

Supplementary Material:

**An LLM-Based Agentic Framework
for Explaining Solution Evolution in Rolling Horizon
Optimization**

Sai Madhukiran Kompalli Hsuan-Han Chiu Can Li*

Davidson School of Chemical Engineering, Purdue University,
West Lafayette, Indiana 47907, USA

* Corresponding author: canli@purdue.edu *This document accompanies the FOCAPO-CPC*

2027 paper of the same title.

Deposited at PSEcommunity.org/LAPSE; not independently peer-reviewed.

1 Notation Reference

The main paper develops a large language model (LLM)-based framework for time-indexed linear programming (LP) and mixed-integer linear programming (MILP) models. Four query types (QT1–QT4) organize model comparison, solution comparison, prior-plan feasibility, and solution-change explanation. This supplement uses the main paper’s reported results and conventions.

Symbol	Definition
T^τ	Set of time periods in cycle τ
I_t^τ	Set of entity indices defined in period t at cycle τ
T_{ov}	Overlap periods: $T^A \cap T^B$
T_{new}	New periods: $T^B \setminus T^A$
T_{rm}	Removed periods: $T^A \setminus T^B$
$I_{t,\text{ov}}$	Overlap indices for period t : $I_t^A \cap I_t^B$
V_{ov}	Set of shared variables, including overlapping time-indexed and time-independent decisions
V_{diff}	Shared variables whose values differ beyond the numerical tolerance
V_{covered}	Variables in traced constraints, including binary variables reached by the shared-constraint coverage rule
$Z^{\tau,*}$	Optimal objective value at cycle τ
$\mathcal{F}^\tau$	Set of constraint families at cycle τ
λ_j^τ	Dual value for constraint j in the LP relaxation at cycle τ
γ	Relative optimality gap for retaining prior decisions, normalized by $\max(Z^{B,*} , \epsilon)$
ρ	Churn rate: fraction of overlapping variables whose value changed
Π_Z	Relative objective change: $\Delta Z / \max(Z^{A,*} , \epsilon)$
δ_k	Adaptive numerical tolerance for variable k
ϵ	Positive denominator floor (10^{-10}) used to prevent division by zero in the relative objective change and retained-plan optimality gap

2 Workforce Scheduling Formulation

The case study is a multi-period workforce scheduling problem based on the framework of Ernst et al. [1]. The complete mathematical formulation follows.

2.1 Sets

$I = \{W01, W02, \dots, W20\}$	Workers available for assignment
$J^A = \{J01, J02, \dots, J15\}$	Baseline jobs/projects
$T^A = \{1, 2, \dots, 20\}$	Baseline planning periods (weeks)
$IJ \subseteq I \times J$	Eligible worker-job pairs (skill-filtered)

For cycle B , $J^B = J^A \cup \{J16\}$ and $T^B = \{6, \dots, 25\}$. The equations below use J , T , and IJ for the sets of the cycle being solved. Eligibility is updated when jobs or worker skills change.

A worker i is eligible for job j (i.e., $(i, j) \in IJ$) if and only if the worker possesses at least one skill required by the job. Skills are drawn from the set {Python, Java, machine learning (ML), development and operations (DevOps), Frontend, Backend, Data, project management (PM)}. Each worker has 2–4 skills and each job requires 1–2 skills. Skill filtering reduces the

eligible worker-job-period combinations. The main paper reports a workforce instance with approximately 5,300 variables and 6,400 constraints; variables comprise binary assignments, continuous assigned hours, and overtime hours.

2.2 Parameters

$h_i \in \mathbb{R}_+$	Hourly labor cost for worker i
$a_{ij} \in \mathbb{R}_+$	Fixed assignment cost for worker i on job j
$m_{it} \in \mathbb{Z}_+$	Maximum available hours for worker i in period t
$d_{jt} \in \mathbb{Z}_+$	Required hours for job j in period t
$r_i \in \mathbb{Z}_+$	Regular hours threshold for worker i (overtime above this)
$\bar{n} \in \mathbb{Z}_+$	Maximum simultaneous job assignments per worker per period
$\underline{x} \in \mathbb{R}_+$	Minimum hours if worker is assigned to a job
$\omega \in \mathbb{R}_+$	Overtime penalty per hour above regular threshold

Default values: $r_i = 40$ for all workers, $\bar{n} = 3$, $\underline{x} = 4$, $\omega = 30$. Baseline worker capacities m_{it} range from 40 to 50 depending on the worker. Baseline hourly rates h_i are drawn uniformly from $[50, 120]$; selected rates are increased in cycle B . Rates shown in narrative examples are approximate whole-dollar values.

