http://dx.doi.org/10.17159/2411-9717/2018/v118n12a5

Underground mine scheduling modelled as a flow shop: a review of relevant work and future challenges

by M. Åstrand*†, M. Johansson†, and J. Greberg‡

Synopsis

Advanced planning and automation are increasingly important in modern mines. Sophisticated methods for long-term mine planning are often used, and the advent of autonomous machines makes the actual operation more predictable. However, the interface between these two timescales, *i.e.* the scheduling of the mobile production fleet, often limits the ability to operate mines at maximum profitability. We show how scheduling the production fleet in an underground mine can be modelled as a flow shop. A flow shop is a general abstract process formulation that captures the key properties of a scheduling problem without going into specific details. Thus, the flow shop enables mine scheduling to reap the benefits of scheduling research from other industries. We review recent results from the mining community and the flow shop community, and introduce scheduling methods used in these two fields. This work aims at providing value to researchers from the mining community who want to leverage their skill set, as well as to theoretical researchers by presenting the mining process as a potential application area. Lastly, we discuss the results, and outline some future challenges and opportunities facing the industry.

Keywords

underground mining, scheduling, flow shop, optimization.

Introduction

The margins in underground mining are constantly under pressure since costs increase as the mine becomes deeper. One major component influencing the operational performance, and thus the profitability of an underground mine, is how the mobile machinery and staff are coordinated. Today, the coordination of the underground fleet, *i.e.* scheduling of the machines, is mostly done manually, with methods that are stretched to their limit and with no guarantees on performance. According to a survey of more than 200 high-level executives in mining companies around the world (Mincom, 2011), the top challenge for modern mines is to maximize production effectiveness, rather than improving the reliability of individual items of equipment. This highlights the importance of coordination on a system level.

The effectiveness of mine production can be increased by introducing supportive algorithms in the scheduling process. These algorithms enable mines to construct optimized schedules with respect to custom metrics. Additionally, manual scheduling is a tedious and error-prone task where the performance is heavily dependent on the scheduler. To reduce the staff dependency, supportive algorithms can help to achieve more uniform outcomes, resulting in a more predictable process. In turn, this increases the transparency of the mining process through the entire production chain. It is worth noting that automatic scheduling has proved beneficial in many other industries, such as chemicals and metals (Floudas and Lin, 2004; Tang *et al.*, 2001).

The operation of a mine is often planned on different levels, each with its own time horizon and task granularity (see Figure 1). The life-of-mine plan, which contains a rough plan of which year in which to extract what parts of the orebody until depletion, has the longest timespan. Based on the life-of-mine plan, extraction plans of various granularities are constructed that have a shorter time horizon and include more details about what amount of ore is planned to be produced during a shorter period of time. These extraction plans are then scheduled, *i.e.* the exact time of each activity is determined and the necessary resources (machines/personnel) are allocated. After a schedule has been constructed, typically spanning 1–2 weeks, it is a matter of realizing the schedule by routing the correct vehicle to the production area and performing the actual activity. Commonly, a supervisor underground acts as a real-time controller reacting to unforeseen changes and disturbances. Several authors (*e.g.* Song *et al.*, 2015) note that decisions taken by the supervisors are often based on intuition, which can result in suboptimal reactions to disturbances and disruptions.

* *Control, Optimization and Analytics, Automation Solutions, ABB Corporate Research, Sweden.*

† *Department of Automatic Control, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden.*

‡ *Department of Civil, Environmental and Natural Resources Engineering, Division of Mining and Geotechnical Engineering, Luleå University of Technology, Sweden.*

© *The Southern African Institute of Mining and Metallurgy, 2018. ISSN 2225-6253. Paper received Nov. 2017.*

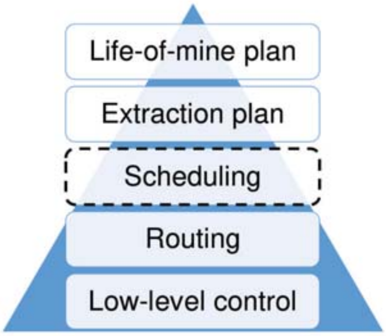

Figure 1—Different levels of planning, scheduling, and control found in underground mining

It is interesting to note that for the top two levels in the pyramid in Figure 1, sophisticated tools and methods are commonly used to ensure optimized (although perhaps not optimal) plans, while the lower levels still rely on a lot of manual and suboptimal work. With the advent of autonomous vehicles affecting the bottom two levels in the pyramid, it is obvious that the long-term plan and its realization need to be connected by scheduling in an efficient manner. Scheduling can thus be seen as the 'glue' that unites high-level planning (how to maximize dividends) with low-level control (excavating ore).

The incentives for theoretical research on scheduling often come from different underlying industries. To formalize the methods, general abstract formulations known as 'flow shops' have been developed, which capture the key properties of the scheduling problem without going into specific details. For instance, when scheduling an assembly line, it is important in which order parts should be assembled, but not if the actual part is the tyre of a bicycle or the button on a shirt. The flow shop thus provides a framework for developing scheduling algorithms separated from irrelevant process details. People working in a specific industry often believe that their processes differ from other industries in some key aspects, rendering results from other industries inapplicable. However, we argue that at least those industries which can be categorized as process industries have more things in common than things that differ. By modelling the mining process as a flow shop one can see in black-and-white which scheduling methods developed for other industries are applicable for mining.

Publications on underground mine scheduling sometimes briefly mention the flow shop concept as a way to justify methodological choices. However, there has been no systematic approach to highlight the specifics of underground mining and how it can be modelled as a flow shop in order to reap the benefits of scheduling research from other industries. The question that we address is thus how the diverse activities in underground mining can be adequately represented in a flow shop setting. To answer this, we review literature from both the mining community and the flow shop community, and study the methods used in both of these fields.

The goal for this paper is to be of value for both the theoretical researcher by presenting the mining process as a potential application area for the flow shop concept, and for researchers from the mining community who want to leverage their skill set or get an overview of available methods. In order to address this broad audience a common ground must be established. Therefore, the underground mining process is introduced, followed by a review of previous work on underground mining scheduling. The flow shop concept is then introduced, together with some relevant results from previous studies in this field. We conclude with a discussion, together with an outlook on future challenges and opportunities in underground mine scheduling.

Underground mining operations

Mining is the process of extracting minerals from the Earth, by either open pit or underground methods. In general, mining consists of a large number of discrete and continuous activities. Activities can be related to rock excavation, such as drilling and charging, or supporting activities such as managing the steady inflow of water and ventilating blast fumes. Underground mines typically have large uncertainties in process parameters (for instance task durations), due partly to the fact that the rock properties are not known until the rock has been physically encountered. This brings uncertainty, as one does not know if an activity will take 4, 6, or even 12 hours. Historically, underground mining has also been a very non-transparent process, where the state of the mine is evident only between shift breaks, if at all. With the adaption of new communication technologies underground, this has started to change. Still, profitable mining requires a variety of uncertain and partially non-transparent activities to be coordinated and steered towards a common goal.

An underground mine is in operation for many years, and decisions need to be taken on different timescales. It is natural to categorize these decisions as strategic, tactical, and operational. The most vital strategic decision is the choice of mining method, which sets the stage for all downstream decisions at the tactical and operational levels. An example of a tactical decision is the sequence of extraction. Underground mining essentially involves excavating blocks of ore under complex precedence constraints. A common approach found in the literature for finding a good sequence of extraction is to solve a mathematical programming problem which optimizes the net present value of the mine subject to production targets from the strategic level. The operational scale deals with the shortest time horizon: once the sequence of extraction has been determined, how should equipment and personnel be allocated to meet the production targets in the extraction plan?

The life of a mine can be divided into five stages (Newman, Kuchta, and Martinez, 2010). The first two stages, prospecting and exploration, are concerned with determining the profitability of exploiting an orebody. The third stage is development, *i.e.* making the orebody accessible for production, while the fourth stage, exploitation, deals with the actual excavation of ore. Finally, the fifth stage (reclamation) is restoring the affected environment as much as possible to its original state. Out of the five stages in the life of a mine the only profitable stage is exploitation. Hence, how the exploitation is realized, *i.e.* choosing a mining method, is vital for the profitability of a mine. The mining method is selected based on the geometry, geology, and rock mass characteristics of the orebody and the surrounding rock. Since different mining methods have different process dynamics they pose different scheduling problems.

