A Sliding Time Window Heuristic for Open Pit Mine Block Sequencing ⋆

Christopher Cullenbine[1], R. Kevin Wood[2], and Alexandra Newman[1]

[1] Division of Economics and Business, Colorado School of Mines, Golden, CO 80401, USA,
ccullenb@mymail.mines.edu, newman@mines.edu,
[2] Operations Research Dept., Naval Postgraduate School, Monterey, CA 93943, USA,
kwood@nps.edu

Abstract. The open pit mine block sequencing problem (OPBS) seeks a discrete-time production schedule that maximizes the net present value of the orebody extracted from an open-pit mine. This integer program (IP) discretizes the mine's volume into blocks, imposes precedence constraints between blocks, and limits resource consumption in each time period. We develop a "sliding time window heuristic" to solve this IP approximately. The heuristic recursively defines, solves and partially fixes an approximating model having: (i) fixed variables in early time periods, (ii) an exact submodel defined over a "window" of middle time periods, and (iii) a relaxed submodel in later time periods. The heuristic produces near-optimal solutions (typically within 2% of optimality) for model instances that standard optimization software fails to solve. Furthermore, it produces these solutions quickly, even though our OPBS model enforces standard upper-bounding constraints on resource consumption along with less standard, but important, lower-bounding constraints.

Keywords: mine scheduling, mine planning, open pit mining, surface mining, integer programming applications

1 Introduction

The *open pit mine block sequencing problem* (OPBS) is an integer program (IP) whose solution is critical for the profitable operation of an open pit mine [16]. A solution yields a $\mathcal{T}$-period schedule for the extraction (i.e., "excavation" or "mining") of notional three-dimensional blocks of ore that contain valuable minerals, or costly waste, or both. (Test bores, coupled with geological prediction models, yield a reasonable estimate of each block's content.) The goal is to maximize the net present value of the extracted blocks, subject to spatial precedence constraints and resource constraints. This IP can be extremely difficult to solve, because: (i) a mine model may have from 10^4 to over 10^6 blocks, (ii) the time

⋆ We acknowledge the assistance of Professor Daniel Espinoza of the Universidad de Chile in running computation for upper bound verification.

horizon may have $\mathcal{T} = 20$ periods or more, and (iii) the resulting model may have millions of variables and constraints (e.g., [4], [8], and [3]). The purpose of this paper is to present a *sliding time window heuristic* (STWH) to solve OPBS approximately, and to demonstrate the heuristic's effectiveness on problems with 15 time periods and over 25,000 blocks.

Versions of STWH have been used for other optimization problems (e.g., [10], [5]), but our paper appears to be the first on this topic in the open-pit mining literature. Our particular application of Lagrangian relaxation also appears to be unique.

A typical STWH (e.g., [5]) first solves a "restricted exact model" over a time window that covers, say, periods 1 through τ of a full time horizon of $\mathcal{T} > \tau$ time periods. The heuristic then fixes the first period's variables to the solution values found, slides the time window up to periods 2 through $\tau+1$, solves a $\tau+1$ time horion model with the first period's variables fixed, fixes the second period's variables to the solution values just found, slides the time window up to periods 3 through $\tau + 2$, and continues the process until all periods have been covered in the time window. The user may want to solve a "global model" that covers all $\mathcal{T}$ periods, but that model is simply too difficult to solve. An STWH may provide an answer to the user's dilemma by producing good solutions quickly, even though it temporarily ignores the "out periods" $\tau + 1, \ldots, \mathcal{T}$.

We have tried the above method for OPBS using, for example, five-period "exact windows" for a model having $\mathcal{T} = 20$. Sometimes the method works well, and sometimes its myopia leads to a poor-quality solution. Instead of completely ignoring the out periods, this paper develops an STWH that maintains an approximate submodel in the out periods, along with an exact submodel in the "window" between the fixed and approximate parts of the overall model. Computational results show that this "Lagrangian approximation" produces high-quality solutions quickly, avoiding the difficulties that some Lagrangian methods have in even finding feasible solutions for OPBS.

"Precedence constraints" comprise the bulk of the constraints in any OPBS model and require some explanation. For simplicity, one may think of a mine's blocks as uniformly shaped cubes, defined by evenly spaced, parallel planes in the x-, y- and z-axes of three-dimensional space. Without loss of generality, we enforce spatial precedence constraints by specifying that for block b at a given z-level to be extracted: (i) the blocks adjacent to each face of b, but on the level directly above, must be extracted in the same time period or an earlier one, and (ii) one of the four blocks facing block b on block b's level must also be extracted. Figure 1 illustrates. (The configurations of a partially excavated mine seen in Figure 2 show how precedence constraints appropriately enforce the extraction of a sequence of "nested pits.")

