A heuristic approach for scheduling activities with or-precedence at an underground mine

P. Nancel-Penard[ab]*, N. Morales[ab], V. Rojas[ab] & T. González[ab]

[a]*Delphos Mine Planning Laboratory, Department of Mining Engineering, Faculty of Physical and Mathematics Sciences, University of Chile;* [b]*Advanced Mining Technology Center, University of Chile, Santiago, Chile*

Provide full correspondence details here including e-mail for the *corresponding author

Correspondence address: Delphos Mine Planning Laboratory, Universidad de Chile, Tupper 2069, Santiago, Chile.

e-mail for the corresponding author: pierre.nancel@amtc.cl

ORCID for the corresponding author: 0000-0001-7311-0073

Acknowledgments

The authors would like to thank the Advanced Mining Technology Center and the organizations that funded this research. This work was supported by CONICYT basal Funding Grant AFB180004, and by CORFO under Grant 16CONTEC-67413.

A heuristic approach for scheduling activities with "OR"-precedence constraints at an underground mine

Medium-term development planning of underground mines requires scheduling multiple activities to comply with long-term milestones, and to obtain a time span as short as possible. However, the planning must also respect the availability of construction resources and precedence constraints, which in our case can be *disjunctive*, that is with more than one alternative predecessor.

In this paper, we present an optimization model to find the schedule of minimum length, satisfying all the constraints mentioned. We develop a heuristic approach to solve it and show that it can be used to produce feasible development plans in a real mine.

Keywords: underground mine planning; mixed-integer programming; heuristic; "OR"-precedence constraints; mine development

1. Introduction

Mining development comprises all the activities necessary to establish the infrastructure required to sustain the exploitation of a mineral resource. These activities must be carefully planned in order not to delay mining production or interfere with production activities.

The mine development plans are created by mine planners, who use common criteria and historical data to build these plans. Contrary to other areas of mine planning, they do not have access to methodologies based on well-established optimization tools that help to automate the process. As a result, their plans may not be robust enough, leading to non-compliance with the development plan within the established period for the execution of the mine development during the operation. Therefore, the elaboration of methodologies in this domain would allow planning more efficiently and therefore produce results close to optimal ones. These methodologies would contribute to enhance the mine planning process, by giving the planners more analytical tools that could help to minimize the non-compliance and lead to more optimal use of the resources during the mine development stage.

From a mathematical programming point of view, the planning of mining development can be modeled as a problem of scheduling activities inside a given horizon. These activities are related to each other by precedences, and subject to a series of constraints that can be operational, geotechnical, milestones or deadlines. In this context, activities are all those tasks of the plan that must be executed. Precedence constraints establish the order in which these activities can be performed, and other constraints establish conditions that must be met to find a feasible solution.

The optimization of mine planning has been widely addressed by various authors, both in open-pit and underground mining. However, despite being a crucial issue, the optimization of planning mining development has not received enough attention, and most works focus on the optimization of production. A review of optimization techniques applied to underground mine planning can be found in [1].

In [2], the author proposes a mixed integer programming (MIP) model to schedule the production of 55 stopes over 17 time periods at an Australian copper mine located in Mt Isa, and shows that applying integer programming improves the net present value (NPV) by 23% over a solution obtained manually. In [3], a MIP model is developed to optimize a production schedule that seeks to minimize deviation from monthly planned production quantities of three types of ore products for Kiruna Mine, an iron mine located in Sweden. The paper by [4] also implements a MIP model to underground mine planning. They schedule basic operations at a platinum and palladium mine without providing details of their model. The paper by [5] presents a MIP model for a sub-level stoping operation that improves the computing times of a previous model and permits to solve larger instances.

Other works that extend their scope beyond the classical production optimization issue are the following. For example, [6–8] focus on the transition between open-pit and underground operations, while the paper by [9] shows the interest of integrating both underground and open-pit activities into a unique model. In [10], the authors show that integrating different operation areas of the same underground mine into a unique optimization model leads to better results than considering one zone only at a time. The paper by [11] proposes an integer programming model integrating both "in-mine" recovery by a leaching process and conventional underground operations. The paper by [12] proposes a stochastic integer programming model that incorporates the uncertainty of delays from hang-ups and grades into the production scheduling process of a block caving operation.

In [13], the authors recognize that one of the most relevant distinctions between open-pit and underground operations is that the latter requires more complex precedences. In this regard, the paper by [14] also studies "AND" and "OR" precedences, but in the context of job scheduling, while we focus on mine construction. This paper also proposes some algorithms to check the feasibility of the precedences and to compute early start times of jobs, among others.

It is worth noting that the model in this article is an extension of the model presented in [15], which also considered "AND" and "OR" constraints, but has much simpler resource constraints. It also requires a number of precedence constraints which could be exponentially large in the number of activities.

Finally, another work that is close to ours is presented in [16], even though their focus is on short-term scheduling. There is also a reference schedule that comes from the medium-term, indicating when some activities should be done. They propose to use constraint programming to look for a short-term schedule (i.e. more detailed, with higher time fidelity) such that the deviations from the reference plan are minimized. Apart from the focus on short-term scheduling, their approach is different in terms of technique and goal function and does not consider “OR”-precedence constraints.