Excavating drifts (tunnels) is called development and is a key component in most mining methods. The drifts are used not only for transporting rock and equipment, but also for supporting activities such as supplying air, water, and electricity. Development is implemented in a cyclic process, often called the production cycle (see Figure 2 and Table I). In most mines, development takes place simultaneously in many parallel drifts, and different types of machine are needed at different steps in the production cycle. This makes scheduling development a non-trivial activity.

Different mining methods have different strengths and weaknesses. The choice of mining method depends, among other factors, on the depth and inclination of the orebody. Orebodies close to the surface are naturally mined in open pits, whereas if the orebody goes deep, underground mining methods are used. A common underground mining method for near-horizontal orebodies such as deposits of copper, coal, or potash, is room-and-pillar mining (see Figure 3a). As the name suggests, in this method the ore is excavated in blocks, with some blocks being left as pillars to provide support for the surrounding rock. Common mining methods for steep orebodies include sublevel caving, block caving, and cut-and-fill mining. Sublevel caving, depicted in Figure 3b, is based on collapsing the ore and adjacent waste rock by blasting at sublevels and progressing downwards. Regularly spaced drifts are developed into the orebody to enable drilling and blasting so as to initiate this caving. The ore is then dumped into orepasses connecting production levels to haulage levels. Another method for steep orebodies is block caving (Figure 3c), which is used for large-scale production where parts of the orebody are induced to cave by removing the support from underneath. The ore is fractured naturally by gravity and rock stress, and caves down to drawpoints. Cut-and-fill mining (Figure 3d) differs from the abovementioned methods in that it uses backfill. Proceeding upwards, the excavated voids are backfilled with concrete, waste rock, and sand. The backfill acts as support for the surrounding rock and becomes the platform for subsequent excavation on a higher level as production progresses.

Although some mining methods are similar, most of them differ in key aspects such as the type and number of machines required, where these machines are operating, whether rock support is needed, and how fast the mine evolves. This means that the choice of mining method does not only affect the strategic and tactical levels, but also propagates down to the operational level. Some categorization is, however, evident since the process dynamics of room-and-pillar and cut-and-fill are similar to development as the process follows a production cycle at one single location. This may not be the case for other methods where different parts of the production cycle take place at different locations. For more on underground mining methods, and mining in general, see Hustrulid (1982) or Darling (2011).

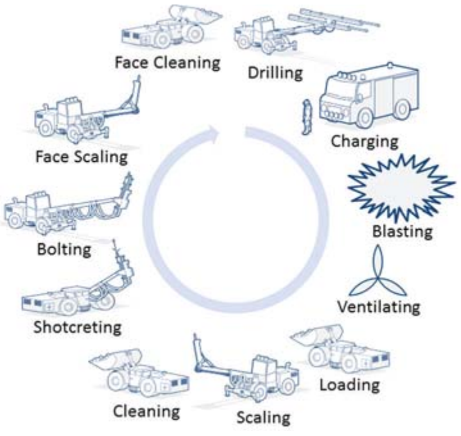

Figure 2—A typical production cycle includes a diverse set of mobile machines realizing the activities in the cycle

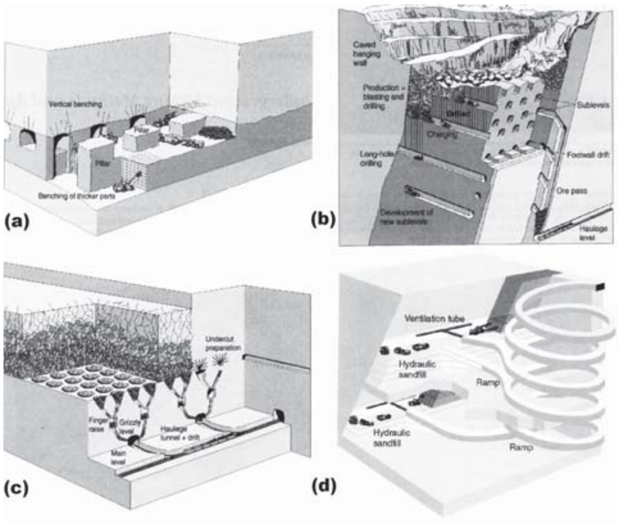

Figure 3—A selection of mining methods. Room-and-pillar mining (a) is used for horizontal orebodies. Sublevel caving (b), block caving (c), and cut-and-fill mining (d) are primarily used for steep orebodies. (Hamrin 2001)

Table I

Some of the activities in a common production cycle

Activity	Machine	Description
Drilling	Drill rig	Drilling holes in the rock face
Charging	Wheel loader with platform	Charging holes with explosives
Loading	LHD	Removing loose rock from a drift after blasting or scaling
Scaling	Scale rig	Mechanically removing loosely attached rock from the walls of a drift
Cleaning	Wheel loader with bucket	Removing small amounts of rock from the drift after scaling
Bolting	Bolter	Securing a drift by installing bolts in the rock
Shotcreting	Shotcreter	Reinforcing by spraying the walls of a drift with concrete

This introduction contains the essence of underground mine operations, as viewed from a scheduling perspective. That being said, each mine site has its own peculiarities which affect the operation and thus the scheduling problem. Moreover, a schedule is not only dependent on production goals. It must also adhere to legislation and safety standards, and it needs to consider additional information from other sources such as maintenance, localization, and geomechanical systems. In summary, underground mining requires the coordination of a multitude of heterogeneous machines, under large process uncertainties, in a dynamic environment, while obeying numerous process constraints. This is a non-trivial task, and it is part of the reason why even in modern mines the average utilization of some mobile machines can be well below 50% (Gustafson *et al.*, 2014). Altogether, this indicates that scheduling is a crucial component in the overall mining process.

Scheduling in underground mining

There are some previous works on scheduling in underground mining, although these are scarce. The terminology in the mining community regarding planning, scheduling, and dispatching varies between authors and is unfortunately inconsistent at times. Outside the mining community, planning is used to answer what and how, while scheduling is answering who and when. Following this convention, mine planning thus determines in what sequence the orebody is exploited and how (by which method) this is done. The scheduling process then determines exactly when each activity is to be conducted, and by who (what resource). Another common term is dispatching, which represents the actual realization of the activities contained in a schedule. For instance, several parallel schedules might exist but only one schedule is actually dispatched. In the literature there is also a categorization of scheduling and planning into long-term, medium-term, and short-term, where there is no consensus on where one time horizon ends and another starts. The works introduced here will be focused on short-term questions rather than long-term, *i.e.* on operational level rather than strategic or tactical. For a review on research on long-term issues, refer to Newman *et al.* (2010) or Kozan and Liu (2011).

With a few exceptions (*e.g.* Williams, Smith, and Wells, 1973), research on scheduling in underground mining is a recent topic. Song *et al.* (2015) developed a decision-support instrument to aid the scheduling process by constructing a schedule of some of the activities in the production cycle. The sequencing is done by calculating all possible permutations of the activities and selecting the one with shortest total duration. The total duration of a schedule, *i.e.* the difference between the start time of the first activity and the end time of the last activity, is called the makespan and is a common quality metric in the scheduling literature. To cope with the factorial growth of possible permutations the authors cluster the faces based on geographical distance, and schedule one machine set at each cluster. The method was tested on data from the Kittilä mine in Finland, where the makespan achieved by using the algorithm was far shorter than the actual outcome from manual scheduling. However, the authors recognize that this analysis is an open loop in the sense that it does not incorporate the dynamic nature of uncertain activity durations and disturbances.

