Short-Term Scheduling of an Open-Pit Mine with Multiple Objectives

Online Supplementary

Michelle Blom*, Adrian R. Pearce, Peter J. Stuckey
Department of Computing and Information Systems, University of Melbourne

This document presents material supplementary to the companion paper *Short-Term Scheduling of an Open-Pit Mine with Multiple Objectives*. In this companion paper, a novel Concurrent Rolling Horizon-based (CRH) algorithm for the generation of multiple short-term production schedules for an open-pit mine is presented, in which several objectives, of varying priority, characterize the quality of each solution. A short-term schedule selects regions of a mine site, known as ‘blocks’, to be extracted in each week of a planning horizon (typically spanning 13 weeks). The companion paper presents a tool in which multiple, diverse, short-term schedules are constructed, meeting a range of common objectives. This supplementary document presents formal definitions of a range of objectives supported by this solver, and practical advice on how the CRH algorithm can be efficiently implemented.

Keywords: short-term open-pit mine production scheduling; multi-objective optimization; mixed-integer programming; decision support

Notation used in this document is defined in Section 1. All references to material in the companion paper are prefixed with CP.

1. Notation

Indices, Sets, and Parameters

$t \in \mathcal{T}$	Index over time period-aggregates in the discretized horizon $\mathcal{T}$.
$b \in \mathcal{B}$	Set of blocks in the block model of the mine.
$l \in \mathcal{L}$	Set of modeled material types (e.g., high grade, low grade, waste).
$l \in \mathcal{W}$	Set of material types considered to be waste.
$s \in \mathcal{S}$	Set of stockpiles available at the mine.
$s \in \mathcal{S}^W$	Set of waste dumps at the mine.
$p \in \mathcal{P}$	Set of plants (processing options) available at the mine.
$q \in \mathcal{Q}$	Attributes defining the composition of ore and waste.
$\tau \in \mathcal{R}^k$	Set of truck types operating at the mine site.
$k \in \mathcal{A}_b$	Each k is a group of blocks that, if removed, will provide access to block $b \in \mathcal{B}$. There can be <i>multiple</i> ways of accessing b , by removing <i>different</i> groups of blocks ($\mathcal{A}_b$ defines the different ways in which b can be accessed).
$j \in \hat{\mathcal{B}}$	Each j is a group of blocks, each group defining a region of the mine site.
G_{blqp}	Fractional percentage of $q \in \mathcal{Q}$ in the material of type $l \in \mathcal{L}$ in $b \in \mathcal{B}$ <i>after</i> it has been through processing in plant $p \in \mathcal{P}$.

*Corresponding author. Email: michelle.blom@unimelb.edu.au

G_{sq1p}	Fractional percentage of attribute $q \in \mathcal{Q}$ in stockpile $s \in \mathcal{S}$ ($t = 1$) <i>after</i> it has been through processing in plant $p \in \mathcal{P}$.
$\underline{L}_{qt}$	Lower bound on the level of attribute $q \in \mathcal{Q}$ in ore produced in $t \in \mathcal{T}$.
$\bar{L}_{qt}$	Upper bound on the level of attribute $q \in \mathcal{Q}$ in ore produced in $t \in \mathcal{T}$.
$\underline{W}_t$	Lower bound on the tons of waste mined in $t \in \mathcal{T}$.
$\bar{W}_t$	Upper bound on the tons of waste mined in $t \in \mathcal{T}$.
$\underline{E}_{jt}$	Lower bound on the tons of material mined from region $j \in \hat{\mathcal{B}}$.
$\bar{E}_{jt}$	Upper bound on the tons of material mined from region $j \in \hat{\mathcal{B}}$.

Functions

$size(t)$	Returns the number of time periods in period-aggregate $t \in \mathcal{T}$.
$blocks(j)$	Returns the set of blocks in region $j \in \hat{\mathcal{B}}$ of the mine.
$blocks(k)$	Returns the set of blocks in group $k \in \mathcal{A}_b$ for some $b \in \mathcal{B}$.
$nbors(\mathcal{B}')$	Returns the set of blocks outside of $\mathcal{B}' \subset \mathcal{B}$ that are a dependency (according to mining precedences) of some block inside $\mathcal{B}'$.

