SELLERFLOW RESEARCH SERIES · WHITE PAPER 001

Decision Intelligence for Amazon Marketplace

From price optimization to decision optimization

SIGNAL — INTERPRETATION — CONSTRAINT — DECISION — PRICE

Elias Sultan
Founder of SellerFlow
Marketplace Decision Systems Research
Valencia, Spain

Version 1.0
July 2026
eliassultan.pro

ABOUT THIS PUBLICATION

Title
Decision Intelligence for Amazon Marketplace: From Price Optimization to Decision Optimization

Author
Elias Sultan — Founder of SellerFlow, Marketplace Decision Systems Research, Valencia, Spain

Series
SellerFlow Research Series · White Paper 001 · Version 1.0 · July 2026

Scope

This paper sets out a framework for treating marketplace pricing as the output of a commercial decision rather than as the decision itself. It describes the limitations observed in conventional rule-based repricing, defines the variables a decision layer must evaluate, presents the architecture that implements that layer in SellerFlow, and documents the research method used to develop and test it.

Intended audience

Amazon Marketplace professionals; ecommerce technology companies; pricing specialists and marketplace consultants; software developers building commercial decision systems; and researchers working on pricing and decision-support methodologies.

Disclaimer

This white paper presents the author's research perspective and the methodology implemented within SellerFlow. The discussion reflects applied observations and documented development work carried out on a live marketplace operation. Marketplace behaviour is influenced by many factors outside the control of any individual seller or software platform, and outcomes vary with the specific conditions of each business. Nothing in this document should be read as a performance guarantee.

Amazon, the Amazon Marketplace and the Buy Box are the property of their respective owners. This publication is independent and is not affiliated with, endorsed by or sponsored by Amazon.

Citation

> Sultan, E. (2026). *Decision Intelligence for Amazon Marketplace: From Price Optimization to Decision Optimization.* SellerFlow Research Series, White Paper 001, Version 1.0.

CONTENTS

SECTION 01

Executive summary

Amazon Marketplace is one of the most complex pricing environments in commerce. This paper argues that the difficulty is not primarily a pricing problem, and that treating it as one is the reason so much marketplace automation destroys margin while working exactly as designed.

Sellers on Amazon do not compete on price alone. They compete inside a system shaped by the Buy Box allocation, by automated competitors reacting to one another, by the marketplace's own retail arm, by inventory position, by fulfilment channel and delivery promise, and by a cost structure that is only fully known after settlement. A price set without reference to those conditions is a number, not a decision.

Conventional repricing software optimises numerical variables — profit, ROI, margin, markup, minimum and maximum price — and emits a price. The variables are sound and the execution is reliable. The limitation is structural: the system's objective is the price. It has no representation of what the contested position is worth, of what the competitor is likely to do next, of whether the inventory supports the strategy, or of what the business is trying to achieve this month. Those judgements remain with the operator, and at catalogue scale the operator cannot make them one product at a time.

This paper proposes an alternative structure. Before a price is calculated, the system determines the appropriate commercial strategy under current conditions. Price then follows from the strategy. We call this layer **Decision Intelligence**: a stage that consumes market signal, interprets it as behaviour, applies commercial constraint, and produces a decision — of which the price is only the visible output.

> CENTRAL PROPOSITION
>
> A repricer optimises the price. A decision system decides whether the price should move at all, in which direction, by how much, for how long, and what should happen if the market does not respond as expected. The price is the consequence of that decision, not its objective.

The framework is not theoretical. It was developed through several years of continuous observation of a live Amazon operation in Spain, and is implemented in SellerFlow, a platform built by the author to run that operation. Every element described here was adopted because a rule-based approach had failed in a way that could be measured in settled margin.

Sections 3 and 4 establish what conventional repricing does and where it breaks. Section 5 defines the decision framework and its five variables. Section 6 gives the reference architecture. Section 7 describes the research method, including the decision to treat the buyer-visible marketplace as ground truth rather than relying on reported data. Section 8 presents three case studies. Sections 9 to 11 cover differentiation, limitations and open problems.