In this paper, a methodology is proposed to address the scheduling problem for optimizing an underground mine development plan in the context of a medium-term plan (one year). The objective is to minimize the length of the plan, considering operational, geotechnical, and deadline constraints that require disjunctive precedences to capture the nature of the scheduling problem. The first contribution is that the model is oriented to minimize the time it takes to execute the plan, which is different from most literature, that aims to maximize NPV or minimize costs. Indeed, this approach would suit better the case study where planners are given yearly goals that are not clearly reachable within the planning horizon. It is also the case when plans need to be recomputed during the year to respond to actual performance. The second contribution is the fact that we implement “OR”-precedences that allow starting a certain activity from more than one predecessor without the need to finish them all. “OR”-precedences have the advantage of capturing some flexibility that some underground mining methods have, which is to reach certain locations from different places. The proposed methodology lets the optimization model decide how to build certain tunnels, specifically drives, which can be constructed from both ends simultaneously, hence the meeting point of both excavations is a result of the optimization process. In a standard “AND”-precedence approach, the mine planner would have to decide this in advance, and strongly determining the overall sequence of construction before optimization. A third contribution is the design and application of a heuristic approach that generates good feasible solutions in a very short time span and without the need of any commercial software.

This paper is organized as follows. The problem stated in Section 2, is formulated as a MIP model in Section 3 and for which a heuristic approach is proposed in Section 4. Section 5 presents the case study, and Section 6 presents and discusses the results of the application. Finally, Section 7 contains the conclusions of this research.

2. Problem statement

The problem that we consider is a scheduling problem over a time discretized horizon in time periods $t \in \{1, 2, \ldots, T\}$, each of a length (in days) of L_t. We assume preemptive resources, i.e., an activity can start during a specific period, have some progress, be suspended, and restart during another period. The restarting of activities is done without setup times. Therefore, the problem is to define the time intervals at which the activities progress and how much progress the activities achieve in each period within the time horizon T. It is important to note that an activity is not forced to start or end at the beginning or at the end of a time period, respectively.

The optimal schedule must satisfy different constraints that deal with the duration of the activities, due dates, resource capacities and the precedence between activities. To begin with, we assume that the maximum fraction that an activity can progress is $\bar{V}_i \in (0,1)$. These values are used to model constraints in the progress that do not depend on the availability of resources (for example, the time required for a cemented fill to solidify and dry). Notice that the actual progress of an activity could be smaller than $\bar{V}_i$ if there are not enough resources available, and that the actual maximum progress of an activity can be scaled, depending on the duration of the period.

In terms of due dates, we take a general modeling approach: For each activity i we consider an interval $[MStart_i, MEnd_i]$ of periods and demand that the activity is completely executed within that timeframe.

As it is characteristic of scheduling problems, there is a set $R = \{r\}$ of resources that are needed to execute the activities. Activity i requires a total R_r^i of resource r for its completion. However, it consumes an amount proportional to its progress, i.e., if the activity advances a fraction α at some period, then it consumes αR_r^i of the resource during that period. However, contrary to other cases, instead of having a unique resource availability per period, we extend this notion for intervals. That is, for each resource r we consider a set $J_m = \{1, 2, \dots, |J_m|\}$ of periods $RStart_r^m, REnd_r^m$ for each $m \in J_r$ and constraint that the total consumption of resource r over all the activities and all periods in the interval is at most equal to $\bar{R}_r^m$. This more general approach is useful, for example, to express that the horizontal development of drives of a specific level of the mine between March and June cannot exceed a certain amount of meters over these three months. In this formulation, the notion of resource is extended to a characteristic of the construction of drives (quantity of developed meters) and the consumption of this "resource" corresponds to the meters developed by the corresponding construction activities for this level and 0 for all the other activities. This example comes from the case study, which is explained in Section 5 in details.

We consider two different types of precedence between activities. The first one is the classic "AND"-precedence. An activity cannot start before the end of any of the activities of a given set. In Figure 1(a) where circular nodes represent activities, activities a_1, a_2 and a_3 are "AND"-predecessors of a_4, i.e. all these activities must end before the start of activity a_4. The second one is an "OR"-precedence in which we do not have one set of predecessors, but many. In this case, we require that there must be at least one set of predecessors whose elements are finished before starting the activity being constrained. An example of this is shown in Figure 1(b). All the activities of the set of predecessors

$\{a_1, a_2\}$, **or** all the activities of the set of predecessors $\{a_3, a_4, a_5\}$ have to end before the start of activity a_6.

(a) (b)

Figure 1. (a) An example of "AND"-precedences, where activities a_1, a_2 and a_3 must be finished to start a_4. (b) An example of "OR"-precedences, where either activities a_1 and a_2 must be finished *or* activities a_3, a_4 and a_5 have to be finished to start activity a_6.

To encode these "AND"/"OR"-precedences, let us denote the set of activities as A. The model considers that for each Activity i, there is a set P_i whose elements we call *precedence groups*. Each precedence group consists of several activities, i.e., $P_i = \{P_i^1, P_i^2, \dots, P_i^h\}$ where $P_i^k \subset A$ is a set of activities that are predecessors of i (in particular we assume that $\emptyset \neq P_i^k \subset A - \{i\}$). The precedence constraint then works as follows: i cannot start unless there is at least one k such that all activities in P_i^k have all been finished. If all activities in P_i^k were finished, we say that these activities *enabled* the execution of activity i. In the example of Figure 1(b), we have that $P_{a_6} = \{ P_{a_6}^1, P_{a_6}^2\}$ where $P_{a_6}^1 = \{a_1, a_2\}$ and $P_{a_6}^2 = \{a_3, a_4, a_5\}$. Therefore, in order to start activity a_6 it is sufficient to finish activities a_1 **and** a_2, **or** to finish activities a_3 **and** a_4 **and** a_5. We call all the activities $j \in \bigcup_k P_i^k$ the predecessors of activity k, however, we observe that under this framework, it may be possible that some predecessors of i are not executed (not all activities have enabled i), but that i is executed. In the example of Figure 1(b), if a_1 and

a_2 are finished, then all activities in P_i^1 have been finished and therefore these activities enable i to start, regardless of the state of activities a_3, a_4 and a_5.

