A novel branch and bound algorithm for optimal development of gas fields under uncertainty in reserves

Vikas Goel[*], Ignacio E. Grossmann[†]
Department of Chemical Engineering, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213

Amr S. El-Bakry and Eric L. Mulkay
ExxonMobil Upstream Research Company,
Houston, TX

Abstract

We consider the problem of optimal investment and operational planning for development of gas fields under uncertainty in gas reserves. Assuming uncertainties in the size and initial deliverabilities of the gas fields, the problem has been formulated as a multistage stochastic program by Goel and Grossmann (2004b). In this paper, we present a set of theoretical properties satisfied by any feasible solution of this model. We also present a Lagrangean duality based branch and bound algorithm that is guaranteed to give the optimal solution of this model. It is shown that the properties presented here achieve significant reduction in the size of the model. In addition, the proposed algorithm generates significantly superior solutions than the deterministic approach and the heuristic proposed by Goel and Grossmann (2004b). The optimality gaps are also much tighter.

[*]E-mail: vgoel@andrew.cmu.edu
[†]To whom all correspondence should be addressed. Tel.: (412) 268-3642. Fax: (412) 268-7139. E-mail: grossmann@cmu.edu.

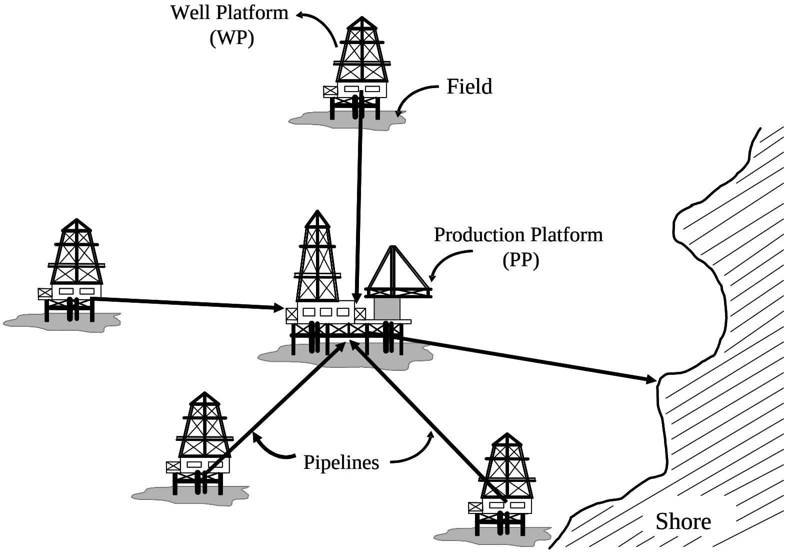

Figure 1: A typical gas production asset

1 Introduction

A gas production asset consists of a number of gas fields with infrastructure including well platforms, production platforms and connecting pipelines, as shown in Figure 1. A dedicated well platform (WP) is installed at each field being exploited for gas. Each WP is connected to another WP, or to a production platform (PP), through pipelines. Gas produced at all WPs is sent to the PPs and from there to the shore through this network of pipelines. Investment decisions to be taken during a gas production project include selection of WPs, PPs and pipelines to be installed together with their capacities, while operation decisions include the production profiles for all fields.

Each project in this industry typically lasts 10-30 years or more and can be worth in excess of $1 Billion. Installation of a WP or PP itself can cost millions of dollars and such a decision affects the profitability of the entire project. Given the large potential profits and high investments in each project, there is significant interest in developing optimization models for planning in the oil and gas exploration and production industry.

A major challenge in this area is to address the uncertainty in various parameters, quality of reserves being one of the most important among them. The existence of gas reserves at a site is indicated by seismic surveys and preliminary exploration tests. However, the actual sizes and the initial deliverabilities (represent efficacy of extraction of gas) of the gas reserves remain largely uncertain until after the capital investments have been made. Both these factors directly affect the profitability of the project and hence it is important to consider the impact of uncertainty in these parameters when formulating the decision policy.

However, most of the available literature that deals with planning of oil and gas field infrastructures uses a deterministic approach (Ierapetritou et al. (1999), Iyer et al. (1998), Grothey and McKinnon (2000), Van den Heever and Grossmann (2000), Van den Heever et al. (2001), Barnes et al. (2002), Kosmidis et al. (2002), Lin and Floudas (2003), Ortiz-Gomez et al. (2002)). For a recent review of deterministic approaches for these problems, please refer to Van den Heever et al. (2001). Goel and Grossmann (2004b) present a review of literature that deals with uncertainty in these problems. These authors have concluded that most of this work deals with simplified cases, where either the investment or operation decisions are assumed to be fixed, or the problem includes only one field (as examples, see Jornsten (1992), Haugen (1996), Jonsbraten (1998)).

Recently, Goel and Grossmann (2004b) addressed the general problem of optimizing investment and operation decisions simultaneously for a multi-field site under uncertainty in quality of reserves. They identified that while the standard modeling approach in stochastic programming (Birge and Louveaux (1997)) can address problems with *exogenous* uncertainty only(Jonsbraten (1998); scenario tree is independent of the optimization decisions), the gas field problem has *endogenous* uncertainty where the investment decisions determine the structure of the scenario tree. As a result, the standard approach in stochastic programming cannot be used to formulate an optimization model for this problem. For an example illustrating the dependence of the scenario tree on the investment decisions and its impact on the optimization model, please refer to Goel and Grossmann (2004b). These authors presented a novel hybrid mixed-integer/disjunctive optimization model where the interdependence between the scenario tree and the investment decisions is captured by conditional non-anticipativity constraints. They also developed a heuristic algorithm to solve this model. While the heuristic gave significantly better solutions than the deterministic method and was computationally more efficient than solving the the model in full space using off-the-shelf solvers, it is not guaranteed to find the optimal solution. Further, the upper bounding scheme in the heuristic produced weak bounds in some examples and the algorithm had to be stopped with fairly large optimality gaps.

The goal of this paper is to make the solution process more rigorous and efficient. We present properties that significantly reduce the size of the model presented by Goel and Grossmann (2004b) without affecting the feasible space, thus making solution of the full space model using an off-the-shelf solver more attractive. We also present a Lagrangean duality based branch and bound algorithm that, unlike the heuristic of Goel and Grossmann (2004b), is guaranteed to converge to the optimal solution. The algorithm also gives much tighter bounds than the heuristic.

This paper is organized as follows. Sections 2 and 3 present the problem statement and the stochastic programming model (Goel and Grossmann (2004b)) for the gas field problem. Theoretical properties of this model and a smaller but equivalent form of this model that is derived based on these properties is presented in section 4. The proposed Lagrangean duality branch and bound algorithm is presented in section 5. Finally, we illustrate the advantages of our approach with a set of example problems in section 6.

2 Problem Statement

Given is a gas production asset with a set of gas fields. To produce gas from a field, a dedicated WP will have to be installed at that field. Candidate WP to WP and WP to PP pipeline connections for the WP at each field are given. Candidate locations for the PPs are also given. Note that since each field is associated with a dedicated WP, we will use the terms field and WP interchangeably.

Investment and operation decisions have to be made over the project horizon of T years, which is discretized into T time periods of one year each. Investment decisions include selecting which WPs, PPs and pipeline connections should be installed in which time periods and the capacities of the WPs and PPs. Operation decisions include determining the production rates for each field in each time period. It is assumed that investments are instantaneous and take place at the beginning of a time period, while operation takes place throughout the time period. Also, once installed, the capacity of a WP or PP cannot be expanded further.

The quality of reserves of a field are characterized by the “size” and “deliverability” of the field. The size of a field corresponds to the total amount of gas that can be recovered from the field, while deliverability at any time represents the maximum rate of production that can be obtained from the field. As shown in Figure 2, the deliverability of a field is highest (initial deliverability) before production from the field has begun and decreases with increase in cumulative production from the field. In this paper, as in Goel and Grossmann (2004b), we assume that this decrease is linear.

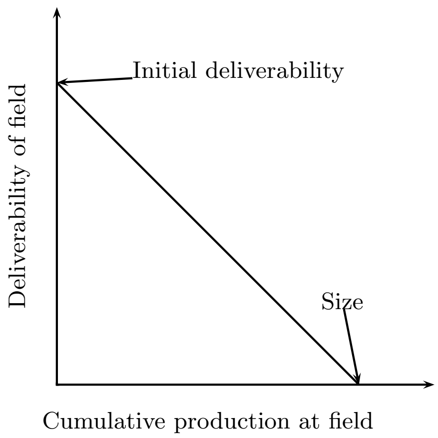

Figure 2: Linear Reservoir Model for a field

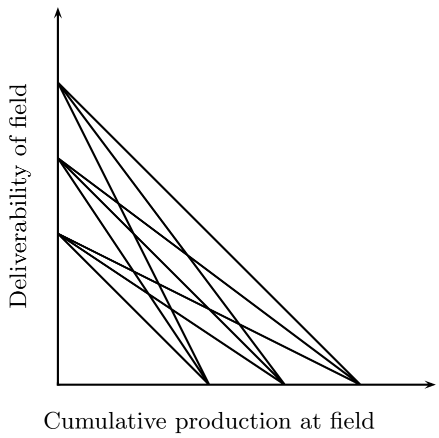

Figure 3: Nine scenarios arising from uncertainty in size and initial deliverability of a field

We assume discrete probability distributions for the sizes and initial deliverabilities of all fields. Let $\Theta_{wp,1}$ and $\Theta_{wp,2}$ represent the set of possible realizations in the distributions for the size and initial deliverability, respectively, of field wp. The overall set of possibilities is represented by a set of "scenarios", where each scenario is one possible combination of values for the sizes and initial deliverabilities of all fields, and has a given probability. We assume that the set of scenarios consists of every possible combination of realizations for the sizes and initial deliverabilities of various fields. Thus, the set of scenarios is given by $\times_{wp} (\Theta_{wp,1} \times \Theta_{wp,2})$, where the $\times$ operator computes the Cartesian product of sets. Figure 3 shows the nine possible reservoir models for a field with three realizations each for its size and its initial deliverability. For a problem where decisions have to be made only for this field, each of the nine reservoir models shown in Figure 3 would correspond to a scenario.

We use indices s, s' to refer to individual scenarios. $\mathcal{D}(s, s')$ represents the set of fields (or equivalently, WPs) that have different sizes and/or initial deliverabilities in scenarios s, s'. Mathematically,

$$\mathcal{D}(s, s') = \{wp | (\theta^s_{wp,1} \neq \theta^{s'}_{wp,1}) \vee (\theta^s_{wp,2} \neq \theta^{s'}_{wp,2})\}$$

For example, consider a problem where the uncertainty is restricted to uncertainty in the sizes

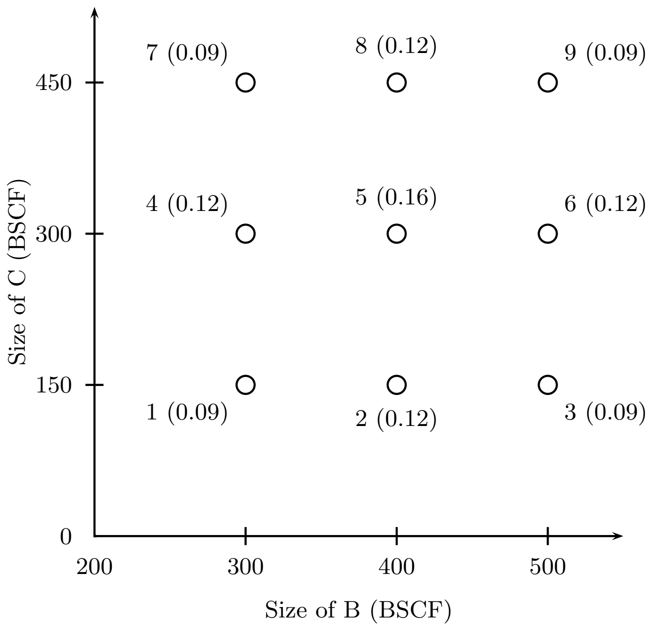

Figure 4: Scenarios for problem with uncertainty in sizes of fields B and C

of fields B and C and possible realizations for the size of field B include 300, 400 and 500 BSCF with[1] probabilities 0,3, 0.4 and 0.3, respectively, while possible realizations for the size of field C include 150, 300 and 450 BSCF with probabilities 0,3, 0.4 and 0.3, respectively. Each circle in Figure 4 represents one of the nine scenarios with the scenario probabilities given in parenthesis next to the scenario indices. The set $\mathcal{D}(s, s')$ for scenarios s, s' such that $s < s'$ is given in Table 1. As an example, $\mathcal{D}(1, 2) = \{B\}$ because scenarios 1 and 2 differ only in the size of field B (see Figure 4). Also, note that $\mathcal{D}(s, s') = \mathcal{D}(s', s)$.

