Les Cahiers du GERAD ISSN: 0711–2440

Scheduling of drilling machines in open-pit mines: Stochastic and non-probabilistic CP approaches

M. Maftah, M. Gendreau, B. Agard, M. Gamache

G–2024–62

October 2024

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Scheduling of drilling machines in open-pit mines: Stochastic and non-probabilistic CP approaches

Mohamed Maftah [a, b]

Michel Gendreau [a]

Bruno Agard [a]

Michel Gamache [a, b]

[a] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[b] *GERAD, Montréal (Qc), Canada, H3T 1J4*

mohamed.maftah@polymtl.ca
michel.gendreau@polymtl.ca
bruno.agard@polymtl.ca
michel.gamache@polymtl.ca

October 2024
Les Cahiers du GERAD
G–2024–62

Abstract : This paper addresses the scheduling of electrical drilling machines in open-pit mines, proposing three constraint programming formulations to account for uncertain drilling durations: two-stage stochastic, a novel probability-free method, and a chance-constrained model. These approaches aim to create robust schedules for drilling operations. The probability-free method introduces a "Resilient" constraint for deterministic representation of uncertainty, while the chance-constrained approach combines elements of two-stage stochastic and resilient programming using scenario approximation. Evaluated using simulated instances from real coal mine data, all models efficiently handle problem sizes comparable to or exceeding typical daily operations. The probability-free model demonstrates particular efficiency and scalability. This study contributes to mining operations research by providing flexible, robust constraint programming models for drill rig scheduling under uncertainty, offering mining practitioners tools to optimize operations in uncertain environments.

Keywords: Open-pit mining, drill rig scheduling, constraint programming, stochastic optimization, probability-free optimization, chance-constrained optimization

Résumé : Cet article traite de la planification des foreuses électriques dans les mines à ciel ouvert, proposant trois formulations de programmation par contraintes pour tenir compte des durées de forage incertaines : stochastique à deux étapes, une nouvelle méthode sans probabilité, et un modèle à contraintes de chance. Ces approches visent à créer des calendriers robustes pour les opérations de forage. La méthode sans probabilité introduit une contrainte "Résiliente" pour une représentation déterministe de l'incertitude, tandis que l'approche à contraintes de chance combine des éléments de la programmation stochastique à deux étapes et de la programmation résiliente en utilisant l'approximation par scénarios. Évalués à l'aide d'instances simulées à partir de données réelles de mines de charbon, tous les modèles gèrent efficacement des tailles de problèmes comparables ou supérieures aux opérations quotidiennes typiques. Le modèle sans probabilité démontre une efficacité et une évolutivité particulières. Cette étude contribue à la recherche opérationnelle minière en fournissant des modèles de programmation par contraintes flexibles et robustes pour la planification des foreuses sous incertitude, offrant aux praticiens miniers des outils pour optimiser les opérations dans des environnements incertains.

Mots clés : Mines à ciel ouvert, ordonnancement de foreuses, programmation par contraintes, optimisation stochastique, optimisation sans probabilité, optimisation sous contraintes probabilistes

Acknowledgements: The authors would like to express their gratitude for the support received through the NSERC Cooperative Research Development program. This work was made possible by the grant RDCPJ 531815018, which played a crucial role in the research and development of this project.

1 Introduction

Open-pit mining occupies a major place in global mineral extraction. A pivotal stage in this operation is the drilling phase, where tactical decisions can profoundly influence both the efficiency and cost of the following mining activities. A specific challenge during this phase is scheduling electrical drill rigs operating closely within the mine. Precise scheduling is crucial, ensuring not just timely task execution but also the safety and efficiency of the entire operation.

In a recent contribution Maftah et al. (2024), we introduced a constraint programming (CP) model addressing the drill rig scheduling challenge encountered in open-pit mining operations. This model aimed to maximize the number of drilling tasks completed within a given time horizon, while adhering to constraints related to drill rig positioning, collision avoidance, and task precedences.

While the deterministic CP model effectively produced high-quality schedules for practical problem sizes, real-world drilling operations often face substantial uncertainties. Factors such as rock conditions, equipment malfunctions, and unforeseen disruptions can cause variations in drilling durations. To address the inherent unpredictability in drilling duration, we suggested indirectly managing this uncertainty by re-solving the problem whenever we receive more precise estimates of drilling durations. This reactive approach was feasible because our deterministic model could generate solutions with small optimality gaps for practical-sized instances in under 2 minutes when run on a standard computer.

In this paper, we offer extensions to our previous CP model, focusing on proactive scheduling approaches to address the uncertainty in drilling durations. A proactive approach, as opposed to a reactive approach, seeks to account for uncertainty in the duration of drilling activities during the construction of the drilling sequence, rather than reacting when drilling activity durations deviate from what was planned. This anticipatory strategy aims to create robust schedules that can accommodate variations in task durations without requiring frequent rescheduling. By incorporating uncertainty directly into our extended scheduling models, we aim to produce more resilient and reliable drilling plans. These extensions to our previous work represent a significant advancement in addressing the inherent unpredictability of open-pit mining operations. In line with this proactive strategy, we propose three scheduling approaches:

1. A two-stage stochastic programming formulation to optimize schedules across multiple task duration scenarios.
2. A probability-free optimization model that protects against uncertainty in the problem data, offering a middle-ground approach between deterministic and stochastic modeling.
3. A chance-constrained model that takes a risk-averse approach, ensuring that schedules meet a predefined level of reliability.

The core contribution of this paper is the formulation, validation, and juxtaposition of three non-deterministic constraint programming models intended for the coordination of multiple drill rigs in open-pit mines. Our goal is to equip mining practitioners with a suite of models designed to hedge against operational uncertainty in different ways. By comparing deterministic and non-deterministic methodologies, we aim to highlight the flexibility and utility of CP approaches, ensuring optimal drill rig scheduling in the face of potential uncertainties.

Following this introduction, Section 2 reviews related work on stochastic optimization for scheduling challenges prevalent in mining operations. Section 3 presents a description of the problem. Section 4 presents the three non-deterministic CP models, while Section 5 showcases their application through simulated instances derived from a coal mine's drilling data. We conclude with insights gained and directions for future research.

2 Literature review

Mining is a domain replete with uncertainties, ranging from geological variability, task duration, equipment reliability, to fluctuating demand and prices. Effective scheduling under such uncertain conditions becomes crucial for optimizing productivity, reducing costs, and ensuring the safety of operations. The literature has seen various approaches to tackle these challenges, including scenario-based optimization, chance-constrained optimization, and probability-free optimization.

2.1 Scenario-based optimization

Scenario-based optimization makes use of multiple stochastically simulated scenarios to describe uncertainties in the mining process. It is a proactive approach that prepares for various possible future scenarios by developing solutions that perform well on average across them. Multiple authors have tackled scheduling issues in the mining industry through the application of scenario-based optimization.

In the context of gold mining, Ramazan and Dimitrakopoulos (2013) utilized a two-stage stochastic integer model to schedule annual production, incorporating uncertainty regarding the availability of mineralized materials in the ground. Expanding on this, Lamghari and Dimitrakopoulos (2016) broadened the scope of mine production scheduling by integrating decisions about potential destinations for mined materials under metal uncertainty, developing a two-stage stochastic programming model solved with a heuristic based on network flow techniques.

Focusing on mining complexes, Goodfellow and Dimitrakopoulos (2016a) explored global optimization for generating economically viable production schedules across multiple mines and processing streams, introducing a two-stage stochastic programming model for a copper-gold mining complex. Similarly, Del Castillo and Dimitrakopoulos (2019) developed a multistage stochastic programming model to determine the optimal annual production schedule for a copper mining complex, notably integrating investment decisions throughout the asset's life.

In the realm of open-pit mines, Khan (2018) addressed production scheduling under grade uncertainty, proposing two population-based metaheuristics: one based on particle swarm optimization and another utilizing the bat algorithm. These approaches offer computational advantages in terms of tractability and feasibility compared to typical solvers used for two-stage stochastic programming models. Along similar lines, Danish et al. (2023) presented a stochastic optimization algorithm based on Simulated Annealing, leveraging multiple simulated realizations of an orebody to account for geological uncertainties.

Within the context of underground gold mines, Furtado E Faria et al. (2022) introduced a two-stage stochastic integer program for optimizing stope and development network designs under grade uncertainty and variability. Their model aims to maximize discounted revenues, minimize development costs, and manage production target risks. In a similar vein, Aalian et al. (2024) proposed a two-stage stochastic constraint programming model for short-term scheduling in underground mining, determining robust sequences of activities for available resources based on various scenarios derived from real data sets. Additionally, they introduce an alternative model based on the chance-constraint paradigm, bridging the gap between scenario-based and chance-constrained optimization approaches.

2.2 Chance-constrained optimization

Chance-constrained optimization is a paradigm that seeks to find solutions that satisfy constraints with a certain probability, making it particularly well-suited for environments like mining where uncertainties abound. Several researchers have addressed scheduling challenges in the mining sector using chance-constrained optimization techniques. In their work, Golamnejad et al. (2006) introduced a long-term production scheduling model utilizing chance-constrained binary integer programming. Their model adeptly accounts for the inherent uncertainty in ore block grades. Additionally, Kumral and Sari (2017) proposed an extraction sequencing methodology that prioritizes maximizing the net present value of mining projects, all while maintaining a specified risk tolerance. Their approach combines chance-constrained programming with Monte Carlo simulation techniques, drawing data from a gold mine for validation. Moreover, Gholamnejad et al. (2020) applied a chance-constrained integer programming approach to an open-pit iron and gold mine production scheduling problem with a single stockpile for storing low-grade material.

Shifting the focus to short-term planning, Mohtasham et al. (2021) considered a truck allocation problem in a copper open-pit mine. By employing a chance-constrained goal programming model, the authors demonstrate how uncertainties in truck-shovel systems can be managed. Their model proves effective across various confidence levels, ensuring the short-term production schedule's objectives are met even in high-risk scenarios. In a parallel context focusing on underground mines, Aalian et al. (2024) studied short-term planning within an underground gold mine, considering the confined space and numerous uncertainties such as activity duration variations. The authors employ a constraint programming model with `Confidence` constraints, adapting chance-constrained programming techniques as demonstrated by Mercier-Aubin et al. (2020) for use in constraint programming. This approach allows mine planners to control the risk level of the generated solution, ensuring that the produced schedule meets certain reliability criteria, given actual activity durations.

2.3 Probability-free optimization

Probability-free optimization approaches make decisions based solely on available data, without relying on probabilistic models or making assumptions about underlying probability distributions. This approach is particularly valuable when such distributions are difficult to ascertain or when the available data does not conform to standard probability distributions. In the context of mining operations scheduling, only one paper was found that presents solutions without explicitly relying on stochastic or probabilistic parameters. Alipour et al. (2018) studied production scheduling problems in open-pit mines with the goal of finding the best block extraction sequence to maximize profit. The authors introduced a robust counterpart linear optimization model, incorporating the ellipsoidal set-based counterpart to account for uncertainties in block economic value, block weights, and operational capacity. To solve the nonlinear ellipsoidal counterpart, the authors proposed a genetic algorithm.

While applications in mining are scarce, probability-free and robust optimization methods have been successfully applied in various other fields. This scarcity in mining applications presents an opportunity for novel research rather than indicating a lack of utility. In the context of Internet of Things (IoT) applications, Martinez et al. (2022) employed robust optimization for the design and dimensioning of fault-tolerant fog computing infrastructures. Chen et al. (2016) applied robust optimization to daily maintenance routing problems in road networks with uncertain service times. In facility location problems, Cheng et al. (2021) used robust optimization to address demand uncertainty and facility disruptions.

Recent years have seen an increased interest in data-driven robust optimization across diverse domains. Huang et al. (2023) applied this approach to industrial utility systems integrating wind and solar energies. Asgari et al. (2024) developed a novel method based on position-regulated support vector clustering for creating data-driven uncertainty sets. In the renewable energy sector, Sadeghi Darvazeh et al. (2024) used data-driven robust optimization to design an integrated sustainable forest biomass-to-electricity network. The versatility of these approaches is further demonstrated by their application in steel supply chain network design (Khalili-Fard et al., 2024), operation optimization of industrial power stations (Ashraf and Dua, 2024), and PVC production scheduling (Wang and Su, 2024).

These diverse applications underscore the potential of probability-free and robust optimization methods to address complex uncertainties across various industries. Given the successful application of these methods in other fields, there is significant potential for their adaptation and application to mining operations, particularly in addressing the inherent uncertainties in production scheduling and resource allocation.

2.4 Hybrid approaches

While individual optimization paradigms have been extensively studied and applied in mining operations, there is a growing interest in hybrid approaches that combine elements from different optimization methods to address complex uncertainties and improve overall system performance. These hybrid approaches have shown promise not only in mining but also in various other fields dealing with complex systems and uncertainties.

In the context of mining operations, several researchers have developed innovative hybrid methods. Goodfellow and Dimitrakopoulos (2016b) proposed a novel approach that integrates stochastic integer programming with sequential Gaussian simulation to simultaneously optimize mining complexes and mineral value chains. This method allows for the consideration of geological uncertainty throughout the entire mineral value chain, from extraction to processing and transportation, demonstrating improved net present value and risk management compared to conventional methods. Similarly, Benndorf and Yueksel (2015) developed an approach that combines simulation and optimization techniques for continuous mining systems, integrating a detailed simulation model of the mining process with an optimization algorithm to improve operational efficiency.

Beyond mining, hybrid approaches have been successfully applied in various fields, particularly in the context of robust optimization and energy systems. Hashemi Doulabi et al. (2021) developed a hybrid Benders algorithm for two-stage robust optimization models with exponential scenarios, applying it to nurse planning and supply chain problems. Cheng et al. (2018) proposed a two-stage robust approach for reliable logistics network design, combining uncertainty sets with recourse decisions. In the energy sector, Nourollahi et al. (2022) presented a hybrid approach combining robust optimization and stochastic programming to optimize the operation of a residential hybrid energy system, effectively handling different types of uncertainties. Similarly, Yan et al. (2022) developed a two-stage stochastic-robust optimization model for a hybrid renewable energy system, addressing multiple scenario-interval uncertainties.

Innovative hybrid methods have also emerged in other optimization domains. For instance, Butkeraites et al. (2022) introduced a sampling-based multi-objective iterative robust optimization method for the Bandwidth Packing Problem, combining sampling techniques with unsupervised learning to explore the topology of uncertain parameter sets.

These diverse applications of hybrid approaches demonstrate their potential to address complex uncertainties and improve system performance across various fields. In the context of mining operations, such approaches offer promising avenues for addressing the multifaceted uncertainties inherent in mining, from geological variability to operational complexities and market fluctuations. The success of hybrid methods in other domains suggests their potential for enhancing the efficiency and robustness of optimization in mining operations.

2.5 Conclusion

The mining sector, characterized by inherent uncertainties ranging from geological variances to economic fluctuations, has long relied on optimization techniques to ensure optimal scheduling and, consequently, maximize returns on investments. Through our review of the literature, three predominant paradigms emerge as key avenues for addressing these uncertainties: scenario-based optimization, chance-constrained optimization, and probability-free optimization.

Scenario-based optimization, given its robustness and ability to prepare for diverse potential future events, remains a popular choice among researchers. It is evident from our study that many researchers have developed and proposed models under this paradigm, with applications ranging from production scheduling of entire mining complexes to underground mine planning.

On the other hand, chance-constrained optimization offers an interesting alternative, aiming to find solutions that satisfy constraints within specified probabilistic bounds. A fair amount of research has been conducted in this space, with applications ranging from production scheduling with ore block grade uncertainty to truck allocation problems in open-pit mines.

Probability-free optimization, the third paradigm, represents a departure from probabilistic models, instead opting for data-driven decisions without probabilistic assumptions. Surprisingly, our literature review reveals a conspicuous scarcity of research in this domain. Only a single paper was identified that delves into mining operations scheduling without leaning on stochastic or probabilistic parameters.

While these paradigms have been widely studied and applied individually, there is growing interest in hybrid approaches that combine elements from different optimization paradigms to address complex uncertainties in mining operations. Such approaches could leverage the strengths of various methods to provide more comprehensive solutions to the multifaceted uncertainties in mining. Examples from both mining and other fields demonstrate the potential of these hybrid methods to enhance the robustness and efficiency of optimization models.

It is worth noting that Constraint Programming (CP) has been successfully applied to various mining problems, primarily focusing on deterministic scenarios. Recent years have seen an increased interest in applying CP to mining operations. Strand et al. (2020) used CP for scheduling mobile machines in underground mines. Valenca Mariz et al. (2024) proposed a multi-stage CP approach for solving clustering problems in open-pit mine planning. Oleynik and Zuenko (2022) applied CP to open-pit mine production scheduling. Campeau and Gamache (2022) optimized short- and medium-term underground mine planning using CP. Kumar et al. (2023) explored CP for open-pit mine production scheduling with stockpiling. Aalian et al. (2023) developed a CP model for short-term scheduling in an underground gold mine. However, the application of CP to address uncertainties in mining operations remains limited. The use of stochastic CP in mining, which could potentially address uncertainties more comprehensively, is notably scarce, with only one identified study (Aalian et al., 2024) exploring this approach.

Overall, the literature addresses a range of problems with a clear focus on production scheduling and an overarching emphasis on maximizing profits. The reviewed studies not only provide theoretical frameworks but also validate their proposed methodologies in real-world settings. Nevertheless, in light of the existing literature, several gaps can be identified:

- While mining scheduling problems have been studied, there remains a largely unexplored domain in terms of the unique scheduling challenges that arise during the drilling process.
- Although CP has been applied to various deterministic mining problems, its application to mining scheduling problems under uncertainty remains largely unexplored, indicating a significant research gap.
- As uncertainties in mining are manifold, there is a pressing need to explore diverse methods to address parameter uncertainty. Our literature review indicates that while some methods have been proposed, a comprehensive and comparative study of multiple paradigms, especially using stochastic constraint programming models, remains largely unexplored.

In light of these observations, this paper seeks to bridge these gaps by:

- Introducing a novel scheduling problem under uncertainty specific to the drilling operations of electrical drill rigs in open-pit mines.
- Employing constraint programming to model this new problem, building on the limited precedent in the literature.

- Proposing three distinct constraint programming models, each corresponding to the aforementioned paradigms, offering a comprehensive approach to tackle parameter uncertainty in mining operations.

Through these contributions, this paper aims to enrich the existing body of knowledge, offering innovative solutions to hitherto unaddressed challenges in the mining industry. By introducing a hybrid model that combines multiple optimization paradigms, we seek to provide a more comprehensive and flexible approach to addressing the complex uncertainties inherent in mining operations.

3 Problem description

In a previous work Maftah et al. (2024), we introduced a deterministic discrete optimization problem encountered during the drilling phase in open-pit mines, which we called the Drill Coordination Problem (DCP). The objective of the DCP is to optimize the drilling operations of multiple machines while satisfying various constraints. These constraints include maintaining safe distances between machines, preserving the integrity of power cables, and adhering to specific movement restrictions.

The problem setting in this paper is similar to the one described in Maftah et al. (2024), where a set of drilling machines $\mathcal{M}$ must visit and drill a set of targets $\mathcal{J}$ (also referred to as tasks) arranged in a pattern of rows and columns, as depicted in Figure 1. The constraints on machine movements and cable management remain the same. However, in this work, we consider the case where the drilling durations at each target are uncertain.

To provide context for our current contribution, we briefly summarize the key elements of the original drill scheduling problem:

- There is a set of targets $\mathcal{J}$ that must be drilled, with each target indexed by $j \in \{1, 2, \dots, |\mathcal{J}|\}$.
- There is a set of drilling machines $\mathcal{M}$, indexed by $m \in \{1, 2, \dots, |\mathcal{M}|\}$, each connected to a power source via a cable.
- The machines must transition from target to target in a specific order, which is determined by the column structure of the drilling pattern $\mathcal{C}$, indexed by $c \in \{1, 2, \dots, |\mathcal{C}|\}$:
  - Precedence constraints over targets within a column, represented by $\mathcal{J}_c$.
  - Direct lateral movement between adjacent columns is prohibited.
- Safety constraints enforce minimum distances between machines and cable locations, represented by the parameter δ, which indicates the number of columns required as spacing between machine pairs. For example, if $\delta = 2$, machines must maintain at least two columns of separation at all times.
- Collision avoidance constraints mandate that machines maintain their initial relative positions with respect to each other throughout the drilling process. For example, if Machine A starts to the left of Machine B, it must remain to the left of Machine B for the entire operation.

To visualize these elements and constraints, Figure 1 provides a schematic representation of a typical DCP instance.

In this study, we focus on uncertainty in drilling durations (p_{jm}) while treating travel times between tasks (t_{ij}^{m}) as deterministic. This decision is based on several factors:

1. Drilling times are subject to greater variability due to factors such as rock hardness, equipment performance, and unforeseen geological conditions.
2. Travel times between drilling locations are generally more predictable, with fixed distances and relatively constant machine speeds.
3. The impact of drilling time uncertainty on the overall schedule is typically more significant than that of travel time variations.

4. Focusing on drilling time uncertainty allows us to capture the most significant source of variability while maintaining model tractability.

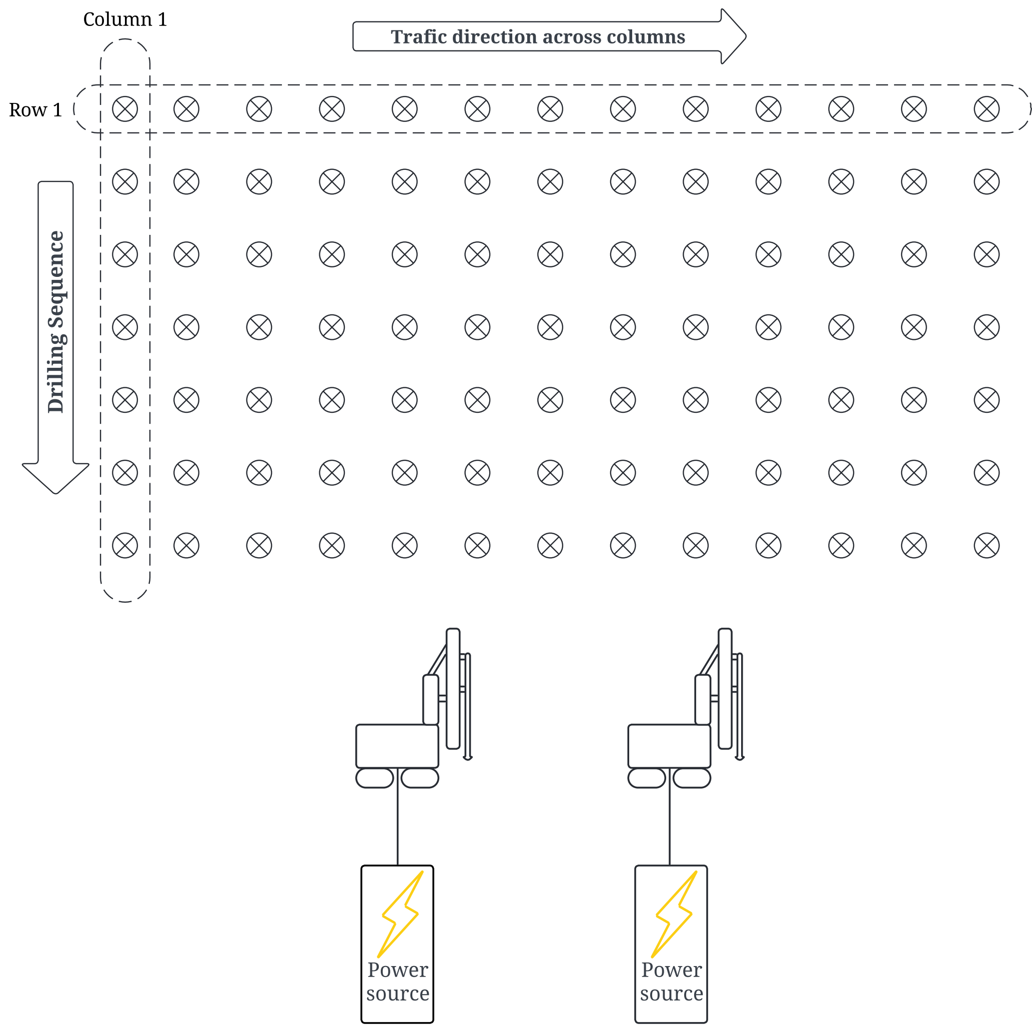

Figure 1: An instance of the DCP.

While some level of uncertainty in travel times exists, we believe our approach adequately represents the primary sources of operational uncertainty in the DCP. For a more detailed discussion of practical considerations in open-pit mining operations, readers are directed to the second section of our earlier paper Maftah et al. (2024). The goal of our study is to find a drilling schedule that maximizes the total number of drilled targets within a specified timeframe H (planning horizon), while satisfying all of the constraints. The planning horizon H represents the total available time for the drilling operation, typically determined by operational constraints such as shift durations or project deadlines. In practice, this could range from a single shift to multiple days, depending on the specific operational context and planning needs. The planning horizon is typically discretized into fixed time intervals to facilitate scheduling and resource allocation decisions. The granularity of time discretization can impact both the precision of the resulting schedule and the computational complexity of solving the problem.

To model this discretized time framework and represent the scheduling decisions, we employ three types of variables in our constraint programming formulation:

- Interval variables x_{jm}: These represent the assignment and timing of task j to machine m. They capture both the decision of whether machine m performs task j and, if so, when the task starts and ends.
- Sequence variables z_m: These represent the order of tasks on each machine m. They determine the sequence in which machine m performs its assigned tasks, ensuring that the scheduling respects the precedence and movement constraints described earlier.
- State functions l_m: These denote the location of machine m throughout the scheduling horizon. They track the column position of each machine at any given time, allowing us to enforce spatial constraints such as safety distances and relative positioning between machines.