In the model used in this article, an activity can start and have some progress at a given time period provided that:

- The precedence constraints are satisfied before or during that time period, and
- There is still some time left by the predecessors and that can be allocated to the activity during the same time period.

For example, if a time period t corresponds to a week, and the last activity that enabled Activity i was finished by Tuesday, then Activity i has still 5 days left to progress.

Another aspect of the model, which is standard in scheduling problems, is the availability and utilization of resources. In our model, the execution of each activity may require the consumption of certain resources, like construction materials or equipment time. We assume that, for each activity and resource, there is a total requirement to complete the execution of the activity and that if an activity progresses a fraction of the total, it requires an amount of the resource which is proportional to that fraction. There may be several resources available, and each activity may need some of them.

3. The mathematical model

The mathematical model that we present can be seen as a variation of the classic Preemptive Resource-Constrained Project Scheduling Problem (PRCPSP). In our model, we compute the fraction of the progress of each activity in each time period and the time consumed by each activity and its predecessors within each time period. From these results and the parameters of the activities, the time intervals to complete the activities can be deduced. Tables 1-3 include the sets, variables, and parameters used in the formulation (1-17).

Table 1. Overview of the sets used for the subscripts and the superscripts in the formulation.

Symbol	**Sets**
$\{1,\dots,T\}$	Set of the time periods for the scheduling
A	Set of the activities to be scheduled
$P_i = \{P_i^k\}_k$	P_i is the set of precedence groups of activity i: $P_i^1, P_i^2, \dots, P_i^{h(i)} \subset A$. All activities in at least one P_i^k have to be completed to proceed with i
$A_R \subset A$	Set of the activities of A without predecessors, i.e., $A_R = \{i \in A: P_i = \emptyset\}$, these are called *root* activities
R	Set of the resources (materials, equipment time, …)
J_r	Set of the indexes of constraints of capacities related to resource *r*

Table 2. Overview of the decision variables.

	Decision variables
$s_{i,t}$	Binary variable denoting the start of activity *i* before or during time period *t*
$e_{i,t}$	Binary variable denoting the end of activity *i* before or during time period *t*
$w_{i,t}^k$	Binary variable denoting the end of all the activities of P_i^k before or during time period *t*
$x_{i,t}$	Continuous variable denoting the fraction of progress of activity *i* in time period *t*
$y_{i,t}$	Continuous variable denoting the time consumed by activity *i* and its enabling ancestors within time period t.

Table 3. Overview of the parameters.

Symbol	Description
L_t	The duration of time period *t*
$\bar{V}_i$	The maximum speed of activity i (percentage per period)
$PStart_i^m$	The first period of the time window of the minimum progress constraint related to activity *i*

$PEnd_i^m$	The last period of the time window of the minimum progress constraint related to activity *i*
R_r^i	The total quantity of the resource *r* consumed by activity *i* inside one time period
$\overline{R}_r^m$	The quantity of resource *r* available over the time window of the mth capacity constraint related to resource *r*
$RStart_r^m$	The first period of the time window of the *m*th constraints of capacities related to resource *r*
$REnd_r^m$	The last period of the time window of the *m*th constraints of capacities related to resource *r*
a_N	An auxiliary activity that is a successor of all other activities of *A*.
C_N	A large positive number associated with the "benefit" of a_N.

The objective function (1) corresponds to the discounted value of activity a_N, which is used as a proxy for total completion time. Notice that an alternative approach would be to consider the execution of a_N as a constraint, however, having this in the goal gives more information in the case of infeasibility, therefore we preferred this approach.

$$\max \textstyle\sum_{t=1}^{T} D_t \, C_N x_{N,t} \qquad (1)$$

Constraint (2) states that if an activity has started during period *t* then it is still considered as started for any posterior period.

$$s_{i,t+1} \geq s_{i,t} \qquad \forall i \in A, \forall t \leq T - 1 \qquad (2)$$

Constraint (3) states that if an activity has ended during period t then it is still considered as ended for any future period.

$$e_{i,t+1} \geq e_{i,t} \qquad \forall i \in A, \forall t \leq T - 1 \qquad (3)$$

Constraint (4) states that activity *i* cannot progress if it has not started.

$$\textstyle\sum_{u \leq t} x_{i,u} \leq s_{i,t} \qquad \forall i \in A, \forall t \leq T \qquad (4)$$

Constraint (5) states that the cumulated progress over the scheduling periods of activity *i* cannot be superior to 1.

$$\sum_{t=1}^{T} x_{i,t} \leq 1 \qquad \forall i \in A, \forall t \leq T \tag{5}$$

Constraint (6) states that while the cumulated progress of activity *i* does not reach the value 1, the activity is not ended.

