Fundamental behaviours of production traffic in underground mine haulage ramps

David Haviland a, Joshua Marshall b,⇑

a Golder Associates Ltd., Burnaby V5C 6C6, Canada
b The Robert M. Buchan Department of Mining, Queen’s University, Kingston K7L 3N6, Canada

Article info
Article history:
Received 2 March 2014
Received in revised form 15 April 2014
Accepted 15 July 2014
Available online 7 February 2015

Keywords:
Underground mining
Fleet management
Discrete-time simulation
Vehicle dispatching

Abstract
Ramps (or declines) are often used in underground mines to transport ore, waste, materials, and personnel. This paper studies mine ramp productivity and presents results from a set of computer simulations designed to model the fundamental behaviours of ramp haulage systems. Simulations show that, under fundamental assumptions without random disturbances, the haulage system always converges to a periodic behaviour in the steady state, but that productivities vary between equilibria. Simulations also demonstrate how productivity per vehicle does not necessarily decrease as more vehicles are added and, for example, in the five-vehicle case, how a 3.1% improvement can be achieved over the use of four vehicles. The result reveals the inefficiency of commonly-used lockout-style vehicle coordination strategies, and suggests a possible avenue for improving the productivity of haulage ramps by controlling the system to achieve more productive behaviours.

© 2015 Published by Elsevier B.V. on behalf of China University of Mining & Technology.

1. Introduction
This paper demonstrates how the periodic patterns, underlying the behaviour of vehicles that operate on underground mine ramps, might be used to improve the overall productivity of a ramp-based materials transport system. To the best of our knowledge, such observations have never been reported before.

All underground mines face the challenge of productively transporting ore to the surface for processing. Shaft hoisting is often the most efficient means of moving the ore, which is normally blasted rock, but in some instances (especially in shallow mines) the high cost of shaft-sinking is not justified [1,2]. In these and other specialized cases, the burden of ore and/or waste rock haulage is passed on to mine trucks, which travel up and down a narrow access ramp (sometimes also called a decline). Fig. 1 provides an example configuration where the access to the mine ore body is by way of the levels that are connected to a single-lane spiral ramp. The example ramp portal and example haulage machines are shown in Fig. 2.

When left uncontrolled, traffic flow in mine ramps can be congested, inefficient, and even dangerous. To mitigate these problems, some operators use lockout systems to manage the flow of vehicles. When a vehicle enters the segment of ramp between two levels, for example, all other vehicles are “locked out” of that segment. This can be done with a traffic light or by voice-over-radio. When lights are used they can be automatically or manually activated (e.g., by a pull-rope system). Once the vehicle reaches the next level, the segment is unlocked. While such systems do improve safety, they leave room for improvement in terms of efficiency. An ideal system would minimize delays in the ramp and optimize productivity, e.g., the system with the maximum rate at which material is delivered to the top of the ramp.

To this end, a discrete-time simulator was created with MATLAB R2011b. Although not necessary, specialized tools such as Arena, GPSS, SimEvents, could also have been used. A variety of simulations were run with simple ramp layouts and small vehicle fleets with a common lockout policy as a baseline. The goal was to develop an essential understanding of the dynamics of traffic flow in narrow mine haulage ramps. Based on these output data, this paper describes four fundamental observations about emergent behaviours and the relative mine productivity under different system parameters. Sets of more complex simulations were also run with larger ramps and more vehicles in an attempt to extend these hypotheses to situations more like those found in producing mines. The observations made in this study aim to provide the foundation for a new and systematic approach to improving the productivity of underground haulage ramps.

1.1. Underground fleet management

The monitoring of traffic flow and real-time dispatching of vehicles in open pit mining is now commonplace in industry, and
improvements based on the use of satellite-based GPS continue to be developed. However, there is very little in the published literature about underground mine ramp traffic dynamics and, also, about the coordination of machines on such ramps. The most relevant work is perhaps that of [4], who used the GPSS/H programming language to simulate ramp traffic flow at the Greens Creek Mine in Alaska. Their simulations were based on two different traffic management policies. In the first model, only one vehicle was allowed on a given ramp segment at any time (i.e., a lockout system). The second model gave priority to upward-travelling vehicles and allowed more than one vehicle to travel upward in the same segment. Due to the prioritizing, only downward-travelling vehicles utilised the passing bays. In both cases the capacity of any given passing bay was restricted to just one vehicle. The findings from their simulations were used to assist in determining the feasibility of adding an additional decline to the Greens Creek Mine in Alaska to handle return mine traffic.

