{J} d_{jt}^{k} x_j, t = 1, \ldots, T, k = 1, \ldots, K,$$

where d_{0t}^{k} represents the native load; i.e., the original load that is used to schedule the generating units. Then, for a given set of probabilities p_k, the standard stochastic programming model is

$$\begin{array}{rl} \max_{x_j, y_{it}^k} & \sum_{j=1}^{J} r_j x_j - \sum_{k=1}^{K} p_k \sum_{t=1}^{T} \sum_{i=1}^{I} g_{it}^{k}(y_{it}^{k}) \\ \text{s.t.} & \sum_{i=1}^{I} y_{it}^{k} = d_{0t}^{k} + \sum_{j=1}^{J} d_{jt}^{k} x_j, \ t = 1, \ldots, T, \ k = 1, \ldots, K, \\ & x_j \text{ binary}, \ y^k \in Y^k, \ k = 1, \ldots, K, \end{array} \tag{1}$$

where x_j is a binary variable indicating whether contract j is to be accepted $x_j = 1$ or rejected $x_j = 0$. The function $g_{it}^{k}(y_{it}^{k})$ is the cost of producing y_{it}^{k} units of power and is assumed to be convex (Muckstadt and Koenig 1977).

As it might be beneficial to supplement the generating capacity by purchasing power from the spot market, we assume that the generation set has a spot-market generator with a cost function that reflects spot-market prices. Note that our formulation (1) allows the cost g associated with the spot generator, i.e., the spot price, to vary from one period to another and between scenarios. Furthermore, we assume that the spot generator may have a negative production; i.e., the set of y variables associated with the spot generator is unrestricted in sign. By allowing the production of the spot generator to be negative, the producer may opt to sell the power directly to the spot market, assuming that the price is attractive. The spot market—spot generator—can also be used to sell any excess capacity that is not sold in the form of firm contracts. In case of transaction costs or if spot prices are expected to change as a result of transacting with the spot market, one can use two spot generators: one for selling and the other for buying.

In (1), the notation $y^k \in Y^k$ denotes the constraints imposed on the production vector $y^k = \{y_{it}^{k}, \ i = 1, \ldots, I, \ t = 1, \ldots, T\}$, under scenario k. For example, let us say that generator i is committed at time period t. Then, its production cannot exceed its maximum capacity and cannot go below its minimum capacity; i.e., we enforce the constraint $q_{it}^k \leq y_{it}^k \leq Q_{it}^k$, where $[q_{it}^k, Q_{it}^k]$ is the operating range of unit i. Note that we allow available capacity to vary with time and scenarios. This reflects the fact that available capacity is determined by solving another optimization problem—a stochastic unit commitment—that attempts to minimize the cost needed to cover the native load d_{0t}^k. For an introduction to the unit commitment and its stochastic extensions, we refer the reader to Baldick (1995), Jacobs et al. (1995), Carpentier, Cohen, and Culioli (1996), and Takriti, Birge, and Long (1996). The set Y^k may include additional constraints depending on operational requirements. Our approach is valid as long as Y^k is convex and does not couple the decisions across stages; i.e., the resulting model is a two-stage program. Since Y^k represents the constraints imposed by the solution of a stochastic unit commitment problem when d_{0t}^k is used as the native load, the problem in (1) has a feasible solution in which all bids are declined; i.e., $x_1 = \ldots = x_J = 0$.

The model in (1) is a two-stage program in which the first stage determines the set of contracts to be chosen by assigning x_j the value of 0 or 1, while the second stage determines an optimal production strategy y_{it}^k in response to the demand dictated by the contracts. Viewing the problem as a two-stage program allows for the use of decomposition techniques that are widely used in stochastic programming (Birge and Louveaux 1997). The formulation in (1) assumes that the probabilities p_k, $k = 1, \ldots, K$, are known with complete certainty. As mentioned earlier, an alternative is to allow these probabilities to vary in order to reflect the collective view of our experts. To do so, we seek the probability distribution that provides the most conservative solution.

