A Wave Is a Batch, and a Batch Is a Deadline: Strategy Selection, the Travel-Latency Trade, and the Recovery Window

Architecture, Computational Methods, and a Modelled Scenario for MileSoft Order Fulfillment

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/order-fulfillment

Abstract

The product's material draws a distinction this paper takes as its starting point: a pick-list printer is not a fulfilment engine. Printing the work in a sensible order is a formatting problem. Deciding which orders travel together, by which strategy, and whether the result can be finished before the carrier leaves is a scheduling problem with a hard deadline in it.

This paper presents the architecture and computational methods of MileSoft Order Fulfillment. Its organising claim is that a wave is a batch and a batch is a commitment: grouping orders amortises travel across them, and every order in the group waits for the group. That trade is computable, and it is the reason wave size is a decision rather than a preference.

Three results follow. The four pick strategies the product selects between — single order, batch, cluster and zone — are not a quality ranking but four points on one trade, each moving travel into sortation, consolidation or latency, so the right choice is a function of the order mix rather than of the warehouse. Under zone picking a wave's completion time is the maximum across zones rather than the sum, which makes labour imbalance a direct cost in makespan and explains why zone strategies need balancing that batch strategies do not. And the value of surfacing a promise failure at wave time rather than at the dock is not a faster answer but a larger set of available answers, because substitution, splitting, expediting and re-promising all require time that the dock no longer has.

This module is named in a published third-party logistics deployment but no result is attributed to it there, and none is claimed here. The product page's cycle-time and accuracy claims carry no baseline and are reported as claims. Section 5 is a modelled scenario with every assumption printed, and every figure is an authored schematic because this product has no captures of any kind.

Keywords: Order fulfillment software; Wave planning; Pick path optimization; Batch picking; Zone picking; Cluster picking; Carrier cutoff scheduling; Order cycle time; Pick accuracy; Returns and RMA processing; B2B and B2C fulfilment; Outbound staging

1 Introduction

The product's material makes a distinction worth taking literally: a pick-list printer is not a fulfilment engine. Printing today's orders in a sensible sequence is a formatting problem, and most systems in this category solve it well.

The scheduling problem underneath is harder. Which orders should travel together. By which picking strategy. Whether the resulting work can be finished, packed and staged before a carrier that will not wait. Those are decisions with consequences, and a system that does not make them is leaving them to whoever releases the work.

This paper's organising claim is that a wave is a batch and a batch is a commitment. Grouping orders is what makes picking efficient, and it is also what makes an order wait — so wave size is a trade with two computable sides, and it should be set against a target rather than inherited.

1.1 What this paper is about

This paper concerns the outbound batching decision: how orders become waves, which pick strategy each wave uses, how labour is distributed across it, and whether the result meets its deadline.

It does not concern where stock is slotted — that is the warehouse system's decision and the companion paper on it develops the travel-weighted objective. It does not concern where stock actually is, which is the tracking layer's job. And it does not concern which door the load leaves through, which belongs to the dock paper. This module consumes all three and produces the work.

> The dependency runs one way and it is worth naming. A perfect wave plan against a wrong position produces a perfectly sequenced failed pick, which is why the accuracy work in the tracking layer is upstream of everything here.

1.2 Contributions

1. The wave stated as a batch, with the travel saving and the latency cost both computed (Section 4.1).
2. The four pick strategies placed on one trade rather than ranked, with the order-mix conditions that select between them (Section 4.2).
3. Zone makespan as a maximum across zones, making labour imbalance a direct cost in completion time (Section 4.3).
4. The recovery window — what surfacing a promise failure early actually buys, expressed as a set of responses rather than as time (Section 4.4).

5. An eight-dimension capability reference framework for fulfilment engines (Section 6.3).

2 Background and Related Work

Order picking is the most studied warehouse activity and the results relevant here are settled. What is rarely made explicit is which of them a given wave-planning decision is trading against which.

2.1 Travel dominates, which is why batching works at all

De Koster, Le-Duc and Roodbergen's review establishes that travel dominates picking time. That is the premise of every result in this paper: batching is worth doing because the walk, not the pick, is the expense.

