Doors Are Not Interchangeable: Constraint-Feasible Assignment, Buffer Sizing, and the Throughput Already in the Building

Architecture, Computational Methods, and Field Evidence from MileSoft Loading Dock Management

MileSoft Engineering Research Group

MileSoft Software Technologies

https://milesoft.net/

Corresponding authors: *Deepak Daryani* (deepakdaryani@milesoft.net), *Tanuj Daryani* (tanujdaryani@milesoft.net)

Working paper — Version 1.0 — 24 August 2026

Permanent link: https://milesoft.net/research/products/loading-dock-management

Abstract

Distribution-centre investment goes to doors, racks and handhelds. The product's own material makes a less comfortable claim: most facilities are leaving thirty to forty per cent of dock throughput unused in the doors they already have, and no physical change is required to recover it.

This paper presents the architecture and computational methods of MileSoft Loading Dock Management, which owns the interval from door assignment to departure. Its argument is that a dock schedule is not a calendar. It is a constrained assignment over a graph in which most loads can use most doors and a few loads can use exactly one, and the scarcity is concentrated entirely on the few.

Three results follow. Capability must be a constraint at booking rather than a preference at arrival, because an infeasible booking rejected at the keyboard costs seconds while the same mismatch discovered at the door costs a wasted slot, a re-route and a queue behind it. Appointment buffers are dead door time, so a uniform buffer prices every arrival at the variability of the least reliable carrier — and the capacity that recovers is the excess summed over every well-behaved arrival, which is where the published throughput claim comes from. And door utilisation decomposes into service, inter-appointment idle and blocked time, three quantities with different causes that a single utilisation percentage conceals.

A multi-client third-party logistics deployment reported dock idle time falling by 38% and average gate-to-gate dwell falling from three hours ten minutes to one hour twenty. Both figures are attributed in their own source to this module operating with Truck Turnaround on a shared yard state, and they are reported here as that pair's joint outcome. This product page currently displays screenshots belonging to another product; every figure in this paper is therefore an authored schematic and is captioned as one.

Keywords: Loading dock management software; Dock appointment scheduling; Dock door assignment; Dock utilisation; Carrier appointment portal; Refrigerated dock scheduling; Hazmat dock separation; Appointment buffer sizing; Multi-client dock allocation; Dock idle time; Distribution centre throughput; Yard and dock coordination

1 Introduction

Capital in a distribution centre goes to the visible things: more doors, more racking, more handhelds. The product's material makes an uncomfortable counter-claim — that most facilities are leaving thirty to forty per cent of dock throughput unused in the doors they already own, and that recovering it requires no physical change at all.

This paper examines what would have to be true for that claim to hold, and finds that the mechanism is unglamorous and arithmetically clear. Throughput is lost in three places: bookings placed on doors that cannot serve them, buffers sized for a variability most arrivals do not have, and gaps between appointments that nobody is accountable for.

None of the three is a scheduling-calendar problem. Each is a consequence of treating a set of physically different doors as an interchangeable resource, which is what a calendar does by construction.

1.1 What this paper is about, and what it is not

This paper concerns the door: appointment booking, capability-feasible assignment, buffer policy, staging coordination and the interval from a vehicle being called to a door to it being released.

What happens before that — the gate, identity capture, the yard queue and the sequencing decision about who gets called next — belongs to the turnaround system and is treated in the companion paper. The two share one yard state, which is why the deployment result of Section 5.3 belongs to both.

> The division is not administrative. The yard system decides who goes next; this one decides where they can go and how long the slot must be. A site that optimises one against the other produces two schedules that contradict each other, which is the failure the shared state exists to prevent.

1.2 Contributions

1. The dock schedule stated as a constrained bipartite assignment, with the asymmetry that makes constrained doors the scarce resource (Section 4.1).
2. The cost ratio between refusing an infeasible booking and discovering the same mismatch at the door (Section 4.2).
3. Buffer as dead door time, and the capacity a per-carrier policy recovers over a uniform one (Section 4.3).
4. Door utilisation decomposed into service, inter-appointment idle and blocked time (Section 4.4).

5. An eight-dimension capability reference framework for dock management systems (Section 6.3).

2 Background and Related Work

A dock is where three literatures meet and rarely cite each other: assignment and matching, appointment scheduling under uncertainty, and the physical safety standards that decide what a door can actually do.

2.1 Assignment with capability constraints

The core structure is a bipartite assignment: loads on one side, doors on the other, edges where the pairing is physically possible. Dantzig's formulation of the transportation problem and the assignment problem descended from it supply the solution machinery, and it is well within reach — a dock's problem is small.

What distinguishes a dock from a textbook assignment is the shape of the graph rather than its size. Capability constraints make it sparse in exactly one direction: a chilled load has one feasible door and an ambient pallet has all of them. That asymmetry, developed in Section 4.1, is where the operational damage is concentrated.