SECTION 02

Introduction

Marketplace selling has a deceptively simple surface. A product has a page, several sellers offer it, one of those offers is presented to the buyer by default, and most of the demand flows through that offer. Everything that makes the environment difficult sits underneath: how that default offer is chosen, how quickly the other sellers react to a change, what each of them can afford, and what the marketplace itself is doing on the same listing.

The seller's response to this complexity has been automation. Repricing software is now standard equipment, and it works: it observes competing offers, applies a rule, and updates prices faster and more consistently than any human operator could. The practical consequence is that a large share of the visible price movement on a marketplace listing is machines responding to machines.

That is where the problem begins. Two automated systems facing each other do not converge on a price that reflects the value of the product or the needs of either business. They converge on the lower of the two floors that their operators configured, because each system's only instruction is to be positioned relative to the other. The outcome is determined by the configuration limits, not by any commercial judgement, and the seller who automates most aggressively is frequently not the one who ends up better off.

Automation removed the operator from the loop without putting a decision in their place.

The research presented here started from that observation, made while operating rather than while studying. The initial question was narrow: is the price movement of a competitor predictable enough to be treated as behaviour rather than noise? The answer turned out to be yes for most competitors most of the time, and once competitors are modelled as actors with characteristics, a series of further questions follow. What is the contested position actually worth on this product? What does the unit actually cost, after settlement rather than in estimate? Under what conditions is the correct action to do nothing?

Answering those questions required instrumentation that did not exist in the tooling available, and the instrumentation eventually became a platform. This paper documents the framework that emerged and the architecture that implements it.

Terminology

Throughout this document, **featured offer** and **Buy Box** refer to the offer presented by default on a product page. **Position** refers to a seller's standing among the competing offers on a listing. **Floor** refers to the lowest price a seller can accept for a unit given its true cost. **Reaction interval** refers to the elapsed time between a change in the market and a given competitor's response to it. **Decision** refers to the selection of a commercial strategy, of which a price change may or may not be a component.

SECTION 03

The evolution of marketplace pricing

It is useful to separate the development of marketplace pricing into three stages, because each one solved the previous stage's problem and created the next.

3.1 Manual pricing

In the first stage the seller sets prices by hand, reviewing listings periodically. The approach scales poorly and reacts slowly, but it has one property that later stages lose: every price carries a judgement. The operator knows why a particular product is priced where it is, what they are willing to give up on it, and when to walk away from a contest.

3.2 Rule-based repricing

The second stage automates the reaction. The seller configures a rule — typically some form of "undercut the lowest competing offer by a fixed amount, within a minimum and maximum" — and the system applies it continuously. This solves the scale and latency problems completely.

What it does not carry forward is the judgement. The rule is a static object: it does not know which competitor it is facing, whether that competitor is about to run out of stock, whether the position is worth the discount required to take it, or whether the business needs margin or cash this quarter. Every product in the catalogue is treated with the same logic.

Figure 1. The rule-based loop. The system is closed: an observed price produces a new price, which is observed by the competing system, which produces another. No stage of the loop contains a commercial judgement.

3.3 What the third stage has to add

The third stage does not replace the second. Execution still has to be fast, continuous and reliable, and the numerical variables of a conventional repricer remain necessary. What the third stage adds is a layer above execution that decides what the execution should be trying to achieve on this product, against this competitor, under these conditions, with this inventory, for this business.

The remainder of this paper is concerned with that layer.

SECTION 04

The limitations of rule-based repricing

The limitations below are not defects of implementation. They follow from the structure of a system whose objective is the price. Each was observed directly in operation before it was addressed in software.

4.1 The competitor is anonymous

A rule reacts to the lowest competing price. It does not distinguish between a seller with three units left and a seller with three hundred, between one that defends a position and one that abandons it under pressure, between a professional operation running its own automation and an occasional seller clearing stock, or between an independent seller and the marketplace's own retail arm. These are entirely different commercial situations, and they call for different strategies. The rule applies one strategy to all of them.