Mathematically, our formulation amounts to replacing the objective of (1) by the following minimax function.

$$\begin{array}{ll} \max_{x_j, y_{it}^k} & \sum_{j=1}^J r_j x_j - \min_{p \in \mathbb{P}} \sum_{k=1}^K p_k \sum_{t=1}^T \sum_{i=1}^I g_{it}^k(y_{it}^k), \\ \text{s.t.} & \sum_{i=1}^I y_{it}^k = d_{0t}^k + \sum_{j=1}^J d_{jt}^k x_j, \ t = 1, \ldots, T, \ k = 1, \ldots, K, \\ & x_j \text{ binary}, \ y^k \in Y^k, \ k = 1, \ldots, K, \end{array} \tag{2}$$

where $p = (p_1, \ldots, p_K)$ and $\mathbb{P}$ is the set of feasible probability distributions. Note that the set $\mathbb{P}$ is the intersection of the domains defined by the different views. That is, if view

l defines a set of feasible measures $\mathbb{P}_l$, then $\mathbb{P} = \cap_l \mathbb{P}_l$. As $\mathbb{P}$ represents the intersection of several views, it is possible that $\mathbb{P}$ may be infeasible. In this case, the objective function of the maximization problem in p may be augmented by a penalty term measuring the infeasibility with respect to the linear constraints defining $\mathbb{P}$. For the purpose of this paper, we assume that $\mathbb{P}$ is nonempty.

3 Minimax Stochastic Programming

The proposed formulation (2) has the general form:

$$\min_x \left\{ c^T x + h(x) \mid x \in X \cap \{0,1\}^J \right\}, \tag{3}$$

where $h(x) = \max_p \{\sum_{k=1}^K p_k Q_k(x) | (p_1, \ldots, p_k) \in \mathbb{P}\}$ and

$$Q_k(x) = \min_y \{ \sum_{t=1}^T \sum_{i=1}^I g_{it}^k(y_{it}^k) \mid Dy = h_k + T_k x, y \in Y^k \}.$$

Due to the convexity of g and the presence of a spot-market generator, the function $Q_k(x)$, $k = 1, \ldots, K$, is convex and finite valued for all $x \in X$. Hence, the expected cost function $h(x)$ is also convex and finite valued over X. We denote the feasible solutions to (3) by $x^1, \ldots, x^M$; i.e., $X \cap \{0,1\}^J = \{x^1, \ldots, x^M\}$. From the convexity of $h(x)$, we can reformulate (3) into the following mixed-integer linear program:

$$\min_{x,\theta} \left\{ c^T x + \theta \mid x \in X \cap \{0,1\}^J,\ (x, \theta) \in \mathcal{S} \right\}, \tag{4}$$

where $\mathcal{S} = \{(x, \theta) | \theta \geq h(x^m) + \partial h(x^m)^T (x - x^m), m = 1, \ldots, M\}$ and $\partial h(x^m)$ is a subgradient of h evaluated at x^m. That is, $\mathcal{S}$ is represented using a set of linear constraints—cuts, which are binding at all integer solutions x^m. Note that for a solution $\tilde{x}$ to the linear relaxation of (4); i.e., $\tilde{x} \in X \cap [0,1]^J$, the cut $\theta \geq h(\tilde{x}) + \partial h(\tilde{x})^T (x - \tilde{x})$ is a valid cut for $\mathcal{S}$, but may not be binding; i.e., the inequality may be strict.

We propose solving (4) using a branch-and-cut approach (Nemhauser and Wolsey 1998). In this scheme, we begin with a linear relaxation for (4) of the form

$$\min_{x,\theta} \left\{ c^T x + \theta \mid x \in X \cap [0,1]^J,\ (x, \theta) \in \overline{\mathcal{S}} \right\}, \tag{5}$$

where $\overline{\mathcal{S}}$ is defined using a set of valid cuts for $\mathcal{S}$; i.e., $\mathcal{S} \subseteq \overline{\mathcal{S}}$. The solution of (5) provides a lower bound on the value of (4). Integrality is enforced by branching on non-integer solutions in a typical branch-and-bound fashion. In order to tighten the approximation $\overline{\mathcal{S}}$, cuts may be added at any point in the branch-and-bound process as long as they are embedded to all nodes in the tree. We choose to add binding cuts at integer nodes that violate $\mathcal{S}$ so that the approximation is exact.