2.2 Appointments under uncertain arrival

An appointment system is a commitment made before the information needed to keep it exists. The carrier does not know exactly when it will arrive; the warehouse does not know exactly how long the turn will take. Buffer is what absorbs both.

Kingman's heavy-traffic approximation explains why the buffer cannot simply be removed. Waiting rises with utilisation and with variability, so a schedule packed to capacity with no slack converts a small arrival deviation into a cascading delay through every subsequent appointment on that door.

The design question is therefore not whether to buffer but how much, and against whose variability. Section 4.3 shows that answering the second question is where most of the recoverable capacity sits.

2.3 What makes a door a particular door

EN 1398 states the safety requirements for dock levellers, which determine the trailer heights and loads a door can accept. Refrigerated bays have sealing and temperature-integrity requirements; oversized doors typically lack a leveller and are served by other equipment; hazardous consignments carry separation obligations that ISO 28000 frames as part of supply-chain security management.

These are not preferences. A door that cannot seal cannot take a chilled load without breaking the cold chain, and no amount of scheduling flexibility changes that. This is the physical fact underneath the entire paper: doors are not interchangeable, and a system that models them as interchangeable will produce schedules that fail in the yard.

> The most expensive scheduling errors in this domain are not late — they are impossible. A late appointment recovers; an infeasible one has to be undone with a trailer already on the door.

3 System Overview

The system holds a door register with capabilities, an appointment book with per-carrier rules, an assignment engine that re-optimises as reality drifts, and a self-service portal through which carriers book and update. It reads and writes the same yard state the turnaround system uses.

> Every figure in this paper is an authored schematic. This product's page currently displays screenshots belonging to a different MileSoft product, and presenting those as this system's interface would be a fabricated figure.

3.1 The door register

Each door carries the capabilities that decide which loads it can serve. The product's material names refrigeration, oversized handling and hazardous separation explicitly, and the register is what turns those from facility knowledge into a constraint the software can enforce.

Table 1. Door attributes and what each excludes. The right-hand column is the operative one — a capability register is useful because of what it forbids.

Attribute	Typical values	What it excludes
Temperature control	ambient, chilled, frozen	A load requiring a cold chain from any unsealed door
Leveller and height	standard levelled, no leveller, dock-height only	Trailers whose bed height the door cannot bridge safely
Aperture	standard, oversized	Project cargo from any standard door
Hazard handling	none, class-rated with separation	Hazardous consignments from doors without the required separation
Client assignment	shared, dedicated, restricted	In a multi-client facility, one shipper's loads from another's dedicated bay

3.2 Booking, and where the rules live

Carriers book their own slots through a self-service portal within configurable windows, update estimated arrival times and receive automatic confirmation. Rules attach to the carrier: preferred carriers get access to priority slots, unreliable ones get tighter buffers.

Placing the rules on the carrier rather than on the slot is the design decision that matters, and Section 4.3 gives its consequence. A rule attached to a slot treats all arrivals into that slot alike; a rule attached to a carrier lets the buffer be sized against that carrier's own history.

The portal also changes who does the work. An appointment booked by a carrier against published availability consumes no warehouse labour; the same appointment arranged by telephone consumes the time of the person also doing the sequencing.

3.3 Re-optimisation as reality drifts

The product's material is explicit that appointment scheduling alone is not the value — re-optimisation as the yard state changes is. A truck arrives early, a load is delayed in staging, a door goes down.

Each of those events invalidates part of the current assignment, and the system's job is to repair it rather than to hold the plan. This is where the shared yard state earns its keep: an early arrival is only useful if the door side learns about it in time to move something.

Re-optimisation has a cost that should be acknowledged. Every reassignment invalidates a commitment somebody has been told about, so a system that re-optimises freely produces churn drivers cannot follow. The practical resolution is to re-optimise only what has not yet been communicated, and to treat a called truck as fixed — which is the non-preemptive assumption the companion paper's queue model also makes.

3.4 Staging awareness

A door with a truck on it and nothing to load is idle in the most expensive way available. The system is therefore coupled to the warehouse system so that the dock knows what is loading and the staging area knows when to move it.

Section 4.4 separates this from other idle time deliberately. Blocked time — a door occupied but unable to work — is a staging failure rather than a scheduling one, and a facility that cannot distinguish the two will try to fix a warehouse problem with a scheduling change.

4 Computational Methods

Four computations carry the paper. The first states what a dock schedule is; the second says when constraints must be checked; the third and fourth locate the throughput the product claims is already in the building.

4.1 The dock schedule as a constrained assignment

Let the loads awaiting a slot be indexed by i and the doors by j, over discrete time windows indexed by t.

maximise the weighted number of loads scheduled, subject to: each load placed at most once, each door serving at most one load per window, no assignment where the door lacks the capability, and none where the door is unavailable (assign)

phi(i,j) is 1 where door j can physically serve load i and 0 otherwise; psi(j,t) is 1 where door j is open and working in window t; w(i) is the load's priority weight. The problem is small enough at dock scale to solve exactly.