Petersen's comparison of routing policies and Roodbergen and de Koster's aisle-routing work supply the sequencing inside a single tour. Gu, Goetschalckx and McGinnis place both inside the facility design decisions that constrain them.

The consequence for wave planning is direct. If travel is amortised across the orders in a tour, then the marginal travel cost of adding an order to a batch is small whenever that order's lines are near lines already in the batch — and near zero when they are on the same face.

2.2 What batching costs, and where the cost lands

The efficiency argument for batching is usually presented without its counterpart, which is that a batch has a flow time and every order inside it inherits the whole batch's.

Little's Law connects the two. An order's time in the fulfilment system is the wait for its wave to form plus the wave's own duration, and the second term grows with wave size while the per-order travel falls. There is a minimum somewhere, and it moves with how urgent the orders are.

> This is why an e-commerce operation and a wholesale one settle on different wave sizes from the same warehouse and the same software. They are minimising different things.

2.3 The deadline that makes it a scheduling problem

A carrier cutoff is a hard deadline. Work that misses it does not cost more; it ships tomorrow, which is a different outcome entirely and one a customer has been promised against.

This turns wave sizing from an optimisation into a constrained one. A wave that is efficient and finishes after the cutoff is worse than an inefficient wave that finishes before, and the constraint binds most exactly when volume is highest.

It also means the relevant statistic is completion time rather than mean pick rate. A wave is finished when its last line is picked, packed and staged, and Section 4.3 shows that under zone strategies the last line is determined by the slowest zone rather than by the average.

3 System Overview

The system holds an order pool with promise dates and carrier assignments, a wave generator, a strategy selector, a pick-path optimiser, a labour model, an outbound staging plan aligned to the dock, carrier integrations for manifest, label and tracking, and a returns workflow.

> Every figure in this paper is an authored schematic. This product's page carries no screenshots of any kind, and no image belonging to another product is used to stand in for one.

3.1 What a wave has to respect

Wave generation groups orders subject to constraints the product's material names explicitly: carrier cutoff times, customer delivery windows, dock assignments and labour capacity.

Table 1. What each wave constraint is, and what happens when it is ignored. Only the last is recoverable within the shift.

Constraint	Source	Consequence of ignoring it
Carrier cutoff	The carrier's schedule	The load ships tomorrow
Customer window	The order's promise	A delivery outside the agreed window
Dock assignment	The dock plan	Output staged at the wrong door and re-sorted
Labour capacity	The shift roster	A wave that cannot finish in the time available

The dock constraint is the one most often left out, and leaving it out produces a specific and visible failure: correct picking whose output arrives at the staging area in an order nobody can load, so it is sorted again at the door. The product's material describes the alternative as staging in the right sequence before the carrier arrives, which is a wave-sequencing decision rather than a dock one.

3.2 Strategy selection per wave

The engine chooses among single-order, batch, cluster and zone picking per wave, from the order mix. The product's material gives its heuristic directly: high-line multi-unit orders use batch, large counts of small orders use zone, and time-critical orders use single-order.

That is a reasonable heuristic and Section 4.2 supplies the structure underneath it, which is more useful than the heuristic itself because it says why — and therefore what to do with an order mix the heuristic does not name.

3.3 Two order profiles in one engine

Business-to-business orders are pallet-level, bulk and scheduled; consumer orders are unit-level, time-sensitive and carrier-integrated. The product's material states both run in the same wave engine with different rules applied per order type.

Section 4.1 explains why this is harder than it sounds and why the single-engine claim is worth testing. The two profiles want opposite wave sizes — the wholesale order is indifferent to a few hours of latency and benefits from large batches; the consumer order is the reverse — so mixing them in one wave gives each the other's optimum.

> One engine is the right architecture. One wave usually is not, and a system that merges the profiles rather than partitioning them will underperform on both.

3.4 Returns as an inbound flow with a routing decision

Returns generate a return authorisation record with putaway routing configured by reason — restock, refurbish, dispose — and returned stock becomes available in the warehouse record as soon as it is inspected and confirmed.

The design point worth noting is the absence of a separate returns database. A return that lives outside the main record is stock that exists physically and not systemically, which is the drift condition the companion tracking paper defines — and it is a drift a facility creates deliberately every time it takes a return.

