Les Cahiers du GERAD ISSN: 0711–2440

Short- and medium-term optimization of underground mine planning using constraints programming

L.-P. Campeau,
M. Gamache

G–2018–94

November 2018

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, 2018
– Bibliothèque et Archives Canada, 2018

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, 2018
– Library and Archives Canada, 2018

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

Short- and medium-term optimization of underground mine planning using constraints programming

Louis-Pierre Campeau
Michel Gamache

GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7

louis-pierre.campeau@gerad.ca
michel.gamache@gerad.ca

November 2018
Les Cahiers du GERAD
G–2018–94

Abstract: For the past few years, the mining industry has seen a lot of operational changes. Digitalization and automation of many processes have paved the way for an increase in its general productivity. In keeping with this trend, this article presents a novel approach for optimizing underground mine scheduling for the short- and medium-term. This problem is similar to the Resource-Constrained Project Scheduling Problem, with some particularities. The model uses Constraint Programming principles to maximize the Net Present Value of a mining project. It plans work shifts for up to a year in advance, considering specialized equipment, backfilling and operational constraints. Results from its applications to datasets based on a Canadian gold mine demonstrate its ability to find optimal solutions in a reasonable time. A comparison with an equivalent Mixed Integer Programing model proves that the Constraint Programming approach offers clear gains in terms of computability and readability of the constraints.

Keywords: Constraint programming, underground mining, scheduling, short- and medium-term planning

Résumé : Depuis quelques années, l'industrie minière subit de nombreux changements opérationnels. La numérisation et l'automatisation de plusieurs procédés ont permis d'augmenter la productivité globale de cette industrie. Suivant cette tendance, cet article présente une nouvelle approche de résolution du problème d'optimisation de l'ordonnancement des activités dans les mines souterraines pour le court et moyen terme. Il s'agit d'un problème bien connu semblable au problème de gestion de projet avec contraintes de ressources incluant quelques modifications. Le modèle utilise les principes de la programation par contraintes afin de maximiser la valeur actualisée nette d'un projet minier. Il permet la planification des activités par quart de travail, et ce pour l'année à venir, considérant les équipements spécialisés, le remblayage et les contraintes opérationnelles. Les résultats de son application sur des données basées sur une mine d'or canadienne permettent de démontrer la capacité de cette approche à trouver des solutions optimales en des temps raisonnables. Une comparaison avec une formulation de programmation en nombres entiers démontre clairement les avantages de la formulation en programmation par contraintes au niveau des temps de résolution et de la lisibilité des contraintes.

Mots clés : Programmation par contraintes, mines souterraines, ordonancement, planification à court et moyen terme

Acknowledgments: This research was funded with the Discovery Grant no. RGPIN–06402–2016 from The Natural Sciences and Engineering Research Council of Canada (NSERC)

1 Introduction

Underground mines are a unique environment with their own specific challenges. Rock mechanics, dewatering, ventilation and making a choice from one of the many mining methods are only a few examples of the technical aspects that have to be taken into consideration when solving a scheduling problem. In order to consider all of these challenges while keeping focus on the overall profitability, different levels of planning are used over the course of a mine, each of them taking into consideration certain aspects of underground mining. The planning of an underground mine generally starts by creating a "life of mine" planning. This is usually done at the very beginning of a mining project and is revised generally every year with regards to new information gathered from exploration drilling or accordingly with the advance of planned activities. Through this, the general shape of the developments needed to access the ore body is drawn and the ore body itself is split in approximative stopes, i.e. unitary subsections of the ore body, with generic size and shape. Long-term planning is based on this life of mine and plans activities with a very low granularity over periods lasting one year. The focus of this exercise is mostly on the economical considerations, with most of the technicality being considered through rough approximations. The first year of this planning is then used to create a medium-term planning, which will generally be revised every few months. At this stage, activities are planned with more precision over periods ranging typically from one to three months according to the general objectives defined by the long-term planning. Using the first periods of medium-term planning, the short-term planning is then created with periods of one or two weeks and revised every few weeks. This last level of planning is used daily by planners and foremen to schedule activities through the following shifts and the dispatch of equipment between these tasks.

