

# A Multistage Stochastic Programming Approach to Open Pit Mine Production Scheduling with Uncertain Geology

Natashia Boland\*, Irina Dumitrescu†, Gary Froyland‡

October 24, 2008

## Abstract

The Open Pit Mine Production Scheduling Problem (OPMPSP) studied in recent years is usually based on a single geological estimate of material to be excavated and processed over a number of decades. However techniques have now been developed to generate multiple stochastic geological estimates that more accurately describe the uncertain geology. While some attempts have been made to use such multiple estimates in mine production scheduling, none of these allow mining and processing decisions to flexibly adapt over time, in response to observation of the geological properties of the material mined. In this paper, we use multiple geological estimates in a mixed integer multistage stochastic programming approach, in which decisions made in later time periods can depend on observations of the geological properties of the material mined in earlier periods. Since the material mined in earlier periods is determined by our decisions, the information received about uncertain properties, and when that information is available, is decision-dependent. Thus we tackle the difficult case of stochastic programming with endogeneous uncertainty. We extend a successful mixed integer programming formulation of the OPMPSP to this stochastic case, and show that non-anticipativity can be modelled with linear constraints involving variables already present in the model. We extend this observation to the general class of endogenous stochastic programs, and exploit the special structure of our model to show that in some cases we can omit a significant proportion of these constraints. Using data supplied by our industry partner, (a multinational mining company), we show that this approach is reasonably tractable, and demonstrate the improvements that can be made to mine schedules through the explicit use of multiple geological estimates.

*Keywords:* stochastic programming, integer programming, open pit mining, endogeneous uncertainty.

## 1 Introduction

We consider an orebody exploited using open pit mining methods. The orebody is represented as a block model: a discretisation of a volume of earth into blocks subject to mining and processing capacities, as well as precedence constraints with respect to the order in which the blocks can be excavated. The Open Pit Mine Production Scheduling Problem (OPMPSP) consists of finding the

---

\*School of Mathematical and Physical Sciences, U. of Newcastle, Australia, [Natashia.Boland@newcastle.edu.au](mailto:Natashia.Boland@newcastle.edu.au)

†School of Mathematics and Statistics, U. of New South Wales, Australia, [irina.dumitrescu@unsw.edu.au](mailto:irina.dumitrescu@unsw.edu.au)

‡School of Mathematics and Statistics, U. of New South Wales, Australia, [g.froyland@unsw.edu.au](mailto:g.froyland@unsw.edu.au)

sequence in which the blocks should be removed from the pit, over the lifetime of the mine, so that the net present value (NPV) of the operation is maximized. The OPMPSP is of great interest to mining companies as the solution of this problem provides a basis for the strategic future development of a new or existing mine. The solution of the OPMPSP and similar problems have been the focus of research for many decades, with the first integer programming approach appearing in the 1980s. Since then, there have been huge improvements in both commercial MIP solvers and models for the OPMPSP so that today, the solution of very large problems are within reach (see Boland et al. (2008), Caccetta and Hill (2003), and Stone et al. (2004)). A shortcoming of these models is that they are based upon a *single estimate of the geology and mineralogy*. In practice, this estimate arises via a smooth interpolation (for example by kriging, see eg. Goovaerts (1997)) of drillhole information. Despite the best efforts of geologists and geostatisticians, this single estimate will contain inaccuracies, as it is based on incomplete geological sampling data. In this paper we take into account this uncertainty via a multistage stochastic programming approach.

We take as our starting point the mixed integer programming model for the OPMPSP given in Boland et al. (2008) and Fricke (2006), which generalizes an earlier model of Caccetta and Hill (2003) that is widely accepted as the basis for the state of the art; this model will be the starting point of our extension to the stochastic case. We begin by defining some notation. The set of all blocks into which the orebody is discretised is denoted by  $\mathcal{K} = \{1, \dots, K\}$ . For each block, a set of possible *attributes* in the block is quantified. For example, an attribute might be the total material (rock) in the block, some mineral of interest, or some impurity. In this paper, we consider only two attributes: for each block  $k \in \mathcal{K}$  we denote by  $a_k^0$  and  $a_k^1$  the total amount of rock and the total metal contained in the block. The use of only two attributes is for simplicity of exposition only; the key ideas in this paper extend easily to multiple attributes. We assume  $a_k^0 > 0$  for each  $k \in \mathcal{K}$  (there is no point considering an empty block). For each block  $k \in \mathcal{K}$ , the set  $\mathcal{P}(k) \subseteq \mathcal{K}$  defines the set of blocks that must be mined before extraction of block  $k$  can begin, in order for the mining operation to be structurally safe. The set of blocks is partitioned into  $N$  subsets, called aggregates. We will use the notation  $\mathcal{A}_i \subseteq \mathcal{K}$  to denote the  $i$ th aggregate; however, for simplicity, we will refer to  $\mathcal{A}_i$  as “aggregate  $i$ ”. The total amount of rock (attribute 0) in aggregate  $i$  will be denoted by  $a_i^0 = \sum_{k \in \mathcal{A}_i} a_k^0$ , and the total amount of base metal (attribute 1) as  $a_i^1 = \sum_{k \in \mathcal{A}_i} a_k^1$ , for each  $i = 1, \dots, N$ . For any aggregate  $i = 1, \dots, N$ , we denote by  $\mathcal{P}(i)$  the set of indices of the aggregates that must be extracted before aggregate  $i$ , which are defined from the block precedence sets via the requirement that one aggregate is in the precedence set of another if any block in the former is in the precedence set of any block in the latter. Note that the *grade* of a block  $k$  (aggregate  $i$ ) is given by the proportion of metal to rock, i.e. by  $a_k^1/a_k^0$  (by  $a_i^1/a_i^0$  in the case of an aggregate).

The lifetime of the mine is divided into  $T$  time periods. The *mining capacity* for period  $t$ , in units of tonnes of rock excavated, is denoted  $M_t$  and the *processing capacity* for period  $t$ , in units of tonnes of rock processed, is denoted  $P_t$ , for each  $t \in \{1, \dots, T\}$ . Estimates of the cost per tonne of rock extracted and processed in each period  $t \in \{1, \dots, T\}$  are denoted by  $c_t^{mng}$  and  $c_t^{proc}$ , respectively. The forecast of the revenue per tonne of metal sold in period  $t$  is denoted by  $c_t^1$  for each  $t = 1, \dots, T$ . It is assumed that the revenue is realised in the period in which the material is processed. All costs and revenues are given in present value terms, and are assumed to be nonnegative.

Binary variables  $x_{i,t} \in \{0, 1\}$  are defined to indicate whether extraction of aggregate  $i$  has begun by or in period  $t$ , i.e.  $x_{i,t} = 1$ , if excavation of aggregate  $i$  begins in periods  $1, \dots, t$ , and  $x_{i,t} = 0$ , otherwise. Continuous variables  $y_{i,t} \in [0, 1]$  are defined to represent the fraction of aggregate  $i$  to be excavated in time period  $t$ , and continuous variables  $z_{i,t} \in [0, 1]$  are defined to represent the fraction of aggregate  $i$  to be processed in time period  $t$ , for any  $t = 1, \dots, T$ ,  $i = 1, \dots, N$ . It is assumed that an aggregate is excavated uniformly, in the sense that every block in the aggregate is excavated in

the same proportion. In other words, it is assumed that the fraction of any block  $k$  excavated in any time period  $t$  is  $y_{i(k),t}$ , where  $i(k)$  is the (unique) aggregate to which block  $k$  belongs, i.e  $k \in \mathcal{A}_{i(k)}$ . Another assumption is that any unprocessed material in an excavated aggregate is sent to waste, (there is no stockpiling), so for each aggregate  $i$  and time period  $t$ , the value  $y_{i,t} - z_{i,t}$  represents the fraction of aggregate  $i$  that goes to waste. We can now give the (*deterministic*) mixed integer program for OPMPSP that is the starting point for this paper, which we will refer to as D-MIP:

$$\max \sum_{t=1}^T \sum_{i=1}^N ((c_t^1 a_i^1 - c_t^{proc} a_i^0) z_{i,t} - a_i^0 c_t^{mng} y_{i,t})$$

$$\text{s.t. } z_{i,t} \leq y_{i,t}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, \quad (1)$$

$$\sum_{i=1}^N a_i^0 z_{i,t} \leq P_t, \quad \forall t \in \{1, \dots, T\}, \quad (2)$$

$$\sum_{i=1}^N a_i^0 y_{i,t} \leq M_t, \quad \forall t \in \{1, \dots, T\}, \quad (3)$$

$$x_{i,t} \leq \sum_{\hat{i}=1}^t y_{j,\hat{i}}, \quad \forall i \in \{1, \dots, N\}, j \in \mathcal{P}(i), t \in \{1, \dots, T\}, \quad (4)$$

$$\sum_{\hat{i}=1}^t y_{i,\hat{i}} \leq x_{i,t}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, \quad (5)$$

$$x_{i,t} \leq x_{i,t+1}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T-1\}, \quad (6)$$

$$x_{i,t} \in \{0, 1\}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, \quad (7)$$

$$y_{i,t} \geq 0, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\} \text{ and} \quad (8)$$

$$z_{i,t} \geq 0, \quad \forall t \in \{1, \dots, T\}, i \in \{1, \dots, N\}. \quad (9)$$

To better capture geological uncertainty, mining companies can produce multiple stochastically generated geological estimates that we call *scenarios*; see for example Dimitrakopoulos (1998), Goovaerts (1997), or Journel (1996). Each scenario so produced provides (possibly different) attribute values for each block; typically the block model would be held constant. One might also expect rock values to be identical across all scenarios, i.e. uncertainty in metal attribute values is of primary interest. The process used for estimate generation is usually described as conditional simulation, and each scenario so generated is compatible with the measured drillhole information with a probability of occurring derived in the course of the simulation. The model of the material to be excavated is now represented as a finite number of possible scenarios, each with a known probability of occurrence.

While the technology to produce scenarios has existed for some time, there has been relatively little research on augmenting OPMPSP formulations to explicitly incorporate them. The importance of this was noted by Smith and Dimitrakopoulos (1999) who discussed the following experiment. Suppose an NPV-maximal production schedule is created based upon a single kriged geological estimate, and this schedule is applied to several conditionally simulated geological estimates. While the schedule applied to the kriged estimate performed very well with regular production each year, the variations of annual production were significant when the schedule was applied to the various conditionally simulated estimates. Smith and Dimitrakopoulos conclude their study by strongly

advocating the inclusion of stochastic geological estimates in the optimization process to avoid this effect.

In the intervening years, several research groups have addressed the issue. Ramazan and Dimitrakopoulos in a series of papers (see Ramazan and Dimitrakopoulos (2007) and references therein) control ore grade and total production for each individual scenario and impose a penalty in the objective for grade and production deviations from targets, using a predefined cutoff grade policy. Menabde et al. (2004) describe a variable cutoff grade model to maximize expected NPV over several scenarios, however, the production rates are only controlled in an average sense, leading to possible fluctuations in practice. In a more recent paper by Golamnejad et al. (2006), uncertain mineral grades are taken into account via chance constraints on the average blended product grade and also in the objective function with grade affecting revenue. However, the delineation of blocks into waste and ore is done a priori and without regard to grade uncertainty, effectively enforcing a fixed cutoff grade policy based upon some grade average. For a recent review of existing models and algorithms for the OPMPSP with stochastic elements we refer the reader to Osanloo et al. (2007). A major drawback with the existing approaches is that the mining and processing decision variables are completely independent of scenarios; that is, *the same decision is applied to each scenario*.

In practice, however, decisions should *adapt over time* according to material that is excavated. Clearly as mining occurs, and assays of the material extracted are obtained, the mineral properties of the ore body could be better estimated than at the outset. Such new information gained through mining could, and clearly should, influence future mine scheduling decisions. This motivates a multistage stochastic programming approach to mine scheduling, in which decisions made at one period can depend on information revealed through mining in earlier periods. The advantages of such stochastic programming approaches are well known in general (see for example Birge and Louveaux (1997)), and as already noted, are well motivated in the mining context of Smith and Dimitrakopoulos (1999).

However the information that is revealed through mining depends on the mining decisions made in earlier periods; what we learned through mining depends on what blocks or aggregates we chose to mine. This puts us in the case of stochastic programming with *endogenous uncertainty*, defined to occur when the underlying stochastic process depends on the optimization decisions, whether via direct alteration of the probability distributions, or by reduction of the uncertainty through the discovery of new information (referred to as *type 1* and *type 2* forms respectively in Goel and Grossmann (2006)). In such cases, one might see a further influence of stochasticity on decision-making: as noted, for example, in Artstein and Wets (1993), one could, and perhaps should, invest some resources in estimation of parameters defining the uncertainty. In mining terms, one may thus see some mining decisions made primarily to reduce uncertainty in the geological estimates, i.e. the process of mining should, perhaps, be viewed as a process that is part “exploration” and part “extraction”. Multistage stochastic programming with endogenous uncertainty allows all this to be considered as an integrated planning problem, with the single aim of maximizing expected NPV. As noted, for example by Jonsbråten et al. (1998), endogenous uncertainty makes the problem substantially more difficult to solve, and even very recent papers (e.g. Tarhan and Grossmann (2008)) comment on the sparsity of literature addressing such cases.

Interestingly, the key work addressing multistage stochastic programs with endogenous uncertainty has, at least in part, been motivated by another application in the exploitation of geological resources: the development of gas fields. Goel et al. (2006) build on the general approach of Goel and Grossmann (2006) to multistage stochastic programs with endogenous uncertainty to develop a method of optimizing the design of infrastructure and operational planning over time for a gas production asset. Their work clearly demonstrates the benefits of stochastic programming in this

context, which has strong analogies with open-pit mining. Their results on five examples show improvements in expected NPV due to stochastic programming ranging from 2.3% to 8.9%, with an average of 5.4%. This paper, together with that of Tarhan and Grossmann (2008), (also based on the general approach of Goel and Grossmann (2006)), represents the state of the art in methods for multistage stochastic programming with endogenous uncertainty, particularly of type 2; other than references in Goel et al. (2006), Goel and Grossmann (2006), Tarhan and Grossmann (2008), we are aware of no other work that tackles such problems.

In this paper we address the need to take into account geological uncertainty in open-pit mine production scheduling, to produce schedules that adapt over time in response to the information acquired through mining. We develop a multistage stochastic programming approach, which has endogenous uncertainty (of type 2), and which is designed to capitalize on the types of multiple geological estimates that can be produced by methods such as described in Dimitrakopoulos (1998). This introduces an additional difficulty, as such estimates are sampled from a continuous space, without an underlying scenario tree structure. We are not aware of any previous work that deals with such a case. In Section 2 we develop a stochastic model which includes practical measures to deal with this difficulty, and which also reflects some of the decision-making practices that could be expected of mine planning engineers in response to new mineralogical information. We call the OPMPSP that incorporates geological uncertainty in this way the Stochastic OPMPSP (SOPMPSP). In Section 3 we show how the Stochastic OPMPSP can be modelled as a mixed integer linear program. As noted by Goel et al. (2006) in the gas field context, standard approaches in stochastic programming cannot be used, and instead the interdependence between the scenarios (*scenario tree* in their case) and the decisions must be captured by conditional non-anticipativity constraints. Where Goel et al. (2006), (and also Goel and Grossmann (2006)), introduce binary variables for every pair of scenarios and each time period to capture the non-anticipativity condition, within a disjunctive model, we show that the binary variables arising naturally in the formulation can give rise to a linear non-anticipativity constraint. This approach can readily be generalized to provide an alternative to that of Goel and Grossmann (2006). Where Goel and Grossmann use Lagrangian relaxation to deal with the large number of binary variables needed for their non-anticipativity condition, and specialized branch-and-bound to deal with their disjunctions, we use standard branch-and-bound techniques. In cases where the number of non-anticipativity constraints grows large, we suggest implementing them as “lazy” constraints, that are only added as needed. This helps keep the size of the formulation manageable, and may also be viable in the general context. (In our experiments in Section 4, we in fact found performance was better without making the non-anticipativity constraints lazy.) In this section, we also exploit the special structure of our model to show that in some cases we may omit non-anticipativity constraints related to processing decisions. We also provide a direct proof that in general a significant proportion of non-anticipativity constraints can be omitted. (The latter result can be viewed as a special case of one given by Goel and Grossmann (2006), which requires a long, technical proof; in our special case the proof is natural and short.) Using data supplied by our industry partner, (a multinational mining company), we give in Section 4 numerical results demonstrating the improvements in mine schedules that are possible with this approach, and confirming that it is reasonably tractable.