This combination of interval variables, sequence variables, and state functions allows us to comprehensively model the complex spatial and temporal aspects of the drill coordination problem, including task assignments, execution order, and machine movements.

For a comprehensive understanding of the practical considerations leading to these constraints, readers are directed to the second section of our earlier paper Maftah et al. (2024).

4 Constraint programming formulations

In this section, we enhance the deterministic model presented in our previous paper by introducing three non-deterministic counterparts: a two-stage stochastic model, a probability-free model, and a chance-constrained model. These models incorporate mechanisms to address the inherent unpredictability in task durations.

Before delving into the various formulations, we will outline the elements that are common to all three models. Additionally, we will reproduce the deterministic model from our previous paper Maftah et al. (2024). This allows for a direct comparison with the extended models, facilitating a clear understanding of how each model accounts for parameter uncertainty in their respective subsections. In order to express the DCP, we use IBM ILOG CP Optimizer (CPO), a constraint programming solver and modeling language specifically designed for scheduling problems. We chose CPO for its robust set of global constraints tailored to scheduling applications and its efficient handling of complex temporal relationships. Table 1 shows the constructs provided by this modeling language:

Table 1: Variables, constraints, and functions used in the CP models.

Variables	
`Interval`	Represents a time interval for an activity whose exact start time is not yet determined. These variables can be designated as optional, allowing the solver to decide whether to include or exclude the activity when finding a solution.
`Sequence`	Represents the order in which a set of interval variables are scheduled. If an optional interval variable is excluded from the solution, it is automatically omitted from the sequencing.
`State function`	Represents the evolution of a system's status over time. It derives its values from interval variables, taking on non-negative integer values to signify the state when the associated interval variable is included in the solution. Essentially, it encapsulates the change of a system's condition as influenced by the presence or absence of specific intervals in the solution.
Functions	
`EndOf`	Given an interval variable i, this function returns an integer expression representing the completion time of i. If i is included in the solution, the expression equals its end time. Otherwise, it returns a predefined value (typically zero, but this can be customized). This function is useful for coordinating the timing of different activities in a schedule.

Continued on next page

`SizeOf`	Given an interval variable i, this function yields an integer expression denoting the duration of i. When i is part of the solution, the expression corresponds to its length. If i is not included, the function returns a specified default value (usually zero, but can be adjusted as needed). This function is particularly helpful in scenarios where the duration of activities impacts the overall schedule or resource allocation.
`PresenceOf`	Given an interval variable i, this function produces a boolean expression indicating whether i is included in the solution. It returns 1 if i is present and 0 if it is absent. This function is instrumental in formulating logical dependencies between different activities in the schedule, such as ensuring that if one task is performed, another related task must also be included. It can also be employed to calculate costs associated with the inclusion or exclusion of specific activities in the final schedule.
Global constraints	
`Alternative`	Given a "parent" interval variable a and a collection of "child" interval variables $b_1, \ldots, b_n$, this constraint ensures that exactly one child interval from the set is selected to represent the parent interval. The chosen child interval's start and end times must exactly match those of the parent interval a. If a is not present in the solution, none of the child intervals are present, and vice versa. This constraint is useful for modeling alternative ways to perform a task, where a represents the overall task and each b_i represents a specific method (i.e. machine) of performing it.
`AlwaysEqual`	Given a state function f, an interval variable i, and a value v, this constraint ensures that the state of function f must be equal to v within the time span defined by the start and end times of interval variable i, if i is scheduled.
`AlwaysIn`	Given a state function f, an interval variable i, and specified values min and max, this constraint ensures that the state of function f must consistently remain within the defined interval $[min, max]$ during the time span indicated by the start and end times of interval variable i, if i is scheduled.
`EndBeforeStart`	Given two interval variables a_1 and a_2, this constraint ensures that the completion time of a_1 has to be less than or equal to the start time of a_2.
`NoOverlap`	Given a set of interval variables and the setup times between each pair of elements of the set, this constraint ensures that the durations of the activities represented by the interval variables do not overlap. Additionally, it guarantees that the time gap between these activities exceeds or equals the specified setup time for each pair.

For a more detailed overview of this modeling language along with practical examples, we refer the reader to Laborie et al. (2018).

4.1 Deterministic model

The deterministic model, which serves as the foundation for the three non-deterministic counterparts, is outlined as follows:

Sets	
$\mathcal{M}$	set of machines, indexed by $m \in \{1, 2, \ldots, \vert\mathcal{M}\vert\}$.
$\mathcal{M}^*$	set of machines excluding its first element (i.e. $\mathcal{M}^* = \mathcal{M} \backslash \{1\}$).
$\mathcal{J}$	set of tasks, indexed by i and $j \in \{1, 2, \ldots, \vert\mathcal{J}\vert\}$.
$\mathcal{J}^*$	set of tasks excluding its first element (i.e. $\mathcal{J}^* = \mathcal{J} \backslash \{1\}$).
$\mathcal{I}_j$	tasks that become inaccessible once task j is processed.
$\mathcal{C}$	set of columns, indexed by $c \in \{1, 2, \ldots, \vert\mathcal{C}\vert\}$.
$\mathcal{J}_c$	tasks associated with column c, indexed by j.
$\mathcal{J}_c^-$	set $\mathcal{J}_c$ excluding its last element.

Note 1. The numbering of tasks and machines plays a crucial role in implementing the precedence and safety constraints in our model:

- Tasks are numbered from 1 to $|\mathcal{J}|$ in a specific order:
  - Task 1 is located at the top-left corner of the drilling pattern (see Figure 1).
  - Numbering proceeds downward within each column, then continues from the top of the next column to the right.
  - The last task ($|\mathcal{J}|$) is located at the bottom-right corner of the pattern.
- Machines are numbered from 1 to $|\mathcal{M}|$, increasing from left to right as shown in Figure 1.

This systematic numbering allows us to express precedence relationships and maintain safety distances efficiently in our model constraints. For example, constraint set (0.4) uses this numbering to ensure the correct drilling sequence within columns, while constraint sets (0.6) and (0.7) leverage it to preserve relative positions and safety gaps between machines.

Parameters	
col_j	column associated with task j.
p_{jm}	processing duration of task j on machine m.
t_{ijm}	setup duration between tasks i and j when on machine m.
δ	number of columns required as spacing between machine pairs.
H	planning horizon.
Variables	
x_{jm}	optional interval variable of size p_{jm} indicating if task j is performed by machine m.
y_j	optional interval variable representing task j.
z_m	sequence of interval variables x_{jm} for machine m.
l_m	state function denoting the location of machine m.

Note 2. The omission of an index from a variable or parameter within the model implies reference to the array encompassing the omitted index. For example, $\mathbf{x}_{j.}$ denotes the one-dimensional array $[x_{j1},\ x_{j2}, \dots,\ x_{j|\mathcal{M}|}]$, while $\mathbf{t}_{..m}$ denotes the two-dimensional array representing setup times between each pair of tasks for a specific machine m.

Model 0

$$\text{Maximize} \qquad Z = \sum_{j \in \mathcal{J}} \texttt{PresenceOf}(y_j) \tag{0.1}$$

$$\text{s.t.} \qquad \texttt{Alternative}(y_j,\ \mathbf{x}_{j.}), \quad \forall j \in \mathcal{J}, \tag{0.2}$$

$$\texttt{NoOverlap}(z_m,\ \mathbf{t}_{..m}), \quad \forall m \in \mathcal{M}, \tag{0.3}$$

$$\texttt{EndBeforeStart}(y_j,\ y_{j+1}), \quad \forall c \in \mathcal{C},\ \forall j \in \mathcal{J}_c^-, \tag{0.4}$$

$$\texttt{NoOverlap}(\{y_j \mid j \in \mathcal{I}_j\}), \quad \forall j \in \mathcal{J}, \tag{0.5}$$

$$\texttt{AlwaysEqual}(l_m,\ x_{jm},\ col_j), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}, \tag{0.6}$$

$$\texttt{AlwaysIn}(l_m,\ x_{j,m-1},\ col_j + \delta + 1,\ |\mathcal{C}| + \delta + 1), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}^*), \tag{0.7}$$

$$\texttt{PresenceOf}(y_j) \implies \texttt{PresenceOf}(y_{j-1}), \quad \forall j \in \mathcal{J}^*, \tag{0.8}$$

$$\texttt{Max}(\texttt{EndOf}(\mathbf{y}_.)) \leq H. \tag{0.9}$$

To better understand the optimization model, let us explain its primary components:

The objective (0.1) seeks to maximize the count of tasks scheduled within the defined planning horizon.

Constraint set (0.2) guarantees that any scheduled task is uniquely allocated to one machine.

Constraint set (0.3) mandates non-overlapping execution of tasks on the same machine.

Constraint set (0.4) upholds the prescribed drilling sequence within each column.

Constraint set (0.5) enforces a safety distance between active drilling machines.

Constraint set (0.6) monitors and records the position of each machine during its drilling operation.

Constraint set (0.7) preserves the relative positions and safety gaps between operating machines.

Constraint set (0.8) ensures a continuous schedule by requiring that if a task y_j is scheduled, the preceding task y_{j-1} must also be scheduled. This prevents gaps in the drilling sequence where tasks might otherwise be skipped.

Constraint (0.9) limits the completion time of all scheduled tasks to be within the defined planning horizon H, ensuring that the entire schedule fits within the allocated timeframe.

In the following subsections, we will introduce each of the non-deterministic models. While these models share a significant portion of the same notation, we will reiterate the model's notation in each subsection for the sake of clarity and ease of reference. To emphasize the distinctions between them, we will use blue text to highlight the specific parts that vary within each model.

4.2 Two-stage stochastic model

Two-stage stochastic programming is a well-established approach for dealing with uncertainty in optimization problems, introduced by Dantzig (1955). This seminal work laid the foundation for handling uncertainty in linear programming, which was later extended to more general optimization problems. The method is particularly suited to problems where decisions must be made before the realization of uncertain parameters, which aligns well with the drill rig scheduling problem where task durations are not known with certainty at the time of planning.

Our first extension involves a two-stage stochastic CP model, where uncertainty in drilling times is considered. This approach builds on the concepts of stochastic constraint programming as discussed by Walsh (2002) and further developed in the scenario-based framework proposed by Tarim et al. (2006). The drilling times are modeled as random variables, and we aim to optimize the expected performance under this stochastic setting. The key points of this approach are as follows:

First stage decision : In the first stage, decisions are made based on the available information and known parameters such as the predetermined drilling locations and the set of available machines. Within our model, the first stage decision variables are the sequence variables. These variables serve a dual purpose: they denote task assignments to machines and also dictate the order in which tasks are executed on these machines. To maintain consistency across all possible realizations of the uncertain parameter, we enforce a uniform sequencing of tasks using the global constraint `SameSequence`.

Scenario set : Following the first stage of decision-making, uncertain events occur. To represent this uncertainty, we define a set of equiprobable scenarios, $\mathcal{S}$, with each scenario, s, representing a unique realization of drilling durations.

Second stage decision : After observing the actual realization of uncertainty (which corresponds to one of the scenarios), a second decision is made. In our model, second stage decisions consist in determining the start and end times for each task.

Objective function : The aim of two-stage stochastic optimization is to find a decision strategy that minimizes or maximizes an objective function, taking into account both the first and second stage decisions, as well as the uncertainty in the outcomes. Specifically, our objective is to maximize the expected total number of scheduled tasks within a predetermined planning horizon.

This approach provides a solution to the DCP that performs well on average over the long term, as we detail in the formal two-stage constraint programming model that follows. Our model builds upon the classical two-stage stochastic programming framework, adapting it to the specific constraints and objectives of drill rig scheduling in open-pit mines.

Sets	
$\mathcal{S}$	set of scenarios, indexed by s.
$\mathcal{M}$	set of machines, indexed by $m \in \{1, 2, \ldots, \|\mathcal{M}\|\}$.
$\mathcal{M}^*$	set of machines excluding its first element.
$\mathcal{J}$	set of tasks, indexed by i and $j \in \{1, 2, \ldots, \|\mathcal{J}\|\}$.
$\mathcal{J}^*$	set of tasks excluding its first element.
$\mathcal{I}_j$	tasks that become inaccessible once task j starts.
$\mathcal{C}$	set of columns, indexed by $c \in \{1, 2, \ldots, \|\mathcal{C}\|\}$.
$\mathcal{J}_c$	tasks associated with column c, indexed by j.
$\mathcal{J}_c^-$	set $\mathcal{J}_c$ excluding its last element.

Parameters	
col_j	column associated with task j.
p^s_{jm}	processing duration of task j on machine m in scenario s.
t_{ijm}	setup duration between tasks i and j when on machine m.
δ	number of columns required as spacing between machine pairs.
H	planning horizon.
Variables	
x^s_{jm}	optional interval variable of size p^s_{jm} indicating if task j is performed by machine m in scenario s.
y^s_j	optional interval variable representing task j in scenario s.
z^s_m	sequence of interval variables x^s_{jm} for machine m in scenario s.
l^s_m	state function denoting the location of machine m in scenario s.

Model 1

$$\text{Maximize} \quad Z = \frac{\sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}} \texttt{PresenceOf}(y^s_j)}{|\mathcal{S}|} \tag{1.1}$$

$$\text{s.t.} \quad \texttt{Alternative}(y^s_j,\ \mathbf{x}^s_{j\cdot}), \quad \forall j \in \mathcal{J},\ \forall s \in \mathcal{S}, \tag{1.2}$$

$$\texttt{NoOverlap}(z^s_m,\ \mathbf{t}_{\cdot\cdot m}), \quad \forall m \in \mathcal{M},\ \forall s \in \mathcal{S}, \tag{1.3}$$

$$\texttt{EndBeforeStart}(y^s_j,\ y^s_{j+1}), \quad \forall c \in \mathcal{C},\ \forall j \in \mathcal{J}^-_c,\ \forall s \in \mathcal{S}, \tag{1.4}$$

$$\texttt{NoOverlap}(\{y^s_j \mid j \in \mathcal{I}_j\}), \quad \forall j \in \mathcal{J},\ \forall s \in \mathcal{S}, \tag{1.5}$$

$$\texttt{AlwaysEqual}(l^s_m,\ x^s_{jm},\ col_j), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S}, \tag{1.6}$$

$$\texttt{AlwaysIn}(l^s_m,\ x^s_{j,m-1},\ col_j + \delta + 1,\ |\mathcal{C}| + \delta + 1), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}^*,\ \forall s \in \mathcal{S}, \tag{1.7}$$

$$\texttt{PresenceOf}(y^s_j) \implies \texttt{PresenceOf}(y^s_{j-1}), \quad \forall j \in \mathcal{J}^*,\ \forall s \in \mathcal{S}, \tag{1.8}$$

$$\texttt{Max}(\texttt{EndOf}(\mathbf{y}^s_{\cdot})) \le H, \quad \forall s \in \mathcal{S}, \tag{1.9}$$

$$\texttt{SameSequence}(z^{s_1}_m,\ z^{s_2}_m), \quad \forall m \in \mathcal{M},\ \forall s_1,\ s_2 \in \mathcal{S},\ s_1 \neq s_2. \tag{1.10}$$

To better understand this first extension, let us explain its primary components:

The objective (1.1) is to maximize the expected total number of tasks scheduled within the planning horizon.

Formulation (1.2–1.9) enforces, for each scenario, the same constraints as found in formulation (0.2–0.9)

Constraint set `SameSequence`($sequence_1$, $sequence_2$) **(1.10)** enforces that the interval variables comprising both $sequence_1$ and $sequence_2$ maintain identical relative positions. For example, if task a precedes task b in $sequence_1$, then its counterpart a' must also precede b' in $sequence_2$. This constraint is crucial for ensuring non-anticipativity in the model, a key principle in two-stage stochastic optimization Birge and Louveaux (2011). Non-anticipativity requires that decisions made in the first stage must be the same for all scenarios before the uncertainty is revealed. By ensuring that the order of tasks on each machine remains consistent across all scenarios, regardless of the specific realizations of drilling times, the `SameSequence` constraint prevents the model from making first-stage decisions (task sequencing) based on second-stage information (actual drilling times). This maintains the integrity and feasibility of the two-stage stochastic formulation.

In the context of drill rig scheduling, this two-stage stochastic model allows for more robust decision-making under uncertainty. The first-stage decisions (task sequencing) provide a consistent schedule structure, while the second-stage decisions (start and end times) allow for adaptability to the realized drilling times. This balance between consistency and flexibility is crucial in mining operations where adhering to a general plan while accommodating unexpected delays or efficiencies is important.

4.3 Probability-free model

In contrast to the stochastic model, our probability-free model considers a deterministic representation of uncertainty. We aim to find a solution that remains feasible across various scenarios of task durations while maintaining computational tractability. To achieve this, we introduce a constraint in our deterministic model, which we refer to as the `Resilient` constraint. This constraint places a lower limit on the permissible delay in drilling duration, controlled by a user-defined threshold denoted as α (ranging from 0 to 1). The `Resilient` constraint is formally defined as:

$$\texttt{Resilient}([X_1, \ \dots, \ X_n], \ [D_1, \ \dots, D_n], \ \alpha) \Leftrightarrow \sum_{i=1}^{n} X_i \geq \alpha \sum_{i=1}^{n} D_i. \tag{1}$$

Here, the variables $X_1, \dots, X_n$ are our decision variables representing uncertain parameters, while $D_1, \dots, D_n$ signify the maximum allowable deviation from their respective minimal values. The parameter α serves as a threshold governing the degree of resilience we desire.

While our model represents uncertain parameters with sets, it differs significantly from traditional robust optimization techniques. The key differences are:

- **Objective**: Robust optimization typically aims to optimize for the worst-case scenario within an uncertainty set. Our model, in contrast, seeks to ensure a minimum level of delay tolerance across various scenarios, providing a more flexible approach to uncertainty.
- **Constraint handling**: Robust optimization typically aims to optimize for the worst-case scenario within an uncertainty set. Our model, in contrast, seeks to ensure a minimum level of delay tolerance across various scenarios.
- **Solution approach**: Robust optimization often requires reformulation of the original problem. Our model modifies the original constraint programming formulation by:
  - Introducing a new variable for each uncertain parameter.
  - Adding the `Resilient` constraint.

  This approach maintains much of the original problem structure while incorporating uncertainty considerations.
- **Uncertainty distribution**: Our approach distributes uncertainty across variables differently than typical robust optimization approaches, which often focus on worst-case scenarios.
- **Scope of uncertainty consideration**: Our model considers the global impact of uncertainty (total delay across all tasks), while robust optimization often addresses local worst-cases for individual constraints or variables.
- **Risk handling**: Our approach ensures a certain level of deviation by requiring a minimum total deviation, which differs from typical robust optimization approaches.

The parameter α in our model serves a similar role to the uncertainty budget in robust optimization Bertsimas and Sim (2004), but with a different interpretation. While an uncertainty budget in robust optimization typically limits the maximum deviation, our α parameter ensures a minimum level of total deviation.

To implement this feature, we decompose the drilling time into two distinct components for a given task j and machine m. The first component is the minimal drilling duration, denoted as p_{jm}^{min}. This is a predefined parameter reflecting the shortest possible time for completing the drilling operation under ideal conditions. The second component, d'_{jm}, represents the delay. This delay is stochastic in nature, accounting for various unpredictable factors that may prolong the drilling process. Therefore, the total drilling duration, p_{jm}, for task j and machine m is calculated as the sum of these two components:

$p_{jm} = p_{jm}^{min} + d'_{jm}$. In this equation, p_{jm}^{min} is a given constant, while d'_{jm} is a decision variable that reflects the uncertain aspects of the drilling duration. The resulting `Resilient` constraint is:

$$\sum_{j \in \mathcal{J}} d'_{jm} \geq \alpha \sum_{j \in \mathcal{J}} d_{jm}^{max} \quad \forall m \in \mathcal{M}. \tag{2}$$

Although the constraint includes the "greater than or equal" sign, the solver avoids overestimating the total delay due to the influence of the objective function and the upper bounds on individual delays. Specifically, any increase in delay adversely impacts the total number of tasks that can be completed within the planning horizon, and each task-specific delay d'_{jm} is constrained by a maximum allowable delay d_{jm}^{max}. Consequently, this guides the solver to keep the delay at a minimum, while still meeting the threshold set by the `Resilient` constraint.

Such an approach is suitable for any deterministic CP model where the effects of uncertain parameters on the objective function are clearly discernible. The implementation involves following these steps:

1. Identify the uncertain parameters.
2. Replace each uncertain parameter with a variable whose range encompasses possible realizations of the uncertain parameter.
3. When an increase in the uncertain parameter *negatively* affects the objective function, impose that the total value of the variables representing the uncertain parameter should *exceed* the sum of the *upper* limits of the range, scaled by a factor α (which ranges from 0 to 1).
4. When an increase in the uncertain parameter *positively* affects the objective function, impose that the total value of the variables representing the uncertain parameter should remain *below* the total of the *lower* limits of the range, scaled by a factor $\alpha \in [0,\ 1]$.

Note 3. Selecting α as 1 corresponds to optimizing for the worst-case scenario, aligning our approach with traditional robust optimization in this extreme case.

The simplicity and accessibility of the resilient model make it particularly valuable for practitioners and decision-makers who may not have extensive training in operations research (OR). As highlighted in Gurobi's State of Mathematical Optimization Report 2023 Gurobi Optimization (2023), only half (49%) of the surveyed commercial customers have an OR educational background. While this report specifically focuses on Gurobi's mathematical optimization software, it is reasonable to assume that this trend of limited OR education among practitioners is likely similar across the broader optimization community, including those working with constraint programming techniques.

This observation underscores the need for optimization approaches that can be readily adopted and applied by a wider audience, including those without formal OR education. The resilient model addresses this need by providing a straightforward process for transforming any deterministic constraint programming model into one that effectively accounts for uncertainty. By contrast, alternative approaches such as two-stage stochastic programming, chance-constrained optimization, or traditional robust optimization often require a deeper understanding of theoretical concepts, which may hinder their adoption among practitioners lacking specialized knowledge.

Thus, the resilient model offers a practical and accessible solution for incorporating uncertainty into optimization models, empowering a broader range of decision-makers to leverage the benefits of resilient optimization techniques in their respective domains. It maintains the original problem structure with just an additional constraint, potentially making it easier to implement in existing CP solvers while providing a flexible approach to handling uncertainty. This method provides a resilient solution to the DCP, as elaborated in the subsequent model description.

Sets	
$\mathcal{M}$	set of machines, indexed by $m \in \{1, 2, \ldots, \lvert\mathcal{M}\rvert\}$.
$\mathcal{M}^*$	set of machines excluding its first element.
$\mathcal{J}$	set of tasks, indexed by i and $j \in \{1, 2, \ldots, \lvert\mathcal{J}\rvert\}$.
$\mathcal{J}^*$	set of tasks excluding its first element.
$\mathcal{I}_j$	tasks that become inaccessible once task j starts.
$\mathcal{C}$	set of columns, indexed by $c \in \{1, 2, \ldots, \lvert\mathcal{C}\rvert\}$.
$\mathcal{J}_c$	tasks associated with column c, indexed by j.
$\mathcal{J}_c^-$	set $\mathcal{J}_c$ excluding its last element.
Parameters	
col_j	column associated with task j.
p_{jm}^{min}	minimum processing duration of task j on machine m.
d_{jm}^{max}	maximum allowable delay for processing task j on machine m.
t_{ijm}	setup duration between tasks i and j on machine m.
δ	number of columns required as spacing between machine pairs.
α	control parameter (ranging from 0 to 1) influencing model resilience.
H	planning horizon.
Variables	
x_{jm}	optional interval variable indicating if task j is performed by machine m.
d'_{jm}	integer variable representing the delay in the processing of task j on machine m, $d'_{jm} \in [0,\ d_{jm}^{max}]$
y_j	optional interval variable representing a task j.
z_m	sequence of interval variables x_{jm} for machine m.
l_m	state function denoting the location of machine m.

Model 2

$$\text{Maximize} \quad Z = \sum_{j \in \mathcal{J}} \texttt{PresenceOf}(y_j) \tag{2.1}$$

$$\text{s.t.} \quad \texttt{Alternative}(y_j,\ \mathbf{x}_{j.}), \quad \forall j \in \mathcal{J}, \tag{2.2}$$

$$\texttt{NoOverlap}(z_m,\ \mathbf{t}_{..m}), \quad \forall m \in \mathcal{M}, \tag{2.3}$$

$$\texttt{EndBeforeStart}(y_j,\ y_{j+1}), \quad \forall c \in \mathcal{C},\ \forall j \in \mathcal{J}_c^-, \tag{2.4}$$

$$\texttt{NoOverlap}(\{y_j \mid j \in \mathcal{I}_j\}), \quad \forall j \in \mathcal{J}, \tag{2.5}$$

$$\texttt{AlwaysEqual}(l_m,\ x_{jm},\ col_j), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}, \tag{2.6}$$

$$\texttt{AlwaysIn}(l_m,\ x_{j,m-1},\ col_j + \delta + 1,\ \lvert\mathcal{C}\rvert + \delta + 1), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}^*, \tag{2.7}$$

$$\texttt{PresenceOf}(y_j) \implies \texttt{PresenceOf}(y_{j-1}), \quad \forall j \in \mathcal{J}^*, \tag{2.8}$$

$$\texttt{Max}(\texttt{EndOf}(\mathbf{y}_.)) \leq H, \tag{2.9}$$

$$\texttt{SizeOf}(x_{jm}) = (p_{mj}^{min} + d'_{jm}) \cdot \texttt{PresenceOf}(x_{jm}), \quad \forall m \in \mathcal{M},\ \forall j \in \mathcal{J}, \tag{2.10}$$

$$\texttt{Resilient}(\mathbf{d'}_{.m},\ \mathbf{d}_{.m}^{max},\ \alpha), \quad \forall m \in \mathcal{M}, \tag{2.11}$$

$$0 \leq d'_{jm} \leq d_{jm}^{max} \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}. \tag{2.12}$$

To better understand this second extension, let us explain its primary components:

Formulation (2.1–2.9) is identical to the deterministic formulation (0.1–0.9).