Figure 1 is a typical representation of a long-hole stope, one of the many mining methods used underground and Figure 2 represents the popular "Pyramidal" sequence of extraction for long-hole stopes. They will be used here to illustrate some of the more important mining concepts that are necessary to understand this article. Many specialized crews and equipment are required in an underground mine depending on the mining method, the type of development, and the rate of production. These crews can either work one after another over multiple work shifts, or semi-simultaneously one after another within a single shift and repeat this for multiple days. The former generally happens in stopes, where a lot of work is needed to prepare a single blast that will provide haulage material for many shifts. For example, in the case of a long-hole stope, as in Figure 1, all of a stope's holes must be drilled first by the production drilling crew, before they can be loaded by the loading crew. The latter usually happens in the development of galleries where a series of activities need to be repeated for as many blasts as necessary. Taking Figure 1 as an example, the drill accesses are excavated through a series of smaller blasts, so in order to excavate the whole length of the accesses, the development drilling and haulage crews will have to cycle through this site as many times as there are blasts required to completely develop the drift.

Blasts can be seen as a base unit of production in vertical or horizontal developments in underground mines. The length of excavated rock they produce is fixed for a given type of heading as well as the amount of work required. For safety reasons, blasts can only happen in between shifts (generally twice a day), so the time required to excavate a gallery is more dependent on the number of blasts than the sum of work time since it generally takes less than a shift to complete the work sequence leading to a blast. In the case of stopes, several mining methods require backfilling as a last activity. This activity consists of filling the empty stope with a mix of rock and concrete in order to stabilize neighboring stopes. A delay of two to three weeks then has to be allowed for the concrete to solidify before anything can happen in adjacent locations. In Figure 1 for example, the open stope illustrated would have to be backfilled before the section of ore body next to it could be extracted. Stopes are also usually extracted in a predefined order, as represented in Figure 2. This order is mostly dictated by rock mechanics for stability reasons. All of these concepts (e.g. backfilling, fixed-time blast, specialized crews) are unrelated to open-pit mining, which makes the underground planning problem very different.

The goal of this article is to develop a model that would integrate short- and medium-term planning for underground mines into a single model. The model has to create planning over a horizon of more than a year in order to consider long-term objectives, while being detailed enough to be used at the short-term level. It is important to consider long-term objectives when planning for shorter-terms because the accesses to the

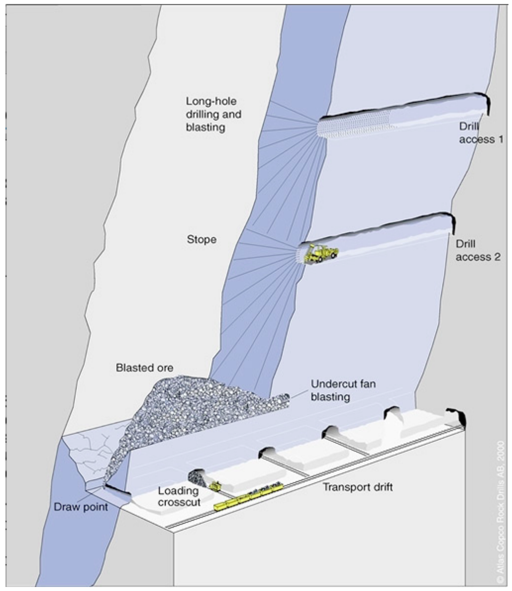

Figure 1: Typical representation of an underground long-hole operation [1]

Stope 15	Stope 13	Stope 9	Stope 12	Stope 14
Stope 11	Stope 8	Stope 4	Stope 7	Stope 10
Stope 6	Stope 3	Stope 1	Stope 2	Stope 5

Figure 2: Pyramidal sequence of extraction for long-hole stopes

ore-rich parts of an underground mine have to be prepared many weeks, if not months, in advance with costly developments. Thus, a production plan corresponding to the optimal solution on a short-term scale would certainly not include these developments for production outside of its time horizon and be sub-optimal on the long-term. Covering a complete long-term period (i.e. one year), the model takes into account the long-term impact of short-term decisions and eases the inclusion of long-term objectives (e.g. complete all developments of veins X and Y by the end of the year, mine Z tonnes of ore) into the short-term planning. Moreover, the model has to be scalable to allow for a frequent re-optimization of the problem accordingly to operation changes and unplanned events. The application of such model to real-world mining operations yields many benefits. First, it produces automatically optimal solutions to a problem that is still solved by hand in most of the mining industry. Second, it addresses the loss of optimality coming from successive resolution of the different planning levels by finding the global optimal solution to short- and medium-term planning. Third, it requires a lot less time to solve and guarantees a much faster response time for the planner to produce these updated schedules every time a change occurs and a new planning needs to be created.

This article will present a Constraint Programming (CP) model that addresses the points mentioned here and compares its capacity to a more traditional Mixed Integer Programming (MIP) model. CP has proven its value for many types of problems, including the closely related Resource-Constrained Project Scheduling Problem. It is a novel approach for solving this problem since, as will be explored in the next section, all of the examples available in the literature are MIP models.

