Underground Mine Scheduling Under Uncertainty

Peter Nesbitt[a], Lewis R. Blake[b], Patricio Lamas[c], Marcos Goycoolea[c], Bernardo Pagnoncelli[c], Alexandra Newman*[d], Andrea Brickey[e]

[a] *Operations Research Department, Naval Postgraduate School, Monterey, California 93943*
[b] *Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, Colorado 80401*
[c] *School of Business, Universidad Adolfo Ibañez, Santiago, Chile*
[d] *Department of Mechanical Engineering, Colorado School of Mines, Golden, Colorado 80401*
[e] *Department of Mining Engineering and Management, South Dakota School of Mines and Technology, Rapid City, South Dakota 57701*

Abstract

Underground mine schedules seek to determine start dates for activities related to the extraction of ore, often with an objective of maximizing net present value; constraints enforce geotechnical precedence between activities, and restrict resource consumption on a per-time-period basis, e.g., development footage and extracted tons. Strategic schedules address these start dates at a coarse level, whereas tactical schedules must account for the day-to-day variability of underground mine operations, such as unanticipated equipment breakdowns and ground conditions, both of which might slow production. At the time of this writing, the underground mine scheduling literature is dominated by a deterministic treatment of the problem, usually modeled as a Resource Constrained Project Scheduling Problem (RCPSP), which precludes mine operators from reacting to unforeseen circumstances. Therefore, we propose a stochastic integer programming framework that: (i) characterizes uncertainty in duration and economic value for each underground mining activity; (ii) formulates a new stochastic variant of the RCPSP; (iii) suggests an optimization-based heuristic; and, (iv) produces implementable, tactical schedules in a practical amount of time and provides corresponding managerial insights.

Key words: OR in natural resources, Stochastic integer programming, Optimization-based heuristics, Project scheduling, Underground mining

Email: nesbitt@nps.edu (Peter Nesbitt), lblake@mines.edu (Lewis R. Blake), plamas@alumnos.uai.cl (Patricio Lamas), marcos.goycoolea@uai.cl (Marcos Goycoolea), bernardo.pagnoncelli@uai.cl (Bernardo Pagnoncelli), anewman@mines.edu (Alexandra Newman*), Andrea.Brickey@sdsmt.edu (Andrea Brickey)

*corresponding author

1. Introduction

Underground mining seeks to extract ore from deep underground through constructed passageways, or tunnels. An underground mine design defines the infrastructure necessary to efficiently gain access to this ore, and a production schedule informs the timing of operational decisions, or the execution of *activities*, given a design. These activities might consist of the extraction of three-dimensional, notional *blocks* in an open pit mine, or the mining, and subsequent backfilling, of a *stope* in an underground mine. Common objectives include maximizing net present value or minimizing deviations from contracts. Constraints: (i) enforce physical precedence between activities, e.g., development in an area before extraction; and, (ii) restrict resource consumption on a per-time-period basis, e.g., development footage and extracted tons. Brickey [2015] presents a generalized underground mine scheduling model as a *resource-constrained project scheduling problem* (RCPSP) in which: (i) the duration of each activity; (ii) lag, or required delay between activities; (iii) resources consumed by each activity; and, (iv) economic value of completing each activity are known. Scheduling synchronizes allocation of labor and mechanical resources within the production process; in practice, schedules often fall short of providing an achievable plan because of uncertainty associated with the parameters; rather, mine planners must make real-time adjustments as better estimates of the data are realized. To mitigate the lack of clairvoyance using a more tailored approach than the ad hoc addition of incorporating slack into the schedule, we propose to include uncertainty for tactical decision making.

Production scheduling is used by mine management to make large financial decisions (e.g., the size and quantity of equipment to purchase) and to meet production goals (e.g., maximize net present value) [Dowd et al., 2016]. In this paper, we develop policies associated with enterprises that are under the control of a single planner. We take exogenous factors (such as the price of a commodity) as given, thereby omitting market influences. However, this planner must still contend with uncertainty inherent to the mining operation. Because the market fluctuates drastically day-to-day, corporate policy tends to take a longer-term approach with respect to metal price. Therefore, the tactical uncertainty we consider addresses the duration of activities that must be executed to extract the mineral, and its grade. These act as proxies for the overall uncertainty in the system, and serve to create stable production levels, which, in turn, generate a stable revenue stream. We assume that all uncertainty associated with an underground mining activity is only resolved completely when the activity is completed. We propose using as a frame of reference a model whose solution yields a typical industry-derived schedule based on mean-value point estimates.

The contributions of this paper are as follows: (i) a means to characterize uncertainty in duration and ore grade through multiple scenarios; (ii) a new stochastic programming formulation that takes into account uncertainty in duration and economic value for each underground mining activity, maximizing expected net present value by defining an interval in which activities start, rather than a precise moment in time; (iii) a corresponding optimization-based heuristic; and, (iv) managerial insights in contrast to those from a deterministic schedule. The remainder of this paper is organized as follows: §2 provides a literature review of deterministic and stochastic mine planning models, with an emphasis on underground operations; §3 introduces the creation of scenarios and a formulation of our integer program; §4 describes our solution techniques, including an optimization-based heuristic and the implications of relaxing certain constraints in our integer programming model; §5 presents results and corresponding analysis, while §6 concludes.

2. Literature Review

Underground scheduling is more difficult than its open-pit counterpart [O'Sullivan et al., 2015]. The following factors are common sources of complexity: (i) the activity data, e.g., durations, are heterogeneous; (ii) practical instances are particularly large, i.e., they contain many (discrete) variables and constraints; and, (iii) there is an unstylized precedence structure and the graph corresponding to the precedence relationships between activities is dense. Trout [1995] first discusses a mixed-integer program to schedule underground ore extraction and backfilling activities. Carlyle and Eaves [2001] expand Trout's work by including development activities for a platinum and palladium mine in Stillwater, Montana. Kuchta et al. [2004] and Newman and Kuchta [2007] demonstrate a means to solve instances of a mixed-integer program that yields lower deviations from contracts compared to manual practice at Kiruna Mine, Sweden. Nehring et al. [2010] integrate operational and tactical underground mining schedules into a single mathematical model through minimizing deviation of targeted mill feed grade while maximizing net present value. O'Sullivan and Newman [2014] develop optimization-based heuristics that produce schedules for an underground lead and zinc mine in Ireland with a complex set of precedence constraints. King et al. [2017] determines the boundary between open pit and underground mining, presenting corresponding schedules for both parts of the mine. Brickey et al. [2019] present five-year tactical schedules at daily fidelity for Barrick's Turquoise Ridge cut-and-fill mine. Some mine production scheduling problems such as King et al. [2017] and Brickey et al. [2019] can be cast in an RCPSP-like framework. The main differences between this framework in a mining context and in a more classical context are: (i) in mine scheduling problems, the goal is to maximize the (expected) Net Present Value (NPV) of the mine while in the classical RCPSP, the typical objective is to minimize the makespan of the project; and (ii), in mining problems, each activity is optionally executed, while in the classical RCPSP, all activities must be executed. Research such as O'Sullivan and Newman [2014], King et al. [2017], and Brickey et al. [2019] incorporate greater operational details than earlier work, which, in turn, produces more adoptable schedules. However, none of these references incorporates uncertainty into their scheduling paradigm, and all are therefore more suited to longer term, strategic mining.

In reality, there is uncertainty associated with most inputs, e.g., production rates, costs, and profits, of the mine planning process; point estimates do not necessarily generate feasible *tactical* schedules. The mining industry addresses uncertainty explicitly, but not necessarily through optimization-based methods. Sari [2009] utilizes stochastic modeling to evaluate the potential for accidents and, correspondingly, worker-days lost, in a Turkish coal mine. The author combines statistical modeling and Monte Carlo simulations. In another safety-related application, Karacan and Luxbacher [2010] model the performance of gob gas ventholes, which are used to remove methane in previously mined areas of longwall coal mines; as in Sari [2009], their techniques include multi-parameter regression models and Monte Carlo simulations to determine the variability in venthole performance. Researchers have begun to incorporate uncertainty in their models to produce more realistic mine plans. Rojas et al. [2007] formulate an optimal control policy for the extraction of ore in an open pit mine, and demonstrate their methodology on a small example. Lamghari and Dimitrakopoulos [2012] develop heuristic search techniques to solve an open pit mine production scheduling problem cast as a stochastic integer program that accounts for uncertainty in metal content. Another common practice in strategic decisions extends deterministic analysis by quantifying the effects of uncertainty at multiple, fixed levels of market conditions [Rossi, 2014]. Reus et al. [2019] consider uncertainty related to production incidents such as strikes and accidents that may slow production and/or decrease expected profits. Their stochastic program represents a strategic mine-planning model which they decompose to enable the inclusion of a very large number of scenarios over a 15-year planning horizon. Caldentey et al. [2019] apply real options to address price uncertainty for making capacity expansion decisions in a long-term copper mining project.

