Optimization of Short-Term Underground Mine Planning Using Constraint Programming

Younes Aalian
Département de mathématiques et de génie industriel, Polytechnique Montréal, Québec, Canada

Gilles Pesant
Département de génie informatique et génie logiciel, Polytechnique Montréal, Québec, Canada

Michel Gamache
Département de mathématiques et de génie industriel, Polytechnique Montréal, Québec, Canada

Abstract

Short-term underground mine planning problems are often difficult to solve due to the large number of activities and diverse machine types to be scheduled, as well as multiple operational constraints. This paper presents a Constraint Programming (CP) model to optimize short-term scheduling for the Meliadine underground gold mine in Nunavut, Canada, taking into consideration operational constraints and the daily development and production targets of the mine plan. To evaluate the efficacy of the developed CP short-term planning model, we compare schedules generated by the CP model with the ones created manually by the mine planner for two real data sets. Results demonstrate that the CP model outperforms the manual approach by generating more efficient schedules with lower makespans.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming; Computing methodologies → Planning and scheduling

Keywords and phrases Mine planning, Constraint Programming, Short-term planning, Underground mine, Scheduling

Digital Object Identifier 10.4230/LIPIcs.CP.2023.6

1 Introduction

The mining industry is an important component of Canada's economic vitality. In 2019, its economic contribution was estimated at $ 109 billion, or 5 % of Canada's GDP [10]. Mining projects involve a variety of operations that handle significant amounts of material and require substantial investment. Even small reductions in costs or increases in ore yield can have a considerable economic impact. These projects can generate significant profits when they are managed efficiently. Furthermore, the mining industry is evolving and transitioning towards automated mining. With the advent of new communication and data collection tools, mining operation data is becoming more easily accessible. This creates opportunities to develop new optimization tools that can use the available data to enhance the operational efficiency in mines.

The model presented in this study is designed for an underground gold mine. The price of gold is set by the market and the same for all mining companies. Among other things, 47 % of the gold produced in Canada is purchased by the London Bullion Market, which trades gold worldwide. The only options for gold mines to increase their profits is to reduce their operating costs. One way to reduce operating costs is to make better use of available resources. Minimizing the makespan indirectly reduces operating costs by doing more activities with the same equipment and reducing downtime.

Short-term planning in underground mines plays a crucial role in ensuring the profitability and success of mining operations. It involves allocating resources to activities and determining the sequence and start time of activities during each work shift over a planning horizon ranging from one to two weeks [1, 3]. Currently, scheduling decisions in underground mines are typically made manually based on the planner's experience. Planning has been done manually for several reasons. First, communication systems in the underground mines were virtually non-existent. As a result, the exchange of information between the planning teams was essentially done between shifts. In addition, the management systems are not yet standardized in the mines, which means that information on geology, equipment maintenance and production management are found in different systems and the transfer of one system to another is not always trivial. However, manual planning is prone to errors and often results in infeasible schedules with low accuracy and efficiency. Therefore, developing a decision tool to optimize short-term scheduling in underground mines can help achieve high-quality schedules, improve mine productivity, and reduce reliance on the planner's experience, while ensuring technical and safety requirements are met [17]. In this paper, a Constraint Programming (CP) model is presented for the short-term scheduling of activities at the Meliadine underground gold mine located in Nunavut, Canada. The model considers both operational constraints and the mine's development and production targets to generate more practical and reliable schedules.

1.1 Why CP?

Previous research has shown that Constraint Programming is an effective and efficient method for solving scheduling problems across various industries, including planning, scheduling, transportation, and automated systems [9]. CP uses a wide variety of variable types, functions, and global constraints to offer modeling at a high level of abstraction, making it a more flexible and intuitive approach than other model-based methods such as Mixed Integer Programming (MIP)[4]. Consequently CP models are more concise and require fewer decision variables and constraints which makes them an attractive tool for addressing large-scale scheduling problems. In the context of underground mining, the use of CP functions (as described in detail in Section 3) makes it easier to model operational constraints in the short-term scheduling problem, resulting in a more compact and efficient model.

1.2 Plan of the Paper

Section 2 describes the problem we address, Section 3 introduces the CP model we developed to solve it, and Section 4 discusses the outcomes of implementing the presented model on two actual data sets. Section 5 highlights the advantages of using CP for this short-term underground mine planning problem. Section 6 presents an overview of related computational approaches in the literature. Finally Section 7 concludes the paper.

2 Problem Description

Underground mining operations involve two primary categories of activities: development and production. In order to access economically valuable ore deposits, development activities are conducted in waste rocks that lack financial value. Production activities take place in economically significant rocks located in areas referred to as *stopes* [5]. Mining activities occur in a cycle at one of several sites that serve as a workplace to perform these activities. Figures 1a and 1b show the development and production cycles with activities arranged in a sequence-dependent order. Table 1 provides the description of activities in the cycle, along with the required machine type. There are several machines available for each type of activity.

Each machine can be viewed as a renewable unitary resource, limited to performing one activity at a time. Short-term scheduling for underground mines includes assigning activities in the cycle to eligible machines and determining the start and end times for each activity [1].

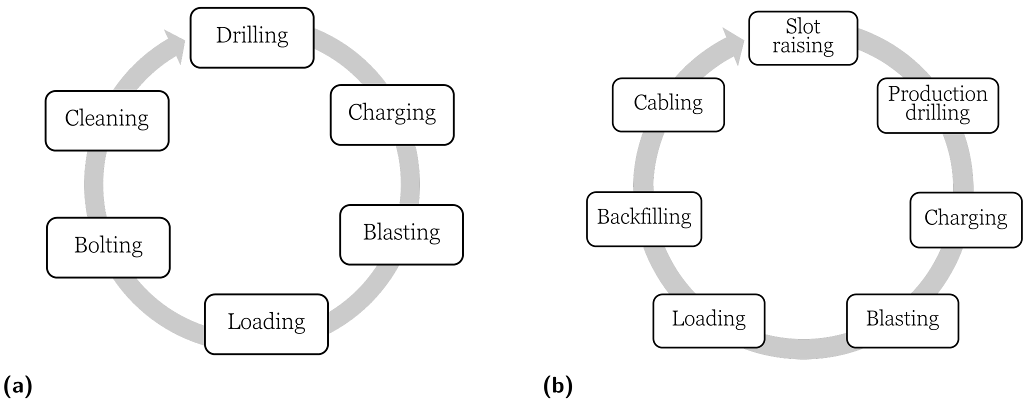


Figure 1 Typical development (a) and production (b) cycles in underground mining.

