



Long-term maintenance optimization for integrated mining operations

Yingying Yang¹ · Ryan Loxton¹ · Andrew L. Rohl¹ · Hoa T. Bui¹

Received: 23 July 2023 / Revised: 12 October 2023 / Accepted: 12 October 2023
© The Author(s) 2023

Abstract

Maintenance activities are inevitable and costly in integrated mining operations. Conducting maintenance may require the whole system, or sub-units of the system, to be shut down temporarily. These maintenance activities not only disrupt the unit being shut down, but they also have consequences for inventory levels and product flow downstream. In this paper, we consider an interconnected mining system in which there are complicated maintenance relationships and stock accumulation at intermediate nodes. We propose a time-indexed mixed-integer linear programming formulation to optimize the long-term integrated maintenance plan and maximize the total throughput. We also devise an algorithm, which combines Benders decomposition and Lagrangian relaxation, to accelerate the computational speed. To validate our mathematical model, we perform simulations for a real-world case study in the iron ore industry. The results show that our method can yield better solutions than CPLEX optimization solver alone in faster time.

Keywords Long-term maintenance planning · Optimization · Mixed integer linear programming · Integrated network · Mining industry

Mathematics Subject Classification 90B06 · 90B50 · 90B90 · 90C11

✉ Yingying Yang
yingying.yang1@postgrad.curtin.edu.au

Ryan Loxton
r.loxton@curtin.edu.au

Andrew L. Rohl
andrew.rohl@curtin.edu.au

Hoa T. Bui
hoa.bui@curtin.edu.au

¹ ARC Training Centre for Transforming Maintenance through Data Science, Curtin University, Perth, Australia

1 Introduction

Maintenance shutdowns (shutdowns for short) are lengthy and necessary stoppages to undertake preventive maintenance, strip-down and overhaul, corrective repair and/or component replacement and occur in all heavy industries (Levitt 2004). They are crucial to restoring plant processes to a good operating condition. In practice, shutdowns significantly impact the net earnings for companies, not only in direct maintenance costs, but also the production loss while the plant is not operating (Schulz et al. 2006). Therefore, having an optimal shutdown maintenance plan that specifies the best timings for maintenance can bring large benefits.

An integrated mining site is a complex production system that consists of many processing assets, intermediate buffer storages and linking facilities. Disruption of any unit may impact upstream/downstream product flow, stock levels, throughput and the ability to satisfy customer demands. It is therefore essential to take into account the impact on production and operations when planning maintenance (Budai et al. 2008). The purpose of this paper is to present a mathematical model for optimizing the interplay between maintenance timings and the performance of the whole system over a long-term planning horizon.

Planning for shutdown maintenance events is a complex process. Since the 1970s, interest in tackling shutdown issues has increased significantly (Lawrence 1976). Two main research streams can be identified: single shutdown planning and multi-shutdown planning. In the first stream, the focus is on determining the optimal sequence of maintenance activities in a single shutdown and minimizing shutdown duration and resource requirements. The second stream is to find the best way to schedule a series of shutdowns to ensure they are appropriately staggered to balance resource needs over the planning horizon. Our work is related to the latter. Optimizing multi-shutdown planning in an integrated system is a challenging problem due to various constraints. For example, certain assets cannot be shut down at the same time, and some maintenance cannot be performed during certain seasons. Maintenance conducted on one unit may impede upstream flows or starve downstream operations, and hence performing maintenance at the wrong times may impede throughput and cause stock to build up to excessive levels. Thus, in an interconnected production network, optimizing integrated maintenance planning at a holistic system level, rather than just at a single sub-unit level, is highly desired. Al-Turki et al. (2013) reviewed the development of shutdown maintenance planning and proposed a framework to encapsulate the holistic view. Duffuaa et al. (2019) provided a comprehensive industry survey to measure the impact of integrated shutdown maintenance planning on system reliability and productivity, which highlighted the importance of maintenance scheduling at the global level. A detailed discussion on shutdown maintenance planning and industry applications can be found in Al-Turki et al. (2019).

Some specific work on integrated shutdown planning in the chemical industry is described below. Cheung et al. (2004) investigated site-wide shutdown maintenance scheduling for an integrated production site in the chemical industry to determine the exact time windows of shutdowns in a short-term time horizon of four to ten weeks. Their objective was to maximize profit, based on material demand profiles, inventories, and variation of utility contracts. Amaran et al. (2015) proposed a math-

emathical model to optimize long-term shutdown planning for integrated chemical sites, incorporating timing constraints, financial performance and resource availability. They later explored medium-term maintenance planning optimization, considering uncertainty in shutdown duration and production decisions (Amaran et al. 2016). A combination of robust optimization and stochastic programming was used for that purpose.

The references discussed in the previous paragraph focus on the chemical industry. In integrated mining operations, the size and scale of the system is much larger including plants, railways and port facilities, and overstock can be more easily accumulated than in chemical applications. Boland et al. (2013) formulated a two-stage model for scheduling annual maintenance to maximize the total throughput for the Hunter Valley Coal Chain in eastern Australia. In a later paper, the same authors developed an integer programming model to schedule maintenance on arcs (e.g. railway tracks, pipes) in a weighted network, maximizing the total flow through the network (Boland et al. 2014a). Since the problem is strongly NP-hard (see Boland et al. 2014b for a complexity analysis), they proposed several local search-based heuristic approaches to solve the problem. The network considered consists of the railway track and terminal equipment, and the only maintenance constraint enforced is that each maintenance activity needs to be conducted once. In this paper, we take a wider view and consider not only rail and terminal equipment, but also upstream assets such as the processing plant. The goal is to generate an integrated maintenance schedule for minimizing production disruptions. The inclusion of the processing plant and other assets means that we need to consider more complex maintenance relationships, such as maintenance on certain units cannot overlap, some maintenance activities can only start after others are completed and some maintenance activities must be performed in alignment with others. This type of situation is common in iron ore mining, for example.

We organize our paper as follows. After a thorough description of the maintenance planning problem and network structure in Sect. 2, we formulate a bi-criteria mixed-integer linear programming model to represent shutdown maintenance planning scenarios in Sect. 3. In our model, we incorporate not only many practical maintenance requirements and timing restrictions, but also the effects of intermediate storage and network flows. In Sect. 4, we design a Benders Lagrangian decomposition algorithm to reduce the computational time. In Sect. 5, we present the computational results of a real-life industry case study involving the production network of an iron ore mining company in Western Australia and perform extensive sensitivity analysis to gain insights into the developed model. Finally, in Sect. 6, we summarize our work and discuss some potential extensions and directions for future work. To the best of our knowledge, our paper is the first to consider long-term maintenance planning in the context of simultaneously optimizing throughput and inventory management in an integrated mining system. We expect that the proposed approach could also be adapted to other asset-intensive industries in which maintenance scheduling is required and production occurs via an interconnected network with associated flows.

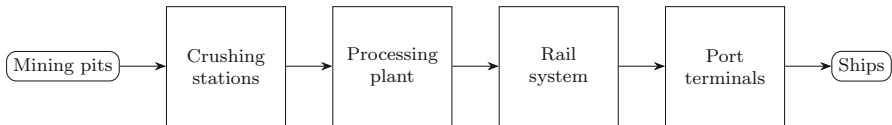


Fig. 1 A block model for an integrated mining operation

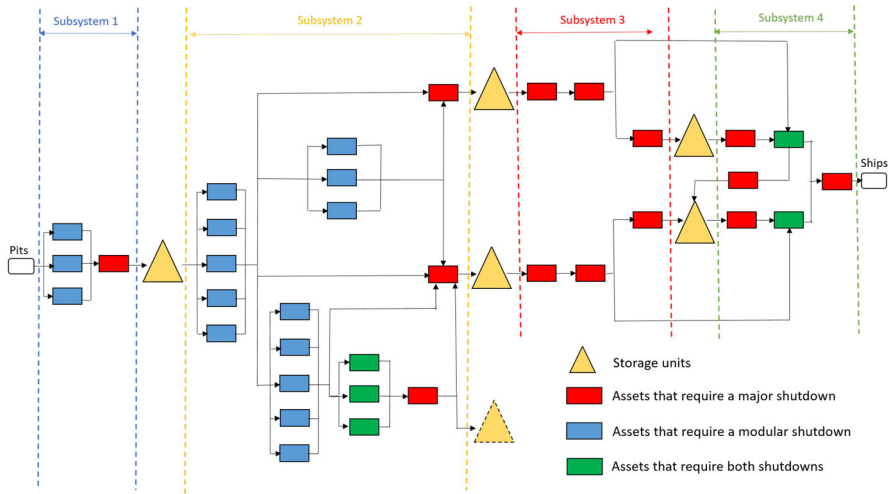


Fig. 2 An example diagram for an iron ore mining operation

2 Problem statement

Generally, mining operations comprise an integrated network of several pits, crushing stations, processing units, rail infrastructure and port terminals. The original material is transported to Run of Mine stockpiles that are located at the edge of the mining pits. After being crushed in the crushing stations, ore is then delivered by conveyors to the processing plant, where it is further processed and classified. All products are then transported along a railway line to the port terminals and loaded onto vessels for shipping to customers. The process can be visualized as a block model as shown in Fig. 1.

As an example, Fig. 2 shows a schematic diagram of a real iron ore mining system. The whole system is divided into four subsystems according to their functions (recall the four squares in Fig. 1). Note that subsystem 3 contains loading facilities (in the processing plant), railway system and unloading facilities (in the port). Maintenance is conducted either during major shutdowns, where an entire subsystem is shut down, or during modular shutdowns, where individual assets within a subsystem are shut down. The rounded rectangles in Fig. 2 represent the start and end points (pits and ships, respectively), and all regular rectangles represent assets that need to be maintained on a periodic basis. The red assets need to be maintained in major shutdowns, which must occur at a certain frequency. The blue assets only require modular shutdowns. The green assets should be maintained during both major and modular

shutdowns, since these assets require either more frequent maintenance or additional maintenance outside of major shutdowns due to resource and time limitations. The triangles with a solid outline represent stockpiles that serve as intermediate buffers between consecutive assets, while the dotted triangle represents the tailings storage for waste treatment. All assets and storage units are connected by solid arrows that indicate the flow of material via conveyor belts or rail tracks. In this paper, we are interested in scheduling maintenance for all assets from the crushing stations to the port.

