



Regular Research Manuscript

Selection of Discrete Event Simulation Software for Simulating Mining Operations

Abubakary Salama^{1*} and Bartłomiej Skawina^{2,3}

¹Department of Mining and Mineral Processing Engineering, University of Dar es Salaam, P.O. Box 35131, Dar es Salaam, Tanzania

²Department of Civil, Environment and Natural Resources Engineering, Division of Mining and Rock Engineering

³Luossavaara-Kiirunavaara Aktiebolag (LKAB), 983-81 Malmberget, Sweden

[†]Corresponding author: Abubakary.Salama@gmail.com

ORCID: <https://orcid.org/0000-0001-6210-978X>

ABSTRACT

Simulation is a tool that can be used to predict and evaluate the performance of mining systems. It has been used for various applications such as fleet optimization in underground mining, comparison of timing and efficiency between drills, and mine-to-mill production systems. Due to the availability of a large number of simulation tools, careful selection should be made depending on the type of problem to be simulated. The study presented in this paper aims to compare two different simulation tools, AutoMod and SimMine, by comparing the two underground loading models created. This is achieved by analysing different equipment alternatives for possible future conditions when the mine depth increases. Both tools produce statistically equivalent results for simulated production. The paper presents a discussion regarding the choice of software and based on the study, SimMine is recommended for easy, fast modelling, whereas AutoMod, with its wider palette of software features and facilities provided, is recommended for more enhanced and detailed mining simulations. The study is based on the mining operation in the Kiirunavaara underground mine in Sweden, and the simulation is conducted based on a fixed production target.

ARTICLE INFO

Submitted: Nov. 7, 2022

Revised: Jan. 29, 2023

Accepted: March 15, 2023

Published: June, 2023

Keywords: Loading system, discrete event simulation, software selection

INTRODUCTION

There are many techniques available for estimating and maintaining the number of pieces of equipment in mining, such as theoretical approaches and simulation. The theoretical estimate of the number of loading units is generally a quick method, but has the drawback of not including some important factors like the variability of tramming distances, variability of vehicle performances, queuing, and traffic congestion when more than one load-haul-

dump (LHD) is utilized (Atkinson, 1992; Raj et al., 2009). The most accurate way to estimate the number of loading units is to utilize simulation (Sturgul, 1999; Raj et al., 2009; Greberg et al., 2016). The simulation model will estimate the required number of LHDs considering the variability of the data, thus resembling the real-life scenario in a more accurate way than when using analytical methods. The use of computer simulation will allow for a much quicker evaluation of different loading units and better control of the processes of the

currently running system. The drawback of using simulation is that generating the model and required output can be a time-consuming process. Common features of these tools such as animation and graphical interface offer a direct approach to the increased understanding of a specified mining environment (Raj et al., 2009; Banks, 2004). These tools enable better interaction between the variables and the system performance in applying different rules and procedures in form of functions, probability distributions, processes, and algorithms. They also help to solve the issues related to fleet requirements or mine planning optimization problems. However, the method requires the collection of a large amount of data and often requires the data to be fitted to statistical distributions. Failure to do so may result in inaccurate values and wrong conclusions. This means that the data from different processes and the model itself should be frequently validated (Sturgul, 1999). Simulation is well-established in operations research and industrial engineering. However, much less literature exists on the results of comparison between two simulation models of the same mining system, where not only the software but also the built-in models are compared with each other. In this study, the simulation models were developed using AutoMod and SimMine tools.

METHODS AND MATERIALS

Selecting simulation software

To select the appropriate software, Nance (1995) proposed six main characteristics. The software should have the capability to generate random numbers and permit the use of other variations of random distributions. The software should also have the capability of listing the processing in such a way that the object can be created, manipulated, and deleted. Additionally, the software should include features such as statistical analysis, report generation, and time flow capability. Many software

programs meet these requirements, thus further evaluation is necessary. An example of such an evaluation was presented by Albrecht (2010), who identifies 87 discrete event simulation and modelling tools which include simulation programming languages. Albrecht (2010) then combines the guidelines for selecting the simulation software from Ahmed et al. (2003), Banks and Gibson (1997), Hlupic, et al. (1999), and Page (1994). When selecting the simulation, the procedure covers seven major areas to look at (Albrecht, 2010):

- a) Modelling Environment – varying degrees of details/levels of description, full accessibility to the underlying model code, and good readability of the code and model animation with virtual reality as a bonus.
- b) Model documentation and structure – varying levels of model description from a very high level to a very low level and independence of the system and computer architecture.
- c) Verification and Validation – tools to support verification and validation of the simulation models.
- d) Experimentation facilities – automatic batch runs including insertion of the warm-up period and independent replications with re-initialization between the runs.
- e) Statistical facilities – alternative statistical distributions together with the possibility of handling large volumes of random numbers. Additionally, the statistical facilities that enable standard experiment analysis functions such as output data analysis.
- f) User support – package maintenance, technical and promotional information including a training course, tutorials, and demonstration models.
- g) Financial and technical features – the cost of the package, ease of use,

and ease of installation together with hardware requirements.

In this study, the simulation tools AutoMod (manufacturing-oriented software) and SimMine (mining-oriented software) were used to model the Kiirunavaara mine. They were selected, as they can model the rock transportation systems of the selected mines with the necessary level of detail, and they are also commonly used simulation language environments capable of satisfying the characteristics necessary for the study. Two models were built using the two different software programs which provided opportunities to explore the variations and limitations of them both.