Table 1 Activities and the required machine type in the cycle.

Activity	Machine	Description
Drilling	Drilling rigs	Drilling blast holes in the rock face
Charging	Anfo loader	Charging drilled holes with explosives
Loading	Scooptram	Removing broken rocks after blasting
Bolting	Bolter	Stabilizing drifts by installing bolts into the rock mass
Cleaning	Scooptram	Removing small rocks from the site (the gallery)
Cabling	Cabling machine	Reinforcing stope by installing steel cables into rock mass
Slot raising	Raise borer	Creating a vertical or inclined hole into the rock

There are several underground mining methods for extracting deep mineral deposits. The Meliadine mine uses the long-hole stoping method, which is one of the most commonly-used underground mining techniques that involve extracting a significant amount of material from each stope (Figure 2). This method is particularly suitable for large-scale and steeply dipping ore deposits with preferably tabular shapes. The long-hole stoping method begins with the development of main shafts or declines for transportation and ventilation purposes. Next, drill drives are excavated to access the intended location of the ore body and to create stopes. In each stope, production holes are drilled and charged with explosives. Once the blasting is completed, the fragmented rock is accessed through draw points developed at the bottom level of the stope. Scoop trams and trucks are used to collect the broken ore and transport it to the surface or other underground locations via drifts or ramps. In the final stage, the evacuated space in the stope is filled with a mixture of waste rocks and concrete to provide sufficient stability for the subsequent adjacent opening stopes [16].

At Meliadine work is organized into a succession of day and night shifts, each lasting 55 time units. Blasting activities are performed only during designated blast windows. A blast window corresponds to the period between shifts in the morning, during which resources cannot be used by operators due to safety regulations. The team rotation takes approximately 1 hour and 30 minutes, and the blast window requires roughly 4 hours and 30 minutes, including the time needed for team rotation and gas clearance (18 working hours for both

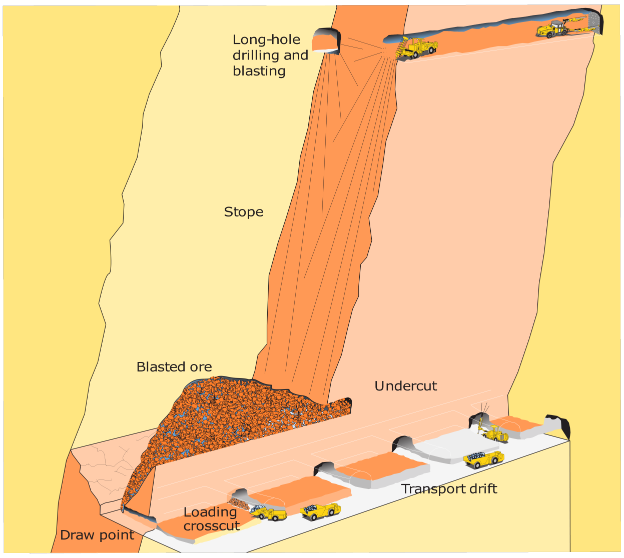


Figure 2 Typical representation of the long-hole stoping operation [8].

day and night shifts + 1 hour 30 minutes for team rotation at the end of day shift + 4 hours 30 minutes for blast window at the end of night shift = 24 hours). The shift at the end of which team rotation occurs is referred to as the day shift, while the subsequent shift, which includes the blasting window at the end, is known as the night shift. Figure 3 illustrates the shift organization in the studied underground mine. Mining activities are preemptive as they can be interrupted at the end of each shift and continue in the next shift.

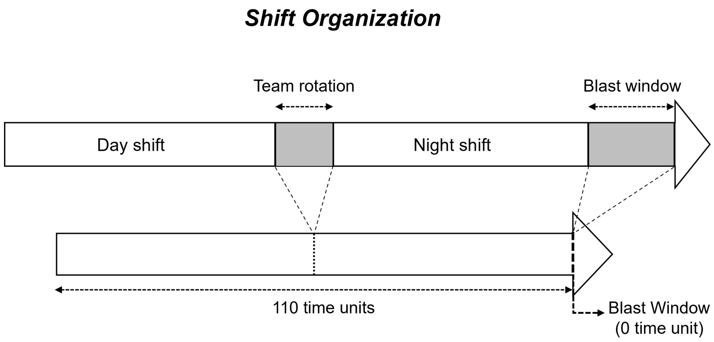


Figure 3 Timeline of alternating day and night shifts including time to rotate the teams and to perform blasting (above). Its representation in the CP model (below).

Short-term planning at the Meliadine mine incorporates several key performance indicators (KPI) such as progress of development rounds in the drift, total length of production holes drilled in the stope, and total amount of material mucked from the stope to meet the medium-term planning goals. The KPI values vary monthly and are updated every three months by the medium-term mine planner. Development and production constraints will be introduced in our CP model to consider the defined KPIs in short-term scheduling.

3 How our Problem is Modeled in CP

An optimization model is developed using Constraint Programming (CP) for short-term underground mine scheduling, taking into account operational requirements of underground mining operations. Additional constraints are introduced to ensure that the mine planning development and production targets are met and that practical and reliable short-term schedules are generated. In other words, the produced schedule determines the detailed execution of mining activities in underground operations considering the required daily rates of development and production. It is important to note that the same model can be used for both development and production activities in underground mining, which ensures consistency and accuracy in short-term scheduling.

