

TECH & AI CASE STUDY - AIR CARGO - 2025

Quantum-Inspired Optimization That Cut Gate-Reassignment Time 35% in an Honest Pilot

Based on a delivery engagement with SkyLift Cargo

-35%

REASSIGNMENT DECISION TIME

-18%

GROUND-TIME OVERRUNS

-22%

OPS-DESK OVERTIME

2 of 5

DISRUPTION CLASSES AUTOMATED

A fillable practice edition: the full story, 8 actions,
6 reflections and a 90-day plan - work it with your team.

Free - no email required
emersontechnology.com

MAKE THIS PRACTICE BOOK YOURS

Work it, do not just read it

This practice book turns the SkyLift Cargo case study into a working session for your team. Read the full story first - what happened, what was built and what it changed. Then tick the 8 actions you can commit to, write short answers to the 6 reflections, and finish with a 90-day plan. Every field in this PDF is fillable: open it in Adobe Acrobat Reader, Chrome, Edge or macOS Preview, type directly into the boxes and save.

YOUR NAME	ROLE
COMPANY	TEAM / DEPARTMENT
DATE	EMAIL (OPTIONAL)

MY GOAL FOR THIS PRACTICE SESSION

- 01 The story** - What happened at SkyLift Cargo - before, during and after the project.
- 02 The results** - The numbers that changed, plus industry benchmarks from public research.
- 03 The 8-action checklist** - Tick boxes as your organization completes each step. Owner and date fields included.
- 04 The 6 reflections** - Fillable questions that pressure-test your own roadmap.
- 05 My 90-day plan** - Three fillable phase cards that turn this case study into your roadmap.

RUN IT AS A 90-MINUTE TEAM SESSION

25 min READ Read the story together. Ask everyone to note the one number that surprised them most.	40 min PRACTICE Work the 8-action checklist and the 6 reflections. Assign an owner and a date to every action you commit to.	25 min COMMIT Fill in the 90-day plan. One goal per phase, one number that tells you it worked. Book a review date before you leave the room.
--	--	---

THE STORY

What happened, and why it worked

01 The Company

SkyLift Cargo runs a freighter hub where the schedule is a living thing that has to be rebuilt several times a week. Twelve gates and 31 stands serve a rotation of freighter aircraft whose arrivals slip for reasons the hub does not control: weather cells over origin airports, late inbound rotations, customs holds that freeze an aircraft on the ramp. When disruption hits, the operations desk must reassign gates, stands, tow tractors and crews — often for 20-plus aircraft at once — under time pressure where every minute of ground time is billed at roughly \$2,800 an hour. The desk's quality of decision-making is, in a very literal sense, the hub's margin.

The economics of the problem were well understood inside the company. Cascading delays triggered overtime and occasional service credits with integrator customers, and the hub averaged three major disruptions a month, which meant the reassignment process was not an edge case — it was a recurring operational workload with a recurring cost. What made the situation delicate was history: a previous ops-software replacement had failed, and the desk staff who would use anything new were openly skeptical of anything sold to them as intelligence.

Leadership, to its credit, responded by approving a time-boxed pilot with explicit, pre-registered success criteria rather than a procurement program.

02 Where Things Stood

Before Emerson Technology engaged, disruption response ran on rules of thumb and a spreadsheet. When weather or a customs hold scrambled the rotation, coordinators worked the reassignment by hand — mentally sequencing aircraft against gate availability, tow-tractor positions and crew duty rules — taking around 45 minutes to settle a full reassignment while freighters burned billed ground time. The spreadsheet encoded the desk's accumulated judgment, but it could not search the space of alternatives; it could only record the one plan a coordinator had already reasoned to. Under pressure, the first feasible plan tended to win, not the best plan. The costs cascaded in familiar ways. Reassignments settled late produced ground-time overruns, overtime shifts and the occasional service credit with integrator customers whose delivery promises are contractual. Three major disruptions a month meant the desk spent a meaningful share of its collective attention rebuilding schedules rather than managing exceptions, and the 45-minute decision cycles compounded: a late decision on the first wave of aircraft constrains every decision after it. Meanwhile the memory of the failed software replacement hung over every conversation about tooling — the desk had been promised transformation before, and it had watched a system that ignored how the work actually happens get abandoned within a year.