2 Literature review

There are many examples of optimization in the mining industry and reviews of the literature in optimization in natural resources and mining can be found in [4] and [20]. Most of these developments, however, are specific to open pit mining, as reviewed in [22] among others. Even if publications are fewer, the reader can find in [18] an overview of recent developments and opportunities in underground mine optimization.

More specifically in planning for underground mines, some models are available for the optimization of long-term planning. In [10], the authors present a model for the optimization of a mining complex including underground and open-pit mines. More recently, in [16] authors describe a method for optimizing a similar problem comprising several underground and open-pit mines while taking into consideration geological uncertainty. On a more technical perspective, GU et al. in [11] present a generalized procedure based on column generation for solving a variation of the Resource Constrained Project Scheduling Problem (RCPSP) which aims at maximizing the discounted cash flow (RCPSPDC). The model is then tested on a long-term scheduling problem from an underground mine. Similarly, in [17] authors demonstrate that the well-known methodology presented in [3] to solve LP relaxations of open-pit problems can be applied to solve relaxations of a much broader category of problems like the RCPSP, including underground mine planning problems.

Some articles present solutions to the integration of other aspects of underground mines. For example, authors in [14] describe a model for the simultaneous optimization of stopes design and scheduling. In a similar fashion, Collard et al. (see [9]) present a model for the optimization of underground mine schedule while considering the cut-off grades i.e. the minimum ore content at which rock is considered to be ore. The authors of [8] later proposed a stochastic variation of this model. The model presented in [5] also optimizes the schedule while taking into consideration variable stope sizes, but does so by using linear approximations of grade versus tonnage curves for the different stopes.

Short- and medium-term models have also been developed but tend to be more application specific. This is because for these kinds of time horizons, many site specific or mining method specific constraints have to be added. The authors of [15] describe a model and a heuristic for the scheduling of activities at LKAB's Kiruna iron ore mine. A model for integrated short- and medium-term planning can also be found in [19] but its application is limited to a conceptual 30 stopes dataset. Another site specific application is described in [21], where the authors describe the results of the application of an optimization model in the planning of the final two years of activity at the Lisheen zinc mine in Ireland. The model displayed in [7] allows for the optimization of short-term planning and is tested on a dataset based on a Canadian gold mine. The model from [6] uses a variable time discretization to extend the solvable planning horizon for the same dataset.

As mentioned before, the problem of mine scheduling has many similarities with the RCPSP. The field of constraint programming has been proven in the past to be very effective for this class of problem. To cite one among others, Kreter et al. (see [12]) show the advantage of CP models over classic MIP models for a variation of the RCPSP called RCPSP/max-cal, where time lags and resource calendars have to be respected while minimizing the total makespan. To the best of the authors' knowledge, the only other underground mining application of CP in the literature is [2], which proposes a model for the optimal dispatch of equipment for time horizons of less than 72h. For a complete overview of the CP solver used for this article as well as examples, the reader is referred to [13].

3 Model

The model uses six different index types. The first one s, is used to designate sites. The term site is used to designate any location where an activity can happen. Long galleries are split into segments for each intersection so that a site represents a single tunnel without any branches. Long galleries without intersections are also split into many sites in order to allow for their development to be segmented into a few parts. Index a represents the different activities happening in each site, e.g. haulage, drilling, explosives loading. Index c refers to the different types of specialized crews or equipment working underground e.g. production drill, development drill, explosive loader. Indexes l and v represent levels and veins, respectively. Finally, index t refers to production periods, used only for the mill feed constraints. The base unit of time used in this model is the work shift (typically 10 hours), so that all durations are rounded up to the next shift. Shifts make a natural unit of time in underground mines since most series of activities have to end with a blast, and a blast can only be done in between shifts. Hence, even if activities in a gallery are finished in a fraction of a shift, its successor gallery will not start before the next shift. The model presented in this article was built using DOcplex.CP Python API from IBM. The syntax for the function and variables was taken from its documentation. No long-term objectives or limitations were included in the datasets in order to make the resolution harder and test its limits since fixed objectives limit the feasible solution domain. Simple indications on how to include long-term objectives are given through the model description.

3.1 Sets

Four groups of sets are present in the model: A refers to activities, C refers to crews, P refers to predecessors or adjacent stopes, and S refers to sites. Table 1 gives a brief description of all sets used in the model.