4.2 The cost floor is assumed rather than known

Minimum prices are usually configured from estimated fees. Estimates and settlements diverge: fulfilment fees vary with measured dimensions and weight bands, referral fees vary by category, returns and their handling are charged after the fact, storage accrues, and promotional participation changes the net. A floor built on estimates is therefore approximately correct on average and materially wrong on specific products — and it is precisely on the specific products that a repricer will spend the difference.

> OBSERVED CONSEQUENCE
>
> When settled transaction data is reconciled per unit and compared with the estimates that were used to configure minimum prices, a portion of a catalogue is typically found to have been trading below its true floor — profitably according to the tool, unprofitably according to the settlement.

4.3 The value of the position is not modelled

Holding the featured offer is treated as the objective, uniformly. In practice its value varies substantially between products: on some listings it carries nearly all the demand, on others the difference between holding it and not holding it is modest. A system that pays the same discount for the position everywhere systematically overpays on the products where the position matters least.

4.4 There is no representation of doing nothing

Rule-based systems act whenever their condition is met. But a substantial share of correct commercial decisions are decisions to wait: the competitor is about to exhaust stock, the contest is structurally unwinnable, the discount required exceeds what the position is worth, or the market is moving for a reason that will resolve on its own. Without an explicit hold state, a system over-trades, and each unnecessary move invites a reaction that costs more than the move gained.

4.5 The system cannot retreat on evidence

When a price change fails to produce the intended result, a rule-based system has no memory of the failure. It will repeat the same move under the same conditions. What is needed instead is a return to the last price at which the intended outcome was actually observed, followed by a controlled attempt to recover from there — and a record ensuring that a move already known to fail is not attempted again.

4.6 Margin and cash are optimised separately, or not at all

A system tuned for margin alone will hold prices high, sell less, and quietly degrade cash conversion — which matters more than margin to an inventory-financed business. A system tuned for volume alone will erode margin until there is nothing to finance the inventory with. The decision has to hold both objectives simultaneously and be told which one the business currently needs. A rule cannot express this, because it is a property of the business, not of the product.

Table 1. What a conventional repricer optimises, and what remains outside its scope.

Handled by rule-based repricing	Not represented
Profit, ROI, margin and markup targets	Whether the contested position is worth its cost on this product
Minimum and maximum price limits	Whether the floor reflects settled costs or estimates
Continuous reaction to competing offers	Who the competitor is and how they behave
Fast, reliable price execution	Whether any action should be taken at all
Per-product configuration	The objective of the business this month
Consistent application of the rule	Evidence of what happened last time the rule was applied

SECTION 05

Decision Intelligence: the framework

The framework replaces a single step — observe, then price — with four, and places the price at the end where it belongs.

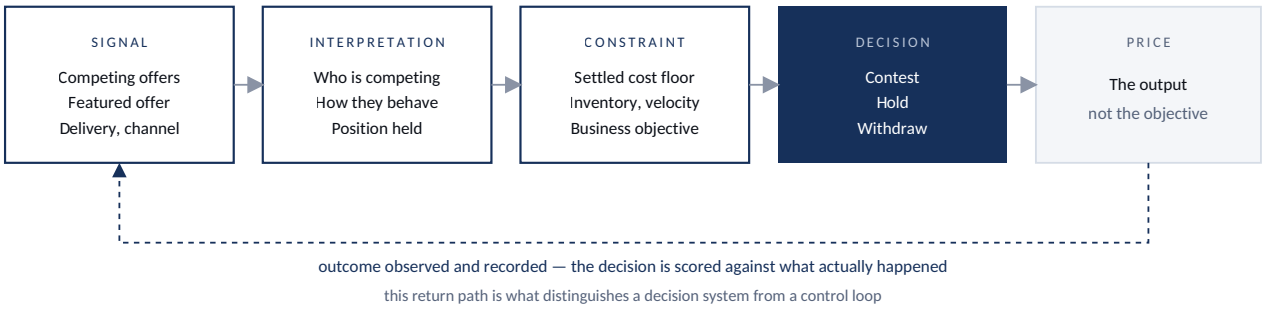

Figure 2. The decision pipeline. Signal is interpreted as behaviour before constraint is applied; the decision may or may not include a price change; the outcome is recorded and informs the next decision on the same product.