## 2 The Stochastic Model

Methods for producing multiple geological estimates, such as those given in Dimitrakopoulos (1998), typically produce a set of scenarios, indexed by  $\mathcal{S} = \{1, \dots, S\}$ , defined over a single block model

with block set  $\mathcal{K}$ , where each scenario  $s \in \mathcal{S}$  specifies attribute values for each block  $k \in \mathcal{K}$ . We denote by  $a_{k,s}^0$  the total rock, and by  $a_{k,s}^1$  the total metal, contained in block  $k$  under scenario  $s$  for each  $k \in \mathcal{K}$  and each  $s \in \mathcal{S}$ . The total amount of rock in aggregate  $i$  under scenario  $s$  will be denoted by  $a_{i,s}^0 = \sum_{k \in \mathcal{A}_i} a_{k,s}^0$ , and the total amount of metal as  $a_{i,s}^1 = \sum_{k \in \mathcal{A}_i} a_{k,s}^1$ . We expect that precedence relationships do not vary from scenario to scenario. It is also likely that the rock attribute will remain constant over all scenarios, in most cases; the primary focus in considering geological uncertainty is the uncertainty in grade. Each scenario  $s$  has a given probability  $p(s)$  of occurring, with  $\sum_{s=1}^S p(s) = 1$ .

Whilst the set of such scenarios is finite, the scenario-dependent attribute values are drawn from a continuous space, and are not generated via a scenario tree structure. Typically, the attribute values for a given block are different under every scenario; no two scenarios are likely to have the same attribute values for a single block, or for a single aggregate. This introduces something of a challenge: if one takes these attributes at face value, then simply by learning the attribute values of a single aggregate, we can correctly identify the entire scenario. A more realistic interpretation is to take each scenario as representing a range of “nearby” realizations of aggregate attribute values; this is more consistent with the positive value the estimate generator has dictated for the probability that the scenario will occur. An alternative view is based on the likelihood of experimental error in assay values: when an aggregate is assayed, the assay attribute values observed are unlikely to be exactly those of the aggregate itself; any “nearby” value may be those of the aggregate. Thus scenarios taking “similar” values on a particular aggregate may not be reliably distinguished when the aggregate is assayed. In both cases the outcome is the same: we should not distinguish between two scenarios just because their attribute values on a single aggregate are different; instead we should ask that the values are “sufficiently different”. Although the context is somewhat different, this is not unlike the practical learning model used in Jonsbråten et al. (1998), in which the uncertainty in cost of production of a family of products can only be resolved if at least a certain threshold fraction of such products are produced.

Formally, we distinguish between two scenarios in the following way, predicated on the primary measure of interest: grade. For any aggregate  $i \in \{1, \dots, N\}$  and scenario  $s \in \mathcal{S}$ , we denote by  $g_{i,s} = a_{i,s}^1/a_{i,s}^0$  the *grade* of aggregate  $i$  under scenario  $s$ . Given two scenarios  $r, s \in \mathcal{S}$  with  $r \neq s$ , the *differentiator* of  $r$  and  $s$  is the set of aggregates  $i \in \{1, \dots, N\}$  for which  $g_{i,r}$  is “sufficiently different” from  $g_{i,s}$ , i.e. for which  $|g_{i,r} - g_{i,s}|$  is “sufficiently large”. Since attribute values could be drawn from somewhat different scales, depending on the type of metal, (or impurity, if one is considering multiple attributes), we suggest a relative measure of difference, taken to be a fraction of the largest range of grades observed under all scenarios, taken over all aggregates. Mathematically, this is stated as below.

**Definition 2.1** For any two (distinct) scenarios  $r, s \in \mathcal{S}$ , we call the set

$$\Delta_{\{r,s\}}^\alpha = \{i \in \{1, \dots, N\} : |g_{i,r} - g_{i,s}| > \alpha\delta\}$$

the  $\alpha$ - $\{r, s\}$ -differentiator, where  $\alpha \geq 0$  and  $\delta = \max_{i \in \{1, \dots, N\}} \left( \max_{r' \in \{1, \dots, S\}} g_{i,r'} - \min_{r' \in \{1, \dots, S\}} g_{i,r'} \right)$ .

We discuss practical choices of  $\alpha$  in Section 4, but assume throughout that  $\alpha \geq 0$ .

We now consider how learning occurs in the mining context, and how and when decisions can be made in response to information learned.

We take an aggregate  $i \in \{1, \dots, N\}$  to be “open for mining” in the earliest period  $t \in \{1, \dots, T\}$  for which all preceding aggregates have been fully excavated; without loss of generality we can assume  $x_{i,t} = 1$  in this case. When an aggregate is open for mining, it is fully exposed, and can be assayed

to estimate its attribute values. So as soon as  $x_{i,t} = 1$ , we assume that we can distinguish any pair of scenarios  $r, s \in \mathcal{S}$ ,  $r \neq s$ , with  $i \in \Delta_{\{r,s\}}^\alpha$ .

How quickly can decision-makers react to information acquired in the course of mining? We believe that in practice, decisions about how much of an aggregate to send to the processing plant are made quite dynamically, and can be changed rapidly in response to new information. In contrast, mining decisions are likely to experience a considerable lag. For example, to change the mining sequence may require a dragline to be moved to another area of the mine. Since draglines are very large, unwieldy pieces of equipment, this decision will entail a disruption in mine production, and is not a decision to be made lightly. In contrast, changing the fraction of an aggregate to send for processing may be simply a matter of re-directing trucks to dump their load in a different area of the mine. Mining decisions are thus typically major strategic decisions, requiring significant time to implement, whereas processing decisions are more tactical, and can be made over a shorter timeframe.

To model these practical differences, we define two different forms of decisions dependent on information learned. If a lag is required between the acquisition of new information and reacting to that information in decision-making, we call the decision-making *conservative*. For simplicity, we assume here that the lag in conservative decision-making is one period, so that conservative decisions must wait until period  $t + 1$  to differentiate based on information gained in period  $t$ . Our industry partners in this research suggest that it may well be appropriate in some mining contexts to allow a longer lag; fortunately it is not difficult to extend the models we present here to allow an arbitrary lag. In this work, we take mining decisions, namely the decisions on mining sequence and rates dictated by the  $x$  and  $y$  variables in D-MIP, to be conservative, with a lag of one period.

If, on the other hand, decision-making can respond rapidly to new information, we call that form of decision *non-conservative*. We allow non-conservative decisions to differentiate within the same period in which the information is gained. Here we take processing decisions, namely the  $z$  variables in the D-MIP model, to be non-conservative. Of course this may be a somewhat optimistic assumption. As we will see in the analysis in Section 3, aggregates open for mining in the same period compete for processing capacity in that period, and whether an aggregate is newly opened for mining early or late within the period may well make a difference to the decision-makers ability to respond. However the OPMPSP model does not sequence mining activities within each period, and so provides no basis to distinguish. Indeed the time discretization inherent in the OPMPSP model already introduces some approximation of this type; we believe taking the processing decisions to be non-conservative is a reasonable assumption along the same lines. If a particular practical mining context does not warrant this assumption, our models are easily modified to make processing decisions, too, conservative, by imitating the approach taken for mining decisions.

In what follows, we develop a stochastic programming model in which processing and mining decisions can depend on the observed properties of material already mined (or open for mining). In particular, we will allow all of the  $x$ ,  $y$  and  $z$  variables to depend on the scenario, subject to appropriate non-anticipativity constraints. We find, as do Goel and Grossmann (2006), (also Goel et al. (2006) and Tarhan and Grossmann (2008)), that the endogeneous uncertainty necessitates *conditional* non-anticipativity constraints. Unlike Goel and Grossmann, we do not introduce new binary variables for each pair of scenarios and each time period to model the condition; instead we show this may be naturally modelled by the  $x$  variables, with linear constraints, and so keep the number of variables at  $\mathcal{O}(NTS)$ . Although  $\mathcal{O}(NTS^2)$  such constraints are required, implementing them as “lazy” constraints may help keep the model size modest. (We shall see in Section 4 that in fact computational performance is better for the data we tested without making these constraints lazy.)

We develop our model in two steps. First, we consider the case that *only* processing decisions

depend on scenario. As well as facilitating exposition, this case is of interest for practical reasons. Highly conservative mine planners may prefer not to change mining sequence at all. By treating and testing stochastic processing decisions as a separate case, we can clearly demonstrate to such planners the gains in expected NPV due solely to stochastic mining decisions. Furthermore, the special structure of the OPMPSP leads to a very nice property for the model with only processing decisions dependent on scenario: provided the scenarios preserve the *ordering* of aggregates induced by grade, over aggregates with “similar” grade, (not in the differentiator), non-anticipativity constraints are unnecessary. (Identical rock attribute values across scenarios are also required, but as mentioned earlier, we believe this is the rule rather than the exception.)

In our second step, we extend the model to allow all decisions to depend on scenario. At first sight, it would appear as if  $4N(T-1)S(S-1)$  non-anticipativity constraints are needed for mining decisions, but it is not difficult to prove that in fact only half of these are necessary. (We provide a short, direct proof, but this observation also follows from Theorem 1 of Goel and Grossmann (2006).) Unlike Goel and Grossmann, we are unable to eliminate many more of these constraints. Their Theorem 2 (in Goel and Grossmann (2006)) allows a dramatic reduction in the number of necessary non-anticipativity constraints, however this theorem depends on scenarios derived from the cross product of random variables taking on discrete values. In our mining case, this would be equivalent to scenarios generated via cross products of attribute values over all aggregates. Such a cross product construction is not supported by geological considerations: attribute properties of aggregates are not independent, but are spatially linked, and furthermore are sampled from an underlying continuous space.

We note that the necessary conditions to eliminate non-anticipativity constraints can be generalized from Theorem 2 of Goel and Grossmann (2006), to apply to arbitrary sets of scenarios. We give a brief discussion in Section 3.3 of how this is possible, where we also explain how, in general, one may avoid the additional binary variables Goel and Grossman use to model the non-anticipativity condition, by using variables already present in their model.

### 3 Integer Programming Models for Stochastic OPMPSP

In this section we show how to extend D-MIP to the stochastic case, first with stochastic processing decisions only, and then with stochastic mining decisions as well. We give a final section discussing how some of our ideas can be used in the general case of multistage stochastic programming with endogenous uncertainty.

#### 3.1 Scenario-Dependent Processing Decisions

In this section we extend D-MIP to the case where information is used to influence processing decisions, but not mining decisions. In other words, we replace the  $z$  variables with variables  $z_{i,t}^s \in [0, 1]$  that represent the fraction of aggregate  $i$  to be processed in time period  $t$ , under scenario  $s$ , for any  $t = 1, \dots, T$ ,  $i = 1, \dots, N$  and  $s \in \{1, \dots, S\}$ . The binary variables  $x_{i,t} \in \{0, 1\}$  and the continuous variables  $y_{i,t} \in [0, 1]$  are defined as in the deterministic case. As in the deterministic case, we assume that an aggregate is excavated uniformly and that any unprocessed material in an aggregate is sent to waste, so for each aggregate  $i$  and time period  $t$ , the value  $y_{i,t} - z_{i,t}^s$  represents the fraction of aggregate  $i$  that goes to waste in period  $t$ , if scenario  $s$  occurs.

Our mixed integer programming model for the SOPMPSP with scenario-dependent processing decisions, the SP-MIP, is given below. Of course we will also require appropriate non-anticipativity



constraints. Subsequently we explain how to model these via linear constraints; for now, we simply ask that  $(x, y, z) \in \mathcal{N}^{proc}$ . Thus we obtain the SP-MIP as follows:

$$\max \sum_{t=1}^T \sum_{i=1}^N \sum_{s=1}^S p(s) ((c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s - a_{i,s}^0 c_t^{mng} y_{i,t})$$

$$\text{s.t. } z_{i,t}^s \leq y_{i,t}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \quad (10)$$

$$\sum_{i=1}^N a_{i,s}^0 z_{i,t}^s \leq P_t, \quad \forall t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \quad (11)$$

$$\sum_{i=1}^N a_{i,s}^0 y_{i,t} \leq M_t, \quad \forall t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \quad (12)$$

$$(x, y, z) \in \mathcal{N}^{proc} \quad (13)$$

$$(4) - (8)$$

$$z_{i,t}^s \geq 0, \quad \forall t \in \{1, \dots, T\}, i \in \{1, \dots, N\}, s \in \{1, \dots, S\}. \quad (14)$$

The main differences between this formulation and the deterministic one are the scenario-wise definition of the processing variables  $z_{i,t}^s$  and the addition of constraints (13). Inequalities (10) ensure that the amount of rock of any aggregate that is processed in any given time period, under any scenario, is less than or equal to the amount of rock extracted from that aggregate in the time period considered. Processing and mining capacities must be respected under each scenario. The precedence and other logical constraints are the same as for D-MIP.

Non-anticipativity constraints for the  $z$  variables are needed to ensure that if we have not been able to distinguish between two scenarios  $r, s \in \mathcal{S}$ ,  $r \neq s$ , in time period  $t \in \{1, \dots, T\}$  or any earlier period, (recall processing decisions are taken to be non-conservative), then we must make the same decisions under both scenarios. The only way we can distinguish between  $r$  and  $s$  in period  $t$  is if some aggregate  $i$  which allows us to differentiate between  $r$  and  $s$ , i.e. some  $i \in \Delta_{\{r,s\}}^\alpha$ , has been opened for mining in or before period  $t$ . In other words, if  $x_{i,t} = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ , then it must be that  $z_{j,t}^s = z_{j,t}^r$  for all  $j \in \{1, \dots, N\}$ . Formally the non-anticipativity constraints in SP-MIP can thus be expressed as

$$\mathcal{N}^{proc} = \{(x, y, z) : z_{i,t}^s = z_{i,t}^r, \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \text{ and} \quad (15) \\ \forall t \in \{1, \dots, T\} \text{ such that } x_{j,t} = 0, \forall j \in \Delta_{\{r,s\}}^\alpha\}.$$

In the terminology of Goel and Grossmann (2006), this is a conditional non-anticipativity constraint, predicated, for each  $r, s \in \{1, \dots, S\}$ ,  $r \neq s$ , and each  $t \in \{1, \dots, T\}$ , on the condition that  $x_{j,t} = 0$  for all  $j \in \Delta_{\{r,s\}}^\alpha$ .

Fortunately, there is a convenient linear expression to capture this condition:  $\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t} = 0$  if the condition holds (and, for binary  $x$ ,  $\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t} \geq 1$  otherwise). Now since  $z_{i,t}^s \in [0, 1]$  for all  $i \in \{1, \dots, N\}$ ,  $t \in \{1, \dots, T\}$  and  $s \in \{1, \dots, S\}$ , it is obvious that (for binary  $x$ )

$$|z_{i,t}^r - z_{i,t}^s| \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}$$

provides a valid model of the non-anticipativity constraint. Linearized in the usual way, we deduce that together

$$z_{i,t}^s - z_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, r, s \in \{1, \dots, S\}, r \neq s, \quad (16)$$

and

$$z_{i,t}^r - z_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, r, s \in \{1, \dots, S\}, r \neq s, \quad (17)$$

provide valid constraints with which to replace (13). We may thus take SP-MIP to be the mixed integer linear program defined as follows:

$$\max \left\{ \sum_{t=1}^T \sum_{i=1}^N \sum_{s=1}^S p(s) \left( (c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s - a_{i,s}^0 c_t^{mng} y_{i,t} \right) : (4)-(8), (10)-(12), (16), (17), \text{ and } (14) \right\}.$$

Of course, as the number of scenarios grows, the above formulation will suffer from the non-anticipativity constraints, of which there are  $\mathcal{O}(NTS^2)$ . However, the OPMPSP has special structure associated with the processing decisions. This can be used to show, under relatively mild assumptions, that these non-anticipativity constraints can be omitted altogether! Define SP-MIP-red, which we refer to as the “reduced” model, to be SP-MIP without the non-anticipativity constraints for  $z$ , i.e. we take SP-MIP-red to be SP-MIP with (16) and (17) omitted. We will show that given any optimal solution to SP-MIP-red, we can construct a feasible solution to SP-MIP that has identical objective function value. Since SP-MIP-red is a relaxation of SP-MIP, the solution so constructed must be optimal for SP-MIP. The assumptions under which we may do this are based on the following definition, which, for convenience in what follows, makes use of the complement of the differentiator, defined by

$$E_{\{r,s\}}^\alpha = \{1, \dots, N\} \setminus \Delta_{\{r,s\}}^\alpha$$

for all  $r, s \in \mathcal{S}$ ,  $r \neq s$ .

