Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

SIMULATION OF MIXED DISCRETE AND CONTINUOUS SYSTEMS: AN IRON ORE TERMINAL EXAMPLE

Vincent Béchard, Eng., MASc.
Normand Côté, Eng.

SNC-Lavalin Inc., GTS
360 St-Jacques West., Suite 800
Montréal, H2Y 1P5
CANADA

ABSTRACT

Modeling industrial systems involving discrete and continuous processes is a challenge for practitioners. A simulation approach to handle these situations is based on flow rate discretization (instead of mass discretization): the discrete simulation unfolds as a series of steady-state flows calculation updated when a state variable changes or a random event occurs. Underlying mass balancing problem can be solved with the linear programming simplex algorithm. This paper presents a novel technique based on maximizing flow through a network where nodes are black-box model units. This network-based method is less sensitive to problem size; the computation effort required to solve the mass balance is proportional to $O(m+n)$ instead of $O(mn)$ with linear programming. The approach was implemented in Flexsim™ software and used to simulate an iron ore port terminal. Processes included in the model were: mine-to-port trains handling, port terminal equipment (processing rate, capacity, operating logic, failures) and ship loading.

1 INTRODUCTION

In discrete systems, state variables change only at a countable number of points in time. These points in time are called "events" and typically represent process cycle times, items in motion, vehicles/operators arrival, travel and departure, machine breakdowns and scheduled activities. In continuous systems, state variables change in a continuous and smooth way from one state to another. These systems typically arise from chemical processes involving liquids and gases, heat and mass exchange, and chemical reactions.

In mixed discrete-continuous systems, both behaviors are present. Some industrial examples of mixed systems are: bulk material handling (ex.: powders, particles, minerals), liquids, ore pulp and oil pipelines, etc. These systems are difficult to represent in continuous simulation software because of the presence of randomness and discrete events which do not easily interact with algebraic and differential equations based solvers (Sezgi et al. 1999). On the other hand, they are difficult to represent in discrete events simulation software because of the presence of flow rates (which are continuous variables).

To model mixed discrete-continuous systems using a discrete events simulation platform, a common workaround is to replace continuous flow rates by flows of discrete masses. However, this can introduce a potentially non-negligible bias in the model unless the mass sizes tend to 0. Furthermore, including components compositions and/or particle size distribution of the streams would make the problem difficult to solve because of the large number of items to handle (Fioroni et al. 2007).

This paper presents a novel technique that can reduce significantly the complexity of simulating mixed discrete-continuous industrial systems without compromising results accuracy. This technique makes it easier for a practitioner to model plants involving a large number of units having both discrete and continuous behaviors, and to perform accurate item parts and mass balances.

This paper is organized in 4 sections. In Section 2, we briefly present the important features of Discrete Rate Simulation (DRS), a technique used to simulate mixed discrete-continuous systems. In Section 3, we expose a novel approach we developed and implemented into Flexsim, a commercial discrete events simulation software. Finally in Section 5, we present a simulation study we have conducted with the mixed discrete and continuous simulation technique presented in Section 3.

2 DISCRETE RATE SIMULATION

2.1 An Extension to Discrete Event Simulation

Simulation of mixed discrete and continuous industrial processes can be achieved using discrete rate simulation (denoted DRS), a combination of rate-based and discrete event systems (Damiron and Nastasi 2008). This concept is illustrated in Figure 1. We assume that industrial processes are hybrid systems "with significant, interacting continuous and discrete dynamics" (Nutaro et al. 2012).

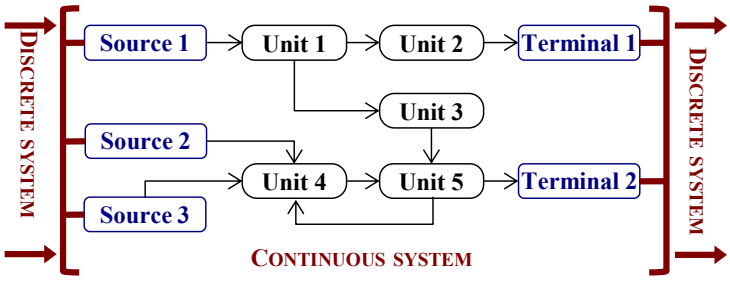

Figure 1: Schematic mixed discrete-continuous process

DRS consists basically of solving the continuous system every time an event occurs. At the time of the event, all flows between units are calculated to perform a differential mass balance. Between two "flow update" events, it is assumed that flow rates are constant. These are the fundamental principles behind simulation of hybrid systems (Kofman 2004; Krahl 2009; Nutaro et al. 2012). Examples of events that could trigger a "flow update" are:

- Events within the continuous system (see Figure 1):
  - Reservoir or stockpile reaching full or empty level marks;
  - Batch processes starting or ending cycles;
  - Valves opening or closing.
- Events at the boundary between the continuous and discrete models (see Figure 1):
  - Discrete item being "discharged" into a source or "pulled" from a terminal;
  - Arrival of a transporter (ex.: truck, train, ship) at a source or a terminal.
- Other traditional events that could result in a change in flow rates: breakdowns, repairs, operators availability, scheduled tasks, etc.

Accuracy of DRS model calculations is strongly influenced by Δt, the amount of time elapsed between two flow updates). Computation time is proportional to the number of flow updates and also depends on the number of units and streams, regardless of the amount of material flowing in the system (Ozgun and Barlas 2009).

In a pure discrete events simulation (denoted DES), the amount of material flowing in the system is represented by items carrying a fixed mass Δm. The accuracy is strongly influenced by Δm, and computation time depends of the number of item moves. Compared to DES, DRS provides a greater accuracy in mass balance calculations and a shorter computation time (Bauer and Schneider 2010).

2.2 Selection of a Simulation Platform

Simulation of hybrid discrete and continuous systems is well supported by scientific literature. In particular, there exists recognized frameworks and formalisms such as DEVS, QSS and their variants (Kofman 2004; Nutaro et al. 2012). Also, several commercial DES software offer "fluid", "continuous", or "system dynamics" rate-based features in an object oriented 3D modeling environments.

To perform DRS simulation, we chose Flexsim software (Flexsim 2013) based on a commercial decision. This software has an open architecture and offers customizing capabilities supported by C-style scripting language. The ability to create custom libraries with associated graphical user interfaces improves productivity and also promotes the development of specialized functionalities. In that regard, we already developed libraries for traffic analysis (Gipps car-following model), reliability simulation using virtual fault-trees, mobile resource fleets and overhead cranes management, and XML export functions. Some of our studies require that we combine several of our custom libraries.

Foreseeing the application of a split system approach (Nutaro et al., 2012), we evaluated that embedding DRS capability into Flexsim would represent for us a competitive advantage in addition to the custom libraries we already have. What needed to be implemented in Flexsim is a mechanism to solve continuous system mass balance.

2.3 Motivations for the Novel Approach

Some authors have suggested using linear programming (LP) to solve the mass balance (Damiron and Nastasi 2008; Fioroni et al. 2007; Krahl 2009; Ozgun and Barlas 2009; Sezgi et al. 1999). The idea behind LP-based DRS solvers is to express all mass balances as a set of linear inequalities where the unknowns are units inter-connecting flows, and constraints are imposed by units mass balances. Then, the so-called simplex algorithm can be used to solve this set of inequalities. Details on this approach can be found in Damiron and Nastasi (Damiron and Nastasi 2008).

Using the simplex algorithm or an interior points method is sensitive to problem size (Nocedal and Wright, 2006). For instance, if m is the number of continuous units and n is the number of connections between continuous units (edges, on Figure 1), then computation effort required to solve the mass balance would be of the order $O(mn)$. If k solid components with their own particle size distribution were modeled, computation effort would be of the order $O(kmn)$. With the simplex algorithm, calculation time increases exponentially with problem size.

Industrial processes we are asked to simulate can involve a hundred of continuous units. For example, the material handling process of an iron ore port terminal presented in Section 5 involves numerous belt conveyors and transfer towers, stockpiles stackers and reclaimers, railcar dumper and ship loaders. In addition to these continuous units, the system includes ore transportation by train and by ship, breakdowns and failures, operation schedules and resources allocation logics.

To simulate three years of operation with an acceptable accuracy, the model would have to evaluate all traditional discrete events plus, assuming a Δt of 1 minute, running 1,576,800 instances of the simplex algorithm. That is a huge computation effort. We developed an alternative approach to LP-based DRS which is less sensitive to problem size.

3 A NOVEL APPROACH FOR DISCRETE RATE SIMULATION

In this section, we provide details of a network approach we developed and implemented into Flexsim to perform continuous units mass balance and its underlying flows optimization problem. The novel algorithm presented below is tailored to the object-oriented approach we used to design unit operations.