Although little has been published about ramp productivity, there has been some research on the problem of allocating tasks to production vehicles on mine levels. For example, the problem of “dispatching” underground mine vehicles on levels has been studied to resolve conflicts and to plan collision-free routes [5]. Part of this work is involved with resolving bi-directional conflicts, which are also present on mine ramps. However, on underground mine ramps there is only one possible route which is wide enough for only one vehicle. And so the problem is not one of dispatching but rather of coordination for optimal productivity.

Others have also tackled the fleet management problem for underground mines [5]. For instance, the advanced approaches to real-time fleet management for underground mines, where routing and scheduling of machines on a level are handled simultaneously, were developed [6–8]. Their work also incorporated the presence of traffic [6] and had to account for the so-called “displacement mode”; i.e., whether the vehicle arrived at its destination in the correct orientation, e.g., forward or reverse [9].

However, none of these examples have explicitly studied the problem of ramp management. Yet, when ramps are used for haulage, this can be the question where productivity is either achieved or broken, and the constraints associated with operating on a ramp are different in comparison with the more general problem of mine dispatching.

1.2. Other related problems

Railway scheduling may be similar to ramp traffic management. Single rail lines are required to handle bi-directional traffic, and in this case passing is only possible at stations along the line. The previous studies investigated traffic flow on these types of single-line rail systems and introduced a useful graphical tool to show the changing positions of all of the trains over time [10]. Other studies have shown how individual train velocities can be strategically varied to minimize the delays and fuel consumption in complex rail systems [11]. Some of the ideas used in rail traffic management could also be applied to traffic management in mine ramps. One important distinction is that railway traffic tends to be precisely scheduled, and is subject to much less variance than the flow of trucks on mine ramps, which makes the rail problem somewhat different from ours.

Path planning for autonomous robots has been studied extensively, but (to our knowledge) never in the specific context of underground mine ramps. The existing research addressed a similar traffic flow problem and dealt with path planning for multiple autonomous vehicles in narrow tunnel environments [12]. Their model described tunnel networks (similar to the underground workings on one level of a mine) as a graph made up of nodes and connecting edges. Each vehicle in the system was assigned a “goal node” that represented its intended destination. A multi-step sequential algorithm was then used to plan the trajectories of each vehicle, in a way it would avoid collisions and ensure that all goals were met. Their method was shown to always guarantee a solution with relatively low computational cost, but the generated solutions were not necessarily optimal in terms of time required to meet all goals. Their method was shown to be useful for large, complex layouts with many vehicles, which would ordinarily require significant computational power to optimise (if a solution is reached at all). Given the fairly simple linear layouts of mine ramps, an alternative path-planning strategy is desirable.

Another related example is the work where they studied the problem (in the context of an autonomous mobile robot) of repeatedly gathering data at some locations and uploading the data to other locations [13,14]. In some respects, this problem is analogous to a typical load-haul-dump cycle in underground mining. These authors attempt to minimize the maximum time between data uploads (or dumps, in mining productivity terms). However, their preliminary work deals only with a single robot and the coordination of multiple robots is the future work.

2. Ramp traffic simulation

This section introduces the simulation model, its underlying assumptions and the key model parameters and policies, which were studied in our numerical experiments. Although simulation has been widely used in the mining industry as a decision-making tool [15,16], what distinguishes this paper is the use of the simulation results to analyse and understand the patterns of vehicle behaviours, rather than trying to output realistic system performance data for comparison studies, e.g., for equipment selection, mine design, identification of bottlenecks.
2.1. Model assumptions

As for any numerical simulator, it is important to recognise that the output results are, at best, approximations of reality. In this case, the simulator was designed with the following set of model limitations and assumptions in mind: (1) Ramps are used for ore haulage only, so there is no light vehicle traffic; (2) All of the vehicles in the ramp are assumed to be identical; (3) Vehicle speeds are assumed to be fixed and not subject to random variations; (4) Loading and dumping times are assumed to be fixed with no random variations; (5) Passing bays are allowed to hold more than one vehicle at a time, if necessary; (6) Delays associated with in pulling into/out of passing bays are not simulated; and (7) Delays due to acceleration/deceleration of the vehicles are not simulated.