**Definition 3.1** *We say that scenario set  $\mathcal{S}$  and differentiator parameter  $\alpha$  are grade-order-preserving if all  $r, s \in \mathcal{S}$ ,  $r \neq s$ , and all  $i, j \in E_{\{r,s\}}^\alpha$  we have that*

$$g_{i,s} \geq g_{j,s} \quad \text{if and only if} \quad g_{i,r} \geq g_{j,r}.$$

In other words,  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving if the scenarios in  $\mathcal{S}$  preserve the ordering of aggregates induced by grade over aggregates with “similar” grade (not in the differentiator defined by  $\alpha$ ). We believe this is a relatively mild assumption: since the aggregates of interest have similar grades across the scenarios, there would seem to be little harm in asking that the grade relationships between such aggregates are maintained by the scenarios. To get our result, we also ask that all rock attribute values are identical over aggregates with similar grades across scenarios; as we discussed earlier, we believe this, too, is a mild assumption, even if asked of all aggregates and scenarios.

Before beginning our proof proper, we require the following technical lemma.

**Lemma 3.2** *Let  $\mathcal{V} \subseteq \mathcal{S}$  and  $\mathcal{E} \subseteq \{\{r, s\} \subset \mathcal{V} : r \neq s\}$  be such that the graph  $(\mathcal{V}, \mathcal{E})$  is connected. Let  $Q \subseteq E_{\{r,s\}}^\alpha$  for all  $\{r, s\} \in \mathcal{E}$ . Suppose  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving. Then there exists an ordering  $\sigma : Q \rightarrow \{1, \dots, |Q|\}$ , (one-to-one), of the elements of  $Q$  that preserves grade order for all scenarios in  $\mathcal{V}$ , i.e. such that*

$$g_{i,s} > g_{j,s} \quad \text{implies} \quad \sigma(i) < \sigma(j)$$

for all  $i, j \in Q$ , for all  $s \in \mathcal{V}$ .

**Proof.** We first construct a directed graph  $G = (Q, R)$ , and then show that  $\sigma$  can be taken to be a topological ordering of its nodes. Define the arcs

$$R = \{(i, j) \in Q \times Q : g_{i,s} > g_{j,s}, \exists s \in \mathcal{V}\}.$$

Our first claim is that if  $(i, j) \in R$  then  $g_{i,s} \geq g_{j,s}$  for all  $s \in \mathcal{V}$ . To prove this claim, we consider some  $(i, j) \in R$ . By the definition of  $R$  there must also exist  $s \in \mathcal{V}$  such that  $g_{i,s} > g_{j,s}$ . Let  $r \in \mathcal{V}$ ,  $r \neq s$ . Now  $(\mathcal{V}, \mathcal{E})$  is connected, so there exists a path in  $(\mathcal{V}, \mathcal{E})$  from  $s$  to  $r$ , say the path is  $(s = q^1, q^2, \dots, q^M = r)$ . Now  $\{q^m, q^{m+1}\} \in \mathcal{E}$  for all  $m = 1, \dots, M-1$ , so  $Q \subseteq E_{\{q^m, q^{m+1}\}}^\alpha$  for all  $m = 1, \dots, M-1$ , and of course  $i, j \in Q$ . But  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving, so since  $g_{i,s} = g_{i,q^1} > g_{j,s} = g_{j,q^2}$ , we also have that  $g_{i,q^2} \geq g_{j,q^2}$ , and hence that  $g_{i,q^3} \geq g_{j,q^3}$ , and so on by induction, to deduce that  $g_{i,q^M} = g_{i,r} \geq g_{j,q^M} = g_{j,r}$ , proving our claim.

Our second claim is that  $G$  is acyclic. Suppose otherwise. Then there exists a cycle  $(i_1, \dots, i_M)$  with  $(i_m, i_{m+1}) \in R$  for all  $m = 1, \dots, M-1$  and  $(i_M, i_1) \in R$ . Now by the definition of  $R$ , there must exist  $s \in \mathcal{V}$  with  $g_{i_1,s} > g_{i_2,s}$ . From our first claim,  $g_{i_2,s} \geq g_{i_3,s} \geq \dots \geq g_{i_M,s} \geq g_{i_1,s}$ , which is a contradiction.

Finally, let  $\sigma$  be any topological ordering on the nodes of  $G$ , i.e.  $\sigma : Q \rightarrow \{1, \dots, |Q|\}$  is one-to-one and  $(i, j) \in R$  implies that  $\sigma(i) < \sigma(j)$ . Now for any  $i, j \in Q$  and any  $s \in \mathcal{V}$ , if  $g_{i,s} > g_{j,s}$  we have that  $(i, j) \in R$  by the definition of  $R$ , and so  $\sigma(i) < \sigma(j)$ , as required.  $\blacksquare$

**Proposition 3.3** *Assume that  $a_{i,r}^0 = a_{i,s}^0$  for all  $r, s \in \mathcal{S}$ ,  $r \neq s$ , and all  $i \in E_{\{r,s\}}^\alpha$ , and that  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving. Then given any optimal solution to SP-MIP-red, we can construct an optimal solution for SP-MIP.*

**Proof.** Let  $(\hat{x}, \hat{y}, (\hat{z}^s)_{s \in \mathcal{S}})$  be an optimal solution of SP-MIP-red. Any  $(z^s)_{s \in \mathcal{S}}$  that solves SP-MIP-red with  $x = \hat{x}$  and  $y = \hat{y}$  fixed will of course be optimal for SP-MIP-red. We will show that we can choose such a  $(z^s)_{s \in \mathcal{S}}$  that satisfies non-anticipativity constraints (16) and (17), and so must be optimal for SP-MIP. Now SP-MIP-red with  $x = \hat{x}$  and  $y = \hat{y}$  fixed is a problem in  $z$  only, that is obviously separable by time period and scenario, i.e. for each  $t \in \{1, \dots, T\}$  and scenario  $s \in \mathcal{S}$  we only need to ask that  $z_{(\cdot),t}^s = (z_{i,t}^s)_{i=1}^N$  solves

$$\max \sum_{i=1}^N (c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s$$

$$\text{s.t. } z_{i,t}^s \leq \hat{y}_{i,t}, \quad \forall i \in \{1, \dots, N\}, \quad (18)$$

$$\sum_{i=1}^N a_{i,s}^0 z_{i,t}^s \leq P_t, \quad (19)$$

$$z_{i,t}^s \geq 0, \quad i \in \{1, \dots, N\}. \quad (20)$$

We call this the “ $t, s$ -processing-only problem”. Observe that it is a continuous knapsack problem. So any sorting of the aggregates that orders the aggregates by non-increasing value of the ratio of the objective and knapsack constraint coefficients corresponds to an optimal solution, found by “packing” the knapsack greedily in sorted order. For a given  $t$  we will argue that we can choose optimal solutions  $z_{(\cdot),t}^s$  for each  $s \in \mathcal{S}$  that are identical on all pairs of scenarios for which non-anticipativity constraints

should apply, i.e. so that if  $\mathcal{E}_t \stackrel{def}{=} \{\{r, s\} \subseteq \mathcal{S} : r \neq s, \sum_{j \in \Delta_{\{r,s\}}^\alpha} \hat{x}_{j,t} = 0\}$  then we have that vector  $z_{(\cdot),t}^r = z_{(\cdot),t}^s$  for all  $\{r, s\} \in \mathcal{E}_t$ .

To begin, note that if  $c_t^1 = 0$  then of course  $z_{(\cdot),t}^s = 0$  solves the  $t, s$ -processing-only problem for all  $s \in \mathcal{S}$ : clearly this gives an identical solution for each scenario, as required. So suppose otherwise, i.e. suppose that  $c_t^1 > 0$ . Then grade order and objective-to-knapsack-coefficient ratio order are the same:

$$\begin{aligned} & g_{i,s} \geq g_{j,s} \\ \Leftrightarrow & c_t^1 g_{i,s} - c_t^{proc} \geq c_t^1 g_{j,s} - c_t^{proc} \\ \Leftrightarrow & c_t^1 \frac{a_{i,s}^1}{a_{i,s}^0} - c_t^{proc} \geq c_t^1 \frac{a_{j,s}^1}{a_{j,s}^0} - c_t^{proc} \\ \Leftrightarrow & \frac{c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0}{a_{i,s}^0} \geq \frac{c_t^1 a_{j,s}^1 - c_t^{proc} a_{j,s}^0}{a_{j,s}^0} \end{aligned}$$

for all  $i, j \in \{1, \dots, N\}$ .

Now observe that for  $i \in \{1, \dots, N\}$ , if  $i \notin Q_t \stackrel{def}{=} \{j \in \{1, \dots, N\} : \hat{y}_{j,t} > 0\}$ , then by (18) it must be that  $z_{i,t}^s = 0$  for all  $s \in \mathcal{S}$ . Further, note that if  $i \in Q_t$ , then  $\hat{y}_{i,t} > 0$ , so by (5), and since  $\hat{x}$  binary, it must be that  $\hat{x}_{i,t} = 1$ , and so  $i \notin \Delta_{\{r,s\}}^\alpha$  for any  $\{r, s\} \in \mathcal{E}_t$ . Thus  $Q_t \subseteq \{1, \dots, N\} \setminus \Delta_{\{r,s\}}^\alpha = E_{\{r,s\}}^\alpha$  for all  $\{r, s\} \in \mathcal{E}_t$ . Consider the graph  $(\mathcal{S}, \mathcal{E}_t)$ , with node set  $\mathcal{S}$ , induced by  $\mathcal{E}_t$ . This graph can be partitioned into a (possibly empty) set of  $M \geq 0$  connected components,  $\{(\mathcal{V}^m, \mathcal{E}_t^m) : m = 1, \dots, M\}$ , and a (possibly empty) set  $\mathcal{V}^0$  of singleton nodes. Note  $\{\mathcal{V}^m : m = 0, 1, \dots, M\}$  is a partition of  $\mathcal{S}$  and  $\{\mathcal{E}_t^m : m = 1, \dots, M\}$  is a partition of  $\mathcal{E}_t$ .

For each  $s \in \mathcal{V}^0$  we may choose  $z_{(\cdot),t}^s$  to be an arbitrary optimal solution of the  $t, s$ -processing-only problem. For each  $m = 1, \dots, M$ , apply Lemma 3.2 with  $\mathcal{V} = \mathcal{V}^m$ ,  $\mathcal{E} = \mathcal{E}_t^m$ , and  $Q = Q_t$ , to yield an ordering  $\sigma^m$  of  $Q_t$ . Now by Lemma 3.2, for every  $s \in \mathcal{V}^m$ ,  $\sigma^m$  sorts the aggregates of  $Q_t$  in non-increasing order of  $g_{(\cdot),s}$ , and thus also sorts the objective-to-knapsack-coefficient ratios for variables not bounded above by zero in the  $t, s$ -processing-only problem, in non-increasing order. For each such scenario  $s \in \mathcal{V}^m$ , choose  $z_{(\cdot),t}^s$  to be the optimal solution of the  $t, s$ -processing-only problem defined by this ordering  $\sigma^m$ . Now since upper bounds on the  $z$  variables, in (18), are independent of scenario, and since for all  $\{r, s\} \in \mathcal{E}_t^m$  and all  $i \in Q_t \subseteq E_{\{r,s\}}^\alpha$ , the knapsack coefficients  $a_{i,s}^0 = a_{i,r}^0$  are also independent of scenario by assumption, it must be that  $z_{i,t}^r = z_{i,t}^s$  for all  $i \in Q_t$ , and all  $\{r, s\} \in \mathcal{E}_t^m$ . Recall  $z_{i,t}^r = z_{i,t}^s = 0$  for all  $i \notin Q_t$ , so in fact we have  $z_{i,t}^r = z_{i,t}^s$  for all  $i \in \{1, \dots, N\}$ , and all  $\{r, s\} \in \mathcal{E}_t^m$ . Since  $\{\mathcal{V}^m : m = 0, 1, \dots, M\}$  is a partition of  $\mathcal{S}$  we have constructed a solution  $z_{(\cdot),t}^s$  for every  $s \in \mathcal{S}$ , and since  $\{\mathcal{E}_t^m : m = 1, \dots, M\}$  is a partition of  $\mathcal{E}_t$ , we have that  $z_{i,t}^r = z_{i,t}^s$  for all  $i \in \{1, \dots, N\}$ , and all  $\{r, s\} \in \mathcal{E}_t$ . Thus  $(\hat{x}, \hat{y}, z)$  so constructed satisfies the non-anticipativity constraints, and so is feasible for SP-MIP. Since each  $z_{(\cdot),t}^s$  also optimizes the  $t, s$ -processing-only problem,  $(\hat{x}, \hat{y}, z)$  is also optimal for SP-MIP-red, a relaxation of SP-MIP. The result follows.  $\blacksquare$

The above proof also demonstrates that even if  $\mathcal{S}$  and  $\alpha$  are not grade-order-preserving, ignoring non-anticipativity constraints is unlikely to greatly affect the outcome: the differences in objective function coefficients in the  $t, s$ -processing-only problems for variables not already forced to zero reflect the differences in grade over scenarios for which such differences are ‘‘small’’. Thus one could expect to obtain a near-optimal solution by ignoring non-anticipativity constraints, then fixing the  $x$  and  $y$  values, and finally re-solving the problem in  $z$  variables only, but this time applying the non-anticipativity constraints, now conditioned on  $x$ , (so expressed simply as equality constraints between pairs of  $z$  variables). We call this heuristic SP-MIP-h. Numerical results for SP-MIP-h and a comparison with SP-MIP are provided in Table 2, in Section 4.

In the case that the grade-order-preserving property does not hold, we can still eliminate some non-anticipativity constraints. In particular, constraints for an aggregate and pair of scenarios are

redundant if their differentiator set intersects the precedence set for the aggregate. This is intuitively obvious: all processing variables are zero until an aggregate is open for mining, and at this point all preceding aggregates have been mined, and so the scenario pair must have been differentiated.

**Proposition 3.4** *The non-anticipativity constraints for the  $z$  variables are redundant for any  $i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, r, s \in \{1, \dots, S\}, r \neq s$ , for which  $\mathcal{P}(i) \cap \Delta_{\{r,s\}}^\alpha \neq \emptyset$ .*

We omit formal proof here, as a similar result holds for the more general case in the next section: see Proposition 3.8. For now, note that we will refer to the model from which these redundant constraints have been removed as SP-MIP-elim.