As can be seen from the diagram, the network is strongly interconnected, and shutting down an asset will reduce the outflow from the asset and cause upstream stockpiles to grow. Thus, a good maintenance plan should not only be feasible with respect to the maintenance requirements, but also ensure stock can progress through the network to meet shipping schedules and maximize total throughput. This means optimizing the timings of maintenance disruptions (major and modular shutdowns) is key to operating the system effectively.

In what follows, we impose some assumptions to facilitate the mathematical modelling of this problem. The first assumption is that there is ample supply of raw materials (no limitations) from the pits to feed the crushing stations. This is because our focus is on reducing bottlenecks and throughput limits caused by maintenance, not on production scheduling. The second assumption is that there is no delay between the inflow and outflow of an asset. Although in practice there is always some delay (normally a few hours), it is smaller than the time discretization of our problem (1 day). The third assumption is that the final products are perfectly separated and product blending is not considered.

There are typically three types of shutdown plans in the mining industry: long-term plans, medium-term plans and short-term plans. Our model focuses on long-term planning and our goal is to give a timeline of shutdowns over 1–2 years. This information then typically flows into medium-term plans of 3–6 months and ultimately plans for a single shutdown. This allows for forward planning, such as labour and contractor hire, inventory and stockpile management, etc. We will focus on providing long-term maintenance plans (which maintenance operation starts at what time) for the whole system to satisfy all maintenance requirements, while maximizing throughput and minimizing excessive accumulation of stock in intermediate storage.

3 Mathematical optimization model

In this section, we give the solution approach and present a time-indexed mixed integer linear programming (MILP) model for the problem described in Sect. 2.

3.1 Sets and parameters

We consider a directed network (N, E) . The nodes in N represent facilities and the edges in E represent links between different facilities. Specifically, the node set N can be partitioned into four parts, i.e., $\{0\} \cup I \cup P \cup \{e\}$, where the start node 0 and

the end node e represent pits and ships respectively, the node set I and P represent all assets and storage units respectively. Moreover, let N_n^+ denote the nodes that receive the flow leaving from node n , let N_n^- denote the nodes that provide flow to node n . Mathematically, they can be formulated as below:

$$N_n^+ := \{m : (n, m) \in E, \forall m \in N\}, \quad N_n^- := \{m : (m, n) \in E, \forall m \in N\}.$$

Let A denote the set of all maintenance operations and A_M denote maintenance operations that require a major shutdown. Each operation $i \in A$ is associated with a duration d_i and a number of resources r_i . Let D be the number of periods in the planning time horizon and D_i be the set of periods during which the corresponding shutdown for i may occur. Let R^{\max} denote the maximum resources provided for the entire system over the whole time horizon.

To define the relationships between maintenance operations, we introduce the following notations.

- \mathcal{A}_{NO} represents the set of pairs of maintenance operations that cannot overlap with each other. For example, in Fig. 2, the modular shutdowns cannot conflict with the major shutdown of the same subsystem due to resource limitations.
- For each operation i , there are parameters a_i^- and a_i^+ and a corresponding set of maintenance operations A_i , such that at most λ_i^{\max} operations in set A_i can start within $-a_i^-$ and a_i^+ of the start of operation i . See Fig. 3 for an illustration. This constraint can be used to represent limits on the number of similar maintenance activities that can be performed during the same time window due to resource limitations, space constraints, and lack of redundancy. The parameter λ_i^{\max} only constrains the numbers of operations in A_i that can start during the specified time window around operation i , but it does not restrict which operation in A_i will be chosen.
- For each operation i , there are parameters b_i^- and b_i^+ and a corresponding set of maintenance operations B_i , such that at least λ_i^{\min} operations in set B_i must start within $-b_i^-$ and b_i^+ of the start of operation i . For example, in subsystem 2 in the network in Fig. 2, it is strongly desired to perform modular shutdowns for some of the adjacent assets simultaneously while product flow is reduced.
- For each pair of maintenance operations (i, j) , the duration between the start of i and the start of j must be within G_{ij}^{\min} (the minimum gap) and G_{ij}^{\max} (the maximum gap). Figure 4 illustrates the start times for a pair of operations. For instance, there is typically a frequency at which the same type of operation must occur on the same asset, such as one shutdown every three months.

Stockyards in the mining industry are designed to manage the inventory while creating sufficient buffer to reduce the impacts of shutdowns of the subsystems. Figure 5 illustrates the structure of a stockyard. For example, product 1 is conveyed to the corresponding stockpiles (canyon B) via a specified stacker (the solid square next to B), and reclaimed by a bucket wheel reclaimer (the solid circle next to B) to feed the next process. All stock in canyon B is called "live stock", as it can be regained directly by the reclaimer. When canyon B is full, product 1 will be stacked in canyon A. Unfortunately, those products cannot be reclaimed by the reclaimer in the normal way, that

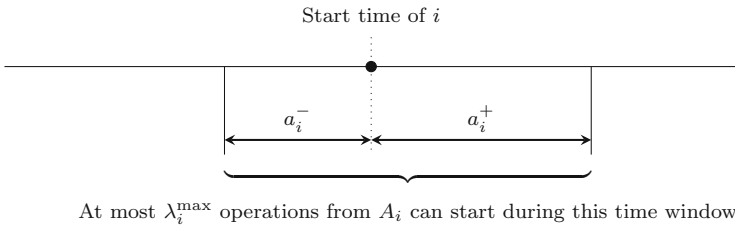


Fig. 3 The definition of A_i , λ_i^{\max} , a_i^- , and a_i^+

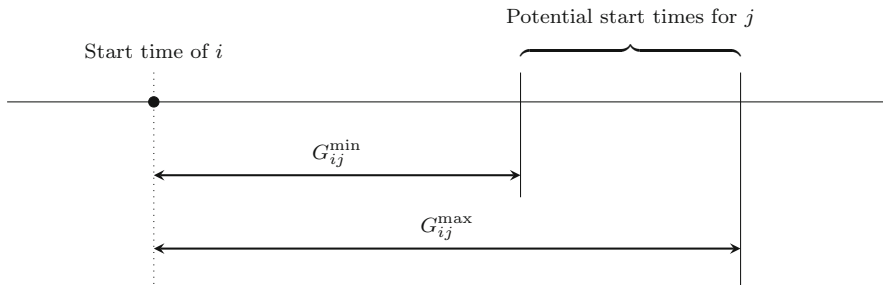


Fig. 4 The start times for a pair of maintenance operations (i, j)

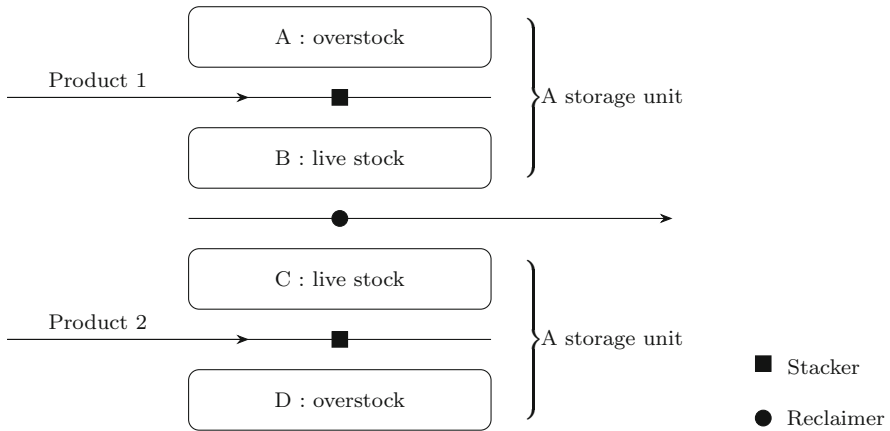


Fig. 5 Illustration of stockpiles and facilities in a stockyard. The rectangles represent stockpiles. The solid squares and circles represent stackers and reclaimers respectively

is why they are also called “overstock”, and this stock can only be recovered using mobile facilities at a high additional cost. In this example, canyons A and B together constitute a storage unit in the network, while C and D constitute another unit.

For the intermediate storages, each unit $n \in P$ is assigned with a maximum live stock capacity z_{nd}^{\max} and a minimum value z_{nd}^{\min} during period $d \in D$, as well as an overstock limit z_{nd}^{\max} . One of our objectives is to avoid overstock or force it to be as small as possible. To control the requirement of avoiding overstock in n , we introduce

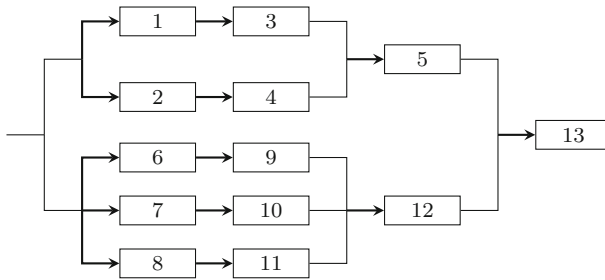


Fig. 6 An example diagram illustrating flow capacity

a penalty parameter denoted by α_n . Moreover, we introduce a dummy time period and let $z_{n,0}$ denote the initial stock level at the beginning of the time horizon.

In terms of flow in the network, let y_{mnd}^{\max} denote the maximum carrying capacity of the edge from node m to n during period d . In practice, the flow to some nodes may have additional capacity limits. For example, in Fig. 6, products from nodes 3 and 4 merge into the same edge to node 5. That is, there exists an inflow limit to node 5. Let N_{SI} denote the set of such nodes that have inflow capacity limits. Let y_n^{\max} denote the maximum flow capacity through node n . Moreover, the downstream flow from some nodes may have restrictions. For instance, recall Fig. 5, the reclaiming is located in the middle between product 1 and product 2 stockpiles. It only has the ability to reclaim one type of product at a time, i.e. products 1 and 2 cannot be reclaimed simultaneously. In Fig. 6, assume nodes 1 and 2 share the same equipment, then there are restrictions on the summation of outflows from nodes 1 and 2. We say the node set $\{1,2\}$ is a group of such nodes. Similarly, if nodes 6, 7 and 8 share the same equipment, then we have another group $\{6,7,8\}$ of such nodes. In general, let N_{SO}^g denote the g^{th} group of nodes that have outflow capacity limits, and G^c denote the number of such groups.

During the course of the process, the coarse ore is beneficiated to create different types of final products, and usually an approximate percentage between the flow leaving a node and the flow entering its downstream adjacent nodes is maintained.

