MININGIQ TECHNICAL SERIES

The Haulage Module

Closing the Loop Between Truck Simulation, Pit Optimisation, and Mine Costing

Author	IMC Mining
Platform	MiningIQ
Date	March 2026

Introduction

Truck cycle time simulation is not new. Dedicated haulage simulation systems like Talpac and SPRY have been used for decades to estimate haul times from rimpull curves, grade resistance, and rolling resistance. These are mature, well-validated tools, and MiningIQ's physics engine draws on the same first-principles foundations. If all you need is a cycle time for a single route, the established systems do the job.

The problem has never been the physics. The problem is what happens *after* the cycle time is calculated. In the conventional workflow, an engineer runs Talpac or SPRY as a standalone tool, exports cycle times, manually maps them to blocks in the mine plan, converts them to costs using a separate fleet model, and then feeds those costs into pit optimisation and scheduling — often in yet another tool. Each handoff introduces manual transcription, version drift, and the temptation to simplify. By the time a haulage cost reaches the pit optimiser, it is frequently a flat $/t assumption that bears little resemblance to the segment-by-segment simulation that produced the original cycle time.

MiningIQ's haulage module eliminates these handoffs. The physics engine, the discrete event simulation, the road network, the block model, the pit optimiser, and the cost model all live in a single integrated platform. A cycle time computed from first-principles kinematics flows directly into a per-block haulage cost, which feeds into the Mining Cost Adjustment Factor (MCAF) used by the pit optimiser, which produces the shell sequence consumed by the scheduler, which defines the production targets sized by the fleet optimiser. There is no export, no re-import, no manual mapping. When a road design changes, the entire chain — from physics to financial model — updates in a single web session.

This article describes each layer of the system: the road network model, the physics engine, the DES, and critically, how these layers connect to pit optimisation, scheduling, and fleet cost modelling to close the loop that conventional workflows leave open.

Conventional Workflow vs MiningIQ

Step	Conventional Approach	MiningIQ
Haul profile design	Vulcan / Deswik → export profiles	Import 3D polylines (CSV/DXF) from any design package
Route assembly	Manual sequencing of segments per route	Automatic graph construction with K-shortest route discovery
Cycle time simulation	Talpac or SPRY (standalone tool)	Embedded physics engine, same platform as block model
Block haulage costing	Export cycle times → manual mapping to blocks	Direct lookup from pre-computed route cycle times per block
Pit optimisation input	Flat $/t or averaged haulage cost	Per-block MCAF incorporating actual haul distance and physics
Fleet sizing	Separate DES tool or spreadsheet estimate	Integrated DES with iterative fleet optimiser
Cost model linkage	Manual transfer of fleet counts to cost model	Fleet results write directly to capital and operating cost schedule
Road design change	Re-run each tool, re-export, re-import	Single update propagates through entire chain

1. Road Network Modelling

Building haul profiles in mine design packages like Vulcan or Deswik is standard practice. The design engineer creates 3D polyline profiles along haul roads, exports them, and feeds them into a haulage simulation tool. That part of the workflow is well established. Where the conventional approach becomes cumbersome is in assembling those profiles into complete source-to-destination routes, managing alternative paths, handling new benches that don't yet have road data, and keeping the route network in sync as the pit design evolves through multiple iterations.

MiningIQ addresses this by building an automated graph-based road network from the imported profiles. Rather than requiring the engineer to manually sequence profile segments into routes, the system constructs a connected graph and discovers routes automatically. It also provides methods to interpolate and extrapolate haul profiles for benches that have not yet been designed — avoiding the need to go back to the design package every time a new bench is added to the study.

Graph Construction

Haul road profiles are imported as 3D polylines in CSV or DXF format from any mine design package. Each profile is a named polyline representing a physical road segment — for example, a ramp from a pit bench to a saddle junction, or a surface haul road from a junction to the crusher. The system processes these profiles through four steps:

1. Parses XYZ point arrays from each profile to capture full 3D geometry

2. Clusters profile endpoints into junction nodes using a KDTree spatial index with a configurable snap tolerance (default 5 m)
3. Builds a NetworkX MultiDiGraph — a directed graph with parallel edges for alternative routes between the same nodes
4. Auto-detects node roles (crusher, stockpile, waste dump, pit source) from profile naming conventions, with user override through the web interface

The result is a connected graph where nodes represent junctions and destinations, and edges are road segments with full 3D geometry attached.