Clearly, as is the case for every model, these assumptions are not perfectly realistic. However, these assumptions were purposefully and prudently chosen. The primary goal of the simulations described in this paper was to allow for the observation of recurring patterns and underlying behaviours displayed by the vehicles, without the effect of random disturbances (work is ongoing to systematically re-introduce such details, but this is beyond the scope of the current paper.). It is important to recognise that the strategically simplifying simulator model in these ways allowed for generalisation of the behaviours by removing the veils of complexity and uncertainty. It would surely be present in a more detailed model (e.g., with random disturbances and a non-homogeneous fleet).

2.2. Parameterization of the haulage ramp

Luckily, the haulage ramps at most mines tend to have similar basic features. The ramp is a spiralling and narrow path that leads from a surface dumpsite to loading point at its base. Numerous passing bays and level entrances are along the way, which allow vehicles to pass one another. These features are consistent in all kinds of mines, and they lend themselves well to an intuitive parameterization strategy.

For the purpose of simulation, the three-dimensional ramp was simplified to a linear one-dimensional path consisting of \( s \in N \) segments of ramp that are connected by \( s + 1 \) “nodes”, where \( N \) denotes the set of natural number not including 0. Node 0 is designated as the surface dump, and node \( s \), the loading point. Nodes 1 through \( s - 1 \) represent the passing bays and level entrances along the length of the ramp. A vector \( L = (L_1, L_2, \ldots, L_s) \) defines the lengths of the ramp segments between each node. Node 1 is the length of the ramp segment between nodes 0 and 1, \( L_2 \) is the length of the segment connecting nodes 1 and 2, and so on. These parameters define the geometry of the ramp. Fig. 3 illustrates these (and other) ramp parameters for a simple one-truck scenario.

For simplicity it was assumed that haul trucks were the only vehicles operated on the mine ramps. These vehicles originate at the surface and head to the base of the ramp to load. Once loaded, the vehicles return to surface to deliver their load to the dump.

This cycle is repeated perpetually for all vehicles in the simulation. In the simulator, a mine’s haul truck fleet consists of \( n = N \) vehicles. In all of the simulations in this study it was assumed that the vehicles were identical and that they were capable of travelling up the ramp at a constant speed of \( v_{up} \), and down the ramp at a speed of \( v_{down} \). The simulation algorithm was designed around these few basic input parameters.

2.3. Algorithm indexing

A discrete-time simulation was used to model the flow of traffic on the ramp. This approach required knowledge of the positions, speeds, and directions of travel of all of the vehicles at each time step. A \( 3 \times n \) array named \( T \) (for trucks) was created to contain this information for a given point in time. \( T \) has three rows, such that \( T = \{ p; v; d \} \), where \( p, v, \) and \( d \) are each vectors with \( n \) elements. In other words, \( T \) describes the state of the simulation at any given time step.

The first row of \( T \) describes the positions \( p \) of the vehicles. Each element of the vector \( p \) corresponds to a particular vehicle. Vehicle positions are described as real numbers which range from 0 to \( s \), inclusively, i.e., \( p_1, p_2, \ldots, p_n \in [0,s] \).

The second row of \( T \) contains the speeds \( v \) of all of the vehicles at a given time. These elements can take on values \( v_1, v_2, \ldots, v_n \in (-v_{up}, v_{down}, 0) \). Vehicles travelling up the ramp are given a negative speed for calculation purposes. As they ascend the ramp, their corresponding values of \( p \) decrease gradually to zero. Vehicles travelling down the ramp have increasing \( p \) values, in time. Any vehicles that are stopped at nodes have temporarily constant \( p \) values.