CP Optimizer (CPO) from IBM ILOG Optimization Studio [9] was used to create the model presented in this article. In this CPO model, *interval variables* are used to represent activities, each with several related *optional interval variables* depicting the choice of resource. Optional interval variables include a Boolean status reflecting the fact that the corresponding activity is present or absent from the solution (i.e. not considered by the constraints). The ordering of resources can be represented by a set of interval variables, known as a *sequence variable*. This sequence variable is used in the scheduling model to prevent activities in the sequence from overlapping in time. More formally, an interval variable a is defined by a start time s and an end time e, which are non-negative integer values, such that $a \in \{[s, e) \mid s, e \in \mathbf{N}, e \geq s\}$. Optional interval variable b is presented such that $b \in \{\emptyset\} \cup \{[s, e) \mid s, e \in \mathbf{N}, e \geq s\}$. Additionally the developed CPO model uses various functions and constraints that are described as follows [9]:

- `endOf`: A function that provides the end value of an interval variable if it exists, or else returns zero.
- `alternative`: This constraint ensures that if a given interval variable is present, then only one related optional interval variable is chosen with the same start and end values.
- `noOverlap`: This constraint is used to ensure that a set of interval variables defined by a sequence variable do not overlap, while maintaining a minimum distance between them as specified by a transition distance matrix.
- `endBeforeStart`: This constraint guarantees that if two interval variables are present, then the first ends before the second starts, with an optional minimum delay between them.
- `forbidExtent`: This constraint makes sure that an interval variable cannot overlap with a forbidden region where the value of the step function is zero. As a result, the interval variable must either end before the forbidden region or start after it.
- `stepAtEnd`: This step function returns an elementary cumulative function with a non-negative integer value at the end of an interval variable. Such functions model a known function of time, such as the resources used during a particular time period, by returning a non-negative integer value (height of the elementary function) within the range of the interval variable and zero outside of it.
- `cumulFunction`: This expression models a known function of time, such as the cumulative amount of resources used by an activity during a specific time period.
- `alwaysIn`: This constraint restricts the potential values of a cumulative function to a specific range during a time interval.

Tables 2 and 3 present lists of sets, parameters, and variables used in the CP model, along with their corresponding descriptions.

Table 2 Sets and parameters of the CP model.

Set	**Description**
J	Index set of activities
M	Index set of all available equipment
M_j	Index set of eligible machines to perform activity j
A_j	Index set of activities that must occur after activity j
B	Index set of blast activities
T	Index set of time windows (starting at 1)
Parameter	**Description**
p_j	Processing time of activity j
D	Matrix of transition time between sites where the value of its element is equal to 0 for the same site and greater than 0 otherwise
d_j	Development (meter) of activity j
h_j	Production hole drilling (meter) of activity j
o_j	Stope ore mucking (ton) of activity j
s_t	Starting time of time window t
e_t	Ending time of time window t
$\underline{d}$	Lower bound for daily development
$\overline{d}$	Upper bound for daily development
$\underline{h}$	Lower bound for daily hole drilling
$\overline{h}$	Upper bound for daily hole drilling
$\underline{o}$	Lower bound for daily ore mucking
$\overline{o}$	Upper bound for daily ore mucking
$Blast\_calendar$	The time periods during which only blasting activities are permitted (all activities except blasting are forbidden to be performed during these periods

Table 3 Decision variables of the CP model.

Variable	**Description**
Y_j	Interval variable for activity j
X_{jm}	Optional interval variable to perform activity j using machine m
S_m	Sequence variable for machine m ($S_m = \{X_{jm} \mid j \in J\}$)
Q^d	Integer variable for total development in drifts
Q^h	Integer variable for total amount of production hole drilling in stopes
Q^o	Integer variable for total amount of ore material mucked from stopes

The CP model is given as (1)-(11):

Objective function

$$\textit{Minimize} \max_{j\in J}(\texttt{endOf}(Y_j)) \tag{1}$$

Constraints

$$\texttt{alternative}(Y_j, X_{jm} \mid m \in M_j) \qquad \forall\, j \in J \tag{2}$$

$$\texttt{noOverlap}(S_m, D) \qquad \forall\, m \in M \tag{3}$$

$$\texttt{endBeforeStart}(Y_j, Y_i) \qquad \forall\, j \in J,\ i \in A_j \tag{4}$$

$$\texttt{forbidExtent}(Y_j, \textit{Blast\_calendar}) \qquad \forall\, j \in J \setminus B \tag{5}$$

$$\texttt{cumulFunction}(Q^d) = \sum_{j\in J} \texttt{stepAtEnd}(Y_j, d_j) \tag{6}$$

$$\texttt{alwaysIn}(Q^d, s_t, e_t, t \times \underline{d}, t \times \overline{d}) \qquad \forall\, t \in T \tag{7}$$

$$\texttt{cumulFunction}(Q^h) = \sum_{j\in J} \texttt{stepAtEnd}(Y_j, h_j) \tag{8}$$

$$\texttt{alwaysIn}(Q^h, s_t, e_t, t \times \underline{h}, t \times \overline{h}) \qquad \forall\, t \in T \tag{9}$$

$$\texttt{cumulFunction}(Q^o) = \sum_{j\in J} \texttt{stepAtEnd}(Y_j, o_j) \tag{10}$$

$$\texttt{alwaysIn}(Q^o, s_t, e_t, t \times \underline{o}, t \times \overline{o}) \qquad \forall\, t \in T \tag{11}$$

Objective (1) of the CP model is to minimize the makespan. Constraint (2) ensures that only one optional variable is chosen for an interval variable i.e. only one machine (with the appropriate type) is used to perform a given activity. Constraint (3) prevents machines from being used simultaneously, meaning that each machine can only be assigned to one activity at a time. Constraint (4) takes into account the order in which activities must be performed at a site, with most activities having only one predecessor and some having none. It is important to note that the site where each activity must be carried out is predefined in the input data. Therefore, all activities can be executed in their respective predetermined sites.