Route Discovery

For any source-to-destination query, the system finds K-shortest simple paths (default K = 5) through the graph. Each candidate route is assembled by concatenating edge polylines into continuous arrays of segment lengths, gradients, bearings, rolling resistances, and road classes. This means the physics engine always evaluates multiple route options and selects the fastest feasible candidate.

Bench Entry — The Three-Tier Logic

Mining does not happen at fixed graph nodes. It happens at specific bench elevations that change as the pit deepens. The system must find where each bench RL intersects the ramp profiles. MiningIQ uses a three-tier approach to handle this robustly:

1. **Tolerance match** — the first point within ±0.75 m of the target bench RL. This deterministic method avoids skipping switchbacks.
2. **Zero-crossing** — the first point where the elevation difference changes sign. This handles cases where no surveyed point falls exactly at the bench RL.
3. **Closest Z fallback** — the global closest point by elevation. This never fails, and always generates a warning for the engineer to review.

The route is truncated at the bench entry point and a flat on-bench segment is prepended. Critically, this happens without mutating the graph — the network stays clean, and per-bench route arrays are ephemeral.

Stage-Aware Subgraphs

Each road profile is linked to the pit stages or pit shells during which it is active. As the mine progresses through its pushback sequence, roads are constructed and decommissioned in step with the evolving pit geometry. The network produces stage-filtered subgraphs, ensuring that route cycle times reflect the roads that will actually exist at each stage of mining.

Interactive 3D Viewer

The road network is rendered in a 3D viewer within the MiningIQ web interface, featuring fat-line polyline rendering, raycaster picking for hover and click interaction, labels on key nodes, and OrbitControls for navigation. Segment codes and road parameters can be edited directly from the 3D view.

2. The Physics Engine

The physics underpinning the haulage module will be familiar to any engineer who has used Talpac or SPRY. The engine simulates a truck traversing a route one segment at a time, computing tractive effort from rimpull curves, resistive forces from grade and rolling resistance, and speed profiles from standard kinematics. This is the same foundation the industry has relied on for decades, and MiningIQ does not attempt to reinvent it.

What is different is how the engine is deployed. Rather than running as a standalone desktop tool whose outputs must be manually transferred into a mine plan, the physics engine operates as an embedded service within MiningIQ's web platform. It reads road geometry directly from the network graph, truck specifications from the project database, and writes cycle times directly to the block model — with no file exports, no copy-paste, and no version mismatches.

Simulation Configuration

All physics constants are encapsulated in a SimConfig dataclass, loaded from database settings. This means multiple projects with different site configurations coexist without interference. Key configurable parameters include:

Parameter	Description	Default
Gravity	Gravitational acceleration (configurable for altitude)	9.81 m/s²
Rolling resistance	Per road surface class (A/B/C), stored as percentage	Site-specific
Speed caps	Per road class (loaded/empty), sitewide, ramp, corner bands	Site-specific
Lookahead distance	Anticipatory braking distance	40 m
Comfort deceleration	Minimum braking rate for operator comfort	0.4 m/s²
Calibration factor	Informed adjustment multiplier applied to travel times	Site-specific
Coefficient of traction	Tyre-road friction limit	0.45

Truck Performance Model

Each truck type is defined by OEM performance data. The system currently supports many truck models, with the architecture accommodating any truck for which rimpull and retarder data is available. Key truck parameters include:

- **Rimpull curve:** force (N) versus speed (m/s) — this defines the engine's available tractive effort at any speed
- **Retarder curve:** braking force (N) versus speed (m/s) — the dynamic braking capacity on downhill segments
- **Physical limits:** empty mass, payload capacity, maximum loaded and empty speeds, maximum acceleration and deceleration rates

- **Tray volume:** for volume-constrained payload calculation using loose density derived from in-situ density and swell factor

Route Geometry Pre-Processing

Before simulation, the route geometry is prepared through several steps. Segment geometry is computed from XYZ point arrays to produce per-segment lengths, gradients, bearings, and cumulative distances. Segments longer than 10 m are split into equal sub-segments to maintain simulation resolution. Flat 50 m buffer segments are prepended and appended to represent acceleration and deceleration zones at loading and dumping areas. For the empty return trip, segments are reversed and gradients negated.