5.1 Signal

The first stage captures the market as the buyer sees it: which offers exist on the listing, at what price, from which seller, with what delivery promise, from which fulfilment channel, and which of them holds the featured position. This is deliberately buyer-visible data. A pricing decision taken on a representation of the market that no buyer ever sees is a decision about a different market.

5.2 Interpretation

The second stage converts observation into behaviour. Competitors are identified across time, so that a price is attributed to an actor rather than to the listing. Each actor accumulates characteristics: the interval within which it typically reacts, how far it follows a price down, where it stops, whether it holds a position under pressure, and whether it appears to be running out of stock. The seller's own position — held, lost, or contested — is established from the same observation rather than inferred from the last action taken.

> DESIGN RULE
>
> The system may not assume it holds a position because it lowered a price. It must observe that it holds the position. Actions produce intentions; only observation produces state.

5.3 Constraint

The third stage introduces everything the market cannot tell you: what the unit actually cost after settlement, how much inventory exists and how fast it is moving, what the product is expected to contribute, and what the business currently needs. Constraint is where a commercially valid decision separates from a merely competitive one.

5.4 Decision

The fourth stage selects a strategy. The important property of this stage is that it has more than one available output:

- **Contest.** Take or defend the position, by an amount and at a pace chosen from the competitor's known behaviour rather than by a fixed increment.
- **Hold.** Take no action. The correct choice when the competitor is likely to move first, when the required discount exceeds the value of the position, or when the current state is already the best available.
- **Withdraw.** Decline the contest and retain margin. The correct choice when the situation is structurally unwinnable, and a decision a rule-based system cannot express.
- **Recover.** Return towards a price previously observed to be viable, gradually, verifying at each step that the position is still held.

5.5 Price

Only at this point is a number produced. Its magnitude comes from the strategy and from what is known about the specific competitor; its permissible range comes from the settled cost floor; its timing comes from the competitor's reaction interval. The same market condition can legitimately produce different prices on two products, because the decision behind them is different.

Five stages, one rule: no stage may be skipped, and the price is never the first thing computed.

SECTION 06

Reference architecture

SellerFlow implements the framework in four layers. The separation matters: each layer can fail, degrade or be replaced without the layer above it producing a wrong decision, provided the contract between them is that state must be observed rather than assumed.

Figure 3. Layered reference architecture. Observation and economics feed the decision layer; execution acts and reports back. No layer above may assume the state of the layer below.

6.1 Observation layer

The observation layer captures competing offers continuously across the catalogue. Three properties proved necessary in practice.

Buyer-visible capture. The offers are read as they are presented on the product page, including the delivery promise and fulfilment channel of each. Reported or aggregated pricing feeds omit exactly the attributes that determine whether a position can be held.

Fair scheduling. Refresh capacity is finite. Without an explicit ceiling, a small number of high-velocity products consume the available capacity and the rest of the catalogue goes stale. A cap on how frequently any single product may be refreshed converts wasted capacity into coverage.

Health monitoring and self-recovery. Any continuous capture process degrades: sessions expire, pages fail to render, individual products become temporarily unreadable. The layer maintains a health state per product and releases products from a failed state automatically after a bounded interval, so that a transient failure does not remove a product from the decision process for hours.

6.2 Economics layer

The economics layer establishes what a unit actually earns. It reconciles settled marketplace transactions against the products they belong to, attaches purchase cost, shipping cost by carrier and destination, and returns. Its output is a floor per unit that reflects

charges rather than estimates, and the same figures feed the accounting record — so that the decision layer and the books are working from one set of numbers rather than two.