Potash is commonly mined using the room-and-pillar method. Schulze *et al.* (2016) studied how to schedule the mobile production fleet in an underground potash mine, also with the objective of minimizing the makespan. This was done by formulating a mixed integer programming (MIP) model and solving it to optimality using a commercial solver for small problem instances. To handle the computational burden when scaling up the problem to realistic sizes, the authors introduced construction procedures embedded in a multi-start environment. By observing that non-delay schedules (all activities placed without buffers) need not include the optimal schedule, the construction procedure is enhanced by introducing conscious delays into the heuristic. The authors additionally introduced a modified Giffler and Thompson procedure (Giffler and Thompson, 1960) for scheduling medium to large problem instances. The different algorithms were tested on a variety of problem and fleet sizes. The authors conclude that for small problem instances, solving the MIP formulation using a commercial solver is efficient. For medium sized problems, the Giffler and Thompson procedure works best, and for large problems the construction procedure, including conscious delays, yields the lowest makespan. The fact that including conscious delays may lower the makespan indicates that the common mine practice that machines should never be idle is not necessarily a good practice. The authors continued to develop the heuristics in Schulze and Zimmermann (2017), where a combined method for scheduling both staff and machines is presented. The heuristic is constructed to minimize deviations from a targeted amount of mined potash, and the results show that the algorithm outperforms manual scheduling of the same problem.

Nehring, Topal, and Knights (2010) studied an application in a sublevel stoping mine. Here, the scheduling problem addresses how to transport the ore from the drawpoints, via intermediate storage, to the haulage shaft. The MIP model allocates machines to different drawpoints on a shift basis over a period of 2 months. The objective is to minimize the deviation from targeted production. The MIP model is solved using CPLEX and evaluated on a simulated mine. This is one of the few papers that also includes secondary ore movements such as transporting ore from an orepass to the crusher. The authors continue to explore this path in Nehring, Topal, and Little (2010) by simplifying the model to decrease computation time. Additionally, in Nehring *et al.* (2012) the medium-term goal of optimizing net present value is included in the MIP formulation. The advantage of using a holistic model spanning both tactical and operational decisions is also evaluated by Little, Knights, and Topal (2013). The authors integrate both stope layout and production scheduling in one model and note that the generated schedules are superior to those created sequentially. However, using a holistic model instead of two segregated models increases the computation time from seconds to days.

Some mining methods require backfill, *i.e.* refilling excavated voids with waste rock, sand, and concrete. This adds complexity to the scheduling problem since the backfill activities need to be synchronized with excavation. O'Sullivan and Newman (2015) studied the Lisheen mine in Ireland, which uses a mining method similar to cut-and-fill. A MIP model is introduced to schedule when a certain area underground should be extracted in order to maximize the production of mineral (*i.e.* maximizing the product of the ore grade and the volume of ore extracted). Unfortunately, the MIP model is unable to solve problems over an extended period of time due to the high computational cost. The authors alleviate this issue by introducing a heuristic decomposition method based on fixing variables in a predefined fashion. The idea is to first consider only high-grade areas and solve the scheduling problem for activities in those areas only. When a solution is found, lower-grade areas are introduced into a new scheduling problem where the solution considering only the high-grade areas is enforced as a constraint. The authors argue that the decomposition-based method provides a feasible way of obtaining good (but not optimal) results.

Another decomposition-based technique can be found in Martinez and Newman (2011), who build on the results first obtained by Kuchta, Newman, and Topal (2004) and later refined by Newman, Kuchta, and Martinez (2007). Similar to O'Sullivan and Newman (2015), Martinez and Newman (2011) decompose a MIP model based on ore grades. The goal is to determine monthly resource allocations over a horizon spanning several years. The objective is to minimize the deviation from monthly targeted quantities while adhering to numerous mining considerations. Upon evaluation on data-sets from the Kiruna mine in Sweden, the authors note that the method often finds a solution within 5% of the targeted quantities, and does so in a reasonable time for this application.

The solution of a scheduling problem contains at least a schedule of the activities' start and end times, together with allocated machines for performing these activities. Executing the activities in a mine often includes routing the correct equipment to the correct location – a distinct problem in production scheduling. For completeness, some recent work in this field is briefly mentioned. Equi *et al.* (1997) note that the characteristics of the open pit routing problem are similar to *e.g.* sugarcane and timber production, while for the underground case, Saayman, Craig, and Camisani-Calzolari (2006) state that routing research from other areas has limited application. The authors proceed by studying the effect of different routing strategies in dispatching LHDs to drawpoints in a block cave diamond mine. Another study of underground routing can be found in Gamache, Grimard, and Cohen (2005), who used Dijkstra's algorithm for routing mobile machinery. Dijkstra's algorithm considers one vehicle at the time, which neglects the interaction between vehicles. This research was extended by Beaulieu and Gamache (2006), who present a method based on dynamic programming which provides a global view on the routing problem.

Although the emphasis in this work is on underground mining, it is appropriate to mention some noteworthy open pit scheduling studies. Beaulieu and Gamache (2006) note that scheduling for an underground mine operation does not simply involve altering existing algorithms for open pit mines; it is a related but distinct problem from scheduling open pit mines. An aspect that differs between the two types of mine is that the transparency of the process in open pits has historically been higher than that of underground operations, *i.e.* more relevant information is available in order to make informed decisions. Another difference is the overall process dynamics, which partially arises from the need of a more diverse machine fleet underground. In the open pit context there are mainly two types of equipment, shovels and trucks. The shovel-truck problem simply involves deciding where a truck should go for loading after it has unloaded. Alarie and Gamache (2002) give a structured overview of the different methods used for open pit dispatching. These methods are categorized as single-stage or multi-stage, where mathematical programming is often used in the first stage in multi-stage programming but rarely in later stages.

One feature that characterizes mining compared to other process industries is the noisy (sometimes chaotic) nature of operation. A common mining practice to deal with variable duration of activities is to place buffers between subsequent activities. Another way is to incorporate the uncertainty into the scheduling model formulation. Based on a similar approach to long-term planning, Matamoros and Dimitrakopoulos (2016) use stochastic integer programming with recourse to produce schedules on a monthly basis for allocating shovels and trucks to different mining areas in an open pit mine. The authors simultaneously optimize the fleet and mining considerations together with the extraction sequence. Uncertainty is accounted for by including fleet parameters and ore quality as stochastic variables in the model formulation. The problem is solved using a commercial solver to produce monthly extraction sequences and machine allocations for a planning horizon of 12 months. This composite way of modelling the mining process has both advantages and disadvantages. It is obvious that a decision made on the tactical level (*e.g.* extraction sequence) will have an effect on the operational level (fleet allocation), and *vice versa*. Optimization using a holistic model incorporates the synergies of considering both timescales simultaneously. The authors use an objective function formulated as a sum of smaller optimization goals spanning both the tactical and the operational level (minimize cost of extraction, minimize shovel movement, maximize shovel utilization, minimize deviation from targeted plan, among others). One drawback is thus the need to determine weighting parameters such as cost per shovel movement and cost per surplus of mining tonnage. Determining these parameters accurately may be very difficult in practice.

A stochastic approach to the shovel-truck problem for open pits was also studied by Ta *et al.* (2005). The authors decompose the problem by solving two sub-problems, where the first step is solving a chance-constrained problem in continuous decision variables, while the second step solves a MIP model in discrete variables. The combinatorial challenge facing questions regarding global optimality is noted by the authors to be hard, which is why heuristic approaches to the dispatching problem for open pit mines are common.

Finally, it is worth mentioning that in order to schedule efficiently, access to accurate process information is crucial. As mentioned earlier, a major difference between open pit and underground operations is the availability of up-to-date process information. However, the adoption of new technology in underground mining has started to change this, where mine-wide WiFi enables real-time monitoring of machines, equipment, and staff. Song, Rinne, and van Wageningen (2013) discuss the impact of information technology underground, as well as summarizing systems and hardware suitable for providing informational transparency. Burger (2006) describes a system of systems for integrating the mining plan of the Finsch mine in South Africa into the control system, showing that the integration of different processes is crucial for future profitability. Howes and Forrest (2012) describe an approach to a similar problem at the Chelopech mine in Bulgaria. They provide the operating context for a scheduling solution including upstream and downstream integrations to *e.g.* long-term production planning and to process control. The authors note that interfacing between systems on different operational levels is non-trivial, and that although no standards exist specifically for the mining industry, there are tentative standards such as ISA-95 that have been used successfully in other industries.