AutoMod

AutoMod software consists of a material movement system with a built-in simulation environment (Rohrer, 1997). The graphics offer two modes: static and dynamic where the moving objects can be observed during the simulation run. The flexibility comes from AutoMod syntax and built-in environments. There is a possibility to modify various parameters via AutoMod syntax or directly in the movement system such as speed, turning speed, acceleration, or deceleration (Banks, 2004; Yuriy and Runciman, 2013). In addition to the AutoMod syntax, the software includes built-in templates: vehicle path mover system, conveyors, automated storage and retrieval system, bridge cranes, power and free conveyors, and kinematic (robotic) systems. These features, combined with panel control in form of drag-and-drop and .m logic create powerful simulation software that can communicate with control systems and other simulation models. Performance reports with statistics and 3-D animation are created automatically, providing a realistic and statistically accurate view of the system, and helping to verify and validate the models. Advanced debugging and trace facilities enable easy tracking of errors and flaws (Banks et al., 2010). In the AutoMod the loads move through

processes and compete for resources (Banks et al., 2010). The AutoMod also includes AutoView, AutoStat, the model zip archiver, the OPC utility, the process server, and the web tools utilities such as SimController or ACE graphics editor (Banks, 2004). In SimController it is possible to view the results of the model and redirect useful output files for multiple runs. SimController also enables comparison of the reports, viewing of Gantt charts, and viewing the model information. The ACE is the graphics editor that offers direct control of the graphics represented in a model. With ACE graphics you can create, edit and delete the model elements. AutoStat is an extension to AutoMod used to design experiments by running multiple simulations under different statistical constraints (Banks et al., 2010). AutoView is an extension to AutoMod where video clips from the AutoMod models can be made. While making the movie there is a set of customization tools that enable the user to follow the vehicles with the camera in the recorded simulation run or set the camera to certain views for the specified time. The possible movie formats are AVI and MPEG.

SimMine

SimMine is a discrete event simulation software that uses a full graphical interface and does not require coding. The logic and behaviour of different entities are already built into the software and can be controlled through their properties. To incorporate the randomness/variance in the processes the tool utilizes statistical distributions and uses a point-and-click interactive spreadsheet for entering the data. An extra feature is the interactive 3D layout which allows the user to point-and-click and set properties directly from the 3D layout but also to visually inspect and analyse the sequence of development in the animation of the development process. Since it is developed specifically for the simulation of mining operations, it is made of interfaces

where the user can set and specify underground mine-related properties such as rock properties, face profiles, activity cycles, etc. Equipment interface and delays can be visually analysed for better evaluation of the statistical results generated by the model (Yuriy and Runciman, 2013). SimMine software offers a direct import of the mine layouts (CAD layouts), comprehensive model output statistics, the capability to evaluate the design of the production facilities, and the selection of production equipment such as trucks and loaders. Results come from simulating real-life underground data that incorporates all the equipment performance data, operational data, and other relevant data such as rock properties.

Case study

The Kiirunavaara mine is an underground iron-ore mine located in the northern part of Sweden that uses the sublevel caving mining method (Figure 1). The mine consists of a high-grade magnetite deposit approximately 4 km long, with a dip of about 65° from the horizontal plane and with an average thickness between 80 and 100 meters in the northeast direction.

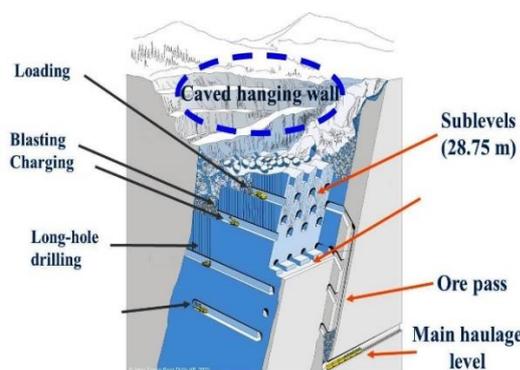


Figure 1: A large-scale sublevel caving mining method (modified after Atlas Copco).

Currently, one electric LHD of 25-tonne capacity works from 6 a.m. until 10 p.m. before blasting, while one semi-automated diesel LHD machine with 21-tonne capacity works from 10 p.m. to 6 a.m. on

each block. These machines load the ore from draw points within each production drift and transport the ore to the ore passes. Large trains, operating on the main level, transport the ore from ore passes to a crusher, which breaks the ore into pieces of four inches or less in size for subsequent hoisting to the surface, which happens through a series of vertical shafts. Once the block starts being mined, to comply with the mining restrictions, continuous production of the production blocks must be maintained until all available ore is removed. Currently, the mine produces around 27 Mtonnes of crude ore per year, and the goal is to reach a production of 35 Mtonnes of crude ore per year from all ten blocks. To achieve this target, the mine set a plan to have a future daily production of 10,000 tonnes on each block (Salama et al., 2015).

Model settings

The analysed production area in this study (Figure 2) consists of 17 production drifts, each with a length of approximately 100 meters, and four ore passes located close to the main drift. A three-meter blasting round on each production drift starts in sequence at the hanging wall, commonly using an upward raise to provide a free face, and then retreats toward the footwall. Mucking out by LHD continues until the waste dilution reaches the set limit. The amount of ore produced is approximately 10,000 tonnes from one blasted ring, which makes the total ore to be mined from this block approximately 6.17 Mtonnes. It is assumed that loading starts from the first drift on the left side of the block and continues to the next drift until the last drift is finished. The same procedure is repeated until the whole production area is mined.