The third and final row of \( T \) describes the direction of travel \( d \) of each of the vehicles in the ramp. Vehicles heading up the ramp are assigned a value of 1, while those travelling down the ramp are given a value of 0, i.e., \( d_1, d_2, \ldots, d_n \in \{0,1 \} \). The values in \( d \) change only when a vehicle reaches a load or dump point. For any vehicles waiting at “interior nodes”, the corresponding values in \( d \) describe their intended directions of travel, or the direction that they will take when their path is clear. For haulage ramps, loaded trucks are given a value of 1, and unloaded trucks are given a value of 0.

2.4. Traffic flow policies

As already mentioned, lockout systems are commonly used to manage traffic in busy underground ramps. When a vehicle enters a segment of ramp, a switch is activated to “lock” that segment by displaying red lights at both ends. No other vehicles may enter the segment until the original vehicle has passed completely through and unlocked it. This paper focuses on imitating this common type of traffic management policy.

A base time step of 1 s was chosen for the discrete-time simulation. In each time step, the vehicles move according to the most recent values stored in the trucks array. The vehicle positions \( p \) are updated according to the vehicles’ speeds \( v \) and their directions of travel \( d \). In each time step it is also necessary to check for conditions, which might cause one or more vehicles to alter their speeds or directions of travel. The rules that define these conditions and govern the decision-making processes are laid out by the traffic management policy.

2.4.1. Basic lockout system

Under a lockout policy, a vehicle will only change its speed when it arrives at or leaves a node. It will only change its direction of travel when it reaches the surface dump or the loading point. In this paper, these discrete events are referred to as breakpoints, because they cause the simulator to break away from its base algorithm in order to make traffic management decisions. These breakpoints are further subdivided into predictable and unpredictable
break points. This breakpoint classification and handling scheme are based on the one used in the field of hybrid systems modelling [17].

Vehicle arrivals at nodes are called predictable breakpoints. The exact times at which each vehicle will next arrive at a node can be calculated easily at any time from the values in the trucks array, in conjunction with the known layout of the ramp. The algorithm creates a vector \( x \in R^n \) that stores the times at which each vehicle will next arrive at a node. The minimum value in \( x \) always corresponds to next time \( t \) at which any of the vehicles will arrive at a node. In other words, \( t^* = \min x \), which represents the time at which the next predictable break point will occur.

Vehicle departures from nodes are classified as unpredictable breakpoints. The exact timing of these events is more difficult to calculate in advance because it can depend on the positions and motions of any of the other vehicles in the system (i.e., a vehicle cannot leave a node if there is another vehicle in its way). Because of the difficulty in forecasting their occurrences, the simulator checks for these events at every iteration.

With these two event scenarios in mind, the lockout policy algorithm was built around three main steps, which are described as follows:

Step 1: Check for predictable breakpoints. Let \( t \) denote the current simulation time. At any time during the simulation, if \( t < t^* = \min x < t + 1 \), then it is known that at least one vehicle will arrive at a node in the next time step of the simulation. When it occurs, the algorithm first adjusts the length of the time step so that the next iteration will end precisely when the arrival occurs. For all vehicles arriving at nodes (usually only one), the algorithm then decides if the conditions allow the vehicles to continue on their paths, or if they must stop and wait before proceeding. If a vehicle has arrived at a load or dump point it must always stop, and its direction of travel is reversed. If no vehicles will arrive at nodes in the next time step, then the positions are updated based on the speeds and directions of the vehicles.

Step 2: Check for unpredictable break points. With the updated vehicle positions, the next step for the algorithm is to check if any of the currently stationary vehicles can be released from their nodes. If there are no stationary vehicles, this step is skipped. For all vehicles waiting at nodes, the algorithm checks whether the conditions are met for their release. In the lockout case, there must be no other vehicles travelling in the segment so that the waiting vehicle intends to enter. If this condition is met, the corresponding element in \( v \) is updated. The vehicle will then leave the node in the next time step. The decision to release a vehicle waiting at a loading or dumping point follows the same logic, albeit slightly more complex due to queuing delays and the time taken to load or dump.

Step 3: Update \( x \). Frequent updating of \( x \) is critical to ensure that no predictable break points are missed. To do so, the distance through which each truck must travel before it will reach a node is calculated, and then it was divided by the speed at which the truck is travelling.