Flow shop scheduling

The scheduling of the mobile machinery is a key aspect of successful underground mining operations. One way to describe a scheduling problem is to use the concept of a flow shop. The term flow shop is used to denote a typical manufacturing set-up where the items that are being produced need to be processed on several stationary machines. For instance, to produce a loaf of bread in an industrial bakery, the ingredients first need to pass through a dough mixer, followed by a portioning machine, and lastly the raw loaf needs to be baked in an oven. Several parallel dough mixers may exist so a particular batch of ingredients can be scheduled on one out of many mixers. In a flow shop, each producible item is known as a job, and the manufacturing steps are denoted stages. The equipment units that process the items at each stage (*e.g.* mixer, oven) are called machines.

More formally, a flow shop is a process description where jobs are scheduled on a set of resources in a given processing order. A k-stage hybrid flow shop (HFS) is a generalization of the flow shop problem where n jobs are to be processed in $k > 2$ stages. Each stage $s = 1 \ldots, k$ holds $M_S \geq 1$ machines, and all jobs are processed in the same relative sequence (stage 1 → … → stage k) but are allowed to skip some stages given that each is processed in at least one. An HFS is sometimes referred to as a flexible flow shop, flexible flow line, or multi-processor flow shop although the meaning of these terms varies between authors.

In a mining context, the flow shop abstraction can be used to model mining methods that follow a production cycle at a single location (such as development or cut-and-fill mining). The individual activities in the production cycle can be thought of as stages (see Figure 4), where each stage contains a set of machines that can perform the activity. In contrast to the common manufacturing set-up (*e.g.* the bakery analogy) it is not jobs that travel between stationary machines, but rather mobile machinery that travel between stationary jobs.

Flow shop modelling

A large vocabulary exists in order to describe the characteristics of an industrial process. For instance, if the time for a machine to perform a job (the processing time) is independent of which machine in the stage is used, and independent of which particular job is being processed, the machines are said to be identical. However, if the machines are independent of which job is processed, but have different processing times based on which machine is used, then the HFS is said to have uniform parallel machines. An example of uniform machines is when the processing time of machine A is 10 minutes for all jobs, while that for machine B is 15 minutes for all jobs. In the scenario when the processing time also explicitly depends on what job is being processed, then the HFS is said to have unrelated parallel machines (*e.g.* on a particular machine job A takes 10 minutes while job B takes 15 minutes). There is some sambiguity on the topic of parallel machines, where some authors claim that unrelated parallel machines are included in the formulation of an HFS, while others claim that an HFS without specification holds identical machines (Ruiz and Maroto, 2006).

Furthermore, if a certain job can only be processed on a subset of the machines in a stage, then the problem is said to have machine eligibility. If a job is to be treated at one stage several times it is said to have revisits or recirculations. An HFS with preemption allows a job to be interrupted and finished later, while an HFS with non-preemption models a situation where jobs which have started on a machine must be completed on that machine without interruption. An HFS naturally has precedence constraints between stages since each job has to pass through the stages sequentially.

In a general setting, starting a job at a certain stage may not be possible until another job is completed. The HFS is then said to have precedence constraints between jobs. Often

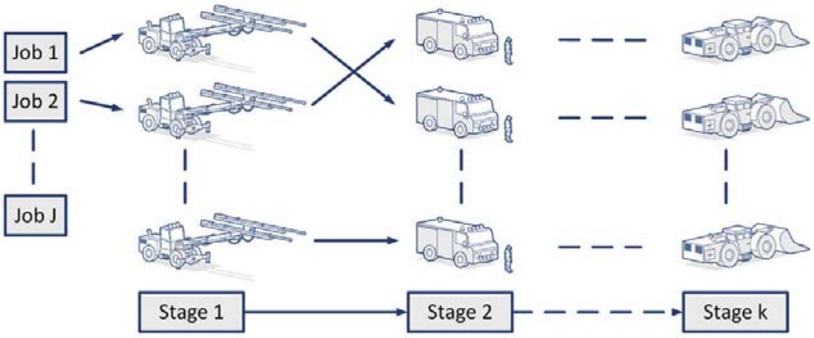

Figure 4—A *k*-stage hybrid flow shop is a process description where jobs sequentially pass through *k* stages. At each stage the job is processed by one of several machines in that machine park. In this context one job would correspond to one complete production cycle at a specified location

a certain time is required to prepare a machine before it can process a job at a certain stage. If the set-up time is constant then it is most often included in the duration of the stage. However, if the set-up time depends on the processing order, the HFS is said to have sequence-dependent set-up times (for instance, if switching between two different types of jobs requires changing some tooling, but scheduling two subsequent jobs of the same type eliminates the need for this time buffer). If the process is constrained by the fact that a sufficient amount of time must pass between certain stages of a job, the problem is said to have time lags. For more on classification of different flow shops see Pinedo (2015).

A scheduling problem is not particularly interesting without an objective. As noted by Ruiz and Vázquez-Rodríguez (2010), the most common objective of an HFS problem is to minimize the makespan, defined as the time difference between the start of the first activity and the end of the last activity. It is commonly reasoned that a low makespan indicates a high utilization of the machines. Let σ denote the feasible schedule, and let $C_{jk}(\sigma)$ be the completion time of job j at the final k-th stage under the schedule σ. Minimizing the makespan of processing J jobs can thus be formulated as

$$\underset{\sigma}{\text{minimize}} \quad \max_{j \in \{1,\dots,J\}} C_{jk}(\sigma) \qquad [1]$$

An example of another common objective in scheduling is to minimize the inventory costs. In some settings, this can

be represented as a weighted sum of all completion times which focuses on reducing the completion times of all jobs simultaneously.

$$\underset{\sigma}{\text{minimize}} \sum_{j} w_j \, C_{jk}(\sigma) \qquad [2]$$

In order to leverage research from other industries it is beneficial to model the mining process as a flow shop. HFS is mentioned in a few previous studies from the mining community. Most notably, Schulze *et al.* (2016) argue that the production cycle in a room-and-pillar mine can be described as an HFS where excavating one cubic block is considered as one job. The authors model the process with unrelated parallel machines, indicating not only machine-dependent but also location-dependent processing times. They complicate the HFS further by including revisits to model the fact that this particular potash mine has time-based rescaling, meaning that if a certain amount of time has passed since scaling the work area it is unsafe. Hence, rescaling is necessary before continuing operations. The authors do not include any transportation time for the machines to travel between faces. Another study that briefly mentions HFS is that by Song *et al.* (2015). The authors note that the mining operation under study is an HFS, but without providing any details or motivations. They claim that the transportation time between work areas in an underground mine is a key characteristic that previous work on HFS scheduling does not consider. However, as discussed above, transportation times can be modelled as sequence-dependent set-up times, for which there are numerous related studies.

By using the introduced vocabulary, we note that cut-and-fill, room-and-pillar, and development can be modelled as an HFS as follows.

- ➤ One job is identified as excavating one geographical volume of rock in a specified location, *i.e.* one job corresponds to one full production cycle.
- ➤ Each location is processed by the activities included in a production cycle, hence the activities in the cycle are seen as stages in a k-stage HFS.
- ➤ If the machine park is heterogeneous (different specifications such as shovel volume or drilling speed), but does not vary in different parts of the mine, then the machines can be considered as uniform.
- ➤ The transportation time between different locations can be modelled as sequence-dependent set-up times. This does not exclude 'ordinary' set-up times, such as connecting a machine to the electrical grid, since these durations can be added on top of the transportation time in a sequence-dependent manner.
- ➤ Multiple scaling, clearing, loading, and rock supporting activities during one production cycle can be modelled by revisits.
- ➤ Precedence constraints can be used to account for (i) considering safety standards, (ii) the impact of operating in a confined environment, and (iii) to model the fact that access to a location needs to be granted before excavation can start.

