Management Science

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

A Decomposition-Based Algorithm for the Scheduling of Open-Pit Networks Over Multiple Time Periods

Michelle L. Blom, Adrian R. Pearce, Peter J. Stuckey

MANAGEMENT SCIENCE
Vol. 62, No. 10, October 2016, pp. 3059–3084
ISSN 0025-1909 (print) | ISSN 1526-5501 (online)

http://dx.doi.org/10.1287/mnsc.2015.2284

A Decomposition-Based Algorithm for the Scheduling of Open-Pit Networks Over Multiple Time Periods

Michelle L. Blom, Adrian R. Pearce, Peter J. Stuckey
Department of Computing and Information Systems, University of Melbourne, Parkville 3010, Victoria, Australia
{michelle.blom@unimelb.edu.au, adrianrp@unimelb.edu.au, pstuckey@unimelb.edu.au}

We consider the multiple-time-period, short-term production scheduling problem for a network of multiple open-pit mines and ports. Ore produced at each mine, in each period, is transported by rail to a set of ports and blended into products for shipping. Each port forms these blends to a specification, as stipulated in contracts with downstream customers. This problem belongs to a class of multiple producer/consumer scheduling problems in which producers are able to generate a range of products, a combination of which are required by consumers to meet specified demands. In practice, short-term schedules are formed independently at each mine, tasked with achieving a grade and quality target outlined in a medium-term plan. Because of uncertainty in the data available to a medium-term planner and the dynamics of the mining environment, such targets may not be feasible in the short term. In this paper, we present an algorithm in which the grade and quality targets assigned to each mine are iteratively adapted, ensuring the satisfaction of blending constraints at each port while generating schedules for each mine that maximise resource utilisation.

Keywords: short-term open-pit mine production scheduling; hybrid optimisation; nonlinear programming
History: Received June 18, 2014; accepted June 7, 2015, by Yinyu Ye, optimization. Published online in *Articles in Advance* January 8, 2016.

1. Introduction

We consider the multiple-time-period, multiple mine planning problem (MTP-MMPP) of scheduling the production of multiple open-pit mines, across a horizon of multiple time periods, to supply several ports with ore that can be blended to form products of a desired composition. This problem belongs to a class of multiple producer/consumer scheduling problems in which producers (mines) are able to generate a range of products (ore of varying grade and quality), and consumers (ports) require a combination (a blend) of these products to meet deterministic (known a priori) demands. We extend existing work by Blom et al. (2014) in which a decomposition-based algorithm for the single-time-period MMPP was developed. In this paper, we consider the significantly more complex multiple-time-period setting, in which multiple-time-period schedules must be generated for each mine that are both feasible to enact and lead to correct blending at the ports. We incorporate additional constraints into our modelling of the MMPP, not present in the single-time-period model of Blom et al. (2014), to develop a higher-fidelity representation of operational behaviour at each mine. Moreover, we describe a general class of multiple producer/consumer problems to which our solving approach can be applied, instances of which appear in a wide range of domains.

The MTP-MMPP is significantly more complex than the single-period MMPP solved by Blom et al. (2014). This is primarily due to the combinatorial complexity of selecting regions of material to extract across multiple time periods and several mines—where many extraction schedules are possible—to optimise an objective that is dependent on the activity at each mine in each period.

A solution to the short-term MTP-MMPP schedules the movement of material from available sources of ore and waste at each mine to appropriate destinations and the transport of ore between each mine and port in each time period of a scheduling horizon. In the short term, for the case study we consider in this paper, this horizon is 13 weeks long, split into weekly periods. During each time period, at each mine, ore from a variety of sources is processed and blended in a stockyard, producing ore of a specific grade and quality. Ore is reclaimed from this stockyard onto trains, railed to a port, and blended with ore from other mines to form desired products. An optimal solution to the MTP-MMPP requires coordination across mines. The grade and quality of production at each mine, in each period, must support the formation of correctly blended products at each port.

In practice, short-term planning in a network of open-pit mines proceeds with the independent construction of block extraction schedules at each mine, by individual short-term planners. In this process, the short-term planner at each mine is guided by a five-year, or medium-term, plan. This plan sets monthly grade and quality targets on mine production—assumed to be both achievable given the estimated composition of material in pit benches and supportive of port blending constraints. In the short term, such targets may not be achievable (in conjunction with full usage of processing plants) at one or more mine sites, during one or more time periods, jeopardising the production of blended products at each port. The collaborative adjustment of grade and quality targets assigned to a set of mines, in the generation of short-term plans, can ensure that each mine is assigned goals that can be met, while achieving maximal use of processing facilities.

We present a mixed integer nonlinear program (MINLP) modelling of the MTP-MMPP. This model is a bilinear program, involving the product of two continuous variables in its constraints. At the short-term horizon, a schedule for a mine site evaluates the average grade of processed material in each time period and assumes that each train departing the mine in that period carries this grade. We can imagine that all material processed at the site feeds into linearly blended "virtual" stockpiles, from which trains are loaded. This abstract view of mining operations is considered to be a reasonable approximation for the purposes of short-term scheduling. Bilinear constraints are required to ensure that the grade of material leaving each virtual stockpile is equivalent to the combined grade of the material entering it. The relaxation of these bilinear constraints, in terms of McCormick (1976) envelopes, results in solutions with significant deviations present between actual port product composition and desired bounds (as shown in §6). Tighter piecewise-linear relaxations of these constraints were considered for the single-period MMPP and were unable to satisfactorily reduce the resulting deviations (Blom et al. 2014).

The MTP-MMPP, as modelled in this paper, is similar in structure to a pooling problem (Haverly 1978). Such problems model the blending of materials in a feedforward network of source nodes (sources of ore at each mine), intermediate blending pools (virtual stockpiles), and terminal nodes (products formed at each port), under the assumption of linear blending at pools and terminals. Optimisation of the network assigns a rate of flow along each arc, such that profit is maximised and correctly blended products formed at terminals (Misener and Floudas 2009). The principal source of nonlinearity in the MTP-MMPP is the modelling of the composition of port products.

The assumption that ore produced by each mine is uniformly blended is reasonable in light of the way this material is stacked in the mine's stockyard and reclaimed onto trains. Each stockyard contains one or more chevron stacked stockpiles. Material is spread across these stockpiles in layers, each layer containing material of a similar composition. Each stockpile is reclaimed, once built to a desired size, in slices, extracting material from each layer in a roughly equal proportion. The blending efficiency of a stacking and reclaiming method is defined as the ratio of variance in the grade of material leaving the stockpile (σ^2_{out}) to that entering (σ^2_{in}) (Kumral 2006). Modern methods can achieve blending ratios of 1:10, reducing variance in grade by a factor of 10 (Müller 2010). For the mines and ports in the case study we consider in this paper, a 1:5 blending effect is achieved. In both planning and practice, running averages are used to represent stockpile composition (Everett 1996). Grade targets to be met by the blended products formed at the ports are placed on *average* grades. As ore moves through the supply chain in large quantities (trainloads are tens of kilotons, and each shipload ranging from tens to hundreds of kilotons), it is safe to assume that the average composition of reclaimed material does not differ appreciably from that of built stockpiles. In §6 we compute the average variance in grade in the blends formed at each port in our case study network, across solutions generated over multiple runs of our decomposition-based algorithm.

At each mine site in our case study, ore that is mined, but not processed, in a time period is placed on one of a number of stockpiles. Material on these stockpiles can be reclaimed and processed in a later period. Tracking the composition of stockpiles over multiple time periods is known to introduce nonlinearities into the modelling of a mine (Bley et al. 2012). Additional bilinear constraints, of the form introduced by Bley et al. (2012), are included in our MINLP to model the changing composition of stockpiles at each mine. We show in this paper that stockpiles cannot be modelled as maintaining a constant grade across the scheduling horizon without introducing significant errors in the estimated (versus actual) composition of blended port products.

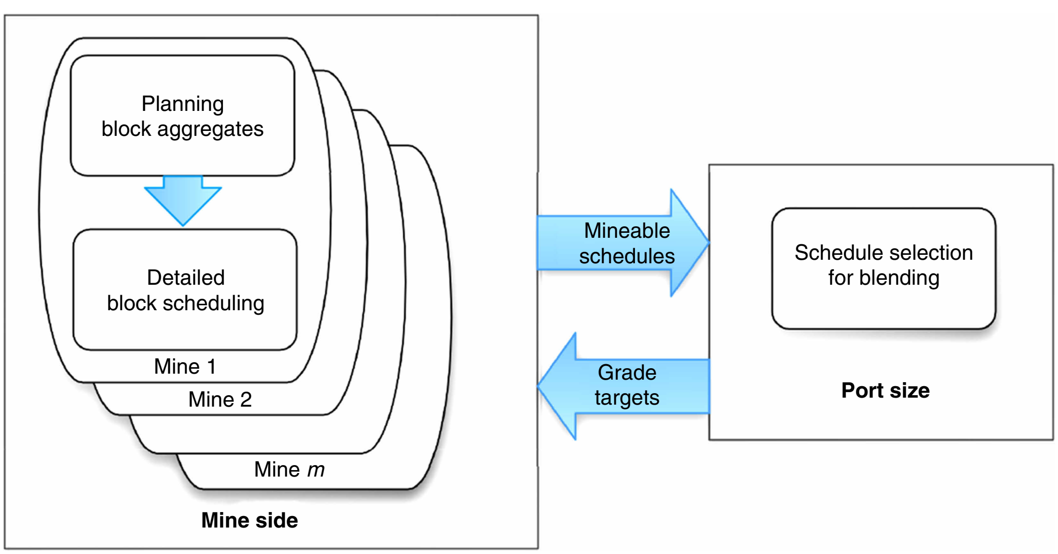
Blom et al. (2014) introduce an algorithm for discovering high-quality solutions to a MINLP model of the single-period MMPP. In this algorithm, the open-pit production scheduling task is decomposed into multiple subproblems: a set of optimisation problems solved on behalf of a set of mines (generating a set of candidate production schedules for each mine) and an optimisation problem solved on behalf of a network of ports (selecting a candidate schedule to be enacted at each mine and routing trains of ore between mines and ports). The solving of these subproblems is iterated, with the solution of each mine-side optimisation providing an input to the port-side problem, and the port-side optimisation informing the decisions made at each mine in subsequent iterations. In this paper, we adapt this algorithm to find high-quality solutions to the MTP-MMPP MINLP. The key differences between the single-period algorithm of Blom et al. (2014) and the multiple-time-period algorithm of this paper lies in the nature and implementation of the mine- and port-side optimisation problems and the feedback mechanism that exists between these problems in each iteration. A diagram of the main components of the algorithm is shown in Figure 1.

The algorithm presented in this paper, to the best of our knowledge, is the first to solve an integrated production scheduling problem involving multiple open-pit mines and a horizon of multiple periods, where the grade and quality of ore to be produced by each mine are not known a priori, but determined as part of the optimisation. We show that the algorithm of Blom et al. (2014) can be successfully adapted to the more complex multiple-time-period setting. Most crucial in this transition is the existence of a sufficiently fast and reliable method to generate multiple-time-period schedules for each mine in our network. Each schedule must achieve acceptable levels of equipment utilisation in each period, while producing ore that is close to a desired grade and quality. The decision of what to mine in each period has a direct influence on what regions of ore will be accessible later in the horizon. These complexities do not arise in the single-period problem of Blom et al. (2014). Full utilisation of available processing capacity is not enforced in this earlier work. Moreover, the need to ensure adequate availability of ore in future periods, or model the changing state of stockpiles, is not considered. Given the absence of appropriate techniques in the literature, we develop and present a novel heuristic approach for constructing such schedules.

In the multiple-time-period setting, each mine-side problem is solved via the use of hierarchical planning methods (Bitran and Hax 1977, Hax and Meal 1973), namely, decomposition, aggregation (of units of extraction, known as "blocks"), and a rolling horizon solution strategy. The task of generating a multiple-time-period extraction schedule for a mine is decomposed into two subproblems: a high-level *planning task*, in which blocks are grouped into aggregates and the mining of these aggregates is scheduled, and a *detailed scheduling problem*, in which we consider the mining of individual blocks. Both problems are solved with a rolling horizon heuristic. The latter problem is restricted to extracting blocks that form part of an aggregate mined in the solution to the high-level planning task. The result is an efficient heuristic for schedule generation at a mine. In the single-time-period setting of Blom et al. (2014), such schedules can be quickly found by solving a single mixed integer program (MIP). Given multiple time periods and the presence of bilinear constraints, generating a schedule by solving a monolithic MINLP is prohibitively time consuming, motivating the need for a heuristic method. Our rolling horizon heuristic monitors the changing state of each stockpile at a mine site, without the need for nonlinear constraints. It is important to emphasise, however, that an MIP model of this scheduling task, under the assumption of constant stockpile grades, is also prohibitively time consuming to solve. This is the result of combinatorial complexities that arise with the introduction of multiple interdependent time periods.

Each mine-side problem is designed to generate a set of (up to) N *mineable schedules* per mine, with the grade of the ore produced across these schedules clustered around a given grade and quality

Figure 1 (Color online) An Overview of the Decomposition-Based Algorithm for the MTP-MMPP



profile. The multiple-time-period port-side subproblem is designed to accept a set of (up to) N mineable block extraction schedules from each mine-side optimisation. Formulated as an MIP, a solution characterises the flow of ore between each mine and port, in each time period, and *selects a schedule*, from the N schedules available for each mine, to be enacted. The objective in this blending problem is to form products at each port whose composition, in each time period, does not deviate from desired bounds on grade and quality. Ore available in the stockyards of each mine in any given period t includes that produced by the mine in t and leftover material (ore produced but not railed to a port) from prior periods. The resulting MIP is more complex than the port-side MIP of Blom et al. (2014). The state of the stockyard of each mine in each period is no longer a known constant, but a variable dependent on the decisions of prior time periods. The port-side problem provides, as an output, *grade targets* forming the input to each mine-side optimisation in the next iteration. These targets are based on the grade and quality of ore produced by each mine, in each period, in the best solution found to the port-side MIP across all prior iterations. The schedules formed by each mine-side optimisation in the next iteration will produce ore whose grade is clustered around these new targets. Each mine is, in this way, guided toward finding solutions to its optimisation problem that allow each port to form correctly blended products.

The MTP-MMPP is an instance of a multiple period, multiple producer/consumer scheduling problem in which a set of independently operating producers supply a set of consumers with products to meet deterministic demands. Each consumer combines products sourced from multiple producers to generate their own products for export to an external market. We present two examples of such problems in the domain of food production: the harvesting, transport, and processing of grapes in a wine production supply chain and the preservation of fruit across a set of producers for supply to exporters, supermarkets, and other consumers. We describe how our decomposition-based approach to the scheduling of supply networks can be applied in these examples.

The key contributions of this paper are the presentation of a MINLP model of the MTP-MMPP; a decomposition-based algorithm for solving the MTP-MMPP, significantly extending that of Blom et al. (2014); an efficient heuristic for the generation of 13-week schedules at individual mine sites (with stockpiles), forming a core component of our approach; the investigation of several methods for the generation of feedback between the port- and mine-side subproblems in our algorithm; an evaluation of our algorithm on a real and currently operating network of open-pit mines, using industry-supplied data; and the identification of a general class of problems to which our approach can be applied. Our case study in this paper is of a currently operating network of eight mines and two ports, producing over 200 million tons of ore annually at the time of data collection.

The remainder of this paper is structured as follows. In §2, we highlight existing work related to the MTP-MMPP. We describe the MTP-MMPP domain in §3, presenting a MINLP representation of the problem in §4. We describe and evaluate our decomposition-based algorithm in §§5 and 6. In §7, we describe a class of multiple producer/consumer production planning problems to which the methodology we describe in this paper can be applied.

2. Related Work

Weintraub et al. (2008) propose the idea of aggregating MIP models, each designed to schedule a mine in a network of copper mines, for the purpose of making integrated decisions. Two types of aggregation are considered: the grouping of blocks into larger units of extraction and the aggregation of columns in an MIP representation of the scheduling problem. Both types of aggregation are applied to a single mine-scheduling problem, and the extent to which problem size is reduced is analysed. These techniques were not, however, applied to a case study involving multiple mines.

Although there exists work in which the mine-to-port transport problem in a network of multiple mines and ports is optimised (Thomas et al. 2012, 2014; Singh et al. 2013), the composition and tons of ore produced at each mine in each time period is known a priori, in contrast to the problem we tackle in this paper. Epstein et al. (2012) present an approach to integrate long-term production scheduling across multiple copper mines (both open-pit and underground) that share downstream processing plants. Each mine produces one or more commercial products, each with defined characteristics (copper grade and contaminant levels). Decision variables determine the tons of each product produced at each mine and the flow of this material through a network of stockpiles and plants. The grade and quality of production at each mine in this network is known a priori, with the tons of each type of product formed determined as part of the optimisation. The problem can thus be modelled as an MIP, in contrast to the MTP-MMPP, which we model as a MINLP, with decision variables denoting the grade and quality of production at each mine.

Our method of generating single mine extraction schedules, forming a key component of our algorithm, embodies the concepts of hierarchical production planning (Hax and Meal 1973, Bitran and Hax

1977). We first aggregate blocks into larger units of extraction and schedule the mining of these aggregates over our planning horizon. The original problem is then solved, removing from consideration all blocks not part of an aggregate scheduled for extraction.

The use of aggregation—of data or time—is commonly used to reduce the complexity of an optimisation problem (Rogers et al. 1991). Newman and Kuchta (2007) aggregate time periods into phases in an approach for long-term production scheduling at an underground mine. The key decision variable is machine placement—when to place machines in, and consequently mine, specific regions in the orebody. The phases in which machine placements are started in a solution to this aggregated model are used to restrict machine placements when solving the original problem. Tabesh and Askari-Nasab (2011) cluster blocks into larger units of mining on the basis of material type, grade, and location to reduce the complexity of long-term scheduling in a single open-pit mine. Ramazan (2007) aggregates blocks, in a long-term production scheduling problem, to form "fundamental trees"—minimal collections of blocks that have a positive total economic value and can be extracted without violating slope constraints—significantly reducing the number of integer variables required in MIP formulations of such problems. Boland et al. (2009) schedule the extraction of block aggregates in the long-term scheduling of an open-pit mine, while allowing decisions regarding what material is sent to a processing plant to be made at a sub-aggregate level.

For pooling problems involving time, such as the scheduling of crude oil refineries (Shah 1996, Wenkai et al. 2002, Màs and Pinto 2003, Reddy et al. 2004, Bengtsson et al. 2013) and the planning or design of oilfield infrastructure (Iyer and Grossmann 1998, van den Heever and Grossmann 2000, Carvalho and Pinto 2006), solution techniques are commonly hierarchical or decomposition-based in nature. Shah (1996) considers a crude oil scheduling problem in which oil arriving on ships is allocated into port tanks, piped into storage tanks at a refinery, and allocated to crude distillation units (CDUs) for processing. This problem is decomposed into two parts, solved sequentially: (1) an upstream problem responsible for the allocation, over time, of crude oil arriving on ships to port tanks and the sequence in which port tanks are discharged to feed a pipeline to the refinery, and (2) a downstream problem in which the refinery tanks being supplied by the pipeline and the tanks being discharged to specific CDUs in each time period are determined. We take a similar approach in our decomposition of the mining supply chain into a (downstream) mine-port transportation and blending problem and a series of (upstream) scheduling problems at each mine.

Sundar and Acharya (1995) present a two-stage approach for short-term scheduling at an open-pit mine, first determining the set of blocks to be blasted over the planning horizon, and then scheduling the extraction of material in some (or all) of these blocks in each period. Our method of schedule generation and the approach of Sundar and Acharya (1995) both select a subset of blocks to be excluded from any schedule formed. We aim to maximise the number of blocks in this set, to reduce the complexity of the scheduling problem, whereas Sundar and Acharya (1995) aim to minimise the size of this set, maximising the number of blocks blasted in the blasting schedule.

We make use of a rolling horizon technique to generate extraction schedules for individual mines. In this approach, the short-term horizon is discretised into two periods of length 1 and $T-1$, where T denotes the length of the horizon. A two-period scheduling problem is solved, in which the grade of stockpiles is assumed to remain constant, whereas their volume is permitted to vary, and the activity of period 1 is fixed. The grade of each stockpile at the mine at the end of period 1 is calculated. The remainder of the horizon (periods 2 to T) is rediscretised, and this process repeated (using the updated grades for each stockpile) until all T periods are scheduled. The use of a rolling horizon allows us to monitor the changing composition of stockpiles over time while avoiding the need for nonlinear constraints. Goodwin et al. (2006) solve a long-term mine-planning problem using a similar approach, termed receding horizon control. Time is discretised into periods of nonuniform size, with the quantisation becoming increasingly coarse toward the end of the horizon. The scheduling problem is solved, and the activities of the first period fixed. The remaining periods are rediscretised, and the scheduling problem solved on the reduced horizon. This process continues, and a schedule—whose time periods are uniformly discretised—is generated.