Every OPBS model enforces upper bounds on resource consumption, but ours, somewhat unusually, also enforces lower bounds. The need for upper bounds is obvious: limited time and limited availability of equipment (shovels and trucks) lead to (upper-bounding) production-capacity constraints, and limited time and the finite capacity of milling facilities lead to (upper-bounding) processing-

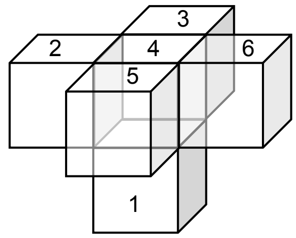

Fig. 1. Spatial precedence constraints in OPBS imply that block 1 cannot be extracted until blocks 2-6 have been extracted in the same or an earlier time period. Additionally, extraction of block 1 also requires that at least one of the blocks directly under blocks 2, 3, 5 or 6 be extracted.

capacity constraints. But, lower bounds can be important, too: the scale and nature of production and processing operations in an open-pit mine imply that large set-up costs accrue if operations are stopped and started repeatedly. Placing lower limits on production and processing reduces the potential for such effects. Contractual agreements and the chemical and physical properties of the milling process may also necessitate positive lower bounds on production and processing rates. Section 2 provides additional details on lower bounds, and points out that their inclusion in an OPBS model may hamper certain, specialized computational methods that have appeared in the literature.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature and motivates our work further. Section 3 defines our version of OPBS as a monolithic IP. Section 4 describes our STWH and the "restricted Lagrangian subproblem" that this heuristic solves repeatedly. Section 5 presents computational comparisons of the STWH to two alternative solution approaches, namely, direct solution of the monolithic IP, and a Lagrangian-based heuristic that does not use a sliding time window. Section 6 concludes the paper.

2 Literature Review

The seminal work of Lerchs and Grossman [18] provides an exact and computationally tractable method for "open-pit design;" Hochbaum and Chen [14], among others, extend this work. A solution to the design problem identifies the economically viable envelope of profitable blocks to be extracted given pit-slope requirements.

The design model relates structurally to OPBS, but it cannot schedule mine operations directly, because it ignores both time and limits on resources. Time periods, typically a few months to a year in length, must be modeled for scheduling purposes, so OPBS incorporates these. OPBS also constrains production (extraction) quantities and processing quantities in each time period to produce implementable schedules. Unfortunately, the generality of OPBS means that a typical mathematical-programming model for OPBS is at least an order of magnitude larger, and much harder to solve, than the corresponding design model.

For computational reasons, early work on OPBS aggregates blocks into strata (e.g., [6]), or ignores the discrete nature of the block-extraction decisions (e.g., [23]); unfortunately, both approaches reduce solution fidelity. Other early work investigates heuristics, but none that provides an indication of solution quality (e.g., [22], [21]; see also [24]). In contrast, our STWH avoids aggregation and, empirically, produces solutions with consistently good quality.

Early work in the literature on an OPBS IP uses variables that specify in, or "at," which time period a block is to be mined. Caccetta and Hill [7] improve this formulation by incorporating variables that represent whether a block is mined "by" time period t. They demonstrate the computational attractiveness of this modification by solving problems with as many as 210,000 blocks and 10 time periods, although optimality gaps range from 5% to 10% after 20 hours of computation. This model is one of the most general in the literature, as it: (i) handles inventory, and (ii) represents a "variable cutoff grade," meaning that the model determines whether an extracted block is to be processed for valuable ore or is to be classified as waste and left unprocessed. This model omits lower bounds on resource consumption, however.

Some OPBS models, including ours, incorporate a "fixed cutoff grade" rather than a variable one. A fixed cutoff grade implies that if a block contains a sufficiently high mineral content, it is always processed if extracted; otherwise, it is never processed. "Sufficiently high" is defined by the cutoff grade. Although a fixed cutoff grade might seem more appropriate for long-term strategic models, and a variable grade for short-term, tactical models, no hard and fast rules appear to exist about when one paradigm should be used over the other, and both are used in practice.