For instance, an example subnetwork in Fig. 7 shows that after being processed in node 1, the product is distributed into three parts at percentages of 10%, 40% and 50%. Let N_C denote the set of such nodes which have certain percentages of outflow to their adjacent nodes. It is worth noting that the percentage of 40% of the total flow to nodes 3 and 4 does not mean there is always 20% of the flow entering node 4. When node 3 is shut down, the flow to node 4 would be 40%, but it may have a certain level of impact on throughput since the maximum flow limit through node 4 must be satisfied. For example, referring to Fig. 7, if node 3 should be maintained during a time period, then the flow from node 1 to 3 is zero. Since the upper bound of flow entering node 4 is 30 kt, the maximum outflow from node 1 should be $(0 + 30) \div 40\% = 75$ kt during the same period. These changes imply that the total throughput is affected slightly, dropping from 100 to 75 kt. We call such nodes 3 and 4 as group adjacent nodes of node 1, and the set $\{3,4\}$ is a set of node 1's group adjacent nodes. Conversely, when node 2 is shut down, the flow to node 2 is zero ($y_{1,2} = 0$). Since flow conservation must be respected ($y_{1,2}$ must be 10% of the total outflow from node 1), the flow from node 1 must be zero, and there is no throughput at all. We call such node 2

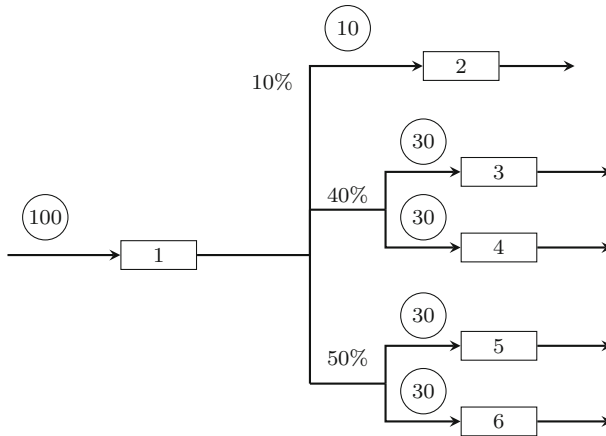


Fig. 7 An example diagram illustrating flow distribution. Note that the numbers in the circles represent the maximum capacity of each edge with the unit kiloton (kt)

as normal adjacent nodes of node 1. To differentiate these two situations, let N_n^{F+} denote all the normal adjacent nodes of n . Let N_{ng}^{P+} denote the g^{th} set of n 's group adjacent nodes, and G_n^d denote the number of such sets. Here, $N_n^{F+} \cup N_{ng}^{P+} = N_n^+$, and $N_n^{F+} \cap N_{ng}^{P+} = \emptyset$. Accordingly, let $\beta_{n,m}$ denote the percentage of flow from n to m out of the total outflow from n , where $m \in N_n^{F+}$, while let β_n^g be the percentage showing how much flow from n to the g th set of n 's group adjacent nodes. For example, in Fig. 7, $\beta_{1,2} = 10\%$, $\beta_1^1 = \beta_{1,\{3,4\}} = 40\%$, $\beta_1^2 = \beta_{1,\{5,6\}} = 50\%$.

When the maintenance operation i is performed, the corresponding shutdown will affect product flow in the network. Let N_i denote the set of nodes that have impacts on the incoming flow during the shutdown for i . Let ρ_n denote the flow impact factor of the shutdown for node n , and this information can be estimated from historical throughput data by specialized staff in the demand chain team.

All sets and parameters discussed above are summarized in Tables 1 and 2.

3.2 Objective function with decision variables

The problem is to determine the timings of all maintenance operations, which are key decision variables, denoted by x_{id} indicating whether operation i starts at period d or not. Apart from these binary variables, there are three classes of continuous variables: y_{mnd} define how much flow entering n from m during period d , z_{nd} define how much stock is left in n at the end of period d , and z_{nd}^δ define how much overstock is produced in n during period d . All variables described above are listed in Table 3.

The objective function is to maximize total throughput and minimize overstock over the entire time horizon, and it can be formulated as follows:

$$\max \sum_{d \in D} \sum_{n \in N_e^-} y_{ned} - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^\delta. \tag{1}$$

Table 1 Sets in the model

Sets	Description
I	Set of all assets
P	Set of stockpiles and tailings storage facility
$N = \{0\} \cup I \cup P \cup \{e\}$	Set of nodes in the network (0 represents the pits, e represents the ships)
$N_n^+ \subseteq N$	Set of nodes that receive the flow leaving from node n
$N_n^- \subseteq N$	Set of nodes that provide the flow to node n
$N_{SI} \subseteq N$	Set of nodes that have inflow capacity limit
$N_{SO}^g \subseteq N$	Set of the g th group of nodes that have outflow capacity limit
$N_C \subseteq N$	Set of nodes that have certain percentage of outflow to different nodes
$N_n^{F+} \subseteq N$	Set of the normal adjacent nodes of n
$N_{ng}^{P+} \subseteq N$	Set of the g th set of n 's group adjacent nodes
$N_i \subseteq N$	Set of nodes that have impact on the inflow when operation i is performed
E	Set of edges that connect two adjacent nodes
A	Set of maintenance operations
$A_M \subseteq A$	Set of operations requiring a major shutdown
$A_i \subseteq A$	Set of operations in which at most λ_i^{\max} operations can occur along with i
$B_i \subseteq A$	Set of operations in which at least λ_i^{\min} operations must be scheduled with i
$\mathcal{A}_{NO} \subseteq A \times A$	Set of pairs of operations that cannot overlap
D	Set of time periods over the planning horizon
$D_i \subseteq D$	Set of potential periods for the shutdown for maintenance operation i

3.3 Constraints

In this subsection, we will list all constraints considered in the model.

3.3.1 Time constraints

- Each maintenance operation must be scheduled.

$$\sum_{d \in D_i} x_{id} = 1, \quad \forall i \in A. \tag{2}$$

- Time constraints on the gap between a pair of operations (i, j) .

$$\sum_{d \in D} dx_{id} + G_{ij}^{\min} \leq \sum_{d \in D} dx_{jd} \leq \sum_{d \in D} dx_{id} + G_{ij}^{\max}, \quad \forall i, j \in A. \tag{3}$$

- Some maintenance operations cannot overlap with each other.

$$\sum_{t=\max\{1, d-d_j+1\}}^{\min\{d+d_i-1, |D|\}} x_{jt} \leq M_1(1 - x_{id}), \quad \forall (i, j) \in \mathcal{A}_{NO}, \quad \forall d \in D \tag{4}$$

where $M_1 = d_i + d_j - 1$.

Table 2 Parameters in the model

Parameters	Description
d_i	Duration of shutdown for operation i
r_i	Resources required in shutdown for operation i
R^{\max}	Max resources limits for the whole system
$G_{ij}^{\max}, G_{ij}^{\min}$	The max/min gap between start times of operation i and j
a_i^-, a_i^+	Time tolerance for operation i when operations in A_i are scheduled
b_i^-, b_i^+	Time tolerance for operation i when operations in B_i are scheduled
λ_i^{\max}	Maximum number of operations in A_i that can be maintained along with operation i
λ_i^{\min}	Minimum number of operations in B_i that must be scheduled with operation i
$y_{m,n,d}^{\max}$	Max flow capacity from node m to n during time period d
y_n^{\max}	Max product flow that go through node n per period
ρ_n	Flow impact factor of shutdown for node n
G^c	Number of groups of nodes that have outflow capacity limits
G_n^d	Number of sets of n 's group adjacent nodes
$\beta_{n,m}(\beta_n^g)$	The percentage of flow from node n to m (the g th set of n 's group adjacent nodes) out of the total outflow from n
$z_{n,0}$	Initial stock in storage n at the beginning of the time horizon
$z_{nd}^{\max}, z_{nd}^{\min}$	Max/Min live stock in storage n at period d
z_{nd}^{\max}	Overstock limit in storage n at period d
α_n	Penalty parameter on overstock in storage n

Table 3 Variables in the model

Variables	Type	Description
x_{id}	Binary	Whether the shutdown for operation i starts at period d , $i \in A$
y_{mnd}	Continuous	Product flow from m to n during period d , $m, n \in N$
z_{nd}	Continuous	Stock level in n at the end of period d , $n \in P$
z_{nd}^δ	Continuous	Overstock level in n during period d , $n \in P$

- Only limited numbers of maintenance operations in A_i can start along with operation i .

$$\sum_{j \in A_i} \sum_{t=\max\{1, d-a_i^-\}}^{\min\{d+a_i^+, |D|\}} x_{jt} \leq \lambda_i^{\max} + M_2(1 - x_{id}), \quad \forall i \in A, \quad \forall d \in D \quad (5)$$

where $M_2 = |A_i|(a_i^+ + a_i^- + 1) - \lambda_i^{\max}$.

- At least certain numbers of maintenance operations in B_i must be performed along with operation i .

$$\sum_{j \in B_i} \sum_{t=\max\{1, d-b_i^-\}}^{\min\{d+b_i^+, |D|\}} x_{jt} \geq \min\{\lambda_i^{\min}, |B_i|\}x_{id}, \quad \forall i \in A, \quad \forall d \in D. \quad (6)$$

Note that if $x_{id} = 0$ or $B_i = \emptyset$, the inequality is redundant since the left-hand side is non-negative. If $x_{id} = 1$ and $B_i \neq \emptyset$, then constraints (6) force at least λ_i^{\min} operations in B_i to be scheduled along with operation i .

3.3.2 Stock status constraints

- Conservation of product stock at each storage.

$$z_{nd} = z_{n(d-1)} + \sum_{m \in N_n^-} y_{mnd} - \sum_{m \in N_n^+} y_{nmd}, \quad \forall n \in P, \quad \forall d \in D \quad (7)$$

where the initial stock in n is z_{n0} which is an input parameter.