Constraint (5) is used to ensure that only blasting activities occur during designated blast windows. In order to model the blasting constraint in the CP model, the day and night work shifts are compressed into a 110-time unit period (each shift consists of 55 time units), where each time unit represents 10 minutes in the real world. This compression allows for blasting activities with a length of zero time units to occur only at the end of compressed periods, every 110 time units (see Figure 3). Multiple blasting activities can be performed at the same time during each blasting window.

Constraints (6) and (7) are introduced to ensure that the progress of development rounds each day (measured in meters per day) is maintained within specific limits based on the defined development target. To model these development constraints, we define time windows $[s_t, e_t)$ each representing a day in the schedule. For each time window, we establish cumulative upper and lower bounds (based on daily bounds) for total development in drifts (in meters) that must be achieved. Next, we use the `cumulFunction` to model the cumulative amount of development per meter and apply the `alwaysIn` constraint to ensure that the cumulative function stays within the target value bounds for each time window. The function $\texttt{stepAtEnd}(i, j)$ returns an elementary cumulative function with a step of height j (a non-negative integer value) at the end of interval variable i. The presented development constraints aim to achieve the desired daily development target in the generated schedule. Furthermore, by using the `cumulFunction` in this constraint, the model is able to flexibly compensate in the following days for any shortfall in achieving the daily development goal (see e.g. Figure 6a). This feature of the constraints closely resembles what is taken into account in actual short-term underground mine planning, making the model more practical for real-world operations.

Constraints (8) and (9) model the production drilling constraint to ensure that the amount of production holes drilled in the stope per day (measured in meters per day) is restricted within certain bounds, defined based on the production drilling objectives. Furthermore, Constraints (10) and (11) are used to model the stope mucking constraint, which ensures that the amount of ore material mucked from stopes each day (measured in tons per day) is maintained within specific limits determined based on the production target. These constraints (Constraints (8)-(11)) aim to achieve the production plan in the produced schedule by controlling the daily amount of production holes drilled and ore mucked. Similar to the development constraints, the production constraints also allow for making up for shortfalls in meeting the daily production goal. An activity can perform either development or production depending on the type of cycle. If the activity is part of a development cycle, it can have development (d_j), and if it is involved in a production cycle, it can have either production hole drilling (h_j) or stope mucking (o_j). The development cycle includes activities that are performed in waste rocks lacking financial value to access economically valuable deposits, while the production cycle is conducted in valuable rocks to extract ore material from the stope.

In the presented short-term mine planning model, operational development and production targets (KPIs) are considered to achieve the tactical decisions made at the medium-term planning level. Specifically, tactical decisions in underground mine planning are typically associated with defining the extraction sequence over a planning horizon of one to three months [7].

4 Implementation and Results

The experiments were conducted on a computer featuring an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and 16 GB of RAM. The CP models were solved using the Constraint Programming Optimizer in IBM ILOG CPLEX Optimization Studio version 12.8.0.

The model was tested on two real data sets collected from the Meliadine underground gold mine in Nunavut, Canada. Both data sets involve scheduling activities for a roughly one-week planning horizon. The first data set (Instance 1) relates to development operations, which consist of 15 machines and 291 activities to be performed across 18 sites. The total advancement achieved by all available development rounds in this instance is equal to 188 meters. Specifically, each round (cycle) results in approximately 4 meters of advancement in the development drift. The second data set (Instance 2) concerns production operations and includes 27 machines, 185 activities, and 27 sites. In this instance, a total of 1500 meters of production holes have been drilled across all accessible stopes, resulting in the extraction of 27,000 tons of ore material. The available resources are categorized into different equipment types, and the number of each type is reported in Table 4. Although the developed CP model takes into account both development and production activities, there was no data available (in the mine) that included both activities together. Therefore, we applied our model separately to two different datasets: one for development and another for production.

Table 4 Number of machines per equipment type for Instances 1 and 2.

Equipment type	Instance 1	Instance 2
Scooptram	2	7
Bolter	6	-
Scooptram clean face	1	-
Jumbo	3	-
Anfo loader	3	3
Truck	-	7
Raise borer	-	1
Production drill rig	-	4
Cabling machine	-	5

The results obtained by implementing the CP model on Instances 1 and 2 are presented in the following subsections.

4.1 Instance 1

Table 5 presents the results of schedules generated for Instance 1 with different daily development upper bounds ($\overline{d}$) in the development constraint. All the models in the table are solved to optimality in a short amount of time. As the primary objective of the scheduling model is to minimize the makespan, the lower bound on daily development specified in the development constraint is readily satisfied. Therefore to evaluate the effect of different development targets on the resulting schedule, we only modify the upper bound value for the total amount of development to be accomplished per day.

Table 5 Results of different CP models on Instance 1.

Model	Development upper bound ($\overline{d}$)	Makespan	Solving time
1	24	882	12 sec
2	28	772	13 sec
3	32	678	12 sec
4	36	635	13 sec
5	40	600	11 sec
6	44	600	10 sec
7	∞	600	10 sec

As can be seen from Table 5, increasing the upper bound in the development constraint results in lower makespans in the produced schedule. Furthermore, the schedule makespan remains unchanged for bounds greater than 40. Therefore, $\overline{d} = 40$ can be considered a suitable daily development target for generating a short-term schedule on Instance 1. Interestingly, this value coincides with the daily development target employed by the human planner at the mine – our model confirms this empirical choice. Figure 4 displays the location-based Gantt chart for the short-term schedule generated on Instance 1 with $\overline{d} = 40$.

The daily and cumulative development resulting from the schedule produced using Model 3 on Instance 1 with $\overline{d} = 32$ are displayed in Figures 5a and 5b. As seen in Figure 5a, the maximum daily development limit of 30 meters is respected, resulting in a total cumulative development of 188 meters in six days (Figure 5b).