Various techniques have been applied to improve solution times for variants of the OPBS IP. Ramazan [20] addresses a model with a fixed cutoff grade, blending constraints, and production and processing constraints; the model also enforces lower-bounding constraints on processing, but not on production. Ramazan constructs aggregated "fundamental trees" to reduce model size. Specifically, his case study contains about 12,000 blocks, which are aggregated into about 1,600 fundamental trees. A four-period model solves to near-optimality in about 30 minutes. Boland et al. [4] develop a model with a variable cutoff grade but with no blending constraints and no lower bounds on resource consumption. These authors aggregate blocks according to precedence rules and solve instances with over 96,000 blocks and up to 25 time periods in a few hundred seconds. The fidelity of their solutions appears good, but it is unclear if their aggregation and disaggregation methods would apply in the presence of lower-bounding constraints. Gleixner [13] adapts the work in [4] to an alternative aggregation scheme and also presents ideas for applying Lagrangian relaxation.

Amaya et al. [2] present a model similar to that in [4] but with a fixed cutoff grade; they enforce upper bounds on resource consumption but not lower bounds. They develop a local-search heuristic that seeks to improve on a heuristically generated incumbent solution by iteratively fixing, relaxing and solving parts of the full model. This heuristic produces approximate solutions for the largest instances of OPBS reported in the literature to date, and uses the linear-programming (LP) relaxation of the OPBS IP to bound solution quality. For instance, one mine model has about four million blocks and 15 time periods (although the quality of the solution obtained is unclear in this case, because the bounding LP relaxation cannot be solved). Chicoisne et al. [8] solve the same formulation, but reduce computational effort by more efficient solution of the LP relaxations that guide the heuristic. Bienstock and Zuckerberg [3] develop a version of OPBS with a variable cutoff grade, but only solve LP relaxations. (They do solve those models quickly, however. For instance, one model with more than 100,000 blocks and 25 time periods solves in just hundreds of seconds.)

Lagrangian relaxation is key to the efficiency of our STWH, so we review previous work related to OPBS models here. (See Fisher [11] for a general discussion of Lagrangian relaxation.) Dagdelen and Johnson [9] present the earliest work, describing a model for extracting a fixed tonnage from a mine subject to precedence constraints; they solve a small, ten-block, two-period, illustrative example. Akaike and Dagdelen [1] suggest a different scheme for updating Lagrangian multipliers for instances with up 129,500 blocks and 5 time periods, but their solutions are not always feasible. Kawahata [17] applies similar techniques to a version of OPBS with a variable cutoff grade for instances with up to 58,970 blocks and 15 time periods; similar to other work, he is unable to consistenly find feasible solutions. Apparently, feasibility of resource constraints is difficult to obtain in a Lagrangian relaxation and, consequently, this technique has helped to solve only small instances of OPBS to date.

We can now better frame the current paper's contributions to the research literature on OPBS. We study a model variant that is more general than some (cf. [2], [8]) in that this variant enforces lower bounds on resource consumption. These constraints are important for practical applications, but can add significantly to solution effort. Even a small problem with lower bounds on resource consumption can be dramatically harder to solve than the same problem with those lower bounds omitted. For example, using the computer with the specifications given in section 5, one 2,880-block, five-period test problem having no lower bounds requires 176 seconds to solve with CPLEX, yet requires 3,688 seconds to solve when lower bounds are added. (Interestingly, the added restrictions reduce the optimal objective value by less than 0.1%.) On the other hand, our model omits certain features of other OPBS models, for instance, an inventory of mined but unprocessed material (e.g., [7]). Incorporating inventory constructs should be easy—this might involve adding fewer than a hundred new variables and constraints—but other generalizations would surely be more difficult (e.g., a variable cutoff grade). An attractive feature of our method is that it avoids aggregation and the complications that aggregation can entail (see [20], [4], [13]). Finally, we note that others have attempted to use Lagrangian relaxation to solve OPBS more quickly, but fail to obtain feasible solutions for even modest-sized problems. Our sliding time window heuristic exploits Lagrangian relaxation for speed, and it reliably produces high-quality, feasible solutions in large test problems.

3 An Integer Programming Model for OPBS

Our OPBS IP applies the following assumptions to a three-dimensional discretization of an orebody, i.e., to a block model: (i) each block must be mined in its entirety, or not at all, (ii) each block requires exactly one time period to mine, (iii) precedence constraints restrict how adjacent blocks may be extracted, (iv) each block contains a known amount of ore (mineral content) and waste, (v) a fixed cutoff grade applies, (vi) both lower and upper bounds apply to production and processing quantities in each period, and (vii) the mining operation holds no inventories of mined but unprocessed material. The following specifies a complete "*by* formulation" of OPBS as an IP (see [7]):

Indices, Indexed Sets, and Parameters:

$b \in B$	mine blocks
$t \in T$	time periods defining the time horizon
$r \in R$	production and processing resources
B_b	blocks above b that must be extracted directly before b
$\hat{B}_b$	blocks at the same level as b, and adjacent, one of which must be extracted in order to extract b
v_{bt}	net present value of block b if extracted in period t (\$)
n_{rb}	consumption of resource r associated with the extraction of block b (tons)
$\overline{C}_{rt}$	amount of resource r available in time period t (tons)
$\underline{C}_{rt}$	minimum level of resource r to be consumed in time period t (tons)

Variables:

y_{bt}	1 if block b is extracted *by* time period t, 0 otherwise (Note that $y_{bt} - y_{b,t-1}$ specifies whether or not block b is mined *at* time t.)