4 Computational Methods

Four computations carry the paper: what a wave costs and buys, how a strategy is chosen, what determines when a wave finishes, and what early detection is actually worth.

4.1 The wave as a batch, both sides

Batching amortises travel and imposes latency. Both are computable and they move in opposite directions in wave size.

work per order falls with wave size - the tour's travel T(b) is divided across b orders, leaving only per-line handling; order flow time rises with it - half the fill time plus the wave's own duration; the two derivatives have opposite signs (batch)

b is the number of orders in a wave, T(b) the travel time of the tour serving them, s-bar the handling time per line, l-bar the mean lines per order, and lambda the order arrival rate. The flow-time expression is Little's Law applied to wave formation.

The shape of T(b) is what decides how much batching is worth. Because travel dominates and a tour visiting more faces grows sublinearly in the number of faces, T(b)/b falls steeply at first and then flattens — so the first few orders added to a batch deliver most of the saving and the twentieth delivers very little.

That asymmetry has an operational consequence worth stating. There is usually a modest wave size capturing most of the travel saving at a small latency cost, and going beyond it buys diminishing efficiency at linearly increasing flow time. A facility whose waves are large because a previous system defaulted to large is probably past that point.

Wave size should be set from a target order cycle time and the shape of T(b), not from what fits on a screen or what the shift pattern suggests.

4.2 Four strategies, one trade

The four strategies are usually presented as a maturity ladder. They are not. Each moves work from travel into something else, and which is cheapest depends on the order mix.

single order costs one tour per order; batch costs one tour plus a sort per line picked; cluster costs a slightly longer tour plus a lighter sort into carts; zone costs the slowest zone's tour plus a consolidation step per order (strategy)

T(1) is a single-order tour, T(b) a batched tour over b orders, T(z) the tour within zone z, sigma the sortation cost per line, sigma(c) the lighter cart-sortation cost, and gamma the downstream consolidation cost per order.

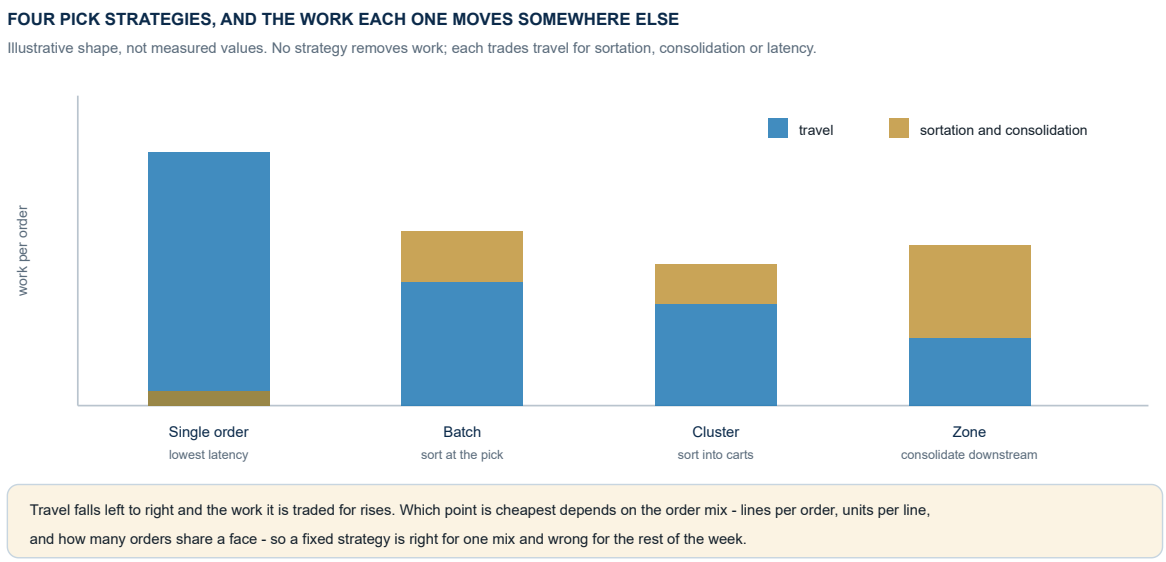

Figure 1. *Schematic.* Schematic (illustrative shape, not measured values): four strategies as four points on one trade. Travel falls from left to right and the work it is exchanged for rises. No strategy removes work; each relocates it into sortation, consolidation or latency, so the cheapest point depends on the order mix rather than on the warehouse.