Decision Variables

$x_{bl dt}$	Tons of material $l \in \mathcal{L}$ in block $b \in \mathcal{B}$ hauled to destination $d \in \mathcal{S} \cup \mathcal{P}$ in $t \in \mathcal{T}$.
x_{sprt}	Tons of stockpile $s \in \mathcal{S}$ hauled to plant $p \in \mathcal{P}$ with truck type $\tau \in \mathcal{R}^k$ in $t \in \mathcal{T}$.
y_{bt}	Binary of value 1 iff block $b \in \mathcal{B}$ is mined by the end of $t \in \mathcal{T}$.
y_{blt}	Binary of value 1 iff material $l \in \mathcal{L}$, from $b \in \mathcal{B}$, is mined by the end of $t \in \mathcal{T}$.
y_{kt}	Binary of value 1 iff the block group $k \in \bigcup_{b \in \mathcal{B}} \mathcal{A}_b$ has been completely extracted by the end of $t \in \mathcal{T}$.
a_{st}	Tons of material on stockpile $s \in \mathcal{S}$ at the end of $t \in \mathcal{T}$.
h_{pt}	Tons of material sent to plant $p \in \mathcal{P}$ in $t \in \mathcal{T}$.
ρ_{qt}	Tons of attribute $q \in \mathcal{Q}$ produced by the mine in $t \in \mathcal{T}$.

Algorithmic Notation

$\mathbf{x}$	Set of all decision variables (defined above) characterizing a schedule.
$M_{\mathcal{T}}$	Instantiation of MIP_{STP} over a discretized horizon $\mathcal{T}$.
α_s, α_b	Split and branch factor used to control the number of schedules generated by the CRH algorithm (described in Section CP-5).

2. Supported Objectives

Each objective z supported by the CRH solver is defined in terms of an expression to be optimized $\xi(z)$, and a set of associated prune constraints $PR(z)$. The CRH solver supports over 40 objectives, including: maximizing the utilization of trucks; maintaining desired material movement between selected regions and stockpiles; maintaining bounds on the tons of waste mined in selected regions; and maintaining the grade of stockpile and plant feed within desired bounds. In Section CP-6, the CRH solver is demonstrated in a range of optimization scenarios, composed of a variety of commonly optimized objectives. These objectives are formally defined in the following sections.

2.1. Maintaining Alignment with a Longer-Term Plan (z_{align})

A longer-term plan identifies the blocks $\mathcal{B}_{LTP} \subset \mathcal{B}$ that should be extracted by the end of the short-term planning horizon. This objective aims to maximize the tons of material extracted from blocks in this set.

$$\xi(z_{align}) \equiv \max \sum_{b \in \mathcal{B}_{LTP}} \sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bl dt} \quad (1)$$

The prune constraints for z_{align} are shown in Equation (2). Let ν_{LTP} denote the tons mined from blocks in $\mathcal{B}_{LTP}$ in the best found solution to MIP_{STP} instantiated with z_{align} .

$$PR(z_{align}) \equiv \sum_{b \in \mathcal{B}_{LTP}} \sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bl dt} \geq \nu_{LTP} \quad (2)$$

2.2. Maintaining Stockpile Levels within Desired Bounds (z_{piles})

This objective is designed to keep the tons on each stockpile $s \in \mathcal{S}$ at the end of each period within desired lower and upper bounds (denoted $\underline{S}_s$ and $\bar{S}_s$, respectively).

The activities of later periods are more likely to change, prior to execution, due to the dynamic nature of the mining environment. Consequently, the performance of earlier periods against objectives should be prioritized. By dividing the contribution of stockpile size deviations arising in each period-aggregate $t \in \mathcal{T}$ by the number of spanned periods $size(t)$, we discount the influence of the future on the value of the objective.

$$\xi(z_{piles}) \equiv \min \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \frac{1}{size(t)} (\max(0, a_{st} - \bar{S}_s) + \max(0, \underline{S}_s - a_{st})) \quad (3)$$

Let ν_{st} denote the size of stockpile $s \in \mathcal{S}$ at the end of period-aggregate $t \in \mathcal{T}$ in the best found solution to MIP_{STP} , instantiated with $\xi(z_{piles})$.

$$PR(z_{piles}) \equiv \min(\underline{S}_s, \nu_{st}) \leq a_{st} \leq \max(\bar{S}_s, \nu_{st}) \quad \forall t \in \mathcal{T}, s \in \mathcal{S} \quad (4)$$

Note that the $\min(\cdot)$ and $\max(\cdot)$ expressions that appear above are linearized when added to MIP_{STP} . The CRH solver supports variations of z_{piles} in which only deviation in stockpile levels below desired lower bounds, or above upper bounds, are considered.