Formulation of $OPBS_{\mathcal{IP}}$:

$$\max \sum_{b \in B} \sum_{t \in T} v_{bt}(y_{bt} - y_{b,t-1}) \tag{1}$$

subject to

$$y_{bt} \le y_{b't} \quad \forall b \in B, b' \in B_b, t \in T \tag{2}$$

$$y_{bt} \le \sum_{\hat{b} \in \hat{B}_b} y_{\hat{b}t} \quad \forall b \in B, t \in T \tag{3}$$

$$y_{b,t-1} \le y_{bt} \quad \forall b \in B, t > 1 \tag{4}$$

$$\sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \le \overline{C}_{rt} \quad \forall r \in R, t \in T \tag{5}$$

$$\sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \ge \underline{C}_{rt} \quad \forall r \in R, t \in T \tag{6}$$

$$y_{bt} \in \{0, 1\} \quad \forall b \in B, t \in T \tag{7}$$

The objective function (1) maximizes the net present value of blocks extracted from the mine over the model's time horizon. Constraints (2) and (3) enforce spatial precedence on block extraction. Constraints (4) enforce "temporal precedence," that is, if a block is extracted by time $t-1$, it must also be extracted by time period t. Constraints (5) and (6) limit the maximum and minimum resource consumption in each time period, respectively. Constraints (7) require that all variables assume binary values.

4 A Sliding Time Window Heuristic

$OPBS_{\mathcal{IP}}$ is well known to be NP-hard (see [14]), and is also difficult to solve in practice. Instances comprising 10,000 blocks and 15 time periods can require hours to solve to near-optimality on a fast computer using state-of-the-art optimization software such as CPLEX [15]; modestly larger instances may not solve at all. This section describes a sliding time window heuristic that greatly extends the size of problems that can be solved. We note that "preprocessing" usually reduces solution times for OPBS dramatically (e.g., [2]), and is now a standard tool. Since we use preprocessing in all three methods compared in our paper, this section begins with a short description of the technique.

4.1 Preprocessing

A set of spatial precedence constraints can imply that a particular block b cannot be accessed until many thousands of overlying blocks b' have been extracted. Because this extraction occurs at rates restricted both by maximum production and maximum processing capacities, an "earliest start time" (earliest extraction period) for each block can be established by a "preprocessing routine." All variables that correspond to extracting a block before its earliest start time can then be eliminated, as they must equal 0 in any feasible solution. See [2], [8] and [19] for more detailed descriptions.

Conversely, spatial precedence constraints imply that not extracting a given block precludes extraction of underlying blocks. These underlying blocks can remain unextracted only as long as mining rates do not fall below minimum production and processing limits. Thus, a "latest start time" for each block can be established, and all variables corresponding to extracting a block at or after its latest start time can be fixed to 1. The above-cited papers that employ "earliest-start-time preprocessing" do not apply the latest-start-time analog because the relevant models omit lower bounds on resource consumption, or because demand requirements imply that lower bounds are elastic. We do apply that analog; see Gaupp [12] for a full description.

4.2 A Restricted Lagrangian Subproblem for STWH

The STWH algorithm repeatedly solves a *restricted Lagrangian subproblem* which we describe here. This subproblem, denoted $OPBS_{\mathcal{H}}$, (i) partitions the time periods T of $OPBS_{\mathcal{IP}}$ into three sequential subsets $T = T_1 \cup T_2 \cup T_3$, (ii) fixes all variables in the earliest group T_1 to feasible values, (iii) represents a "time window" T_2 in which all constraints of $OPBS_{\mathcal{IP}}$ are enforced, and (iv) enforces only a relaxed version of the model for the out periods $t \in T_3$. We present the formulation $OPBS_{\mathcal{H}}$ after making three additional definitions:

$\hat{y}_{bt}$	fixed value for y_{bt} for all $b \in B$ and $t \in T_1$ (all fixed variables constitute a portion of a feasible solution to $OPBS_{\mathcal{IP}}$)
μ_{bt}	non-negative Lagrangian multiplier for relaxing precedence constraint (3) for $b \in B$ and $t \in T_3$
$\overline{\lambda}_{rt}$, $\underline{\lambda}_{rt}$	non-negative Lagrangian multipliers for relaxing upper- and lower-bounding resource constraints, (5) and (6) respectively, for $r \in R$ and $t \in T_3$

Formulation of $OPBS_{\mathcal{H}}$:

$$\max \sum_{b \in B} \sum_{t \in T} v_{bt}(y_{bt} - y_{b,t-1}) + \sum_{t \in T_3, r \in R} \overline{\lambda}_{rt} \left(\overline{C}_{rt} - \sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \right) - \sum_{t \in T_3, r \in R} \underline{\lambda}_{rt} \left(\underline{C}_{rt} - \sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \right) + \sum_{t \in T_3, b \in B} \mu_{bt} \left(\sum_{\hat{b} \in \hat{B}_b} y_{\hat{b}t} - y_{bt} \right) \quad (8)$$

subject to precedence constraints (2), (3) and (4), and

$$y_{bt} \le \sum_{\hat{b} \in \hat{B}_b} y_{\hat{b}t} \quad \forall b \in B, t \in T_1 \cup T_2 \quad (9)$$

$$\sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \le \overline{C}_{rt} \quad \forall r \in R, t \in T_1 \cup T_2 \quad (10)$$

$$\sum_{b \in B} n_{rb}(y_{bt} - y_{b,t-1}) \ge \underline{C}_{rt} \quad \forall r \in R, t \in T_1 \cup T_2 \quad (11)$$

$$y_{bt} \in \{0,1\} \quad \forall b \in B, t \in T_1 \cup T_2 \quad (12)$$

$$0 \le y_{bt} \le 1 \quad \forall b \in B, t \in T_3 \quad (13)$$

$$y_{bt} \equiv \hat{y}_{bt} \quad \forall b \in B, t \in T_1 \quad (14)$$

It is easy to interpret $OPBS_{\mathcal{H}}$ if we first look at extreme cases.

1. If $T_1 = T_3 = \emptyset$, and $T_2 = T$, $OPBS_{\mathcal{H}}$ is identical to $OPBS_{\mathcal{IP}}$.
2. If $T_1 = T_2 = \emptyset$, and $T_3 = T$, we have a full Lagrangian relaxation of $OPBS_{\mathcal{IP}}$, as described in [12]. Note that the constraint matrix is totally unimodular in this case—it is the dual of a single-commodity network-flow model—and binary solutions are automatically obtained from extreme-point solutions (see [14]). It is unlikely that such a solution is feasible, but the structure of this model implies that standard optimization software like CPLEX can solve this mixed-integer problem quickly.

3. If $T_2 = T_3 = \emptyset$, and $T_1 = T$, all variables are fixed to values which are presumed to satisfy all constraints. (Moot constraints remain in the model.)

When none of the sets T_1, T_2 or T_3 is empty, a general mixture of the three cases appears: all variables associated with T_1 are fixed, a complete OPBS IP is represented in the "exact window" T_2, and a Lagrangian relaxation of that model is represented in the out periods T_3. In practice, we solve the full LP relaxation of $OPBS_{\mathcal{IP}}$ and use the optimal dual variables from constraints (5) and (6) as Lagrangian multipliers $\overline{\lambda}_{rt}$ and $\underline{\lambda}_{rt}$, respectively. It also seems natural to use dual variables from constraints (3) to define μ_{bt}, but empirical testing has shown that $\mu_{bt} = 0$ provides faster solutions with almost no reduction in solution quality. None of these multipliers are ever recalculated ("updated"), as we have not found this to be worthwhile computationally.

The STWH solves a sequence of $OPBS_{\mathcal{H}}$ subproblems, each of which can be viewed as an approximation to $OPBS_{\mathcal{IP}}$. It is natural to think that a better approximation would derive from using the full LP relaxation of $OPBS_{\mathcal{IP}}$ in periods T_3, rather than a Lagrangian relaxation. Experiments show that computational times using the LP relaxation can increase by a factor of 20, however, without any improvement in solution quality. The number of simplex iterations in the branch-and-bound solution process for the LP-based approximation increases only modestly compared to the number in the Lagrangian-based approximation, so the difference must be explained by the ease with which those iterations are executed. Apparently, the large dual network structure that $OPBS_{\mathcal{H}}$ presents to the solver is especially tractable.