We assume that all uncertainty in a field is resolved completely as soon as a WP is installed at the field; *i.e.*, the size and initial deliverability of the field are known deterministically once the WP is installed.[2] Goel and Grossmann (2004b) showed that as a result of this interdependence between the resolution of uncertainty and the investment decisions, the scenario tree is not fixed but depends on the investment decisions. However, since the non-anticipativity constraints in the stochastic program depend on the scenario tree, therefore the standard modeling approach in stochastic programming is based on the assumption that the scenario tree is known a priori.

[1]BSCF = Billion Standard Cubic Feet

[2]This is a simplification of reality because, in practice, installing a WP at a field yields a significant amount of information about the field, but may not eliminate the uncertainty completely.

$\mathcal{D}(s,s')$ for $s < s'$		s' 1	2	3	4	5	6	7	8	9
s	1		{B}	{B}	{C}	{B,C}	{B,C}	{C}	{B,C}	{B,C}
	2			{B}	{B,C}	{C}	{B,C}	{B,C}	{C}	{B,C}
	3				{B,C}	{B,C}	{C}	{B,C}	{B,C}	{C}
	4					{B}	{B}	{C}	{B,C}	{B,C}
	5						{B}	{B,C}	{C}	{B,C}
	6							{B,C}	{B,C}	{C}
	7								{B}	{B}
	8									{B}
	9									

Table 1: The set $\mathcal{D}(s,s')$ for scenarios s, s' given in Figure 4 such that $s < s'$

Since this is not the case with the gas field problem, the standard modeling approach in stochastic programming cannot be used to define the optimization model for this problem.

3 Model

(SPM) is the hybrid mixed-integer/disjunctive optimization model for the gas field problem presented by Goel and Grossmann (2004b). Note that while Goel and Grossmann (2004b) had presented the model in abbreviated form with variables and constraints represented by generic vectors, we have presented the unabbreviated model to make the following discussion clearer. In (SPM), index t refers to time periods, indices and s and s' refer to scenarios, indices wp and wp' refer to well-platforms while index pp refers to production platforms. A different set of decision variables is defined for each scenario. For example, $b^s_{wp,t}$ is a binary variable representing whether or not well-platform wp (at the corresponding field) is installed in time period t of scenario s. Please refer to the nomenclature at the end of the paper.

$$(SPM) \qquad \phi = \max \sum_s p^s NPV^s \tag{1}$$

$$\text{s.t.} \quad \frac{q^{deliv,s}_{wp,t}}{\theta^s_{wp,2}} + \frac{q^{cum,s}_{wp,t}}{\theta^s_{wp,1}} = 1 \qquad \forall (wp,t,s) \tag{2}$$

$$q^{cum,s}_{wp,t} = \sum_{\tau=1}^{t} q^{prod,s}_{wp,\tau}\delta_\tau \qquad \forall (wp,t,s) \tag{3}$$

$$q^{prod,s}_{wp,t} \leq q^{deliv,s}_{wp,t} \qquad \forall (wp,t,s) \tag{4}$$

The objective of (SPM) is to maximize the expected net present value (ENPV), which is the probability weighted average of the net present values (NPVs) over all scenarios (1). (2)-(28) represent investment, operational and economic constraints for scenario s. Constraint (2) represents the cumulative production-deliverability curve (Figure 2) for field wp in scenario s. $\theta^s_{wp,1}$ and $\theta^s_{wp,2}$ represent the size and initial deliverability, respectively, of field wp in scenario s. Equation (3) computes the cumulative production for field wp while (4) restricts the production rate for field wp below its current deliverability.

$$q^{out,s}_{wp,t} = q^{prod,s}_{wp,t} + \sum_{wp'} q^{out,s}_{wp',wp,t} \quad \forall(wp,t,s) \tag{5}$$

$$q^{out,s}_{wp,t} = \sum_{wp'} q^{out,s}_{wp,wp',t} + \sum_{pp} q^{out,s}_{wp,pp,t} \quad \forall(wp,t,s) \tag{6}$$

$$q^{out,s}_{pp,t} = \sum_{wp} q^{out,s}_{wp,pp,t} \quad \forall(pp,t,s) \tag{7}$$

$$q^{shr,s}_{t} = (1 - shrink) \sum_{pp} q^{out,s}_{pp,t} \quad \forall(t,s) \tag{8}$$

Equations (5)-(6) together enforce mass balance at well-platform wp while (7) and (8) compute the flow into pp and into the shore, respectively, in time period t. The factor $shrink$ accounts for the shrinkage in flow to the shore through the pipeline.

$$q^{out,s}_{wp,t} \leq cap^s_{wp,t} \quad \forall(wp,t,s) \tag{9}$$

$$q^{out,s}_{pp,t} \leq cap^s_{pp,t} \quad \forall(pp,t,s) \tag{10}$$

$$cap^s_{wp,t} = cap^s_{wp,t-1} + e^s_{wp,t} \quad \forall(wp,t,s) \tag{11}$$

$$cap^s_{pp,t} = cap^s_{pp,t-1} + e^s_{pp,t} \quad \forall(pp,t,s) \tag{12}$$

$$e^s_{wp,t} \leq M_{wp} b^s_{wp,t} \quad \forall(wp,t,s) \tag{13}$$

$$e^s_{pp,t} \leq M_{pp} b^s_{pp,t} \quad \forall(pp,t,s) \tag{14}$$

$$q^s_{wp,wp',t} \leq M_{wp,wp'} \sum_{\tau=1}^{t} b^s_{wp,wp',\tau} \quad \forall(wp,wp',t,s) \tag{15}$$

$$q^s_{wp,pp,t} \leq M_{wp,pp} \sum_{\tau=1}^{t} b^s_{wp,pp,\tau} \quad \forall(wp,pp,t,s) \tag{16}$$

Constraints (9)-(10) impose capacity restrictions on flows out of all platforms. Platform capacities in time period t are computed by equalities (11)-(12). (13)-(14) are upper-bounding constraints on expansion decisions in time period t. Similarly, (15)-(16) are upper-bounding constraints on the flow-rates through pipelines.

$$\sum_t b^s_{wp,t} \leq 1 \quad \forall (wp, s) \tag{17}$$

$$\sum_t b^s_{pp,t} \leq 1 \quad \forall (pp, s) \tag{18}$$

$$\sum_t b^s_{wp,wp',t} \leq 1 \quad \forall (wp, wp', s) \tag{19}$$

$$\sum_t b^s_{wp,pp,t} \leq 1 \quad \forall (wp, pp, s) \tag{20}$$

$$b^s_{wp,t} = \sum_{wp'} b^s_{wp,wp',t} + \sum_{pp} b^s_{wp,pp,t} \quad \forall (wp, t, s) \tag{21}$$

$$b^s_{wp,wp',t} \leq \sum_{\tau=1}^{t} b^s_{wp',\tau} \quad \forall (wp, wp', t, s) \tag{22}$$

$$b^s_{wp,pp,t} \leq \sum_{\tau=1}^{t} b^s_{pp,\tau} \quad \forall (wp, pp, t, s) \tag{23}$$

$$b^s_{wp,wp',t} + b^s_{wp',wp,t} \leq 1 \quad \forall (wp, wp', t, s) \tag{24}$$

Inequalities (17)-(20) state that a platform or a pipeline connection can be installed only once. Equation (21) states that exactly one outgoing connection should be installed when wp is installed. Constraints (22)-(23) state that a connection to a platform can be installed only if that platform has already been installed. Constraint (24) prevents two way connections between platforms.

$$CC^{tot,s}_t = \sum_{wp} \left(FCC_{wp} b^s_{wp,t} + VCC_{wp} e^s_{wp,t} + \sum_{wp'} FCC_{wp,wp'} b^s_{wp,wp',t} + \sum_{pp} FCC_{wp,pp} b^s_{wp,pp,t} \right) + \sum_{pp} \left(FCC_{pp} b^s_{pp,t} + VCC_{pp} e^s_{pp,t} \right) \quad \forall (t, s) \tag{25}$$

$$OC^{tot,s}_t = \sum_{wp} \left(FOC_{wp} b^s_{wp,t} + VOC_{wp} q^{prod,s}_{wp,t} \right) + \sum_{pp} \left(FOC_{pp} b^s_{pp,t} + VOC_{pp} q^{out,s}_{pp,t} \right) \quad \forall (t, s) \tag{26}$$

$$Rev^{tot,s}_t = \sum_t P_t q^{shr,s}_t \delta_t \quad \forall (t, s) \tag{27}$$

$$NPV^s = \sum_t \alpha_t (Rev^{tot,s}_t - CC^{tot,s}_t - OC^{tot,s}_t) \quad \forall s \tag{28}$$

The overall capital cost consists of fixed and variable costs for installation of platforms and fixed costs for the installation of pipelines (25). The total operating cost consists of fixed and variable costs for operation of platforms (26). Equation (27) computes the total revenue for time period t based on the flow to the shore while the overall NPV for scenario s is computed in (28). Constraints (2)-(28) are applied for every scenario.

$$
\begin{aligned}
b^{s}_{wp,1} &= b^{s'}_{wp,1} \quad \forall (wp, s, s'), s < s' && \text{(29a)}\\
e^{s}_{wp,1} &= e^{s'}_{wp,1} \quad \forall (wp, s, s'), s < s' && \text{(29b)}\\
b^{s}_{pp,1} &= b^{s'}_{pp,1} \quad \forall (pp, s, s'), s < s' && \text{(29c)}\\
e^{s}_{pp,1} &= e^{s'}_{pp,1} \quad \forall (pp, s, s'), s < s' && \text{(29d)}\\
b^{s}_{wp,wp',1} &= b^{s'}_{wp,wp',1} \quad \forall (wp, wp', s, s'), s < s' && \text{(29e)}\\
b^{s}_{wp,pp,1} &= b^{s'}_{wp,pp,1} \quad \forall (wp, pp, s, s'), s < s' && \text{(29f)}
\end{aligned}
$$

$$
Z^{s,s'}_{t} \Leftrightarrow \left[\bigwedge_{wp \in \mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} \left(\neg b^{s}_{wp,\tau} \right) \right] \quad \forall (t, s, s'), s < s' \tag{30}
$$

$$
\left[\begin{array}{rcl}
 & Z^{s,s'}_{t} & \\
b^{s}_{wp,t+1} = & b^{s'}_{wp,t+1} & \forall wp \\
e^{s}_{wp,t+1} = & e^{s'}_{wp,t+1} & \forall wp \\
b^{s}_{pp,t+1} = & b^{s'}_{pp,t+1} & \forall pp \\
e^{s}_{pp,t+1} = & e^{s'}_{pp,t+1} & \forall pp \\
b^{s}_{wp,wp',t+1} = & b^{s'}_{wp,wp',t+1} & \forall (wp, wp') \\
b^{s}_{wp,pp,t+1} = & b^{s'}_{wp,pp,t+1} & \forall (wp, pp) \\
q^{prod,s}_{wp,t} = & q^{prod,s'}_{wp,t} & \forall wp
\end{array} \right] \vee \left[\neg Z^{s,s'}_{t} \right] \quad \forall (t, s, s'), s < s' \tag{31}
$$

Decisions for different scenarios are linked to each other by *non-anticipativity* constraints (29)-(31). These constraints ensure that decisions at any time are based only on information available at that time, and not on foresight. Scenarios s, s' are said to be *indistinguishable* at some time if s, s' are identical in realizations for all parameters in which uncertainty has been resolved up till that time. The non-anticipativity rule states that if two scenarios are indistinguishable at some time, then decisions at that time should be the same in the two scenarios. As explained in section 2, we assume that investments are made instantaneously at the beginning of each time period, while operation is performed throughout the time period. Since it is the investment decisions that lead to the resolution of uncertainty, operation decisions in time period t and investment decisions in time period $t+1$ should be the same for scenarios s, s' if these scenarios are indistinguishable after investments in time period t.