2.3. Maintaining Product Grades within Desired Bounds (z_{grade})

This objective aims to keep the levels of each attribute $q \in \mathcal{Q}$ in the ore produced by the mine in each period $t \in \mathcal{T}$ within desired lower ($\underline{L}_{qt}$) and upper ($\bar{L}_{qt}$) bounds. Recall that h_{pt} and ρ_{qt} denote the tons of material sent to plant $p \in \mathcal{P}$ in t , and the tons of attribute q produced by the mine in t , respectively.

$$\xi(z_{grade}) \equiv \min \psi_{grade} \quad (5)$$

where:

$$\psi_{grade} = \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, \rho_{qt} - \bar{L}_{qt} \sum_{p \in \mathcal{P}} Y_p h_{pt}) \quad (6)$$

$$+ \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, L_{qt} \sum_{p \in \mathcal{P}} Y_p h_{pt} - \rho_{qt})$$

$$h_{pt} = \sum_{\tau \in \mathcal{R}^k} \sum_{s \in \mathcal{S}} x_{s\tau t} + \sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}} x_{blpt} \quad \forall t \in \mathcal{T}, p \in \mathcal{P}, \quad (7)$$

$$\rho_{qt} = \sum_{p \in \mathcal{P}} Y_p \left[\sum_{\tau \in \mathcal{R}^k} \sum_{s \in \mathcal{S}} G_{sq1p} x_{s\tau t} + \sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}} G_{blqp} x_{blpt} \right] \quad \forall t \in \mathcal{T}, q \in \mathcal{Q} \quad (8)$$

Let ν denote the value of ψ_{grade} in the best found solution to MIP_{STP} , instantiated with $\xi(z_{grade})$. The prune constraints for z_{grade} are shown in Equation (9).

$$PR(z_{grade}) \equiv \psi_{grade} \leq \nu \quad (9)$$

Several variations of z_{grade} are supported in which grade deviation above desired upper bounds, below desired lower bounds, or from the original bounds narrowed by one half, and three quarters, are minimized. Also supported are objectives that minimize deviation between the level of each $q \in \mathcal{Q}$, individually, and its original or narrowed bounds.

2.4. Encouraging the Continuous Mining of Blocks (z_{cont})

Continuous mining encourages the mining of blocks without interruption, until completely extracted. Given a rolling horizon spanning periods t' to T , where t' is the first period, $OPEN_{t'}$ denotes the set of blocks left in a partially mined state at the start of period t' . Equation 10 maximizes the tons of these blocks mined during t' .

$$\xi(z_{cont}) \equiv \max \sum_{b \in OPEN_{t'}} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bldt'} \quad (10)$$

Let ν_b denote the tons of $b \in \mathcal{B}$ mined in period t' of the best found solution to MIP_{STP} , instantiated with $\xi(z_{cont})$. Constraint (11) prunes from the solution space of MIP_{STP} , all solutions in which less than ν_b tons of each $b \in OPEN_{t'}$ are mined in period t' .

$$PR(z_{cont}) \equiv \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bldt'} \geq \nu_b \quad \forall b \in OPEN_{t'} \quad (11)$$

2.5. Mining Appropriate Quantities of Waste (z_{waste})

This objective aims to keep the tons of waste mined in each period $t \in \mathcal{T}$ within desired lower and upper bounds (denoted $\bar{W}_t$ and $\bar{W}_t$, respectively). The set $\mathcal{S}^W$ denotes the subset of stockpiles $\mathcal{S}$ that are waste dumps, and $\mathcal{W}$ the subset of material types $\mathcal{L}$ that are waste. The prune constraints $PR(z_{waste})$ take the form shown in Equation (4).