Cullenbine et al. (2011) describe a sliding time window heuristic (STWH) for solving a long-term, multiple period, open-pit block-sequencing problem, in which a series of integer programs (IPs) are generated and solved. In the first IP, the full set of problem constraints are enforced in the first τ time periods, and a Lagrangian relaxation of the model in the remainder. The solution to this IP is used to fix the variables in the first time period, after which the window of τ periods is moved forward by one, and a second IP enforcing all constraints between periods 2 and $1+\tau$ is solved. The heuristic repeats this process and terminates once the last period is scheduled. Lambert and Newman (2013) use the STWH to find an initial feasible solution to a constrained ultimate pit- and block-sequencing problem. Whereas Cullenbine et al. (2011) use Lagrangian relaxation to reduce the complexity of the generated IPs, our approach aggregates time periods, while enforcing all problem constraints in those periods. Our heuristic and the STWH do not guarantee that a schedule will be found. However, we demonstrate in §6 that the frequency with which our heuristic fails to form a solution for mines in our case study network is extremely small. Moreover, we show that a less aggressive aggregation of time periods, forming subproblems of more than two periods, does not result in an improvement to the quality of solutions found to the MTP-MMPP and incurs a significant computational penalty. Similarly, the increase in solve time resulting from setting $\tau > 1$ in the work of Cullenbine et al. (2011) was found to outweigh any improvement in solution quality.

Beyond the domain of mine planning, the use of aggregation, disaggregation, and rolling horizon-like techniques is prevalent in the production scheduling literature (for examples, see Rodrigues et al. 1996, Bassett et al. 1996, Dimitriadis et al. 1997, Elkamel et al. 1997, Iyer and Grossmann 1998, van den Heever and Grossmann 2000, Màs and Pinto 2003, Reddy et al. 2004, Méndez et al. 2006, Janak et al. 2006, Maravelias and Sung 2009). We refer the reader to Osanloo et al. (2008), Newman et al. (2010), Espinoza et al. (2012), and Lambert et al. (2014) for thorough reviews on the use of optimisation in open-pit mining, a description of common problems arising in the scheduling and operation of open-pit mines, unaddressed challenges for future research, and techniques for expediting the solving of block-sequencing formulations.

3. Modelling the Multiple Mine Network

Let $\mathcal{M}$ denote a set of mines, connected by rail to a set of ports, Π. We consider the open-pit mining of ores that are sold in two granularities—lumps and fines—distinguished by their particle size. At each mine $m \in \mathcal{M}$ in each period t, $t \in \{1, 2, \ldots, T\}$, ore and waste are extracted by dig units (e.g., loaders, shovels, and excavators) from geological regions (known as "blocks"), processed into lump (6 to 31 mm) and fine (<6 mm) granularities, and loaded onto trains to be railed to a port $\pi \in \Pi$. Ore arriving at each port is blended onto stockpiles, from which it is loaded onto ships for delivery to customers. Appendix A defines the notation used throughout this paper.

Each mine m contains a set of grade, $\mathcal{B}_m^g$, and blast blocks, $\mathcal{B}_m^b$. A short-term plan selects a number of blocks, from either set, to be extracted and the destination of this material (stockpiles or processing plants) in each time period. Grade blocks are regions of similar composition and form the broken stock of a mine—ore and waste that have been primed for extraction (blasted). Each blast block is a region of material that has not yet been blasted and divided into grade blocks. For each block $b \in \mathcal{B}_m = \mathcal{B}_m^g \cup \mathcal{B}_m^b$, $\mathcal{A}_{m,b}^{\wedge}$ denotes the set of blocks that lie directly above b, *all* of which must be mined before b can be accessed, and $\mathcal{A}_{m,b}^{\vee}$ denotes the set of blocks adjacent to $b \in \mathcal{B}_m$ in the same bench, only *one* of which must be mined before b can be accessed.

Each block $b \in \mathcal{B}_m$, at $t = 1$, contains one or more types of material—$T_{hi}^{t=1}(b)$ tons of high grade, $T_{lo}^{t=1}(b)$ tons of low grade, and $T_w^{t=1}(b)$ tons of waste—distributed throughout the block. For each grade block $b \in \mathcal{B}_m^g$, $T_k^{t=1}(b)$ is nonzero for only one value of k. The composition of the high-grade, low-grade, and waste material in $b \in \mathcal{B}_m$ is defined in terms of the percentage of a number of elements, $\mathcal{Q}$, denoting metal grade and impurities in its lump and fine components ($G_{b,l,q}^{m,k}$ for $q \in \mathcal{Q}$, $l \in \mathcal{L}$, and $k = hi, lo, w$). The split of material k in a block $b \in \mathcal{B}_m$, denoted $S_{b,l}^{m,k}$, defines the percentage of material k in b that will split (upon processing) into granularity $l \in \mathcal{L}$.

The waste in a block is hauled by truck to a waste dump ($\delta \in \Delta_m$). High-grade ore is hauled to a dry processing plant (κ) or a high-grade stockpile ($\theta \in \Theta_m$). Low-grade ore is hauled to a low-grade stockpile ($\lambda \in \Lambda_m$) or a wet processing plant (ω), if one exists at m. Material on high- and low-grade stockpiles is fed, if needed, to the dry and wet processing plants, respectively. The tons of material on each stockpile $s \in \Theta_m \cup \Lambda_m \cup \Delta_m$ at $t = 1$ is denoted by $T^{t=1}(s)$. The split of ore in a stockpile $s \in \Theta_m \cup \Lambda_m$ is denoted by $S_{s,l}^{m,t=1}$, and its composition by $G_{s,l,q}^{m,t=1}$, for $q \in \mathcal{Q}$ and $l \in \mathcal{L}$.

A wet processing plant upgrades low-grade ore, producing a stream of (rejected) tailings and a concentrate, as described by Blom et al. (2014). The tons of valuable metal (and other attributes) in this concentrate are proportional to the tons of input feed (as per a recovery factor $R_{l,q}^{m,\omega}$ for $q \in \mathcal{Q}$). The tons of concentrate produced are proportional to the tons of input feed (as per a yield factor $Y_l^{m,\omega}$). This concentrate is blended with lump and fine ore produced by the dry plant.

Capacities exist on the extraction of material at each mine, C_e^m tons per period, on the basis of available dig units; tons of material hauled by truck, C_τ^m tons per period; tons of ore processed by the dry and wet plants, C_κ^m and C_ω^m tons per period; tons of material permitted on each stockpile $s \in \Theta_m \cup \Lambda_m \cup \Delta_m$, C_s^m tons; and tons of ore handled at each port π per period, C_π. The capacity of each plant must be fully utilised to maximise production across the network. To maximise productivity, dig and trucking resources must also be fully utilised. In practice, full utilisation of both types of resource is not possible, because the dig and truck fleets will differ in capacity.

Ore produced by each mine is transported in T_R ton trainloads to a port $\pi \in \Pi$. Ore arriving at each port π is blended to form a set $N_{\pi,l}$ of products of each granularity $l \in \mathcal{L}$. Each product $n \in N_{\pi,l}$ is associated, in each time period t, with bounds on its grade and quality, expressed in terms of a lower ($L^{\pi,l,t}_{n,q}$) and upper ($U^{\pi,l,t}_{n,q}$) bound on the percentage of each $q \in \mathcal{Q}$.

The set of ore and waste sources at mine m is denoted $\mathcal{S}_m = \mathcal{B}_m \cup \Theta_m \cup \Lambda_m$. The set of destinations to which such sources can be transported is denoted $\mathcal{D}_m = \{\kappa, \omega\} \cup \Delta_m \cup \Theta_m \cup \Lambda_m$. Variables $x^{m,t}_{s,d}$ for $s \in \mathcal{S}_m$ and $d \in \mathcal{D}_m$ denote the tons of each source s extracted at m and hauled to each of its possible destinations d in period t. Variable $r^{l,n,t'}_{m,\pi,t}$ denotes the floating point number of trainloads of granularity $l \in \mathcal{L}$ produced by mine m in period t' and transported by rail from m to port π in period t to be blended into product $n \in N_{\pi,l}$. The port system need not rail the entirety of a mine's production to a port in any given time period. We assume that any remaining ore in the stockyard of a mine at the end of a time period t must be transported to a port in period $t+1$.

Our modelling of the MTP-MMPP differs from the single-period problem of Blom et al. (2014) in that scheduling decisions are indexed by time; blocks contain multiple types of material, each of which must be sent to a different destination; wet and dry plants are constrained to be fully utilised in each period; high- and low-grade stockpile composition is tracked over time (via the addition of nonlinear constraints); and the integrality of trainloads is relaxed. The transport of ore between mines and ports is not modelled in great detail, with emphasis placed on mapping ore produced at each mine to the products shipped from each port. Relaxing trainload integrality allows us to model the distribution of trainloads of ore to multiple products at a port and the transport of partial trainloads of "leftover" ore produced at a mine in prior periods but not railed to a port. Although, in practice, "leftover" material is blended with new production and transported in full trainloads, this approximation avoids the need for additional nonlinear constraints modelling the changing composition of stockyard stockpiles. A higher-fidelity modelling of the transport network between mines and ports and the operational processes of each port is planned as future work.

The addition of time periods, in conjunction with the mining precedences that exist between blocks, significantly increases the complexity of the MMPP. Each mining decision has an influence on what can be accomplished in remaining time periods and whether an adequate supply of processable ore, available for extraction, will exist to meet the future needs of each plant.

4. A MINLP Model of the MTP-MMPP

Given a network of mines $\mathcal{M}$, ports Π, and parameters (of Appendix A), the MTP-MMPP is defined as finding an instantiation of variables $x^{m,t}_{s,d}$ for each $m \in \mathcal{M}$, $t \in \{1, 2, \ldots, T\}$, $s \in \mathcal{S}_m$, and $d \in \mathcal{D}_m$, and $r^{l,n,t'}_{m,\pi,t}$ for each $m \in \mathcal{M}$, $\pi \in \Pi$, $t, t' \in \{1, 2, \ldots, T\}$, $l \in \mathcal{L}$, and $n \in N_{\pi,l}$. A solution to the MTP-MMPP defines the flow of ore and waste across each mine and the transport of ore between each mine and port in each of the T time periods in our planning horizon.

Our objective in the MTP-MMPP is to minimise the total deviation present between the composition of products formed at each port, over the given horizon, and desired upper and lower bounds, while maximising the productivity of each mine. We define productivity in terms of the "desirable utilisation" of dig and trucking resources. Dig and trucking capacity not used in the extraction of ore to supply processing plants should be used for other purposes—for example, the removal of waste to expose ore for future extraction. The transfer of ore to a stockpile and then from that stockpile to a plant—a process known as "double handling"—is to be avoided, unless required to ensure that processing plants are able to be run continuously at full capacity. This may occur if a mine needs to enter a phase of increased waste removal, reducing the portion of resource capacities devoted to mining processable ore. In this instance, ore from stockpiles, located near processing plants, can be used to offset the reduction in processable ore being supplied from each pit.

4.1. The Objective

Let variable $\overline{rv}^{m,l,q,t'}_{\pi,n,t}$ denote the tons of attribute $q \in \mathcal{Q}$, from granularity l produced at mine m in period t', blended into product $n \in N_{\pi,l}$ at port π in period t; let β_1 and β_2 be constants such that $\beta_1 \gg \beta_2$; let $\vec{x}$ and $\vec{r}$ be the sets of all $x^{m,t}_{s,d}$ and $r^{m,l,t'}_{\pi,n,t}$ variables, respectively, where $m \in \mathcal{M}$, $s \in \mathcal{S}_m$, $d \in \mathcal{D}_m$, $\pi \in \Pi$, $l \in \mathcal{L}$, $n \in N_{\pi,l}$, and $t, t' \in \{1, 2, \ldots, T\}$; let $\eta(\vec{x}, \vec{r}, t)$ be a measure of the extent to which port products deviate from desired bounds, in period t (Equation (2)); and let $\rho_m(\vec{x}, t)$ be the productivity of mine m in period t (Equation (3)). The objective of the MINLP, $Z_{\text{MTP–MMPP}}$ in Equation (1), is to minimise deviation that exists between the composition of port products and desired bounds and to maximise the productivity achieved at each mine across all time periods:

$$Z_{\text{MTP–MMPP}} = \min \sum_{t=1}^{T} \left[\beta_1 \eta(\vec{x}, \vec{r}, t) - \beta_2 \sum_{m \in \mathcal{M}} \rho_m(\vec{x}, t) \right]. \quad (1)$$

In the following, Δ^+_q denotes a significant change in the percentage of attribute $q \in \mathcal{Q}$ in a body of ore,[1] Φ^m_ω

is a binary parameter whose value is 1 if mine m has facilities to process low-grade ore and 0 otherwise, $f(t)=\max(1,t-1)$, and $g(t)=\min(t+1,T)$:

$$\begin{aligned}\eta(\vec{x},\vec{r},t) &= \sum_{\pi\in\Pi}\sum_{l\in\mathcal{L}}\sum_{n\in N_{\pi,l}}\sum_{q\in\mathcal{Q}}\frac{1}{\Delta_q^+}\\ &\cdot\max\bigg[0,\sum_{m\in\mathcal{M}}\sum_{t'=f(t)}^{t}\overline{rv}_{\pi,n,t}^{m,l,q,t'}-U_{n,q}^{\pi,l,t}\sum_{m\in\mathcal{M}}\sum_{t'=f(t)}^{t}T_R r_{\pi,n,t}^{m,l,t'}\bigg]\\ &+\sum_{\pi\in\Pi}\sum_{l\in\mathcal{L}}\sum_{n\in N_{\pi,l}}\sum_{q\in\mathcal{Q}}\frac{1}{\Delta_q^+}\max\bigg[0,L_{n,q}^{\pi,l,t}\sum_{m\in\mathcal{M}}\sum_{t'=f(t)}^{t}T_R r_{\pi,n,t}^{m,l,t'}\\ &\qquad -\sum_{m\in\mathcal{M}}\sum_{t'=f(t)}^{t}\overline{rv}_{\pi,n,t}^{m,l,q,t'}\bigg]. \qquad (2)\end{aligned}$$

Equation (3) calculates the productivity of a mine in terms of the tons of waste hauled to a dump and the occurrence of undesirable stockpiling. Hauling low-grade ore to a stockpile is considered undesirable only if the mine has facilities for its upgrade:

$$\rho_m(\vec{x},t)=\sum_{s\in\mathcal{S}_m}\bigg[\sum_{\delta\in\Delta_m}x_{s,\delta}^{m,t}-\sum_{\theta\in\Theta_m}x_{s,\theta}^{m,t}+(1-2\Phi_\omega^m)\sum_{\lambda\in\Lambda_m}x_{s,\lambda}^{m,t}\bigg]. \qquad (3)$$

4.2. Constraints

For brevity, we present the subset of constraints in the MTP-MMPP that either do not appear in the single-period MINLP of Blom et al. (2014) or differ significantly in the multiple-period setting. The full set of constraints in our MINLP model of the MTP-MMPP is presented in Appendix B.

Let $\tau_l^{m,t}$ denote the tons of granularity l produced at m in t; let $v_{l,q}^{m,t}$ denote the (fractional) percentage of $q\in\mathcal{Q}$ in the ore of granularity l produced at m in t; let $y_{m,t}^{\sigma,b}$ be a binary variable with value 1 if and only if block b is at least partially mined by the end of period t; let $y_{m,t}^{\tau,b}$ be a binary variable with value 1 if and only if b is completely mined by the end of t; let $a_{m,l}^{t,s,d}$ denote the tons of granularity l in the material on stockpile $s\in\Theta_m\cup\Lambda_m$ sent to destination $d\in\{\kappa,\omega\}$ in t; let $a_{m,l,q}^{t,s,d}$ denote the tons of attribute q in the ore of granularity l sent to destination d from stockpile s in t; let $o_s^{m,t}$ denote the tons of material on stockpile s at the start of t; let $o_{s,l}^{m,t}$ denote the tons of granularity l in the material on stockpile s at the start of t; and let $o_{s,l,q}^{m,t}$ denote the tons of attribute q in the ore of granularity l on stockpile s at the start of t. The values of all variables, excepting binaries $y_{m,t}^{\tau,b}$ and $y_{m,t}^{\sigma,b}$, are restricted to nonnegative reals.

Constraint (4) ensures that in each period, the processing of ore at each plant is equal, within a tolerance ϵ, to C_d^m for $d\in\{\kappa,\omega\}$. Constraint (5) ensures that all ore produced by each mine m in each period t is transported to a port by the end of period $t+1$:

$$C_d^m-\epsilon\le\sum_{s\in\mathcal{S}_m}x_{s,d}^{m,t}\le C_d^m+\epsilon \quad \forall\, m\in\mathcal{M},\, d\in\{\kappa,\omega\},\, t\in\{1,2,\dots,T\}; \qquad (4)$$

$$\sum_{t=t'}^{g(t')}\sum_{\pi\in\Pi}\sum_{n\in N_{\pi,l}}T_R r_{\pi,n,t}^{m,l,t'}=\tau_l^{m,t'} \quad \forall\, m,l,t'\in\{1,2,\dots,T\}. \qquad (5)$$

For $k\in\{hi,lo,w\}$, constraint (6) sets the value of binaries $y_{m,t}^{\sigma,b}$ (1 if the mining of $b\in\mathcal{B}_m$ has been scheduled by or during t) and $y_{m,t}^{\tau,b}$ (1 if b is to be entirely extracted by or during t):

$$y_{m,t}^{\tau,b}\sum_k T_k^{t=1}(b)\le\sum_{t'=1}^{t}\sum_{d\in\mathcal{D}_m}x_{b,d}^{m,t'}\le y_{m,t}^{\sigma,b}\sum_k T_k^{t=1}(b) \quad \forall\, m\in\mathcal{M},\, b\in\mathcal{B}_m,\, t\in\{1,2,\dots,T\}. \qquad (6)$$

Constraint (7) ensures that no blast blocks are mined prior to a specific time period, TB_m, at each mine $m\in\mathcal{M}$. Prior to TB_m, only grade blocks can be extracted at m. Constraints (8)–(11) ensure that stockpile capacities are respected and that no more than $o_s^{m,t}$ tons (the tons of ore on stockpile s at the start of t) can be extracted from any stockpile $s\in\Theta_m\cup\Lambda_m$ in any period t:

$$y_{m,t}^{\sigma,b}=0 \quad \forall\, m\in\mathcal{M},\, b\in\mathcal{B}_m^b,\, t<TB_m; \qquad (7)$$

$$o_s^{m,t}=T^{t=1}(s) \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, t=1; \qquad (8)$$

$$o_s^{m,t}=o_s^{m,t-1}-\sum_{d\in\{\kappa,\omega\}}x_{s,d}^{m,t-1}+\sum_{b\in\mathcal{B}_m}x_{b,s}^{m,t-1} \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, t\in\{2,3,\dots,T\}; \qquad (9)$$

$$o_s^{m,t}-\sum_{d\in\{\kappa,\omega\}}x_{s,d}^{m,t}+\sum_{b\in\mathcal{B}_m}x_{b,s}^{m,t}\le C_s^m \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, t\in\{2,3,\dots,T\}; \qquad (10)$$

$$o_s^{m,t}-\sum_{d\in\{\kappa,\omega\}}x_{s,d}^{m,t}\ge 0 \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, t\in\{2,3,\dots,T\}. \qquad (11)$$

Constraints (12)–(15) define the tons of each granularity $l\in\mathcal{L}$ and attribute $q\in\mathcal{Q}$ residing on each stockpile $s\in\Theta_m\cup\Lambda_m$ at the start of period t:

$$o_{s,l}^{m,t}=S_{s,l}^{m,t=1}T^{t=1}(s) \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, l\in\mathcal{L},\, t=1; \qquad (12)$$

$$o_{s,l}^{m,t}=o_{s,l}^{m,t-1}-\sum_{d\in\{\kappa,\omega\}}a_{s,l}^{m,t-1,d}+\sum_{b\in\mathcal{B}_m}S_{b,l}^m x_{b,s}^{m,t-1} \quad \forall\, m\in\mathcal{M},\, s\in\Theta_m\cup\Lambda_m,\, l\in\mathcal{L},\, t\in\{2,3,\dots,T\}; \qquad (13)$$

[1] The value of Δ_q^+ may be 0.1%, for example, if q denotes metal percentage, or on the order of 0.001% for an impurity.

$$o^{m,t}_{s,l,q} = S^{m,t=1}_{s,l} G^{m,t=1}_{s,l,q} T^{t=1}(s) \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, q \in \mathcal{Q}, t = 1; \tag{14}$$

$$o^{m,t}_{s,l,q} = o^{m,t-1}_{s,l,q} - \sum_{d \in \{\kappa,\omega\}} a^{m,t-1,d}_{s,l,q} + \sum_{b \in \mathcal{B}_m} S^m_{b,l} G^m_{b,l,q} x^{m,t-1}_{b,s} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, q \in \mathcal{Q}, t \in \{2,3,\ldots,T\}. \tag{15}$$

Constraints (16) and (17) ensure that the composition of material leaving each stockpile $s \in \Theta_m \cup \Lambda_m$ for a processing plant $d \in \{\kappa, \omega\}$ in each period t is equal to that of the stockpile at the start of t. These bilinear constraints, introduced in the work of Bley et al. (2012), add $\sum_m T|\mathcal{L}||\Theta_m \cup \Lambda_m|$ and $\sum_m T|\mathcal{L}||\mathcal{Q}| \cdot |\Theta_m \cup \Lambda_m|$ bilinear terms to the model, respectively. Each stockpile can supply either the dry (κ) or wet (ω) processing plants, but not both. Constraints restricting the movement of material along valid source to destination pathways are provided in Appendix B.

$$a^{m,t,d}_{s,l} o^{m,t}_{s} = x^{m,t}_{s,d} o^{m,t}_{s,l} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, d \in \{\kappa,\omega\}, l \in \mathcal{L}, t \in \{1,2,\ldots,T\}; \tag{16}$$

$$a^{m,t,d}_{s,l,q} o^{m,t}_{s,l} = a^{m,t,d}_{s,l} o^{m,t}_{s,l,q} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, d \in \{\kappa,\omega\}, l \in \mathcal{L}, q \in \mathcal{Q}, t \in \{1,2,\ldots,T\}. \tag{17}$$

Variables $\tau^{m,t}_l$ and $v^{m,t}_{l,q}$ are linked by constraint (18), where $\nu^{m,t}_{l,q}$ denotes the tons of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in t. The equations defining variables $\tau^{m,t}_l$, $v^{m,t}_{l,q}$, and the quantity $\nu^{m,t}_{l,q}$, are provided in Appendix B. Constraint (19) defines variable $\overline{rv}^{m,l,q,t'}_{\pi,n,t}$, used in Equation (2). Constraints (18) and (19) introduce $T|\mathcal{M}||\mathcal{L}|\,|\mathcal{Q}|$ and $\sum_{\pi\in\Pi}\sum_{l\in\mathcal{L}}(2T-1) \cdot |\mathcal{Q}||N_{\pi,l}|$ bilinear terms to the model, respectively:

$$v^{m,t}_{l,q} \tau^{m,t}_l = \nu^{m,t}_{l,q} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, l \in \mathcal{L}, q \in \mathcal{Q}; \tag{18}$$

$$T_R r^{m,l,t'}_{\pi,n,t} v^{m,t'}_{l,q} = \overline{rv}^{m,l,q,t'}_{\pi,n,t} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, t' \in \{f(t), t\}, l \in \mathcal{L}, q \in \mathcal{Q}, n \in N_{\pi,l}. \tag{19}$$

5. A Decomposition-Based Algorithm

Blom et al. (2014) present an iterative, decomposition-based algorithm for the single-time-period MMPP. The steps of this algorithm, shown in Algorithm 1, remain the same in the case of multiple time periods. The mine- and port-side subproblems solved in each iteration, however, are significantly more complex in the multiple-time-period setting. The key elements of this algorithm are the decomposition of the MTP-MMPP into mine- and port-side subproblems, denoted $\mathcal{O}_m$ and $\mathcal{O}_\Pi$, respectively; the solving of each $\mathcal{O}_m$ and $\mathcal{O}_\Pi$ in succession to produce a solution to the MTP-MMPP (steps 7 and 8); the recording of the best found solution, denoted by $\vec{s}_{\text{best}}$, after each iteration (step 9); and the passing of feedback, denoted by $\mathcal{O}_\Pi$, to each $\mathcal{O}_m$ at the end of each iteration (step 10).