By definition, all scenarios are mutually indistinguishable before investments are made in the first time period. Thus, non-anticipativity requires that investment decisions at $t = 1$ be the same for all scenarios (constraints (29a)-(29f)). Boolean variable $Z_t^{s,s'}$ represents whether or not scenarios s, s' are indistinguishable after investments have been made in time period t. Expression (30) relates the indistinguishability of scenarios with the investment decisions for the WPs.

(30) states[3] that scenarios s, s' are indistinguishable after investment in time period t if and only if none of the WPs that distinguish scenario s from s' have been installed up till time period t. For scenarios s, s', disjunction (31) imposes non-anticipativity constraints on operation decisions in time period t, and investment decisions in time period $t+1$ only if s, s' are indistinguishable after investment in time period t.

Note that in (SPM), $b^{(\cdot)}_{wp,t+1}, b^{(\cdot)}_{pp,t+1}, b^{(\cdot)}_{wp,wp',t+1}, b^{(\cdot)}_{wp,pp,t+1}, e^{(\cdot)}_{wp,t+1}, e^{(\cdot)}_{pp,t+1}, q^{prod,(\cdot)}_{wp,t}$ are the only decision variables. All other variables are "state variables" that can be eliminated from the model. Hence non-anticipativity is not imposed on those variables. Also, note that to avoid duplication of non-anticipativity constraints for a pair of scenarios, constraints (29)-(31) are applied for (s, s') only if $s < s'$.

Finally, the domains for the variables are specified as below.

$$
\begin{array}{llll}
b^s_{wp,t} \in \{0,1\} & \forall(wp,t,s), & b^s_{pp,t} \in \{0,1\} & \forall(pp,t,s), \\
b^s_{wp,wp',t} \in \{0,1\} & \forall(wp,wp',t,s), & b^s_{wp,pp,t} \in \{0,1\} & \forall(wp,pp,t,s), \\
e^s_{wp,t} \geq 0 & \forall(wp,t,s), & e^s_{pp,t} \geq 0 & \forall(pp,t,s), \\
q^{prod,s}_{wp,t} \geq 0 & \forall(wp,t,s), & Z_t^{s,s'} \in \{True, False\} & \forall(t,s,s'), s < s'
\end{array}
$$

[3] In theory, the logical operator "$\neg$" should only be used with Boolean variables. Since $b^s_{wp,t}$ are binary variables, constraint (30) involves a slight inconsistency in notation. A more rigorous formulation can be obtained at the expense of additional notation by defining (30) in terms of Boolean variables $B^s_{wp,t}$ and specifying an equivalence between variables $B^s_{wp,t}$ and $b^s_{wp,t}$.

4 Model Properties

Next we present a set of properties satisfied by all feasible solutions of (SPM). These properties are combined in Theorem 1 to arrive at an equivalent but significantly smaller model than (SPM). Note that we will follow the convention that a variable name for which an index has been dropped refers to the vector of that variable over all possible values of that index. For example, $b_{wp,t}$ represents the vector of $b^s_{wp,t}$ for all s while b_{wp} represents the vector of $b^s_{wp,t}$ for all (t, s). Similarly, b represents the vector of $b^s_{wp,t}, b^s_{pp,t}, b^s_{wp,wp',t}, b^s_{wp,pp,t}$ for all (wp, wp', pp, t, s). Finally, we will use the set of vectors (b, e, q^{prod}, Z) to refer to a feasible solution of (SPM).

Lemma 1 states that given a feasible solution of (SPM) and scenarios s, s', if the sub-solution for scenario s is such that WPs have not been installed at any of the fields that distinguish scenario s from s', then the sub-solution for scenario s' also satisfies this condition.

Proposition 1 implies that constraints (30)-(31) need to be applied for scenarios s, s' only if these scenarios differ in the realization of exactly one uncertain parameter. We refer to the set of such scenario pairs as $\mathcal{L}^1$, which is mathematically defined as

$$\mathcal{L}^1 = \left\{(s, s')|s < s', \mathcal{D}(s, s') = \{wp^*\}, (\theta^s_{wp^*,1} \neq \theta^{s'}_{wp^*,1}) \veebar (\theta^s_{wp^*,2} \neq \theta^{s'}_{wp^*,2})\right\}.$$

Proposition 2 states that for $(s, s') \in \mathcal{L}^1$ and for wp^* such that wp^* is the (only) field that has different properties in scenarios s, s', the investment decisions for field wp^* always have to be the same for scenarios s, s'. Thus, equalities $b^s_{wp^*,t+1} = b^{s'}_{wp^*,t+1}$, $e^s_{wp^*,t+1} = e^{s'}_{wp^*,t+1}$ hold irrespective of the value of $Z^{s,s'}_t$, unlike as stipulated by (31).

The detailed proofs to Lemma 1 and Proposition 1 can be found in Goel and Grossmann (2004b) and Goel and Grossmann (2004a), respectively. Here we only present brief argumentative proofs to provide physical insight into those relationships.

Lemma 1. *Any feasible solution of* (SPM) *satisfies*

$$\left[\bigwedge_{wp\in\mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} \left(\neg b^s_{wp,\tau}\right)\right] \Leftrightarrow \left[\bigwedge_{wp\in\mathcal{D}(s',s)} \bigwedge_{\tau=1}^{t} \left(\neg b^{s'}_{wp,\tau}\right)\right] \quad \forall (t, s, s'), s < s'$$

Proof. Suppose $(\bar{b}, \bar{e}, \bar{q}^{prod}, \bar{Z})$ is a feasible solution of (SPM). Consider scenarios $s, s', s < s'$. Suppose the solution is such that scenarios s, s' cease to be indistinguishable after investments in time period t^*. By definition of t^*, $\bar{Z}^{s,s'}_{t^*-1} = True, \bar{Z}^{s,s'}_{t^*} = False$ and therefore, using (30) ,

$\exists wp^* \in \mathcal{D}(s,s')$ such that

$$\bar{b}^s_{wp^*,t^*} = 1 \tag{32}$$

$$\bar{b}^s_{wp,t} = 0 \quad \forall (wp,t), wp \in \mathcal{D}(s,s'), 1 \le t \le t^* - 1. \tag{33}$$

Also, since the solution satisfies the non-anticipativity rule, therefore

$$\bar{b}^s_{wp,t} = \bar{b}^{s'}_{wp,t} \quad \forall (wp,t), 1 \le t \le t^*. \tag{34}$$

Thus, combining (32) with (34) and (33) with (34), we get

$$\left[\bigwedge_{wp \in \mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} \left(\neg b^s_{wp,\tau}\right) \right] = \left[\bigwedge_{wp \in \mathcal{D}(s',s)} \bigwedge_{\tau=1}^{t} \left(\neg b^{s'}_{wp,\tau}\right) \right] = True \quad \forall 1 \le t \le t^* - 1,$$

$$\left[\bigwedge_{wp \in \mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} \left(\neg b^s_{wp,\tau}\right) \right] = \left[\bigwedge_{wp \in \mathcal{D}(s',s)} \bigwedge_{\tau=1}^{t} \left(\neg b^{s'}_{wp,\tau}\right) \right] = False \quad \forall t \ge t^*$$

□

Note that as a result of the above lemma, we can use either of $b^s_{wp,\tau}$ and $b^{s'}_{wp,\tau}$ in the right hand side of (30).

Proposition 1. *Consider constraints (35)-(37) defined over the tuple* (t,s,s')*.*

$$b^s_{wp,t} = b^{s'}_{wp,t} \quad \forall wp \tag{35a}$$

$$e^s_{wp,t} = e^{s'}_{wp,t} \quad \forall wp \tag{35b}$$

$$b^s_{pp,t} = b^{s'}_{pp,t} \quad \forall pp \tag{35c}$$

$$e^s_{pp,t} = e^{s'}_{pp,t} \quad \forall pp \tag{35d}$$

$$b^s_{wp,wp',t} = b^{s'}_{wp,wp',t} \quad \forall (wp,wp') \tag{35e}$$

$$b^s_{wp,pp,t} = b^{s'}_{wp,pp,t} \quad \forall (wp,pp) \tag{35f}$$

$$Z^{s,s'}_t \Leftrightarrow \left[\bigwedge_{wp \in \mathcal{D}(s,s')} \bigwedge_{\tau=1}^{t} \left(\neg b^s_{wp,\tau}\right) \right] \tag{36}$$

$$\begin{bmatrix} & Z_t^{s,s'} & \\ b^s_{wp,t+1} = & b^{s'}_{wp,t+1} & \forall wp \\ e^s_{wp,t+1} = & e^{s'}_{wp,t+1} & \forall wp \\ b^s_{pp,t+1} = & b^{s'}_{pp,t+1} & \forall pp \\ e^s_{pp,t+1} = & e^{s'}_{pp,t+1} & \forall pp \\ b^s_{wp,wp',t+1} = & b^{s'}_{wp,wp',t+1} & \forall (wp,wp') \\ b^s_{wp,pp,t+1} = & b^{s'}_{wp,pp,t+1} & \forall (wp,pp) \\ q^{prod,s}_{wp,t} = & q^{prod,s'}_{wp,t} & \forall wp \end{bmatrix} \vee \left[\neg Z_t^{s,s'}\right] \tag{37}$$

Suppose the set of scenarios is given by $\times_{wp} (\Theta_{wp,1} \times \Theta_{wp,2})$*, where* $\Theta_{wp,1}$ *and* $\Theta_{wp,2}$ *represent the set of possible realizations for the size and initial deliverability, respectively, of field wp. If solution* $(\bar{b}, \bar{e}, \bar{q}^{prod}, \bar{Z})$ *satisfies*

(i) Constraint (35) for all (t, s, s') *such that* $s < s', t = 1$

(ii) Constraints (36)-(37) for all (t, s, s') *such that* $(s, s') \in \mathcal{L}^1$

then this solution satisfies (36)-(37) for all (s, s') *such that* $s < s'$*.*

Proof. Consider scenarios s_0, s_k such that $s_0 < s_k$ and s_0, s_k differ in realizations for k parameters. If $k = 1$ then $(s_0, s_k) \in \mathcal{L}^1$ and therefore (36)-(37) hold because of condition (ii) above. If $k > 1$, suppose that vector $\bar{b}_{wp}$ is such that scenarios s_0, s_k are mutually indistinguishable after investments in time period t, and therefore operation decisions in time period t and investment decisions in time period $t+1$ should be the same for these two scenarios. Since the set of scenarios is given by $\times_{wp} (\Theta_{wp,1} \times \Theta_{wp,2})$ (all possible combinations of realizations for the sizes and initial deliverabilities of all fields), the set of scenarios will include scenarios $s_1, s_2, \ldots, s_{k-1}$ such that s_i, s_{i+1} differ in the realization of exactly one of (and mutually exclusive) these k parameters, for all $0 \leq i \leq k-1$. Thus, $(s_0, s_1), (s_1, s_2), (s_2, s_3), \ldots, (s_{k-2}, s_{k-1}), (s_{k-1}, s_k) \in \mathcal{L}^1$. Since s_0, s_k are mutually indistinguishable in t, uncertainty in none of the k parameters should have been resolved up till period t. Since scenarios s_i, s_{i+1} differ in the realization of exactly one of these k parameters, for all $0 \leq i \leq k-1$, these scenarios should also be mutually indistinguishable. Since $(s_i, s_{i+1}) \in \mathcal{L}^1$ for all $0 \leq i \leq k-1$, therefore $\bar{Z}_t^{s_i, s_{i+1}}$ should be $True$. Thus, from (37) we can infer that operation decisions in time period t and investment decisions in time period $t+1$ will be the same for s_i, s_{i+1} for all $0 \leq i \leq k-1$. Thus, operation decisions in time period t and investment decisions in time period $t+1$ will be same for s_0, s_k. □

