Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Hierarchical Benders Decomposition for Open-Pit Mine Block Sequencing

Thomas W. M. Vossen, R. Kevin Wood, Alexandra M. Newman

Operations Research
Vol. 64, No. 4, July–August 2016, pp. 771–793
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2016.1516

Hierarchical Benders Decomposition for Open-Pit Mine Block Sequencing

Thomas W. M. Vossen
Leeds School of Business, University of Colorado at Boulder, Boulder, Colorado 80309, vossen@colorado.edu

R. Kevin Wood
Operations Research Department, Naval Postgraduate School, Monterey, California 93943, kwood@nps.edu

Alexandra M. Newman
Department of Mechanical Engineering, Colorado School of Mines, Golden, Colorado 80401, newman@mines.edu

The open-pit mine block sequencing problem (OPBS) models a deposit of ore and surrounding material near the Earth's surface as a three-dimensional grid of blocks. A solution in discretized time identifies a profit-maximizing extraction (mining) schedule for the blocks. Our model variant, a mixed-integer program (MIP), presumes a predetermined destination for each extracted block, namely, processing plant or waste dump. The MIP incorporates standard constructs but also adds not-so-standard lower bounds on resource consumption in each time period and allows fractional block extraction in a novel fashion while still enforcing pit-wall slope restrictions. A new extension of nested Benders decomposition, "hierarchical" Benders decomposition (HBD), solves the MIP's linear-programming relaxation. HBD exploits time-aggregated variables and can recursively decompose a model into a master problem and two subproblems rather than the usual single subproblem. A specialized branch-and-bound heuristic then produces high-quality, mixed-integer solutions. Medium-sized problems (e.g., 25,000 blocks and 20 time periods) solve to near optimality in minutes. To the best of our knowledge, these computational results are the best known for instances of OPBS that enforce lower bounds on resource consumption.

Keywords: industries: mining; production scheduling: deterministic; sequencing; integer programming: Benders decomposition; heuristic.
Subject classifications: industries: mining/metals; production/scheduling: applications; programming: integer; algorithms: Benders decomposition.
Area of review: Environment, Energy, and Sustainability.
History: Received September 2014; revisions received November 2015; accepted April 2016. Published online in *Articles in Advance* July 8, 2016.

1. Introduction

The mining industry solves the open-pit mine block sequencing problem (OPBS), primarily for strategic planning purposes, with typical models incorporating a yearly level of detail over a 10- to 30-year time horizon (Rojas et al. 2007, Chicoisne et al. 2012, Epstein et al. 2012). This paper extends a standard integer program (IP) for OPBS to a mixed-integer program (MIP) and develops a specialized solution procedure for that MIP. Although open-pit mines may produce diamond ore, coal, and materials other than metal ores, without loss of generality, we discuss OPBS in terms of mining metal ores. Figure 1 illustrates a large open-pit copper mine for reference.

In OPBS, a three-dimensional grid of box-shaped blocks represents a deposit of potentially valuable ore containing metals such as gold or copper, along with inevitable waste. An IP or MIP for OPBS seeks a multi-period schedule for extracting (mining) and processing these blocks, a schedule that (i) maximizes net present value, (ii) satisfies constraints on the shape of the mine as it evolves over time, and (iii) satisfies constraints on resource consumption in each time period.

Our work begins by applying lower-bounding resource constraints, in addition to the standard upper-bounding constraints, to one variant of a binary (0-1) IP for OPBS (Chicoisne et al. 2012). More significantly, we relax the IP, converting it into a MIP that allows selective, fractional extraction of blocks: researchers typically assume that restrictions on the shape of the mine require the use of binary variables, but we show that our relaxed regime also satisfies those restrictions. We then develop a specialized solution procedure for the new MIP that (i) defines the MIP's linear-programming relaxation without explicitly representing relaxed binary variables, (ii) solves that linear program using a new "hierarchical" version of nested Benders decomposition (Ho and Manne 1974), and then (iii) incorporates that linear-programming solution method within a specialized branch-and-bound heuristic that enforces discrete relationships within the MIP through constraint branching. Thus, the method avoids explicit use of binary variables.

Figure 1. The Bingham Canyon Mine as of 2003.

Source. http://commons.wikimedia.org/wik/File:Bingham_mine_5-10-03.jpg, accessed July 11, 2013. *Notes.* This copper mine is one of the largest open-pit mines in the world. Visible in this figure are the benches, or "steps" from which ore and waste are extracted, and the haul roads, which wind down past the benches to the bottom of the pit.

As with most work on OPBS (e.g., Dagdelen and Johnson 1986, Caccetta and Hill 2003, Chicoisne et al. 2012), we solve only a deterministic model, even though uncertainty surely plays a role in strategic mine planning (Johnson 1968). For example, price estimates for metals 10 years in the future must have large variances, and ore quality 500 meters below the Earth's surface cannot be known with certainty. Stochastic programming methods have been suggested for open-pit mine planning (e.g., Ramazan and Dimitrakopoulos 2007, Boland et al. 2008, Gholamnejad and Moosavi 2012), but the current state of the art does not permit the solution of full-scale stochastic programming models. Thus, we assume that (i) core samples from the deposit (Krige 1951) and radio-imaging techniques (Stolarczyk 1992) yield accurate deterministic estimates of each block's weight and grade, with a block's *grade* being the percentage of metal it contains; (ii) those values, together with economic forecasts, yield acceptable deterministic estimates of the profitability of extracting each block in each possible time period; and (iii) sources of uncertainty can be handled in an ad hoc manner using a deterministic model.

Several variants of OPBS exist (Espinoza et al. 2012), but our variant incorporates two key features that at least one standard mine-planning software system also uses (Whittle 1998), namely, a "fixed cutoff grade" and "no inventorying." *Fixed cutoff grade* implies that if a block's estimated grade is at least $g\%$ for some pre-specified value g, that block is sent to a processing plant to be converted into salable ore; otherwise, it is sent to a waste dump. *No inventorying* implies that a block must be processed or dumped in the period in which it is extracted or the block is never extracted at all.

1.1. Technical Background on OPBS

For computational reasons, OPBS normally defines extraction variables of this form: $x_{bt} = 1$ if block $b \in \mathscr{B}$ is extracted *by* (i.e., at or before) time period $t \in \mathscr{T}$, and $x_{bt} = 0$, otherwise (Johnson 1968). Strategic planners typically seek an extraction schedule that covers 10^4–10^7 blocks and 10–100 time periods for a model instance that spans 10–30 years.

If block $b \in \mathscr{B}$ is not at the mine's surface, then a unique block denoted $\bar{b}$ lies directly above b; let $\bar{\mathscr{B}}_b = \{\bar{b}\}$ if $\bar{b}$ exists, and let $\bar{\mathscr{B}}_b = \varnothing$ otherwise. Also, a specially defined set of blocks $\hat{\mathscr{B}}_b$ may lie obliquely above b. The blocks $b' \in \bar{\mathscr{B}}_b \cup \hat{\mathscr{B}}_b \equiv \mathscr{B}_b$ are the *direct spatial predecessors* of b, and most of the constraints in OPBS enforce *spatial precedence:*

$$x_{bt} - x_{\bar{b}t} \leqslant 0 \quad \forall\, b \in \mathscr{B} \mid \bar{\mathscr{B}}_b \neq \varnothing,\ t \in \mathscr{T} \tag{1}$$

$$x_{bt} - x_{b't} \leqslant 0 \quad \forall\, b \in \mathscr{B} \mid \hat{\mathscr{B}}_b \neq \varnothing,\ b' \in \hat{\mathscr{B}}_b,\ t \in \mathscr{T}. \tag{2}$$

That is, block b cannot be extracted by period t unless all of its direct spatial predecessors are extracted by t. Constraints (1) and (2) are typically written as a single set of constraints, but we find it useful to split them into two sets because of different interpretations. In particular, constraints (1) simply imply that a block cannot be extracted until the top of the block forms part of the mine's surface, while constraints (2) mathematically enforce *slope restrictions* on the pit's walls to prevent their collapse (Johnson 1968). The relationships expressed through block b's *oblique predecessors* $b' \in \hat{\mathscr{B}}_b$ may vary throughout the potential mine volume depending on local characteristics of the rock and minerals.

In addition to spatial-precedence constraints, OPBS implements *temporal-precedence* constraints, which imply that if block b is extracted by time period $t < T$, then that block must also be extracted by period $t+1$:

$$x_{bt} - x_{b,\,t+1} \leqslant 0 \quad \forall\, b \in \mathscr{B},\ t = 1, \ldots, T-1. \tag{3}$$

Note that both types of precedence constraints exhibit "dual network structure," with each defining a constraint-matrix row whose nonzero coefficients are a single $+1$ and a single -1. Certain solution methods exploit this structure for efficiency; see Ahuja et al. (2003) for a general discussion of the topic, and see Chicoisne et al. (2012) for a recent discussion with respect to OPBS.

A solution to OPBS must also satisfy resource constraints on *production* and *processing* in each time period (Johnson 1968). Production constraints limit the total weight of the blocks that are extracted, while processing constraints limit the total weight that can be milled, i.e., crushed and refined for sale. Upper-bounding constraints for both production and processing reflect limited equipment and labor capacities, while labor contracts and requirements of exothermic processing reactions may dictate lower bounds on production and processing, respectively.

1.2. Basic Solution Approaches for OPBS

Lerchs and Grossmann (1965) describe an efficient algorithm for a simplified version of OPBS. Ignoring resource constraints and time periods in OPBS produces the ultimate pit limit problem (UPL). A solution to UPL estimates the extent of the pit beyond which no profit is possible because further mining would require the extraction of excessive waste material to reach nominally valuable ore. This model corresponds to a dual network with no complicating structure and, therefore, solves efficiently using network-flow techniques. In fact, the max-flow/min-cut theorem applies, and UPL may be viewed as a classical OR problem, appearing in numerous textbooks and research papers (e.g., Ahuja et al. 1993, pp. 721–722; Hochbaum and Chen 2000).

Johnson (1968) presents the first comprehensive description of OPBS. He gives formulations with "at-time-t variables" x'_{bt} (i.e., x'_{bt} equals 1 if block b is extracted at time t, and x'_{bt} equals 0 otherwise), as well as formulations like ours with "by-time-t variables" x_{bt}. We refer the reader to Lambert et al. (2014) for details on these models and on their direct solution by LP-based branch-and-bound methods. (The abbreviation "LP" means "linear-programming" or "linear program," depending on the context.) For our purposes, the key point in Lambert et al. is this: solution via branch and bound of realistically sized OPBS models lies beyond the capability of current-day integer-programming solvers. We note that Caccetta and Hill (2003) apply branch and cut to a version of OPBS and report promising results on problems with up to 210,000 blocks and 10 time periods. Reported optimality gaps are large, however, and the paper's lack of detail makes its results irreproducible. These difficulties with direct branch-and-bound solutions, and our desire to avoid using heuristics and aggregation schemes that provide no measure of solution quality (e.g., Gershon 1987, Denby and Schofield 1994, Ramazan 2007, Boland et al. 2009), motivate us to pursue a decomposition-based solution approach.

1.3. Decomposition Methods for OPBS

Dagdelen and Johnson (1986) appear to be the first to apply mathematical decomposition in an attempt to solve OPBS. Their Lagrangian relaxation of the model's resource constraints yields a subproblem having pure dual network structure, which results in an integer solution to the continuous relaxation just as the UPL model does. One must eventually find a solution that satisfies the initially relaxed constraints, however, and Dagdelen and Johnson's method often fails in this regard. Other work with Lagrangian relaxation and OPBS has also had limited success; for example, Akaike and Dagdelen (1999), Cai (2001), and Kawahata (2007) all have difficulty finding resource-feasible solutions.

Gleixner (2008) produces promising results using Lagrangian relaxation on the variant of OPBS described by Boland et al. (2009). This model aggregates certain standard constructs and relaxes others, but the validity of these techniques remains unproven. Cullenbine et al. (2011) obtain high-quality solutions using a Lagrangian-based "sliding time window heuristic," which is a type of fix-and-relax heuristic (Pochet and Wolsey 2006). Lambert and Newman (2014) use Lagrangian relaxation to speed solutions of OPBS but guarantee a solution only through what may evolve into a complete OPBS model that must be solved by branch and bound. We seek a decomposition method for solving OPBS that promises to be faster than a brute-force, branch-and-bound solution of a monolithic MIP and that provides an objective measure of solution quality.

Chicoisne et al. (2012) apply an efficient method to solve the continuous relaxation of an OPBS IP and then apply a greedy heuristic to identify an integer solution. Because of a strong bound from the relaxation and an effective heuristic, this method yields solutions with optimality gaps of approximately 2% on problems with up to 10^7 blocks and 25 time periods. These authors report solution times of a few hours on a computer having two Quad-Core Intel Xeon E5420 processors. We also note that Bienstock and Zuckerberg (2010) model a variant of OPBS with a variable cutoff grade and solve the corresponding continuous relaxations efficiently. For instance, models with 10^5 blocks and 25 time periods solve in only hundreds of seconds using a single core of a 3.2 GHz Xeon processor on a computer having 64 GB of memory. However, the papers mentioned in this paragraph omit lower-bounding constraints on resource consumption, and evidence indicates that incorporating both constraint types can increase computation times, even on small problems, by more than an order of magnitude (Cullenbine et al. 2011).

We aim to take advantage of OPBS's *staircase structure*, meaning that variables for period t interact directly through constraints only with variables for periods $t-1$ and $t+1$. Glassey (1971) first shows that LPs having such structure can be solved using a *nested decomposition*, specifically, a nested version of Dantzig-Wolfe decomposition (Dantzig and Wolfe 1960). The key advantage of using a nested decomposition to solve a staircase model is that a "nested subproblem" involves only variables associated with a single time period. "Branch and price" extends Dantzig-Wolfe decomposition to integer problems (Johnson 1968, Barnhart et al. 1998), but this technique seems difficult to adapt to our multistage problem. We have turned, therefore, to *nested Benders decomposition* (NBD), first described by Ho and Manne (1974).

Benders decomposition was originally developed for solving MIPs (Benders 1962). It views the solution of a maximizing MIP having integer variables $\mathbf{x}$ and continuous variables $\mathbf{y}$ as $\max_{\mathbf{x}\in\mathscr{X}} \theta(\mathbf{x})$, where $\theta(\mathbf{x})$ is a piecewise-linear concave function in continuous $\mathbf{x}$, and $\mathscr{X}$ is defined through a polyhedral set with integrality requirements added.

The decomposition algorithm

1. creates an easy-to-evaluate approximating function $\bar{\bar{\theta}}(\mathbf{x})$ with $\bar{\bar{\theta}}(\mathbf{x}) \geqslant \theta(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}$,
2. solves the "relaxed master problem" $\max_{\mathbf{x}\in\mathscr{X}} \bar{\bar{\theta}}(\mathbf{x})$ for $\hat{\mathbf{x}} \in \mathscr{X}$,
3. solves the LP subproblem in variables $\mathbf{y}$ that results from fixing $\mathbf{x} = \hat{\mathbf{x}}$ in the MIP,
4. extracts a dual extreme point or dual extreme ray from the solution to that LP to generate a new constraint called a "Benders cut" to help refine $\bar{\bar{\theta}}(\mathbf{x})$, and
5. repeats steps 2–4 until the best $\hat{\mathbf{x}}$ found meets convergence criteria.

NBD extends the two-stage method to multistage LPs or to multistage MIPs with integer variables in the first stage only. In concept, NBD views an LP in terms of a master problem and subproblem and then recursively decomposes the subproblem into a master problem and subproblem using the basic ideas from standard Benders decomposition. We call the problem solved at any stage t of standard NBD a "nested subproblem" even though it contains constructs of a master problem.

For simplicity, assume that each period-t nested Benders subproblem with variables $\mathbf{x}_t$, $t = 1, \ldots, T$, has a bounded, feasible solution. The following procedure then outlines a standard implementation of NBD, which solves a forward recursion of an LP.

1. Vector $\hat{\mathbf{x}}_0$ defines initial conditions.
2. A *primal pass*, in the order $t = 1, \ldots, T$, solves a period-t nested subproblem for primal solution $\hat{\mathbf{x}}_t$ given $\hat{\mathbf{x}}_{t-1}$. (Note that the existence of $\hat{\mathbf{x}}_t$ implies that a consistent primal solution $\hat{\mathbf{x}}_{t'}$ has been computed for $t' = 1, \ldots, t-1$.) This subproblem involves only variables $\mathbf{x}_t$ but, except when $t = T$, it does incorporate an approximate cost-to-go function $\bar{\bar{\theta}}_{t+1}(\mathbf{x}_t)$, which covers the beginning of period $t+1$ through the end of period T.
3. A *dual pass*, in the order $t = T, \ldots, 2$, solves a period-t nested subproblem for dual solution $\hat{\boldsymbol{\pi}}_t$ to generate a new Benders cut that refines the cost-to-go function for the nested subproblem in period $t-1$. (Actually, the solution to the period-T nested subproblem in the primal pass yields the initial dual-pass result $\hat{\boldsymbol{\pi}}_T$.)
4. Steps 2 and 3 are repeated until the pessimistic bound from step 2 and the optimistic bound from step 3 are sufficiently close.

(Note that some authors use "forward recursion" and "backward recursion" to mean what we call "primal pass" and "dual pass," respectively.)

It is possible to reorganize computations above, for instance, by iterating between a primal solution for period t and a dual solution for period $t+1$ until some local convergence criterion is reached, then iterating between $t+1$ and $t+2$, etc. However, the outline above describes the subproblem-processing method that both Wittrock (1985) and Gassman (1990) find most efficient for implementing NBD and that we therefore adopt or adapt, as needed. We use Wittrock's term for this method, "FASTPASS."

Staircase structure lends nested decomposition its computational advantage, but this structure is also its Achilles heel. If strong linkages exist between distant time periods, eventually those linkages must be represented by generating and applying Benders cuts in many time periods. Slow convergence results. We seek to improve the empirical convergence rate using several techniques.

First, we formulate the model using *cumulative variables*, that is, variables that are cumulative over time. Of course, cumulative extraction variables in OPBS formulations are actually standard: $x_{bt} = 1$ if block b is extracted by time period t, and $x_{bt} = 0$, otherwise.

Next, we add redundant, *aggregate resource constraints* to help guide the decomposition. For example, the decomposition procedure's first subproblem might aggregate all time periods into a single cumulative period and, in essence, ask, "What is the optimal 'relaxed open-pit mine' that could be excavated in a single period if each resource constraint aggregates resource availability over the entire time horizon?" Intuitively, this provides immediate, global information to subsequent subproblems. By contrast, the first primal pass of a standard forward recursion would greedily excavate the "relaxed mine" from one period to the next, with global information appearing only slowly as the procedure refines approximating cost-to-go functions over many iterations. (Section 3.6 covers this topic further and compares our techniques to others in the literature.)