Constraint set (2.10) defines the duration of each task, on each machine, as the sum of the known minimal duration and the delay variable.

Constraint set (2.11) ensures that accumulated delays of scheduled tasks meet or exceed a proportion α of the total allowable delay, where α is a fraction between 0 and 1.

Constraint set (2.12) define the domains of the delay variables.

In the context of the Drill Coordination Problem (DCP), our resilient model addresses the uncertainty in drilling durations while preserving the core structure and constraints of the original problem. By representing uncertain drilling times as the sum of a minimum duration and a delay variable, we can apply the `Resilient` constraint to ensure a balanced distribution of delays across tasks and machines.

This approach is particularly valuable in open-pit mining operations, where geological conditions can vary unpredictably. The α parameter provides mine planners with a tool to adjust the balance between optimistic and conservative schedules, allowing for tailored risk management strategies. Importantly, this method maintains crucial safety and operational constraints such as machine spacing and movement restrictions, while offering a computationally tractable way to account for drilling uncertainties. The resulting model aims to maximize the number of drilled targets within the planning horizon, producing schedules that can proactively absorb a certain level of delays without compromising overall performance or safety requirements.

4.4 Chance-constrained model

Chance-constrained optimization, introduced by Charnes and Cooper (1959), provides a framework for dealing with uncertainty in optimization problems by allowing constraints to be violated with a small probability. This approach is particularly well-suited to mining operations, where the need to maintain operational efficiency must be balanced against the inherent uncertainties in geological conditions and equipment performance. Our approach builds on the scenario-based approximation of chance constraints Nemirovski and Shapiro (2006), but innovates by incorporating a resilience mechanism. This novel combination allows for a more nuanced treatment of uncertainty in drill rig scheduling, balancing the need for operational efficiency with the desire for schedule resilience.

Building upon the two-stage stochastic model and the resilient approach, we propose a chance-constrained resilient model. This hybrid approach aims to balance the benefits of stochastic optimization with the robustness of the resilient model, while avoiding overly conservative solutions. The key idea is to enforce a `Resilient` constraint on a subset of scenarios, similar to a chance constraint in stochastic programming. This ensures a certain level of resilience without requiring it for every possible scenario, which could be unnecessarily restrictive.

To do so, we introduce a constraint called `ConditionalResilient`, which incorporates a binary variable for scenario selection:

$$\texttt{ConditionalResilient}([X_1, \ \ldots, \ X_n], [D_1, \ \ldots, \ D_n], \ \alpha, \ b) \Leftrightarrow \sum_{i=1}^{n} X_i \geq (\alpha \cdot b) \sum_{i=1}^{n} D_i \tag{3}$$

Here, b is a binary variable that determines whether the `Resilient` constraint is enforced ($b = 1$) or relaxed ($b = 0$) for a given scenario.

Key components of the model:

1. **Scenario set**: A set of scenarios $\mathcal{S}$, each representing a possible realization of drilling times.
2. **Conditional resilient constraint**: The `ConditionalResilient` constraint applied for each machine and scenario.
3. **Scenario selection**: Binary variables b_s for each scenario $s \in \mathcal{S}$, where $b_s = 1$ if the `Resilient` constraint is enforced for scenario s, and 0 otherwise.
4. **Chance constraint on resilience**: Ensuring the resilient condition is met for a specified proportion of scenarios:
$$\sum_{s \in \mathcal{S}} b_s \geq \lceil \beta \cdot |\mathcal{S}| \rceil \tag{4}$$
where $\beta \in [0, 1]$ is the desired proportion of scenarios that should satisfy the `Resilient` constraint.
5. **Special case**: When $\beta = 1$, this model becomes equivalent to a two-stage resilient model, as the `ConditionalResilient` constraint is enforced for all scenarios.

This approach complements recent advancements in incorporating chance-constrained programming into constraint programming (CP). In particular, Mercier-Aubin et al. (2020) introduced the `Confidence` constraint, a new global constraint in CP that implements chance-constrained programming concepts. Their method offers a way to model chance constraints without relying on scenario approximation. The authors present a linear-time filtering algorithm that leverages the cumulative distribution functions (CDFs) of the random variables to iteratively refine variable domains. This approach ensures that the constraint is satisfied with a given probability, enhancing its practical applicability in CP solvers. Moreover, they demonstrate its effectiveness in an industrial case study in the textile industry, showing its potential for real-world applications. However, their method assumes independence between random variables, which may not always hold in complex real-world scenarios. In contrast, our scenario-based approach does not require this assumption of independence, potentially allowing for more flexible modeling of interdependent uncertainties in drilling operations, as elaborated in the subsequent model description.

Sets	
$\mathcal{S}$	set of scenarios, indexed by s.
$\mathcal{M}$	set of machines, indexed by $m \in \{1, 2, \ldots, \lvert\mathcal{M}\rvert\}$.
$\mathcal{M}^*$	set of machines excluding its first element.
$\mathcal{J}$	set of tasks, indexed by i and $j \in \{1, 2, \ldots, \lvert\mathcal{J}\rvert\}$.
$\mathcal{J}^*$	set of tasks excluding its first element.
$\mathcal{I}_j$	tasks that become inaccessible once task j starts.
$\mathcal{C}$	set of columns, indexed by $c \in \{1, 2, \ldots, \lvert\mathcal{C}\rvert\}$.
$\mathcal{J}_c$	tasks associated with column c, indexed by j.
$\mathcal{J}_c^-$	set $\mathcal{J}_c$ excluding its last element.
Parameters	
col_j	column associated with task j.
$p_{jm}^{min,s}$	minimum processing duration of task j on machine m in scenario s.
d_{jm}^{max}	maximum allowable delay for processing task j on machine m.
t_{ijm}	setup duration between tasks i and j when on machine m.
δ	number of columns required as spacing between machine pairs.
α	control parameter (ranging from 0 to 1) influencing model resilience.
β	desired proportion of scenarios that should satisfy the `Resilient` constraint.
H	planning horizon.
Variables	
x_{jm}^s	optional interval variable indicating if task j is performed by machine m in scenario s.
d'^s_{jm}	integer variable representing the delay in the processing of task j on machine m in scenario s, $d'^s_{jm} \in [0, d_{jm}^{max}]$.
b_s	binary variable indicating if the `Resilient` constraint is enforced for scenario s.
y_j^s	optional interval variable representing task j in scenario s.
z_m^s	sequence of interval variables x_{jm}^s for machine m in scenario s.
l_m^s	state function denoting the location of machine m in scenario s.

Model 3

$$\text{Maximize} \quad Z = \frac{\sum_{s\in\mathcal{S}} \sum_{j\in\mathcal{J}} \texttt{PresenceOf}(y_j^s)}{\lvert\mathcal{S}\rvert} \tag{3.1}$$

$$\text{s.t.} \quad \texttt{Alternative}(y_j^s,\ \mathbf{x}_{j\cdot}^s), \quad \forall j \in \mathcal{J},\ \forall s \in \mathcal{S}, \tag{3.2}$$

$$\texttt{NoOverlap}(z_m^s,\ \mathbf{t}_{\cdot\cdot m}), \quad \forall m \in \mathcal{M},\ \forall s \in \mathcal{S}, \tag{3.3}$$

$$\texttt{EndBeforeStart}(y_j^s,\ y_{j+1}^s), \quad \forall c \in \mathcal{C},\ \forall j \in \mathcal{J}_c^-,\ \forall s \in \mathcal{S}, \tag{3.4}$$

$$\texttt{NoOverlap}(\{y_j^s \mid j \in \mathcal{I}_j\}), \quad \forall j \in \mathcal{J},\ \forall s \in \mathcal{S}, \tag{3.5}$$

$$\texttt{AlwaysEqual}(l_m^s,\ x_{jm}^s,\ col_j), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S}, \tag{3.6}$$

$$\texttt{AlwaysIn}(l_m^s,\ x_{j,m-1}^s,\ col_j + \delta + 1,\ \lvert\mathcal{C}\rvert + \delta + 1), \quad \forall j \in \mathcal{J},\ \forall m \in \mathcal{M}^*,\ \forall s \in \mathcal{S}, \tag{3.7}$$

$$\texttt{PresenceOf}(y_j^s) \implies \texttt{PresenceOf}(y_{j-1}^s), \quad \forall j \in \mathcal{J}^*, \ \forall s \in \mathcal{S}, \tag{3.8}$$

$$\texttt{Max}(\texttt{EndOf}(\mathbf{y}_\cdot^s)) \leq H, \quad \forall s \in \mathcal{S}, \tag{3.9}$$

$$\texttt{SameSequence}(z_m^{s_1}, \ z_m^{s_2}), \quad \forall m \in \mathcal{M}, \ \forall s_1, \ s_2 \in \mathcal{S}, \ s_1 \neq s_2, \tag{3.10}$$

$$\texttt{SizeOf}(x_{jm}^s) = (p_{jm}^{min,s} + d_{jm}'^s) \cdot \texttt{PresenceOf}(x_{jm}^s), \quad \forall j \in \mathcal{J}, \ \forall m \in \mathcal{M}, \ \forall s \in \mathcal{S}, \tag{3.11}$$

$$\texttt{ConditionalResilient}(\mathbf{d'}_{\cdot m}^s, \ \mathbf{d}_{\cdot m}^{max}, \ \alpha, \ b_s), \quad \forall m \in \mathcal{M}, \ \forall s \in \mathcal{S}, \tag{3.12}$$

$$\sum_{s \in \mathcal{S}} b_s \geq \lceil \beta \cdot |\mathcal{S}| \rceil, \tag{3.13}$$

$$b_s \in \{0, 1\}, \quad \forall s \in \mathcal{S}, \tag{3.14}$$

$$0 \leq d_{jm}'^s \leq d_{jm}^{max}, \quad \forall j \in \mathcal{J}, \ \forall m \in \mathcal{M}, \ \forall s \in \mathcal{S}. \tag{3.15}$$

The key components of this model are:

The objective function (3.1) maximizes the average number of tasks completed across all scenarios. This formulation aims to optimize the overall performance of the schedule, considering all possible scenarios equally. By focusing on the average task completion across all scenarios, the model seeks to create schedules that perform well under various conditions.

Constraints (3.2–3.10) ensure the basic structure and feasibility of the schedule across all scenarios.

Constraint (3.11) defines the duration of each task, including the potential delay.

Constraint (3.12) applies the `ConditionalResilient` constraint for each machine and scenario, enforcing resilience only when $b_s = 1$.

Constraint (3.13) ensures that the `Resilient` constraint is satisfied for at least β proportion of the scenarios.

Constraints (3.14) and (3.15) define the domains of the new variables.

In the context of the Drill Coordination Problem (DCP), the chance-constrained resilient model offers a nuanced approach to handling uncertainty in drilling times. By combining elements of stochastic programming, resilient optimization, and chance-constrained programming, it creates a flexible framework for drill rig scheduling that balances efficiency and robustness. This model allows mine planners to tailor schedules to their specific risk tolerance and operational priorities by adjusting parameters α and β. For instance, a lower β might be chosen during periods of stable geological conditions, allowing for more aggressive scheduling, while a higher β could be used when facing more uncertain terrain, ensuring greater overall schedule resilience. It is worth noting that when $\beta = 1$, this model reduces to a two-stage resilient model, providing a direct link between these approaches and demonstrating the flexibility of our formulation.

Compared to the two-stage stochastic model and the probability-free resilient model, this approach offers a middle ground. It maintains the scenario-based structure of the stochastic model while incorporating the delay tolerance mechanism of the resilient model, all within a framework that allows for controlled violation of the resilience constraint. This flexibility makes it particularly suitable for the dynamic and uncertain environment of open-pit mining operations, where conditions can vary significantly and rapid adaptability is crucial.

5 Computational experiments

This section presents the computational results obtained from testing our three constraint programming models on simulated instances derived from a coal mine's drilling data. The primary aim is to evaluate and compare the performance of the deterministic model (Model 0), the two-stage stochastic model (Model 1), the probability-free model (Model 2), and the new chance-constrained resilient model (Model 3), under various scenarios reflecting the inherent uncertainties in drill rig scheduling.

5.1 Experimental setup

To thoroughly evaluate the performance of our proposed models, we conducted a comprehensive set of computational experiments using simulated instances derived from real-world drilling data obtained from a coal mine over a nine-month period. Our experimental process consists of two main phases: model solving and solution evaluation, as illustrated in Figure 3.

5.1.1 Phase 1: Model solving

As shown in Figure 3, we begin with historical data from which we generate model instances. To ensure our experiments cover a representative range of drilling activities, we first analyzed the distribution of daily drilling operations in our historical data. Figure 2 presents a histogram of the number of holes drilled per day based on our historical data.

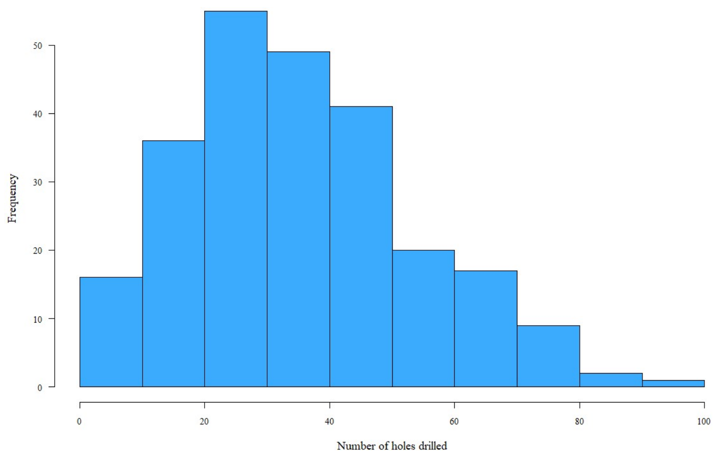

Figure 2: Histogram of the number of holes drilled per day.

As shown in Figure 2, the number of holes drilled per day in our historical data ranges from 0 to about 100, with the majority of days seeing between 20 and 60 holes drilled. Based on this distribution, we chose to generate instances with task numbers ranging from 20 to 110, covering the full spectrum of observed daily drilling activities, from light days to exceptionally busy ones. This range ensures that our models are tested on scenarios that reflect both typical operations and more extreme cases, providing a comprehensive evaluation of their performance and adaptability.

In addition to the number of holes drilled per day, we also analyzed the distribution of individual drilling durations. Here is a summary of the drilling duration distribution (in minutes):

- Minimum: 4
- 25th percentile: 9
- Median: 13
- Mean: 14.14
- 75th percentile: 19
- Maximum: 59
- Standard deviation: 8.14

This distribution reflects the variability in drilling times that our models aim to address, ranging from quick operations lasting just a few minutes to more time-consuming tasks that may take nearly an hour.

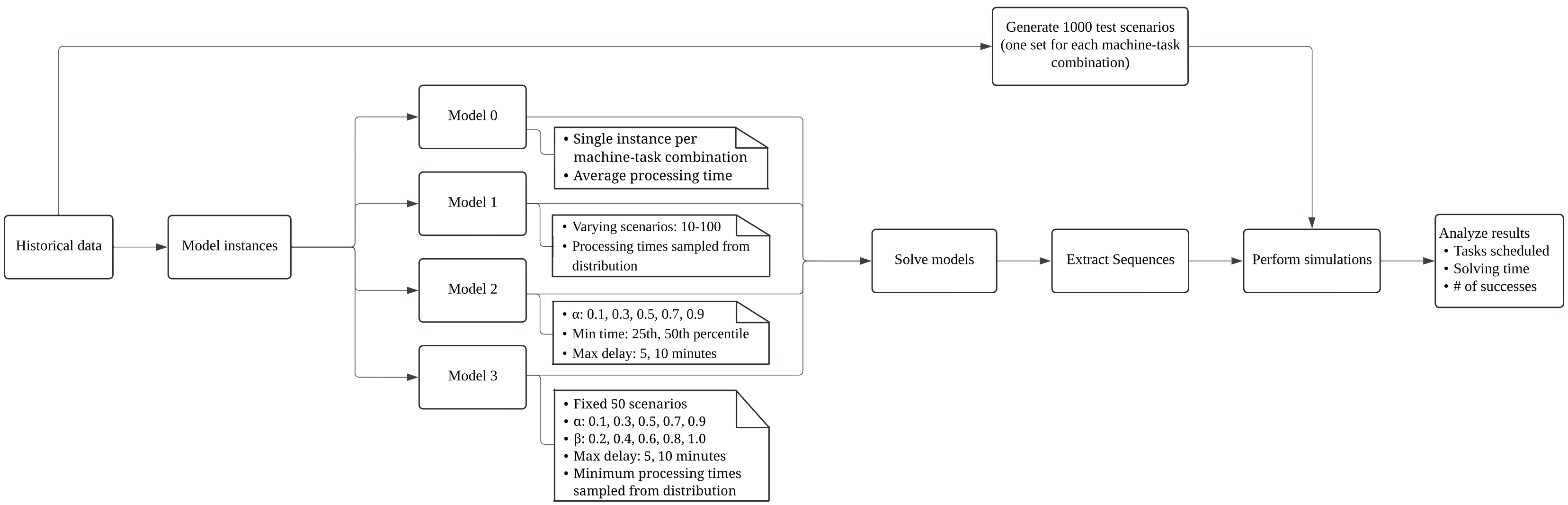

Figure 3: Flowchart of the experimental setup process.

With these insights from our historical data analysis, we proceeded to generate instances for our four models. For each model, we created instances with 2 or 3 machines and varying numbers of tasks (20, 30, 40, 50, 60, 70, 80, 90, 100, or 110), covering the full spectrum of observed daily drilling activities. Each model approaches the uncertainty in drilling durations differently, and our instance generation process reflects these different approaches. The following subsections detail how instances were generated for each specific model:

Model 0 (Deterministic). For each combination of machine and task numbers, we created a single instance with processing times set to the average observed drilling duration. This resulted in a total of 20 instances.

Model 1 (Two-stage stochastic). For each combination of machine and task numbers, we created ten instances with varying numbers of scenarios (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100). Processing times for each scenario were sampled from the historical distribution. This resulted in a total of 200 instances.

Model 2 (Probability-free). We generated instances with varying values of the resilience parameter α (0.1, 0.3, 0.5, 0.7, 0.9). We used two settings for the minimum processing time: the 25th and 50th percentiles of the historical data. These percentiles were chosen to represent conservative and moderate estimates of minimum processing times, respectively. Maximum delays were set to 5 and 10 minutes for all tasks. These delay values were determined through an empirical process, testing various values to find a range that, in conjunction with other parameters, provided good performance in preliminary simulation tests. This approach allows us to evaluate the model's robustness to both smaller and larger variations in task duration. This resulted in a total of 400 instances.

Model 3 (Chance-constrained resilient). Instances were created using a fixed set of 50 scenarios. Minimum processing times for each scenario were sampled from the historical distribution, similar to Model 1. We varied the resilience parameter α (0.1, 0.3, 0.5, 0.7, 0.9) and the proportion parameter β (0.2, 0.4, 0.6, 0.8, 1.0). For each instance, the maximum delay was set uniformly to either 5 or 10 minutes for all tasks, consistent with the delay settings used in Model 2. This consistency allows for a more direct comparison between the probability-free and chance-constrained approaches. This resulted in a total of 1000 instances.

In total, we generated and solved 1,620 instances across all four models. The planning horizon for all instances was set to 24 hours (1440 minutes), with time discretized into 1-minute intervals. After solving these instances, we extracted the task sequencing information from the obtained solutions, as depicted in the flowchart.

5.1.2 Phase 2: Solution evaluation

To assess the robustness and adaptability of the obtained sequences, we evaluated their performance against a common set of test scenarios. As illustrated in Figure 3, we generated a set of 1000 test scenarios for each combination of machines and tasks, with processing times sampled from the historical distribution to ensure a realistic representation of the observed variability. For each solution obtained in Phase 1 (where a solution represents a sequence of tasks for each machine), we evaluated its performance on these 1000 test scenarios by solving a constraint satisfaction problem. This problem aimed to find feasible start and end times for each task within the 24-hour planning horizon, adhering to the predetermined sequencing and ensuring that all constraints were satisfied. In total, this resulted in 1,620,000 simulations, providing a broad basis for evaluating our models' performance.

As shown in the final step of our flowchart, we collected the following metrics common to all models for each evaluation:

- Number of tasks scheduled within the time horizon.
- Computational time for solving the original instance in Phase 1.
- Number of scenarios where the pre-determined sequencing was feasible.

It is important to note that the computational time metric refers specifically to the time taken to solve each original problem instance in Phase 1 and obtain the initial solution, not including the time for subsequent scenario evaluations in Phase 2. In addition to these common metrics, we also collected and analyzed model-specific metrics, which are discussed in detail in the respective sections for each model.

This two-phase approach, clearly visualized in Figure 3, allows us to compare the performance of all four models across a wide range of scenarios and uncertainty levels, providing insights into the trade-offs between solution quality, computational time, and robustness for each approach.

5.2 Results and analysis

This section presents a comprehensive analysis of the computational results obtained from our four constraint programming models: the deterministic model (Model 0), the two-stage stochastic model (Model 1), the probability-free model (Model 2), and the chance-constrained resilient model (Model 3). We evaluate each model's performance across various metrics, including computational efficiency, solution quality, robustness to uncertainty, and scalability. Our analysis aims to provide insights into the strengths and limitations of each approach in addressing the inherent uncertainties in drill rig scheduling. We begin with the deterministic model as a baseline and progressively examine the more sophisticated models, culminating in a comparative analysis that highlights the trade-offs between the different approaches. This systematic evaluation will help inform the selection of the most appropriate model for different operational scenarios in drilling operations.

5.2.1 Deterministic model

The performance of the deterministic model is presented in Table 2. This model provides optimal solutions when the drilling durations are known with certainty and equal to their average observed values.

Our analysis of the deterministic model reveals several key insights:

Computational efficiency. The model demonstrates excellent solving times across all instance sizes, with all problems solved in less than one second. This indicates high computational efficiency regardless of the problem size or number of machines.

Feasibility. The model shows perfect feasibility (1000 out of 1000 scenarios) for instances up to 70 tasks, regardless of the number of machines. However, feasibility begins to decrease for larger instances:

- 80 tasks: Still highly feasible (997/1000 for 2 machines, 996/1000 for 3 machines).
- 90 tasks: Feasibility drops to about 78% (785/1000 for 2 machines, 773/1000 for 3 machines).
- 100 tasks: Further decrease to about 59% (597/1000 for 2 machines, 576/1000 for 3 machines).
- 110 tasks: Significant drop to 44% (440/1000) for 2 machines and 26.2% (262/1000) for 3 machines.

Impact of problem size. The model's performance in terms of feasibility clearly deteriorates as the number of tasks increases beyond 80, suggesting limitations in handling larger, more complex scheduling scenarios.

Machine quantity. Interestingly, increasing the number of machines from 2 to 3 does not consistently improve feasibility, and in some cases, slightly reduces it. This result stems from the increased complexity in resource allocation and scheduling constraints when an additional machine is introduced.

These findings highlight that while the deterministic model is computationally efficient, its ability to produce feasible solutions decreases for larger problem instances. This underscores the challenges in deterministic approaches when handling the inherent variability in drill rig scheduling, particularly for larger-scale operations. The model's performance provides a solid foundation for comparing with more advanced modeling approaches in subsequent sections.

Table 2: Results for the deterministic model

Tasks	2 machines		3 machines	
	Time (seconds)	Feasible	Time (seconds)	Feasible
20	¡ 1	1000	¡ 1	1000
30	¡ 1	1000	¡ 1	1000
40	¡ 1	1000	¡ 1	1000
50	¡ 1	1000	¡ 1	1000
60	¡ 1	1000	¡ 1	1000
70	¡ 1	1000	¡ 1	1000
80	¡ 1	997	¡ 1	996
90	¡ 1	785	¡ 1	773
100	¡ 1	597	¡ 1	576
110	¡ 1	440	¡ 1	262

5.2.2 Two-stage stochastic model

The two-stage stochastic model was tested with varying numbers of scenarios (10 to 100) for different combinations of tasks (20 to 110) and machines (2 and 3). This model optimizes the task sequencing considering the uncertainty in drilling durations by incorporating multiple scenarios. The model can schedule all tasks for instances with up to 110 tasks, regardless of the number of machines (2 or 3), while ensuring the feasibility of the obtained task sequencing across most evaluated scenarios. A summary of the key findings is presented here, with the full results available in Appendix A.

Computational efficiency: Solving times generally increased with the number of tasks and scenarios, with a notable turning point observed around 90 tasks. For instance, with 100 scenarios:

- For 2 machines:
  - 20 tasks: 7.61s.
  - 80 tasks: 28.86s.
  - 90 tasks: 369.84s (marking a significant turning point).
  - 110 tasks: 1652.06s.
- For 3 machines:
  - 20 tasks: 9.98s.
  - 80 tasks: 16.70s.
  - 90 tasks: 528.74s (similar turning point).
  - 110 tasks: 705.63s.