### 3.2 Scenario Dependent Mining and Processing Decisions

We now look at the case when both the processing and the mining decisions are allowed to depend on scenarios. In order to model scenario dependence of the mining decisions, we modify the variables  $x$  and  $y$  from SP-MIP. We define binary variables  $x_{i,t}^s \in \{0, 1\}$  to indicate whether extraction of aggregate  $i$  has begun by or in period  $t$  under scenario  $s$ ,  $x_{i,t}^s = 1$ , if excavation of aggregate  $i$  begins in periods  $1, \dots, t$  under scenario  $s$  and  $x_{i,t}^s = 0$  otherwise, and continuous variables  $y_{i,t}^s \in [0, 1]$  to represent the fraction of aggregate  $i$  to be excavated in time period  $t$  under scenario  $s$  for any aggregate  $i = 1, \dots, N$ , time period  $t = 1, \dots, T$  and scenario  $s \in \{1, \dots, S\}$ . We also need to consider non-anticipativity conditions for  $x^s$  and  $y^s$ , and reconsider those for  $z^s$ . Our mixed integer programming model, the SPM-MIP, is given below. Of course we will also require appropriate non-anticipativity constraints. Subsequently we explain how to model these via linear constraints; for now, we simply ask that  $(x, y, z) \in \mathcal{N}^{proc-min}$ . Thus we obtain the SPM-MIP as follows:

$$\max \sum_{t=1}^T \sum_{i=1}^N \sum_{s=1}^S p(s) ((c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s - a_{i,s}^0 c_t^{mg} y_{i,t}^s)$$

$$\text{s.t. } z_{i,t}^s \leq y_{i,t}^s, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \quad (21)$$

$$\sum_{i=1}^N a_{i,s}^0 z_{i,t}^s \leq P_t, \quad \forall t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \quad (22)$$

$$\sum_{i=1}^N a_{i,s}^0 y_{i,t}^s \leq M_t, \quad \forall t \in \{1, \dots, T\}, \quad (23)$$

$$x_{i,t}^s \leq \sum_{\hat{t}=1}^t y_{j,\hat{t}}^s, \quad \forall i \in \{1, \dots, N\}, j \in \mathcal{P}(i), t \in \{1, \dots, T\}, s \in \{1, \dots, S\}, \quad (24)$$

$$\sum_{\hat{t}=1}^t y_{i,\hat{t}}^s \leq x_{i,t}^s, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, s \in \{1, \dots, S\}, \quad (25)$$

$$x_{i,t}^s \leq x_{i,t+1}^s, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T-1\}, s \in \{1, \dots, S\}, \quad (26)$$

$$(x, y, z) \in \mathcal{N}^{proc-min} \quad (27)$$

$$x_{i,t}^s \in \{0, 1\}, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, s \in \{1, \dots, S\}, \quad (28)$$

$$y_{i,t}^s \geq 0, \quad \forall i \in \{1, \dots, N\}, t \in \{1, \dots, T\}, s \in \{1, \dots, S\} \text{ and} \quad (29)$$

$$z_{i,t}^s \geq 0, \quad \forall t \in \{1, \dots, T\}, i \in \{1, \dots, N\}, s \in \{1, \dots, S\}. \quad (30)$$

As for the SP-MIP, inequalities (21) ensure that the amount of rock of any block which is processed in any given time period, under any scenario, is less than or equal to the amount of rock extracted from that block in the time period considered. Inequalities (22) and (23) represent the processing, respectively the mining constraints. Inequalities (24), (25) and (26) ensure that the precedence relationships between aggregates are satisfied.

Non-anticipativity constraints for all variables are needed to ensure that if we have not been able to distinguish between two scenarios by some time, then we make the same decisions under both scenarios. Recall that we take the  $x$  and  $y$  decisions to be conservative, and the  $z$  decisions to be non-conservative. In the conservative case, if we have not been able to distinguish between  $r, s \in \mathcal{S}$ ,  $r \neq s$ , before some time period  $t \in \{1, \dots, T\}$ , then we must make the same decisions under both scenarios, i.e., if  $x_{i,t-1}^s = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ , or if  $x_{i,t-1}^r = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ , (or if  $t = 1$ ), then it must be that  $x_{j,t}^s = x_{j,t}^r$  and  $y_{j,t}^s = y_{j,t}^r$  for all  $j \in \{1, \dots, N\}$ . We note that it seems somehow ambiguous to predicate on either  $x_{i,t-1}^s = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$  or  $x_{i,t-1}^r = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ ; in fact we will show shortly that one side of this disjunction cannot hold without the other. For now, we go on to define non-anticipativity for the non-conservative  $z$  variables. These are nearly the same as given in Section 3.1: since now the  $x$  variables also depend on  $s$  we ask that if  $x_{i,t}^s = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ , or  $x_{i,t}^r = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$ , then it must be that  $z_{j,t}^s = z_{j,t}^r$  for all  $j \in \{1, \dots, N\}$ . Formally the non-anticipativity constraints in SPM-MIP can thus be expressed as

$$\begin{aligned} \mathcal{N}^{proc-min} = \{ (x, y, z) : & x_{i,1}^s = x_{i,1}^r, \quad \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \\ & x_{i,t}^s = x_{i,t}^r, \quad \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \text{ and} \\ & \quad \forall t \in \{2, \dots, T\} \text{ such that } x_{j,t-1}^s = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha \\ & \quad \quad \quad \text{or } x_{j,t-1}^r = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha, \\ & y_{i,1}^s = y_{i,1}^r, \quad \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \\ & y_{i,t}^s = y_{i,t}^r, \quad \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \text{ and} \\ & \quad \forall t \in \{2, \dots, T\} \text{ such that } x_{j,t-1}^s = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha \\ & \quad \quad \quad \text{or } x_{j,t-1}^r = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha, \\ & z_{i,t}^s = z_{i,t}^r, \quad \forall i \in \{1, \dots, N\}, \forall r, s \in \{1, \dots, S\}, r \neq s, \text{ and} \\ & \quad \forall t \in \{1, \dots, T\} \text{ such that } x_{j,t}^s = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha \\ & \quad \quad \quad \text{or } x_{j,t}^r = 0, \quad \forall j \in \Delta_{\{r,s\}}^\alpha \} . \end{aligned}$$

Since all variables lie in the range  $[0, 1]$ , and the  $x^s$  variables are all binary, it is easy to see how

to model these as linear constraints using the same approach as in Section 3.1:

$$x_{i,1}^s = x_{i,1}^r, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, \quad (31)$$

$$x_{i,t}^s - x_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (32)$$

$$x_{i,t}^r - x_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (33)$$

$$x_{i,t}^s - x_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^r, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (34)$$

$$x_{i,t}^r - x_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^r, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (35)$$

$$y_{i,1}^s = y_{i,1}^r, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, \quad (36)$$

$$y_{i,t}^s - y_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (37)$$

$$y_{i,t}^r - y_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (38)$$

$$y_{i,t}^s - y_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^r, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (39)$$

$$y_{i,t}^r - y_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^r, \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{2, \dots, T\} \quad (40)$$

$$z_{i,t}^s - z_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^s, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{1, \dots, T\} \quad (41)$$

$$z_{i,t}^r - z_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^s, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{1, \dots, T\} \quad (42)$$

$$z_{i,t}^s - z_{i,t}^r \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^r, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{1, \dots, T\} \quad (43)$$

$$z_{i,t}^r - z_{i,t}^s \leq \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^r, \quad \forall i \in \{1, \dots, N\}, r, s \in \{1, \dots, S\}, r \neq s, t \in \{1, \dots, T\} \quad (44)$$

We may thus take SPM-MIP to be the mixed integer linear program defined as follows:

$$\max \left\{ \sum_{t=1}^T \sum_{i=1}^N \sum_{s=1}^S p(s) \left( (c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s - a_{i,s}^0 c_t^{mng} y_{i,t}^s \right) : (21) - (26), (28) - (30), \text{ and } (31) - (44) \right\}.$$

Clearly, the non-anticipativity constraints constitute a rather large class of constraints. Fortunately, as mentioned earlier, one side of the disjunction  $x_{i,t}^s = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$  or  $x_{i,t}^r = 0$  for all  $i \in \Delta_{\{r,s\}}^\alpha$  cannot hold without the other. This allows us to eliminate about half of the non-anticipativity constraints, as we only need to use one of  $x^r$  or  $x^s$  on the right-hand side, we do not need constraints for both.

**Lemma 3.5** *If  $x$  satisfies (26), (28), (31), (32) and (33), then for any pair of distinct scenarios  $r, s \in \mathcal{S}$ ,*

$$\mathbf{Ind} \left( \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^r \right) = \mathbf{Ind} \left( \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^s \right), \forall t = 1, \dots, T,$$

where  $\mathbf{Ind}$  denotes the indicator function, i.e., for  $\xi \geq 0$ ,  $\mathbf{Ind}(\xi) = 0$  if  $\xi = 0$  and  $\mathbf{Ind}(\xi) = 1$  otherwise.

**Proof.** The case  $t = 1$  follows from (31). Let  $t \in \{1, \dots, T - 1\}$  be the earliest period for which

$$\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t}^s > 0, \quad (45)$$

or if no such period exists, take  $t = T$ . Then for all  $t' = 1, \dots, t - 1$ , and since  $x$  is binary, it must be that

$$\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t'}^s = 0.$$

Then by (32) and (33), and (31), we have that  $x_{i,t'}^r = x_{i,t'}^s$  for all  $t' = 1, \dots, t$ . Thus

$$\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t'}^r = \sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t'}^s$$

for all  $t' = 1, \dots, t$ . We conclude that since  $x_i^s, x_i^r$  are increasing vectors over time by (26), and by (45), the result must follow.  $\blacksquare$

This lemma easily proves that constraints (34), (35), (39), (40), (43) and (44) may all be removed. In other words, when enforcing non-anticipativity based on a pair of scenarios  $r, s$ , we can choose arbitrarily whether to use  $x^r$  or  $x^s$  on the right-hand side, and do not need to include constraints for both cases. The proof is as follows.

**Proposition 3.6** *SPM-MIP is equivalent to:*

$$\max \left\{ \sum_{t=1}^T \sum_{i=1}^N \sum_{s=1}^S p(s) \left( (c_t^1 a_{i,s}^1 - c_t^{proc} a_{i,s}^0) z_{i,t}^s - a_{i,s}^0 c_t^{mng} y_{i,t}^s \right) : (21) - (26), (28) - (30), \right. \\ \left. (31) - (33), (36) - (38), (41) \text{ and } (42) \right\}.$$

**Proof.** Suppose  $(x, y, z)$  is a feasible point for the above formulation. Since  $|x_{i,t}^s - x_{i,t}^r| \leq 1$  (as  $x$  binary),  $x$  satisfies (34) and (35) if  $|x_{i,t}^s - x_{i,t}^r| \leq \mathbf{Ind}(\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^r)$  for all  $t = 2, \dots, T$ . But the right-hand side of this constraint is equal to  $\mathbf{Ind}(\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s)$ , by Lemma 3.5, and the constraint is satisfied since  $x$  satisfies (32) and (33). The same argument shows that  $y$  satisfies (39) and (40), and that  $z$  satisfies (43) and (44), since all elements in both variables lie between zero and one, and since  $0 \leq \xi \leq 1$  implies  $|\xi_{i,t}^s - \xi_{i,t}^r| \leq 1$ , where we may take  $\xi = y$  or  $\xi = z$ .  $\blacksquare$

This result could also be observed from Theorem 1 of Goel and Grossmann (2006), and by understanding the relationship between their disjunctive formulation and our linear one. We provide the above direct proof for the convenience of the reader.



As for the case where only processing decisions are scenario-dependent, we are able to show in this case that if  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving, (and rock attribute values are scenario-independent), then all of the non-anticipativity constraints associated with the processing constraints can be omitted, i.e. from any optimal solution for SPM-MIP *without* constraints (41) and (42), which we refer to as SPM-MIP-red, we can construct an optimal solution for SPM-MIP.

**Proposition 3.7** *Assume that  $a_{i,r}^0 = a_{i,s}^0$  for all  $r, s \in \mathcal{S}$ ,  $r \neq s$ , and all  $i \in E_{\{r,s\}}^\alpha$ , and that  $\mathcal{S}$  and  $\alpha$  are grade-order-preserving. Then given any optimal solution to SPM-MIP-red, we can construct an optimal solution for SPM-MIP.*

We omit the proof here, as it is very similar to that of Proposition 3.3. Here we simply note that the key difference between the two proofs revolves around the upper bounds on the  $z_{(\cdot),t}^s$  variables in the  $t, s$ -processing-only problem. In the earlier case, these bounds were clearly scenario-independent (being given by  $\hat{y}_{(\cdot),t}$ ). Here they will be given by  $\hat{y}_{(\cdot),t}^s$ , where  $(\hat{x}, \hat{y}, \hat{z})$  is an optimal solution to SPM-MIP-red. If  $t = 1$  then by (36) they will still be scenario-independent. For  $t \geq 2$ , for each  $\{r, s\} \in \mathcal{E}_t$ , (defined in the proof of Proposition 3.3 to indicate the pairs of scenarios on which the  $z$  variables must satisfy non-anticipativity, here taken similarly to be  $\mathcal{E}_t = \{\{r, s\} \subseteq \mathcal{S} : r \neq s, \sum_{j \in \Delta_{\{r,s\}}^\alpha} \hat{x}_{j,t}^s = 0\}$ , which is well defined by Lemma 3.5), we know that  $\sum_{j \in \Delta_{\{r,s\}}^\alpha} \hat{x}_{j,t}^s = 0$ , so by (26) it must be that  $\sum_{j \in \Delta_{\{r,s\}}^\alpha} \hat{x}_{j,t-1}^s = 0$ , and hence  $\hat{y}_{i,t}^s = \hat{y}_{i,t}^r$  for all  $i \in \{1, \dots, N\}$  by (37) and (38) (and Proposition 3.6). Arguments in the proof of Proposition 3.3 show that we can find common sortings of the aggregates so that for every pair  $\{r, s\} \in \mathcal{E}_t$ , the same sorting applies and sorts both objective-to-knapsack-coefficient ratios in non-decreasing order. We have just argued that variable upper bounds in both the  $t, r$ -processing-only and  $t, s$ -processing-only problems will be identical. As in the proof of Proposition 3.3, the knapsack coefficients are also scenario-independent, except on variables forced to zero, so the common sorting must induce identical solutions  $z_{(\cdot),t}^s = z_{(\cdot),t}^r$ , thus satisfying non-anticipativity. Even if the grade-order preserving property is not satisfied, the above reasoning suggest that initially ignoring non-anticipativity for the processing variables, and then later re-solving with the other variables fixed, could be a good heuristic, similar to SP-MIP-h. We will refer to this heuristic as SPM-MIP-h.

In the case that the grade-ordering-preserving property does not apply, we can still omit some of the non-anticipativity constraints for the  $z$  variables: they can be omitted if the differentiator set for an aggregate and pair of scenarios has any intersection with the precedence set of the aggregate. In the following proposition, we show that we can do this in addition to halving the number of non-anticipativity constraints via Proposition 3.6.

**Proposition 3.8** *Let  $(x, y, z)$  satisfy (21), (24), (25), (26), (28)-(30), (31)-(33), and (36)-(38). If  $x$  and  $z$  also satisfy (41) and (42) for all  $i \in \{1, \dots, N\}$ ,  $r, s \in \{1, \dots, S\}, r \neq s$ , for which  $\mathcal{P}(i) \cap \Delta_{\{r,s\}}^\alpha = \emptyset$ , for all  $t \in \{1, \dots, T\}$ , then  $(x, y, z) \in \mathcal{N}^{proc-min}$ .*

**Proof.** Let  $(x, y, z)$  satisfy the constraints indicated. Then by Lemma 3.5, and the proof of Proposition 3.6,  $x$  and  $y$  must also satisfy (34)-(35) and (39)-(40). Furthermore the same arguments show that  $x$  and  $z$  must satisfy (43) and (44) for  $i \in \{1, \dots, N\}$ ,  $r, s \in \{1, \dots, S\}, r \neq s$ , with  $\mathcal{P}(i) \cap \Delta_{\{r,s\}}^\alpha = \emptyset$ , for all  $t \in \{1, \dots, T\}$ .

Now consider the situation when for some  $i \in \{1, \dots, N\}$ ,  $r, s \in \{1, \dots, S\}, r \neq s$ , we have  $\mathcal{P}(i) \cap \Delta_{\{r,s\}}^\alpha \neq \emptyset$ . If  $|z_{i,t}^r - z_{i,t}^s| = 0$ , for any  $t \in \{1, \dots, T\}$ , then (41)-(44) are naturally satisfied. If for some  $t \in \{1, \dots, T\}$ , we have that  $|z_{i,t}^r - z_{i,t}^s| \neq 0$ , then there are two cases.