Each $\mathcal{O}_m$ subproblem is responsible for generating a set of (up to) N block extraction schedules for mine $m \in \mathcal{M}$ (step 7). A grade and quality target, denoted by $\vec{\phi}_m$, stating the desired percentage of each attribute $q \in \mathcal{Q}$ in each granularity $l \in \mathcal{L}$ to be produced by m in each time period, is provided as input. A set of (up to) N schedules, whose production lies in the vicinity of this target, is formed. Recall that each schedule defines the movement of material across mine m, from each source $s \in \mathcal{S}_m$ (blocks and stockpiles) to each destination $d \in \mathcal{D}_m$ (stockpiles and plants), in each time period t, instantiating variables $x^{m,t}_{s,d}$ for all $t \in \{1, 2, \ldots, T\}$.

Diversity in the grade of ore produced, across the N schedules formed by $\mathcal{O}_m$, is controlled by a vector of standard deviations $\vec{\sigma}_m$, containing one standard deviation for each combination of $q \in \mathcal{Q}$, $l \in \mathcal{L}$, and $t \in \{1, 2, \ldots, T\}$, forming a second input to each $\mathcal{O}_m$. Smaller standard deviations lead to a set of schedules across which the grade of produced ore is more tightly clustered around the assigned target, $\vec{\phi}_m$. Section 5.1 describes in more detail how each $\mathcal{O}_m$ is formulated and solved.

The $\mathcal{O}_\Pi$ subproblem is given a set of (up to) N schedules from each $\mathcal{O}_m$ and must select one schedule in each set to be enacted (step 8). The goal of $\mathcal{O}_\Pi$ is to form correctly blended products at each port from the ore produced at each mine. If $\mathcal{O}_\Pi$ cannot find such a selection of schedules, one for each mine, for which port products are correctly formed, the selection that allows it to minimise deviation between the grade and quality of port products and desired bounds is made. In doing so, the remaining variables in our MINLP are instantiated, forming a solution to the MTP-MMPP. Section 5.3 describes, in more detail, how $\mathcal{O}_\Pi$ is formulated and solved.

Our decomposition-based algorithm repeats the solving of each $\mathcal{O}_m$ and $\mathcal{O}_\Pi$, in sequence, generating a series of monotonically improving solutions to the MTP-MMPP. The quality of solution $\vec{s}_i$, found by $\mathcal{O}_\Pi$ in iteration i, is given by its objective value, $Z_{\text{MTP–MMPP}}(\vec{s}_i)$, as per Equation (1). A record of the best solution found by $\mathcal{O}_\Pi$, $\vec{s}_{\text{best}}$, is maintained over the course of the algorithm. A new solution, $\vec{s}_i$, replaces $\vec{s}_{\text{best}}$ if and only if $Z_{\text{MTP–MMPP}}(\vec{s}_i) < Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$ (step 9). In each iteration $i > 1$, $\mathcal{O}_\Pi$ is able to select the schedule chosen for mine m in $\vec{s}_{\text{best}}$, denoted by $\vec{s}_{\text{best},m}$, in place of those newly generated, ensuring that $Z_{\text{MTP–MMPP}}(\vec{s}_i) \le Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$, for all $i > 1$.

Our algorithm encourages each $\mathcal{O}_m$ to construct schedules that will allow $\mathcal{O}_\Pi$ to form correctly blended products at each port. This is accomplished via the use of feedback, passed from $\mathcal{O}_\Pi$ to each $\mathcal{O}_m$, at the end of each iteration (step 10). The composition of

Algorithm 1 (A decomposition-based algorithm for the MTP-MMPP, where Δ_q^+ and Δ_q^- denote significant and insignificant changes in $q \in \mathcal{Q}$ percentage, respectively; Ξ_m denotes a medium-term grade and quality target assigned to mine $m \in \mathcal{M}$; and MAX_i is a cap on the number of iterations of the algorithm executed)

1: $\vec{s}_{\text{best}} \leftarrow \varnothing$
2: $\vec{\sigma}^+ \leftarrow \{\sigma_{l,q}^+ = \Delta_q^+ \mid l \in \mathcal{L},\ q \in \mathcal{Q}\}$
3: $\vec{\sigma}^- \leftarrow \{\sigma_{l,q}^- = \Delta_q^- \mid l \in \mathcal{L},\ q \in \mathcal{Q}\}$
4: $i \leftarrow 1$
5: Initialise expected mine targets and standard deviation sets: $\vec{\phi}_m^i \leftarrow \Xi_m$ and $\vec{\sigma}_m^i \leftarrow \vec{\sigma}^+$, for all $m \in \mathcal{M}$.
6: **repeat**
7: Solve each $\mathcal{O}_m$ to find (up to) N schedules for mine m, Ω_m^i, producing ore whose composition is clustered around $\vec{\phi}_m^i$ with a spread determined by the standard deviations in $\vec{\sigma}_m^i$.
8: Solve $\mathcal{O}_\Pi$ given sets $\Omega_m^i \cup \{\vec{s}_{\text{best},m}\}$ from each $m \in \mathcal{M}$, where $\vec{s}_{\text{best},m} \in \vec{s}_{\text{best}}$ is the schedule to be enacted by m in the best solution found thus far. Select a schedule to be enacted at each mine, and a routing of ore between mines and ports, forming a solution $\vec{s}_i$ to the MTP-MMPP.
9: Update best solution $\vec{s}_{\text{best}}$ if and only if $Z_{\text{MTP–MMPP}}(\vec{s}_i) < Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$.
10: Generate feedback to each $\mathcal{O}_m$ by adapting $\vec{\phi}_m^i$ and $\vec{\sigma}_m^i$ to form $\vec{\phi}_m^{i+1}$ and $\vec{\sigma}_m^{i+1}$.
11: $i \leftarrow i+1$
12: **until** $[Z_{\text{MTP–MMPP}}(\vec{s}_i) \geq Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}}) \wedge \nexists m \in \mathcal{M}(\vec{\sigma}_m^i \neq \vec{\sigma}^-)] \vee i > \text{MAX}_i$
13: **return** $\vec{s}_{\text{best}}$

production at each mine m, in $\vec{s}_{\text{best}}$, forms a new grade target, given to $\mathcal{O}_m$ as input in the next iteration.

The set of standard deviations given to each $\mathcal{O}_m$ in each iteration serves a dual purpose. These deviations are increased and decreased by $\mathcal{O}_\Pi$ over the course of the algorithm to encourage more or less diversity in the composition of produced ore across the schedule sets generated by each $\mathcal{O}_m$. When these standard deviations reach a minimum size, termination of the algorithm is triggered, at which point the best found solution, $\vec{s}_{\text{best}}$, is returned (steps 12 and 13). This termination mechanism and the feedback that takes place between $\mathcal{O}_\Pi$ and each $\mathcal{O}_m$, influencing production across the schedules formed by $\mathcal{O}_m$ in successive iterations, are described in more detail in §5.4.

The algorithm presented in this section is an anytime approach, forming a solution to the MTP-MMPP in each iteration. A cap on the number of iterations performed, MAX_i, can be specified to keep runtime within an acceptable range. In practice, we have found the algorithm to terminate within 200 iterations in each of the experiments described in §6.

5.1. Optimisation at the Mines: The $\mathcal{O}_m$ Problem

A solution to each $\mathcal{O}_m$, in iteration i, is a set of (up to) N extraction schedules, denoted by Ω_m^i. Each schedule instantiates variables $x_{s,d}^{m,t}$, from the MINLP of §4, for all $t \in \{1, 2, \ldots, T\}$, $s \in \mathcal{S}_m$, and $d \in \mathcal{D}_m$. A grade and quality target $\vec{\phi}_m^i = \{\phi_{l,q}^{m,t,i} \mid t \in \{1, 2, \ldots, T\}, l \in \mathcal{L}, q \in \mathcal{Q}\}$ and standard deviation vector $\vec{\sigma}_m^i = \{\sigma_{l,q}^{m,t,i} \mid t \in \{1, 2, \ldots, T\}, l \in \mathcal{L}, q \in \mathcal{Q}\}$ are given to $\mathcal{O}_m$ as input. We devised a heuristic to generate Ω_m^i, across which the percentage of $q \in \mathcal{Q}$ in the ore of granularity l produced by m in each t is clustered around $\phi_{l,q}^{m,t,i}$, with a spread determined by the standard deviation $\sigma_{l,q}^{m,t,i}$.

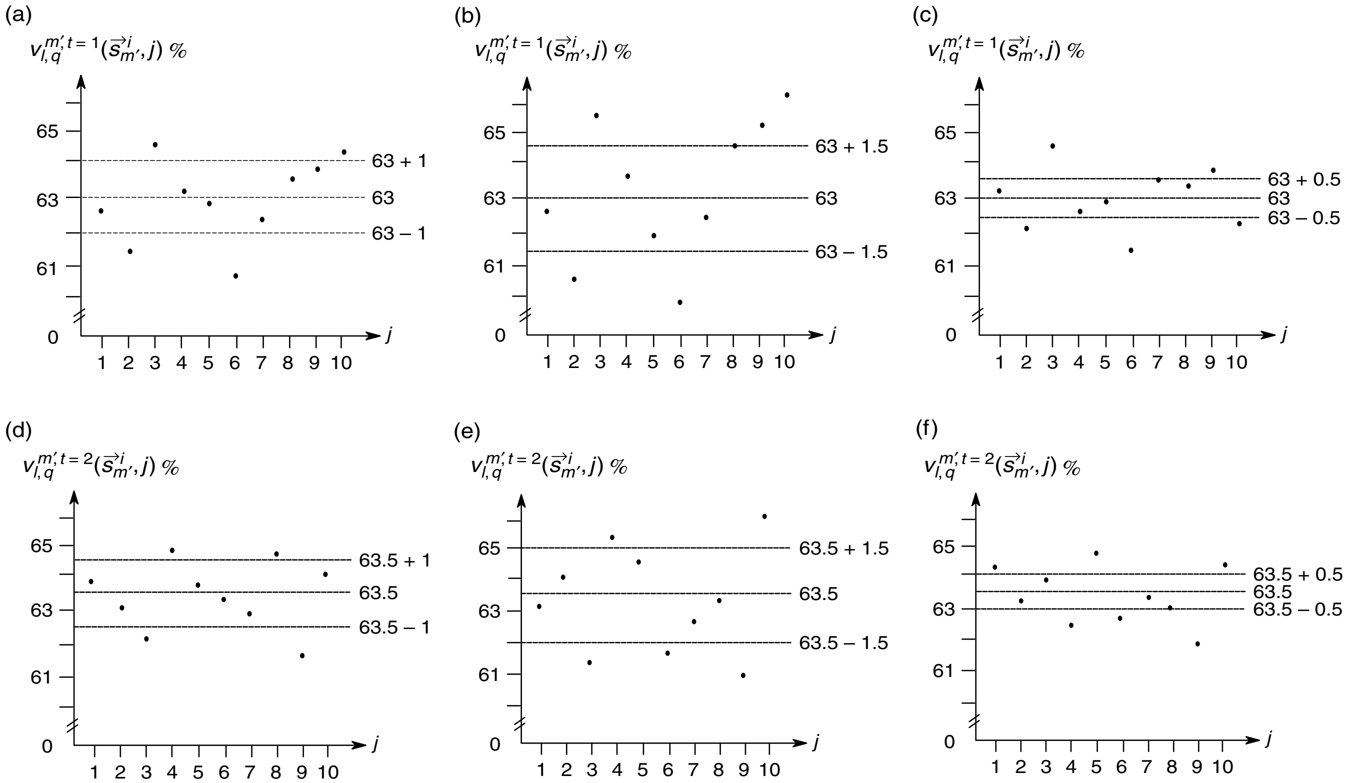
Let $\vec{s}_{m,j}^i \in \Omega_m^i$ denote the jth schedule generated by $\mathcal{O}_m$, in iteration i, and let $v_{l,q}^{m,t}(\vec{s}_{m,j}^i)$ denote the value of variable $v_{l,q}^{m,t}$ in schedule $\vec{s}_{m,j}^i$. Given a mine $m' \in \mathcal{M}$, a grade and quality target $\vec{\phi}_{m'}^{i=1}$, and a vector of standard deviations $\vec{\sigma}_{m'}^{i=1}$, Example 5.1 presents the first schedule generated by $\mathcal{O}_{m'}$ in iteration $i=1$, $\vec{s}_{m',j=1}^{i=1}$. Examples 5.3–5.7 demonstrate how this schedule is formed.

Example 5.1. Mine m' produces ore of a single granularity l, characterised by a single attribute q (metal grade) over three periods $t = 1, 2, 3$. For all t, $\phi_{l,q}^{m',t,i=1}$ is 62.5%, and $\sigma_{l,q}^{m',t,i=1}$ is 0.4. Mine m' contains grade blocks $\mathcal{B}_{m'}^g = \{b_0, b_1, b_2, b_3, b_4\}$, blast blocks $\mathcal{B}_{m'}^b = \{b_5, b_6, \ldots, b_{14}\}$, and a waste dump δ. For each block $b \in \mathcal{B}_m \setminus \{b_7, b_8, b_9\}$, $T_b^{hi} = 10$ kt. Moreover, $G_{b,l,q}^{m',hi} = 63\%$ for $b = b_0, b_5, b_6$; $G_{b,l,q}^{m',hi} = 62\%$ for $b = b_1, b_2, b_{10}$; $G_{b,l,q}^{m',hi} = 62.5\%$ for $b = b_3, b_{11}, b_{12}$; and $G_{b,l,q}^{m',hi} = 63.5\%$ for $b = b_4, b_{13}, b_{14}$. Blocks b_7, b_8, and b_9, each contain 20 kt of waste, and no low- or high-grade ore. The capacity of the dry plant κ at m' is 20 kt (per period). Mine m' can extract up to 20 kt of material in $t=1$ and 30 kt in periods $t = 2, 3$. The first schedule generated by $\mathcal{O}_{m'}$ in iteration $i=1$ is shown below. Schedule $\vec{s}_{m',j=1}^{i=1}$ produces ore of grades 62.88%, 62.4%, and 62.98% in periods 1, 2, and 3, respectively:

$$\vec{s}_{m',j=1}^{i=1} = \begin{cases} x_{b_0,\kappa}^{m,t=1} = 2\text{ kt}, & x_{b_0,\kappa}^{m,t=2} = 8\text{ kt}, \quad x_{b_2,\kappa}^{m,t=3} = 7\text{ kt}, \\ x_{b_2,\kappa}^{m,t=1} = 1\text{ kt}, & x_{b_1,\kappa}^{m,t=2} = 10\text{ kt}, \quad x_{b_4,\kappa}^{m,t=3} = 3\text{ kt}, \\ x_{b_3,\kappa}^{m,t=1} = 10\text{ kt}, & x_{b_2,\kappa}^{m,t=2} = 2\text{ kt}, \quad x_{b_{13},\kappa}^{m,t=3} = 10\text{ kt}, \\ x_{b_4,\kappa}^{m,t=1} = 7\text{ kt}, & x_{b_7,\delta}^{m,t=2} = 10\text{ kt}, \quad x_{b_8,\delta}^{m,t=3} = 10\text{ kt}. \end{cases}$$

For $i=1$, $\vec{\phi}_m^{i=1}$ is initialised with an expected target, denoted by Ξ_m, derived from a medium-term (five-year) plan. This plan provides a target composition

Figure 2 **A Set of $N = 10$ Schedules for Mine m', Producing a Single Granularity l, Characterised by One Quality Attribute q, Over Two Periods, Formed with Varying $\vec{\phi}^i_{m'}$ and $\vec{\sigma}^i_{m'}$**



Notes. For $\phi^{m',t=1,i}_{l,q} = 63\%$ and $\phi^{m',t=2,i}_{l,q} = 63.5\%$, (a) and (d) plot the value of variable $v^{m',t}_{l,q}$ across schedules when $\sigma^{m',t,i}_{l,q} = 1$, for all t. Similarly, (b) and (e) and (c) and (f) depict the values of $v^{m',t}_{l,q}$ across schedules when $\sigma^{m',t,i}_{l,q} = 1.5$ and $\sigma^{m',t,i}_{l,q} = 0.5$, respectively, for all t.

for the ore mine m should produce in each period t of the short-term horizon, to ensure correct blending at the ports. We initialise $\vec{\sigma}^{i=1}_m$ with large enough values to ensure significant diversity across Ω^i_m in the grade of produced ore.

Example 5.2. Consider mine m' of Example 5.1 in periods $t = 1, 2$. Figure 2 shows the value of $v^{m',t}_{l,q}(\vec{s}^i_{m',j})$ across 10 possible schedules for differing targets $\vec{\phi}^i_{m'}$ and standard deviations $\vec{\sigma}^i_{m'}$. Larger values for each $\sigma^{m',t,i}_{l,q}$ encourage the creation of a schedule set, across which the percentage of q in produced ore exhibits a greater range of values. Smaller values for each $\sigma^{m',t,i}_{l,q}$ result in schedules across which the percentage of q in produced ore is more tightly clustered around $\vec{\phi}^i_{m'}$.

To construct $\vec{s}^i_{m,j}$, for each $j = 1, \ldots, N$, we first define the desired upper and lower bounds on each $v^{m,t}_{l,q}(\vec{s}^i_{m,j})$, denoted by $[L^{m,t,j}_{l,q}, U^{m,t,j}_{l,q}]$, via Algorithm 2. A normally distributed random value $\Delta^{j,t}_{l,q}$, for each $j \in \{1 \ldots N\}$, $t \in \{1, 2, \ldots, T\}$, $l \in \mathscr{L}$, and $q \in \mathscr{Q}$ is generated from a distribution with mean 0 and standard deviation $\sigma^{m,t,i}_{l,q}$ (step 4). Shifting $\phi^{m,t,i}_{l,q}$ by $\Delta^{j,t}_{l,q}$ and subtracting (adding) a small quantity, Δ^-_q, to the result,[2] computes our lower $L^{m,t,j}_{l,q}$ (and upper $U^{m,t,j}_{l,q}$) bound (steps 5 and 6). By forming a set of varying bounds on the grade and quality of production in each period t, we are able to generate multiple schedules that produce varying grades of ore in each period t.

Example 5.3 (Example 5.1 Continued). Let $\Delta^-_q = 0.01\%$. Algorithm 2, for $j = 1$, computes $\Delta^{j,t=1}_{l,q} = 0.3$, $\Delta^{j,t=2}_{l,q} = -0.2$, and $\Delta^{j,t=3}_{l,q} = 0.5$. The following bounds on the percentage of q in ore produced by m' in periods $t = 1, 2$, and 3 are formed. Recall that $\phi^{m',t,i=1}_{l,q} = 62.5\%$ and $\sigma^{m',t,i=1}_{l,q} = 0.4$ for all t;

$$L^{m',t=1,j=1}_{l,q} = 62.7\%, \quad L^{m',t=2,j=1}_{l,q} = 62.2\%,$$
$$L^{m',t=3,j=1}_{l,q} = 62.9\%,$$
$$U^{m',t=1,j=1}_{l,q} = 62.9\%, \quad U^{m',t=2,j=1}_{l,q} = 62.4\%,$$
$$U^{m',t=3,j=1}_{l,q} = 63.1\%.$$

We apply a two-stage process to generate a schedule, $\vec{s}^i_{m,j}$, for which the deviation between each

[2] Here Δ^-_q denotes an insignificant change in the percentage of $q \in \mathscr{Q}$ in an orebody.