Reading the terms explains the product's heuristic. Batch wins when orders share faces, because T(b) barely grows while b does. Zone wins on very large counts of small orders, because the max over zones is small when each zone's work is small and consolidation is cheap per order when orders are one or two lines. Single order wins when latency matters more than efficiency, which is exactly the time-critical case.

It also identifies the case the heuristic does not name: many orders each with many lines spread across zones. There the sortation term in batch and the consolidation term in zone both grow, and cluster — a middle point with lighter sortation into physical carts — is often the answer.

4.3 Why a zone wave finishes when its slowest zone does

Total labour and completion time are different quantities, and under zone picking they behave very differently.

total labour is the sum across zones; makespan is the worst zone's labour divided by its pickers, plus consolidation; the ideal makespan is total labour spread evenly across all pickers (makespan)

L(z) is the pick work in zone z, n(z) the pickers assigned to it, and gamma(cons) the consolidation step after zone picks converge. The gap between M and M(ideal) is pure imbalance cost.

The wave's deadline is met or missed on M, not on L. A wave whose total labour comfortably fits the shift can still miss a cutoff because one zone holds a third of the lines and a fifth of the pickers — and no amount of idle capacity elsewhere helps, because a picker in zone A cannot pick zone C's lines without becoming a zone C picker.

This is the specific reason labour balancing is a first-class concern under zone strategies and a second-order one under batch. In a batched tour every picker draws from the same pool of work, so imbalance self-corrects; in zone picking the pools are disjoint by construction and imbalance persists until somebody reassigns.

> Balancing should follow the wave, not the roster. Zone workload varies with what was ordered today, so an allocation fixed at shift start is balanced against yesterday's order mix.

4.4 What early detection buys, precisely

The product's material puts promise-date exceptions surfacing at wave time rather than at the dock, two to four hours earlier in the shift. The natural reading is that the answer arrives sooner. The more useful reading is that a different question is being asked.

the response set at time t is the actions whose duration still fits before the cutoff; it shrinks as the cutoff approaches; the value of early detection is the difference between the cheapest response available at wave time and at the dock (recovery)

A is the set of possible responses — substitute an item, split the shipment, expedite a replenishment, re-promise the customer, defer to the next wave — with d(a) the time each takes and c(a) its cost. t(cut) is the carrier cutoff.

