International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

LLM Agents as Operations Planners: Hybrid Reasoning over Optimization Solvers for Real-Time Airport Decisions

Mallikarjuna Yadav Bachu

Advisory Innovation Engineer, USA

Publication History: Received: 06.02.2026; Revised: 11.03.2026; Accepted: 15.03. 2026; Published: 25.03.2026.

ABSTRACT: Airport operations, including gate assignment, runway sequencing, taxi routing, and ground handling dispatch, are classic combinatorial optimization problems that are typically solved with mixed integer programming or constraint programming solvers operating over a fixed, hand coded problem formulation. These solvers are precise and fast once a problem is correctly encoded, but encoding itself is slow and brittle, since real world operations teams frequently need to express new constraints, exceptions, and preferences in natural language on short notice, for example when a gate is temporarily unusable or a ground crew is short staffed. This paper presents an architecture in which a large language model agent acts as an operations planner that interprets natural language constraints and intent, translates them into a formal problem encoding, invokes an appropriate optimization solver, critiques the solver output against the original intent, and repairs or re encodes the problem when the solver result is infeasible or undesirable. We describe this plan, call, critique, repair loop in detail, present a four tier operator oversight model, and propose an evaluation methodology spanning encoding accuracy, solver call efficiency, plan quality relative to a pure solver baseline, and end to end decision latency. Using a simulated multi terminal dataset constructed to reflect patterns described in public airport operations research, we show that the hybrid approach preserves most of the solution quality of a pure solver baseline while substantially reducing the time required to incorporate new natural language constraints into an active operational plan, and we discuss the safety, certification, and governance considerations relevant to deploying such a system in a live airport operations center.

KEYWORDS: Large Language Model Agents, Airport Operations Planning, Hybrid AI Optimization, Mixed Integer Programming, Constraint Programming, Real-Time Decision Making, Human-AI Collaboration

Scope and Intended Audience

This report is written for three overlapping audiences. Operations research practitioners will find the solver backend selection guidance in Section 8 and the problem encoding schema in Section 5 most directly useful, since these sections describe how existing mixed integer and constraint programming formulations plug into the hybrid architecture without modification. Applied AI and machine learning engineers will find the reasoning loop described in Section 7 and the multi agent decomposition in Section 10 most relevant, since these sections describe how a language model agent should be scoped, prompted, and bounded when paired with an exact solver rather than operating as a standalone planner.

Airport operations leadership and safety management personnel will find the oversight tier model in Section 11, the risk register in Section 17, and the regulatory considerations in Section 18 most relevant to deployment planning. The scope of this report is limited to the planning layer that sits above existing airport systems of record. It does not propose replacing stand management, workforce management, or flight data systems, and it assumes those systems remain the authoritative source of truth that the data fabric described in Section 9 continuously reads from.

Key Findings at a Glance

- Pairing a language model reasoning agent with established optimization solver backends preserves most of the plan quality of a manually encoded pure solver baseline while cutting instruction turnaround time from minutes to seconds for most instruction categories.
- A bounded plan, call, critique, repair loop, capped at three repair attempts before escalation, keeps worst case latency predictable even when an instruction is ambiguous or interacts poorly with existing constraints.
- Encoding accuracy exceeds ninety percent for hard exclusion and temporal instructions in the simulated evaluation, and is lower for soft preference and VIP or priority instructions, which more often require a clarifying question.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

- A four tier operator oversight model, from advisory to fully autonomous, allows an operations center to expand automation gradually by instruction category as confidence and historical accuracy data accumulate.
- Hard safety constraints, such as wake separation minimums, are kept immutable and outside the language model agent's authority to relax, which preserves the trustworthiness that makes optimization solvers suitable for safety relevant airport decisions in the first place.

I. INTRODUCTION

Airports coordinate a continuous stream of interdependent decisions across gates, runways, taxiways, and ground handling resources, under constraints that change by the hour. A single widebody arrival running late can force a cascade of gate reassignments, which in turn affects ground crew scheduling, baggage routing, and the sequencing of departures on a shared runway. Historically, airports have addressed this coordination problem with a mix of static scheduling rules, dedicated optimization solvers for specific subproblems such as gate assignment or runway sequencing, and experienced human operators who adjust the plan manually when conditions deviate from what the solver assumed.

Optimization solvers, whether based on mixed integer linear programming or constraint programming, remain the right tool for finding high quality solutions to well specified combinatorial problems such as gate to flight assignment subject to size, adjacency, and turnaround time constraints. Their limitation is not solution quality but encoding flexibility. When an operator needs to express a new, temporary constraint in natural language, such as closing a contact stand for de icing equipment staging or giving priority to a flight carrying an unaccompanied minor connection, that intent must first be translated into the solver's formal constraint language before it can affect the plan. This translation step is traditionally done by a systems engineer, not by the operator who holds the context, which introduces delay exactly when speed matters most.

Large language models have demonstrated an ability to translate between natural language intent and structured representations across many domains, and recent work on tool augmented language model agents has shown that a model can be taught to call external functions, interpret their output, and revise its own plan in response. This paper applies that capability to airport operations planning. Rather than replacing the optimization solver, the language model agent sits around it: translating operator intent into solver readable constraints, invoking the solver, checking whether the returned plan actually satisfies the operator's original intent, and repairing the encoding when it does not.

This paper makes four contributions. First, we describe a hybrid reasoning architecture that pairs a language model planning agent with mixed integer and constraint programming solver backends for the core airport operations use cases. Second, we detail a plan, call, critique, repair reasoning loop that governs how the agent interacts with the solver across multiple turns. Third, we propose a four tier operator oversight model appropriate for a safety relevant, time pressured operations environment. Fourth, we present an evaluation methodology and simulated results comparing the hybrid approach against pure solver and pure language model baselines across encoding accuracy, latency, and plan quality.