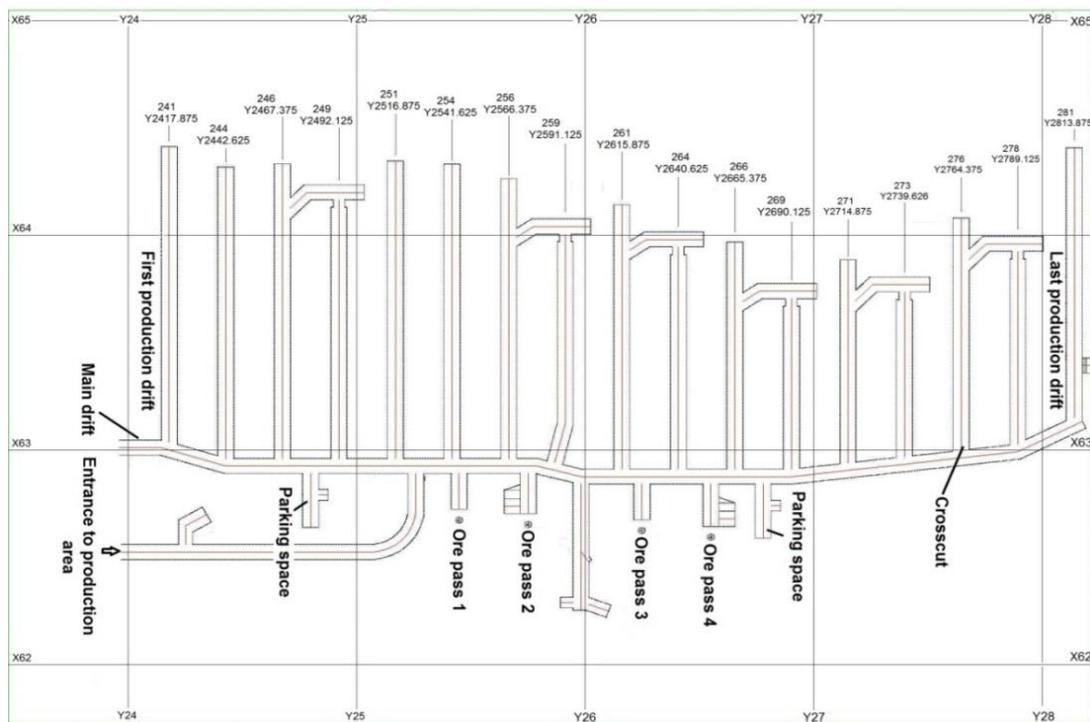


Figure 2: Production area considered in this study (Salama et al., 2015).

As seen in Figure 2, two ‘parking areas’ were defined, which were used as waiting places for LHD machines during breaks. When dumping, a machine working on the left-hand side of the mine block will use the first two ore passes, while when working on the other side of the block, the machine will dump material to the remaining ore passes. The usage of the two ore passes on each side of the working area depends on the buffer capacity and boulder frequency. If the buffer capacity limit is reached, then the LHD operator will move to the next ore pass. If a boulder enters the ore pass, the LHD operator will wait for up to five minutes; when the waiting time exceeds five minutes, the LHD operator will move to the next ore pass. If the rock breaker is working to clear the boulders, and at the same time the next ore pass buffer capacity limit is reached, the LHD operator will wait for the rock breaker to finish removing the boulders. The simulation of the base case model includes one 25-tonne electric LHD machine working from 6 a.m. until 10 p.m. before blasting, and one 21-tonne diesel LHD machine working from 10 p.m. until 6 a.m. The results from the base case

scenario are compared with the results from the analysed scenarios.

The procedure of building SimMine and AutoMod models begins with importing the planned layout into the graphical environment. The next step is to define all the necessary parameters such as performance characteristics of the equipment, rock properties (ore/waste), face profile, mining cycles, shift schedules, blasting times, and/or costs. Further actions are to implement necessary procedures and interactions between the operating machines. In SimMine this is done by choosing appropriate options in point-and-click menus. In the case of AutoMod, it is necessary to incorporate the programming code. The base case model is developed to benchmark against the planned development/production rate or already existing system. Further steps involve running alternative scenarios and verification and validation of the models utilizing the physical and statistical tests. During the process of model development, the reporting and documentation on various optimization scenarios including variations such as additional equipment, shift, or

different performance characteristics are arranged. The final stage involves making recommendations to improve or optimize the studied processes in the system.

AutoMod cycle settings

To emulate the movement of LHD machines the AutoMod uses a process-interaction method. The LHD machine would move until it arrives at the particular location, gets delayed or leaves the process. The process is created and controlled in the AutoMod by use of source codes (.m

```

begin P_vehicle arriving procedure
  call F_FindNextWorkLoc(VvehPtr(Ai_ID))
  set VvehPtr(Ai_ID) A_WorkLoc = A_WorkLoc
  call F_SetOperParam(VvehPtr(Ai_ID))
  dispatch VvehPtr(Ai_ID) to A_dest
  while A_Done = 0 begin
    if V_locData(A_Drift, A_PointIndex) > 0 begin
      set VvehPtr(Ai_ID) A_dest = A_dest
      dispatch VvehPtr(Ai_ID) to A_WorkLoc
      wait to be ordered on OL_vehicle(procindex)
      wait for u Vload_mean, Vload_STD sec
      if V_locData(A_Drift, A_PointIndex) < VvehPtr(Ai_ID) A_Capacity then
        set VvehPtr(Ai_ID) A_current_amount to V_locData(A_Drift, A_PointIndex)
      else
        set VvehPtr(Ai_ID) A_current_amount = VvehPtr(Ai_ID) A_Capacity
        dec V_locData(A_Drift, A_PointIndex) by VvehPtr(Ai_ID) A_current_amount
        if VvehPtr(Ai_ID) defined velocity <> 0 then
          call F_SetSpeed(VvehPtr(Ai_ID), Vr_Machine_speed(Ai_ID, Loaded), Vr_Machine_acc(Ai_ID, Loaded))
        dispatch VvehPtr(Ai_ID) to A_dest
        wait to be ordered on OL_vehicle(procindex)
        if V_orepass = 2 then begin
          wait for u Vdump_mean, Vdump_STD sec
        end
      else begin
        call F_boulder(V_boulder(A_destIndex), VvehPtr(Ai_ID) A_destIndex)
        wait for u Vdump_mean_STD sec
      end
      inc V_received(A_destIndex) by VvehPtr(Ai_ID) A_current_amount
      increment C_tonnage by VvehPtr(Ai_ID) A_current_amount
      inc C_cycles by 1
      call F_SetSpeed(VvehPtr(Ai_ID), Vr_Machine_speed(Ai_ID, Empty), Vr_Machine_acc(Ai_ID, Empty))
    end
  end
  call F_FindNextWorkLoc(VvehPtr(Ai_ID))
  call F_update(VvehPtr(proc)
end
end

```

Figure 3: AutoMod cycle settings.

Subsequent statements ensure that the work is continued until all of the material has been drawn from the production drifts and if necessary appropriate changes have been made to the specified destinations. This is done by sending the LHD to the assigned working locations. Upon arrival, if necessary, the machine is sent to the order list to await further instructions. Further down the list, the loading information is updated. After updating the machine and production area information, the function that sets speed is invoked to send the LHD machine to the closest dumping location. If there is a boulder in the ore pass, the LHD

syntax) or through point-and-click control panels. An example of such a process used in this study for diesel and electric LHD machines is shown in Figure 3. The code emulates the cycle procedure. The process begins by calling the function that finds the next work location to which the LHD machine should travel. Next is to set operational parameters. The operational parameters, in this case, are the bucket capacities, acceleration, deceleration, turning time, and empty/loaded vehicle speeds.

operator waits for the ore pass to be cleared and then dumps the material into the ore pass. However, if the time to dump the material into the ore pass is predicted to take longer than five minutes the LHD operator drives on and dumps the material in the adjacent ore pass. Afterwards, the information such as the amount of transported material is updated. Next, the cycle repeats until all the material has been extracted.