The leadership team faced a genuine technology question as well. Optimization vendors were marketing quantum and quantum-inspired capabilities, McKinsey's Quantum Technology Monitor estimates quantum computing could create up to \$1.3 trillion in economic value by 2035 with logistics among the concentrated beneficiaries, and Gartner has advised that practical quantum advantage remains years away for most enterprises — leaving a plain question: was there value available today, on classical hardware, from better optimization of one well-chosen decision? The pilot was designed to answer that question honestly, which is why its success criteria were written before any code.

03 The Challenge

The challenge was to prove, or disprove, that a modern optimizer could beat experienced humans at the hub's most expensive recurring decision — without breaking the trust of a desk that had already survived one failed system. The decision itself was genuinely hard: reassigning 20-plus aircraft across 12 gates and 31 stands under tow-tractor availability, crew duty rules and turnaround-time constraints is a combinatorial problem where the number of feasible plans explodes faster than any human can enumerate them. The question was never whether the mathematics was interesting; it was whether the mathematics, fed the hub's messy live data, would produce decisions a duty manager could act on faster and better than judgment alone.

The organizational constraints were as real as the mathematical ones. Coordinators would not accept a system that issued verdicts, and they should not have — the desk retains accountability for live operations, and a tool that removes the human from the decision removes the human's reason to trust it. The pilot also had to be honest by construction: pre-registered metrics, a defined scope and explicit exit criteria agreed before work began, so that neither commercial enthusiasm nor sunk cost could move the goalposts after results arrived. Finally, the data reality had to be faced: the hub's live flight and ground data lived in two legacy feeds with mismatched identifiers and update cadences, and every ambitious optimization in the world was worthless until those feeds were trustworthy.

04 Discovery and Diagnosis

Emerson Technology began with a two-week discovery sprint spent almost entirely on the desk. The team observed real disruption events, catalogued the constraints coordinators actually weighed — some formal, some habitual — and reconstructed the anatomy of the 45-minute decision cycle. The diagnosis found that coordinators were not slow; the search space was simply too large for sequential human reasoning, and the first workable plan was usually accepted because generating a second one cost minutes the operation did not have. The team also classified disruptions into five recurring classes by cause and shape, which later proved essential for defining honest scope.

The technical audit quantified the data problem early. Two legacy feeds — the flight-management system and the ground-operations tracker — disagreed on identifiers, refreshed at different cadences and contained edge cases any solver would inherit. The team priced that integration candidly and told leadership the honest number: feeds of this kind would consume roughly 40% of total project effort, and any plan that assumed the algorithm was the schedule risk was planning the wrong risk. Gartner's guidance that practical quantum advantage remains years away for most enterprises shaped the recommendation to evaluate quantum-inspired classical algorithms — annealing-style solvers running on today's cloud hardware — rather than waiting on hardware logistics cannot yet buy. The pilot charter that emerged specified two disruption classes, pre-registered metrics for decision time, ground-time overruns and overtime, and explicit exit criteria covering partial success.

05 The Solution

Emerson Technology modeled the hub as a constrained scheduling problem and built the decision tool around a quantum-inspired annealing-style commercial solver running on classical cloud hardware, orchestrated in Python. The architecture has three layers.

The model of the hub

The model encoded the hub as it actually operates: 12 gates, 31 stands, tow-tractor availability, crew duty rules and turnaround times, each with the real constraints the desk recognizes rather than the tidy constraints a textbook would prefer. The solver ingests live flight and ground data feeds, ranks feasible reassignments and presents the duty manager with three scored options rather than a single verdict, because operations staff adopt

tools that keep them in command. Scoring reflected the desk's own priorities — ground time, delay propagation risk and crew feasibility — so the tool's logic was legible to the people accountable for its recommendations.