Case 1:  $z_{i,t}^r > z_{i,t}^s$ . In this case (41) and (43) are naturally satisfied. Also  $z_{i,t}^r > 0$ , and so  $y_{i,t}^r \geq z_{i,t}^r > 0$  by (21). Hence  $x_{i,t}^r = 1$  by (25) and (28). Thus by (24) it must be that for all  $j \in \mathcal{P}(i)$ , we have  $\sum_{\hat{t}=1}^t y_{j,\hat{t}}^r \geq 1$ , and hence via (25) and (28) again we have  $x_{j,t}^r = 1$  for all  $j \in \mathcal{P}(i)$ . Since  $\mathcal{P}(i) \cap \Delta_{\{r,s\}}^\alpha \neq \emptyset$ , this means there must exist  $k \in \Delta_{\{r,s\}}^\alpha$  with  $x_{k,t}^r = 1$ , and so  $x$  and  $z$  must satisfy (44). If  $x_{k,t}^s = 1$ , then (42) is also satisfied. Otherwise,  $x_{k,t}^s = 0$ , and by (33) it must be that  $\sum_{j \in \Delta_{\{r,s\}}^\alpha} x_{j,t-1}^s \geq 1$ . So for some  $j \in \Delta_{\{r,s\}}^\alpha$ ,  $x_{j,t-1}^s = 1$ , and by (26), it must be  $x_{j,t}^s = 1$ , and so (42) is still satisfied.

Case 2:  $z_{i,t}^r < z_{i,t}^s$ . This case is similar, but appeals to (34) instead of (33). ■

Note that non-anticipativity constraints for the  $x$  and  $y$  variables cannot be eliminated in a similar way, due to the learning associated with the mining decisions being conservative.

### 3.3 Further Elimination of Non-Anticipativity Constraints and Links to General Multistage Stochastic Programming with Endogenous Uncertainty

As discussed in our introduction, the work of Goel and Grossmann (2006) represents the state of the art in general multistage stochastic programming with endogenous uncertainty. Later papers, such as Goel et al. (2006), Tarhan and Grossmann (2008), are applications based on the same underlying theory given in Goel and Grossmann (2006).

Here we mention a couple of ways in which we believe our approach offers an alternative to that of Goel and Grossmann (2006). Since their work is very general, and so requires a lot of complex notation, we try as far as possible to work within our own notation, while still conveying the implications for the general setting. We also simplify where possible, for ease of presentation. For example, we discuss their work without reference to exogenous uncertainty, which they also include. We first note key notation and concepts in their general formulation, and link them to ours.

Goel and Grossmann (2006) have binary variables  $b_{p,t}^s$  for each scenario  $s$ , each period or stage  $t$ , and for each  $p \in \mathcal{I}$  indexing the sources of endogenous uncertainty represented by (discrete-valued) random variables  $\theta_p$ . The vector  $b_{p,(\cdot)}^s$  is an indicator vector, indicating the period in which  $\theta_p$  is realized, under scenario  $s$ . The set of possible realizations for  $\theta_p$  is given by  $\Theta_p$ . They also have other variables, which could be real or integer; here we simply use a general variable vector  $\chi_t^s \in \mathbb{R}^L$  for each scenario  $s$  and period  $t$  to represent other decisions that are affected by the endogenous uncertainty. (Here we use  $\chi$  in place of Goel and Grossman's  $x$  and  $y$  variables, which we avoid to prevent confusion with our own variables of the same name. Note that their  $y$  variables correspond to conservative decisions, and their  $x$  variables to non-conservative; here for simplicity we take all  $\chi$  decisions to be non-conservative.) Goel and Grossman also use a differentiator, which they call  $\mathcal{D}(r, s) \subseteq \mathcal{I}$ , indicating the sources of endogenous uncertainty which allow distinction between the two scenarios. (A scenario  $s$  is characterized by a vector of realizations  $(\theta_p^s)_{p \in \mathcal{I}}$ .) So our aggregates  $i \in \{1, \dots, N\}$  play the role of their sources of uncertainty,  $p \in \mathcal{I}$ , and our uncertain mineral attribute  $a_{i,s}^1$  takes the role of their uncertain parameters  $\theta_p^s$ . Our source of uncertainty in aggregate  $i$  is resolved under scenario  $s$  when our  $x_{i,t}^s - x_{i,t-1}^s = 1$ ; our vector  $(x_{i,t}^s - x_{i,t-1}^s)_{t=1}^T$  (taking  $x_{i,0}^s = 0$ ) takes the role of their indicator vector  $b_{p,(\cdot)}^s$ .

The way Goel and Grossmann (2006) handle non-anticipativity in their general case is to introduce a boolean  $Z_t^{r,s}$  for each pair of scenarios, which is interpreted as *True* if and only if  $r$  and  $s$  have *not* been distinguished by period  $t$  and formulate the non-anticipativity logic using disjunctions

(equivalent to) the following:

$$\neg Z_t^{r,s} \vee (b_{p,t+1}^s = b_{p,t+1}^r), \quad \forall p \in \mathcal{I}, r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T-1\} \quad (46)$$

$$\neg Z_t^{r,s} \vee (\chi_t^s = \chi_t^r), \quad \forall r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T\} \quad (47)$$

$$Z_t^{r,s} \Leftrightarrow \bigwedge_{p \in \mathcal{D}(s,r)} \left( \bigwedge_{\tau=1}^t \neg b_{p,\tau}^s \right) \quad \forall r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T\} \quad (48)$$

In their initial formulation they also include

$$Z_t^{r,s} \Leftrightarrow \bigwedge_{p \in \mathcal{D}(s,r)} \left( \bigwedge_{\tau=1}^t \neg b_{p,\tau}^r \right), \quad \forall r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T\},$$

but, in an analagous conclusion to that we reach in our Proposition 3.6, deduce that this can be omitted.

Goel and Grossmann (2006) mention modelling these constraints linearly, which they do via the introduction of variables  $\xi_t^{r,s}$ , ( $\xi$  is  $z$  in their notation, see page 370), taking the role of the boolean  $Z_t^{r,s}$ . But of course this leads to  $\mathcal{O}(TS^2)$  variables. Furthermore, to model (48), they ask that

$$\xi_t^{r,s} \leq 1 - b_{p,\tau}^s, \quad \forall t \in \{1, \dots, T\}, \tau \in \{1, \dots, t\}, p \in \mathcal{D}(r, s), \quad (49)$$

which leads to  $\mathcal{O}(T^2S^2)$  constraints. This is of course very unwieldy, so they do not consider solving such a formulation with standard mixed integer programming techniques. Instead they develop a Lagrangean relaxation-based branch-and-bound method. Their Lagrangean relaxation relaxes the disjunctions (48) entirely, and penalizes the non-anticipativity constraints (46) and (47) via Lagrangean dual variables in the objective. The resulting Lagrangean subproblem decomposes nicely, but the Lagrangean dual still has  $\mathcal{O}(TS^2)$  dual variables. This makes the dual quite difficult to solve; in Goel et al. (2006), for example, subgradient optimization is used.

Our approach suggests an alternative. First, we note that, like our  $x$  variables, it would be helpful to use cumulative variables, rather than the indicator vector  $b$ , i.e. we suggest using

$$\hat{b}_{p,t}^s = \text{Ind} \left( \sum_{\tau=1}^t b_{p,\tau}^s \right), \quad \forall s \in \mathcal{S}, p \in \mathcal{I}, t \in \{1, \dots, T\},$$

instead of  $b_{p,t}^s$ . (We also note that allowing the  $b_{p,(\cdot)}^s$  vector to indicate more than one period seems unhelpful – periods after the first add no new information – and the cumulative variables capture only the first period, as needed.) This approach has been effective in the mining context, (see, for example, Caccetta and Hill (2003)), where branching on these cumulative variables leads to more balanced trees in branch-and-bound. Furthermore, it allows, for example, (49) to be re-modelled via

$$\xi_t^{r,s} \leq 1 - \hat{b}_{p,t}^s, \quad \forall t \in \{1, \dots, T\}, p \in \mathcal{D}(r, s),$$

reducing the number of such constraints by order a factor of  $S$ . (The correct logic is assured as we of course also require that  $\hat{b}_{p,t}^s \leq \hat{b}_{p,t+1}^s$  for all  $s \in \mathcal{S}, p \in \mathcal{I}$ , and  $t \in \{1, \dots, T-1\}$ .)

Furthermore, we observe that the booleans  $Z$  and additional variables  $\xi$  are actually unnecessary: the non-anticipativity constraints (46) and (47) can be modelled directly in a manner analagous to

our SPM-MIP via

$$\hat{b}_{p,t+1}^s - \hat{b}_{p,t+1}^r \leq \sum_{j \in \mathcal{D}(r,s)} \hat{b}_{j,t}^s, \quad \forall p \in \mathcal{I}, r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T-1\}, \quad (50)$$

$$\hat{b}_{p,t+1}^r - \hat{b}_{p,t+1}^s \leq \sum_{j \in \mathcal{D}(r,s)} \hat{b}_{j,t}^s, \quad \forall p \in \mathcal{I}, r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T-1\}, \quad (51)$$

$$\chi_{l,t}^s - \chi_{l,t}^r \leq M_{l,t}^{r,s} \sum_{j \in \mathcal{D}(r,s)} \hat{b}_{j,t}^s, \quad \forall l \in \{1, \dots, L\}, r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T\}, \text{ and} \quad (52)$$

$$\chi_{l,t}^r - \chi_{l,t}^s \leq M_{l,t}^{r,s} \sum_{j \in \mathcal{D}(r,s)} \hat{b}_{j,t}^s, \quad \forall l \in \{1, \dots, L\}, r, s \in \mathcal{S}, r \neq s, t \in \{1, \dots, T\}. \quad (53)$$

Here (52) and (53) are “big-M” constraints, and  $M_{l,t}^{r,s}$  can be taken to be any upper bound known on the value of  $|\chi_{l,t}^s - \chi_{l,t}^r|$ . In our mining models, all our “big-M”s could be taken to be one, as all our variables lay between zero and one.

We suggest that the above approach, solved with standard branch-and-bound techniques, could offer a viable alternative to the approach of Goel and Grossmann (2006). In cases with a very large number of the non-anticipativity constraints (50)–(53), these could be handled as “lazy” constraints. Our results in Section 4, in our mining context, show that lazy constraints were not helpful; the model with all constraints included a priori solved quite effectively. In general, the need for “big-M” constraints may weaken the formulation, however modern solvers are adept at handling such issues. The  $\mathcal{O}(TS^2)$  can, as mentioned, be addressed by treating them as “lazy” constraints. However in special cases they may be able to be further reduced. Indeed, in the special case that the scenario set is the entire set of all cross products of realizations of the uncertain parameters, i.e.  $\{\theta^s : s \in \mathcal{S}\} = \times_{p \in \mathcal{I}} \Theta_p$ , Goel and Grossmann (2006) are able to prove (in their Theorem 2) that only constraints corresponding to pairs  $r, s \in \mathcal{S}$  with  $|\mathcal{D}(r, s)| = 1$  are needed. This very substantially reduces the number of such constraints, and may well make our approach quite attractive.

Though Theorem 2 of Goel and Grossmann (2006) is very powerful in reducing the number of non-anticipativity constraints that are needed, it does require a quite special condition on the scenario set, which will in general not be satisfied. It is certainly not satisfied in our mining case, where scenarios are not constructed via a scenario tree, and are sampled from a continuous, not discrete, distribution. Even if not, for mining problems the number of sources of uncertainty (the number of aggregates) is typically very large, so if all cross products were constructed, the number of scenarios would become enormous. In the remainder of this section, we go to the heart of Goel and Grossmann’s result, to show how the same ideas can be used to reduce the number of non-anticipativity constraints needed, but without the strong requirement on the scenario set. Our result is summarized as follows, where first we require a key definition. In all that follows we use  $\mathcal{T} = \{\{r, s\} \subseteq \mathcal{S} : r \neq s\}$  to denote the set of all scenario pairs.

**Definition 3.9** *We say  $\mathcal{G} \subseteq \mathcal{T}$  is a generator of  $\mathcal{T}$  if for all  $\{r, s\} \in \mathcal{T}$ , there exist  $q^1, q^2, \dots, q^U \in \mathcal{S}$  with  $r = q^1$ ,  $s = q^U$ ,  $\{q^m, q^{m+1}\} \in \mathcal{G}$  for all  $m = 1, \dots, U-1$ , and*

$$\mathcal{D}(r, s) \supseteq \bigcup_{m=1, \dots, U-1} \mathcal{D}(q^m, q^{m+1}). \quad (54)$$

We show that it suffices to consider non-anticipativity constraints for scenario pairs in  $\mathcal{G}$ . To do so, we may take  $t$  to be fixed, and consider a single generic variable  $\phi^s$  to take the role of variables

$\hat{b}_{p,t+1}^s$  or  $\chi_{l,t}^s$  on the left-hand side of the constraints, and use binary  $\beta^s$  on the right-hand side. In other words, we view our prototypical non-anticipativity constraint as (in logical form):

$$\sum_{p \in \mathcal{D}(r,s)} \beta_p^s = 0 \Rightarrow \phi^s = \phi^r, \quad \forall \{r, s\} \in \mathcal{T}. \quad (55)$$

Clearly all non-anticipativity constraints (50)–(53) can be obtained from this form via the natural linearization.

**Theorem 3.10** *For  $\mathcal{G}$  a generator of  $\mathcal{T}$ , we have that if  $(\beta_p^s)_{\substack{p \in \mathcal{I} \\ s \in \mathcal{S}}} \geq 0$  and  $(\phi^s)_{s \in \mathcal{S}}$  satisfy*

$$\sum_{p \in \mathcal{D}(r,s)} \beta_p^s = 0 \Rightarrow \phi^s = \phi^r, \quad \forall \{r, s\} \in \mathcal{G}, \quad (56)$$

then they also satisfy (55).

**Proof.** Assume  $(\beta_p^s)_{\substack{p \in \mathcal{I} \\ s \in \mathcal{S}}} \geq 0$  and  $(\phi^s)_{s \in \mathcal{S}}$  satisfy (56) and consider  $\{r, s\} \in \mathcal{T} \setminus \mathcal{G}$ . Suppose that  $\sum_{p \in \mathcal{D}(r,s)} \beta_p^s = 0$ . Then since  $(\beta_p^s)_{\substack{p \in \mathcal{I} \\ s \in \mathcal{S}}} \geq 0$  it must be that  $\beta_p^s = 0$  for all  $p \in \mathcal{D}(r, s)$ . Now by the definition of a generator, there must exist  $q^1, q^2, \dots, q^U \in \mathcal{S}$  with  $q^1 = r, q^U = s$ , such that  $\{q^m, q^{m+1}\} \in \mathcal{G}$  and  $\mathcal{D}(q^m, q^{m+1}) \subseteq \mathcal{D}(r, s)$ , for all  $m = 1, \dots, U - 1$ . Thus  $\beta_p^s = 0$  for all  $p \in \mathcal{D}(q^m, q^{m+1})$  and all  $m = 1, \dots, U - 1$ , and hence  $\sum_{p \in \mathcal{D}(q^m, q^{m+1})} \beta_p^s = 0$  for all  $m = 1, \dots, U - 1$ . By (56) we deduce that  $\phi^{q^m} = \phi^{q^{m+1}}$  for all  $m = 1, \dots, U - 1$ . Hence  $\phi^r = \phi^{q^1} = \phi^{q^2} = \dots = \phi^{q^U} = \phi^s$ , and the non-anticipativity constraint for  $\{r, s\}$  is satisfied as required.  $\blacksquare$

It is not hard to see that Goel and Grossmann's Theorem 2 in Goel and Grossmann (2006) follows from this result as a corollary. When  $\{\theta^s : s \in \mathcal{S}\} = \times_{p \in \mathcal{I}} \Theta_p$ , the set  $\mathcal{L}^1 = \{\{r, s\} \subseteq \mathcal{S} : |\mathcal{D}(r, s)| = 1\}$  is clearly a generator for  $\mathcal{T}$ . To see why, consider  $\{r, s\} \in \mathcal{T}$  with  $U' \stackrel{\text{def}}{=} |\mathcal{D}(r, s)| \geq 2$  (note that for simplicity we have ignored exogeneous uncertainty and can assume that all scenario pairs must have a non-empty differentiator set). We can without loss of generality assume  $\mathcal{D}(r, s) = \{1, \dots, U'\}$ , so  $\theta^s = \theta^r + \sum_{p=1}^{U'} (\theta_p^s - \theta_p^r) e^p$ , where  $e_p$  is the  $p$ th standard basis vector in  $\mathbb{R}^{|\mathcal{I}|}$ . Take for each  $m = 1, \dots, U'$ , take  $q^{m+1} \in \mathcal{S}$  to be the scenario index such that  $\theta^{q^{m+1}} = \theta^r + \sum_{p=1}^m (\theta_p^s - \theta_p^r) e^p$ . Such an index must exist since  $\{\theta^s : s \in \mathcal{S}\} = \times_{p \in \mathcal{I}} \Theta_p$ , and each element  $\theta^{q^{m+1}} \in \{\theta_p^r, \theta_p^s\} \subseteq \Theta_p$  for each  $p \in \mathcal{I}$ . Set  $q^1 = r$ . Taking  $U = U' + 1$  we have  $\theta^{q^U} = \theta^s$ , so  $q^U = s$ . Furthermore  $\theta^{q^{m+1}} = \theta^{q^m} + (\theta_m^s - \theta_m^r) e^m$ , so  $\mathcal{D}(q^m, q^{m+1}) = \{m\}$  for all  $m = 1, \dots, U'$ . Thus  $\{q^m, q^{m+1}\}$  is in  $\mathcal{L}^1$ , and furthermore  $\mathcal{D}(r, s) = \{1, \dots, U'\} = \bigcup_{m=1, \dots, U'} \{m\} = \bigcup_{m=1, \dots, U'} \mathcal{D}(q^m, q^{m+1})$ . Thus  $\mathcal{L}^1$  is a generator of  $\mathcal{T}$ .