While these works consider uncertainty at an aggregate planning level, other researchers focus on uncertainty at the block level in the production planning process. For instance, Alonso-Ayuso et al. [2014] provide an example of the inclusion of uncertainty in underground mining with respect to copper price in a block caving (underground) mine scheduling problem; their stochastic program considers many scenarios, and is then transformed into a deterministic equivalent. By testing value-at-risk and conditional-value-at-risk strategies, they conclude that a very modest reduction in expected profit with respect to the risk-neutral model can offer significantly better risk control, measured as the probability of having negative profit, and the expected losses given that losses will occur. Carpentier et al. [2016] seek a robust cut-off grade for a cluster of underground nickel mines that use the same labor and material resources; their two-stage stochastic program includes decisions related to mine opening and closure, and incorporates precedence and elastic constraints on mining operations (e.g., development and extraction); the objective maximizes net present value and minimizes deviation from target production levels. Dirkx and Dimitrakopoulos [2018] also account for uncertainty in grade and drawdown rate in determining feasibility of meeting long-term production targets for a potential mineral deposit using block cave mining. The authors use stochastic mixed-integer programming to maximize the net present value and minimize production target deviation with respect to mining capacity, continuous extraction, production grade, inter-drawpoint precedence, and milling operations. Del Castillo and Dimitrakopoulos [2019] optimize production planning in the face of price and geologic uncertainty for an open-pit mining complex. Their model considers long-term design and fleet sizing, as well as shorter term tactical decisions. They apply their multi-stage model to a copper mine, and contrast their results with those from a two-stage model.

In a related vein, the literature of the RCPSP under uncertainty focuses on activity durations while neglecting that in profits and costs. Most of this literature assumes that the uncertain duration of the activities are random variables with a known (joint) probability distribution, with corresponding modeling frameworks broadly classified as *proactive* and *reactive* [Herroelen and Leus, 2004, 2005; Demeulemeester and Herroelen, 2011; Ortiz-Pimiento and Diaz-Serna, 2018]. In proactive scheduling, the focus is on computing an initial *baseline schedule* that is protected as much as possible against future uncertainty. On the other hand, the goal of reactive scheduling is to amend a given baseline schedule or to create a schedule in real time, as uncertainty is revealed and activities and resources become available. Corresponding solution techniques are tested on instances having very few activities (i.e., fewer than 120). Moreover, almost no solution method integrates the proactive and reactive approaches. The only exception to the latter restriction is Davari and Demeulemeester [2019], but with the limitation that the problem is modeled as a Markov Decision Process over a very restricted set of possible reactive policies.

Our approach consists of an integrated proactive-reactive approach for mine scheduling under stochastic activity duration and ore grade uncertainty. It generates an initial baseline schedule that seeks to maximize the expected NPV of the mine, and initially assigns to each activity a fixed time interval in which such an activity must start to be executed for all possible realizations (i.e., scenarios) of stochastic duration and ore grade. The only constraint imposed on the reactive policy is that each activity must start within its corresponding baseline time interval. Therefore, our approach is integrated since the (proactive) baseline schedule and the reactive policy are both jointly determined. To the best of our knowledge, our work is the first in the literature to propose the concept of baseline schedule *with associated time intervals*, which lends a different interpretation to the resulting schedule; specifically, having a time interval for each activity provides the practitioner with a global view of the project's execution times.

The optimization model that must be solved to obtain a baseline schedule and its associated reactive policy corresponds to a multi-stage stochastic integer programming (MSIP) problem; decision-dependent uncertainty is revealed as decisions are made. Adding constraints linking randomness and decisions complicates the problem to such an extent that its solution must usually be derived using heuristics [Jonsbråten et al., 1998]. The challenge is to maintain non-anticipativity; that is, decisions for scenarios that are indistinguishable up to some time period t must be the same for all time periods up to t. Building on ideas from Goel and Grossmann [2004], we propose a novel multi-stage stochastic integer program for our mine scheduling problem that can handle non-anticipativity constraints in the context of decision-dependent uncertainty.

Two-stage stochastic integer programming problems, which are a simpler subclass of MSIP problems, are challenging to solve due to the absence of convexity and to the presence of discontinuities in the expected cost function [Ahmed, 2010]. Several corresponding algorithms have been proposed, usually leveraging problem structure, e.g., simple recourse [Haneveld et al., 1995], continuous second-stage variables [Liu et al., 2009], and binary variables [Ntaimo, 2010]. We refer the reader to several excellent surveys [Haneveld and van der Vlerk, 1999; Schultz et al., 1996; Carøe and Schultz, 1999; Küçükyavuz and Sen, 2017].

Given the complexity of our MSIP problem and the large size of mine scheduling problem instances, we do not attempt to solve the monolith directly. Rather, we propose a three-step optimization-based heuristic that proceeds as follows. First, we relax the non-anticipativity and integrality constraints, and then we solve the resulting two-stage stochastic *linear* programming problem. Second, based on the optimal solution of the problem solved in the first step, we create a priority list of activities. Finally, using the priority list, we build the baseline schedule and create the corresponding reactive policy. Two-stage stochastic linear programming problems have been traditionally solved by the L-shaped method [Van Slyke and Wets, 1969] and variations. We propose an alternative approach that takes advantage of the RCPSP structure of our problem. Specifically, we use a decomposition approach [Muñoz et al., 2018] based on Bienstock and Zuckerberg [2010].

3. Modeling

We begin by conceptualizing our MSIP problem. Then, we model uncertain problem parameters as random variables; the probability distributions of these random parameters are estimated from geological and geotechnical data. Finally, we formally state the problem, i.e., we define the instance, decision variables, objective function, and constraints that form our stochastic mine scheduling problem.

We develop an integer-programming model that seeks to determine when, if ever, various activities in an underground mine start so as to maximize net present value subject to precedence constraints between the activities, and resource constraints associated with the execution of any given set of activities in a particular time period. However, rather than solving the deterministic formulation, as given, e.g., in Brickey [2015] or Brickey et al. [2019], we add uncertainty in the forms of: (i) the amount that the execution of an activity contributes to the net present value, and (ii) the duration required to execute any given activity. The solution to this new stochastic programming model defines an interval in which activities start, rather than a precise moment in time, and constitutes an integrated proactive-reactive approach by generating an initial baseline schedule with the reactive policy that each activity must start within its corresponding baseline time interval. Such solutions have an advantage over those generated by alternative approaches in the literature: time-interval width, δ, provides the decision maker a control on the variability of the starting times of the activities. We define the following notation:

Sets

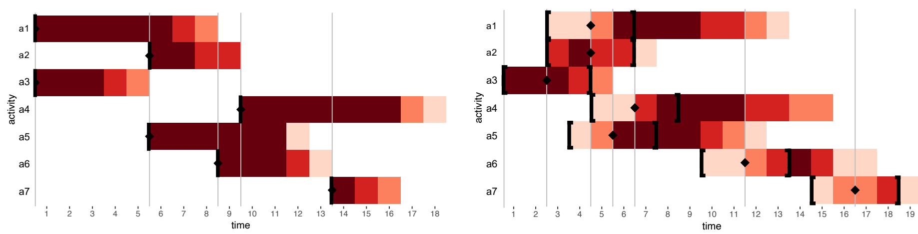
symbol	definition
$\mathcal{A}$	all activities
Ω	sample space, ordered by scenario: 1, 2, ..., $\lvert\Omega\rvert$

Parameters

symbol	definition	[units]
$\delta \in \mathbb{N}_0$	time-interval width	[time periods]

The following example illustrates. Let us consider the scheduling of seven activities, i.e., $|\mathcal{A}| = 7$, over a sample space consisting of four scenarios, i.e., $\Omega = \{1, 2, 3, 4\}$. Furthermore, we alternatively examine the cases of $\delta = 0$ and $\delta = 2$. The remaining components of the instances (which we define when presenting our complete mathematical formulation) are identical in both cases and are irrelevant for the purpose of this example.

Figure 1 contains a graphical representation of optimal solutions for the cases $\delta = 0$ and $\delta = 2$, where the shades demonstrate that the execution times of each activity (i.e., the time periods at which the activity is under execution) present less variability in case $\delta = 0$ than in case $\delta = 2$. For $\delta = 0$, the variability of the execution times is explained only by the variability of the parameters (duration) because each activity starts at the same time period (represented by the black diamond in the Figure 1a) in all scenarios. Alternatively, for the case in which $\delta = 2$, activities do not necessarily begin at the baseline start time; the higher variability in the execution times is explained both by the variability in the parameters and by the variability in the starting times.



(a) Optimal solution $\delta = 0$

(b) Optimal solution $\delta = 2$

Figure 1: The square brackets correspond to the time intervals associated with the baseline schedules; a colored square represents the corresponding activity being under execution at a given time period, and darker squares indicate that more scenarios cast an activity as in progress at that time. The optimal baseline schedule is represented by black diamonds.

A solution that provides less variability in the starting times, and therefore less variability in the execution times, is attractive to the decision maker: it allows for anticipation regarding the time at which activities will be carried out. Nevertheless, less variability in the starting times requires smaller values of δ, which implies a lower expected net present value of the schedule. Consequently, there is a trade-off between starting-time variability (controlled by δ) and expected net present value of the schedule.

3.1. Representation of Uncertainty in Activity Value and Duration

Geological uncertainty is inevitable with widely spaced drill holes [Koushavand et al., 2014] from which geological information is gained to construct a block model, and represents the inability to accurately represent the grade, geologic boundaries, or other conditions of a rock mass [Bruno, 2019, Chapter 12].

This uncertain information is used to define activities and their associated characteristics such as ore content and resource requirements for their execution, and gives rise to two uncertain parameters of interest in our application: economic and geotechnic. The economic value of completing an activity depends on the mineralogical properties of the rock (such as grade concentration, rock hardness, grain size, and oxidation intensity), the capability of the mining operation (such as equipment capacity and market demand), and the metallurgical efficiency of the milling process, *inter alia*. Matheron [1962] provides foundations for applying statistical techniques to mineral resource reserve estimation and grade control. The related procedure using stochastic or geostatistical simulation is mature and well established [Goovaerts, 1997]. Block models record estimated grade for each unit of a spatially discretized orebody, and are often a product of a simulation. While the procedure is valid, we seek an improvement by exploiting all available information.