As for the root node, we start without any cuts in the system; i.e., $\overline{\mathcal{S}} = \Re^{J+1}$, and use Benders' decomposition to solve the model: solve the first-stage problem, solve the second-stage problems to create a cut, add the cut to $\overline{\mathcal{S}}$, and resolve the first-stage problem. The process iterates until an optimal solution for the linear relaxation is reached or until the number of cuts reaches its maximum limit. The cuts added to the $\overline{\mathcal{S}}$ are of the form $\theta \geq h(\tilde{x}) + \partial h(\tilde{x})^T(x - \tilde{x})$ and can be carried to the branch-and-cut process as they are valid for the integer program (4).

A formal description of the algorithm follows.

A Branch-and-cut algorithm for Solving (4)

Step 0. Set $\overline{\mathcal{S}}$ to $\Re^{J+1}$ and $\theta \leftarrow -\infty$. Solve the linear relaxation (4) as follows

Step a. Choose a solution $\tilde{x} \in X \cap [0,1]^J$ and $p \in \mathbb{P}$.

Step b. For each scenario k, solve the second-stage problem and determine the optimal objective value $Q_k(\tilde{x})$.

Step c. Evaluate $h(\tilde{x}) = \max_p\{\sum_k Q_k(\tilde{x})p_k | p \in \mathbb{P}\}$. Construct a new cut $\theta \geq h(\tilde{x}) + \partial h(\tilde{x})^T(x - \tilde{x})$ and add it to $\overline{\mathcal{S}}$.

Step d. If $\theta \approx h(\tilde{x})$ or if the maximum number of cuts is reached, go to Step f.

Step e. Solve the first-stage problem (5) and determine a new $\tilde{x}$ and θ. Go to Step b.

Step f. This is the end of the root node iteration. Note that $\overline{\mathcal{S}}$ contains valid cuts that are added during the solution process. In order to begin the branch-and-bound procedure, define $\mathcal{L}$ to be the set of unfathomed problems. Set $\mathcal{L}$ so that it has a single element, $\mathcal{L}^0$, which is the linear problem defined by (5). Set $x^1 \leftarrow \tilde{x}$, n to 1, m^* to -1, and $\overline{z}$ to ∞. As will become clear, n represents the number of nodes in the tree, m^* points to the optimal node, and $\overline{z}$ serves as an upper bound on the optimal objective value of (4). Proceed to Step 1.

Step 1. If $\mathcal{L}$ is empty, terminate and declare x^{m^*} to be an optimal solution for (4).

Step 2. Select a problem m from $\mathcal{L}$. We denote the linear program corresponding to this problem by $\mathcal{L}^m$. If $\mathcal{L}^m$ is infeasible, set $\mathcal{L} \leftarrow \mathcal{L} - \mathcal{L}^m$, and go to Step 1.

Step 3. Let (x^m, θ^m) be an optimal solution and z^m be the optimum value for problem $\mathcal{L}^m$. If $z^m \geq \overline{z}$, set $\mathcal{L} \leftarrow \mathcal{L} - \mathcal{L}^m$ and go to Step 1.

Step 4. If $x^m \notin \{0,1\}^J$, create two new problems $\mathcal{L}^n$ and $\mathcal{L}^{n+1}$, by fixing a non-integer element of x^m to 0 and 1, respectively. Set $\mathcal{L} \leftarrow \mathcal{L} - \mathcal{L}^m$, increment n by 2, and go to Step 1.

Step 5. Solve the second-stage problems and calculate a new probability measure as in Step b and Step c above. If $c^T x^m + h(x^m) < \overline{z}$, set $\overline{z} \leftarrow c^T x^m + h(x^m)$ and $m^* \leftarrow m$.