3.1 The Black-Boxes Network Problem

The system illustrated on Figure 1 can be seen as a network where nodes are continuous units and edges are inter-unit flows. Although some processes have a linear configuration (a bottling line, for example),

we use the "network" terminology to denote a system made of inter-connected units with continuous behavior. In classical network optimization theory, nodes have no behavior/logic and do not dynamically influence flow calculations (Bradley et al. 1977; Elder 2011; Nocedal and Wright 2006). Nodes only indicate connections between streams and illustrate possible routings.

In order to represent a mixed discrete-continuous system, nodes must represent the dynamic nature of the equipment being simulated. Using the object-oriented design paradigm, we developed a template black-box node capable of calculating its own mass balance. The equations used to model a specific equipment mass balance are only known within the node's scope. This network node is a black box because the solving algorithm presented in the next section only exploits the following generic "visible" properties:

- *FlowError*: determine if there is a "flow excess" that prevents the unit from being well mass balanced (if *FlowError* > 0: inlet is too high; if *FlowError* < 0: outlet is too high; if *FlowError* = 0: well balanced). This node property indicates any component-wise mass balance error.
- *ReduceOutlet*: for the given inlet flow rate, adjust outlet flow rates such that *FlowError* = 0.
- *ReduceInlet*: for the given outlet flow rate, adjust inlet flow rates such that *FlowError* = 0.
- *Update*: for the given inlet and outlet flow rates, adjust states and interval variables.

Specific implementations of these properties allow units to represent real-world equipment such as: reservoirs or stockpiles, pipes, conveyors, batch processors, flow combiner/divider, components combiner and divider, process controllers, control valves. Table 1 provides selected examples that illustrates flow continuity between black-box network nodes.

Table 1: Implementation examples of the generic properties

Equipment	*FlowError*	*ReduceOutlet*	*ReduceInlet*
Reservoir	If ($L=L_{min}$) or ($L=L_{max}$), $= (s^{prev} - s^{next})$ Else, = 0	$s^{next}=s^{prev}$	$s^{prev}=s^{next}$
Vane	$s^{prev} - s^{next}$	$s^{next}=s^{prev}$	$s^{prev}=s^{next}$
Batch processor: filling: processing: purging:	$\min(0, -s^{next})$ $\max(s^{next}, s^{prev})$ $\max(0, s^{prev})$	$s^{next}=0$ $s^{next}=0$ (…)	(…) $s^{prev}=0$ $s^{prev}=0$
Flow divider	$s^{prev} - \Sigma_i s_i^{next}$	Complex procedure	$s^{prev} = \Sigma_i s_i^{next}$
Flow combiner	$\Sigma_i s_i^{prev} - s^{next}$	$s^{next} = \Sigma_i s_i^{prev}$	Complex procedure

Legend: L = level, s^{next} = downstream flow rate, s^{prev} = upstream flow rate

3.2 Solving the Mass Balance

A network made of our generic black-box nodes cannot be solved using the well-known Ford-Fulkerson algorithm. The fundamental reason is that our nodes are more than a "mathematical abstraction indicating edge incidences (or associations)" (Nocedal and Wright, 2006). Our nodes have the ability to dynamically restrict, enable or disable material routes based on their current state and candidate inlet/outlet flow values. The only assumptions that can be made about our black-box nodes is that they implement the generic properties listed in Section 3.1.

When a "flow update" is triggered during the simulation, all edges flow must be re-evaluated. Given the mass balance and state calculations performed inside the generic black-box units, finding the maximal network throughput becomes an optimization problem of refining edges flow rate guesses until all units reach component-wise mass balance (*FlowError* = 0).

We created an algorithm to calculate all edges flow such that system throughput is maximized while mass balance constraints in each node are satisfied. This algorithm, outlined in Figure 2, starts with an initial possibly infeasible optimistic solution (maximal flow rate on all edges) and progressively reduces flow rates until the sum of absolute flow error of all units is exactly 0. Therefore, the algorithm does not exploit any explicit mass balance equation. It requires that: 1) each node must return the expected quantities associated to the four generic black-box properties and 2) network nodes must be sorted in downstream direction.