$$e_{i,t} \leq \sum_{t' \leq t} x_{i,t'} \qquad \forall i \in A, \forall t \leq T \tag{6}$$

Constraint (7) states the minimum progress of activity *i*. A given due date for ending an activity that corresponds to a milestone or a deadline is modeled by adding a constraint with the minimum progress of 1 over the time periods corresponding to the due date.

$$1 \leq \sum_{t=PStart_i}^{PEnd_i} x_{i,t} \qquad \forall i \in A \tag{7}$$

Constraint (8) defines the availability of resource *r*.

$$\sum_{t=RStart_r^m}^{REnd_r^m} \sum_{i \in A} R_r^i \; x_{i,t} \leq \overline{R}_r^m \qquad \forall r \in R, \forall m \in J_r \tag{8}$$

Constraint (9) defines the relationship between the progress and the time consumed by an activity *i* that does not have any predecessor (root activities).

$$x_{i,t} = \bar{V}_i \,.\, y_{i,t} \qquad \forall i \in A_R, \forall t \leq T \tag{9}$$

Constraint (10) states that the variable $w_{i,t}^k$ can be set to one only if the corresponding activities within the set P_i^k have been finished.

$$w_{i,t}^k \leq e_{j,t} \qquad \forall i \in A - A_R, \forall j \in P_i^k, \forall t \leq T \tag{10}$$

Constraint (11) states that if activity i with "OR"-precedence starts at a given time period, then all activities of at least one precedence group have to be ended inside or before the same period.

$$s_{i,t} \leq \sum_k w_{i,t}^k \qquad \forall i \in A - A_R,\ \forall t \leq T \tag{11}$$

Constraint (12) states that the time consumed by activity i and all the activities of the enabling ancestors must not exceed the duration of the corresponding period.

$$y_{i,t} \ \geq \ y_{j,t} \ - \ L_t \ (1 \ - \ w_{i,t}^k \ ) \ + \ x_{i,t} \ /\bar{V}_i \quad \forall i \in A - A_R, \forall j \in \cup_k^{h(i)} P_i^k \ , \forall t \leq T \tag{12}$$

Finally, Constraints (13-17) reflect the nature and state the bounds of the variables.

$$0 \leq \ x_{i,t} \leq \ \min\{1, L_t \cdot \ \bar{V}_i)\} \qquad \forall i \in A, \forall t \leq T \tag{13}$$

$$0 \leq \ y_{i,t} \leq \ L_t \qquad \forall i \in A, \forall t \leq T \tag{14}$$

$$s_{i,t} \in \{0,1\} \qquad \forall i \in A, \forall t \leq T \tag{15}$$

$$e_{i,t} \in \{0,1\} \qquad \forall i \in A, \forall t \leq T \tag{16}$$

$$w_{i,t}^k \in \{0,1\} \qquad \forall i \in A, \forall k \leq h(i), \forall t \leq T \tag{17}$$

4. A heuristic approach to find good solutions

Many scheduling problems similar to ours are known to be hard to solve. For example, the paper by [17] proves that minimizing the total weighted completion time on a single machine for the "OR"-precedence type is NP-hard. Such a model could be reduced to our model to show that finding optimal solutions is NP-Hard. However, we prefer to focus this paper on applied aspects to find good lower bounds, i.e., solutions to the problem.

Apart from the theoretical motivation, the practical utilization of the model is envisioned as an analytical tool to construct a plan evaluating multiple combinations of parameters. Therefore, we develop a heuristic approach to the problem to find good solutions in a very short time and with a small memory footprint.

4.1. Overview of the heuristic approach

The heuristic approach we present in this paper is composed of three different sub-routines. The first subroutine is a Parallel Schedule Generation Scheme (PSGS, see [18] for a description of this heuristic algorithm), which is applied to activities ordered by a topological sort method using expected times, such that the resulting schedule respects the “AND”/“OR”-precedences. To obtain this topological order, we adapt the topological sort presented in [19,20], which is used as part of a heuristic in [21,22] to schedule blocks in open-pit mines with only “AND”-precedences.

The first sub-routine gives a solution that may not respect the due dates expressed in the mathematical model by minimum progress constraints. Hence the second sub-routine is a repair heuristic algorithm that iteratively lowers the expected times of the start of the activities and its predecessors for which a minimum progress constraint is not respected. Then, the PSGS described above is recomputed to obtain another solution to the problem. It iterates while there is at least one minimum progress constraint that is not respected or while a given maximum of attempts is not reached.

Finally, the third sub-routine iteratively enforces activities not fully scheduled in the target horizon, to be scheduled in that target horizon through adding to the current model, minimum progress constraints of 1 over this horizon for these activities. Then, the second sub-routine is applied to respect the maximum number of those constraints.