Summarizing, the mining methods that follow a production cycle at a single location can be modelled as a k-stage HFS with uniform parallel machines, sequence-dependent set-up times, revisits, and precedence constraints between jobs. A possible objective could be to minimize the makespan (see Equation [1]), indicating that sought-after schedules include efficient use of the mobile production fleet.

Flow shop scheduling techniques

The methods used for solving the HFS scheduling problem can be categorized as exact, heuristic, or metaheuristic. The exact methods solve the HFS problem to optimality. A simple example of an exact method is the scheduling of all different combinations of activities and selecting the one that optimizes some metric (given that it is feasible). It is worth mentioning that even one of the simplest HFS (two stages, where the first stage has only a single machine and the other stage has two machines) is known to be *NP*-hard (Gupta, 1988). Thus, it may be very hard to solve certain scheduling problems to optimality in a reasonable time using exact methods. Therefore, heuristic methods are often introduced as the size of the problem increases. A heuristic is an algorithm that is typically fast, but lacks optimality guarantees. Metaheuristics, in turn, are systematic techniques for introducing randomness to improve on heuristics in the search for optimal solutions.

Exact methods

To be categorized as exact, a scheduling algorithm needs to be guaranteed to find the globally optimal solution on every problem instance. The most commonly used exact technique for HFS scheduling is branch and bound (B&B), where the problem is solved by searching over a binary decision tree. At each node, the search space is branched (split) into two disjoint sets. Two statistics are tracked, an upper bound and a lower bound. For minimization, the upper bound is simply the best solution so far, while the lower bound is calculated by solving a relaxed problem. The relaxation could, for instance, be to consider all discrete variables as continuous (known as linear relaxation). Obviously, a solution to the relaxed problem will have at least as good an objective as the original problem. In each node in the search tree the lower bound is calculated and compared to the upper bound. If the lower bound is higher than the upper bound (*i.e.* the best solution found so far) then that part of the tree cannot contain the solution with the minimal cost. Thus, the whole subtree can be ignored, which results in a reduced search space. Much of the previous work on B&B in HFS scheduling considers two- or three-stage HFS with only a few parallel machines at each stage. One representative example (Haouari, Hidri, and Gharbi, 2006) use a B&B algorithm to schedule a two-stage HFS under the objective of minimizing the makespan. The authors derive several efficient lower bounds by relaxing the HFS under study. Upper bounds are calculated in each node by an algorithm that gives priority to jobs with a lot of remaining processing time. The authors note that the B&B algorithm with the proposed bounds and dominance rules can handle larger problem than previous work in the field.

A related method is constraint programming, which is a modern graph-based method increasingly used for scheduling (Baptiste, le Pape, and Nuijten, 2012). This approach leverages the common B&B method by systematically exploiting the structure of the problem. Constraint programming has been used successfully in many diverse areas including planning, scheduling, and vehicle routing (Baptiste, le Pape, and Nuijten, 2012).

An alternative method is to formulate the scheduling problem as a mixed integer program (MIP) and solve it using a commercial solver. The solvers often use B&B internally to calculate the upper and lower bounds using a linear relaxation. Early methods for scheduling HFS are criticized by Liu and Karimi (2008) due to their incapability of handling more complex HFS problems. The authors use MIP to study a *k*-stage HFS which produces multiple products in batches. It is common in modelling HFS by MIP to use variables either representing time-slots, or by variables representing the particular sequence in which jobs are scheduled on stages. Liu and Karimi (2008) continue to note that even though sequence-based models often involve fewer binary variables than the time-slot counterpart, the relaxation of sequence-based models is often inferior. To tighten the relaxation the authors propose to combine both models. However, the work indicates that models with fewer binary variables, or tighter relaxations, do not necessarily always perform better (a remark which is noted by the authors to be frequent in related literature) (Table II).

For concreteness, a basic MIP model is introduced for scheduling jobs in a k-stage HFS with the objective of minimizing makespan. The full model can be seen in Model 1. Here, C_{js} denotes the completion time of job at stage s. The makespan $C_{\max}$ is introduced as the minimum value such that $C_{\max} \geq C_{js}$ holds for all j and s. Further, each stage has M_s uniform parallel machines, and p_{js} denotes the processing time of job j at stage s. By introducing two binary decision variables, Y_{jsm} and $X_{jj's}$, we can adequately represent the k-stage HFS. The variable Y_{jsm} represents machine allocation, while $X_{jj's}$ keeps track of the order in which the jobs are scheduled.

In the MIP model given in Model 1, constraint [4] ensures that at each stage only one machine is scheduled to process a certain job. Constraint [5] enforces that the completion time of a stage is dependent on the processing time of the allocated machine. The two constraints in [6] and [7] make sure that one machine cannot process several jobs simultaneously, where a large number Q is used to enforce the disjunctive constraint (Griva, Nash, and Sofer, 2009). The domains of the decision variables are specified in constraint [8]. This MIP model can be seen as a baseline where modifications further specify characteristics of the particular process under study.

Heuristic methods

As noted by many authors (*e.g.* Ruiz and Vázquez-Rodríguez, 2010), most exact methods are incapable of handling medium and large problem sizes. Therefore, research often turns to heuristic methods as the problem size increases. Heuristic methods (including dispatching rules, scheduling policies, and construction procedures) are commonly deployed to deal with the computational challenge of

Table II

Model 1: An MIP model for minimizing makespan in a basic k-stage flow shop

Decision variables	
$Y_{jsm} = \begin{cases} 1 \text{ if job } j \text{ is processed in stage } s \text{ by machine } m \\ 0 \text{ otherwise} \end{cases}$	$X_{jj's} = \begin{cases} 1 \text{ if job } j \text{ precedes job } j' \text{ in stage } s \\ 0 \text{ otherwise} \end{cases}$
C_{js} = completion time of job j at stage s	$C_{\max}$ = the makespan of the schedule
MIP model	
minimize $C_{\max}$ s.t.	
$C_{\max} \geq C_{js}, \forall(j,s)$ [3]	$Q(2 - Y_{jsm} - Y_{j'sm} + X_{jj's}) +$ $C_{js} - C_{j's} \geq p_{js}, \ \forall(j,j',s,m), j < j'$ [6]
$\sum_{m=1}^{M_s} Y_{jsm} = 1, \forall(j,s)$ [4]	$Q(3 - Y_{jsm} - Y_{j'sm} - X_{jj's}) +$ $C_{j's} - C_{js} \geq p_{j's}, \ \forall(j,j',s,m), j < j'$ [7]
$C_{j,s-1} + \sum_{m=1}^{M_s} Y_{jsm} p_{js} \leq C_{js}, \ \forall(j,s)$ [5]	$Y_{jsm}, X_{jj's} \in \{0,1\},$ $C_{\max}, C_{js} \geq 0, \ \forall(j,j',s,m)$ [8]

complex HFS problems. Heuristics are computationally cheap and often intuitive algorithms for constructing a feasible schedule. Although heuristics may produce good solutions, they come with no performance guarantees. Moreover, intuitive heuristics might perform poorly on some problem instances. One famous example was introduced by Graham (1978), who describes a bicycle factory. Scheduling of workers to assemble the bicycles is done according to the scheduling policy that (i) no assembler can be idle if there is some work task that can be done and (ii) an assembler must continue working on a task until it is completed (it is not permitted to pause and do another task). Although this policy seems reasonable, Graham showed that both reducing the duration of each task and hiring more workers results in fewer bicycles produced per day. Such unexpected and unpredictable behaviours are common for many dispatching rules.