1- Set all edges flow rate to maximal allowed value
2- Evaluate *FlowError* property for all units
3- Forward scan each unit in downstream direction
  - If unit *FlowError* > 0:
    - Call *ReduceOutlet* property
    - Re-evaluate its *FlowError* property
4- Backward scan each unit in upstream direction
  - If unit *FlowError* > 0
    - Call *ReduceInlet* property
5- Evaluate *FlowError* property for all units and calculate the network $\Sigma|FlowError|$
  - If $\Sigma|FlowError| > tolerance$, go back to 3
  - Else, convergence is achieved, go to 6
6- Perform all units *Update* (content and state)

Figure 2: Black-box network maxflow pseudo-algorithm

3.3 Reducing the Computation Effort

Prior to running the algorithm, the order in which units are calculated is determined. During simulation, sequential calculation becomes possible: all predecessors of a given unit are calculated first; hence, unit's outputs are calculated using best up-to-date input values. Recirculation line are problematic since the unit calculating the stream is located downstream.

A loop tearing procedure helps handling this situation (Piela and Westerberg 1994). For example, an appropriate evaluation sequence for the continuous system on Figure 1 is: Source 1 – Unit 1 – Unit 2 – Unit 3 – Source 2 – Source 3 – Unit 4 – Unit 5 – Terminal 1 – Terminal 2. The downside of recirculation lines is that iterative convergence is required.

In our implementation, determining the units order for network-based sequential evaluations is performed only once before simulation begins. This pre-processing phase reduces the complexity of the calculation because network topology is analyzed only once. It is similar to labeling algorithms used to solve classical network maxflow problems (Bradley et al. 1977, Elder 2011).

With LP-based DRS, a simultaneous calculation scheme is used: solving the set of inequalities implies that network topology is implicitly analyzed by the simplex algorithm (Nocedal and Wright 2006) at each flow update.

3.4 Increasing Calculations Accuracy

Calculations accuracy is mostly governed by three key elements: 1) internal units model for mass balance, 2) the *tolerance* parameter at step 5 in Figure 2, and 3) the capability to estimate the exact time at which continuous system needs to be updated.

Internal units mass balance model are usually errorless. Some units such as vane, flow divider and combiners do not accumulate mass. For units that accumulate mass, such as reservoirs and filling or purging batch processors, internal content is updated with the relationship $\Delta L = \Delta t \cdot (s^{prev} - s^{next})$.

The *tolerance* parameter is the numerical precision of the convergence loop. This is a user-defined value of, for example, 0.01%. Reducing *tolerance* value will demand more iterations to achieve convergence of the algorithm in Figure 2 but will yield more accurate flow values.

Estimating the exact time at which continuous system needs to be updated can be approximated using a fixed time step. This approach is easy to manage during simulation and provides an upper error bound on single event time estimate. However, some events could occur in between two updates and errors can be cumulative, resulting in loss of accuracy. A common workaround is to use a finer time step. The accuracy is improved, but simulation time is longer and several consecutive edges flow re-evaluations can be performed without being needed (no flow rate change, no random event, etc.).

In order to increase the probability of estimating exactly the time of process changes, we developed a predictive time step approach. The concept of event time prediction in continuous system has been extensively explained in Bauer and Schneider (Bauer and Schneider 2010). The basic idea is that continuous units can compute in how much time a change will occur. Table 2 shows examples of event time prediction functions we implemented. To remain conservative, all units are scanned and next flow update is triggered in the smallest delay before a change. Events such as failures, breakdowns and repairs, and interactions at the boundaries of the continuous system can also force a flow update.

Table 2: Examples of event time predictions

Equipment	Event time prediction
Reservoir	$\min((L-L_{min})/s^{next}, (L_{max}-L)/s^{prev})$
Vane	∞
Batch processor: filling:	$(L_{max}-L)/s^{prev}$
processing:	Time remaining before end of cycle
purging:	$(L_{max}-L)/s^{prev}$
Flow divider	∞
Flow combiner	∞

Legend: L = level, s^{next} = downstream flow rate, s^{prev} = upstream flow rate

3.5 Properties of the Algorithm

One property of interest is that if the continuous system has no recirculation line, this algorithm converges in only one iteration with $\Sigma|FlowError|=0$. The presence of recirculation lines requires iterations, because at least one predecessor of one unit is not evaluated yet during first iteration (see Section 3.3). In practice, we have observed that less than 10 iterations lead to a mass balance accuracy $\Sigma|FlowError| \leq 0.01\%$.