So how can we find generators in general? Naturally we would seek a minimal generator; ideally it would be one of minimum cardinality. For any set  $\mathcal{F} \subseteq \mathcal{T}$  define  $\text{Graph}_{\mathcal{F}}(\{r, s\})$  to be the graph with node set  $\mathcal{S}$  and arcs  $\{\{r', s'\} \in \mathcal{F} : \mathcal{D}(r', s') \subseteq \mathcal{D}(r, s)\}$ . Then it is not hard to see that  $\mathcal{F}$  is a generator for  $\mathcal{T}$  if and only if for every  $\{r, s\} \in \mathcal{T}$  there exists a path between  $r$  and  $s$  in  $\text{Graph}_{\mathcal{F}}(\{r, s\})$ . This suggests the following naive algorithm for finding a minimal generator.

**Algorithm 3.1** *Minimal Generator Algorithm*

Initialize  $\mathcal{F} = \mathcal{T}$   
**for** each scenario pair  $\{r, s\} \in \mathcal{T}$  **do**

```

/* consider removing {r, s} */
Set can := True
for each scenario pair  $\{r', s'\} \in \mathcal{T} \setminus (\mathcal{F} \setminus \{\{r, s\}\})$  do
    if there is no path between  $r'$  and  $s'$  in  $\text{Graph}_{\mathcal{F} \setminus \{\{r, s\}\}}(\{r', s'\})$  then
        Set can := False
        Break
    endif
endfor
if can = True then set  $\mathcal{F} := \mathcal{F} \setminus \{\{r, s\}\}$ 
enddo
Return minimal generator  $\mathcal{F}$ 

```

Of course, this algorithm could be quite computationally expensive, and even with a good search order for considering scenario pairs in  $\mathcal{T}$  to remove from  $\mathcal{F}$ , is unlikely, in general, to give a generator with minimum cardinality. We give it here primarily to show that it is possible, and to show that even for general scenario sets, it may be possible to reduce the set of scenario pairs for which non-anticipativity constraints must be enforced. We reserve further study of minimal generators for further work.

For the ten scenario sets we consider in the next section, all have  $|\mathcal{T}| = 10$ . We determined their minimal generators by inspection. In nine cases, there was a unique minimal generator. Of these, one had cardinality 10, (no reduction was possible at all), three had cardinality 9, four had cardinality 8, and one had cardinality 7. In the remaining case there were two, one of cardinality 8, the other of 9. Since the reductions in number of scenario pairs was modest for this data, we did not attempt to automate the process; the numerical results in the next section are taken from models that enforce non-anticipativity over all 10 pairs.

However we believe that in general, for problems with larger numbers of scenarios, the minimal generator concept may be useful.

## 4 Numerical Results

In this section we describe a series of numerical experiments to demonstrate the benefit of explicitly incorporating scenario dependent decisions in the optimization process.

We first mention that we used the preprocessing of Boland et al. (2007) in which some variables are fixed before the model is solved. The preprocessing is based on the idea that any aggregate  $i$  may be extracted in a period  $t$  only if all aggregates that need to be extracted before  $i$ , according to the precedence relationships, are extracted in time period  $t$  or earlier. If the mining capacity required to achieve this exceeds the cumulative mining capacity available up to and including time period  $t$ , aggregate  $i$  cannot be extracted in time period  $t$  or earlier. We apply this under each scenario  $s = 1, \dots, S$ , and calculate the mining capacity required according to the  $a_{i,s}^0$  values, yielding  $x_{i,t}^s = 0$ , for all  $t' = 1, \dots, t$ , and for all  $t \in \{1, \dots, T\}$  in which the cumulative available mining capacity is exceeded.

We test our algorithms using realistic mining data. The block model we use was supplied by our industry partner BHP Billiton Pty Ltd. It includes the aggregates and precedence relationships between aggregates, the attributes for each block, and costs and revenues per tonnes mined, processed, and sold. It has 1643 blocks and 115 aggregates. We take our planning problem to run over ten time periods, each period being two years long. From this block model, we constructed 50 scenarios, each with the same total metal tonnage as the original block model, but possibly different metal

attribute values for each block. (All other data, in particular the block rock attribute values, are identical across scenarios.) Our scenarios are generated so that blocks in the upper layers (higher in the mine) have similar attribute values across scenarios, while the lower layers show much greater variation between scenarios. We create 10 instances containing 5 scenarios each, by taking the first 5 scenarios that were generated, then the next 5 and so on, until every scenario belongs to exactly one instance. Each such instance represents a Stochastic OPMPSP data instance, so we have a data set consisting of 10 instances, each with 5 scenarios. We consider the probabilities associated with the scenarios to be equal. In all cases differentiators are calculated using  $\alpha = 0.1$ , and we do not use the grade-order-preserving property.

We solve these instances using ILOG CPLEX 9.1 on an Intel(R) Core(TM)2 CPU 6600 at 2.4GHz, with 3.5GB of RAM, with optimality gap set to 1%, and we use CPLEX default parameter settings.

In order to provide a full comparison for all algorithms considered in this paper, we first investigate the case in which both the processing and the mining decisions are stochastic. We consider two main models for the SOPMPSP. Firstly, we consider the model with the full set of non-anticipativity constraints, i.e. (31)-(44), and we refer to it as SPM-MIP. We then solve the model obtained, using CPLEX with default parameters, but we also solve it by declaring all non-anticipativity constraints as CPLEX lazy constraints. We provide numerical results for both methods. Secondly, we consider the model with the set of non-anticipativity constraints halved according to Proposition 3.6. As shown in Proposition 3.6, for any pair of scenarios  $r$  and  $s$ ,  $r, s \in \{1, \dots, S\}, r \neq s$ , only half of the non-anticipativity constraints need to be added to the model. In other words, it is enough to use either (32)-(33) or (34)-(35) for the  $x$  variables, either (37)-(38) or (39)-(40) for the  $y$  variables, and either (41)-(42) or (43)-(44) for the  $z$  variables. The question that arises is how do we choose which non-anticipativity constraints to keep and which ones to eliminate as redundant? In our experiments, we test two heuristics. In the first one, we keep the non-anticipativity constraints for which the right-hand-side of the inequalities may have the best chance to be the smallest. For any given pair of scenarios  $r$  and  $s$ ,  $r \neq s$ , we calculate the total grade of the aggregates in  $\Delta_{r,s}^\alpha$  under each scenario. If the total grade is smaller for scenario  $r$  than for scenario  $s$ , then the non-anticipativity constraints that we use are (34)-(35), (39)-(40), and (43)-(44); otherwise we use (32)-(33), (37)-(38), and (41)-(42). We refer to the model that uses this selection heuristic as SPM-MIP-half-min. The second heuristic is the reverse of the first one: we use the non-anticipativity constraints (34)-(35), (39)-(40), and (43)-(44) if the total grade of the blocks in the differentiator is *greater* for scenario  $r$  than for scenario  $s$ , and we use (32)-(33), (37)-(38), and (41)-(42) otherwise. We refer to the model that uses this selection heuristic as SPM-MIP-half-max. Thirdly, we investigate the effect of eliminating redundant  $z$ -variable non-anticipativity constraints with differentiator/precedence set intersection. We call the model with such constraints removed SPM-MIP-elim. We also test the removal of such constraints from the model with halved non-anticipativity constraints, according to Proposition 3.8. However, we only investigate the effect of elimination on the best algorithm for which the halving was used. From Table 1 it can be seen that the best algorithm is SPM-MIP-half-max. We denote by SPM-MIP-half-max-elim the model given by SPM-MIP-half-max with redundant non-anticipativity constraints for the  $z$  variables eliminated via Proposition 3.8.

From the results in Table 1 it can be seen that the running time needed when the non-anticipativity constraints were added to SPM-MIP as lazy constraints was much longer than when the full model was solved with default parameters. While we may think that adding the non-anticipativity constraints only when needed would help the solution of the problem, this is not the case in practice. Therefore we did not further test this option.

From Table 1 it can be also be seen that, for most instances, the running time was greatly improved by halving the number of non-anticipativity constraints. For Instance 2, however both SPM-MIP-

half-min and SPM-MIP-half-max were slower than SPM-MIP (by 7.73% and 2.23% respectively), while SPM-MIP-half-min was also 22.5% slower than SPM-MIP in the case of Instance 7. On the remaining instances the average speed-up was 27.48% for SPM-MIP-half-min (calculated over 8 instances) and 33.83% for SPM-MIP-half-max (calculated over 9 instances). We can therefore claim that out of the four methods tested for solving OPMPSP with stochastic mining and processing decisions, the best performance was recorded by SPM-MIP-half-max.

From Table 1 we see that the elimination of redundant non-anticipativity constraints produces mixed results. When SPM-MIP and SPM-MIP-elim are compared, one can see that for four instances the running time needed was longer (on average by 45.93%), while for the remaining six instances the running time was reduced (on average by 19.20%). Also, in most cases, the results for SPM-MIP-elim are worse than those obtained by simply halving the non-anticipativity constraints (SPM-MIP-half-min and SPM-MIP-half-max). Even when the elimination of the redundant constraints is applied to the best algorithm, SPM-MIP-half-max, no significant and consistent improvement is recorded. The elimination reduced the number of non-anticipativity constraints in SPM-MIP-half-max by over 21% on average. In spite of that, it can be seen that the performance worsened in 60% of the cases (by 24.01% on average), while the improvement recorded in the case of the four remaining instances was only 14.19% on average. It therefore appears that the redundant constraints make the problem easier to solve and we choose to keep them in the model.

Since we wish to provide a comparison between algorithms that use the same techniques as far as the elimination of non-anticipativity constraints is concerned, we will not use elimination of redundant constraints for SP-MIP either. In the rest of this section we use SPM-MIP-half-max for our numerical comparisons and we also use the scenario selection of SPM-MIP-half-max to test the heuristic SPM-MIP-h, which will also only use half of the non-anticipativity constraints.

Next, we investigate the results obtained by our heuristics SP-MIP-h and SPM-MIP-h. From Table 2 we see that, as expected, the optimal values obtained by SP-MIP-h are very close to those obtained after running SP-MIP (two exceptions are Instance 1 and Instance 3). For nine instances, the running time needed by SP-MIP-h decreased on average by 35%, with instance time improvements ranging from 12.68% (Instance 2) to 50.76% (Instance 8). The only instance for which the running time of SP-MIP-h was slower was Instance 3, for which the time needed almost doubled compared to the time needed by SP-MIP. In the case of running times of SPM-MIP-h and SPM-MIP-half-max the situation is reversed: for eight instances the time needed by the heuristic was significantly longer (on average by 49.43%), with only two instances for which a speed-up was recorded. We can therefore say that the non-anticipativity constraints for  $z$  speed up the model when the mining and the processing decisions are stochastic, (recall that SPM-MIP-h first drops those constraints), but not when only processing decisions are stochastic.

In order to measure the effects of planning with knowledge of multiple geological scenarios, we consider two benchmarks. The first reverts to the current standard approach using a single geological estimate, which we take to be defined by the average (expected) attribute values over all scenarios. We call this the *Base Algorithm*, which is simply Step (i) of the SP-MIP-heur heuristic, (Algorithm 4.1), given below. Clearly this provides a feasible solution for both SP-MIP and SPM-MIP, (assuming that the rock attribute values are scenario-independent, which is the case for our mining data), and its objective value is a lower bound on that of both these problems. We then use the usual “perfect information” solution to give an upper bound. We note that in general one hopes to do this simply by solving the deterministic problem, in this case D-MIP, for each scenario independently, and then taking the expected value of the resulting objective values, i.e. for each  $s \in \mathcal{S}$ , solve D-MIP with  $a^0 = a^0_{(\cdot),s}$  and  $a^1 = a^1_{(\cdot),s}$  (call this problem D-MIP<sup>s</sup>) to obtain optimal value  $w^s$ , and calculate the objective value under the assumption of perfect information as  $w^{PI} = \sum_{s \in \mathcal{S}} p(s)w^s$ . In theory, this



should be equivalent to solving SPM-MIP with all non-anticipativity constraints relaxed, (call this PI-MIP), and more computationally tractable. However, solving mixed integer programs such as D-MIP for realistic mining data to a high degree of accuracy can be very time consuming, so we usually impose a tolerance, namely the value of the *gap* between lower and upper bound at which the branch-and-bound may terminate, and report the gap for the final solution found. Fortunately, it is not hard to show that if we solve each D-MIP<sup>s</sup> to within a gap of  $\delta^s > 0$  and  $\delta^s \leq \delta$  for all  $s \in \mathcal{S}$ , then the expected value of  $\{w^s : s \in \mathcal{S}\}$  is within a gap of  $\delta > 0$  of optimal for PI-MIP. Each D-MIP<sup>s</sup> terminates having produced a lower bound  $L^s$  and an upper bound  $U^s$  on its value, with  $\delta^s \geq (U^s - L^s)/L^s$ , (as long as all lower bounds are positive, which is true for typical mining data). Now  $L^{PI} \stackrel{def}{=} \sum_{s \in \mathcal{S}} p(s)L^s$  and  $U^{PI} \stackrel{def}{=} \sum_{s \in \mathcal{S}} p(s)U^s$  must be upper and lower bounds on PI-MIP, since PI-MIP decomposes precisely into the sum of the D-MIP<sup>s</sup> problems (weighted by  $p(s)$ ) over  $s \in \mathcal{S}$ . Now

$$U^{PI} - L^{PI} = \sum_{s \in \mathcal{S}} p(s)(U^s - L^s) \leq \sum_{s \in \mathcal{S}} p(s)\delta^s L^s \leq \sum_{s \in \mathcal{S}} p(s)\delta L^s = \delta L^{PI},$$

as required. Note that of course  $w^{PI}$  is unlikely to be achievable by the stochastic model SPM-MIP; it indicates only what could be achieved with complete *a priori* knowledge of the scenarios.

The Base Algorithm gives the result of ignoring multiple scenarios altogether, while SP-MIP allows processing decisions to make use of knowledge obtained during mining. Although SP-MIP still constructs a single mining schedule, to be used irrespective of scenario, it nevertheless does construct the schedule so that scenario knowledge may be exploited in processing. We ask how important is that, and how much do we gain in expected NPV through the use of such a schedule? To test this, we take the mining schedule produced by the base algorithm as a heuristic solution for SP-MIP, and investigate how far from optimal for SP-MIP it is. More formally, the heuristic we suggest is as follows.