Finally, we show that the standard recursion used to create a multistage Benders decomposition is a special case of a *tree decomposition*: a standard decomposition recursively decomposes a multistage problem into a master problem and a subproblem, while the tree decomposition can recursively decompose that problem into a master problem and two subproblems. This more general decomposition framework may be viewed in terms of a binary tree and, consequently, resembles the decomposition scheme proposed by Kallio and Porteus (1977) for solving a set of linear equations having a tree structure. Kallio and Porteus intend for their decomposition to improve computational efficiency but provide no supporting, computational results. We also note that these authors assume a given tree structure, whereas we create various tree structures through problem reformulations. In the literature on decomposition for optimization, only the work by Entriken (1989) seems closely related to ours. Entriken proposes a framework for decomposing an LP that could, in principle, yield a formulation having a general tree structure. He provides computational results only for sequentially structured decompositions, however; see also Entriken (1996).

For simplicity, we use the phrase *hierarchical Benders decomposition* (HBD) to refer to the combination of all three techniques just described, i.e., cumulative variables, aggregate constraints, and a tree-structured decomposition. While somewhat specialized, HBD should also apply to a number of multistage production-scheduling problems in the literature, including production-planning problems (Gabbay 1979), production-distribution problems (Brown et al. 2001), short-term scheduling for forest harvesting (Karlsson et al. 2003), and short-term open-pit mine scheduling (Eivazy and Askari-Nasab 2012). The key is that production efficiencies or yields should not change substantially over time.

Eventually we need a solution to a MIP, not just an LP. For the same reason that Benders decomposition applies to problems with integer variables in the first stage only, NBD and HBD apply (directly) to staircase MIPs with integer variables in a single stage only (e.g., Wollmer 1980). To address this limitation, we develop a branch-and-bound heuristic that produces high-quality binary solutions for a variety of test problems. For efficiency, the methods represent binary variables only implicitly.

1.4. Outline of the Remainder of the Paper

Section 2 begins the remainder of this paper by describing a new MIP for OPBS and justifying the changes from more standard models. Section 3 demonstrates how standard NBD applies to solve the LP relaxation of the MIP and then generalizes that method to HBD; Section 4 presents corresponding computational results. Based on solving LP relaxations with HBD, Section 5 devises a branch-and-bound heuristic to obtain mixed-integer solutions to OPBS; Section 6 presents corresponding computational results. Section 7 summarizes all computational results, and Section 8 concludes the paper.

2. A New Model for OPBS

Beginning with a standard IP for OPBS, this section develops a new MIP for that problem; defines a useful restriction and a useful relaxation of the MIP; and then validates the key, novel feature of the MIP, which is the modeling of fractional block extraction.

2.1. A Mixed-Integer Programming Formulation

We use the "C-PIT" IP of Chicoisne et al. (2012) as a starting point and describe a new MIP for OPBS by (i) imposing lower as well as upper bounds on resource consumption in each period and (ii) allowing selective, fractional block extraction. Normally, spatial-precedence constraints (2), together with strict integrality of variables, enforce pit-wall slope restrictions, but we show that these restrictions remain enforced even with the relaxation implied by (ii). As is standard, our model represents the potential mine volume as a three-dimensional grid of blocks having common dimensions, with blocks stacked directly on top of each other. We denote the new model simply as "**MIP**."

Indices and Index Sets

- $b \in \mathcal{B}$ extractable blocks
- $\mathcal{B}_b \subset \mathcal{B}$ all direct spatial predecessors of block b
- $\bar{b} \in \mathcal{B}_b$ if it exists, the direct spatial predecessor that lies above block b
- $\bar{\mathcal{B}}_b \subset \mathcal{B}_b$ $\bar{\mathcal{B}}_b = \{\bar{b}\}$ if block $\bar{b}$ exists, and $\bar{\mathcal{B}}_b = \varnothing$, otherwise
- $\hat{\mathcal{B}}_b \subset \mathcal{B}_b$ $\mathcal{B}_b \backslash \bar{\mathcal{B}}_b$, i.e., the oblique direct spatial predecessors of block b
- $t \in \mathcal{T}$ time periods defining the time horizon; $\mathcal{T} = \{1, \ldots, T\}$
- $r \in \mathcal{R}$ production and processing resources

Data: [*units*]

- v'_{bt} net present value of block b if extracted in period t [dollars]
- v_{bt} $v'_{bt} - v'_{b,t+1}$, with $v'_{b,T+1} \equiv 0$ for all $b \in \mathcal{B}$
- q_{rb} consumption of resource r associated with the extraction of block b [tons] (Note that $q_{rb} = 0$ if r corresponds to processing a waste block b.)
- q^L_{rt} (q^U_{rt}) minimum (maximum) consumption limits for resource r in time period t [tons]

Variables: [*units, if defined*]

- x_{bt} 1 if block b is completely extracted by time period t, 0 otherwise
- y_{bt} fraction of block b extracted by time t; nominally, $y_{b0} \equiv 0$ for all $b \in \mathcal{B}$

Formulation:

MIP:

$$\theta^*_{\mathbf{MIP}} = \max_{\mathbf{x},\mathbf{y}} \sum_{b\in\mathcal{B}} \sum_{t\in\mathcal{T}} v_{bt} y_{bt} \tag{4}$$

$$\text{s.t.} \quad -\sum_{b\in\mathcal{B}} q_{rb}(y_{bt} - y_{b,t-1}) \leqslant -q^L_{rt} \quad \forall\, r \in \mathcal{R},\, t \in \mathcal{T} \tag{5}$$

$$\sum_{b\in\mathcal{B}} q_{rb}(y_{bt} - y_{b,t-1}) \leqslant q^U_{rt} \quad \forall\, r \in \mathcal{R},\, t \in \mathcal{T} \tag{6}$$

$$-(y_{bt} - y_{b,t-1}) \leqslant 0 \quad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{7}$$

$$y_{bt} - x_{\bar{b}t} \leqslant 0 \quad \forall\, b \in \mathcal{B} \mid \bar{\mathcal{B}}_b \neq \varnothing,\, t \in \mathcal{T} \tag{8}$$

$$y_{bt} - y_{b't} \leqslant 0 \quad \forall\, b \in \mathcal{B} \mid \hat{\mathcal{B}}_b \neq \varnothing,\, b' \in \hat{\mathcal{B}}_b,\, t \in \mathcal{T} \tag{9}$$

$$x_{bt} - y_{bt} \leqslant 0 \quad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{10}$$

$$x_{bt} \in \{0,1\} \quad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{11}$$

$$y_{bt} \geqslant 0 \quad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{12}$$

$$y_{b0} \equiv 0 \quad \forall\, b \in \mathcal{B} \tag{13}$$

Through its objective function (4), **MIP** seeks to maximize the total net present value of extracted blocks. For each time period, constraints (5) and (6) restrict minimum and maximum resource consumption, respectively. Constraints (7) enforce (relaxed) temporal-precedence relationships for each block: if a fraction of block b is extracted by time $t-1$, then at least that fraction must be extracted from block b by time t.

Note that some blocks at a mine's surface may be "partial" at the beginning of time period 1 because the undisturbed surface is uneven or because some partial extraction has already taken place. Although we assume that $y_{b0}=0$ for all computational tests, partial blocks could be handled by fixing certain instances of y_{b0} to appropriate nonzero values in (13).

In effect, a standard OPBS model enforces spatial precedence through constraints (8) and (9) while using only binary variables: block b cannot be extracted until all blocks $b' \in \bar{\mathscr{B}}_b \cup \hat{\mathscr{B}}_b$ have been extracted completely. Our OPBS model **MIP** uses a combination of binary and continuous variables to enforce the following, relaxed, spatial-precedence requirements:

- assuming block $\bar{b}$ lies above block b, constraints (8) imply that block b cannot begin to be extracted until $\bar{b}$ is completely extracted; and
- assuming block b has some oblique spatial predecessors, i.e., $\hat{\mathscr{B}}_b \neq \varnothing$, constraints (9) imply that the fraction of block b that is extracted by time t cannot exceed the fraction that is extracted by time t from any $b' \in \hat{\mathscr{B}}_b$.

We validate this use of fractional block extraction in Section 2.3 after describing important ways that we restrict and relax **MIP**.

2.2. Restricting and Relaxing MIP

Later, we need to compare solutions of **MIP** to those of a standard IP for OPBS. Since we derived **MIP** from such an IP, it is easy to recreate it: (i) restrict all variables y_{bt} to be binary; (ii) replace all x_{bt} with y_{bt}; (iii) delete constraints (10)–(12), which have become redundant; and (iv) call the resulting model "**IP**."

As is standard, we begin the solution process for **MIP** by first solving its LP relaxation. Actually, we solve a special form of this relaxation, denoted **RMIP**: this is identical to the LP relaxation of **IP**, just described

After solving **RMIP**, we apply a branch-and-bound heuristic, which dynamically and implicitly enforces (8) and (10) in **MIP**, ensuring that these restrictions hold for every block b such that $\bar{\mathscr{B}}_b \neq \varnothing$. Specifically, $y_{bt} > 0 \Rightarrow y_{\bar{b}t} = 1$, whenever $\bar{b}$ exists. The resulting solution $(\hat{\mathbf{y}}_1^*, \ldots, \hat{\mathbf{y}}_T^*)$ is said to be "MIP valid" for **MIP**.

2.3. Validating Fractional Block Extraction

The validity of constraints (8) as relaxed versions of constraints (1) is clear: no fraction of a block b can be extracted until the block $\bar{b}$, which lies directly above b, is completely extracted. That statement is true whether the fraction in question lies between 0 and 1 or the "fraction" must be exactly 0 or 1. The validity of constraints (9) as relaxed versions of (2) is less clear, however. This section demonstrates the geometrical validity of the fractional block extraction modeled in **MIP** and then solves some small instances of **MIP** and **IP** to investigate practical implications. We note that Gershon (1983) also considers fractional block extraction but uses a more restrictive definition: fractional extraction of a block b is allowed provided that all blocks $b' \in \mathscr{B}_b$ have been extracted fully.

2.3.1. Theory. In **IP**, each spatial-precedence constraint defined by (2) restricts the local pit-wall slope angle. We show here that each constraint defined by (9) restricts the local slope in a solution to **MIP** to an angle that is no steeper than that enforced by the corresponding constraint in **IP**; Figure 2 illustrates. We demonstrate informally, first, assuming that each block is a cube with each side having a length of one in arbitrary units.

Figure 2(a) reflects a standard set of spatial-precedence relations: block b_0 cannot be extracted until each block $b' \in \bar{B}_{b_0} \cup \hat{\mathscr{B}}_{b_0} = \{\bar{b}_0\} \cup \{b_1, b_2, b_3, b_4\}$ is extracted. We simplify to the two-dimensional model of Figure 2(b) so that block b_0 has oblique predecessors $\widehat{\mathscr{B}}_{b_0} = \{b_1, b_2\}$, only.

Dropping the subscript t for simplicity, these constraints define the standard spatial-precedence relationships for the two-dimensional example:

$$x_{b_0} - x_{\bar{b}_0} \leqslant 0 \tag{14}$$

$$x_{b_0} - x_{b_1} \leqslant 0 \tag{15}$$

$$x_{b_0} - x_{b_2} \leqslant 0. \tag{16}$$

Assuming that the block below b_0 is not extracted, standard slope restrictions associated with b_0 may be interpreted as follows (see Figure 2(c)):

(i) constraint (14) requires that $\bar{b}_0$ be extracted completely before b_0 is extracted;

(ii) constraint (15) requires that the slope φ_{01}, measured from the center of the extracted face of b_0 to the center of the extracted face of b_1, not exceed $\arctan(d_{01}/D_{01}) = \arctan(1/1) = 45°$; and, similarly,

(iii) constraint (16) requires that the slope φ_{02}, which is analogous to φ_{01}, not exceed $\arctan(d_{02}/D_{02}) = \arctan(1/1) = 45°$.

Thus, constraints (14)–(16) enforce pit-slope restrictions of "at most 45°."

Now, when allowing fractional block extraction in **MIP** (see Figure 2(d)), the following constraints replace (14)–(16), respectively:

$$y_{b_0} - x_{\bar{b}_0} \leqslant 0 \tag{17}$$

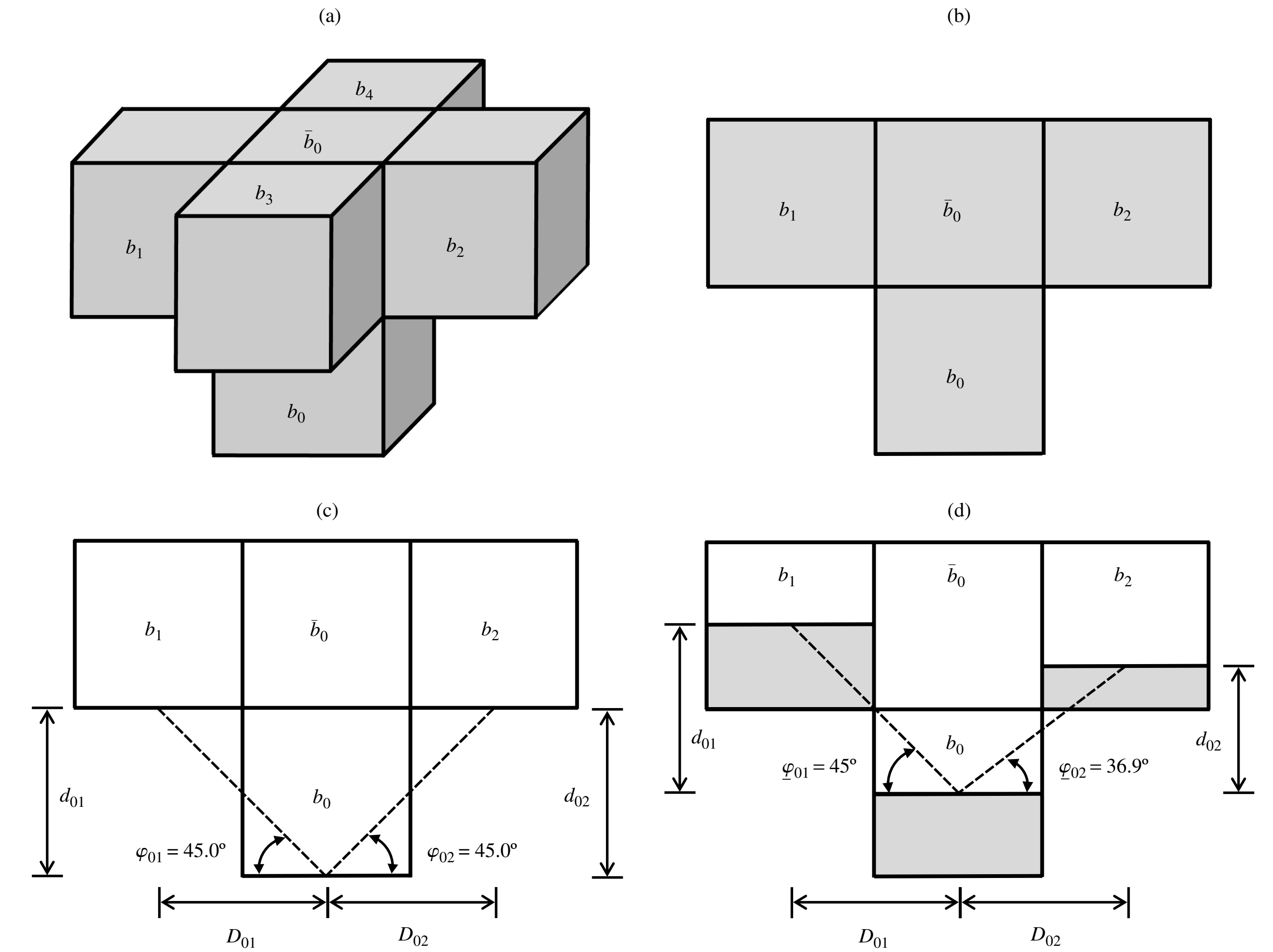
$$y_{b_0} - y_{b_1} \leqslant 0 \tag{18}$$

$$y_{b_0} - y_{b_2} \leqslant 0. \tag{19}$$

If $y_{b_0} = 0$, then block b_0 has not been extracted, and these constraints impose no restrictions on the extraction of $\bar{b}_0$, b_1, or b_2. (An analogous situation arises in the all-binary model when $x_{b_0} = 0$; see (14)–(16).) If $y_{b_0} = 1$, and the block below b_0 has not been extracted, then standard slope restrictions are enforced; i.e., $y_{b_1} = x_{\bar{b}_0} = y_{b_2} = 1$. Thus, we only need to ensure that, when $0 < y_{b_0} < 1$,

(i) block $\bar{b}_0$ is completely extracted;

Figure 2. Fractional extraction maintains slope restrictions.



Notes. Each block side has unit length. (a) Shows the six blocks, b_0, $\bar{b}_0$, $b_1, \ldots, b_4$, that are involved in maintaining slopes as measured from b_0. (b) Simplifies to two dimensions for purposes of illustration. (c) Shows how slope angles are defined from the blocks' *extracted faces* (i.e., the bottom faces of the blocks) when fractional extraction is disallowed. Assuming the blocks (not shown) directly beneath b_0, b_1, and b_2 have not been extracted, φ_{01} measures the slope from b_0 toward b_1 and φ_{02} measured from b_0 toward b_2. The illustrated angles of 45° are the maximum allowable: $\varphi_{01} = \arctan(d_{01}/D_{01}) = \arctan(1/1) = 45°$ and $\varphi_{02} = \arctan(d_{02}/D_{02}) = \arctan(1/1) = 45°$. (d) Shows slope angles $\underline{\varphi}_{01}$ and $\underline{\varphi}_{02}$, corresponding to φ_{01} and φ_{02}, respectively, but with fractional extraction allowed; unshaded regions have been extracted, while shaded regions have not. Now we see that $\underline{\varphi}_{01} = \arctan(d_{01}/D_{01}) = \arctan(1/1) = 45° = \varphi_{01}$, but $\underline{\varphi}_{02} = \arctan(d_{02}/D_{02}) = \arctan(0.75/1) = 36.9° < 45° = \varphi_{02}$.

(ii) the slope $\underline{\varphi}_{01}$ between extracted faces of b_0 and b_1 does not exceed 45°; and

(iii) the analogous slope $\underline{\varphi}_{02}$ does not exceed 45°.