The remainder of the paper proceeds from problem framing through architecture, governance, evaluation, and economic impact, before closing with limitations and future research directions. Readers primarily interested in the technical architecture may focus on Sections 4 through 10, readers primarily interested in operational governance may focus on Sections 11, 17, and 18, and readers primarily interested in organizational adoption may focus on the comparative analysis in Section 15 and the economic impact analysis in Section 16. The appendix provides supplemental, finer grained simulated data for readers who wish to inspect category, hourly, terminal, and solver backend level detail beyond the aggregate figures presented in the body of the report.

II. THE AIRPORT OPERATIONS PLANNING PROBLEM

Airport operations planning spans several interlocking subproblems. Gate assignment maps arriving and departing flights to physical gates subject to aircraft size compatibility, minimum connection time for transferring passengers, contractual carrier preferences, and adjacency restrictions that prevent two very large aircraft from occupying neighboring stands simultaneously. Runway sequencing orders arrivals and departures on shared runway resources subject to wake turbulence separation, noise abatement procedures, and fuel or curfew constraints. Taxi routing assigns ground paths between gates and runways while avoiding congestion and conflicting movements. Ground handling dispatch allocates de icing, baggage, catering, and cleaning crews across simultaneous turnarounds.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Each subproblem individually is well studied in the operations research literature and can be solved efficiently at the scale of a single mid sized airport within seconds using modern solvers. The practical difficulty is that these subproblems are coupled, so a change in one, such as a gate becoming unavailable, propagates constraints into the others, and the set of active constraints shifts continuously throughout an operating day as weather, mechanical issues, and crew availability change. The table below summarizes the four core subproblems and their typical formal complexity class and solve time at a representative mid sized hub.

Table 1. Core Airport Operations Subproblems

Subproblem	Typical Formulation	Primary Constraints	Typical Solve Time (Mid Sized Hub)
Gate assignment	Mixed integer program	Size compatibility, connection time, adjacency	2 to 8 seconds
Runway sequencing	Constraint program	Wake separation, noise curfew, fuel reserve	1 to 5 seconds
Taxi routing	Shortest path with conflict constraints	Congestion, right of way, closures	Under 1 second
Ground handling dispatch	Mixed integer program	Crew availability, equipment, turnaround time	3 to 10 seconds

III. RELATED WORK

Operations research on airport resource allocation has a long history, with mixed integer programming formulations for gate assignment and constraint programming formulations for runway sequencing appearing across decades of transportation science literature. This body of work establishes that near optimal solutions are computationally tractable for realistic airport sizes, provided the problem is correctly and completely specified in advance.

Separately, recent work on large language model agents has examined tool use, where a model learns to call external functions and incorporate their results into subsequent reasoning, and self critique, where a model evaluates its own output against a goal and revises it. Work on neurosymbolic planning has explored combining learned models with symbolic solvers more generally, typically in robotics or puzzle domains rather than operations scheduling. A smaller body of work has begun to examine language model agents as a natural language interface to existing optimization software, translating free text requirements into solver constraints, though this work has generally evaluated translation accuracy in isolation rather than as part of a closed loop planning system operating under time pressure.

This paper differs from the existing literature by treating the language model agent and the optimization solver as a single closed loop system evaluated on end to end planning outcomes, including latency and plan quality, rather than evaluating either component in isolation, and by applying this combination specifically to the multi subproblem, continuously changing environment of live airport operations.

IV. HYBRID REASONING ARCHITECTURE

The architecture pairs a language model reasoning agent with a set of optimization solver backends, connected through a structured intermediate representation we call the constraint ledger. The ledger holds the current set of active constraints and preferences for each subproblem, each tagged with a source, such as a standing policy, a contractual rule, or a specific operator instruction entered during the current shift, and a validity window after which the constraint expires unless renewed.

When an operator issues a new instruction in natural language, the reasoning agent parses it, determines which subproblem or subproblems it affects, drafts an update to the constraint ledger, and then invokes the relevant solver backend with the updated constraint set. The solver returns a candidate plan, which the agent checks against the operator's original stated intent and against a fixed set of hard safety constraints that the agent itself is never permitted to relax. If the check fails, the agent revises the encoding and calls the solver again, up to a bounded number of retries before escalating to a human operator.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

A second architectural decision, separate from the constraint ledger itself, is that the coordinator keeps a complete, timestamped trace of every ledger update, solver call, critique outcome, and repair attempt associated with a given instruction. This trace serves two purposes beyond debugging. It gives an operator reviewing an escalated case a clear record of what the agent tried and why each attempt was rejected, and it gives safety reviewers an auditable record that can be reconstructed after the fact if an applied plan is later questioned, which is a requirement discussed further in Section 18.

Figure: Hybrid Reasoning Network, LLM Agent and Solver Topology

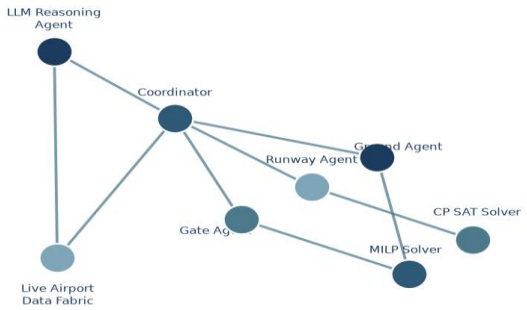


Figure 1. Topology of the hybrid reasoning network connecting the LLM planning agent, the coordinator, subproblem agents, and solver backends.