Step 6. If $\theta^m = h(x^m)$, set $\mathcal{L} \leftarrow \mathcal{L} - \mathcal{L}^m$ and go to Step 1. Otherwise, update the set $\overline{\mathcal{S}}$ by adding a binding cut at (x^m, θ^m); i.e., $\overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \cap \{(x,\theta) | \theta \geq h(x^m) + \partial h(x^m)^T (x - x^m)\}$, to all problems $\mathcal{L}^m, \ldots, \mathcal{L}^{n-1}$. Set $\mathcal{L} \leftarrow \mathcal{L} - \mathcal{L}^m$. Go to Step 1.

The algorithm requires the computation of a subgradient of h at $\tilde{x}$ in Step c and at x^m in Step 5. The following result establishes this procedure.

Proposition 1 *Consider a point $x^m \in X$. Let $\lambda_{m,k}$, $k = 1, \ldots, K$, be the optimal Lagrange multiplier associated with the constraint $Dy = h_k + T_k x^m$ in the evaluation of $Q_k(x^m)$ and let $(p_1^m, \ldots, p_K^m) \in \arg\max_{\mathbb{P}} \{\sum_{k=1}^K p_k Q_k(x^m)\}$. Then a subgradient of h at*

x^m *is given by:*

$$\partial h(x^m) = \sum_{k=1}^{K} p_k^m \lambda_{m,k}^T T_k.$$

Proof. From Lagrangian duality, for any $x \in X$

$$Q_k(x) \geq Q_k(x^m) + \lambda_{m,k}^T T_k(x - x^m) \text{ for all } k.$$

Multiplying the inequalities by $(p_1^m, \ldots, p_K^m)$ and summing,

$$\sum_{k=1}^{K} p_k^m Q_k(x) \geq \sum_{k=1}^{K} p_k^m Q_k(x^m) + \sum_{k=1}^{K} p_k^m \lambda_{m,k}^T T_k(x - x^m).$$

Noting that $h(x) = \max_{(p_1,\ldots,p_K)\in\mathbb{P}} \sum_{k=1}^{K} p_k Q_k(x) \geq \sum_{k=1}^{K} p_k^m Q_k(x)$, and $h(x^m) = \sum_{k=1}^{K} p_k^m Q_k(x^m)$ completes the proof. ■

Thus, Step 5 involves solving $Q_k(x^m)$ for each scenario $k = 1, \ldots, K$, collecting the corresponding optimal Lagrange multipliers, and then solving for the optimal probabilities $(p_1^m, \ldots, p_K^m)$, to construct a subgradient $\partial h(x^m)$. Note that the solution of $Q_k(x^m)$ for each scenario k can be carried out in a computationally convenient decomposed fashion. From the fact that the set $\mathcal{S}$ is defined by a finite number of inequalities, and since the set of feasible integer solutions is finite, we immediately have the following result.

Proposition 2 *Provided that $h(x)$ and a subgradient $\partial h(x)$ can be evaluated in finite time, then the proposed branch-and-cut algorithm terminates finitely with an optimal solution to (4).*

Proposition 2 indicates that our algorithm is valid as long as the maximization problem in p can be evaluated. Note that the proposed method is applicable to any two-stage stochastic program with a finite number of scenarios and continuous recourse. In the case of continuous first-stage variables and a fixed probability measure, our method is equivalent to the L-shaped algorithm that is widely used in solving stochastic linear programs.

4 Numerical Results

In this section, we present our experience with the proposed solution technique when applied to the system of a utility that is based in the Midwestern US. The system has 33 generators which are committed in advance in response to the native load using a unit commitment solver. As mentioned before, we assume that the commitment of a unit cannot be changed and that the only possible action is that of changing the production level of a generator. The generation system contains a committed spot generator with a linear cost function that changes across scenarios and periods in order to reflect possible spot market prices. The goal of our numerical experiment is to verify the validity of our solution method and to investigate its scalability with respect to the number of scenarios, views, and number of contracts. To do so, we implemented the algorithm in C++ on a Sun Sparc Workstation 450 MHz running Solaris 5.7. We used CPLEX 7.0 and its mixed-integer programming solver as the backbone for solving the first-stage problem (5).