$$\begin{aligned} \xi(z_{waste}) \equiv \min & \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, W_t - \sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{W}} \sum_{d \in \mathcal{S}^W} x_{bldt}) \\ & + \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, \sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{W}} \sum_{d \in \mathcal{S}^W} x_{bldt} - \bar{W}_t) \end{aligned} \quad (12)$$

2.6. Restricting Mining in Specific Regions ($z_{regions}$)

This objective aims to keep the tons of material mined from specific regions of the mine site within desired bounds. Let $\hat{\mathcal{B}}$ denote the set of regions with such restrictions, and $\underline{E}_{jt}$ and $\bar{E}_{jt}$ a lower and upper bound on the material mined from region $j \in \hat{\mathcal{B}}$ in period $t \in \mathcal{T}$.

$$\begin{aligned} \xi(z_{regions}) \equiv \min & \sum_{j \in \hat{\mathcal{B}}} \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, \underline{E}_{jt} - \sum_{b \in blocks(j)} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bldt}) \\ & + \sum_{j \in \hat{\mathcal{B}}} \sum_{t \in \mathcal{T}} \frac{1}{size(t)} \max(0, \sum_{b \in blocks(j)} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bldt} - \bar{E}_{jt}) \end{aligned} \quad (13)$$

Let ν_{jt} denote the tons of material mined from region $j \in \hat{\mathcal{B}}$ in period $t \in \mathcal{T}$ of the best found solution to MIP_{STP} instantiated with objective $z_{regions}$.

$$PR(z_{regions}) \equiv \min(\underline{E}_{jt}, \nu_{jt}) \leq \sum_{b \in blocks(j)} \sum_{l \in \mathcal{L}} \sum_{d \in \mathcal{S} \cup \mathcal{P}} x_{bldt} \leq \max(\bar{E}_{jt}, \nu_{jt}) \quad \forall t \in \mathcal{T}, j \in \hat{\mathcal{B}} \quad (14)$$

3. Implementation Details

This section describes a range of modeling choices that have been made and optimizations that have been used, in the implementation of the CRH solver.

Recall that each block $b \in \mathcal{B}$ has a set of dependencies, $\mathcal{A}_b$, where each dependency is a group of blocks. To schedule the mining of b in period t , one of the block groups in $\mathcal{A}_b$ must be mined completely by the end of t . As a group of blocks can act as a dependency for more than one block, a *single* binary variable y_{kt} is created for each distinct group of blocks k for which there exists a $b \in \mathcal{B}$ such that $k \in \mathcal{A}_b$. Constraints CP-12 to CP-13 define these variables, and map each y_{kt} to the blocks for which block group k is a dependency.

Each instance of MIP_{STP} constructed by the CRH algorithm of Section CP-5 can be solved more efficiently if care is taken to avoid the creation of variables that cannot take on a value other than zero in any feasible schedule. For example, the set of permitted destinations for material mined from a block is dependent on the material type $l \in \mathcal{L}$ and location of the block. Truck and dig unit types will be restricted from being used in specific regions of the mine site, and plants will accept material from only a subset of stockpiles.

Recall that constraint set CP-14 in MIP_{STP} (Section CP-3) ensures that, in any $t \in \mathcal{T}$, all contiguous collections of blocks $\mathcal{B}' \subset \mathcal{B}$ scheduled for mining, where each $\mathcal{B}'$ is maximal with respect to set inclusion, contain at least one block with no dependencies ($\exists b \in \mathcal{B}' . \mathcal{A}_b = \emptyset$). Adding this constraint set to MIP_{STP} would cause a combinatorial explosion in its size. Instead, selected instances of constraint set CP-14 are added to the MIP lazily, following

Figure 1 A set of blocks $b_0, \dots, b_9$, and their precedences. For example, block b_0 can be accessed by mining one of blocks b_1 , b_2 , b_3 , or b_4 . Similarly, block b_7 can be accessed by either mining block b_4 or block b_9 .

the process shown in Listing 1. For each new incumbent solution, and each $t \in \mathcal{T}$, the set of all (maximal with respect to set inclusion) contiguous collections of blocks, among those scheduled for extraction, that *do not* contain a block without dependencies, is computed (Step 3). This set is labeled $\mathcal{B}^2$. Consider the scenario in Figure 1, in which the shaded blocks have been scheduled for mining. In this incumbent solution, blocks b_0 and b_4 have been mined, yet neither are immediately accessible. The immediately accessible blocks are b_5 , b_6 , b_8 , and b_9 . Here, $\mathcal{B}^2 = \{\{b_0, b_4\}\}$.