Next, Figure 2(d) illustrates the case in which $y_{b_0} = y_{b_1} = 0.5$, $y_{\bar{b}_0} = 1.0$, and $y_{b_2} = 0.75$. Now,

(i) is satisfied in general through constraint (17);

(ii) is satisfied because

$\underline{\varphi}_{01} = \arctan(d_{01}/D_{01}) = \arctan(1/1) = 45° = \varphi_{01}$; and

(iii) is satisfied because

$\underline{\varphi}_{02} = \arctan(d_{02}/D_{02}) = \arctan(0.75/1) = 36.9° < 45°$.

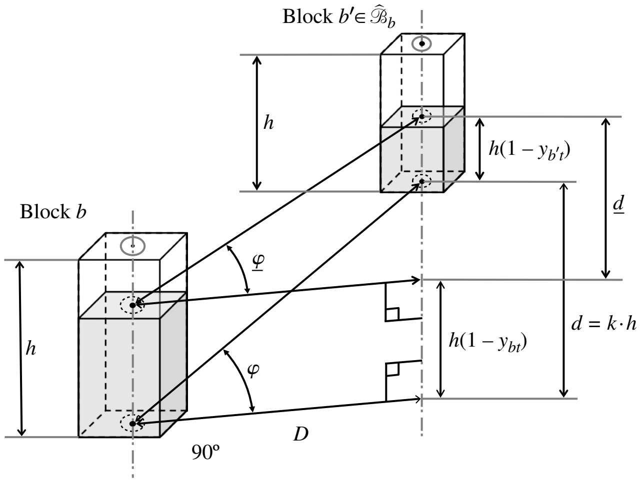
In general, with respect to (ii) and (iii) when $0 < y_{b_0} < 1$, a solution to **MIP** defines a slope $\underline{\varphi}$ between b_0 and any block $b' \in \hat{\mathscr{B}}_b$ such that $\underline{\varphi} = \arctan((1 + (1 - y_{b'}) - (1 - y_{b_0}))/1) \leqslant 45°$ because (9) $\Rightarrow 1 - y_{b'} + y_{b_0} \leqslant 1$. Thus, the relaxed model also enforces pit-slope restrictions of "at most 45°."

In the following theorem, we extend the discussion above to more general slope relationships and more general block geometries.

Theorem 1. *In allowing fractional block extraction, an instance of* **MIP** *ensures that pit-wall slope angles do not exceed those that the corresponding (all-binary) instance of* **IP** *would enforce, provided that each block has common dimensions and a rectangular base.*

Proof. Given the discussion above, it suffices to show that for any pair of blocks (b, b') such that $b' \in \hat{\mathscr{B}}_b$, the relevant constraint from (9) enforces an angle between the extracted faces of b and b' that does not exceed the angle enforced

Figure 3. Figure for proof of Theorem 1.



Notes. Block shapes are exaggerated for clarity; h denotes the blocks' common height. The shaded portions of the blocks illustrate unextracted portions that would be valid in **MIP**. Assuming an all-integer version of **MIP**, φ indicates the enforced angle from the extracted face of b to the extracted face of $b' \in \hat{\mathcal{B}}_b$. For **MIP**, $\underline{\varphi}$ indicates the corresponding angle when the extracted fraction of block b is y_{bt} and the extracted fraction of b' is $y_{b't}$. Note that $y_{bt} < y_{b't}$ in the figure, satisfying constraints (9) in **MIP**.

constraint in (2). Figure 3 illustrates a general case in which (i) each block has height h; (ii) the bottom center of block b is located at general grid coordinates (x, y, z); and (iii) the bottom center of block b' is located at coordinates $(\mathrm{x} + D_\mathrm{x}, \mathrm{y} + D_\mathrm{y}, \mathrm{z} + d)$ such that $D_\mathrm{x} \geqslant 0$, $D_\mathrm{y} \geqslant 0$, $D_\mathrm{x} + D_\mathrm{y} > 0$, and $d = k \cdot h$ for some positive integer k. (None of x, y, z, D_x, or D_y is indicated in the figure.)

Now, $D = (D_\mathrm{x}^2 + D_\mathrm{y}^2)^{1/2} > 0$ defines the horizontal distance between the blocks' centers, as indicated in the figure, and we know from earlier discussion that constraints (2) in the all-integer version of **MIP** enforce a slope of $\varphi = \arctan(d/D)$, when that slope is defined. For **MIP**, let the corresponding angle be $\underline{\varphi}$, and assume that this angle is defined in period t. Thus,

$$\underline{\varphi} = \arctan(\underline{d}/D) \tag{20}$$
$$= \arctan((d + (1 - y_{b't})h - (1 - y_{bt})h)/D) \tag{21}$$
$$= \arctan((d + (y_{bt} - y_{b't})h)/D) \tag{22}$$
$$\leqslant \arctan(d/D) \text{ because constraints (9) require that } y_{bt} - y_{b't} \leqslant 0 \tag{23}$$
$$= \varphi. \quad \square \tag{24}$$

2.3.2. Practical Implications. Here, using data sets for five different open-pit mine scenarios, we compare solutions of **MIP** to solutions of **IP**. The data sets can create problem instances that cover 10,819, 18,300 and 25,620 blocks, for 1 to 20 time periods, although we use a maximum of 10 time periods here. Cullenbine et al. (2011) use the same data sets, plus one denoted "BD10819F," and we follow that paper's naming conventions, with each data set's label specifying the relevant number of blocks. (The current section omits results for BD10819F simply because of space limitations in Table 1. Subsequent computational results for decomposition-based methods do cover that data set, however.)

We consider only small values of T so that the problems can be solved using LP-based branch and bound, that is, without requiring the decomposition techniques developed later in the paper. Computations are performed on a Lenovo W541 laptop computer having a 64-bit, quad-core, Intel processor running at 2.9 GHz. The computer has 16 GB RAM and runs the Windows 7 Professional operating system. A C++ program generates all models and CPLEX 12.6 (IBM Corp. 2014) solves them. We override CPLEX's default parameters in five different ways:

(i) the solver may use at most four threads (Threads = 4);

(ii) a relative optimality tolerance of 0.1% applies (EpGap = 0.001);

(iii) computations are limited to 7,200 seconds of elapsed time (TiLim = 7,200);

(iv) the barrier algorithm solves root-node LPs (RootAlg invokes BarAlg); and

(v) because of the cumulative nature of the models' variables, branching priorities for x_{bt} are set to t, i.e., priority increases with t.

Table 1 displays results and "Notes" provide detailed explanations of the table's entries. We highlight the following points from these results:

- Average profit for a solution to **MIP** compared to **IP** improves by at least 1.0% but no more than 1.9%; the inability to solve many instances of **IP** accurately makes more precise statements impossible.
- No clear trend in improved profit for **MIP** versus **IP** appears as T increases, i.e., as the mine pit expands.
- Because counting all fractional variables y_{bt} in **MIP** would imply some double counting when $0 < y_{bt} \approx y_{b,t+1} < 1$, the table lists the number of fractional variables only for the last time period, T. No clear trend in that number appears as T increases.
- The number of fractional variables y_{bT} in **MIP** may constitute a small percentage of the total number of positive variables in period T (less than 3% for the two largest instances of BD25620A), or it may constitute a substantial percentage (almost 50% for the smallest instance of BD18300A).
- Despite having more variables and constraints, the flexibility provided by fractional block extraction in **MIP** makes that model much easier to solve than **IP**.

2.3.3. Conclusion. The qualities of solutions to **MIP** deserve further investigation, but Section 2.3 has shown that (i) **MIP**'s fractional block extraction leads to extraction schedules that satisfy pit-wall slope restrictions; (ii) solutions to **MIP** may yield profits that are 1%–2% higher than with **IP**; and (iii) **MIP** has computational advantages over

Table 1. Validating fractional block extraction: Comparing solutions of **MIP** and **IP** produced by a standard branch-and-bound method.

		MIP								**IP**							
Model name[a]	T	Vars.[b] (num.)	Cons.[c] (num.)	Soln. time[d] (sec.)	B&B nodes[e] (num.)	$\hat{\theta}^{f}_{\mathbf{MIP}}$ ($)	Ones[g] (num.)	Fracs.[h] (num.)	Avg. fracs.[i] (num.)	Soln. time[d] (sec.)	B&B nodes[e] (num.)	$\hat{\theta}^{j}_{\mathbf{IP}}$ ($)	$\bar{\theta}^{k}_{\mathbf{IP}}$ ($)	Ones[g] (num.)	Opt. gap[l] (%)	Obs. profit incr.[m] (%)	Min. profit incr.[n] (%)
BD10819A	2	43,285	129,937	41.4	0	5,593,649	370	111	55.5	29.4	251	5,464,135	5,468,601	398	0.1	2.3	2.3
	4	86,569	270,693	220.8	0	8,312,442	773	92	23.0	7,201.3	778,769	8,270,561	8,315,961	775	0.5	0.5	0.0
	6	129,853	411,449	541.9	0	9,427,565	1,082	177	29.5	7,202.5	99,474	9,325,300	9,389,419	1,141	0.7	1.1	0.4
	8	173,137	552,205	2,451.8	3,927	10,252,760	1,463	219	27.4	7,203.3	77,603	10,178,989	10,248,853	1,497	0.7	0.7	0.0
	10	216,421	692,961	2,490.3	0	10,921,067	1,783	150	15.0	7,202.9	11,060	10,808,960	10,901,314	1,839	0.8	1.0	0.2
BD18300A	1	36,605	90,536	5.0	38	19,227,602	37	33	33.0	3.0	7	18,801,982	18,811,748	38	0.1	2.3	2.2
	2	73,209	199,372	31.8	72	37,448,347	44	34	17.0	3,429.2	31,943	36,638,599	36,638,599	76	0.1	2.2	2.2
	3	109,813	308,208	99.1	137	54,342,392	81	53	17.7	7,206.3	4,878	52,734,812	54,074,466	116	2.5	3.0	0.5
	4	146,417	417,044	413.8	1,342	70,033,174	140	18	4.5	7,201.8	1,569	68,242,623	69,773,789	155	2.2	2.6	0.4
	5	183,021	525,880	3,491.2	7,390	83,970,633	181	24	4.8	7,202.3	2,949	80,791,165	83,675,968	193	3.6	3.9	0.4
BD18300B	1	36,605	90,536	1.2	0	22,971,028	37	2	2.0	1.4	0	22,672,289	22,677,547	38	0.0	1.3	1.3
	2	73,209	199,372	52.5	0	40,622,148	75	3	1.5	1,404.7	1,033	40,232,788	40,268,474	76	0.1	1.0	0.9
	3	109,813	308,208	39.6	0	55,170,601	111	7	2.3	4,987.9	2,142	54,626,239	54,661,314	115	0.1	1.0	0.9
	4	146,417	417,044	104.8	0	67,664,348	147	16	4.0	7,201.6	1,803	66,966,310	67,276,808	153	0.5	1.0	0.6
	5	183,021	525,880	476.7	0	78,475,904	185	7	1.4	7,202.3	623	77,294,665	78,128,742	193	1.1	1.5	0.4
BD25620A	1	51,245	133,972	2.6	0	28,081,757	35	4	4.0	2.8	0	27,672,169	27,683,547	38	0.0	1.5	1.4
	2	102,489	293,564	80.6	0	49,559,101	74	4	2.0	349.1	732	48,880,305	48,928,136	77	0.1	1.4	1.3
	3	153,733	453,156	327.1	0	67,577,898	110	7	2.3	7,202.0	4,807	66,660,574	66,778,644	115	0.2	1.4	1.2
	4	204,977	612,748	297.0	0	83,070,180	150	4	1.0	7,202.6	2,412	81,937,083	82,379,219	154	0.5	1.4	0.8
	5	256,221	772,340	1,766.3	0	96,291,336	187	5	1.0	7,203.0	2,372	95,052,790	95,837,081	192	0.8	1.3	0.5
BD25620B	1	51,245	133,972	6.5	0	12,953,190	35	4	4.0	7.4	0	12,593,340	12,594,550	38	0.0	2.9	2.9
	2	102,489	293,564	71.5	0	23,660,764	51	30	15.0	7,202.3	48,043	23,021,102	23,302,878	76	1.2	2.8	1.5
	3	153,733	453,156	201.3	0	33,519,809	92	30	10.0	7,205.3	36,627	32,604,828	33,132,682	114	1.6	2.8	1.2
	4	204,977	612,748	1,791.1	393	42,592,824	109	68	17.0	7,207.6	20,160	41,284,258	42,113,244	153	2.0	3.1	1.1
	5	256,221	772,340	1,628.6	156	50,878,612	176	81	16.2	7,203.2	5,757	49,191,860	50,477,075	191	2.6	3.4	0.8

[a]Name of the model, which indicates the number of blocks that it contains. [b]Number of variables in **MIP**; **IP** has 50% fewer. [c]Number of constraints in **MIP**; **IP** averages 17% fewer. [d]Wall-clock time for CPLEX to solve the model instance, **MIP** or **IP**, but limited to 7,200 seconds. This time omits about three seconds of overhead from model generation. [e]Number of nodes in CPLEX's branch-and-bound tree. [f]Best solution's objective value for **MIP**. Because all solutions to **MIP** satisfy the tight optimality tolerance of 0.1%, no column for "observed relative optimality gap" is provided and computations of increased profit for **MIP** versus **IP** assume that $\hat{\theta}_{\mathbf{MIP}} = \theta^*_{\mathbf{MIP}}$. [g]Number of binary variables x_{bT} such that $x_{bT} = 1$. [h]Number of fractional variables, i.e., number of continuous variables y_{bT} such that $0 < y_{bT} < 1$. [i]Number of fractional variables y_{bT}. [j]Best solution's objective value for **IP**, although this may not satisfy the desired relative optimality tolerance of 0.1%. [k]Best upper bound on $\theta^*_{\mathbf{IP}}$. [l]Relative optimality gap observed in solution to **IP**. [m]Observed percentage increase in profit for **MIP** versus **IP**: $100\% \times (\hat{\theta}_{\mathbf{MIP}} - \hat{\theta}_{\mathbf{IP}})/\hat{\theta}_{\mathbf{IP}}$. [n]Minimum possible percentage increase in profit for **MIP** versus **IP**, which hypothesizes that a solution to **IP** can be found with objective value $\bar{\theta}_{\mathbf{IP}}$: $100\% \times (\hat{\theta}_{\mathbf{MIP}} - \bar{\theta}_{\mathbf{IP}})/\bar{\theta}_{\mathbf{IP}}$.

IP, at least when trying to solve those models by LP-based branch and bound. As discussed later, we have attempted to solve **IP** using the decomposition-based heuristic that successfully solves **MIP**. The branch-and-bound portion of that heuristic requires orders of magnitude more time when operating on **IP** than when operating on **MIP**, however. Thus, **MIP** also appears to have computational advantages over **IP** in the context of decomposition.

3. Solving RMIP by Decomposition

This section describes how to solve **RMIP** using standard nested Benders decomposition and then develops our new hierarchical variant of that decomposition, HBD. A novel application of aggregate resource constraints, made possible by the use of cumulative variables, turns out to be crucial for the computational effectiveness of HBD. For ease of exposition, we add a dummy time period $t = T+1$ and present **RMIP** using matrix notation.

Additional Notation

- $\mathcal{T}^+$ $\{1, \ldots, T+1\}$
- $\mathcal{T}^{0+}$ $\{0, \ldots, T+1\}$
- $\bar{T}$ $T+1$
- $\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}$ initial conditions and upper bound on ending conditions for the mine, respectively; $\hat{\mathbf{y}}_0 = \mathbf{0}$ and $\hat{\mathbf{y}}_{\bar{T}} = \mathbf{1}$ may be assumed
- $\mathbf{y}_t$ $(y_{1t}, \ldots, y_{|\mathcal{B}|t})^\top$ for all $t \in \mathcal{T}^{0+}$; however, $\mathbf{y}_0 \equiv \hat{\mathbf{y}}_0$ and $\mathbf{y}_{\bar{T}} \equiv \hat{\mathbf{y}}_{\bar{T}}$
- $\mathbf{v}_t$ $(v_{1t}, \ldots, v_{|\mathcal{B}|t})$ for all $t \in \mathcal{T}^{0+}$; however, $\mathbf{v}_0 = \mathbf{v}_{\bar{T}} = \mathbf{0}$
- $\mathbf{q}_t$ $((-q^L_{1t}, \ldots, -q^L_{|\mathcal{R}|t}), (q^U_{1t}, \ldots, q^U_{|\mathcal{R}|t}))^\top$ for all $t \in \mathcal{T}^+$, where $-q^L_{b,\bar{T}}$ and $q^U_{b,\bar{T}}$ are large enough to make constraints (26) vacuous when $t = \bar{T}$

The continuous relaxation of **MIP** then has the following form:

$$\textbf{RMIP}: \quad \theta^*_{\textbf{RMIP}} = \max_{\mathbf{y}_0, \ldots, \mathbf{y}_{\bar{T}}} \sum_{t \in \mathcal{T}^+} \mathbf{v}_t \mathbf{y}_t \tag{25}$$

$$\text{s.t.} \quad A(\mathbf{y}_t - \mathbf{y}_{t-1}) \leqslant \mathbf{q}_t \quad \forall t \in \mathcal{T}^+ \tag{26}$$

$$-I(\mathbf{y}_t - \mathbf{y}_{t-1}) \leqslant \mathbf{0} \quad \forall t \in \mathcal{T}^+ \tag{27}$$

$$H_t \mathbf{y}_t \leqslant \mathbf{0} \quad \forall t \in \mathcal{T} \tag{28}$$

$$\mathbf{y}_t \geqslant \mathbf{0} \quad \forall t \in \mathcal{T} \tag{29}$$

$$\mathbf{y}_0 \equiv \hat{\mathbf{y}}_0 \tag{30}$$

$$\mathbf{y}_{\bar{T}} \equiv \hat{\mathbf{y}}_{\bar{T}}, \tag{31}$$

where

1. the matrix A derives from constraints (5)–(6);
2. the $|\mathcal{B}| \times |\mathcal{B}|$ identity matrix I in (27) derives from constraints (7);
3. the matrix H_t derives from constraints (9) as well as the constraints that result from aggregating pairs of constraints taken from (8) and (10); and
4. we define a feasible solution $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_0, \ldots, \hat{\mathbf{y}}_{\bar{T}})$ to **RMIP** to be *MIP-valid*, if there exists $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_{\bar{T}})$ such that $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a feasible solution to **MIP**.

Note that upper bounds $\mathbf{y}_t \leqslant \mathbf{1}$ for all t are implied by (27) and (31).