In order to create a scenario tree for each contract and to associate a probability measure with its nodes, we rely on high-resolution weather forecasts for the delivery point of the contract. As the subject of weather forecasting is outside the scope of this paper, we refer the reader to the web site `http://www.research.ibm.com/weather` for a detailed discussion of the weather modeling work at IBM Research. Given a particular weather forecast, we use several load profiles to construct the scenario tree for a contract as described in Hoy, Takriti, and Wu (1998). Then, experts—traders—are presented with the scenario tree and asked to provide their views regarding the load, which are in turn translated into a set of linear constraints representing the set of possible probability measures.

The optimizer starts at the root node with a model that has bound constraints $0 \leq x_j \leq 1$, $j = 1, \ldots, J$, on the first-stage variables while it allows the approximate value of the second-stage objective θ to take any value. Then, by solving the second-stage problem and the corresponding maximization model in p, we construct a cut that is added to the root node, hence tightening the quality of the solution. The process is repeated until we find an optimal solution for the linear relaxation of the model. At this point the branch-and-bound starts.

In our implementation of the algorithm of Section 3, CPLEX maintains the list of unfathomed nodes $\mathcal{L}$, selects a new node as suggested in Step 2, solves the linear program of Step 3, and determines the branching variable of Step 4. At every node, we use CPLEX's call-back facility to check if the solution x^m at hand is integer. If it is, then the algorithm moves to Step 5, where a cut of the form $\theta \geq h(x^m) + \partial h(x^m)^T(x - x^m)$ is added to all active nodes $\mathcal{L}$ within the branch-and-bound tree. If x is non-integer, CPLEX proceeds with its branch-and-bound without alternating the set of cuts in the model. Our experience indicates that adding cuts at the root and at integer nodes only, outperforms other methods, such as adding cuts at every node in the tree, as it keeps the total number of cuts under control.

In order to create the cuts efficiently, we developed a Lagrangian-based C++ code for solving the second-stage problems $Q_k(x^m)$, $k = 1, \ldots, K$, as described in economic power dispatch literature. Briefly, the economic power dispatch is the problem of determining an optimal generation level so that the electric load is covered at each time period. As demand constraints are the only constraints linking the different generators, the problem can be decomposed by relaxing the demand requirement. The Lagrange multiplier $\lambda_{m,k}$ associated with each period can be interpreted as the value of a unit of power in that particular period. We refer the reader to Wood and Wollenberg (1996) for more detail.

Given the optimal second-stage values $Q_k(x^m)$, $k = 1, \ldots, K$, the maximization problem in p is $\max_p\{\sum_k Q_k(x^m)p_k \mid p \in \mathbb{P}\}$. If the set of feasible measures $\mathbb{P}$ is a convex polytope, then the maximization problem in p is a linear program, which is solved using the linear-programming library of CPLEX. In case $\mathbb{P}$ represents a finite set of measures, an optimal measure is found by evaluating the objective function at each $p \in \mathbb{P}$ and choosing the maximum value. A cut is then constructed and added to all active nodes in the branch-and-bound tree.

In order to compare the proposed method to CPLEX's mixed-integer programming solver, we consider the case when g^k_{it} is piecewise linear and convex, and $\mathbb{P}$ is finite; i.e., $\mathbb{P} = \{p^1, \ldots, p^L\}$ and $p^l = (p^l_1, \ldots, p^l_K)$ for all $l = 1, \ldots, L$. Then, the deterministic equivalent of (3) is

$$\begin{aligned} \min_{x,y,\phi}\{c^T x + \phi \mid x \in X \cap \{0,1\}^J,\ Dy^k = h_k + T_k x,\ y^k \in Y^k,\ k = 1, \ldots, K, \\ \phi \geq \textstyle\sum_{k=1}^K p^l_k \sum_{t=1}^T \sum_{i=1}^I g^k_{it}(y^k_{it}),\ l = 1, \ldots, L\}, \end{aligned} \tag{6}$$

which is a large-scale mixed-integer program that we pass to CPLEX. In the general case when $\mathbb{P}$ is a polytope, it is sufficient to consider the finite set of extreme points of $\mathbb{P}$. Then, the deterministic equivalent can be constructed as in (6), where L corresponds to the number of vertices.