4.2. Extended Topological Sort Algorithm

In this section, we introduce the Extended Topological Sort Algorithm (ETSA) which generates a feasible activity schedule from a fractional solution of the relaxed problem. The first step of the first subroutine computes the solution to the integer relaxation of the mathematical problem presented in Section 3 to obtain a fractional solution $(\hat{s}_{i,t}, \hat{x}_{i,t}, \hat{e}_{i,t}, \widehat{w}_{i,t}^{k}, \hat{y}_{i,t})$. The expected times of the start of the activities are computed using Equation (18):

$$\mathrm{ET}_{i,start} = \quad (\mathrm{T} + 1) \cdot \left(1 - \sum_{t=1}^{T} \hat{x}_{i,t}\right) + \mathrm{T} + 1 - \sum_{t=1}^{T} \hat{s}_{i,t} \qquad (18)$$

The topological sort algorithm needs to work on an acyclic graph, but the graph constructed with the “AND”/“OR”-precedences for underground operations may not be acyclic. The paper by [14] shows that feasibility, as well as many questions related to the transitivity of “AND”/“OR”-precedence constraint problems, can be solved by applying nearly the same linear time algorithms as for “AND”-precedences. Then, we adapt the topological sort presented in [19,20] with only “OR”-precedence to the considered “AND”/“OR”-precedence framework.

Considering a topological ordering of the activities $\{a_1, \ldots, a_n\}$, the ordering is feasible for the considered precedences P_i if all the constraints of precedence defined in the mathematical model of Section 3 are respected. Defining G = *(V,E)* as the directed graph for which each node of *E* is an activity, $\delta^{-}(i)$ as the incoming arcs to the node that corresponds to Activity *i*, each Arc (j, i) of G corresponds to an “AND”-precedence such that Activity *j* is an “AND”-predecessor of activity *i* and $j \in P_i^1$ and P_i^1 is the only precedence group for activity *i* ($|P_i| = 1$). Defining W_i as the set of the precedences groups for activity *i* only if there is more than one precedence group ($|P_i| > 1$), we present the “Extended Topological Sort Algorithm” (ETSA) in Figure (2).

In this algorithm, the set Q is initialized with activities that do not have any "OR"-predecessor. This set is iteratively updated each time a node is taken away from the graph, adding nodes that respect at least one precedence group. At each iteration, the node that does not have any incoming arc (i.e. without any "AND"-predecessor) with a minimum expected start time is inserted into the topological ordering.

Algorithm 1: ETSA (Extended Topological Sort Algorithm)

Result: A topological feasible ordering of vertices v_1 ... v_n in G.
Input: $G = (V,E)$ an acyclic directed graph, $W = (W_i)$ a set of precedence groups, $w = (\mathrm{ET}_{i,start})$ the vector of the expected start time of the activities
Output: v_1 ... v_n

```
i ← 1
Q ← ∅
For u ∈ V Do
  | If |{ (P_u^k) ∈ W_u }| = 0 Then
  |   | Q ← Q ∪ {u}
End
While Q ≠ ∅ do
  | j ← Argmin(w_j : j ∈ Q | δ⁻(j) = ∅)
  | G ← G \ {j}      (remove j from G and all arcs outgoing from j)
  | Q ← Q \ {j}
  | For P_i^k ∈ W_i | j ∈ P_i^k do
  |   | P_i^k ← P_i^k / j     (remove j from P_i^k ⊂ W_i ⊂ W)
  | For u ∈ V do
  |   | If ∃k | (P_u^k) = ∅ then
  |   |   | Q ← Q ∪ {u}
  | v_i ← j
  | i ← i + 1
End
Return (v_i)
```

Figure 2. Extended Topological Sort Algorithm.

5. Case study

The mathematical model and the heuristic approach were applied to the mining development plan of a copper underground mining operation located in Chile that is part of the El Teniente mining complex. The four levels of a Panel Caving operation: sinking, production, ventilation, and haulage, in addition to the ore pass systems are considered (see Figure 3) for scheduling.

The construction sequence of a panel caving operation can be observed in Figure 4. The base plan used in this case study is a monthly plan within the horizon of one year. The construction sequence aims to deliver the volumes of works considered during the annual period, the growth guidelines for each sector, and the monthly requirements for the incorporation of the area. It also incorporates all the milestones of mining development to ensure sustainability and continuity of production. Table 4 gives the maximum speed for the activities of the base plan at the production level.

Table 4. Overview of the maximum speed for the activities of the base plan at the production level.

Activity	Unity	Quantity
Horizontal development	Meters/month	245
Vertical development	Meters/month	95
Road surface	Meters/month	125
Construction of draw point	Unity/month	10
Construction of draw point brow	Unity/month	10
Construction of draw point ground	Meters/month	10
Construction of retaining wall	Unity/month	9
Crossing reinforcement	Unity/month	6
Drilling for hydraulic fracturing	Meters/month	300
Hydraulic fracturing	Unity/month	3
Drainage hole	Meters/month	200
Special reinforcements	Meters/month	65

The base plan also indicates when some of the main milestones have to be developed. Some of the milestones deal with the Interaction Zone (IZ) that is related to the fragmentation of the considered zone as well as those that come from other parts of the mine.

All the original activities presented in the base plan were included. A discretization of the activities that are extensive in mining development was carried out, to avoid a unique advancing front. For example, an activity of horizontal development of 100 meters was discretized or divided into five smaller activities of 20 meters each. The discretization for the modeling was made based on the discretization used by the mine planners of the operation for the construction of the plan, which complied with the operational requirements of the mine. It is important to note that there is a contractor change in July for Undercut and Production levels, with the consequence that none of the activities of horizontal development for these levels can be planned during this month.

Historical data of construction were taking into account to generate the upper bound of some resource constraints presented in Section 2 that are part of constraints (8) of the mathematical model described in Section 3. The objective of these constraints is to model that the developed meters cannot exceed a certain amount of meters over a defined set of periods.

