OPT-Engine: Benchmarking the Limits of LLMs in Optimization Modeling via Complexity Scaling

Yitian Chen, Cheng Cheng, Yinan Sun, Zi Ling, Dongdong Ge

ICML 2026

Outline

Limitations: The "Static" Benchmark Gap

- **Small-Scale & Static:** Existing benchmarks (e.g., IndustryOR) are often restricted to textbook-size instances with fixed parameters(problem size <20) [2].
- **Evaluation Blind Spot:** Cannot distinguish between failures in *mathematical formulation* vs. *numerical execution*.
- **Data Contamination:** Static problems are likely present in LLM pre-training sets, leading to "memorized" rather than "reasoned" solutions.

Dataset	**Number After Correction**	**Affiliation**	problem size
NL4OPT	245	NeurIPS 2022 Competition	5-10
MAMO_EasyLP	642	CUHK-SZ	5-15
MAMO_ComplexLP	203	CUHK-SZ	5-15
IndustryOR	100	Cardinal Operations	5-20
OptMATH	166	PKH	5-20
Optibench	605	HKUST & Huawei	5-20

The Need

A benchmark that moves beyond "fixed snapshots" to **controllable complexity scaling**.

Research Directions: PTR vs. SIR

Research bifurcation in how LLMs approach optimization:

Pure-Text Reasoning (PTR)

Paradigm: End-to-end deduction [8, 4]

- **Mechanism:** Chain-of-Thought (CoT) text derivation [7].
- **Strength:** Mimics human intuition; no external tools needed.
- **Weakness:** Vulnerable to calculation errors.

Solver-Integrated Reasoning (SIR)

Paradigm: Formulation + Execution

- **Mechanism:** NL $\rightarrow$ Code (Pyomo/Gurobi) $\rightarrow$ Solver.
- **Strength:** Global Constraint modeling guarantees + numerical precision.
- **Weakness:** Requires strict syntax modeling.

The Hybrid Frontier: **CompPTR**

Augmenting text reasoning with **computational code tools** (e.g., calculators) to bridge the gap without full solver integration.

The Paradox: Superhuman Math vs. Brittle Optimization

- **The Phenomenon:** SOTA LLMs achieve $> 95\%$ on MATH500 [1], yet their performance **drops sharply** on optimization tasks of comparable difficulty [5].
- **Why the Gap?** Optimization modeling is a **subset** of mathematical reasoning — but with two critical differences:
  - Requires **numerical precision** across multiple interdependent variables
  - Demands **global constraint satisfaction**, not just a local or approximate answer.

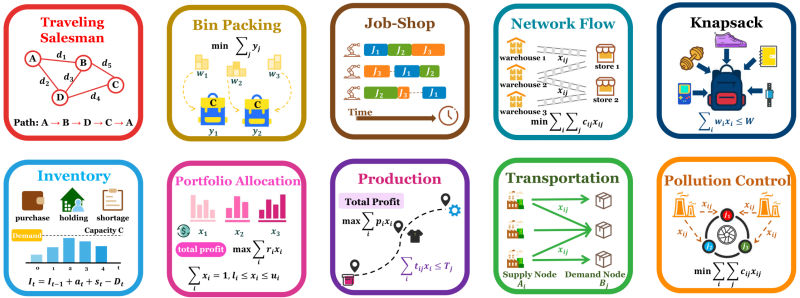
Research Question:

Why does a persistent performance gap exist between LLMs' deductive reasoning capabilities (for math) and their capacity for robust optimization modeling?

Our Proposal: The OPT-Engine Framework

OPT-Engine: An extensible framework with quantifiable and controllable complexity.

- Covers 10 canonical OR problems (from LP to MILP).
- Dynamically scales structural parameters (Constraints, Variables).



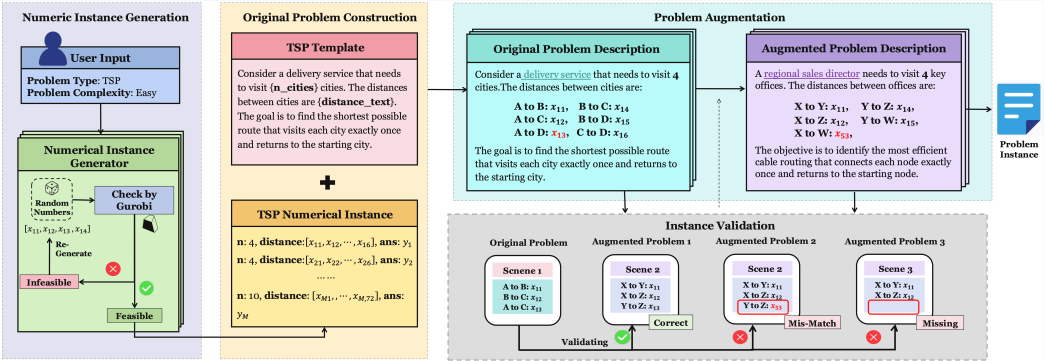
Key Research Questions

Three Pivotal Questions Addressed by OPT-Engine:

1. **Scalability:** While the current SOTA adopts Solver-Integrated Reasoning (SIR), can Pure-Text Reasoning (PTR) via classical Chain-of-Thought efficiently tackle optimization tasks as complexity scales?
2. **Tool Utility:** Can integrating external computational tools mitigate PTR's arithmetic weaknesses and achieve performance comparable to SIR?
3. **Bottleneck Identification:** What is the critical bottleneck limiting the efficacy of the current SOTA SIR approach throughout its generation process?
   - **Linguistic Ambiguity & Complexity:** Investigating whether performance degradation stems from the model's inability to parse nuanced natural language descriptions into precise mathematical variables.
   - **Objective Function Perturbations:** Assessing the model's sensitivity to changes in the objective goal.
   - **Scaling via Constraint Augmentation:** Stress-testing the "reasoning load" by incrementally increasing the number and type of constraints

OPT-Engine: Pipeline Workflow

1. **Numeric instance generation**: $G : \mathcal{D} \times \Theta \to \mathcal{I}$, with solver-based ground truth.
2. **Canonical problem construction**: $M : \mathcal{I} \times \mathcal{T} \to \mathcal{S}_C$.
3. **Problem augmentation**: $R : \mathcal{S}_C \times \mathcal{L} \to \mathcal{S}_R$ (LLM rephrasing).
4. **Instance validation**: LLM-as-judge + rule-based verifier ensures consistency.



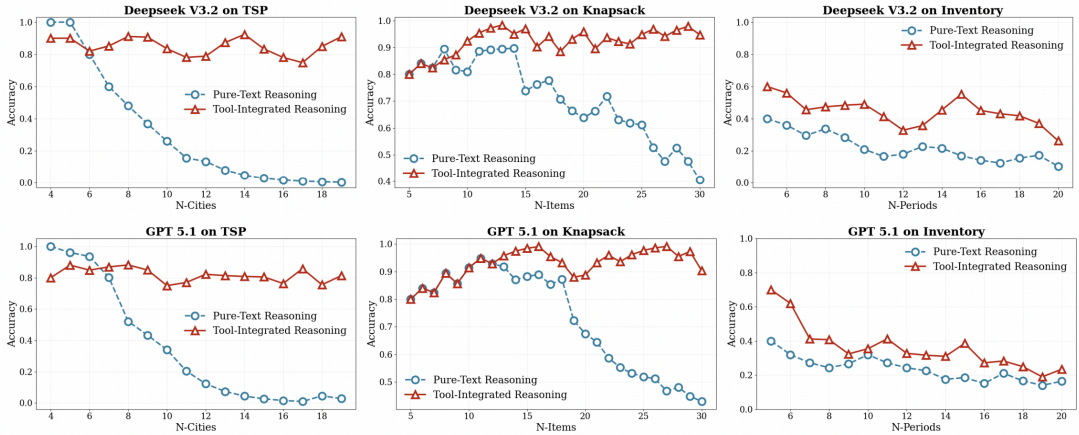
Experimental Setup & Bottleneck Analysis

- **Models:** DeepSeek-V3.2, GPT-5.1, Qwen3-4B-Instruct, Qwen3-4B-RL.
- **Accuracy Criterion:** Relative error $< 10^{-3}$. For each problem class and complexity level, we generate ten distinct instances. A solution $\hat{y}_i^{(p)}$ is correct if:

$$\frac{|\hat{y}_i^{(p)} - y_i^*|}{|y_i^*| + 10^{-6}} < 10^{-3}.$$

  We report avg@10 (mean success rate over the ten instances) as the standard accuracy measure.
- **Optimization task reasoning approaches:**
  - **SIR:** Gurobi execution (100s time limit).
  - **PTR:** Pure Chain-of-Thought, no tools.
  - **CompPTR:** PTR augmented with Python (numpy, itertools) but **no external solvers**.
- **Bottleneck Identification – What limits current SOTA SIR?**
  1. **Linguistic ambiguity & complexity:** base on base problem template generate problem with different PPL perperxity.
  2. **Objective function perturbations:** do both linear and nolinear pertuabation when generate new questions.
  3.
  4. **Constraint augmentation:** incrementally increase the number and type of constraints

SIR vs PTR: Frontier Models



- **SIR (red)**: accuracy remains high or degrades only slightly as problem scale increases.
- **PTR (blue)**: severe performance ceiling – accuracy collapses as as problem scale increases..