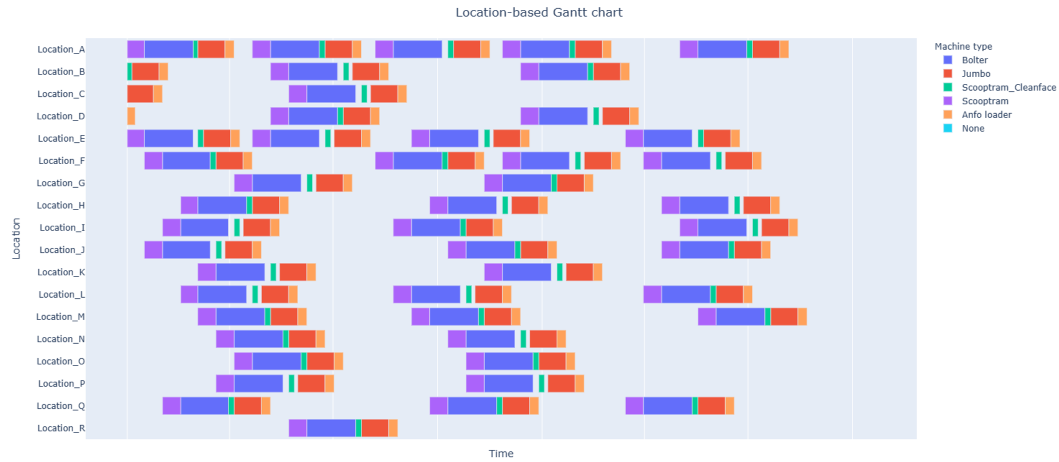
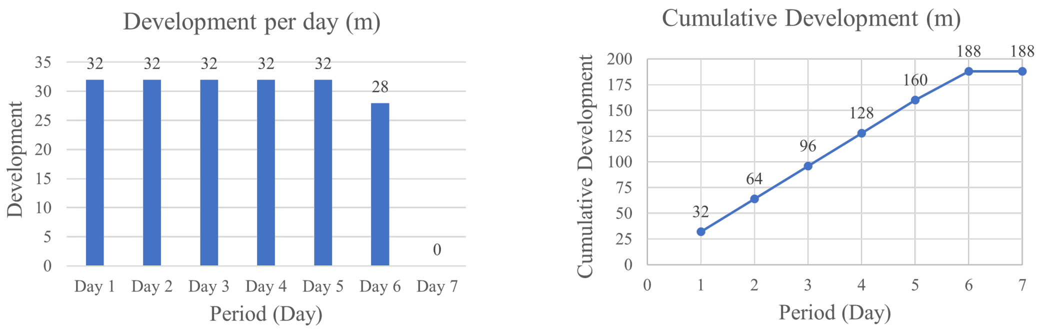


Figure 4 Location-based Gantt chart for the generated schedule on Instance 1.

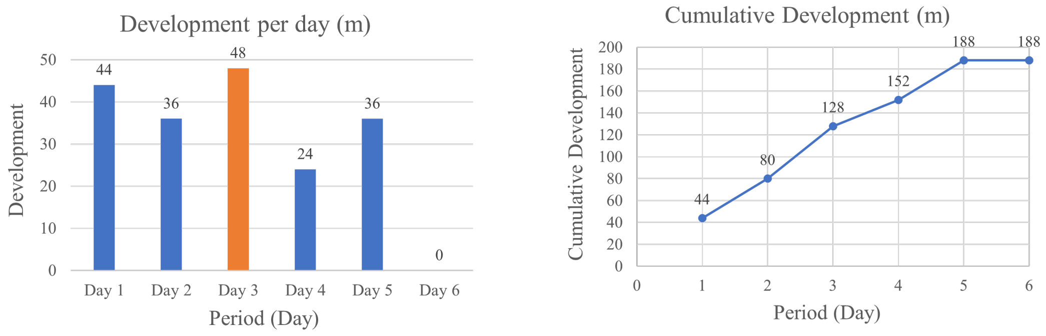


(a) **(b)**

Figure 5 Daily (a) and cumulative (b) development in Model 3 ($\overline{d} = 32$) on Instance 1.

Figures 6a and 6b show the daily and cumulative development obtained from Model 6 on Instance 1 with $\overline{d} = 44$. According to Figure 6a, 36 meters of development are achieved on Day 2, which is lower than the maximum daily target of 44 meters. However, this shortfall is made up on Day 3 by completing 48 meters of development, above the maximum daily target. In other words, 48 meters of development are completed on Day 3 to compensate for the shortfall on Day 2. After Day 3, it is not possible to meet the maximum daily goal due to the limited number of drifts available. As shown in Figure 6b, the total cumulative development of 188 meters is reached in five days. This feature of the development constraints in the CP model can be practical for short-term planning in underground mines, where operational restrictions or a relatively small number of accessible drifts (sites) prevent the achievement of the development target on certain days.

Figure 7 shows the comparison of the average utilization rate of several machine types in schedules produced using CP models on Instance 1 with different $\overline{d}$ for the development constraint. The utilization rate of a machine is the total amount of time units during which the machine was actively operating at the site relative to the total amount of time for which it was available for use. As can be seen from this figure, increasing $\overline{d}$ results in a higher



(a) **(b)**

Figure 6 Daily (a) and cumulative (b) development in Model 6 ($\overline{d} = 44$) on Instance 1.

average utilization rate of machines in the schedule. This is due to the fact that larger $\overline{d}$ values lead to more compact schedules with lower makespans, which in turn, reduces waiting time for machines.

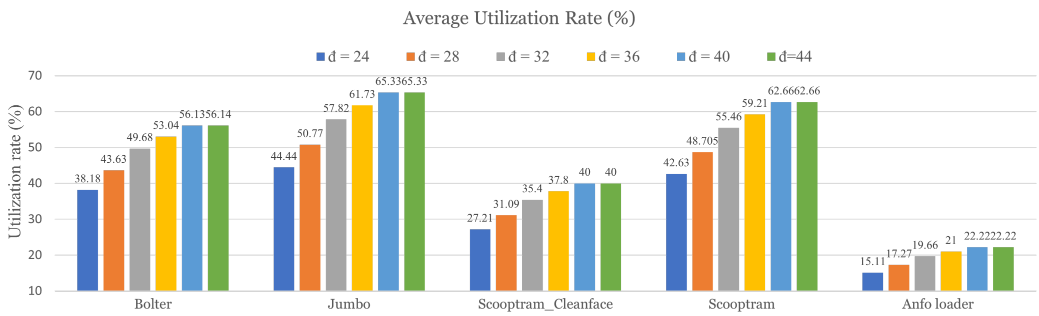


Figure 7 Average utilization rate of machines in schedules with different development bounds ($\overline{d}$) on Instance 1.