To ensure that there is a starting activity that is not inside a cycle of “OR”-precedences, we add a dummy activity that is the predecessor of some of the activities inside an “OR”-precedence cycle. Another dummy activity that is the “AND”-successor of all the other activities was also added to fit the mathematical model presented in Section 3.

The implementation of the base plan into the mathematical model generates 1,442 activities, 2,682 precedence constraints. 1,834 are “AND”-precedence constraints, where there is only one set of predecessors for an activity and so that all need to be finished before that activity starts. 848 are “OR”-precedence constraints, where there is more than one set of predecessors that enables the execution of the activity. There are also 1,392 other constraints.

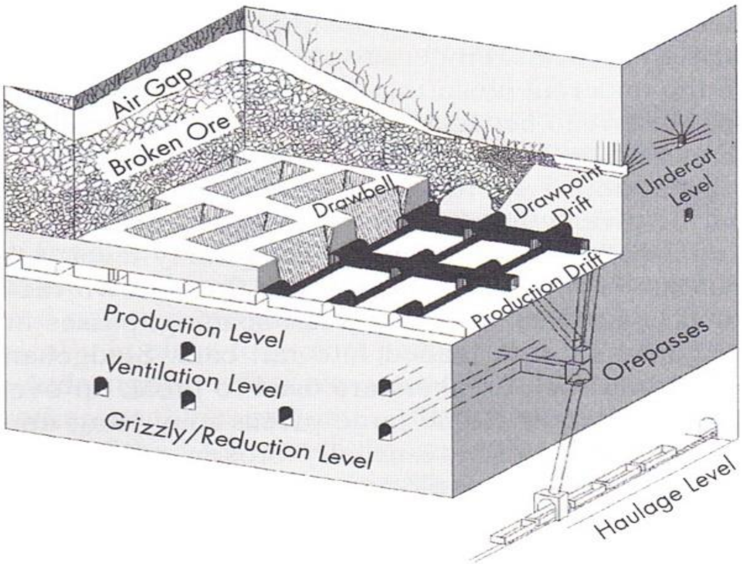


Figure 3. Post or conventional undercutting from [23].

6. Results and Analysis

In this section, we present the results obtained using this model and provide several analyses. First, a brief comparison in terms of performance, with regards to utilizing a commercial software to solve the problem. Second, we describe the solution obtained by showing some parts of the development plan. Third, we compare the plan obtained by the model with a base plan, generated independently by the mine planners.

6.1. Comparison with a commercial software

To implement the heuristic approach, we used the UDESS library developed at Delphos Mine Planning Laboratory at Universidad de Chile [24], which implements data structures to store the activities, "AND"/"OR"-precedences and the other constraints. This library also implements the mixed-integer program presented in Section 3. The code of the heuristic approach is written in C++, but the module also provides a wrapper to use it from Python. This code calls the coin-or Clp that is an open-source solver through the coin-or/Cbc library version 2.9 [25] to solve the linear programming relaxation of the mixed-integer program of Section 3. This implementation obtained a feasible schedule with all the activities of the base plan scheduled within 11.84 months, with a run time of 108 seconds on a notebook Intel core i7 2.60 GHz with 8 GB of RAM.

We also utilized the Gurobi version 8.0 software for computing the mixed-integer program of Section 3. Gurobi found the same result for the mixed-integer program with a relative MIP gap of 0% as the one found by the heuristic approach, the solution obtained by the heuristic is optimal.

6.2. Comparison of development plan obtained by the model and base case

Most of the activities are developed in undercut and production levels. Thus, for the sake of the length of the paper, we focus on the results corresponding to these levels, as they provide a good representation of the overall outcome for the rest of the mine. The activities with the most volume of work in both levels correspond to horizontal development. As shown in Figure 4, the mathematical model can schedule all the plan activities leaving a small volume of activities to be carried out towards the final periods. The results for the rest of the activities at all levels are similar. All the activities of the development plan are scheduled within the 12-month horizon, respecting all constraints and leaving more time available towards the final periods.

Figure 4. Horizontal development plan for undercut and production levels for the base development plan and the heuristic approach result.