2.3 Decision Variables

$y_{ijt} \in \{0, 1\}$	1 if worker i is assigned to job j in period t
$x_{ijt} \in \mathbb{R}_{\geq 0}$	Hours worker i spends on job j in period t
$o_{it} \in \mathbb{R}_{\geq 0}$	Overtime hours for worker i in period t

2.4 Objective Function

Minimize total labor cost:

$$\min \sum_{(i,j) \in IJ} \sum_{t \in T} a_{ij} y_{ijt} + \sum_{(i,j) \in IJ} \sum_{t \in T} h_i x_{ijt} + \sum_{i \in I} \sum_{t \in T} \omega o_{it} \quad (1)$$

The three terms represent fixed assignment fees, hourly wages, and overtime penalties respectively.

2.5 Constraints

Worker capacity. Total hours assigned to a worker cannot exceed availability:

$$\sum_{j: (i,j) \in IJ} x_{ijt} \leq m_{it} \quad \forall i \in I, t \in T \quad (2)$$

Job demand satisfaction. Total hours allocated to a job must meet demand:

$$\sum_{i: (i,j) \in IJ} x_{ijt} \geq d_{jt} \quad \forall j \in J, t \in T \quad (3)$$

Linking (upper). Hours are allocated only if the worker is assigned:

$$x_{ijt} \leq m_{it} y_{ijt} \quad \forall (i, j) \in IJ, t \in T \quad (4)$$

Linking (lower). A minimum commitment is required if assigned:

$$x_{ijt} \geq \underline{x} y_{ijt} \quad \forall (i, j) \in IJ, t \in T \quad (5)$$

Maximum jobs per worker. Limits simultaneous assignments:

$$\sum_{j:(i,j) \in IJ} y_{ijt} \leq \bar{n} \quad \forall i \in I, t \in T \quad (6)$$

Overtime definition. Overtime hours exceed the regular threshold:

$$o_{it} \geq \sum_{j:(i,j) \in IJ} x_{ijt} - r_i \quad \forall i \in I, t \in T \quad (7)$$

3 Prompt Templates

Structured inputs are represented in JavaScript Object Notation (JSON). An irreducible infeasible subsystem (IIS) identifies an irreducible set of conflicting constraints and variable bounds. In the templates, RHS and LHS denote the right-hand side and left-hand side of a constraint; MIP denotes mixed-integer programming.

This section documents prompt templates for the four analytical approaches. The wording has been edited for clarity and consistency with the revised methodological description; these listings are not unmodified execution logs. The root agent classifies the question, and the comparison agent receives a query-specific template and structured numerical results. The reported experiment results are unchanged; no new evaluations of these edited templates are reported.

3.1 Root Agent Classification Prompt

You are the Root Agent of a system for explaining changes in a rolling horizon planning system.

In this system, operational plans are regenerated periodically as new information becomes available. Each planning cycle locks in near-term decisions and re-plans the remaining horizon. Your job is to classify the user's question into one of the types below.

CROSS-CYCLE TYPES (comparing two planning cycles):

- QT1 - What changed in the INPUTS/CONDITIONS between cycles
- QT2 - How the PLAN/SOLUTION differs between cycles
- QT3 - Can the PREVIOUS PLAN still be executed under new conditions
- QT4 - Evidence explaining WHY the plan changed

DISAMBIGUATION RULES:

1. "Why did X change?" -> QT4 (solution-change explanation).
2. "Can I still use last week's plan?" -> QT3.
3. "What's different in the conditions/inputs?" -> QT1.
"What's different in the schedule/plan?" -> QT2.
4. If the user asks about both what changed in conditions AND why

```

    the plan changed -> QT4 (it subsumes QT1).

Select one query_type from QT1, QT2, QT3, or QT4.
Output ONLY a JSON object with these fields (QT1 is an example):
{
  "query_type": "QT1",
  "original_query": "<the user's exact query>"
}

Do NOT answer the query. Do NOT explain your reasoning.
Output only the JSON.