Using k, m and n as defined in Section 2.3, another property of interest is that running this algorithm requires a computation effort of the order $O(km+n)$. This order of complexity is driven by the algorithm's six steps: one step on all n connections (Step 1) and five steps on all m units (Steps 2 to 6). If there are k components, then each unit performs a mass balance for each component. The network-based DRS algorithm $O(km+n)$ complexity means that computation time will increase slower with problem size than for LP-based DRS, which is $O(kmn)$.

4 THEORETICAL EXAMPLE: TANK LEVEL CONTROL

The following storage tank level problem from Damiron and Nastasi (2008) is useful to illustrate and compare DRS and DES performances on a purely continuous system. The problem illustrated in Figure 3 is designed to test the capability to determine event times accurately. Storage tank has a capacity 10 tons,

initial is content is 5 tons. Source flow rate is 1 ton/minute. Outlet flow rate is initially 0.3 ton/minute. When tank is full, outlet flow rate is set to 2.1 ton/minute. When tank is empty, outlet flow rate is set to 0.3 ton/minute. The objective is to determine the tank content after 100 minutes. This problem can be solved exactly by hand as in Figure 4; solution is 9.545454…

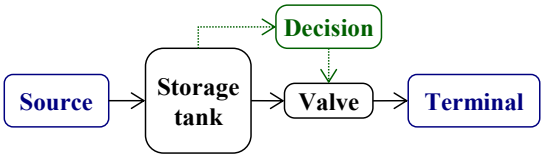

Figure 3: Storage tank flow chart

The difficulty in solving this problem using simulation resides in the fact that exact events (tank full or empty) occur at times that are not multiples of common discretization values such as 1/5, 1/100, etc. First event is "tank full" at time 5/0.7 = 7.142857… minutes. Fixed mass values Δm (DES with finite masses) and fixed time steps Δt (DRS non-predictive time step) approaches might fail at estimating exactly event times.

We performed several experiments using both discrete events and discrete rate approaches. Results presented in Table 3 include: model key settings, tank content after 100 minutes, absolute error and required number of events in simulation (from Flexsim's events counter).

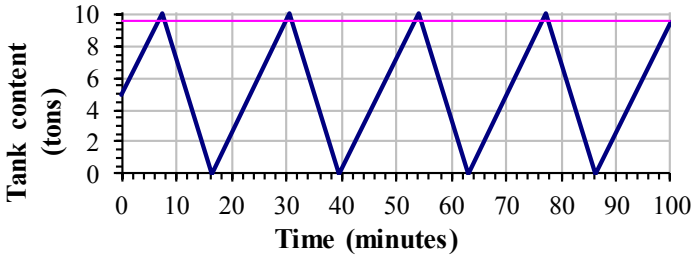

Figure 4: Exact tank content trend curve

Table 3: Tank content after 100 minutes

Modeling approach	No. events	Content (t)	Error (%)
Discrete events (using items as finite masses)			
Blocks: Δm = 0.1 t	3,243	8.9	6.76
Blocks: Δm = 0.05 t	9,644	9.85	3.19
Blocks: Δm = 0.001 t	483,785	9.552	0.07
Discrete rate (as described in Section 3)			
Fixed time step: 0.1 min	1,001	9.449	1.01
Fixed time step: 0.01 min	10,001	9.501	0.46
Fixed time step: 0.001 min	100,001	9.544	0.01
Predictive time step	18	9.545454	0
Exact analytical solution	**9**	**9.545454**	

Results for discrete events illustrate the lack of precision associated to using finite masses. A very small block size was required to obtain a satisfying precision, but the number of events is close to 500,000. And this is a very simple example, not a large scale system. Results for DRS with fixed time

step indicate that results precision is better with a smaller step size. Benefits of the DRS approach with predictive time step are well illustrated in Table 3: exact results achieved with very few number of events.

5 CASE STUDY: IRON ORE TERMINAL DE-BOTTLENECKING

In this section, we use our DRS library to perform a debottlenecking study of an iron ore port terminal. The study also evaluated options to increase production capacity beyond its original design. Abbreviations used in this section are listed in Table 4.

The scope of simulated processes is illustrated in Figure 5. Processes included in the model were: trains handling (loading at mine, mine-port-mine travelling and unloading at port), port terminal equipment (processing rate, capacity and operating logic) and vessels handling (TSV vessels loading, travelling and unloading into OGV vessels).