SIR vs PTR: Qwen3-4B Series

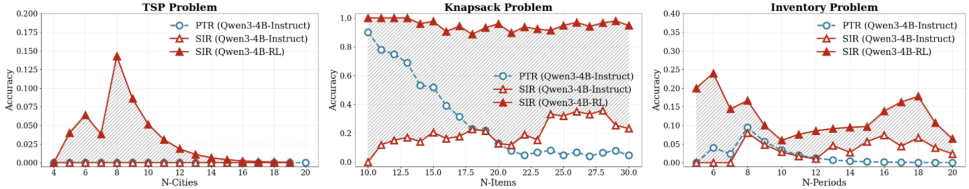


Figure 4. **Performance scaling of PTR (blue) vs. SIR (red) on the Qwen3-4B series**. The panel displays results from Qwen3-4B-RL, indicating RLVR training significantly enhances SIR mode accuracy.

Baseline Performance

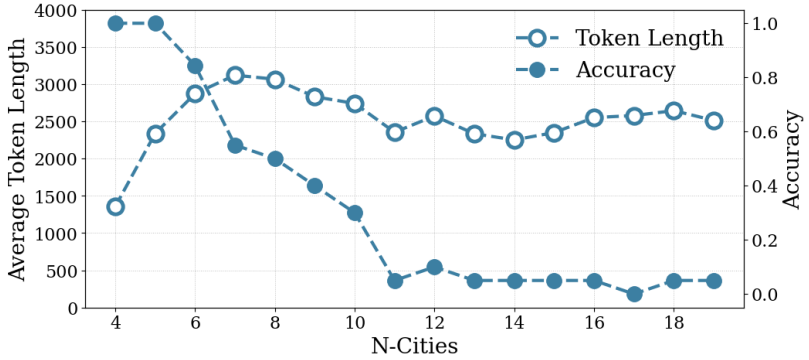
- **PTR vs. SIR:** At low complexity, **PTR** initially leads due to the Instruct model's limited code generation capacity.
- **Execution Ceiling:** Pure text reasoning acts as a temporary fallback for weaker coding models.

Impact of RLVR Fine-tuning

- **Scaling Shift:** Post-RLVR, **SIR** consistently surpasses PTR across all complexity levels.
- **Diagnostic:** RLVR significantly elevates code execution success rates, transforming SIR from a bottleneck into a strength.

Case Study: PTR Behavior on the TSP Problem

- **Small problem sizes:** Explicit enumeration of Hamiltonian cycles → high accuracy, token usage grows with complexity.
- **Beyond a critical threshold:** Lightweight heuristics (nearest neighbor, cheapest insertion) → token usage plateaus, solution quality drops.
- the model dynamically shifts to near-optimal, practical heuristics when exact solutions become intractable.



Implication: This adaptive strategy explains the efficacy of PTR in prior work [4, 6].

Constraint Infeasibility vs. Computational Inaccuracy

Hypothesis: PTR functions as a *stochastic approximation* — directionally accurate but numerically imprecise.

- **Feasibility:** Measures violations of hard logical or physical constraints.
- **Accuracy:** Measures numerical precision and residual computational errors.

Problem	Approaches	Easy		Medium		Hard	
		Feas/Acc	OptGap	Feas/Acc	OptGap	Feas/Acc	OptGap
TSP	PTR	73.3% / 58.3%	1.07%	70.0% / 10.0%	10.14%	64.0% / 2.0%	11.72%
	SIR	98.3% / 91.7%	0.01%	96.0% / 86.0%	0.25%	88.0% / 80.0%	0.24%
Knapsack	PTR	93.9% / 89.2%	5.62%	93.2% / 78.0%	0.29%	61.8% / 45.6%	0.42%
	SIR	100.0% / 92.3%	0.00%	100.0% / 91.7%	0.00%	100.0% / 95.7%	0.00%
Netflow	PTR	40.0% / 10.0%	3.57%	15.0% / 0.0%	2.64%	0.0% / 0.0%	0.00%
	SIR	100.0% / 100.0%	0.00%	100.0% / 100.0%	0.00%	100.0% / 100.0%	0.00%

Divergent Failure Signatures:

1. **SIR (Structural Logic):** Success is strictly contingent on **feasibility**. The solver guarantees optimality provided the logical grounding (constraints + objectives) is well-defined.
2. **PTR (Probabilistic Inference):** Exhibits a **dual-failure trajectory**:
   - **Reasoning Collapse:** Loss of global constraint consistency (feasibility) as task complexity scales.
   - **Precision Deficit:** Inherent lack of deterministic **accuracy**, even when operating on a correct logical manifold.