```

3.2 QT4 Comparison Agent Prompt

The shared translation rules and vocabulary guidance below apply to each comparison template. All explanations should remain within the supplied numerical evidence. The QT4-specific instructions are:

```

You are explaining WHY the recommended plan differs between two
planning cycles.

ROLE: Explain the evidence associated with plan changes in plain
domain language. Distinguish observed differences from possible
explanations; do not assert unsupported causal chains.

YOU RECEIVE (JSON):
- summary: objective values, change amount, and coverage statistics
- traced: list of operational limits ranked by LP dual magnitude,
  each with:
    constraint: which specific limit
    family: category of limit (e.g., job_demand, worker_capacity)
    category: what happened to this limit
    dual: ranking score from a continuous LP relaxation; this is
      not a measured global impact on the MILP objective
    rhs_a, rhs_b: the limit values in each cycle
    variables: who/what was affected and by how much

ANSWERING RULES:
1. State the reported objective change using the supplied label
   and units. Mention the horizon context. Different horizons do
   not support an unqualified claim of operational improvement.
2. Organize the rest by FAMILY. For each family with meaningful
   changes, write one paragraph explaining:
   - What happened to the requirements or limits in that family
   - Who gained work and who lost work (translate names, state hours)
   - How changed requirements and direct costs help interpret the
     observed reallocation, accounting for coupled restrictions
   - If a rate changed between cycles, say so explicitly
3. Category meanings (use to frame explanation, never use the
   category names themselves):
   - "adjusted": a binding rule has changed bounds or coefficients.
     Describe the supplied change, using values from both cycles.
   - "binding_stable": requirement stayed the same but work was
     reshuffled. Say "Within the same X requirements, the plan
     redistributed work because..."
   - "tightened": an existing limit became active at the solution
   - "new_binding": a newly added limit is active

```

```

- "loosened": an existing limit is no longer active
- "removed_binding": a previously active limit was removed
4. Use plain domain language for costs, benefits, requirements,
   and limits. Use "hourly rate" for workforce wage coefficients;
   use the supplied domain descriptions for other components.
5. Summarize the supported contributors. Do not name a unique
   or biggest cause unless the supplied evidence establishes it.
6. Keep under 300 words. Be specific with numbers.

DATA:
{data}

USER QUERY: {query}

```

3.3 Restricted Vocabulary

The templates discourage the following optimization-specific terms in user-facing responses and request plain-language equivalents:

```

dual, shadow price, binding, LP, MIP, constraint, objective
function, feasible, infeasible, relaxation, slack, IIS, RHS, LHS,
coefficient, variable, parameter, sigma, alpha, repn,
index set, basis, bottleneck_evolution, delta_z,
delta_coeff, delta_struct

```

3.4 Interpretation Guide

The following translation rules guide each QT prompt. Domain names and objective units must be obtained from the model descriptions. Workforce-specific examples illustrate this mapping; other applications require their own descriptive metadata:

```

TRANSLATION RULES:
- Time units are "periods" unless context maps them to hours,
  days, weeks, or shifts.
- You MUST translate all model names into plain domain language.
  Examples:
    "W[('Reaction1', 'Reactor1', 3)]"
    -> "Reaction 1 running on Reactor 1 in period 3"
    "y[('W05', 'J11', 13)]"
    -> "Worker W05 assigned to Job 11 in period 13"
    "x[('W02', 'J01', 17)]"
    -> "Worker W02 hours on Job 01 in period 17"
- NEVER expose raw model names in your response.
- Use the objective label, units, and direction supplied by the
  model metadata. Report a change as an improvement only when
  the compared objectives have a comparable scope and horizon.

```

3.5 QT1 Comparison Agent Prompt

You are reporting what changed in the operating environment between two planning cycles.

ROLE: You summarize condition changes - NOT how those changes affected the plan.