Regulatory Speed Caps

Before the kinematic simulation runs, per-segment speed caps are computed from multiple sources, with the most restrictive value applying:

1. **Road class cap:** each road surface type has separate loaded and empty speed limits
2. **Ramp cap:** downhill segments receive a lower speed cap; flat and uphill segments use the sitewide cap
3. **Corner cap:** bearing changes between consecutive segments trigger speed restrictions. The system uses a multi-band angle system (e.g., 30–60° = 25 km/h, 60–90° = 15 km/h, >90° = 10 km/h) with an influence zone extending the reduction to segments either side of the apex
4. **Truck maximum speed:** the truck's own loaded or empty maximum velocity
5. **Terminal cap:** a low-speed cap on the final segment to model deceleration into loading or dumping position

The Forward Pass — Segment-by-Segment Kinematics

The simulation executes a single forward pass through all segments. For each segment, the engine computes the resistive force:

$$F\_res = mass \times g \times (grade/100 + rolling\_resistance)$$

It then determines whether the truck is accelerating or braking by comparing the current speed to the look-ahead anticipation target. When accelerating, the net engine force is the rimpull at the current speed minus the resistive force. Acceleration is clamped to the truck's maximum, and the exit speed is computed using standard kinematics:

$$v\_end = \min\left(\sqrt{v_0^2 + 2 \cdot a \cdot L},\ \ v\_cap \right)$$

Segment travel time is computed by averaging entry and exit speeds for short segments, with an exact acceleration-plus-cruise split for longer segments where the truck reaches the speed cap.

When braking, achievable deceleration is derived from the retarder curve and clamped to both the comfort deceleration limit and the truck's maximum deceleration rate.

Look-Ahead Anticipation

The look-ahead mechanism scans ahead up to the configured distance (default 40 m) and computes the maximum safe entry speed that allows comfortable braking to any lower speed cap ahead. This prevents the truck from overshooting into a low-speed zone — a critical refinement that flat-haul calculators miss entirely.

Cycle Simulation

A full cycle simulation runs the forward pass twice: once loaded (pit to destination, using loaded mass and loaded speed caps) and once empty (destination to pit, on the reversed route, using empty mass and empty speed caps). The result is a CycleResult containing loaded travel time, empty travel time, per-segment velocity profiles, per-segment times, and stall detection.

Flat Haul Simulation

For blocks that require additional flat-haul distance from a pit floor to the ramp entry, the engine creates a synthetic flat route (0% grade, default rolling resistance, no corners) and simulates it through the same physics engine. This is not a simple speed-times-distance lookup — it accounts for acceleration from rest and deceleration at the end, maintaining physical consistency with the main simulation.

3. Batch Simulation

The physics engine simulates individual routes. The batch simulator orchestrates mass computation across all required combinations.

The Combinatorial Grid

For each combination of pit, destination, bench RL, and truck type, the system:

1. Finds the bench entry point on the pit ramp using the three-tier logic
2. Discovers K candidate routes from the bench entry to the destination
3. Builds route arrays for each candidate — concatenating edges, truncating at the bench, adding on-bench and buffer segments
4. Simulates each candidate through the physics engine
5. Selects the fastest non-stalled candidate
6. Computes KPIs: loaded and empty speeds, total cycle time, calibration-adjusted cycle time

Extrapolation for Non Route Defined Benches

For benches above the last design level, the system extrapolates from a base result by computing the vertical delta and horizontal distance from the configured ramp gradient, simulating this extra segment through the physics engine, and adding it to the base result's travel times. This ensures that early-stage studies can still produce meaningful haulage estimates for future pit stages.