- The bound on the stock quantity.

$$z_{nd}^{\min} \leq z_{nd} \leq z_{nd}^{\max} + z_{nd}^{\text{omax}}, \quad \forall n \in P, \quad \forall d \in D. \quad (8)$$

- There exists sufficient stock to feed the corresponding assets.

$$z_{nd} \geq \sum_{m \in N_n^+} y_{nmd}, \quad \forall n \in P, \quad \forall d \in D. \quad (9)$$

- Constraints on the overstock in n during period d .

$$z_{nd}^\delta = \max\{0, z_{nd} - z_{nd}^{\max}\}, \quad \forall n \in P, \quad \forall d \in D. \quad (10)$$

Note that since the “max” term is non-linear, we linearize and replace it by introducing the following two constraints:

$$0 \leq z_{nd}^\delta \leq z_{nd}^{\text{omax}}, \quad \forall n \in P, \quad \forall d \in D. \quad (11)$$

$$z_{nd}^\delta \geq z_{nd} - z_{nd}^{\max}, \quad \forall n \in P, \quad \forall d \in D. \quad (12)$$

See Proposition 1 for the detailed proof.

3.3.3 Flow constraints

- Bound constraints on the product flow.

$$0 \leq y_{mnd} \leq y_{mnd}^{\max}, \quad \forall (m, n) \in E, \quad \forall d \in D. \quad (13)$$

- The inflow to an asset will always be equal to the outflow (recall our assumption in Sect. 2 that processing assets cannot store any product).

$$\sum_{m \in N_n^-} y_{mnd} = \sum_{m \in N_n^+} y_{nmd}, \quad \forall n \in I, \quad \forall d \in D. \tag{14}$$

- The flow is affected at a predefined level when shutdown is performed for maintenance operation i .

$$y_{mnd} \leq y_{mnd}^{\max} \left(1 - \rho_n \sum_{t=\max\{1, d-d_i+1\}}^d x_{it} \right), \quad \forall i \in A, \quad \forall n \in N_i, \quad \forall m \in N_n^-, \quad \forall d \in D. \tag{15}$$

Note that constraints (15) represent three types of restrictions on the flow. If d is within the time window of the shutdown for i , then the inflow to an affected asset n will be not greater than $y_{mnd}^{\max}(1 - \rho_n)$. In most cases, ρ_n is 1, which enforces the flow to be zero during shutdown. For some assets that have redundancy, i.e. $0 < \rho_n < 1$, the flow will be at certain percentage of the maximum flow capacity. If there is no shutdown during period d , then we only have flow capacity y_{mnd}^{\max} to control the inflow to node n . We call these constraints "linking constraints" that build connections between shutdown and flow.

- Flow cannot exceed capacity limits on some shared equipment.

$$\sum_{m \in N_n^-} y_{mnd} \leq y_n^{\max}, \quad \forall n \in N_{SI}, \quad \forall d \in D. \tag{16}$$

$$\sum_{n \in N_{SO}^g} \sum_{m \in N_n^+} y_{nmd} \leq \max_{n \in N_{SO}^g} \{y_n^{\max}\}, \quad \forall g \in G^c, \quad \forall d \in D. \tag{17}$$

- The flow from the node in N_C to its adjacent downstream nodes follows certain predefined percentage distribution.

$$y_{nmd} = \beta_{n,m} \sum_{u \in N_n^+} y_{nud}, \quad \forall n \in N_C, \quad \forall m \in N_n^{F+}, \quad \forall d \in D. \tag{18}$$

$$\sum_{m \in N_{ng}^{P+}} y_{nmd} = \beta_n^g \sum_{u \in N_n^+} y_{nud}, \quad \forall n \in N_C, \quad \forall g \in G_n^d, \quad \forall d \in D. \tag{19}$$

3.3.4 Resources constraints

- The resources for the whole system must be sufficient.

$$\sum_{d \in D} \sum_{i \in A \setminus A_M} r_i x_{id} \leq R^{\max}. \tag{20}$$

3.3.5 Variable domains

- Non-negativity constraints.

$$y_{mnd} \geq 0, \quad \forall m, n \in N, \quad \forall d \in D. \quad (21)$$

$$z_{nd}, z_{nd}^{\delta} \geq 0, \quad \forall n \in P, \quad \forall d \in D. \quad (22)$$

- Binary constraints.

$$x_{id} \in \{0, 1\}, \quad \forall i \in A, \quad \forall d \in D. \quad (23)$$

The formulations of most constraints are self-explanatory, except the constraints (11)–(12) to control overstock. We now give a further explanation with mathematical justification.

Proposition 1 Constraints (11)–(12) correctly represent the formulation of overstock.

Proof Recall the definition of overstock in a stockyard, that is the ore which exceeds the maximum level of live stock. Therefore, overstock can be formulated as

$$z_{nd}^{\delta} = \max \{0, z_{nd} - z_{nd}^{\max}\},$$

which can be rewritten as

$$z_{nd}^{\delta} = \begin{cases} 0, & \text{if } z_{nd} \leq z_{nd}^{\max}; \\ z_{nd} - z_{nd}^{\max}, & \text{if } z_{nd} > z_{nd}^{\max}. \end{cases} \quad (24)$$

Clearly, the overstock z_{nd}^{δ} is non-negative and within the corresponding upper bound z_{nd}^{\max} , which is shown in constraint (11).

If $z_{nd} \leq z_{nd}^{\max}$, then constraint (12) is redundant, since z_{nd}^{δ} cannot be negative. Besides, we aim to minimize overstock in the objective function (1), thus the value of z_{nd}^{δ} should be as small as possible, which follows that the optimal value satisfying constraint (11) must be 0.

On the contrary, if $z_{nd} > z_{nd}^{\max}$, from constraints (11)–(12), we have $z_{nd} - z_{nd}^{\max} \leq z_{nd}^{\delta} \leq z_{nd}^{\max}$, then the optimal value (minimization) of z_{nd}^{δ} must be $z_{nd} - z_{nd}^{\max}$, which completes the proof. \square

Formally, the complete MILP model involved in this paper can be stated as follows: choose the decision variables x_{id} , y_{mnd} , z_{nd} and z_{nd}^{δ} to maximize the objective function (1) subject to constrains (2)–(23).

4 Solution strategies

The proposed time-indexed MILP model is challenging to solve for large scale industry instances. To speed up the process, we designed an algorithm which is a combination of Benders decomposition (Benders 1962) and Lagrangian relaxation (Held and

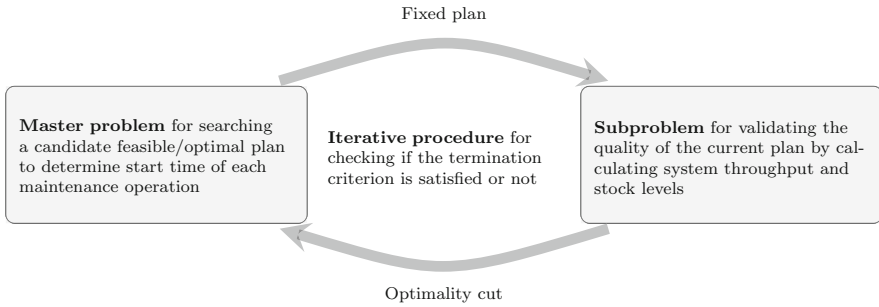


Fig. 8 A schematic overview of the decomposition approach

Karp 1971; Fisher 2004). We call this the Benders Lagrangian Decomposition (BLD) method. In this section, we present the proposed decomposition approach in detail and give pseudocode for the entire algorithm.

Recalling that in our long-term maintenance planning problem, we have two types of decisions to make: (i) scheduling decisions which determine the start time of each maintenance operation, and (ii) continuous decisions which provide the value of product flow along each edge connecting two adjacent nodes and inventory level of each storage unit. There are independent sets of constraints to control the feasible solution space of these two types of variables. Linking constraints (15) connect these two types of variables and establish the relationship between the maintenance schedule and product flow. Such a special structure makes it possible to apply partitioning approaches to effectively solve the model. In addition, note that the objective function only depends on continuous variables. Therefore, we can allocate all binary variables into the so-called master problem for searching for a feasible maintenance schedule and check how this fixed schedule affects product flow and stock in the subproblem. We then derive appropriate cuts that will be added to the master problem to obtain a better schedule in a subsequent iteration. Instead of solving the classical dual of the subproblem, we price out only the linking constraints in the subproblem objective function and make use of the corresponding Lagrangian relaxation to generate optimality cuts. The process repeats until the optimal schedule is achieved. Figure 8 illustrates the scheme for iteratively searching for the optimal solution. Readers can refer to Geoffrion (1972), Sahinidis and Grossmann (1991), Song and Cheng (2022) for a convergence analysis and more applications. In the following, we discuss the procedure in detail mathematically.

We denote the entire model by the following simplified formulation:

$$\max_{x,s} \{c^T s : x \in X, As + Bx \leq b, s \in S\} = \max_{x \in X} \left\{ \max_{s \in S} \{c^T s : As \leq b - Bx\} \right\}, \quad (25)$$

where $x = x_{id}, s = (y_{md}, z_{nd}, z_{nd}^\delta)$ are decision variables, constraints $As + Bx \leq b$ represent linking constraints (15) that connect integer variables and continuous variables, X, S are feasible regions defined by time (resource) constraints ((2)–(6), (20)) and all remaining constraints. The inner maximization problem is a subproblem whose purpose is to get the maximum throughput and minimum overstock for a given main-

tenance schedule x . Let $\theta(x)$ denote the optimal objective value of the subproblem. The original model is then equivalent to:

$$\max_{x, \theta} \{ \theta : x \in X, \theta \leq \theta(x) \}, \tag{26}$$

where $\theta \in \mathbb{R}$ is a continuous variable. Note that, since there exists capacity limits for flow and stock, we have

$$\theta \leq \theta(x) = \sum_{d \in D} \sum_{n \in N_e^-} y_{ned} - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^\delta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned} \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^{\max}. \tag{27}$$

We impose a bound on θ and define the initial master problem (IMP) as follows:

$$\begin{aligned} \text{(IMP)} : \quad & \max_{x_{id}, \theta} \theta \\ & \text{s.t. constraints : (2) – (6), (20)} \\ & \theta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^{\max} \\ & x_{id} \in \{0, 1\}, \end{aligned} \tag{28}$$

from which we can obtain a feasible schedule determining start times for all maintenance operations.