Table 4: List of abbreviations

Abbreviation	Description
Avrg	Average
C.I.	Confidence interval
Mtpa	Million metric tonnes per annum
OGV	Ocean Going Vessel
t/h	Metric tonnes per hour
TSV	Transshipment Vessel

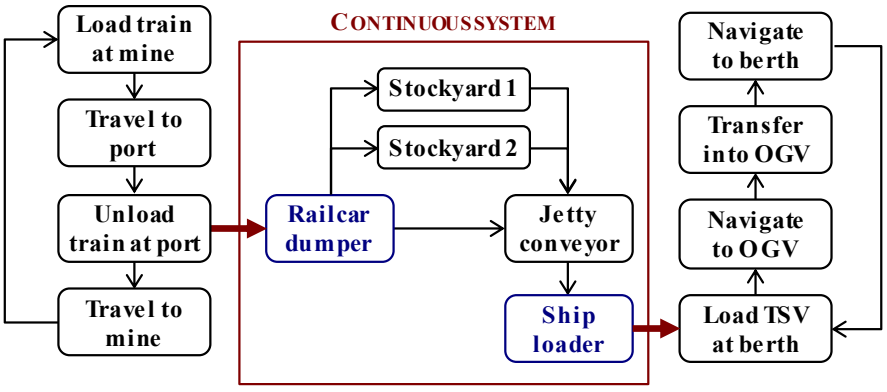

Figure 5: Iron ore mine-to-port process map

Processes related to trains and vessels handling can be modeled using "classical" discrete events simulation, but material handling at port involves conveying, stacking and reclaiming, and handling of empty/full stockpile constraints. These processes use quantities expressed in tons per hour and imply performing mass balances.

5.1 Model Description

Material flow and main processes included in the model are illustrated in Figure 5. In addition to this conceptual map, important data and constraints related to existing equipment and operations are:

- Trains handling: train fleet consists of 6 trains made of 85 cars with an average content of 65 t. Due to port terminal space constraints, trains are divided in rakes, rakes are unloaded, then rakes are shunted and all cars are reconnected. Statistical distributions have been built using historical

data on travel times, loading and unloading durations, shunting reconnection times, and railcar content. Average arrival rate at port is 4.5 trains per day.

- Material handling: belt conveyors and transfer towers capacity is 3,500 t/h, and jetty conveyor capacity is 7,000 t/h. Stockyard stackers and reclaimers capacities are respectively 3,500 t/h and 4,800 t/h. Ore is routed directly from railcar dumper to ship loader whenever possible. If no TSV is being loaded, ore is stacked in non-full stockyards. If no train is being unloaded, TSV are loaded from non-empty stockyards. Statistical distributions have been built using historical data on railcar dumper throughput and belt conveyors and transfer towers stops and breakdowns.
- Vessels handling: iron ore is transported from berth to an OGV standing at an offshore transfer location using TSV vessels. TSV fleet consists of 2 ships with a maximum payload of 50,000 t. The transshipment operation arises from limited water depth nearby port terminal, a narrow navigation channel connects with deep ocean water. Effective TSV load is limited to 41,000 t due to channel depth restriction. Moreover for safety purposes, only one TSV can navigate in channel during a tide low/high window. There is room for only 1 TSV at berth, other TSV have to wait at navigation channel entrance. As a consequence, ship loader can operate only 12 hours per day. Tide tables have been included in the model using statistical forecasting equations.

Some additional features were included in the model in order to account for maintenance periods, unplanned shutdown periods and wet season days when production efficiency is reduced

5.2 Model Validation

Model has been calibrated using existing process data and real operation logics. Comparison between observed daily exports and simulated daily throughput is presented on Figure 6. Simulated values (red diamonds) are close to data central values. In Table 5, several important variables for which historical data were provided are compared to simulation results. Simulation results are close to observed values. Model is able to represent and predict key characteristics of the terminal, including performing an accurate mass balance.