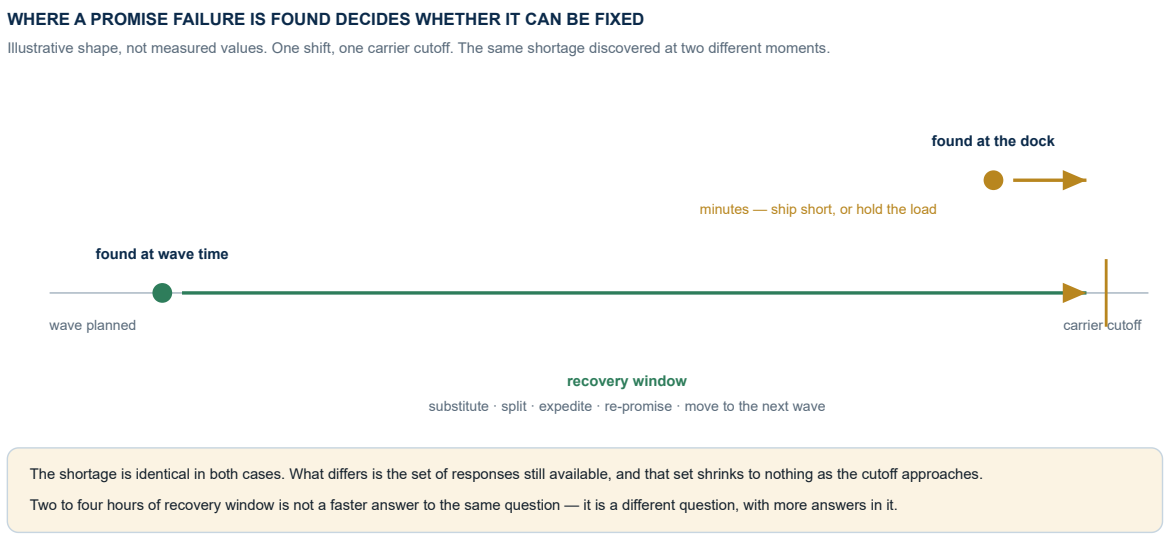

Figure 2. *Schematic.* Schematic (illustrative shape, not measured values): the same shortage found at two moments in a shift. Detected at wave time, five responses remain available. Detected at the dock, two remain — ship short or hold the load — and both are expensive. The value is the size of the response set, not the speed of the answer.

The formulation makes the value depend on the response set rather than on the hours, which is the right dependency. Two hours of warning in a facility with no substitution policy, no split-shipment capability and no ability to expedite is worth almost nothing; the same two hours in a facility with all three is worth the difference between them.

It also identifies what a buyer should ask for alongside the software. Early exception surfacing is a precondition for recovery, not recovery itself, and a system delivering exceptions into an organisation with no defined responses will produce two to four hours of earlier anxiety.

5 Modelled Scenario

No published outcome is attributed to this module. It is named in a third-party logistics deployment, but the four results that deployment reports belong to other modules, and this paper claims none of them. Everything below is a claim, a design statement, or a modelled scenario with its assumptions printed.

5.1 Claims published for this module, and what they lack

Table 2. Published figures for this module, with what each is missing.

Claim	Value	What it does not state
Order cycle time reduction	50% or better	Baseline, order profile, and what the cycle spans
Picking accuracy	99.9%	Baseline, and whether the denominator is lines, units or orders
Exception surfacing	2-4 hours earlier in the shift	The baseline process it is measured against

All three are product-page claims and this paper reports them as claims. The third is the most useful of them because Section 4.4 supplies a mechanism, but even there the value depends on the responses a facility can actually execute rather than on the hours.

The accuracy claim deserves a specific caution. Ninety-nine point nine per cent means something quite different across lines, units and orders: a facility at 99.9% of lines with five lines per order is at roughly 99.5% of orders, and the customer experiences orders. A figure quoted without its denominator can differ from the figure a customer feels by an order of magnitude in error rate.

> This is a general failing in the category rather than a criticism of one vendor. Ask any accuracy claim in this domain what its denominator is; the answer is rarely on the page.

5.2 The deployment this module appears in

A multi-client third-party logistics operator deployed six MileSoft warehouse modules across a distribution centre. The published approach describes this module closing the loop with fleet routing — wave plans flowing into route plans that re-optimise as orders mature.

> No result is attributed. The deployment's four published outcomes — dock idle time, truck dwell, inventory accuracy and client onboarding — all belong to other modules, and this paper converts none of them into a claim for this one.

What the account does establish is the coupling Section 3.1 treats as a wave constraint. A wave plan feeding a route plan means the outbound sequence is being produced against a downstream commitment rather than in isolation, which is the difference between staging that loads and staging that has to be sorted at the door.

5.3 Modelled pick, pack and ship labour

The published return model for this module prices the saving as labour recovered across pick, pack and ship. Its assumptions are printed here so a reader can substitute their own.

Table 3. Modelled scenario — not a deployment result. Assumptions are the published defaults of the return-on-investment model for this product.

Assumption	Value
Pick and pack staff in scope	12
Labour gain attributed to the module	15%
Where the gain comes from	orchestration of the work, not faster picking
Modelled recovery	the equivalent of 1.8 full-time staff

Fifteen per cent is an assumption, and Section 4.1 says where it would have to come from: travel amortised across better-formed waves. That is a real mechanism with a real ceiling — T(b)/b flattens — so the gain is largest at facilities currently picking single-order or waving badly, and smallest at facilities already batching sensibly.

It also cannot come from the picking itself. Nobody walks faster because the wave was planned better; they walk less far per order, which is a different and bounded quantity.

6 Discussion

6.1 One engine, several wave streams

Section 3.3 notes that business-to-business and consumer orders want opposite wave sizes. The general form of that observation is worth drawing out, because it applies to more than those two profiles.