Shadow mode and the interface

Before the tool ever influenced a live decision, it ran for six weeks in shadow mode, comparing its recommendations against human decisions on every real disruption that hit the hub. Each event produced a paired record — what the desk chose, what the solver proposed, and the ground-time consequences of each — which built the evidence base and a shared vocabulary between the desk and the model. The interface was built around the disruption classes in scope: the duty manager sees the event classified, the three scored options, and the constraint conflicts behind each score. The tool was deliberately designed as decision support, because the previous failed replacement had tried to be the system of record, and the desk remembered.

What the data integration really cost

The project's honest center of gravity was not the algorithm — it was the plumbing. Integrating the two legacy feeds, reconciling mismatched identifiers, aligning update cadences and handling the edge cases live operations produce consumed roughly 40% of total project effort, exactly as the discovery phase had predicted. The team treated this as a feature of the design rather than an overrun: the feeds were hardened once, validated continuously, and the solver's confidence in its inputs became visible to the desk, so a recommendation built on stale data could be recognized as such. Scenarios beyond about 30 aircraft still fell back to manual, a limitation documented rather than hidden, because the pilot's charter required publishing what failed alongside what worked.

06 The First 90 Days

Weeks one and two were spent on the charter and the model: success metrics, scope, budget and exit criteria signed by leadership before code, and the hub's constraints formalized with the coordinators who owned them. Weeks three through six built the solver integration and began the data plumbing, with the first end-to-end recommendation produced against a replayed historical disruption in week five — a moment the desk watched with professional skepticism rather than applause. Weeks seven through twelve ran shadow mode across every real disruption in the two in-scope classes, and the paired-record discipline turned each event into evidence: where the desk's plan beat the solver, the logs showed why; where the solver beat the desk, the ground-time deltas accumulated into the business case.

The shadow phase produced its share of humbling findings, which the team published internally as required by the charter. The solver's first recommendations sometimes violated unwritten sequencing habits the desk followed for good reasons the model had not captured, and two constraints had to be re-encoded after coordinators demonstrated why they mattered. Conversely, the solver consistently surfaced alternatives nobody on the desk had time to consider, and on several shadowed events its second-ranked option would have avoided overruns the desk's plan incurred. That repeated pattern is what converted skepticism into curiosity.

Weeks eleven and twelve moved the tool live for the two disruption classes in scope, with the desk retaining full authority to override. A weekly review of every live event — decision time, chosen option, outcome — fed tuning, and the team kept a public log of model failures so the desk could see the tool's limits stated as plainly as its wins. By day 90 the pattern was set: faster decisions on the in-scope classes, manual handling preserved for the classes the model could not yet serve, and a desk asking when the third class would be added.

07 The Results

For the two disruption classes in scope, the improvements were measurable and honest. Gate-reassignment decision time fell from about 45 minutes to 29 — a 35% reduction — and ground-time overruns dropped 18% as the desk began its decisions from three scored options instead of one hand-built plan. Ops-desk overtime fell 22%, and the desk now refuses to run those events without the tool. The honest ledger, published alongside the wins: no quantum advantage materialized over well-tuned classical heuristics, three of the five disruption classes remained manual, and the team kept the solver as decision support rather than declaring transformation.

METRIC	BEFORE	AFTER
Reassignment decision time	-	-35%
Ground-time overruns	-	-18%
Ops-desk overtime	-	-22%
Disruption classes automated	-	2 of 5

The durable gains were as much methodological as operational. The desk's trust — the scarcest resource in the program — was earned through shadow-mode evidence and the right to override, and it now expresses itself as demand for scope expansion rather than as tolerance. The hub has a documented, pre-registered methodology for evaluating optimization tools — charter, shadow mode, paired records, published failures — which will outlast this particular solver. Volkswagen's widely reported pilot optimizing public-bus routing in Lisbon showed the same shape years ago: value comes from classical-grade engineering applied to a well-chosen decision, not from the label on the algorithm. And because the failures were published in the same document as the wins, every number in the wins is believed.