$v^{m,t}_{l,q}(\vec{s}^{\,i}_{m,j})$ and bounds $[L^{m,t,j}_{l,q}, U^{m,t,j}_{l,q}]$ is minimised as a first priority, and the total productivity at mine m maximised as a second. The productivity achieved at mine m in period t of $\vec{s}^{\,i}_{m,j}$, denoted by $\rho_m(\vec{s}^{\,i}_{m,j}, t)$, is computed as per Equation (3). In the first stage of this process, we aggregate blocks in the grade and blast block models of each mine into larger units of extraction, given a maximum aggregate size of M_A blocks. The set of block aggregates at mine m is denoted by $\mathscr{B}^A_m$, where $IN(a)$ denotes the subset of $\mathscr{B}_m$ in aggregate $a \in \mathscr{B}^A_m$ ($IN(a) \subset \mathscr{B}_m$). The procedure used to create these block aggregates is, for brevity, omitted from this paper. A rolling horizon heuristic, described in Algorithm 3 and §5.2, generates a block extraction schedule, identifying which of these aggregated blocks are to be mined in each period t and the destination of this mined material.

Algorithm 2 (Generation of bounds on the blend of produced ore at mine $m \in \mathscr{M}$)
1: **for** $j \leftarrow 1$ to N **do**
2: **for** $t \leftarrow 1$ to T **do**
3: **for** each $l \in \mathscr{L}$ and $q \in \mathscr{Q}$ **do**
4: $\Delta^{j,t}_{l,q} \leftarrow$ RandNormal$(0, \sigma^{m,t}_{l,q})$ where $\sigma^{m,t}_{l,q} \in \vec{\sigma}_m$
5: $L^{m,t,j}_{l,q} \leftarrow \phi^{m,t}_{l,q} + \Delta^{j,t}_{l,q} - \Delta^-_q$
6: $U^{m,t,j}_{l,q} \leftarrow \phi^{m,t}_{l,q} + \Delta^{j,t}_{l,q} + \Delta^-_q$
7: **end for**
8: **end for**
9: **end for**

Let $\tilde{s}^i_{m,j}$ and $\bar{\mathscr{B}}^A_m$ respectively denote the schedule generated by our rolling horizon heuristic, given blocks $\mathscr{B}^A_m$, and the set of aggregates in $\mathscr{B}^A_m$ that have been mined (partially or completely) in $\tilde{s}^i_{m,j}$. In the second stage of our method of mine schedule generation, we reapply the rolling horizon heuristic of §5.2 with respect to a new set of blocks, $\mathscr{B}^*_m = \mathscr{B}^g_m \cup \{b \mid b \in \mathscr{B}^b_m \wedge \exists a \in \bar{\mathscr{B}}^A_m \cdot b \in IN(a)\}$, to generate schedule $\vec{s}^{\,i}_{m,j}$. The set $\mathscr{B}^*_m$ denotes the union of the set of grade blocks $\mathscr{B}^g_m$ with the set of all blast blocks in $\mathscr{B}^b_m$ that appear in a mined aggregate $a \in \bar{\mathscr{B}}^A_m$. Our first application of the rolling horizon heuristic seeks to reduce the complexity of the problem of generating $\vec{s}^{\,i}_{m,j}$, by reducing the number of blocks that can be considered for mining. The result is an efficient method for the generation of extraction schedules, given a large number of available blocks.

We show in §6 that the use of an initial scheduling pass to reduce the number of blocks under consideration in the generation of schedules by each $\mathscr{O}_m$ substantially reduces the runtime of our algorithm while preserving, on average, the quality of solutions found.

Example 5.4 (Example 5.3 Continued). Let $M_A = 2$. We form the following aggregates of blocks in $\mathscr{B}_{m'}$: $a_0 = \{b_0, b_3\}$; $a_1 = \{b_1, b_2\}$; $a_2 = \{b_4\}$; $a_3 = \{b_5, b_6\}$; $a_4 = \{b_{10}\}$; $a_5 = \{b_{11}, b_{12}\}$; $a_6 = \{b_{13}, b_{14}\}$; and waste aggregates $a_7 = \{b_7, b_8\}$ and $a_8 = \{b_9\}$. The percentages of q in a_0–a_6 are 62.75%, 62%, 63.5%, 63%, 62%, 62.5%, and 63.5%, respectively. Examples 5.5–5.7 show how our rolling horizon heuristic generates $\tilde{s}^{i=1}_{m',j=1}$, scheduling the mining of aggregates $\mathscr{B}^A_{m'} = \{a_0, \dots, a_8\}$ given the bounds on grade shown in Example 5.3. In $\tilde{s}^{i=1}_{m',j=1}$, aggregates a_0–a_2 and a_6–a_7 are mined. Thus, $\bar{\mathscr{B}}^A_{m'} = \{a_0, a_1, a_2, a_6, a_7\}$ and $\mathscr{B}^*_{m'} = \{b_0, b_1, b_2, b_3, b_4, b_7, b_8, b_{13}, b_{14}\}$. Reapplication of the rolling horizon heuristic, restricted to the mining of blocks in $\mathscr{B}^*_{m'}$, yields $\vec{s}^{\,i=1}_{m',j=1}$, as shown in Example 5.1.

5.2. A Rolling Horizon Heuristic

Algorithm 3 presents a rolling horizon heuristic for the generation of an extraction schedule, $\tilde{s}_m$, for a single mine $m \in \mathscr{M}$. Given a set of sources $\mathscr{S}$ containing blocks $\mathscr{B} \subset \mathscr{S}$ and stockpiles $\Theta_m \cup \Lambda_m \subset \mathscr{S}$, a set of destinations $\mathscr{D}_m$, T time periods, and bounds on the composition of production $[L^{m,t}_{l,q}, U^{m,t}_{l,q}]$ for all $l \in \mathscr{L}$, $q \in \mathscr{Q}$, and $t \in \{1, 2, \dots, T\}$, $\tilde{s}_m$ is formed by solving $T-1$ two-time-period MIPs and a final one-time-period MIP. Starting at $t' = 1$, the horizon is decomposed into two periods, h_1 and h_2, of size 1 and $T - t'$ (steps 1 and 2). A two-time-period extraction schedule is formed (steps 7–11), and the activities of h_1 become part of schedule $\tilde{s}_m$ (step 12). This process is repeated for $t' \in \{2, 3, \dots, T\}$ (steps 15 and 16). The heuristic terminates, returning $\tilde{s}_m$, after period T is scheduled.

Given time periods h_1 and h_2, a two-time-period extraction schedule is constructed as follows. In step 7, the bounds $[L^{m,t}_{l,q}, U^{m,t}_{l,q}]$, for each $l \in \mathscr{L}$, $q \in \mathscr{Q}$, and $t \in \{t', \dots, T\}$, are averaged across the periods represented by h_1 and h_2 to form new bounds $[L^{m,h}_{l,q}, U^{m,h}_{l,q}]$ for all combinations of $l \in \mathscr{L}$, $q \in \mathscr{Q}$, and $h \in \{h_1, h_2\}$. For $t' = 1$, h_1 denotes period 1, and h_2 denotes periods $\{2, 3, \dots, T\}$. Let $\tilde{x}^{m,h}_{s,d}$ denote the tons of source $s \in \mathscr{S}$ hauled to destination $d \in \mathscr{D}_m$ in period h, and $\tilde{\mathbf{x}}_m = \{\tilde{x}^{m,h}_{s,d} \mid s \in \mathscr{S},\ d \in \mathscr{D}_m,\ h \in \{h_1, h_2\}\}$. A schedule for periods h_1 and h_2 is formed by solving two MIPs, $\mathscr{O}_{m,1}$ and $\mathscr{O}_{m,2}$, in succession.

Example 5.5 (Example 5.4 Continued). To generate $\tilde{s}^{i=1}_{m',j=1}$ in Example 5.4, we first assign period $t = 1$ to h_1 and periods $t = 2, 3$ to h_2. The bounds shown in Example 5.3 for periods $t = 2, 3$ are averaged to form bounds on the percentage of q in the ore produced at m' in h_1 and h_2:

$$L^{m',h_1}_{l,q} = 62.7\%, \qquad L^{m',h_2}_{l,q} = 62.55\%,$$
$$U^{m',h_1}_{l,q} = 62.9\%, \qquad U^{m',h_2}_{l,q} = 62.75\%.$$

The objective of $\mathscr{O}_{m,1}$, denoted $Z_{\mathscr{O}_{m,1}}$, acts to minimise deviation between the composition of each granularity $l \in \mathscr{L}$ formed at mine m in each period

Algorithm 3 (Generation of schedule $\tilde{s}_m$ for mine $m \in \mathcal{M}$, given a set of material sources $\mathcal{S}$ (including a set of blocks $\mathcal{B} \subset \mathcal{S}$), material destinations $\mathcal{D}_m$, T time periods, and bounds $[L^{m,t}_{l,q}, U^{m,t}_{l,q}]$ on each $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ produced at m in each period $t \in \{1, 2, \ldots, T\}$)

1: $h_1 \leftarrow 1$
2: $h_2 \leftarrow \{2, 3, \ldots, T\}$
3: Initialise block tonnages $T^{t=1}_k(b)$ for all $k \in \{hi, lo, w\}$ and $b \in \mathcal{B}$.
4: Initialise stockpile tonnages $T^{t=1}(s)$, ore splits $S^{t=1}_{s,l}$, and grades $G^{t=1}_{s,l,q}$ for all $s \in \Theta_m \cup \Lambda_m$, $l \in \mathcal{L}$, and $q \in \mathcal{Q}$.
5: $\tilde{s}_m \leftarrow \varnothing$
6: **while** $h_1 \leq T$ **do**
7: Compute bounds on the percentage of $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ to be produced at m in periods h_1 and h_2, denoted $[L^{m,h}_{l,q}, U^{m,h}_{l,q}]$, by averaging $[L^{m,t}_{l,q}, U^{m,t}_{l,q}]$ over $t = h_1$ and (if $h_1 < T$) $t = h_1 + 1$ to $t = T$.
8: Solve an MIP, denoted $\mathcal{O}_{m,1}$, to schedule the extraction of blocks in $\mathcal{B}$ across time periods h_1 and h_2, while minimising deviation between the chemistry of produced ore and desired bounds.
9: If a solution to $\mathcal{O}_{m,1}$ could not be found, terminate and **return** $\varnothing$.
10: Minimally shift $L^{m,h}_{l,q}$ or $U^{m,h}_{l,q}$ for each $q \in \mathcal{Q}$, $l \in \mathcal{L}$, and $h \in \{h_1, h_2\}$ to cover the solution obtained to $\mathcal{O}_{m,1}$. Incorporate these bounds into $\mathcal{O}_{m,1}$, as hard constraints, forming a new MIP, $\mathcal{O}_{m,2}$.
11: Alter the objective of $\mathcal{O}_{m,2}$ to maximise productivity at mine m, and solve $\mathcal{O}_{m,2}$ to schedule the extraction of blocks in $\mathcal{B}$ across time periods h_1 and h_2.
12: Fix the values of variables $\tilde{x}^{m,t=h_1}_{s,d}$ in schedule $\tilde{s}_m$, for each $s \in \mathcal{S}$ and $d \in \mathcal{D}_m$, to those of corresponding variables $\tilde{x}^{m,h_1}_{s,d}$ in the solution obtained to $\mathcal{O}_{m,2}$, if found, and to $\mathcal{O}_{m,1}$ otherwise.
13: Compute remaining block tonnages $T^{t=h_1+1}_k(b)$ for all $k \in \{hi, lo, w\}$ and $b \in \mathcal{B}$.
14: Compute updated stockpile tonnages $T^{t=h_1+1}(s)$, ore splits $S^{t=h_1+1}_{s,l}$, and grades $G^{t=h_1+1}_{s,l,q}$ for all $s \in \Theta_m \cup \Lambda_m$, $l \in \mathcal{L}$, and $q \in \mathcal{Q}$.
15: $h_1 \leftarrow h_1 + 1$
16: $h_2 \leftarrow \{h_1 + 1, \ldots, T\}$ if $h_1 < T$ and $\varnothing$ otherwise.
17: **end while**
18: **return** $\tilde{s}_m$

$h \in \{h_1, h_2\}$ and desired bounds $[L^{m,h}_{l,q}, U^{m,h}_{l,q}]$ for all $q \in \mathcal{Q}$ (step 8). A measure of the extent of this deviation is denoted by $\eta_m(\tilde{\mathbf{x}}_m, h)$ and is computed as shown in Equation (22), where $\nu^m_{l,q}(\tilde{\mathbf{x}}_m, h)$ denotes the tons of attribute $q \in \mathcal{Q}$ in the ore of granularity l formed by m in period h (see Equation (25)), and $\tau^m_l(\tilde{\mathbf{x}}_m, h)$ denotes the tons of granularity l formed by m in h (see Equation (24)). Constraints (B1)–(B3), (B6)–(B13), (B20)–(B24), and (B29) from the MINLP of Appendix B form the constraints of $\mathcal{O}_{m,1}$, redefined for a single mine m in terms of variables $\tilde{\mathbf{x}}_m$. In both $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$, the composition of each low- and high-grade stockpile s at m is assumed to have a constant split and grade, denoted by $S^{t=h_1}_{s,l}$ and $G^{t=h_1}_{s,l,q}$, respectively, for granularity $l \in \mathcal{L}$ and attribute $q \in \mathcal{Q}$ across periods h_1 and h_2. We place a time limit on the solving of $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$, accepting the best solution found by the time this limit is reached. If $\mathcal{O}_{m,1}$ is found to be infeasible or no solution is found within the prescribed time limit, the heuristic terminates in failure (step 9). Across the experiments conducted in §6, our rolling horizon heuristic fails 3.7% (with a standard deviation of 1.4%) of the time, on average. The objectives of $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$, denoted $Z_{\mathcal{O}_{m,1}}$ and $Z_{\mathcal{O}_{m,2}}$, are defined in Equations (20) and (21):

$$Z_{\mathcal{O}_{m,1}} = \min[\eta_m(\tilde{\mathbf{x}}_m, h_1) + \eta_m(\tilde{\mathbf{x}}_m, h_2)]; \tag{20}$$

$$Z_{\mathcal{O}_{m,2}} = \max[\rho_m(\tilde{\mathbf{x}}_m, h_1) + \rho_m(\tilde{\mathbf{x}}_m, h_2)]; \tag{21}$$

$$\begin{aligned}\eta_m(\tilde{\mathbf{x}}_m, h) &= \sum_{l\in\mathcal{L}}\sum_{q\in\mathcal{Q}} \frac{1}{\Delta^+_q}\max[0, \nu^m_{l,q}(\tilde{\mathbf{x}}_m, h) - U^{m,h}_{l,q}\tau^m_l(\tilde{\mathbf{x}}_m, h)] \\ &\quad + \sum_{l\in\mathcal{L}}\sum_{q\in\mathcal{Q}} \frac{1}{\Delta^+_q}\max[0, L^{m,h}_{l,q}\tau^m_l(\tilde{\mathbf{x}}_m, h) - \nu^m_{l,q}(\tilde{\mathbf{x}}_m, h)];\end{aligned} \tag{22}$$

$$\rho_m(\tilde{\mathbf{x}}_m, t) = \sum_{s\in\mathcal{S}}\left[\sum_{\delta\in\Delta_m}\tilde{x}^{m,h}_{s,\delta} - \sum_{\theta\in\Theta_m}\tilde{x}^{m,h}_{s,\theta} + (1 - 2\Phi^m_\omega)\sum_{\lambda\in\Lambda_m}\tilde{x}^{m,h}_{s,\lambda}\right]; \tag{23}$$

$$\begin{aligned}\tau^m_l(\tilde{\mathbf{x}}_m, h) &= \sum_{\lambda\in\Lambda_m} S^{t=h_1}_{\lambda,l} Y^{m,\omega}_l \tilde{x}^{m,h}_{\lambda,\omega} + \sum_{\theta\in\Theta_m} S^{t=h_1}_{\theta,l}\tilde{x}^{m,h}_{\theta,\kappa} \\ &\quad + \sum_{b\in\mathcal{B}} S^{m,hi}_{b,l}\tilde{x}^{m,h}_{b,\kappa} + S^{m,lo}_{b,l} Y^{m,\omega}_l \tilde{x}^{m,h}_{b,\omega} \qquad \forall h \in \{h_1, h_2\}, l \in \mathcal{L};\end{aligned} \tag{24}$$

$$\begin{aligned}\nu^m_{l,q}(\tilde{\mathbf{x}}_m, h) &= \sum_{\lambda\in\Lambda_m} S^{t=h_1}_{\lambda,l} G^{t=h_1}_{\lambda,l,q} R^{m,\omega}_{l,q}\tilde{x}^{m,h}_{\lambda,\omega} \\ &\quad + \sum_{\theta\in\Theta_m} S^{t=h_1}_{\theta,l} G^{t=h_1}_{\theta,l,q}\tilde{x}^{m,h}_{\theta,\kappa} + \sum_{b\in\mathcal{B}} S^{m,hi}_{b,l} G^{m,hi}_{b,l,q}\tilde{x}^h_{b,\kappa} \\ &\quad + S^{m,lo}_{b,l} G^{m,lo}_{b,l,q} R^{m,\omega}_{l,q}\tilde{x}^h_{b,\omega} \qquad \forall h \in \{h_1, h_2\}, l \in \mathcal{L}, q \in \mathcal{Q}.\end{aligned} \tag{25}$$

The objective of $\mathcal{O}_{m,2}$, denoted by $Z_{\mathcal{O}_{m,2}}$, acts to maximise the productivity of mine m across periods h_1 and h_2, denoted by $\rho_m(\tilde{\mathbf{x}}_\mathrm{m}, h_1) + \rho_m(\tilde{\mathbf{x}}_\mathrm{m}, h_2)$ in Equation (21), given hard constraints on the percentage of each attribute $q \in \mathcal{Q}$ in the ore of each granularity $l \in \mathcal{L}$ produced at m in each period (steps 10 and 11). Let $\tilde{Q}^{m,h}_{l,q}$ denote the percentage of q in granularity l produced in period h by m in the solution found to $\mathcal{O}_{m,1}$. The $\mathcal{O}_{m,2}$ MIP applies the constraints of $\mathcal{O}_{m,1}$, with the addition of constraints (26) and (27) (step 10). With these additional constraints, $\mathcal{O}_{m,2}$ ensures that the solution found to $\mathcal{O}_{m,1}$ remains a solution of $\mathcal{O}_{m,2}$, and that deviation between the composition of production and desired bounds is limited to that present in the solution to $\mathcal{O}_{m,1}$:

$$\nu^m_{l,q}(\tilde{\mathbf{x}}_\mathbf{m}, h) \geq \min[\tilde{Q}^{m,h}_{l,q}, L^{m,h}_{l,q}]\tau^m_l(\tilde{\mathbf{x}}_\mathrm{m}, h) \quad \forall\, h \in \{h_1, h_2\},\ l \in \mathcal{L},\ q \in \mathcal{Q}; \tag{26}$$

$$\nu^m_{l,q}(\tilde{\mathbf{x}}_\mathrm{m}, h) \leq \max[\tilde{Q}^{m,h}_{l,q}, U^{m,h}_{l,q}]\tau^m_l(\tilde{\mathbf{x}}_\mathrm{m}, h) \quad \forall\, h \in \{h_1, h_2\},\ l \in \mathcal{L},\ q \in \mathcal{Q}. \tag{27}$$

Example 5.6 (Example 5.5 Continued). Consider the set of aggregates, $\mathcal{B}^A_{m'}$, in Example 5.4. Solving $\mathcal{O}_{m',1}$, with respect to the bounds of Example 5.5, leads to the following instantiation of $\tilde{\mathbf{x}}_{\mathrm{m}'}$:

$$\tilde{\mathbf{x}}_{\mathrm{m}'} = \{\tilde{x}^{m',h_1}_{a_0,\kappa} = 20 \text{ kt},\ \tilde{x}^{m',h_2}_{a_1,\kappa} = 20 \text{ kt},\ \tilde{x}^{m',h_2}_{a_2,\kappa} = 10 \text{ kt},\ {}^{m',h_2}_{a_6,\kappa} = 10 \text{ kt}\}.$$

In this solution to $\mathcal{O}_{m',1}$, $\tilde{Q}^{m',h_1}_{l,q} = \tilde{Q}^{m',h_2}_{l,q} = 62.75\%$. Constraints (26) and (27) in $\mathcal{O}_{m',2}$ are instantiated as shown in Equations (28) and (29):

$$0.627\tau^{m'}_l(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_1) \leq \nu^{m'}_{l,q}(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_1) \leq 0.629\tau^{m'}_l(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_1); \tag{28}$$

$$0.6255\tau^{m'}_l(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_2) \leq \nu^{m'}_{l,q}(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_2) \leq 0.6275\tau^{m'}_l(\tilde{\mathbf{x}}_{\mathrm{m}'}, h_2). \tag{29}$$

The solution to problem $\mathcal{O}_{m',2}$ extracts and processes the same high-grade material in periods h_1 and h_2 as does that of $\mathcal{O}_{m',1}$. To maximise productivity, waste aggregate a_7 is extracted in h_2;

$$\tilde{\mathbf{x}}_{\mathrm{m}'} = \{\tilde{x}^{m',h_1}_{a_0,\kappa} = 20 \text{ kt},\ \tilde{x}^{m',h_2}_{a_1,\kappa} = 20 \text{ kt},\ \tilde{x}^{m',h_2}_{a_2,\kappa} = 10 \text{ kt},\ \tilde{x}^{m',h_2}_{a_6,\kappa} = 10 \text{ kt},\ \tilde{x}^{m',h_2}_{a_7,\delta} = 20 \text{ kt}\}.$$

For each $s \in \mathcal{S}$ and $d \in \mathcal{D}_m$, we fix $\tilde{x}^{m,t=1}_{s,d}$ to the value of $\tilde{x}^{m,h_1}_{s,d}$ in the solution found to $\mathcal{O}_{m,2}$ (step 12). If no such solution is found in the time limit allocated to $\mathcal{O}_{m,2}$, each $\tilde{x}^{m,t=1}_{s,d}$ is fixed to the value of $\tilde{x}^{m,h_1}_{s,d}$ in the solution found to $\mathcal{O}_{m,1}$. We set $h_1 = 2$ and $h_2 = \{3, 4, \ldots, T\}$ (steps 15 and 16) and reconstruct $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$. Each $T^{t=1}_k(b)$, for $k = \{hi, lo, w\}$ and $b \in \mathcal{B}$, appearing in the constraints of $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$ is replaced with $T^{t=2}_k(b)$ (computed in step 13). Constants $T^{t=1}(s)$, $S^{m,t=1}_{s,l}$, and $G^{m,t=1}_{s,l,q}$, for each stockpile $s \in \Theta_m \cup \Lambda_m$, granularity $l \in \mathcal{L}$, and attribute $q \in \mathcal{Q}$, are replaced with $T^{t=2}(s)$, $S^{t=2}_{s,l}$, and $G^{t=2}_{s,l,q}$ (computed in step 14). The computation of updated block tonnages and stockpile compositions is straightforward, and thus explicit equations are omitted from this paper.