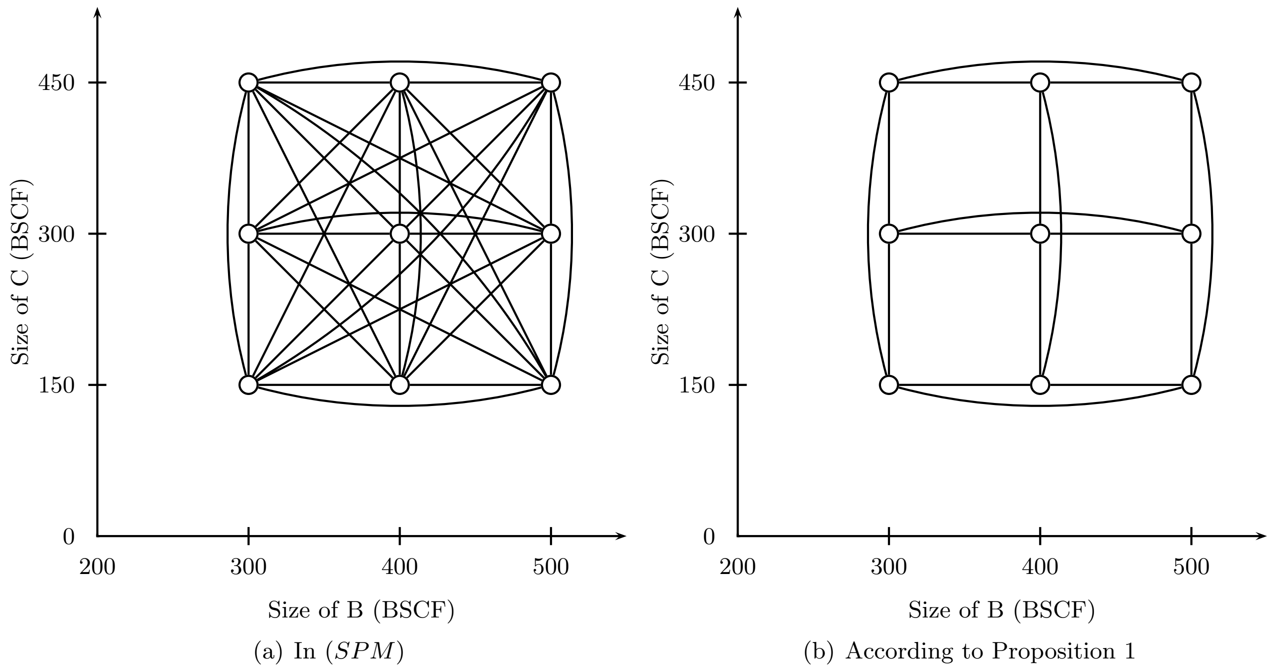

Figure 5: Pairs (s, s') for which (30)-(31) have to be applied for example introduced in section 2

Figure 5 is a pictorial representation of the statement of Proposition 1 for the two field example introduced in section 2. Linkages between scenarios s, s' mean that constraints (30)-(31) will be applied for (t, s, s') for all t.

Proposition 2. *Any feasible solution of* (SPM) *satisfies*

$$\left.\begin{aligned} b^s_{wp,t} &= b^{s'}_{wp,t} \\ e^s_{wp,t} &= e^{s'}_{wp,t} \end{aligned}\right\} \forall (wp, t, s, s'), (s, s') \in \mathcal{L}^1, \{wp\} = \mathcal{D}(s, s')$$

Proof. Suppose $(\bar{b}, \bar{e}, \bar{q}^{prod}, \bar{Z})$ is a feasible solution of (SPM). Choose (wp, s, s') such that $(s, s') \in \mathcal{L}^1, \{wp\} = \mathcal{D}(s, s')$. For $t = 1$, the proof follows directly from (29a) and (29b). Consider $t > 1$. Suppose $\bar{b}^s_{wp,t} \neq \bar{b}^{s'}_{wp,t}$. Since $\bar{b}^s_{wp,t}, \bar{b}^{s'}_{wp,t} \in \{0, 1\}$, exactly one of $\bar{b}^s_{wp,t}, \bar{b}^{s'}_{wp,t}$ is equal to 1.

If $\bar{b}^s_{wp,t} = 1$, then using (17), we can infer that $\bar{b}^s_{wp,\tau} = 0$ for all τ such that $1 \leq \tau \leq t - 1$. Then, using (30) and (31), we get $\bar{Z}^{s,s'}_{t-1} = True$ and therefore $\bar{b}^s_{wp,t} = \bar{b}^{s'}_{wp,t}$. Hence we have a contradiction to the hypothesis that $\bar{b}^s_{wp,t} \neq \bar{b}^{s'}_{wp,t}$.

On the other hand, if $\bar{b}^{s'}_{wp,t} = 1$, then from (17), we have $\bar{b}^{s'}_{wp,\tau} = 0$ for all τ such that $1 \leq \tau \leq t-1$. Then, using Lemma 1, we can infer that $\bar{b}^s_{wp,\tau} = 0$ for all τ such that $1 \leq \tau \leq t - 1$. Using (30) and (31) as above, we get $\bar{Z}_{t-1}^{s,s'} = True$ and therefore $\bar{b}_{wp,t}^{s} = \bar{b}_{wp,t}^{s'}$. Again we have contradicted our hypothesis that $\bar{b}_{wp,t}^{s} \neq \bar{b}_{wp,t}^{s'}$.

Now suppose that $\bar{e}_{wp,t}^{s} \neq \bar{e}_{wp,t}^{s'}$. This implies that at least one of $\bar{e}_{wp,t}^{s}$ and $\bar{e}_{wp,t}^{s'}$ is non-zero. Therefore we can use (13) to infer that $\bar{b}_{wp,t}^{(\cdot)} = 1$ for the corresponding scenario. Using the same logic as above, we can infer that $\bar{Z}_{t-1}^{s,s'} = True$, and therefore $\bar{e}_{wp,t}^{s} = \bar{e}_{wp,t}^{s'}$, which contradicts our hypothesis. □

Based on Propositions 1 and 2 together with model (SPM), we define model $(SPM2)$ where the disjunctions and logic constraints in (30)-(31) are applied over a reduced set of combinations for s, s' ((39)-(40)). Also, some of the equality constraints inside the disjunct corresponding to $Z_t^{s,s'} = True$ in (31) are now applied unconditionally (38).

$$(SPM2) \qquad \phi = \qquad \max \textstyle\sum_s p^s NPV^s$$

$$\text{s.t. (2)-(28), (29)}$$

$$b_{wp,t}^{s} = b_{wp,t}^{s'} \quad \forall (t,s,s'), t > 1, (s,s') \in \mathcal{L}^1, \{wp\} = \mathcal{D}(s,s') \tag{38a}$$

$$e_{wp,t}^{s} = e_{wp,t}^{s'} \quad \forall (t,s,s'), t > 1, (s,s') \in \mathcal{L}^1, \{wp\} = \mathcal{D}(s,s') \tag{38b}$$

$$Z_t^{s,s'} \Leftrightarrow \left[\bigwedge_{\tau=1}^{t} \left(\neg b_{wp,\tau}^{s} \right) \right] \quad \forall (t,s,s'), (s,s') \in \mathcal{L}^1, \{wp\} = \mathcal{D}(s,s') \tag{39}$$

$$\begin{bmatrix} & Z_t^{s,s'} & \\ b_{wp,t+1}^{s} = & b_{wp,t+1}^{s'} & \forall wp \notin \mathcal{D}(s,s') \\ e_{wp,t+1}^{s} = & e_{wp,t+1}^{s'} & \forall wp \notin \mathcal{D}(s,s') \\ b_{pp,t+1}^{s} = & b_{pp,t+1}^{s'}, & \forall pp \\ e_{pp,t+1}^{s} = & e_{pp,t+1}^{s'} & \forall pp \\ b_{wp,wp',t+1}^{s} = & b_{wp,wp',t+1}^{s'} & \forall (wp,wp') \\ b_{wp,pp,t+1}^{s} = & b_{wp,pp,t+1}^{s'} & \forall (wp,pp) \\ q_{wp,t}^{prod,s} = & q_{wp,t}^{prod,s'} & \forall wp \end{bmatrix} \vee \left[\neg Z_t^{s,s'} \right] \quad \forall (t,s,s'), (s,s') \in \mathcal{L}^1 \tag{40}$$

$$b_{wp,t}^{s} \in \{0,1\} \qquad \forall (wp,t,s), \qquad b_{pp,t}^{s} \in \{0,1\} \qquad \forall (pp,t,s),$$

$b^s_{wp,wp',t} \in \{0,1\}$	$\forall(wp,wp',t,s),$	$b^s_{wp,pp,t} \in \{0,1\}$	$\forall(wp,pp,t,s),$
$e^s_{wp,t} \geq 0$	$\forall(wp,t,s),$	$e^s_{pp,t} \geq 0$	$\forall(pp,t,s),$
$q^{prod,s}_{wp,t} \geq 0$	$\forall(wp,t,s),$	$Z^{s,s'}_t \in \{True, False\}$	$\forall(t,s,s'), s < s'$

Theorem 1. *The optimal solutions of* (SPM) *and* $(SPM2)$ *are the same.*

Proof. Since the objective functions of (SPM) and $(SPM2)$ are the same, it is sufficient to show that the feasible regions of these models are the same.

Any feasible solution of (SPM) will satisfy (2)-(28), (29) and (38a)-(38b) (from Proposition 2). In addition, since any feasible solution of (SPM) will satisfy (30)-(31), therefore it will also satisfy (39)-(40) (since the domain of (s, s') in (39)-(40) is a subset of the domain of (s, s') in (30)-(31)). Thus, any feasible solution of (SPM) is also a feasible solution of $(SPM2)$.

Now consider a feasible solution of $(SPM2)$. This solution will satisfy (2)-(28), (29)-(29f). Also, this solution will satisfy (35a)-(35f) for $s < s', t = 1$ (from (29a)-(29f)) and (36)-(37) for all t and $(s, s') \in \mathcal{L}^1$ (combining (38)-(40)). Thus, from Proposition 1, this solution will satisfy (30)-(31). Thus, any feasible solution of $(SPM2)$ is also a feasible solution of (SPM). □

5 Branch and Bound Algorithm

One approach for solving model $(SPM2)$ is to reformulate it as an MILP by converting the logic constraints and disjunctions into linear constraints using a big-M approach (Raman and Grossmann (1994)). Since the size of the resulting MILP problem can be very large, we consider in this section a Lagrangean duality based branch and bound algorithm. Model $(SPM2)$ is coupled in scenarios through the non-anticipativity constraints. In the proposed branch and bound algorithm, an upper bound to the optimal solution at any node is obtained by solving a Lagrangean dual problem that is obtained by relaxing these non-anticipativity constraints. Lower bounds at each node are obtained by generating feasible solutions heuristically from the solution of the Lagrangean dual. Non-anticipativity constraints violated by the solution of the Lagrangean dual are used in the branching step of the algorithm to partition the feasible space.

5.1 Upper bound generation

We illustrate the formulation of the Lagrangean dual problem at the root node. Model $(SPM2_{RLR})$ is obtained from $(SPM2)$ by relaxing constraints (39)-(40) and replacing equality constraints (29), (38) by penalty terms in the objective.