2.4.2. Priority-based policies

In a priority-based policy, instead of allowing trucks to enter segments of ramp on a basis of “first-come, first-served”, priority is given to the vehicles that are travelling up the ramp. This also prevents vehicles from being locked out by other vehicles that are travelling in the same direction up or down the ramp. In order to work in a real mine, this type of system requires more precise information about vehicle locations than the above lockout systems. This information might be supplied, for example, by a system, which involves RFID tags on the vehicles and is strategically placed tag-readers throughout the ramp, or by a underground navigation system [18,19]. This paper does not present the results about priority-based policies, but our simulation efforts show that priority-based policies can be analysed in a similar way to the lock-out policy.

2.4.3. Delays, queuing, and piggybacking

Reasonable (constant) delays \( t_{\text{load}} \) and \( t_{\text{dump}} \) were added to the simulation to account for the time it takes to load and dump trucks, respectively. Queuing behaviour was also incorporated to model to reflect the fact that a vehicle cannot begin loading or dumping until the vehicle ahead of it has finished.

In theory, the lockout system allows only one truck to travel in a given ramp segment at a time. In actual practice, however, vehicles that are closely following one another tend to “piggyback” by following the lead vehicle into the segment even though they should be locked out. To model this behaviour, the conditions to allow passing through nodes (Step 1), and the conditions for release from nodes (Step 2) were both modified.

Basic lockout rules were still applied, but they were overruled if the algorithm found a vehicle travelling in the right direction (or hypotheses) made from the outputs of the performed simula-

### Table 1

Ramp parameters in Section 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L (m) )</td>
<td>(200, 200, 200)</td>
</tr>
<tr>
<td>( t_{\text{load}} (s) )</td>
<td>30</td>
</tr>
<tr>
<td>( t_{\text{dump}} (s) )</td>
<td>15</td>
</tr>
<tr>
<td>( v_{\text{up}} (m/s) )</td>
<td>5</td>
</tr>
<tr>
<td>( v_{\text{down}} (m/s) )</td>
<td>10</td>
</tr>
</tbody>
</table>

3. Fundamental results

Under the lockout policy described in Section 0, a very large number of computer simulations were run by using the simulator described in Section 0. This section presents several fundamental observations based on relatively short and symmetrical ramps, which are designed with evenly spaced passing bays when there are only two or three trucks in the haulage system. These experiments all used the parameters shown in Table 1. Simulations based on more realistic parameters (e.g., for loading and dumping times) are presented in Section 4.

The goal of these simulations was to gain a fundamental understanding of traffic behaviour within mine ramps in a way which could possibly be extended to more realistic cases. It is important to note that, in all of the observations made in this section, the significance of the results lies in the trends observed, and not in the specific output values. The input parameters in Table 1 were chosen primarily for the ease of observation. The specific numerical results of each of the simulations are subject to variation when the parameters are changed. The observations made here do not only hold for the parameters shown in Table 1. Section 0 demonstrates that, although the outputs may change drastically, the fundamental observations made from these initial simulations can be extended to more complex scenarios, with more vehicles and more realistic ramp environments.

The remainder of this section describes four key observations (or hypotheses) made from the outputs of the performed simula-
tion experiments: (1) The system always converges to a steady-state, periodic behaviour on limit sets in the simulation state space; (2) Some steady-state limit sets are more productive than others; (3) Some limit sets are reached more frequently than others; and (4) Per-vehicle productivity diminishes as fleet size increases.

3.1. Convergence to steady-state

Simulations, which are run from a very large set of random initial conditions, showed that, shortly after releasing the vehicles, the traffic flow consistently converges to some form of steady state, which presents the periodic cycle. In steady state, each vehicle predictably repeats a set of actions, stopping at the same nodes and delaying for the same lengths of time in every load-haul-dump-return cycle. It was also observed that there can be many different steady-state configurations. In this paper, these steady-state cycles are referred to as limit sets. This term is borrowed from the field of dynamic systems theory, where it is used to describe the state to which a system will converge in time. Parker et al. [20] define a limit set as “the set of points in a state space that a trajectory repeatedly visits”.

Recall that the state of our haul ramp system at any time $t$ is given by $T$. It follows that the “state space” refers to the continuous set of all possible configurations of $T$. As the vehicles move through the ramp system, they define a trajectory through the state space that is described by the elements of $T$ at each time step. When the traffic flow converges to a periodic cycle (i.e., a limit set), this trajectory forms a loop. The set of points (snapshots of $T$) that make up the loop are what define the particular limit set.