YOU RECEIVE (JSON):

- horizon_context: how the planning window shifted; includes a "narrative" field (plain English summary)
- summary.set_changes: which index sets grew or shrank (new tasks, units, products, periods added/removed)
- summary.added/removed_constraint_families: new or removed operating rules
- summary.added/removed_variable_families: new or removed decision types
- summary.objective_coefficient_changes: which revenue or cost rates changed
- items: specific condition changes ranked by magnitude, each tagged:
 - "deliberate" = an intentional business update (new demand forecast, equipment change, price update)
 - "derived" = a consequence of executing the prior plan (e.g., inventory carried forward)

TRANSLATION RULES:

- Time units are "periods" unless context maps them to hours, days, weeks, or shifts.
- You MUST translate all model names into plain domain language.
- NEVER expose raw model names in your response.
- Use the objective label, units, and direction supplied by the model metadata. Report a change as an improvement only when the compared objectives have a comparable scope and horizon.

ANSWERING RULES:

1. Open with the rolling horizon context (use horizon_context.narrative verbatim or closely paraphrased).
2. Report STRUCTURAL changes first (new capabilities, removed options): new index sets, new/removed operating rules, new/removed decision types.
3. Then report DELIBERATE parameter changes ranked by magnitude. State exact values with percentage changes.
4. Items tagged "derived" are starting conditions inherited from executing the prior plan. Mention once briefly - do NOT list individually.
5. Group related changes (same equipment, same time range, same category).
6. End with a 1-sentence summary of the overall direction of change.
7. Do NOT discuss how these changes affect the plan.
8. Do NOT use optimization terms (dual, shadow price, binding, LP, MIP, constraint, objective function, feasible, infeasible, relaxation, slack, IIS, RHS, LHS, coefficient, variable, parameter).
9. Keep under 200 words.

DATA:

{data}

USER QUERY: {query}

3.6 QT2 Comparison Agent Prompt

You are reporting how the recommended operational plan differs between two planning cycles.

ROLE: You state FACTUAL differences in the plan - NOT why they occurred.

YOU RECEIVE (JSON):

- horizon_context: how the planning window shifted; includes a "narrative" field
- summary.objective_sense: "maximize" or "minimize"; use the objective label and units supplied by the model metadata
- summary.obj_a, obj_b: objective values for each cycle
- summary.delta_z, pct_change: how much the objective changed (absolute and percentage)
- summary.churn_rate: fraction of overlapping decisions that changed (0 = identical plan, 1 = completely different)
- summary.activations: discrete operations started in the new plan (not in the old)
- summary.deactivations: discrete operations stopped (were in the old plan, not in the new)
- items: specific decision changes ranked by magnitude
- new_variables: decisions enabled by new periods or entities

TRANSLATION RULES:

- Time units are "periods" unless context maps them to hours, days, weeks, or shifts.
- You MUST translate all model names into plain domain language.
- NEVER expose raw model names in your response.
- Use the objective label, units, and direction supplied by the model metadata. Report a change as an improvement only when the compared objectives have a comparable scope and horizon.

ANSWERING RULES:

1. Open with a brief mention of the horizon shift (use horizon_context.narrative).
2. Report the objective change in the supplied units and as a percentage. If horizons or model scope differ, describe a difference in totals rather than a like-for-like improvement.
3. If churn_rate is supplied, report "[X]% of the overlapping decision values changed" (convert the fraction to a percentage).
4. Use bullets for the top 3-5 most significant decision changes. Translate all names.
5. If supplied, separately note discrete changes: operations started (activations) and stopped (deactivations).
6. If new_variables exist, distinguish decisions for newly visible periods from those for newly introduced entities.
7. State FACTS ONLY. Do NOT explain causes.
8. End with a 1-sentence factual summary of the overall plan shift.
9. Do NOT use optimization terms (dual, shadow price, binding, LP, MIP, constraint, objective function, feasible, infeasible, relaxation, slack, IIS, RHS, LHS, coefficient, variable, parameter).
10. Keep under 200 words.

DATA:

{data}

USER QUERY: {query}

3.7 QT3 Comparison Agent Prompt

You are assessing whether the previous plan can still be executed given updated conditions.

ROLE: You deliver a clear YES/NO verdict and supporting evidence.