Geotechnic uncertainty, which arises from the inability to accurately estimate the quality of the rock, i.e., strength, composition, and structure, has a direct effect on an activity's duration. Specifically, because rock masses can be unbroken (at one extreme) or highly fractured (at the other), impacting their strength, the amount and type of resources to develop the necessary underground infrastructure can vary considerably, and sometimes unpredictably. Ground control mitigates poor rock quality through engineering protocols such as roof bolts, shockcrete, and other supports, and the extent to which this control must be implemented affects the time required to complete various activities [Darling, 2011, Chapter 8].

We describe the nature of both economic and geotechnical uncertainty, defining notation using the conventions that lower case letters are parameters and indices; upper case letters in calligraphic font are sets, and upper case letters in roman font are variables (see Teter et al. [2016] for guidelines). Hats and overbars differentiate sets that represent similar entities. Without loss of generality with respect to our optimization framework, we treat random parameters v_a^ω and d_a^ω as independent of each other.

Parameters

symbol	definition	[units]
$v_a^\omega \in \mathbb{R}$	value of completing activity $a \in \mathcal{A}$ in scenario $\omega \in \Omega$	[dollars]
$d_a^\omega \in \mathbb{N}$	duration of activity $a \in \mathcal{A}$ in scenario $\omega \in \Omega$	[time periods]

To estimate value parameters, v_a^ω, we use a standard geostatistical approach. We consider a continuously varying quantity over a spatial domain $D \subset \mathbb{R}^3$, and model the estimates as a Gaussian Process, defined by the property that any finite combination of observations from D follows a multivariate normal distribution. Within this framework, we use a procedure based on the Cholesky decomposition of the data variance-covariance matrix Σ to simulate values [Cressie, 1991]. Modeling the random duration parameters d_a^ω requires an ad-hoc approach given that the available data in typical mines consists of *estimates* with only one value for each activity.

3.2. Problem Statement

An instance of our problem requires the following additional inputs:

Sets

symbol	definition
$\mathcal{P}_a \subseteq \mathcal{A}$	activities that must be completed before activity $a \in \mathcal{A}$ can start
$\mathcal{R}$	resource types required to execute activities
$\mathcal{T}$	time periods

Parameters

symbol	definition	[units]
$\gamma \in \mathbb{R}^+$	value discount factor per time period	[–]
$\bar{q}_r \in \mathbb{R}^+$	quota of resource $r \in \mathcal{R}$ available (renewable)	[units/time period]
$q_{ar} \in \mathbb{R}^+$	amount of resource $r \in \mathcal{R}$ consumed by activity $a \in \mathcal{A}$	[units/time period]

Regarding random value and random duration of each activity, we note that, aside from very special cases, MSIP problems cannot be solved explicitly for arbitrary probability distributions due to the presence of the expected multi-stage cost. A common approach is to replace the expected value by an average over a finite set of scenarios using the Sample Average Approximation [Kleywegt et al., 2002; Linderoth et al., 2006] by generating samples from the distribution of the random elements. Consequently, we assume that the sample set Ω is finite and that the probability of occurrence of each scenario is equal to $1/|\Omega|$. We also assume that the number of time periods in $\mathcal{T}$ corresponds to the set of consecutive natural numbers between 1 and T, where T is a given natural number. The decision variables of the problem are the following:

Variables

symbol	definition
X_{at}^{ω}	1 if activity a starts at time period t in scenario ω, 0 otherwise
Y_{at}	1 if the baseline start time of activity a is equal to time period t, 0 otherwise
$Z_t^{\omega\omega'}$	0 if $X_{at}^{\omega} = X_{at}^{\omega'}$ for activity a, time period t, and scenario pair $\{\omega, \omega'\}$

Finally, our MSIP problem can be stated as follows.

$$
(\mathbf{S}) \quad \max_{X^{\omega}_{at},\, Y_{at},\, Z^{\omega\omega'}_{t}} \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \sum_{t\in\mathcal{T}} \sum_{a\in\mathcal{A}} \gamma^{t+d^{\omega}_a-1} v^{\omega}_a X^{\omega}_{at} \tag{1}
$$

$$
\text{s.t.} \sum_{t\in\mathcal{T}} X^{\omega}_{at} \le 1 \qquad a\in\mathcal{A}; \omega\in\Omega \tag{2}
$$

$$
\sum_{t'=1}^{t} X^{\omega}_{at'} \le \sum_{t'=1}^{t-d^{\omega}_{a'}} X^{\omega}_{a't'} \qquad a\in\mathcal{A};\ a'\in\mathcal{P}_a;\ t\in\mathcal{T};\ \omega\in\Omega \tag{3}
$$

$$
\sum_{a\in\mathcal{A}} \sum_{t'=\max\{1,t-d^{\omega}_a+1\}}^{t} q_{ar}\, X^{\omega}_{at'} \le \bar{q}_r \qquad r\in\mathcal{R};\ t\in\mathcal{T};\omega\in\Omega \tag{4}
$$

$$
\sum_{t'=1}^{t} Y_{at'} \le \sum_{t'=1}^{\min\{T,t+\delta\}} X^{\omega}_{at'} \qquad a\in\mathcal{A};\ t\in\mathcal{T};\ \omega\in\Omega \tag{5}
$$

$$
\sum_{t'=1}^{t} X^{\omega}_{at'} \le \sum_{t'=1}^{\min\{T,t+\delta\}} Y_{at'} \qquad a\in\mathcal{A};\ t\in\mathcal{T};\ \omega\in\Omega \tag{6}
$$

$$
Z^{\omega\omega'}_t \le \sum_{a\in\mathcal{D}^{\omega\omega'}} \left(\sum_{t'=1}^{t-d^{\omega}_a} X^{\omega}_{at'} + \sum_{t'=1}^{t-d^{\omega'}_a} X^{\omega'}_{at'} \right) \qquad t\in\mathcal{T};\omega,\omega'<\omega\in\Omega \tag{7}
$$

$$
X^{\omega'}_{at} - Z^{\omega\omega'}_t \le X^{\omega}_{at} \le X^{\omega'}_{at} + Z^{\omega\omega'}_t \qquad a\in\mathcal{A};\ t\in\mathcal{T};\omega,\omega'<\omega\in\Omega \tag{8}
$$

$$
X^{\omega}_{at}, Y_{at}, Z^{\omega\omega'}_t \ \text{binary} \qquad a\in\mathcal{A};\ t\in\mathcal{T};\ \omega,\omega'<\omega\in\Omega \tag{9}
$$

The objective, represented by (1), is to maximize the expected net present value of the reactive policy; constraints (2) allow activities to start within the time horizon, and only once, if at all. Constraints (3) impose the precedence constraints: activities can start only after their predecessors have been completed. Constraints (4) enforce resource constraints: the resources consumed in each time period cannot exceed the amount of available resource. Constraints (5) and (6) impose the time-interval condition: activities must start within δ time periods of the baseline for all scenarios. Finally, constraints (7) and (8) enforce the non-anticipativity constraints: if a pair of scenarios are indistinguishable up to a given time period, then the schedule for those scenarios must coincide up to that time period. In particular, given a pair of scenarios ω and ω', the set of *differentiating activities*, $\mathcal{D}^{\omega\omega'} \subseteq \mathcal{A}$, contains the activities that allow us to differentiate both scenarios, i.e., those activities that have different duration and/or value for scenarios ω and ω'. Formally, this set (see Goel and Grossmann [2004]) is defined as follows:

$$\mathcal{D}^{\omega\omega'} = \{a \in \mathcal{A} : d_a^{\omega} \neq d_a^{\omega'} \vee v_a^{\omega} \neq v_a^{\omega'}\}.$$

.

Note that constraints (7) and (8) depend on the information gained through completing activities. Specifically, when $Z_t^{\omega\omega'} = 0$, then constraint (8) forces $X_{at}^{\omega} = X_{at}^{\omega'}$; and, when $Z_t^{\omega\omega'} = 1$, constraint (8) is void. For the sake of simplicity, we refer to these constraints as non-anticipativity constraints, despite the fact that they are *conditional* non-anticipativity constraints, and should not be confused with the traditional non-anticipativity constraints found in standard stochastic programming textbooks [Birge and Louveaux, 2011]. The following example shows how non-anticipativity constraints (7) and (8) distinguish scenarios.

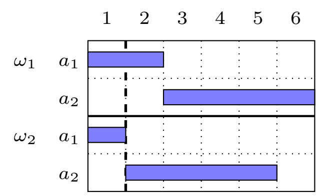
Example: We assume that there are two activities (a_1 and a_2) and two scenarios (ω_1 and ω_2). The time limit of the project, T, is equal to 6. Moreover, there is only one unit of resource available per time period, and activities consume one unit of resource per time period of execution. For simplicity, we assume that: (i) we need not create a baseline schedule; and, (ii) there are no precedence constraints. Table 1 provides the value and duration for each activity in each scenario.

	v_a^{ω}		d_a^{ω}	
	ω_1	ω_2	ω_1	ω_2
a_1	1	1	2	1
a_2	1	1	4	4

Table 1: We provide the value, v_a^{ω}, and duration, d_a^{ω}, for each activity and scenario in the example.