The behavior is similar for both 2 and 3 machines, with a sharp increase in solving time around 90 tasks. However, the increase for larger instances is less dramatic with 3 machines. Notably, even the longest solving time was less than 30 minutes, indicating reasonable computational efficiency for practical use.

Feasibility: The model demonstrated high feasibility across most instances:

- Perfect feasibility (1000/1000 scenarios) for most instances up to 70 tasks.
- Slight decrease in feasibility for 80 tasks (consistently 997/1000 for 2 machines, 996/1000 for 3 machines).

- High performance maintained for larger instances (90-110 tasks), with feasibility often reaching 1000/1000, especially with higher numbers of scenarios.

Impact of scenario number: Increasing the number of scenarios generally improved solution robustness, particularly for larger instances (90-110 tasks). Perfect feasibility was often achieved with 60 or more scenarios, and in some cases, even with fewer scenarios.

Machine quantity: The impact of increasing from 2 to 3 machines varied:

- Generally similar feasibility for smaller instances (up to 80 tasks).
- For larger instances (90-110 tasks), both 2 and 3 machines often achieved high feasibility, especially with higher numbers of scenarios.

The increased solving times for larger instances can be attributed to the complexity introduced by considering multiple scenarios. As the number of tasks grows, the model must account for a larger number of potential realizations of drilling durations, leading to a more computationally demanding optimization process.

Overall, the two-stage stochastic model demonstrates its effectiveness in addressing the uncertainty in the drilling coordination problem. It provides robust task sequencing that remains feasible across most evaluated scenarios, even for larger instances with up to 110 tasks. The model's ability to achieve high feasibility, often perfect feasibility, for larger instances highlights its robustness in handling uncertain drilling durations.

These results suggest that the two-stage stochastic model offers improved robustness compared to the deterministic model, particularly for larger instances and with higher numbers of scenarios. The trade-off between solution quality and computational expense becomes more pronounced as the problem size and number of scenarios increase, but the model maintains high feasibility even for the largest instances tested.

Table 3 provides a snapshot of the model's performance for selected instance sizes and scenario numbers. The full results, including all scenario numbers and instance sizes, are available in Appendix A.

Table 3: Selected results for the two-stage stochastic model (10 and 100 scenarios).

Tasks	10 Scenarios				100 Scenarios			
	2 machines		3 machines		2 machines		3 machines	
	Time (s)	Feasible	Time (s)	Feasible	Time (s)	Feasible	Time (s)	Feasible
20	0.82	1000	1.2	1000	7.61	1000	9.98	1000
50	2.63	1000	2.84	1000	19.75	1000	24.33	1000
90	8.35	968	17.09	998	369.84	1000	528.74	1000
100	14.06	1000	12.37	802	625.72	1000	817.95	1000
110	65.08	1000	68.55	1000	1652.06	999	705.63	1000

5.2.3 Probability-free model

The probability-free model incorporates a deterministic representation of uncertainty by ensuring that the accumulated delays of scheduled tasks meet or exceed a specified fraction of the total allowable delay, controlled by the α parameter. This model was tested with various combinations of parameters: number of tasks (20 to 110), number of machines (2 or 3), maximum delay (5 or 10 minutes), α values (0.1, 0.3, 0.5, 0.7, 0.9), and two percentile settings for minimum processing times (25th and 50th percentiles).

Table 4 presents a selection of results from the probability-free model. The complete set of results is available in Appendix B.

Table 4: Selected results for the probability-free model.

Tasks	Machines	Max delay	α	Percentile	Time (s)	Feasible	Delay per machine
110	2	10	0.9	50	3.92	1000	990
110	3	10	0.9	50	3.80	1000	990
100	2	10	0.7	50	1.40	1000	700
100	3	5	0.9	25	0.40	598	450
90	2	5	0.1	25	0.07	785	45

Key findings from the analysis include:

Minimum total delay and delay per machine: The resilient constraint in our model introduces a minimum amount of delay that must be incorporated into the schedule. This minimum delay is a key feature of our approach to handling uncertainty. The mechanism operates as follows:

1. **Resilient constraint:** Our model requires that the total delay across all tasks and machines must be at least a certain fraction (α) of the maximum possible delay. This fraction is controlled by the resilience parameter α.
2. **Minimum total delay:** Based on the resilient constraint, we can calculate the minimum total delay as:
$$\text{Minimum total delay} = \alpha \cdot |\mathcal{J}| \cdot d_{max} \cdot |\mathcal{M}|, \tag{6}$$
where α is the resilience parameter, $|\mathcal{J}|$ is the number of tasks, d_{max} is the maximum allowable delay per task (set to either 5 or 10 minutes as described in the experimental setup), and $|\mathcal{M}|$ is the number of machines.
3. **Minimum delay per machine:** From this, we can derive the minimum delay per machine:
$$\text{Minimum delay per machine} = \frac{\text{Minimum total delay}}{|\mathcal{M}|} = \alpha \cdot |\mathcal{J}| \cdot d_{max}. \tag{7}$$
4. **Actual delay:** While the model must incorporate at least this minimum delay, it could potentially include more. The actual delay per machine is calculated as:
$$\text{Actual delay per machine} = \frac{\sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} d'_{jm}}{|\mathcal{M}|}, \tag{8}$$
where d'_{jm} represents the delay variable for job j on machine m.
5. **Solver behavior:** In practice, we observe that the solver typically sets the actual delay to the minimum required by the resilient constraint. This allows the model to schedule as many tasks as possible while still meeting the resilience requirement.

To illustrate, let us consider an example with 110 tasks, maximum delay of 10 minutes, and $\alpha = 0.1$:

- Minimum total delay: $0.1 \times 110 \times 10 \times |\mathcal{M}| = 110|\mathcal{M}|$ minutes.
- For 2 machines:
  - Minimum total delay: $110 \times 2 = 220$ minutes.
  - Minimum delay per machine: $220/2 = 110$ minutes.
- For 3 machines:
  - Minimum total delay: $110 \times 3 = 330$ minutes.
  - Minimum delay per machine: $330/3 = 110$ minutes.

Note that while the minimum total delay increases with the number of machines, the minimum delay per machine remains constant. This pattern holds across different parameter combinations, with the minimum total delay scaling proportionally with the number of tasks, maximum delay, α value, and number of machines.

Feasibility: For specific combinations of percentile, delay, and alpha values, the model achieves a 1000/1000 success rate, even for the largest instances. For example, with 110 tasks, 2 machines, max delay = 10, $\alpha = 0.9$, and 50th percentile, we achieve 1000/1000 feasible scenarios with a solve time of 3.92 seconds.

Computational efficiency: The model solves most instances in under 1 second, with even the most challenging scenarios (110 tasks) typically solved in less than 5 seconds. It is worth noting that despite its effectiveness in managing uncertainty, the `Resilient` constraint is fundamentally a linear constraint. This means it can be directly implemented and efficiently handled by standard CP solvers without requiring custom filtering algorithms. Linear constraints are generally well-supported by CP solvers, which have efficient propagation techniques for such constraints. This simplicity in implementation, combined with its power in addressing uncertainty, contributes to the model's computational efficiency and makes our approach both practical and accessible for real-world applications.

Scalability: As the problem size increases, the model continues to perform well. For instance, with 100 tasks, 2 machines, max delay = 10, $\alpha = 0.7$, and 50th percentile, we achieve 1000/1000 feasible scenarios in 1.4 seconds.

Parameter impact: Higher α values often lead to improved feasibility for larger instances. Increasing max delay from 5 to 10 minutes generally improves feasibility. Using the 50th percentile for minimum processing times often results in better feasibility compared to the 25th percentile, particularly for larger instances.

Machine configurations: The model performs well with both 2 and 3 machine configurations. In some cases, especially for larger instances and higher α values, using 3 machines can lead to improved feasibility.

While the model faces some challenges with very large instances under certain parameter combinations, these limitations can often be overcome by adjusting the parameters. The model's ability to achieve 1000/1000 feasibility for even the largest instances under appropriate settings is noteworthy.

Compared to the deterministic model, the probability-free model exhibits a significant improvement in handling uncertainty. While the deterministic model struggles with infeasible scenarios for instances with 90, 100, and 110 tasks, the probability-free model consistently generates proactive task sequencing, even for the largest instances considered. This highlights the effectiveness of the resilient approach in addressing the inherent variability in drilling durations.

In conclusion, the probability-free model meets its primary objective of creating proactive, practical schedules for a wide range of problem sizes. Its combination of computational efficiency, scalability, and adaptability through parameter tuning makes it a useful tool for drill rig scheduling under uncertainty.

5.2.4 Chance-constrained model

The chance-constrained resilient model aims to balance stochastic optimization benefits with resilient approach robustness while avoiding overly conservative solutions. It introduces a chance constraint on schedule resilience, ensuring a specified proportion of scenarios incorporate a minimum level of proactive delay allocation. The model employs two key parameters: α, determining the minimum proportion of maximum allowable delay to be proactively incorporated, and β, specifying the proportion of scenarios that must satisfy this resilience condition. Testing involved various parameter combinations: number of tasks (20 to 110), number of machines (2 or 3), maximum delay per task (5 or 10 minutes), α values (0.1, 0.3, 0.5, 0.7, 0.9), and β values (0.2, 0.4, 0.6, 0.8, 1.0). The results reveal several key insights:

Table 5 presents a selection of results from the chance-constrained model. The complete set of results is available in Appendix C.

Table 5: Selected results for the chance-constrained model.

Tasks	Machines	Delay	α	β	Solve time (s)	Considered scenarios	Delay per machine	Feasible
110	2	10	0.9	0.2	3811.36	10	990	1000
110	3	10	0.9	0.2	2460.59	10	990	1000
100	2	5	0.1	0.4	1556.30	20	50	981
90	3	5	0.3	0.6	1171.38	30	270	1000
80	2	5	0.1	0.2	26.35	10	40	999

Key findings from the analysis include:

Scenario selection: The model consistently selects the minimum number of scenarios required to satisfy the β constraint, as defined by:

$$\sum_{s\in\mathcal{S}} b_s \geq \beta|\mathcal{S}|, \tag{9}$$

where $\mathcal{S}$ is the set of all scenarios, b_s is a binary variable indicating whether scenario s is considered, and $|\mathcal{S}|$ is the total number of scenarios (50 in this case). The number of considered scenarios is always equal to $\beta|\mathcal{S}|$. For example, when $\beta = 0.2$, the number of considered scenarios is 10 (0.2 * 50).

This behavior is expected, as selecting more scenarios than necessary would hamper the solver's ability to schedule more tasks. Each additional scenario considered introduces more constraints that the solution must satisfy, potentially reducing the flexibility of the schedule and limiting the number of tasks that can be feasibly included. By selecting only the minimum required number of scenarios, the model maintains the desired level of resilience specified by β while maximizing its ability to schedule tasks efficiently.

Total delay per machine: We observe the same behavior as in the probability-free model. The solver consistently chooses to set this value to its minimum, as defined by the resilient constraint, to maximize the number of scheduled tasks.

Impact of α and β: Increasing α leads to higher total and average delays, as directly computed from the delay constraint. Increasing β results in a higher number of considered scenarios, directly proportional to the β value as per the scenario selection constraint, but does not affect the average delay per scenario.

Computational efficiency: Solving times vary significantly based on problem size and parameter settings. Smaller instances (up to 70 tasks) typically solve in under 30 seconds, while larger instances, especially with 90 or more tasks, can take from a few minutes to over an hour. For example, with $\alpha = 0.1$, $\beta = 0.2$, delay = 5, and 2 machines:

- 20 tasks: 7.5s.
- 80 tasks: 26.35s.
- 110 tasks: 1256.63s.

Feasibility: The model maintains high feasibility across most parameter combinations, often achieving 1000/1000 feasible scenarios even for the largest instances (110 tasks). This demonstrates the model's effectiveness in managing uncertainty while maintaining schedule feasibility.

Impact of delay setting: Increasing the maximum delay from 5 to 10 minutes results in proportionally higher average delays but does not significantly impact feasibility. It often leads to longer solving times for larger instances.

Number of machines: The effect of increasing from 2 to 3 machines varies. For smaller instances, it often leads to slightly longer solving times. For larger instances, especially with higher α values, 3 machines can sometimes lead to shorter solving times and improved feasibility.

In conclusion, the chance-constrained model demonstrates the ability to create robust schedules across a wide range of problem sizes and uncertainty levels. It offers a flexible approach to balancing schedule robustness and computational efficiency through the α and β parameters. The model's behavior in selecting the minimum number of scenarios and implementing the minimum required delay showcases its optimization capability, while its high feasibility rates, even for large instances, suggest its potential for practical application in complex drill rig scheduling scenarios under uncertainty.

5.2.5 Comparative analysis

To provide a comprehensive comparison of the four models developed in this study, we analyze their performance across various metrics: computational efficiency, solution quality, robustness to uncertainty, and applicability based on data quality. Table 6 presents a summary of key results for each model.

Table 6: Comparative summary of model performance.

Metric	Model 0 (Deterministic)	Model 1 (Two-stage)	Model 2 (Resilient)	Model 3 (Chance-constrained)
Avg. Solve Time (110 tasks, 2 machines)	¡ 1s	1652.06s	3.61s	1256.63s
Feasibility (110 tasks, 2 machines)	440/1000	999/1000	1000/1000	1000/1000
Scalability (Solve time increase 20 to 110 tasks)	Minimal	Significant	Moderate	Significant
Robustness to Uncertainty	Low	High	High	High
Flexibility in Uncertainty Management	None	Limited	Moderate	High
Data Dependency	High	High	Low	Moderate

Computational efficiency: Model 0 (deterministic) consistently demonstrates the fastest solving times, typically under 1 second even for large instances. Model 2 (resilient) also shows impressive efficiency, with solving times generally under 5 seconds for most instances. Models 1 (two-stage) and 3 (chance-constrained) exhibit longer solving times, especially for larger instances, due to their more complex handling of uncertainty.

Solution quality and feasibility: All models can achieve optimal solutions (scheduling all tasks) for smaller instances. However, as the problem size increases, differences emerge:

- Model 0 struggles with feasibility for larger instances, dropping to 440/1000 feasible scenarios for 110 tasks.
- Model 1 maintains high feasibility (999/1000 for 110 tasks) with 100 scenarios, showing improved robustness over Model 0.
- Models 2 and 3 consistently achieve perfect or near-perfect feasibility (1000/1000) even for the largest instances, demonstrating superior robustness to uncertainty.

Notably, these models allow for scheduling 90 to 110 tasks consistently, a level of productivity that was very rarely achieved in the historical data, as evident from the histogram in Figure 2. This capability represents a significant potential for operational improvement.

Scalability: Model 0 and Model 2 show the best scalability, with solving times increasing only moderately as the number of tasks grows. Models 1 and 3 exhibit more significant increases in solving time for larger instances, reflecting the increased complexity of their uncertainty handling mechanisms. Despite this, all models can handle the full range of task numbers observed in the historical data (see Figure 2), and even beyond, up to 110 tasks.

Robustness to uncertainty: Models 1, 2, and 3 all show significant improvements in robustness compared to Model 0:

- Model 1 achieves this through scenario-based optimization.
- Model 2 incorporates a minimum level of delay directly into the schedule.

- Model 3 combines scenario-based optimization with chance constraints on schedule resilience.

This enhanced robustness is particularly valuable for handling the variability in daily drilling operations observed in the historical data (Figure 2).

Flexibility in uncertainty management: Model 3 offers the highest flexibility, allowing fine-tuning of both the proportion of scenarios considered (β) and the level of resilience (α). Model 2 provides moderate flexibility through the α parameter, while Model 1's flexibility is limited to adjusting the number of scenarios. Model 0 does not provide any mechanism for uncertainty management. This flexibility is crucial for adapting to the wide range of daily drilling rates observed in the historical data.

Applicability based on data quality: The choice of model also depends significantly on the quality and availability of historical data:

- Models 0 and 1 are most suitable when historical data (like that shown in Figure 2) is reliable, complete, and can be reasonably assumed to be representative of future conditions. Model 1, in particular, leverages this data to consider multiple possible realizations of uncertainty, similar to how machine learning models are trained on historical data to make future predictions.
- Model 2 is particularly valuable when historical data is unavailable, unreliable, or not representative of future conditions. This might occur in new mining locations, unexplored geological formations, or when using new equipment. By incorporating a minimum level of delay directly into the schedule, it provides robustness without relying on historical data.
- Model 3 offers a hybrid approach, suitable for situations where data is partially available or trustworthy. It combines the scenario-based approach of Model 1 with the resilience mechanism of Model 2, allowing for a more nuanced treatment of uncertainty when data quality is mixed.

In conclusion, each model presents distinct trade-offs and is suited to different scenarios:

- Model 0 offers unparalleled speed but at the cost of robustness for larger instances and requires reliable historical data.
- Model 1 provides a balanced approach, improving robustness significantly over Model 0 while maintaining reasonable solve times, and is ideal when historical data is representative of future conditions.
- Model 2 combines excellent computational efficiency with high robustness, making it particularly suitable for time-sensitive applications or when historical data is unreliable.
- Model 3 offers the most comprehensive approach to uncertainty management, at the cost of increased computational complexity, and is well-suited for situations with mixed data quality.

The choice between these models would depend on the specific requirements of the drilling operation, balancing the need for computational speed, solution robustness, flexibility in uncertainty management, and the quality and availability of historical data. This approach to decision-making under uncertainty shares similarities with machine learning methodologies, where the quality and representativeness of historical data play a crucial role in model selection and performance.

Importantly, all models demonstrate the capability to consistently schedule a number of tasks that falls in the upper range or even exceeds the typical daily drilling rates observed in the historical data (Figure 2). This suggests that these models have the potential to significantly improve operational efficiency in drilling operations, potentially pushing productivity beyond historically observed levels while maintaining robustness to uncertainty.

6 Conclusion

This paper has addressed the challenge of scheduling electrical drill rigs in open-pit mines under uncertain drilling durations. We proposed three non-deterministic CP models incorporating different optimization paradigms: a two-stage stochastic approach, a probability-free (resilient) approach, and a chance-constrained model.

Our computational experiments, based on simulated instances derived from real-world drilling data, demonstrated the effectiveness of these models in generating robust and reliable schedules. The comparative analysis highlighted trade-offs between computational efficiency, solution quality, and robustness:

- The deterministic model, while computationally efficient, struggled with feasibility under drilling duration variations, emphasizing the need for robust approaches.
- The two-stage stochastic model showed excellent robustness but at the cost of increased computational effort, especially for larger instances.
- The probability-free model emerged as a promising approach, balancing computational efficiency and robustness without relying on probabilistic assumptions.
- The chance-constrained model offered the most comprehensive approach to uncertainty management, providing unparalleled flexibility in balancing schedule robustness and operational efficiency.

For mining practitioners, our study underscores the importance of explicitly considering uncertainty in drill rig scheduling. The proposed models, particularly the probability-free and chance-constrained approaches, can serve as valuable decision support tools, enhancing the efficiency and effectiveness of drilling operations.

From a research perspective, our work highlights the value of constraint programming in modeling and solving complex scheduling problems under uncertainty. Future research could explore integrating these models with other aspects of the mining supply chain and investigating their performance under different types of uncertainty.

In conclusion, this paper has made significant contributions to drill rig scheduling in open-pit mines under uncertainty. The proposed non-deterministic CP models provide mining practitioners with a suite of tools to effectively manage uncertain drilling durations, potentially driving more efficient and robust drilling operations and contributing to the overall success and sustainability of the mining industry.

Appendix A

Table A1: Results for the Two-Stage Stochastic Model.

Tasks	Machines	Scenarios	Objective function	Solve Time (s)	Feasible
20	2	10	20	0.82	1000
30	2	10	30	0.87	1000
40	2	10	40	0.94	1000
50	2	10	50	2.63	1000
60	2	10	60	2.57	1000
70	2	10	70	3.71	1000
80	2	10	80	3.76	997
90	2	10	90	8.35	968
100	2	10	100	14.06	1000
110	2	10	110	65.08	1000
20	3	10	20	1.2	1000
30	3	10	30	0.31	1000

Continued on next page

Table A1 – continued

Tasks	Machines	Scenarios	Objective	Solve	Feasible
40	3	10	40	1.13	1000
50	3	10	50	2.84	1000
60	3	10	60	3.09	1000
70	3	10	70	3.67	1000
80	3	10	80	4.49	996
90	3	10	90	17.09	998
100	3	10	100	12.37	802
110	3	10	110	68.55	1000
20	2	20	20	1.41	1000
30	2	20	30	3.18	1000
40	2	20	40	1.67	1000
50	2	20	50	3.96	1000
60	2	20	60	2.5	1000
70	2	20	70	2.99	1000
80	2	20	80	6.15	997
90	2	20	90	37.67	968
100	2	20	100	276.26	998
110	2	20	110	243.62	1000
20	3	20	20	2.65	1000
30	3	20	30	2.89	1000
40	3	20	40	2.17	1000
50	3	20	50	6.21	1000
60	3	20	60	6.3	1000
70	3	20	70	4.05	1000
80	3	20	80	7.34	996
90	3	20	90	38.39	985
100	3	20	100	68.09	972
110	3	20	110	599.47	1000
20	2	30	20	2.6	1000
30	2	30	30	4.42	1000
40	2	30	40	4.18	1000
50	2	30	50	3.35	1000
60	2	30	60	6.25	1000
70	2	30	70	7.38	1000
80	2	30	80	8.42	997
90	2	30	90	73.54	975
100	2	30	100	997.66	1000
110	2	30	110	1323.73	1000
20	3	30	20	3.59	1000
30	3	30	30	2.39	1000
40	3	30	40	5.25	1000
50	3	30	50	6.73	1000
60	3	30	60	8.54	1000
70	3	30	70	10	1000
80	3	30	80	11.74	996
90	3	30	90	85.88	999
100	3	30	100	100.31	970
110	3	30	110	1443.02	1000
20	2	40	20	3.92	1000
30	2	40	30	5.81	1000
40	2	40	40	6.45	1000
50	2	40	50	6.84	1000
60	2	40	60	9.67	1000
70	2	40	70	11.44	1000
80	2	40	80	7.7	997
90	2	40	90	95.5	986
100	2	40	100	1364.18	1000
110	2	40	110	3296.25	1000
20	3	40	20	4.03	1000
30	3	40	30	3.43	1000
40	3	40	40	8.92	1000

Continued on next page

Table A1 – continued

Tasks	Machines	Scenarios	Objective	Solve	Feasible
50	3	40	50	5.88	1000
60	3	40	60	13.03	1000
70	3	40	70	13.57	1000
80	3	40	80	16.52	996
90	3	40	90	192.33	990
100	3	40	100	102.99	1000
110	3	40	110	246.9	1000
20	2	50	20	3.7	1000
30	2	50	30	5.3	1000
40	2	50	40	4.46	1000
50	2	50	50	9.7	1000
60	2	50	60	14.1	1000
70	2	50	70	8.67	1000
80	2	50	80	15.23	997
90	2	50	90	325.06	981
100	2	50	100	202.69	1000
110	2	50	110	571.41	1000
20	3	50	20	5.3	1000
30	3	50	30	7.23	1000
40	3	50	40	5.93	1000
50	3	50	50	11.83	1000
60	3	50	60	9.42	1000
70	3	50	70	18.76	1000
80	3	50	80	6.8	996
90	3	50	90	116.9	998
100	3	50	100	191.25	1000
110	3	50	110	279.87	1000
20	2	60	20	4.64	1000
30	2	60	30	8.34	1000
40	2	60	40	5.61	1000
50	2	60	50	7.23	1000
60	2	60	60	8.74	1000
70	2	60	70	16.7	1000
80	2	60	80	20.9	997
90	2	60	90	129.51	1000
100	2	60	100	321.71	1000
110	2	60	110	340.54	1000
20	3	60	20	8.55	1000
30	3	60	30	8.8	1000
40	3	60	40	11.84	1000
50	3	60	50	15.67	1000
60	3	60	60	18.67	1000
70	3	60	70	7.15	1000
80	3	60	80	13.54	996
90	3	60	90	170.71	1000
100	3	60	100	170.33	999
110	3	60	110	237.84	1000
20	2	70	20	6.49	1000
30	2	70	30	4.82	1000
40	2	70	40	10.13	1000
50	2	70	50	14.97	1000
60	2	70	60	16.42	1000
70	2	70	70	21.1	1000
80	2	70	80	14.07	997
90	2	70	90	131.96	1000
100	2	70	100	341.71	1000
110	2	70	110	765.69	1000
20	3	70	20	7.16	1000
30	3	70	30	10.4	1000
40	3	70	40	8.89	1000
50	3	70	50	18.66	1000

Continued on next page

Table A1 – continued

Tasks	Machines	Scenarios	Objective	Solve	Feasible
60	3	70	60	7.32	1000
70	3	70	70	16.47	1000
80	3	70	80	27.74	996
90	3	70	90	319.65	1000
100	3	70	100	313.25	1000
110	3	70	110	299.76	1000
20	2	80	20	7.8	1000
30	2	80	30	8.99	1000
40	2	80	40	12.03	1000
50	2	80	50	10.29	1000
60	2	80	60	20.62	1000
70	2	80	70	14.42	1000
80	2	80	80	8.11	997
90	2	80	90	217.23	1000
100	2	80	100	402.3	1000
110	2	80	110	1479.37	1000
20	3	80	20	5.05	1000
30	3	80	30	7.53	1000
40	3	80	40	17.95	1000
50	3	80	50	13.75	1000
60	3	80	60	14.31	1000
70	3	80	70	9.14	1000
80	3	80	80	25.25	996
90	3	80	90	349.88	1000
100	3	80	100	295.9	1000
110	3	80	110	416.73	1000
20	2	90	20	7.64	1000
30	2	90	30	10.05	1000
40	2	90	40	13.83	1000
50	2	90	50	19.67	1000
60	2	90	60	7.09	1000
70	2	90	70	8.84	1000
80	2	90	80	30.64	997
90	2	90	90	440.81	1000
100	2	90	100	589.03	1000
110	2	90	110	1812.63	1000
20	3	90	20	8.88	1000
30	3	90	30	9.12	1000
40	3	90	40	12.37	1000
50	3	90	50	8	1000
60	3	90	60	26.79	1000
70	3	90	70	24.92	1000
80	3	90	80	30	996
90	3	90	90	400.49	1000
100	3	90	100	176.39	1000
110	3	90	110	903.22	1000
20	2	100	20	7.61	1000
30	2	100	30	11.48	1000
40	2	100	40	16.03	1000
50	2	100	50	19.75	1000
60	2	100	60	14.97	1000
70	2	100	70	19.86	1000
80	2	100	80	28.86	997
90	2	100	90	369.84	1000
100	2	100	100	625.72	1000
110	2	100	110	1652.06	999
20	3	100	20	9.98	1000
30	3	100	30	14.72	1000
40	3	100	40	14.19	1000
50	3	100	50	24.33	1000
60	3	100	60	25.45	1000

Continued on next page

Table A1 – continued

Tasks	Machines	Scenarios	Objective	Solve	Feasible
70	3	100	70	23.49	1000
80	3	100	80	16.7	996
90	3	100	90	528.74	1000
100	3	100	100	817.95	1000
110	3	100	110	705.63	1000

Appendix B

Table B1: Results for the Probability-Free Model.