YOU RECEIVE (JSON):

- horizon_context: how the planning window shifted; includes a "narrative" field
- summary.compatible: true = old plan still workable, false = old plan violates updated limits
- IF COMPATIBLE (summary.compatible = true):
 - summary.obj_a_in_b: what the old plan achieves under new conditions
 - summary.obj_b_star: what re-planning achieves (the optimal under new conditions)
 - summary.suboptimality_gap: for positive-cost minimization, (retained-plan cost - re-planned cost) / retained-plan cost; 0.15 means re-planning saves 15% of retained-plan cost
 - summary.gap_class: "near-optimal" / "moderate" / "significant"
- IF NOT COMPATIBLE (summary.compatible = false):
 - summary.num_blockers: how many operational limits are violated
 - items: list of specific violations, each with:
 - constraint: which operational limit is broken
 - slack_needed: how much the limit would need to be relaxed
 - stage: diagnostic pass number; order is not causal evidence

TRANSLATION RULES:

- Time units are "periods" unless context maps them to hours, days, weeks, or shifts.
- You MUST translate all model names into plain domain language.
- NEVER expose raw model names in your response.
- Use the objective label, units, and direction supplied by the model metadata. Report a change as an improvement only when the compared objectives have a comparable scope and horizon.

ANSWERING RULES:

1. Open with a clear verdict: "The previous plan CAN still be executed" or "The previous plan CANNOT be executed as-is."
2. IF NOT EXECUTABLE:
 - List specific operational limits that are violated.
 - For each, state: what limit is broken, by how much (slack_needed), and in which period/equipment.
 - Group related violations (same equipment, same time range, same type of limit).
 - Suggest adjustments as possibilities; do not guarantee they restore overall feasibility unless that has been checked.
3. IF EXECUTABLE:
 - Report the cost of keeping the old plan in plain language.
 - For positive-cost minimization: "Re-planning saves \$X, or Y% of the cost of retaining the prior decisions."
 - For other objective conventions, report the objective values and absolute difference in the supplied units; do not reuse a percentage with an incompatible denominator.
 - Interpret severity: near-optimal = minor, moderate = meaningful trade-off, significant = strong reason to re-plan.
4. End with a 1-sentence actionable recommendation.
5. Do NOT use optimization terms (dual, shadow price, binding, LP,

```

MIP, constraint, objective function, feasible, infeasible,
relaxation, slack, IIS, RHS, LHS, coefficient, variable,
parameter).
6. Keep under 200 words.

DATA:
{data}

USER QUERY: {query}

```

3.8 LLM Configuration

The reported experiments used Sonnet 4.6 with temperature 0.0 to reduce output variability; this setting does not guarantee identical responses. The root classification agent used a maximum token limit of 200, and the comparison agent used a maximum token limit of 1000.

4 Query-Response Examples

This section presents author-edited query-response examples for the four query types on the workforce scheduling instance. These examples explain the reported numerical results and are not verbatim conversation transcripts. A saved history could support future aggregation of decision changes, recurring operating patterns, and persistent resource scarcity; this extension is not part of the present pairwise evaluation. Cycle *A* covers periods 1–20 (baseline). Cycle *B* covers periods 6–25 with five changes: Worker W05 on leave, new urgent Job J16, 20% rate increase for Workers W01–W03, approximately 50% demand increase for Job J01, and Worker W12 gaining a new skill.

4.1 QT1: Model Change Detection

User query: “What changed in the conditions between the previous and current planning cycles?”

Response:

The planning horizon rolled forward by 5 periods. Periods 1–5 are now committed and periods 21–25 are newly visible. The overlap covers periods 6–20.

Structural changes: A new job (J16) was added with demand of 60 hours per period, with ML and Data listed as eligible skill categories. Worker W12 gained the ML skill, expanding eligibility to additional jobs.

Updated operating conditions:

- Worker W05 availability dropped from 40 hours to 0 across all periods (on leave)
- Job J01 demand increased by approximately 50%; for example, from 40 to 60 hours in period 10 and from 33 to 50 hours in period 16 (about a 52% increase)
- Hourly rates for Workers W01, W02, and W03 increased by 20% (e.g., W01: \$110 → \$132)

Overall direction: increased workload (new job, higher demand) combined with reduced capacity (W05 unavailable) and higher costs for senior workers.

4.2 QT2: Solution Differences

User query: “How different is the new workforce plan from the previous plan?”

Response:

The planning horizon advanced by 5 periods (now covering periods 6–25).

Total cost decreased by \$44,570 (6.3% reduction, from \$712,767 to \$668,197).

Changes are compared on overlapping periods 6–20, with decisions for newly visible periods reported separately.