SimMine cycle settings

In SimMine, to emulate the loading and transportation of the LHD machines the

user should use the point-and-click menus. There is no need to use coding and, since the software is mining-related, many additional features can be implemented into the model by checking the right option from the drop-down menu list. For each activity, a set of additional properties can be changed depending on the requirements. Further adjustments can be made in the mine layout. To control the movement of the LHD machines, the transport method (LHD, Truck) is set from the loading location to the chosen destination in the 'Loading and transport activity' tab. The fleet properties can be changed by navigating to the 'Fleet' tab submenu. The user can set the occurrences of boulder frequency based on either tonnage (mean tonnage between failures) or rounds (risk of boulder per round) but also set the time to fix the boulders based on selecting the probability distribution. At the end of the simulation, the results are presented under several tabs. Each tab consists of different information such as location and vehicle sequences but also production statistics, time statistics, vehicle statistics, and cost information. The location and vehicle sequences are presented in the form of Gantt charts or tables. Others covers a wide range of information such as the number of cycles, travelled distance, working/idle time, utilization, downtime, breaks, operating costs, or investment costs.

Input data

Input data for the simulation consists of the mine layout (Figure 2), availability of the vehicles, availability of the production areas, and operational data of the vehicles. Data was collected from the mine and the equipment manufacturer. Before model formulation, data analysis from different sources was required to ensure valid simulation results. The techniques used for data analysis vary depending on the amount of available data, when available data is too small or data is missing, and when there is no data available. In cases when more data

is available, such data can be fitted to probability distributions that characterize the uncertainty and randomness of the operation. In some cases, when only a small amount of data is available, an attempt to fit into distributions may become inappropriate, and then the empirical distribution (actual data values), point estimate, uniform, or triangular distribution can be applied. When no data is available due to, for example, the fact that the system being modelled does not yet exist, a subjective estimate can be made based on guesses and assumptions. However, since the input data provide the driving force for the simulation, a careful check of the physical characteristics, limits, or nature of the process should be performed by experts.

The input data (Table 1) shows the operational parameters for all analysed machines. The diesel LH 621D is the 21-tonne engine machine, while LH 625E, LH 514E, and LH 621E are the electric LHDs with 25-tonne, 14-tonne, and 21-tonne bucket capacities respectively. The electric LH 621E is not yet released onto the market, and thus the data for this machine was subjectively estimated based on guesses and assumptions by experts from the manufacturer. The simulation results were used to obtain the LHD operating times, after excluding time lost for maintenance, production area availability, meal breaks, shift changes, and interference with other mining activities, such as drilling, charging, and ore pass maintenance. The production area availability due to other mining activities was estimated to be 80%, which was applied for all LHDs. In the simulation, a machine availability of 90% was used for all LHDs. The value was based on the mean time between failure (MTBF) of 85.5 hours and the mean time to repair (MTTR) of 9.5 hours for the diesel LH 621D (Gustafson et al., 2013).

Table 1. Operational parameters

Machine type	Forward & Reverse speed (km/h)	Acceleration & Deceleration (m/s ²)	Bucket capacity (tonnes)	Drive power (kW)
	Empty/Loaded	Empty/Loaded		
LH514E	23.5/20.5	0.1/0.05-0.6	25	315
LH625E	18/16.0	0.1/0.05-0.6	21	283
LH621D	33.9/28	0.1/0.08-0.6	14	132
LH621E	21/18	0.1/0.08-0.6	21	345

Scenarios

Several combinations of machines were analysed for fifteen different scenarios (Table 2). The machines were scheduled to work from 6:00 a.m. until 00:30 a.m. before blasting which comprises 18.5 hours. After the blasting, the LHD was scheduled to work for 3.5 hours from 2:30 a.m. to 6:00 a.m. In the simulation, all breaks and delays were excluded to obtain the actual, effective machine working time. The

simulation ended when the LHD operators finished removing all the available ore from the studied production areas. The results obtained from the simulation were then statistically analysed. As depicted in Table 2, scenarios one to twelve were selected to analyse the performance when only one type of machine is used in all available time, and the remaining three scenarios were analysed when machines work only before blasting.

Table 2. The Fifteen Simulated Scenarios

Scenario number	6:00 a.m. to 0:30 a.m.				2:30 a.m. to 6:00 a.m.		
	Bucket capacity (tonnes)	LHD type	Scheduled time (hours)	No. of LHDs	LHD type	Scheduled time (hours)	No. of LHDs
1	25	Electric	18.5	1	Electric	3.5	1
2	25	Electric	18.5	2	Electric	3.5	2
3	21	Electric	18.5	1	Electric	3.5	1
4	21	Electric	18.5	2	Electric	3.5	2
5	14	Electric	18.5	1	Electric	3.5	1
6	14	Electric	18.5	2	Electric	3.5	2
7	21	Diesel	18.5	1	Diesel	3.5	1
8	21	Diesel	18.5	2	Diesel	3.5	2
9	25	Electric	18.5	2	Electric	3.5	1
10	21	Electric	18.5	2	Electric	3.5	1
11	14	Electric	18.5	2	Electric	3.5	1
12	21	Diesel	18.5	2	Diesel	3.5	1
13	25	Electric	18.5	2			
14	21	Diesel	18.5	2			
15	21	Electric	18.5	2			

The first eight scenarios involved the analysis of the same type (electric or diesel) and the same number (one or two) of LHDs

of all capacities operating for the duration of the whole scheduled time (6:00 a.m. to 0:30 a.m. and 2:30 a.m. to 6:00 a.m.). In all

of the analysed scenarios, when two LHDs are employed, each machine will work on one side and use separate ore passes, meaning that there is no interaction of the LHD machines during production.