Example 5.7 (Example 5.6 Continued). We fix the mining and processing of aggregate a_0 in period $t = 1$ of schedule $\vec{s}^{\,i=1}_{m',j=1}$ and set $T^{t=2}_{hi}(a_0) = 0$. Periods h_1 and h_2 are redefined ($h_1 = 2$ and $h_2 = \{3\}$) and problems $\mathcal{O}_{m',1}$ and $\mathcal{O}_{m',2}$ solved to complete schedule $\vec{s}^{\,i=1}_{m',j=1}$:

$$\vec{s}^{\,i=1}_{m',j=1} = \begin{cases} \tilde{x}^{m,t=1}_{a_0,\kappa} = 20 \text{ kt},\ \tilde{x}^{m,t=2}_{a_1,\kappa} = 17 \text{ kt},\ \tilde{x}^{m,t=3}_{a_1,\kappa} = 3 \text{ kt}, \\ \tilde{x}^{m,t=2}_{a_2,\kappa} = 3 \text{ kt},\ \tilde{x}^{m,t=3}_{a_2,\kappa} = 7 \text{ kt}, \\ \tilde{x}^{m,t=2}_{a_7,\delta} = 10 \text{ kt},\ \tilde{x}^{m,t=3}_{a_6,\kappa} = 10 \text{ kt}, \\ \tilde{x}^{m,t=3}_{a_7,\delta} = 10 \text{ kt}. \end{cases}$$

5.3. The $\mathcal{O}_\Pi$ Problem

In each iteration i, $\mathcal{O}_\Pi$ is given a set of (up to) N block extraction schedules for each mine $m \in \mathcal{M}$, denoted by Ω^i_m. One schedule in the set $\Omega^i_m \cup \vec{s}_{\mathrm{best},m}$ is selected by $\mathcal{O}_\Pi$ to be enacted, where $\vec{s}_{\mathrm{best},m} \in \vec{s}_\mathrm{best}$ denotes the schedule to be implemented at mine m in the best solution found by the algorithm, $\vec{s}_\mathrm{best}$, across all iterations prior to i. The jth schedule available for selection at mine m is denoted by $\vec{s}_{m,j} \in \Omega^i_m \cup \{\vec{s}_{\mathrm{best},m}\}$. Ore railed from each mine m to each port π must originate from only one $\vec{s}_{m,j}$. Let $v^{m,t}_{l,q}(\vec{s}_{m,j})$ denote the percentage of attribute $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ produced by mine m in period t of schedule $\vec{s}_{m,j}$; let $\rho_m(\vec{s}_{m,j}, t)$ denote the productivity of m in period t of schedule $\vec{s}_{m,j}$, as per Equation (3); let P^m_j denote the total productivity achieved at m by $\vec{s}_{m,j}$ (the sum of $\rho_m(\vec{s}_{m,j}, t)$ over all t); let $\tau^m_l(\vec{s}_{m,j}, t)$ denote the tons of $l \in \mathcal{L}$ produced by mine m in period t of $\vec{s}_{m,j}$; and let $N_m = |\Omega^i_m \cup \{\vec{s}_{\mathrm{best},m}\}|$.

We now present an MIP formulation of the $\mathcal{O}_\Pi$ subproblem. In contrast to the instantiation of $\mathcal{O}_\Pi$ presented by Blom et al. (2014), a continuous variable $r^{m,l,j,t'}_{\pi,n,t}$ denotes the number of trainloads of granularity $l \in \mathcal{L}$ produced by mine m in period t' of schedule $\vec{s}_{m,j} \in \Omega^i_m \cup \{\vec{s}_{\mathrm{best},m}\}$ contributing to product $n \in N_{\pi,l}$ at port π in period t. Variable $s_{m,j}$ is assigned a value of 1 if and only if schedule $\vec{s}_{m,j}$ is chosen for implementation at mine m, and 0 otherwise. The objective of the port-side MIP is twofold. As a first priority, deviation between the composition of port products and desired bounds, denoted by $[L^{\pi,l,t}_{n,q}, U^{\pi,l,t}_{n,q}]$ for product $n \in N_{\pi,l}$ of granularity $l \in \mathcal{L}$ at port π in period t, is minimised. The total productivity achieved across the network of mines is maximised, over the scheduling horizon, as a second priority. The objective of the port-side MIP, $Z_{\mathcal{O}_\Pi}$, is defined in Equation (30), where β_1 and β_2 are constants such that $\beta_1 \gg \beta_2$; $f(t) = \max(1, t-1)$ and $g(t) = \min(T, t+1)$ for $t \in \{1, 2, \ldots, T\}$; $\vec{r}'$ denotes the set of all $r^{m,l,j,t'}_{\pi,n,t}$ variables for $m \in \mathcal{M}$, $l \in \mathcal{L}$, $j \in \{1, N_m\}$, $\pi \in \Pi$, $n \in N_{\pi,l}$,

$t \in \{0, 1, \ldots, T\}$, and $t' \in \{f(t), t\}$; and $\eta(\vec{r}', t)$ denotes the total deviation present between port products and desired bounds in period t, as defined in Equation (31).

The domains of variables $r^{m,l,j,t'}_{\pi,n,t}$ and $s_{m,j}$ are respectively restricted to nonnegative reals and binary values:

$$Z_{\mathcal{O}_\pi} = \min \sum_{t=1}^{T} \left[\beta_1 \eta(\vec{r}', t) - \beta_2 \sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} s_{m,j} P^m_i \right]; \tag{30}$$

$$\begin{aligned} \eta(\vec{r}', t) = \sum_{\substack{\pi \in \Pi,\, l \in \mathcal{L} \\ n \in N_{\pi,l} \\ q \in \mathcal{Q}}} \frac{T_R}{\Delta^+_q} \max \Bigg[0, &\sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} \sum_{t'=f(t)}^{t} r^{m,l,j,t'}_{\pi,n,t} v^{m,t'}_{l,q}(\vec{s}_{m,j}) \\ &- U^{\pi,l,t}_{n,q} \sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} \sum_{t'=f(t)}^{t} r^{m,l,j,t'}_{\pi,n,t} \Bigg] \\ + \sum_{\substack{\pi \in \Pi,\, l \in \mathcal{L} \\ n \in N_{\pi,l} \\ q \in \mathcal{Q}}} \frac{T_R}{\Delta^+_q} \max \Bigg[0, &L^{\pi,l,t}_{n,q} \sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} \sum_{t'=f(t)}^{t} r^{m,l,j,t'}_{\pi,n,t} \\ &- \sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} \sum_{t'=f(t)}^{t} r^{m,l,j,t'}_{\pi,n,t} v^{m,t'}_{l,q}(\vec{s}_{m,j}) \Bigg]. \end{aligned} \tag{31}$$

Constraint (32) ensures that only one schedule is selected to be implemented at each mine m. Port capacities are enforced by constraint (33). Constraint (34) ensures that all ore produced by each mine m in each period t is railed to a port by period $t+1$. We assume that the stockyards at each mine are empty at the start of $t = 1$ and that they must be emptied by the end of $t = T$:

$$\sum_{j=1}^{N_m} s_{m,j} = 1 \quad \forall\, m \in \mathcal{M}; \tag{32}$$

$$\sum_{l \in \mathcal{L}} \sum_{n \in N_{\pi,l}} \sum_{m \in \mathcal{M}} \sum_{j=1}^{N_m} \sum_{t'=f(t)}^{t} T_R r^{m,l,j,t'}_{\pi,n,t} \le C^t_\pi \quad \forall\, \pi \in \Pi,\ t \in \{1, 2, \ldots, T\}; \tag{33}$$

$$\sum_{\pi \in \Pi} \sum_{n \in N_{\pi,l}} \sum_{t=t'}^{g(t')} T_R r^{m,l,j,t'}_{\pi,n,t} = s_{m,j} \tau^m_l(\vec{s}_{m,j}, t') \quad \forall\, m \in \mathcal{M},\ l \in \mathcal{L},\ t' \in \{1, 2, \ldots, T\}. \tag{34}$$

5.4. Port to Mine Feedback

Feedback passed between $\mathcal{O}_\Pi$ and each $\mathcal{O}_m$ drives our algorithm toward a solution to the MTP-MMPP that minimises $Z_{\text{MTP-MMPP}}$ (Equation (1)). In each iteration i, $\mathcal{O}_\Pi$ provides each $\mathcal{O}_m$ with feedback in the form of a grade and quality target $\vec{\phi}^{i+1}_m$ and a vector of standard deviations $\vec{\sigma}^{i+1}_m$ to be used as its input in iteration $i+1$. We investigate three methods for the generation of these new targets and standard deviation vectors, one of which (F3) extends the mechanism of Blom et al. (2014) to the multiple-period setting. We evaluate each of these methods in §6.

5.4.1. Method 1 (F1). Equations (35) and (36) present the first of these methods. Equation (35) states that if $\mathcal{O}_\Pi$ does not find a solution better than $\vec{s}_{\text{best}}$, in iteration i, the grade and quality target provided to each mine m does not change, $\vec{\phi}^{i+1}_m = \vec{\phi}^i_m$. Each standard deviation in $\vec{\sigma}^i_m$, however, is reduced by a predetermined factor γ, where $0 < \gamma < 1$. The intuition behind F1 is that there may be a target in the vicinity of $\vec{\phi}^i_m$ that, if produced by m, will allow $\mathcal{O}_\Pi$ to improve upon $\vec{s}_{\text{best}}$. If such schedules were not proposed in iteration i, it is possible that each $\mathcal{O}_m$ has focused on achieving too large a spread in the composition of produced ore about $\vec{\phi}^i_m$. Reducing $\vec{\sigma}^i_m$ encourages each $\mathcal{O}_m$ to generate schedules across which ore compositions are more tightly clustered around $\vec{\phi}^{i+1}_m$.

$$\text{F1:} \quad Z_{\text{MTP-MMPP}}(\vec{s}_i) \ge Z_{\text{MTP-MMPP}}(\vec{s}_{\text{best}}) \rightarrow \vec{\phi}^{i+1}_m = \vec{\phi}^i_m \wedge \vec{\sigma}^{i+1}_m = \max(\vec{\sigma}^-, \gamma \vec{\sigma}^i_m) \quad \forall\, m \in \mathcal{M}. \tag{35}$$

Equation (36) states that if $\mathcal{O}_\Pi$ finds a solution $\vec{s}_i$ that is better than the current best, $\vec{s}_{\text{best}}$, in iteration i, the grade and quality target given to mine m in iteration $i+1$ is equal to the composition of the ore produced by m in $\vec{s}_i$, where $v^{m,t}_{l,q}(\vec{s}_i)$ denotes the percentage of $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ produced by m in period t of schedule $\vec{s}_i$:

$$\text{F1:} \quad Z_{\text{MTP-MMPP}}(\vec{s}_i) < Z_{\text{MTP-MMPP}}(\vec{s}_{\text{best}}) \rightarrow \sigma^{m,t,i+1}_{l,q} = \sigma^{m,t,i}_{l,q} \wedge \phi^{m,t,i+1}_{l,q} = v^{m,t}_{l,q}(\vec{s}_i) \quad \forall\, m \in \mathcal{M},\ t \in \{1, 2, \ldots, T\},\ l \in \mathcal{L},\ q \in \mathcal{Q}. \tag{36}$$

5.4.2. Method 2 (F2). We generalise F1 to form a second feedback method, F2, in which standard deviations are reduced, by γ, only after N_f successive iterations have been performed in which $\mathcal{O}_\Pi$ was unable to improve upon $\vec{s}_{\text{best}}$, where $N_f \ge 1$. If $N_f = 1$, F2 reduces to F1.

5.4.3. Method 3 (F3). Equations (37) and (38), together with Equation (35), form our third feedback method, F3. If $\mathcal{O}_\Pi$ finds a solution, $\vec{s}_i$, that is better than $\vec{s}_{\text{best}}$, we consider each $\sigma^{m,t,i}_{l,q}$ in $\vec{\sigma}^i_m$ and increase it by a predetermined factor γ, $0 < \gamma < 1$, if $v^{m,t}_{l,q}(\vec{s}_i)$ is sufficiently distant from the given target $\phi^{m,t,i}_{l,q}$ ($|v^{m,t}_{l,q}(\vec{s}_i) - \phi^{m,t,i}_{l,q}| > \sigma^{m,t,i}_{l,q}$). The intuition is that prior reductions in the size of $\sigma^{m,t,i}_{l,q}$ may have been premature. Increasing $\sigma^{m,t,i}_{l,q}$, for iteration $i+1$ will encourage $\mathcal{O}_m$ to form schedules, denoted by $\vec{s}^{i+1}_{m,j}$ for $j = 0, 1, \ldots, N$, across which there is more diversity in $v^{m,t}_{l,q}(\vec{s}^{i+1}_{m,j})$:

$$\text{F3:} \quad Z_{\text{MTP-MMPP}}(\vec{s}_i) < Z_{\text{MTP-MMPP}}(\vec{s}_{\text{best}}) \wedge |v^{m,t}_{l,q}(\vec{s}_i) - \phi^{m,t,i}_{l,q}| \le \sigma^{m,t,i}_{l,q} \rightarrow \sigma^{m,t,i+1}_{l,q} = \sigma^{m,t,i}_{l,q} \quad \forall\, m \in \mathcal{M},\ t \in \{1, 2, \ldots, T\},\ l \in \mathcal{L},\ q \in \mathcal{Q}; \tag{37}$$

$$\text{F3:}\quad Z_{\text{MTP–MMPP}}(\vec{s}_i) < Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}}) \wedge |v^{m,t}_{l,q}(\vec{s}_i) - \phi^{m,t,i}_{l,q}| > \sigma^{m,t,i}_{l,q} \rightarrow \sigma^{m,t,i+1}_{l,q} = \min(\sigma^{+}_{l,q}, \gamma^{-1}\sigma^{m,t,i}_{l,q}) \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, l \in \mathcal{L}, q \in \mathcal{Q}. \tag{38}$$

6. Computational Results

To evaluate our decomposition-based algorithm, for the MTP-MMPP, we consider a currently operating system of eight mines, connected to two ports, producing over 200 million tons of ore annually. Our planning horizon spans 13 weeks, divided into weekly time periods. We construct a set of test cases, each test characterising each mine in terms of a set of grade blocks sufficient to supply the mine's processing plants for two to three weeks, a set of blast blocks sufficient to supply these plants for at least four months, the composition and tonnage of each block and stockpile, the mining precedences that exist between blocks, and all relevant capacities. Capacities at each mine and port and bounds on the composition of port products are constant across the set of test cases. Varied across test instances are the set and number of grade and blast blocks available for extraction at each mine. In each test, each port forms one product of each granularity ($|N_{\pi,l}| = 1$ for all $\pi \in \Pi$ and $l \in \mathcal{L}$).

The number of blocks available for scheduling at each mine ranges from 102 to 437 across the test suite. The total number of blocks in the network ranges from 1,967 to 2,095. Test cases were formed using data provided by an industry partner, following the approach used by Blom et al. (2014) in the single-time-period setting. All experiments were run on a machine with 12 Intel(R) Xeon(R) E5-2440 central processing units and 64 GB random access memory, and were afforded 12 hours of (wall) clock time to complete. All MIPs were solved with CPLEX 12.6.

We first consider whether the MINLP of §4 can be solved directly, or if good solutions can be discovered by solving an MIP relaxation. For each test, we find a lower bound on the value of $Z_{\text{MTP–MMPP}}$ (Equation (1)) by solving an MIP relaxation of the MINLP. This relaxation is formed by replacing each bilinear term in constraints (16)–(17) and (18)–(19) with its McCormick (1976) envelope. We use this lower bound to evaluate the quality of solutions found by our decomposition-based algorithm. Table 1 states, for each test, the time required (in seconds) to find the first feasible solution of the MIP relaxation (alongside the gap between its objective value and the best known lower bound); the time required to solve the MIP to optimality; and the maximum deviation present, in the best solution obtained, between the actual (metal) grade of port products formed across the scheduling horizon and desired bounds. Constraints (16) and (17) track the composition of stockpiles at each mine, whereas constraints (18) and (19) define the grade of ore produced by each mine and transported to each port in each time period. Their relaxation leads to discrepancies between the actual grade of products formed at each mine and port and that inferred by the relaxed MIP. Because of the size of these discrepancies, it is not possible to solve the MIP relaxation with narrowed bounds on the composition of port products and obtain solutions in which these products are correctly blended. Blom et al. (2014) consider piecewise-linear relaxations (Gounaris et al. 2009) of the bilinear constraints present in the single-period MMPP to generate an MIP relaxation of greater fidelity. For each bilinear term, the domain of one variable is partitioned into several intervals and its value constrained to lie within one of these intervals. Given a 12-hour timeout, a solution to the relaxed single-period MMPP could not be found without significant deviations in port product compositions present.

Table 1 MIP Relaxation of the MTP-MMPP MINLP: Time Taken (s) to the First Feasible Solution (and Its Duality Gap) and to Solve to Optimality for Each Test

No.	Time to 1st (s)	Gap (%)	Time to opt. (s)	Max. deviation (metal grade, %)
1	9,539	0.61	23,904	2.12
2	6,672	0	6,672	1.40
3	5,077	0.09	6,807	1.27
4	3,087	0	3,087	1.29
5	—	—	—	—
6	2,693	0.28	2,751	1.67
7	4,520	0.11	4,899	1.79
8	14,198	0.07	27,177	1.89
9	6,261	0.29	9,020	1.67
10	5,181	0	5,181	2.52
11	—	—	—	—
12	8,846	0.26	—	1.76
13	8,825	0.10	9,929	1.62
14	13,306	0	13,306	1.49
15	13,427	0	13,427	1.51
16	11,818	0.38	16,364	1.79
17	29,833	0.04	—	1.15
18	6,415	0.35	6,569	1.71
19	24,917	0.42	34,968	1.69
20	39,761	0.39	—	1.47

Note. The maximum deviation in metal grade present (in a product formed at a port across the time horizon, from desired bounds) in the best found solution is stated. Dashes denote a failure to find an integer feasible or optimal solution within a 12-hour time limit.