Table 1: Sets description

$\mathcal{A}_s^F$	All activities required at site s
$\mathcal{A}_s^F$	First activity at site s
$\mathcal{A}_s^L$	Last activity at site s
$\mathcal{A}_s^H$	Haulage activity at site s
$\mathcal{A}_{sa}^P$	Predecessor activity to activity a at site s
$\mathcal{C}_{sa}$	Crews needed for activity a at site s
$\mathcal{P}_s$	Predecessor sites of site s
$\mathcal{P}_s^{Adj}$	Adjacent stopes of stope $s \in \mathcal{E}^{Stope}$
$\mathcal{P}_s^{Stope}$	Predecessor stopes of stope $s \in \mathcal{E}^{Stope}$
$\mathcal{S}_l$	Sites located on level l
$\mathcal{S}_v$	Sites located in vein v
$\mathcal{S}^B$	Sites requiring backfill
$\mathcal{S}^O$	Sites containing ore
$\mathcal{S}^{Stope}$	Sites that are stopes

3.2 Parameters

Parameter A_c, representing the number of available crews for each type c, is expressed as a percentage to ease the representation of fractional usage. For example, if two crews of type c were available for a scenario, A_c would equal 200. This is made necessary by the fact that in underground mine planning, crews are typically assigned to a fixed number of sites until their completion. For example, each production drilling crew will be assigned 3 sites, which corresponds to the number of blasts one crew can complete in one day (2 shifts). Assigning crews to fewer sites would significantly reduce their productivity since a delay has to be respected after a blast to secure the site. Thus, having a crew that cycles through two sites or fewer would greatly diminish their working time. Assigning a crew to more sites, on the other hand, would slow down the development rate of each of them and dilute the development effort. Parameter U_{sac} represents the percentage of a crew c required by an activity a in a site s. Coming back to our example with the development drilling crew, each activity where it is required would have $U_{sac} = 33$.

Parameter M is the equivalent of DOcplex.CP Python API parameter `INTERVAL_MAX`. It represents a very large number used to set upper bounds to a value so large that it effectively equates to not setting any bound. Parameters O_t^L and O_t^U are lower and upper bounds derived from the global objective of ore production for the mine. Each mine has its own objectives of ore production where the lower bound is usually the minimum feed necessary to keep the ore mill active. Parameters P_t^S and P_t^E define the start and the end of each production period. The production periods are the periods over which the ore production objectives are defined. For example, a mine could define its minimal ore production to be of 3000 tonnes of ore per week, where O_t^L would increase weekly by 3000 and P_t^S and P_t^E represents the start and end of each week.

Parameters R^M, R_l^L, and R_v^V set different limits for different parts of the mine. These limits are often imposed by planners to avoid congestion in different zones or simply to avoid having too much fragmented rock to haul back to the surface for the mine capacity. M, L, and V stand for mine, level, and vein respectively. Parameter $T^D_{sas'a'}$ represents the delay imposed in between two activities a in site s and a' in site s'. These delays can be imposed for many reasons, including for example to wait for the drilling samples to be analyzed by the geology department in a drift leading to stopes. Finally parameter T^{Max}_s imposes a maximum duration for the activities in one site. This is mostly due to ground stability reasons like in the stopes, where backfilling activities have to take place not too long after the haulage is completed. Table 2 summarizes the description of the parameters used in the model.

Table 2: Parameters description

A_c	Step function of the available crews of type c (%)
F_{sa}	Step function of the discounted cash flow associated with the activity a at site s with quarterly steps ($)
M	Large number representing the maximum possible number of time units
O^L_t	Lower bound on the cumulative total tonnage of ore extracted at period t (tonnes)
O^U_t	Upper bound on the cumulative total tonnage of ore extracted at period t (tonnes)
P^S_t	Starting time unit of production period t
P^E_t	Ending time unit of production period t
Q_s	Rock tonnage in site s (tonnes)
R_s	Rate of extraction at site s (tonnes/shift)
R^M	Maximum possible total rate of extraction in the mine at any given time (tonnes/shift)
R^L_l	Maximum possible total rate of extraction in level l at any given time (tonnes/shift)
R^V_v	Maximum possible total rate of extraction in vein v at any given time (tonnes/shift)
T^B	Backfill curing time to be respected in between adjacent stopes (shifts)
$T^D_{sas'a'}$	Time delay imposed by planner in between activity a at site s and activity c' at site s' (shifts)
T^{Max}_s	Maximum time span between the start of the first activity and the end of the last at site s (shifts)
U_{sac}	Percentage of crew type c required for activity a at site s