Once a schedule \bar{x}_{id} is obtained, we then solve the following subproblem:

$$\begin{aligned} \text{(SP}(\bar{x}_{id})) : \quad & \max_{y_{mnd}, z_{nd}, z_{nd}^\delta} \sum_{d \in D} \sum_{n \in N_e^-} y_{ned} - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^\delta \\ & \text{s.t. } y_{mnd} \leq y_{mnd}^{\max} (1 - \rho_n \sum_{t=\max\{1, d-d_i+1\}}^d \bar{x}_{it}), \\ & \forall i \in A, \quad \forall n \in N_i, \quad \forall m \in N_n^-, \quad \forall d \in D \\ & \text{constraints : (7)–(14), (16)–(19), (21)–(23)}. \end{aligned} \tag{29}$$

Note that problem $\text{SP}(\bar{x}_{id})$ is a linear continuous programming problem with $y_{mnd}, z_{nd}, z_{nd}^\delta$. $\theta(\bar{x}_{id})$ provides a lower bound of the original problem, as all binary variables \bar{x}_{id} have a fixed value. It is obvious that, no matter how maintenance is scheduled, we always have $\{y_{mnd} = 0, z_{n0}, z_{n0}^\delta\}$ in the feasible region of problem (SP). On the other hand, refer to (27), $\theta(\bar{x}_{id})$ has a bound. Hence, the subproblem is feasible and bounded. Only optimality cuts will be generated and added to the master problem for finding a better schedule.

Now we explain how to derive optimality cuts with respect to x_{id} from the incumbent subproblem. We employ Lagrangian relaxation on the subproblem by pricing out linking constraints (15) into the objective function with a non-negative multiplier μ , and the Lagrangian dual problem is:

$$\begin{aligned}
 \text{(LDP)} : \quad & \min_{\mu} \max_{y_{mnd}, z_{nd}, z_{nd}^{\delta}} \left\{ \sum_{d \in D} \sum_{n \in N_e^-} y_{ned} - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^{\delta} \right. \\
 & \left. - \mu^T \left(y_{mnd} - y_{mnd}^{\max} \left(1 - \rho_n \sum_{t=\max\{1, d-d_i+1\}}^d \bar{x}_{it} \right) \right) \right\}.
 \end{aligned} \tag{30}$$

Then, we can derive an optimality cut:

$$\theta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^* - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^{\delta*} - (\mu^*)^T \left(y_{mnd}^* - y_{mnd}^{\max} \left(1 - \rho_n \sum_{t=\max\{1, d-d_i+1\}}^d x_{it} \right) \right), \tag{31}$$

where $y_{mnd}^*, z_{nd}^{\delta*}, \mu^*$ are optimal solutions of problem (LDP). The type of cut (31) will be added to problem (IMP) and the updated master problem has the following form:

$$\begin{aligned}
 \text{(MP)} : \quad & \max_{x_{id}, \theta} \theta \\
 & \text{s.t. constraints : (2)–(6), (20)} \\
 & \theta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^* - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^{\delta*} \\
 & \quad - (\mu^*)^T \left(y_{mnd}^* - y_{mnd}^{\max} \left(1 - \rho_n \sum_{t=\max\{1, d-d_i+1\}}^d x_{it} \right) \right) \\
 & \theta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^{\max} \\
 & x_{id} \in \{0, 1\}.
 \end{aligned} \tag{32}$$

In the new master problem, we can look for another schedule x_{id} and update θ that is an upper bound of the original problem. The procedure repeats until the stopping criterion is satisfied. In our case, since the subproblem is solved by continuous linear programming, we can update multipliers by obtaining the optimal dual variables associated with linking constraints in the subproblems. The pseudocode is described more precisely in Algorithm 1.

5 Computational results and discussion

To validate the mathematical formulation proposed in Sect. 3, we implemented the MILP model in Python, using the optimization solver CPLEX v12.10 with up to 8 threads. All computations were conducted on a HP laptop with an Intel Core i7-8565U CPU running at 1.80 GHz and 16G of RAM. Our test problem is based on realistic data provided by an integrated iron ore mining company in Western Australia. Our testing process was split into three parts. In the first part, we tested the model on

Algorithm 1: BLD Algorithm for Maintenance Planning Problem

```

1 Set  $U \leftarrow +\infty, L \leftarrow -\infty, k \leftarrow 0$ 
2 while  $U \neq L$  and  $k < k_{max}$  do
3   Solve MP( $k$ ) (IMP at the beginning)
4   if MP( $k$ ) infeasible then
5     | Algorithm stops and return "the problem is infeasible"
6   else
7     | Obtain optimal solution  $(\bar{x}_{id}^k, \bar{\theta}^k)$ 
8   end if
9   Update the upper bound:  $U \leftarrow \min\{U, \bar{\theta}^k\}$ 
10  Solve LDP( $k$ ) with fixed  $\bar{x}_{id}^k$ 
11  Obtain optimal solution  $y_{mnd}^k, z_{nd}^{\delta k}$  and the multipliers  $\mu^k$ 
12  Update the lower bound:  $L \leftarrow \max\left\{L, \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^k - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^{\delta k}\right\}$ 
13  Add optimality cut
    
$$\theta \leq \sum_{d \in D} \sum_{n \in N_e^-} y_{ned}^k - \sum_{d \in D} \sum_{n \in P} \alpha_n z_{nd}^{\delta k} - (\mu^k)^T \left( y_{mnd}^k - y_{mnd}^{\max} \left( 1 - \rho_n \sum_{i=\max\{1, d-d_i+1\}}^d x_{it} \right) \right)$$

    to MP( $k$ ) to obtain MP( $k+1$ )
14   $k \leftarrow k + 1$ 
15 end while

```

a real case study. In the second part, we performed sensitivity analysis by running the model with adjusted parameters to gain practical insights such as how gaps between major shutdowns affect throughput, and the key factors affecting overstock. In the third part, we generated 10 random scenarios to test the computational performance further and provided a comparison between CPLEX and BLD in terms of objective values at certain running times.

5.1 Problem set-up and case study

Our aim is to determine an optimal long-term maintenance plan for all assets in the iron ore mining operation in Fig. 2 and provide an upper limit of throughput.

Note that some simplifications can be made in Fig. 2. First, since the red assets are maintained while their corresponding subsystems are shut down, we do not require determining separate times for these assets, and hence they can be omitted for the purpose of optimization. Second, the two groups of five parallel assets in subsystem 2 are always maintained one by one in the same modular shutdown. Hence, we can replace each of these groups with a "super-node" that requires modular shutdowns. Based on this, we have a simplified version of the system as shown in Fig. 9, in which all nodes are labelled by integers.

Compared with the previous version, all red assets are omitted, except nodes 12 and 13 in subsystem 3 to connect the processing plant and the port. Nodes 4 and 5 represent the two groups of five parallel assets in subsystem 2. The symbols have the same meaning as in Fig. 2.

5.1.1 Data set for case study

In the case study, we consider a 1-year time horizon, which is divided into time periods of 1 day (365 periods in total). Each subsystem requires 4 major shutdowns per year

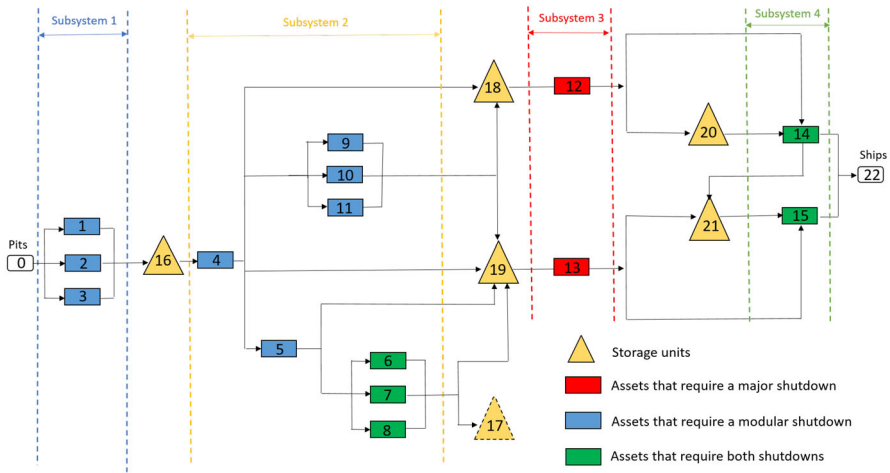


Fig. 9 A simplified diagram for the example iron ore mining operation

Table 4 Number of modular shutdowns required for each asset in Fig. 9

	Asset														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Modular shutdowns	8	8	8	4	4	1	1	1	8	8	8	-	-	4	4

and the number of modular shutdowns required for each blue and green asset is given in Table 4. In total there are 91 maintenance operations required in the simplified network.

Recalling Table 2, the input data can be categorised into three groups: data for maintenance operations, data for product flow and data for storage. Note that all data is provided by our iron ore industry partner, except the penalty parameter α_n on overstock. We set the penalty parameter on overstock for all storage units to be 1, i.e., $\alpha_n = 1, \forall n \in P$. In what follows, we present the rest of the data in detail.

For a maintenance operation i , the data set defines the duration d_i , the resource requirement r_i and the potential time window D_i . Note that in the case study, we only have restrictions on the time windows for major shutdowns as they must be scheduled during certain seasons, but allow modular shutdowns to occur at any time over the whole time horizon. For a pair of operations (i, j) , the minimum and maximum gaps (G_{ij}^{\min} and G_{ij}^{\max}) are defined by the following business rules.

- Consecutive maintenance operations for the same asset (subsystem) must be within plus or minus 3 days of the maintenance frequency.
- The blue “super-assets” 4 and 5 are always maintained at the same time, and red assets 12 and 13 should be maintained at the same time as the major shutdowns of subsystem 3.
- Modular shutdowns of green asset 14 at the port must start at least 7 days before the corresponding modular shutdowns for asset 15.

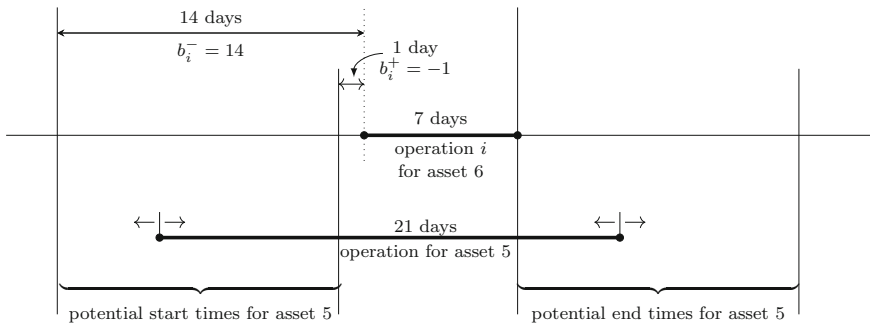


Fig. 10 Parameters b_i^-, b_i^+ are chosen so that asset 6 operations occur during the modular shutdowns of asset 5

- The subsystem shutdowns occur consecutively with precisely 1-day gap between each.