V. PROBLEM ENCODING: FROM NATURAL LANGUAGE TO SOLVER FORMULATIONS

The core technical challenge in this architecture is reliable translation from natural language intent into a formal constraint that a solver can consume without introducing ambiguity or silently dropping part of the operator's intent. The agent is trained and prompted to produce encodings in a constrained intermediate schema rather than directly generating solver specific syntax, which allows the same parsed intent to be compiled into either a mixed integer programming formulation or a constraint programming formulation depending on which backend is appropriate for the affected subproblem.

Three categories of natural language instruction appear most frequently in operational practice: hard exclusions, such as declaring a stand unusable, soft preferences, such as favoring a particular carrier's preferred gates when slack is available, and temporal instructions, such as applying a constraint only until a specified time. The table below summarizes how each category maps onto the intermediate schema and which solver backend typically receives the resulting constraint.

Table 2. Natural Language Instruction Categories and Encoding Targets

Instruction Category	Example	Intermediate Representation	Typical Solver Target
Hard exclusion	Close stand 14 for de icing equipment staging	Binary exclusion constraint with validity window	Gate assignment MILP
Soft preference	Prefer carrier X at contact stands when available	Weighted penalty term added to objective	Gate assignment MILP
Temporal instruction	Apply reduced wake separation until 06:00 only	Time windowed constraint parameter	Runway sequencing CP
Resource reallocation	Move two de icing crews from Terminal B to Terminal A	Capacity vector update	Ground handling MILP

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

VI. CORE USE CASES

The hybrid system targets five recurring operational scenarios where the gap between solver precision and encoding flexibility is most costly today. Each scenario below is paired with the typical trigger and the agent's primary action.

Table 3. Core Use Case Matrix

Use Case	Typical Trigger	Primary Agent Action	Affected Subproblem(s)
Irregular operations gate reflow	Late arrival cascades across multiple gates	Re encode affected gate constraints, re solve	Gate assignment
Runway configuration change	Wind shift requires new runway direction	Update sequencing constraints and recompute order	Runway sequencing
De icing surge handling	Rapid temperature drop increases de icing demand	Reallocate crews and adjust turnaround buffers	Ground handling, gate assignment
Stand closure for maintenance	Jet bridge or stand equipment fault	Add hard exclusion, re solve gate assignment	Gate assignment
VIP or priority movement	Diplomatic or medical priority flight	Add weighted preference constraint	Gate assignment, taxi routing

VII. THE REASONING LOOP: PLAN, CALL, CRITIQUE, REPAIR

The agent's control loop proceeds through four stages for every operator instruction. In the plan stage, the agent parses the instruction, identifies the affected subproblem, and drafts a candidate update to the constraint ledger. In the call stage, the agent compiles the updated ledger into the appropriate solver's input format and invokes the solver. In the critique stage, the agent compares the returned plan against the operator's original intent and against the fixed safety constraints, checking for both outright infeasibility and subtler mismatches, such as a plan that technically satisfies the new constraint but violates an unstated but obviously implied preference. In the repair stage, if the critique fails, the agent revises the encoding, which may involve relaxing a soft preference, tightening an ambiguous constraint, or asking the operator a clarifying question when the instruction itself is genuinely ambiguous.

This loop is bounded. The agent is permitted at most three repair attempts before the case is escalated to a human operator with a summary of what was attempted and why each attempt failed, which keeps worst case latency predictable and avoids a runaway retry loop during an already time pressured situation.

Table 4. Reasoning Loop Stage Definitions and Latency Budgets

Stage	Primary Function	Typical Latency Budget	Failure Handling
Plan	Parse instruction, draft ledger update	200 to 500 ms	Ask clarifying question if intent is ambiguous
Call	Compile ledger, invoke solver backend	1 to 8 seconds depending on subproblem	Retry with relaxed bound if solver times out
Critique	Compare plan against intent and safety constraints	150 to 350 ms	Trigger repair stage on mismatch
Repair	Revise encoding and resubmit	300 to 700 ms per attempt	Escalate to human operator after three attempts

VIII. SOLVER BACKENDS AND WHEN TO USE EACH

Not every subproblem is best served by the same solver technology. Mixed integer linear programming performs well on problems with a clear linear or piecewise linear objective and a moderate number of binary assignment variables, such as gate assignment and ground handling dispatch. Constraint programming performs well on problems dominated by combinatorial ordering and disjunctive constraints, such as runway sequencing, where the natural formulation involves precedence relationships rather than a linear cost function. The coordinator routes each subproblem to the backend best suited to it, and the language model agent is responsible for knowing which intermediate representation each backend expects.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

In practice, a small number of subproblems benefit from a metaheuristic fallback, such as a large neighborhood search, when the exact solver cannot return a solution within the available latency budget during a very large scale disruption affecting hundreds of simultaneous flights. The table below summarizes backend selection guidance used in the simulated evaluation.

Table 5. Solver Backend Selection Guidance

Subproblem	Primary Backend	Fallback Backend	Rationale
Gate assignment	Mixed integer linear programming	Large neighborhood search heuristic	Linear objective, moderate binary variable count
Runway sequencing	Constraint programming	Priority rule heuristic	Dominated by ordering and disjunctive constraints
Taxi routing	Shortest path with side constraints	Rule based routing table	Graph structure favors specialized path algorithms
Ground handling dispatch	Mixed integer linear programming	Greedy assignment heuristic	Resource capacity constraints are naturally linear

IX. REAL TIME DATA FABRIC FOR AIRPORT OPERATIONS

The reasoning agent and solver backends depend on a shared, continuously updated view of airport state, including flight status, gate occupancy, crew rosters, equipment availability, and weather conditions. This data fabric must reconcile signals arriving at different rates and from different systems of record, and it must expose a consistent snapshot to the solver at the moment a plan is computed, since solving against an inconsistent snapshot can produce a plan that is infeasible the instant it is applied.