Blom et al. (2014) demonstrate that their decomposition-based algorithm was able to find solutions to the single-period MMPP that were as good as or better than a range of alternative approaches commonly applied to problems with a pooling component, in orders of magnitude less time. The best performing of these approaches was the alternating heuristic (ALT) of Audet et al. (2004). The ALT heuristic fixes the

Table 2 **Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 3, 5$, and 7, $\gamma = 0.25$, and $M_A = 4$, for Each Test**

	$N=3$, $\gamma=0.25$, $M_A=4$				$N=5$, $\gamma=0.25$, $M_A=4$				$N=7$, $\gamma=0.25$, $M_A=4$			
	Time (s)		Gap to (%) MINLP_{lb}		Time (s)		Gap to (%) MINLP_{lb}		Time (s)		Gap to (%) MINLP_{lb}	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	5,317	747	2.36	1.21	6,261	1,434	1.87	0.48	9,273	1,179	1.74	1.00
2	**4,035**	**962**	**11.65**	**29.48**	6,502	1,263	1.12	0.49	8,994	1,247	0.88	0.21
3	5,739	906	1.72	0.91	7,089	1,450	0.96	0.62	10,843	1,772	0.89	0.58
4	5,572	981	1.54	0.52	6,601	1,108	1.12	0.6	9,792	1,380	1.03	0.54
5	**4,362**	**934**	**12.29**	**29.29**	7,912	1,098	0.88	0.27	10,030	1,712	0.77	0.43
6	4,948	945	1.48	0.74	6,068	1,320	1.08	0.54	9,815	2,201	0.4	0.27
7	4,062	321	2.37	1.08	6,350	1,323	0.94	0.75	8,736	2,469	0.71	0.46
8	5,309	1,471	2.65	1.98	6,858	1,245	2.09	1.32	9,103	1,862	1.5	1.08
9	5,029	1,270	2.1	1.18	6,132	956	0.99	0.61	10,145	2,067	0.57	0.23
10	4,607	1,541	3.18	1.43	6,770	1,285	1.35	0.62	11,736	1,806	0.77	0.29
11	**4,849**	**690**	**11.4**	**29.55**	6,792	1,362	0.65	0.27	9,715	1,203	0.59	0.55
12	**5,105**	**994**	**13.11**	**29.07**	6,822	1,156	1.89	1.98	11,006	2,826	1.27	0.76
13	4,872	1,056	2.77	1.31	7,474	1,368	1.2	0.81	9,108	1,208	1.32	0.48
14	**4,973**	**1,253**	**22.18**	**38.94**	8,078	1,588	1.64	0.76	9,851	2,076	0.96	0.58
15	**4,642**	**1,584**	**22.61**	**38.72**	7,522	809	1.48	0.53	8,952	2,155	0.93	0.44
16	4,538	653	1.44	0.26	7,208	729	1.11	0.53	9,222	2,094	0.85	0.40
17	**5,366**	**1,162**	**12.57**	**29.18**	8,637	1,475	1.43	0.69	9,205	2,035	1.44	0.56
18	4,165	558	2.36	1.12	7,543	1,820	1.22	0.83	9,335	1,054	1.28	0.77
19	**5,240**	**919**	**31.26**	**45.03**	8,203	1,221	1.34	0.4	10,734	1,307	0.74	0.34
20	**5,135**	**1,127**	**21.84**	**39.1**	6,893	574	0.93	0.57	9,490	1,749	0.77	0.58

Note. The elapsed time to solve (seconds) and the gap (percentage) between $Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$ and the best known lower bound are shown. An average (μ) and standard deviation (σ) for all quantities over 10 seeded runs are recorded. The bold entries indicate that one or more of the 10 runs of our algorithm on the associated test case did not result in correctly blended port products in one or more periods.

value of one variable in each bilinear term, producing an MIP that is solved to find values for the remaining bilinear variables. Fixing these alternate variables forms a second MIP that is solved to find new values for the first variable set. This process is repeated until a fixed point is reached. We have applied the ALT heuristic to the MINLP of §4, partitioning and alternately fixing its bilinear variables. Given a time limit of 12 hours and 2,000 seconds afforded to each MIP solve, the ALT heuristic could not find a solution, in any of our tests, in which port products were correctly blended.

We now consider the performance of our decomposition-based algorithm in varying settings. We first examine the impact of varying the N and γ parameters on the quality of solutions found by our algorithm (Tables 2 and 3). Recall that N denotes the number of schedules formed during the solving of each $\mathcal{O}_m$; γ denotes the degree to which the standard deviations given to each $\mathcal{O}_m$ as input are increased or decreased (a larger γ results in smaller changes) during feedback generation; and M_A denotes the maximum size of block aggregates formed during the generation of schedules by each $\mathcal{O}_m$. We next demonstrate the need to monitor evolving stockpile states by evaluating the difference between expected and achieved port product composition under the assumption of constant stockpile grades. The merits of our two-stage approach for the generation of single-mine schedules (see §5.1), relative to a single application of the rolling horizon heuristic of §5.2, are highlighted in Table 4. We then examine the relative performance of the algorithm when instantiated with each of the feedback methods of §5.4. Feedback method F3 is used in all other experiments. We conclude by examining the impact of splitting the scheduling horizon into more than two time periods in the application of our rolling horizon heuristic. In the remainder of this section, each average value is reported with an associated standard deviation following it in parentheses.

Table 2 records the results of our decomposition-based algorithm, averaged over 10 seeded runs, on each of our benchmark tests, with $N=3$, 5, and 7; $\gamma=0.25$; $M_A=4$; and feedback method F3 implemented. We record, for the best solution found by the algorithm, $\vec{s}_{\text{best}}$, the elapsed time to termination (seconds) and the gap (percentage) between $Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$ and its best known lower bound. Quantities have been averaged over 10 seeded runs, with the average (μ) and standard deviation (σ) recorded. Across all tests, our decomposition-based algorithm was able to find solutions in which plant capacities at each mine are fully utilised. For $N = 5$ and 7, port products are formed to specification in all solutions, in each period of the scheduling horizon. The bold entries in Table 2, for $N = 3$, indicate that one or more of the 10 runs of our algorithm on the associated test case did not

Table 3 **Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 5$, $M_A = 4$, and $\gamma = 0.25, 0.50$, and 0.75**

	$N=5,\ \gamma=0.25$				$N=5,\ \gamma=0.50$				$N=5,\ \gamma=0.75$			
	Time (s)		Gap to (%) MINLP_{lb}		Time (s)		Gap to (%) MINLP_{lb}		Time (s)		Gap to (%) MINLP_{lb}	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	6,261	1,434	1.87	0.48	7,208	1,185	1.72	0.71	10,633	1,874	1.04	0.42
2	6,502	1,263	1.12	0.49	6,923	1,287	0.63	0.2	10,877	1,763	0.4	0.22
3	7,089	1,450	0.96	0.62	8,355	1,580	0.84	0.38	12,465	1,531	0.54	0.29
4	6,601	1,108	1.12	0.6	7,802	1,288	0.7	0.27	11,021	1,367	0.59	0.35
5	7,912	1,098	0.88	0.27	7,887	1,612	0.74	0.25	10,929	1,628	0.4	0.22
6	6,068	1,320	1.08	0.54	7,194	1,322	0.47	0.25	10,656	1,375	0.32	0.29
7	6,350	1,323	0.94	0.75	7,222	915	0.63	0.52	10,471	1,095	0.5	0.26
8	6,858	1,245	2.09	1.32	8,162	1,933	1.01	0.51	12,533	3,146	0.8	0.38
9	6,132	956	0.99	0.61	7,548	1,503	0.65	0.5	13,753	1,194	0.44	0.23
10	6,770	1,285	1.35	0.62	7,697	799	1.02	0.56	13,433	1,318	0.77	0.31
11	6,792	1,362	0.65	0.27	7,614	935	0.8	0.67	14,366	1,550	0.28	0.17
12	6,822	1,156	1.89	1.98	7,876	1,684	1.17	0.52	13,039	1,877	0.8	0.57
13	7,474	1,368	1.2	0.81	7,837	1,407	1.1	0.42	13,097	2,188	1.16	0.66
14	8,078	1,588	1.64	0.76	8,309	1,174	0.77	0.3	12,907	1,973	0.84	0.76
15	7,522	809	1.48	0.53	7,734	1,253	1.25	0.6	12,977	1,302	0.75	0.71
16	7,208	729	1.11	0.53	7,550	369	0.67	0.35	13,609	991	0.44	0.25
17	8,637	1,475	1.43	0.69	7,921	1,126	1.11	0.73	14,782	2,124	0.87	0.47
18	7,543	1,820	1.22	0.83	7,251	983	0.99	0.23	12,672	1,247	0.84	0.42
19	8,203	1,221	1.34	0.4	8,286	1,093	1.36	0.47	13,635	1,337	0.88	0.47
20	6,893	574	0.93	0.57	7,075	995	0.62	0.36	11,976	1,485	0.56	0.29

Notes. Columns are defined as in Table 2. An average (μ) and standard deviation (σ) for all quantities over 10 seeded runs are recorded.

result in correctly blended port products, in one or more periods. Across these instances, small deviations in one contaminant were present, and the duality gaps of solutions found, across all tests, ranged between 0.6% and 100%. Increasing N from 3 to 5 increases solve times by 2,192 s (847 s) on average, and solution quality by 7.88% (9.11%). For $N = 5$, duality gaps range between 0.2% and 5.6%. Increasing N from 5 to 7 increases solve times by 2,668 s (1,071 s) on average, with a 0.29% (0.27%) average improvement in quality and duality gaps ranging between 0.1% and 3.5%. These results demonstrate that a value of N that is too small prevents the algorithm from sufficiently exploring the space of producible grades at each mine site before it terminates.

Reducing the gap between $Z_{\text{MTP–MMPP}}(\vec{s}_{\text{best}})$ and its lower bound by 1% can be achieved by mining on the order of 10 kt more waste at each mine across the horizon, or hauling 10 kt less material at each mine to stockpiles. The large optimality gaps present in some entries of Table 2, for $N = 3$, are the result of port product deviations, which are heavily penalised.

Table 3 records the results of our decomposition-based algorithm, averaged over 10 seeded runs, in each of our benchmark tests, with $N = 5$; $M_A = 4$; $\gamma = 0.25, 0.50,$ and 0.75; and feedback method F3 implemented. Increasing γ from 0.25 to 0.50 increases solve times by 587 s (589 s) on average, while increasing solution quality by only 0.35% (0.30%). For $\gamma =$ 0.50, duality gaps range between 0% and 3%. Increasing γ from 0.50 to 0.75 increases solve times and solution quality by an average of 4,819 s (1,158) and 0.25% (0.19%), respectively, with a range in duality gaps of 0.1% to 3%. A larger γ results in smaller changes to standard deviation parameters at each iteration, and as a result, more iterations are performed prior to termination of the algorithm. Smaller values of γ ($\gamma =$ 0.25) are sufficient, however, for consistently generating solutions of reasonable quality.

The impact of varying N and γ on the quality of solutions found by our algorithm in the multiple-time-period setting is consistent with the results of the same experiment performed in the single-period case (Blom et al. 2014). In both settings, increasing both N and γ improves, in general, the quality of solutions found by the algorithm, but at a cost of longer solve times.

In our modelling of the MTP-MMPP, we keep track of the composition of stockpiles at each mine, across the scheduling horizon. We have used a modified version of our algorithm (with $N = 5$, $\gamma = 0.50$, and $M_A = 4$), in which step 14 of our rolling horizon heuristic (Algorithm 3) does not update the composition of stockpiles, to solve each of our tests. For each test, and each solution obtained from 10 differently seeded runs of our algorithm, we compute the actual state of each stockpile at each mine in each period and the actual deviation that exists (if any) between the composition of port products formed in each time period (based on actual stockpile states) and desired bounds. We find that although the average deviations reported are small, across the set of relevant quality attributes, the maximum deviations experienced, over all obtained solutions, are significant. The average deviation in the metal grade of port products (from desired bounds), across all time periods and tests, is 0.01% (0.03%), with a maximum experienced deviation of 0.67%. The distance between the upper and lower bounds on metal grade, in each port product, is 1%. These results demonstrate the importance of monitoring the state of stockpiles over time.

We have calculated, via experiment, the variance in the grade of ore produced by each mine and port, providing an indication of how grade varies between reclaimed slices of stockpiles. We consider the schedules produced for each mine, across our test suite, in the first iteration of our decomposition-based algorithm. We calculate the average variance in each quality attribute, across each schedule formed, in 10 seeded runs of the algorithm with $N = 5$, $M_A = 4$, and $\gamma = 0.25$. The average variance in metal content over the scheduling horizon was 0.23 (0.12), indicating that samples of ore produced at each mine will be distributed around the average with a standard deviation of 0.48%. As ore arrives at each port, it is blended onto stockpiles and reclaimed onto ships. Given a 1:5 blending effect arising from stacking and reclaiming at the ports, the variance in metal content of shipped ore is, on average, 0.25 (0.03). Samples of the shipped ore will be distributed about the average grade with a standard deviation of 0.50%. Such variability in blended products of this type lies within expectations (Everett et al. 2002, Minnitt and Pitard 2008).

To generate a single schedule, $\vec{s}_m$, for a mine m, $\mathcal{O}_m$ applies the two-stage process described in §5.1. First, the set of available grade and blast blocks are aggregated to form a smaller number of larger units. The rolling horizon heuristic of §5.2 determines how these aggregates are to be mined over the course of the scheduling horizon. The original set of blocks is culled by removing all blocks that do not appear in an aggregate mined in the resulting schedule. A second application of this heuristic, in which only blocks in this restricted set and the set of grade blocks can be mined, generates our schedule, $\vec{s}_m$. The intuition is that this two-stage process is likely to be less time consuming than a single application of the rolling horizon heuristic to the unrestricted, original block set. Table 4 records the results of our algorithm with $N = 5$, $\gamma = 0.25$, $M_A = 4$, and each $\mathcal{O}_m$ instructed to generate schedules with only one application of our rolling horizon heuristic (one stage). In this setting, the duality gaps of solutions found, across all tests, range between 0.1% and 5.6%. For comparison,

Table 4 **Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 5$, $\gamma = 0.25$, $M_A = 4$, and a Single Application of the Rolling Horizon Heuristic of §5.2 (One Stage) Used by Each $\mathcal{O}_m$ to Generate Schedules**

	One stage				Two stage			
	Time (s)		Gap to MINLP$_{\text{lb}}$ (%)		Time (s)		Gap to MINLP$_{\text{lb}}$ (%)	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	9,629	1,991	1.93	1.23	6,261	1,434	1.87	0.48
2	10,147	3,300	1.52	0.97	6,502	1,263	1.12	0.49
3	8,859	1,142	1.43	0.78	7,089	1,450	0.96	0.62
4	9,201	2,157	1.07	0.46	6,601	1,108	1.12	0.6
5	8,743	1,975	1.15	0.93	7,912	1,098	0.88	0.27
6	8,480	1,876	0.97	0.49	6,068	1,320	1.08	0.54
7	7,882	1,776	1.54	1.04	6,350	1,323	0.94	0.75
8	9,509	3,231	1.95	1.01	6,858	1,245	2.09	1.32
9	9,520	1,892	1.21	0.55	6,132	956	0.99	0.61
10	7,507	935	0.9	0.62	6,770	1,285	1.35	0.62
11	7,430	1,252	1.08	0.4	6,792	1,362	0.65	0.27
12	8,363	1,647	2.23	1.3	6,822	1,156	1.89	1.98
13	9,110	2,084	1.91	1.33	7,474	1,368	1.2	0.81
14	7,480	2,033	0.92	0.47	8,078	1,588	1.64	0.76
15	8,908	3,599	2.28	1.09	7,522	809	1.48	0.53
16	8,034	1,548	1.18	1.01	7,208	729	1.11	0.53
17	7,426	2,196	2.25	1.49	8,637	1,475	1.43	0.69
18	8,660	1,470	1.52	0.62	7,543	1,820	1.22	0.83
19	8,293	1,567	1.46	0.99	8,203	1,221	1.34	0.4
20	8,647	1,969	0.8	0.37	6,893	574	0.93	0.57

Notes. Results obtained using two applications of this heuristic (two stage, from Table 2) are listed for comparison. Columns are defined as in Table 2. An average (μ) and standard deviation (σ) for all quantities over 10 seeded runs are recorded.

Table 4 replicates the results of Table 2 for $N = 5$, $\gamma = 0.25$, and $M_A = 4$, where two applications of the rolling horizon heuristic were completed (two stage), and duality gaps ranged between 0.2% and 5.6%. In the one-stage setting, our algorithm solves, on average, 1,505 s (1,284 s) slower, while producing solutions that are 0.2% (0.40%) further from lower bounds.

We examine the performance of our decomposition-based algorithm for different values of maximum aggregate size M_A. Table 5 reports the results of our algorithm given $N = 5$, $\gamma = 0.25$, and $M_A = 2, 4$, and 8. For $M_A = 2$, solution duality gaps range between 0.2% and 9.3%. Increasing M_A from 2 to 4 reduces solve times by 2,491 s (1,298 s), on average, with no significant change in average solution quality. The maximum duality gap observed, however, reduces to 5.6%. Increasing M_A from 4 to 8 reduces solve times by 248 s (760 s), with an average reduction in solution quality of 0.42% (0.68%). Duality gaps range between 0.2% and 9.9%. Reducing M_A allows our algorithm to be more selective in the blocks it discards from consideration in the first scheduling stage, but results in a greater number of blocks available for scheduling in the second, increasing solve times.

Table 5 Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 5$, $\gamma = 0.25$, and $M_A = 2, 4$, and 8

	$M_A = 2$				$M_A = 4$				$M_A = 8$			
	Time (s)		Gap to (%) MINLP$_{\text{lb}}$		Time (s)		Gap to (%) MINLP$_{\text{lb}}$		Time (s)		Gap to (%) MINLP$_{\text{lb}}$	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	10,146	1,856	1.88	0.87	6,261	1,434	1.87	0.48	6,627	1,210	1.81	1.38
2	9,410	1,538	0.68	0.29	6,502	1,263	1.12	0.49	5,688	860	1.57	0.77
3	10,471	1,535	1.14	0.82	7,089	1,450	0.96	0.62	7,370	984	0.96	0.3
4	10,617	1,629	0.79	0.48	6,601	1,108	1.12	0.6	6,308	1,626	1.8	1.31
5	9,331	1,190	1.19	0.65	7,912	1,098	0.88	0.27	6,856	843	0.99	0.62
6	8,989	1,550	0.92	0.92	6,068	1,320	1.08	0.54	7,005	863	1.29	0.59
7	10,044	1,241	0.81	0.3	6,350	1,323	0.94	0.75	5,856	1,434	3.23	2.4
8	10,689	1,718	1.4	0.76	6,858	1,245	2.09	1.32	7,528	1,736	1.41	0.32
9	9,265	1,093	0.88	0.19	6,132	956	0.99	0.61	7426	1,508	1.19	0.88
10	7,765	2,514	0.94	0.41	6,770	1,285	1.35	0.62	7,436	1,246	2.1	1.36
11	9,646	1,485	1.07	0.65	6,792	1,362	0.65	0.27	6,429	665	1.41	0.53
12	11,062	2,295	2.26	2.55	6,822	1,156	1.89	1.98	6,873	1,159	1.4	0.72
13	10,482	2,025	1.02	0.38	7,474	1,368	1.2	0.81	6,414	1,123	2.58	1.07
14	9,992	1,238	1.33	0.56	8,078	1,588	1.64	0.76	7,134	1,692	2.62	1.45
15	9,632	2,011	1.67	0.76	7,522	809	1.48	0.53	7,188	1,743	2.6	2.65
16	9,283	1,707	1.23	0.62	7,208	729	1.11	0.53	7,072	901	1.28	0.73
17	8,897	2,105	1.4	0.88	8,637	1,475	1.43	0.69	7,031	1,398	1.95	1.04
18	8,620	1,547	1.96	1.24	7,543	1,820	1.22	0.83	7,023	915	1.11	0.32
19	7,783	2,317	0.98	0.69	8,203	1,221	1.34	0.4	7,507	1,417	1.33	0.9
20	9,411	2,153	1.05	0.77	6,893	574	0.93	0.57	5,975	7,15	1.12	0.55

Notes. Columns are defined as in Table 2. An average (μ) and standard deviation (σ) for all quantities over 10 seeded runs are recorded.

Table 6 reports the results of our algorithm when instantiated with feedback methods F1, F2 with $N_f = 2$, and F3, for $N = 5$, $\gamma = 0.25$, and $M_A = 4$. Each of these methods is described in §5.4. Using F1, solve times are 7,875 s (1,295 s) and 3,120 s (768 s) faster, on average, than when using F2 ($N_f = 2$) and F3, respectively. In five test cases, however, our algorithm (with F1 implemented) did not find solutions

Table 6 Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 5$, $\gamma = 0.25$, $M_A = 4$, with Feedback Methods F1, F2 with $N_f = 2$, and F3 Implemented

	F1				F2 with $N_f = 2$				F3			
	Time (s)		Gap to (%) MINLP$_{\text{lb}}$		Time (s)		Gap to (%) MINLP$_{\text{lb}}$		Time (s)		Gap to (%) MINLP$_{\text{lb}}$	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	**3,438**	**592**	**21.82**	**39.13**	11,721	2,362	1.37	0.72	6,261	1,434	1.87	0.48
2	3,786	522	1.14	0.54	11,354	1,443	0.65	0.4	6,502	1,263	1.12	0.49
3	4,702	709	1.04	0.33	12,442	2,905	1.01	0.72	7,089	1,450	0.96	0.62
4	4,409	853	1.54	0.89	11,013	1,255	0.89	0.51	6,601	1,108	1.12	0.6
5	**3,370**	**501**	**21.13**	**39.48**	12,129	1,996	1.03	0.91	7,912	1,098	0.88	0.27
6	3,449	457	1.71	0.6	13,038	2,454	0.59	0.26	6,068	1,320	1.08	0.54
7	3,652	606	1.64	0.96	12,244	2,299	0.72	0.51	6,350	1,323	0.94	0.75
8	4,219	874	2.25	1.31	14,611	2,647	0.85	0.45	6,858	1,245	2.09	1.32
9	4,195	665	1.08	0.32	12,918	2,371	0.6	0.27	6,132	956	0.99	0.61
10	4,150	720	1.48	0.77	13,332	2,464	0.85	0.42	6,770	1,285	1.35	0.62
11	3,987	500	1.22	0.79	12,862	2,454	0.6	0.28	6,792	1,362	0.65	0.27
12	**4,288**	**570**	**11.76**	**29.44**	12,767	2,250	1.58	0.98	6,822	1,156	1.89	1.98
13	**3,711**	**397**	**11.92**	**29.42**	11,773	1,869	1.23	0.6	7,474	1,368	1.2	0.81
14	3,955	580	2.15	1.08	11,643	2,216	0.73	0.38	8,078	1,588	1.64	0.76
15	3,779	796	2.33	1.77	9,018	1,854	0.98	0.43	7,522	809	1.48	0.53
16	4,114	500	1.15	0.29	10,518	1,700	0.56	0.14	7,208	729	1.11	0.53
17	**4,015**	**770**	**21.71**	**39.2**	10,860	1,954	0.83	0.51	8,637	1,475	1.43	0.69
18	4,158	879	1.6	1.17	12,146	1,178	0.63	0.25	7,543	1,820	1.22	0.83
19	4,282	688	2.53	1.24	10,686	1,909	1.02	0.65	8,203	1,221	1.34	0.4
20	3,673	685	1.89	1.15	9,756	1,875	0.64	0.33	6,893	574	0.93	0.57

Notes. Columns are defined as in Table 2. An average (μ) and standard deviation (σ) for all quantities, over 10 seeded runs, are recorded. Bold values indicate instances where our algorithm (with F1 implemented) did not find solutions in which port products were correctly blended.

in which port products were correctly blended (these instances are highlighted in bold in Table 6). The duality gaps of solutions, across all tests, range from 0.1% to 100%. Using F3 in place of F2 ($N_f = 2$) results in solve times that are 4,756 s (1,725 s) faster and solutions with duality gaps that are 0.78% (0.37%) higher, on average. Duality gaps range from 0% to 3.4% when F2 is implemented, and from 0.2% to 5.6% for F3. Although solution quality improves, in general, with method F2, it does so at a cost of longer solve times.