For each $\mathcal{B}' \in \mathcal{B}^2$, all dependencies $k \in \mathcal{A}_b$, for some $b \in \mathcal{B}'$, which have not been completely extracted by the end of t are computed (Step 6). This set is labeled $\mathcal{A}'$. In the example of Figure 1, $\mathcal{A}' = \{\{b_2\}, \{b_1\}, \{b_3\}, \{b_7\}\}$. A set $\mathcal{U}$ is initialized with the blocks in set $\mathcal{B}'$. A constraint is added to MIP_{STP} stating that if a block in $\mathcal{U}$ is mined, then y_{kt} , for some $k \in \mathcal{A}'$, must take on a value of 1 (Step 8). In the example of Figure 1, a constraint is added to MIP_{STP} requiring one of b_2 , b_1 , b_3 , and b_7 to be mined if either b_0 or b_4 are mined.

If the union of the block groups in $\mathcal{A}'$ does not contain a block without dependencies, further constraints can be added to MIP_{STP} by repeating Steps (7)–(11) of Listing 1. The set $\mathcal{U}$ is expanded to contain all blocks in the union of block groups in $\mathcal{A}'$ (Step 9). In the example of Figure 1, blocks b_2 , b_1 , b_3 , and b_7 are added to $\mathcal{U}$. The set $\mathcal{A}'$ is then recalculated (Step 10). Here, $\mathcal{A}' = \{\{b_5\}, \{b_6\}, \{b_8\}, \{b_9\}\}$. In the next execution of Step 8, a constraint is added to MIP_{STP} requiring one of b_5 , b_6 , b_8 , or b_9 to be mined if any of blocks $b_{0-4,7}$ are mined. Repeating Steps (7)–(11) reduces solve times, in general.

The constraints (or cuts) added in the repeated execution of Steps (7)–(11) are labeled *ring cuts*. Table 1 reports the time required, by the CRH solver, to generate schedules in scenarios $DS1_{a-e}$ and $DS2_{a-e}$, when repeating Steps (7)–(11) for each $\mathcal{B}' \in \mathcal{B}^2$ (*with ring cuts*) and when executing these steps only once for each $\mathcal{B}' \in \mathcal{B}^2$ (*without ring cuts*).

The addition of some of the instances of constraint set CP-14 to each MIP formed by the CRH algorithm, prior to solving it, improves solver performance. The CRH solver considers each pair of blocks (b, b') , for which $b' \in \cup_{k \in \mathcal{A}_b} \text{blocks}(k)$ and $b \in \cup_{k \in \mathcal{A}_{b'}} \text{blocks}(k)$ (the mining of either block can provide access to the other), and performs Steps (7)–(11) of Listing 1 with $\mathcal{U}$ initially containing blocks b and b' . The set of added constraints are labeled *basic dependency cuts*. Moreover, in each subsequent MIP, $M_{\mathcal{T}}$, solved over the course of the