Can Computational Tools Rescue PTR?

Mechanism: CompPTR augments PTR with a Python interpreter for local arithmetic and deterministic sub-tasks (excluding solver calls).

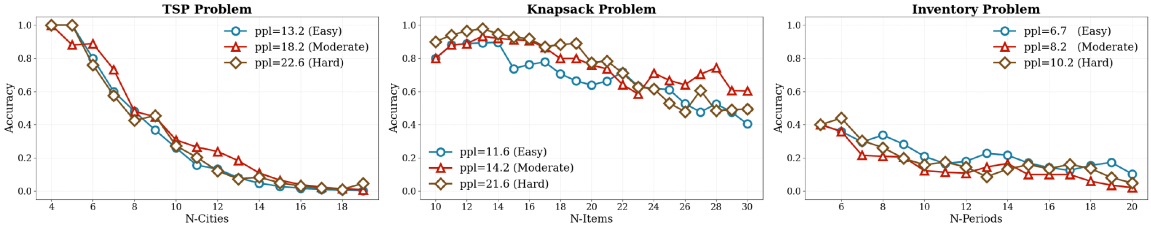
Problem	Approaches	Easy		Medium		Hard	
		Feas/Acc	OptGap	Feas/Acc	OptGap	Feas/Acc	OptGap
TSP	PTR	73.3% / 58.3%	1.07%	70.0% / 10.0%	10.14%	64.0% / 2.0%	11.72%
	CompPTR	86.7% / 85.0%	0.15%	72.0% / 70.0%	0.05%	52.0% / 44.0%	0.43%
Knapsack	PTR	93.9% / 89.2%	5.62%	93.2% / 78.0%	0.29%	61.8% / 45.6%	0.42%
	CompPTR	98.5% / 87.7%	0.00%	98.3% / 83.3%	0.01%	87.1% / 78.6%	0.00%
Netflow	PTR	40.0% / 10.0%	3.57%	15.0% / 0.0%	2.64%	0.0% / 0.0%	0.00%
	CompPTR	90.0% / 90.0%	0.00%	85.0% / 85.0%	0.00%	75.0% / 75.0%	0.00%

Conclusion: The "Global" Ceiling

External tools mitigate **local execution errors** but fail to enforce the **global optimization strategy** required for combinatorial consistency.

Bottleneck Analysis:Linguistic Complexity

- Start from a canonical problem template.
- Create two derivative versions with progressive syntactic and lexical complexity.
- All three templates share the **same numeric instance and constraint set** $\rightarrow$ linguistic variation is the sole independent factor.



When the underlying mathematical structure is held fixed, solution accuracy remains stable as Perplexity (PPL) [3] increases.

Bottleneck Analysis: Objective Perturbations

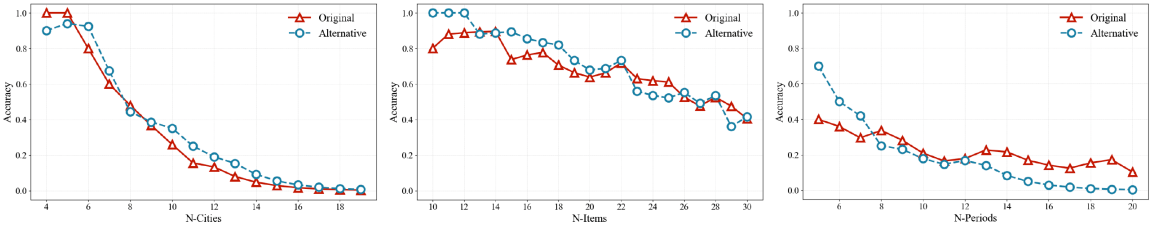
Intuition: Models might rely on "memorized signatures", a shift in the objective could potentially break heuristic-based reasoning.

Formal Perturbation Model

For an original objective $f(x)$, we apply a linear shift using a randomly sampled constant K (independent of decision variables):

$$\min_x f(x) \quad \Longrightarrow \quad \min_x f(x) + K$$

Property: The optimal solution x^* remains **invariant** under this shift.



- **Key Finding:** Empirical results show a **negligible impact** on accuracy across all problem classes.