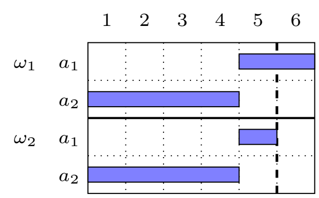
Let us consider two execution policies: in the first, activity a_1 is executed first, followed by activity a_2; in the second, the reverse. Figure 2 represents both execution policies.

t		1	2	3	4	5	6
ω_1	a_1	1	0	0	0	0	0
	a_2	0	0	1	0	0	0
ω_2	a_1	1	0	0	0	0	0
	a_2	0	1	0	0	0	0



(a) Policy 1: X_{at}^{ω} values and corresponding Gantt chart

t		1	2	3	4	5	6
ω_1	a_1	0	0	0	0	1	0
	a_2	1	0	0	0	0	0
ω_2	a_1	0	0	0	0	1	0
	a_2	1	0	0	0	0	0



(b) Policy 2: X_{at}^{ω} values and corresponding Gantt chart

Figure 2: Here, we provide the Gantt charts and tables showing time along the x-axis. Activity a_1 and a_2 share resource r_1, resulting in two possible policies. In a policy, each scenario has its own schedule, and schedules within a policy are identical to the left of the dotted line.

We note that the set of differentiating activities is composed only by activity a_1, i.e., $\mathcal{D}^{\omega_1\omega_2} = \{a_1\}$. Let us first analyze Policy 1. Looking at its corresponding Gantt chart, we see that, from the beginning of time period 2 on, both scenarios are differentiated since activity a_1 is completed at the end of time period 1 in scenario ω_2. Thus, constraints (7) impose $Z_1^{\omega_1\omega_2} = 0$. This fact, together with constraints (8), implies that $X_{a_1 1}^{\omega_1} = X_{a_1 1}^{\omega_2}$ and $X_{a_2 1}^{\omega_1} = X_{a_2 1}^{\omega_2}$. On the other hand, constraints (7) imply that $Z_{t'}^{\omega_1\omega_2}$ is equal to 0 or 1 for the remaining time periods $t' = 2, \ldots, 6$. Therefore, constraints (8) are inactive, which implies that $X_{a_1 t'}^{\omega_1}$ is not necessarily equal to $X_{a_1 t'}^{\omega_2}$ for each time period $t' = 2, \ldots, 6$. Analogously, $X_{a_2 t'}^{\omega_1}$ is not necessarily equal to $X_{a_2 t'}^{\omega_2}$ for each time period $t' = 2, \ldots, 6$.

Now, let us analyze Policy 2. In its Gantt chart, we see that differentiating activity a_1 is completed only at the end of time period 5 in scenario ω_2. Therefore, constraints (7) imply that $Z_{t'}^{\omega_1\omega_2} = 0$, and, by constraints (8), $X_{a_1 t'}^{\omega_1} = X_{a_1 t'}^{\omega_2}$

and $X^{\omega_1}_{a_2 t'} = X^{\omega_2}_{a_2 t'}$ for each time period $t' = 1, \ldots, 5$. At time period 6, $Z^{\omega_1 \omega_2}_6$ is equal to 0 or 1, then constraints (8) are inactive. Thus, $X^{\omega_1}_{a_1 6}$ is not necessarily equal to $X^{\omega_2}_{a_1 6}$, and $X^{\omega_1}_{a_2 6}$ is not necessarily equal to $X^{\omega_2}_{a_2 6}$.

4. Solution Methodology

Instances of problem (**S**) cannot be solved in polynomial time (under the assumption that P $\neq$ NP). The RCPSP is known to be NP-hard [Blazewicz et al., 1983], and reduces to (**S**) with $|\Omega|=1$ (and, therefore, without constraints (5), (6), (7) and (8)). Realistic instances of tactical underground mining problems are large, often including thousands of activities, hundreds of time periods, and multiple scenarios, making it impossible to solve (**S**) in an operationally feasible amount of time (e.g., hours) by directly applying a standard mixed-integer programming solver to the monolith. Furthermore, *ad hoc* algorithms designed for scheduling problems with deterministic parameters exploit structure that is absent in our multiple-scenario case.

It is possible to strengthen (**S**) by reformulating constraint (7); for example, the number of terms on its right-hand side could be reduced by including only the more limiting of the two summations based on activity duration. Another potential formulation enhancement sums constraint (7) over a and its union of predecessors. While valid and potentially useful, numerical results indicate that the linear relaxation of our proposed formulation is tight; at any rate, the first suggestion increases the density of the constraint set. On the other hand, preliminary numerical testing indicates that the RAM storage requirements (which grow with the density of the constraint matrix) are more limiting than the quality of the linear programming relaxation. Modeling conditional non-anticipativity requires constraints which are theoretically necessary to craft solutions given our multi-scenario setting. However, the number and density of these constraints, specifically, constraints (7) and (8), contribute significantly to the difficulty of solving (**S**). We therefore initially relax these constraints, calling the resulting problem ($\mathbf{S}^-$). Not only does this relaxation remove “difficult” constraints, it reduces the model to one with an RCPSP-like structure, amenable to solution via an academic research solver. Specifically, OMP SOLVER [Rivera et al., 2015] is capable of quickly finding the optimal solution to the linear relaxation of problem ($\mathbf{S}^-$) for realistically sized instances by using a tailored linear programming algorithm [Muñoz et al., 2018] which decomposes the problem [Bienstock and Zuckerberg, 2010]. We refer to the linear programming relaxation of problem ($\mathbf{S}^-$) as ($\mathbf{LS}^-$).

Table 2 describes the linear programming-based heuristic $\mathcal{H}$ in three steps. First, $\mathcal{H}_1$ solves the linear program, ($\mathbf{LS}^-$). Let (X', Y') be the corresponding optimal solution. Second, $\mathcal{H}_2$ conducts a SIMPLE SORT as follows: (i) determine a *mean baseline start time*, $MS_a = \sum_{t \in \mathcal{T}} tY'_{at}$, for each activity $a \in \mathcal{A}$; (ii) discard all activities $a \in \mathcal{A}$ with $MS_a < 0.5$; and, (iii) sort, non-decreasing by mean start time, to produce a PRIORITY LIST (Appendix A, Algorithm 1). Third and finally, $\mathcal{H}_3$, given the priority list, applies a list-scheduling heuristic (Appendix A, Algorithm 2) in order to obtain a feasible solution, (X, Y), for problem ($\mathbf{S}$). A proof of correctness of Algorithm 2 is found in Appendix B.

Phase	Input	→	Algorithm	→	Output
$\mathcal{H}_1$	Instance of ($\mathbf{S}$)		BIENSTOCK-ZUCKERBERG		(X', Y')
$\mathcal{H}_2$	(X', Y')		SIMPLE SORT (See Algorithm 1, Appendix A)		PRIORITY LIST
$\mathcal{H}_3$	PRIORITY LIST		LIST SCHEDULING (See Algorithm 2, Appendix A)		(X, Y)

Table 2: We give a description of the heuristic by phases, which includes inputs, algorithms, and outputs. Phase $\mathcal{H}_3$ produces a feasible solution, (X, Y).

5. Case Study

The case study for this investigation is a United States-based, large-scale underground mine at which annual production is approximately 1.8 million tons of material (ore and waste) and 370,000 troy ounces of gold [Brickey, 2015]. The mine uses an underground stoping method that consumes five resources (see Table 3) associated with development, extraction, backfill, and other ancillaries. We use a value of 1 time period for δ and a daily, i.e., per time period, discount rate of 0.02%. Each of the 15,773 activities has (i) a type, (ii) precedence and resource requirements, (iii) a `value` (which can be negative) and (iv) a `duration`. We describe first how we generate scenarios based on attributes (iii) and (iv) to populate instances of our MSIP problem, (**S**), and then how we solve it via the method outlined in Table 2.

Constraint	Constrained Activity Types	Upper Bound	Units
Total tonnage	Development, mining and all backfill	11,000	[tons/day]
Total tonnage	Cement and paste backfill	5,000	[tons/day]
Total tonnage	Unconsolidated rock backfill	2,500	[tons/day]
Ore tonnage	Development and mining	6,000	[tons/day]
Footage	Development	155	[feet/day]
Concurrent activities	Vertical development	1	[activity/day]

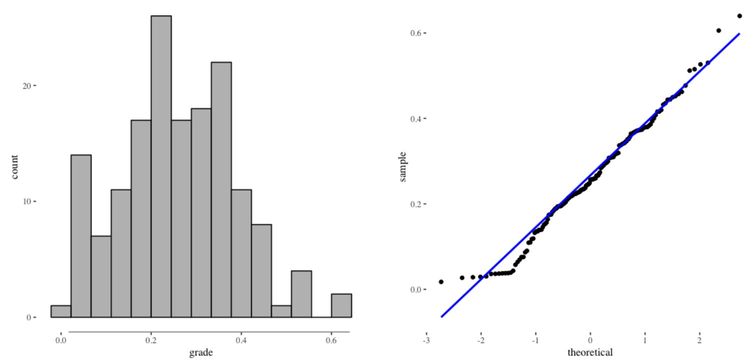
Table 3: Five resources adapted from our case study constrain activity completion.