The 9th to 12th scenarios involved two electric or diesel LHDs working from 6 a.m. until 0:30 a.m. before blasting, and only one electric or diesel LHD working after blasting. The aim is to check the possibility of using two LHDs before blasting and one LHD after the blast. This is because of the short available time after the blast until the new shift machine schedule begins. After the blast the same LHD, which was working in a manual mode, will be prepared to work in semi-automated mode. The arrangement of these changes will delay the loading operation and reduce the LHD's effective working time.

The last three scenarios were selected to check if it is worth loading from morning until the time before blasting and to include no loading activities after blasting. The scenarios involved two electric 25-tonne LHD machines, and two diesel 21-tonne LHD machines. The evaluation of these scenarios was based on the production rate, energy consumption, and gas emissions. Summary of the scenario settings:

- a) The first setting involves one or two LHD machines operating from morning until before the blast, and one or two after the blast.
- b) The second setting involves two electric or diesel LHD machines operating from morning until before the blast, and only one electric or diesel LHD machine after the blast.
- c) The last setting involves two electric or diesel LHD machines from morning until before the blast, and no LHD machine after the blast.

Model validation and verification

Verification is the process of ensuring that the conceptual model design has been transformed into a computer model with sufficient accuracy. Validation is the

process of ensuring that the model is sufficiently accurate for a certain purpose (Muller, 2014). Verification and validation of the model aims to ensure that the created model is accurate and represents the real system. Several techniques can be used for verification, such as testing the model logic, using debugging techniques, running the model under varying conditions, making logic flow diagrams, or building diagnostics into the model. The validation can be performed by using a degenerate test, testing internal validity, using an extreme condition test, comparing historical data, testing face validity, comparing output results with an actual system, or through a Turing test (Banks et al., 2010). A verified and validated simulation model should provide results that are very close to those seen in the actual operating system. In this study, verification was done using debugging techniques, animations check, model inspection, and by running the model under varying conditions. The debugging features were used to make sure that everything was running correctly before resuming execution. Simulation runs were initially conducted with the conceptual estimated size of equipment, storage facilities, and haulage systems structures. Initial results allow these parameters to be redefined and radically changed; changes involve extra programming but enhance the versatility of the program to conform with proposed mine logistics. The validation was done by using internal validity and by comparing the output from the model with the output from the real system.

RESULTS AND DISCUSSION

Results from the simulation runs

The simulation was first conducted for the base case scenario, which involves using one 25-tonne electric LHD machine and one 21-tonne semi-automated diesel LHD machine. The current loading and hauling operations modelled in AutoMod and SimMine simulation software for the base

case scenario produce daily averages of 6,035 tonnes and 6,120 tonnes, respectively. This was seen to be close to the real-life production in the mine. Next, the scenarios shown in Table 2 were run to compare the output difference between AutoMod and SimMine models. The results are shown in Figure 4.

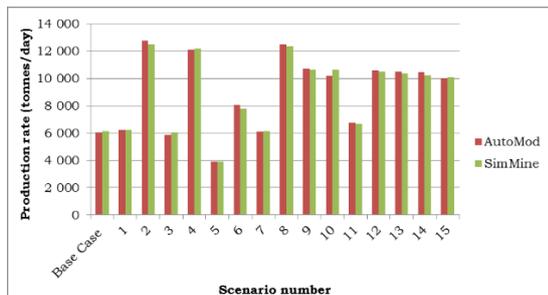


Figure 4. Daily production rates for the selected scenarios

When comparing AutoMod and SimMine simulation results (Figure 4), the highest variation in the production rates is observed in the 10th scenario (432 tonnes) whereas the lowest variation in the production rates is observed in the 1st scenario (similar results). The 1st and 5th scenarios can be considered to achieve the smallest variation. These scenarios involve the use of one electric LHD machine from 6 a.m. until 0:30 a.m. before blasting, and one electric LHD machine after the blast. The 2nd, 6th, 10th, and 14th scenarios can be considered to achieve the highest difference variation. These scenarios involve the use of two electric or diesel LHD machines from 6 a.m. until 0:30 a.m. before blasting, and one, two, or none of the electric LHD machines after the blast.

The result shows that simulation using these two different simulation software results in similar production rates, thus suggesting that both software tools are capable of capturing similar results when simulating an underground loading environment. Different results (Figure 4) are observed since the loading and dumping operations are continuous and were simulated with randomly generated input

parameters (based on the specified probability distributions).

Statistical and practical relevance for the AutoMod and SimMine models

The simulation analysis in this research was conducted using both AutoMod and SimMine simulation software. To increase the credibility and validate the model assumptions, output results and various settings were discussed and improved with the help of experts who are knowledgeable about the studied system and processes. From experience and from observations the SimMine model and AutoMod model in this study provide similar results. Despite the similarity observed in these tools, it is still necessary to investigate the statistical relevance of the two separate simulations. In other words, to compare the relative performance of two or more system designs (Banks et al., 2010). For this the statistical technique correlated sampling, also known as Common Random Number (CRN), was used. This means that the same random numbers are used to simulate both alternative system designs. However, the same is not true when comparing between the relative performance of the AutoMod model and the relative performance of the SimMine model. In this case, independent sampling (different random numbers) is used. In order to evaluate the tools under the same conditions, a statistical hypothetical comparison using t-test was performed. Statistical significance tests tell us how likely it is that there will be differences between sample groups. The t-test was chosen because the variances of the two models were not known, and the sample size was less than 25 (Kanji, 1999; Cohen et al., 2013). The sample size of 16 from both models was used and then tested with a five percent level of significance. The purpose is to test the null hypothesis (H_0) of the two population means, which is: $H_0: \mu_1 = \mu_2$, where μ_1 and μ_2 stand for the mean daily production rate, energy consumption, gas emissions for diesel units, and time to mine the entire block

from AutoMod and SimMine simulation models, respectively. The hypothesis is accepted if true; otherwise, it is rejected based on obtained P-Value. The P-value is

the probability of getting the results (or more extreme results) given that the null hypothesis is true. The result of this test is shown in Table 3.