Bottleneck Analysis: Constraint Augmentation

- **Setup:** For each problem class, we add mathematically straightforward constraints:
  - No new variables, only $\mathcal{O}(1)$ constraints $\rightarrow$ intrinsic mathematical difficulty and asymptotic complexity remain unchanged.
  - The modeling burden shifts from computation to **semantics**: the model must correctly parse and integrate auxiliary conditions.

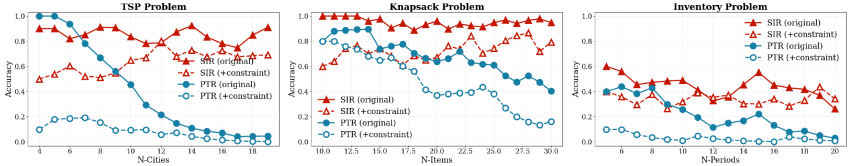


Figure 7. **Performance scaling under baseline vs. augmented constraints on the Deepseek-V3.2 Model.** PTR is shown in blue and SIR in red. Augmented constraints used in evaluation are as follows: **TSP**: Exactly one of the following two roads must be included in the tour: the road between city 1 and city 2, the road between city 2 and city 3. ($x_{01} = 0$, $x_{12} + x_{23} = 1$); **Knapsack**: Exactly one of item 1 and item 2 must be selected. If item 3 is selected, the effective backpack capacity is reduced by 2 kg. ($x_1 + x_2 = 1$, $\sum_{i=1}^{n} w_i x_i \leq C - 2x_3$); **Inventory**: On day t, the on-hand inventory must be at least $I_{\min}$ units. ($I_t \geq I_{\min}$). The introduction of augmented constraints leads to a consistent accuracy drop across problem classes.

- **Key result:** Both PTR and SIR show significant accuracy degradation compared to canonical problems.

Summary of Key Findings

1. **OPT-Engine Benchmark:** Introduced the first extensible framework capable of quantifying LLM performance across granular **complexity scaling** dimensions.
2. **The PTR Ceiling:** Pure-Text Reasoning suffers from a dual failure: **logical collapse** under high dimensionality and inherent **numerical imprecision**.
3. **CompPTR Limitations:** While tool-augmentation (Python) mitigates arithmetic errors, it remains insufficient for enforcing **global logical constraints** as problem size increases.
4. **The Formulation Bottleneck:** Structural constraint formulation is the primary barrier; even frontier models exhibit fragility when faced with incremental constraint additions.

Resource Release: Code and Data available at
`https://github.com/CardinalOperations/OPTEngine`

Conclusion and Future Directions

- **The SIR Necessity:** We establish that **Solver-Integrated Reasoning** is the essential paradigm for maintaining robustness in high-complexity industrial optimization.
- **Future Research Streams:**
  - **Advanced Metrics:** Developing tracks for reporting **optimality gaps**, computational runtime, and feasibility robustness.
  - **Heuristic Synthesis:** Evaluating LLMs' capacity to design meta-heuristics for NP-hard problems.
  - **Domain Expansion:** Extending OPT-Engine to support **nonlinear, stochastic**, and **dynamic** programming.

Acknowledgements

We thank **Siyu Shao (HKU)** and the anonymous reviewers for their invaluable feedback and contributions to this work.

Thank you!

References I

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt.

Measuring mathematical problem solving with the math dataset.

In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Chenyu Huang, Zhengyang Tang, Shixi Hu, Ruoqing Jiang, Xin Zheng, Dongdong Ge, Benyou Wang, and Zizhuo Wang.

Orlm: A customizable framework in training large models for automated optimization modeling.

Operations Research, 2025.

Frederick Jelinek.

Interpolated estimation of markov source parameters from sparse data.

In *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

Xia Jiang, Yaoxin Wu, Minshuo Li, Zhiguang Cao, and Yingqian Zhang.

Large language models as end-to-end combinatorial optimization solvers.

In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

References II


Hongliang Lu, Zhonglin Xie, Yaoyu Wu, Can Ren, Yuxuan Chen, and Zaiwen Wen.
OptMATH: A scalable bidirectional data synthesis framework for optimization modeling.
In *Forty-second International Conference on Machine Learning*, 2025.


Jianheng Tang, Qifan Zhang, Yuhan Li, Nuo Chen, and Jia Li.
Grapharena: Evaluating and exploring large language models on graph computation.
In *The Thirteenth International Conference on Learning Representations*, 2025.


Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al.
Chain-of-thought prompting elicits reasoning in large language models.
Advances in neural information processing systems, 35:24824–24837, 2022.


Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen.
Large language models as optimizers.
In *The Twelfth International Conference on Learning Representations*, 2024.