Listing 1 A separation algorithm adding instances of constraint set CP-14 to MIP_{STP} , given a discretized horizon $\mathcal{T}$, and an incumbent solution $\mathbf{x}_I$.

```

1: for each  $t \in \mathcal{T}$  do
2:    $\mathcal{B}^1 \leftarrow$  The set of all (maximal with respect to set inclusion) contiguous sets of
     blocks  $\mathcal{B}' \subset \mathcal{B}$  scheduled for mining in period-aggregate  $t$  of solution  $\mathbf{x}_I$ .
3:    $\mathcal{B}^2 \leftarrow \{\mathcal{B}' | \mathcal{B}' \in \mathcal{B}^1, \nexists b \in \mathcal{B}' . \mathcal{A}_b = \emptyset\}$ 
4:   for each  $\mathcal{B}' \in \mathcal{B}^2$  do
5:      $\mathcal{U} \leftarrow \mathcal{B}'$ 
6:      $\mathcal{A}' \leftarrow \{k | \exists b \in \mathcal{U} . k \in \mathcal{A}_b, \exists b' \in \text{blocks}(k) . b' \notin \mathcal{U}\}$ 
7:     while  $\nexists b \in \mathcal{U} . \mathcal{A}_b = \emptyset$  do
8:       Add constraint  $\frac{1}{|\mathcal{U}|} \sum_{b \in \mathcal{U}} y_{bt} \leq \sum_{k \in \mathcal{A}'} y_{kt}$  to  $MIP_{STP}$ .
9:       Expand  $\mathcal{U}$  by adding to it all blocks that appear in a set within  $\mathcal{A}'$ .
10:       $\mathcal{A}' \leftarrow \{k | \exists b \in \mathcal{U} . k \in \mathcal{A}_b, \exists b' \in \text{blocks}(k) . b' \notin \mathcal{U}\}$ 
11:     end while
12:   end for
13: end for

```

algorithm, all instances of constraint set CP-14 that have been added to a prior MIP, and are still relevant (refer to blocks that have not yet been mined), are added to $M_{\mathcal{T}}$. These constraints are labeled *prior cuts*. Adding these constraints to each MIP, prior to solving it, avoids the need to repeatedly reinforce constraints that have been added in the past.

To generate the results presented in Section CP-6, the CRH solver uses all of the described cuts (prior cuts, basic dependency cuts, and ring cuts). Tables 1 to 3 consider the impact of ‘turning off’ each of these cut types on solver performance. Table 1 shows that the time required to generate schedules increases, in general, if *ring cuts* are not used. Table 2 records the number of schedules generated for scenarios $DS2_{a-d}$, alongside solve time, with all cut types and with the use of *prior cuts* turned off. Without *prior cuts*, the CRH solver fails to find *any* feasible schedules to scenarios $DS1_{a-e}$, for the parameter settings used ($\alpha_s = 2$, $\alpha_b = 3$, and $N = 2$). Table 3 records the number of schedules generated for scenarios $DS1_{a-e}$, alongside solve time, with all cut types and with the use of *basic dependency cuts* turned off. $DS2$ contains no disjunctive precedences, and no cuts of this type are generated. In the presence of disjunctive precedences, the addition of these cuts leads to more feasible schedules being formed (for $DS1_{a-d}$), in less time. Tables 1 to 3 show that switching off any of these cuts is detrimental to solver performance, in general.

Let $\mathcal{T}$ denote a discretized horizon spanning periods t' to T and $\mathcal{T}'$ a discretized horizon spanning $t' + 1$ to T . As described in Section CP-5, the CRH algorithm rolls forward from period t' to $t' + 1$ and creates a new instance of MIP_{STP} , $M_{\mathcal{T}'}$. Given a solution to $M_{\mathcal{T}}$, $\mathbf{x}$, it is possible to both warm start and guide the solving of $M_{\mathcal{T}'}$. Let $\mathcal{B}' \subset \mathcal{B}$ denote the set of blocks that have been completely extracted by the end of the horizon in $\mathbf{x}$. For each $b \in \mathcal{B}'$, the CRH solver places an increased priority on the branching of variable $y_{bt''}$, where t'' is the final period-aggregate in $\mathcal{T}'$, and on branching up, in place of down. Similarly, if material type $l \in \mathcal{L}$ has been mined from block $b \in \mathcal{B}'$ by the end of $\mathcal{T}$ in $\mathbf{x}$, an increased priority is placed on the branching of variable $y_{blt''}$, and on branching up. A starting solution in which $y_{bt''}$ and $y_{blt''}$ variables are set to 1 is used to warm start the solving of $M_{\mathcal{T}'}$ (using the *addMIPStart* method of CPLEX 12.6). Without this warm start, Table 4 reports the number of schedules generated for $DS1_{a-e}$ and $DS2_{a-e}$, alongside solve time. In the

Table 1 Time (s) used to solve scenarios $DS1_{a-e}$ and $DS2_{a-e}$ with and without the addition of ring cuts.