To be completely correct, the definition of the state of a system must also account for the current queuing conditions at the loading and dumping points, because the state of any queues in the ramp cannot be determined by the values in $T$. It is conceivable that, in some cases, the limit set that a system converges to after initialization with a given $T$ could change depending on the queuing conditions at the time of the initialization. This difference would only be of significance in simulations which are initialized with vehicles in the processes of loading, dumping, or queuing, and would not change our fundamental observation.

It is implied by the definition of a limit set that the steady state configuration is determined by the system’s initial condition. In this way, each limit set can be thought of as having a convergence field within the state space, and any initialization state that falls within that field will eventually converge to that limit set. By definition, the convergence fields of any two limit sets in a given system must be mutually exclusive. If the assumption is made that all possible states of initialization will eventually converge to steady state behaviour, then the state space can also be completely defined as the set of convergence fields of all the possible limit sets.

In the cases of two and three vehicle ramp systems, the collective behaviour of the vehicles can be observed by plotting the positions of the vehicles against one another over time in Cartesian coordinates. This type of plot is shown for the two-vehicle case in Fig. 4. The loops in Fig. 4 are generated by mapping the first row of $T$ (which contains the vehicle position vector, $(p_1, p_2)$) over time while the system operates in steady state. Each distinct loop represents a different steady state configuration.

In Fig. 4, the overall symmetry of the plot about the line $p_1 = p_2$ is explained by the reassignment of vehicle numbers within the same type of limit set. Because the vehicles are identical, reassigning the vehicle numbers has no practical consequence other than complicating the task of identifying unique limit sets. The loops labelled as “B” in Fig. 4 actually represent the same vehicle behaviours, but with the opposite assignment of numbers. The loop labelled as “A” can also have a mirrored configuration, but it is not visible on the plot because the loop itself is symmetrical about the line of reflection.

For two-vehicle ramp systems, a given limit set can have two possible vehicle numbering configurations. In the general case with $n$ vehicles, the vehicle numbers can be assigned in $n!$ different ways. These reconfigurations are in fact distinct limit sets, but in this paper they are grouped as one. This assumption is made for simplicity, because the collective vehicle behaviours and productivities achieved are the same.

When a ramp system has converged to a steady state, plotting the positions of the vehicles over time allows for a clear visualisation of the individual vehicle behaviours characteristic of the limit set in which the system is operating. It is shown for the two-vehicle case in Fig. 5. The upper plot shows the two-vehicle system operating in limit set “A”, and the lower plot shows the same system operating in limit set “B”.

3.2. Some more productive limit sets than others

Let productivity be determined by the average time taken for a vehicle to complete a single load-haul-dump-return cycle while operating in steady state, and the shorter cycle times represent the greater productivity. Simulations were repeatedly run from random initializations of $T$. By exhaustive simulation in this way, nearly all of the possible limit sets for the system (in both the two-vehicle and three-vehicle cases) could be observed. It was discovered that, in the same ramp with the same number of vehicles, some of the limit sets were more productive than others.
In these tests, all of the vehicles were observed to complete just one load-haul-dump-return cycle in each iteration of a limit set period (i.e., the vehicle cycle times equaled the limit set periods). The discussion in Section 0 will show that this is not always the case, and we should necessitate a clear distinction between the measurements of limit set period lengths and average vehicle cycle times when discussing ramp productivity. With two vehicles operating in the short ramp, the vehicle cycle times in the most productive limit set ("B") were found to be 4.0% shorter than those in the least productive limit set ("A"). In the three-vehicle case, the difference was 7.4%. In mining, due to scale and timeframe of operations, even small improvements, such as this, can result in significant revenue.

3.3. Some limit sets reaching more frequently than others

One convenient method of cataloguing limit sets is by their periods. Using the same repeated random initialization technique described in Section 0, the periods of all of the possible limit sets in the basic two-vehicle and three-vehicle cases were recorded. After many simulations were randomly initialized and run to convergence, conclusions could be made regarding the relative likelihoods of convergence to limit sets with different periods. Using this technique, it was found that some limit set periods occurred significantly more frequently than others. The results from the basic two-vehicle and three-vehicle simulations are presented as histograms in Fig. 6.