Table 1 compares the decomposition approach with CPLEX's branch-and-bound solver when applied to 20 problems, each with a horizon of 24 periods. In Table 1, a problem is identified using three numbers representing the number of scenarios, views, and contracts, respectively. For example, Problem $10\times20\times40$ refers to the problem of selecting contracts from a pool of 40 contracts, where the future load of each contract is represented by 10 scenarios. The probability space has 20 possible probability measures, where each measure is represented by a vector of dimension 10.

For CPLEX, we report the CPU time needed to solve the linear relaxation—root node, the number of nodes in the branch-and-bound tree, and the total execution time for solving the integer program (6). These results are listed in the columns labeled "R-CPU," "Nodes," and "CPU," respectively. All execution times are measured in seconds and represent the total user and system times as reported by UNIX's `times` function. For the decomposition approach, the column "Root" reports the number of Benders' cuts needed to solve the linear-programming relaxation to within 10^{-9} of optimality. However, for the problems in Table 1, we limit the maximum number of cuts at the root node to 200. The number of additional cuts imposed during the branch-and-bound process is reported in "Int." Note that this is also the number of integer solutions—incumbents—encountered during the branch-and-bound process. The column "Nodes" under "Decomposition" reports the total number of nodes searched by the first-stage branch-and-bound while "CPU" indicates the total execution time in seconds. That is, it is the time needed to solve the first-stage mixed-integer program, to solve the second-stage problems, and to create and add the appropriate cuts.

The "Ratio" of the execution time of CPLEX to the CPU time of the algorithm of Section 3 provides the speed up that is achieved as a result of decomposing the model. This ratio varies between a minimum of $208.20/8.46 = 24.61$ for Problem $10 \times 20 \times 80$ and a maximum of 2132.23 for Problem $50 \times 50 \times 80$, with an average speed up of 447.53. Note that we do not report the execution time for solving the root node using decomposition as this time is relatively small. For example, for Problem $20 \times 20 \times 100$, it takes 1.44 seconds to solve the root node, while it takes 1.64 seconds to solve the root node of Problem $50 \times 50 \times 40$.

An interesting issue is that of deciding whether to solve the linear-programming relaxation to optimality or to stop after a certain number of cuts. For example, Table 1 indicates that the number of cuts needed to solve the root node of Problem $10 \times 20 \times 80$ is 162 cuts. It might be beneficial to terminate the process of adding cuts at the root node after a 100 cuts. In this case, we may lose accuracy in approximating the region around the continuous optimal solution, but gain by reducing the size of the first-stage problem. Table 2 studies the impact of capping the number of cuts at the root node by looking at Problem $10 \times 20 \times 100$, Problem $20 \times 20 \times 100$, and Problem $50 \times 50 \times 100$.

In Table 2, the column "Max." indicates the maximum number of cuts permitted at the root node. The algorithm for solving the root node terminates when the maximum number of cuts is reached or when the current solution is within 10^{-9} of optimality. The actual number of cuts at the root is reported in the column labeled "Root". When solving the linear relaxation of the root node terminates as a result of reaching the maximum number of cuts, we calculate the "Error" which is the ratio of the upper bound to the lower bound minus one. When the error exceeds 100%, we do not report its value. As in Table 1, columns labeled "Int.," "Nodes," and "CPU" provide the number of integer solutions found during the branch-and-bound process, the number of nodes searched, and the execution time in seconds, respectively. There does not appear to be a direct correlation between the execution time and the number of cuts at the root node. However, it is clear that as the number of cuts introduced at the root level increases, the total number of nodes in the branch-and-bound tree decreases.