Observe that H_t is actually stationary in our application (i.e., $H_t = H$ for all t), but it need not be. The matrix A need not be stationary for standard NBD, but we exploit stationarity of A when using aggregate resource constraints. (The final paragraph of Section 3.6 discusses extensions to "nearly stationary" matrices A_t.) To simplify later descriptions of NBD and HBD, we develop necessary notation here in the context of standard, "non-nested" Benders decomposition applied to a staircase LP.

To begin, let $\underline{t}, \bar{t} \in \mathcal{T}^{0+}$, $\underline{t} < \bar{t}$, and suppose that $\hat{\mathbf{y}}_{\underline{t}}$ and $\hat{\mathbf{y}}_{\bar{t}}$ are given such that (i) $\hat{\mathbf{y}}_0 \leqslant \hat{\mathbf{y}}_{\underline{t}} \leqslant \hat{\mathbf{y}}_{\bar{t}} \leqslant \hat{\mathbf{y}}_{\bar{T}}$, (ii) $H_{\underline{t}} \hat{\mathbf{y}}_{\underline{t}} \leqslant \mathbf{0}$, and (iii) $H_{\bar{t}} \hat{\mathbf{y}}_{\bar{t}} \leqslant \mathbf{0}$. That is, except for not necessarily being MIP-valid, $\hat{\mathbf{y}}_{\underline{t}}$ defines a valid pit through time period $\underline{t}$, which is "nested" inside of the pit defined by $\hat{\mathbf{y}}_{\bar{t}}$.

Given the above conditions, the following model generalizes the standard concept of a cost-to-go function to a *cost-to-operate function*, which covers the end of period $\underline{t}$ to the beginning of period $\bar{t}$:

$$\theta^*(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}}) \equiv \max_{\mathbf{y}_{\underline{t}}, \ldots, \mathbf{y}_{\bar{t}}} \sum_{t=\underline{t}+1}^{\bar{t}-1} \mathbf{v}_t \mathbf{y}_t \tag{32}$$

$$\text{s.t.} \quad (26), (27) \text{ for } t = \underline{t}+1, \ldots, \bar{t}, \text{ only} \tag{33}$$

$$(28), (29) \text{ for } t = \underline{t}+1, \ldots, \bar{t}-1, \text{ only} \tag{34}$$

$$\mathbf{y}_{\underline{t}} \equiv \hat{\mathbf{y}}_{\underline{t}} \tag{35}$$

$$\mathbf{y}_{\bar{t}} \equiv \hat{\mathbf{y}}_{\bar{t}}. \tag{36}$$

Remark 1. Strictly speaking, $\theta^*(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$ should display an additional identifier such as a subscript $[\underline{t}, \bar{t}]$, but the relevant information will be clear from the function's arguments so we omit it. Note also that we have dropped the subscript **RMIP** for notational simplicity.

Remark 2. The actual implementation of **RMIP** uses elastic resource constraints, so given conditions (i)–(iii) just specified, $\theta^*(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$ always has a finite value. For simplicity here, we omit explicit representation of elastic constructs, but Appendices A and B do cover the details.

Remark 3. The optimal objective value for **RMIP** is $\theta^*_{\textbf{RMIP}} = \theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})$.

Now, for any $\underline{t}, t', \bar{t} \in \mathcal{T}^{0+}$ with $\underline{t} < t' < \bar{t}$, define

$$Y_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}}) = \left\{ \hat{\mathbf{y}}_{\underline{t}} \leqslant \mathbf{y}_{t'} \leqslant \hat{\mathbf{y}}_{\bar{t}} \;\middle|\; \begin{array}{c} H_{t'} \mathbf{y}_{t'} \leqslant \mathbf{0}, \\ \text{and } A(\mathbf{y}_{t'} - \hat{\mathbf{y}}_{t'-1}) \leqslant \mathbf{q}_{t'} \text{ if } t' = \underline{t}+1 \\ \text{and } A(\hat{\mathbf{y}}_{t'+1} - \mathbf{y}_{t'}) \leqslant \mathbf{q}_{t'+1} \text{ if } t' = \bar{t}-1 \end{array} \right\}. \tag{37}$$

Note that bounds $\hat{\mathbf{y}}_{\underline{t}} \leqslant \mathbf{y}_{t'} \leqslant \hat{\mathbf{y}}_{\bar{t}}$ must hold for any model using cumulative variables $\mathbf{y}_{t'}$; see constraints (9).

Next, consider the optimal solution of **RMIP** computed using two functions of $\mathbf{y}_{t'}$:

$$\theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) = \theta_{t'}(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) \tag{38}$$

$$\equiv \max_{\mathbf{y}_{t'} \in Y_{t'}(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_{t'}\mathbf{y}_{t'} + \theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}) + \theta^*(\mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}}). \tag{39}$$

We know that constraint (37) is valid because $\mathbf{y}_{t'}$ must satisfy the constraints of a (relaxed) pit that lies "nested between" the pits defined by $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_{\bar{T}}$. From standard LP theory, we also know that the functions $\theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'})$ and $\theta^*(\mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ are piecewise linear and concave for $\mathbf{y}_{t'} \in Y_{t'}(\hat{\mathbf{y}}_{t_0}, \hat{\mathbf{y}}_{\bar{T}})$.

To solve **RMIP**, standard Benders decomposition would

1. view $\theta^*(\hat{\mathbf{y}}_1, \mathbf{y}_{t'}) + \theta^*(\mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ as a single, piecewise-linear, concave function, say $\psi^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$;
2. replace $\psi^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ with a piecewise-linear approximating function, say $\bar{\bar{\psi}}(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$, such that $\bar{\bar{\psi}}(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}}) \geqslant \psi^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ for all $\mathbf{y}_{t'} \in Y_{t'}$; and
3. solve a sequence of nested subproblems that successively improves the approximating function and converges to an optimal solution. (Of course, the decomposition algorithm typically terminates when the best solution found satisfies a prespecified optimality criterion.)

Maintaining separate approximating functions for $\theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'})$ and $\theta^*(\mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ leads to a multicut master problem defined with respect to two separate subproblems (Birge and Louveaux 1988). Two special cases arise, however: if $t' = 1$, then (39) simplifies to

$$\theta_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) = \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_1\mathbf{y}_1 + \theta^*(\mathbf{y}_1, \hat{\mathbf{y}}_{\bar{T}}), \tag{40}$$

and if $t' = T$, then (39) simplifies to

$$\theta_T(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) = \max_{\mathbf{y}_T \in Y_T(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_T). \tag{41}$$

Section 3.5 shows how to recursively decompose *both* functions $\theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_{t'})$ and $\theta^*(\mathbf{y}_{t'}, \hat{\mathbf{y}}_{\bar{T}})$ to create a general tree decomposition. First, however, Section 3.1 presents a standard forward recursion for NBD, Section 3.2 indicates how applying aggregate resource constraints may improve the decomposition algorithm's efficiency, and Sections 3.3 and 3.4 describe simple variants of NBD that make use of those constraints. From this point on, "subproblem" implies "nested subproblem."

3.1. A Forward Recursion for Nested Benders Decomposition (FBD)

RMIP exhibits a staircase structure, which seems ideal for solution through a nested decomposition, either nested Dantzig-Wolfe decomposition (Glassey 1973) or NBD (Ho and Manne 1974). We implement NBD because constructing a MIP valid solution for **MIP** from the continuous solution to **RMIP** seems easier with NBD. For reference, then, this section describes a standard version of NBD. We note that NBD was first proposed for solving deterministic problems but has become particularly important for solving multistage stochastic programs (Birge 1997). Perhaps this fact will make NBD useful for solving certain stochastic versions of OPBS, for example, with probabilistically modeled net present values for blocks.

The following equations describe a forward recursion of **RMIP**, which leads to a "forward NBD" (FBD). This is the classical nested Benders decomposition of Wittrock (1985).

$$\theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})$$

$$= \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_1\mathbf{y}_1 + \cancelto{0}{\theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_1)} + \theta^*(\mathbf{y}_1, \hat{\mathbf{y}}_{\bar{T}}) \tag{42}$$

$$= \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_1\mathbf{y}_1 + \theta_2(\mathbf{y}_1, \hat{\mathbf{y}}_{\bar{T}}) \tag{43}$$

$$= \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_1\mathbf{y}_1 + \Big\{ \max_{\mathbf{y}_2 \in Y_2(\mathbf{y}_1, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_2\mathbf{y}_2 + \theta_3(\mathbf{y}_2, \hat{\mathbf{y}}_{\bar{T}}) \Big\} \tag{44}$$

$$\vdots$$

$$= \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_1\mathbf{y}_1 + \Big\{ \max_{\mathbf{y}_2 \in Y_2(\mathbf{y}_1, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_2\mathbf{y}_2 + \Big\{ \cdots \Big\{ \max_{\mathbf{y}_T \in Y_T(\mathbf{y}_{T-1}, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \cancelto{0}{\theta^*(\mathbf{y}_T, \hat{\mathbf{y}}_{\bar{T}})} \Big\} \cdots \Big\} \Big\} \tag{45}$$

More simply, the FBD recursion may be summarized through the repeated application of the following relationships, starting with $\underline{t} = 0$ and with fixed values $\bar{t} = \bar{T}$, $\mathbf{y}_0 = \hat{\mathbf{y}}_0$, and $\mathbf{y}_{\bar{T}} = \hat{\mathbf{y}}_{\bar{T}}$:

$$\theta^*(\mathbf{y}_{\underline{t}}, \mathbf{y}_{\bar{t}}) = \theta_{\underline{t}+1}(\mathbf{y}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{T}}) \tag{46}$$

$$= \max_{\mathbf{y}_{\underline{t}+1} \in Y_{\underline{t}+1}(\mathbf{y}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_{\underline{t}+1}\mathbf{y}_{\underline{t}+1} + \theta^*(\mathbf{y}_{\underline{t}+1}, \hat{\mathbf{y}}_{\bar{T}}). \tag{47}$$

To exploit the FBD recursion, an FBD algorithm replaces each function $\theta_{t+1}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{T}})$ with an upper-bounding, piecewise-linear, concave approximation $\bar{\bar{\theta}}^k_{t+1}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{T}})$, which it improves in each iteration k using standard optimality and feasibility cuts (although our implementation requires only optimality cuts). In our FASTPASS implementation (see below), this approximating function actually depends on dual variables $\hat{\boldsymbol{\pi}}^{k'}_{t+1}$, evaluated in iterations $k' = 1, \ldots, k-1$. Thus, more explicitly,

$$\max_{\mathbf{y}_t \in Y_t(\hat{\mathbf{y}}_{t-1}, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_t\mathbf{y}_t + \bar{\bar{\theta}}^k_{t+1}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{T}}; \hat{\boldsymbol{\pi}}^1_{t+1}, \ldots, \hat{\boldsymbol{\pi}}^{k-1}_{t+1}) \tag{48}$$

defines the general, primal subproblem for period t and iteration k. Slightly different dual subproblems are also solved, as described below.

One might "process" subproblems using a variety of sequences or methods, and we simply use the FASTPASS method identified by Wittrock (1985) as the most effective method among the three tests. (See also Gassman 1990.) Specifically, iteration k includes (i) a *primal pass*, which, for $t = 1, \ldots, T$, uses $\hat{\mathbf{y}}_{t-1}$ and the period-t subproblem to solve for $\hat{\mathbf{y}}_t$, and (ii) a *dual pass*, which, for

$t = T-1, \ldots, 2$, uses the most recent dual solution $\hat{\boldsymbol{\pi}}_{t+1}$ to solve for $\hat{\boldsymbol{\pi}}_t$, which generates a new Benders cut to improve the approximating function for period $t-1$. (The dual pass initializes with $\hat{\boldsymbol{\pi}}_T$ taken from the last step in the preceding primal pass.) Figure 4 illustrates a generic iteration k of this algorithm for $T = 4$; Figure 5(a) gives a condensed illustration, which simplifies comparison to other algorithmic variants. Appendix A provides details on the subproblems solved in an FBD algorithm, including the elastic formulation, an expanded representation of the vector $\hat{\boldsymbol{\pi}}_t$, and the recursive definition of optimality cuts.

3.2. Aggregate Resource Constraints

Computational results in Section 4 show that standard FBD runs slowly. To improve upon these results, we exploit aggregate resource constraints. Note that, in effect, we have already exploited aggregations of temporal-precedence constraints (27) to define the bounds $\hat{\mathbf{y}}_{\underline{t}} \leqslant \mathbf{y}_{t'} \leqslant \hat{\mathbf{y}}_{\bar{t}}$ used in $Y_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$; see Equation (37).

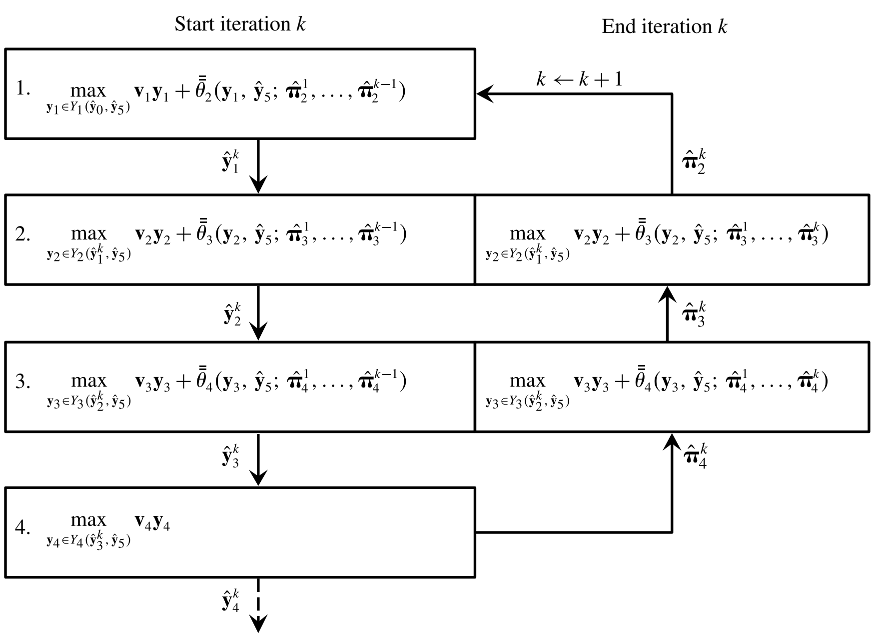
The idea is simple: total resource consumption from the end of time period t_1 to the end of time period t_2, $t_2 > t_1$, must satisfy both lower and upper bounds on resource consumption accumulated from periods $t_1 + 1$ through t_2. More specifically, by summing constraints (26) appropriately and noting the cancellations that occur because of the stationary matrix A, we see that $Y_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$ as used in (39) can be replaced by the following set of constraints, again for any $\underline{t}, t', \bar{t} \in \mathcal{T}^{0+}$ with $\underline{t} < t' < \bar{t}$:

$$Y^+_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}}) = \left\{ \hat{\mathbf{y}}_{\underline{t}} \leqslant \mathbf{y}_{t'} \leqslant \hat{\mathbf{y}}_{\bar{t}} \;\middle|\; \begin{array}{c} H_{t'}\mathbf{y}_{t'} \leqslant \mathbf{0}, \\ \text{and } A(\mathbf{y}_{t'} - \hat{\mathbf{y}}_{\underline{t}}) \leqslant \sum_{\tau=\underline{t}+1}^{t'} \mathbf{q}_\tau \ \text{ if } t' \geqslant \underline{t}+1 \\ \text{and } A(\hat{\mathbf{y}}_{\bar{t}} - \mathbf{y}_{t'}) \leqslant \sum_{\tau=t'+1}^{\bar{t}} \mathbf{q}_\tau \ \text{ if } t' \leqslant \bar{t}-1 \end{array} \right\}. \tag{49}$$

In general, $Y^+_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}}) \subseteq Y_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$, with strict inclusion possible, except that the two constraint sets become identical when $t' = \underline{t} + 1 = \bar{t} - 1$.

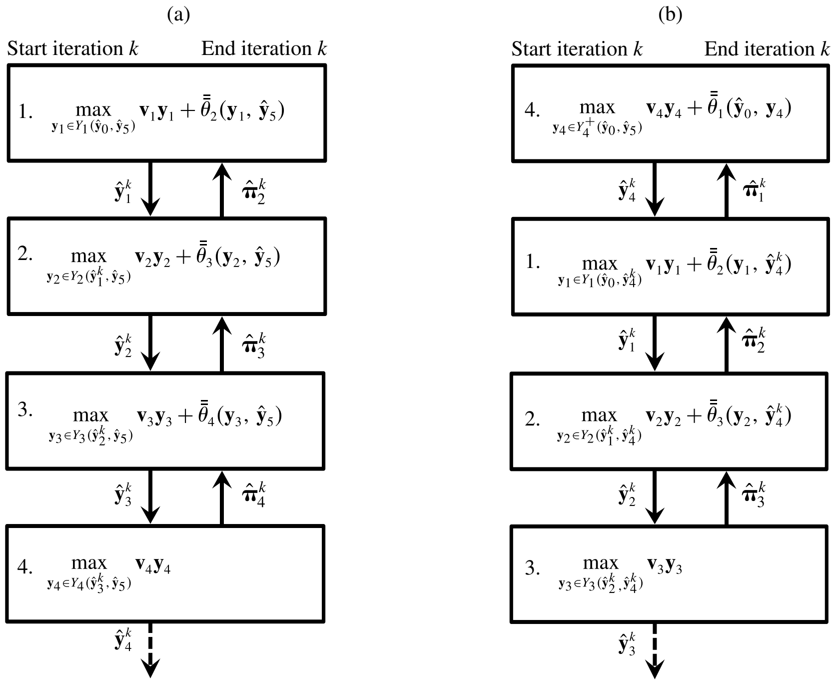
The aggregate constraints in (49) (i.e., the constraints involving the matrix A) link period t' with periods $\underline{t}$ and $\bar{t}$, thereby enabling the use of (39) to derive more general problem recursions. To illustrate, the following subsections describe three different HBD recursions. Variations on these recursions could be made theoretically valid for any staircase model with cumulative variables, but they seem unlikely to be computationally attractive without the use of aggregate constraints. We also note that the ideas described below have led us to experiment with more direct aggregation/disaggregation heuristics for solving **MIP** and **IP**

Figure 4. A generic FASTPASS iteration of a standard NBD algorithm that solves the forward recursion of a staircase LP with $T = 4$ periods; see constraints (42)–(45).