Equation (batch) has a minimum whose position depends on how much latency an order can absorb. Orders that can absorb a lot should be batched heavily; orders that cannot should not. Mixing them into a common wave gives each the other's optimum and neither its own.

The right architecture is therefore one engine with several concurrent wave streams, partitioned by urgency class rather than by order type as such — because a rush wholesale order belongs with the consumer stream and a standing consumer subscription belongs with the wholesale one.

> Partition by what the order can tolerate, not by what channel it came from. Channel is a proxy for urgency and it is a poor one at exactly the edges where it costs most.

6.2 An exception is only worth what the response is

Section 4.4 formulates the value of early detection as the size of the remaining response set. That reframing has a practical consequence for how this class of software should be bought and deployed.

The software's contribution is the detection. The responses — substitution rules, split-shipment authority, expedite budgets, a re-promising policy that customer service is permitted to execute — are organisational, and most of them require someone to be allowed to make a decision without escalating.

A facility that deploys exception surfacing without those is measurably worse off than before in one respect: it now knows about failures two to four hours earlier and can do nothing about them, which consumes attention without producing outcomes.

The recommendation follows directly. Define the response set before the deployment, and size the expected benefit from the responses that will actually be available rather than from the detection alone.

6.3 A capability reference framework for fulfilment engines

Table 4. Capability reference framework. Each dimension separates a fulfilment engine from a pick-list printer, and each is answerable by demonstration against a live wave rather than by reading a specification.

Dimension	Question the system must answer by demonstration
D1 Wave size reasoned	What sets wave size — a target order cycle time, or a default?
D2 Strategy per wave	Do two waves in the same shift ever use different pick strategies?
D3 Deadline feasibility	Does the system refuse to release a wave that cannot finish before the cutoff?
D4 Makespan not labour	Does it report a wave's completion time, or only its total pick hours?
D5 Balance to the wave	Is zone staffing set from today's order mix, or from the roster?
D6 Urgency partition	Can a rush wholesale order join a fast wave, or is it batched by channel?
D7 Dock-aware staging	Is outbound staged in load sequence, or sorted at the door?
D8 Exception with response	When a promise failure surfaces, does the system offer the actions still available?

D3 is the one that separates planning from printing. A system that will happily release a wave which cannot finish in the time remaining is not scheduling; it is formatting, and the deadline is being managed by whoever notices.

6.4 Generalisability

The batch-latency trade of Section 4.1 generalises to every batching decision anywhere: laboratory sample runs, print jobs, payment settlement, kitchen order firing. In all of them the efficiency argument is made loudly and the latency cost is paid quietly by whoever is inside the batch.

The makespan result generalises to any partitioned workflow where the partitions cannot help each other, which is a good working definition of when to worry about balance at all.

What does not generalise are the product-page percentages. A 50% cycle-time reduction is a claim against an unnamed baseline, and Section 4.1 bounds what any wave-planning change can deliver — it is the travel term, and only the travel term.

7 Threats to Validity and Limitations

1. No outcome is attributed to this module. It is named in a deployment whose four published results belong to other modules, so this paper reports no field evidence and the absence should be read as informative.
2. The 50% cycle-time claim has no baseline. It states neither what the previous process was, nor what the cycle spans — order receipt to pick, to pack, to dispatch, or to delivery — and the four differ by hours.
3. The 99.9% accuracy claim has no denominator. Lines, units and orders give materially different figures, and the customer experiences orders.
4. The 2-4 hour figure is measured against an unstated baseline. Section 4.4 argues its value depends on the available response set rather than on the interval, and the interval alone cannot be converted into a benefit.
5. The travel model is stylised. T(b) is treated as growing sublinearly in wave size; the actual shape depends on layout, slotting and routing policy, all of which belong to other systems.
6. Strategy costs are compared at the level of terms, not calibrated. Equation (strategy) identifies which quantities dominate under which mix; it does not give numbers a specific facility can substitute without local measurement.
7. The wave plan depends on position being correct. A perfectly formed wave against a drifted position produces a well-sequenced failed pick, and that dependency is entirely outside this module.
8. No figure here is a product capture. This product's page carries no screenshots, so nothing in this paper demonstrates the described interface exists in the form modelled.