08 What Your Business Should Do

SkyLift's playbook is replicable precisely because it was honest, and it starts before any code. Write the pilot charter first: define success metrics, scope, budget and exit criteria in writing, and have leadership sign them, because pre-registration is what keeps a pilot honest under commercial pressure. Model one decision, not the network — pick a single recurring decision with measurable timing and cost, since hub-wide transformations stall in committees while one well-chosen decision class can prove value in a quarter. Budget for data plumbing, not demos: expect integrating legacy feeds, mismatched units and edge cases to consume 40% or more of total project effort, because the algorithm is rarely the schedule risk. Benchmark against tuned classical methods, comparing any novel optimizer against your best current heuristics rather than against nothing at all — beating a spreadsheet baseline proves very little and convinces nobody technical.

Then design for the humans who will carry the decision. Run in shadow mode first, scoring the system's recommendations against real human decisions for several weeks before allowing live use, because shadow mode builds both the evidence and the trust you need. Offer options, not verdicts — present duty managers with three scored alternatives plus explicit tradeoffs instead of a single authoritative answer, since operations staff adopt tools that keep them in command. Publish what failed, not just wins: report which scenarios the system could not handle and why, in the same document as the wins, because credibility compounds and does not transfer from other projects. Define exit criteria up front, deciding before starting what partial success looks like and which gains get kept — most pilots end somewhere in the middle, and the discipline is in banking the middle. The quantum-era context makes this discipline more valuable, not less: McKinsey projects up to \$1.3 trillion in value by 2035 while Gartner advises that practical advantage remains years away for most enterprises, so the organizations that profit first will be the ones that learned to run honest optimization pilots on classical hardware

today.

Before the charter is signed, sit with the questions SkyLift's leadership had to answer honestly. Which single recurring operational decision causes your team the most overtime when disruptions hit? Would your planners accept three scored options from a system, or only explanations they can interrogate? How would you prove a new optimizer beats the heuristics your veterans already trust? What share of a promising pilot's budget will realistically go to integrating your legacy systems? If quantum advantage is still years away, which classical optimization problems are you neglecting today? Does your organization have a safe way to declare a pilot a partial failure and keep the gains? Answer these with your team alongside the action checklist in this practice book, and revisit them each quarter as your operation, your data feeds and the optimization tooling landscape evolve.

REAL-WORLD PROOF & BENCHMARKS

McKinsey's Quantum Technology Monitor estimates quantum computing could create up to \$1.3 trillion in economic value by 2035, concentrated in logistics, chemistry and finance. - McKinsey, 2023

Volkswagen's widely reported pilot used quantum computing to optimize public-bus routing in Lisbon, one of the earliest logistics proof-of-concepts of its kind. - Volkswagen, reported 2019

Gartner has advised that practical quantum advantage remains years away for most enterprises, fueling interest in quantum-inspired classical algorithms that run on today's hardware. - Gartner

THE PRACTICE PAGES

The 8-action checklist

Work through the list with your team. Tick each box as your organization completes the step, and fill in an owner and a target date so every action has a name and a deadline.

☐**1. Write the pilot charter before code**

Define success metrics, scope, budget and exit criteria in writing, and have leadership sign them. Pre-registration is what keeps a pilot honest under commercial pressure.

OWNER

TARGET DATE

STATUS

--	--	--

☐**2. Model one decision, not the network**

Pick a single recurring decision with measurable timing and cost. Hub-wide transformations stall in committees; one well-chosen decision class can prove value in a quarter.

OWNER

TARGET DATE

STATUS

--	--	--

☐**3. Budget for data plumbing, not demos**

Expect integrating legacy feeds, mismatched units and edge cases to consume 40% or more of the total project effort. The algorithm is rarely the schedule risk.