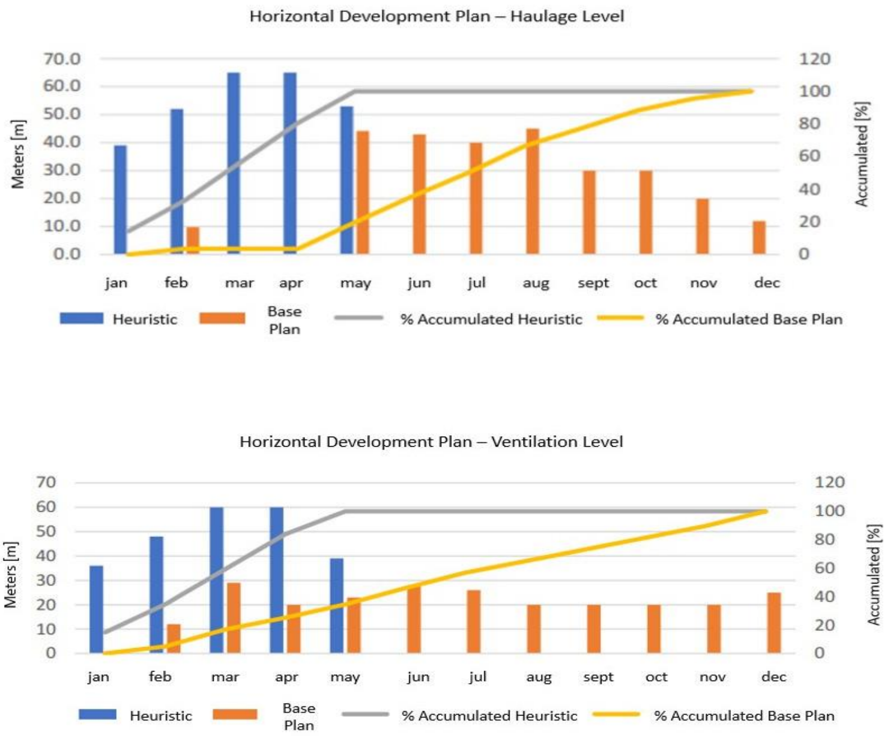
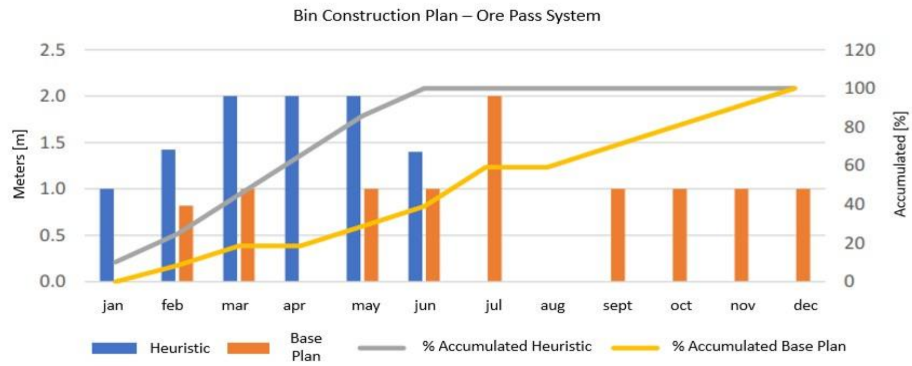


Figure 5. Horizontal development plan for haulage and ventilation levels for the base development plan and the heuristic approach result



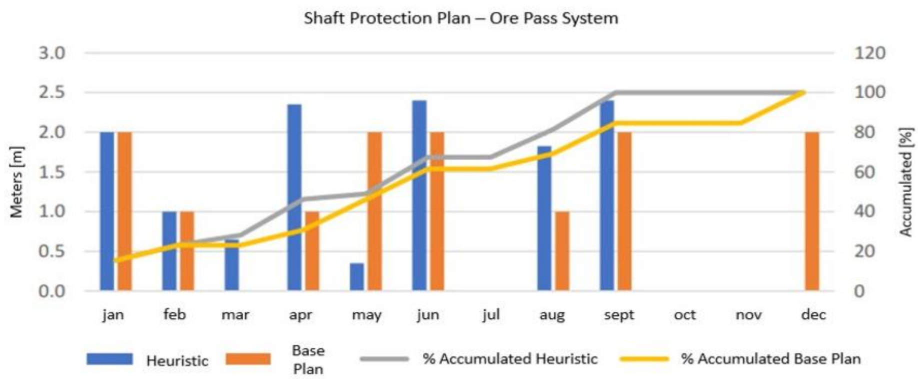


Figure 6. Bin construction and shaft protection plans in the ore pass system for the base development plan and the heuristic approach result.

In Figure 5, a similar distribution of jobs is observed along the time horizon for undercut and production levels, which does not occur in the plans of haulage and ventilation levels, as seen in Figure 6. The availability of resources by level, giving priority to higher impact works such as ore pass system ones may explain the differences.

In the base plan, the works of the ore pass systems are scheduled for the second semester, while the plan generated by the proposed heuristic approach tends to do this work in advance towards the first semester, as seen in Figure 6. A large part of the activities of the ore pass systems does not have any precedences, and the fact that these activities are not limited in terms of resources permits to bring them forward.

Another very significant point of using the model is that the schedule given by the proposed heuristic approach fulfills all the milestones required (100% compliance), while the base plan was only not able to meet the established deadlines for 3 of the 19 required milestones (84% of compliance).

7. Conclusions

We propose an approach for generating medium-term development plans in underground mining. Our approach relies on a mathematical model and a heuristic algorithm that optimizes the length of the development plan, generating operationally feasible solutions. Most of the commercial scheduling software (or even models present in the literature) only allow “AND”-precedence constraints; thus, the generated plans are more rigid in terms of possible outcomes. The proposed methodology provides greater flexibility to the activity schedule by incorporating “OR”-precedence, allowing the generation of plans that are closer to the best possible operational ones.

Another advantage of the proposed methodology is the computation time for the generation of plans (about two minutes, without the need of a commercial software), which allows sensitivity analysis and also, for the case study, the possibility of recalculating plans during its execution through the year.

When compared to a reference plan generated using the regular methodology at the mine, the mathematical optimization redistributes the activities, bringing forward some activities when precedences and resources allow it. Both plans scheduled all the activities within the established maximum period of 12 months. However, the dates of completion for milestones were not the same, the plan proposed by the heuristic approach complied with 100% of the established due dates, while the base plan did not. The result obtained by the heuristic approach corresponds to a 16% improvement in the compliance of the milestones of the base plan.