6.3 Decision layer

The decision layer holds the framework of Section 5. It maintains competitor profiles, evaluates the current position from observation, selects a strategy, and produces a price only as a consequence. Two implementation properties are worth stating explicitly, because both were adopted after failures:

- **Explainability.** Every decision records the reason it was taken. At catalogue scale an operator cannot verify individual prices, but they can audit reasons — and a system whose reasoning cannot be inspected cannot be trusted with a catalogue.
- **Evidence-based recovery.** Where a decision must be reversed, the system returns to a price at which the desired outcome was actually observed, and recovers from there in controlled steps rather than jumping to a target it has no evidence for.

6.4 Execution layer

Execution submits prices, fulfils orders, books shipments across carriers and tracks them. Its relevance to this paper is limited but not zero: execution is where a decision meets the real constraints of an operation, and delays or failures there are themselves signals the decision layer must account for.

6.5 Resilience of the observation source

A decision system that depends on a single view of the market inherits that view's fragility. The architecture therefore separates the *decision logic* from the *source of market data*, and supports switching the source without changing the logic. The two sources differ in fidelity — the buyer-visible capture is richer, the programmatic interface is more stable — and the system records which source produced each observation, so that the decision layer knows how much of the picture it currently has.

SECTION 07

Research methodology

The research is observational and continuous, conducted on a live Amazon operation in Spain rather than in simulation. That choice defines both the method's limitations and its value.

7.1 Ground truth

The canonical record of the market is what a buyer sees on the product page. Where a programmatic data source and the buyer-visible page disagree, the page is treated as correct and the discrepancy is logged. This principle resolved a recurring class of error in which decisions were taken against a representation of the market that was internally consistent but did not correspond to what any customer was being shown.

7.2 The unit of observation

The unit is not the listing price but the *offer*: seller identity, price, delivery promise, fulfilment channel, and whether it holds the featured position, timestamped. A listing observed repeatedly yields a sequence of offers per seller, which is what makes behavioural characterisation possible.

7.3 Hypothesis and test

Hypotheses are formulated from operational anomalies — a product losing a position it should hold, a category with margin worse than modelled, a competitor behaving inconsistently with its profile. Each is tested by applying a change to a bounded part of the catalogue and measuring the outcome against the unchanged remainder, over a period long enough to survive weekly seasonality.

7.4 What counts as a result

Only outcomes that can be settled: units sold, margin realised after settled fees, position held or lost as observed, and cash converted. Internal indicators — that a service is running, that a rule fired, that a price was submitted — are explicitly not results. This distinction is not pedantic. A system can report healthy operation while producing nothing of value, and the discipline of measuring flow rather than status has repeatedly been the difference between a diagnosis and a false conclusion.

> METHODOLOGICAL RULE
>
> Audit real flow, not status reports. That a process is active does not establish that it is producing anything; verify the arrival of fresh observations and settled outcomes.

7.5 Documentation

Each significant change is recorded with the date, the problem observed, the change applied, the module affected, and the measured or expected effect. The record is contempor-

aneous, which allows the development of the system to be reconstructed from evidence rather than from memory.

7.6 Limitations of the method

Three limitations should be stated plainly. First, the operation is a single business in a single national marketplace, so findings may not transfer uniformly to other categories or countries. Second, the market is not controlled: competitors enter and leave, and the marketplace changes its own behaviour, so isolation of a single variable is imperfect. Third, observation is sampled rather than continuous at every instant, which bounds the resolution at which reaction intervals can be measured. Each of these argues for reporting findings as robust patterns rather than as precise constants.

SECTION 08

Case studies

Three studies are summarised here. Each began as an operational problem, was investigated with the method of Section 7, and resulted in a change to the decision layer.

> NOTE ON QUANTITATIVE RESULTS
>
> Findings are reported here as directional patterns. The underlying measurements are drawn from the operational record of a single business, and per-product magnitudes vary with category, competitor and period; publishing them as constants would overstate their generality.

CASE STUDY 01

Reaction interval and the cost of exposure

Problem. A price reduction taken to win a position was matched by a competitor, leaving both sellers lower and the position unchanged. Repeated across a catalogue, the pattern transfers margin to buyers without changing the distribution of sales.