5.1. Scenario Development

Activity grade is derived from simulations of the gold concentration in the orebody given borehole sample data; the feature `grade` represents the concentration of gold estimated in troy ounces per ton, and yields a way to compare concentrations of gold over space because, for each activity, the feature accounts for the mass of rock to be mined. We restrict for which activities to model uncertain `grade` and for which to hold their values constant. Grade values used to calculate the revenue component of `value` (from the sale of gold extracted) are adjusted from the block model values, which are based on the physical estimated value of gold in the orebody, and incorporate recovery rates associated with mining and processing.

To model value, we only consider activities associated with mining-specific types, i.e., we do not consider development or ancillary activities. These are Stope-Mining, Up-Hole, Cut-Fill, and Floor-Pull. Of the original 15,773 activities, this leaves 1,509. We further limit this number to high-grade activities based on the assumption that the majority of the grade uncertainty lies in this set. This further reduces the set to 159 activities. Let $\{\boldsymbol{s}_1, ..., \boldsymbol{s}_{159}\} \in \mathbb{R}^3$ be the locations of the data and $\{\widehat{v}(\boldsymbol{s}_1), ..., \widehat{v}(\boldsymbol{s}_{159})\}$ be the values of `grade` observed at those locations.

Figure 3 shows that $\{\widehat{v}(\boldsymbol{s}_1), ..., \widehat{v}(\boldsymbol{s}_{159})\}$ appears within a tolerance of normality to accept the Gaussian Process assumption as a model for these data. We conduct a formal test for spatial dependence with Moran's I-score [Moran, 1950], a type of correlation coefficient which measures spatial dispersion or correlation present in a data set based on observation proximity. We can formally check for spatial dependence by testing a null hypothesis of purely random spatial observations. Figure 4a shows Moran's I-score as a function of the number of neighbors, which we determine to be 0.50 with $k = 3$ neighbors, suggesting moderate spatial autocorrelation. For each number of neighbors k, the Moran's I-score tests as significant. We then center the data to form a mean-zero Gaussian Process.



(a) Histogram of `grade` **(b)** Normal Q-Q Plot

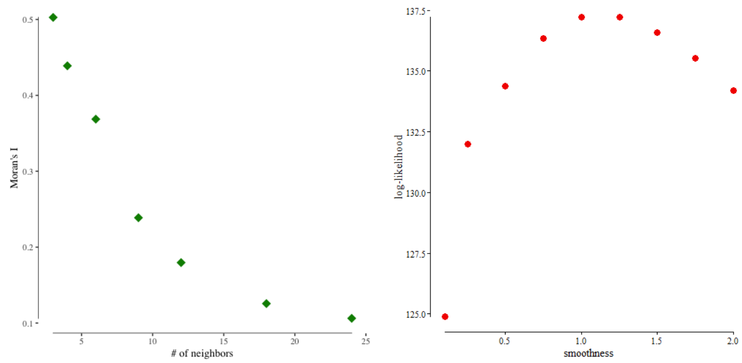
Figure 3: We show the normality of `grade`, and provide a histogram of 159 `grade` observations used for simulations, displayed in fifteen bins. We also depict the Normal Quantile-Quantile (Q-Q) Plot of `grade` with the theoretical reference line superimposed in blue.

We determine whether the resulting mean-zero Gaussian Process forms a second-order stationary random field. The stationarity assumption must be checked to validate subsequent analysis and to produce accurate simulations although, in practice, it is almost always an approximation. Bandyopadhyay and Rao [2017] provide a method for evaluating the presence of non-stationarity with irregularly spaced spatial data, which uses a Discrete Fourier Transform of the observations. If the resulting Fourier coefficients are "nearly uncorrelated," then the underlying spatial process is second-order stationary; otherwise, this property does not hold. We pose a null hypothesis that $v(\cdot)$ is a second-order stationary random field; tests yield a statistic of 6.60 with a corresponding p-value of 0.22. We therefore fail to reject the null hypothesis, and maintain the stationarity assumption.

We investigate appropriate covariance functions to model the centered data. A classic family consists of the Matérn covariance functions. While flexible, they depend upon a collection of estimated parameters: The smoothness parameter, ν, is particularly difficult to estimate directly from the data, so instead we evaluate the performance of a set of Matérn covariance functions for a range of chosen values for ν: 0.10, 0.25, 0.50, 0.75, 1.00, 1.25, and 1.50. Figure 4b shows the log-likelihood of a Matérn covariance function for these values. The maximum log-likelihood occurs where $\nu = 1.25$; however, a close second maximum occurs where $\nu = 1$. In fact, the log-likelihood values for each of these choices of ν agree up to two decimal places, and so, in practice, would perform quite similarly. Given these two options, we select $\nu = 1$ because: (i) a Matérn covariance function with smoothness ν assumes that the underlying spatial field is $\lceil \nu - 1 \rceil$ times differentiable (implying that with $\nu = 1$, there is no differentiability assumption); and, (ii) taking $\nu = 1$ with a Matérn covariance function is a special case known as a Whittle covariance function [Guttorp and Gneiting, 2006]. Because we are reverse-engineering the simulation process that gave rise to the values of `grade` present in our data, it seems more likely that the simulators would choose a Whittle covariance function over setting $\nu \approx 1.25$ given its popularity in geostatistical applications. A Whittle covariance function is also dependent upon a range parameter; checking a fine grid yields $\theta = 54$ ft to maximize the likelihood.

With our chosen covariance function, we construct the variance-covariance matrix Σ. We then use the Cholesky decomposition method to simulate `grade` across the spatial field [Cressie, 1991]. This method is valid for general multivariate Gaussian random variables and does not require a stationary or isotropic covariance function.

We now turn our attention to the second source of uncertainty, that asso-



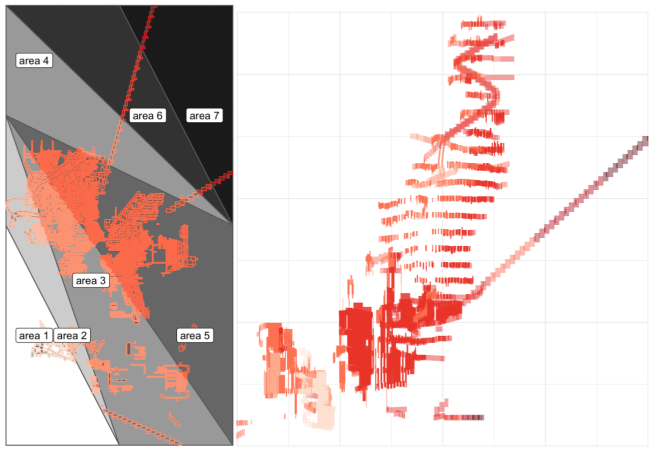
(a) Moran's I-score as a function of the number of neighbors.

(b) Log-likelihood as a function of smoothness.

Figure 4: We show Moran's I-score as a function of the number of neighbors, and the log-likelihood as a function of smoothness assuming a Matérn covariance function over the centered `grade` data.

ciated with an activity's `duration`. Related to this, we note that there are six rock densities reflecting the different rock types present in the mine, which is partitioned into seven regions such that each region is labeled a *ground risk area* (see Figure 5). Incorporating information regarding these qualitatively different areas of the mine into our duration simulations enables us to account for geotechnical uncertainty. There is a unique observation for each activity in each geological risk area in our data set.

For a scenario $\omega \in \Omega$ and activity $a \in \mathcal{A}$, we model duration as $d_a^\omega = \lfloor \widehat{d}_a + \beta_a^\omega \rceil$, where $\widehat{d}_a$ is the duration of activity a in the data set used in a deterministic model derived from industry standards and β_a^ω accounts for variability associated with geotechnical uncertainty. (Note that the rounding function, which we denote $\lfloor \cdot \rceil$, should equal at least 1, because we assume that the duration is a positive integer – for compatibility with the way in which our integer program handles time fidelity.) Let f_g be a scaling factor representing the "worst-case" duration increase resulting from an activity occurring in a ground risk area g. For each scenario $\omega \in \Omega$ and ground risk area g, we generate $U_g^\omega \sim U[-1, 1]$ and define $\beta_a^\omega = f_g \widehat{d}_a U_g^\omega$. We make this modeling decision because, within a window for a given activity duration, we assume all other



(a) Design top view **(b)** Design side view

Figure 5: Given a design, the location of activities can be mapped to ground risk areas g through expert analysis of borehole data.

durations are equiprobable. An additional benefit is that the expected value of β_a^ω is zero, in which case we recover in expectation the initial duration estimate $\widehat{d}_a$.

We incorporate only a modest number of scenarios (five), commensurate with the intuition of mine operators and on par with the resulting size of instances in the literature (based also on the number of activities and length of time horizon) [Leite and Dimitrakopoulos, 2007; Consuegra and Dimitrakopoulos, 2010]. Using these scenarios, we demonstrate how our procedure yields solutions in an operationally feasible amount of time, whereas a straightforward application of a state-of-the-art solver to the monolith solves only the smallest instance. Then, we compare solution quality of the stochastic programming model to that of a deterministic problem, (**D**); the latter assumes a single scenario with means for value and duration of each activity $a \in \mathcal{A}$ over all scenarios, given as $\bar{v}_a$ and $\bar{d}_a$, respectively.

5.2. Results

All experiments were run using the following hardware: a Sun Fire X2270 M2 with two Intel Xeon X5675 processors at 3.07 GHz, 48 GB RAM and under the Ubuntu 18.04 operating system. All algorithms were implemented in Julia 1.0.5 and run in serial mode, i.e., on a single thread, to avoid memory overflows. The exact solution of all linear and integer programming problems were obtained with CPLEX 12.10.0.0 through its Julia interface, JuMP.