There are two additional business rules that require the inputs $a_i^-, a_i^+, \lambda_i^{\max}, b_i^-, b_i^+$ and λ_i^{\min} as follows.

(1) The modular shutdowns of green assets 6, 7 and 8 must occur during the modular shutdowns of asset 5. To enforce this, for each operation i corresponding to assets 6, 7 and 8, we let the set B_i contain all the operations for asset 5, and choose the parameters b_i^-, b_i^+ and λ_i^{\min} as follows:

$$b_i^- = 14, b_i^+ = -1, \lambda_i^{\min} = 1.$$

Since the duration of i is 7 days and the duration of asset 5’s modular shutdowns are 21 days, these constraints ensure that the maintenance for i can be completed during the modular shutdown of asset 5 as shown in Fig. 10.

(2) Each modular shutdown of asset 5 only allows at most one of assets 6, 7 and 8 to be maintained along with it. To achieve this, for each operation i corresponding to asset 5, we let the set A_i contain all the operations for assets 6, 7 and 8, and choose the parameters a_i^-, a_i^+ and λ_i^{\max} as follows:

$$a_i^- = 0, a_i^+ = d_i - 1, \lambda_i^{\max} = 1,$$

where d_i is the duration of operation i .

The maximum flow capacities $y_{mnd}^{\max}, y_n^{\max}$ were determined from historical data. Flow distribution percentages are listed in Table 5. The first six columns in Table 5 report the values of the parameters $\beta_{n,m}$. The last two columns show the flow distribution percentages β_n^g from node 4 to group {9, 10, 11} and from node 5 to group {6, 7, 8}. With assets 4, 5, 14 and 15, there is redundancy, so maintenance operations on these assets do not completely stop product flow. The impact factor data for these nodes are

$$\rho_4 = \rho_5 = 0.1, \rho_{14} = \rho_{15} = 0.05.$$

Table 5 Flow distribution percentages

n	m							
	5	17	18	19	21	22	{6, 7, 8}	{9, 10, 11}
4	30%	–	25%	35%	–	–	–	10%
5	–	–	–	20%	–	–	80%	–
6	–	80%	–	20%	–	–	–	–
7	–	80%	–	20%	–	–	–	–
8	–	80%	–	20%	–	–	–	–
9	–	–	80%	20%	–	–	–	–
10	–	–	80%	20%	–	–	–	–
11	–	–	80%	20%	–	–	–	–
14	–	–	–	–	13%	87%	–	–

The maximum flow capacities y_{mnd}^{\max} , y_n^{\max} were determined from historical data. Flow distribution percentages are listed in Table 5

For all other nodes, the flow impact factor $\rho_n = 1$, meaning that flow completely stops during maintenance operations for these nodes.

For all storage units $n \in P$ (the triangles in Fig. 9), the maximum and minimum stock capacities z_{nd}^{\max} , z_{nd}^{\min} were obtained from historical data. Note that, we assume the tailings (node 17) capacity is unlimited because this part is irrelevant to our model and its capacity is not a bottleneck in practice. The initial stock levels z_{n0} at the beginning of the time horizon were estimated based on historical data and the quantities were chosen as

$$z_{17,0} = 0 \text{ kt}, z_{16,0} = z_{18,0} = z_{19,0} = 300 \text{ kt}, z_{20,0} = z_{21,0} = 500 \text{ kt}.$$

5.1.2 Results for case study

Although the simplified network in Fig. 9 has far fewer variables and constraints than the full network in Fig. 2, it still involves 33,215 binary variables, 18,250 continuous variables and 666,311 constraints. This is a large number of variables, especially binary variables, which makes the model hard to solve in a short time. We first solved the model using the default CPLEX, which found the first feasible solution in about 1 h and obtained another slightly better solution with increased throughput after 2 h of computation. To accelerate the computation, we applied the Benders Lagrangian decomposition (BLD) algorithm (see Sect. 4). The numerical results show that the proposed algorithm outperforms the default Branch and Cut method in CPLEX. Our algorithm can achieve a feasible schedule faster, and succeed in finding better schedules with more throughput than the default CPLEX. See Fig. 11 for a visualization of the detailed comparison after 2 h of computation.

A Gantt chart for the optimised schedule using the BLD method is given in Fig. 12a. The vertical axis denotes all assets and subsystems, and the horizontal axis denotes the time horizon. Pink bars represent major shutdowns, and the other coloured bars

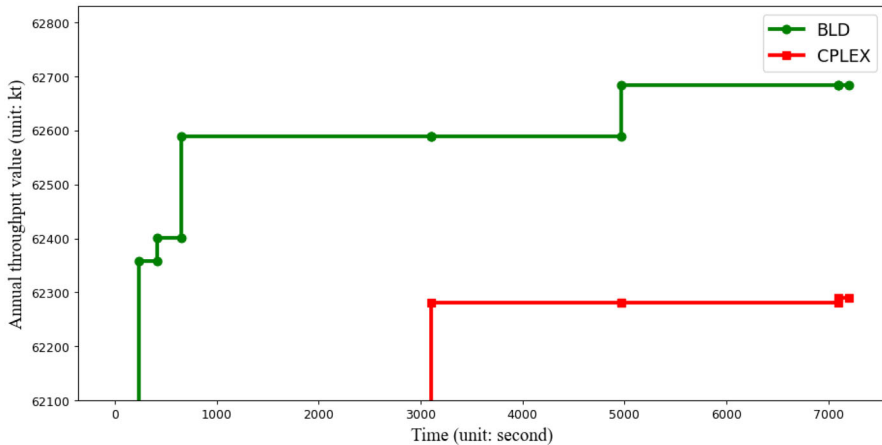


Fig. 11 A comparison of the annual throughput improvement of CPLEX and the proposed BLD method in 2 h of computation

represent other shutdown operations with the same colours as in Fig. 9. The plan in Fig. 12a indicates which units need to be maintained on which days.

Ideally, maintenance on assets within the same subsystem should be performed in the same time interval, so that they can be taken back online at the same time to reduce production loss outside this time interval. However, sometimes this is impossible due to lack of resources. Our model is designed to find the best sequence of shutdowns, taking into account the trade-off between maintenance and throughput. The model minimizes disruptions to throughput and chooses the maintenance schedule that increases throughput by as much as possible. To give a comparison, we recorded the first feasible solution from BLD method. This solution yields a throughput value of 62,358 kt and is shown in Fig. 12b. As we can see, compared with the plan with throughput 62,683 kt in Fig. 12a, both of them satisfy all maintenance requirements, but the timings and sequence of maintenance operations are different. That is, different combinations of maintenance operations will lead to different throughput. The purpose here is to remove maintenance as a bottleneck to production by finding the best shutdown plan that maximises potential throughput. Note that the maximized throughput is a relative quantity to optimize the maintenance plan rather than actual production. Forecasting real production will involve more uncertainties (e.g. extreme weather factors, unplanned downtime on assets, etc.) and will need another more complex optimization model, which is beyond the scope of this paper.

Asset n must be maintained at a frequency f_n . However, this does not mean that the maintenance must be performed exactly on the due date and it is allowed to be advanced or delayed if it is of benefit to do so. For example, for a pair of adjacent maintenance operations (i, j) on the same asset n , operation j is considered to be advanced if the start time of j is less than the sum of start time of i and maintenance



Fig. 12 Maintenance plans for the integrated mining system based on Fig. 9

frequency f_n . More generally:

$$\sum_{d \in D} dx_{jd} - \left(\sum_{d \in D} dx_{id} + f_n \right) \begin{cases} < 0, & \text{maintenance operation } j \text{ is advanced,} \\ = 0, & \text{maintenance operation } j \text{ is on time,} \\ > 0, & \text{maintenance operation } j \text{ is delayed.} \end{cases} \quad (33)$$

Note that, there are limits on the extent to which maintenance can be advanced or delayed, since parameters G_{ij}^{\min} and G_{ij}^{\max} restrict the maximum number of days that maintenance j can be brought forward and backward respectively. Table 6 reports the maintenance deviations over the whole time horizon. In this problem, most of maintenance operations are taken in advance or delayed, which is reasonable because the dynamic stock status needs to be adjusted to respect storage capabilities and overstock avoidance. Furthermore, about 80% of maintenance activities are brought forward instead of being delayed. This is consistent with practical requirements, as frequencies are determined by experienced maintenance schedulers, and it is undesirable to delay maintenance too much due to safety risk and system reliability considerations.

Table 6 Maintenance operation deviations over the whole time horizon

Operations advanced			Operations delayed			Operations on time		
Total	Average	Percentage	Total	Average	Percentage	Total	Average	Percentage
58	2.7 days	80.5%	12	2.3 days	16.7%	2	0 days	2.7%

5.2 Sensitivity analysis

In this section, we discuss the sensitivity of the shutdown plan with respect to subsystem shutdown gaps, overstock penalty and individual product limits. All scenarios were run with a 1 h time limit, using the same network as the case study.

5.2.1 Sensitivity to the gap between subsystem shutdowns

To analyse how the gap between different major shutdowns affects final throughput, we performed simulations in which the values of parameters G_{ij}^{\min} and G_{ij}^{\max} were changed for the consecutive major shutdowns: subsystem 1 and subsystem 2, subsystem 2 and subsystem 3, and subsystem 3 and subsystem 4. The baseline is a gap of 1 day as used in the previous section. The scenarios were generated either by having a constant gap (0, 3, 5, or 7 days) between each pair of consecutive shutdowns, or by changing one gap to be 7 days and keeping the other gaps to the baseline value of 1 day. The first two columns of Table 7 show the different combinations of gap values, with changes to the baseline shown in bold.

Numerical results in Table 7 show that the gap between the major shutdowns has a significant impact on system throughput and the throughput decreases as the gap between major shutdowns increases. The last two columns give the throughput values and the percentage change compared with the baseline throughput. For example, when the gap increases from 1 to 3 days, the throughput decreases by about 2 million tons, which is a significant amount for this mining operation. The effects on throughput can potentially be alleviated by changing the capacities of storage units such as buffers. The results show that major shutdowns of different subsystems should overlap as much as possible to reduce the loss from process interruptions. Furthermore, we can observe that increasing the gap to 3 days for all pairs of consecutive major shutdowns leads to more reduction on the throughput, compared with changing the gap to 7 days for only one pair of consecutive major shutdowns. This suggests that we should avoid increasing the gaps between all consecutive major shutdowns to reduce the production loss.