A common heuristic found in the scheduling literature is the Nawaz-Enscore-Ham algorithm (NEH algorithm) introduced by Nawaz, Enscore, and Ham (1983). The basic idea of the NEH algorithm is that jobs which require a lot of processing time should be scheduled before jobs which require less time (so as to fit the smaller jobs in-between larger ones). The NEH algorithm is based on the average processing time (APT) of a job at stage s. For unrelated parallel machines the processing times depend on the job, stage, and machine ($p = p_{jsm}$) and can be calculated as

$$APT_{js} = \sum_{m=1}^{M_s} \frac{p_{jsm}}{M_s}, \qquad [9]$$

which in turn is used to the form the total average processing time for all jobs $TAPT_j = \sum_s APT_{js}$. In Algorithm 1 (Table III) the NEH algorithm for minimizing makespan is given in pseudocode. In order to be specific, assume that in step 2 we select job j_1 and j_2. The makespans of the two schedules are then calculated, one for the job sequence (j_1,j_2) and one for the job sequence (j_2,j_1). Assume that the makespan of (j_1,j_2) is lower than that of (j_2,j_1), the sequence (j_2,j_1) is then discarded. At step 3 we consider a new job j_3; we then evaluate all possible insertions of the job into the current sequence of jobs. In this case we evaluate the makespan of three schedules; (j_3,j_1,j_2), (j_1,j_3,j_2), and (j_1,j_2,j_3). Similarly, the schedule with the lowest makespan is kept, while the others are discarded. This continues until the list of unscheduled jobs is empty. Once a sequence has been determined the criterion used for allocating machines at each stage is often based on the earliest finishing time. The NEH algorithm is very flexible since the assignment of jobs to machines can be based on any criteria, not only makespan. (Table III).

Brah and Loo (1999) compare five HFS scheduling algorithms and conclude that the NEH algorithm produces high-quality schedules. The NEH algorithm is also deemed appropriate by Ruiz, Şerifoğlu, and Urlings (2008), who studied a complex *k*-stage HFS including unrelated parallel machines, release dates for machines, sequence-dependent set-up times, time lags, precedence constraints between jobs, and machine eligibility. They furthermore compare heuristic algorithms with solving MIP models, and conclude that the NEH algorithm scales best with increased complexity of the scheduling problem.

Another interesting heuristic for the *k*-stage HFS is the shifting bottleneck procedure (SBP) found in *e.g.* Cheng, Karuno, and Kise (2001). A bottleneck is a term used for a limiting resource in a process chain, and the idea behind SBP is to give priority to a bottleneck resource. For instance, consider a three-stage HFS with a large number of identical machines at stages 1 and 3, but only a single machine at stage 2. If all machines have the same processing time, the single machine in stage 2, through which all jobs need to pass, becomes the bottleneck in the process. The SBP for this three-stage HFS can be thought of as decomposing the three stages into three single-stage parallel machine problems, and iteratively sequencing and re-optimizing a schedule based on which machine is the bottleneck in the current iteration. The idea behind this heuristic is similar to the theory of constraints (Nave, 2002).

Metaheuristic methods

The drawback of using heuristics is that there is no guarantee that the solution will be optimal. In fact, heuristic solutions are typically suboptimal. However, the quality of a constructed schedule can often be improved by randomly perturbing a schedule and selecting a solution which shows the best improvement. These methods, called metaheuristics, include a variety of optimization techniques such as simulated annealing, tabu search, and genetic algorithms, with the common concept of combining randomness and local search (Pham and Karaboga, 2012). Metaheuristics are initialized by a solution, and explore the neighbourhood of that solution in hope of finding a solution with an improved objective. An example is running a heuristic algorithm to produce an initial schedule, and then studying all feasible swaps of two activities in that schedule. The explored neighbourhood would then correspond to all pairwise swaps such that machines allocations M_i, M_j to jobs J_i, J_j are changed from ($M_i \rightarrow J_i$, $M_j \rightarrow J_j$) to ($M_i \rightarrow J_i$, $M_j \rightarrow J_i$) as long as it is a feasible change. For all schedules in this neighbourhood, the schedule with the lowest makespan is then selected.

A representative metaheuristic commonly deployed to complex HFS is the genetic algorithm (GA). This algorithm takes inspiration from biology by mimicking the evolutionary process of natural selection. Based on a set of tunable parameters (such as probability of mutation) the algorithm 'cross-breeds' schedules to produce new schedules. In each iteration the newly produced schedules are evaluated and a subset of these is kept to breed the next generation of

Table III

Algorithm 1. NEH algorithm

```
1 . Sort all jobs in non-decreasing TAPT
2.  Select the two jobs $j$, $j''$ which have highest TAPT and remove them from the list
3.  Construct two schedules, one where $j$ is scheduled first and one where $j'$ is scheduled first. Discard the schedule with highest makespan
4.  for all jobs left in the list do
5.      Select the job with highest TAPT and remove it from list
6.      Calculate all possible insertions of the job into the current job sequence
7.      Select the insertion which yields the lowest makespan
8.  end for
```

schedules based on their fitness, *i.e.* objective function. One example of GA can be found in Ruiz and Maroto (2006). Unlike previous approaches with GA, which decompose the problem into first sequencing the jobs and then finding a feasible allocation of machines, in this paper the authors embed the makespan directly into the fitness function of the GA. Inspired by an industrial setting, the authors study a complex *k*-stage HFS with unrelated parallel machines, sequence-dependent set up times, and machine eligibility. Furthermore, they benchmark their implementation to other metaheuristic methods and show that their method outperforms the next best metaheuristic by at least 50%. It is noted that the GA outperformed manual scheduling done by experts at the industry by almost 10%, indicating the industrial relevance of these methods when it comes to scheduling complex problems. In general, a drawback of GA is the presence of numerous optimization parameters which are typically case-dependent. In the example by Ruiz and Maroto (2006) they determine these parameters by performing a full factorial experiment on simulated test data.

Another study of sequence-dependent set-up times, but with another metaheuristic method, can be found in Naderi *et al.* (2009), where simulated annealing (SA) is used instead of GA. SA is an optimization technique that resembles the physical process that molecules undergo when a heated metal is cooled. At first the mobility of the molecules is high (corresponding to evaluating a lot of diverse schedules), whereas when the metal cools the mobility becomes increasingly impaired (corresponding to focusing in on further optimizing a subset of these schedules) and the molecules finally freeze in place (resulting in one final schedule). The problem under study in this paper is a *k*-stage HFS with transportation times and sequence-dependent set-up times with the objective of minimizing, among other things, the sum of completion times (see Equation [2]). The method is compared to other metaheuristics, and the analysis indicates that the proposed SA method works well for this class of problem.

Discussion and conclusion

We have argued that scheduling, the 'glue' that unites high-level planning and low-level control, is increasingly important for profitable mining. After establishing a common ground by giving an overview of underground mining and the flow shop concept, representative previous research from both the mining and flow shop communities has been reviewed. A simple HFS model for underground mining was then introduced, and a number of existing state-of-the art scheduling techniques for HFS were described.

Flow shop modelling of underground mining

We have shown that cut-and-fill, room-and-pillar, and development can all be modelled as a *k*-stage HFS with uniform parallel machines, sequence-dependent set-up times, revisits, and precedence constraints between jobs.

This HFS serves as a baseline that can be extended and adapted to a specific mining process. For instance, the processing time of a machine might depend on rock mass properties, which could vary within the mine. This can be modelled as unrelated parallel machines. Furthermore, in some activities preemption is commonly allowed while in others (for instance, activities involving concrete) preemption is not possible. Some activities might also have an after-lag, such as shotcreting, where the concrete needs to cure. The machine is then free to do other jobs, while the location remains unavailable for a certain duration. Mining practice might also impose additional constraints on the HFS scheduling problem. An example could be the safety hazard of simultaneously drilling in adjacent drifts. This differs from precedence constraints between jobs in the aspect that it is dependent on both stage and job.

The HFS model presented in this paper covers only some mining methods. When the scope is expanded to other mining techniques, or to holistic modelling of the entire underground mining process, the activities in the production cycle might take place at different locations. Consider, for instance, an activity that does not take place solely at the face of a drift but at several locations, such as loading in a block caving mine. Due to the confined environment, only one machine can work at one location at a time. This means that different locations in the mine need to be allocated at different times during different stages of a job. Two ways of dealing with this come to mind. One is to explicitly model locations as necessary resources for different parts of the production cycle, and the other is to use a segregated approach where the HFS scheduling problem represents production on a higher level and routing of vehicles is left to tailored algorithms on a lower level. The routing algorithms then handle the allocation of locations in order to ensure coordinated routing. The segregated approach will naturally be suboptimal, but in the context in which the algorithm is situated it might be more suitable since machine routing may be provided by *e.g.* machine manufacturers.