4.2 Instance 2

Table 6 displays the makespan of schedules generated by implementing different models on Instance 2, with distinct upper bounds for the daily production drilling ($\overline{h}$) and stope mucking ($\overline{o}$) in production constraints. According to Table 6, reducing the upper bound values in production constraints leads to longer makespans in the generated schedule. Specifically, for the production drilling constraint, the suitable $\overline{h}$ value is 400 meters, as it leads to the lowest makespan value that remains unchanged for larger upper bounds. Similarly, for the stope mucking constraint, $\overline{o} = 6,000$ appears to be an appropriate stope mucking target for the schedule generated on Instance 2. Figure 8 shows the location-based Gantt chart for the created schedule on Instance 2 with $\overline{o} = 6,000$.

Figures 9a and 9b present the daily and cumulative production drilling rates in the schedule generated using Model 5 with $\overline{h} = 500$ on Instance 2. Figure 9a demonstrates that the drilling rate exceeds the daily limit by reaching 600 meters on Day 2 to compensate for the shortfall on Day 1. As depicted in Figure 9b, the total production drilling of 1500 meters is achieved within four days.

Table 6 Results of different CP models on Instance 2.

Model	Production drilling upper bound ($\overline{h}$)	Stope mucking upper bound ($\overline{o}$)	Makespan	Solving time
1	∞	∞	730	16 sec
2	200	–	1060	17 sec
3	300	–	840	16 sec
4	400	–	730	16 sec
5	500	–	730	16 sec
6	–	4000	881	17 sec
7	–	5000	771	16 sec
8	–	6000	730	17 sec
9	–	7000	730	17 sec

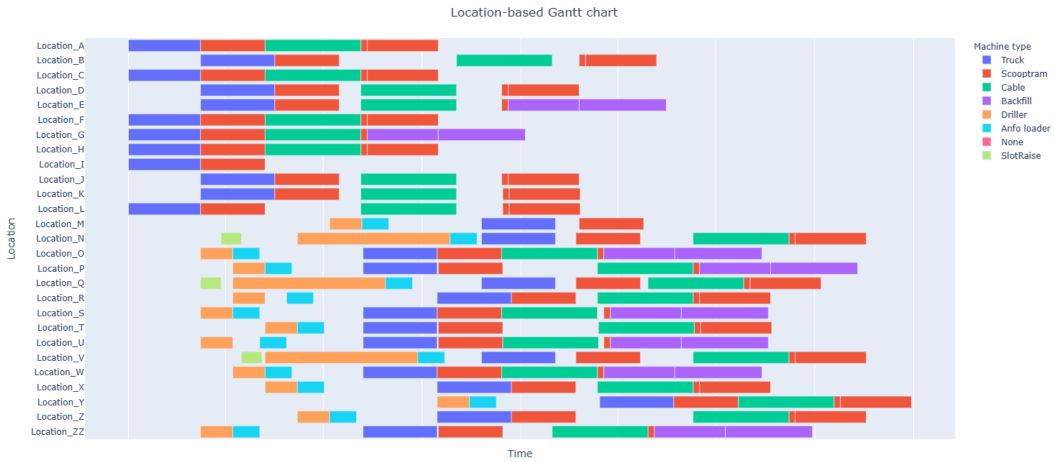
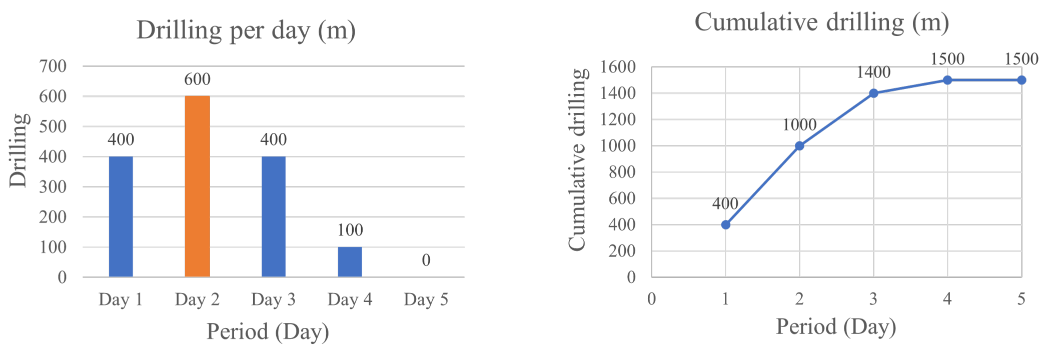


Figure 8 Location-based Gantt chart for the generated schedule on Instance 2.



(a) **(b)**

Figure 9 Daily (a) and cumulative (b) drilling in Model 5 ($\overline{h} = 500$) on Instance 2.

Figures 10 and 11 compare the average utilization rate of several machine types in schedules produced using CP models on Instance 2, with different values for $\overline{h}$ and $\overline{o}$, respectively. According to these figures, the average utilization rate of machines increases for schedules with higher upper bounds in production constraints.

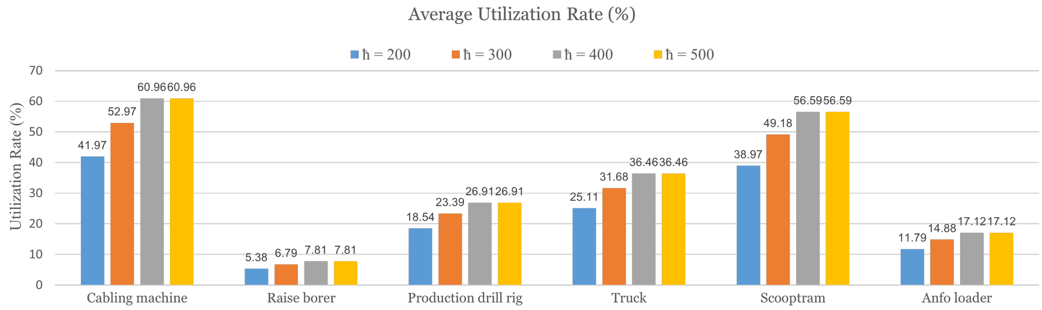


Figure 10 Average utilization rates of machines in schedules with different production drilling bounds ($\overline{h}$) on Instance 2.