**Algorithm 4.1** *SP-MIP-heur*

- Step (i): Calculate the expected value of the scenario attributes values to produce  $\bar{a}^0$  and  $\bar{a}^1$ .  
Solve D-MIP with attribute coefficients  $a^0 = \bar{a}^0$  and  $a^1 = \bar{a}^1$ .  
Store  $(\bar{x}, \bar{y}, \bar{z})$ , the optimal solution obtained, and  $w^{Ave}$ , the optimal value.
- Step (ii): Fix  $x = \bar{x}, y = \bar{y}$  and optimize  $(z^s)_{s \in \mathcal{S}}$  in SP-MIP to give optimal value  $w^{Heur}$ .

We now compare the Base Algorithm, the heuristic SP-MIP-heur, the solution of SP-MIP, SPM-MIP-half-max, and PI-MIP. The results are reported in Table 3. Results reported in the ‘‘Gap’’ column in Table 3 represent the optimality gap for SP-MIP and SPM-MIP, the optimality gap obtained at the end of Step (i) of Algorithm 4.1 for the Base Algorithm and SP-MIP-heur, and the largest optimality gap obtained in solving D-MIP<sup>s</sup> over all  $s \in \mathcal{S}$  when calculating  $w^{PI}$ .

First, we note that improvement in the optimal value is recorded for SP-MIP-heur over the Base Algorithm. The largest improvement is recorded for Instance 2 and 8 (1.98% and 1.03% respectively), with an average improvement of 0.56%.

When we compare the optimal values from Table 3, we see that for some instances the improvement of SP-MIP over the Base Algorithm is larger than for others, for example Instances 2, 3, and 8 show an improvement of over 2% (2.59 %, 2.43%, and 2.12% respectively), while Instances 5 and 7 show hardly any improvement (0.01% and 0.03% respectively). The average improvement, calculated over all instances, is 1.07%.

From Table 3 we see that the improvement in the objective values obtained by SPM-MIP-half-max over those obtained by the Base Algorithm increases when the mining decisions are also allowed

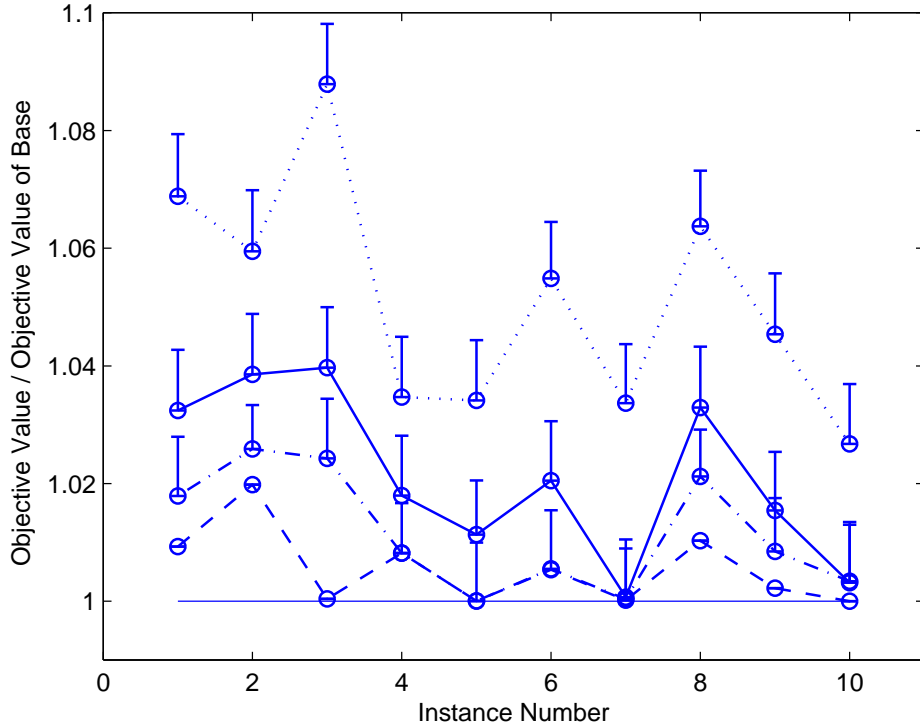


Figure 1: The relative objective values obtained, with respect to the Base Algorithm. The algorithm corresponding to value 1 on the Objective Value/Objective Value of Base axis is the Base Algorithm. The first algorithm up is SP-MIP-heur (dashed line), followed by SP-MIP (dash-dot line), SPM-MIP-half-max (solid line) and PI-MIP (dotted line). The “whiskers” indicate the range of relative values possible for each model over the Base Algorithm, taking into account the gap values.

to depend on the scenario. The best such improvements are recorded for Instance 1 (3.24%), Instance 2 (3.86%), Instance 3 (3.97%), and Instance 8 (3.29%). The average improvement, over all instances, is 2.13%. We note that this average improvement is bounded by the average improvement obtained by the Perfect Information Algorithm, which is 5.09%. The relative improvement with respect to the Base Algorithm is clearly illustrated in Figure 1 for all algorithms. We finally note that SP-MIP was further improved by allowing the mining decisions to depend on scenarios (i.e. by solving SPM-MIP-half-max) by 0.96% on average.

Instances 7 and 10 in which the improvement of SPM-MIP-half-max over the Base Algorithm is very small (0.07%, resp. 0.32%), illustrate the effect of the non-anticipativity constraints. Without non-anticipativity constraints, the Perfect Information Algorithm is able to use prior knowledge of the scenarios to (unrealistically) improve the expected net present value for both Instance 7 and 10. Instances 7 and 10 represent the instances in which differentiation of scenario pairs occurred the latest; no pair of scenarios could be distinguished before period 5 (see Table 4). In SPM-MIP-half-max the non-anticipativity constraints correctly enforced all mining decisions made in periods 1 through 5 to be identical for all scenarios, and the processing decisions made in periods 1 through 4 to be identical. Thus each schedule  $(x^s, y^s, z^s)$ ,  $s = 1, \dots, 5$  of SPM-MIP-half-max is committed to the same excavation over the first 5 periods, and so once new information arises in period 5 that allows us to distinguish between some scenarios, there is little chance for altering the remainder of

the schedule in a highly NPV-beneficial way.

## 5 Conclusions

We have extended a state-of-the-art MIP formulation of the OPMPSP to incorporate multiple conditional simulations of the resource geology, using a multistage stochastic programming approach. Our extension took two forms: we first allowed processing decisions to depend on geological information in a non-anticipative fashion (SP-MIP); then we additionally allowed mining decisions to depend on geological information (SPM-MIP). The main advantages of our new formulations are: (i) the additional flexibility of mining and processing decisions and the ability to alter these decisions as new geological information is revealed; (ii) the ability to precisely control production rates for each possible geological outcome; (iii) the maximization of net present value without the need for additional penalties in the objective; and (iv) simultaneous optimization of cutoff grade decisions.

The new stochastic models SP-MIP and SPM-MIP were naturally more beneficial in cases where the optimal mining schedules for each scenario, considered independently, show significant differences. This is not necessarily related to having a large grade variability; the latter does not necessarily lead to different excavation decisions being made under different scenarios. The stochastic models also naturally yield better NPV when it is possible to learn distinctive geological information relatively soon. As we see from Instances 7 and 10, and Table 4, when no scenarios can be distinguished until many years from the present, decisions cannot be altered until far in the future: because of discounting in the objective, this will make little difference to the expected net present value. As can be seen from Figure 3, the optimal value of the Perfect Information Algorithm relative to the Base Algorithm is quite a good predictor for when the stochastic models are likely to yield greatest potential benefit.

In the numerical experiments reported here, we used five scenarios per instance in order to fully report all objective values for all scenarios. In practice, we advise using many more scenarios per instance to achieve a more accurate description of the uncertain geology. The runtimes of SP-MIP are not appreciably greater than the Base Algorithm, which is already solvable in practical situations in an acceptable timeframe. Indeed, employing the methodology of Boland et al. (2008), a very accurate geological description can be maintained within reasonable runtimes. The runtimes of SPM-MIP are presently larger than the Base Algorithm and more work is required to make the numerical solution of SPM-MIP more efficient. This is a focus of future research, in which we will consider generalizing approaches of Boland et al. (2008) to problems with the stochastic model structure, as well as strengthening the stochastic models through the use of cutting planes.

This paper has also contributed to the general case of multistage stochastic programming with endogenous uncertainty: we are able to streamline the approach of Goel and Grossmann (2006), and generalize their ability to reduce the number of non-anticipativity constraints that must be added to general scenarios spaces, not necessarily constructed from cross products.

## 6 Acknowledgements

The authors are very grateful to Merab Menabde, Peter Stone, and Mark Zuckerberg (BHP Billiton) for their ongoing support and guidance on a variety of practical mining-related issues and for numerous technical suggestions and insightful feedback that improved the content and exposition of this work. We also thank Hamish Waterer for helpful suggestions and proof-reading, and Ambros Gleixner for his help with the code implementation.

This research is supported by the Australian Research Council Linkage Project LP0561744 and by BHP Billiton Limited.

## References

- Artstein, Z., R. J.-B. Wets. 1993. Sensors and information in optimization under stochastic uncertainty. *Mathematics of Operations Research* **28** 523–547.
- Birge, J.R., F. Louveaux. 1997. *Introduction to stochastic programming*. Springer Verlag.
- Boland, N., C.Fricke, G.Froyland. 2007. A strengthened formulation for the open pit mine production scheduling problem. Optimization Online.
- Boland, N., I. Dumitrescu, G. Froyland, A.M. Gleixner. 2008. LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers & Operations Research* To appear.
- Caccetta, L., S.P. Hill. 2003. An application of branch and cut to open pit mine scheduling. *Journal of Global Optimization* **27** 349–365.
- Dimitrakopoulos, R. 1998. Conditional simulation algorithms for modelling orebody uncertainty in open pit optimisation. *International Journal of Surface Mining, Reclamation and Environment* **12** 173–179.
- Fricke, C. 2006. Applications of integer programming in open pit mining. Ph.D. thesis, The University of Melbourne.
- Goel, V., I. E. Grossmann. 2006. A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming Series B* **108** 355–394.
- Goel, V., I. E. Grossmann, A. S. El-Bakry, E. L. Mulkay. 2006. A novel branch and bound algorithm for optimal development of gas fields under uncertainty in reserves. *Computers and Chemical Engineering* **30** 1076–1092.
- Golamnejad, J., M. Osanloo, B. Karimi. 2006. A chance-constrained programming approach for open pit long-term production scheduling in stochastic environments. *The Journal of the South African Institute of Mining and Metallurgy* **106** 105–114.
- Goovaerts, P. 1997. *Geostatistics for Natural Resources Evaluation*. Oxford University Press.
- Jonsbråten, T.W., R. J.-B. Wets, D. L. Woodruff. 1998. A class of stochastic programs with decision dependent random elements. *Annals of Operations Research* **82** 83–106.
- Journel, A.G. 1996. Modelling uncertainty and spatial dependence: Stochastic imaging. *Int. J. Geographical Information Systems* **10** 517–522.
- Menabde, M., G. Froyland, P. Stone, G. Yeates. 2004. Mining schedule optimisation for conditionally simulated orebodies. *Proceedings of the International Symposium on Orebody Modelling and Strategic Mine Planning: Uncertainty and Risk Management*. Perth, WA, 347–352.
- Osanloo, M., J. Gholamnejad, B. Karimi. 2007. Long-term open pit mining production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment* **22** 3–35.
- Ramazan, S., R. Dimitrakopoulos. 2007. *Orebody modelling and Strategic Mine Planning*, chap. Stochastic optimisation of long-term production scheduling for open pit mines with a new integer programming formulation. 2nd ed. Spectrum Series, The Australian Institute of Mining and Metallurgy, 385–392.
- Smith, M., R. Dimitrakopoulos. 1999. The influence of deposit uncertainty on mine production scheduling. *International Journal of Mining, Reclamation and Environment* **13** 173–178.
- Stone, P., G. Froyland, M. Menabde, B. Law, R. Pasyar, P. Monkhouse. 2004. Blasor - blended iron ore mine planning optimisation at Yandi. *Proceedings of Orebody Modelling and Strategic Mine Planning - Uncertainty and Risk Management*. Perth, WA, 285–288.

Tarhan, B., I. E. Grossmann. 2008. A multistage stochastic programming approach with strategies for uncertainty reduction in the synthesis of process networks with uncertain yields. *Computers and Chemical Engineering* **32** 766–788.

Table 1: Numerical comparison between SPM-MIP and the models that use only half of the non-anticipativity constraints, elimination of redundant non-anticipativity constraints for the  $z$  variables, or both halving and elimination. Column 3 indicates whether or not the non-anticipativity constraints (NAC) are added as lazy constraints.