Key decision changes:

- Worker W05 dropped from all assignments in periods 6–20 across Jobs J11, J12, J15
- Worker W02 increased hours on Job J01 by 43 hours in period 17
- Worker W17 picked up Job J07 work previously handled by W02
- Workers W07 and W09 absorbed Jobs J11, J12, J15 from W05

New assignments and hour allocations are reported separately for periods 21–25 and the newly introduced job.

Overall, assignments and hours changed across the overlapping periods, alongside additional decisions in the new periods. The total cost comparison spans different horizons.

4.3 QT3: Feasibility of the Prior Plan

User query: “Can we still use the previous workforce plan?”

Response:

The previous plan cannot be executed as-is. The diagnostic procedure reports 20 violations of operational requirements.

Worker W05 availability: W05’s available hours dropped to zero, whereas the prior schedule assigned 30–40 hours per week on Jobs 11, 12, and 15 in overlapping periods 6–20. The diagnostics associate 17 violations with these unavailable-worker assignments.

Worker W15 overtime accounting: Three further diagnostic violations involve W15’s overtime accounting in periods 13, 14, and 17, with reported shortfalls of 35–40 hours.

The schedule needs revision to satisfy the updated requirements. Reassigning W05’s work is one necessary adjustment; the complete revised schedule must also satisfy the other operating requirements.

Methodological note. All shared decision values are fixed to their cycle A values, including overlapping overtime variables; only new variables remain free. The violation counts refer to diagnostic constraint instances, not necessarily distinct periods. The overtime equation defines required overtime hours; it is not an independent overtime cap. Diagnostic pass order does not establish causation between conflicts.

4.4 QT4: Solution Change Attribution

User query: “Why did the workforce plan change?”

Response:

The total cost decreased by \$44,570 (6.3%) across the two planning horizons.

Job requirement changes: Job 01 requirements jumped from 40 to 60 hours in period 10 and from 33 to 50 hours in period 16. These expanded requirements accompanied a reallocation. Worker 14 (\$85/hour) and Worker 12 (\$111/hour) absorbed the additional demand. Worker 01, whose rate jumped from \$110 to \$132/hour, was removed from Job 01. Worker 02 (rate increased from \$51 to \$61/hour) gained 43 hours on Job 01 in period 17.

Workforce rebalancing within stable requirements: Worker 03’s rate increased from approximately \$100 to \$121/hour. They lost substantial hours across multiple jobs and periods. Worker 01’s rate rose to approximately \$132/hour, accompanied by similar reassignments. The system shifted work to more cost-effective alternatives: Worker 12 (\$111/hour), Worker 16

(\$114/hour), Worker 20 (\$93/hour), and Worker 14 (\$85/hour) collectively gained hundreds of hours.

Worker availability: Worker 05's available hours dropped to zero. Their assignments on Jobs 11, 12, and 15 were redistributed to Workers 07 and 09 who had available hours in the affected periods.

The traced evidence connects the reallocations to changed job requirements, hourly rates, and worker availability.

5 Structured Analytical Evidence

This section illustrates selected fields in the structured evidence supplied to QT4. The examples are edited representations of the reported case, not raw execution logs. Hourly-rate fields use the approximate whole-dollar values reported in the main paper. Omitted fields have no implied value.

5.1 Summary Dictionary

The 571 traced variables out of 703 changed overlapping variables give 81.2% coverage, reported as 81% in the main paper. The trace contains 133 constraints. Coverage is the fraction reached by the stated structural rule, not the fraction of explanations independently verified.

```
{
  "qt": "QT4",
  "cycle_a": "wf_A",
  "cycle_b": "wf_B",
  "summary": {
    "objective_sense": "minimize",
    "obj_a": 712767.0,
    "obj_b": 668197.0,
    "delta_z": -44570.0,
    "n_diff_vars": 703,
    "n_traced_vars": 571,
    "n_traced_constraints": 133,
    "coverage": 0.812,
    "threshold": 0.80
  }
}
```