Fixed Times and Calibration Adjustment

The total cycle time combines travel times with fixed operational times:

```
Total CT = loaded_travel + empty_travel + spot + load + dump + queue
Adjusted CT = Total CT / calibration_factor
```

Fixed times can be derived from shovel dig-rate calculations (bucket capacity, fill factor, swing time, propel time, number of passes), user-defined defaults per fleet assignment, or fallback constants. The calibration factor is an informed adjustment derived from site-specific operational data — GPS fleet records, dispatch system logs, or time studies — that accounts for real-world inefficiencies not captured in the deterministic physics model, such as operator variability, minor delays, and road condition variance. This is not a generic correction; it is calibrated to the operation.

4. Block Model Integration

The haulage module connects to the geological block model through a three-layer architecture designed to separate expensive computation from fast per-block lookups.

Layer 1: Pre-Compute Route Cycle Times

For every fleet assignment and bench RL combination, the physics engine runs over the road network and stores the result as a RouteCycleTime record. This is the computationally expensive step — it involves graph traversal, route assembly, and segment-by-segment simulation. It runs once when the user triggers the precompute action.

Layer 2: Import Block Model

Block model data is imported with centroid coordinates, tonnages, material types, and destination assignments. Each block carries its bench RL, its pit identifier, and its destination (ore to crusher, waste to dump).

Layer 3: Compute Block Haulage

For each block, the system looks up the pre-computed RouteCycleTime for that block's fleet, bench RL, pit, and destination combination. It adds flat-haul time if the block's centroid is offset from the ramp entry, then computes per-block haulage cost from cycle time multiplied by the truck operating cost rate. This step is fast — a lookup plus arithmetic rather than a simulation. The expensive physics was already completed in Layer 1.

5. Discrete Event Simulation

Discrete event simulation of truck-shovel fleets is itself well-established — commercial DES tools have been available for years. The value MiningIQ adds is not the simulation technique, but the integration. The DES engine reads its travel times directly from the physics engine (which reads road geometry from the network graph), reads its operational delays from the same MineCost roster that drives the cost model, and writes its fleet sizing results directly into the capital and operating cost schedule. There is no intermediate file, no manual re-entry, and no risk of the DES assumptions diverging from the cost model assumptions.

The deterministic physics engine computes what a single truck *can* do on a given route. The DES engine models what a fleet of trucks *actually achieves* when they share roads, queue for shovels, experience breakdowns, and work through shift changes.

SimPy Process Model

Built on the SimPy discrete-event framework, each truck operates as an independent process cycling continuously through dispatch, empty travel, queuing and loading at a shovel, loaded travel, queuing and dumping at a destination, and return. Dispatch strategies include least-queue, minimum-wait, and round-robin assignment. Travel times come directly from the physics engine with configurable stochastic variation, while loading, spotting, and dumping times are drawn from triangular distributions. Trucks compete for shovels and dump points as shared resources — if a shovel is occupied or broken down, arriving trucks queue until it becomes available.

Stochastic Distributions

Every activity time in the DES is drawn from configurable probability distributions, allowing the simulation to capture the inherent variability of mining operations:

Activity	Distribution Options	Default
Cycle times (load, dump, spot)	Triangular, normal, uniform	Triangular
Travel times	Deterministic base ± 10% variation	Normal
Equipment breakdowns	Exponential MTBF, Weibull, lognormal	Exponential
Payload variation	Fixed, normal (±5% CV), lognormal	Normal

Operational Delays

The DES reads operational delays directly from MineCost roster records — the same roster data that drives the cost model. These are not abstract efficiency factors; they are specific, time-classified delay categories:

- **Blast exclusion:** scheduled blast events per shift, with total duration from the roster. All trucks are held during blasting.
- **Shift change:** fires at each shift boundary. All trucks held for the configured duration.
- **Crib and smoko breaks:** crib at mid-shift, smoko at quarter-shift.
- **Wet weather:** converts annual wet weather days to a daily probability. During rain events, travel times increase by 25%.
- **Loader relocation:** seizes all shovel capacity for the roster's configured relocation time per shift.
- **Service, lube, and fuel:** per-shift downtime from the roster.

Each truck accumulates delay time separately, broken down by category. The simulation distinguishes productive time from queue time, operational delay time, and breakdown time.