Notes. The number on the left indicates time period t, the left box represents the primal subproblem for t, and the right box represents the dual subproblem for t. Downward-pointing arrows correspond to the primal outputs of the subproblems, and upward pointing arrows to dual outputs. The dashed arrow here and in other figures indicates that no other subproblem uses the indicated output. (That output is saved, however, in case it constitutes part of an optimal primal solution.) The dual vector $\hat{\boldsymbol{\pi}}_t^k$ represents $(\hat{\boldsymbol{\alpha}}_t^k, \hat{\boldsymbol{\beta}}_t^k, \hat{\boldsymbol{\delta}}_t^k, \hat{\boldsymbol{\gamma}}_t^k)$ as described in Appendix A.

Figure 5. Simplified diagrams for a FASTPASS iteration of NBD for a staircase LP with $T = 4$.



Notes. (a) Represents the standard forward recursion (FBD), which Figure 4 depicts in more detail. (b) Represents an iteration of FBD enhanced with aggregate constraints (FBD-A); see Equations (50)–(52).

approximately; for example, see Van Den Heever and Grossmann (2000). Thus far we have been unsuccessful, however.

3.3. Forward Nested Decomposition with Aggregate Resource Constraints (FBD-A)

The use of aggregate resource constraints, together with a slight reordering of the recursion, can improve FBD substantially. Intuitively, by initiating the recursion with a model that aggregates constraints over the complete time horizon, the solution process obtains some initial guidance from a "weakly constrained UPL solution," which the standard forward recursion cannot supply. Specifically, this recursion, denoted "FBD-A," can begin based on period T to take advantage of the aggregate constraints defined through $Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})$:

$$\begin{aligned}
&\theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) \\
&\quad = \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \theta_1(\hat{\mathbf{y}}_0, \mathbf{y}_T) && (50)\\
&\quad = \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \left\{ \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \mathbf{y}_T)} \mathbf{v}_1\mathbf{y}_1 + \theta_2(\mathbf{y}_1, \mathbf{y}_T) \right\} && (51)\\
&\quad \vdots \\
&\quad = \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \left\{ \max_{\mathbf{y}_1 \in Y_1(\hat{\mathbf{y}}_0, \mathbf{y}_T)} \mathbf{v}_1\mathbf{y}_1 \right.\\
&\qquad\qquad \left. + \left\{ \cdots \left\{ \max_{\mathbf{y}_{T-1} \in Y_{T-1}(\mathbf{y}_{T-2}, \mathbf{y}_T)} \mathbf{v}_{T-1}\mathbf{y}_{T-1} \right\} \cdots \right\} \right\}. && (52)
\end{aligned}$$

Figure 5(b) depicts a single FASTPASS iteration of this recursion for $T = 4$. Note that, after the first step to identify a value for $\mathbf{y}_T$, the recursion simply follows the pattern set out in Equations (46) and (47), but with T replacing $\bar{T}$.

To gain some insight into the value of the aggregate subproblem, consider the first subproblem of the first primal pass of FBD-A. (See Equation (50) and Figure 5(b).) That subproblem is

$$\max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T + \bar{\bar{\theta}}_1(\hat{\mathbf{y}}_0, \mathbf{y}_T) = \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T\mathbf{y}_T \tag{53}$$

because $\bar{\bar{\theta}}_1(\hat{\mathbf{y}}_0, \mathbf{y}_T)$, the approximation to $\theta_1(\hat{\mathbf{y}}_0, \mathbf{y}_T)$, is null. Given the aggregate constraints, this recursion begins by identifying a weakly constrained UPL solution and then identifies a period-by-period extraction schedule for this "estimated ultimate pit." (Each iteration of FBD-A then updates its estimates of both the aggregate and disaggregate quantities.) Compare that to FBD, which begins by solving an LP that corresponds to greedily excavating the pit from period 1 to period T: profitable but poor, initial, global myopic decisions in early time periods may lead to a poor initial global solution, from which the algorithm recovers only at the expense of extra iterations.

3.4. Reverse Nested Decomposition with Aggregate Resource Constraints (RBD-A)

For a standard staircase model without cumulative variables, say, a production-inventory problem (e.g., Gabbay 1979), a reverse nested decomposition does not seem sensible, especially in early iterations. In particular, in the order $t = T, T-1, \ldots, 2$, a reverse decomposition would try to estimate optimal production quantities in time period t, but without having a reasonable approximation of the optimal, total production up to time period $t-1$ (i.e., without having a reasonable approximation of the optimal inventory at the beginning of period t). Cumulative variables $\mathbf{y}_t$ in **RMIP** do represent total production (extraction) for each block up through time period t, but a mirror image in time of the simple forward recursion (see Equations (42)–(45)) would provide little guidance as to resource consumption in early iterations.

By applying aggregate resource constraints, however, we can also obtain global guidance in a reverse recursion. Specifically, using (49), the following recursion describes a "reverse decomposition with aggregate constraints" (RBD-A):

$$\theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})$$

$$= \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T \mathbf{y}_T + \theta_{T-1}(\hat{\mathbf{y}}_0, \mathbf{y}_T) \tag{54}$$

$$= \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T \mathbf{y}_T + \left\{ \max_{\mathbf{y}_{T-1} \in Y_{T-1}^+(\hat{\mathbf{y}}_0, \mathbf{y}_T)} \left\{ \mathbf{v}_{T-1}\mathbf{y}_{T-1} + \theta_{T-2}(\hat{\mathbf{y}}_0, \mathbf{y}_{T-1}) \right\} \right\} \tag{55}$$

$$\vdots$$

$$= \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T \mathbf{y}_T + \left\{ \max_{\mathbf{y}_{T-1} \in Y_{T-1}^+(\hat{\mathbf{y}}_0, \mathbf{y}_T)} \mathbf{v}_{T-1}\mathbf{y}_{T-1} + \left\{ \cdots \left\{ \max_{\mathbf{y}_1 \in Y_1^+(\hat{\mathbf{y}}_0, \mathbf{y}_2)} \mathbf{v}_1 \mathbf{y}_1 + \theta_0 \cancelto{0}{(\hat{\mathbf{y}}_0, \mathbf{y}_1)} \right\} \cdots \right\} \right\} \tag{56}$$

More simply, the RBD-A recursion may be summarized through the repeated application of the following relationships, starting with $\bar{t} = \bar{T}$ and with fixed values $\underline{t} = 0$, $\mathbf{y}_0 = \hat{\mathbf{y}}_0$, and $\mathbf{y}_{\bar{T}} = \hat{\mathbf{y}}_{\bar{T}}$:

$$\theta^*(\mathbf{y}_{\underline{t}}, \mathbf{y}_{\bar{t}}) = \theta_{\bar{t}-1}(\hat{\mathbf{y}}_0, \mathbf{y}_{\bar{t}}) \tag{57}$$

$$= \max_{\mathbf{y}_{\bar{t}-1} \in Y_{\bar{t}-1}^+(\hat{\mathbf{y}}_0, \mathbf{y}_{\bar{t}})} \mathbf{v}_{\bar{t}-1}\mathbf{y}_{\bar{t}-1} + \theta^*(\hat{\mathbf{y}}_0, \mathbf{y}_{\bar{t}-1}). \tag{58}$$

Figure 6(a) illustrates a FASTPASS iteration of RBD-A for $T=4$.

3.5. A Tree Decomposition Using Aggregate Resource Constraints (BBD-A)

We would not use a reverse decomposition without aggregate resource constraints, and we would not create a generalizing tree decomposition without them, either. For a particular application, a user might tailor a recursion to characteristics of the modeled system but, for simplicity, we describe "BBD-A," which defines a "bisection decomposition with aggregate resource constraints." Also for simplicity, we specialize to $T = 2^n$ for some integer $n > 1$:

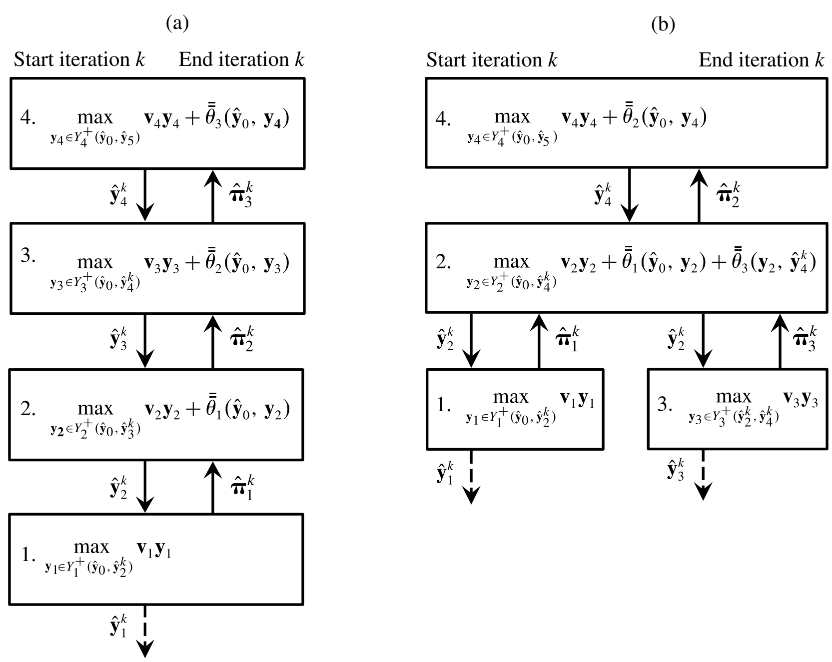
$$\theta^*(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) = \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T \mathbf{y}_T + \theta_{T/2}(\hat{\mathbf{y}}_0, \mathbf{y}_T) \tag{59}$$

$$= \max_{\mathbf{y}_T \in Y_T^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_T \mathbf{y}_T + \left\{ \max_{\mathbf{y}_{T/2} \in Y_{T/2}^+(\hat{\mathbf{y}}_0, \mathbf{y}_T)} \mathbf{v}_{T/2}\mathbf{y}_{T/2} + \theta_{T/4}(\hat{\mathbf{y}}_0, \mathbf{y}_{T/2}) + \theta_{3T/4}(\mathbf{y}_{T/2}, \mathbf{y}_T) \right\}, \tag{60}$$

where $\theta_{T/4}(\hat{\mathbf{y}}_0, \mathbf{y}_{T/2}) = \max_{\mathbf{y}_{T/4} \in Y_{T/4}^+(\hat{\mathbf{y}}_0, \mathbf{y}_{T/2})} \mathbf{v}_{T/4}\mathbf{y}_{T/4} + \theta_{T/8}(\hat{\mathbf{y}}_0, \mathbf{y}_{T/4})$

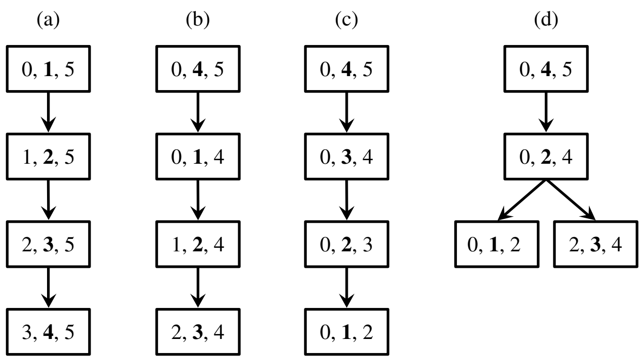
$$+ \theta_{3T/8}(\mathbf{y}_{T/4}, \mathbf{y}_{T/2}), \tag{61}$$

Figure 6. A single iteration for two versions of nested Benders decomposition with cumulative variables and aggregate constraints: (a) a reverse decomposition (RBD-A) and (b) a bisection decomposition (BBD-A).



Notes. A FASTPASS processing method, or a generalization thereof, applies to both recursions. Note that $Y_1^+(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_2^k) = Y_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_2^k)$ in both (a) and (b) and that $Y_3^+(\hat{\mathbf{y}}_2^k, \hat{\mathbf{y}}_4^k) = Y_3(\hat{\mathbf{y}}_2^k, \hat{\mathbf{y}}_4^k)$ in (b).

Figure 7. Tree representation of nested and hierarchical Benders decompositions: (a) FBD (b) FBD-A (c) RBD-A (d) BBD-A.



Note. The numbers in each vertex are $[\underline{t}, t, \bar{t}]$, with t in a bold font to emphasize that the value for $\mathbf{y}_t$ is being determined at that vertex.

and $$\theta_{3T/4}(\mathbf{y}_{T/2}, \mathbf{y}_T) = \max_{\mathbf{y}_{3T/4} \in Y^+_{3T/4}(\mathbf{y}_{T/2}, \mathbf{y}_T)} \mathbf{v}_{3T/4}\mathbf{y}_{3T/4} + \theta_{5T/8}(\mathbf{y}_{T/2}, \mathbf{y}_{3T/4}) + \theta_{7T/8}(\mathbf{y}_{3T/4}, \mathbf{y}_T), \tag{62}$$

etc.

After the first step to identify a value for $\mathbf{y}_T$, we may summarize the recursion for BBD-A through the repeated application of the following relationships, starting with $\underline{t} = 0$ and $\bar{t} = T$, and with fixed values $\mathbf{y}_0 = \hat{\mathbf{y}}_0$ and $\mathbf{y}_{\bar{t}} = \hat{\mathbf{y}}_T$:

$$\theta^*(\mathbf{y}_{\underline{t}}, \mathbf{y}_{\bar{t}}) = \theta_t(\mathbf{y}_{\underline{t}}, \mathbf{y}_{\bar{t}}), \quad \text{where } t = (\underline{t} + \bar{t})/2 \tag{63}$$

$$= \max_{\mathbf{y}_t \in Y^+_t(\mathbf{y}_{\underline{t}}, \mathbf{y}_{\bar{t}})} \mathbf{v}_t\mathbf{y}_t + \theta^*(\mathbf{y}_{\underline{t}}, \mathbf{y}_t) + \theta^*(\mathbf{y}_t, \mathbf{y}_{\bar{t}}). \tag{64}$$

If T is not a multiple of 2, t may be defined as the integer floor or ceiling of $(\underline{t} + \bar{t})/2$; in fact, computational tests in Sections 4 and 6 apply the integer-floor operator.

Figure 6(b) illustrates an iteration of BBD-A for $T = 4$. We generalize the FASTPASS subproblem processing method by (i) viewing the decomposition diagram as a directed tree with all arcs pointing downward, away from the root vertex; (ii) processing primal subproblems in any topological (acyclic) ordering of that tree; and (iii) processing dual problems in a reverse topological ordering of the tree, but skipping leaf vertices. Figure 7(d) shows the diagram of Figure 6(b) viewed as a tree, while Figures 7(a)–(c) show how other, simpler decomposition schemes may be viewed as trees, also.

3.6. Hierarchical Benders Decomposition (HBD)

For simplicity, we refer to a tree decomposition that uses cumulative variables and aggregate resource constraints as a hierarchical Benders decomposition" (HBD). Both RBD-A and BBD-A are examples, even though the tree associated with RBD-A has an especially restricted structure. FBD-A uses aggregate constraints only in its first stage, but we also include that as a special case of HBD. On the other hand, it should be clear that HBD allows even more general recursions than those described. For instance, seasonal effects might make this decomposition scheme possible over two years at a monthly level of detail: two years, one year, one quarter, one month.

One difficulty with standard Benders decomposition is that early iterations make only slow progress to an optimal solution because the few cuts available provide a poor approximation of the subproblem's or subproblems' contribution to the overall objective function (Geoffrion and Graves 1974). When NBD is used to solve multistage stochastic programming problems, several authors have shown how the application of special "preliminary cuts" can help address this issue (Infanger 1994, pp. 96–99; Morton 1996). In particular, Infanger generates preliminary cuts based on an aggregate, "expected-value model." By contrast, we exploit an aggregate model directly in the Benders recursion rather than through specialized cuts. To the best of our knowledge, complete aggregate models have not been exploited in multistage stochastic programming.

As a final point in this discussion, we note that, in theory, HBD could be applied to certain staircase LPs that lack the stationary matrix A that appears in resource constraints (26). For example, suppose that **RMIP** represents a production-inventory-distribution model with time-of-year effects in production-line yields (e.g., Brown et al. 2001) and that these are represented by replacing constraints (26) with the following:

$$A_t(\mathbf{y}_t - \mathbf{y}_{t-1}) \leqslant \mathbf{q}_t \quad \forall\, t \in \mathcal{T}. \tag{65}$$

The cancellations that enable creation of aggregate constraints in (49) no longer apply. But if we define $A = \min_{t \in \{\underline{t}, \ldots, \bar{t}\}} A_t$, where the minimization is taken elementwise across the matrices A_t, then $Y^+_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$ defined using this matrix A is valid, although it may not be as tight as when $A_t = A$ for all t. Intuitively, the weaker constraints would still give useful guidance to the decomposition provided that the matrices A_t vary only modestly with t, i.e., are "nearly stationary."

4. Computational Tests: Solving RMIP

This section describes computational tests of HBD for solving instances of **RMIP**. We use all six data sets described by Cullenbine et al. (2011), which create problem instances that cover 10,819, 18,300, and 25,620 blocks, for 1 to 20 time periods; the problem's name indicates the number of blocks modeled in the data. (Five of these data sets were used for testing in Section 2.3.2.) Potential solution methods include a simplex algorithm applied to the "monolithic LP" and each of the four variants of nested Benders decomposition: FBD, FBD-A, RBD-A and BBD-A.

A 64-bit workstation with 16 GB RAM and a 3 GHz quad-core Intel processor performs all computations, running under a Windows operating system. A C++ program generates all LPs, and CPLEX 12.5 (IBM Corp.

2013) solves those LPs. Default parameters apply except as follows:

(i) the solver may use at most four threads (Threads = 4);

(ii) CPLEX's barrier algorithm solves the monolithic LPs (LPMethod = Barrier);

(iii) monolithic LPs are limited to 7,200 seconds of elapsed computation time (TiLim = 7,200);

(iv) the dual simplex algorithm solves the decomposition's LPs (LPMethod = Dual); and

(v) that algorithm emphasizes numerical stability (NumericalEmphasis = true).

Note that the barrier algorithm typically solves an LP "from scratch" more quickly than does the dual simplex algorithm, hence (ii). But within a decomposition algorithm with cuts being added to an LP from one iteration to the next, the dual simplex algorithm becomes quicker because it can exploit standard "warm starts," hence (iv).

A penalty of 100 dollars/ton, discounted at the model's standard rate, applies to the violation of each resource constraint in period t; this penalty corresponds to p^-_{rt} and p^+_{rt} defined in Appendix A. Also, all decomposition algorithms enforce a relative optimality tolerance of $\epsilon_{\mathbf{RMIP}} = 10^{-4}$.

Problem-specific preprocessing, adjusted for potentially fractional blocks, helps reduce model sizes. This preprocessing eliminates variables and constraints by bounding the earliest and latest time periods in which a block can be extracted. (See Lambert et al. 2014 for details, but note that we do not use the "enhanced early starts" described in that paper.) Preprocessing requires only a few seconds of computation and is performed just once for each data set, so we do not report the corresponding computation times. Reported times do include CPLEX's standard "prereduce" methods, however.