Table 3. Results of t-test for two population means: AutoMod and SimMine

Parameter	AutoMod		SimMine		Statistical relevance	
	Mean	Std	Mean	Std	P-Value	Mean diff.
Production rate (tonnes/day)	8,927	2,772	8,907	2,728	0.984	20
Time to mine block (days)	757	302	756	300	0.905	1

To accept or reject the null hypothesis, the obtained P-values were compared to a 0.05 significance level. The condition is to reject the null hypothesis if the P-value is less than or equal to 0.05; otherwise, the null hypothesis is accepted. As can be seen in Table 3, the P-value is greater than 0.05, which means the null hypothesis is accepted for all analysed parameters. This means that there is no significant difference between the means of the daily production rate and time to mine for the entire mine block from the AutoMod and SimMine simulation models. This indicates that both simulation models produce statistically equivalent results.

Software performance

Currently, when making an evaluation the user tends to focus not only on the product design arena but also on the high software performance that is both short and reliable; therefore, the results of the runtimes for AutoMod and SimMine models are

presented in Figure 5. Each tool had a varying number of tools available for processing. There was also the possibility to run the model in real time in the built-in environment or run the model in windowless mode. In case of SimMine, the time was clocked at the end of each run, and in the case of the AutoMod, the time was read straight from the output report. To reduce the external influence of bias and make sure that the AutoMod and SimMine models both were run in a similar environment, the same computer settings were used. This meant disabling the Wi-Fi connection and reducing the number of programs running in the background. Additionally, the simulation models were run multiple numbers of times to report differences. When changing the seed number, the difference in simulation runtimes ranged within tenths of seconds and the difference between multiple simulation runtimes did not vary by more than few seconds, thus an average runtime value was used.

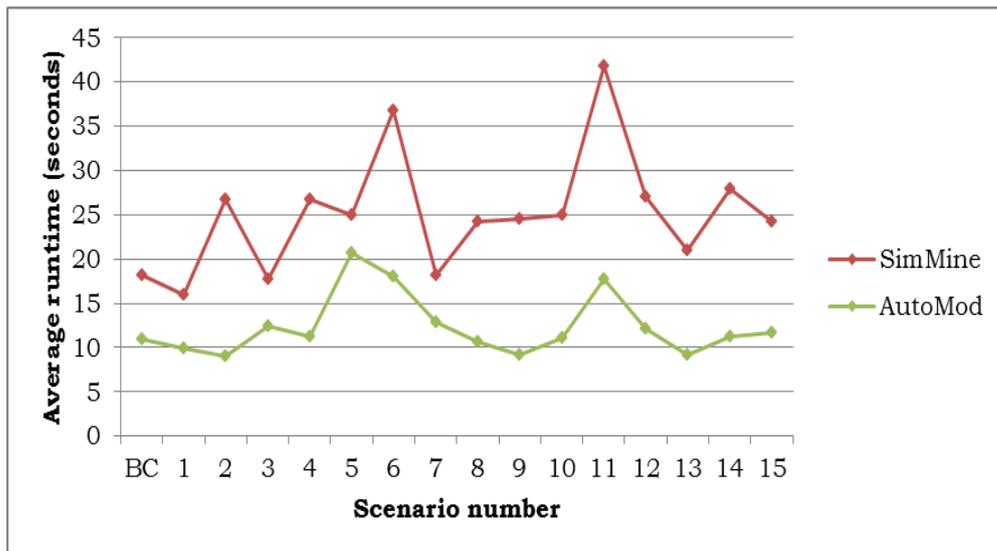


Figure 5: Comparison of average simulation runtimes for AutoMod and SimMine models.

The results in Figure 5 range between 9 to 21 seconds and 16 to 42 seconds for the AutoMod and SimMine respectively. An observation was made that using more than one LHD of the same type in the SimMine model created complexity that increased the simulation runtime. Even when one LHD was used it still took longer to finish loading operations. Examples of these occurrences are between 1st and 2nd scenario, 3rd and 4th scenario, 5th and 6th scenario and 7th and 8th scenario. However, it does not mean that bucket size and speed of the vehicle had not influenced the simulation runtime. For example, the 5th, 6th, and 11th scenarios when running 14-tonne electric LHD produced the longest runtimes. In this case, the runtimes are the longest with reaching values of over 40 seconds of CPU time clock for scenario number 11. Overall, the differences between the SimMine and the AutoMod runtimes do not significantly vary, however, the AutoMod might take a longer time to simulate the scenarios depending on the way the model is constructed/coded. Observations were made that running the AutoMod in windowless mode with no switched functions improved the simulation runtime without affecting the

output results. What this means is that there is room for further improvement of the codes. For example, using the logic that invokes reporting the loaded material in the graphical user interface (GUI), each time the vehicle performs a cycle, consumes more time than using the logic that invokes reporting at the end of the simulation run. This line of code is not necessary for the windowless run and therefore can be excluded from the windowless simulation run. However, if necessary, the line of the removed code can be reconnected again later.

Simulation software evaluation

The AutoMod and SimMine software evaluation (Table 4) is based on the modelling package evaluation table from Albrecht (2010). The software features presented in Table 4 differ extensively favouring the AutoMod due to its wide range of additional facilities that are provided for modelling. Certain categories have to be further elaborated such as the time to build the model, the structure of the system, and system capabilities.

Table 4: AutoMod and SimMine package evaluation

Category		Modelling package	
		AutoMod	SimMine
Modeling environment			
1.	Model development environment provided	yes	yes
2.	Alternative model creation methods	yes	yes
3.	Visual modelling (drag-and-drop)	yes	yes
4.	Accessibility of model code	yes	no
5.	Ease of use	no	yes
5.	Animation	yes	yes
6.	Virtual reality	no	no
Model documentation and structure			
1.	Operating system independence	no	no
2.	Model chaining (Linking outputs from different models)	yes	no
3.	Efficient translation to executable form	yes	yes
4.	Alternative worldviews	no	no
Verification and validation			
1.	Facilitates the verification of simulation models	yes	no
2.	Provides interactive model checking	yes	no
3.	Trace file provided	yes	no
Experimentation facilities			
1.	Automatic batch run	yes	no
2.	Warm-up periods	yes	no
3.	Independent replications of experiments	yes	no
4.	Re-initialization (initializing again)	yes	no
5.	Speed adjustment	yes	no
6.	Breakpoints (intentional stopping)	yes	no
7.	Experimental design capability	yes	no
8.	Automatic determination of run length	yes	yes
Statistical facilities			
1.	Alternative statistical distributions	yes	yes
2.	User-defined distributions	yes	no
3.	Random number streams	yes	yes
4.	User-specified seeds of random number streams	yes	yes
5.	Distribution fitting	no	no
6.	Goodness-of-fit tests	no	no
7.	Output data analysis	yes	yes
User support			
1.	User manuals	yes	yes
2.	Technical and promotional information (e-mail, internet, discussion groups)	yes	yes
3.	Lecturer's guide for educational licenses	yes	yes