3.3 Variables

The model uses three kinds of variables. The first one, interval variables, is used to represent the activities. An interval variable has a size, a start, an end time, and can be optional or not. An optional variable is a variable that can be absent from the final solution. In a case where long-term objectives would call for specific sites to be completed before or after a certain date, this can be included directly in the variable definition. The second type, sequence variables, represents unordered sequences of interval variables over which one can impose special constraints. Sequence variables are used in this model to represent the relations between adjacent stopes. Finally, regular integer variables are used to represent quantity variables in the problem. In order to represent potentially fractional usage of crew, variable u_c is used as a percentage. For example, if the activities taking place at a given time require the work from 1.5 crew of type c, the value of u_c will be 150.

Table 3: Variables description

a_{sa}	Optional interval variable for the execution of activity a at site s with start time
$b_{sas'}$	Sequence variable linking variables a_{sa} and $a_{s'a'}$ $\forall s \in \mathcal{S}^B, a \in \mathcal{A}^F_s, s' \in \mathcal{S}^{Adj}_s, a' \in \mathcal{A}_{s'}$
q^O	Integer variable indicating the total tonnage of ore extracted
r^M	Integer variable indicating the total rate of extraction in the mine
r^L_l	Integer variable indicating the total rate of extraction in level l
r^V_v	Integer variable indicating the total rate of extraction in v
u_c	Integer variable indicating the percentage of available crew c being used

3.4 Objective

The objective of the model is to maximize the Net Present Value associated with the activities executed. The function `startEval` bellow simply evaluates the value of the discounted cash flow function F_{sa} for each activity a_{sa} at its start time.

$$Max \qquad \sum_s \sum_c \texttt{startEval}(F_{sa}, a_{sa}) \tag{1}$$

3.5 Constraints

3.5.1 Renewable resources constraints

One type of function and one type of constraint are used to constraint the renewable resources. The function $\texttt{pulse}(i, j)$ creates a step function with a step of height j for the duration of interval variable i. Constraint type $\texttt{alwaysIn}(i, j, k, l, m)$ forces a variable i, in the interval j to k, to take values in between l and m. Constraints 2 link variables u_c to the usage of each type of crew and Constraints 3 make sure that the amount required does not exceed the number available. Constraints 4 link variables r^M to the total rate of mining at all time in the mine and Constraints 5 limit this rate to its maximum value. Pairs of constraints 6–7 and 8–9 limit rates of extraction similarly but for levels and veins respectively. Some of the models discussed previously in this article include blending constraints i.e. constraint limiting the variation of ore grade fed to the mill. These constraints represent the fact that the recuperation of ore in the mill is always better for a constant feed grade. These types of constraints were not included in our model since, as confirmed by industry professionals, these constraints are generally not considered in underground gold mines. If such constraints were required though for a given application, one could simply include them in the form of a renewable resource constraint.

$$u_c = \sum_s \sum_a \texttt{pulse}(a_{sa}, U_{sac}) \qquad \forall c \tag{2}$$

$$\texttt{alwaysIn}(u_c, 0, M, 0, A_c) \qquad \forall c \tag{3}$$

$$r^M = \sum_s \texttt{pulse}(a_{sa}, R_s) \qquad \forall a \in \mathcal{A}_s^H \tag{4}$$

$$\texttt{alwaysIn}(r^M, 0, M, 0, R^M) \qquad \forall c \tag{5}$$

$$r_l^L = \sum_s \texttt{pulse}(a_{sa}, R_s) \qquad \forall l, s \in \mathcal{S}_l, a \in \mathcal{A}_s^H \tag{6}$$

$$\texttt{alwaysIn}(r_l^L, 0, M, 0, R_l^L) \qquad \forall l \tag{7}$$

$$r_v^V = \sum_s \texttt{pulse}(a_{sa}, R_s) \qquad \forall v, s \in \mathcal{S}_v, a \in \mathcal{A}_s^H \tag{8}$$

$$\texttt{alwaysIn}(r_v^V, 0, M, 0, R_v^V) \qquad \forall v \tag{9}$$

3.5.2 Nonrenewable resources constraints

One new function is used to model the nonrenewable resources contraints. $\texttt{stepAtStart}(i, j)$ creates a step function with a step of height j at the start of interval variable i. Contrarily to $\texttt{alwaysIn}$, the value of the step is kept after the end of the interval variable. Constraints 10 and 11 assure that enough ore is extracted to keep the mill constantly fed. The decision to represent the ore feed as a nonrenewable resource with cumulative value along the time horizon comes from the fact that mines often use stock piles where ore is stored at the surface, waiting to be processed at the mill. Hence, it is possible for a mine to extract more ore in a given period in order to compensate for less ore extraction in subsequent periods. If long-term objectives were in the form of a minimum amount of development or production, they could also be included in the model using constraints similar to Constraints 10 and 11.