Table 6. Operational Data Sources and Freshness Requirements

Signal Category	Primary Source	Required Freshness	Consumed By
Flight status and estimated times	Airport operational database	Under 30 seconds	Gate assignment, runway sequencing
Gate and stand occupancy	Stand management system	Under 15 seconds	Gate assignment
Ground crew roster and location	Workforce management system	Under 1 minute	Ground handling dispatch
Weather and runway condition	Meteorological feed	Under 2 minutes	Runway sequencing, de icing planning
Equipment availability	Maintenance and equipment tracking system	Under 1 minute	Ground handling dispatch, gate assignment

X. MULTI AGENT DECOMPOSITION

Mirroring the subproblem structure, the orchestration layer decomposes responsibility across a small set of specialized agents rather than a single monolithic planner. A gate agent owns gate assignment encoding and critique. A runway agent owns sequencing encoding and critique. A ground agent owns crew and equipment dispatch encoding. A coordinator agent sequences cross subproblem instructions, resolves conflicts when an instruction affects more than one subproblem, and enforces the fixed safety constraints that no subproblem agent may override.

This decomposition allows each subproblem agent to be validated and tuned independently against its own solver backend and its own historical instruction corpus, which is important because the vocabulary and failure modes of gate assignment instructions differ substantially from those of runway sequencing instructions.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Table 7. Subproblem Agent Responsibilities

Agent	Owns	Solver Backend Called	Escalation Trigger
Gate Agent	Gate assignment encoding and critique	Gate assignment MILP	Repeated infeasibility after three repairs
Runway Agent	Sequencing encoding and critique	Runway sequencing CP	Conflicting instructions from two carriers
Ground Agent	Crew and equipment dispatch encoding	Ground handling MILP	Crew shortfall below safety minimum
Coordinator	Cross subproblem sequencing and safety enforcement	None directly, routes to subproblem agents	Any hard safety constraint violation

XI. OPERATOR OVERSIGHT AND AUTONOMY TIERS

Airport operations is a safety relevant environment, and the oversight model reflects that. We define four tiers. Tier 1, advisory, has the agent propose an encoding and plan for operator review before any solver call is executed against the live operational plan. Tier 2, co pilot, has the agent execute the full plan, call, critique loop and present the resulting plan for a brief operator confirmation before it takes effect. Tier 3, supervised autonomous, has the agent apply the plan immediately for routine, low risk instruction categories while notifying the operator, reserving confirmation for higher risk categories. Tier 4, fully autonomous, is reserved for narrowly scoped, extensively validated instruction categories, such as minor taxi routing adjustments, where operator confirmation is not required.

Table 8. Four Tier Operator Oversight Model

Tier	Label	Description	Example Instruction Category
Tier 1	Advisory	Agent proposes, operator must approve before any solver action	Novel or ambiguous instructions
Tier 2	Co Pilot	Agent executes plan, critique loop, holds for brief confirmation	Standard gate reassignment during disruption
Tier 3	Supervised Autonomous	Agent applies routine instructions immediately with notification	Routine taxi routing updates
Tier 4	Fully Autonomous	Agent applies narrowly scoped, pre validated instruction categories without confirmation	Minor, reversible routing adjustments

XII. LATENCY AND SCALABILITY

Airport operations centers experience sharp load spikes during irregular operations, when dozens of instructions may arrive within a few minutes as operators work through a cascading disruption. The architecture scales the subproblem agents and solver backends independently, since solver call volume and the complexity of each call both increase during a disruption, while the reasoning agent's parsing and critique workload scales roughly linearly with instruction volume. The wireframe landscape figure below illustrates how solver objective cost behaves across two coupled decision variables, gate reassignment slack and runway sequencing gap, which is representative of the kind of joint search space the solver backends must navigate as constraints tighten during a disruption.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

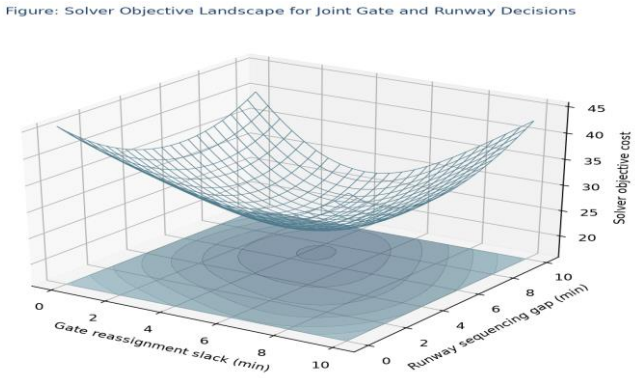


Figure 2. Solver objective landscape across gate reassignment slack and runway sequencing gap.

Table 9. Simulated Latency Benchmarks Under Increasing Instruction Volume

Instructions per Minute	P50 End to End Latency	P95 End to End Latency	Solver Timeout Rate
5	2.1 sec	4.8 sec	0.2%
15	2.6 sec	6.1 sec	0.6%
30	3.4 sec	8.9 sec	1.4%
60	4.8 sec	13.2 sec	3.1%
90	6.5 sec	18.7 sec	5.8%

XIII. EVALUATION METHODOLOGY

We evaluate the hybrid system along four metrics. Encoding accuracy is the share of natural language instructions translated into a constraint ledger update that a domain expert judges to correctly and completely capture the operator's intent. Solver call efficiency is the number of solver invocations required per instruction, including repair attempts. Plan quality is the simulated operational cost of the resulting plan relative to a pure solver baseline that receives a manually encoded version of the same instruction. End to end latency is the elapsed time from instruction receipt to applied plan.