In order to test the efficacy of our heuristic and the quality of the solutions it provides relative to (i) solving the deterministic equivalent and (ii) solving the stochastic program in its monolith form, we present a variety of numerical experiments (Table 4).

Problem	Solution technique	Method	Treatment of uncertainty	Non-anticipativity	$\lvert\Omega\rvert$	Value	Duration
(**S**)	exact	CPLEX	stochastic	yes	5	v_a^ω	d_a^ω
(**S**$^-$)	exact	CPLEX	stochastic	no	5	v_a^ω	d_a^ω
(**D**)	heuristic	$\mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$	deterministic	NA	1	$\bar{v}_a$	$\bar{d}_a$
(**S**)	heuristic	$\mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$	stochastic	yes	5	v_a^ω	d_a^ω

Table 4: We conduct numerical experiments on these four problem variants using the methods outlined in the third column and with the problem characteristics given in the remaining columns.

Table 5 reports solution times. Solving the monolith directly for (**S**), and even for (**S**$^-$), i.e., without the non-anticipativity constraints, is only possible for the smallest instance, i.e., that containing 56 activities, and requires an order of magnitude more time than our proposed heuristic (§4). For the smallest three instances, solutions from the stochastic program require more time to obtain than those from the deterministic equivalent; however, as problem size increases, solving the linear-programming relaxation can be as difficult for the deterministic as the stochastic case. In the largest instances, both cases reach a time limit. In all instances, solution times are dominated by the execution of the Bienstock-Zuckerberg algorithm, and fall within ten hours (even if an eight-hour time limit for the execution of the Bienstock-Zuckerberg algorithm is reached for the largest three); nonetheless, we obtain good quality solutions.

Instance		Solution Times			
		Exact Solve		$\mathcal{H}_1+\mathcal{H}_2+\mathcal{H}_3$	
$\lvert\mathcal{A}\rvert$ [activities]	T [days]	$(\mathbf{S})$ [sec]	$(\mathbf{S}^-)$ [sec]	$(\mathbf{D})$ [sec]	$(\mathbf{S})$ [sec]
56	50	41	29	1	1
396	200	†	†	8	16
646	300	†	†	57	109
1,453	600	†	†	1,507	1,231
2,323	900	†	†	6,293	4,466
3,150	1,200	†	†	15,723	14,987
3,828	1,500	†	†	28,495	33,769
4,330	1,800	†	†	30,323	32,731
4,764	2,100	†	†	31,017	31,076

†Exceeds available computer memory

Table 5: We provide solution times for the problems given in Table 4. All instances contain five scenarios ($|\Omega|$=5) and five resources ($|\mathcal{R}|$=5). Solutions for the linear programming relaxation found via the Bienstock-Zuckerberg algorithm (used in the heuristic) subscribe to the following termination criterion: **minimum** {eight hours of computation time, a duality gap of less than or equal to 0.01%}.

Table 6 shows the expected net present value, given by (1), for each case listed in Table 4. The smallest instance, which is solvable via both exact and heuristic methods, demonstrates equal objective function values. We note that the linear programming relaxation objective function values tend to be particularly tight for the following three reasons. First, despite the fractionation in the LP relaxation solution, activities are usually completed by the end of the horizon; so, the effect of the fractionation is that parts of activities can be moved around in the schedule, rather than "canceled" altogether. This movement affects the objective function value only by the discount factor, which, for our daily-fidelity model, is 0.02%. And, many fractionated activities tend to be adjacent in time because of the precedence constraints, implying that large costs cannot be relegated to the end of the horizon at an unrealistically highly discounted value. Second, our objective function lacks large fixed charges; correspondingly, the values assessed are all associated with the same "type" of decision variable, namely, the execution of an activity. Third, in the optimal integer-programming solution, the resource constraints tend to be reasonably tight; that is, the linear programming solution is not able to "cram" fractions into the bottleneck resource constraints in a way that the integer solution precludes.

In fact, TopoSort handles the resource constraints well; other authors have observed this same behavior with similarly structured problems, e.g., Brickey et al. [2019]. Furthermore, this performance extends to instances with as many as 30 scenarios, which we can solve to within 3.06% of optimality within about thirty hours of computation time, even for the largest instances tested here (where the bulk of the computation is spent in input and output procedures).

The objective function values for all instances show negligible differences between those produced by the deterministic (**D**) versus stochastic (**S**) and ($\mathbf{LS}^-$) models. This might suggest that incorporating stochasticity is not important. However, we further analyze the solutions via three metrics: *makespan*, *feasibility*, and *expected count of completed activities*, and conclude that the results from the stochastic program are more realistic, and therefore implementable in a production setting, while not sacrificing significant objective function value.

Instance		Objective Function Values					Optimality Gap
		Exact Solve		$\mathcal{H}_1+\mathcal{H}_2+\mathcal{H}_3$		BZ	$\left(\frac{\hat{Z}_{(\mathbf{LS}^-)}-Z_{(\mathbf{S})}}{Z_{(\mathbf{S})}}\right)^{\ddagger}$
$\lvert\mathcal{A}\rvert$ [activities]	T [days]	(**S**) [\$M]	($\mathbf{S}^-$) [\$M]	(**D**) [\$M]	(**S**) [\$M]	($\mathbf{LS}^-$) [\$M]	[%]
56	50	0.96	0.96	0.96	0.96	0.96	0.02
396	200	†	†	14.79	14.78	14.78	0.03
646	300	†	†	33.14	33.12	33.12	0.01
1,453	600	†	†	101.00	100.69	100.75	0.07
2,323	900	†	†	187.90	186.84	186.99	0.09
3,150	1,200	†	†	267.24	265.05	265.35	0.12
3,828	1,500	†	†	335.66	332.20	332.66	0.19
4,330	1,800	†	†	375.02	370.42	370.94	0.16
4,764	2,100	†	†	409.83	403.50	404.35	0.56

†Exceeds available computer memory

Table 6: We provide expected net present value for the problems given in Table 4. All instances contain five scenarios ($|\Omega|$=5) and five resources ($|\mathcal{R}|$=5).
‡$\hat{Z}_{(\mathbf{LS}^-)}$ corresponds to an upper bound on the objective value of problem ($\mathbf{LS}^-$) obtained via the dual bound; $Z_{(\mathbf{S})}$ corresponds to the objective value of the best feasible solution found for problem (**S**).

In order to assess the quality of the solutions, we introduce a variety of metrics, the first of which is the *makespan*, given by μ and defined in Equation (10) as the last time period with an activity under execution, as follows:

$$\mu = \max_{a \in \mathcal{A}, \omega \in \Omega} \left\{ \sum_{t \in \mathcal{T}} t \cdot X_{at}^{\omega} + d_a^{\omega} - 1 \right\} \tag{10}$$

We also measure the *feasibility* of a schedule, which is necessarily satisfied for any solution of the stochastic programming models, (**S**) and ($\mathbf{S}^{-}$). For the deterministic model (**D**), feasibility implies, for the original five scenarios in Ω, the satisfaction of integrality, and constraints (3) and (4) (shown again here):

$$\sum_{t'=1}^{t} X_{at'}^{\omega} \leq \sum_{t'=1}^{t-d_{a'}^{\omega}} X_{a't'}^{\omega} \qquad a \in \mathcal{A};\ a' \in \mathcal{P}_a;\ \omega \in \Omega$$

$$\sum_{a \in \mathcal{A}} \sum_{t'=\max\{1, t-d_a^{\omega}+1\}}^{t} q_{ar} \cdot X_{at'}^{\omega} \leq \bar{q}_r \qquad r \in \mathcal{R};\ \omega \in \Omega$$

Invariably, there exists some time period(s) in which one or more of these precedence and/or resource constraints is not satisfied, and our measure ϕ is given as the last feasible time period in the schedule, i.e., the last time period before rescheduling is required to resolve the infeasibility:

$$\phi = \max_{\hat{t} \in \mathcal{T}} \left\{ \hat{t} \text{ such that (3) and (4) both hold for all } t \leq \hat{t} \right\} \tag{11}$$

That is, in the absence of scenario consideration that the stochastic program affords, the schedule in the "out years" cannot be executed without a correction policy; it is precisely the ambiguity of determining such a policy that motivates us to avoid re-solving model instances that become infeasible. The longer term schedules we produce are used to commit equipment and other resources throughout the planning horizon. So, postponing the schedule leads to a misallocation of resources whose resulting cost is difficult to quantify. Activities cannot be skipped because precedence would be violated. (For example, ore cannot be extracted from a stope before the area is drilled and the rock blasted.) Mine planners do reschedule, because unanticipated occurrences happen frequently. However, an optimization model affords a mine planner with the best possible action plan given the current information, and minimizing the number of rescheduling activities minimizes disruption to this plan.

Finally, unlike in typical project scheduling in which all activities are executed, activities are optional in an underground mine. Deterministic models have the clairvoyance not to schedule activities that offer little value. Equation (12) defines the measure η as the *expected count of completed activities*:

$$\eta = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{t \in \mathcal{T}} \sum_{a \in \mathcal{A}} X_{at}^{\omega} \tag{12}$$

We record each of these metrics, μ, ϕ, and η in Table 7 for the nine instances given in Tables 5 and 6 in both the deterministic, (**D**), and stochastic, (**S**), settings. Corresponding to intuition, the makespans are all longer for the stochastic programming solutions for which the corresponding model incorporates uncertainty from a variety of scenarios, resulting in some longer durations that (**S**) accommodates owing to feasibility requirements. As expected, the deterministic model becomes infeasible early on in the schedule relative to the entire horizon, while the stochastic program maintains feasibility for the entire horizon. We now see that the small degradation in objective function in the stochastic program (Table 6) is more than offset by the gain in feasibility with respect to the five scenarios. Finally, the number of activities executed is similar for solutions from both the stochastic and deterministic programs, indicating that the real quantitative difference lies in the makespan. This signifies that the uncertainty prolongs the duration of the activities and, hence, the schedule, but does not, generally speaking, transform a profitable activity into an unprofitable one.