> The seventh limitation is the one to sequence a programme around. Fulfilment orchestration multiplies the value of accurate position and cannot substitute for it, so a facility with a drift problem should fix that first.

8 Future Work

- Publishing the wave-size curve. Measuring T(b) at a facility and showing where travel saving flattens against rising flow time would replace a default with a decision, and the data is in the pick history already.
- Makespan as the released figure. Reporting projected wave completion time rather than total pick hours would make Section 4.3 an instrument and would make the deadline feasibility check of D3 possible.
- Balance recomputed per wave. Zone staffing derived from today's order mix rather than the roster, with the imbalance cost reported so a supervisor can see what a reassignment is worth.

- Urgency-class partitioning in place of channel. Streaming waves by what an order can tolerate would give each profile its own optimum rather than the average of two.
- Exceptions delivered with their response set. Surfacing a promise failure alongside the actions still feasible before the cutoff turns a warning into a decision, which is what Section 4.4 says the value actually was.

9 Conclusion

A wave is a batch, and a batch is a commitment. Grouping orders is what makes picking efficient and what makes an order wait, so wave size is a trade with two computable sides rather than a setting inherited from the previous system.

Three results follow. The four pick strategies are four points on one trade rather than a maturity ladder — each moves travel into sortation, consolidation or latency — so the right choice follows from the order mix, which is why selecting per wave rather than per warehouse is the design decision that matters. Under zone picking a wave finishes when its slowest zone finishes, so imbalance costs completion time while leaving total labour untouched, and a deadline is met or missed on the maximum rather than the mean. And surfacing a promise failure at wave time rather than at the dock does not buy a faster answer; it buys a larger set of answers, because substitution, splitting, expediting and re-promising all need time the door no longer has.

That last result carries the strongest practical recommendation in the paper. Detection is the software's contribution and the responses are the organisation's, so a facility should define what it is permitted to do about an exception before it buys the ability to see one earlier.

The framework of Section 6.3 is offered as the durable contribution, and its third dimension separates the category in one question: will the system release a wave that cannot finish before the cutoff?

Appendix A Nomenclature

Table 5. Symbols and abbreviations used in this paper.

Symbol / term	Meaning
b	Wave size — the number of orders grouped into one wave
T(b)	Travel time of the tour serving a wave of b orders
s-bar, l-bar	Handling time per line, and mean lines per order
w-bar(b)	Work per order at wave size b
phi-bar(b)	Mean order flow time — wait for the wave plus the wave's duration
lambda	Order arrival rate
sigma, sigma(c)	Sortation cost per line under batch, and the lighter cost under cluster
gamma	Downstream consolidation cost per order under zone picking
L, L(z)	Total pick labour, and the labour in zone z
n(z)	Pickers assigned to zone z
M, M(ideal)	Wave makespan, and the makespan under perfect balance
A, R(t)	The set of possible responses, and those still feasible at time t
d(a), c(a)	Duration and cost of response a
t(cut)	The carrier cutoff
RMA	Return merchandise authorisation — the record a return generates

Appendix B Worked Numerical Examples

Appendix B.1 Choosing a wave size instead of inheriting one

A facility receives 90 orders an hour at 4 lines each. Handling is 0.4 minutes per line. Measured tour times are 22 minutes for 5 orders, 34 for 15, 44 for 30 and 58 for 60.

Applying Equation (batch): work per order is 22/5 + 1.6 = 6.0 minutes at b = 5; 34/15 + 1.6 = 3.9 at 15; 44/30 + 1.6 = 3.1 at 30; and 58/60 + 1.6 = 2.6 at 60.

Now the other side. Mean flow time is b/(2 x 90) hours plus the wave duration. At b = 15 that is 5.0 minutes of fill plus 34 of tour, so 39 minutes. At b = 60 it is 20 minutes of fill plus 58, so 78.

Going from 15 to 60 orders per wave saves 1.3 minutes of work per order and doubles order flow time. Going from 5 to 15 saves 2.1 minutes per order and adds only 12 minutes of flow. The first move is clearly worth making and the second usually is not — and a facility waving at 60 because that is what its previous system did is paying 39 extra minutes of latency for a third of the saving it already banked.

> Every input here comes from the facility's own pick history. The curve can be plotted before any software is purchased and it will settle the wave-size argument better than any vendor benchmark.