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
20	2	5	0.1	25	20	0.16	10	1000
30	2	5	0.1	25	30	0.03	15	1000
40	2	5	0.1	25	40	0.04	20	1000
50	2	5	0.1	25	50	0.04	25	1000
60	2	5	0.1	25	60	0.05	30	1000
70	2	5	0.1	25	70	0.05	35	1000
80	2	5	0.1	25	80	0.06	40	997
90	2	5	0.1	25	90	0.07	45	785
100	2	5	0.1	25	100	0.07	50	116
110	2	5	0.1	25	110	0.69	55	2
20	3	5	0.1	25	20	0.24	10	1000
30	3	5	0.1	25	30	0.04	15	1000
40	3	5	0.1	25	40	0.04	20	1000
50	3	5	0.1	25	50	0.05	25	1000
60	3	5	0.1	25	60	0.06	30	1000
70	3	5	0.1	25	70	0.06	35	1000
80	3	5	0.1	25	80	0.08	40	996
90	3	5	0.1	25	90	0.09	45	773
100	3	5	0.1	25	100	0.09	50	124
110	3	5	0.1	25	110	0.55	55	2
20	2	5	0.3	25	20	0.17	30	1000
30	2	5	0.3	25	30	0.03	45	1000
40	2	5	0.3	25	40	0.04	60	1000
50	2	5	0.3	25	50	0.05	75	1000
60	2	5	0.3	25	60	0.05	90	1000
70	2	5	0.3	25	70	0.05	105	1000
80	2	5	0.3	25	80	0.06	120	997
90	2	5	0.3	25	90	0.06	135	785
100	2	5	0.3	25	100	0.07	150	116
110	2	5	0.3	25	110	0.67	165	2
20	3	5	0.3	25	20	0.25	30	1000
30	3	5	0.3	25	30	0.04	45	1000
40	3	5	0.3	25	40	0.04	60	1000
50	3	5	0.3	25	50	0.05	75	1000
60	3	5	0.3	25	60	0.06	90	1000
70	3	5	0.3	25	70	0.06	105	1000
80	3	5	0.3	25	80	0.07	120	996
90	3	5	0.3	25	90	0.08	135	773
100	3	5	0.3	25	100	0.09	150	124
110	3	5	0.3	25	110	0.96	165	2
20	2	5	0.5	25	20	0.17	50	1000
30	2	5	0.5	25	30	0.04	75	1000
40	2	5	0.5	25	40	0.04	100	1000
50	2	5	0.5	25	50	0.04	125	1000
60	2	5	0.5	25	60	0.05	150	1000
70	2	5	0.5	25	70	0.05	175	1000
80	2	5	0.5	25	80	0.06	200	997

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	5	0.5	25	90	0.07	225	785
100	2	5	0.5	25	100	0.07	250	116
110	2	5	0.5	25	110	1.73	275	31
20	3	5	0.5	25	20	0.22	50	1000
30	3	5	0.5	25	30	0.04	75	1000
40	3	5	0.5	25	40	0.04	100	1000
50	3	5	0.5	25	50	0.05	125	1000
60	3	5	0.5	25	60	0.05	150	1000
70	3	5	0.5	25	70	0.07	175	1000
80	3	5	0.5	25	80	0.07	200	996
90	3	5	0.5	25	90	0.09	225	773
100	3	5	0.5	25	100	0.09	250	124
110	3	5	0.5	25	110	1.91	275	103
20	2	5	0.7	25	20	0.17	70	1000
30	2	5	0.7	25	30	0.03	105	1000
40	2	5	0.7	25	40	0.04	140	1000
50	2	5	0.7	25	50	0.04	175	1000
60	2	5	0.7	25	60	0.05	210	1000
70	2	5	0.7	25	70	0.06	245	1000
80	2	5	0.7	25	80	0.06	280	997
90	2	5	0.7	25	90	0.07	315	785
100	2	5	0.7	25	100	0.32	350	467
110	2	5	0.7	25	110	2.17	385	35
20	3	5	0.7	25	20	0.24	70	1000
30	3	5	0.7	25	30	0.04	105	1000
40	3	5	0.7	25	40	0.04	140	1000
50	3	5	0.7	25	50	0.05	175	1000
60	3	5	0.7	25	60	0.06	210	1000
70	3	5	0.7	25	70	0.07	245	1000
80	3	5	0.7	25	80	0.07	280	996
90	3	5	0.7	25	90	0.08	315	773
100	3	5	0.7	25	100	0.39	350	758
110	3	5	0.7	25	110	2.53	385	90
20	2	5	0.9	25	20	0.15	90	1000
30	2	5	0.9	25	30	0.03	135	1000
40	2	5	0.9	25	40	0.04	180	1000
50	2	5	0.9	25	50	0.05	225	1000
60	2	5	0.9	25	60	0.05	270	1000
70	2	5	0.9	25	70	0.06	315	1000
80	2	5	0.9	25	80	0.06	360	997
90	2	5	0.9	25	90	0.07	405	785
100	2	5	0.9	25	100	0.33	450	498
110	2	5	0.9	25	110	3.77	495	1000
20	3	5	0.9	25	20	0.23	90	1000
30	3	5	0.9	25	30	0.04	135	1000
40	3	5	0.9	25	40	0.04	180	1000
50	3	5	0.9	25	50	0.05	225	1000
60	3	5	0.9	25	60	0.06	270	1000
70	3	5	0.9	25	70	0.06	315	1000
80	3	5	0.9	25	80	0.07	360	996
90	3	5	0.9	25	90	0.08	405	773
100	3	5	0.9	25	100	0.4	450	598
110	3	5	0.9	25	110	2.31	495	171
20	2	10	0.1	25	20	0.18	20	1000
30	2	10	0.1	25	30	0.04	30	1000
40	2	10	0.1	25	40	0.04	40	1000
50	2	10	0.1	25	50	0.04	50	1000
60	2	10	0.1	25	60	0.05	60	1000
70	2	10	0.1	25	70	0.06	70	1000
80	2	10	0.1	25	80	0.06	80	997

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	10	0.1	25	90	0.07	90	785
100	2	10	0.1	25	100	0.07	100	116
110	2	10	0.1	25	110	0.08	110	2
20	3	10	0.1	25	20	0.27	20	1000
30	3	10	0.1	25	30	0.04	30	1000
40	3	10	0.1	25	40	0.05	40	1000
50	3	10	0.1	25	50	0.05	50	1000
60	3	10	0.1	25	60	0.06	60	1000
70	3	10	0.1	25	70	0.07	70	1000
80	3	10	0.1	25	80	0.07	80	996
90	3	10	0.1	25	90	0.08	90	773
100	3	10	0.1	25	100	0.09	100	124
110	3	10	0.1	25	110	0.81	110	2
20	2	10	0.3	25	20	0.22	60	1000
30	2	10	0.3	25	30	0.03	90	1000
40	2	10	0.3	25	40	0.04	120	1000
50	2	10	0.3	25	50	0.04	150	1000
60	2	10	0.3	25	60	0.05	180	1000
70	2	10	0.3	25	70	0.06	210	1000
80	2	10	0.3	25	80	0.06	240	997
90	2	10	0.3	25	90	0.07	270	785
100	2	10	0.3	25	100	0.07	300	116
110	2	10	0.3	25	110	0.38	330	42
20	3	10	0.3	25	20	0.24	60	1000
30	3	10	0.3	25	30	0.04	90	1000
40	3	10	0.3	25	40	0.05	120	1000
50	3	10	0.3	25	50	0.05	150	1000
60	3	10	0.3	25	60	0.06	180	1000
70	3	10	0.3	25	70	0.06	210	1000
80	3	10	0.3	25	80	0.07	240	996
90	3	10	0.3	25	90	0.1	270	773
100	3	10	0.3	25	100	0.09	300	124
110	3	10	0.3	25	110	2.42	330	51
20	2	10	0.5	25	20	0.18	100	1000
30	2	10	0.5	25	30	0.03	150	1000
40	2	10	0.5	25	40	0.04	200	1000
50	2	10	0.5	25	50	0.05	250	1000
60	2	10	0.5	25	60	0.05	300	1000
70	2	10	0.5	25	70	0.06	350	1000
80	2	10	0.5	25	80	0.06	400	997
90	2	10	0.5	25	90	0.07	450	785
100	2	10	0.5	25	100	0.34	500	560
110	2	10	0.5	25	110	0.38	550	47
20	3	10	0.5	25	20	0.21	100	1000
30	3	10	0.5	25	30	0.04	150	1000
40	3	10	0.5	25	40	0.04	200	1000
50	3	10	0.5	25	50	0.05	250	1000
60	3	10	0.5	25	60	0.06	300	1000
70	3	10	0.5	25	70	0.06	350	1000
80	3	10	0.5	25	80	0.08	400	996
90	3	10	0.5	25	90	0.08	450	773
100	3	10	0.5	25	100	0.4	500	779
110	3	10	0.5	25	110	2.52	550	103
20	2	10	0.7	25	20	0.19	140	1000
30	2	10	0.7	25	30	0.04	210	1000
40	2	10	0.7	25	40	0.04	280	1000
50	2	10	0.7	25	50	0.05	350	1000
60	2	10	0.7	25	60	0.05	420	1000
70	2	10	0.7	25	70	0.06	490	1000
80	2	10	0.7	25	80	0.06	560	997

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	10	0.7	25	90	0.34	630	934
100	2	10	0.7	25	100	0.34	700	467
110	2	10	0.7	25	110	1.6	770	1000
20	3	10	0.7	25	20	0.22	140	1000
30	3	10	0.7	25	30	0.04	210	1000
40	3	10	0.7	25	40	0.05	280	1000
50	3	10	0.7	25	50	0.05	350	1000
60	3	10	0.7	25	60	0.06	420	1000
70	3	10	0.7	25	70	0.07	490	1000
80	3	10	0.7	25	80	0.08	560	996
90	3	10	0.7	25	90	0.35	630	994
100	3	10	0.7	25	100	0.41	700	758
110	3	10	0.7	25	110	4.08	770	110
20	2	10	0.9	25	20	0.17	180	1000
30	2	10	0.9	25	30	0.04	270	1000
40	2	10	0.9	25	40	0.04	360	1000
50	2	10	0.9	25	50	0.05	450	1000
60	2	10	0.9	25	60	0.05	540	1000
70	2	10	0.9	25	70	0.06	630	1000
80	2	10	0.9	25	80	0.42	720	1000
90	2	10	0.9	25	90	1.14	810	1000
100	2	10	0.9	25	100	1.36	900	1000
110	2	10	0.9	25	110	1.95	990	1000
20	3	10	0.9	25	20	0.25	180	1000
30	3	10	0.9	25	30	0.04	270	1000
40	3	10	0.9	25	40	0.04	360	1000
50	3	10	0.9	25	50	0.07	450	1000
60	3	10	0.9	25	60	0.06	540	1000
70	3	10	0.9	25	70	0.06	630	1000
80	3	10	0.9	25	80	0.35	720	1000
90	3	10	0.9	25	90	0.56	810	997
100	3	10	0.9	25	100	1.49	900	998
110	3	10	0.9	25	110	4.24	990	1000
20	2	5	0.1	50	20	0.17	10	1000
30	2	5	0.1	50	30	0.03	15	1000
40	2	5	0.1	50	40	0.04	20	1000
50	2	5	0.1	50	50	0.04	25	1000
60	2	5	0.1	50	60	0.05	30	1000
70	2	5	0.1	50	70	0.05	35	1000
80	2	5	0.1	50	80	0.06	40	997
90	2	5	0.1	50	90	0.07	45	785
100	2	5	0.1	50	100	0.32	50	387
110	2	5	0.1	50	110	4.05	55	287
20	3	5	0.1	50	20	0.25	10	1000
30	3	5	0.1	50	30	0.03	15	1000
40	3	5	0.1	50	40	0.04	20	1000
50	3	5	0.1	50	50	0.05	25	1000
60	3	5	0.1	50	60	0.06	30	1000
70	3	5	0.1	50	70	0.07	35	1000
80	3	5	0.1	50	80	0.07	40	996
90	3	5	0.1	50	90	0.08	45	773
100	3	5	0.1	50	100	0.4	50	821
110	3	5	0.1	50	110	3.97	55	508
20	2	5	0.3	50	20	0.17	30	1000
30	2	5	0.3	50	30	0.03	45	1000
40	2	5	0.3	50	40	0.04	60	1000
50	2	5	0.3	50	50	0.04	75	1000
60	2	5	0.3	50	60	0.05	90	1000
70	2	5	0.3	50	70	0.05	105	1000
80	2	5	0.3	50	80	0.06	120	997

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	5	0.3	50	90	0.34	135	946
100	2	5	0.3	50	100	0.33	150	590
110	2	5	0.3	50	110	4.04	165	988
20	3	5	0.3	50	20	0.21	30	1000
30	3	5	0.3	50	30	0.04	45	1000
40	3	5	0.3	50	40	0.05	60	1000
50	3	5	0.3	50	50	0.05	75	1000
60	3	5	0.3	50	60	0.06	90	1000
70	3	5	0.3	50	70	0.07	105	1000
80	3	5	0.3	50	80	0.07	120	996
90	3	5	0.3	50	90	0.37	135	990
100	3	5	0.3	50	100	0.39	150	439
110	3	5	0.3	50	110	3.71	165	292
20	2	5	0.5	50	20	0.17	50	1000
30	2	5	0.5	50	30	0.03	75	1000
40	2	5	0.5	50	40	0.04	100	1000
50	2	5	0.5	50	50	0.04	125	1000
60	2	5	0.5	50	60	0.05	150	1000
70	2	5	0.5	50	70	0.05	175	1000
80	2	5	0.5	50	80	0.06	200	997
90	2	5	0.5	50	90	0.34	225	966
100	2	5	0.5	50	100	0.35	250	616
110	2	5	0.5	50	110	3.33	275	985
20	3	5	0.5	50	20	0.03	50	1000
30	3	5	0.5	50	30	0.04	75	1000
40	3	5	0.5	50	40	0.05	100	1000
50	3	5	0.5	50	50	0.06	125	1000
60	3	5	0.5	50	60	0.06	150	1000
70	3	5	0.5	50	70	0.06	175	1000
80	3	5	0.5	50	80	0.08	200	996
90	3	5	0.5	50	90	0.38	225	991
100	3	5	0.5	50	100	0.4	250	659
110	3	5	0.5	50	110	3.53	275	218
20	2	5	0.7	50	20	0.18	70	1000
30	2	5	0.7	50	30	0.03	105	1000
40	2	5	0.7	50	40	0.04	140	1000
50	2	5	0.7	50	50	0.05	175	1000
60	2	5	0.7	50	60	0.05	210	1000
70	2	5	0.7	50	70	0.05	245	1000
80	2	5	0.7	50	80	0.32	280	1000
90	2	5	0.7	50	90	0.45	315	986
100	2	5	0.7	50	100	1.5	350	1000
110	2	5	0.7	50	110	3.8	385	1000
20	3	5	0.7	50	20	0.24	70	1000
30	3	5	0.7	50	30	0.04	105	1000
40	3	5	0.7	50	40	0.04	140	1000
50	3	5	0.7	50	50	0.05	175	1000
60	3	5	0.7	50	60	0.06	210	1000
70	3	5	0.7	50	70	0.06	245	1000
80	3	5	0.7	50	80	0.36	280	1000
90	3	5	0.7	50	90	0.37	315	991
100	3	5	0.7	50	100	0.46	350	803
110	3	5	0.7	50	110	5	385	1000
20	2	5	0.9	50	20	0.19	90	1000
30	2	5	0.9	50	30	0.03	135	1000
40	2	5	0.9	50	40	0.04	180	1000
50	2	5	0.9	50	50	0.04	225	1000
60	2	5	0.9	50	60	0.05	270	1000
70	2	5	0.9	50	70	0.05	315	1000
80	2	5	0.9	50	80	0.39	360	1000

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	5	0.9	50	90	1.04	405	1000
100	2	5	0.9	50	100	1.51	450	1000
110	2	5	0.9	50	110	3.61	495	1000
20	3	5	0.9	50	20	0.21	90	1000
30	3	5	0.9	50	30	0.04	135	1000
40	3	5	0.9	50	40	0.05	180	1000
50	3	5	0.9	50	50	0.05	225	1000
60	3	5	0.9	50	60	0.06	270	1000
70	3	5	0.9	50	70	0.07	315	1000
80	3	5	0.9	50	80	0.35	360	1000
90	3	5	0.9	50	90	0.59	405	1000
100	3	5	0.9	50	100	1.4	450	1000
110	3	5	0.9	50	110	4.17	495	1000
20	2	10	0.1	50	20	0.19	20	1000
30	2	10	0.1	50	30	0.03	30	1000
40	2	10	0.1	50	40	0.04	40	1000
50	2	10	0.1	50	50	0.04	50	1000
60	2	10	0.1	50	60	0.05	60	1000
70	2	10	0.1	50	70	0.06	70	1000
80	2	10	0.1	50	80	0.06	80	997
90	2	10	0.1	50	90	0.07	90	785
100	2	10	0.1	50	100	0.32	100	387
110	2	10	0.1	50	110	1.62	110	267
20	3	10	0.1	50	20	0.22	20	1000
30	3	10	0.1	50	30	0.04	30	1000
40	3	10	0.1	50	40	0.04	40	1000
50	3	10	0.1	50	50	0.05	50	1000
60	3	10	0.1	50	60	0.06	60	1000
70	3	10	0.1	50	70	0.07	70	1000
80	3	10	0.1	50	80	0.07	80	996
90	3	10	0.1	50	90	0.08	90	773
100	3	10	0.1	50	100	0.4	100	821
110	3	10	0.1	50	110	4.12	110	462
20	2	10	0.3	50	20	0.18	60	1000
30	2	10	0.3	50	30	0.04	90	1000
40	2	10	0.3	50	40	0.04	120	1000
50	2	10	0.3	50	50	0.04	150	1000
60	2	10	0.3	50	60	0.05	180	1000
70	2	10	0.3	50	70	0.05	210	1000
80	2	10	0.3	50	80	0.06	240	997
90	2	10	0.3	50	90	0.35	270	942
100	2	10	0.3	50	100	0.33	300	590
110	2	10	0.3	50	110	1.35	330	630
20	3	10	0.3	50	20	0.24	60	1000
30	3	10	0.3	50	30	0.04	90	1000
40	3	10	0.3	50	40	0.05	120	1000
50	3	10	0.3	50	50	0.05	150	1000
60	3	10	0.3	50	60	0.05	180	1000
70	3	10	0.3	50	70	0.07	210	1000
80	3	10	0.3	50	80	0.07	240	996
90	3	10	0.3	50	90	0.37	270	990
100	3	10	0.3	50	100	0.4	300	439
110	3	10	0.3	50	110	4.38	330	458
20	2	10	0.5	50	20	0.21	100	1000
30	2	10	0.5	50	30	0.04	150	1000
40	2	10	0.5	50	40	0.04	200	1000
50	2	10	0.5	50	50	0.04	250	1000
60	2	10	0.5	50	60	0.05	300	1000
70	2	10	0.5	50	70	0.06	350	1000
80	2	10	0.5	50	80	0.3	400	1000

Continued on next page

Table B1 – continued

Tasks	Machines	Max delay	α	Percentile	Objective function	Solve Time (s)	Total machine delay	Feasible
90	2	10	0.5	50	90	0.34	450	979
100	2	10	0.5	50	100	0.37	500	616
110	2	10	0.5	50	110	1.33	550	1000
20	3	10	0.5	50	20	0.21	100	1000
30	3	10	0.5	50	30	0.04	150	1000
40	3	10	0.5	50	40	0.04	200	1000
50	3	10	0.5	50	50	0.05	250	1000
60	3	10	0.5	50	60	0.06	300	1000
70	3	10	0.5	50	70	0.07	350	1000
80	3	10	0.5	50	80	0.36	400	1000
90	3	10	0.5	50	90	0.37	450	991
100	3	10	0.5	50	100	0.4	500	659
110	3	10	0.5	50	110	3.77	550	517
20	2	10	0.7	50	20	0.2	140	1000
30	2	10	0.7	50	30	0.03	210	1000
40	2	10	0.7	50	40	0.04	280	1000
50	2	10	0.7	50	50	0.05	350	1000
60	2	10	0.7	50	60	0.05	420	1000
70	2	10	0.7	50	70	0.26	490	1000
80	2	10	0.7	50	80	0.38	560	1000
90	2	10	0.7	50	90	1.07	630	1000
100	2	10	0.7	50	100	1.4	700	1000
110	2	10	0.7	50	110	1.9	770	1000
20	3	10	0.7	50	20	0.21	140	1000
30	3	10	0.7	50	30	0.04	210	1000
40	3	10	0.7	50	40	0.04	280	1000
50	3	10	0.7	50	50	0.05	350	1000
60	3	10	0.7	50	60	0.06	420	1000
70	3	10	0.7	50	70	0.33	490	1000
80	3	10	0.7	50	80	0.36	560	1000
90	3	10	0.7	50	90	0.45	630	999
100	3	10	0.7	50	100	0.87	700	843
110	3	10	0.7	50	110	3.79	770	754
20	2	10	0.9	50	20	0.19	180	1000
30	2	10	0.9	50	30	0.03	270	1000
40	2	10	0.9	50	40	0.04	360	1000
50	2	10	0.9	50	50	0.05	450	1000
60	2	10	0.9	50	60	0.1	540	1000
70	2	10	0.9	50	70	0.5	630	1000
80	2	10	0.9	50	80	1	720	1000
90	2	10	0.9	50	90	1.36	810	1000
100	2	10	0.9	50	100	1.61	900	1000
110	2	10	0.9	50	110	3.92	990	1000
20	3	10	0.9	50	20	0.25	180	1000
30	3	10	0.9	50	30	0.04	270	1000
40	3	10	0.9	50	40	0.04	360	1000
50	3	10	0.9	50	50	0.05	450	1000
60	3	10	0.9	50	60	0.27	540	1000
70	3	10	0.9	50	70	0.37	630	1000
80	3	10	0.9	50	80	0.7	720	1000
90	3	10	0.9	50	90	1.28	810	1000
100	3	10	0.9	50	100	1.68	900	1000
110	3	10	0.9	50	110	3.8	990	1000

Appendix C

Table C1: Results for the Chance-Constrained Model (50 Scenarios).