4.3 The Heuristic Algorithm, ASTWH

We can now provide details of our sliding time window heuristic algorithm **ASTWH**. The algorithm assumes: (i) a window of τ time periods, i.e., $|T_2| = \tau$; (ii) $\tau < |T|$; and (iii) no subproblems become infeasible.

Algorithm ASTWH

1. $T_1 \leftarrow \emptyset$; $T_2 \leftarrow \{1, \ldots, \tau\}$; $T_3 \leftarrow \{\tau + 1, \ldots, |T|\}$; $t' \leftarrow 1$; $\hat{\mathbf{y}} \leftarrow \mathbf{0}$;
   /* t' is always the first period of the window T_2 */
2. Define $OPBS_{\mathcal{H}}$ with respect to T_1, T_2, T_3 and $\hat{\mathbf{y}}$, and solve for $\mathbf{y}^*$;
   /* Note that $\hat{\mathbf{y}}$ is irrelevant in the first iteration */
3. For (all $b \in B$) $\hat{y}_{b,t'} \leftarrow y^*_{b,t'}$; /* That is, fix all variables in period t' to the values just found */
4. $T_1 \leftarrow T_1 \cup \{t'\}$; $t' \leftarrow t' + 1$; $T_2 \leftarrow \{t', \ldots, \min\{t' + \tau - 1, |T|\}\}$; $T_3 \leftarrow \{\tau + t', \ldots, |T|\}$; /* That is, slide the window ahead one period, adjusting for the finite horizon, as appropriate */
5. If ($T_3 \neq \emptyset$) go to Step 2;
6. Print ("Solution from STWH is," $\mathbf{y}^*$) and halt.

In computational experiments, we apply a window width of $\tau = 1$. In numerous tests, any small improvements in solution quality derived from using

$\tau > 1$ are outweighed by the increase in computational effort. As a final note, we point out that **ASTWH**'s myopic strategy is not guaranteed to find a feasible solution if one exists. If a subproblem becomes infeasible, a method that relaxes fixed variables, expands T_2 and to tries to recover feasibility would be necessary.

5 Computational Results

We first test **ASTWH** on five instances of $OPBS_{\mathcal{IP}}$ comprising 10,819 blocks and 15 time periods [12]. These include a baseline instance denoted "10,819A," and four variations, denoted by suffixes B through E. The variations are obtained by randomly and independently perturbing mineral content in each block in the range $[-5\%, +5\%]$ according to a uniform probability distribution. Figures 2 and 3 illustrate the mine's layout and provide a rough idea of a block-sequencing solution for 10,819A.

We also test **ASTWH** on four larger 15-period problem instances, each of which is a subset of the 53,668-block mine model known as "Marvin." Marvin is included as a test problem in the Whittle Four-X mine-planning software, and comprises an artificially constructed copper-and-gold orebody with 17 vertical levels. All test problems are derived from horizontal "slices" of the mine. Instances "18,300A" and "18,300B" represent the same slice of 18,300 blocks, but with slightly different mineral content. Instances "25,620A" and "25,620B" are similar, but have 25,620 blocks. We have created these "submines" for testing purposes because they are larger and more challenging computationally and, admittedly, we cannot yet solve the full, 53,668-block model.

All models are generated in AMPL 12.1 and solved with CPLEX 12.1 (see [15]) on a 64-bit workstation running the Linux operating system. The workstation has four Intel processors running at 2.27 GHz and is supplied with 12 GB of RAM. "Branching priorities" for branch-and-bound solutions are set so that high-profit blocks are branched on before low- or negative-profit blocks; other branching rules state "branch up first on positive-valued blocks," and "branch down first on negative-valued blocks." We apply these CPLEX options (see [15]) for solving the LP relaxation of $OPBS_{\mathcal{IP}}$: "predual 1," "netopt 2" and "primalopt." In addition, "mipbasis 0," "mipemphasis 3" and "mipcuts 2" apply when solving any $OPBS_{\mathcal{H}}$ subproblem.

Table 1 presents computational results for three different solution procedures: (i) direct solution of $OPBS_{\mathcal{IP}}$ with CPLEX, (ii) Gaupp's optimization-based heuristic [12], and (iii) **ASTWH**. All procedures begin with early- and late-start preprocessing; computational times are negligible and are not reported. The Gaupp procedure applies Lagrangian relaxation with subgradient updates and a special "feasing heuristic" that attempts to convert infeasible Lagrangian solutions into feasible ones. Lagrangian multipliers are obtained for **ASTWH** by first solving the LP relaxation of $OPBS_{\mathcal{IP}}$, and this computation time is included in the total solution times reported.