OWNER

TARGET DATE

STATUS

--	--	--

☐**4. Benchmark against tuned classical methods**

Compare any novel optimizer against your best current heuristics, not against nothing at all. Beating a spreadsheet baseline proves very little and convinces nobody technical.

OWNER

TARGET DATE

STATUS

--	--	--

☐**5. Run in shadow mode first**

Score the system's recommendations against real human decisions for several weeks before allowing live use. Shadow mode builds the evidence and the trust you need.

OWNER

TARGET DATE

STATUS

--	--	--

☐

6. Offer options, not verdicts

Present duty managers with three scored alternatives plus explicit tradeoffs instead of a single authoritative answer. Operations staff adopt tools that keep them in command.

OWNER

TARGET DATE

STATUS

--	--	--

☐

7. Publish what failed, not just wins

Report which scenarios the system could not handle and why, in the same document as the wins. Credibility compounds; it does not transfer from other projects.

OWNER

TARGET DATE

STATUS

--	--	--

☐

8. Define exit criteria up front

Decide before starting what partial success looks like and which gains get kept. Most pilots end somewhere in the middle; plan to bank the middle.

OWNER

TARGET DATE

STATUS

--	--	--

AFTER THE CHECKLIST: OUR TOP 3 PRIORITIES

From the eight actions above, write down the three you will start first - and the single number you expect each one to move. Three priorities with owners beat eight without any.

PRIORITY 1 NUMBER IT SHOULD MOVE	PRIORITY 2 NUMBER IT SHOULD MOVE	PRIORITY 3 NUMBER IT SHOULD MOVE
--	--	--

THE PRACTICE PAGES

Reflections for your team

Use these questions in your next planning session and type short answers into the boxes. The gaps you find are usually the fastest wins.

Q1. Which single recurring operational decision causes your team the most overtime when disruptions hit?

Q2. Would your planners accept three scored options from a system, or only explanations they can interrogate?

Q3. How would you prove a new optimizer beats the heuristics your veterans already trust?

Q4. What share of a promising pilot's budget will realistically go to integrating your legacy systems?

Q5. If quantum advantage is still years away, which classical optimization problems are you neglecting today?

Q6. Does your organization have a safe way to declare a pilot a partial failure and keep the gains?

THE PRACTICE PAGES

My 90-day plan

Finish the session by committing to a plan. Three phases, one goal each, one number that tells you it worked. Bring this page to your next leadership review - or to a free consultation with Emerson Technology, and we will pressure-test it with you.

Days 1-30 Focus: audit and alignment

PRIMARY GOAL**KEY ACTIONS****HOW WE MEASURE SUCCESS****OWNER**

Days 31-60 Focus: build and prove

PRIMARY GOAL**KEY ACTIONS****HOW WE MEASURE SUCCESS****OWNER**

Days 61-90 Focus: scale and measure

PRIMARY GOAL

KEY ACTIONS

HOW WE MEASURE SUCCESS

OWNER

Tip: save a copy per team so each group keeps its own answers. Revisit the reflections every quarter - the questions stay the same even as the numbers move.

BOOK THE REVIEW BEFORE YOU CLOSE THE BOOK

Plans without a review date drift. Before you close this practice book, agree on when you will come back to these pages, who will run that session, and when you will check the numbers again. If you want an outside pair of eyes, bring this page to a free consultation with Emerson Technology and we will pressure-test the plan with you.

REVIEW DATE	REVIEWED BY	NEXT CHECK-IN
------------------------------------	------------------------------------	--------------------------------------

Ready for a transformation like SkyLift Cargo's?

Emerson Technology designs and builds the data, AI, cloud and automation systems behind results like these. On a free consultation call we will map this exact playbook to your business - your data, your constraints, your numbers.

Book a free consultation

support@emersontechnology.com

or start online: emersontechnology.com/contact