Road Congestion — BPR Model

When multiple trucks share a route, the DES applies a Bureau of Public Roads (BPR) style congestion model:

```
delay_fraction = α × max(0, trucks_on_route − free_flow_capacity)
congestion_delay = base_travel_time × delay_fraction × direction_factor
```

The default alpha is 0.10 (10% delay per excess truck), with a free-flow capacity of 4 trucks per route. Loaded trucks are affected more than empty trucks (direction factor 1.0 versus 0.5), reflecting the reality that heavier, slower trucks create more congestion on uphill hauls. Trucks register entry and exit on each route segment, and the congestion tracker maintains a real-time count.

Operational Realism

Beyond queuing and travel, the DES captures several additional sources of variability. Shovels experience random breakdowns with time-to-failure sampled from configurable MTBF distributions (exponential, Weibull, or lognormal) and repair times drawn from triangular distributions. Each truck is assigned a random operator efficiency factor that scales all its travel times, introducing realistic inter-truck variability without requiring individual operator profiles. A configurable warmup period excludes the initial transient from reported statistics, ensuring that KPIs reflect steady-state fleet performance.

DES Outputs

The simulation produces comprehensive statistics at multiple levels:

- **Per truck:** cycles completed, total payload, travel/loading/dumping/queue/spot/breakdown/delay minutes, productivity (tph), utilisation, queue fraction, congestion fraction
- **Per shovel:** loads completed, total payload, utilisation, maximum queue length, breakdown time
- **Fleet-level:** total tonnes, total cycles, fleet tph, effective efficiency (productive time divided by total fleet time), average queue fraction, average delay fraction

6. Fleet Optimiser

The DES engine is wrapped in an iterative optimiser that determines the required fleet size to meet an annual production target. This closes the loop between the mine schedule (which defines how much material must move in each period) and the fleet cost model (which needs truck and shovel counts).

Optimisation Algorithm

The fleet optimiser follows an iterative convergence approach:

4. **Estimate initial shovel count** from the target tonnes per annum divided by shovel theoretical capacity at 80% utilisation

5. **Binary search for truck count:** for the current shovel count, find the truck count where achieved production meets the target (8 binary search steps)
6. **Analyse the DES result** to classify the fleet state: shovel-limited (add a shovel), over-shovelled (remove a shovel), truck-limited (add a shovel to unlock capacity), or balanced (fine-tune trucks)
7. **Repeat** until the target is met within tolerance (default 2%) or maximum iterations (default 20) are reached

Annualisation

The DES simulates a single shift or multiple shifts. Annualisation uses the bottleneck equipment's effective roster hours, derived from the MineCost roster — the same roster that calculates scheduled hours, availability, utilisation, and operating efficiency. This ensures consistency between the simulation's production estimate and the cost model's operating hours.

Per-Period Fleet Scheduling

The fleet optimiser integrates with the mine schedule. For each scheduling period, it reads production targets, annualises them based on the period's calendar months, runs the fleet optimisation, and records shovel counts, truck counts, production rates, and utilisation. This produces life-of-mine fleet requirements that feed directly into capital and operating cost models.

7. The Simple Sim — Pre-Computed Cycle Times

Not every analysis requires a full discrete event simulation. The Simple Sim path provides a faster alternative for studies where fleet dynamics are not the primary focus.

The Simple Sim operates in three steps:

6. **Precompute:** run the physics engine over all fleet and bench combinations, storing results as RouteCycleTime records
7. **Block haulage:** for each block, look up the matching cycle time, add flat-haul if needed, compute cost
8. **Result:** per-block haulage costs ready for pit optimisation and scheduling — no DES, no queuing, no stochastic variation

The physics is still first-principles — every segment of every route was simulated with full kinematic detail. The simplification is that fixed times (load, dump, queue, spot) are deterministic inputs rather than DES outputs. This makes the Simple Sim appropriate for early-stage scoping studies, sensitivity analyses on road design alternatives, and rapid iteration during pit optimisation where the haulage cost per block needs to be recomputed frequently.

8. Integration with the Mine Optimisation Pipeline

This is the section that matters most, because it is where MiningIQ diverges from the conventional workflow. In a typical study, the haulage engineer produces cycle times in Talpac or SPRY, the pit optimisation engineer imports a flat $/t haulage cost into Whittle or similar, the scheduling engineer works in yet another tool, and the cost engineer builds a fleet model in Excel. Each handoff is a manual process. Each handoff is an opportunity for assumptions to diverge.