$$q^O = \sum_s \texttt{stepAtStart}(a_{sa}, Q_s) \qquad \forall s \in \mathcal{S}^O, a \in \mathcal{A}_s^H \tag{10}$$

$$\texttt{alwaysIn}(q^O, P_t^S, P_t^E, O_t^L, O_t^U) \qquad \forall t \tag{11}$$

3.5.3 Precedence constraints

One new constraint type is used for the precedence constraints. The constraints $\texttt{endBeforeStart}(i, j, k)$ assure that interval variable i ends before interval variable j starts, with a minimum delay of k time units between them. Constraints 12 make the precedences links between the first activity of a site and the last of its predecessor while enforcing the required delay between the two activities. Constraints 13 make similar links but between predecessor and successor activities of the same site. Constraints 14 enforce the particular precedence relations between stopes linking the first activities from both stopes. The reason for this special link is that stope precedences are not as strict as regular precedences. As mentioned earlier, these precedences are mostly for rock mechanics reasons and not because of the availability of the stopes. Only the first activities are linked together to represent the fact that when the first crew is done in a stope, it can start its work in the successor stope while the other crews finish their activities in the predecessor stope. Constraints 15 use the `endBeforeStart` formulation to assure that the maximum time span between all activities is respected. This particular formulation is recommended in article [13] over one of the forms, `endOf`(j)-`startOf`$(i) < k$.

$$\texttt{endBeforeStart}(a_{s'a'}, a_{sa}, T^D_{sas'a'}) \qquad \forall s, a \in \mathcal{A}^F_s, s' \in \mathcal{P}_s, a' \in \mathcal{A}^L_{a'} \tag{12}$$

$$\texttt{endBeforeStart}(a_{sa'}, a_{sa}, T^D_{sasa'}) \qquad \forall s, a, a' \in \mathcal{A}^P_{sa} \tag{13}$$

$$\texttt{endBeforeStart}(a_{s'a'}, a_{sa}, 0) \qquad \forall s \in \mathcal{S}^{Stope}, a \in \mathcal{A}^F_s, s' \in \mathcal{P}^{Stope}_s, a' \in \mathcal{A}^F_{s'} \tag{14}$$

$$\texttt{endBeforeStart}(a_{sa'}, a_{sa}, -T^{Max}_s) \qquad \forall s, a \in \mathcal{A}^F_s, a' \in \mathcal{A}^L_{s'} \tag{15}$$

$$\tag{16}$$

3.5.4 Backfilling constraints

One last type of constraint is needed to represent the backfill constraints. `noOverlap`(i, j) constraints restrict all of the interval variables included in the sequence variable i so they do not to overlap, with a minimal time delay of j between them. Constraints 17 assure that the curing time for backfill is respected between adjacent backfilled stopes.

$$\texttt{noOverlap}(b_{sas'}, T^B) \qquad \forall s \in \mathcal{S}^B, a \in \mathcal{A}^F_s, s' \in \mathcal{P}^{Adj}_s \tag{17}$$

4 Results

In order to test our model and compare it with an MIP formulation, we used the five datasets presented in [6]. These five datasets, based on data from a Canadian gold mine, represent five different planning scenarios with a number of possible activities between 842 for dataset 1 and 2229 for dataset 5 with ten different crew types involved. Table 4 shows the main characteristics of each dataset. The starting time of all activities was limited to one year after the scenario's start time. For comparison purposes, the parameters and inputs for the model presented in [6] were modified for these five datasets in order to change the resolution from one week to one shift, so that both models can be equally compared over the datasets.

Table 4: Datasets characteristics

	D1	D2	D3	D4	D5
Sites	385	385	612	838	892
Activities	842	842	1495	2145	2229
Crews	10	10	10	10	10
Levels	4	4	4	4	4
Veins	4	4	5	6	7

Tests for the CP and the MIP models were carried on the same computer with an Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 16 GB of RAM. For the CP model, DOcplex.CP Python API using the Constraint Programming Optimizer of IBM ILOG CPLEX Optimization Studio 12.8.0.0 was used. For the MIP model, the Mathematical Programming Optimizer of IBM ILOG CPLEX Optimization Studio 12.8.0.0 was used. The relative gap tolerance was set to 0.01%, meaning that solutions 0.01% away from the best known upper bound were considered optimal and the time limit was set to 3600 seconds for both models.