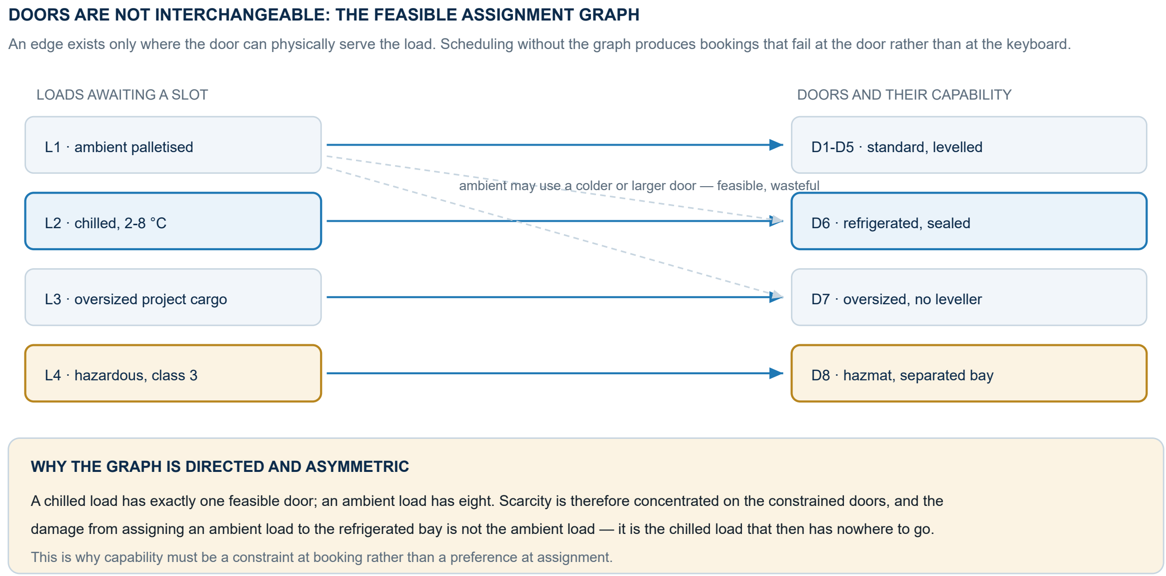

Figure 1. *Schematic.* Schematic (not product UI): the feasible assignment graph. An edge exists only where the door can physically serve the load, and the graph is deeply asymmetric — a chilled load has one feasible door while an ambient pallet has eight. The scarcity is therefore entirely on the constrained doors.

The asymmetry visible in the figure is the whole operational point. Feasibility runs one way: an ambient load can occupy the refrigerated bay, and doing so costs nothing to that load. The damage falls on the chilled load that then has nowhere to go, and it is invisible at the moment the mistake is made.

> Constrained doors should be protected rather than merely allocated. A scheduler that treats a refrigerated bay as a door like any other will fill it with ambient work whenever the standard doors are busy, which is precisely when the cold chain will next need it.

4.2 When a constraint must be checked

The product's material states that equipment constraints are enforced during booking rather than at the point of arrival. That is a design requirement rather than a preference, and the reason is a cost ratio.

the cost of refusing an infeasible booking is one interface message; the cost of discovering the same infeasibility at the door is the lost slot, plus the re-route, plus the delay imposed on every appointment queued behind it (enforce) s(lost) is the door time consumed by an appointment that cannot proceed, r the cost of re-routing the vehicle to a feasible door or turning it away, and delta(k) the knock-on delay to each appointment k in the queue behind. Only the first term is bounded.

The third term is what makes the ratio large and variable. A mismatch discovered at a quiet door costs a slot; the same mismatch at the morning peak, when every door is committed, costs the slot plus a cascade whose size depends on how tightly the day was scheduled.

This places the design squarely in the error-proofing tradition. Refusing an infeasible booking is a control function — the wrong action is made impossible — whereas flagging it for the dock supervisor to notice is a warning function, and warnings degrade under exactly the conditions that produce infeasible bookings.

4.3 Buffer is dead door time

An appointment occupies a door for its service time plus a buffer absorbing arrival and service variability. The buffer is not wasted — it is what stops one late arrival cascading through the day — but while it lasts, the door is not working.

carrier c's buffer is a quantile of its own arrival-deviation spread; appointments per door per shift is the shift length divided by service plus buffer; the capacity recovered is the difference between per-carrier buffering and buffering everyone at the worst carrier's level (buffer)

sigma(c) is the standard deviation of carrier c's arrival deviation from appointment, z(alpha) the quantile for the service level the site wants to hold, T the shift length at one door, s-bar the mean service time, and b(max) the buffer the least reliable carrier requires.