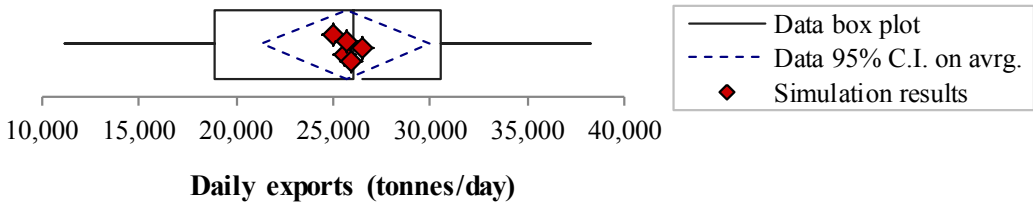

Figure 6: Terminal throughput comparison

Table 5: Simulated vs observed values

Metric	Observed values Avrg. ± 95% C.I.	Simulation results Avrg.
Unloaded trains (trains/day)	4.51 ± 0.17	4.58
Train unloading duration (h/train)	4.43 ± 0.31	4.39
Direct train to TSV loading (%)	52.5 ± 6.6	47.8
Loaded TSV (TSV/day)	0.62 ± 0.11	0.57

5.3 Terminal Performance Assessment

With the help of material handling specialists and port terminal design experts, it has been confirmed that bottleneck was the railcar dumper design and location. Several significant equipment changes have been proposed to ensure terminal could handle projected throughput volumes:

1. Relocate (no train splitting) and increase capacity of railcar dumper
2. Increase number of cars in trains and use cars with a greater capacity
3. Build a lay-by berth for an additional TSV (ship loader can be operated 24 hours per day)
4. Add a 3rd TSV
5. Expand train fleet
6. Dredge navigation channel (increase TSV payload)
7. Add a 4th TSV

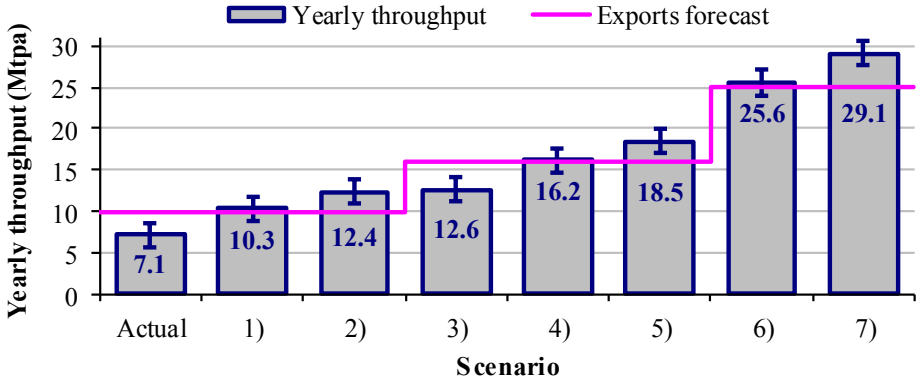

Figure 7: Comparative simulation results

All these changes have been evaluated using the simulation model; results are presented on Figure 7. Comparison to planned production targets (purple line on Figure 7) indicates that without any change, terminal throughput would never have reached expectations. Simulation results of proposed changes indicate that terminal will be capable of meeting projected volumes at the end of the construction phases.

5.4 Benefits of Using DRS

Our DRS library used to simulate the iron ore port terminal had the following advantages:

- Iron ore handling system was "easy" to model since continuous objects already implement the logic needed to represent bulk material handling equipment like conveyors, stockpiles and transfer towers.
- Evaluating the impact of empty or full stockpiles on train unloading and ship loading was critical. No special modeling effort was required to simulate stockyards since these constraints are "built-in" features of the DRS framework.
- Ship loader had a specific operation logic. Modeling this equipment was simple, it only required adequate formulation of the generic node's properties listed in Section 3.1. The possibility to easily build custom unit models was a great benefit in this project.
- Model behavior and results were accurate at each "flow update". Variability observed in simulation results is due to the system itself (modeled randomness is detailed in Section 5.2), and not to modeling approximations by finite masses.

- Interactions between continuous and "classical" discrete event system (failures and breakdowns, trains and ships handling, schedules) were accurately evaluated within a single model.

A final remark concerning the benefits of our network-based DRS approach, continuous system was made of 63 units linked by 75 connections. The smallest Δt used was 0.5 minutes. Simulating a period of 3 years required approximately 3 minutes using a computer equipped with a Intel Core i7 processor. During these 3 minutes, over than 10,500,000 instances of our black-box network algorithm have been used. This aspect is important in the context of a fast pace project.

6 CONCLUSIONS

The DRS technique is a modeling tool to be used to simulate industrial processes involving discrete events and commodities expressed as mass or volume per time unit. It ensures more accurate calculations and facilitates the modeling of flow routing, composition and capacity constraints. We developed and implemented in Flexsim a new black-boxes network-based DRS calculation technique. This novel approach facilitates the development of custom model units and makes calculation time less sensitive to problem size. We successfully used our DRS implementation in the context of an iron ore port terminal de-bottlenecking study. The robust and accurate model we obtained was helpful in identifying bottlenecks and evaluating proposed equipment modifications. Future works should focus on improving the black-box network solving algorithm speed by removing unnecessary calculations and making it converging faster.