Appendix B.2 Total labour fits; the wave still misses

A zone-picked wave carries 620 minutes of pick work across four zones: 300 in zone A, 140 in B, 110 in C and 70 in D. Twelve pickers are on shift, allocated three per zone by the roster. Consolidation adds 25 minutes. There are 140 minutes to the cutoff.

Applying Equation (makespan): M = max(300/3, 140/3, 110/3, 70/3) + 25 = 100 + 25 = 125 minutes. That fits, with 15 minutes to spare.

Now suppose the order mix shifts and zone A carries 420 minutes with the total unchanged at 620. Total labour still fits comfortably — 620 minutes across 12 pickers is 52 minutes each — but M = 420/3 + 25 = 165, and the wave misses by 25 minutes while pickers in zones C and D finish early and stand idle.

Rebalancing to the actual mix — six pickers in A, three in B, two in C, one in D — gives M = max(70, 46.7, 55, 70) + 25 = 95. The same twelve people, the same total work, forty minutes inside the deadline instead of twenty-five outside it.

Appendix B.3 What two hours is worth, and when it is worth nothing

A shortage is detected on a 40-unit line. Available responses and their durations: substitute an equivalent item (20 minutes, cost 15), expedite a replenishment from reserve (75 minutes, cost 40), split the shipment (30 minutes, cost 120), re-promise the customer (10 minutes, cost 200), ship short (0 minutes, cost 480).

Detected at wave time with 150 minutes to the cutoff, R contains all five and the cheapest is substitution at 15. Detected at the dock with 12 minutes remaining, R contains only re-promising and shipping short, and the cheapest is 200.

Applying Equation (recovery), V = 200 — 15 = 185 per exception. At 30 such exceptions a week that is 5,550 a week, and none of it comes from working faster.

Now remove the responses. A facility with no substitution policy and no authority to split loses the two cheapest options; its best action at wave time is expediting at 40, and V falls to 160. A facility that can only re-promise or ship short has V = 0 — the exception arrives two hours earlier and changes nothing, which is the case Section 6.2 warns about.

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
Pick strategies selected per wave from the order mix	single-order, batch, cluster, zone	**Product specification** /products/order-fulfillment
How much earlier a promise-date exception surfaces at wave time than at the dock	2-4 hours earlier in the shift	**Product specification** /products/order-fulfillment
Order cycle time reduction claimed on the product page	50% or better	**Product specification** /products/order-fulfillment — A product-page benefit claim with no baseline, no order profile and no definition of what the cycle spans.
Picking accuracy claimed on the product page	99.9%	**Product specification** /products/order-fulfillment — A product-page benefit claim without a baseline or a stated denominator - lines, units or orders.
Pick, pack and ship labour recovered	15% across 12 staff (modelled)	**Modelled estimate** 12 pick and pack staff in scope; 15% labour gain attributed to the module; Model and defaults published in src/data/roiModels.ts

References

[1] 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.

[2] Petersen, C. G. (1997). An evaluation of order picking routeing policies. International Journal of Operations & Production Management, 17(11), 1098-1111.

[3] Roodbergen, K. J., and de Koster, R. (2001). Routing methods for warehouses with multiple cross aisles. International Journal of Production Research, 39(9), 1865-1883.

[4] 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.

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

[6] 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

[7] 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

[8] F. W. Harris (1913). How Many Parts to Make at Once. Factory, The Magazine of Management, vol. 10, no. 2, pp. 135-136, 152; the first statement of the economic order quantity, reprinted in Operations Research 38(6), 1990, pp. 947-950. https://pubsonline.informs.org/doi/10.1287/opre.38.6.947

[9] H. F. Dickie (1951). ABC Inventory Analysis Shoots for Dollars, Not Pennies. Factory Management and Maintenance, vol. 109, no. 7, July 1951, pp. 92-94; the origin of ABC classification, applying Pareto and Lorenz to stock value.

[10] 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

[11] 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

[12] 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

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

[14] GS1 (2023). GS1 Logistic Label Guideline. GS1 AISBL; the SSCC is the only mandatory field on a GS1 logistic label. https://www.gs1.org/standards/gs1-logistic-label-guideline/1-3

[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.