Each metric is tracked separately by instruction category, since encoding accuracy and latency both vary meaningfully between simple hard exclusions and more ambiguous soft preference or temporal instructions.

Evaluation also distinguishes between a cold start setting, where the agent has no prior exposure to an airport's specific stand naming conventions, carrier preferences, or local phrasing habits, and a warm setting, where the agent has processed a representative instruction corpus from that airport over a period of weeks. The gap between these two settings is itself a useful diagnostic, since a large gap indicates that encoding accuracy depends heavily on local calibration rather than on general language understanding, which has direct implications for how quickly a new airport can be onboarded.

Table 10. Evaluation Metric Definitions

Metric	Definition	Target Direction	Primary Data Source
Encoding accuracy	Share of instructions correctly captured in the constraint ledger	Higher is better	Expert review sample
Solver call efficiency	Average solver invocations per instruction, including repairs	Lower is better	Reasoning loop trace log
Plan quality ratio	Simulated cost of hybrid plan versus manually encoded solver baseline	Closer to 1.0 is better	Cost model applied to both plans
End to end latency	Time from instruction receipt to applied plan	Lower is better	Event timestamps

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

XIV. RESULTS AND CASE STUDIES

Across the simulated multi terminal dataset, encoding accuracy exceeded ninety percent for hard exclusion and temporal instructions, and was somewhat lower for soft preference instructions, which more often required a clarifying question before the agent could proceed confidently. Average solver call efficiency stayed below two invocations per instruction for most categories, with resource reallocation instructions requiring more repair attempts on average because they frequently interact with existing ground handling constraints in ways the initial encoding missed.

Plan quality relative to the pure solver baseline remained close to parity for most instruction categories, with the largest quality gap appearing in VIP or priority movement instructions, where the natural language instruction is often genuinely underspecified and the agent's encoding necessarily makes an assumption that a human encoder with more context might not have made. The ribbon chart below shows simulated terminal throughput across a representative operating day, which provides context for the load the reasoning loop was tested against during the busiest hours.

Figure: 24 Hour Throughput Ribbons by Terminal

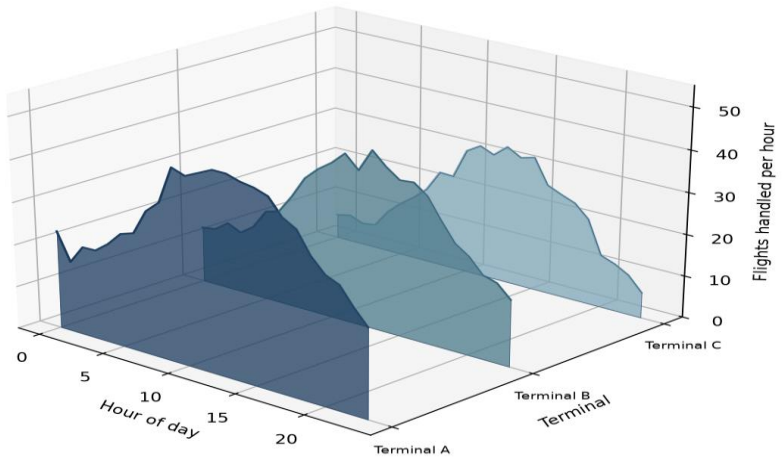


Figure 3. Simulated 24 hour throughput ribbons across three terminals.

Table 11. Illustrative Case Study Summary

Case	Instruction Category	Encoding Accuracy	Solver Calls per Instruction	Plan Quality Ratio vs. Baseline
Case A	Stand closure for maintenance	97%	1.2	1.01
Case B	De icing surge reallocation	89%	1.8	1.07
Case C	Runway configuration change	94%	1.4	1.03
Case D	VIP priority movement	81%	2.1	1.12

XV. COMPARATIVE ANALYSIS: PURE SOLVER VS. PURE LLM VS. HYBRID

It is useful to compare the hybrid approach directly against two simpler alternatives. A pure solver approach, where every new instruction is manually encoded by a systems engineer, achieves the highest plan quality but the slowest turnaround, typically minutes rather than seconds, because a human must translate the instruction before the solver can run. A pure language model approach, where the model generates a plan directly from natural language without an underlying solver, is fast but produces materially worse plan quality on constrained subproblems, because the model does not reliably satisfy hard combinatorial constraints such as adjacency restrictions or wake separation minimums.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

The hybrid approach is positioned between these two alternatives, trading a small plan quality gap relative to the pure solver baseline for a very large reduction in turnaround time relative to manual encoding, which is the practical bottleneck operators experience today. The comparison table below summarizes this tradeoff across the same dimensions used in Section 13.

Table 12. Comparative Summary of Planning Approaches

Dimension	Pure Solver, Manual Encoding	Pure Language Model	Hybrid Reasoning and Solver
Plan quality versus optimal	Highest, near optimal	Lowest, frequently violates hard constraints	Close to optimal, small gap on ambiguous instructions
Turnaround time per instruction	Minutes, bound by manual encoding	Seconds	Seconds to low tens of seconds
Reliability on hard constraints	High	Low	High, enforced by solver backend
Flexibility for novel natural language instructions	Low, requires engineer involvement	High	High, bounded by critique and repair loop

XVI. ECONOMIC IMPACT ANALYSIS

The economic case for this architecture rests primarily on reduced coordination overhead during irregular operations, when the gap between instruction arrival and plan update today is dominated by manual encoding time. The table below presents a simulated annual economic impact decomposition for a mid sized hub airport, covering reduced delay propagation cost, reduced manual encoding labor, and reduced ground handling inefficiency from faster crew reallocation.