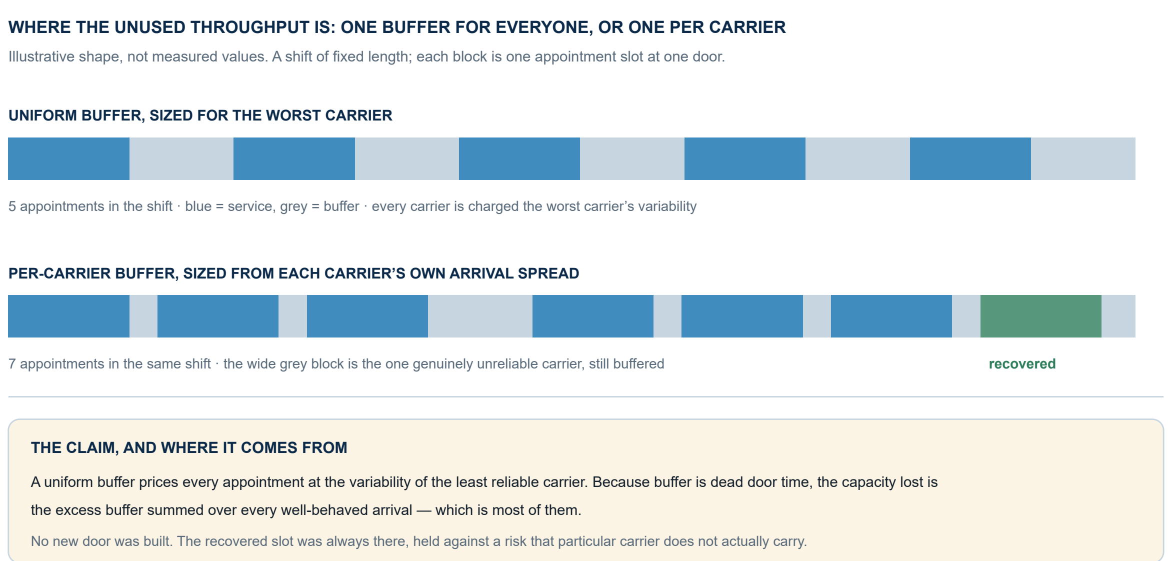

Figure 2. *Schematic.* Schematic (illustrative shape, not measured values): the same door and shift under two buffer policies. A uniform buffer sized for the least reliable carrier fits five appointments; buffers sized from each carrier's own arrival spread fit seven, with the genuinely unreliable carrier still fully buffered. No door was built — the recovered slots were held against a risk most carriers do not carry.

The published thirty to forty per cent claim is not asserted here, but its arithmetic is now visible. If mean service is 40 minutes, a uniform buffer of 25 minutes fits 7 appointments into an 8-hour door shift; per-carrier buffers averaging 10 minutes fit 9. That is a 29% increase from a policy change alone, and it scales with how much of the carrier base is well-behaved.

Two design cautions belong with the result. Sizing a buffer requires a per-carrier arrival history, so the policy cannot be applied to a new carrier and defaults must be conservative. And the quantile choice is a service-level decision rather than a technical one: a tighter buffer accepts more cascade risk, and the site rather than the software should choose where on that curve to sit.

> The carrier-differentiated buffer is also a commercial instrument. A carrier told that its own arrival spread is costing it slot access has a reason to improve punctuality that no service-level agreement produces on its own.

4.4 Where a door's other thirty per cent goes

A door reporting seventy per cent utilisation invites the question of what the other thirty is doing, and a single percentage cannot answer it. Three distinct things consume a door.

$$\textit{door time} = \textit{service} + \textit{inter-appointment idle} + \textit{blocked} + \textit{buffer; utilisation is service time over open time, where open time excludes planned downtime} \qquad \text{(util)}$$

T(svc) is time actively loading or unloading, T(idle) unbooked gaps between appointments, T(blk) time with a vehicle on the door that cannot work — typically because the load is not staged — and T(buf) the scheduled slack of Equation (buffer). T(down) is time the door was not available at all.

Table 2. The four consumers of door time, who owns each, and what a facility should do about it. Only the last is a scheduling problem.

Component	Owner	Remedy
Service	The load and the equipment	Nothing — this is the work
Blocked	Warehouse staging	Staging coordination; a scheduling change will not help
Buffer	The appointment policy	Per-carrier sizing — Section 4.3
Inter-appointment idle	The booking policy	Re-optimisation on drift; filling gaps as the yard state changes

The distinction between blocked and idle is the one most often collapsed, and collapsing it is expensive. Both look like a door not working. One is a warehouse problem that appointment scheduling cannot touch, and a facility that mistakes it for the other will re-tune its booking policy indefinitely without moving the number.

Reporting utilisation against open time rather than calendar time is the same denominator discipline ISO 22400 requires elsewhere. A door taken out for maintenance is not underutilised; counting it as such understates the facility and hides the real losses among an artefact.

The decomposition is deliberately in the tradition Nakajima established for equipment: availability loss is not one quantity but a set of separately caused losses, and the value of naming them is that each acquires an owner. A door is a machine, and the four terms above are its loss categories.