The rolling horizon heuristic that we use to generate mine-side schedules (§5.2) splits the horizon into two periods, solves a two-period problem to schedule the first period, and then repeats this process on the remaining periods in the horizon. Table 7 records the results of our algorithm when this heuristic splits the horizon into three periods. The results obtained using the original two-period split are also shown. In the three-period setting, our algorithm is 11,602 s (1,485 s) slower, on average, finding solutions with duality gaps that are, on average, 0.89% (0.57%) higher—ranging from 0.3% to 5.9%. In the two-period setting, duality gaps range between 0.2% and 5.6%. Recall that time limits are placed on the solving of all MIPs by the rolling horizon heuristic. Although dividing the horizon into three periods, in place of two, is a better approximation of the horizon, each three-period MIP is more time consuming to solve. The resulting solutions have higher duality gaps, with respect to the relevant objectives, than those found by solving two-period MIPs.

Table 7 **Best Solution $\vec{s}_{\text{best}}$ Found by Our Algorithm for $N = 5$, $\gamma = 0.25$, $M_A = 4$, F3 Implemented, and the Horizon Split into Three Periods When the Rolling Horizon Heuristic of §5.2 Is Applied**

	Two-period split				Three-period split			
	Time (s)		Gap to (%) MINLP$_{\text{lb}}$		Time (s)		Gap to (%) MINLP$_{\text{lb}}$	
No.	μ_T	σ_T	μ_G	σ_G	μ_T	σ_T	μ_G	σ_G
1	6,261	1,434	1.87	0.48	18,539	2,945	2.24	0.77
2	6,502	1,263	1.12	0.49	18,592	3,386	2.35	1.26
3	7,089	1,450	0.96	0.62	18,914	3,234	2.91	1.3
4	6,601	1,108	1.12	0.6	17,921	3,507	2.88	0.97
5	7,912	1,098	0.88	0.27	19,880	3,608	1.46	0.71
6	6,068	1,320	1.08	0.54	20,021	2,720	1.05	0.67
7	6,350	1,323	0.94	0.75	15,923	2,698	1.66	0.77
8	6,858	1,245	2.09	1.32	14,156	2,926	3.75	1.02
9	6,132	956	0.99	0.61	17,786	2,626	1.31	0.84
10	6,770	1,285	1.35	0.62	17,732	1,984	2.07	1.44
11	6792	1,362	0.65	0.27	17,572	2,273	1.43	0.8
12	6,822	1,156	1.89	1.98	18,411	2,813	2.68	0.97
13	7,474	1,368	1.2	0.81	20,308	2,065	2.56	0.99
14	8,078	1,588	1.64	0.76	18,966	4,134	1.65	0.61
15	7,522	809	1.48	0.53	20,937	2,317	2.56	1.49
16	7,208	729	1.11	0.53	19,270	2,164	1.51	0.81
17	8,637	1,475	1.43	0.69	22,140	4,634	2.83	1.4
18	7,543	1,820	1.22	0.83	19,740	4,821	1.75	0.52
19	8,203	1,221	1.34	0.4	18,328	3,082	2.74	0.96
20	6,893	574	0.93	0.57	18,618	3,690	1.62	0.65

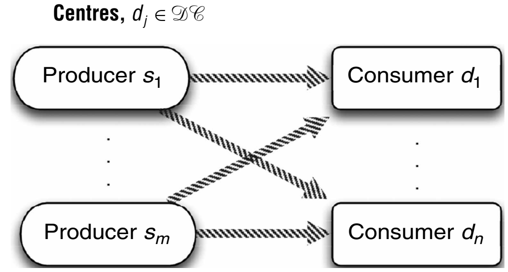
Notes. Columns are defined as in Table 2. An average (μ) and standard deviation (σ) for all quantities over 10 seeded runs are recorded.

7. Multiple Period, Multiple Producer/Consumer Production Planning

The MTP-MMPP is an instance of a general class of multiple-period, multiple producer/consumer production planning problems. Consider a network of independently operating production sites, $\mathcal{PS}$, and multiple distribution centres, $\mathcal{DC}$, as shown in Figure 3. Each production site $s \in \mathcal{PS}$ in each time period $t \in \mathcal{T}$ is able to produce a set of products, $\mathcal{P}_s$, each characterised by a vector of numerical attributes, $\vec{a}^{\,t}_{p,s}$, $p \in \mathcal{P}_s$. These attributes may include the quantity of p produced in period t and/or its quality. Each distribution centre $d \in \mathcal{DC}$, in each period $t \in \mathcal{T}$, combines products formed across the set of sites to form its own products, $\mathcal{P}_d$, similarly characterised by a vector of attributes, $\vec{a}^{\,t}_{p,d}$, $p \in \mathcal{P}_d$, to be shipped to external markets. Each of these products, in each period, is associated with deterministic demands on the values of its associated attributes (for example, on the quantity and quality of production). In the open-pit supply network, each production site is a mine, each distribution centre is a port, lump and fine products are formed at each mine and port, and the attributes of mine and port products are metal percentage and impurity levels.

Such problems can be decomposed into $|\mathcal{PS}+1|$ subproblems, one subproblem for each production site $s \in \mathcal{PS}$, denoted by $\mathcal{O}_s$, and one distribution subproblem involving the set of production sites and distribution centres, denoted by $\mathcal{O}_D$. Our decomposition-based algorithm, as defined in §5, can be readily applied to problems of this form. In the application of this algorithm, each $\mathcal{O}_s$ is designed to accept, as input, a vector of target attributes and an associated standard deviation for each attribute for each of the products it is capable of producing.

Figure 3 **A Network of Production Sites, $s_i \in \mathcal{PS}$, and Distribution Centres, $d_j \in \mathcal{DC}$**



A solution to subproblem $\mathcal{O}_s$ is a set of schedules for site s, each schedule producing a set of products in each period with varying attributes. The $\mathcal{O}_D$ subproblem accepts these schedule sets, selects one schedule to be implemented at each site, and solves a distribution problem in which products are transported between sites and centres. The attributes of the products formed at each centre are dependent on the attributes of those produced at each site. A feedback mechanism, such as those described in §5.4, must be defined to give each $\mathcal{O}_s$ a new vector of target attributes, and standard deviations, for each product $p \in \mathcal{P}_s$. The successive solving of each $\mathcal{O}_s$ and $\mathcal{O}_D$, in conjunction with the standard deviations supplied to each $\mathcal{O}_s$ reducing in size over the course of the algorithm, yields a sequence of monotonically improving solutions to the original multiperiod, multisite planning problem.

General production planning problems of this form can be found in a range of domains, including the natural resources sector, food production, and the chemical process industry. Although our model of the MTP-MMPP exhibits the structure of a pooling problem, our approach is not restricted to solving problems that involve pooling. We require only that the problem be decomposable into several optimisation problems for the upstream components of the supply chain and a single optimisation problem representing the downstream component. The application of our approach to problems in this general class is planned as future work.

8. Concluding Remarks

In this paper, we presented a decomposition-based algorithm for a challenging multiple mine, multiple-time-period, open-pit production scheduling problem. Nonlinear in nature, this problem can be formulated as a MINLP with on the order of 100,000 variables in realistic instances. In a simpler instantiation of this problem—the single-time-period MMPP—this decomposition-based algorithm was able to find solutions that were, in a majority of cases, higher in quality than those discovered by a range of alternative techniques (Blom et al. 2014). Where this was not the case, the algorithm was able to find good quality solutions for which the alternative methods required orders of magnitude more time to match. We have shown in this paper that this algorithm is able to scale to the significantly more complex multiple-time-period setting, generating solutions with optimality gaps within 6% of known lower bounds on the objective of the MINLP problem representation in less than two hours, on average. The best performing alternative approach considered by Blom et al. (2014) in solving the single-period MMPP could not scale to the multiple-period setting.

The extension of the algorithm to the multiple-time-period setting required the development of a new heuristic for the generation of multiple-time-period extraction schedules for each mine and a new MIP model to represent the port-side optimisation problem. The modelling of time significantly increases the complexity of the MMPP. The composition of stockpiles at each mine must be tracked over time to avoid errors in the evaluation of port product grades. Each decision made at each mine has an influence on the material that will be available for mining in the future, and in consequence the ability to satisfy constraints on plant use. These decisions determine the composition of ore produced by each site, which in turn influences the decisions of other sites to ensure the correct blending of products at each port—resulting in a problem with high combinatorial complexity.

The decomposition-based algorithm presented in this paper is, to the best of our knowledge, the first approach to solve an integrated multiple-period open-pit scheduling problem, across multiple mines and ports, where the grade and quality of the ore to be produced by each mine is not known a priori, but determined as part of the optimisation.

Acknowledgments

The authors thank the anonymous reviewers of this paper for their constructive feedback and comments. This work is funded by an Australian Research Council (ARC) Linkage grant [LP110100115] in conjunction with an industry partner.

Appendix A. Modelling Notation

$\mathcal{O}_m$, $\mathcal{O}_\Pi$ Mine-side and port-side subproblems

Sets and Indices

$m, \mathcal{M}$	Mines
$b, \mathcal{B}_m$	Blocks at mine m
$\mathcal{B}_m^b$	Blast blocks at mine m
$\mathcal{B}_m^g$	Grade blocks at mine m
$l, \mathcal{L}$	Granularities (lump/fines)
$q, \mathcal{Q}$	Quality attributes
δ, Δ_m	Waste dumps at mine m
θ, Θ_m	High-grade stockpiles at mine m
$s, \mathcal{S}_m$	Set of material sources at mine m (blocks and stockpiles)
κ, ω	Dry and wet processing plant
$\mathcal{A}_{m,b}^{\wedge}$	Set of blocks that must be removed before block $b \in \mathcal{B}_m$ can be accessed
$\mathcal{A}_{m,b}^{\vee}$	Set of blocks, one of which must be removed, before $b \in \mathcal{B}_m$ can be accessed
i	Iteration of the decomposition-based algorithm
$\vec{s}_i$	Solution to the MTP-MMPP found by our algorithm in iteration i
Ω_m^i	Set of extraction schedules generated by $\mathcal{O}_m$ in iteration i of the algorithm

$\vec{\phi}_m$, $\vec{\phi}_m^i$ Grade and quality target used by $\mathcal{O}_m$, in iteration i, to generate Ω_m^i
$\phi_{l,q}^{m,t,i}$ Grade/quality target in $\vec{\phi}_m^i$ corresponding to granularity $l \in \mathcal{L}$, attribute $q \in \mathcal{Q}$, and time period t
j, $\vec{s}_{m,j}^i$ The jth schedule in a set of schedules
$\mathcal{B}_m^A$ A set of aggregates of blocks in $\mathcal{B}_m$
$\bar{\mathcal{B}}_m^A$ Set of grade or blast block aggregates mined (partially or completely) in an extraction schedule, for mine m, formed during the solving of $\mathcal{O}_m$
π, Π Ports
t Time period
k, hi, lo, w Index k refers to high grade (hi), low grade (lo), or waste (w)
λ, Λ_m Low-grade stockpiles at mine m
d, $\mathcal{D}_m$ Set of material destinations at mine m (stockpiles and plants)
n, $N_{\pi,l}$ Set of products, of granularity $l \in \mathcal{L}$, formed by port π
$\vec{s}_{\text{best}}$ Best solution to the MTP-MMPP found by our algorithm, defines values of variables in the MINLP of §4
$\vec{\sigma}_m$, $\vec{\sigma}_m^i$ Set of standard deviations used by $\mathcal{O}_m$, in iteration i, to generate Ω_m^i
$\sigma_{l,q}^{m,t,i}$ Standard deviation in $\vec{\sigma}_m^i$ corresponding to granularity $l \in \mathcal{L}$, attribute $q \in \mathcal{Q}$, and time period t
$\vec{s}_{\text{best},m}$ Schedule to be implemented at mine m in the best solution to the MTP-MMPP found by our algorithm, $\vec{s}_{\text{best}}$
$IN(a)$, $a \in \mathcal{B}_m^A$ Set of grade or blast blocks forming part of aggregate $a \in \mathcal{B}_m^A$
$\mathcal{B}_m^*$ Set of grade blocks at mine m, together with all grade and blast blocks forming part of an aggregate in $\bar{\mathcal{B}}_m^A$

Parameters

T Number of periods in planning horizon
$T_k^{t=1}(b)$ Tons of material k in block b at the start of the planning horizon
MAX_i Cap on the number of iterations of the algorithm performed
$S_{b,l}^{m,k}$ Percentage of granularity $l \in \mathcal{L}$ in the material k within block $b \in \mathcal{B}_m$
$S_{s,l}^m$ Percentage of granularity $l \in \mathcal{L}$ in stockpile s at mine m
$Y_l^{m,\omega}$ Percentage of $l \in \mathcal{L}$ in any source $s \in \mathcal{S}_m$ that will be recovered after wet processing
C_π Tons of material that can pass through port $\pi \in \Pi$ per time period
C_e^m Tons of material that can be extracted from blocks at mine m per time period
T_R Tons of ore in a trainload
$U_{n,q}^{\pi,l,t}$ Upper bound on $q \in \mathcal{Q}$ percentage in product $n \in N_{\pi,l}$ formed by port π in t
Φ_ω^m Binary parameter equal to 1 if and only if mine m has a wet processing plant
M_A Maximum number of blocks permitted in an aggregate
Ξ_m Grade/quality target to be achieved by m over the planning horizon, as stated in a medium-term (five-year) plan
N Maximum number of schedules formed by each mine-side subproblem, $\mathcal{O}_m$
$T^{t=1}(s)$ Tons of material on stockpile s at the start of the planning horizon
$G_{b,l,q}^{m,k}$ Percentage of $q \in \mathcal{Q}$ in the granularity $l \in \mathcal{L}$ within material k of block $b \in \mathcal{B}_m$
$G_{s,l,q}^m$ Percentage of $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ of stockpile s at mine m
$R_{l,q}^{m,\omega}$ Percentage of $q \in \mathcal{Q}$ in the granularity $l \in \mathcal{L}$ of any source $s \in \mathcal{S}_m$ that will be recovered after wet processing
C_τ^m Tons of material haulable by truck per time period at mine m
γ Factor by which standard deviations are increased/reduced during our decomposition-based algorithm
C_κ^m, C_ω^m Tons of material that can be processed by the dry and wet plants at m, per period
Δ_q^+, Δ_q^- Significant and insignificant change in the percentage of attribute $q \in \mathcal{Q}$
$L_{n,q}^{\pi,l,t}$ Lower bound on the percentage of $q \in \mathcal{Q}$ in product $n \in N_{\pi,l}$ formed by port π in period t
TB_m Time period prior to which blast blocks at mine m cannot be mined
$\vec{\sigma}^+$, $\vec{\sigma}^-$ Set of upper and lower bounds on the standard deviation set $\vec{\sigma}_m^i$, $\forall\, i, m \in \mathcal{M}$

Functions

$f(t)$ Equal to $\max(1, t-1)$
$v_{l,q}^{m,t}(\vec{s})$ Percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period t of the MTP-MMPP solution, $\vec{s}$
$g(t)$ Equal to $\min(t+1, T)$, where T is the number of periods in the horizon
$\rho_m(\vec{s}, t)$ Productivity of mine m in period t in a solution to the MTP-MMPP, $\vec{s}$

Decision Variables and Expressions: **Monolithic MINLP**

$x_{s,d}^{m,t}$ Tons of source $s \in \mathcal{S}_m$, at mine m, hauled to destination $d \in \mathcal{D}_m$ in period t
$v_{l,q}^{m,t}$ Percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced at m in t
$\tau_l^{m,t}$ Tons of granularity $l \in \mathcal{L}$ produced at mine m in period t
$\rho_m(\vec{x}, t)$ Measure of productivity (equipment usage) at mine m in period t
$y_{m,t}^{\sigma,b}$ Binary, taking a value of 1 if and only if block $b \in \mathcal{B}_m$ has been mined (partially or completely) by, or in, period t at m
$y_{m,t}^{\tau,b}$ Binary, taking a value of 1 if and only if block $b \in \mathcal{B}_m$ has been completely mined by, or in, period t at m
$o_s^{m,t}$ Tons of material on stockpile s at mine m at the start of period t
$r_{m,\pi,t}^{l,n,t'}$ (Floating point) number of trains of granularity $l \in \mathcal{L}$, produced by mine m in period t', railed to port π to form part of product $n \in N_{\pi,l}$ in period t

$\overline{rv}^{m,l,q,t'}_{\pi,n,t}$ Tons of attribute $q \in \mathcal{Q}$, from ore produced by mine m in period t', blended into product $n \in N_{\pi,l}$ in period t

$\vec{x}$ Set of $x^{m,t}_{s,d}$ variables for all mines m, periods t, material sources $s \in \mathcal{S}_m$, and material destinations $d \in \mathcal{D}_m$

$\eta(\vec{x},\vec{r},t)$ Measure of deviation present between port product composition and desired bounds across all ports in period t

$\nu^{m,t}_{l,q}$ Tons of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period t

Decision Variables, Expressions, and Constants: $\mathcal{O}_\Pi$

$s_{m,j}$ Binary variable with value 1 if and only if $\mathcal{O}_\Pi$ selects the jth schedule in set $\Omega^i_m \cup \{\vec{s}_{\text{best},m}\}$, $\vec{s}_{m,j}$, to be enacted at mine m

$\tau^m_l(\vec{s}_{m,j},t)$ The tons of granularity $l \in \mathcal{L}$ produced by mine m in period t of schedule $\vec{s}_{m,j}$

N_m Number of candidate schedules available for $\mathcal{O}_\Pi$ to choose among for mine m in an iteration i ($N_m = |\Omega^i_m \cup \{\vec{s}_{\text{best},m}\}|$)

$\vec{r}'$ Set of all $r^{m,l,j,t'}_{\pi,n,t}$ variables for $m \in \mathcal{M}$, $l \in \mathcal{L}$, $j \in \{1,2,\ldots,N_m\}$, $t,t' \in \{1,2,\ldots,T\}$

$v^{m,t}_{l,q}(\vec{s}_{m,j})$ The percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period t of schedule $\vec{s}_{m,j}$

$\mathcal{P}^m_j$ The productivity achieved by mine m in schedule $\vec{s}_{m,j}$

$r^{m,l,j,t'}_{\pi,n,t}$ (Floating point) number of trainloads of granularity l produced by mine m in period t' of schedule $\vec{s}_{m,j}$ blended into product $n \in N_{\pi,l}$ at port π in period t

$\eta(\vec{r}',t)$ Measure of deviation present between composition of port products and desired bounds in period t

Decision Variables, Expressions, and Constants: $\mathcal{O}_m$

$\mathcal{O}_{m,1}$ Two-time-period MIP, solved as part of the rolling horizon heuristic for schedule generation at a mine m, which minimises deviation between the grade of production at m and a grade target

$\mathcal{O}_{m,2}$ Two-time-period MIP, solved as part of the rolling horizon heuristic for schedule generation at a mine m, which maximises productivity

h_1, h_2 Periods scheduled in $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$

$L^{m,t,j}_{l,q}$ Lower bound on the percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period t of the jth schedule formed by $\mathcal{O}_m$

$\tilde{x}^{m,h}_{s,d}$ Tons of source s hauled to destination d, at m, in period h of $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$

$\tilde{\mathbf{x}}_m$ Set of all $\tilde{x}^{m,h}_{s,d}$ variables for $m \in \mathcal{M}$, $h \in \{h_1,h_2\}$, source s, destination d

$\nu^m_{l,q}(\tilde{\mathbf{x}}_m,h)$ Tons of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ formed by mine m in period h

$\tau^m_l(\tilde{\mathbf{x}}_m,h)$ Tons of granularity $l \in \mathcal{L}$ produced by mine m in period h