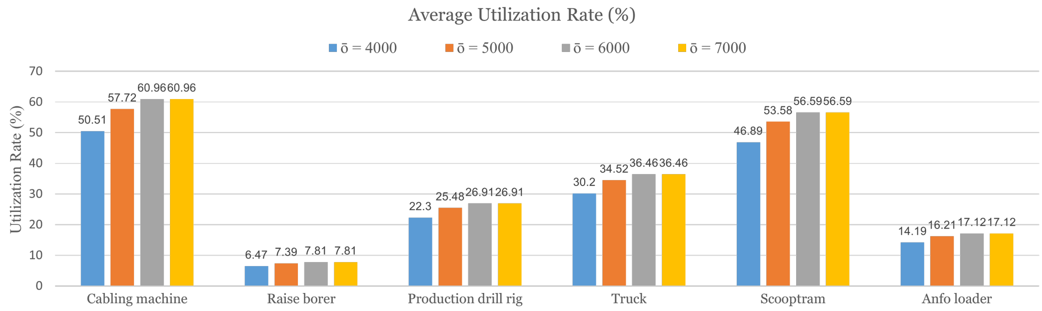


Figure 11 Average utilization rates of machines in schedules with different stope mucking bounds ($\overline{o}$) on Instance 2.

5 Added Value of CP

Constraint Programming allowed us to efficiently address short-term underground mine planning by quickly producing optimal schedules minimizing the makespan. Moreover the model identifies $\overline{d} = 40$ as the appropriate daily development target, which aligns with the value selected by the mine planner and confirms the practice of setting the development upper bound at 40. Additionally, the model can explore what-if scenarios by varying parameter values, such as the impact of changing daily development or production targets in the model on machine utilization rates in the generated schedule. These results demonstrate the practicality and efficiency of using the CP model for short-term scheduling in real-world underground mining operations.

5.1 Comparison of CP model and manual approach

In order to demonstrate the effectiveness of our optimization model, we compared the short-term schedules produced by the CP model with those manually created by the mine planner for the same instance. Since a detailed schedule of activities with similar time fidelity to the schedule produced using the CP model was not provided in the studied mine, we only compared the schedule makespan. In particular, we compared the number of shifts required to complete all activities in the generated short-term schedule using the CP model and manual approach for both Instances 1 and 2, as shown in Table 7. The development KPI considered for scheduling activities in Instance 1 is 40 meters per day (m/day). In Instance

2, the production drilling KPI is 400 m/day, and the stope mucking KPI is 6,000 tons per day. As previously mentioned, each day consists of two working shifts, where each shift is equivalent to 55 time units in the CP model.

Table 7 Comparison of schedule makespan between CP model and manual approach for Instances 1 and 2.

Instance	CP model	Manual approach
1	11 Shifts	14 Shifts
2	14 Shifts	16 Shifts

Table 7 shows that the CP approach outperforms the manual scheduling method on both Instances 1 and 2 by creating more compact schedules with lower makespans (based on the number of shifts) while satisfying the daily development and production targets (KPIs). Additionally, the CP models are quickly solved to optimality, making it an efficient tool for mine planners to rapidly generate updated short-term schedules whenever changes occur in the underground mine plan.

The results of this study demonstrate advantages of the developed CP model for optimizing short-term planning in underground mines and reducing the reliance on manual scheduling, which is highly dependent on the planner's experience. Moreover, the CP model can be easily adjusted to accommodate or exclude additional activity types and related constraints based on the specific requirements of underground mining operations.

6 Literature review

Short-term underground mine planning models are often difficult to solve (NP-hard) due to various operational constraints to consider and to the large number of variables involved. However there has been notable research interest in developing new mathematical models and algorithms to optimize short-term scheduling in underground mines.

Nehring et al. (2010) designed a MIP model to optimize the short-term scheduling and allocation of loader-trucks in sublevel stoping mines. The model allows for the reallocation of equipment in response to changes in underground operations. The proposed model was applied to a copper mine, demonstrating satisfactory results in terms of tonnage deviations from predetermined amounts throughout the planning period [11]. O'Sullivan and Newman (2015) introduced an Integer Programming (IP) model for scheduling activities in an Irish lead and zinc underground mine to maximize the discounted amount of produced metal. Both exact and heuristic solutions were used to reduce the number of variables in the model. Additionally, an optimization-based decomposition heuristic was developed to generate feasible schedules in less computation time for complicated problem instances [12]. Song et al. (2015) developed a decision support tool to determine the scheduling of activities in underground mines. The tool was tested on a real mine dataset in Finland and significantly decreased the makespan compared to manual scheduling methods, thereby improving operational performance. However, the proposed method did not take into account uncertainty related to unexpected activities in underground operations [15].