As a future direction of research, it would be appropriate to adapt the earliest start times compute algorithm proposed in [14] to the considered problem, to reduce the feasible domain of some of the variables. Stochastic versions of this problem is also a very relevant research line.

References

[1] C. Musingwini, *Presidential address: Optimization in underground mine planning-developments and opportunities*, J. South. African Inst. Min. Metall. 116 (2016), pp. 809-820.

[2] L. Trout, Underground mine production scheduling using mixed integer programming, in 25th International APCOM symposium, 1995, pp. 395-400.

[3] M. Kuchta, A. Newman and E. Topal, *Implementing a Production Schedule at LKAB's Kiruna Mine*, Interfaces (Providence). 34 (2004), pp. 124-134.

[4] W.M. Carlyle and B.C. Eaves, *Underground planning at Stillwater Mining Company*, Interfaces (Providence). 31 (2001), pp. 50-60.

[5] M. Nehring, E. Topal and J. Little, *A new mathematical programming model for production schedule optimization in underground mining operations*, J. South. African Inst. Min. Metall. 110 (2010), pp. 437-446.

[6] J. Chung, E. Topal and A.K. Ghosh, *Where to make the transition from open-pit to underground? Using integer programming*, J. South. African Inst. Min. Metall. 116 (2016), pp. 801-808.

[7] B. King, M. Goycoolea and A. Newman, *Optimizing the open pit-to-underground mining transition*, Eur. J. Oper. Res. 257 (2017), pp. 297-309.

[8] D. Whittle, M. Brazil, P.A. Grossman, J.H. Rubinstein and D.A. Thomas, *Combined optimisation of an open-pit mine outline and the transition depth to underground mining*, Eur. J. Oper. Res. 268 (2018), pp. 624-634.

[9] R. Epstein, M. Goic, A. Weintraub, J. Catalán, P. Santibáñez, R. Urrutia et al., *Optimizing Long-Term Production Plans in Underground and Open-Pit Copper Mines,* Oper. Res. 60 (2012), pp. 4-17.

[10] J. Little, P. Knights and E. Topal, *Integrated optimization of underground mine design and scheduling*, J. South. African Inst. Min. Metall. 113 (2013), pp. 775-785.

[11] A. Mousavi and E. Sellers, *Optimisation of production planning for an innovative hybrid underground mining method*, Resour. Policy 62 (2019), pp. 184-192.

[12] R. Dirkx, V. Kazakidis and R. Dimitrakopoulos, *Stochastic optimisation of long-term block cave scheduling with hang-up and grade uncertainty*, Int. J. Mining, Reclam. Environ. 33 (2019), pp. 371-388.

[13] A. O´Sullivan, D; Brickey, A; Newman, *Is openpit production scheduling "easier" than its underground counterpart?*, Min. Eng. 67 (2015), pp. 68-73.

[14] R.H. Möhring, M. Skutella and F. Stork, *Scheduling with AND/OR Precedence Constraints*, SIAM J. Comput. 33 (2004), pp. 393-415.

[15] W. Rocher, N. Morales and E. Rubio, Eight-Dimensional Planning: Construction of an Integrated Model for the Mine Planning Involving Constructability, in 35th International APCOM symposium, 2011, pp. 393-406.

[16] L.P. Campeau and M. Gamache, *Short-term planning optimization model for underground mines*, Comput. Oper. Res. (2019). Available at https://www.sciencedirect.com/science/article/pii/S0305054819300309.

[17] B. Johannes, *On the complexity of scheduling unit-time jobs with OR-precedence constraints*, Oper. Res. Lett. 33 (2005), pp. 587-596.

[18] R. Kolisch and S. Hartmann, *Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*, in *Project Scheduling. International Series in Operations Research & Management Science*, *14*, W. J., ed., Springer, Boston, MA, 1999, pp. 147-178.

[19] M. Gershon, *Heuristic approaches for mine planning and production scheduling*, Int. J. Min. Geol. Eng. 5 (1987), pp. 1-13.

[20] R. Chicoisne, M. Goycoolea, E. Moreno and E. Rubio, *A new algorithm for the open-pit mine scheduling problem*, Min. Technol. 60 (2009), pp. 1-21.

[21] M. Samavati, D. Essam, M. Nehring and R. Sarker, *A new methodology for the open-pit mine production scheduling problem*, Omega 81 (2018), pp. 169-182.

[22] E. Jélvez, N. Morales, P. Nancel-Penard and F. Cornillier, *A new hybrid heuristic algorithm for the Precedence Constrained Production Scheduling Problem: A mining application*, Omega (United Kingdom) (2019). Available at https://www.sciencedirect.com/science/article/pii/S0305048318301750.

[23] P. Darling, *SME Mining Engineering Handbook*, 3rd ed.,Society for Mining, Metallurgy, and Exploration, 2011.

[24] N. Morales, D. Mancilla, and C. Moreno, UDESS lib. 2.0, reg. 293.951,

Universidad de Chile, 2016. Description of the software available at http://delphoslab.cl/.

[25] J. Forrest, T. Ralphs, S. Vigerske, L. Hafer, B. Kristjansson, Jpfasano, E. Straver, M. Lubin, H. Santos, Rlougee, M. Saltzmann, coin-or/Cbc: Version 2.9.9, 2019. Available online at https://dx.doi.org/10.5281/zenodo.1317566.