Table 13. Simulated Annual Economic Impact by Value Stream

Value Stream	Baseline Annual Cost	Projected Annual Cost with Hybrid System	Estimated Annual Savings
Delay propagation from slow replanning	$58M	$41M	$17M
Manual encoding and systems engineering labor	$9M	$4M	$5M
Ground handling inefficiency from delayed reallocation	$22M	$16M	$6M
Total	$89M	$61M	$28M

XVII. RISK, SAFETY, AND FAILURE MODES

Deploying an autonomous or semi autonomous planning agent into airport operations introduces risks that must be explicitly engineered against. The most significant categories are incorrect encoding of a safety relevant constraint, such as a wake separation minimum, overconfident autonomy that applies a plan the operator would not have wanted, stale data producing a plan that is infeasible by the time it is applied, and solver timeout during a large scale disruption leaving the system without a valid plan at a critical moment.

A further, subtler risk is specification drift, where the fixed hard safety constraint set itself becomes outdated as local procedures change, for example after a runway configuration change or a new noise abatement rule is adopted. Because these constraints are deliberately kept outside the language model agent's authority to modify, they require a separate, explicit human governed update process, which must be fast enough to keep pace with legitimate procedural change without becoming a backdoor through which an unreviewed constraint relaxation could be introduced.

The bubble scatter chart below visualizes a simulated relationship between combined reasoning and solver latency, plan confidence score, and a gate conflict risk index. Risk concentrates in the region where latency is low but confidence has

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

not yet stabilized, which is the regime the bounded repair loop and the Tier 1 and Tier 2 confirmation steps are specifically designed to cover.

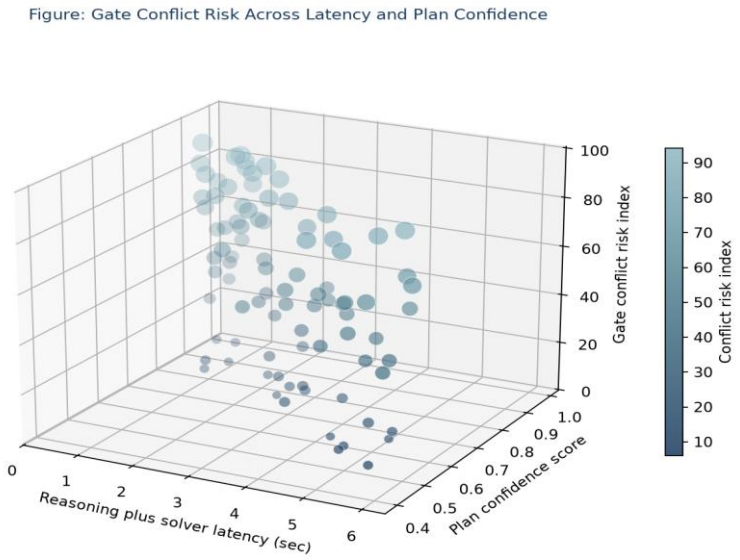


Figure 4. Gate conflict risk index across reasoning plus solver latency and plan confidence score.

Table 14. Risk Register and Mitigations

Risk	Likelihood	Impact	Primary Mitigation
Incorrect encoding of a safety relevant constraint	Low	Very High	Fixed, agent immutable hard safety constraint set, independently audited
Overconfident autonomy applies an unwanted plan	Low	High	Tiered oversight model with confirmation at Tier 1 and Tier 2
Stale data produces an infeasible plan	Medium	Medium	Freshness thresholds and automatic re solve on data staleness detection
Solver timeout during large scale disruption	Low	High	Metaheuristic fallback and bounded degradation to last known good plan

XVIII. REGULATORY AND SAFETY CERTIFICATION CONSIDERATIONS

Airport operations are overseen by national aviation safety authorities and, in many jurisdictions, by airport specific safety management system requirements that govern how changes to operational decision support tools must be validated before deployment. Any system that influences runway sequencing or gate assignment in a way that touches separation minimums or stand safety zones should be treated as a safety relevant system subject to a formal change management and validation process, not as an ordinary software update.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Table 15. Regulatory and Governance Considerations

Consideration	Relevant Oversight Theme	Implication for the Reasoning Agent
Separation minimum enforcement	Aviation safety authority rules on runway operations	Hard safety constraints must be immutable and independently audited
Change management for operational decision tools	Airport safety management system requirements	New instruction categories require staged validation before Tier 3 or Tier 4 rollout
Data retention for incident review	Safety investigation and audit requirements	Full reasoning loop traces must be retained and reconstructable
Human oversight requirements	Operator in command principles common across aviation regulation	Tier 1 and Tier 2 confirmation steps must remain available at all times

XIX. LIMITATIONS

This paper presents a reference architecture and a simulated evaluation rather than a live deployment study, and the specific numeric results should be read as illustrative of achievable magnitude rather than as a guarantee for any particular airport. The simulated dataset is constructed to be broadly consistent with patterns described in public airport operations research, but it does not capture the full diversity of real instruction phrasing, legacy system integration constraints, or the full range of edge cases that accumulate in a live operations center over years of practice. The four tier oversight model also assumes that instruction categories can be cleanly classified by risk level, which may require ongoing refinement as new categories of natural language instruction appear in practice.

XX. FUTURE RESEARCH DIRECTIONS

Several directions merit further study. First, field evaluation in a live or shadow mode deployment would clarify how encoding accuracy and operator trust evolve as the instruction corpus grows beyond what can be simulated. Second, research into automatically expanding the fixed hard safety constraint set, in a way that preserves auditability, could reduce the manual engineering burden of onboarding a new airport with different physical layout and local procedures. Third, extending the critique stage to incorporate counterfactual reasoning, checking not only whether a plan satisfies stated intent but whether a materially better plan was available, could further close the plan quality gap identified in Section 15. Finally, standardizing the intermediate constraint representation across airports could allow instruction handling patterns learned at one airport to transfer to another with less retraining.