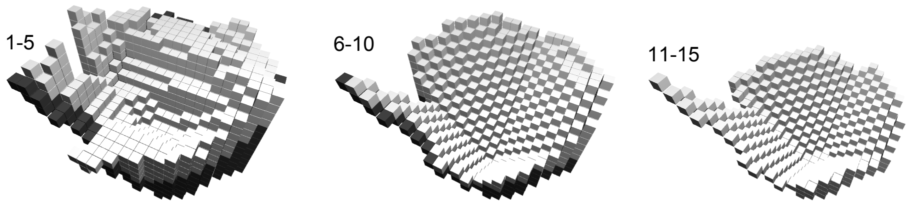

Fig. 2. A three-dimensional block extraction schedule, in five-period increments, for the baseline OPBS instance, "10,819A;" the sheer walls in the far left diagram represent awkward initial conditions.

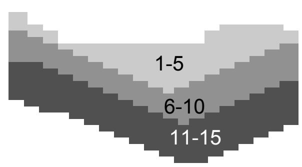

Fig. 3. A two-dimensional slice of the block-extraction schedule, in five-period increments, for the baseline OPBS instance "10,819A."

Table 1. Numerical results comparing direct solution of $OPBS_{\mathcal{IP}}$, Gaupp's method ("Gaupp;" see [12]) and our sliding time window heuristic **ASTWH**. A time limit of 36,000 seconds, i.e., 10 hours, applies. All methods first employ early- and late-start preprocessing. The IP $OPBS_{\mathcal{IP}}$ fails to solve in most instances: † indicates that the problem could not be solved to a tolerance of 2% within the time limit, and ‡ indicates that no feasible solution was obtained within that limit.

	$OPBS_{\mathcal{IP}}$		Gaupp		STWH	
Instance name	Solution time (sec)	Optimality gap (%)	Solution time (sec)	Optimality gap (%)	Solution time (sec)	Optimality gap (%)
10,819A	†	‡	†	4.2	1,596	1.9
10,819B	†	‡	35,370	2.3	1,463	2.0
10,819C	†	‡	14,376	1.8	1,405	1.9
10,819D	†	6.3	16,770	2.2	1,949	2.0
10,819E	†	‡	28,572	2.3	1,466	2.0
18,300A	†	‡	†	‡	2,461	2.4
18,300B	†	‡	†	‡	1,250	1.4
25,620A	†	‡	†	‡	3,045	1.6
25,620B	†	‡	†	‡	9,823	4.3

The solution quality for the two heuristics is given with respect to the optimal objective-function value from the LP relaxation of $OPBS_{\mathcal{IP}}$. The Gaupp procedure terminates when the optimality gap drops to 2% or less, so that heuristic might produce somewhat better solutions given more time. Of course, this criterion cannot be successful unless the LP relaxation for $OPBS_{\mathcal{IP}}$ is quite tight, but it is for problems tested here. **ASTWH** does not prespecify an overall optimality criterion, but simply solves each mixed-integer subproblem to within 0.1% of optimality. (A near-optimal solution procedure for the first $OPBS_{\mathcal{H}}$ subproblem can provide an upper bound for SWTH that is better than the LP bound. This improvement is marginal, however, so we simply use the LP bound for all gap computations.)

For the 10,819-block instances, Table 1 shows that **ASTWH** produces results of similar or better quality than the Gaupp heuristic, but 10 to 20 times faster. Within the ten-hour limit, $OPBS_{\mathcal{IP}}$ cannot solve these problems reliably. **ASTWH** also successfully solves the four larger problem instances, although the optimality gap for one instance rises modestly to 4.3%. The gap stays below 2.5% for the other three instances, and the longest computation time is only about two and three-quarter hours. These are promising results for an IP that contains as many as 200,500 variables and over 1,173,000 constraints (after CPLEX eliminates extraneous variables and constraints in its "presolve" routine). Note that neither the Gaupp procedure nor CPLEX can even find a feasible solution to these problems in ten hours of computation.

6 Conclusions

This paper has presented a sliding time window heuristic (STWH) for approximately solving an integer-programming formulation (IP) of the open pit mine block sequencing problem (OPBS). OPBS models the extraction of blocks of material from a mine over a discretized time horizon, subject to spatial precedence constraints and subject to lower and upper limits on production and processing in each time period. The STWH is based on solving a sequence of mixed-integer programs that have fixed variables in early time periods, a full model representation in at least one middle period, and a relaxed representation in later periods. The use of Lagrangian relaxation is critical for computational efficiency.