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
20	2	5	0.2	0.1	20	7.5	10	10	1000
30	2	5	0.2	0.1	30	12.63	10	15	1000
40	2	5	0.2	0.1	40	17.75	10	20	1000
50	2	5	0.2	0.1	50	22.8	10	25	1000
60	2	5	0.2	0.1	60	27.09	10	30	1000
70	2	5	0.2	0.1	70	15.9	10	35	1000
80	2	5	0.2	0.1	80	26.35	10	40	999
90	2	5	0.2	0.1	90	544.1	10	45	1000
100	2	5	0.2	0.1	100	815.13	10	50	1000
110	2	5	0.2	0.1	110	1256.63	10	55	1000
20	3	5	0.2	0.1	20	11.96	10	10	1000
30	3	5	0.2	0.1	30	17.58	10	15	1000
40	3	5	0.2	0.1	40	26.07	10	20	1000
50	3	5	0.2	0.1	50	16.78	10	25	1000
60	3	5	0.2	0.1	60	20.19	10	30	1000
70	3	5	0.2	0.1	70	29.16	10	35	1000
80	3	5	0.2	0.1	80	17.32	10	40	995
90	3	5	0.2	0.1	90	1114.96	10	45	1000
100	3	5	0.2	0.1	100	2506.38	10	50	1000
110	3	5	0.2	0.1	110	3186.55	10	55	1000
20	2	5	0.2	0.3	20	6.71	10	30	1000
30	2	5	0.2	0.3	30	11.76	10	45	1000
40	2	5	0.2	0.3	40	18.15	10	60	1000
50	2	5	0.2	0.3	50	25.55	10	75	1000
60	2	5	0.2	0.3	60	29.25	10	90	1000
70	2	5	0.2	0.3	70	15.17	10	105	1000
80	2	5	0.2	0.3	80	10.46	10	120	999
90	2	5	0.2	0.3	90	446.05	10	135	1000
100	2	5	0.2	0.3	100	539.46	10	150	999
110	2	5	0.2	0.3	110	1240.66	10	165	1000
20	3	5	0.2	0.3	20	10.88	10	30	1000
30	3	5	0.2	0.3	30	17.45	10	45	1000
40	3	5	0.2	0.3	40	24.36	10	60	1000
50	3	5	0.2	0.3	50	13.67	10	75	1000
60	3	5	0.2	0.3	60	23.52	10	90	1000
70	3	5	0.2	0.3	70	20.78	10	105	1000
80	3	5	0.2	0.3	80	701.94	10	120	1000
90	3	5	0.2	0.3	90	1050.02	10	135	1000
100	3	5	0.2	0.3	100	2347.12	10	150	1000
110	3	5	0.2	0.3	110	2931.68	10	165	1000
20	2	5	0.2	0.5	20	8.25	10	50	1000
30	2	5	0.2	0.5	30	11.98	10	75	1000
40	2	5	0.2	0.5	40	19.25	10	100	1000
50	2	5	0.2	0.5	50	25.46	10	125	1000
60	2	5	0.2	0.5	60	29.08	10	150	1000
70	2	5	0.2	0.5	70	15.84	10	175	1000
80	2	5	0.2	0.5	80	661.78	10	200	1000
90	2	5	0.2	0.5	90	478.41	10	225	1000
100	2	5	0.2	0.5	100	1623.09	10	250	1000
110	2	5	0.2	0.5	110	457.75	10	275	994
20	3	5	0.2	0.5	20	11.5	10	50	1000
30	3	5	0.2	0.5	30	18.02	10	75	1000
40	3	5	0.2	0.5	40	25.12	10	100	1000
50	3	5	0.2	0.5	50	13.07	10	125	1000
60	3	5	0.2	0.5	60	19.59	10	150	1000
70	3	5	0.2	0.5	70	24.5	10	175	1000
80	3	5	0.2	0.5	80	494.44	10	200	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.2	0.5	90	754	10	225	1000
100	3	5	0.2	0.5	100	2278.95	10	250	1000
110	3	5	0.2	0.5	110	2409.7	10	275	1000
20	2	5	0.2	0.7	20	8.73	10	70	1000
30	2	5	0.2	0.7	30	12.41	10	105	1000
40	2	5	0.2	0.7	40	17.84	10	140	1000
50	2	5	0.2	0.7	50	21.88	10	175	1000
60	2	5	0.2	0.7	60	26.43	10	210	1000
70	2	5	0.2	0.7	70	18.71	10	245	1000
80	2	5	0.2	0.7	80	262.41	10	280	1000
90	2	5	0.2	0.7	90	788.72	10	315	1000
100	2	5	0.2	0.7	100	636.61	10	350	1000
110	2	5	0.2	0.7	110	1140.82	10	385	1000
20	3	5	0.2	0.7	20	10.13	10	70	1000
30	3	5	0.2	0.7	30	17.67	10	105	1000
40	3	5	0.2	0.7	40	24.71	10	140	1000
50	3	5	0.2	0.7	50	16.48	10	175	1000
60	3	5	0.2	0.7	60	19.1	10	210	1000
70	3	5	0.2	0.7	70	384.74	10	245	1000
80	3	5	0.2	0.7	80	434.34	10	280	1000
90	3	5	0.2	0.7	90	662.42	10	315	1000
100	3	5	0.2	0.7	100	1414.05	10	350	1000
110	3	5	0.2	0.7	110	1903.21	10	385	1000
20	2	5	0.2	0.9	20	9.67	10	90	1000
30	2	5	0.2	0.9	30	13.75	10	135	1000
40	2	5	0.2	0.9	40	17.32	10	180	1000
50	2	5	0.2	0.9	50	22.73	10	225	1000
60	2	5	0.2	0.9	60	27.3	10	270	1000
70	2	5	0.2	0.9	70	151.44	10	315	1000
80	2	5	0.2	0.9	80	690.29	10	360	1000
90	2	5	0.2	0.9	90	1486.26	10	405	1000
100	2	5	0.2	0.9	100	783.61	10	450	1000
110	2	5	0.2	0.9	110	1211.49	10	495	1000
20	3	5	0.2	0.9	20	10.07	10	90	1000
30	3	5	0.2	0.9	30	18.05	10	135	1000
40	3	5	0.2	0.9	40	25.23	10	180	1000
50	3	5	0.2	0.9	50	15.55	10	225	1000
60	3	5	0.2	0.9	60	23.13	10	270	1000
70	3	5	0.2	0.9	70	378.17	10	315	1000
80	3	5	0.2	0.9	80	347.13	10	360	1000
90	3	5	0.2	0.9	90	624.59	10	405	1000
100	3	5	0.2	0.9	100	2018.46	10	450	1000
110	3	5	0.2	0.9	110	970.48	10	495	1000
20	2	5	0.4	0.1	20	8.84	20	10	1000
30	2	5	0.4	0.1	30	11.93	20	15	1000
40	2	5	0.4	0.1	40	20.32	20	20	1000
50	2	5	0.4	0.1	50	21.03	20	25	1000
60	2	5	0.4	0.1	60	29.63	20	30	1000
70	2	5	0.4	0.1	70	18.07	20	35	1000
80	2	5	0.4	0.1	80	17.03	20	40	999
90	2	5	0.4	0.1	90	1024.15	20	45	1000
100	2	5	0.4	0.1	100	1556.3	20	50	981
110	2	5	0.4	0.1	110	1704.17	20	55	1000
20	3	5	0.4	0.1	20	11.57	20	10	1000
30	3	5	0.4	0.1	30	17.5	20	15	1000
40	3	5	0.4	0.1	40	25.47	20	20	1000
50	3	5	0.4	0.1	50	18	20	25	1000
60	3	5	0.4	0.1	60	20.09	20	30	1000
70	3	5	0.4	0.1	70	22.06	20	35	1000
80	3	5	0.4	0.1	80	31.06	20	40	995

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.4	0.1	90	774.47	20	45	1000
100	3	5	0.4	0.1	100	1716.26	20	50	1000
110	3	5	0.4	0.1	110	1912.12	20	55	1000
20	2	5	0.4	0.3	20	7.34	20	30	1000
30	2	5	0.4	0.3	30	13.23	20	45	1000
40	2	5	0.4	0.3	40	16.88	20	60	1000
50	2	5	0.4	0.3	50	22.97	20	75	1000
60	2	5	0.4	0.3	60	27.33	20	90	1000
70	2	5	0.4	0.3	70	13.37	20	105	1000
80	2	5	0.4	0.3	80	325.96	20	120	1000
90	2	5	0.4	0.3	90	953.02	20	135	1000
100	2	5	0.4	0.3	100	815.77	20	150	1000
110	2	5	0.4	0.3	110	740.89	20	165	1000
20	3	5	0.4	0.3	20	10.96	20	30	1000
30	3	5	0.4	0.3	30	20.42	20	45	1000
40	3	5	0.4	0.3	40	23.12	20	60	1000
50	3	5	0.4	0.3	50	17.09	20	75	1000
60	3	5	0.4	0.3	60	19.3	20	90	1000
70	3	5	0.4	0.3	70	25.04	20	105	1000
80	3	5	0.4	0.3	80	501.42	20	120	1000
90	3	5	0.4	0.3	90	334.17	20	135	1000
100	3	5	0.4	0.3	100	1236.9	20	150	1000
110	3	5	0.4	0.3	110	3256.09	20	165	1000
20	2	5	0.4	0.5	20	6.9	20	50	1000
30	2	5	0.4	0.5	30	12	20	75	1000
40	2	5	0.4	0.5	40	12.71	20	100	1000
50	2	5	0.4	0.5	50	22.63	20	125	1000
60	2	5	0.4	0.5	60	30.51	20	150	1000
70	2	5	0.4	0.5	70	13.97	20	175	1000
80	2	5	0.4	0.5	80	266.55	20	200	1000
90	2	5	0.4	0.5	90	445.59	20	225	1000
100	2	5	0.4	0.5	100	734.31	20	250	1000
110	2	5	0.4	0.5	110	2996.55	20	275	1000
20	3	5	0.4	0.5	20	11.87	20	50	1000
30	3	5	0.4	0.5	30	16.41	20	75	1000
40	3	5	0.4	0.5	40	23.96	20	100	1000
50	3	5	0.4	0.5	50	16.84	20	125	1000
60	3	5	0.4	0.5	60	22.18	20	150	1000
70	3	5	0.4	0.5	70	28.85	20	175	1000
80	3	5	0.4	0.5	80	537.61	20	200	1000
90	3	5	0.4	0.5	90	1003.03	20	225	1000
100	3	5	0.4	0.5	100	1184.9	20	250	1000
110	3	5	0.4	0.5	110	2696.74	20	275	1000
20	2	5	0.4	0.7	20	9.25	20	70	1000
30	2	5	0.4	0.7	30	11.95	20	105	1000
40	2	5	0.4	0.7	40	18.59	20	140	1000
50	2	5	0.4	0.7	50	23.5	20	175	1000
60	2	5	0.4	0.7	60	30.04	20	210	1000
70	2	5	0.4	0.7	70	21	20	245	1000
80	2	5	0.4	0.7	80	288.05	20	280	1000
90	2	5	0.4	0.7	90	328.28	20	315	1000
100	2	5	0.4	0.7	100	488.3	20	350	1000
110	2	5	0.4	0.7	110	587.22	20	385	1000
20	3	5	0.4	0.7	20	12.63	20	70	1000
30	3	5	0.4	0.7	30	12.53	20	105	1000
40	3	5	0.4	0.7	40	26.24	20	140	1000
50	3	5	0.4	0.7	50	20.25	20	175	1000
60	3	5	0.4	0.7	60	22.05	20	210	1000
70	3	5	0.4	0.7	70	300.94	20	245	1000
80	3	5	0.4	0.7	80	422.24	20	280	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.4	0.7	90	596.84	20	315	1000
100	3	5	0.4	0.7	100	1065.55	20	350	1000
110	3	5	0.4	0.7	110	1273.32	20	385	1000
20	2	5	0.4	0.9	20	5.3	20	90	1000
30	2	5	0.4	0.9	30	11.71	20	135	1000
40	2	5	0.4	0.9	40	16.94	20	180	1000
50	2	5	0.4	0.9	50	23.78	20	225	1000
60	2	5	0.4	0.9	60	30.56	20	270	1000
70	2	5	0.4	0.9	70	146.75	20	315	1000
80	2	5	0.4	0.9	80	132.15	20	360	1000
90	2	5	0.4	0.9	90	443.68	20	405	1000
100	2	5	0.4	0.9	100	639.96	20	450	1000
110	2	5	0.4	0.9	110	642.53	20	495	1000
20	3	5	0.4	0.9	20	11.68	20	90	1000
30	3	5	0.4	0.9	30	11.52	20	135	1000
40	3	5	0.4	0.9	40	22.09	20	180	1000
50	3	5	0.4	0.9	50	16.5	20	225	1000
60	3	5	0.4	0.9	60	19.93	20	270	1000
70	3	5	0.4	0.9	70	219	20	315	1000
80	3	5	0.4	0.9	80	463.65	20	360	1000
90	3	5	0.4	0.9	90	607.2	20	405	1000
100	3	5	0.4	0.9	100	1112.03	20	450	1000
110	3	5	0.4	0.9	110	804.54	20	495	1000
20	2	5	0.6	0.1	20	8.89	30	10	1000
30	2	5	0.6	0.1	30	12.02	30	15	1000
40	2	5	0.6	0.1	40	18.86	30	20	1000
50	2	5	0.6	0.1	50	22.79	30	25	1000
60	2	5	0.6	0.1	60	28.42	30	30	1000
70	2	5	0.6	0.1	70	13.75	30	35	1000
80	2	5	0.6	0.1	80	9.64	30	40	999
90	2	5	0.6	0.1	90	490.3	30	45	1000
100	2	5	0.6	0.1	100	982	30	50	1000
110	2	5	0.6	0.1	110	884.84	30	55	1000
20	3	5	0.6	0.1	20	10.88	30	10	1000
30	3	5	0.6	0.1	30	17.6	30	15	1000
40	3	5	0.6	0.1	40	23.24	30	20	1000
50	3	5	0.6	0.1	50	14.07	30	25	1000
60	3	5	0.6	0.1	60	23.89	30	30	1000
70	3	5	0.6	0.1	70	23.91	30	35	1000
80	3	5	0.6	0.1	80	26.99	30	40	995
90	3	5	0.6	0.1	90	486.37	30	45	984
100	3	5	0.6	0.1	100	1459.1	30	50	1000
110	3	5	0.6	0.1	110	3595.62	30	55	1000
20	2	5	0.6	0.3	20	7.45	30	30	1000
30	2	5	0.6	0.3	30	12.07	30	45	1000
40	2	5	0.6	0.3	40	16.41	30	60	1000
50	2	5	0.6	0.3	50	24.29	30	75	1000
60	2	5	0.6	0.3	60	25.82	30	90	1000
70	2	5	0.6	0.3	70	15.5	30	105	1000
80	2	5	0.6	0.3	80	250.17	30	120	1000
90	2	5	0.6	0.3	90	435.86	30	135	1000
100	2	5	0.6	0.3	100	1185.08	30	150	1000
110	2	5	0.6	0.3	110	1292.92	30	165	1000
20	3	5	0.6	0.3	20	10.31	30	30	1000
30	3	5	0.6	0.3	30	17.45	30	45	1000
40	3	5	0.6	0.3	40	26.1	30	60	1000
50	3	5	0.6	0.3	50	12.96	30	75	1000
60	3	5	0.6	0.3	60	17.28	30	90	1000
70	3	5	0.6	0.3	70	21.62	30	105	1000
80	3	5	0.6	0.3	80	569.9	30	120	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.6	0.3	90	875.95	30	135	1000
100	3	5	0.6	0.3	100	1564.63	30	150	1000
110	3	5	0.6	0.3	110	1538.72	30	165	998
20	2	5	0.6	0.5	20	6.63	30	50	1000
30	2	5	0.6	0.5	30	12.44	30	75	1000
40	2	5	0.6	0.5	40	18.57	30	100	1000
50	2	5	0.6	0.5	50	24.55	30	125	1000
60	2	5	0.6	0.5	60	27.21	30	150	1000
70	2	5	0.6	0.5	70	17.65	30	175	1000
80	2	5	0.6	0.5	80	268.61	30	200	1000
90	2	5	0.6	0.5	90	484.5	30	225	1000
100	2	5	0.6	0.5	100	720.29	30	250	1000
110	2	5	0.6	0.5	110	738.1	30	275	1000
20	3	5	0.6	0.5	20	11.23	30	50	1000
30	3	5	0.6	0.5	30	18.17	30	75	1000
40	3	5	0.6	0.5	40	24.28	30	100	1000
50	3	5	0.6	0.5	50	16.73	30	125	1000
60	3	5	0.6	0.5	60	21.67	30	150	1000
70	3	5	0.6	0.5	70	27.59	30	175	1000
80	3	5	0.6	0.5	80	337.71	30	200	1000
90	3	5	0.6	0.5	90	848.04	30	225	1000
100	3	5	0.6	0.5	100	1544.91	30	250	1000
110	3	5	0.6	0.5	110	1876.78	30	275	1000
20	2	5	0.6	0.7	20	8.81	30	70	1000
30	2	5	0.6	0.7	30	11.77	30	105	1000
40	2	5	0.6	0.7	40	17.52	30	140	1000
50	2	5	0.6	0.7	50	22.92	30	175	1000
60	2	5	0.6	0.7	60	32.08	30	210	1000
70	2	5	0.6	0.7	70	20.75	30	245	1000
80	2	5	0.6	0.7	80	405.48	30	280	1000
90	2	5	0.6	0.7	90	486.02	30	315	1000
100	2	5	0.6	0.7	100	493.06	30	350	1000
110	2	5	0.6	0.7	110	519.17	30	385	1000
20	3	5	0.6	0.7	20	11.2	30	70	1000
30	3	5	0.6	0.7	30	17.16	30	105	1000
40	3	5	0.6	0.7	40	24.59	30	140	1000
50	3	5	0.6	0.7	50	14.53	30	175	1000
60	3	5	0.6	0.7	60	22.7	30	210	1000
70	3	5	0.6	0.7	70	266.34	30	245	1000
80	3	5	0.6	0.7	80	365.05	30	280	1000
90	3	5	0.6	0.7	90	432.5	30	315	1000
100	3	5	0.6	0.7	100	900.19	30	350	1000
110	3	5	0.6	0.7	110	1072.25	30	385	1000
20	2	5	0.6	0.9	20	8.26	30	90	1000
30	2	5	0.6	0.9	30	12.3	30	135	1000
40	2	5	0.6	0.9	40	17.26	30	180	1000
50	2	5	0.6	0.9	50	20.87	30	225	1000
60	2	5	0.6	0.9	60	27.72	30	270	1000
70	2	5	0.6	0.9	70	229.3	30	315	1000
80	2	5	0.6	0.9	80	363.8	30	360	1000
90	2	5	0.6	0.9	90	440.05	30	405	1000
100	2	5	0.6	0.9	100	738.79	30	450	1000
110	2	5	0.6	0.9	110	390.55	30	495	1000
20	3	5	0.6	0.9	20	9.98	30	90	1000
30	3	5	0.6	0.9	30	19.66	30	135	1000
40	3	5	0.6	0.9	40	26.2	30	180	1000
50	3	5	0.6	0.9	50	19.62	30	225	1000
60	3	5	0.6	0.9	60	16.12	30	270	1000
70	3	5	0.6	0.9	70	216.11	30	315	1000
80	3	5	0.6	0.9	80	294.17	30	360	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.6	0.9	90	365.84	30	405	1000
100	3	5	0.6	0.9	100	468.07	30	450	1000
110	3	5	0.6	0.9	110	1654.32	30	495	1000
20	2	5	0.8	0.1	20	7.38	40	10	1000
30	2	5	0.8	0.1	30	11.73	40	15	1000
40	2	5	0.8	0.1	40	20.5	40	20	1000
50	2	5	0.8	0.1	50	23.92	40	25	1000
60	2	5	0.8	0.1	60	30.75	40	30	1000
70	2	5	0.8	0.1	70	17.03	40	35	1000
80	2	5	0.8	0.1	80	11.12	40	40	999
90	2	5	0.8	0.1	90	533.21	40	45	1000
100	2	5	0.8	0.1	100	880.7	40	50	996
110	2	5	0.8	0.1	110	842.22	40	55	1000
20	3	5	0.8	0.1	20	11.73	40	10	1000
30	3	5	0.8	0.1	30	17.8	40	15	1000
40	3	5	0.8	0.1	40	24.8	40	20	1000
50	3	5	0.8	0.1	50	14.3	40	25	1000
60	3	5	0.8	0.1	60	22.7	40	30	1000
70	3	5	0.8	0.1	70	24.05	40	35	1000
80	3	5	0.8	0.1	80	30.75	40	40	995
90	3	5	0.8	0.1	90	1133	40	45	1000
100	3	5	0.8	0.1	100	1323.37	40	50	1000
110	3	5	0.8	0.1	110	2041.68	40	55	1000
20	2	5	0.8	0.3	20	6.54	40	30	1000
30	2	5	0.8	0.3	30	11.09	40	45	1000
40	2	5	0.8	0.3	40	16.7	40	60	1000
50	2	5	0.8	0.3	50	23.81	40	75	1000
60	2	5	0.8	0.3	60	31.93	40	90	1000
70	2	5	0.8	0.3	70	13.28	40	105	1000
80	2	5	0.8	0.3	80	223.63	40	120	1000
90	2	5	0.8	0.3	90	330.75	40	135	1000
100	2	5	0.8	0.3	100	424.92	40	150	1000
110	2	5	0.8	0.3	110	2348.27	40	165	1000
20	3	5	0.8	0.3	20	10.43	40	30	1000
30	3	5	0.8	0.3	30	18.89	40	45	1000
40	3	5	0.8	0.3	40	23.48	40	60	1000
50	3	5	0.8	0.3	50	15.02	40	75	1000
60	3	5	0.8	0.3	60	21.61	40	90	1000
70	3	5	0.8	0.3	70	27.42	40	105	1000
80	3	5	0.8	0.3	80	586.76	40	120	1000
90	3	5	0.8	0.3	90	1632.76	40	135	1000
100	3	5	0.8	0.3	100	2035.7	40	150	1000
110	3	5	0.8	0.3	110	2511.47	40	165	1000
20	2	5	0.8	0.5	20	8.5	40	50	1000
30	2	5	0.8	0.5	30	12.71	40	75	1000
40	2	5	0.8	0.5	40	11.84	40	100	1000
50	2	5	0.8	0.5	50	21.59	40	125	1000
60	2	5	0.8	0.5	60	29	40	150	1000
70	2	5	0.8	0.5	70	21.61	40	175	1000
80	2	5	0.8	0.5	80	263.41	40	200	1000
90	2	5	0.8	0.5	90	698.31	40	225	1000
100	2	5	0.8	0.5	100	1241.97	40	250	1000
110	2	5	0.8	0.5	110	930.91	40	275	1000
20	3	5	0.8	0.5	20	12.34	40	50	1000
30	3	5	0.8	0.5	30	17.59	40	75	1000
40	3	5	0.8	0.5	40	23.25	40	100	1000
50	3	5	0.8	0.5	50	16.52	40	125	1000
60	3	5	0.8	0.5	60	22.48	40	150	1000
70	3	5	0.8	0.5	70	23.99	40	175	1000
80	3	5	0.8	0.5	80	392.09	40	200	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	0.8	0.5	90	786.59	40	225	1000
100	3	5	0.8	0.5	100	1475.46	40	250	1000
110	3	5	0.8	0.5	110	1234.04	40	275	1000
20	2	5	0.8	0.7	20	7.85	40	70	1000
30	2	5	0.8	0.7	30	12.75	40	105	1000
40	2	5	0.8	0.7	40	18.81	40	140	1000
50	2	5	0.8	0.7	50	24.55	40	175	1000
60	2	5	0.8	0.7	60	27.75	40	210	1000
70	2	5	0.8	0.7	70	16.37	40	245	1000
80	2	5	0.8	0.7	80	210.75	40	280	1000
90	2	5	0.8	0.7	90	470.41	40	315	1000
100	2	5	0.8	0.7	100	440.07	40	350	1000
110	2	5	0.8	0.7	110	891.71	40	385	1000
20	3	5	0.8	0.7	20	10.6	40	70	1000
30	3	5	0.8	0.7	30	17.84	40	105	1000
40	3	5	0.8	0.7	40	25.24	40	140	1000
50	3	5	0.8	0.7	50	13.18	40	175	1000
60	3	5	0.8	0.7	60	19.34	40	210	1000
70	3	5	0.8	0.7	70	229.9	40	245	1000
80	3	5	0.8	0.7	80	237.36	40	280	1000
90	3	5	0.8	0.7	90	756.48	40	315	1000
100	3	5	0.8	0.7	100	523.04	40	350	1000
110	3	5	0.8	0.7	110	768.17	40	385	1000
20	2	5	0.8	0.9	20	7.92	40	90	1000
30	2	5	0.8	0.9	30	13.42	40	135	1000
40	2	5	0.8	0.9	40	16.8	40	180	1000
50	2	5	0.8	0.9	50	22.65	40	225	1000
60	2	5	0.8	0.9	60	29.17	40	270	1000
70	2	5	0.8	0.9	70	202.51	40	315	1000
80	2	5	0.8	0.9	80	308.05	40	360	1000
90	2	5	0.8	0.9	90	275.88	40	405	1000
100	2	5	0.8	0.9	100	683.96	40	450	1000
110	2	5	0.8	0.9	110	579.35	40	495	1000
20	3	5	0.8	0.9	20	10.85	40	90	1000
30	3	5	0.8	0.9	30	18.34	40	135	1000
40	3	5	0.8	0.9	40	23.89	40	180	1000
50	3	5	0.8	0.9	50	15.14	40	225	1000
60	3	5	0.8	0.9	60	20.67	40	270	1000
70	3	5	0.8	0.9	70	116.82	40	315	1000
80	3	5	0.8	0.9	80	130.69	40	360	1000
90	3	5	0.8	0.9	90	246.87	40	405	1000
100	3	5	0.8	0.9	100	1083.72	40	450	1000
110	3	5	0.8	0.9	110	987.14	40	495	1000
20	2	5	1	0.1	20	6.96	50	10	1000
30	2	5	1	0.1	30	13.34	50	15	1000
40	2	5	1	0.1	40	17.17	50	20	1000
50	2	5	1	0.1	50	23.76	50	25	1000
60	2	5	1	0.1	60	25.38	50	30	1000
70	2	5	1	0.1	70	17.88	50	35	1000
80	2	5	1	0.1	80	19.12	50	40	999
90	2	5	1	0.1	90	592.64	50	45	1000
100	2	5	1	0.1	100	757.91	50	50	1000
110	2	5	1	0.1	110	1312.9	50	55	1000
20	3	5	1	0.1	20	11.3	50	10	1000
30	3	5	1	0.1	30	18.13	50	15	1000
40	3	5	1	0.1	40	25.72	50	20	1000
50	3	5	1	0.1	50	19.98	50	25	1000
60	3	5	1	0.1	60	22.12	50	30	1000
70	3	5	1	0.1	70	28	50	35	1000
80	3	5	1	0.1	80	16.79	50	40	995