Initial computation focuses on one of the two largest data sets, BD25620A. For reference, Table 2 displays some model-size statistics for (i) the monolithic LP that is generated for $T = 5$ and for $T = 20$ and (ii) the average subproblem sizes observed while applying the decomposition algorithms for $T = 20$. Table 3 shows solution times over a range of values for T.

General trends seen for BD25620A in Table 3 hold for all data sets, so we use these results to cull the clearly inferior methods, which are solution as a monolithic LP and solution via FBD. In particular, the barrier algorithm applied to the monolith cannot compete with FBD, but FBD cannot compete with the hierarchical decomposition methods. (When $T = 3$, BBD-A does perform poorly, and FBD is faster, but this is the only such case for BD25620A.)

Table 2. BD25620A: Model sizes for instances of **RMIP** generated as monolithic LPs and sizes for nested subproblems encountered during solution by decomposition.

	Initial		Reduced	
LP or nested subproblem	Variables (num.)	Constraints (num.)	Variables (num.)	Constraints (num.)
RMIP, $T = 20$	643,424	3,008,805	486,008	2,982,412
RMIP, $T = 5$	128,101	644,240	102,072	591,389
Avg. FBD subproblem w/o cuts	32,040	129,862	27,904	125,726
Avg. FBD-A subproblem w/o cuts	25,623	109,634	7,130	2,241
Avg. RBD-A subproblem w/o cuts	25,622	108,353	107	1,022
Avg. BBD-A subproblem w/o cuts	25,623	109,634	1,455	7,009

Notes. "**RMIP**, $T = 20$" corresponds to the largest monolithic LP generated, and "**RMIP**, $T = 5$" corresponds to the largest such model that solves in under 7,200 seconds. Averages ("Avg.") are taken across all periods in a 20-period model. "Initial" statistics are generated using problem-specific preprocessing, while "Reduced" statistics are those obtained after applying CPLEX's "prereduce" methods.

Table 3. BD25620A: Solution statistics for instances of **RMIP** solved by a barrier algorithm and by decomposition.

		Decomposition											
	Mono-lith	FBD			FBD-A			RBD-A			BBD-A		
T	(sec.)	(sub.)	(piv.)	(sec.)	(sub.)	(piv.)	(sec.)	(sub.)	(piv.)	(sec.)	(sub.)	(piv.)	(sec.)
1	5.7	—	—	—	—	—	—	—	—	—	—	—	—
2	48.6	4	6,091	9.0	4	6,773	8.4	4	6,773	8.4	4	6,773	8.4
3	333.1	11	12,965	15.7	7	20,302	26.2	7	7,099	8.8	7	20,302	26.3
4	1,240.8	16	19,854	22.8	10	7,098	11.3	10	7,114	11.3	9	7,050	11.2
5	6,320.4	29	29,830	33.5	21	8,716	15.5	13	8,217	14.4	19	10,736	16.8
10	†	82	76,141	89.6	46	17,510	34.8	64	10,171	23.6	40	11,334	22.1
15	†	211	178,743	235.3	99	18,228	44.1	127	17,744	43.9	103	29,631	73.8
20	†	248	184,473	235.6	172	16,957	57.0	324	21,825	76.1	175	16,000	60.7

Notes. This table lists solution seconds "(sec.)" for all methods and, for the decomposition algorithms, it lists the number of subproblems solved "(sub.)" and simplex pivots "(piv.)." The symbol "†" indicates that the problem could not be solved within 7,200 seconds. Decomposition algorithms skip the solution of a subproblem if that subproblem immediately follows the generation of a nonviolated cut.

The smaller subproblem sizes shown in Table 2 for HBD compared to FBD result from HBD's aggregate constraints enabling the fixing of more variables to zero or one, and these statistics do help explain the computational superiority of HBD methods over FBD. Other problem instances do not show such stark differences, but in no case does FBD outperform any HBD method.

We have experimented with several methods to improve the computational performance of FBD, but with no success. First, neither the primal simplex algorithm nor the barrier algorithm in CPLEX is faster than the dual simplex algorithm for solving the relevant LPs. Second, our attempts at "stabilizing" primal and/or dual solutions by solving primal and/or dual subproblems using CPLEX's barrier algorithm all fail. In particular, the use of interior-point solutions can improve the empirical convergence of decomposition algorithms (e.g., Rousseau et al. 2007, Singh et al. 2009), but the extra time required to find such solutions produces at best minor reductions in the number of major iterations for FBD while greatly increasing solution times. With some significant implementational effort, other specialized techniques might yield computational benefits (Ruszczyński 1986, Gondzio et al. 1997, Elhedhli and Goffin 2004). However, as seen in Table 3, the use of aggregate constraints to create FBD-A produces definite computational benefits; furthermore, the implementational effort for this "specialized technique" is modest.

Table 4 shows the results for all six data sets using the solution methods that pass the "culling test" described above. Solution times remain reasonable for all hierarchical decompositions, even as the number of time periods becomes large. This suggests that the technique common to all, namely, the use of aggregate constraints, is key to computational efficiency. We postpone further discussion of these computational results until Section 7.

Table 4. Solution times, in seconds, for 10k-, 18k-, and 25k-block instances of **RMIP** solved using HBD.

	BD10819A			BD10819F		
T	FBD-A	RBD-A	BBD-A	FBD-A	RBD-A	BBD-A
2	0.8	0.8	0.8	0.8	0.8	0.8
3	1.4	1.5	1.4	1.6	1.6	1.6
4	2.3	2.5	2.3	2.3	2.5	2.5
5	3.4	3.8	3.9	3.3	3.7	3.5
10	16.7	17.6	16.8	17.4	19.1	15.9
15	45.4	45.4	37.2	44.9	46.0	35.7
20	88.2	97.5	66.0	104.9	94.3	55.0
	BD18300A			BD18300B		
T	FBD-A	RBD-A	BBD-A	FBD-A	RBD-A	BBD-A
2	19.1	19.1	19.1	1.2	1.1	1.1
3	2.5	2.4	2.5	1.8	1.7	1.7
4	3.3	3.2	3.5	5.3	5.3	5.2
5	4.1	18.6	4.6	10.0	8.3	3.1
10	11.3	16.0	16.0	10.5	9.9	14.5
15	114.7	34.4	26.7	18.8	26.3	27.1
20	36.0	56.6	41.3	47.5	69.0	32.6
	BD25620A			BD25620B		
T	FBD-A	RBD-A	BBD-A	FBD-A	RBD-A	BBD-A
2	8.4	8.4	8.4	7.0	7.0	7.0
3	26.2	8.8	26.3	9.5	9.9	9.5
4	11.3	11.3	11.2	13.3	13.2	13.2
5	15.5	14.4	16.8	16.0	16.0	15.6
10	34.8	23.6	22.1	66.5	59.3	63.0
15	44.1	43.9	73.8	160.4	99.8	70.4
20	57.0	76.1	60.7	251.2	574.7	198.3

5. Combining HBD and Heuristic Methods to Solve MIP

Computational tests above show that HBD can solve medium-sized instances of **RMIP** that have both upper- and lower-bounding resource constraints, but OPBS requires MIP-valid solutions for **MIP**. This section describes a heuristic that combines HBD and branch and bound to produce high-quality solutions for **MIP** to all of the test problems solved as LPs above.

As in Section 3.5, a graph $G = (V, E)$ describes the hierarchy tree, with vertices $i \in V$ labeled in the order of a primal pass for HBD. We adjust the notation so that $\mathbf{RMIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$, which has solution $\hat{\mathbf{y}}_{t(i)}$, denotes the LP subproblem at vertex i in the tree. The notation $\mathbf{MIP}_{t(i)}(\hat{\mathbf{y}}_{\underline{t}(i)}, \hat{\mathbf{y}}_{\bar{t}(i)})$ indicates the corresponding MIP (without explicit binary variables), which has MIP-valid solution $\hat{\mathbf{y}}^*_{t(i)}$. The following describes the complete heuristic for solving **MIP**.

Procedure HMIPH ("Hierarchical MIP Heuristic")
Input: Full problem data for **MIP**, relative optimality tolerance $\epsilon_{\mathbf{SUB}} > 0$ for subproblem MIPs.
Output: A MIP valid solution to **MIP**, $(\hat{\mathbf{y}}^*_1, \ldots, \hat{\mathbf{y}}^*_T)$.
Notation: $\hat{\mathbf{y}}_t$ ($\hat{\mathbf{y}}^*_t$) denotes an LP (MIP-valid) solution of a model for period t.

{
Step 0: Using a specific version of HBD, solve **RMIP** for $(\hat{\mathbf{y}}_1,\ldots,\hat{\mathbf{y}}_T)$;
For ($i = 1$ to T) { /* In the order of a primal pass of the decomposition tree */
Step 1: If ($i \neq 1$) {
Beginning with cuts generated in step 0, and generating new cuts as necessary, use the same version of HBD to solve $\mathbf{RMIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$ for $\hat{\mathbf{y}}_{t(i)}$;
/* $\hat{\mathbf{y}}_{t(i)}$ serves as a starting point to solve $\mathbf{MIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$. The MIP-valid inputs $\hat{\mathbf{y}}^*_{\underline{t}(i)}$ and $\hat{\mathbf{y}}^*_{\bar{t}(i)}$ are available because of the vertex-processing order. */
}

/* The next step solves a "MIP subproblem." */
Step 2: With relative optimality tolerance $\epsilon_{\mathbf{SUB}}$, solve $\mathbf{MIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$ for $\hat{\mathbf{y}}^*_{t(i)}$ using the branch-and-bound procedure outlined in the text below;
}
Print("The MIP-valid, approximate solution to **MIP** is," $(\hat{\mathbf{y}}^*_1, \ldots, \hat{\mathbf{y}}^*_T)$);
}

The branch-and-bound procedure used in step 2 resembles the "SOS2 constraint branching" of Beale and Tomlin (1970). Specifically, if at vertex i the solution $\hat{\mathbf{y}}_{t(i)}$ to $\mathbf{RMIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$ does not satisfy integrality requirements for an implicitly determined $\hat{\mathbf{x}}_{t(i)}$, blocks b and $\bar{b}$ must exist such that $\hat{y}_{bt(i)} > 0$, but $\hat{y}_{\bar{b}t(i)} < 1$. That is, some block b is partially extracted in the current solution, yet block $\bar{b}$ directly above b is not completely extracted. When this happens, we branch by enforcing "$y_{bt} = 0$" or "$y_{\bar{b}t} = 1$." (This does not enforce a partition of **MIP**'s feasible region in terms of $\mathbf{y}$, but it does correspond to a partition of the full feasible region defined in terms of $(\mathbf{x}, \mathbf{y})$.) Nodes in the branch-and-bound tree are always feasible and thus are fathomed by bound. The following three rules control branching:

1. Choose the next node for branching based on a "best-bound criterion" (e.g., Linderoth and Savelsbergh 1999).
2. Given the branching node i, among blocks b such that $\hat{y}_{bt(i)} > 0$ and $\hat{y}_{\bar{b}t(i)} < 1$, branch on the block $\underset{\sim}{b}$ having the largest number of fractional, direct predecessors and successors. (Block b' is a *direct successor* of b if b is a direct predecessor of b'.)
3. Branch first in a direction analogous to "branch up," by enforcing this restriction: $y_{\bar{\underset{\sim}{b}}t} = 1$. (The complementary branching direction is thus $y_{\underset{\sim}{b}t} = 0$.)

HMIPH is not an exact algorithm because branch and bound applies only to individual MIP subproblems in the hierarchy. But the always-feasible, MIP-valid, final solution $(\hat{\mathbf{y}}^*_1, \ldots, \hat{\mathbf{y}}^*_T)$ defines a lower bound on $\theta^*_{\mathbf{MIP}}$, and the optimal objective value for $\mathbf{MIP}_{t(1)}(\hat{\mathbf{y}}^*_{\underline{t}(1)}, \hat{\mathbf{y}}^*_{\bar{t}(1)})$ defines an upper bound, a bound that may be better than $\theta^*_{\mathbf{RMIP}}$.

6. Computational Results: Solving MIP

Based on the instances of **RMIP** already investigated (see Table 4), this section evaluates the computational performance of HMIPH for solving **MIP**, i.e., the full MIP model for OPBS. Settings for LP subproblems in steps 0 and 1 of the procedure remain as in Section 4; MIP subproblems in step 2 incorporate a relative optimality tolerance of $\epsilon_{\mathbf{SUB}} = 5 \times 10^{-4}$. (The values for $\epsilon_{\mathbf{SUB}}$ and $\epsilon_{\mathbf{RMIP}}$ are selected together based on empirical tests, which show the pair to yield a good trade-off between solution speed and "accuracy," that is, observed optimality gap.)

Table 5 shows the results for all six instances of **MIP** using HMIPH with the three versions of HBD. Solution times remain reasonable. For example, based on BBD-A, each 20-period instance of **MIP** solves in less than twice the time required to solve the corresponding instance of **RMIP**. Moreover, the computed relative optimality gaps indicate that the heuristic consistently yields high-quality solutions. We also note that any resource-constraint violations are negligible.

The bisection decomposition BBD-A leads to good solutions to **MIP** more quickly than do the other decompositions. The next section indicates why this may be true.

7. Discussion of Computational Results for Both RMIP and MIP

The results presented in Table 3 indicate the superiority of all HBD variants over a standard implementation of nested Benders decomposition (FBD) for solving **RMIP**: the HBD variants are usually faster than FBD, and when $T \geqslant 10$, they are always two to five times faster. No clear trend appears in Table 4, however, to indicate a clear computational winner among the HBD variants. But it should be easy to parallelize BBD-A and, with $T/2$ processors and sufficient computer memory, total solution time might reduce by a factor approaching $(\log T)/T$. Thus, we believe that, of the HBD variants, BBD-A holds the greatest promise for solving staircase LPs.

Although no HBD method is clearly faster than any other for solving **RMIP**, Table 5 shows that BBD-A does gain a computational advantage when used to help solve the largest instances of **MIP**. Specifically, applications of HMIPH based on BBD-A produce solution times on the 20-period instances that range from 47% to 94% of the nearest rival, with an average of 77%. Average subproblem sizes for **RMIP** based on BBD-A are not smaller than for the other HBD variants (see Table 2), so we attribute most of this advantage to the more "balanced branching" that must take place in HMIPH when basing computations on BBD-A.

More specifically, well-balanced branch-and-bound enumeration avoids branching such that one side of the branching partition is strongly restricted while the other side is not. Balanced branching explains much of the efficiency improvements seen with (i) branching based on special ordered sets (Beale and Tomlin 1970, Hummeltenberg 1984), (ii) the implicit-constraint branching exploited by Ryan and Foster (1981) to help solve set-partitioning problems, and (iii) the explicit-constraint branching exploited by Appleget and Wood (2000) to help solve certain binary IPs.

To give a simple, intuitive example, suppose that based on FBD-A or BBD-A, HMIPH is applied to an all-binary, T-period instance of OPBS with variables x_{bt}. For either variant, at vertex $i = 1$, HMIPH determines $\hat{\mathbf{x}}_T$, which specifies for each block b whether or not the block is extracted over the full time horizon. At vertex $i = 2$, HMIPH based on FBD-A would select a fractional variable x_{b1} and branch as follows: (i) block b is extracted in period $t = 1$, or (ii) if it is extracted, block b is extracted in some

Table 5. Solution statistics for 10k-, 18k- and 25k-block instances of **MIP** solved using HMIPH.

	BD10819A						BD10819F					
	Gap (%)			Soln. time (sec.)			Gap (%)			Soln. time (sec.)		
T	FA	RA	BA	FA	RA	BA	FA	RA	BA	FA	RA	BA
2	0.8	0.8	0.8	2.2	2.2	2.3	0.8	0.8	0.8	2.3	2.3	2.4
3	0.7	0.7	0.7	3.1	3.1	3.1	0.8	0.7	0.8	3.2	3.2	3.2
4	0.6	0.6	0.6	4.8	4.6	4.3	0.8	0.7	0.7	4.7	4.8	4.8
5	0.7	0.6	0.6	5.9	6.0	6.1	0.7	0.6	0.6	5.6	6.1	6.1
10	0.5	0.5	0.5	23.9	22.6	24.6	0.5	0.5	0.5	25.2	24.6	22.3
15	0.6	0.6	0.6	60.1	58.8	52.9	0.5	0.5	0.5	59.7	58.0	48.2
20	0.5	0.6	0.6	121.1	120.3	96.2	0.5	0.6	0.6	135.7	117.7	89.5
	BD18300A						BD18300B					
	Gap (%)			Soln. time (sec.)			Gap (%)			Soln. time (sec.)		
T	FA	RA	BA	FA	RA	BA	FA	RA	BA	FA	RA	BA
2	0.2	0.2	0.2	25.1	24.7	24.8	0.0	0.0	0.0	1.2	1.2	1.2
3	0.4	0.4	0.4	9.7	12.0	9.6	0.0	0.0	0.0	2.0	2.0	2.0
4	0.4	0.4	0.4	9.0	8.8	8.4	0.0	0.0	0.0	5.7	5.9	5.6
5	0.3	0.3	0.3	10.5	25.6	9.5	0.0	0.0	0.0	11.4	9.4	4.4
10	0.2	0.2	0.2	21.1	29.3	29.4	0.0	0.0	0.0	14.8	14.6	18.5
15	0.2	0.2	0.2	138.0	82.2	58.4	0.0	0.0	0.0	38.3	39.3	35.3
20	0.2	0.2	0.2	83.0	128.6	78.2	0.0	0.0	0.0	76.0	110.9	54.8
	BD25620A						BD25620B					
	Gap (%)			Soln. time (sec.)			Gap (%)			Soln. time (sec.)		
T	FA	RA	BA	FA	RA	BA	FA	RA	BA	FA	RA	BA
2	0.0	0.0	0.0	8.9	8.5	8.5	0.0	0.0	0.0	10.8	10.6	10.6
3	0.0	0.0	0.0	26.7	9.3	26.9	0.1	0.1	0.1	24.7	21.7	24.4
4	0.0	0.0	0.0	12.3	11.9	12.0	0.2	0.2	0.2	27.9	25.1	25.6
5	0.0	0.0	0.0	17.4	15.3	18.1	0.4	0.4	0.5	53.4	48.6	50.1
10	0.0	0.0	0.0	44.4	29.0	31.9	1.2	1.2	1.2	197.2	130.8	289.6
15	0.0	0.0	0.0	61.2	68.3	90.0	1.3	1.4	1.2	853.4	266.4	158.3
20	0.0	0.0	0.0	96.3	154.0	86.2	1.2	1.2	1.1	852.6	1,441.4	403.6

Note. “FA” = FBD-A, “RA” = RBD-A, “BA” = BBD-A, and “Gap” = relative optimality gap.

period $t > 1$. The first element in this partition is strongly restricted while the second is only weakly restricted, so this branching scheme is unbalanced. By contrast, at vertex $i = 2$, HMIPH based on BBD-A would identify a fractional variable $x_{b,\lfloor T/2\rfloor}$, and branching would execute this better-balanced partition: (i) block b is extracted in the first half of the time horizon, or (ii) if it is extracted, block b is extracted in the second half of the time horizon.