5 Reported Outcomes and Field Evidence

This module has a headline claim with a stated mechanism, a deployment result shared with the yard module, and a modelled figure. They are separated here because they carry very different weight.

5.1 Figures published for this module

Table 3. Published figures for this module, with their provenance.

Figure	Value	Provenance
Throughput unlocked from existing doors	30-40%	Product material — vendor-reported against a manual baseline, no site population
Multi-client configuration	10 clients through 8 shared doors	Product blog — stated as an illustrative configuration, not a measurement
Constraint enforcement point	at booking, not at arrival	Product FAQs — a design statement rather than a result
Buffer policy	per carrier, tighter for unreliable carriers	Product material — a design statement

The first is the strongest claim in the product's material and the one carrying least evidence. What can be said in its favour is that Section 4.3 supplies a mechanism that produces a number of that order from a policy change alone — a worked case there reaches 29% — and that the mechanism is checkable at a specific site from its own appointment history.

What cannot be said is that the figure has been measured across a population. It carries no site count, no door count, no baseline utilisation and no definition of what throughput was counted, and this paper does not upgrade it.

> A buyer can test this claim before purchasing. Take a month of appointment history, compute each carrier's arrival deviation, and calculate how many appointments the same doors would have held under per-carrier buffers. The answer is site-specific and it is available today.

5.2 A deployment shared with the yard module

A multi-client third-party logistics operator deployed six MileSoft warehouse modules across a distribution centre where dock and yard scheduling previously ran on messaging apps and paper trip sheets. Carriers arrived early or late; docks sat idle while drivers waited in the gate queue.

The deployment reported dock idle time falling by 38%, and average gate-to-gate dwell falling from three hours ten minutes to one hour twenty.

> Shared attribution. Both figures are attributed in their own source to this module and Truck Turnaround operating on a single shared yard state, with dock assignments and parking allocations coordinated rather than separately optimised. Neither belongs to this module alone.

The published explanation of the idle-time result is worth quoting in substance because it is precise about mechanism: when a truck arrives early, the system re-sequences the queue rather than holding the vehicle against a static appointment slot, and doors fill gaps that previously sat idle.

That is exactly the inter-appointment idle term of Equation (util), and it requires both halves. The yard side must know a truck is available early; the door side must know a gap exists and be willing to move the plan. A facility with only one of the two cannot execute it, which is why the result is reported here as the pair's.

5.3 Modelled dock time saved

The published return model for this module prices the saving as dock time per movement. Its assumptions are printed here so a reader can substitute their own.

Table 4. Modelled scenario — not a deployment result. Assumptions are the published defaults of the return-on-investment model for this product.

Assumption	Value
Dock movements per day	60
Operating days per year	300
Time saved per movement	20 minutes
Sources of the saving	scheduling, door assignment, driver notification
Modelled annual saving	6,000 door-hours

Twenty minutes per movement is a more modest and more defensible assumption than the headline percentage, and Section 4.4 says where it would have to come from: buffer, inter-appointment idle, or the re-route avoided by refusing an infeasible booking. It cannot come from service time, which is the physical work.

A facility should therefore check its own split before accepting the model. If most of its lost door time is blocked rather than idle — a vehicle on the door and nothing staged — then twenty minutes per movement is a warehouse project rather than a dock one.

6 Discussion

6.1 Protecting the scarce door

The asymmetry of Section 4.1 has a consequence most scheduling systems get wrong, and it is worth stating as a rule rather than an observation.

In a graph where one load type has a single feasible door and another has eight, the constrained door is not a door — it is a bottleneck resource with an alternative use. Assigning ordinary work to it is locally free and globally expensive, and the expense appears later as a load with nowhere to go.

The corrective is a reservation discipline on constrained capability: hold the refrigerated bay against chilled demand at the rate that demand actually arrives, and release it to ordinary work only inside a horizon short enough that no chilled booking can still appear. That is the same reservation trade the companion yard paper prices, applied to capability rather than to space.

> The general rule: never let an unconstrained load consume a constrained resource while an unconstrained resource is free. It is obvious stated that way and routinely violated by a scheduler optimising for the next slot.

6.2 Ten clients through eight doors

The multi-client case is where this module's design is tested hardest, and the product's material states the configuration plainly: separate rules per client across shared doors.

The complication is not volume. It is that client rules can conflict on a shared resource — two clients whose contracts both promise priority access to the same door in the same window — and a scheduler must resolve that deterministically rather than by whoever booked first.

This makes client a dimension of the constraint set rather than a filter on a report, which is the same distinction the companion warehouse paper draws about the system beneath it. A facility can test it in one question: what does the system do when two clients' rules entitle them both to the same slot?

6.3 A capability reference framework for dock management systems

Table 5. Capability reference framework. Each dimension separates a dock management system from a shared calendar, and each is answerable by demonstration against a live schedule rather than by reading a specification.