Table 5 shows the results of the application of the CP model to the 5 different datasets. In the first section of the table, the objective value of the best solution found is displayed on the "Objective" line. The gap between this value and the best known upper bound is in the "Gap" line and the total number of feasible solutions found in the branching process is displayed in the "No of Solutions" line. The second section of the table shows the computational time needed by the solver to reach solutions within 5% and 1%, respectively, of the best solution found as well as the time at which the best solution was found and the total solving time. The differences between the times to find the best solutions and the total computation time are explained by the fact that in the majority of cases, the branching algorithm needs time to prove the optimality of a solution by lowering its upper bound after the discovery of the optimal solution.

Table 5: Constraints Programming model results for datasets 1 to 5

Solution	**D1**	**D2**	**D3**	**D4**	**D5**
Objective	9.99E+06	1.21E+07	1.59E+07	1.53e+07	1.52e+07
Gap (%)	0.1	0.1	0.1	0.1	0.1
No of Solutions	130	158	830	1084	611
Time (s)					
5%	23.5	13.2	420.7	1086.7	669.0
1%	24.5	31.5	422.1	1182.0	679.1
Best Solution	26.4	48.3	433.4	1240.4	774.3
Total	26.4	70.3	433.4	1252.0	1155.5

Table 6 shows the results of the application of the MIP model to the same five datasets presented before. Once again, the value of the best solution found and the difference with the best known upper bound can be found respectively in the lines "Objective" and "Gap". The LP Relaxation section shows the time needed to solve the linear relaxation to the problem and the value of its solution. The total computing time is displayed in the "Total" line in the last section.

Table 6: Mixed Integer Programing model results for datasets 1 to 5

Solution	**D1**	**D2**	**D3**	**D4**	**D5**
Objective	1.02E+06	-	-	-	-
Gap (%)	1240.5	-	-	-	-
LP Relaxation					
Time	514.7	862.7	-	-	-
Value	1.42E+07	2.08E+07	-	-	-
Time					
Total (s)	3600	2577.3	-	-	-

From the results displayed in Table 5, one can see that the CP model proves to be very effective in solving the five problems. All of them were solved to the optimality limit with many feasible solutions found along the branching process. The times to reach the different percentage away from optimality also shows that the optimality gap limit does not seem to have a large effect on the solution time; most of the good solutions found relatively close to the optimal solution. On the other hand, Table 6 clearly shows that the MIP formulation performs a lot worse than its CP equivalent. It could only find a feasible solution to the simplest scenario within the time limit of 3600 seconds. The solution found was very far from the optimality when compared to either the CP solution to the scenario (9.99E+06) or the gap to the upper bound found by the MIP branch and bound algorithm (1240.5%). All the other scenarios filled the available memory, causing a memory error and halting the process before reaching the time limit. Only Scenario 2 could find the solution of the LP relaxation, but it took more than ten times the amount of time that the CP model took to find the optimal solution.

5 Conclusion

The purpose of this article was to compare different formulations of the same problem, that is, the scheduling of short- and medium-term activities of an underground mine. The two models were based on different solving methods. The one presented in this article was based on the principle of CP, where its comparison was based on mathematical programming. The objective was to solve instances for up to a year ahead in order to allow for the long-term objectives to be considered. The results displayed in this article clearly show that the CP approach surpasses the MIP approach for all the scenarios tested. Of course, some assumptions and approximations needed to be made in order to model the problem as a pure CP problem, but so does the mixed integer approach. The level of precision reached by the model presented in this article (shifts planned for the next year) is probably too detailed for the reality of an underground mine, where planning often has to be redone weekly to address the many changes and unplanned events that happen daily. Considering that the computational time required to solve all instances are very low, however, there is no reason not to plan with this precision for such a long time horizon. These results are also promising for longer-term planning models where time units could be extended in order to take into account time horizons of many years. Other than its computational results, the CP approach also offers advantages in an application context. The many constraints specific to each mine sites and mining techniques can be more easily modeled using the rich dictionary of CP function rather than having to use linear and integer variables. Of course, mixed integer constraints can ultimately model almost every situation but often with the sacrifice of readability and complexity. The dictionary of function makes the model more readable, which makes it easier to maintain once implemented. From the promising results demonstrated for shorter-term planning in underground mines, the authors believe that the application of CP to these time horizons will become more and more present in the literature.

References

[1] Sub-level open stoping. https://goo.gl/qNwo7r. Accessed: 05.11.2018.