For computational evidence of balanced branching, consider the most computationally challenging problem instance, BD25620B. BBD-A and FBD-A have about the same number of violated integrality restrictions when branching commences at each decomposition-tree vertex, about 950 when summed over all i. But BBD-A creates a total of only 98 branch-and-bound nodes across all MIP subproblems while FBD-A creates 369. (RBD-A is worse than FBD-A on both measures.)

As a final point on computation, note that HMIPH can be applied to solve **IP** rather than **MIP**. Steps 0 and 1 of HMIPH remain the same because the solution to **RMIP** is also the solution to the LP relaxation of **IP**. But then, step 2 uses branch and bound to solve the integer-programming analog of $\mathbf{RMIP}_{t(i)}(\hat{\mathbf{y}}^*_{\underline{t}(i)}, \hat{\mathbf{y}}^*_{\bar{t}(i)})$. We have experimented with such a heuristic, but with little success. Specifically, while the heuristic should yield an elastically feasible solution, the lack of flexibility in the model formulation yields individual integer-programming subproblems in step 2 that, typically, do not solve in an hour of computation time.

8. Conclusions and Future Research

This paper has described a new mixed-integer-programming model for an open-pit block sequencing problem and has developed a special decomposition procedure for that model's solution. Unlike most other work on OPBS, our formulation **MIP** incorporates lower bounds on resource consumption in each period in addition to the standard upper bounds. Uniquely, this formulation also allows for fractional block extraction in a mine while still satisfying pit-wall slope restrictions.

The staircase constraint structure of **MIP** enables a new "hierarchical" Benders decomposition for solving **MIP**'s linear-programming relaxation "**RMIP**." HBD generalizes nested Benders decomposition, taking advantage of two special techniques: it formulates the model using cumulative variables and adds time-aggregated resource constraints that provide useful guidance to the overall solution procedure.

Computational testing in this paper does not exploit parallel solution of nested subproblems, but we believe that this will be a fruitful line of work to follow. A standard implementation of a forward recursion in NBD leaves little room for parallel computation because the kth primal pass of approximate subproblems $\mathbf{RMIP}_t^k$ solves those in the order $t=1,\ldots,T$, and a dual pass solves them in the reverse order. But HBD includes "tree decompositions" in which a subproblem $\mathbf{RMIP}_t$ decomposes into $\mathbf{RMIP}_{t_1}$ and $\mathbf{RMIP}_{t_2}$ such that approximating subproblems $\mathbf{RMIP}_{t_1}^k$ and $\mathbf{RMIP}_{t_2}^k$ could be solved in parallel, both in primal and dual passes. In fact, given $\lceil T/2\rceil$ processors, it may be possible to reduce solution time for **RMIP** by a factor approaching $(\log T)/T$.

We also note that NBD has been used extensively for solving multistage stochastic programs, so the usefulness of HBD needs to be explored for such applications.

Acknowledgments

The second author thanks the Office of Naval Research and the Air Force Office of Scientific Research for support of his research.

Appendix A. Solving RMIP with a Forward Nested Benders Decomposition, FBD

This appendix outlines how standard nested Benders decomposition solves a variant of **RMIP** in Section 2.2 using the forward recursion (FBD), as described by Equations (42)–(45) in Section 3.1.

The variant of **RMIP** allows penalized violation of resource constraints (26) using elastic variables

$$\mathbf{s}_t = ((s_{1t}^-,\ldots,s_{|\mathcal{R}|t}^-),(s_{1t}^+,\ldots,s_{|\mathcal{R}|t}^+))^\top, \tag{A1}$$

where s_{rt}^- and s_{rt}^+ correspond to violations of the lower-bounding and upper-bounding constraints (5) and (6), respectively. The corresponding vector of constraint-violation penalties, in units of dollars/ton, is

$$\mathbf{p}_t = ((p_{1t}^-,\ldots,p_{|\mathcal{R}|t}^-),(p_{1t}^+,\ldots,p_{|\mathcal{R}|t}^+)). \tag{A2}$$

Extending the model (46)–(47) with elastic resource constraints, the following recursively defines the full model and the cost-to-go function $\theta_t(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$. Given $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_{\bar{T}}$, for $t=1,\ldots,T$,

$$\mathbf{RMIP}_t(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}}):\ \theta_t(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$$
$$= \max_{\mathbf{0}\leqslant\mathbf{y}_t\leqslant\hat{\mathbf{y}}_{\bar{T}},\,\mathbf{s}_t\geqslant\mathbf{0}} \mathbf{v}_t\mathbf{y}_t - \mathbf{p}_t\mathbf{s}_t + \theta_{t+1}(\mathbf{y}_t,\hat{\mathbf{y}}_{\bar{T}}) \tag{A3}$$
$$\text{s.t.}\quad A\mathbf{y}_t - I\mathbf{s}_t \leqslant \mathbf{q}_t + A\hat{\mathbf{y}}_{t-1} \tag{A4}$$
$$-I\mathbf{y}_t \leqslant -I\hat{\mathbf{y}}_{t-1} \tag{A5}$$
$$H_t\mathbf{y}_t \leqslant \mathbf{0}, \tag{A6}$$

where $\theta_T(\mathbf{y}_{T-1},\mathbf{y}_{\bar{T}})\equiv 0$, and the constraints $-A\hat{\mathbf{y}}_T \leqslant \mathbf{q}_{\bar{T}} - A\hat{\mathbf{y}}_{\bar{T}}$ implied by (37) are omitted because they are constructed so as to be vacuous. (See the definition of $\mathbf{q}_t$ under "Additional Notation" in Section 3. Note that constraints analogous to these do appear in BBD-A, namely, constraints (B2), provided that $\bar{t}\neq\bar{T}$.)

Theorem 2. *FBD, as described in Section 3.1 for solving* **RMIP**, *converges when applied to the version of* **RMIP** (A3)–(A6) *that replaces standard resource constraints with elastic ones.*

Proof. (For reference, (25)–(31) directly define **RMIP** in the text, and (42)–(47) define that model recursively.) Viewing $\mathbf{y}_{t-1}=\hat{\mathbf{y}}_{t-1}$ as a parameter vector, $\mathbf{RMIP}_t(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$ above may be rewritten as

$$\mathbf{RMIP}_t(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}}):$$
$$\theta_t(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}}) = \max_{\mathbf{y}_t\in Y(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}})} \mathbf{v}_t\mathbf{y}_t - \mathbf{p}_t(A(\mathbf{y}_t-\mathbf{y}_{t-1})-\mathbf{q}_t)^+ + \theta_{t+1}(\mathbf{y}_t,\hat{\mathbf{y}}_{\bar{T}}), \tag{A7}$$

where $Y_t(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}})\equiv\{\mathbf{y}_t \mid \mathbf{y}_{t-1}\leqslant\mathbf{y}_t\leqslant\hat{\mathbf{y}}_{\bar{T}},\,H_t\mathbf{y}_t\leqslant\mathbf{0}\}$. Letting $\underline{t}=t-1$ and $\bar{t}=\bar{T}$, we see that $\mathbf{RMIP}_t(\mathbf{y}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$ simply relaxes $Y_{\underline{t}+1}(\mathbf{y}_{\underline{t}},\hat{\mathbf{y}}_{\bar{T}})$ in the formulation (46)–(47) and also replaces the piecewise-linear concave function of $\mathbf{y}_{\underline{t}}$, $\theta_{\underline{t}+1}(\mathbf{y}_{\underline{t}},\hat{\mathbf{y}}_{\bar{T}})$, with a different piecewise-linear concave function of $\mathbf{y}_{\underline{t}}$. Thus, standard convergence theory holds. □

The variables $\mathbf{s}_t$ are auxiliaries that help us compute approximations to a revised piecewise-linear function, but the model can be stated without them. The revised formulation ensures feasibility of the period-t subproblem provided that that $\hat{\mathbf{y}}_0\leqslant\hat{\mathbf{y}}_t\leqslant\hat{\mathbf{y}}_{\bar{T}}$, which is guaranteed, so only Benders optimality cuts need be generated.

FBD replaces $\theta_{t+1}(\mathbf{y}_t,\hat{\mathbf{y}}_{\bar{T}})$ in $\mathbf{RMIP}_t(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$ for each iteration k with a piecewise-linear, concave, upper approximation:

$$\bar{\bar{\theta}}_{t+1}^k(\mathbf{y}_t,\hat{\mathbf{y}}_{\bar{T}}) \equiv \min_{k'=1,\ldots,k-1} h_t^{k'} + \mathbf{g}_t^{k'}\mathbf{y}_t, \tag{A8}$$

with the cut coefficients $h_t^{k'}$ and $\mathbf{g}_t^{k'}$ defined in Equations (A14) and (A15) below. Making the replacement yields the following "approximate (nested) subproblem," with dual variables for the relevant constraints shown in brackets:

$$\mathbf{RMIP}_t^k(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}}):\ \bar{\bar{\theta}}_t^k(\hat{\mathbf{y}}_{t-1},\hat{\mathbf{y}}_{\bar{T}})$$
$$\equiv \max_{\mathbf{y}_t\geqslant\mathbf{0},\,\mathbf{s}_t\geqslant\mathbf{0},\,\bar{\bar{\theta}}_{t+1}} \mathbf{v}_t\mathbf{y}_t - \mathbf{p}_t\mathbf{s}_t + \bar{\bar{\theta}}_{t+1} \tag{A9}$$
$$\text{s.t.}\quad A\mathbf{y}_t - I\mathbf{s}_t \leqslant \mathbf{q}_t + A\hat{\mathbf{y}}_{t-1} \qquad [\boldsymbol{\alpha}_t^k] \tag{A10}$$
$$-I\mathbf{y}_t \leqslant -I\hat{\mathbf{y}}_{t-1} \qquad [\boldsymbol{\beta}_t^k] \tag{A11}$$
$$H_t\mathbf{y}_t \leqslant \mathbf{0} \qquad [\boldsymbol{\delta}_t^k] \tag{A12}$$
$$-\mathbf{g}_t^{k'}\mathbf{y}_t + \bar{\bar{\theta}}_{t+1} \leqslant h_t^{k'} \quad \text{for } k'=1,\ldots,k-1, \qquad [\gamma_t^{kk'}] \tag{A13}$$

where $\bar{\bar{\theta}}_{\bar{T}}\equiv 0$ and cuts (A13) are omitted for $t=T$. Note that (i) $\boldsymbol{\alpha}_t^k$, $\boldsymbol{\beta}_t^k$ and $\boldsymbol{\delta}_t^k$ are vectors while $\gamma_t^{kk'}$ is not; (ii) the generic dual variables $\boldsymbol{\pi}_t^k$ in the body of the paper now correspond to $(\boldsymbol{\alpha}_t^k,\boldsymbol{\beta}_t^k,\boldsymbol{\delta}_t^k,\boldsymbol{\gamma}_t^k)$, where we do now define the vector $\boldsymbol{\gamma}_t^k=(\gamma_t^{k1},\ldots,\gamma_t^{kk})$; and (iii) the vector $\boldsymbol{\delta}_t^k$ is not actually used in the cut-creation process because the right-hand side of the relevant constraints is $\mathbf{0}$.

Finally, we recursively define cut coefficients as follows:

$$\mathbf{g}^k_{t-1} = \hat{\boldsymbol{\alpha}}^k_t A - \hat{\boldsymbol{\beta}}^k_t I \quad \text{for } t = T, T-1, \ldots, 2, \tag{A14}$$

$$h^k_{t-1} = \hat{\boldsymbol{\alpha}}^k_t \mathbf{q}_t + \sum_{k'=1}^{k-1} \hat{\boldsymbol{\gamma}}^{kk'}_t h^{k'}_t \quad \text{for } t = T, T-1, \ldots, 2. \tag{A15}$$

FBD adds cuts (A13) dynamically, alternating between a primal pass and a dual pass. In this forward decomposition, the kth primal pass solves each approximate subproblem $\mathbf{RMIP}^k_t(\hat{\mathbf{y}}_{t-1}, \hat{\mathbf{y}}_{\bar{T}})$ for $\hat{\mathbf{y}}^k_t$ in the order $t = 1, \ldots, T$, given $\hat{\mathbf{y}}_0 \equiv \mathbf{0}$. Next, in the order $t = T, \ldots, 2$ and using values $\hat{\mathbf{y}}_t$ obtained in the most recent primal pass, the kth dual pass solves $\mathbf{RMIP}^k_t(\hat{\mathbf{y}}_{t-1}, \hat{\mathbf{y}}_{\bar{T}})$ for dual variables and adds cuts to update the approximating functions $\bar{\bar{\theta}}^k_t(\mathbf{y}_{t-1}, \hat{\mathbf{y}}_{\bar{T}})$. (The last step of the primal pass, i.e., when $t = T$, actually implements the first step of the dual pass.)

Given elastic resource constraints, the kth primal pass yields a primal feasible solution to **RMIP**, namely, $(\hat{\mathbf{y}}^k_1, \ldots, \hat{\mathbf{y}}^k_T)$. Thus, $\underline{\theta}^k \equiv \sum_{t=1}^T \mathbf{v}_t \hat{\mathbf{y}}^k_t \leqslant \theta^*$. Because $\bar{\bar{\theta}}^k_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) \geqslant \theta^*$, testing $\bar{\bar{\theta}}^k_1(\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_{\bar{T}}) - \max_k \underline{\theta}^k \leqslant \epsilon_{\mathbf{RMIP}}$, for some $\epsilon_{\mathbf{RMIP}} > 0$, yields an appropriate stopping criterion for the decomposition algorithm.

Appendix B. Solving RMIP with Hierarchical Benders Decomposition, BBD-A

This appendix describes the implementation of BBD-A, a bisection version of HBD that solves an elastic version of **RMIP**.

Assume that time periods $\underline{t} < t < \bar{t}$ are given, with

$$t = f(\underline{t}, \bar{t}) = \begin{cases} T, & \text{if } \underline{t} = 0 \text{ and } \bar{t} = \bar{T}; \\ \lfloor (\underline{t} + \bar{t})/2 \rfloor, & \text{otherwise.} \end{cases}$$

Also, assume the existence of solution estimates $\hat{\mathbf{y}}_{\underline{t}}$ and $\hat{\mathbf{y}}_{\bar{t}}$, with $\hat{\mathbf{y}}_0 \leqslant \hat{\mathbf{y}}_{\underline{t}} \leqslant \hat{\mathbf{y}}_{\bar{t}} \leqslant \hat{\mathbf{y}}_{\bar{T}}$. As in Appendix A, we solve a revised model with elastic resource constraints. In general, two groups of constraints must be elasticized, however, so we (i) define $\underline{\mathbf{s}}_t$ and $\bar{\mathbf{s}}_t$ analogous to $\mathbf{s}_t$ (see (A1)) and (ii) define $\underline{\mathbf{p}}_t$ and $\bar{\mathbf{p}}_t$ analogous to $\mathbf{p}_t$ (see (A2)).

Expanding on Equations (63) and (64), the following recursion defines the LP that is solved through BBD-A.

$\mathbf{RMIP}_t(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$: $\theta_t(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$

$$= \max_{\mathbf{y}_t \geqslant \mathbf{0},\, \underline{\mathbf{s}}_t,\, \bar{\mathbf{s}}_t \geqslant \mathbf{0}} \mathbf{v}_t \mathbf{y}_t - \underline{\mathbf{p}}_t \underline{\mathbf{s}}_t - \bar{\mathbf{p}}_t \bar{\mathbf{s}}_t + \theta_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \mathbf{y}_t) + \theta_{t''}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{t}}) \tag{B1}$$

$$\text{s.t.} \quad A\mathbf{y}_t - I\underline{\mathbf{s}}_t \leqslant \sum_{\tau=\underline{t}}^{t} \mathbf{q}_\tau + A\hat{\mathbf{y}}_{\underline{t}} \tag{B2}$$

$$-A\mathbf{y}_t - I\bar{\mathbf{s}}_t \leqslant \sum_{\tau=t+1}^{\bar{t}} \mathbf{q}_\tau - A\hat{\mathbf{y}}_{\bar{t}} \tag{B3}$$

$$-I\mathbf{y}_t \leqslant -I\hat{\mathbf{y}}_{\underline{t}} \tag{B4}$$

$$I\mathbf{y}_t \leqslant I\hat{\mathbf{y}}_{\bar{t}} \tag{B5}$$

$$H_t \mathbf{y}_t \leqslant \mathbf{0}, \tag{B6}$$

where $t' = f(\underline{t}, t)$ and $t'' = f(t, \bar{t})$. Of course, the recursion begins with $\underline{t} = t_0$ and $\bar{t} = \bar{T} = T + 1$ and is not carried further when $\bar{t} = \underline{t} + 1$.