5.2.2 Sensitivity to overstock penalty

In practice, managing inventory at optimal operational levels is crucial. Changing the penalty imposed on overstock through the coefficient α_n will change the optimal maintenance strategy. In this section, we present results for different values of $\alpha = \alpha_n, \forall n \in P$ (all stockyards have the same overstock penalty).

Table 7 Effect of major shutdown gap length on throughput

Subsystem pair	Gap (days)	Throughput (kt)	Throughput change (%)
(1, 2), (2, 3), (3, 4)	1, 1, 1	62,281.3	0 (baseline)
(1, 2), (2, 3), (3, 4)	0, 0, 0	63,571.0	+2.07
(1, 2), (2, 3), (3, 4)	3, 3, 3	59,922.0	-3.79
(1, 2), (2, 3), (3, 4)	5, 5, 5	57,462.4	-7.74
(1, 2), (2, 3), (3, 4)	7, 7, 7	56,427.7	-9.40
(1, 2), (2, 3), (3, 4)	7, 1, 1	60,148.8	-3.42
(1, 2), (2, 3) , (3, 4)	1, 7 , 1	60,443.4	-2.95
(1, 2), (2, 3), (3, 4)	1, 1, 7	61,489.8	-1.27

The throughput change is calculated relative to the baseline throughput when all gaps are 1 day. Bold numbers indicate there exists changes to the baseline gaps

Table 8 Numerical results for different overstock penalties

α	Overstock (kt)						
	Total	Node 16	Node 17	Node 18	Node 19	Node 20	Node 21
5	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1.5	53.4	0	0	0	53.4[1]	0	0
1	83.3	0	0	0	83.3[1]	0	0
0.5	162.1	90[2]	0	0	72.1[2]	0	0
0.1	784.6	635.7[18]	0	20.0[2]	129.0[6]	0	0
0.01	10,888.8	8152.5[107]	0	1420.2[13]	1316.1[11]	0	0

The values in square brackets represent the number of days with overstock, and the values next to the brackets represent the amount of overstock in each storage unit over the time horizon

The numerical results are summarized in Table 8. From the first two columns, we can see that the total overstock increases as the penalty value α decreases. Note that there is no overstock when $\alpha \geq 2$. For a further comparison, in the other columns, we record the total amount of overstock in each storage unit (node 16 to node 21) across the time horizon, and the number of time periods when there is overstock (number of days in square brackets). Note that node 17's capacity is unlimited, and thus there is no overstock in node 17 no matter what value α takes. Although we do have stock capacity limits for nodes 20 and 21 in the port, there is no overstock in all test scenarios, which implies that the current capacity for nodes 20 and 21 is sufficient to cater for dynamic stock levels. The nodes 16, 18 and 19 produce overstock easily, implying that the capacity of such stockyards needs to be adjusted to improve the flow rate. To further explore the trade-off between throughput and overstock, we need to take into account the actual cost of regaining overstock in terms of personnel and equipment time, to define better values for the parameter α . We did not have access to this data, but it is something to explore in future work.

5.2.3 Sensitivity to individual product limits

In practice, the split between lump and fines product is around 35–65% based on historical data. Accordingly, the case study included limits on the lump and fines production of 70kt and 130kt shipped per day. This gives consistent lump and fines product breakdown every day. However, since there are stockyards acting as buffers in the port, it is possible to vary this ratio to have a more flexible shipping schedule to increase throughput and satisfy additional market demand. In this sensitivity study, we remove the individual limits for lump and fines product, but keep the overall limit of 200 kt of product to the ships every day to assess how moving different ratios of lump and fines on different days can increase throughput.

Figure 13 shows a comparison of the results with and without the specific daily product limits. The lines in Fig. 13a, b depict the final lump, fines and total product flows. It can be observed that without the individual product limits, the final flows are more variable. Specifically, when individual limits are imposed, the daily total product loaded onto the ships never reaches the maximum value of 200 kt, represented by the orange dashed line in Fig. 13. On the other hand, when only the total limitation of 200 kt is maintained without individual limits, the total amount reaches the maximum capacity on 172 days, which accounts for approximately half of the entire time horizon. This implies we can get more throughput when imposing more flexible product limits.

5.3 Random scenarios

To test the computational performance of BLD method (recall Sect. 4) further, we ran simulations for 10 additional random problem scenarios.

5.3.1 Data set for random scenarios

We considered the same network as in the case study. Therefore, all data inputs for maintenance operations except for duration are kept the same. For each maintenance operation i , we generated the new duration data by randomly choosing an integer uniformly in the interval bounded by 60% and 140% of the original duration d_i (rounded down and up to the nearest integer).

For the flow data, we kept the flow distribution percentages the same as the case study, but randomly chose the flow bottleneck limits y_n^{\max} (integers) from the following ranges:

$$y_{16}^{\max} \in [230, 350], \quad y_{18}^{\max} = y_{19}^{\max} \in [180, 300], \quad y_{20}^{\max} = y_{21}^{\max} \in [180, 300], \\ y_{22}^{\max} \in [180, 300].$$

The maximum flow capacities y_{mnd}^{\max} were defined by the product of flow bottleneck limits y_n^{\max} and flow distributions $\beta_{n,m}$, β_n^g . The flow impact factors for assets 4, 5, 14, 15 were randomly selected from the range (0, 1), i.e.,

$$\rho_4, \rho_5, \rho_{14}, \rho_{15} \in (0, 1).$$

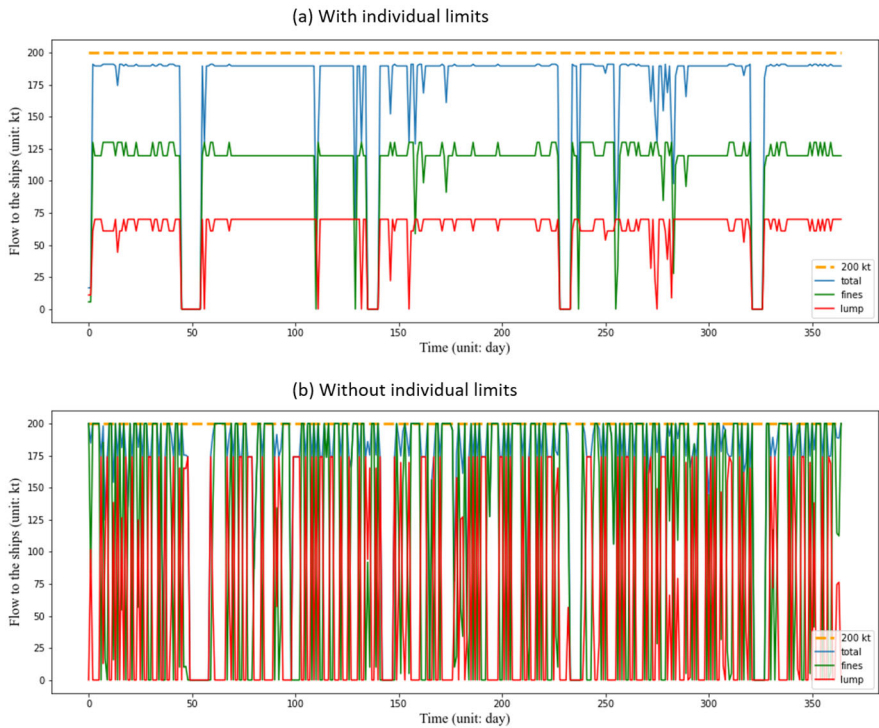


Fig. 13 A comparison of final product flows associated with and without individual product limits. The green line represents the fines product, the red line represents the lump product and the blue line represents the total product flow. The orange dashed line represents the maximum flow (200 kt)

Table 9 The specific ranges for generating random stock data

Node	$z_{n,0}$ (kt)	z_{nd}^{\max} (kt)	z_{nd}^{\min} (kt)	z_{nd}^{omax} (kt)
16	[50, 400]	[400, 500]	[50, 100]	[100, 200]
17	[50, 400]	–	[50, 100]	[400, 600]
18	[50, 400]	[400, 500]	[50, 100]	[400, 600]
19	[50, 400]	[400, 500]	[50, 100]	[400, 600]
20	[50, 700]	[700, 900]	[50, 100]	[700, 900]
21	[50, 700]	[700, 900]	[50, 100]	[700, 900]

For other nodes, the flow impact factor is 1, the same as the case study. Note that all non-integer random numbers are rounded to 2 decimal places.

The stock data were generated by selecting random integers from the corresponding range in Table 9. Note that, since we assume the tailing storage (node 17) has unlimited capacity, we set the maximum stock $z_{17,d}^{\max}$ to be a sufficiently large number 1×10^8 (kt).

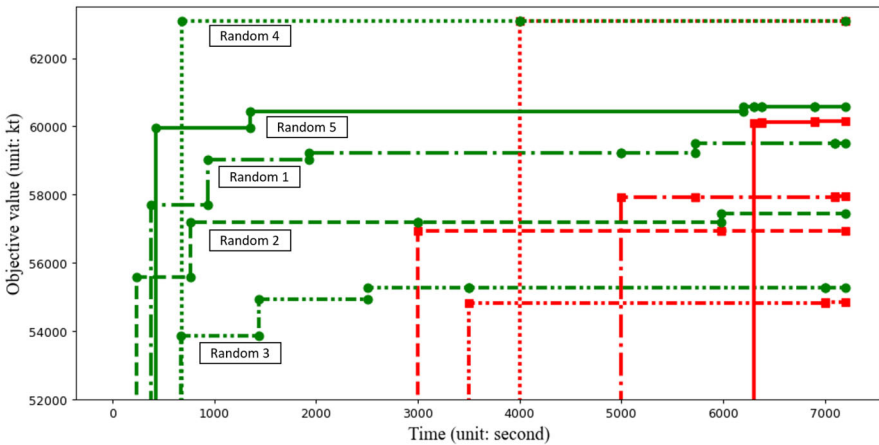


Fig. 14 The performance of CPLEX and the proposed BLD method for 5 random scenarios in 2h of computation. The five scenarios are differentiated by five line styles. The green, red colours represent the results from BLD and CPLEX respectively

5.3.2 Results for random scenarios

We ran simulations for each scenario using the CPLEX and BLD methods. The time limit was set as 2h for all scenarios. Since we consider the same network as the case study, each random problem involves the same number of variables and constraints.