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	1	0.1	90	1445.37	50	45	1000
100	3	5	1	0.1	100	1326.04	50	50	1000
110	3	5	1	0.1	110	1834.58	50	55	1000
20	2	5	1	0.3	20	8.25	50	30	1000
30	2	5	1	0.3	30	12.65	50	45	1000
40	2	5	1	0.3	40	18.27	50	60	1000
50	2	5	1	0.3	50	23.84	50	75	1000
60	2	5	1	0.3	60	27.54	50	90	1000
70	2	5	1	0.3	70	14.7	50	105	1000
80	2	5	1	0.3	80	258.26	50	120	1000
90	2	5	1	0.3	90	571.86	50	135	1000
100	2	5	1	0.3	100	565.62	50	150	1000
110	2	5	1	0.3	110	1172.27	50	165	1000
20	3	5	1	0.3	20	9.64	50	30	1000
30	3	5	1	0.3	30	15.68	50	45	1000
40	3	5	1	0.3	40	23.45	50	60	1000
50	3	5	1	0.3	50	16.63	50	75	1000
60	3	5	1	0.3	60	21.65	50	90	1000
70	3	5	1	0.3	70	23.73	50	105	1000
80	3	5	1	0.3	80	565.3	50	120	1000
90	3	5	1	0.3	90	1437.2	50	135	1000
100	3	5	1	0.3	100	1453.65	50	150	1000
110	3	5	1	0.3	110	2454.4	50	165	1000
20	2	5	1	0.5	20	8.33	50	50	1000
30	2	5	1	0.5	30	12.46	50	75	1000
40	2	5	1	0.5	40	16.95	50	100	1000
50	2	5	1	0.5	50	22.2	50	125	1000
60	2	5	1	0.5	60	30.47	50	150	1000
70	2	5	1	0.5	70	13.59	50	175	1000
80	2	5	1	0.5	80	100.89	50	200	1000
90	2	5	1	0.5	90	492.09	50	225	1000
100	2	5	1	0.5	100	501.31	50	250	1000
110	2	5	1	0.5	110	1366.55	50	275	1000
20	3	5	1	0.5	20	13.31	50	50	1000
30	3	5	1	0.5	30	16.88	50	75	1000
40	3	5	1	0.5	40	26.96	50	100	1000
50	3	5	1	0.5	50	15.15	50	125	1000
60	3	5	1	0.5	60	18.18	50	150	1000
70	3	5	1	0.5	70	25.21	50	175	1000
80	3	5	1	0.5	80	550.7	50	200	1000
90	3	5	1	0.5	90	711.93	50	225	1000
100	3	5	1	0.5	100	1556.85	50	250	1000
110	3	5	1	0.5	110	3757.09	50	275	1000
20	2	5	1	0.7	20	8.81	50	70	1000
30	2	5	1	0.7	30	11.01	50	105	1000
40	2	5	1	0.7	40	16.88	50	140	1000
50	2	5	1	0.7	50	19.51	50	175	1000
60	2	5	1	0.7	60	26.55	50	210	1000
70	2	5	1	0.7	70	121.99	50	245	1000
80	2	5	1	0.7	80	326.2	50	280	1000
90	2	5	1	0.7	90	439.11	50	315	1000
100	2	5	1	0.7	100	281.5	50	350	1000
110	2	5	1	0.7	110	915.14	50	385	1000
20	3	5	1	0.7	20	12.39	50	70	1000
30	3	5	1	0.7	30	17.99	50	105	1000
40	3	5	1	0.7	40	25.37	50	140	1000
50	3	5	1	0.7	50	18.62	50	175	1000
60	3	5	1	0.7	60	14.89	50	210	1000
70	3	5	1	0.7	70	173.23	50	245	1000
80	3	5	1	0.7	80	250.5	50	280	1000

Continued on next page

Table **C1** – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	5	1	0.7	90	257.08	50	315	1000
100	3	5	1	0.7	100	478.03	50	350	1000
110	3	5	1	0.7	110	736.36	50	385	1000
20	2	5	1	0.9	20	8.17	50	90	1000
30	2	5	1	0.9	30	7.8	50	135	1000
40	2	5	1	0.9	40	17.87	50	180	1000
50	2	5	1	0.9	50	23.95	50	225	1000
60	2	5	1	0.9	60	28.21	50	270	1000
70	2	5	1	0.9	70	203	50	315	1000
80	2	5	1	0.9	80	174.15	50	360	1000
90	2	5	1	0.9	90	175.94	50	405	1000
100	2	5	1	0.9	100	321.63	50	450	1000
110	2	5	1	0.9	110	966.49	50	495	1000
20	3	5	1	0.9	20	11.09	50	90	1000
30	3	5	1	0.9	30	17.52	50	135	1000
40	3	5	1	0.9	40	25.71	50	180	1000
50	3	5	1	0.9	50	15.63	50	225	1000
60	3	5	1	0.9	60	18.62	50	270	1000
70	3	5	1	0.9	70	124.16	50	315	1000
80	3	5	1	0.9	80	233.09	50	360	1000
90	3	5	1	0.9	90	309.97	50	405	1000
100	3	5	1	0.9	100	439.81	50	450	1000
110	3	5	1	0.9	110	567.71	50	495	1000
20	2	10	0.2	0.1	20	8.54	10	20	1000
30	2	10	0.2	0.1	30	13.52	10	30	1000
40	2	10	0.2	0.1	40	17.84	10	40	1000
50	2	10	0.2	0.1	50	24.04	10	50	1000
60	2	10	0.2	0.1	60	29.78	10	60	1000
70	2	10	0.2	0.1	70	16.8	10	70	1000
80	2	10	0.2	0.1	80	21.94	10	80	999
90	2	10	0.2	0.1	90	600.15	10	90	1000
100	2	10	0.2	0.1	100	868.55	10	100	1000
110	2	10	0.2	0.1	110	1649.84	10	110	1000
20	3	10	0.2	0.1	20	10.47	10	20	1000
30	3	10	0.2	0.1	30	11.38	10	30	1000
40	3	10	0.2	0.1	40	23.24	10	40	1000
50	3	10	0.2	0.1	50	14.5	10	50	1000
60	3	10	0.2	0.1	60	19.26	10	60	1000
70	3	10	0.2	0.1	70	25.81	10	70	1000
80	3	10	0.2	0.1	80	644.08	10	80	1000
90	3	10	0.2	0.1	90	924.27	10	90	1000
100	3	10	0.2	0.1	100	1199.65	10	100	1000
110	3	10	0.2	0.1	110	1347.09	10	110	1000
20	2	10	0.2	0.3	20	8.03	10	60	1000
30	2	10	0.2	0.3	30	12.08	10	90	1000
40	2	10	0.2	0.3	40	17.88	10	120	1000
50	2	10	0.2	0.3	50	24.42	10	150	1000
60	2	10	0.2	0.3	60	27.39	10	180	1000
70	2	10	0.2	0.3	70	20.7	10	210	1000
80	2	10	0.2	0.3	80	212.66	10	240	1000
90	2	10	0.2	0.3	90	631.34	10	270	993
100	2	10	0.2	0.3	100	801.65	10	300	1000
110	2	10	0.2	0.3	110	586.25	10	330	1000
20	3	10	0.2	0.3	20	10.71	10	60	1000
30	3	10	0.2	0.3	30	18.7	10	90	1000
40	3	10	0.2	0.3	40	26.77	10	120	1000
50	3	10	0.2	0.3	50	16.52	10	150	1000
60	3	10	0.2	0.3	60	22.83	10	180	1000
70	3	10	0.2	0.3	70	29.77	10	210	1000
80	3	10	0.2	0.3	80	986.74	10	240	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.2	0.3	90	884.67	10	270	1000
100	3	10	0.2	0.3	100	1345.78	10	300	1000
110	3	10	0.2	0.3	110	2385.05	10	330	1000
20	2	10	0.2	0.5	20	5.45	10	100	1000
30	2	10	0.2	0.5	30	13.44	10	150	1000
40	2	10	0.2	0.5	40	19.76	10	200	1000
50	2	10	0.2	0.5	50	22.66	10	250	1000
60	2	10	0.2	0.5	60	27.55	10	300	1000
70	2	10	0.2	0.5	70	135.06	10	350	1000
80	2	10	0.2	0.5	80	269.18	10	400	1000
90	2	10	0.2	0.5	90	510.51	10	450	1000
100	2	10	0.2	0.5	100	539	10	500	1000
110	2	10	0.2	0.5	110	2485.08	10	550	1000
20	3	10	0.2	0.5	20	10.22	10	100	1000
30	3	10	0.2	0.5	30	18.82	10	150	1000
40	3	10	0.2	0.5	40	25.51	10	200	1000
50	3	10	0.2	0.5	50	16.33	10	250	1000
60	3	10	0.2	0.5	60	22.09	10	300	1000
70	3	10	0.2	0.5	70	379.18	10	350	1000
80	3	10	0.2	0.5	80	755	10	400	1000
90	3	10	0.2	0.5	90	982.55	10	450	1000
100	3	10	0.2	0.5	100	1495.86	10	500	1000
110	3	10	0.2	0.5	110	2087.35	10	550	1000
20	2	10	0.2	0.7	20	8.13	10	140	1000
30	2	10	0.2	0.7	30	11.19	10	210	1000
40	2	10	0.2	0.7	40	19.32	10	280	1000
50	2	10	0.2	0.7	50	21.55	10	350	1000
60	2	10	0.2	0.7	60	30.25	10	420	1000
70	2	10	0.2	0.7	70	215.4	10	490	1000
80	2	10	0.2	0.7	80	561.6	10	560	1000
90	2	10	0.2	0.7	90	242.19	10	630	1000
100	2	10	0.2	0.7	100	682.21	10	700	1000
110	2	10	0.2	0.7	110	1313.84	10	770	1000
20	3	10	0.2	0.7	20	10.02	10	140	1000
30	3	10	0.2	0.7	30	18.61	10	210	1000
40	3	10	0.2	0.7	40	24.41	10	280	1000
50	3	10	0.2	0.7	50	15.95	10	350	1000
60	3	10	0.2	0.7	60	232.05	10	420	1000
70	3	10	0.2	0.7	70	289.55	10	490	1000
80	3	10	0.2	0.7	80	546.71	10	560	1000
90	3	10	0.2	0.7	90	630.2	10	630	1000
100	3	10	0.2	0.7	100	1764.25	10	700	1000
110	3	10	0.2	0.7	110	2990.46	10	770	1000
20	2	10	0.2	0.9	20	7.24	10	180	1000
30	2	10	0.2	0.9	30	13.59	10	270	1000
40	2	10	0.2	0.9	40	17.07	10	360	1000
50	2	10	0.2	0.9	50	23.62	10	450	1000
60	2	10	0.2	0.9	60	89.25	10	540	1000
70	2	10	0.2	0.9	70	286.91	10	630	1000
80	2	10	0.2	0.9	80	1193.13	10	720	1000
90	2	10	0.2	0.9	90	339.31	10	810	1000
100	2	10	0.2	0.9	100	1634.88	10	900	1000
110	2	10	0.2	0.9	110	3811.36	10	990	1000
20	3	10	0.2	0.9	20	11.65	10	180	1000
30	3	10	0.2	0.9	30	18.44	10	270	1000
40	3	10	0.2	0.9	40	24.67	10	360	1000
50	3	10	0.2	0.9	50	18.23	10	450	1000
60	3	10	0.2	0.9	60	121.95	10	540	1000
70	3	10	0.2	0.9	70	452.13	10	630	1000
80	3	10	0.2	0.9	80	856.67	10	720	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.2	0.9	90	984.25	10	810	1000
100	3	10	0.2	0.9	100	1612.59	10	900	1000
110	3	10	0.2	0.9	110	2460.59	10	990	1000
20	2	10	0.4	0.1	20	8.19	20	20	1000
30	2	10	0.4	0.1	30	12.66	20	30	1000
40	2	10	0.4	0.1	40	21.2	20	40	1000
50	2	10	0.4	0.1	50	21.96	20	50	1000
60	2	10	0.4	0.1	60	29.58	20	60	1000
70	2	10	0.4	0.1	70	18.91	20	70	1000
80	2	10	0.4	0.1	80	21.51	20	80	999
90	2	10	0.4	0.1	90	263.58	20	90	990
100	2	10	0.4	0.1	100	853.49	20	100	998
110	2	10	0.4	0.1	110	741.17	20	110	1000
20	3	10	0.4	0.1	20	11.39	20	20	1000
30	3	10	0.4	0.1	30	19.58	20	30	1000
40	3	10	0.4	0.1	40	23.66	20	40	1000
50	3	10	0.4	0.1	50	18.4	20	50	1000
60	3	10	0.4	0.1	60	22.49	20	60	1000
70	3	10	0.4	0.1	70	27.77	20	70	1000
80	3	10	0.4	0.1	80	722.66	20	80	1000
90	3	10	0.4	0.1	90	1139.96	20	90	1000
100	3	10	0.4	0.1	100	2050.44	20	100	1000
110	3	10	0.4	0.1	110	3075.67	20	110	1000
20	2	10	0.4	0.3	20	7.66	20	60	1000
30	2	10	0.4	0.3	30	12.42	20	90	1000
40	2	10	0.4	0.3	40	16.07	20	120	1000
50	2	10	0.4	0.3	50	23.51	20	150	1000
60	2	10	0.4	0.3	60	28.24	20	180	1000
70	2	10	0.4	0.3	70	20.92	20	210	1000
80	2	10	0.4	0.3	80	262.63	20	240	998
90	2	10	0.4	0.3	90	600.28	20	270	1000
100	2	10	0.4	0.3	100	1427.96	20	300	1000
110	2	10	0.4	0.3	110	1391.67	20	330	1000
20	3	10	0.4	0.3	20	11.13	20	60	1000
30	3	10	0.4	0.3	30	18.16	20	90	1000
40	3	10	0.4	0.3	40	22.96	20	120	1000
50	3	10	0.4	0.3	50	14.94	20	150	1000
60	3	10	0.4	0.3	60	23.68	20	180	1000
70	3	10	0.4	0.3	70	22.84	20	210	1000
80	3	10	0.4	0.3	80	408.12	20	240	1000
90	3	10	0.4	0.3	90	585.74	20	270	998
100	3	10	0.4	0.3	100	2164.85	20	300	1000
110	3	10	0.4	0.3	110	2565.53	20	330	1000
20	2	10	0.4	0.5	20	7.96	20	100	1000
30	2	10	0.4	0.5	30	13.26	20	150	1000
40	2	10	0.4	0.5	40	12.15	20	200	1000
50	2	10	0.4	0.5	50	23.59	20	250	1000
60	2	10	0.4	0.5	60	27.89	20	300	1000
70	2	10	0.4	0.5	70	262.84	20	350	1000
80	2	10	0.4	0.5	80	318.18	20	400	1000
90	2	10	0.4	0.5	90	367.09	20	450	1000
100	2	10	0.4	0.5	100	582.74	20	500	1000
110	2	10	0.4	0.5	110	449.01	20	550	1000
20	3	10	0.4	0.5	20	11.31	20	100	1000
30	3	10	0.4	0.5	30	19.23	20	150	1000
40	3	10	0.4	0.5	40	24.88	20	200	1000
50	3	10	0.4	0.5	50	13.38	20	250	1000
60	3	10	0.4	0.5	60	19.03	20	300	1000
70	3	10	0.4	0.5	70	368.68	20	350	1000
80	3	10	0.4	0.5	80	566.21	20	400	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.4	0.5	90	1021.75	20	450	1000
100	3	10	0.4	0.5	100	1435.67	20	500	1000
110	3	10	0.4	0.5	110	2121.25	20	550	1000
20	2	10	0.4	0.7	20	8.25	20	140	1000
30	2	10	0.4	0.7	30	11.48	20	210	1000
40	2	10	0.4	0.7	40	11.86	20	280	1000
50	2	10	0.4	0.7	50	24.52	20	350	1000
60	2	10	0.4	0.7	60	30.28	20	420	1000
70	2	10	0.4	0.7	70	112.34	20	490	1000
80	2	10	0.4	0.7	80	345.39	20	560	1000
90	2	10	0.4	0.7	90	463.14	20	630	1000
100	2	10	0.4	0.7	100	364.65	20	700	1000
110	2	10	0.4	0.7	110	1860.1	20	770	1000
20	3	10	0.4	0.7	20	10.15	20	140	1000
30	3	10	0.4	0.7	30	15.77	20	210	1000
40	3	10	0.4	0.7	40	23.13	20	280	1000
50	3	10	0.4	0.7	50	15.13	20	350	1000
60	3	10	0.4	0.7	60	167.83	20	420	1000
70	3	10	0.4	0.7	70	473.06	20	490	1000
80	3	10	0.4	0.7	80	430.5	20	560	1000
90	3	10	0.4	0.7	90	654.08	20	630	1000
100	3	10	0.4	0.7	100	837.59	20	700	1000
110	3	10	0.4	0.7	110	880.13	20	770	1000
20	2	10	0.4	0.9	20	7.71	20	180	1000
30	2	10	0.4	0.9	30	12.09	20	270	1000
40	2	10	0.4	0.9	40	16.85	20	360	1000
50	2	10	0.4	0.9	50	22.01	20	450	1000
60	2	10	0.4	0.9	60	120.13	20	540	1000
70	2	10	0.4	0.9	70	392.65	20	630	1000
80	2	10	0.4	0.9	80	393.12	20	720	1000
90	2	10	0.4	0.9	90	218.25	20	810	1000
100	2	10	0.4	0.9	100	1753.86	20	900	1000
110	2	10	0.4	0.9	110	2366.63	20	990	1000
20	3	10	0.4	0.9	20	11.83	20	180	1000
30	3	10	0.4	0.9	30	13.34	20	270	1000
40	3	10	0.4	0.9	40	26	20	360	1000
50	3	10	0.4	0.9	50	15.97	20	450	1000
60	3	10	0.4	0.9	60	138.09	20	540	1000
70	3	10	0.4	0.9	70	403.65	20	630	1000
80	3	10	0.4	0.9	80	549.4	20	720	1000
90	3	10	0.4	0.9	90	690.72	20	810	1000
100	3	10	0.4	0.9	100	1953.3	20	900	1000
110	3	10	0.4	0.9	110	621.14	20	990	1000
20	2	10	0.6	0.1	20	7.85	30	20	1000
30	2	10	0.6	0.1	30	12.75	30	30	1000
40	2	10	0.6	0.1	40	18.3	30	40	1000
50	2	10	0.6	0.1	50	20.93	30	50	1000
60	2	10	0.6	0.1	60	28.5	30	60	1000
70	2	10	0.6	0.1	70	14.82	30	70	1000
80	2	10	0.6	0.1	80	17.25	30	80	999
90	2	10	0.6	0.1	90	690.02	30	90	1000
100	2	10	0.6	0.1	100	1157.55	30	100	998
110	2	10	0.6	0.1	110	2713.77	30	110	999
20	3	10	0.6	0.1	20	9.77	30	20	1000
30	3	10	0.6	0.1	30	11.49	30	30	1000
40	3	10	0.6	0.1	40	27.21	30	40	1000
50	3	10	0.6	0.1	50	13.58	30	50	1000
60	3	10	0.6	0.1	60	22.48	30	60	1000
70	3	10	0.6	0.1	70	24.79	30	70	1000
80	3	10	0.6	0.1	80	513.01	30	80	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.6	0.1	90	960.26	30	90	1000
100	3	10	0.6	0.1	100	1555.21	30	100	1000
110	3	10	0.6	0.1	110	2450.92	30	110	1000
20	2	10	0.6	0.3	20	8.1	30	60	1000
30	2	10	0.6	0.3	30	11.88	30	90	1000
40	2	10	0.6	0.3	40	17.16	30	120	1000
50	2	10	0.6	0.3	50	22.77	30	150	1000
60	2	10	0.6	0.3	60	25.55	30	180	1000
70	2	10	0.6	0.3	70	17.21	30	210	1000
80	2	10	0.6	0.3	80	292.66	30	240	1000
90	2	10	0.6	0.3	90	295.63	30	270	999
100	2	10	0.6	0.3	100	1035.38	30	300	1000
110	2	10	0.6	0.3	110	1063.38	30	330	1000
20	3	10	0.6	0.3	20	11.49	30	60	1000
30	3	10	0.6	0.3	30	17.94	30	90	1000
40	3	10	0.6	0.3	40	22.05	30	120	1000
50	3	10	0.6	0.3	50	15.45	30	150	1000
60	3	10	0.6	0.3	60	16.77	30	180	1000
70	3	10	0.6	0.3	70	404.97	30	210	1000
80	3	10	0.6	0.3	80	691.48	30	240	1000
90	3	10	0.6	0.3	90	1171.38	30	270	1000
100	3	10	0.6	0.3	100	2013.85	30	300	1000
110	3	10	0.6	0.3	110	2121.68	30	330	1000
20	2	10	0.6	0.5	20	7.84	30	100	1000
30	2	10	0.6	0.5	30	12.13	30	150	1000
40	2	10	0.6	0.5	40	18.55	30	200	1000
50	2	10	0.6	0.5	50	22.75	30	250	1000
60	2	10	0.6	0.5	60	27.76	30	300	1000
70	2	10	0.6	0.5	70	232.07	30	350	1000
80	2	10	0.6	0.5	80	229.28	30	400	1000
90	2	10	0.6	0.5	90	435.13	30	450	992
100	2	10	0.6	0.5	100	1206.64	30	500	1000
110	2	10	0.6	0.5	110	1396.21	30	550	998
20	3	10	0.6	0.5	20	11.42	30	100	1000
30	3	10	0.6	0.5	30	17.66	30	150	1000
40	3	10	0.6	0.5	40	23.28	30	200	1000
50	3	10	0.6	0.5	50	15.8	30	250	1000
60	3	10	0.6	0.5	60	18.84	30	300	1000
70	3	10	0.6	0.5	70	339.84	30	350	1000
80	3	10	0.6	0.5	80	598.14	30	400	1000
90	3	10	0.6	0.5	90	843.54	30	450	1000
100	3	10	0.6	0.5	100	1277.76	30	500	1000
110	3	10	0.6	0.5	110	2218.18	30	550	1000
20	2	10	0.6	0.7	20	7.24	30	140	1000
30	2	10	0.6	0.7	30	12.34	30	210	1000
40	2	10	0.6	0.7	40	17.46	30	280	1000
50	2	10	0.6	0.7	50	22.11	30	350	1000
60	2	10	0.6	0.7	60	108.24	30	420	1000
70	2	10	0.6	0.7	70	151.42	30	490	1000
80	2	10	0.6	0.7	80	426.21	30	560	1000
90	2	10	0.6	0.7	90	266.03	30	630	1000
100	2	10	0.6	0.7	100	1000.55	30	700	1000
110	2	10	0.6	0.7	110	2189.05	30	770	1000
20	3	10	0.6	0.7	20	10.68	30	140	1000
30	3	10	0.6	0.7	30	17.83	30	210	1000
40	3	10	0.6	0.7	40	24.86	30	280	1000
50	3	10	0.6	0.7	50	15.59	30	350	1000
60	3	10	0.6	0.7	60	166.02	30	420	1000
70	3	10	0.6	0.7	70	230.55	30	490	1000
80	3	10	0.6	0.7	80	215.43	30	560	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.6	0.7	90	551.92	30	630	1000
100	3	10	0.6	0.7	100	708.58	30	700	1000
110	3	10	0.6	0.7	110	874.35	30	770	1000
20	2	10	0.6	0.9	20	7.31	30	180	1000
30	2	10	0.6	0.9	30	12.94	30	270	1000
40	2	10	0.6	0.9	40	18.3	30	360	1000
50	2	10	0.6	0.9	50	23.37	30	450	1000
60	2	10	0.6	0.9	60	111.34	30	540	1000
70	2	10	0.6	0.9	70	335.05	30	630	1000
80	2	10	0.6	0.9	80	239.38	30	720	1000
90	2	10	0.6	0.9	90	587.46	30	810	1000
100	2	10	0.6	0.9	100	903	30	900	1000
110	2	10	0.6	0.9	110	9795.83	30	990	1000
20	3	10	0.6	0.9	20	12.1	30	180	1000
30	3	10	0.6	0.9	30	18.48	30	270	1000
40	3	10	0.6	0.9	40	25.94	30	360	1000
50	3	10	0.6	0.9	50	19.84	30	450	1000
60	3	10	0.6	0.9	60	139.73	30	540	1000
70	3	10	0.6	0.9	70	313.88	30	630	1000
80	3	10	0.6	0.9	80	170.96	30	720	1000
90	3	10	0.6	0.9	90	316.66	30	810	1000
100	3	10	0.6	0.9	100	742.71	30	900	1000
110	3	10	0.6	0.9	110	1262.49	30	990	1000
20	2	10	0.8	0.1	20	7.67	40	20	1000
30	2	10	0.8	0.1	30	12.84	40	30	1000
40	2	10	0.8	0.1	40	18.45	40	40	1000
50	2	10	0.8	0.1	50	21.07	40	50	1000
60	2	10	0.8	0.1	60	26.66	40	60	1000
70	2	10	0.8	0.1	70	16.14	40	70	1000
80	2	10	0.8	0.1	80	24.38	40	80	999
90	2	10	0.8	0.1	90	1178.05	40	90	1000
100	2	10	0.8	0.1	100	1425.71	40	100	1000
110	2	10	0.8	0.1	110	844.05	40	110	1000
20	3	10	0.8	0.1	20	11.88	40	20	1000
30	3	10	0.8	0.1	30	19.05	40	30	1000
40	3	10	0.8	0.1	40	25.66	40	40	1000
50	3	10	0.8	0.1	50	15.07	40	50	1000
60	3	10	0.8	0.1	60	19.41	40	60	1000
70	3	10	0.8	0.1	70	24.55	40	70	1000
80	3	10	0.8	0.1	80	658.16	40	80	1000
90	3	10	0.8	0.1	90	394.38	40	90	999
100	3	10	0.8	0.1	100	2515.95	40	100	1000
110	3	10	0.8	0.1	110	2326.63	40	110	1000
20	2	10	0.8	0.3	20	7.97	40	60	1000
30	2	10	0.8	0.3	30	13	40	90	1000
40	2	10	0.8	0.3	40	18.27	40	120	1000
50	2	10	0.8	0.3	50	24.91	40	150	1000
60	2	10	0.8	0.3	60	26.32	40	180	1000
70	2	10	0.8	0.3	70	18.51	40	210	1000
80	2	10	0.8	0.3	80	295.89	40	240	1000
90	2	10	0.8	0.3	90	426.17	40	270	979
100	2	10	0.8	0.3	100	534.39	40	300	1000
110	2	10	0.8	0.3	110	1241.23	40	330	1000
20	3	10	0.8	0.3	20	12.42	40	60	1000
30	3	10	0.8	0.3	30	17.8	40	90	1000
40	3	10	0.8	0.3	40	23.35	40	120	1000
50	3	10	0.8	0.3	50	13.52	40	150	1000
60	3	10	0.8	0.3	60	17.3	40	180	1000
70	3	10	0.8	0.3	70	329.8	40	210	1000
80	3	10	0.8	0.3	80	431.38	40	240	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.8	0.3	90	1360.69	40	270	1000
100	3	10	0.8	0.3	100	2512.76	40	300	1000
110	3	10	0.8	0.3	110	1906.34	40	330	1000
20	2	10	0.8	0.5	20	6.84	40	100	1000
30	2	10	0.8	0.5	30	12	40	150	1000
40	2	10	0.8	0.5	40	16.81	40	200	1000
50	2	10	0.8	0.5	50	22.88	40	250	1000
60	2	10	0.8	0.5	60	29.52	40	300	1000
70	2	10	0.8	0.5	70	182.67	40	350	1000
80	2	10	0.8	0.5	80	150.88	40	400	1000
90	2	10	0.8	0.5	90	481.88	40	450	1000
100	2	10	0.8	0.5	100	554.42	40	500	999
110	2	10	0.8	0.5	110	816.17	40	550	1000
20	3	10	0.8	0.5	20	11.34	40	100	1000
30	3	10	0.8	0.5	30	16.95	40	150	1000
40	3	10	0.8	0.5	40	26.19	40	200	1000
50	3	10	0.8	0.5	50	15.74	40	250	1000
60	3	10	0.8	0.5	60	22.21	40	300	1000
70	3	10	0.8	0.5	70	370.21	40	350	1000
80	3	10	0.8	0.5	80	589.13	40	400	1000
90	3	10	0.8	0.5	90	842.23	40	450	1000
100	3	10	0.8	0.5	100	1225.36	40	500	997
110	3	10	0.8	0.5	110	1308.59	40	550	1000
20	2	10	0.8	0.7	20	8.45	40	140	1000
30	2	10	0.8	0.7	30	12	40	210	1000
40	2	10	0.8	0.7	40	18.56	40	280	1000
50	2	10	0.8	0.7	50	22.73	40	350	1000
60	2	10	0.8	0.7	60	111.2	40	420	1000
70	2	10	0.8	0.7	70	162.23	40	490	1000
80	2	10	0.8	0.7	80	303.29	40	560	1000
90	2	10	0.8	0.7	90	1353.99	40	630	1000
100	2	10	0.8	0.7	100	419.8	40	700	1000
110	2	10	0.8	0.7	110	710.25	40	770	1000
20	3	10	0.8	0.7	20	9.95	40	140	1000
30	3	10	0.8	0.7	30	17.33	40	210	1000
40	3	10	0.8	0.7	40	21.75	40	280	1000
50	3	10	0.8	0.7	50	17.58	40	350	1000
60	3	10	0.8	0.7	60	158.92	40	420	1000
70	3	10	0.8	0.7	70	298.1	40	490	1000
80	3	10	0.8	0.7	80	381.21	40	560	1000
90	3	10	0.8	0.7	90	433.91	40	630	1000
100	3	10	0.8	0.7	100	907.59	40	700	1000
110	3	10	0.8	0.7	110	402.17	40	770	1000
20	2	10	0.8	0.9	20	8.33	40	180	1000
30	2	10	0.8	0.9	30	12.74	40	270	1000
40	2	10	0.8	0.9	40	18.17	40	360	1000
50	2	10	0.8	0.9	50	21.68	40	450	1000
60	2	10	0.8	0.9	60	111.38	40	540	1000
70	2	10	0.8	0.9	70	152.63	40	630	1000
80	2	10	0.8	0.9	80	345.72	40	720	1000
90	2	10	0.8	0.9	90	709.37	40	810	1000
100	2	10	0.8	0.9	100	1083.22	40	900	1000
110	2	10	0.8	0.9	110	14738.55	40	990	1000
20	3	10	0.8	0.9	20	9.91	40	180	1000
30	3	10	0.8	0.9	30	18.12	40	270	1000
40	3	10	0.8	0.9	40	24.64	40	360	1000
50	3	10	0.8	0.9	50	14.59	40	450	1000
60	3	10	0.8	0.9	60	111.91	40	540	1000
70	3	10	0.8	0.9	70	189.39	40	630	1000
80	3	10	0.8	0.9	80	432.42	40	720	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	0.8	0.9	90	872.53	40	810	1000
100	3	10	0.8	0.9	100	605.9	40	900	1000
110	3	10	0.8	0.9	110	437.88	40	990	1000
20	2	10	1	0.1	20	7.14	50	20	1000
30	2	10	1	0.1	30	12.32	50	30	1000
40	2	10	1	0.1	40	17.38	50	40	1000
50	2	10	1	0.1	50	24.43	50	50	1000
60	2	10	1	0.1	60	28.53	50	60	1000
70	2	10	1	0.1	70	16.6	50	70	1000
80	2	10	1	0.1	80	198.74	50	80	1000
90	2	10	1	0.1	90	500.53	50	90	1000
100	2	10	1	0.1	100	1545.18	50	100	1000
110	2	10	1	0.1	110	997.13	50	110	1000
20	3	10	1	0.1	20	10.32	50	20	1000
30	3	10	1	0.1	30	17.91	50	30	1000
40	3	10	1	0.1	40	25.66	50	40	1000
50	3	10	1	0.1	50	15.9	50	50	1000
60	3	10	1	0.1	60	18.05	50	60	1000
70	3	10	1	0.1	70	20.16	50	70	1000
80	3	10	1	0.1	80	367.57	50	80	1000
90	3	10	1	0.1	90	1321.18	50	90	1000
100	3	10	1	0.1	100	1224.99	50	100	999
110	3	10	1	0.1	110	1575.34	50	110	1000
20	2	10	1	0.3	20	8.15	50	60	1000
30	2	10	1	0.3	30	12.84	50	90	1000
40	2	10	1	0.3	40	12.23	50	120	1000
50	2	10	1	0.3	50	22.55	50	150	1000
60	2	10	1	0.3	60	28.78	50	180	1000
70	2	10	1	0.3	70	13.65	50	210	1000
80	2	10	1	0.3	80	193.22	50	240	998
90	2	10	1	0.3	90	461.77	50	270	1000
100	2	10	1	0.3	100	1333.13	50	300	1000
110	2	10	1	0.3	110	975.13	50	330	1000
20	3	10	1	0.3	20	10.71	50	60	1000
30	3	10	1	0.3	30	16.57	50	90	1000
40	3	10	1	0.3	40	23.34	50	120	1000
50	3	10	1	0.3	50	15.62	50	150	1000
60	3	10	1	0.3	60	18.05	50	180	1000
70	3	10	1	0.3	70	769.69	50	210	1000
80	3	10	1	0.3	80	480.18	50	240	1000
90	3	10	1	0.3	90	1187.59	50	270	1000
100	3	10	1	0.3	100	2146.21	50	300	1000
110	3	10	1	0.3	110	1918.96	50	330	1000
20	2	10	1	0.5	20	7.29	50	100	1000
30	2	10	1	0.5	30	12.22	50	150	1000
40	2	10	1	0.5	40	11.53	50	200	1000
50	2	10	1	0.5	50	22.34	50	250	1000
60	2	10	1	0.5	60	27.68	50	300	1000
70	2	10	1	0.5	70	207	50	350	1000
80	2	10	1	0.5	80	273.07	50	400	1000
90	2	10	1	0.5	90	296.38	50	450	989
100	2	10	1	0.5	100	399.88	50	500	1000
110	2	10	1	0.5	110	1486.73	50	550	1000
20	3	10	1	0.5	20	10.67	50	100	1000
30	3	10	1	0.5	30	16.42	50	150	1000
40	3	10	1	0.5	40	23.24	50	200	1000
50	3	10	1	0.5	50	19.71	50	250	1000
60	3	10	1	0.5	60	22.18	50	300	1000
70	3	10	1	0.5	70	389.29	50	350	1000
80	3	10	1	0.5	80	518.56	50	400	1000