If the assumption is made that each bar in the histograms corresponds to one unique limit set the result demonstrating the varying likelihoods of convergence implies that the convergence fields of the different limit sets are of non-uniform size. With random initializations, the frequency of convergence to a particular limit set must be directly proportional to the size of its convergence field. Unfortunately, the results in Section 0 demonstrate that it is possible and common for otherwise unique limit sets to have the same period lengths, which complicates the analysis. In any case, it is still a significant result to note that certain limit set periods and vehicle cycle times appear to occur more frequently than others.

3.4. Per vehicle diminishing productivity with fleet size increasing

With all other factors left unchanged, the three-vehicle simulations produced limit sets with longer average cycle times than the two-vehicle simulations. Because of the increased delays due to traffic congestion in the three-vehicle case, the individual vehicles were required to stop and wait more frequently, and/or for longer periods of time. As expected, the overall productivity of the ramp (the total haulage rate) was seen to increase with the addition of extra vehicles, but the average productivity per vehicle decreased.

For comparison, the simulation was also run with just one vehicle on the ramp in order to create a benchmark by removing all delays associated with traffic congestion. In that case, the lone vehicle was able to travel up and down the ramp without stopping at any of the passing bays, and it was able to achieve a cycle time of 225 s. This cycle time can be viewed as a benchmark against which the per-vehicle productivities of the two-vehicle and three-vehicle cases can be measured in the same ramp environment. With two vehicles, the average cycle time increased to 244.8 s, which indicated a decrease in per-vehicle productivity of 8.8%. With the third vehicle added, the average per-vehicle productivity decreased by 22.1% compared with the one-vehicle case.

4. Advanced ramp scenario

The set of simulations described in this section were run for a larger ramp environment made up of seven 250-m-long ramp segments, and the number of vehicles deployed in the ramp varied from one to nine. The input parameters for the simulations are shown in Table 2. Although these parameters were selected to approximate reality (based on discussions with mine operators), the objective of the simulations was to draw general conclusions about traffic patterns rather than to model a specific scenario. Relatively small changes to the vehicle speeds due to some factors (e.g., type of machine or ramp grade) do not change the resulting fundamental observations about the existence of limit cycles in ramp traffic.

The primary goal of these simulations was to determine if the basic observations made in Section 0 could be extended to the cases involving a larger ramp and more vehicles, which would be more typical of a real mining operation. This paper presents the results of simulations involving up to five vehicles (Simulations were run up to nine vehicles. But, for brevity’s sake, the most important findings can be explained with only five vehicles).

4.1. One-vehicle baseline

With no traffic congestion from other vehicles, a lone vehicle can complete one full cycle in 645 s for the ramp parameterized by Table 2. This represents the minimum possible per-vehicle cycle time, which serves as a baseline for what follows.

4.2. Two and three vehicles

Fig. 7 shows the possible limit sets for the two-vehicle case. After accounting for symmetry, there are three possible limit sets. Interestingly, despite displaying very distinct behaviours, they all have exactly the same period of 660 s, which demonstrates that analysis of the histogram plots alone (e.g., Fig. 6) is not sufficient.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L )</td>
<td>(250, 250, 250, 250, 250, 250, 250)</td>
<td>m</td>
</tr>
<tr>
<td>( r_{\text{load}} )</td>
<td>90</td>
<td>s</td>
</tr>
<tr>
<td>( v_{\text{dump}} )</td>
<td>30</td>
<td>s</td>
</tr>
<tr>
<td>( v_{\text{up}} )</td>
<td>5</td>
<td>m/s</td>
</tr>
<tr>
<td>( v_{\text{down}} )</td>
<td>10</td>
<td>m/s</td>
</tr>
</tbody>
</table>
4.3. Four and five vehicles

Comparison of the four-vehicle and five-vehicle cases represents the main result of this section. With four vehicles in the same eight-node ramp, all of the limit sets included only one cycle per period. Repeated simulations revealed only two possible limit set periods and an average period length of 735.1 s, which is an increase of 52.2 s over the three-vehicle case. Once again, it is worth noting that, although there were only two different possible limit set periods identified, the system was shown to regularly converge to more than two types of limit sets.