Schulze and Zimmermann (2017) introduced a solution approach for short-term production scheduling in underground mining. The developed approach assigns staff and machines to mining activities while considering operational constraints with the goal of minimizing deviations from targeted production in a potash mine. The method was tested on various instances and demonstrated superior performance when compared to manual scheduling [13]. Seifi et al. (2019) proposed a two-stage solution approach for scheduling machines and staff in an underground potash mine in Germany. The first step involves solving the relaxation of the MIP model, and in the second step, a heuristic algorithm is used to modify the solutions obtained from the relaxation model to achieve feasible schedules. The experiments conducted on real-word datasets show that the developed approach outperforms the heuristic procedure presented by Schulze and Zimmermann (2017)[14]. Wang et al. (2020) utilized a genetic algorithm (GA) for optimizing the scheduling of equipment used in underground mining. A Non-Linear Programming (NLP) model is presented with a significant number of decision variables associated with multiple mining sites and equipment types [17]. A MIP model was presented by Campeau and Gamache (2020) to optimize short-term planning in underground mines. The goal was to maximize material extraction while ensuring a minimum ore production rate to keep the mill active. The model considers operational and resource constraints to generate feasible schedules. When applied to a gold mine data set, the model produced an optimal short-term schedule [5]. Campeau et al. (2022) introduced a novel MIP model to address short- and medium-term planning in underground mines. The model integrated continuous variables for time discretization, resulting in realistic schedules. The effectiveness of the model was demonstrated by applying it to a dataset from a Canadian gold mine, which produced promising results [7].

Over the last few years, several CP approaches have been proposed to tackle the short-term underground mine planning problem. A model using CP was suggested by Astrand et al. (2018) for scheduling a mobile fleet in underground operations, which was tested on data from an actual underground mine [2]. Astrand et al. (2020) extended the previously developed CP model by incorporating the time it takes for mobile machinery to travel between different sites in an underground cut-and-fill mine. They also proposed a revised CP model with compressed blasting time and post-processed solutions to obtain schedules for the primary problem. In order to improve the quality of schedules and reduce computation time, a specialized neighborhood definition was implemented in a Large Neighborhood Search (LNS) algorithm. The effectiveness of this algorithm was assessed using several instances of an underground mine in Sweden. The outcome showed that the suggested method successfully enhanced the initial feasible solution and generated high-quality schedules [3]. Campeau and Gamache (2022) presented a CP model for short- and medium-term planning in underground mining. They evaluated the model's ability to address long-term production planning objectives by testing it on five data sets from a Canadian underground gold mine, considering a planning horizon of up to one year. The outcomes revealed that the CP model was superior to the equivalent MIP model in terms of computational efficiency and application [6].

These previous CP approaches for short-term underground mine planning exhibit a limited ability to incorporate daily mine planning development and operational goals during the short-term scheduling process. To overcome this limitation, this paper introduced a CP model for the short-term scheduling of activities in underground mining that takes into consideration operational constraints and the development and production targets of the mine plan to generate more practical and reliable schedules.

7 Conclusion

This paper presented a CP model that takes into account various operational constraints and daily development and production targets for short-term scheduling optimization in underground mines. The model was tested on two data sets from the Meliadine gold mine using the long-hole stoping mining method. We conducted a comparative analysis of the schedules generated by our CP model and those created manually by the mine planner. The experiments showed that the CP model outperforms the manual approach, resulting in more efficient schedules with lower makespans. Results highlight the potential benefits of implementing the CP model in actual underground mining operations to improve both development and production through optimized short-term mine planning. Underground mines are somewhat unpredictable environments which may affect how long an activity actually takes. For future work, it could be beneficial to incorporate uncertainty in activity durations which would improve the robustness of the short-term schedule.

References

1 Max Åstrand, Mikael Johansson, and Jenny Greberg. Underground mine scheduling modelled as a flow shop: a review of relevant work and future challenges. *Journal of the Southern African Institute of Mining and Metallurgy*, 118(12):1265–1276, 2018.
2 Max Åstrand, Mikael Johansson, and Alessandro Zanarini. Fleet scheduling in underground mines using constraint programming. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26–29, 2018, Proceedings 15*, pages 605–613. Springer, 2018.
3 Max Åstrand, Mikael Johansson, and Alessandro Zanarini. Underground mine scheduling of mobile machines using constraint programming and large neighborhood search. *Computers & Operations Research*, 123:105036, 2020.
4 Philippe Baptiste, Claude Le Pape, and Wim Nuijten. *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer Science & Business Media, 2001.
5 Louis-Pierre Campeau and Michel Gamache. Short-term planning optimization model for underground mines. *Computers & Operations Research*, 115:104642, 2020.
6 Louis-Pierre Campeau and Michel Gamache. Short-and medium-term optimization of underground mine planning using constraint programming. *Constraints*, 27(4):414–431, 2022.
7 Louis-Pierre Campeau, Michel Gamache, and Rafael Martinelli. Integrated optimisation of short-and medium-term planning in underground mines. *International Journal of Mining, Reclamation and Environment*, 36(4):235–253, 2022.
8 Atlas Copco. Mining methods in underground mining. *Atlas Copco: Nacka, Sweden*, 2007.
9 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*, 23:210–250, 2018.
10 B. Marshall. Facts and figures : The state of canada's mining industry. Technical report, The Mining Association of Canada, 2020.
11 Micah Nehring, Erkan Topal, and Peter Knights. Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. *Mining Technology*, 119(4):212–220, 2010.
12 Dónal O'Sullivan and Alexandra Newman. Optimization-based heuristics for underground mine scheduling. *European Journal of Operational Research*, 241(1):248–259, 2015.
13 Marco Schulze and Jürgen Zimmermann. Staff and machine shift scheduling in a german potash mine. *Journal of Scheduling*, 20:635–656, 2017.
14 Cinna Seifi, Marco Schulze, and Jürgen Zimmermann. A two-stage solution approach for a shift scheduling problem with a simultaneous assignment of machines and workers. In *Mining Goes Digital*, pages 377–385. CRC Press, 2019.
15 Zhen Song, Håkan Schunnesson, Mikael Rinne, and John Sturgul. Intelligent scheduling for underground mobile mining equipment. *PloS one*, 10(6):e0131003, 2015.
16 Farzad Sotoudeh, Micah Nehring, Mehmet Kizil, Peter Knights, and Amin Mousavi. Production scheduling optimisation for sublevel stoping mines using mathematical programming: A review of literature and future directions. *Resources Policy*, 68:101809, 2020.
17 Hao Wang, Victor Tenorio, Guoqing Li, Jie Hou, and Nailian Hu. Optimization of trackless equipment scheduling in underground mines using genetic algorithms. *Mining, Metallurgy & Exploration*, 37:1531–1544, 2020.