$(SPM2_{RLR})$ $\quad \phi_{RLR}({}^{b}\lambda, {}^{e}\lambda) =$

$$
\begin{aligned}
\max \quad & \sum_{s} p^{s} NPV^{s} \\
& + \sum_{s<s'} \Bigg[\sum_{wp} \left({}^{b}\lambda^{s,s'}_{wp,1} \left(b^{s}_{wp,1} - b^{s'}_{wp,1} \right) + {}^{e}\lambda^{s,s'}_{wp,1} \left(e^{s}_{wp,1} - e^{s'}_{wp,1} \right) \right) \\
& \qquad + \sum_{pp} \left({}^{b}\lambda^{s,s'}_{pp,1} \left(b^{s}_{pp,1} - b^{s'}_{pp,1} \right) + {}^{e}\lambda^{s,s'}_{pp,1} \left(e^{s}_{pp,1} - e^{s'}_{pp,1} \right) \right) \\
& \qquad + \sum_{(wp,wp')} {}^{b}\lambda^{s,s'}_{wp,wp',1} \left(b^{s}_{wp,wp',1} - b^{s'}_{wp,wp',1} \right) + \sum_{(wp,pp)} {}^{b}\lambda^{s,s'}_{wp,pp,1} \left(b^{s}_{wp,pp,1} - b^{s'}_{wp,pp,1} \right) \Bigg] \\
& + \sum_{(s,s') \in \mathcal{L}^1} \sum_{t>1} \left[\sum_{wp \in \mathcal{D}(s,s')} {}^{b}\lambda^{s,s'}_{wp,t} \left(b^{s}_{wp,t} - b^{s'}_{wp,t} \right) + {}^{e}\lambda^{s,s'}_{wp,t} \left(e^{s}_{wp,t} - e^{s'}_{wp,t} \right) \right] \qquad (41) \\
\text{s.t.} \quad & (2) - (28)
\end{aligned}
$$

$$
\begin{aligned}
& b^{s}_{wp,t} \in \{0,1\} && \forall (wp,t,s), && b^{s}_{pp,t} \in \{0,1\} && \forall (pp,t,s), \\
& b^{s}_{wp,wp',t} \in \{0,1\} && \forall (wp,wp',t,s), && b^{s}_{wp,pp,t} \in \{0,1\} && \forall (wp,pp,t,s), \\
& e^{s}_{wp,t} \geq 0 && \forall (wp,t,s), && e^{s}_{pp,t} \geq 0 && \forall (pp,t,s), \\
& q^{prod,s}_{wp,t} \geq 0 && \forall (wp,t,s), && &&
\end{aligned}
$$

In $(SPM2_{RLR})$, parameters ${}^{b}\lambda^{s,s'}_{(\cdot),t}$ and ${}^{e}\lambda^{s,s'}_{(\cdot),t}$ represent the Lagrange multipliers for equality constraints $b^{s}_{(\cdot),t} = b^{s'}_{(\cdot),t}$ and $e^{s}_{(\cdot),t} = e^{s'}_{(\cdot),t}$, respectively. ${}^{b}\lambda$ and ${}^{e}\lambda$ represent the vectors of ${}^{b}\lambda^{s,s'}_{(\cdot),t}$ and ${}^{e}\lambda^{s,s'}_{(\cdot),t}$, respectively, for all relevant combinations of indices $((\cdot), t, s, s')$. $(SPM2_{RLR})$ is an MILP model and clearly a relaxation of $(SPM2)$ for any values of the Lagrange multipliers. Further, for fixed Lagrange multipliers, the solution of $(SPM2_{RLR})$ can be obtained by solving one MILP sub-problem for each scenario. The sub-problem for scenario s will consist of constraints (2)-(28) for scenario s with the objective of maximizing the sum of all terms in (41) involving variables associated with scenario s. We use $\phi^{s}_{RLR}({}^{b}\lambda, {}^{e}\lambda)$ to denote the optimal objective value for the sub-problem corresponding to scenario s for given ${}^b\lambda, {}^e\lambda$. Thus,

$$\phi_{RLR}({}^b\lambda, {}^e\lambda) = \sum_s \phi^s_{RLR}({}^b\lambda, {}^e\lambda) \geq \phi \quad \forall({}^b\lambda, {}^e\lambda)$$

Since this upper bounding property is valid for all values of ${}^b\lambda, {}^e\lambda$, the tightest upper bound on ϕ can be obtained by solving the following Lagrangean dual problem (Guignard and Kim (1987), Nemhauser and Wolsey (1988))

$$\phi_{RLD} = \min_{{}^b\lambda, {}^e\lambda} \phi_{RLR}({}^b\lambda, {}^e\lambda).$$

The above Lagrangean dual problem can be solved approximately by using the sub-gradient algorithm (Fisher, 1981).

5.2 Branching

Let $(\hat{b}, \hat{e}, \hat{q}^{prod})$ represent the solution[4] of the Lagrangean dual. In general, this solution may not satisfy equality non-anticipativity constraints (29) and (38) and/or the conditional non-anticipativity constraints implied by (39) and (40). The branching procedure eliminates infeasibility in the solution of the Lagrangean dual by partitioning the feasible space based on these violated constraints.

5.2.1 Branching on equality non-anticipativity constraints

Similar to Caroe and Schultz (1999), our strategy for branching on equality constraints depends on whether the variables involved are binary or continuous. Suppose the constraint $b^{s_1}_{wp,t} = b^{s_2}_{wp,t} = \ldots = b^{s_k}_{wp,t}$ involving binary variables $b^{(\cdot)}_{wp,t}$ is violated by the solution of the Lagrangean dual. The branching scheme partitions the feasible space in to two regions where $b^{s_1}_{wp,t} = b^{s_2}_{wp,t} = \ldots = b^{s_k}_{wp,t} = 1$ and $b^{s_1}_{wp,t} = b^{s_2}_{wp,t} = \ldots = b^{s_k}_{wp,t} = 0$, respectively. The same strategy is used for branching on violated equality constraints involving other binary variables ($b^{(\cdot)}_{pp,t}, b^{(\cdot)}_{wp,wp',t}$ and $b^{(\cdot)}_{wp,pp,t}$).

On the other hand, if an equality constraint $e^{s_1}_{wp,t} = e^{s_2}_{wp,t} = \ldots = e^{s_k}_{wp,t}$ involving continuous variables $e^{(\cdot)}_{wp,t}$ is violated, we create branches with bounds $e^{s_1}_{wp,t}, e^{s_2}_{wp,t}, \ldots, e^{s_k}_{wp,t} \geq \tilde{e}_{wp,t}$ and $e^{s_1}_{wp,t}, e^{s_2}_{wp,t}, \ldots, e^{s_k}_{wp,t} \leq \tilde{e}_{wp,t}$, where $\tilde{e}_{wp,t}$ is the mean value of variables $e^{s_i}_{wp,t}$ for $1 \leq i \leq k$ in the

[4] Note that $SPM2_{RLR}$ does not have any Z variables

solution of the Lagrangean dual. Mathematically,

$$\tilde{e}_{wp,t} = \frac{\sum_{i=1}^{k} p^{s_i} \hat{e}^{s_i}_{wp,t}}{\sum_{i=1}^{k} p^{s_i}} \tag{42}$$

The same strategy is used for branching on equality constraints involving other continuous variables[5] ($e^{(\cdot)}_{pp,t}$ and $q^{prod,(\cdot)}_{wp,t}$).

Note that $(SPM2)$ involves equality constraints linking variables for only two scenarios at a time, such as $e^{s_1}_{wp,t} = e^{s_2}_{wp,t}$. Similar to the strategy of Caroe and Schultz (1999), we could simply branch on this equality constraint by partitioning the feasible space at $\tilde{e}_{wp,t}$, which is computed from (42) by setting $k = 2$. Instead, we combine all equality constraints involving $e^{(\cdot)}_{wp,t}$ to identify a maximal set of scenarios $s_1, s_2, \ldots, s_k$, with $k \geq 2$, such that $e^{s_1}_{wp,t} = e^{s_2}_{wp,t} = \ldots = e^{s_k}_{wp,t}$ has to hold at the current node. Intuitively, it seems that a large value for k in (42) would yield a partitioning point that removes infeasibility in a large number of equality constraints involving $e^{(\cdot)}_{wp,t}$ (rather than in just one equality constraint, $e^{s_1}_{wp,t} = e^{s_2}_{wp,t}$), and thus lead to more effective branching.

5.2.2 Branching on conditional non-anticipativity constraints

Consider (wp^*, t, s, s') such that $(s, s') \in \mathcal{L}^1$ and $\{wp^*\} = \mathcal{D}(s, s')$. Suppose $\bigwedge_{\tau=1}^{t} \left(\neg \hat{b}^s_{wp^*,\tau} \right) = True$, but at least one of the following equality constraints is not satisfied.

$$b^s_{wp,t+1} = b^{s'}_{wp,t+1} \quad \forall wp \notin \mathcal{D}(s, s') \tag{43a}$$

$$e^s_{wp,t+1} = e^{s'}_{wp,t+1} \quad \forall wp \notin \mathcal{D}(s, s') \tag{43b}$$

$$b^s_{pp,t+1} = b^{s'}_{pp,t+1} \quad \forall pp \tag{43c}$$

$$e^s_{pp,t+1} = e^{s'}_{pp,t+1} \quad \forall pp \tag{43d}$$

$$b^s_{wp,wp',t+1} = b^{s'}_{wp,wp',t+1} \quad \forall (wp, wp') \tag{43e}$$

$$b^s_{wp,pp,t+1} = b^{s'}_{wp,pp,t+1} \quad \forall (wp, pp) \tag{43f}$$

$$q^{prod,s}_{wp,t} = q^{prod,s'}_{wp,t} \quad \forall wp \tag{43g}$$

Thus, constraints (39)-(40) are violated for (t, s, s'). Figure 6 illustrates the branching strategy

[5] Although there may not be non-anticipatitivity constraints in equality form involving variables $q^{prod,(\cdot)}_{wp,t}$ at the root node, they may be introduced at other nodes.

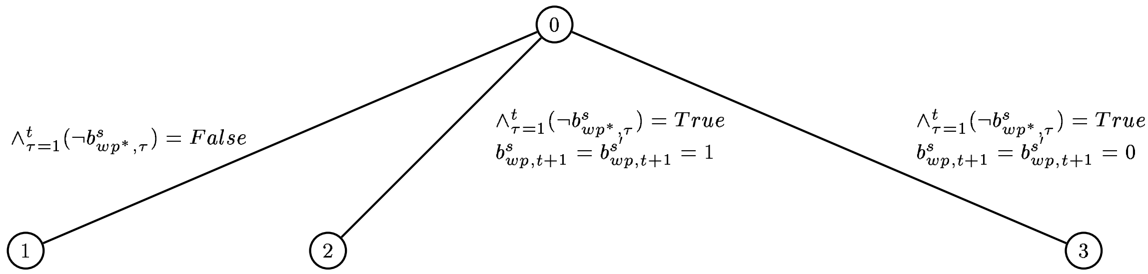

Figure 6: Branching on conditional non-anticipativity constraints (39)-(40)

when $\bigwedge_{\tau=1}^{t} \left(\neg \hat{b}_{wp^*,\tau}^{s}\right) = True$ but $\hat{b}_{wp,t+1}^{s} \neq \hat{b}_{wp,t+1}^{s'}$ for some $wp \notin \mathcal{D}(s,s')$. The three-way branching scheme enforces the rule that either scenarios s, s' are not indistinguishable after investments in time period t or they are indistinguishable and $b_{wp,t+1}^{s} = b_{wp,t+1}^{s'}$ holds. Note that constraints (43) will be dualized in addition to (29) and (38) when the Lagrangean dual is formulated at nodes 2 and 3 in Figure 6.

The same branching strategy is used if $\bigwedge_{\tau=1}^{t} \left(\neg \hat{b}_{wp^*,\tau}^{s}\right) = True$, but an equality in (43) involving other binary variables ((43c), (43e), (43f)) is violated. On the other hand, if $\bigwedge_{\tau=1}^{t} \left(\neg \hat{b}_{wp^*,\tau}^{s}\right) = True$, but $\hat{e}_{wp,t+1}^{s} \neq \hat{e}_{wp,t+1}^{s'}$ for some $wp \notin \mathcal{D}(s,s')$, then the branching scheme enforces the rule that either scenarios s, s' are not indistinguishable after investments in time period t or they are indistinguishable and $e_{wp,t+1}^{s}, e_{wp,t+1}^{s'} \leq \tilde{e}_{wp,t+1}$ or $e_{wp,t+1}^{s}, e_{wp,t+1}^{s'} \geq \tilde{e}_{wp,t+1}$, where $\tilde{e}_{wp,t+1}$ is the probability weighted mean of $\hat{e}_{wp,t+1}^{s}$ and $\hat{e}_{wp,t+1}^{s'}$. The same branching strategy is used if $\bigwedge_{\tau=1}^{t} \left(\neg \hat{b}_{wp^*,\tau}^{s}\right) = True$, but an equality in (43) involving other continuous variables ((43d), (43g)) is violated.