With the fifth vehicle added, the simulator produced some very interesting and unexpected results. Approximately 75% of the randomly generated initial states were observed to converge to limit sets with surprising period lengths of 33,455 s, which allowed every vehicle to complete 46 cycles per period. 20% of the simulations yielded results are more like those observed in the other tests, having period lengths of 1425 s with two cycles in each period. More infrequently, the system converged to limit sets with periods that had even greater lengths of 243,300 s, with 322 cycles per period.

Most interestingly, it was found that with five vehicles, a significant majority of the initial states converged to behaviours that yielded average cycle times of just over 727 s. These included all of the cases reported to have converged to limit sets with periods of 33,455 and 243,300 s, along with the few non-convergence cases. The cases with shorter limit set periods (1425 s) had notably shorter cycle times of just 712.5 s.

Oddly, these vehicle cycle times were shorter than any of the cycle times achieved in the four-vehicle simulations. The most productive limits set in the five-vehicle case achieved a 3.1% improvement in per-vehicle production over the most productive four-vehicle configurations. This is counterintuitive result because one would expect that adding vehicles to the system should result in the increased congestion and, thus, the decreased productivity per unit vehicle.

These results make a strong case for the potential of multi-vehicle coordination in mine haulage ramps. At the very least, this anomaly serves as an indication that the basic lockout system is certainly a sub-optimal policy. Ideally, the four-vehicle case should be able to at least duplicate and likely improve upon the results achieved in the five-vehicle case. This can be easily demonstrated by considering the addition of a "phantom vehicle" to the same four-vehicle simulation, which would perturb the behaviour and allow the system to achieve the improved cycle times reached in the five-vehicle case.

Fig. 10 compares the fastest limit sets in the two scenarios. Qualitatively, one can see that the five-vehicle scenario keeps the
vehicles fairly evenly spread throughout the ramp, effectively avoiding delays due to queuing. The opposite was true in the four-vehicle case, where some of the vehicles trailed each other closely, resulting in alternating periods of high traffic congestion followed by long intervals of inactivity at the loading and dumping points. A more optimal policy would clearly stabilize and distribute the arrivals of vehicles at the loading and dumping points in order to minimize queuing delays.

5. Conclusions

(1) This study demonstrates that, at a fundamental level, periodic patterns underlie the behaviours of vehicles that operate in environments designed to resemble underground mine ramps. Understanding these patterns is important because, as is shown by way of simulations, the overall productivity of a ramp depends on the type of steady-state configuration (limit set) to which the system converges.

(2) Different limit sets achieve different productivities, even with the same number of vehicles operating in the same ramp layout. Some of the limit sets resulted in “bunching” of the vehicles together, sometimes causing queuing at the loading or dumping points, while others did a better job of distributing the vehicles throughout the ramp. Practically, the simulations show that there would be productivity advantages to preventing this kind of congestion.

(3) In general, it seems intuitive that the productivity achieved by each vehicle in a ramp should decrease gradually as more and more vehicles are added, as a result of the increasing traffic congestion. However, the simulations revealed two surprising exceptions that could be exploited. These vehicle cycle times were shorter than any of the cycle times achieved in the four-vehicle simulations. The most productive limits set in the five-vehicle case achieved a 3.1% improvement in per-vehicle production over the most productive four-vehicle configurations.

(4) The results from the simulation tests provide a strong case for the potential of improving upon the non-optimal basic lockout policy, albeit within an idealised ramp environment. For example, the inclusion of a “virtual vehicle” in the four-vehicle system could be used to improve productivity. The notion of asymptotically stable limit sets may be affected by the inclusion of stochastic variables and other random perturbations. But the existence of fundamental behaviours clearly provides significant insight about the problem of ramp production optimization.

Acknowledgments

Thanks to Ivan Zelina and Dennis Kattowitz (Micromine Pty Ltd., Western Australia) and Jon Peck (Peck Tech Consulting Ltd., Montreal, Canada) for fruitful discussions about the problems and realities of underground ramp management. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (No. 371452-2009).

References