Methods for solving the mine scheduling problem

When examining the previous work on scheduling in underground mining, a synthesized picture emerges. Almost all studies reported by the mining industry use optimal methods. Heuristics are sometimes introduced, but often only naïve versions are in place, only to benchmark the optimal methods. This could stem from the fact that to obtain some results with enough theoretical height, researchers are more likely to rely on rigorous mathematical programming than heuristics or metaheuristics, with which it is difficult to ensure that an optimal result has been found. However, most authors admit that the optimal methods struggle when scaling to industrial-size problems.

When considering the recent work from the flow shop community, another picture emerges. In published papers on HFS research the methods are distributed roughly as 50% heuristics, 25% optimal methods, and the rest metaheuristics and unclassified methods (Ruiz and Vázquez-Rodríguez, 2010). Historically, the development of efficient heuristics has been the focus, while lately many authors use metaheuristics. The metaheuristic approach (*e.g.* genetic algorithms) is considered by many to be the state-of-the-art for complex industrial scheduling problems.

We introduced an HFS model for underground mining in order to reap the benefit of research from other industries. A selection of relevant literature on isomorphic HFS scheduling problems includes (i) the metaheuristic implementation by Alfieri (2009) to schedule cardboard production, (ii) the benchmark of heuristics (and MIP) by Ruiz, Şerifoğlu, and Urlings (2008) which has applications in the ceramic tile industry, and (iii) the heuristics introduced by Botta-Genoulaz (2000) solving similar HFS problems with applications in *e.g*. the textile and paper industries.

Outlook and future challenges

Increased effectiveness by improving scheduling

In order to increase safety and maintain profitability in underground mining, there has been a shift from hand-held to mechanized equipment, followed by an increasing use of continuous automation (*e.g*. conveyor belts, mine hoists, crushers). Today, as noted in the survey of 200 mining executives mentioned in the Introduction, modern mines are primarily struggling with maximizing the total production effectiveness. This means that the big picture must be considered. More and more elaborate methods are being used for long-term planning, and the advent of autonomous vehicles makes the implementation of this plan more predictable. Unfortunately, the interface between these two timescales, the scheduling process, has not yet received the same amount of interest. In order to achieve a lean, transparent, and predictable mining process, high-performance scheduling will be a crucial part of the planning/production chain. The amount of published results from the flow shop community is large. However, Ruiz and Vázquez-Rodríguez (2010) note that in a selection of 200 papers on HFS scheduling the most general case with $k > 2$ stages and unrelated parallel machines is only considered in 7% of all papers. Unfortunately for the industry, it is in this category where most real-world applications are found. This lack of works calls for research not only on mine scheduling in particular, but also on complex flow shop scheduling in general.

A benefit of using automatic scheduling compared to manual scheduling is that it is easier to take into account a large amount of facts when constructing a schedule. It is, for instance, possible to integrate maintenance information directly in production scheduling. Today, production scheduling and maintenance planning are often separate activities. However, it is clear that these two topics are connected, since they both operate on the same resources: the machines. In a mine, the maintenance is often supported by some software system keeping track of e.g. machine health and spare parts. It would be beneficial if information from this maintenance system could be integrated in the scheduling algorithm. This would enable the construction of a coordinated schedule where maintenance is planned to minimize the adverse impact on production. This is especially interesting if one considers predictive maintenance, since it would allow an operator to make informed decisions that account for the trade-off between machine health and production goals. Here we considered maintenance integration; however, this is only an example of many similar benefits that can be gained from integrating other supporting activities into the scheduling process, such as integrating the management of infrastructure or rock mechanics inspections. When the number of integrated systems increases, the overall mining process becomes more coordinated and predictable.

Another important topic is the study of what metric is a suitable quantifier of schedule quality in underground mining. Most process industries have internal organizations which are evaluated by different KPIs. However, on an aggregated level, the goal is a safe and profitable mining process. In order to reach that goal, a holistic metric may be appropriate. For instance, an objective function that reflects both scheduling and routing might produce high-quality schedules. Another way would be to consider the entire value chain (from mine to port) in the objective function to tightly couple long-term and short-term goals. Most previous research in both mine scheduling and flow shop scheduling evaluates schedules based solely on the makespan. However, due to the noisy nature of mining the probable use for automatic scheduling is in an online 'rolling horizon' approach. With that in mind, is makespan a good quality indicator even though the schedule might be revised several times a day? A metric representing smart utilization of the parallel drifts might be more suitable in this case. A valuable contribution to the field would be a study of which objective functions are appropriate to use in a rolling horizon approach, and how these can be constructed to reflect common KPIs found in mining.

Acceptance in industry

Many authors note that implementing automatic scheduling in the industry is non-trivial (*e.g*. McKay, Pinedo, and Webster, 2002; Ruiz, Şerifoğlu, and Urlings, 2008; Harjunkoski, 2016). The first test of any scheduling algorithm is a comparison with previous manual methods. Therefore, to prove its value in an industrial setting, automatic scheduling must be designed such that it will be used by the manual scheduler on site. Continuing this train of thought, this means that the algorithm needs to be flexible enough to deal with most of the corner cases that the manual scheduler can handle. If it is not in the same ballpark to enable comparison, it will simply not be used. This conclusion is supported by Harjunkoski *et al.* (2014), who note that successful scheduling implementations in other industries emphasize empowering the current manual scheduler, and do not aim at replacing the scheduler. This requires an intuitive and flexible way to interface the scheduling algorithm, thus encouraging mine schedulers to actually use it. This part of automation is often overlooked by researchers, who are inclined to view design topics as second in importance to algorithm development, although from an industrial acceptance point of view the design might even be more important than the actual algorithm being used.

From published works, it is easy to recognize recent trends and the state-of-the-art at the moment. However, when targeting an industry with a lot of competition there is always an incentive to keep results secret. Compared to academia, where success is measured by published works, it is harder to estimate the state-of-the-art in mine scheduling practice, since many mines do not disclose any information about their scheduling process. Some secrecy concerns can, however, be alleviated by using flow shops. The flow shop abstraction can be used to separate algorithmic scheduling issues from sensitive process or business details, enabling the discussion to be held at an appropriate level. We believe that an open research community that supports development and collaboration would benefit the entire industry.

Summarizing, there are several opportunities to increase production effectiveness in underground mining by optimizing the scheduling process. As methods for long-term planning becomes better, and excavation is to a larger extent handled by autonomous vehicles, research on scheduling methods will become increasingly important. By addressing these challenges, the mining industry will, hopefully, stay profitable for many years to come.

Acknowledgement

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

References

Alarie, S. and Gamache, M. 2002. Overview of solution strategies used in truck dispatching systems for open pit mines. *International Journal of Surface Mining, Reclamation and Environment*, vol. 16. pp. 59–76.

Alfieri, A. 2009. Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study. *International Journal of Production Research*, vol. 47. pp. 5129–5145.

Baptiste, P., le Pape, C., and Nuijten, W. 2012. Constraint-Based Scheduling. Applying Constraint Programming to Scheduling Problems. *International Series in Operations Research & Management Science*. Springer.

Beaulieu, M. and Gamache, M. 2006. An enumeration algorithm for solving the fleet management problem in underground mines. *Computers and Operations Research*, vol. 33. pp. 1606–1624.

Botta-Genoulaz, V. 2000. Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, vol. 64. pp. 101–111.

Brah, S. A. and Loo, L.L. 1999. Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research*, vol. 113. pp. 113–122.

Burger, D. 2006. Integration of the mining plan in a mining automation system using state-of-the-art technology at De Beers Finsch mine. *Journal of the South African Institute of Mining and Metallurgy*, vol. 106. pp. 553–560.