At any node in the branch and bound algorithm, the set of non-anticipativity constraints in equality form will consist of constraints (29) and (38), and equalities (43) for (t, s, s') such that $(s, s') \in \mathcal{L}^1$ and $Z_t^{s,s'}$ is fixed to $True$ at that node. Non-anticipativity constraints in conditional form will consist of logic constraints (39) and disjunctions (40) for (t, s, s') such that $(s, s') \in \mathcal{L}^1$ and $Z_t^{s,s'}$ is a free variable at the node. The Lagrangean dual at any node is formulated by dualizing the non-anticipativity constraints in equality form and relaxing those in conditional form. As part of the branching step, the feasible region is partitioned based on one of these non-anticipativity constraints in equality or conditional form by using the appropriate strategy.

5.3 Lower bound generation

Lower bounds at each node are obtained by generating feasible solutions of $(SPM2)$ heuristically from the solution of the Lagrangean dual. In this procedure, we first use a heuristic to fix all binary decisions for installation of WPs and PPs so that (29a), (29c), (38a), (39) and (40) are satisfied. With these decisions fixed, $(SPM2)$ is reformulated as an MILP and solved in full space.

Binary investment decisions for WPs and PPs are fixed in order of increasing t. We illustrate the procedure for fixing the binary investment decisions for WPs. Let $\bar{b}^s_{wp,t}$ represent the value to which $b^s_{wp,t}$ is fixed by this heuristic procedure. In order to fix decisions $b_{wp,t}$, we use the values of $\bar{b}^s_{wp',\tau}, 1 \leq \tau \leq t-1$ (since binary investment decisions for WPs are fixed in order of increasing t) together with (39) and (40) to identify all s' such that $(s, s') \in \mathcal{L}^1$ and $b^s_{wp,t} = b^{s'}_{wp,t}$ has to be satisfied. These equalities are combined with (29a) and (38a) to identify the set of scenarios $s_1, s_2, \ldots, s_k$ such that $b^{s_1}_{wp,t} = b^{s_2}_{wp,t} = \ldots = b^{s_k}_{wp,t}$ has to be satisfied. Using scenario probabilities as weights, we compute α, the net weight in favor of fixing $b^{s_i}_{wp,t} = 1$ for all i such that $1 \leq i \leq k$. If $\alpha > 0$, where α is computed as shown below, then we fix $\bar{b}^{s_i}_{wp,t} = 1$ for all i such that $1 \leq i \leq k$. Otherwise, we fix $\bar{b}^{s_i}_{wp,t} = 0$ for all i such that $1 \leq i \leq k$.

$$\alpha = \sum_{i=1}^{k} \hat{b}^{s_i}_{wp,t} p^{s_i} - \sum_{i=1}^{k} (1 - \hat{b}^{s_i}_{wp,t}) p^{s_i}$$

Binary investment decisions for PPs are fixed using the same logic.

5.4 Remarks

(i) It is important to note that logic inferencing (Hooker (2000)) on Boolean and discrete variables can significantly impact the quality of upper bounds in the proposed algorithm. Specifically, constraints (29a), (29c), (29e), (29f), (38a), (39) and (40) can be used for logic inferencing on variables b and Z. For example, if a node in the branch and bound tree has the restriction $b^{s_1}_{wp=A,1} = 0$, then we can use (39) to infer that $Z^{s_1,s_2}_1 = True$ for s_2 such that $\mathcal{D}(s_1, s_2) = \{A\}$. Thus, equality constraints (43) for $(t = 1, s = s_1, s' = s_2)$ can be added to the model at that node and can be dualized when the Lagrangean dual at that node is formulated. These modifications may significantly impact the value of the Lagrangean dual at that node. Therefore, the logic inferencing step should be executed before the Lagrangean dual is solved at any node.

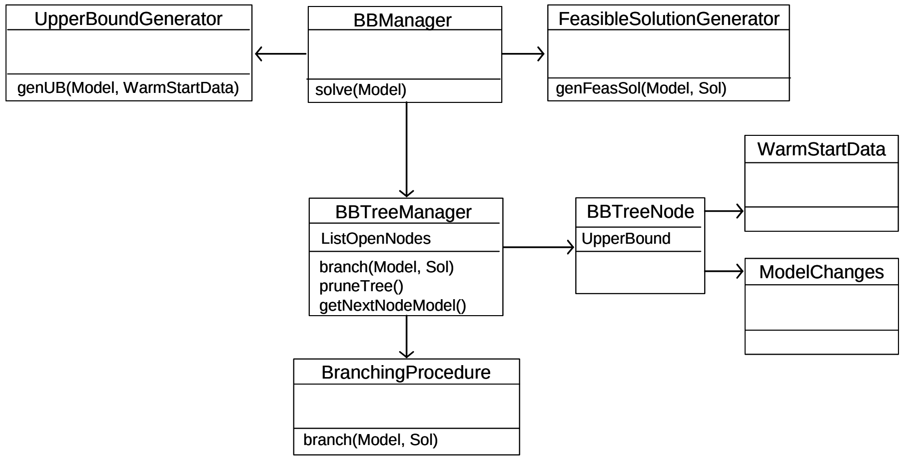

Figure 7: Object oriented design for implementation of branch and bound algorithms

(ii) It should be noted that since this algorithm involves branching on continuous variables, some stopping criterion is needed to avoid infinite branching on these variables. As explained by Caroe and Schultz (1999), if the feasible region is bounded and if we branch parallel to the coordinate axes, then we can stop after the l_∞-diameter of the feasible sets of the sub-problems has fallen below a certain threshold. The algorithm is then guaranteed to converge finitely.

5.5 Implementation

Figure 7 represents an object-oriented design (Deitel and Deitel (2003)) for the implementation of branch and bound algorithms. In this representation, each rectangular box represents a *class* while an arrow connecting class A to class B represents that each *object* of class A has a reference to an object of class B among its data members. Only the most important data members and member functions for each class are shown. In this design, the class `BBManager` controls the branch and bound algorithm. A `BBManager` object has access to objects of classes `BBTreeManager`, `UpperBoundGenerator` (maximization case) and `FeasibleSolutionGenerator`. The `BBTreeManager` maintains the branch and bound tree by pruning unwanted nodes and by creating child nodes, when needed, with the help of the class `BranchingProcedure`. Each node in the branch and bound tree is represented by an object of class `BBTreeNode`. The `ModelChanges` object associated with each `BBTreeNode` object stores information on how the model at that node differs from that at its parent. The WarmStartData object associated with each BBTreeNode object stores warm start information to be used when upper bounds are being generated for any of its child nodes.

The user invokes the branch and bound algorithm on Model by calling function solve(Model), which is a member of class BBManager. At any iteration of the algorithm, this function calls function getNextNodeModel(), a member of class BBTreeManager, which returns NodeModel, the model at the next node, and the WarmStartData to be used for solving this model. The BBTreeManager generates NodeModel by using the ModelChanges information at the node and all its ancestors. NodeModel is passed to function genUB(NodeModel,WarmStartData), a member of class UpperBoundGenerator, which develops and solves a relaxation of NodeModel. NodeSol, the solution of the relaxation of NodeModel, is passed to the function genFeasSol(Model,NodeSol), a member of class FeasibleSolutionGenerator, which attempts to generate a feasible solution to Model by using NodeSol as a starting point. Based on the upper and lower bounds thus generated, the BBManager either asks the BBTreeManager to prune the tree or asks it to branch on the current node. In the latter case, the BBManager collects the necessary information from the UpperBoundGenerator and passes it to BBTreeManager as a WarmStartData object. This information is stored at the current node and will be used when upper bounds are being generated at child nodes of the current node.

We believe that this object-oriented design can be used to implement any branch and bound algorithm by customizing the upper-bound generation procedure, the feasible solution generation scheme and the branching procedure. We refer to the classes providing to these functionalities as the *variant* classes. Since the warm start data will depend on the solution procedure being used in the UpperBoundGenerator, while the changes in model from one node to another will depend on the branching strategy, WarmStartData and ModelChanges are also variant classes. The remaining classes, namely, BBManager, BBTreeManager and BBTreeNode will remain the same irrespective of the algorithm being implemented. Thus, we will refer to these as the *invariant* classes. The proposed object-oriented design can be used to develop a library that provides concrete implementations for the invariant classes and abstract implementations for the variant classes. By implementing *derived* classes that embed algorithm specific functionality for each of the variant classes, this library can be used to implement any branch and bound algorithm with significant reduction in development time.

We have implemented the proposed algorithm in C++ based on this object-oriented design. The model has been implemented using ILOG's Concert Technology modeling libraries. In our implementation, the class `GFUpperBoundGenerator`, which is derived from `UpperBoundGenerator`, has access to objects of user defined classes `PreProcessor` and `LagrangeanDualSolver`. The `PreProcessor` object embeds a logic-model in variables b and Z based on constraints (29a), (29c), (29e), (29f), (38a), (39) and (40). Before function `genUB(NodeModel,WarmStartData)`, member of class `GFUpperBoundGenerator`, develops the relaxation of `NodeModel`, it passes `NodeModel` to the `PreProcessor`. The `PreProcessor` uses the logic-model to fix additional variables in `NodeModel`. The pre-processed model is passed to the `LagrageanDualSolver` object, which uses the sub-gradient algorithm (Fisher (1981)) to solve the Lagrangean dual. The `LagrangeanDualSolver` has access to an `IloCplex` object (ILOG (2003)) that is called to solve each of the MIP sub-problems.

The derived class corresponding to the `FeasibleSolutionGenerator` also has access to an `IloCplex` object that is called to solve the reduced MIP after binary investment variables for WPs and PPs have been fixed by the `FeasibleSolutionGenerator`. The derived class corresponding to `WarmStartData` stores the optimal Lagrange multipliers at a node while the derived class corresponding to `ModelChanges` stores the bounds on the branch connecting a node to its parent node.

Finally, the derived class corresponding to `BranchingProcedure` implements the branching scheme explained in section 5.2. In this scheme, we gave higher priority to branching on equality constraints than on conditional non-anticipativity constraints. Among the equality constraints, we gave highest priority to branching on constraints involving $b^{(\cdot)}_{wp,t}$, followed by those involving $e^{(\cdot)}_{wp,t}$, $b^{(\cdot)}_{pp,t}$, $e^{(\cdot)}_{pp,t}$, $b^{(\cdot)}_{wp,wp',t}$, $b^{(\cdot)}_{wp,pp,t}$ and $q^{prod,(\cdot)}_{wp,t}$, in order of decreasing priority. This priority can be justified as follows. During the solution of the Lagragean dual, decisions for each scenario are optimized independently and therefore they violate the relaxed non-anticipativity constraints. Since the scenarios differ only in the properties associated with WPs (or, more precisely, with the fields corresponding to the WPs), it is natural to expect that this violation comes about because investment decisions for WPs in every scenario will be optimal with respect to the properties of the WPs in that scenario. Thus, enforcing the non-anticipativity constraints on investment decisions for WPs seems intuitively most important.