Continued on next page

Table C1 – continued

Tasks	Machines	Max delay	β	α	Objective function	Solve Time (s)	Considered scenarios ($\sum_s b_s$)	Total machine delay	Feasible
90	3	10	1	0.5	90	914.88	50	450	1000
100	3	10	1	0.5	100	843.71	50	500	1000
110	3	10	1	0.5	110	2310.65	50	550	1000
20	2	10	1	0.7	20	8.46	50	140	1000
30	2	10	1	0.7	30	11.89	50	210	1000
40	2	10	1	0.7	40	17.94	50	280	1000
50	2	10	1	0.7	50	22.46	50	350	1000
60	2	10	1	0.7	60	108.71	50	420	1000
70	2	10	1	0.7	70	140.11	50	490	1000
80	2	10	1	0.7	80	506.07	50	560	1000
90	2	10	1	0.7	90	185.69	50	630	1000
100	2	10	1	0.7	100	395.68	50	700	1000
110	2	10	1	0.7	110	613.38	50	770	1000
20	3	10	1	0.7	20	10.61	50	140	1000
30	3	10	1	0.7	30	17.59	50	210	1000
40	3	10	1	0.7	40	25.43	50	280	1000
50	3	10	1	0.7	50	15.83	50	350	1000
60	3	10	1	0.7	60	118.53	50	420	1000
70	3	10	1	0.7	70	194.52	50	490	1000
80	3	10	1	0.7	80	293.13	50	560	1000
90	3	10	1	0.7	90	516.82	50	630	1000
100	3	10	1	0.7	100	585.2	50	700	1000
110	3	10	1	0.7	110	1130.01	50	770	1000
20	2	10	1	0.9	20	7.27	50	180	1000
30	2	10	1	0.9	30	12.09	50	270	1000
40	2	10	1	0.9	40	15.54	50	360	1000
50	2	10	1	0.9	50	22.05	50	450	1000
60	2	10	1	0.9	60	115.82	50	540	1000
70	2	10	1	0.9	70	104.69	50	630	1000
80	2	10	1	0.9	80	246.88	50	720	1000
90	2	10	1	0.9	90	244.85	50	810	1000
100	2	10	1	0.9	100	605.4	50	900	1000
110	2	10	1	0.9	110	22978.32	50	990	1000
20	3	10	1	0.9	20	11.64	50	180	1000
30	3	10	1	0.9	30	18.91	50	270	1000
40	3	10	1	0.9	40	23.61	50	360	1000
50	3	10	1	0.9	50	19.15	50	450	1000
60	3	10	1	0.9	60	78.13	50	540	1000
70	3	10	1	0.9	70	153.49	50	630	1000
80	3	10	1	0.9	80	301.24	50	720	1000
90	3	10	1	0.9	90	322.09	50	810	1000
100	3	10	1	0.9	100	489.44	50	900	1000
110	3	10	1	0.9	110	825.96	50	990	1000

References

Younes Aalian, Gilles Pesant, and Michel Gamache. Optimization of Short-Term Underground Mine Planning Using Constraint Programming. In Roland H. C. Yap, editor, 29th International Conference on Principles and Practice of Constraint Programming (CP 2023), volume 280 of Leibniz International Proceedings in Informatics (LIPIcs), pages 6:1–6:16, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.CP.2023.6. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CP.2023.6.

Younes Aalian, Michel Gamache, and Gilles Pesant. Short-term underground mine planning with uncertain activity durations using constraint programming. Journal of Scheduling, pages 1–17, 04 2024. doi: 10.1007/s10951-024-00808-x.

Aref Alipour, Ali Asghar Khodaiari, Ahmad Jafari, and Reza Tavakkoli-Moghaddam. Uncertain production scheduling optimization in open-pit mines and its ellipsoidal robust counterpart. International Journal of Management Science and Engineering Management, pages 1–9, January 2018. doi: 10.1080/17509653.2017.1410454. URL https://www.tandfonline.com/doi/full/10.1080/17509653.2017.1410454.

Somayeh Danesh Asgari, Emran Mohammadi, Ahmad Makui, and Mostafa Jafari. Data-driven robust optimization based on position-regulated support vector clustering. Journal of Computational Science, 76, 2024. URL http://dx.doi.org/10.1016/j.jocs.2024.102210.

Waqar Muhammad Ashraf and Vivek Dua. Data-driven robust optimization for the operation optimization of industrial power station. In 2024 7th International Conference on Energy Conservation and Efficiency, ICECE 2024 - Proceedings, Lahore, Pakistan, 2024. URL http://dx.doi.org/10.1109/ICECE61222.2024.10505267.

Jörg Benndorf and Can Yueksel. A hybrid approach for optimizing continuous mining systems. European Journal of Operational Research, 241(3):770–778, 2015.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. Operations Research, 52(1):35–53, 2004. URL http://www.jstor.org/stable/30036559.

John R. Birge and François Louveaux. Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering. Springer New York, New York, NY, 2011. doi: 10.1007/978-1-4614-0237-4. URL https://link.springer.com/10.1007/978-1-4614-0237-4.

Renan Brito Cano Butkeraites, Luiz Leduino De Salles Neto, and Michel Gendreau. A sampling-based multi-objective iterative robust optimization method for Bandwidth Packing Problem. Expert Systems with Applications, 203:117337, October 2022. doi: 10.1016/j.eswa.2022.117337. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417422006947.

Louis-Pierre Campeau and Michel Gamache. Short- and medium-term optimization of underground mine planning using constraint programming. Constraints, 27(4):414 – 431, 2022. URL http://dx.doi.org/10.1007/s10601-022-09337-w.

A. Charnes and W. W. Cooper. Chance-Constrained Programming. Management Science, 6(1):73–79, October 1959. doi: 10.1287/mnsc.6.1.73. URL https://pubsonline.informs.org/doi/10.1287/mnsc.6.1.73.

Lu Chen, Michel Gendreau, Minh Hoàng Hà, and André Langevin. A robust optimization approach for the road network daily maintenance routing problem with uncertain service time. Transportation Research Part E: Logistics and Transportation Review, 85:40–51, 2016. doi: https://doi.org/10.1016/j.tre.2015.11.006. URL https://www.sciencedirect.com/science/article/pii/S1366554515002094.

Chun Cheng, Mingyao Qi, Ying Zhang, and Louis-Martin Rousseau. A two-stage robust approach for the reliable logistics network design problem. Transportation Research Part B: Methodological, 111:185–202, May 2018. doi: 10.1016/j.trb.2018.03.015. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261517304241.

Chun Cheng, Yossiri Adulyasak, and Louis-Martin Rousseau. Robust facility location under demand uncertainty and facility disruptions. Omega, 103:102429, 2021. doi: https://doi.org/10.1016/j.omega.2021.102429. URL https://www.sciencedirect.com/science/article/pii/S0305048321000384.

Abid Ali Khan Danish, Asif Khan, and Khan Muhammad. A simulated annealing based stochastic long-term production scheduling of open-pit mines with stockpiling under grade uncertainty. International Journal of Mining, Reclamation and Environment, 37(1):43–65, 2023. doi: 10.1080/17480930.2022.2140543.

George B. Dantzig. Linear Programming under Uncertainty. Management Science, 1(3-4):197–206, April 1955. doi: 10.1287/mnsc.1.3-4.197. URL https://pubsonline.informs.org/doi/10.1287/mnsc.1.3-4.197.

M. Fernanda Del Castillo and Roussos Dimitrakopoulos. Dynamically optimizing the strategic plan of mining complexes under supply uncertainty. Resources Policy, 60:83–93, 2019. doi: 10.1016/j.resourpol.2018.11.019.

Matheus Furtado E Faria, Roussos Dimitrakopoulos, and Cláudio Lúcio Lopes Pinto. Integrated stochastic optimization of stope design and long-term underground mine production scheduling. Resources Policy, 78:102918, 2022. doi: 10.1016/j.resourpol.2022.102918.

Javad Gholamnejad, Ali Azimi, Reza Lotfian, Sara Kasmaeeyazdi, and Francesco Tinti. The application of a stockpile stochastic model into long-term open pit mine production scheduling to improve the feed grade for the processing plant. Rudarsko-geološko-naftni zbornik, 35(4):115–129, 2020. doi: 10.17794/rgn.2020.4.10.

B. Golamnejad, Osanloo, and Mojtaba Karimi. A chance-constrained programming approach for open pit long-term production scheduling in stochastic environments. Journal of The South African Institute of Mining and Metallurgy, 106:105–114, 2006.

Ryan C. Goodfellow and Roussos Dimitrakopoulos. Global optimization of open pit mining complexes with uncertainty. Applied Soft Computing, 40:292–304, 2016a. doi: 10.1016/j.asoc.2015.11.038.

Ryan C Goodfellow and Roussos Dimitrakopoulos. Simultaneous stochastic optimization of mining complexes and mineral value chains. Mathematical Geosciences, 48(6):619–639, 2016b.

LLC Gurobi Optimization. State of mathematical optimization 2023. Technical report, Gurobi Optimization, LLC, 2023. URL https://p1.aprimocdn.net/gurobi/fbd5f05a-59db-4d41-8672-b058013ec217/originalfile/Report---State-of-Mathematical-Optimization-2023.pdf.

Hossein Hashemi Doulabi, Patrick Jaillet, Gilles Pesant, and Louis-Martin Rousseau. Exploiting the Structure of Two-Stage Robust Optimization Models with Exponential Scenarios. INFORMS Journal on Computing, 33(1):143–162, January 2021. doi: 10.1287/ijoc.2019.0928. URL https://pubsonline.informs.org/doi/10.1287/ijoc.2019.0928.

Liqian Huang, Xuechun Zheng, Pengyu Wang, Wenkai Wang, Yaowei Zhang, Renjie Li, Qipeng Wang, and Liang Zhao. Data-driven robust optimization for industrial utility system integrating wind and solar energies. In Proceedings - 2023 China Automation Congress, CAC 2023, pages 8947 – 8951, Chongqing, China, 2023. URL http://dx.doi.org/10.1109/CAC59555.2023.10450545.

Alireza Khalili-Fard, Fatemeh Sabouhi, and Ali Bozorgi-Amiri. Data-driven robust optimization for a sustainable steel supply chain network design: Toward the circular economy. Computers and Industrial Engineering, 195, 2024. URL http://dx.doi.org/10.1016/j.cie.2024.110408.

A. Khan. Long-term production scheduling of open pit mines using particle swarm and bat algorithms under grade uncertainty. Journal of the Southern African Institute of Mining and Metallurgy, 118 (4):361–368, 2018. doi: 10.17159/2411-9717/2018/v118n4a5.

Rahul Kumar, Pranjal Pathak, and Biswajit Samanta. Open pit mine production scheduling with stockpiling using constraint programming. In APCOM 2023 Proceedings: Intelligent Mining: Innovation, Vision, and Value, 2023.

Mustafa Kumral and Yuksel Asli Sari. Simulation-based mine extraction sequencing with chance constrained risk tolerance. Simulation, 93(6):527–539, 2017. doi: 10.1177/0037549717692415.

Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Petr Vilím. IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. Constraints, 23(2):210–250, 2018. doi: 10.1007/s10601-018-9281-x.

Amina Lamghari and Roussos Dimitrakopoulos. Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty. European Journal of Operational Research, 250(1):273–290, 2016. doi: 10.1016/j.ejor.2015.08.051.

Mohamed Maftah, Michel Gamache, Bruno Agard, and Michel Gendreau. Coordination of electrical drilling machines in open-pit mines: A constraint programming approach. Les Cahiers du GERAD G-2024-62, Groupe d'études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada, September 2024. URL https://www.gerad.ca/en/papers/G-2024-62.

Ismael Martinez, Abdelhakim Senhaji Hafid, and Michel Gendreau. Robust and fault-tolerant fog design and dimensioning for reliable operation. IEEE Internet of Things Journal, 9(19):18280–18292, Oct 2022. doi: 10.1109/JIOT.2022.3157557.

Alexandre Mercier-Aubin, Ludwig Dumetz, Jonathan Gaudreault, and Claude-Guy Quimper. The Confidence Constraint: A Step Towards Stochastic CP Solvers. In Helmut Simonis, editor, Principles and Practice of Constraint Programming, volume 12333, pages 759–773. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-58475-7_44. URL https://link.springer.com/10.1007/978-3-030-58475-7_44.

Mehrnaz Mohtasham, Hossein Mirzaei-Nasirabad, and Behrooz Alizadeh. Optimization of truck-shovel allocation in open-pit mines under uncertainty: a chance-constrained goal programming approach. Mining Technology, 130(2):81–100, 2021. doi: 10.1080/25726668.2021.1916170.

Arkadi Nemirovski and Alexander Shapiro. Scenario Approximations of Chance Constraints. In Giuseppe Calafiore and Fabrizio Dabbene, editors, Probabilistic and Randomized Methods for Design under Uncertainty, pages 3–47. Springer-Verlag, London, 2006. doi: 10.1007/1-84628-095-8_1. URL http://link.springer.com/10.1007/1-84628-095-8_1.

Ramin Nourollahi, Vahid Sohrabi Tabar, Saeid Ghassem Zadeh, and Alireza Akbari-Dibavar. A hybrid optimization approach to analyze the risk-constrained operation of a residential hybrid energy system incorporating responsive loads. Computers & Chemical Engineering, 157:107603, January 2022. doi: 10.1016/j.compchemeng.2021.107603. URL https://linkinghub.elsevier.com/retrieve/pii/S0098135421003811.

Yurii A. Oleynik and Alexander A. Zuenko. Open pit mine production scheduling using new global constraint. In 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences, SIBIRCON 2022, pages 1700 – 1704, 2022. URL http://dx.doi.org/10.1109/SIBIRCON56155.2022.10016915.

Salih Ramazan and Roussos Dimitrakopoulos. Production scheduling with uncertain supply: a new solution to the open pit mining problem. Optimization and Engineering, 14(2):361–380, 2013. doi: 10.1007/s11081-012-9186-2.

Saeed Sadeghi Darvazeh, Farzaneh Mansoori Mooseloo, Fatemeh Gholian-Jouybari, Maghsoud Amiri, Hossein Bonakdari, and Mostafa Hajiaghaei-Keshteli. Data-driven robust optimization to design an integrated sustainable forest biomass-to-electricity network under disjunctive uncertainties. Applied Energy, 356, 2024. URL http://dx.doi.org/10.1016/j.apenergy.2023.122404.

Max Strand, Mikael Johansson, and Alessandro Zanarini. Underground mine scheduling of mobile machines using constraint programming and large neighborhood search. Computers and Operations Research, 123, 2020. URL http://dx.doi.org/10.1016/j.cor.2020.105036.

S. Armagan Tarim, Suresh Manandhar, and Toby Walsh. Stochastic Constraint Programming: A Scenario-Based Approach. Constraints, 11(1):53–80, January 2006. doi: 10.1007/s10601-006-6849-7. URL http://link.springer.com/10.1007/s10601-006-6849-7.

Jorge Luiz Valenca Mariz, Rodrigo de Lemos Peroni, Ricardo Martins de Abreu Silva, Mohammad Mahdi Badiozamani, and Hooman Askari-Nasab. A multi-stage constraint programming approach to solve clustering problems in open-pit mine planning. Engineering Optimization, 2024. URL http://dx.doi.org/10.1080/0305215X.2024.2346935.

Toby Walsh. Stochastic constraint programming. In Proceedings of the 15th European Conference on Artificial Intelligence (ECAI), volume 2, pages 111–115, January 2002.

Yuhong Wang and Jian Su. A data-driven robust optimization method based on scenario clustering for pvc production scheduling under uncertainty. Computers and Chemical Engineering, 188, 2024. URL http://dx.doi.org/10.1016/j.compchemeng.2024.108782.

Rujing Yan, Jiangjiang Wang, Jiahao Wang, Lei Tian, Saiqiu Tang, Yuwei Wang, Jing Zhang, Youliang Cheng, and Yuan Li. A two-stage stochastic-robust optimization for a hybrid renewable energy CCHP system considering multiple scenario-interval uncertainties. Energy, 247:123498, May 2022. doi: 10.1016/j.energy.2022.123498. URL https://linkinghub.elsevier.com/retrieve/pii/S0360544222004017.