[2] Max Astrand, Mikael Johansson, and Alessandro Zanarini. Fleet Scheduling in Underground Mines Using Constraint Programming. In International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research, volume 10848, pages 605–613, 2018.

[3] Daniel Bienstock and Mark Zuckerberg. Solving LP relaxations of large-scale precedence constrained problems, pages 1–14. Springer, 2010.

[4] T Bjørndal, I Herrero, Alexandra Newman, C Romero, and Andres Weintraub. Operations research in the natural resource industry. International Transactions in Operational Research, 19(1–2):39–62, 2012.

[5] Andreas Bley and S.E. Fanie Terblanche. An improved formulation of the underground mine scheduling optimization problem when considering selective mining. Optimization Online, 31(1):1–16, 2015.

[6] Louis-Pierre Campeau, Michel Gamache, and Rafael Martinelli. Integrated optimization of short- and medium-term planning in underground mine. OR Spectrum, In revision process, 2018.

[7] Louis-Pierre Campeau and Gamache Michel. Short-term planning optimization model for underground mines. Computer & Operations Research, In revision process, 2018.

[8] Sabrina Carpentier, Michel Gamache, and Roussos Dimitrakopoulos. Underground long-term mine production scheduling with integrated geological risk management. Mining Technology, 125(2):93–102, 2016.

[9] Jean Collard, Rafael Martinelli, and Michel Gamache. Strategic Planning of an Underground Mine with Variable Cut-off Grades. Les Cahiers du GERAD, G–2013–86, 2013.

[10] Rafael Epstein, Marcel Goic, Andrés Weintraub, Jaime Catalán, Pablo Santibáñez, Rodolfo Urrutia, Raúl Cancino, Sergio Gaete, Augusto Aguayo, and Felipe Caro. Optimizing Long-Term Production Plans in Underground and Open-Pit Copper Mines. Operations Research, 60(1):4–17, 2012.

[11] Hanyu Gu, Peter J. Stuckey, and Mark G. Wallace. Maximising the Net Present Value for Resource-Constrained Project Scheduling. In Principles and Practice of Constraint Programming, pages 362–378, 2012.

[12] Stefan Kreter, Andreas Schutt, and Peter J. Stuckey. Using constraint programming for solving RCPSP/max-cal. Constraints, 22(3):432–462, 2017.

[13] 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.

[14] J. Little, Peter Knights, and Erkan Topal. Integrated optimization of underground mine design and scheduling. Journal of the Southern African Institute of Mining and Metallurgy, 113(10):775–785, 2013.

[15] Michael A. Martinez and Alexandra M. Newman. A solution approach for optimizing long- and short-term production scheduling at LKAB's Kiruna mine. European Journal of Operational Research, 211(1):184–197, 2011.

[16] Luis Montiel, Roussos Dimitrakopoulos, and Kazuhiro Kawahata. Globally optimising open-pit and underground mining operations under geological uncertainty. Mining Technology, 125(1):2–14, 2016.

[17] Gonzalo Muñoz, Daniel Espinoza, Marcos Goycoolea, Eduardo Moreno, Maurice Queyranne, and Orlando Rivera Letelier. A study of the Bienstock–Zuckerberg algorithm: applications in mining and resource constrained project scheduling, volume 69. Springer US, 2018.

[18] Cutberth Musingwini. Presidential address: Optimization in underground mine planning-developments and opportunities. Journal of the Southern African Institute of Mining and Metallurgy, 2016.

[19] Micah Nehring, Erkan Topal, Mehmet Kizil, and Peter Knights. Integrated short- and medium-term underground mine production scheduling. Journal of the Southern African Institute of Mining and Metallurgy, 112(5):365–378, 2012.

[20] Alexandra M. Newman, Enrique Rubio, Rodrigo Caro, Andrés Weintraub, and Kelly Eurek. A review of operations research in mine planning. Interfaces, 40(3):222–245, 2010.

[21] Dónal O'Sullivan and Alexandra Newman. Optimization-based heuristics for underground mine scheduling. European Journal of Operational Research, 241(1):248–259, 2015.

[22] Vidal Félix Navarro Torres, Beck Nader, Carlos Enrique Arroyo Ortiz, Felipe Ribeiro Souza, Hudson Rodrigues Burgarelli, Leonardo Soares Chaves, Luiz Alberto Carvalho, Taís Renata Câmara, Eunírio Zanetti, and Roberto Galery. Classical and stochastic mine planning techniques, state of the art and trends. Mining, 71(2):289–297, 2018.