	Example 1 (3 scenarios)		Example 2 (9 scenarios)		Example 5 (81 scenarios)	
	(SPM)	$(SPM2)$	(SPM)	$(SPM2)$	(SPM)	$(SPM2)$
Constraints	7,541	7,457	43,337	29,045	2,735,597	386,597
Variables	4,480	4,480	13,843	13,573	169,696	125,956
0-1 Variables	594	594	1,782	1,782	16,281	16,281
CPU time (sec.)	51	41	1,200	1,200	Out	75,000
Lower Bound ($ Million)	99.551	99.551	85.534	85.566	of	137.834
Optimality gap	$< 1\%$	$< 1\%$	84.74%	21.63%	memory	77.70%

Table 2: MILP reformulations of (SPM) and $(SPM2)$

6 Results

In this section, we present results for five sample problems to highlight the effectiveness of the properties presented in section 4 and the performance of the proposed branch and bound algorithm. Examples 1, 2 and 5 have been taken from Goel and Grossmann (2004b) while examples 3 and 4 are new. Examples 1, 2, 3, 4 and 5 have 3, 9, 9, 81 and 81 scenarios, respectively. All results presented below have been obtained on a Pentium-IV, 2.4 GHz Linux machine. We present detailed qualitative results for example 3 only. Results for other examples are summarized.

Table 2 illustrates the effectiveness of propositions 1 and 2 by comparing the sizes of the big-M reformulations of models (SPM) and $(SPM2)$. The status of ILOG CPLEX 9.0 during the solution in full space of these reformulations is also compared. An optimality gap of 1% has been used as a stopping criterion. It can be seen that for examples 1, 2 and 5, the propositions achieve 1%, 33% and 86% reduction, respectively, in the number of constraints. While the impact on the performance of ILOG CPLEX 9.0 is relatively small for example 1, the reduction in model size leads to a much tighter optimality gap for example 2 after 1,200 CPU seconds. In the case of example 5, the computer ran out of memory even before the root node could be solved during the solution of the big-M reformulation of (SPM). It is clear that the smaller size of $(SPM2)$ makes solution of this model more manageable. Thus it is evident that propositions 1 and 2 have a significant impact on model size and their importance increases with increase in the number of scenarios. On the other hand, it is also clear that even with these reductions, the size of the problem can still become very large, thereby neccessitating the use of the proposed branch and bound algorithm.

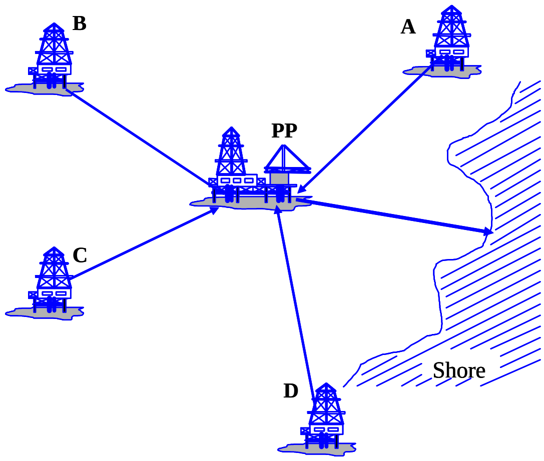

Figure 8: Superstructure for Example 3

6.1 Example 3

Figure 8 shows the super-structure for an offshore site with four fields, denoted by letters A-D, and one potential location for a production platform, PP. Investment and operation decisions have to be made for this site over a project horizon of 15 years (15 time periods). For ease of illustration, we assume that the uncertainty is restricted to the sizes of fields B and C. Possible realizations for the size of field B include 300, 400 and 500 BSCF with probabilities 0.3, 0.4 and 0.3, respectively. Similarly, possible realizations for the size of field C include 150, 300 and 450 BSCF with probabilities 0.3, 0.4 and 0.3, respectively. Fields A and D have sizes 300 and 250 BSCF, respectively. Initial deliverabilities for fields A, B, C, and D are 130, 120, 150 and 140 MSCF/D, respectively[6]. The sizes of fields B and C in each of the nine scenarios together with the probabilities of these scenarios are shown in Figure 4. The set $\mathcal{D}(s, s')$ is given in Table 1.

Figure 9 shows the branch and bound tree traversed by the proposed algorithm. The nodes are numbered in the order they are visited. All Lagrangean multipliers are initialized to zero during the solution of the Lagrangean dual at the root node. The solution of the Lagrangean dual yields an upper bound of 68.912 while the feasible solution generation scheme generates a lower bound of 65.767. In the solution of the Lagrangean dual, variables $b^s_{wp=C,t=1}$ take values $1, 0, 0, 0, 0, 1, 0, 0, 1$ for $s = 1, 2, \ldots, 9$, which clearly violates the equality constraints $b^1_{C,1} = b^2_{C,1} = \ldots = b^8_{C,1} = b^9_{C,1}$ implied by (29a). Thus, the feasible space is bifurcated in

[6]MSCF/D = Million Standard Cubic Feet per Day

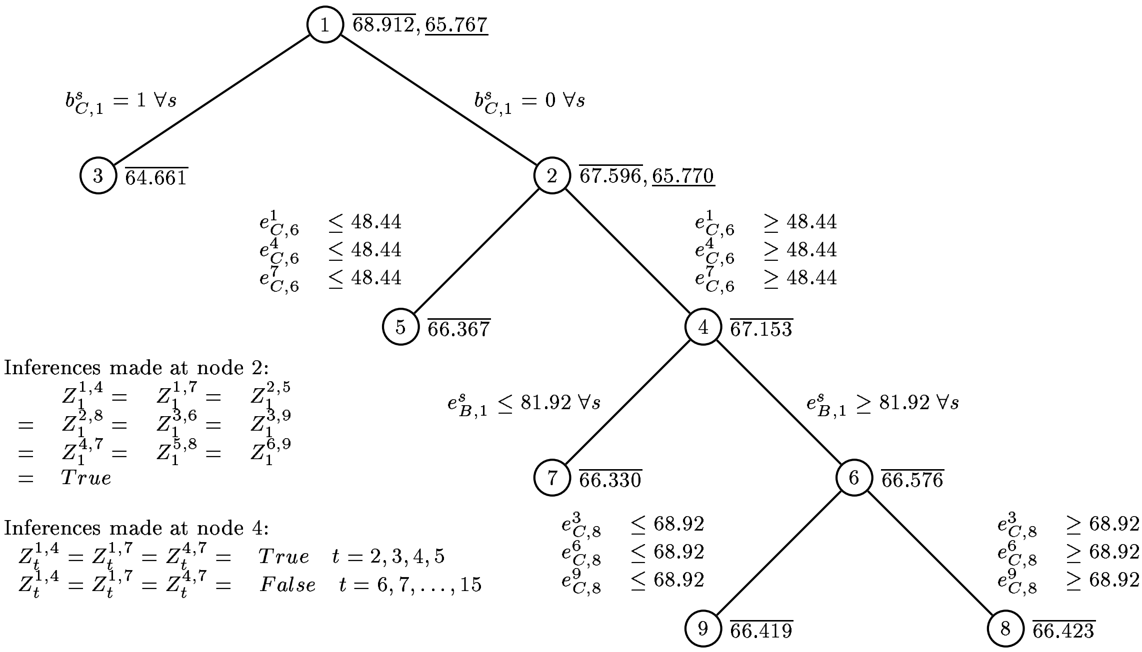

Figure 9: Branch and bound tree for Example 3 (1% optimality gap used as stopping criterion)

to regions $b^1_{C,1} = b^2_{C,1} = \ldots = b^8_{C,1} = b^9_{C,1} = 0$ and $b^1_{C,1} = b^2_{C,1} = \ldots = b^8_{C,1} = b^9_{C,1} = 1$, represented by nodes 2 and 3, respectively.

During the solution of the Lagrangean dual at nodes 2 and 3, the Lagrange multipliers are initialized to the optimal values of these multipliers at node 1. A logic inferencing step is executed at nodes 2 and 3 before the Lagrangean dual problems are solved at these nodes. As shown in Figure 9, this step is able to fix $Z^{s,s'}_t = True$ for $(t,s,s') \in \{(1,1,4),(1,1,7),(1,2,5),(1,2,8),(1,3,6),(1,3,9),(1,4,7),(1,5,8),(1,6,9)\}$. Equality constraints (43) for these combinations of (t,s,s') are also dualized when the Lagrangean dual at node 2 is formulated. Lagrange multipliers for these constraints are initialized to zero during the solution of the Lagrangean dual. At node 2, upper and lower bounds of 67.596 and 65.770, respectively, are generated. In the solution of the Lagrangean dual, variables $e^1_{C,6}, e^4_{C,6}$, and $e^7_{C,6}$ take values 69.17, 69.21 and 0.00 respectively. These values violate the equality constraints $e^1_{C,6} = e^4_{C,6} = e^7_{C,6}$ (implied by (38a) since $(1,4),(1,7) \in \mathcal{L}^1$ and $\mathcal{D}(1,4) = \mathcal{D}(1,7) = \{C\}$; see Table 1). Thus, the feasible space is bifurcated into two regions about the mean of $69.17, 69.21, 0.00$, which is 48.44.

The algorithm is stopped after the optimality tolerance of 1% is reached. The lower bound

	Scenario								
	1	4	7	2	5	8	3	6	9
$t = 1$	A(66.68), B(83.32), PP(150.00)								
$t = 6$	C(73.86)								
$t = 7$				C(73.33)					
$t = 8$							C(73.33)		
$t = 10$	D(66.26)								
$t = 12$		D(69.75)		D(70.54)					
$t = 13$			D(49.01)		D(55.18)		D(60.99)		
ENPV	\$65.770 Million								

Table 3: Solution for Example 3 from branch and bound algorithm

	Scenario								
	1	4	7	2	5	8	3	6	9
$t = 1$	A(67.66), B(82.34), PP(150.00)								
$t = 7$	C(72.13)			C(64.70)			C(69.21)		
$t = 10$	D(63.95)								
$t = 12$		D(67.83)	D(44.25)	D(70.54)					
$t = 13$					D(54.27)		D(62.09)		
ENPV	\$62.794 Million								

Table 4: Solution for Example 3 from heuristic developed by Goel and Grossmann (2004b)

generated at node 2 corresponds to the best solution found by the algorithm. This solution (Table 3) proposes installation of WPs at fields A and B in year 1 with capacities 66.68 and 83.32 MSCF/D, respectively. Installation of the WP at field C is proposed in years 6, 7 or 8 depending on whether the size of field B is 300 (scenarios 1, 4 and 7), 400 (scenarios 2, 5 and 8) or 500 BSCF (scenarios 3, 6 and 9), respectively. Finally, installation of the WP at field D is proposed in six of the nine scenarios.

Table 4 gives the best solution found by the heuristic proposed by Goel and Grossmann (2004b). In that paper, the authors aimed to obtain a good feasible solution to (SPM), rather than the optimal solution. Specifically, their heuristic aims to obtain the best solution that satisfies $b^s_{wp,t} = b^{s'}_{wp,t}$ for all (t, s) and all WPs wp that have uncertainty associated with them, in addition to satisfying (2)-(28) and (29)-(31). This extra restriction prevents solutions from proposing investment at field wp in different time periods based on information obtained about other fields if the quality of reserves of field wp is uncertain. Hence, the best solution found by this heuristic (ENPV = \$62.794 Million) forces the WP at field C to be installed in year 7

	Scenario								
	1	4	7	2	5	8	3	6	9
$t = 1$	B(80.79), C(69.21), PP(150.00)								
$t = 4$	A(66.87)			A(63.42)			A(57.37)		
$t = 6$		A(66.68)	A(58.57)						
$t = 7$					A(66.68)				
$t = 8$						A(61.67)		A(66.02)	
$t = 10$	D(70.54)								A(65.81)
$t = 11$				D(66.39)					
$t = 12$		D(70.67)					D(63.88)		
$t = 13$					D(56.13)				
ENPV	\$61.214 Million								

Table 5: Solution for Example 3 from deterministic approach

irrespective of the size information obtained for field B. The solution obtained by the proposed branch and bound algorithm (ENPV = \$65.770 Million) cannot be found by the heuristic. Note that since field D does not have any associated uncertainty, the heuristic allows the WP at field D to be installed in different time periods in different scenarios.