Many papers on OPBS report results on problems that only enforce upper-bounding constraints on production and on processing. But lower-bounding constraints can be important to maintain smooth mine operations and to meet contractual agreements. Consequently, we test difficult problems that include lower and upper bounds on both production and processing. We solve problem instances with 15 time periods and up to 25,000 blocks. On average, the largest problems require about one and a quarter hours to run on a fast workstation, and exhibit an average optimality gap of 2.4%.

References

1. Akaike, A. and Dagdelen, K.: A strategic production scheduling method for an open pit mine. In Proceedings of the 28th Applications of Computers and Operations in the Mineral Industries Conference (APCOM), C. Dardano, M. Francisco and J. Proud, eds., Golden, CO, 729–738. (1999)
2. Amaya, J., Espinoza, D., Goycoolea, M., Moreno, E., Prevost, T. Rubio, E.: A scalable approach to optimal block scheduling. In 35th APCOM, Vancouver, Canada. (2009)
3. Bienstock, D., Zuckerberg, D.: Solving LP relaxations of large-scale precedence constrained problems. Lecture Notes in Computer Science, 6080/2010: 1–14 (2010)
4. Boland, N., Dumitrescu, I., Froyland, G., Gleixner, A.: LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. Computers and Operations Research 36, 1064–1089 (2009)
5. Brown, G., Keegan, J., Vigus, B. Wood, R.K.: The Kellogg Company optimizes production, inventory and distribution. Interfaces 31, 1–15 (2001)
6. Busnach, E., Mehrez, A., Sinuany-Stern, Z.: A production problem in phosphate mining. Journal of the Operational Research Society 36, 285–288 (1985)
7. Caccetta, L., Hill, S.: An application of branch and cut to open pit mine scheduling. Journal of Global Optimization 27, 349–365. (2003)
8. Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., Rubio, E.: A new algorithm for the open-pit mine scheduling problem. Working Paper, Universidad Adolfo Ibañez and Universidad de Chile (2009)
9. Dagdelen, K., Johnson, T.: Optimum open pit mine production scheduling by Lagrangian parameterization. In 19th APCOM, University Park, PA, 127–141 (1986)
10. Ferland, J., Fortin, L.: Vehicles scheduling with sliding time windows. European Journal of Operational Research 38, 213–226 (1989)
11. Fisher, M.: The Lagrangian relaxation method for solving integer programming problems. Management Science 27, 1–18 (1981)
12. Gaupp, M.: Methods for improving the tractability of the block sequencing problem for open pit mining. Ph.D. thesis, Colorado School of Mines. (2008)
13. Gleixner, A.: Solving large-scale open pit mining production scheduling problems by integer programming. Master's thesis, Technische Universität Berlin. (2008)
14. Hochbaum, D., Chen, A.: Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. Operations Research 48, 894–914 (2000)
15. IBM: IBM ILOG AMPL Version 12.1 Users guide: Standard (command-line) version including CPLEX directives, Incline Village, NV (2010)
16. Johnson, T.: Optimum open pit mine production scheduling. In A Decade of Digital Computing in the Mining Industry, A. Weiss, ed., American Institute of Mining Engineers, New York, chap. 4. (1969)
17. Kawahata, K.: A new algorithm to solve large scale mine production scheduling problems by using the Lagrangian relaxation method. Ph.D. thesis, Colorado School of Mines. (2007)
18. Lerchs, H., Grossmann, I.: Optimum design of open-pit mines. Canadian Mining and Metallurgical Bulletin LXVIII, 17–24 (1965)
19. Martinez, M., Newman, A.: Using decomposition to optimize long- and short-term production planning at an underground mine. European Journal of Operational Research 211, 184–197 (2011)

20. Ramazan, S.: The new fundamental tree algorithm for production scheduling of open pit mines. European Journal of Operational Research 177, 1153–1166 (2007)
21. Sevim, H., Lei, D.: The problem of production planning in open pit mines. Information Systems and Operational Research Journal (INFOR) 36, 1–12 (1998)
22. Sundar, D., Acharya, D.: Blast schedule planning and shiftwise production scheduling of an opencast iron ore mine. Computers and Industrial Engineering 28, 927–935 (1995)
23. Tan, S., Ramani, R.: Optimization models for scheduling ore and waste production in open pit mines. In 23rd APCOM, SME of the AIME, Tucson, AZ, 781–791 (1992)
24. Zhang, M.: Combining genetic algorithms and topological sort to optimize open-pit mine plans. In 15th Mine Planning and Equipment Selection, M. Cardu, R. Ciccu, E. Lovera and E. Michelotti, eds., FIORDO S.r.l., Torino, Italy, 1234–1239. (2006)