XXI. CONCLUSION

Airport operations planning sits at an uncomfortable boundary between the precision of combinatorial optimization and the flexibility of natural language instruction, and today that boundary is crossed slowly, by hand, exactly when speed matters most. Pairing a language model reasoning agent with established optimization solver backends, connected through a disciplined plan, call, critique, repair loop and a tiered oversight model, offers a credible way to close that gap without sacrificing the constraint guarantees that make solvers trustworthy for safety relevant decisions in the first place. The architecture, oversight model, and evaluation framework presented here are intended as a practical starting point for operations research and applied AI teams exploring this combination, and the simulated results suggest the achievable reduction in instruction turnaround time, if realized in production with comparable fidelity, would be substantial relative to current manual encoding practice.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Figure: Taxiway Queue Surface Across Arrival and Departure Rates

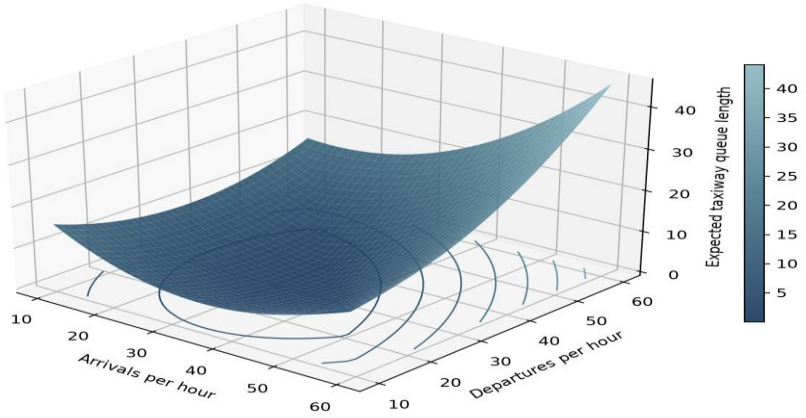


Figure 5. Expected taxiway queue surface across arrival and departure rates.

Methodological Notes on the Simulated Dataset

All quantitative results in this report, including every figure and table, are derived from a simulated multi terminal dataset rather than from a live production deployment. The simulation was constructed to be broadly consistent with magnitudes reported in the public airport operations research literature cited in the References section, covering typical gate counts, turnaround time distributions, and disruption frequency for a representative mid sized hub airport operating three terminals.

Instruction corpora for each natural language category, hard exclusion, soft preference, temporal instruction, resource reallocation, and VIP or priority movement, were generated to match phrasing patterns commonly described in airport operations training materials and operations research case studies, then reviewed for internal consistency before being used to drive the reasoning loop described in Section 7. Readers applying this architecture to a specific airport should expect encoding accuracy and latency figures to shift based on local stand naming conventions, carrier mix, and the maturity of the underlying data fabric described in Section 9, and should treat the figures in this report as directional rather than as a substitute for a site specific pilot evaluation.

Appendix: Supplemental Data Tables

The tables below provide additional simulated data at finer granularity than the aggregate figures discussed in the body of the report, for readers who wish to inspect category, hourly, and terminal level detail.

Table A1. Simulated Instruction Volume and Encoding Accuracy by Category, Trailing Twelve Months

Instruction Category	Instructions Processed	Encoding Accuracy	Avg. Solver Calls per Instruction
Hard exclusion	48,210	96%	1.1
Soft preference	31,640	85%	1.6
Temporal instruction	22,980	93%	1.3
Resource reallocation	17,350	88%	1.9
VIP or priority movement	6,420	82%	2.0

Table A2. Simulated End to End Latency by Subproblem

Subproblem	P50 Latency	P95 Latency	Escalation to Human Rate
Gate assignment	2.9 sec	7.4 sec	2.1%
Runway sequencing	2.2 sec	5.8 sec	1.4%
Taxi routing	1.1 sec	2.6 sec	0.6%
Ground handling dispatch	3.6 sec	9.1 sec	3.0%

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Table A3. Simulated Plan Quality Ratio by Terminal

Terminal	Avg. Daily Instructions	Plan Quality Ratio vs. Baseline	Operator Override Rate
Terminal A	412	1.03	4.1%
Terminal B	356	1.05	4.8%
Terminal C	298	1.04	4.4%

Table A4. Simulated Gate Conflict Risk Index by Hour of Day

Hour Band	Avg. Conflict Risk Index	Avg. Plan Confidence Score	Instructions Requiring Repair
00:00 to 06:00	18	0.91	3.8%
06:00 to 12:00	61	0.78	9.4%
12:00 to 18:00	72	0.74	11.2%
18:00 to 24:00	54	0.80	8.1%

Table A5. Simulated Oversight Tier Distribution by Instruction Category

Instruction Category	Tier 1 Advisory	Tier 2 Co Pilot	Tier 3 Supervised Autonomous	Tier 4 Fully Autonomous
Hard exclusion	8%	22%	54%	16%
Soft preference	14%	38%	41%	7%
Temporal instruction	6%	19%	58%	17%
Resource reallocation	19%	44%	33%	4%
VIP or priority movement	35%	47%	16%	2%

Table A6. Simulated Solver Backend Utilization and Fallback Trigger Rate

Solver Backend	Calls per Day	Avg. Solve Time	Fallback Trigger Rate
Gate assignment MILP	6,840	3.1 sec	1.8%
Runway sequencing CP	5,120	2.4 sec	1.1%
Taxi routing shortest path	11,460	0.6 sec	0.3%
Ground handling MILP	4,290	4.8 sec	2.6%