Table 5 gives the solution found by using a deterministic approach. In this approach, a deterministic problem obtained by assuming mean values for all uncertain parameters is first solved. The decisions proposed by the solution are implemented until more information is obtained. The new information obtained is then used to update the model which is then re-solved to find "optimal" future decisions given that decisions up till that time have already been implemented. The solution from the above approach proposes investment in both uncertain fields in the first year, and has an ENPV of \$61.214 Million which is \$4.516 Million less than the ENPV of the solution obtained by the proposed branch and bound algorithm. It should be noted that the feasible space of (SPM) contains the solution spaces searched by the deterministic approach and the heuristic of Goel and Grossmann (2004b). Since the branch and bound algorithm gives the optimal solution of (SPM), it is guaranteed to always give solutions at least as good as these two methods.

Computational Performance

Table 6 compares the performance of the proposed branch and bound algorithm with that of ILOG CPLEX 9.0 applied to the big-M reformulation of $(SPM2)$ in full space. Note that similar

	Branch and Bound			Full space with ILOG CPLEX 9.0			
				In time $2 \cdot t$		In time $10 \cdot t$	
	t (CPU sec.)	Lower Bound ($ Million)	Opt. gap	Lower Bound ($ Million)	Opt. gap	Lower Bound ($ Million)	Opt. gap
Example 1	169	99.551	$< 1\%$	In 41 CPU sec. ($< t$), lower bound = 99.551, Opt. gap $< 1\%$			
Example 2	556	92.057	$< 1\%$	85.566	21.63%	91.694	7.45%
Example 3	1,683	65.770	$< 1\%$	65.567	20.55%	65.567	15.38%
Example 4	18,552	86.746	$< 1\%$	27.883	655.57%	71.3067	151.97%
Example 5	37,204	146.695	$< 1\%$	137.834	77.70%	141.578	32.90%

Table 6: Comparison of proposed branch and bound algorithm with ILOG CPLEX 9.0

	Heuristic (Goel and Grossmann (2004b))		
	CPU time (sec.)	Lower Bound ($ Million)	Opt. gap
Example 1	34	99.551	0.00%
Example 2	127	92.057	0.21%
Example 3	80	62.794	16.21%
Example 4	589	74.881	37.35%
Example 5	3,034	146.316	9.77%

Table 7: Results for heuristic proposed by Goel and Grossmann (2004b)

	Branch and bound: root node		
	CPU time (sec.)	Lower Bound ($ Million)	Opt. gap
Example 1	152	99.551	1.57%
Example 2	455	92.047	3.68%
Example 3	390	65.767	4.78%
Example 4	3,798	86.094	6.76%
Example 5	3,901	146.695	3.25%

Table 8: Status of branch and bound algorithm at root node

	Solution from deterministic approach ($ Million)	Best solution found for (SPM) ($ Million)
Example 1	94.557	99.551
Example 2	84.507	92.057
Example 3	61.214	65.770
Example 4	84.188	86.746
Example 5	143.432	146.695

Table 9: Comparison of deterministic and stochastic approaches

to Example 3, Example 4 also involves optimizing the investment and operation decisions for a four field site over a 15 year time horizon. However, in Example 4 we consider uncertainty in the sizes of *all* four fields, with three possible realizations for each uncertain parameter. Thus, Example 4 involves 81 scenarios. Also, the field data is different from that in Example 3. Further details for Example 4 are not provided for sake of brevity.

The branch and bound algorithm is stopped when the optimality gap falls below 1%. The CPU time required by the branch and bound algorithm is represented by t in Table 6. The status of ILOG CPLEX 9.0 is reported after $2t$ and $10t$ CPU seconds for each of the examples. It can be seen that the proposed algorithm out-performs the solution of the full space model using ILOG CPLEX in all examples except for example 1, which is also the smallest one. For all other examples, ILOG CPLEX 9.0 could not achieve the same optimality gaps even if provided one order of magnitude greater time than the proposed algorithm. In addition, for most of the examples the best feasible solution generated by ILOG CPLEX 9.0 in $10t$ CPU seconds is significantly inferior to that generated by the proposed algorithm in t CPU seconds.

Results obtained using the heuristic proposed by Goel and Grossmann (2004b) are presented in Table 7. It can be seen that the proposed branch and bound algorithm requires significantly more CPU time (see Table 6). The trade-off, however, is that the proposed algorithm yields improved solutions in three of the five test problems. Furthermore, it is interesting to note that the proposed algorithm generates very good solutions at the root node itself (see Table 8). In fact, the feasible solutions generated at the root node are better than the solutions obtained from the heuristic by Goel and Grossmann (2004b) in three of the five test problems. The solution times for solving the root node in the proposed algorithm are still greater than the solution times for the heuristic by Goel and Grossmann (2004b), but the differences are considerably smaller.

Finally, Table 9 shows that for each of the five examples, the best known solution for ($SPM2$)

(obtained from the proposed algorithm) is significantly better than the solution obtained from a deterministic approach, thus validating the importance of addressing the uncertainty in the gas field problem.

7 Conclusions

Goel and Grossmann (2004b) presented a hybrid mixed-integer/disjunctive model stochastic programming model and a solution heuristic for the optimal planning for development of gas fields under uncertainty in reserves. We have presented a set of theoretical properties satisfied by any feasible solution of this model with the goal of achieving a reduction in model size. It is shown that the effectiveness of these properties increases as the number of scenarios increases. The number of constraints in the model is reduced by 86% for the largest test problem considered. This in turn has a direct impact on the solvability of the model.

We have also presented a Lagrangean duality based branch and bound algorithm to solve the stochastic programming model rigorously and efficiently. It was shown that the proposed algorithm achieves significantly better solutions and tighter optimality gaps than the heuristic of Goel and Grossmann (2004b), especially for the larger problems. Also, ILOG CPLEX 9.0, the state-of-the-art commercial solver as applied to the full space model cannot match the performance of the proposed algorithm even if provided one order of magnitude greater CPU time.

Acknowledgments

The authors acknowledge financial support from ExxonMobil Upstream Research Company and from the Gas Research Institute under award R-510-RPSEA.

Nomenclature

Indices

t, τ	Time periods, $t \in \{1, 2, \ldots, T\}$
wp, wp'	Well platform or corresponding field

pp	Production platform
s, s'	Scenario

Sets

$\Theta_{wp,1}$	Set of realizations for the size of field wp
$\Theta_{wp,2}$	Set of realizations for the initial deliverability of field wp
$\mathcal{D}(s, s')$	Set of fields that have different properties in scenarios s and s'
$\mathcal{L}^1$	Set of scenario pairs (s, s') such that $s < s'$ and s, s' differ in realization of exactly one uncertain parameter

Optimization Variables

$b^s_{(\cdot),t}$	Whether or not structure $(\cdot)$ is installed in time period t
$Z^{s,s'}_t$	Whether or not scenarios s, s' are indistinguishable after investment in time period t
$q^{prod,s}_{wp,t}$	Gas production rate at field wp in time period t; scenario s
$q^{deliv,s}_{wp,t}$	Deliverability of field wp in time period t; scenario s
$q^{cum,s}_{wp,t}$	Cumulative production from field wp at end of time period t; scenario s
$q^{out,s}_{(\cdot),t}$	Gas flow-rate out of structure $(\cdot)$ in time period t; scenario s
$q^{shr,s}_t$	Gas flow-rate to shore in time period t; scenario s
$cap^s_{wp,t}$	Capacity of well platform wp in time period t
$cap^s_{pp,t}$	Capacity of production platform pp in time period t; scenario s
$e^s_{wp,t}$	Expansion in capacity of well platform wp in time period t; scenario s
$e^s_{pp,t}$	Expansion in capacity of production platform pp in time period t; scenario s
$CC^{tot,s}_t$	Total capital costs in time period t; scenario s
$OC^{tot,s}_t$	Total operating costs in time period t; scenario s
$Rev^{tot,s}_t$	Total revenues in time period t; scenario s
NPV^s	Net Present Value in scenario s

Parameters

p^s	Probability of scenario s
$\theta^s_{wp,1}$	Size of field wp in scenario s

$\theta^{s}_{wp,2}$	Initial deliverability of field wp in scenario s
δ_t	Number of days in time period t
$shrink$	Shrinkage factor for flow in pipeline towards shore
$M_{(\cdot)}$	Upper bound on capacity of $(\cdot)$
α_t	Discounting factor for time period t
$FCC_{(\cdot)}$	Fixed operating cost for structure $(\cdot)$
$VCC_{(\cdot)}$	Variable capital cost for structure $(\cdot)$
$FOC_{(\cdot)}$	Fixed operating cost for structure $(\cdot)$
$VOC_{(\cdot)}$	Variable operating cost for structure $(\cdot)$
P_t	Price of gas in time period t

References

Barnes, R., Linke, P., Kokossis, A., 2002. Optimization of oil-field development production capacity. In: European Symposium on Computer Aided Process Engineering. Vol. 12. p. 631.

Birge, J. R., Louveaux, F. c., 1997. Introduction to stochastic programming. Springer-Verlag, New York.

Caroe, C. C., Schultz, R., 1999. Dual decomposition in stochastic integer programming. Oper. Res. Lett. 24, 37–45.

Deitel, H., Deitel, P., 2003. C++ How To Program, 4th Edition. Prentice Hall.

Fisher, M. L., 1981. The lagrangian-relaxation method for solving integer programming-problems. Management Science 27 (1), 1–18.

Goel, V., Grossmann, I. E., 2004a. A class of stochastic programs with decision dependent uncertainty. Submitted for publication.

Goel, V., Grossmann, I. E., 2004b. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Computers and Chemical Engineering 28 (8), 1409–1429.

Grothey, A., McKinnon, K., 2000. Decomposing the optimization of a gas lifted oil well network. Tech Rept. MS 00-005, Department of Mathematics and Statistics, University of Edinburgh.

Guignard, M., Kim, S., 1987. Lagrangean decomposition - a model yielding stronger lagrangean bounds. Mathematical Programming 39 (2), 215–228.

Haugen, K. K., 1996. A stochastic dynamic programming model for scheduling of offshore petroleum fields with resource uncertainty. European Journal of Operational Research 88 (1), 88–100.

Hooker, J., 2000. Logic-based methods for optimization: combining optimization and constraint satisfaction. New York : John Wiley and Sons.

Ierapetritou, M. G., Floudas, C. A., Vasantharajan, S., Cullick, A. S., 1999. Optimal location of vertical wells: Decomposition approach. Aiche Journal 45 (4), 844–859.

ILOG, 2003. Ilog cplex 9.0: User's manual. Tech. rep.

Iyer, R. R., Grossmann, I. E., Vasantharajan, S., Cullick, A. S., 1998. Optimal planning and scheduling of offshore oil field infrastructure investment and operations. Ind. Eng. Chem. Res. 37, 1380–1397.

Jonsbraten, T. W., 1998. Optimization models for petroleum field exploitation. Ph.D. thesis, Norwegian School of Economics and Business Administration.

Jornsten, K. O., 1992. Sequencing offshore oil and gas-fields under uncertainty. European Journal of Operational Research 58 (2), 191–201.

Kosmidis, V., Perkins, J., Pistikopoulos, E., 2002. A mixed integer optimization strategy for integrated gas/oil production. In: European Symposium on Computer Aided Process Engineering. Vol. 12.

Lin, X., Floudas, C., 2003. A novel continuous-time modeling and optimization framework for well platform planning problems. Optimization and Engineering 4, 65–95.

Nemhauser, G. L., Wolsey, L. A., 1988. Integer and combinatorial optimization. Wiley, New York, english Series: Wiley-Interscience series in discrete mathematics and optimization.

Ortiz-Gomez, A., Rico-Ramirez, V., Hernandez-Castro, S., 2002. Mixed-integer multiperiod model for the planning of oilfield production. Computers & Chemical Engineering 26 (4-5), 703–714.

Raman, R., Grossmann, I. E., 1994. Modeling and computational techniques for logic-based integer programming. Computers and Chemical Engineering 18 (7), 563–578.

Van den Heever, S. A., Grossmann, I. E., 2000. An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. Ind. Eng. Chem. Res. 39, 1955–1971.

Van den Heever, S. A., Grossmann, I. E., Vasantharajan, S., Edwards, K., 2001. A lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. Ind. Eng. Chem. Res. 40, 2857–2875.