4.	Tutorial	yes	no
5.	Training course	yes	yes
6.	Package maintenance	yes	yes
7.	Portability	no	no
8.	File conversion	no	no
Financial and technical features			
1.	Package cost includes life cycle maintenance costs	yes	yes
2.	Ease of installation	yes	yes
3.	Low hardware/system requirements	yes	yes
4.	Frequent and comprehensive updates	yes	yes

Depending on the size and complexity of the model, it would usually take much longer for an experienced user to build the AutoMod model than the SimMine model. This is excluding the time needed to gather the input data and run any other tests such as verification and validation tests. In AutoMod, it is possible to code all the parameters and functions however this increases the development time and level of necessary skills required for coding. For example, in case of the AutoMod, the degree of details controllable within the model depends on the user (how and what is included in the model), whereas in case of the SimMine there are already many parameters that are useful and it does not require extra time spent on programming the input and output parameters. The same applies to the readability of model code. Therefore, the degree of details controllable within the model is higher for AutoMod and readability depends on the user as opposed to the SimMine tool. However, the user has an advantage when it comes to SimMine since it already has a mining infrastructure and many mining-related parameters already encoded in the software.

The SimMine software produces mining-related reports providing additional information such as development time, production outputs, or cost-related information. For the AutoMod software, extra coding is required to generate the output in the appropriate form, otherwise, the report comes in a standard and advanced format that includes traces of the variables, paths, etc. For

both AutoMod and SimMine the portability is limited to the Windows operating system. Portability, in this case, is the usability of the same software in different operating systems. When it comes to the incorporation of the layouts there is a possibility to import the layouts to both software programs. In the case of AutoMod, the files can be imported apart from AutoMod (.mod or .arc) from SIMUL8 or as .sdx files. In the case of the SimMine, the files can be a database (.mdb extension) or AutoCAD files. In case of the AutoMod, the imported AutoCAD layout (if without an extension that transforms the AutoCAD layout to a suitable package) requires creating additional paths on top of the imported layout. The AutoMod provides interactive checking control in form of the internal run debugger (IRC) that is a perfect tracer of the bugs and functionality of the model created. This means the user can go line by line and see how the program handles and changes the variables of the created code. There is a possibility to do the same in the SimMine, however, it is not available for the normal user and is restricted to the development package. When it comes to random number generation, the AutoMod uses Combined Multiple Recursive Generator (CMRG) to generate a reproducible sequence of random numbers of length up to 2^{191} distributed continuously and uniformly on the open interval 0 to 1 (Banks, 2004; Banks et al., 2010). Whenever random numbers are required the AutoMod software draws a sample from the sequence of numbers. The sequence of random numbers is split into streams of length up to 2^{127} (each

stream is split into 2^{51} random number sets, each with a length of 2^{76} . These random numbers, if necessary, are then converted to the desired statistical distribution or random variate (Banks, 2004). The random variate is a variable generated from uniformly distributed numbers, for example, the random variate represents the random event in the model (Banks, 2004). Several probability distributions are available in AutoMod such as continuous distribution (user-defined based on the cumulative frequency), exponential distribution, gamma distribution, lognormal distribution, one of distribution (user-defined based on the frequency with the possibility to mix two distributions), normal distribution, triangular distribution, uniform distribution, Weibull distribution, next of distribution (based on round-robin manner) and constant distribution (Brook Automation Inc., 2003). Additionally, there is a possibility of encoding the user-specified distribution in the syntax. In the case of the SimMine software, the generation of the random numbers is performed by use of a linear congruential generator (LCG). The LCG is the most widely used technique for generating random numbers where the initial seed for a linear congruential random-number generator is the integer value that initializes the random-number sequence (Banks et al., 2010). Whenever random numbers are required, the SimMine software draws a sample from the sequence of numbers and converts them to the desired statistical distribution or random variate. The probability distributions available in SimMine are triangular distribution, normal distribution, uniform distribution, lognormal distribution, exponential distribution, or constant distribution.

There are also differences in software capabilities. For example, let's suppose that two trucks enter the loading area. The first truck requires two buckets to fill up, whereas the second requires only one bucket to do so. However, there is only one loader at the moment. In this case, let's consider three alternatives. First: neither of the trucks claims the loader. Second: the first truck (two

buckets to fill) claims the loader and waits for another load. Thirdly: the second truck (one load to fill up) claims the loader and continues to transport the ore further downstream, releasing the location for the first truck. Now, in the case of the SimMine, the possibilities are to follow the First-In-First-Out (FIFO) selection in which the first truck would claim the loader or follow the Last-In-First-Out (LIFO) selection in which the last truck that had entered the waiting list would claim the loader. When it comes to AutoMod the flexibility is higher and offers the possibility not only to claim the loader according to FIFO rule or LIFO rule but also any other method that can be implemented as a code. For example, the selection can be adjusted according to any dispatching rule based on the processing time, due date, number of operations, machine priority, according to a particular event, or any other user-defined rule.

Software recommendations

In the case of the AutoMod software, the model uses different processes in the system to interact with each other. This is achieved by focusing on the job-driven model, whereas, in the case of the SimMine, the model is already there with preinstalled functionalities and options related to underground mining operations. From that perspective, even though the AutoMod software has better functionality and flexibility in building a new mining model it is still considered a time-consuming and complex task for the user. Furthermore, AutoMod is not mine-related software, therefore the user has to understand and decide which parameters to use as input and output values. When it comes to SimMine software, there is room for improvement since the user may require extra functionality in the software. For additional parameters, there is a development group at SimMine that could incorporate the mine-specific changes/parameters depending on the customer's needs. However, it is important to use a software program which can imitate the real-world environment to an acceptable level

of detail depending on the requirements and objectives of the model, since otherwise the simulation results can differ and mislead in providing the information to the user. Nonetheless, the SimMine software is recommended for easy and fast modelling, whereas for more enhanced and detailed simulations the AutoMod is recommended, with its wider palette of software features and facilities provided.