Investigation. Rather than treating the match as instantaneous, the interval between the seller's change and the competitor's response was measured per competitor over an extended period.

Finding. Reaction is not instantaneous, and the interval is consistent enough per competitor to be treated as a characteristic rather than as noise. Between the action and the response there is a window during which the seller holds the position at a price the competitor has not yet matched.

Application. The decision layer treats that window as a resource. The size and timing of a move are chosen from the specific competitor's known interval, and the objective becomes holding the position for the majority of the time rather than permanently occupying the lowest price. A move justified only by a competitor's slow response is also, by construction, a move that can be reversed when the response arrives.

CASE STUDY 02

Settled fees against estimated fees

Problem. Catalogue profitability reported by conventional tooling did not match the settlement. Products modelled as profitable were, after settlement, not.

Investigation. Settled marketplace transactions were reconciled to the unit level and compared against the estimated fees that had been used to configure minimum prices, product by product, together with purchase cost, shipping by carrier and destination, and returns.

Finding. The divergence between estimate and settlement is not uniform. It concentrates in particular dimension and weight bands, in products with above-average return rates, and in promotional participation. A floor configured from estimates is therefore approximately right on average and materially wrong exactly where it matters.

Application. The settled figures became the cost basis for all pricing limits, and the same reconciliation feeds the accounting record. The immediate operational consequence was a set of products whose minimum prices had permitted trading below their true floor, corrected once the real figures were attached.

CASE STUDY 03

The structurally unwinnable contest

Problem. On certain listings the system pursued a position it could not obtain, descending towards its floor against a competitor with a different cost structure and different objectives — frequently the marketplace's own retail arm.

Investigation. Contests were classified by the identity and behaviour of the opposing seller, and the outcome of pursuit was measured: what was conceded in price, and whether the position changed hands as a result.

Finding. Some contests cannot be won at any price the seller can afford, because the opposing party is not solving the same problem. Pursuing them transfers margin without changing the outcome. The correct action is to withdraw and retain margin — but a rule-based system has no way to express withdrawal, so it descends by default.

Application. The decision layer classifies the competitive situation before selecting a strategy, and withdrawal became an explicit, recorded decision rather than a failure state. Related to this, the system was prohibited from raising its price on the assumption that it holds a position: it may only recover upwards when observation confirms the position is held.

SECTION 09

Technical differentiation

The comparison that follows is between *approaches*, not between named products. Commercial repricing tools differ in their configuration surfaces and roadmaps, and several offer features that partially address the points below. The distinction being drawn is structural: whether the price is the system's objective or its output.

Table 2. Structural comparison of the two approaches.

Dimension	Rule-based repricing	Decision Intelligence
Objective	Produce a price	Select a commercial strategy; price follows
Competitor	An anonymous lowest price	An identified actor with measured behaviour
Cost floor	Configured from estimates	Derived from settled transactions per unit
Position	Assumed from the action taken	Established by observation before acting
Available outputs	Change the price	Contest, hold, withdraw or recover
Timing	As fast as possible	Chosen from the competitor's reaction interval
Failure handling	Repeats the rule	Returns to an evidenced price and recovers gradually
Business objective	Not represented	Explicit input: margin, volume or cash
Auditability	The rule that fired	The reason the decision was taken

9.1 Why observation fidelity is the hard part

Most of the framework is inexpensive to state and expensive to run. Interpretation requires a continuous, catalogue-wide, buyer-faithful record of offers including delivery promise and fulfilment channel — and that record has to survive session expiry, rendering failures, rate limits and partial outages without silently degrading into a stale picture. In practice, the majority of the engineering effort in a decision system of this kind is spent on keeping the observation layer honest, not on the decision logic itself.

9.2 Why explainability is a commercial requirement

An operator who cannot understand why a system acted will eventually override it, and a system that is routinely overridden provides no benefit over manual pricing. Recording the reason for each decision is therefore not a reporting feature but a precondition for the system being used as designed.