In MiningIQ, there are no handoffs. The haulage module's outputs feed directly into pit optimisation, scheduling, and fleet costing within the same platform, using the same database, updated in the same session.

Into Pit Optimisation

Block-level haulage costs from the pre-computed route cycle times are incorporated into the Mining Cost Adjustment Factor (MCAF). Deeper benches with longer haul distances carry higher mining costs, which directly affects which blocks are economic and where pit shell boundaries fall. The pit optimiser — powered by the Hochbaum Pseudo-Flow algorithm — sees the true cost of extracting each block, not a flat average. This produces more realistic pit shells that reflect actual haulage economics.

Into Scheduling

The pit shell sequence defines the mining order that the Bienstock-Zuckerberg scheduling algorithm respects. Each shell represents a practical pushback phase with minimum width and tonnage constraints. The scheduler assigns these pushbacks to time periods, respecting equipment capacity constraints, to maximise discounted value. Haulage costs influence which blocks appear in which shells, and therefore when they are scheduled.

Into Fleet Costing

Once the schedule defines tonnes per period, the DES fleet optimiser determines how many trucks and shovels are needed in each period to achieve those production targets. These fleet counts feed into the capital schedule (when to purchase equipment) and the operating cost model (fuel, tyres, maintenance, operator labour). The result is a fully integrated cost estimate where haulage assumptions flow through from road geometry to the financial model.

9. Why This Matters

The technical architecture described above is a means to an end. What matters to study managers, project directors, and investors is what it delivers in practice.

- **More defensible pit economics.** When every block carries a haulage cost derived from actual road geometry and truck physics — not a flat $/t assumption — pit shell boundaries reflect real cost structures. Shells that appear economic under averaged assumptions can move significantly when per-block haulage is modelled correctly, particularly for deep benches with long haul distances.
- **Fewer assumption mismatches.** In the conventional multi-tool workflow, the haulage model, the pit optimiser, the scheduler, and the cost model each carry their own version of key inputs. When one changes, the others can fall out of sync. A single integrated platform eliminates this category of error entirely.
- **Faster study iteration.** Evaluating a road design alternative, a different truck fleet, or a revised pushback sequence no longer requires re-running a chain of standalone

tools and manually transferring results between them. The entire pipeline — from road geometry to financial model — updates in one session.

- **Tighter capital timing.** Because the fleet optimiser sizes trucks and shovels per scheduling period and writes directly to the cost model, the capital acquisition schedule reflects the actual production profile rather than a static fleet assumption applied uniformly across the life of mine.
- **Full auditability.** Every haulage cost on every block can be traced back through the route it was computed on, the segments that route traverses, the physics parameters applied, and the truck performance curves used. There is no black box and no hidden averaging.

Conclusion

The physics of truck haulage simulation is well understood. Rimpull curves, grade resistance, rolling resistance, and segment-by-segment kinematics have been standard practice since dedicated simulation systems like Talpac and SPRY established the foundations. Discrete event simulation of truck-shovel fleets is equally mature. MiningIQ does not claim to have reinvented either of these disciplines.

What MiningIQ does is eliminate the gap between simulation and decision-making. In the conventional workflow, haulage analysis lives in one tool, pit optimisation in another, scheduling in a third, and costing in a fourth. Each boundary is a manual handoff where assumptions can diverge, data can become stale, and simplifications accumulate. MiningIQ closes that gap by embedding the physics engine, the DES, the road network, the block model, the pit optimiser, and the cost model in a single integrated platform. A change to a haul road profile propagates through cycle times, block costs, pit shells, the schedule, fleet sizing, and the financial model — in a single web session, with full auditability at every step.

The system supports both rapid analysis through the Simple Sim path and detailed fleet studies through the full DES, giving engineers the flexibility to match the level of analysis to the stage of the project. Whether at scoping study or feasibility level, the underlying physics remains the same — first-principles, per-segment, per-block — and the integration with the broader mine optimisation pipeline remains seamless.

For more information about MiningIQ and IMC Mining’s mine-to-market consulting services, visit **imcm.au** or contact our team directly.