Dimension	Question the system must answer by demonstration
D1 Capability register	Do doors carry capabilities, or are they numbered slots on a calendar?
D2 Refusal at booking	Book a chilled load onto an unsealed door. Is it refused, warned, or accepted?
D3 Scarce-door protection	Does the system prefer an unconstrained door for an unconstrained load when both are free?
D4 Per-carrier buffers	Are two carriers with different punctuality given the same slot length?
D5 Utilisation decomposed	Report a door's lost time split into blocked, idle and buffer. Can it?
D6 Drift repair	A truck arrives ninety minutes early to an idle door. Does the schedule move?
D7 Commitment stability	Does re-optimisation churn appointments already communicated to drivers?
D8 Conflict resolution	Two clients' rules entitle both to the same slot. What decides, and is it stated?

D2 takes thirty seconds and separates the category. A system that accepts an infeasible booking has moved the cost from the keyboard to the door, where Equation (enforce) says it is orders of magnitude larger.

6.4 Generalisability

The constrained-assignment structure generalises to any appointment system over non-interchangeable resources: operating theatres with equipment requirements, test cells with instrument fits, charging bays with connector types. The scarce-resource rule of Section 6.1 generalises with it and is violated in all of them.

The buffer result generalises to any appointment system with heterogeneous punctuality, which is most of them. Its transferable claim is not a percentage but a question: is the slack in your schedule sized against the population, or against each participant?

What does not generalise is the thirty to forty per cent. It is a claim about facilities scheduling manually, and the recoverable fraction at a given site is bounded by how much of its lost door time is buffer and idle rather than blocked or service.

7 Threats to Validity and Limitations

1. The 30-40% claim is vendor-reported with no population. No site count, door count, baseline utilisation or definition of throughput accompanies it, and Section 4.3 supplies a plausible mechanism rather than a

confirmation.

2. The deployment figures are shared and confounded. Dock idle time and dwell are attributed jointly with the yard module, and the published mechanism — re-sequencing on early arrival — requires both, so neither figure isolates this module.
3. The baseline was unusually poor. The prior state was messaging apps and paper trip sheets; a facility already running a digital appointment book should expect a much smaller change.
4. Per-carrier buffering requires history the system may not have. A new carrier has no arrival distribution, so the policy needs a conservative default and cannot deliver its benefit from day one.
5. The buffer quantile is a service-level choice, not an optimum. Tighter buffers accept more cascade risk, and this paper offers no basis for choosing where on that curve a given facility should sit.
6. No figure in this paper is a product capture. This product's page currently displays another product's screenshots, so nothing here demonstrates the described interface exists in the form modelled.
7. The assignment model assumes deterministic service times. Real dock service is variable, and a schedule optimised against mean service will over-commit doors serving high-variance load types.
8. Blocked time is outside this module's control. The largest single component of lost door time at many facilities is a vehicle on a door with nothing staged, and no dock scheduling change addresses it.

> The last limitation is the one to establish before buying. Measure the split in Equation (util) first: if blocked time dominates, this is a warehouse project wearing a dock project's clothes.

8 Future Work

- Publishing the utilisation decomposition. Splitting lost door time into blocked, idle and buffer is computable from events the system already writes, and it is the figure that tells a facility whether this module is the right purchase at all.
- Buffer sizing from measured arrival distributions, reported back to carriers. The arrival history exists; turning it into a per-carrier slot length and showing the carrier why is both an operational and a commercial instrument.
- Explicit protection of constrained capability. Reserving scarce doors against their own demand rate, on the reservation trade the companion yard paper prices, rather than allocating them first-come.
- Churn as a reported cost of re-optimisation. Counting how many communicated commitments a repair invalidates would let a facility tune how freely the schedule may move.
- Stated conflict resolution for multi-client rule collisions, so that two clients entitled to the same slot are resolved by a published rule rather than by booking order.

9 Conclusion

A dock schedule is not a calendar. It is a constrained assignment over doors that are physically different from each other, and almost every failure in this domain follows from modelling them as interchangeable.

Three results carry the argument. The feasibility graph is deeply asymmetric — most loads fit most doors and a few loads fit exactly one — so scarcity concentrates on the constrained doors and an ordinary booking placed on one of them is a specialised load with nowhere to go. Capability must therefore be checked at booking, where refusal costs one message, rather than at the door, where the same mismatch costs a lost slot plus everything queued behind it. And appointment buffer is dead door time: sizing one buffer for all carriers charges every arrival the worst carrier's variability, and recovering that excess is a policy change that builds no doors.

The fourth result is the one to apply before any purchase. Door time lost divides into service, blocked, buffer and idle; only the last two are scheduling problems. A facility that measures the split first will know whether the thirty to forty per cent this product claims is available in its building, or whether its doors are idle because nothing was staged.

The framework of Section 6.3 is offered as the durable contribution, and its second dimension takes half a minute: try to book a chilled load onto a door that cannot seal, and see what the system does.