SECTION 10

Limitations and open problems

The framework has been tested in one operation, in one marketplace, over a period of years. Several limitations follow, and are stated here rather than in a footnote because they define where the work goes next.

10.1 Generalisation

Category structure varies substantially. Products with few competing offers and high position value behave differently from commodity listings with many sellers. The framework is expected to transfer; the specific parameters are not.

10.2 Competitor identity is not always stable

Behavioural characterisation depends on identifying a competitor consistently over time. Sellers change names, operate multiple accounts, or intermittently disappear from a listing. Profiles must therefore degrade gracefully when identity is uncertain rather than assume continuity.

10.3 The observed market is sampled

Capture is periodic. Events between observations are invisible, which places a resolution limit on measured reaction intervals and can hide short-lived offers entirely. Increasing frequency raises cost and risk without proportionate gain, so the practical answer is to allocate observation capacity by value rather than uniformly.

10.4 Objectives are set by a human

The system is told whether the business needs margin, volume or cash. It does not derive that objective, and should not: it is a judgement about financing, seasonality and strategy that belongs to the operator. A decision system that silently chose its own objective would be harder to trust, not easier.

10.5 The marketplace is not a passive environment

Allocation rules, fee schedules and presentation change without notice. Any model of the environment is provisional, which is the strongest argument for building the system around continuous observation rather than around a fixed model of how the marketplace works.

SECTION 11

Future research

Four directions follow directly from the material presented here.

11.1 Predictive competitor classification

Current profiles are descriptive: they record how a competitor has behaved. The next step is predictive — estimating, from a competitor's recent behaviour, the probability that it will match a given move within a given interval, and using that estimate to price the move before making it.

11.2 Position value estimation per product

The value of the featured position varies by product and is currently estimated coarsely. A per-product estimate of what the position is worth, derived from observed demand with and without it, would allow the system to price the contest itself rather than the product.

11.3 Objective balancing

Margin, volume and cash conversion are currently balanced by an operator-supplied objective. A formal treatment — in which the system proposes a catalogue mix that satisfies a cash constraint at maximum margin — is a natural extension, and the more valuable one for inventory-financed businesses.

11.4 Cross-marketplace generalisation

Extending the observation and economics layers across national marketplaces raises questions that do not exist in a single market: shared inventory, differing fee structures and cost floors, and competitors that are present in one market and not another.

SECTION 12

Conclusion

The argument of this paper is narrow and, we would suggest, consequential. Marketplace pricing has been automated as an execution problem, and executed well. What was not carried across from manual operation was the judgement that used to sit behind each price: what the position is worth, who is on the other side of it, what the unit really costs, and what the business needs right now.

Reinstating that judgement is not a matter of better rules. It requires a distinct layer, with its own inputs — behaviour rather than prices, settled costs rather than estimates, observed position rather than assumed position — and with more than one available output, including the option to do nothing and the option to withdraw.

The practical claim is modest and testable: a system that decides before it prices will act less often, concede less margin in contests it cannot win, and be able to explain itself to the operator who is accountable for the result. SellerFlow is one implementation of that structure, developed against a live catalogue where the alternative to being right is measurable.

Price optimisation asks what number to publish. Decision optimisation asks whether to publish one at all.

Subsequent papers in this series will treat the individual components in depth: behavioural characterisation of automated competitors, probability modelling of the featured position, fee reconciliation methodology, and the architecture of the observation layer.

SECTION 13

References and further reading

This paper is based on primary observation rather than on secondary literature. The references below are the author's own research entries and the public documentation of the marketplace mechanisms discussed.

Documents in the SellerFlow Research Series carry a permanent reference of the form **WP-nnn** (white papers), **RP-nnn** (research papers), **EP-nnn** (engineering papers), **TN-nnn** (technical notes) and **CS-nnn** (case studies). This paper is **WP-001**.