In the most general case, an iteration k of HBD identifies cut coefficients $h^{k'}_{\underline{t}t}$, $\underline{\mathbf{g}}^{k'}_{\underline{t}t}$, $\bar{\mathbf{g}}^{k'}_{\underline{t}t}$, $h^{k'}_{t\bar{t}}$, $\underline{\mathbf{g}}^{k'}_{t\bar{t}}$, and $\bar{\mathbf{g}}^{k'}_{t\bar{t}}$ for $k' = 1, \ldots, k-1$ (see Equations (B17)–(B19), below) and replaces $\theta_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \mathbf{y}_t)$ and $\theta_{t''}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{t}})$. with these piecewise-linear, concave, upper approximations

$$\bar{\bar{\theta}}^k_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \mathbf{y}_t) = \min_{k'=1,\ldots,k-1} \{h^{k'}_{\underline{t}t} + \underline{\mathbf{g}}^{k'}_{\underline{t}t} \hat{\mathbf{y}}_{\underline{t}} + \bar{\mathbf{g}}^{k'}_{\underline{t}t} \mathbf{y}_t\} \quad \text{and} \tag{B7}$$

$$\bar{\bar{\theta}}^k_{t''}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{t}}) = \min_{k'=1,\ldots,k-1} \{h^{k'}_{t\bar{t}} + \underline{\mathbf{g}}^{k'}_{t\bar{t}} \mathbf{y}_t + \bar{\mathbf{g}}^{k'}_{t\bar{t}} \hat{\mathbf{y}}_{\bar{t}}\}, \quad \text{respectively.} \tag{B8}$$

Note that because t' is determined by $[\underline{t}, t]$ and t'' is determined by $[t, \bar{t}]$, $\underline{t}'$ and t'' are omitted in defining (B7) and (B8). Also note that $\bar{\bar{\theta}}^k_{t'}(\hat{\mathbf{y}}_{\underline{t}}, \mathbf{y}_t) \equiv 0$ if $t = \underline{t} + 1$ and $\bar{\bar{\theta}}^k_{t''}(\mathbf{y}_t, \hat{\mathbf{y}}_{\bar{t}}) \equiv 0$ if $t = \bar{t} - 1$. In the decomposition's kth iteration, the approximating model for $\mathbf{RMIP}_t(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$ is thus

$\mathbf{RMIP}^k_t(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$: $\theta^k_t(\hat{\mathbf{y}}_{\underline{t}}, \hat{\mathbf{y}}_{\bar{t}})$

$$= \max_{\mathbf{y}_t,\, \underline{\mathbf{s}}_t,\, \bar{\mathbf{s}}_t \geqslant \mathbf{0}} \mathbf{v}_t \mathbf{y}_t - \underline{\mathbf{p}}_t \underline{\mathbf{s}}_t - \bar{\mathbf{p}}_t \bar{\mathbf{s}}_t + \bar{\bar{\theta}}_{\underline{t}t} + \bar{\bar{\theta}}_{t\bar{t}} \tag{B9}$$

$$\text{s.t.} \quad A\mathbf{y}_t - I\underline{\mathbf{s}}_t \leqslant \sum_{\tau=\underline{t}}^{t} \mathbf{q}_\tau + A\hat{\mathbf{y}}_{\underline{t}} \qquad [\boldsymbol{\alpha}^k_{\underline{t}t}] \tag{B10}$$

$$-A\mathbf{y}_t - I\bar{\mathbf{s}}_t \leqslant \sum_{\tau=t+1}^{\bar{t}} \mathbf{q}_\tau - A\hat{\mathbf{y}}_{\bar{t}} \qquad [\boldsymbol{\alpha}^k_{t\bar{t}}] \tag{B11}$$

$$-I\mathbf{y}_t \leqslant -I\hat{\mathbf{y}}_{\underline{t}} \qquad [\boldsymbol{\beta}^k_{\underline{t}t}] \tag{B12}$$

$$I\mathbf{y}_t \leqslant I\hat{\mathbf{y}}_{\bar{t}} \qquad [\boldsymbol{\beta}^k_{t\bar{t}}] \tag{B13}$$

$$H_t \mathbf{y}_t \leqslant \mathbf{0} \qquad [\boldsymbol{\delta}^k_t] \tag{B14}$$

$$-\bar{\mathbf{g}}^{k'}_{\underline{t}t} \mathbf{y}_t + \bar{\bar{\theta}}_{\underline{t}t} \leqslant h^{k'}_{\underline{t}t} + \underline{\mathbf{g}}^{k'}_{\underline{t}t} \hat{\mathbf{y}}_{\underline{t}} \quad \text{for } k' = 1, \ldots, k-1 \quad [\gamma^{kk'}_{\underline{t}t}] \tag{B15}$$

$$-\underline{\mathbf{g}}^{k'}_{t\bar{t}} \mathbf{y}_t + \bar{\bar{\theta}}_{t\bar{t}} \leqslant h^{k'}_{t\bar{t}} + \bar{\mathbf{g}}^{k'}_{t\bar{t}} \hat{\mathbf{y}}_{\bar{t}} \quad \text{for } k' = 1, \ldots, k-1 \quad [\gamma^{kk'}_{t\bar{t}}] \tag{B16}$$

Note that (i) $\boldsymbol{\alpha}^k_{\underline{t}t}$, $\boldsymbol{\alpha}^k_{t\bar{t}}$, $\boldsymbol{\beta}^k_{\underline{t}t}$, $\boldsymbol{\beta}^k_{t\bar{t}}$ and $\boldsymbol{\delta}^k_t$ are vectors, while $\gamma^{kk'}_{\underline{t}t}$ and $\gamma^{kk'}_{t\bar{t}}$ are not; (ii) the generic dual vectors $\boldsymbol{\pi}^k_t$ in the text (see Figure 6(b)) correspond here to $(\boldsymbol{\alpha}^k_{\underline{t}t}, \boldsymbol{\alpha}^k_{t\bar{t}}, \boldsymbol{\beta}^k_{\underline{t}t}, \boldsymbol{\beta}^k_{t\bar{t}}, \boldsymbol{\delta}^k_t, \boldsymbol{\gamma}^k_{\underline{t}t}, \boldsymbol{\gamma}^k_{t\bar{t}})$, where we do now define vectors $\boldsymbol{\gamma}^k_{\underline{t}t} = (\gamma^{k1}_{\underline{t}t}, \ldots, \gamma^{kk'}_{\underline{t}t})$ and $\boldsymbol{\gamma}^k_{t\bar{t}} = (\gamma^{k1}_{t\bar{t}}, \ldots, \gamma^{kk'}_{t\bar{t}})$; and (iii) the vectors $\hat{\boldsymbol{\delta}}^k_t$ remain unused as in FBD and any other decomposition.

As before, we recursively define cut coefficients for (B15) and (B16):

$$h^k_{\underline{t}\bar{t}} = \hat{\boldsymbol{\alpha}}^k_{\underline{t}t} \sum_{\tau=\underline{t}}^{t} \mathbf{q}_\tau + \hat{\boldsymbol{\alpha}}^k_{t\bar{t}} \sum_{\tau=t+1}^{\bar{t}} \mathbf{q}_\tau + \sum_{k'=1}^{k-1} \hat{\gamma}^{kk'}_{\underline{t}t} h^{k'}_{\underline{t}t} + \sum_{k'=1}^{k-1} \hat{\gamma}^{kk'}_{t\bar{t}} h^{k'}_{t\bar{t}}, \tag{B17}$$

$$\underline{\mathbf{g}}^k_{\underline{t}\bar{t}} = \hat{\boldsymbol{\alpha}}^k_{\underline{t}t} A - \hat{\boldsymbol{\beta}}^k_{\underline{t}t} I + \sum_{k'=1}^{k-1} \hat{\gamma}^{kk'}_{\underline{t}t} \underline{\mathbf{g}}^{k'}_{\underline{t}t}, \tag{B18}$$

$$\bar{\mathbf{g}}^k_{\underline{t}\bar{t}} = -\hat{\boldsymbol{\alpha}}^k_{t\bar{t}} A + \hat{\boldsymbol{\beta}}^k_{t\bar{t}} I + \sum_{k'=1}^{k-1} \hat{\gamma}^{kk'}_{t\bar{t}} \bar{\mathbf{g}}^{k'}_{t\bar{t}}, \tag{B19}$$

where these formulas must be applied in a dual-pass order in the decomposition tree, with leaf vertices omitted.

We do not provide a formal proof that BBD-A converges because it may be viewed as a variant on FBD-A, which does converge. To see this equivalence, suppose that BBD-A performs primal passes in *level order* and dual passes in the reverse order. This means that all subproblems at the same depth in the tree are solved sequentially. But because the individual subproblems within a level are effectively independent, we could solve them simultaneously, as part of a single "level subproblem." Suppose we take this view and modify the decomposition algorithm to apply a single cut for each level subproblem. The "new algorithm" is really just FBD-A applied to a model that has been rearranged into $O(\log T)$ stages. Because BBD-A derives from this new algorithm by using a cut for each independent subproblem, we see that BBD-A is just a multicut version of FBD-A.

References

Ahuja R, Magnanti T, Orlin J (1993) *Network Flows* (Prentice Hall, Englewood Cliffs, NJ).

Ahuja RK, Hochbaum DS, Orlin JB (2003) Solving the convex cost integer dual network flow problem. *Management Sci.* 49(7):950–964.

Akaike A, Dagdelen K (1999) A strategic production scheduling method for an open pit mine. Dardano C, Francisco M, Proud J, eds. *Proc. 28th Internat. Sympos. Comput. Appl. Minerals Industries, APCOM '99* (Colorado School of Mines, Golden, CO), 729–738.

Appleget JA, Wood RK (2000) Explicit-constraint branching for solving mixed-integer programs. Laguna M, González-Velarde JL, eds. *Computing Tools for Modeling, Optimization and Simulation* (Kluwer Academic Publishers, Boston), 245–261.

Barnhart C, Johnson E, Nemhauser G, Savelsbergh M, Vance P (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Beale EML, Tomlin JA (1970) Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. Lawrence J, ed. *Proc. 5th IFORS Conf.* (Tavistock, London), 447–454.

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(3):238–252.

Bienstock D, Zuckerberg D (2010) Solving LP relaxations of large-scale precedence constrained problems. Eisenbrand F, Shepherd FB, eds. *Proc. 14th Internat. Conf. Integer Programming Combinatorial Optimization, IPCO '10*, Lecture Notes Comput. Sci., Vol. 6080 (Springer, Berlin), 1–14.

Birge JR (1997) State-of-the-art survey—stochastic programming: Computation and applications. *INFORMS J. Computing* 9(2):111–133.

Birge JR, Louveaux FV (1988) A multicut algorithm for two-stage stochastic linear programs. *Eur. J. Oper. Res.* 34(3):384–392.

Boland N, Dumitrescu I, Froyland G (2008) A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. http://www.optimization-online.org/DB_HTML/2008/10/2123.html.

Boland N, Dumitrescu I, Froyland G, Gleixner A (2009) LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Comput. Oper. Res.* 36(4):1064–1089.

Brown G, Keegan J, Vigus B, Wood K (2001) The Kellogg Company optimizes production, inventory and distribution. *Interfaces* 31(6):1–15.

Caccetta L, Hill S (2003) An application of branch and cut to open pit mine scheduling. *J. Global Optim.* 27(2–3):349–365.

Cai W (2001) Design of open-pit phases with consideration of schedule constraints. Xie H, Wang Y, Jiang Y, eds. *Proc. 29th Internat. Sympos. Comput. Appl. Minerals Industries, APCOM '01* (A.A. Balkema, Lisse, Netherlands), 217–221.

Chicoisne R, Espinoza D, Goycoolea M, Moreno E, Rubio E (2012) A new algorithm for the open-pit mine scheduling problem. *Oper. Res.* 60(3):517–528.

Cullenbine C, Wood K, Newman A (2011) A sliding time window heuristic for open pit mine block sequencing. *Optim. Lett.* 88(3):365–377.

Dagdelen K, Johnson T (1986) Optimum open pit mine production scheduling by Lagrangian parameterization. *Proc. 19th Internat. Sympos. Comput. Appl. Minerals Industries APCOM '86* (SME, Littleton, CO), 127–141.

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. *Oper. Res.* 8(1):101–111.

Denby B, Schofield D (1994) Open-pit design and scheduling by use of genetic algorithms. *Trans. Institution Mining Metallurgy Section A. Mining Indust.* 103:A21–A26.

Eivazy H, Askari-Nasab H (2012) A mixed integer linear programming model for short-term open pit mine production scheduling. *Mining Tech.* 121(2):97–108.

Elhedhli S, Goffin J-L (2004) The integration of an interior-point cutting plane method within a branch-and-price algorithm. *Math. Programming* 100(2):267–294.

Entriken R (1989) The parallel decomposition of linear programming. Doctoral dissertation, Stanford University, Stanford, CA.

Entriken R (1996) Parallel decomposition: Results for staircase linear programs. *SIAM J. Optim.* 6(4):961–977.

Epstein R, Goic M, Weintraub A, Catalán J, Santibáñez P, Urrutia R, Cancino R, Gaete S, Aguayo A, Caro F (2012) Optimizing long-term production plans in underground and open-pit copper mines. *Oper. Res.* 60(1):4–17.

Espinoza D, Goycoolea M, Moreno E, Newman A (2012) MineLib: A library of open pit mining problems. *Ann. Oper. Res.* 206(1):1–22.

Gabbay H (1979) Multi-stage production planning. *Management Sci.* 25(11):1138–1148.

Gassman H (1990) MSLiP: A computer code for the multistage stochastic linear programming problem. *Math. Programming* 47(1–3):407–423.

Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Sci.* 20(5):822–844.

Gershon M (1983) Optimal mine production scheduling: Evaluation of large scale mathematical programming approaches. *Internat. J. Mining Engrg.* 1(4):315–329.

Gershon M (1987) Heuristic approaches for mine planning and production scheduling. *Internat. J. Mining Geological Engrg.* 5(1):1–13.

Gholamnejad J, Moosavi E (2012) A new mathematical programming model for long-term production scheduling considering geological uncertainty. *J. Southern African Inst. Mining Metallurgy* 112(2):77–81.

Glassey CR (1971) Dynamic linear programs for production scheduling. *Oper. Res.* 19(1):45–56.

Glassey CR (1973) Nested decomposition and multi-stage linear programs. *Management Sci.* 20(3):282–292.

Gleixner A (2008) Solving large-scale open pit mining production scheduling problems by integer programming. Master's thesis, Technische Universität Berlin, Berlin, Germany.

Gondzio J, Sarkissian R, Vial J-P (1997) Using an interior point method for the master problem in a decomposition approach. *Eur. J. Oper. Res.* 101(3):577–587.

Ho J, Manne A (1974) Nested decomposition for dynamic models. *Math. Programming* 6(1):121–140.

Hochbaum D, Chen A (2000) Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. *Oper. Res.* 48(6):894–914.

Hummeltenberg W (1984) Implementations of special ordered sets in MP software. *Eur. J. Oper. Res.* 17(1):1–15.

IBM Corp. (2013) IBM ILOG CPLEX Optimization Studio CPLEX User's Manual, Version 12 Release 5. http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r5/topic/ilog.odms.studio.help/pdf/usrcplex.pdf.

IBM Corp. (2014) IBM ILOG CPLEX Optimization Studio CPLEX User's Manual, Version 12 Release 6. http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r6/topic/ilog.odms.studio.help/pdf/usrcplex.pdf.

Infanger G (1994) *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs* (Boyd & Fraser, Danvers, MA).

Johnson T (1968) Optimum open pit mine production scheduling. Doctoral dissertation, University of California, Berkeley, Berkeley.

Kallio MJ, Porteus EL (1977) Decomposition of arborescent linear programs. *Math. Programming* 33(1):348–356.

Karlsson J, Rönnqvist M, Bergström J (2003) Short-term harvest planning including scheduling of harvest crews. *Internat. Trans. Oper. Res.* 10(5):413–431.

Kawahata K (2007) A new algorithm to solve large scale mine production scheduling problems by using the Lagrangian relaxation method. Doctoral dissertation, Colorado School of Mines, Golden, CO.

Krige D (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. Chemical Metallurgical Mining Soc. South Africa* 52(6):119–139.

Lambert WB, Newman AM (2014) Tailored Lagrangian relaxation for the open pit block sequencing problem. *Ann. Oper. Res.* 222(1): 419–438.

Lambert WB, Brickey A, Newman AM, Eurek K (2014) Open-pit block-sequencing formulations: A tutorial. *Interfaces* 44(2):127–142.

Lerchs H, Grossmann I (1965) Optimum design of open-pit mines. *Canadian Mining Metallurgical Bull.* 68:17–24.

Linderoth JT, Savelsbergh MWP (1999) A computational study of search strategies for mixed integer programming. *INFORMS J. Comput.* 11(2):173–187.

Morton DP (1996) An enhanced decomposition algorithm for multi-stage stochastic hydroelectric scheduling. *Ann. Oper. Res.* 64(1): 211–235.

Pochet Y, Wolsey L (2006) *Production Planning by Mixed Integer Programming* (Springer, New York).

Ramazan S (2007) The new fundamental tree algorithm for production scheduling of open pit mines. *Eur. J. Oper. Res.* 177(2): 1153–1166.

Ramazan S, Dimitrakopoulos R (2007) Stochastic optimisation of long-term production scheduling for open pit mines with a new integer programming formulation. *Orebody Modelling and Strategic Mine Planning* (Australasian Institute of Mining and Metallurgy, Melbourne, Australia), 385–391.

Rojas CR, Goodwin GC, Seron MM, Zhang MM (2007) Open-cut mine planning via closed-loop receding-horizon optimal control. Sánchez Peña RS, Casín JQ, Cayuela VP, eds. *Identification and Control* (Springer, London), 43–62.

Rousseau L-M, Gendreau M, Feillet D (2007) Interior point stabilization for column generation. *Oper. Res. Lett.* 35(5):660–668.

Ruszczyński A (1986) A regularized decomposition method for minimizing a sum of polyhedral functions. *Math. Programming* 35(3): 309–333.

Ryan DM, Foster BA (1981) An integer programming approach to scheduling. Wren A, ed. *Comput. Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* (North-Holland, Amsterdam, Netherlands), 269–280.

Singh KJ, Philpott AB, Wood RK (2009) Dantzig-Wolfe decomposition for solving multistage stochastic capacity-planning problems. *Oper. Res.* 57(5):1271–1286.

Stolarczyk L (1992) Definition imaging of an orebody with the radio imaging method. *IEEE Trans. Indust. Appl.* 28(5):1141–1147.

Van Den Heever SA, Grossmann IE (2000) An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. *Indust. Engrg. Chemistry Res.* 39(6):1955–1971.

Whittle J (1998) *Four-X™ Strategic Planning Software for Open Pit Mines, Reference Manual* (Whittle Programming Pty Ltd., Melbourne, Australia).

Wittrock RJ (1985) Dual nested decomposition of staircase linear programs. *Math. Programming Study* 24:65–86.

Wollmer RD (1980) Two stage linear programming under uncertainty with 0–1 integer first stage variables. *Math. Programming* 19(1):279–288.

Thomas W. M. Vossen is an associate professor in the Management and Entrepreneurship division at Leeds School of Business, University of Colorado Boulder. His main research interest is the development and analysis of large-scale optimization methods to support planning and scheduling decisions, in application areas ranging from cutting-stock problems and open-pit mining to air traffic flow management and network revenue management.

R. Kevin Wood is a Distinguished Professor Emeritus of Operations Research at the Naval Postgraduate School (NPS). Retired as of 2015, he is continuing research on (a) mathematical-programming methods, (b) designing infrastructure that is robust against attack, and (c) skiing.

Alexandra M. Newman is a professor in the Mechanical Engineering Department at the Colorado School of Mines. Her research focuses on applied optimization, especially as it pertains to mine production scheduling and energy systems.