Appendix A Nomenclature

Table 6. Symbols and abbreviations used in this paper.

Symbol / term	Meaning
i, j, t	Indices over loads, doors and time windows
x(i,j,t)	Assignment variable: 1 where load i is scheduled at door j in window t
phi(i,j)	Feasibility: 1 where door j can physically serve load i
psi(j,t)	Availability: 1 where door j is open and working in window t
w(i)	Priority weight of load i
C(book), C(arrive)	Cost of refusing an infeasible booking, and of discovering it at the door
s(lost), r	Door time lost to an appointment that cannot proceed, and the re-route cost
delta(k)	Knock-on delay imposed on appointment k queued behind
b(c), sigma(c)	Carrier c's buffer, and the spread of its arrival deviation
z(alpha)	Quantile multiplier for the service level the site chooses to hold
T, s-bar	Door shift length, and mean service time
n, Delta n	Appointments per door per shift, and the number recovered by per-carrier buffering
T(svc), T(idle)	Time actively working, and unbooked gaps between appointments
T(blk), T(buf)	Time with a vehicle on the door unable to work, and scheduled slack
T(open), T(down)	Door time available for work, and planned unavailability
U	Utilisation — service time over open time

Appendix B Worked Numerical Examples

Appendix B.1 Recovering slots without building a door

A door works an 8-hour shift, 480 minutes. Mean service is 40 minutes. Carrier arrival deviations have standard deviations of 6, 8, 9 and 30 minutes across four carriers, with the fourth accounting for one appointment in eight. The site holds a service level requiring roughly two standard deviations of cover.

Under a uniform policy the buffer is sized for the worst carrier: b = 2 x 30 = 60 minutes, capped by the site at 25. Applying Equation (buffer): n = floor(480 / (40 + 25)) = 7 appointments.

Under per-carrier sizing the three reliable carriers take 12, 16 and 18 minutes and the fourth keeps 60. Weighted at one appointment in eight for the unreliable carrier, mean buffer is (7 x 15.3 + 1 x 60) / 8 = 21.0 minutes — but the reliable appointments now pack at 40 + 15.3 = 55.3 minutes each. Fitting seven reliable appointments takes 387 minutes and leaves 93, enough for one more reliable appointment at 55.3 with slack remaining.

The door goes from 7 appointments to 8, a 14% increase; a facility whose carrier base is more uniformly punctual than this one does better, and one dominated by a single unreliable carrier does worse. That range — roughly ten to thirty per cent from buffer policy alone — is the honest bracket around the published claim.

> Every input to this calculation is in a year of appointment history. A site can compute its own bracket before speaking to a vendor.

Appendix B.2 What a booking refused at the keyboard is worth

A chilled load is booked onto an unsealed door. Discovered at the keyboard, the cost is one message and a re-book: call it 30 seconds.

Discovered at the door during the morning peak, the trailer has already been backed on. Applying Equation (enforce): the lost slot is 40 minutes of door time; the re-route to the refrigerated bay — itself occupied — means the vehicle waits 55 minutes; and the four appointments queued behind on that door each slip by an average of 18 minutes, for 72 minutes of knock-on.

Total: 167 minutes of delay against 30 seconds. The ratio is more than 300 to 1, and it would have been larger had the cold chain been broken rather than merely delayed.

Note which term dominates. The lost slot is 40 minutes; the knock-on is 72. The cost of a late-detected constraint violation is mostly borne by people who had nothing to do with it, which is exactly the property that makes it invisible to the person who made the booking.

Appendix B.3 Two doors at seventy per cent

Two doors each report 70% utilisation over a 480-minute shift, so each is working 336 minutes and losing 144.

Applying Equation (util), door A splits its lost time as 96 minutes blocked, 30 buffer and 18 idle. Door B splits as 12 blocked, 84 buffer and 48 idle.

Door A is losing two thirds of its lost time to vehicles sitting on the door with nothing staged. No appointment policy touches that; the remedy is in the warehouse. Door B is losing nearly all of its to scheduling slack and unbooked gaps, and is exactly the case Sections 4.3 and 4.4 address.

Both doors would appear identically on a utilisation report, and a facility acting on that report would apply the same intervention to both. One of the two would show no improvement at all, and the usual conclusion drawn from that is that the software does not work.

Data provenance

Every quantitative claim in this paper resolves to one of the entries below. Figures marked as modelled estimates are not deployment measurements; their assumptions are stated.