SellerFlow Research Series

1. Sultan, E. (2024–). *Behavioural Analysis of Automated Repricers.* Research entry RP-001, SellerFlow Research Series.
2. Sultan, E. (2025–). *Buy Box Probability Modelling.* Research entry RP-002, SellerFlow Research Series.
3. Sultan, E. (2025–). *Amazon Retail Competition Strategies.* Research entry RP-003, SellerFlow Research Series.
4. Sultan, E. (2025–). *Amazon Fee Intelligence.* Research entry RP-004, SellerFlow Research Series.
5. Sultan, E. (2026–). *Decision Intelligence Framework.* Research entry RP-005, SellerFlow Research Series.

Primary sources

1. Amazon Services LLC. *Selling Partner API documentation.* Product pricing, orders, and finances references.
2. Amazon Services Europe. *Fulfilment by Amazon fee schedules and referral fee schedules*, Spain marketplace.
3. Amazon Seller Central. *Featured offer (Buy Box) eligibility and requirements.*

Operational record

1. SellerFlow development log, 2024–2026. Contemporaneous record of problems observed, changes applied, modules affected and measured effects.

Research entries are published at eliassultan.pro/research. Where an entry is cited as ongoing, its findings are subject to revision as observation continues.

SECTION 14

Appendix A — Decision variables

The variables the decision layer evaluates, with their source and the reason each is required. A system missing any one of them will produce commercially valid decisions only by coincidence.

Table 3. Decision variables, sources and function.

Variable	Source	Function in the decision
Competing offers	Observation layer, buyer-visible	Establishes the field: who is present and at what price
Featured position	Observation layer	Establishes actual state; may not be inferred from own actions
Delivery promise	Observation layer	Determines the price at which a position can be held
Fulfilment channel	Observation layer	Distinguishes structurally different competing offers
Competitor identity	Observation layer, over time	Attributes behaviour to an actor rather than to the listing
Reaction interval	Behavioural profile	Sets the timing and reversibility of a move
Competitor depth	Behavioural profile	How far the competitor will follow before stopping
Settled cost floor	Economics layer	Bounds the permissible range of any decision
Inventory position	Operations	Determines whether a strategy can be sustained
Sales velocity	Operations	Weights the value of the contested position
Business objective	Operator	Selects between margin, volume and cash conversion
Decision history	Decision layer	Prevents repetition of moves already known to fail

A.1 The four decision outputs

Table 4. Available outputs and the conditions that select them.

Output	Selected when
Contest	The position is worth more than the discount required, the floor permits it, and the competitor's profile suggests the move can be held long enough to matter
Hold	The current state is already the best available, the competitor is likely to move first, or the required concession exceeds the value of the position
Withdraw	The contest is structurally unwinnable, or winning it would require trading below the settled floor
Recover	The position is confirmed held and evidence exists that a higher price was previously viable; recovery proceeds in controlled steps with verification at each

SECTION 15

About the author

Elias Sultan is an ecommerce operator, researcher and software developer working on marketplace decision systems. He trades on the Amazon marketplace in Spain and builds the technology that runs that operation, which places him on both sides of the problem he studies: the person making the commercial decision, and the person writing the system that makes it.

His research addresses the layer between market signal and commercial action — how competitive behaviour, featured-offer conditions, delivery promise, real cost structure, inventory position and business objectives combine into a single decision. That work is implemented in SellerFlow, a decision platform he designed and built, which operates a live catalogue and serves as the environment in which each hypothesis is tested against settled commercial outcomes.

Research: eliassultan.pro · Platform: sellerflow.pro · Contact: contact@eliassultan.pro

SELLERFLOW RESEARCH SERIES

This is White Paper 001 of an ongoing series documenting applied research on marketplace decision systems. Subsequent papers treat behavioural characterisation of automated competitors, probability modelling of the featured offer, fee reconciliation methodology, and the architecture of the observation layer. Published papers are available at eliassultan.pro/white-papers.

Sultan, E. (2026). *Decision Intelligence for Amazon Marketplace: From Price Optimization to Decision Optimization.*
SellerFlow Research Series, White Paper 001, Version 1.0. July 2026.