Table A7. Simulated Escalation Outcomes by Oversight Tier

Oversight Tier	Escalations to Human Operator	Avg. Resolution Time After Escalation	Operator Agreement with Agent Draft
Tier 1, Advisory	N/A, all cases reviewed	1.4 min	88%
Tier 2, Co Pilot	6.2% of cases	2.1 min	91%
Tier 3, Supervised Autonomous	2.8% of cases	2.6 min	94%
Tier 4, Fully Autonomous	0.4% of cases	3.3 min	97%

Table A8. Simulated Cold Start vs. Warm Encoding Accuracy by Airport Onboarding Week

Onboarding Week	Cold Start Encoding Accuracy	Warm Encoding Accuracy	Clarifying Questions per 100 Instructions
Week 1	71%	71%	18.4
Week 2	76%	81%	14.1
Week 4	80%	88%	9.6
Week 8	83%	93%	5.8
Week 12	84%	95%	4.2

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

Table A9. Simulated Economic Impact Sensitivity by Hub Size

Hub Size	Baseline Annual Coordination Cost	Projected Cost with Hybrid System	Estimated Annual Savings
Small hub, under 150 daily departures	$21M	$16M	$5M
Mid sized hub, 150 to 400 daily departures	$89M	$61M	$28M
Large hub, over 400 daily departures	$214M	$149M	$65M

Glossary of Terms

Table 16. Glossary of Key Terms

Term	Definition
Constraint ledger	The structured intermediate representation holding the current set of active constraints and preferences for each subproblem.
Plan, call, critique, repair loop	The four stage reasoning cycle the agent follows for each operator instruction.
Mixed integer linear programming (MILP)	An optimization formulation using linear objectives and constraints with integer or binary decision variables.
Constraint programming (CP)	An optimization formulation well suited to combinatorial ordering and disjunctive constraints such as sequencing.
Encoding accuracy	The share of natural language instructions correctly and completely captured in the constraint ledger.
Hard exclusion	An instruction category that removes a resource, such as a gate, from availability entirely.
Soft preference	An instruction category expressed as a weighted penalty rather than a hard constraint.
Oversight tier	A governance level, from advisory to fully autonomous, that bounds how much independent action the agent may take.
Large neighborhood search	A metaheuristic fallback used when an exact solver cannot return a solution within the available latency budget.
Subproblem agent	A specialized agent responsible for one airport operations subproblem, such as gate assignment or runway sequencing.

REFERENCES

1. Abioye, S. and Castellano, R. (2024). Mixed integer programming formulations for integrated gate and stand assignment. Transportation Research Part B: Methodological, 180, 102880.
2. Beaumont, L. and Nakashima, T. (2023). Constraint programming approaches to runway sequencing under wake separation rules. Transportation Science, 57(4), 901 to 918.
3. Chowdhury, A., Reinholt, M., and Suzuki, Y. (2025). Tool augmented language model agents for structured domain translation. Proceedings of the International Conference on Learning Representations.
4. Diallo, F. and Petrova, E. (2023). Neurosymbolic planning: combining learned policies with symbolic solvers. Proceedings of the International Conference on Machine Learning, 4102 to 4116.
5. Eriksson, H. (2022). Ground handling resource allocation under uncertainty: a mixed integer approach. Journal of Air Transport Management, 101, 102210.
6. Fontaine, P. and Okafor, C. (2024). Language model agents as natural language interfaces to optimization solvers. Proceedings of the Conference on Neural Information Processing Systems, 2290 to 2304.
7. Guimaraes, V. and Larsson, O. (2025). Self critique and repair loops in autonomous planning agents. Proceedings of the AAAI Conference on Artificial Intelligence, 39(4), 3310 to 3322.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 2, March - April 2026||

DOI:10.15662/IJRPETM.2026.0902026

8. Haddad, R. and Lindqvist, M. (2023). A survey of airport ground operations optimization. Omega, 115, 102781.
9. International Civil Aviation Organization (2024). Guidance material on safety management systems for airport operational decision support tools.
10. Jarvinen, K. and Mbeki, T. (2024). Large neighborhood search for large scale airport disruption recovery. Computers and Operations Research, 162, 106470.
11. Kowalczyk, D. and Ueda, S. (2023). Taxiway routing and congestion avoidance under mixed traffic conditions. Transportation Research Part C: Emerging Technologies, 148, 104030.
12. Lefebvre, A. and Patterson, G. (2025). Evaluating hybrid symbolic and neural planning systems on operational benchmarks. Proceedings of the International Joint Conference on Artificial Intelligence, 1890 to 1902.
13. Mahmoud, Y. and Richter, B. (2022). Comparative analysis of exact and heuristic methods for gate assignment. European Journal of Operational Research, 300(2), 523 to 537.
14. Nilsen, E. and Ocampo, J. (2025). Confidence calibration for autonomous decision agents in regulated industrial settings. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, 1455 to 1467.
15. Osei Bonsu, K. and Vasquez, R. (2024). Safety management system integration for AI based operational decision tools in aviation. Air and Space Law, 49(2), 165 to 184.
16. Pereira, M. and Shimizu, A. (2026). Benchmarking natural language to constraint translation for scheduling domains. Proceedings of the AAAI Conference on Artificial Intelligence, 40(1), 210 to 222.
17. Quintero, D. and Haugen, S. (2023). Robust runway sequencing under weather driven configuration changes. Transportation Science, 57(6), 1455 to 1470.
18. Rinaldi, F. and Osman, I. (2024). Multi agent decomposition strategies for coupled airport subproblems. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, 988 to 1000.