Table 10 provides a comparison of objective values at certain times (300s, 600s, 1000s, 3000s, 5000s, 7200s). In all of the random scenarios, CPLEX failed to find a feasible solution within about 1h, but BLD method could always yield a feasible solution in a few hundred seconds. When applying BLD method, the final objective values after 2h computation are larger than the results from CPLEX alone. Note that, in the fourth scenario, both CPLEX and BLD can obtain the optimal solution within 2h of computation. However, by using BLD, such a solution can be achieved earlier at 680s, instead of 4000s. Additionally, in the ninth scenario, CPLEX cannot find a feasible solution after 2h, while BLD can obtain a solution after 221s. To better understand the performance of our method, the first five scenarios are visualized in Fig. 14.

To evaluate the quality of the feasible solutions generated by the BLD method, we calculated two gaps as shown in the last two columns of Table 10. Gap_1 is the original gap obtained from the BLD method, while Gap_2 is the gap obtained when we use the upper bound from CPLEX. Numerical results in Table 10 show that Gap_2 is better than Gap_1 and less than 10% on average. This means that the feasible solutions from BLD method are of high quality. The noticeable difference between Gap_1 and Gap_2 suggests that there is room for strengthening the optimality cuts in the BLD method.

Table 10 Numerical results for 10 randomly-generated scenarios

Scenario	Method	Objective value (kt) vs time (s)					Gap1 (%)	Gap2 (%)
		300 (s)	600 (s)	1000 (s)	3000 (s)	5000 (s)		
1	CPLEX	-	-	-	-	57,934.0	57,960.0	-
	BLD	-	57,702.3	59,013.1	59,231.2	59,231.2	59,497.9	18.50
2	CPLEX	-	-	-	56,936.2	56,936.2	56,936.2	56,936.2
	BLD	55,573.0	55,573.0	57,195.0	57,195.0	57,195.0	57,446.7	21.31
3	CPLEX	-	-	-	-	54,814.3	54,855.1	-
	BLD	-	-	53,855.5	55,283.9	55,283.9	55,283.9	24.27
4	CPLEX	-	-	-	-	63,078.8	63,078.8	-
	BLD	-	-	63,078.8	63,078.8	63,078.8	63,078.8	11.88
5	CPLEX	-	-	-	-	-	60,140.5	-
	BLD	-	59,969.9	59,969.9	60,439.6	60,439.6	60,574.4	16.99
6	CPLEX	-	-	-	-	62,313.1	62,313.1	-
	BLD	62,148.7	62,148.7	62,838.5	62,838.5	62,838.5	62,838.5	13.86
7	CPLEX	-	-	-	-	-	52,252.0	-
	BLD	51,053.5	51,053.5	51,053.5	51,053.5	52,504.6	52,504.6	28.07
8	CPLEX	-	-	-	-	-	50,982.1	-
	BLD	47,251.7	51,067.4	51,067.4	51,067.4	51,385.2	51,385.2	27.64
9	CPLEX	-	-	-	-	-	-	-
	BLD	62,148.7	62,148.7	62,838.5	62,838.5	62,838.5	62,838.5	28.05
10	CPLEX	-	-	-	-	59,269.2	59,269.2	-
	BLD	-	57,609.1	59,833.6	59,833.6	59,833.6	59,874.9	18.03

Gap1 is the original gap from BLD. Gap2 is an improved gap for BLD that uses the upper bound from CPLEX

6 Conclusions and future perspectives

In this paper, we investigated a long-term maintenance planning optimization problem for an integrated mining system and developed a MILP model to obtain optimal timings of all maintenance operations. There are two optimization criteria: the maximization of the total throughput and minimization of overstock quantity. The model includes various practical constraints to represent the inter-relationships between different maintenance operations and the inter-relationships between maintenance and production. These include non-conflicting conditions, alignment of related maintenance activities, inventory management and resource limitations. Our computational tests show that the proposed algorithm working in conjunction with CPLEX yields better solutions in faster time than CPLEX alone. The proposed model can be used as a tool to simulate different scenarios from several perspectives, such as changing maintenance requirements, flow impacts, storage capacities and initial conditions.

Additional work remains and the research in this paper can be extended in several directions as described below.

1. This work only considered one process plant and one port terminal. More complex systems with multiple process plants or several independent port terminals can be explored using similar strategies. For example, if there are two process plants feeding one port terminal, we need to coordinate the maintenance for all of them, considering a common resource pool and stock capacities, to minimize the disruptions caused by shutdowns. Efficient coordination and strategic planning are essential to ensure a smooth flow of products and maximize the throughput. With a more complex network, we suspect that more advanced algorithms will be required to deal with the dimensionality challenge.
2. Bringing in demand considerations such as ship arrival times will be an important extension, since the current model maximizes throughput without any regard for when ships are coming to take the product away. Aligning production with demand and adapting to variable flow rates will make the problem more realistic. To do so, one needs to take into account shipping schedules and meet demand at the specified times rather than just maximise overall throughput.
3. In the case study, we imposed the overstock penalty α_n to be 1 for all storage units. The sensitivity analysis shows that these penalty parameters are critical to control overstock. To better quantify the penalty, one can consider the cost of regaining overstock in terms of personnel and equipment time, and define better values for the parameters α_n for each storage unit.
4. Our model only involves separate products. However, in practice, blending of different products from different production lines will occur. In such cases, a more complex model considering blending constraints is required (Boland et al. 2016).
5. The model will be more powerful if the objective considers economic trade-offs (see Amaran et al. (2015)). Postponing maintenance helps to defer spending that can unlock capital for investment elsewhere. However, this may increase the risk of potential throughput loss due to degradation and unplanned maintenance. Therefore, an economic objective involving production revenue, maintenance costs,

- material costs, inventory holding costs and expenses related to the recovery of overstock would provide a more comprehensive assessment of a maintenance plan.
6. More strategies can be applied to improve the current decomposition method to solve the model. Although we can obtain good feasible solutions (the lower bound) faster than CPLEX alone, which is important for industry use, the upper bound can be improved as shown in Table 10. This can be achieved by developing strategies to tighten the optimality cuts. For example, in each iteration, instead of considering one subproblem, one can construct p smaller parallel subproblems and generate p corresponding optimality cuts. In this case, the continuous variable θ in the master problem would be replaced by p continuous variables. Another way is to solve the relaxed continuous master problem and/or apply the branch and bound procedure to alleviate the burden of repeatedly solving the integer master problem (Huang and Dinavahi 2019; Rahmaniani et al. 2020).

Acknowledgements This research was funded by the ARC Training Centre for Transforming Maintenance through Data Science (CTMTDS) (Grant Number IC180100030). The authors acknowledge the support from our iron ore industry partner, and thank Josh Boath, Murray McIntosh, Paul May, Sam Grech, Rob Leadbeater, Ella Kashi for their valuable feedback and suggestions. The authors would like to express their gratitude to the editor and the anonymous referees for their constructive feedback and insightful comments on this paper.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Al-Turki U, Duffuaa S, Bendaya M (2019) Trends in turnaround maintenance planning: literature review. *J Qual Maintain Eng* 25:253–271
- Al-Turki UM, Duffuaa SO, Ben-Daya MA (2013) A holistic system approach for turnaround performance management. *Maintenance Performance Measurement and Management, MPMM* (September)
- Amaran S, Sahinidis NV, Sharda B, Morrison M, Bury SJ, Miller S, Wassick JM (2015) Long-term turnaround planning for integrated chemical sites. *Comput Chem Eng* 72:145–158
- Amaran S, Zhang T, Sahinidis NV, Sharda B, Bury SJ (2016) Medium-term maintenance turnaround planning under uncertainty for integrated chemical sites. *Comput Chem Eng* 84:422–433
- Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer Math* 4(1):238–252
- Boland N, Kalinowski T, Waterer H, Zheng L (2013) Mixed integer programming based maintenance scheduling for the Hunter Valley coal chain. *J Sched* 16(6):649–659
- Boland N, Kalinowski T, Waterer H, Zheng L (2014a) Scheduling arc maintenance jobs in a network to maximize total flow over time. *Discret Appl Math* 163(PART 1):34–52
- Boland N, Kapoor R, Kaur S, Kalinowski T (2014b) Scheduling unit time arc shutdowns to maximize network flow over time: complexity results. *Networks* 63(2):196–202
- Boland N, Kalinowski T, Rigterink F, Savelsbergh M (2016) A special case of the generalized pooling problem arising in the mining industry, pp 1–8

- Budai G, Dekker R, Nicolai RP (2008) Maintenance and production: a review of planning models. Springer series in reliability engineering, vol 8, pp 321–344
- Cheung KY, Hui CW, Sakamoto H, Hirata K, O'Young L (2004) Short-term site-wide maintenance scheduling. *Comput Chem Eng* 28:91–102
- Duffuaa SO, Al-Turki UM, Daya MB (2019) Status of integrated turnaround maintenance. In: 2019 industrial and systems engineering conference, ISEC 2019, pp 0–3
- Fisher ML (2004) The Lagrangian relaxation method for solving integer programming problems. *Manag Sci* 50(12):1861–1874
- Geoffrion AM (1972) Generalized Benders decomposition. *J Optim Theory Appl* 10(4):237–260
- Held M, Karp RM (1971) The traveling-salesman problem and minimum spanning trees: part II. *Math Program* 1(1):6–25
- Huang S, Dinavahi V (2019) A branch-and-cut benders decomposition algorithm for transmission expansion planning. *IEEE Syst J* 13(1):659–669
- Lawrence M (1976) Maintenance management. DC Heath and Company, Toronto
- Levitt J (2004) Managing maintenance shutdowns and outages. Industry Press, New York
- Rahmaniani R, Ahmed S, Crainic TG, Gendreau M, Rei W (2020) The Benders dual decomposition method. *Oper Res* 68(3):878–895
- Sahinidis NV, Grossmann IE (1991) Convergence properties of generalized benders decomposition. *Comput Chem Eng* 15(7):481–491
- Schulz EP, Bandoni JA, Diaz MS (2006) Optimal shutdown policy for maintenance of cracking furnaces in ethylene plants. *Ind Eng Chem Res* 45:2748–2757
- Song M, Cheng L (2022) A generalized Benders decomposition approach for the mean-standard deviation shortest path problem. *Transp Lett* 15:823–833

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.