Scenario	Without Ring Cuts		With Ring Cuts	
	Time (s)	# Schedules	Time (s)	# Schedules
$DS1_a$	812	9	530	9
$DS1_b$	506	6	537	9
$DS1_c$	566	6	414	6
$DS1_d$	661	9	558	9
$DS1_e$	448	9	323	9
$DS2_a$	288	9	162	9
$DS2_b$	316	9	205	9
$DS2_c$	250	9	146	9
$DS2_d$	322	9	183	9

Table 2 Time (s) used to solve scenarios $DS2_{a-d}$ with and without the addition of prior cuts.

Scenario	Without Prior Cuts		With Prior Cuts	
	Time (s)	# Schedules	Time (s)	# Schedules
$DS2_a$	285	9	162	9
$DS2_b$	318	9	205	9
$DS2_c$	248	9	146	9
$DS2_d$	328	9	183	9

Table 3 Time (s) used to solve scenarios $DS1_{a-e}$ with and without the addition of basic dependency cuts.

Scenario	Without Basic Dependency Cuts		With Basic Dependency Cuts	
	Time (s)	# Schedules	Time (s)	# Schedules
$DS1_a$	851	8	530	9
$DS1_b$	730	8	537	9
$DS1_c$	498	2	414	6
$DS1_d$	–	0	558	9
$DS1_e$	559	9	323	9

Table 4 Time (s) to solve $DS1_{a-e}$ and $DS2_{a-d}$ with and without the warm starting of each MIP.

Scenario	Without Warm Start		With Warm Start	
	Time (s)	# Schedules	Time (s)	# Schedules
$DS1_a$	514	3	530	9
$DS1_b$	384	6	537	9
$DS1_c$	467	9	414	6
$DS1_d$	536	6	558	9
$DS1_e$	360	8	323	9
$DS2_a$	152	9	162	9
$DS2_b$	150	9	205	9
$DS2_c$	141	9	146	9
$DS2_d$	202	9	183	9

presence of disjunctive precedences, warm starting improves solver performance, increasing the number of schedules generated in scenarios $DS1_{a,b,d,e}$. In $DS1_a$, warm starting increases the number of schedules generated from 3 to 9, and from 6 to 9 in scenario $DS1_b$. Scenarios $DS2_{a-d}$, and $DS1_c$, do not benefit from the use of warm starting.

4. Discretization of the Horizon

Listing 2 defines a simple procedure for discretizing a horizon spanning periods t' to T into N aggregates, where the first aggregate contains 1 period (period t'), and the subsequent

Listing 2 Discretizing a horizon spanning periods t' to T into N period-aggregates.

```

1: procedure DISCRETIZE( $N, t', T$ )
2:    $H \leftarrow T - t'$ 
3:    $\text{spacings} \leftarrow \{1, 1, \dots, 1\}$      $\triangleright$  spacings is a  $N$  element array of 1's
4:    $\text{counter} \leftarrow N$ 
5:   while  $\text{counter} < H$  do
6:     for  $i = 1$  to  $N$  do
7:       for  $j = N$  to  $\min(N, i + 1)$  do     $\triangleright j$  is decremented after each loop iteration
8:          $\text{spacings}[j] \leftarrow \text{spacings}[j] + 1$ 
9:          $\text{counter} \leftarrow \text{counter} + 1$ 
10:        if  $\text{counter} \geq T$  then break
11:      end for
12:      if  $\text{counter} \geq T$  then break
13:    end for
14:  end while
15:   $\mathcal{T} \leftarrow \text{MAP}(t', T, \text{spacings})$ 
16:  return  $\mathcal{T}$ 
17: end procedure

```

aggregates increase monotonically in size. In the DISCRETIZE procedure, each element in the spacings array denotes the number of periods in the corresponding aggregate. For a 13-period horizon, and $N = 3$, this procedure will terminate with $\text{spacings} = [1, 4, 8]$.

5. Concluding Remarks

This document has formally defined a series of objectives supported by the CRH solver presented in the companion paper *Short-Term Scheduling of an Open-Pit Mine with Multiple Objectives*. Practical advice for the efficient performance of an implementation of this solver has been provided, including: the description of a separation algorithm for lazily adding mining precedence constraints; the definition of performance-improving cuts; and a mechanism for warm starting each MIP solved by the CRH solver.