Finally, Table 3 presents results related to several large test problems with the maximum number of cuts allowed at the root set to 100. The results indicate that the decomposition approach scales well with the number of contracts, views, and scenarios.

5 Conclusions

We proposed formulating the problem of selecting bids in the short-term electricity market as a minimax stochastic program. The load associated with a bid was modeled using a scenario tree, while the probability was allowed to vary based on an expert's view. In order to limit risk, we chose a probability measure that minimized the maximum expected profit. The resulting problem is a large-scale mixed-integer program which was solved using a decomposition approach. Our numerical results indicate that the suggested approach can solve large instances of the problem in 1–2 minutes.

References

J. J. Ancona. A bid solicitation and selection method for developing a competitive spot priced electric market. *IEEE Transactions on Power Systems*, 12(2):743–748, 1997.

J. R. Birge and R. J.-B. Wets. Computing bounds for stochastic programming problems by means of a generalized moment problem. *Mathematics of Operations Research*, 12:149–162, 1987.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

R. Baldick The generalized unit commitment problem. *IEEE Transactions on Power Systems*, 10:465–475, 1995.

M. Breton and S. El Hachem Algorithms for the solution of stochastic dynamic minimax problems. *Computational Optimization and Applications*, 4:317–345, 1995.

A. Carpentier, G. Cohen, and J.-C. Culioli. Stochastic optimization of unit commitment: A new decomposition framework. *IEEE Transactions on Power Systems*, 11(2):1067–1073, May 1996.

J. Dupačová. The minimax approach to stochastic programming and an illustrative application. *Stochastics*, 20:73–88, 1987.

J. Dupačová. Minimax stochastic programs with nonseparable penalties. *Optimization techniques (Proc. Ninth IFIP Conf., Warsaw, 1979)*, *Lecture Notes in Control and Information Science*, 22(1):157–163, Springer, Berlin, 1980.

Y. Ermoliev, A. Gaivoronsky, and C. Nedeva. Stochastic optimization problems with partially known distribution functions. *SIAM Journal on Control and Optimization*, 23:697–716, 1985.

Federal Energy Regulatory Commission. *Staff Report to the Federal Energy Regulatory Commission on the Causes of Wholesale Electricity Pricing Abnormalities in the Midwest During June 1998.* September 1998.

H. Gassman and W. T. Ziemba. A tight upper bound for the expectation of convex function of a multi-variate random variable. *Mathematical Programming Study*, 27:39–53, 1986.

E. Hoy, S. Takriti, and L. S.-Y. Wu. Divide and conquer: A Threshold model. *Energy and Power Risk Management*, 2(10):14–17, March 1998.

J. Jacobs, G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus, and J. Stedinger. SOCRATES: A system for scheduling hydroelectric generation under uncertainty. *Annals of Operations Research*, 59:99–133, 1995.

P. Kall. An upper bound for SLP using first and total second moments. *Annals of Operations Research*, 30:670–682, 1991.

V. Kaminski, S. Gibner, and K. Pinnamaneni. Energy exotic options. In *Managing Energy Price Risk*, Risk Books, 1999.

J. A. Muckstadt and S. A. Koenig. An application of Lagrangian relaxation to scheduling in power-generation systems. *Operations Research*, 25(3):387–403, 1977.

G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, 1998.

A. Shapiro and A. Kleywegt. Robust analysis of stochastic programs. Working paper, School of Industrial & Systems Engineering, Georgia Institute of Technology, 2000.

S. Takriti, J. R. Birge, and E. Long. A stochastic model of the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.

S. Takriti, B. Krasenbrink, and L. S.-Y. Wu. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research*, 48(2):268–280, March-April 2000.