$\Delta^{j,t}_{l,q}$ Random real number generated from a normal distribution

$U^{m,t,j}_{l,q}$ Upper bound on the percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period t of the jth schedule formed by $\mathcal{O}_m$

$L^{m,h}_{l,q}, U^{m,h}_{l,q}$ Bounds on the percentage of $q \in \mathcal{Q}$ in granularity $l \in \mathcal{L}$ produced by m in period h when solving $\mathcal{O}_{m,1}$ and $\mathcal{O}_{m,2}$

$\tilde{\mathcal{Q}}^{m,h}_{l,q}$ Percentage of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in period h of a solution to $\mathcal{O}_{m,1}$

$\eta(\tilde{\mathbf{x}}_m,h)$ Measure of total deviation present between the composition of production at mine m in period h and bounds given by $[L^{m,h}_{l,q}, U^{m,h}_{l,q}]$, for all $l \in \mathcal{L}$, $q \in \mathcal{Q}$

Appendix B. A MINLP Model of the MTP-MMPP

Constraints (B1)–(B4) ensure that in each period t, the capacity of dig units at each mine m, C^m_e, is not exceeded; material hauled is limited by C^m_τ; the processing of ore at each plant is equal, within a tolerance ϵ, to C^m_d for $d \in \{\kappa,\omega\}$; and the tons of ore railed to each port π does not exceed C_π. Constraint (B5) ensures that all ore produced by each mine m in each period t is transported to a port by the end of period $t+1$:

$$\sum_{b\in\mathcal{B}_m}\sum_{d\in\mathcal{D}_m} x^{m,t}_{b,d} \le C^m_e \quad \forall m\in\mathcal{M}, t\in\{1,2,\ldots,T\}; \tag{B1}$$

$$\sum_{s\in\mathcal{S}_m}\sum_{d\in\mathcal{D}_m} x^{m,t}_{s,d} \le C^m_\tau \quad \forall m\in\mathcal{M}, t\in\{1,2,\ldots,T\}; \tag{B2}$$

$$C^m_d - \epsilon \le \sum_{s\in\mathcal{S}_m} x^{m,t}_{s,d} \le C^m_d + \epsilon$$
$$\forall m\in\mathcal{M}, d\in\{\kappa,\omega\}, t\in\{1,2,\ldots,T\}; \tag{B3}$$

$$\sum_{m\in\mathcal{M}}\sum_{l\in\mathcal{L}}\sum_{n\in N_{\pi,l}}\sum_{t'=f(t)}^{t} T_R r^{m,l,t'}_{\pi,n,t} \le C_\pi \quad \forall \pi\in\Pi, t\in\{1,2,\ldots,T\}; \tag{B4}$$

$$\sum_{t=t'}^{g(t')}\sum_{\pi\in\Pi}\sum_{n\in N_{\pi,l}} T_R r^{m,l,t'}_{\pi,n,t} = \tau^{m,t'}_l \quad \forall m,l,t'\in\{1,2,\ldots,T\}. \tag{B5}$$

Constraint (B6) constrains the value of binaries $y^{\sigma,b}_{m,t}$ (1 if the mining of b has been scheduled by or during t) and $y^{\tau,b}_{m,t}$ (1 if b is scheduled to be entirely extracted by or during t). Vertical and disjunctive block precedences are respectively expressed in constraints (B7) and (B8):

$$y^{\tau,b}_{m,t}\sum_k T^{t=1}_k(b) \le \sum_{t'=1}^{t}\sum_{d\in\mathcal{D}_m} x^{m,t'}_{b,d} \le y^{\sigma,b}_{m,t}\sum_k T^{t=1}_k(b)$$
$$\forall m\in\mathcal{M}, b\in\mathcal{B}_m, t\in\{1,2,\ldots,T\}; \tag{B6}$$

$$y^{\tau,b'}_{m,t} \ge y^{\sigma,b}_{m,t} \quad \forall m\in\mathcal{M}, b\in\mathcal{B}_m, b'\in\mathcal{A}^{\wedge}_{m,b}, t\in\{1,2,\ldots,T\}; \tag{B7}$$

$$\sum_{b'\in\mathcal{A}^{\vee}_{m,b}} y^{\tau,b'}_{m,t} \ge y^{\sigma,b}_{m,t} \quad \forall m\in\mathcal{M}, b\in\mathcal{B}_m, t\in\{1,2,\ldots,T\}. \tag{B8}$$

Constraint (B9) ensures that no blast blocks are mined prior to a specific time period, TB_m, at each mine $m \in \mathcal{M}$. Prior to TB_m, only grade blocks can be extracted at m. Constraints (B10)–(B13) ensure that stockpile capacities are respected and that no more than $o^{m,t}_s$ tons (the tons of ore on stockpile s at the start of t) can be extracted from $s \in \Theta_m \cup \Lambda_m$ in t:

$$y^{\sigma,b}_{m,t} = 0 \quad \forall m \in \mathcal{M}, b \in \mathcal{B}^b_m, t < TB_m; \tag{B9}$$

$$o^{m,t}_s = T^{t=1}(s) \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, t = 1; \tag{B10}$$

$$o^{m,t}_s = o^{m,t-1}_s - \sum_{d \in \{\kappa,\omega\}} x^{m,t-1}_{s,d} + \sum_{b \in \mathcal{B}_m} x^{m,t-1}_{b,s} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, t \in \{2,3,\ldots,T\}; \tag{B11}$$

$$o^{m,t}_s - \sum_{d \in \{\kappa,\omega\}} x^{m,t}_{s,d} + \sum_{b \in \mathcal{B}_m} x^{m,t}_{b,s} \le C^m_s \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, t \in \{2,3,\ldots,T\}; \tag{B12}$$

$$o^{m,t}_s - \sum_{d \in \{\kappa,\omega\}} x^{m,t}_{s,d} \ge 0 \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, t \in \{2,3,\ldots,T\}. \tag{B13}$$

Constraints (B14)–(B17) define the tons of each granularity $l \in \mathcal{L}$ and attribute $q \in \mathcal{Q}$ residing on each stockpile $s \in \Theta_m \cup \Lambda_m$ at the start of period t:

$$o^{m,t}_{s,l} = S^{m,t=1}_{s,l} T^{t=1}(s) \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, t = 1; \tag{B14}$$

$$o^{m,t}_{s,l} = o^{m,t-1}_{s,l} - \sum_{d \in \{\kappa,\omega\}} a^{m,t-1,d}_{s,l} + \sum_{b \in \mathcal{B}_m} S^m_{b,l} x^{m,t-1}_{b,s} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, t \in \{2,3,\ldots,T\}; \tag{B15}$$

$$o^{m,t}_{s,l,q} = S^{m,t=1}_{s,l} G^{m,t=1}_{s,l,q} T^{t=1}(s) \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, q \in \mathcal{Q}, t = 1; \tag{B16}$$

$$o^{m,t}_{s,l,q} = o^{m,t-1}_{s,l,q} - \sum_{d \in \{\kappa,\omega\}} a^{m,t-1,d}_{s,l,q} + \sum_{b \in \mathcal{B}_m} S^m_{b,l} G^m_{b,l,q} x^{m,t-1}_{b,s} \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Lambda_m, l \in \mathcal{L}, q \in \mathcal{Q}, t \in \{2,3,\ldots,T\}. \tag{B17}$$

Constraints (B18) and (B19) ensure that the composition of material leaving each stockpile $s \in \Theta_m \cup \Lambda_m$ for a processing plant $d \in \{\kappa, \omega\}$ in each period t is equal to that of the stockpile at the start of t:

$$a^{m,t,d}_{s,l} o^{m,t}_s = x^{m,t}_{s,d} o^{m,t}_{s,l} \quad \forall m \in \mathcal{M},\ s \in \Theta_m \cup \Lambda_m,\ d \in \{\kappa,\omega\},\ l \in \mathcal{L},\ t \in \{1,2,\ldots,T\}; \tag{B18}$$

$$a^{m,t,d}_{s,l,q} o^{m,t}_{s,l} = a^{m,t,d}_{s,l} o^{m,t}_{s,l,q} \quad \forall m \in \mathcal{M},\ s \in \Theta_m \cup \Lambda_m,\ d \in \{\kappa,\omega\},\ l \in \mathcal{L},\ q \in \mathcal{Q},\ t \in \{1,2,\ldots,T\}. \tag{B19}$$

Constraints (B20)–(B24) prevent material movement along invalid pathways and ensure that no more high-grade (hi), low-grade (lo), or waste (w) material is extracted from a block than exists at the start of $t = 1$:

$$x^{m,t}_{s,\kappa} = 0 \quad \forall m \in \mathcal{M}, s \in \Lambda_m \cup \Delta_m, t \in \{1,2,\ldots,T\}; \tag{B20}$$

$$x^{m,t}_{s,\omega} = 0 \quad \forall m \in \mathcal{M}, s \in \Theta_m \cup \Delta_m, t \in \{1,2,\ldots,T\}; \tag{B21}$$

$$\sum_{t=1}^{T} \sum_{d \in \mathcal{D}_m} x^{m,t}_{b,d} x^{m,t}_{b,\kappa} + \sum_{s \in \Theta_m} x^{m,t}_{b,s} \le T^{t=1}_{hi}(b) \quad \forall m \in \mathcal{M}, b \in \mathcal{B}_m; \tag{B22}$$

$$\sum_{t=1}^{T} x^{m,t}_{b,\omega} + \sum_{s \in \Lambda_m} x^{m,t}_{b,s} \le T^{t=1}_{lo}(b) \quad \forall m \in \mathcal{M}, b \in \mathcal{B}_m; \tag{B23}$$

$$\sum_{t=1}^{T} \sum_{s \in \Delta_m} x^{m,t}_{b,s} \le T^{t=1}_{w}(b) \quad \forall m \in \mathcal{M}, b \in \mathcal{B}_m. \tag{B24}$$

Variables $\tau^{m,t}_l$ and $v^{m,t}_{l,q}$ are defined in constraints (B26) and (B27). The tons of $q \in \mathcal{Q}$ in the ore of granularity $l \in \mathcal{L}$ produced by mine m in t, denoted by $\nu^{m,t}_{l,q}$, is defined in Equation (B25):

$$\nu^{m,t}_{l,q} = \sum_{\lambda \in \Lambda_m} R^{m,\omega}_{l,q} a^{m,t,\omega}_{\lambda,l,q} + \sum_{\theta \in \Theta_m} a^{m,t,\kappa}_{\theta,l,q} + \sum_{b \in \mathcal{B}_m} S^{m,hi}_{b,l} G^{m,hi}_{b,l,q} x^{m,t}_{b,\kappa} + S^{m,lo}_{b,l} G^{m,lo}_{b,l,q} R^{m,\omega}_{l,q} x^{m,t}_{b,\omega} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, l \in \mathcal{L}, q \in \mathcal{Q}; \tag{B25}$$

$$\tau^{m,t}_l = \sum_{\lambda \in \Lambda_m} Y^{m,\omega}_l a^{m,t,\omega}_{\lambda,l} + \sum_{\theta \in \Theta_m} a^{m,t,\kappa}_{\theta,l} + \sum_{b \in \mathcal{B}_m} S^{m,hi}_{b,l} x^{m,t}_{b,\kappa} + S^{m,lo}_{b,l} Y^{m,\omega}_l x^{m,t}_{b,\omega} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, l \in \mathcal{L}; \tag{B26}$$

$$v^{m,t}_{l,q} \tau^{m,t}_l = \nu^{m,t}_{l,q} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, l \in \mathcal{L}, q \in \mathcal{Q}. \tag{B27}$$

Constraint (B28) defines variable $\overline{rv}^{m,l,q,t'}_{\pi,n,t}$, used in Equation (2), to compute the extent to which the composition of products formed at each port deviates from desired bounds:

$$T_R r^{m,l,t'}_{\pi,n,t} v^{m,t'}_{l,q} = \overline{rv}^{m,l,q,t'}_{\pi,n,t} \quad \forall m \in \mathcal{M}, t \in \{1,2,\ldots,T\}, t' \in \{f(t), t\}, l \in \mathcal{L}, q \in \mathcal{Q}, n \in N_{\pi,l}. \tag{B28}$$

We define a set of separation constraints to prevent the extraction of blocks that are not immediately accessible on the mining face. Blom et al. (2014) provide a set of such constraints for the single-time-period MMPP, which we generalise to the multiple-time-period setting. A block $b \in \mathcal{B}_m$ is defined to lie on a mining face if $|\mathcal{A}^{\vee}_{m,b}| = 0$ (i.e., no adjacent blocks must be mined before b can be accessed). The set $\mathcal{P}'(\mathcal{B}_m)$ contains all contiguous sets of blocks $\mathcal{B}'_m \subset \mathcal{B}_m$ for which $\nexists b' \in \mathcal{B}'_m$ such that $|\mathcal{A}^{\vee}_{m,b'}| = 0$. The set $\mathcal{N}(\mathcal{B}_m, \mathcal{B}'_m)$ contains all blocks $b'' \in \mathcal{B}_m \backslash \mathcal{B}'_m$ for which $\exists b' \in \mathcal{B}'_m$ such that $b'' \in \mathcal{A}^{\vee}_{m,b'}$ (i.e., the "neighbours" of the blocks in set $\mathcal{B}'_m$). Selected instances of constraint set (B29) are added to the MINLP via a separation algorithm.[3] Each constraint in this set states that if a block in set $\mathcal{B}'_m$ is mined in period t, then at least one of the neighbours of $\mathcal{B}'_m$ must have been completely extracted by or during t:

$$\sum_{b'' \in \mathcal{N}(\mathcal{B}_m, \mathcal{B}'_m)} y^{\tau,b''}_{m,t} \ge \frac{1}{|\mathcal{B}'_m|} \sum_{b' \in \mathcal{B}'_m} y^{\sigma,b'}_{m,t} \quad \forall m \in \mathcal{M},\ \mathcal{B}'_m \in \mathcal{P}'(\mathcal{B}_m),\ t \in \{1,2,\ldots,T\}. \tag{B29}$$

Indicators $y^{\tau,b}_{m,t}$ and $y^{\sigma,b}_{m,t}$, for all $m \in \mathcal{M}$, $b \in \mathcal{B}_m$, and $t \in \{1,2,\ldots,T\}$, are restricted to binary values, whereas all remaining variables are restricted to nonnegative reals.

References

Audet C, Brimberg J, Hansen P, Digabel SL, Mladenovic N (2004) Pooling problem: Alternate formulations and solution methods. *Management Sci.* 50(6):761–776.

[3] Instances of constraint (B29) are added to our implementation of this model via the use of lazy constraint callbacks. For brevity, further details of this separation algorithm have been omitted from this paper.

Bassett MH, Pekny JF, Reklaitis GV (1996) Decomposition techniques for the solution of large-scale scheduling problems. *Amer. Inst. Chem. Engineers J.* 42(12):3373–3387.

Bengtsson J, Bredström D, Flisberg P, Rönnqvist M (2013) Robust planning of blending activities at refineries. *J. Oper. Res. Soc.* 64(6):848–863.

Bitran GR, Hax AC (1977) On the design of hierarchical production planning systems. *Decision Sci.* 8(1):28–55.

Bley A, Boland N, Froyland G, Zuckerberg M (2012) Solving mixed integer nonlinear programming problems for mine production planning with stockpiling. Working paper, Institute for Mathematics, Technical University of Berlin, Berlin.

Blom ML, Burt CN, Pearce AR, Stuckey PJ (2014) A decomposition-based heuristic for collaborative scheduling in a network of open-pit mines. *INFORMS J. Comput.* 26(4):658–676.

Boland N, Dumitrescu I, Froyland G, Gleixner AM (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.

Carvalho M, Pinto J (2006) A bilevel decomposition technique for the optimal planning of offshore platforms. *Brazilian J. Chemical Engrg.* 23(1):67–82.

Cullenbine C, Wood RK, Newman A (2011) A sliding time window heuristic for open pit mine block sequencing. *Optim. Lett.* 5(3):365–377.

Dimitriadis AD, Shah N, Pantelides CC (1997) RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Comput. Chemical Engrg.* 21(12):1061–1066.

Elkamel A, Zentner M, Pekny F, Reklaitis GV (1997) A decomposition heuristic for scheduling the general batch chemical plant. *Engrg. Optim.* 28(4):299–330.

Epstein R, Goic M, Weintraub A, Catalán J, Nez PS, 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): 93–114.

Everett JE (1996) Iron ore handling procedures enhance export quality. *Interfaces* 26(6):82–94.

Everett JE, Kamperman M, Howard TJ (2002) Information for decision support, information for performance evaluation—But don't mix your drinks. *Proc. Informing Sci. IT Ed. Conf., Cork, Ireland*, 427–439.

Goodwin GC, Seron MM, Middleton RH, Zhang M, Hennessy BF, Stone PM, Menabde M (2006) Receding horizon control applied to optimal mine planning. *Optimal Control Appl. Management Sci.* 8(8):1337–1342.

Gounaris CE, Misener R, Floudas CA (2009) Computational comparison of piecewise-linear relaxations for pooling problems. *Indust. Engrg. Chemistry Res.* 48(12):5742–5766.

Haverly CA (1978) Studies of the behaviour of recursion for the pooling problem. *ACM SIGMAP Bull.* 25(1):19–28.

Hax AC, Meal HC (1973) Hierarchical integration of production planning and scheduling. Technical report, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

Iyer RR, Grossmann IE (1998) Optimal planning and scheduling of offshore oil field infrastructure investment and operations. *Indust. Engrg. Chemistry Res.* 37(4):1380–1397.

Janak SL, Floudas CA, Kallrath J, Vormbrock N (2006) Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling. *Indust. Engrg. Chemical Res.* 45(25):8234–8252.

Kumral M (2006) Bed blending design incorporating multiple regression modelling and genetic algorithms. *J. South African Inst. Mining and Metallurgy* 106(3):229–237.

Lambert WB, Newman AM (2013) Tailored lagrangian relaxation for the open pit block sequencing problem. *Ann. Oper. Res.* 222(1):1–20.

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

Maravelias CT, Sung C (2009) Integration of production planning and scheduling: Overview, challenges, and opportunities. *Comput. Chemical Engrg.* 33(12):1919–1930.

Màs R, Pinto JM (2003) A mixed-integer optimization strategy for oil supply in distribution complexes. *Optim. Engrg.* 4(1):23–64.

McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part I–Convex underestimating problems. *Math. Programming* 10(1):147–175.

Méndez CA, Cerdá J, Grossmann IE, Harjunkoski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chemical Engrg.* 30(1): 931–946.

Minnitt RCA, Pitard FF (2008) Application of variography to the control of species in material process streams. *J. Southern African Inst. Mining and Metallurgy* 108(2):109–122.

Misener R, Floudas CA (2009) Advances for the pooling problem: modelling, global optimisation, and computational studies. *Appl. Comput. Math.* 8(1):3–22.

Müller KP (2010) Stacking, reclaiming and blending effects. *Mechanical Tech.* (November):14–17.

Newman AM, Kuchta M (2007) Using aggregation to optimize long-term production planning at an underground mine. *Eur. J. Oper. Res.* 176(2):1205–1218.

Newman AM, Rubio E, Caro R, Weintraub A, Eurek K (2010) A review of operations research in mine planning. *Interfaces* 40(3):222–245.

Osanloo M, Gholamnejad J, Karimi B (2008) Long-term open pit mine production planning: A review of models and algorithms. *Internat. J. Mining, Reclamation, Environment* 22(1):3–35.

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

Reddy PP, Karimi I, Srinivasan R (2004) A new continuous-time formulation for scheduling crude oil operations. *Chemical Engrg. Sci.* 59(6):1325–1341.

Rodrigues MTM, Gimeno L, Passos CAS, Campos MD (1996) Reactive scheduling approach for multipurpose chemical batch plants. *Comput. Chemical Engrg.* 20(Supplement 2):S1215–S1220.

Rogers D, Plante RD, Wong RT, Evans JR (1991) Aggregation and disaggregation techniques and methodology in optimization. *Oper. Res.* 39(4):553–582.

Shah N (1996) Mathematical programming techniques for crude oil scheduling. *Comput. Chemical Engrg.* 20(Supplement 2): S1227–S1232.

Singh G, García-Flores R, Ernst A, Welgama P, Zhang M, Munday K (2013) Medium-term rail scheduling for an iron ore mining company. *Interfaces* 4(2):222–240.

Sundar DK, Acharya D (1995) Blast schedule planning and shift-wise production scheduling of an opencast iron ore mine. *Comput. Indust. Engrg.* 28(4):927–935.

Tabesh M, Askari-Nasab H (2011) Two-stage clustering algorithm for block aggregation in open pit mines. *Mining Tech.* 120(3):158–169.

Thomas A, Singh G, Krishnamoorthy M, Venkateswaran J (2012) Distributed optimisation method for multi-resource constrained scheduling in coal supply chains. *Internat. J. Production Res.* 51(9):2740–2759.

Thomas A, Venkateswaran J, Singh G, Krishnamoorthy M (2014) A resource constrained scheduling problem with multiple independent producers and a single linking constraint: A coal supply chain example. *Eur. J. Oper. Res.* 236(3):946–956.

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.

Weintraub A, Pereira M, Schultz X (2008) A priori and a posteriori aggregation procedures to reduce model size in MIP mine planning models. *Electronic Notes Discrete Math.* 30(1):297–302.

Wenkai L, Hui CW, Hua B, Tong Z (2002) Scheduling crude oil unloading, storage, and processing. *Indust. Engrg. Chemistry Res.* 41(26):6723–6734.