Instance		Measures of Utility					
		(**D**)			(**S**)		
$\|\mathcal{A}\|$ [activities]	T [days]	μ [days]	ϕ [days]	η [activities]	μ [days]	ϕ [days]	η [activities]
56	50	10	2	27	11	50	27
396	200	69	2	229	88	200	229
646	300	117	2	423	149	300	423
1,453	600	296	4	1,117	368	600	1,119
2,323	900	685	3	1,952	874	900	1,953
3,150	1,200	929	4	2,766	1,194	1,200	2,764
3,828	1,500	1,159	2	3,366	1,480	1,500	3,364
4,330	1,800	1,280	4	3,812	1,671	1,800	3,814
4,764	2,100	1,339	4	4,179	1,715	2,100	4,178

Table 7: We measure μ (makespan), ϕ (the last feasible time period in the schedule), and η (expected count of completed activities) for solutions found via the deterministic and stochastic optimization models. Model (**D**) uses a single scenario derived from the mean, while (**S**) uses five scenarios ($|\Omega|$=5); both models consider five resources ($|\mathcal{R}|$=5).

6. Conclusions

Assuming perfect knowledge of value and duration for each activity in an underground mining operation may yield inaccurate mine schedules. Mine planning decisions require input parameters for which only estimates are available. We present a realistic, but intractable, stochastic programming model and demonstrate that by relaxing certain constraints and developing a heuristic that exploits the resulting mathematical structure, we can obtain good-quality solutions, feasible for practical time horizon lengths, even in the presence of the relaxed (non-anticipativity) constraints, within hours. We further demonstrate empirically that the solution quality improves relative to that from a deterministic equivalent based on point estimates of value and duration data.

Alternate heuristic solution strategies might incorporate a priority list π_a of activities a for (**S**) from a mine planner. Additionally, our solution approach could be adapted for solving problems in more general settings. In particular, the principle of first solving a two-stage linear programming problem and then heuristically enforcing non-anticipativity and integrality constraints can be applied to general MSIP problems with decision-dependent uncertainty. Finally, we assume that while duration is uncertain, resource consumption is deterministic. Future work might incorporate the ideas of Demeulemeester et al. [2000] to relax this assumption, though a corresponding solution technique for large instances remains elusive.

Acknowledgements

The authors gratefully acknowledge Chris Alford of Alford Mining Systems for his support. We would also like to thank Dorit Hammerling for conversations regarding scenario generation. This research has been partially funded by the National Institute of Occupational Safety and Health as part of the Mine Ventilation and Safety Research and Capacity Building program, contract number: 0000HCCR-2019-36404, and by the National Agency for Research and Development (ANID), Chile, Scholarship Program, Becas de Doctorado Nacional: 2017-21180460.

References

Ahmed, S., 2010. Two-stage stochastic integer programming: A brief introduction. Wiley Encyclopedia of Operations Research and Management Science.

Alonso-Ayuso, A., Carvallo, F., Escudero, L. F., Guignard, M., Pi, J., Puranmalka, R., Weintraub, A., 2014. Medium range optimization of copper extraction planning under uncertainty in future copper prices. European Journal of Operational Research 233 (3), 711–726.

Bandyopadhyay, S., Rao, S. S., 2017. A test for stationarity for irregularly spaced spatial data. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79 (1), 95–123.

Bienstock, D., Zuckerberg, M., 2010. Solving LP Relaxations of Large-Scale Precedence Constrained Problems. In: Integer Programming and Combinatorial Optimization. Springer, pp. 1–14.

Birge, J. R., Louveaux, F., 2011. Introduction to Stochastic Programming, 2nd Edition. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY.

Blazewicz, J., Lenstra, J. K., Kan, A. R., 1983. Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics 5 (1), 11–24.

Brickey, A., Chowdu, A., Newman, A., Goycoolea, M., Godard, R., 2019. Barrick's Turquoise Ridge Gold Mine optimizes underground production scheduling operations. INFORMS Journal on Applied Analytics, accepted.

Brickey, A. J., 2015. Underground production scheduling optimization with ventilation constraints. Ph.D. thesis, Colorado School of Mines. Arthur Lakes Library.

Bruno, L., 2019. Mineral Resource Estimation. In: Journal of Chemical Information and Modeling. Vol. 53. pp. 1689–1699.

Caldentey, R., Castro, F., Epstein, R., Sauré, D., 2019. Optimal timing of a multi-stage project under market uncertainty. Tech. rep., Working Paper.

Carlyle, W. M., Eaves, B. C., 2001. Underground planning at Stillwater Mining Company. Interfaces 31 (4), 50–60.

Carøe, C. C., Schultz, R., 1999. Dual decomposition in stochastic integer programming. Operations Research Letters 24 (1-2), 37–45.

Carpentier, S., Gamache, M., Dimitrakopoulos, R., 2016. Underground long-term mine production scheduling with integrated geological risk management. Mining Technology 125 (2), 93–102.

Consuegra, F. A., Dimitrakopoulos, R., 2010. Algorithmic approach to pushback design based on stochastic programming: Method, application and comparisons. Mining Technology 119 (2), 88–101.

Cressie, N. A., 1991. Statistics for Spatial Data. A Wiley-Interscience Publication.

Darling, P., 2011. Project alternatives. In: SME Mining Engineering Handbook, 3rd Edition. Society for Mining, Metallurgy, and Exploration (SME), pp. 877–901.

Davari, M., Demeulemeester, E., 2019. The proactive and reactive resource-constrained project scheduling problem. Journal of Scheduling 22 (2), 211–237.

Del Castillo, M. F., Dimitrakopoulos, R., 2019. Dynamically optimizing the strategic plan of mining complexes under supply uncertainty. Resources Policy 60, 83–93.

Demeulemeester, E., De Reyck, B., Herroelen, W., 2000. The discrete time/resource trade-off problem in project networks: A branch-and-bound approach. IIE Transactions 32 (11), 1059–1069.

Demeulemeester, E., Herroelen, W., 2011. Robust project scheduling. Vol. 9. Now Publishers Inc.

Dirkx, R., Dimitrakopoulos, R., 2018. Optimizing infill drilling decisions using multi-armed bandits: Application in a long-term, multi-element stockpile. Mathematical Geosciences 50 (1), 35–52.

Dowd, P. A., Xu, C., Coward, S., 2016. Strategic mine planning and design: Some challenges and strategies for addressing them. Mining Technology 125 (1), 22–34.

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

Goovaerts, P., 1997. Geostatistics for natural resources evaluation. Applied geostatistics series. Oxford University Press, New York.

Guttorp, P., Gneiting, T., 2006. Studies in the history of probability and statistics XLIX on the Matérn correlation family. Biometrika 93 (4), 989–995.

Haneveld, W. K. K., Stougie, L., Van der Vlerk, M. H., 1995. On the convex hull of the simple integer recourse objective function. Annals of Operations Research 56 (1), 209–224.

Haneveld, W. K. K., van der Vlerk, M. H., 1999. Stochastic integer programming: General models and algorithms. Annals of Operations Research 85, 39.

Herroelen, W., Leus, R., 2004. Robust and reactive project scheduling: A review and classification of procedures. International Journal of Production Research 42 (8), 1599–1620.

Herroelen, W., Leus, R., 2005. Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research 165 (2), 289–306.

Jonsbråten, T. W., Wets, R. J., Woodruff, D. L., 1998. A class of stochastic programs with decision dependent random elements. Annals of Operations Research 82, 83–106.

Karacan, C. Ö., Luxbacher, K., 2010. Stochastic modeling of gob gas venthole production performances in active and completed longwall panels of coal mines. International Journal of Coal Geology 84 (2), 125–140.

King, B., Goycoolea, M., Newman, A. M., 2017. Optimizing the open pit-to-underground mining transition. European Journal of Operational Research 257 (1), 297–309.

Kleywegt, A. J., Shapiro, A., Homem-de Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization 12 (2), 479–502.

Koushavand, B., Askari-Nasab, H., Deutsch, C. V., 2014. A linear programming model for long-term mine planning in the presence of grade uncertainty and a stockpile. International Journal of Mining Science and Technology 24 (4), 451–459.

Kuchta, M., Newman, A., Topal, E., 2004. Implementing a production schedule at LKAB's Kiruna Mine. Interfaces 34 (2), 124–134.

Küçükyavuz, S., Sen, S., 2017. An introduction to two-stage stochastic mixed-integer programming. In: Leading Developments from INFORMS Communities. INFORMS, pp. 1–27.

Lamghari, A., Dimitrakopoulos, R., 2012. A diversified tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. European Journal of Operational Research 222 (3), 642–652.

Leite, A., Dimitrakopoulos, R., 2007. Stochastic optimisation model for open pit mine planning: Application and risk analysis at copper deposit. Mining Technology 116 (3), 109–118.