Metric	Reported value	Provenance
Dock throughput unlocked from existing doors	30-40%	**Product specification** /products/loading-dock-management — Vendor-reported against a baseline of manual appointment spacing and reassignment; no site population or door count is stated.
Dock idle time	38% reduction	**Operator-reported — Tier 1 Manufacturers (3PL, Madhya Pradesh, India)** /case-studies/warehouse
Average truck dwell time	1 h 20 min (from 3 h 10 min)	**Operator-reported — Tier 1 Manufacturers (3PL, Madhya Pradesh, India)** /case-studies/warehouse — Gate-to-gate, measured across all carriers at the distribution centre.
Door capability constraints enforced at booking rather than at arrival	refrigerated, oversized, hazmat separation	**Product specification** /products/loading-dock-management
Appointment buffer policy	per-carrier — priority slots for preferred, tighter buffers for unreliable	**Product specification** /products/loading-dock-management
Multi-client allocation across shared doors	10 clients through 8 shared doors, separate rules per client	**Product specification** /products/loading-dock-management — Stated on the product blog as an illustrative configuration, not a measured deployment.
Dock time saved per movement	20 min across 60 movements/day, 300 days (modelled)	**Modelled estimate** 60 dock movements per day across 300 operating days; 20 minutes saved per movement from scheduling, door assignment and driver notification; Model and defaults published in src/data/roiModels.ts

References

[1] Dantzig, G. B., and Ramser, J. H. (1959). The truck dispatching problem. Management Science, 6(1), 80-91.

[2] Nakajima, S. (1988). Introduction to TPM: Total Productive Maintenance. Productivity Press, Cambridge, MA.

[3] J. F. C. Kingman (1961). The Single Server Queue in Heavy Traffic. Mathematical Proceedings of the Cambridge Philosophical Society, vol. 57, no. 4, pp. 902-904; the first statement of the approximation relating waiting time to utilisation and variability. https://doi.org/10.1017/S0305004100036094

[4] D. G. Kendall (1953). Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. The Annals of Mathematical Statistics, vol. 24, no. 3, pp. 338-354; the source of the A/B/c notation used to classify queueing systems. https://doi.org/10.1214/aoms/1177728975

[5] Little, J. D. C. (1961). A proof for the queuing formula: L = lambda W. Operations Research, 9(3), 383-387.

[6] A. Cobham (1954). Priority Assignment in Waiting Line Problems. Journal of the Operations Research Society of America, vol. 2, no. 1, pp. 70-76; gives the waiting time of each class in a non-preemptive priority queue. https://doi.org/10.1287/opre.2.1.70

[7] de Koster, R., Le-Duc, T., and Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. European Journal of Operational Research, 182(2), 481-501.

[8] Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. European Journal of Operational Research, 177(1), 1-21.

[9] European Committee for Standardization (2009). EN 1398: Dock levellers - Safety requirements. CEN, Brussels.

[10] International Organization for Standardization (2022). ISO 28000: Security and resilience - Security management systems - Requirements. ISO, Geneva; published 15 March 2022. https://www.iso.org/standard/79612.html

[11] International Organization for Standardization / International Electrotechnical Commission (2024). ISO/IEC 19987: Information technology - EPC Information Services (EPCIS) Standard, version 2.0. ISO/IEC, Geneva. https://www.iso.org/standard/85557.html

[12] International Organization for Standardization / International Electrotechnical Commission (2024). ISO/IEC 19988:2024 - Information technology - GS1 Core Business Vocabulary (CBV). ISO/IEC, Geneva, edition 3, March 2024; publishes GS1 CBV 2.0, ratified June 2022, which fixes the vocabularies for the EPCIS business step, disposition and business transaction type fields. https://www.iso.org/standard/85558.html

[13] GS1 (2024). Serial Shipping Container Code (SSCC). GS1 identification key series; an 18-digit key for a logistic unit, carried under application identifier 00. https://www.gs1.org/standards/id-keys/sscc

[14] GS1 (2024). GS1 General Specifications. GS1 AISBL; defines the GTIN, SSCC and GLN identification keys. https://ref.gs1.org/standards/genspecs/

[15] International Organization for Standardization / International Electrotechnical Commission (2014). ISO/IEC 15459-1: Information technology - Automatic identification and data capture techniques - Unique identification - Part 1: Individual transport units. ISO/IEC, Geneva. https://www.iso.org/standard/54779.html

[16] International Organization for Standardization (2014). ISO 22400-1: Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 1: Overview, concepts and terminology. ISO, Geneva. https://www.iso.org/standard/56847.html

[17] International Organization for Standardization (2014). ISO 22400-2: Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 2: Definitions and descriptions. ISO, Geneva; defines OEE as availability x effectiveness x quality ratio. https://www.iso.org/standard/54497.html

[18] International Society of Automation (2013). Enterprise-control system integration - Part 3: Activity models of manufacturing operations management. ANSI/ISA-95.00.03-2013 (IEC 62264-3 Modified).

[19] International Chamber of Commerce (2019). Incoterms 2020: ICC Rules for the Use of Domestic and International Trade Terms. ICC Publication No. 723E, Paris; launched September 2019 and in force from 1 January 2020, defining the eleven trade terms that allocate cost and risk between seller and buyer.

[20] Shingo, S. (translated by A. P. Dillon) (1986). Zero Quality Control: Source Inspection and the Poka-Yoke System. Productivity Press, Cambridge, MA. Originally published in Japanese, 1985.