Cheng, J., Karuno, Y., and Kise, H. 2001. A shifting bottleneck approach for a parallel-machine flowshop scheduling problem. *Journal of the Operations Research Society of Japan*, vol. 44. pp. 140–156.

Darling, P. 2011. SME Mining Engineering Handbook. Society for Mining, Metallurgy & Exploration, Littleton, CO.

Equi, L., Gallo, G., Marziale, S., and Weintraub, A. 1997. A combined transportation and scheduling problem. *European Journal of Operational Research*, vol. 97. pp. 94–104.

Floudas, C.A. and Lin, X. 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering*, vol. 28. pp. 2109–2129.

Gamache, M., Grimard, R., and Cohen, P. 2005. A shortest-path algorithm for solving the fleet management problem in underground mines. *European Journal of Operational Research*, vol. 166. pp. 497–506.

Giffler, B.,and Thompson, G.L. 1960. Algorithms for solving production-scheduling problems. *Operations Research*, vol. 8. pp. 487–503.

Graham, R.L. 1978. Combinatorial scheduling theory. *Mathematics Today: Twelve Informal Essays*. Springer. pp. 183–211.

Griva, I., Nash, S.G., and Sofer, A. 2009. Linear and Nonlinear Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Gupta, J N. 1988. Two-stage, hybrid flowshop scheduling problem. *Journal of the Operational Research Society*, vol. 39. pp. 359–364.

Gustafson, A., Lipsett, M., Schunnesson, H., Galar, D., and Kumar, U. 2014. Development of a Markov model for production performance optimisation. application for semi-automatic and manual LHD machines in underground mines. *International Journal of Mining, Reclamation and Environment*, vol. 28. pp. 342–355.

Hamrin, H. 2001. Underground mining methods and applications. W.A. Hustrulid & R.L.Bullock (eds.), Underground mining methods: Engineering fundamentals and international case studies, Littleton. *SME*. pp. 3–14.

Haouari, M., Hidri, L., and Gharbi, A. 2006. Optimal scheduling of a two-stage hybrid flow shop. *Mathematical Methods of Operations Research*, vol. 64. pp. 107–124.

Harjunkoski, I. 2016. Deploying scheduling solutions in an industrial environment. *Computers and Chemical Engineering*, vol. 91. pp. 127–135.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., and Wassick, J. 2014. Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, vol. 62. pp. 161–193.

Howes, R. and Forrest, C. 2012. Short interval control in today's underground mine: A case study. *Proceedings of MINExpo International*, Las Vegas, 24–26 September. https://edstechnologies.com/Mailer/Sep15/Newsletter_September/images/ShortIntervalControl.pdf

Hustrulid, W.A. 1982. Underground Mining Methods Handbook. Society of Mining Engineers of AIME, New York.

Kozan, E. and Liu, S.Q. 2011. Operations research for mining: a classification and literature review. *ASOR Bulletin*, vol. 30. pp. 2–23.

Kuchta, M., Newman, A., and Topal, E. 2004. Implementing a production schedule at LKAB's Kiruna mine. *Interfaces,* vol. 34, no. 2. pp. 124–134.

Little, J., Knights, P., and Topal, E. 2013. Integrated optimization of underground mine design and scheduling. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 113. pp. 775–785.

Liu, Y. and Karimi, I. 2008. Scheduling multistage batch plants with parallel units and no interstage storage. *Computers and Chemical Engineering*, vol. 32. pp. 671–693.

Martinez, M.A. and Newman, A.M. 2011. A solution approach for optimizing long- and short-term production scheduling at LKAB's Kiruna mine. *European Journal of Operational Research*, vol. 211, no. 1.pp. 184–197.

Matamoros, M.E.V. and Dimitrakopoulos, R. 2016. Stochastic short-term mine production schedule accounting for fleet allocation, operational considerations and blending restrictions. *European Journal of Operational Research*, vol. 255. pp. 911–921.

McKay, K., Pinedo, M., and Webster, S. 2002. Practice-focused research issues for scheduling systems. *Production and Operations Management*, vol. 11, no.2. pp. 249–258..

Mincom. 2011. Annual Study: Mining Executive Insights 2011. Denver, CO.

Naderi, B., Zandieh, M., Balagh, A.K.G., and Roshanaei, V. 2009. An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications*, vol. 36. pp. 9625–9633.

Nave, D. 2002. How to compare six sigma, lean and the theory of constraints. *Quality Progress*, vol. 35, no. 3. pp. 73-78.

Nawaz, M., Enscore, E.E., and Ham, I. 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, vol. 11, no. 1. pp. 91–95.

Nehring, M., Topal, E., Kizil, M., and Knights, P. 2012. Integrated short-and medium-term underground mine production schedul-ing. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 112. pp. 365–378.

Nehring, M., Topal, E., and Knights, P. 2010. Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. *Mining Technology*, vol. 119. pp. 212–220.

Nehring, M., Topal, E., and Little, J. 2010. A new mathematical programming model for production schedule optimization in underground mining operations. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 110. 437–446.

Newman, A. M., Kuchta, M., and Martinez, M. 2007. Long- and short-term production scheduling at LKAB's Kiruna mine. *Handbook Of Operations Research In Natural Resources. International Series in Operations Research and Management Science*, vol. 99. pp. 579–593. pp. 579.

Newman, A.M., Rubio, E., Caro, R., Weintraub, A., and Eurek, K. 2010. A review of operations research in mine planning. *Interfaces*, vol. 40. pp. 222–245.

O'Sullivan, D. and Newman, A. 2015. Optimization-based heuristics for underground mine scheduling. *European Journal of Operational Research*, vol. 241. pp. 248–259.

Pham, D. and Karaboga, D. 2012. Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. Springer.

Pinedo, M. 2015. Scheduling. Springer.

Ruiz, R. and Maroto, C. 2006. A genetic algorithm for hybrid flow-shops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, vol. 169. pp. 781–800.

Ruiz, R., Şerifoğlu, F.S., and Urlings, T. 2008. Modeling realistic hybrid flexible flowshop scheduling problems. *Computers and Operations Research*, vol. 35. pp. 1151–1175.

Ruiz, R. and Vázquez-Rodríguez, J.A. 2010. The hybrid flow shop scheduling problem. European Journal of Operational Research, vol. 205. pp. 1–18.

Saayman, P., Craig, I., and Camisani-Calzolari, F. 2006. Optimization of an autonomous vehicle dispatch system in an underground mine. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 106, no. 2. pp. 77–86.

Schulze, M., Rieck, J., Seifi, C., and Zimmermann, J. 2016. Machine scheduling in underground mining: an application in the potash industry. *OR Spectrum*, vol. 38. pp. 365–403.

Schulze, M, and Zimmermann, J. 2017. Staff and machine shift scheduling in a German potash mine. *Journal of Scheduling*, vol. 20, no. 6. pp. 635–656.

Song, Z., Rinne, M., and van Wageningen, A. 2013. A review of real-time optimization in underground mining production. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 113, no. 12. http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2225-62532013001200004

Song, Z., Schunnesson, H., Rinne, M., and Sturgul, J. 2015. Intelligent scheduling for underground mobile mining equipment. *PloS One*. https://doi.org/10.1371/journal.pone.0131003

Ta, C.H., Kresta, J.V., Forbes, J.F., and Marquez, H.J. 2005. A stochastic optimization approach to mine truck allocation. *Journal of Surface Mining, Reclamation and Environment*, vol. 19. pp. 162–175.

Tang, L., Liu, J., Rong, A., and Yang, Z. 2001. A review of planning and scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, vol. 133. pp. 1–20.

Williams, J., Smith, L., and Wells, P. 1972. Planning of underground copper mining. *Proceedings of the 10th International Symposium on the Application of Computer Methods in the Mineral Industry*, Johannesburg, South Africa, 10–14 April 1972. Southern African Institute of Mining and Metallurgy, Johannesburg. pp. 251–254. http://www.saimm.co.za/Conferences/Apcom72/251-Williams.pdf ◆