A. J. Wood and B. F. Wollenberg. *Power Generation, Operation, and Control.* John Wiley, New York, 1996.

J. Žáčková. On minimax solution of stochastic linear programming problems. *Cas. Pest. Mat.*, 91:423–430, 1966.

Problem	CPLEX			Decomposition				Ratio
	R-CPU	Nodes	CPU	Root	Int.	Nodes	CPU	
$10 \times 20 \times 20$	13.85	113	59.58	94	8	306	0.65	92
$10 \times 20 \times 40$	23.00	169	137.98	147	33	761	2.43	57
$10 \times 20 \times 60$	24.86	357	181.24	171	51	1626	5.97	30
$10 \times 20 \times 80$	31.11	340	208.20	162	65	2594	8.46	25
$10 \times 20 \times 100$	31.03	282	211.38	144	48	1422	4.87	43
$20 \times 20 \times 20$	91.25	251	448.93	95	22	1011	1.68	267
$20 \times 20 \times 40$	128.54	452	645.13	136	44	2889	6.17	105
$20 \times 20 \times 60$	145.72	583	1129.61	136	54	2912	7.18	157
$20 \times 20 \times 80$	151.28	389	861.98	134	36	1628	5.24	165
$20 \times 20 \times 100$	145.92	583	1128.92	200	97	5490	19.82	60
$40 \times 30 \times 20$	560.27	518	3919.79	162	77	2367	5.62	697
$40 \times 30 \times 40$	999.26	113	2637.74	200	18	897	4.19	630
$40 \times 30 \times 60$	1076.77	182	3585.73	127	44	2940	7.17	500
$40 \times 30 \times 80$	1052.01	383	4878.95	144	79	4022	11.42	427
$40 \times 30 \times 100$	1129.17	954	7204.66	200	228	10268	41.79	172
$50 \times 50 \times 20$	1015.99	311	5439.08	144	47	2499	5.13	1060
$50 \times 50 \times 40$	1219.29	357	7213.40	200	134	8487	26.03	277
$50 \times 50 \times 60$	1553.08	214	6235.42	200	29	2356	8.20	760
$50 \times 50 \times 80$	1446.61	52	6034.21	163	6	229	2.83	2132
$50 \times 50 \times 100$	1668.75	83	7100.50	138	21	1140	4.74	1498

Table 1: Performance comparison between the proposed decomposition algorithm and CPLEX's mixed-integer programming solver. The maximum number of cuts at the root node is set to 200.

Problem	Root			BB		
	Max.	Act.	Error	Int.	Nodes	CPU
$10 \times 20 \times 100$	1	1	∞	89	2659	3.71
	10	10	∞	98	3098	5.38
	50	50	0.51%	48	1736	2.73
	100	100	0.01%	47	1241	3.53
	≥ 200	144	0.00%	48	1422	4.87
$20 \times 20 \times 100$	1	1	∞	117	7044	11.18
	10	10	14.36	137	7927	14.83
	50	50	0.90%	100	5855	11.94
	100	100	0.06%	91	5544	12.30
	200	200	0.00%	97	5490	19.82
	≥ 300	210	0.00%	95	5220	19.88
$50 \times 50 \times 100$	1	1	∞	90	5090	7.28
	10	10	5.52%	46	4178	4.76
	50	50	0.27%	32	1530	3.08
	100	100	0.00%	21	1325	4.14
	≥ 200	138	0.00%	21	1140	4.74

Table 2: The impact of the maximum number of cuts allowed at the root node.

Problem	Int.	Nodes	CPU	Problem	Int.	Nodes	CPU
$50 \times 50 \times 100$	21	1325	4.14	$50 \times 50 \times 200$	69	3528	22.65
$100 \times 50 \times 100$	63	4930	11.63	$100 \times 50 \times 200$	319	17849	136.31
$200 \times 50 \times 100$	25	2418	10.28	$200 \times 50 \times 200$	74	3663	43.64
$50 \times 100 \times 100$	89	9631	31.09	$50 \times 100 \times 200$	233	13949	85.29
$100 \times 100 \times 100$	20	1258	9.78	$100 \times 100 \times 200$	350	22664	156.58
$200 \times 100 \times 100$	97	7076	37.19	$200 \times 100 \times 200$	105	4149	47.94

Table 3: The scalability of the proposed algorithm. The maximum number of cuts at the root node is set to 100.