Linderoth, J., Shapiro, A., Wright, S., 2006. The empirical behavior of sampling methods for stochastic programming. Annals of Operations Research 142 (1), 215–241.

Liu, C., Fan, Y., Ordóñez, F., 2009. A two-stage stochastic programming model for transportation network protection. Computers & Operations Research 36 (5), 1582–1590.

Matheron, G., 1962. Traité de Géostatistique Appliquée.

Moran, P. A. P., 1950. Notes on continuous stochastic phenomena. Biometrika 37 (1/2), 17–23.

Muñoz, G., Espinoza, D., Goycoolea, M., Moreno, E., Queyranne, M., Letelier, O. R., 2018. A study of the Bienstock–Zuckerberg algorithm: Applications in mining and resource constrained project scheduling. Computational Optimization and Applications 69 (2), 501–534.

Nehring, M., Topal, E., Knights, P., 2010. Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. Transactions of the Institutions of Mining and Metallurgy, Section A: Mining Technology 119 (4), 212–220.

Newman, A. M., Kuchta, M., 2007. Using aggregation to optimize long-term production planning at an underground mine. European Journal of Operational Research 176 (2), 1205–1218.

Ntaimo, L., 2010. Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. Operations Research 58 (1), 229–243.

Ortiz-Pimiento, N. R., Diaz-Serna, F. J., 2018. The project scheduling problem with non-deterministic activities duration: A literature review. Journal of Industrial Engineering and Management 11 (1), 116–134.

O'Sullivan, D., Brickey, A. J., Newman, A. M., 2015. Is openpit production scheduling "easier" than its underground counterpart? Mining Engineering 67 (4), 68–73.

O'Sullivan, D., Newman, A., 2014. Extraction and backfill scheduling in a complex underground mine. Interfaces 44 (2), 204–221.

Reus, L., Pagnoncelli, B., Armstrong, M., 2019. Better management of production incidents in mining using multistage stochastic optimization. Resources Policy 63, 101404.

Rivera, O., Goycoolea, M., Moreno, E., Espinoza, D., 2015. The OMP Guide.

Rojas, C. R., Goodwin, G. C., Seron, M. M., Zhang, M., 2007. Open-cut mine planning via closed-loop receding-horizon optimal control. In: Identification and Control. Springer, pp. 43–62.

Rossi, M. E., 2014. Mineral Resource Estimation. Springer Netherlands.

Sari, M., 2009. The stochastic assessment of strength and deformability characteristics for a pyroclastic rock mass. International Journal of Rock Mechanics and Mining Sciences 46 (3), 613–626.

Schultz, R., Stougie, L., Van Der Vlerk, M. H., 1996. Two-stage stochastic integer programming: A survey. Statistica Neerlandica 50 (3), 404–416.

Teter, M. D., Newman, A. M., Weiss, M., 2016. Consistent notation for presenting complex optimization models in technical writing. Surveys in Operations Research and Management Science 21 (1), 1–17.

Trout, L. P., 1995. Underground mine production scheduling using mixed integer programming. In: 25th International APCOM Symposium Proceedings. pp. 395–400.

Van Slyke, R. M., Wets, R., 1969. L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17 (4), 638–663.

Appendix - A

Algorithm 1: SIMPLE SORT $\mathcal{H}_2$

Data: LP relaxation values Y' from having solved $(\mathbf{LS}^-)$ in $\mathcal{H}_1$ with OMP
Result: List of mean starting times, MS, and sorted list of activities, SL

compute mean starting time, $MS[a]$, of each activity $a \in \mathcal{A}$: $MS[a] \leftarrow \sum_{t \in \mathcal{T}} t \cdot {Y'}_{at}$;
construct set, $\mathcal{A}'$, of activities that will be executed: $\mathcal{A}' \leftarrow \{a \in \mathcal{A} : MS[a] \geq 0.5\}$;
sort activities in set $\mathcal{A}'$ non-decreasing by MS[a] and assign to ordered list SL;
return lists MS and SL;

Algorithm 2: LIST-SCHEDULING HEURISTIC $\mathcal{H}_3$

Data: List of mean starting times, MS, and sorted list of activities, SL
Result: Integer feasible solution, (X, Y)

$Y_{at} \leftarrow 0$ for each activity $a \in \mathcal{A}$, time period $t \in \mathcal{T}$;
$X^{\omega}_{at} \leftarrow 0$ for each activity $a \in \mathcal{A}$, time period $t \in \mathcal{T}$, scenario $\omega \in \Omega$;
while *list SL is not empty* **do**
    select the first activity, a', in list SL and delete it from list SL;
    $t' \leftarrow \lfloor MS[a'] \rceil$;
    while *period $t' \leq T$* **do**
        $Y_{a't'} \leftarrow 1$;
        $X^{\omega}_{a't} \leftarrow 0$ for each time period $t \in \mathcal{T}$, scenario $\omega \in \Omega$;
        for $\omega \in \Omega$ **do**
            $feas\_scenario \leftarrow FALSE$;
            for $t^* \in \{\max\{1, t' - \delta\}, \ldots, \min\{T, t' + \delta\}\}$ **do**
                if *assigning $X^{\omega}_{a't^*}$ to 1 is precedence- and resource-feasible* **then**
                    $X^{\omega}_{a't^*} \leftarrow 1$;
                    $feas\_scenario \leftarrow TRUE$;
                    break;
                end
            end
            if $feas\_scenario = FALSE$ **then**
                break;
            end
        end
        if $feas\_scenario = FALSE$ **then**
            $Y_{a't'} \leftarrow 0$;
            $t' \leftarrow t' + 1$;
        end
    end
end
return (X, Y);

Appendix - B

Proposition 1. *A solution, (X, Y), generated by Algorithm 2 is a feasible solution for problem* $(\mathbf{S})$.

Proof. We focus only on non-anticipativity constraints (7) and (8) since precedence constraints (3), resource constraints (4), and time-interval constraints (5) and (6), are trivially satisfied by construction (see Algorithm 2, lines 11 and 12).

We will prove that solution (X, Y) satisfies non-anticipativity constraints (7) and (8) by induction.

- Base case, activity at position 1 in list SL: The non-anticipativity constraints associated with activity $a_1 = SL[1]$ hold since $\mathcal{P}_{a_1} = \emptyset$ and, by construction, $X^{\omega}_{a_1 t^*} = 1$ for each scenario $\omega \in \Omega$, where $t^* = \max\{1, \lfloor MS[a_1] \rceil - \delta\}$.

- Inductive step: Now, we prove that if the non-anticipativity constraints hold for each activity $SL[1], \ldots, SL[n]$, then the non-anticipativity constraints hold for each activity $SL[1], \ldots, SL[n+1]$.

  We have to check whether the non-anticipativity constraints hold for activity $a_{n+1} = SL[n+1]$. First, we note that, by construction, Algorithm 2 decides whether activity a_{n+1} is scheduled or not. If a_{n+1} is scheduled, it is scheduled in all scenarios. If it is not scheduled, it is not scheduled in any scenario. Therefore, we have two cases to check:

  1. Activity a_{n+1} is not scheduled: In this case, the non-anticipativity constraints associated with activity a_{n+1} are trivially satisfied for each scenario pair $\{\omega_1, \omega_2\} \subseteq \Omega$.
  2. Activity a_{n+1} is scheduled: Given any scenario pair $\{\omega_1, \omega_2\} \subseteq \Omega$, let t_1 and t_2 be the time periods at which activity a_{n+1} starts in scenarios ω_1 and ω_2, respectively. In other words, $X^{\omega_1}_{a_{n+1} t_1} = X^{\omega_2}_{a_{n+1} t_2} = 1$. Moreover, let $\hat{t} = \min\{t_1, t_2\}$. By the beginning of time period $\hat{t}$, scenarios ω_1 and ω_2 are either differentiated or indistinguishable. If scenarios ω_1 and ω_2 are already differentiated by the beginning of time period $\hat{t}$, then the non-anticipativity constraints associated with activity a_{n+1} are trivially satisfied for scenario pair $\{\omega_1, \omega_2\}$. On the other hand, if scenarios ω_1 and ω_2 are indistinguishable by the beginning of time period $\hat{t}$, then we know that (i) exactly the

same activities have been completed in both scenarios; (ii) for each of these completed activities, its corresponding duration in scenarios ω_1 and ω_2 are the same; (iii) the starting times of all activities $SL[1], \ldots, SL[n]$ in scenario ω_1 are equal to the corresponding starting times in scenario ω_2, since, by the inductive hypothesis, activities $SL[1], \ldots, SL[n]$ satisfy the non-anticipativity constraints; (iv) the activities in $\{SL[1], \ldots, SL[n]\}$ that are not yet completed are under execution the same amount of time in both scenarios ω_1 and ω_2. By (i), (ii), and (iii), it follows that the activities in $\mathcal{P}_{a_{n+1}}$, which are contained in $\{SL[1], \ldots, SL[n]\}$, start and end at exactly the same period in both scenarios ω_1 and ω_2. Moreover, by (iv), the resource availability at time period $\hat{t}$ is the same in both scenarios ω_1 and ω_2. Thus, given that both time periods t_1 and t_2 correspond to the first time period at which it is feasible to start activity a_{n+1} in the respective scenario (see Algorithm 2, lines 11 and 12), we have that $\hat{t} = t_1 = t_2$. Thus, the non-anticipativity constraints associated with activity a_{n+1} are satisfied for scenarios ω_1 and ω_2.

Therefore, solution (X, Y) satisfies non-anticipativity constraints (7) and (8).

$\square$