5.2 Traced Constraint Item (Job Demand, Adjusted)

```
{
  "constraint": "job_demand(['J01', 10])",
  "family": "job_demand",
  "category": "adjusted",
  "dual": 141.31,
  "rhs_a": 40.0,
  "rhs_b": 60.0,
  "lhs_coeff_changes": [],
  "variables": [
    {
      "name": "x(['W14', 'J01', 10])",
      "status": "overlap",
      "val_a": 10.0, "val_b": 45.0,
      "obj_coeff_a": 85.0, "obj_coeff_b": 85.0,
      "constr_coeff_a": 1.0, "constr_coeff_b": 1.0
    }
  ]
}
```

```

    {"name": "x[('W01', 'J01', 10)]", "status": "overlap",
     "val_a": 30.0, "val_b": 0.0,
     "obj_coeff_a": 110.0, "obj_coeff_b": 132.0,
     "constr_coeff_a": 1.0, "constr_coeff_b": 1.0},
    {"name": "x[('W12', 'J01', 10)]", "status": "overlap",
     "val_a": 0.0, "val_b": 15.0,
     "obj_coeff_a": 111.0, "obj_coeff_b": 111.0,
     "constr_coeff_a": 1.0, "constr_coeff_b": 1.0}
  ]
}

```

5.3 Traced Constraint Item (Job Demand, Binding Stable)

The field value `binding_stable` corresponds to the rebalanced category in the main paper: the requirement stays binding and unchanged while its participating decisions change.

```

{
  "constraint": "job_demand[('J08', 6)]",
  "family": "job_demand",
  "category": "binding_stable",
  "dual": 143.88,
  "rhs_a": 41.0,
  "rhs_b": 41.0,
  "lhs_coeff_changes": [],
  "variables": [
    {"name": "x[('W03', 'J08', 6)]", "status": "overlap",
     "val_a": 41.0, "val_b": 0.0,
     "obj_coeff_a": 100.0, "obj_coeff_b": 121.0,
     "constr_coeff_a": 1.0, "constr_coeff_b": 1.0},
    {"name": "x[('W16', 'J08', 6)]", "status": "overlap",
     "val_a": 0.0, "val_b": 41.0,
     "obj_coeff_a": 114.0, "obj_coeff_b": 114.0,
     "constr_coeff_a": 1.0, "constr_coeff_b": 1.0}
  ]
}

```

5.4 Binary Variable Coverage

QT4 extends coverage to a binary variable when it shares a constraint with a continuous variable already reached by the trace. This is a structural bookkeeping rule and does not, for a general MILP, prove that the continuous change caused the binary flip. In the workforce formulation, $x_{ijt} \geq \underline{x}y_{ijt}$ with $\underline{x} > 0$ implies that $x_{ijt} = 0$ requires $y_{ijt} = 0$; the upper link $x_{ijt} \leq m_{it}y_{ijt}$ alone does not establish this implication. Assignment indicators also appear in the maximum-jobs constraints. Dual values of linking constraints are not assumed to be identically zero.

6 Cycle Store Structure

When a solved model is registered, the cycle store extracts and saves the model records used by the analytical modules. Numerical comparisons rely on consistent component identities and time labels; readable explanations also depend on model names and descriptions. The current analysis compares two cycles. Aggregating evidence across a longer history is a future extension, as discussed in the main paper.

6.1 Stored Files

Each registered cycle produces the following files:

File	Contents
<code>model.pkl</code>	Serialized Pyomo ConcreteModel instance (via cloudpickle). Enables re-solving for QT3 feasibility checks.
<code>solution.json</code>	Objective value and all variable values keyed by Pyomo name with index (e.g., <code>x[('W02', 'J01', 17)] : 43.2</code>).
<code>structure.json</code>	Sets (members), variable families (index sets, domain), constraint families (size), constraint LHS coefficients (per-index variable-to-coefficient mapping), and doc strings.
<code>params.json</code>	All parameter values organized by family. Includes the raw data dictionary plus extracted objective coefficients.
<code>meta.json</code>	Summary statistics: cycle ID, number of variables, number of constraints, objective value, objective sense, solver status.
<code>bounds.json</code>	Constraint bounds per family per index: upper and lower bound values. Enables binding status comparison without loading the full model.
<code>duals.json</code>	LP dual values per constraint, computed on demand from the continuous LP relaxation when QT4 is first invoked, as described in the main paper. Cached after first computation and used as diagnostic ranking scores. The values characterize the continuous relaxation rather than the full MILP.

References

- [1] Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. “Staff scheduling and rostering: A review of applications, methods and models”. In: *European journal of operational research* 153.1 (2004), pp. 3–27.