CONCLUSION

This research compares AutoMod and SimMine models by simulating the Kiirunavaara mine with different LHD machines, with different sizes, operating between the production draw points and a group of ore passes. The analysis was based on the daily production and energy consumption and gas emissions using discrete event simulation. The following conclusions have been reached:

- a) AutoMod and SimMine models achieve adequate and statistically equivalent production results.
- b) Software features differ extensively favouring AutoMod due to its wide range of additional facilities that are provided for modelling.
- c) In AutoMod the degree of details controllable within the model and readability depends on the user. This means that all the parameters and functions have to be coded by the user increasing the development time and level of the necessary skills required for coding.
- d) In SimMine the parameters and functions are preinstalled providing easy and fast access in developing a simulation model related to mining operations.
- e) SimMine is recommended for easy and fast modelling, whereas for more enhanced and detailed simulations AutoMod is recommended, with its wider palette of software features and facilities provided.

Future studies may involve the detailed evaluation of the capital and operating costs

of the proposed loading equipment models but also enable further enhancement towards adaptive control. Future work should incorporate adaptive control and auto-tuning that can be easily implemented in the typical process control computer. It should have the ability to present information to the user/operator at the right time from the system including the directions and consequences of the selection being made. Implementation of process control units would help to reduce the variability, increase efficiency and maintain the production targets.

ACKNOWLEDGMENT

The authors acknowledge the I² Mine project within the EU 7th framework program for funding this research and Luossavaara-Kiirunavaara Aktiebolag (LKAB) as to a great extent both helped towards the completion of this work.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

- Ahmed, R., Hall, T., and Wernick, P., (2003). A Proposed Framework for Evaluating Software Process Simulation Models, In Proceedings of the Prosim'03 May 3-4, 2003, Portland State University, USA.
- Albrecht, M.C., (2010). Introduction to DES Available at: <http://www.albrechts.com/mike/DES/Introduction%20to%20DES.pdf>.
- Atkinson, T., (1992), Selection and sizing of excavating equipment, in SME Mining Engineering Handbook, H.L Hartman, 2nd ed., Littleton CO SME, 1311–1333.
- Banks, J., (2004). Getting started with AutoMod, Bountiful, UT: Brooks-PRI Automation, USA.
- Banks, J. and Gibson, R., (1997). Selecting simulation software, IIE Solutions, pp. 30-32. 1991_0003.pdf (ncsu.edu)
- Banks, J., Carson, J.S., Nelson, B.L., and Nicol, D.M., (2010). Discrete Event System Simulation, Pearson Education, New Jersey, USA.

- Brooks Automation Inc., (2003). Procedures dictionary. Brooks Automation Inc., USA.
- Cohen, J., Cohen, P., West, S.G., and Aiken, L.S., (2003). Applied multiple regression/correlation analysis for the behavioral sciences. Lawrence Erlbaum Associates, Mahwah. <https://doi.org/10.4324/9781410606266>
- Greberg, J., Salama, A., Gustafson, A., and Skawina, B., (2016). Alternative Process Flow for Underground Mining Operations: Analysis of Conceptual Transport Methods Using Discrete Event Simulation. *Minerals* **6**(3): 65. <https://doi.org/10.3390/min6030065>.
- Gustafson, A., Schunnesson, H., Galar, D., and Kumar, U., (2013). Production and maintenance performance analysis: Manual versus semiautomatic LHDs. *Journal of Quality in Maintenance Engineering*, **19**(1):74–88. <https://doi.org/10.1108/13552511311304492>.
- Hlupic, V., Irani, Z., and Paul, R.J., (1999). Evaluation framework for simulation software, *International Journal of Advanced Manufacturing Technology*, **15**(5):366-382. <https://doi.org/10.1007/s001700050079>.
- Kanji, G., (1999). 100 statistical tests. Sage: Thousand Oaks, CA, 23–30.
- Muller, D., (2014). AutoMod-modeling complex manufacturing, distribution, and logistics, system for over 30 years, WSC 14: in the Proceeding of the 2014 Winter Simulation Conference, S. Jain, R.R Creasey, J. Himmelspach, K.P. White, and M. FuA. Tolk, S.Y. Diallo, I.O. Ryzhov, L. Yilmaz, S. Buckley, and J.A. Miller, eds., Salt Lake City, UT 84116 USA.
- Nance, R.E., (1995). A history of discrete event simulation programming languages. *ACM SIGPLAN Notices*, **28**(3):149-175. <https://doi.org/10.1145/234286.1057822>.
- Page, E. H., (1994). Simulation Modeling Methodology Principles and Etiology of Decision Support, Ph.D. Dissertation Virginia Polytechnic Institute and State University, USA.
- Raj, M.G., Vardhan, H., and Rao, Y.V., (2009). Production optimization using simulation models in mines: a critical review. *International Journal of Operational Research*, **6**(3):330–359. <https://doi.org/10.1504/IJOR.2009.026937>.
- Rohrer, M.W., (1997). AutoMod tutorial, WSC 97: In Proceedings of the 1997 Winter Simulation Conference, Andradottir, K.J.H., Withers, D.H., and Nelson, B.L. eds., USA, 657-662.
- Salama, A., Greberg, J., Skawina, B., and Gustafson, A., (2015). Analyzing energy consumption and gas emissions of loading equipment in underground mining, *CIM Journal*, **6**(4):179-188.
- Sturgul, J.R., (1999). Discrete mine system simulation in the United States, *International Journal of Mining Reclamation and Environment*, **13**(2): 37–41. <https://doi.org/10.1080/09208119908944207>.
- Yuriy, G. and Runciman, N., (2013). Discrete-event simulation in underground mining – applicability and experience at Vale Canada operations. World Mining Congress 05-WMC, Canada. <https://doi.org/10.1016/J.POLYMER.2005.01.083>