REFERENCES

Bauer, K., and Schneider, K., 2010. "Predicting Events for the Simulation of Hybrid Systems". In *2010 10th IEEE International Conference on Computer and Information (CIT 2010)*, 1833-1840.

Bradley, S.P., Hax, A.C., and Magnanti, T.L., 1977. "A Labeling Algorithm for the Maximum-Flow Network Problem". In *Applied Mathematical Programming*, Addison-Wesley.

Damiron, C., and Nastasi, A., 2008. "Discrete rate simulation using linear programming". In *Proceedings of the 2008 Winter Simulation Conference,* Edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 740-749. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Elder, J., 2011. "07 Network flow algorithms." In CSE 3101Z Design and analysis of algorithms (lecture notes), York University. Accessed January 19, 2012. http://elderlab.yorku.ca/~elder/teaching/cse3101/index.php

Fioroni, M. M., Franzese, L.A.G., Zanin, C.E., Perfetti, L.T., Leonoardo, D, da Silva, N.L., and Furia, J., 2007. "Simulation of continuous behavior using discrete tools: ore conveyor transport". In *Proceedings of the 2007 Winter Simulation Conference*, Edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1655-1662. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

FlexSim Software Products, Inc., 2013. Accessed March 15, 2013. http://www.flexsim.com/

Kofman, E., 2004. "Discrete Event Simulation of Hybrid Systems". In *SIAM Journal on Scientific Computing*, 2004, vol. 25, 1771-1795.

Krahl, D., 2009. "Extendsim advanced techology: discrete rate simulation". In *Proceedings of the 2009 Winter Simulation Conference*, Edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 333-338. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nutaro, J., Kuruganti, P.T., Protopopescu, V., and Shankar, M., 2012. "The Split System Approach to Managing Time in Simulations of Hybrid Systems having Continuous and Discrete Event Components". In *SIMULATION*, March 2012, vol. 88 no. 3, 281-298.

Nocedal, J, and Wright, S., 2006. "Numerical Optimization, 2nd edition". Springer, 664 p., ISBN 978-0-387-40065-5.

Ozgun, O., and Barlas, Y., 2009. "Discrete vs. Continuous Simulation: When Does It Matter?" In *Proceedings of the 27th International Conference of The System Dynamics Society*, 1199-1221.

Piela, P. and Westerberg, A.W., 1994. "Equation-Based Process Modeling". Carnegie Mellon University, EDRC Technical Report, Pittsburgh, PA 15213.

Sezgi, U.S., Jameel, H., Wheles, J., and Kirkman, A.G., 1999. "A combined discrete-continuous simulation". In *1999 Engineering Conference Proceedings*, EN99179.

SNC-Lavalin Inc., Engineering and construction firm.. http://www.snclavalin.com/

Venkateswaran, J., and Son, Y.-J., 2005. "Hybrid system dynamic—discrete event simulation-based architecture for hierarchical production planning". In *International Journal of Production Research*, vol. 43, no. 20, 4397-4429

AUTHOR BIOGRAPHIES

VINCENT BÉCHARD, Eng. MASc., has a Bachelor degree in chemical engineering and a Master degree in applied mathematics from École Polytechnique de Montréal, in Canada. He has several years of experience in the field of applied mathematics: simulation and optimization of industrial processes, consulting and teaching in statistics. His diversified working experiences made him know better pulp and paper, mine-to-port ore handling, traffic simulation, call centers and aluminum production. His current position is Discrete Event Simulation Designer at SNC-Lavalin Inc., div GTS. His LinkedIn profile is http://ca.linkedin.com/ in/vincentbechard and his email address is vincent.bechard@snclavalin.com.

NORMAND CÔTÉ, Eng., has a Bachelor degree in mechanical engineering from École Polytechnique de Montréal, in Canada. He is director of the discrete event simulation (DES) group at SNC-Lavalin Inc., div. GTS, a firm specialized in training and simulation solutions. He has 20 years of experience, of which 12 were spent in simulation and computer-based training software development. He has mainly worked in the metallurgy and pulp and paper industries. His email address is normand.cote@snclavalin.com.