Inst. No.	Method	Lazy NAC	No. of nodes	Opt. val. divided by $10^8$	Gap (%)	Time (sec)	NPV realised under each scenario, divided by $10^8$ (the first four significant figures)				
1	SPM-MIP	yes	10179	3.91516500	1.00	44802	3.9283	3.6905	4.4197	3.6525	3.8846
	SPM-MIP	no	6702	3.91195818	1.00	16640	3.9247	3.6902	4.4197	3.6513	3.8736
	SPM-MIP-elim	no	7859	3.91154351	1.00	18254	3.9283	3.6905	4.4015	3.6525	3.8846
	SPM-MIP-half-min	no	8279	3.91444371	1.00	15620	3.9247	3.6905	4.4197	3.6525	3.8846
	SPM-MIP-half-max	no	6312	3.91225202	1.00	9871	3.9247	3.6905	4.4197	3.6525	3.8736
	SPM-MIP-half-max-elim	no	7268	3.91297332	1.00	11167	3.9283	3.6905	4.4197	3.6525	3.8736
2	SPM-MIP	yes	1225	3.97434460	0.97	10914	4.0533	3.9235	4.1408	4.1485	3.6054
	SPM-MIP	no	917	3.97643384	1.00	3625	4.0547	3.9235	4.1439	4.1544	3.6054
	SPM-MIP-elim	no	1438	3.97340148	0.99	6402	4.0533	3.9235	4.1408	4.1485	3.6007
	SPM-MIP-half-min	no	1469	3.97530900	0.98	3905	4.0508	3.9182	4.1480	4.1565	3.6029
	SPM-MIP-half-max	no	1028	3.97643384	0.99	3705	4.0547	3.9235	4.1439	4.1544	3.6054
	SPM-MIP-half-max-elim	no	796	3.97643384	0.99	3126	4.0547	3.9235	4.1439	4.1544	3.6054
3	SPM-MIP	yes	1019	3.94818376	0.96	11715	3.8098	4.1156	4.0094	3.9275	3.8784
	SPM-MIP	no	654	3.95056258	0.95	3026	3.8195	4.0957	4.0195	3.9512	3.8666
	SPM-MIP-elim	no	454	3.94944415	0.92	2181	3.8087	4.1178	4.0110	3.9318	3.8777
	SPM-MIP-half-min	no	489	3.94836744	0.96	2401	3.8100	4.1121	4.0114	3.9318	3.8764
	SPM-MIP-half-max	no	559	3.94886834	0.99	2049	3.8093	4.1054	4.0113	3.9404	3.8777
	SPM-MIP-half-max-elim	no	622	3.94554872	1.00	2888	3.8098	4.1128	4.0059	3.9275	3.8715
4	SPM-MIP	yes	4424	3.90627170	0.99	34795	3.9594	3.8163	3.5891	4.0078	4.1585
	SPM-MIP	no	2011	3.90595715	0.99	11321	3.9590	3.8163	3.5891	4.0078	4.1574
	SPM-MIP-elim	no	1902	3.90595715	1.00	7845	3.9590	3.8163	3.5891	4.0078	4.1574
	SPM-MIP-half-min	no	1636	3.90595909	0.98	5928	3.9594	3.8163	3.5891	4.0062	4.1585
	SPM-MIP-half-max	no	2082	3.90619353	1.00	6267	3.9590	3.8163	3.5891	4.0078	4.1585
	SPM-MIP-half-max-elim	no	1812	3.90619353	1.00	6353	3.9590	3.8163	3.5891	4.0078	4.1585
5	SPM-MIP	yes	824	3.82248678	0.99	18421	3.6768	3.8366	3.9861	3.8307	3.7820
	SPM-MIP	no	683	3.82013976	0.93	4603	3.6702	3.8315	3.9861	3.8307	3.7820
	SPM-MIP-elim	no	894	3.82248678	0.99	4291	3.6768	3.8366	3.9861	3.8307	3.7820
	SPM-MIP-half-min	no	674	3.82248678	1.00	4271	3.6768	3.8366	3.9861	3.8307	3.7820
	SPM-MIP-half-max	no	645	3.82248678	0.91	2830	3.6768	3.8366	3.9861	3.8307	3.7820
	SPM-MIP-half-max-elim	no	562	3.82248678	0.98	3074	3.6768	3.8366	3.9861	3.8307	3.7820
6	SPM-MIP	yes	5823	3.90121379	0.99	28429	4.0182	4.1815	3.8937	3.7978	3.6146
	SPM-MIP	no	2622	3.90037571	1.00	9428	4.0548	4.0353	3.8904	3.8554	3.6657
	SPM-MIP-elim	no	2259	3.90263115	0.99	7148	4.0192	4.1793	3.8930	3.7982	3.6232
	SPM-MIP-half-min	no	2633	3.90305409	1.00	7374	4.0192	4.1815	3.8930	3.7982	3.6232
	SPM-MIP-half-max	no	2741	3.90296153	0.99	7621	4.0182	4.1815	3.8937	3.7978	3.6233
	SPM-MIP-half-max-elim	no	3470	3.90231068	1.00	10016	4.0192	4.1815	3.8930	3.7982	3.6195
7	SPM-MIP	yes	828	3.89112692	0.97	9687	3.9649	3.6514	4.0813	4.1080	3.6499
	SPM-MIP	no	410	3.89325698	0.97	3155	4.0526	3.6773	3.9928	4.0502	3.6931
	SPM-MIP-elim	no	705	3.89325698	0.96	5860	4.0526	3.6773	3.9928	4.0502	3.6931
	SPM-MIP-half-min	no	706	3.89316521	0.99	3865	4.0525	3.6719	3.9922	4.0510	3.6980
	SPM-MIP-half-max	no	540	3.89325698	0.98	2647	4.0526	3.6773	3.9928	4.0502	3.6931
	SPM-MIP-half-max-elim	no	700	3.88558596	0.96	3935	3.9647	3.6436	4.0692	4.1087	3.6414
8	SPM-MIP	yes	5282	3.96581796	0.99	25681	3.7865	3.9557	3.8612	4.1572	4.0683
	SPM-MIP	no	3943	3.96569746	1.00	12700	3.7865	3.9557	3.8594	4.1578	4.0689
	SPM-MIP-elim	no	5089	3.96569746	1.00	14181	3.7865	3.9557	3.8594	4.1578	4.0689
	SPM-MIP-half-min	no	4256	3.96566440	1.00	9271	3.7865	3.9557	3.8594	4.1578	4.0687
	SPM-MIP-half-max	no	4525	3.96514360	1.00	8303	3.7820	3.9557	3.8612	4.1578	4.0689
	SPM-MIP-half-max-elim	no	3964	3.96470608	1.00	6760	3.7809	3.9557	3.8612	4.1578	4.0677
9	SPM-MIP	yes	2020	3.97682287	0.98	12488	3.7477	4.0165	4.0733	3.9288	4.1076
	SPM-MIP	no	880	3.98132711	0.96	5550	3.7572	4.0165	4.0954	3.9294	4.1079
	SPM-MIP-elim	no	814	3.97707900	1.00	4944	3.7575	4.0165	4.0738	3.9294	4.1079
	SPM-MIP-half-min	no	680	3.98138991	0.99	2817	3.7575	4.0165	4.0954	3.9294	4.1079
	SPM-MIP-half-max	no	544	3.98132711	0.98	2793	3.7572	4.0165	4.0954	3.9294	4.1079
	SPM-MIP-half-max-elim	no	713	3.98128888	0.99	2585	3.7570	4.0165	4.0954	3.9294	4.1079
10	SPM-MIP	yes	2735	3.75644050	0.99	17008	3.6285	3.8499	3.8443	3.7154	3.7439
	SPM-MIP	no	1233	3.75984015	0.99	6538	3.6499	3.8451	3.8307	3.7275	3.7458
	SPM-MIP-elim	no	1324	3.75984015	0.98	5576	3.6499	3.8451	3.8307	3.7275	3.7458
	SPM-MIP-half-min	no	1442	3.75954496	1.00	3909	3.6499	3.8451	3.8307	3.7267	3.7452
	SPM-MIP-half-max	no	1585	3.75737210	0.98	4652	3.6499	3.8451	3.8307	3.7151	3.7458
	SPM-MIP-half-max-elim	no	1467	3.75984015	0.97	3950	3.6499	3.8451	3.8307	3.7275	3.7458

Table 2: Numerical results for SP-MIP-h and SPM-MIP-h compared with SP-MIP and SPM-MIP-half-max. SPM-MIP-h also used only half of the non-anticipativity constraints, the halving process being identical to that of SPM-MIP-half-max. The number of nodes reported for SPM-MIP-h is that used by the first step of the heuristic, when no non-anticipativity constraints are considered for the  $z$  variables, but non-anticipativity constraints are included in the model for the  $x$  and  $y$  variables (once the  $x$  and  $y$  variables are fixed, at the second step, the model to solve is simply a linear programme).

Inst. No.	Method	No. of nodes	Opt. val. divided by $10^8$	Time (sec)
1	SP-MIP	200	3.85719671	200
	SP-MIP-h	386	3.82478302	160
	SPM-MIP-half-max	6312	3.91225202	9871
	SPM-MIP-h	8864	3.90774027	17470
2	SP-MIP	114	3.92782149	153
	SP-MIP-h	240	3.92729956	133
	SPM-MIP-half-max	1028	3.97643384	3705
	SPM-MIP-h	2132	3.97593213	4868
3	SP-MIP	232	3.89034536	116
	SP-MIP-h	486	3.82574769	227
	SPM-MIP-half-max	559	3.94886834	2049
	SPM-MIP-h	1464	3.88686308	4845
4	SP-MIP	350	3.86873294	229
	SP-MIP-h	310	3.86826895	122
	SPM-MIP-half-max	2082	3.90619353	6267
	SPM-MIP-h	2677	3.90580771	8637
5	SP-MIP	411	3.77991867	226
	SP-MIP-h	240	3.77859649	118
	SPM-MIP-half-max	645	3.82248678	2830
	SPM-MIP-h	559	3.82081281	2237
6	SP-MIP	449	3.84565550	259
	SP-MIP-h	784	3.84594777	203
	SPM-MIP-half-max	2741	3.90296153	7621
	SPM-MIP-h	4452	3.90300971	9772
7	SP-MIP	163	3.89196654	151
	SP-MIP-h	190	3.88898816	88
	SPM-MIP-half-max	540	3.89325698	2647
	SPM-MIP-h	499	3.89064751	2735
8	SP-MIP	420	3.92021316	322
	SP-MIP-h	373	3.91866870	158
	SPM-MIP-half-max	4525	3.96514360	8303
	SPM-MIP-h	4045	3.96426079	9678
9	SP-MIP	192	3.95403529	167
	SP-MIP-h	290	3.93567840	108
	SPM-MIP-half	544	3.98132711	2793
	SPM-MIP-h	1192	3.98004085	4604
10	SP-MIP	421	3.75835091	158
	SP-MIP-h	295	3.75831813	96
	SPM-MIP-half	1585	3.75737210	4652
	SPM-MIP-h	1112	3.75839130	2256

Table 3: Numerical results for the Base Algorithm, SP-MIP-heur, SP-MIP, SPM-MIP-half-max, and PI-MIP.

Inst. No.	Method	Opt. val. divided by $10^8$	Gap (%)	Time (sec)	NPV realised under each scenario, divided by $10^8$				
					Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
1	Base Alg.	3.78942783	0.00	68	3.98924852	3.70509133	4.23610316	3.29513410	3.72156191
	SP-MIP-heur	3.82463950	0.00	68	3.99024336	3.70310109	4.38619264	3.29933234	3.74432807
	SP-MIP	3.85719671	0.99	200	3.98253850	3.66301025	4.37115149	3.49519977	3.77408354
	SPM-MIP-half-max	3.91225202	1.00	9871	3.92477078	3.69057897	4.41970684	3.65253240	3.87367112
	PI-MIP	4.05021755	0.99	386	4.07998030	3.86707723	4.42275812	3.95406058	3.92721152
2	Base Alg.	3.82875926	0.00	68	3.46907170	4.03802202	4.03389099	3.99431987	3.60849169
	SP-MIP-heur	3.90466509	0.00	68	3.83830889	4.00742029	4.05938158	4.01817208	3.60004262
	SP-MIP	3.92782149	0.73	153	3.85478297	3.90951249	4.13295744	4.14893668	3.59291788
	SPM-MIP-half-max	3.97643384	0.99	3705	4.05470308	3.92358685	4.14396101	4.15446756	3.60545068
	PI-MIP	4.05642527	0.98	380	4.18460300	4.05420818	4.14889205	4.15979259	3.73463053
3	Base Alg.	3.79810769	0.00	65	3.76737171	3.88061503	3.89707460	3.53734963	3.90812748
	SP-MIP-heur	3.79968284	0.00	65	3.76763502	3.88256371	3.89707460	3.54301339	3.90812748
	SP-MIP	3.89034536	0.99	116	3.78567426	4.07096571	4.00760660	3.72353636	3.86394386
	SPM-MIP-half-max	3.94886834	0.99	2049	3.80934093	4.10544602	4.01139688	3.94040822	3.87774966
	PI-MIP	4.13191321	0.94	193	4.16587952	4.36493762	4.03382302	4.10923639	3.98568949
4	Base Alg.	3.83726980	0.00	82	3.83857815	3.78961945	3.52416162	3.94285479	4.09113504
	SP-MIP-heur	3.86873294	0.00	82	3.95291123	3.79434475	3.49660795	3.94338158	4.15641920
	SP-MIP	3.86873294	0.84	229	3.95291123	3.79434475	3.49660795	3.94338158	4.15641920
	SPM-MIP-half-max	3.90619353	1.00	6267	3.95903142	3.81638484	3.58914084	4.00781154	4.15859900
	PI-MIP	3.97048368	0.99	332	4.01797047	3.89854546	3.73599435	4.02282651	4.17708159
5	Base Alg.	3.77971841	0.00	142	3.70673516	3.84020373	3.93441321	3.63412810	3.78311201
	SP-MIP-heur	3.77991867	0.00	142	3.70673516	3.84020373	3.93541435	3.63412810	3.78311201
	SP-MIP	3.77991867	0.99	226	3.70673516	3.84020373	3.93541435	3.63412810	3.78311201
	SPM-MIP-half-max	3.82248678	0.91	2830	3.67688606	3.83662531	3.98616441	3.83070894	3.78204917
	PI-MIP	3.90869876	0.99	366	3.81120046	3.87151741	4.08643766	3.97113305	3.80320520
6	Base Alg.	3.82456135	0.00	50	3.99097372	3.99097372	3.90031785	3.83655643	3.58922382
	SP-MIP-heur	3.84506840	0.00	50	3.99558623	3.90111612	3.90445418	3.83723438	3.58695111
	SP-MIP	3.84565550	0.99	259	3.99557691	3.90284709	3.90970300	3.83438807	3.58576243
	SPM-MIP-half-max	3.90296153	0.99	7621	4.01828835	4.18150137	3.89379797	3.79786645	3.62335349
	PI-MIP	4.03443542	0.91	450	4.08377619	4.18708260	4.05598637	3.89768580	3.94764612
7	Base Alg.	3.89059855	0.00	48	4.05477990	3.66867724	3.98716204	4.04343593	3.69893768
	SP-MIP-heur	3.89129920	0.00	48	4.05555101	3.66867724	3.98858629	4.04474376	3.69893768
	SP-MIP	3.89196654	0.86	151.86	4.05620154	3.68306105	3.98854662	4.04250982	3.68951370
	SPM-MIP-half-max	3.89325698	0.98	2647	4.05269121	3.67735087	3.99283279	4.05029015	3.69311988
	PI-MIP	4.02162974	0.97	425	4.11501095	3.82253462	4.19145593	4.15553499	3.82361221
8	Base Alg.	3.83875903	0.00	63	3.63455435	3.83636990	3.78846683	3.98647565	3.94792836
	SP-MIP-heur	3.87832435	0.00	63	3.64503223	3.97277836	3.80077430	4.02672160	3.94631524
	SP-MIP	3.92021316	0.78	322	3.76146995	3.85880821	3.82808406	4.14756592	4.00513768
	SPM-MIP-half-max	3.96514360	1.00	8303	3.78203532	3.95574066	3.86120231	4.15782313	4.06891657
	PI-MIP	4.08337774	0.89	366	3.83794400	4.16446719	3.93432036	4.37151143	4.10864570
9	Base Alg.	3.92086733	0.00	51	3.55760283	3.75494300	4.18179151	3.98510886	4.12489034
	SP-MIP-heur	3.92961564	0.00	51	3.57270910	3.75638430	4.18777114	3.98789961	4.14331406
	SP-MIP	3.95403529	0.90	167	3.73236255	3.95049626	4.06739165	3.92522211	4.09470387
	SPM-MIP-half-max	3.98132711	0.98	2793	3.75725225	4.01651975	4.09542979	3.92948367	4.10795006
	PI-MIP	4.09878742	0.99	407	3.91160354	4.10589843	4.28744817	4.01942120	4.16956576
10	Base Alg.	3.74555077	0.00	89	3.66643260	3.86306443	3.80963431	3.64179710	3.74682541
	SP-MIP-heur	3.74555077	0.00	90	3.66643260	3.86306443	3.80963431	3.64179710	3.74682541
	SP-MIP	3.75835091	1.00	158	3.64974311	3.84514164	3.83011089	3.72216300	3.74459593
	SPM-MIP-half-max	3.75737210	0.98	4652	3.64997828	3.84511314	3.83070398	3.71518655	3.74587855
	PI-MIP	3.84578129	0.99	470	3.80789581	3.90518356	3.86517066	3.88769297	3.76296346



Table 4: The number of scenario pairs separated at any time, for each instance.

Instance No.	Method	The number of scenario pairs separated in time period $t$									
		$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 8$	$t = 9$	$t = 10$
1	SPM-MIP-half	0	4	4	4	10	10	10	10	10	10
	SP-MIP-heur	0	4	4	4	9	10	10	10	10	10
2	SPM-MIP-half	0	0	0	7	9	9	9	10	10	10
	SP-MIP-heur	0	0	0	7	9	9	9	10	10	10
3	SPM-MIP-half	0	0	3	4	8	8	10	10	10	10
	SP-MIP-heur	0	0	3	4	6	8	10	10	10	10
4	SPM-MIP-half	0	0	0	4	7	10	10	10	10	10
	SP-MIP-heur	0	0	0	4	7	10	10	10	10	10
5	SPM-MIP-half	0	0	0	2	6	9	9	9	10	10
	SP-MIP-heur	0	0	0	2	6	6	9	10	10	10
6	SPM-MIP-half	0	4	4	6	9	9	10	10	10	10
	SP-MIP-heur	0	0	0	6	6	9	10	10	10	10
7	SPM-MIP-half	0	0	0	0	6	8	8	10	10	10
	SP-MIP-heur	0	0	0	0	6	8	8	10	10	10
8	SPM-MIP-half	0	0	0	10	10	10	10	10	10	10
	SP-MIP-heur	0	0	0	9	10	10	10	10	10	10
9	SPM-MIP-half	0	0	0	2	10	10	10	10	10	10
	SP-MIP-heur	0	0	2	2	7	10	10	10	10	10
10	SPM-MIP-half	0	0	0	0	6	10	10	10	10	10
	SP-MIP-heur	0	0	0	0	2	6	10	10	10	10