OptiTree: Hierarchical Thoughts Generation with Tree Search for LLM Optimization Modeling

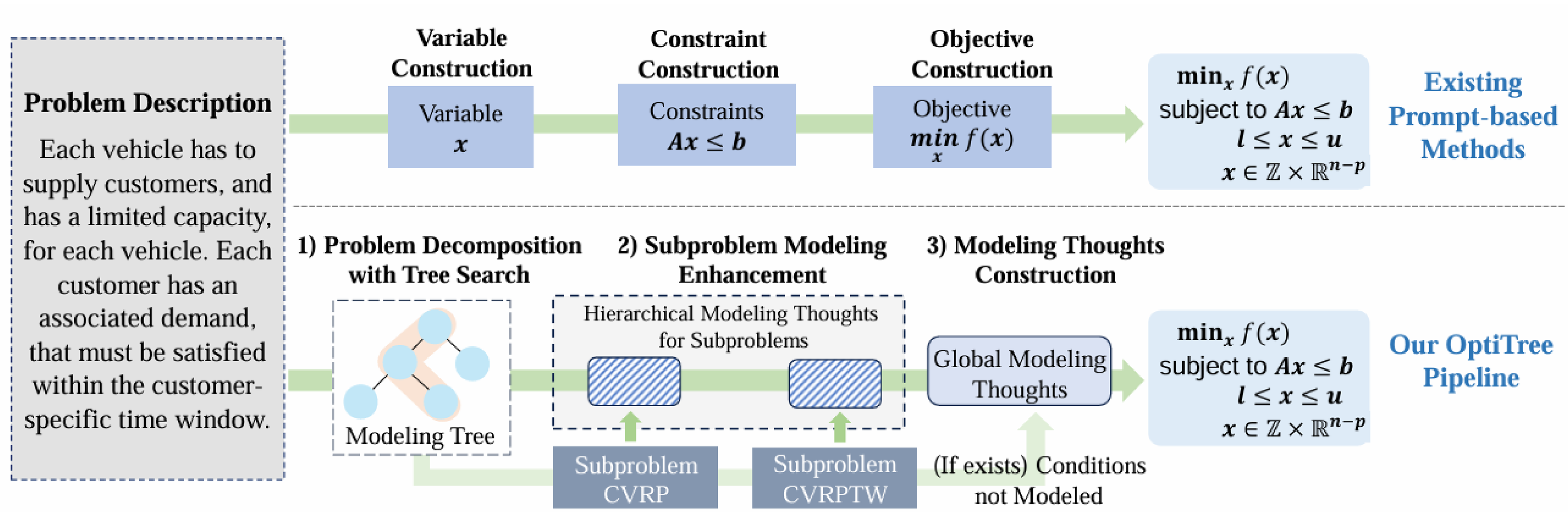
Haoyang Liu, Jie Wang, Yuyang Cai, Xiongwei Han, Yufei Kuang, Jianye Hao
University of Science and Technology of China
&Noah's Ark Lab

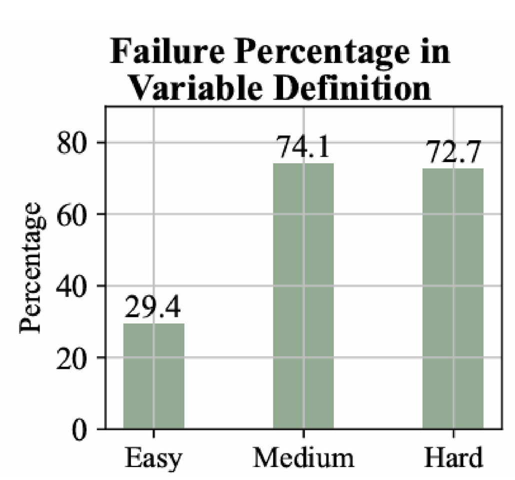
Introduction

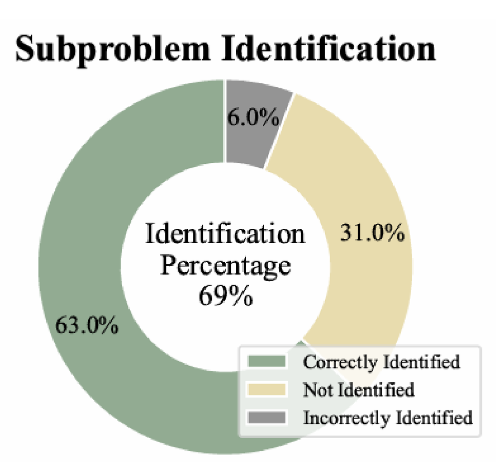
- Optimization modeling is a core technical part of operations research (OR) with wide applications, but manual modeling is time-consuming and requires professional expertise.
- Existing LLM-based prompt methods use fixed-step decomposition, failing to adapt to problem complexity and leading to suboptimal accuracy, especially for complex problems.
- OptiTree proposes a tree-search-based hierarchical decomposition approach, leveraging common subproblem patterns in OR problems to improve modeling accuracy.

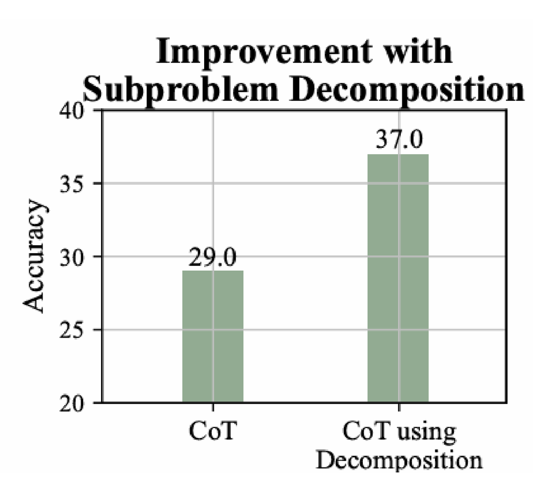
Motivation

- Fixed-step decomposition struggles with complex problems, with most errors stemming from incorrect variable construction in medium and hard tasks.
- Most complex OR problems can be decomposed into standard subproblems, and these decomposition patterns are widely applicable across scenarios.
- Guiding LLMs to model subproblems first significantly enhances their modeling performance for the original problem.



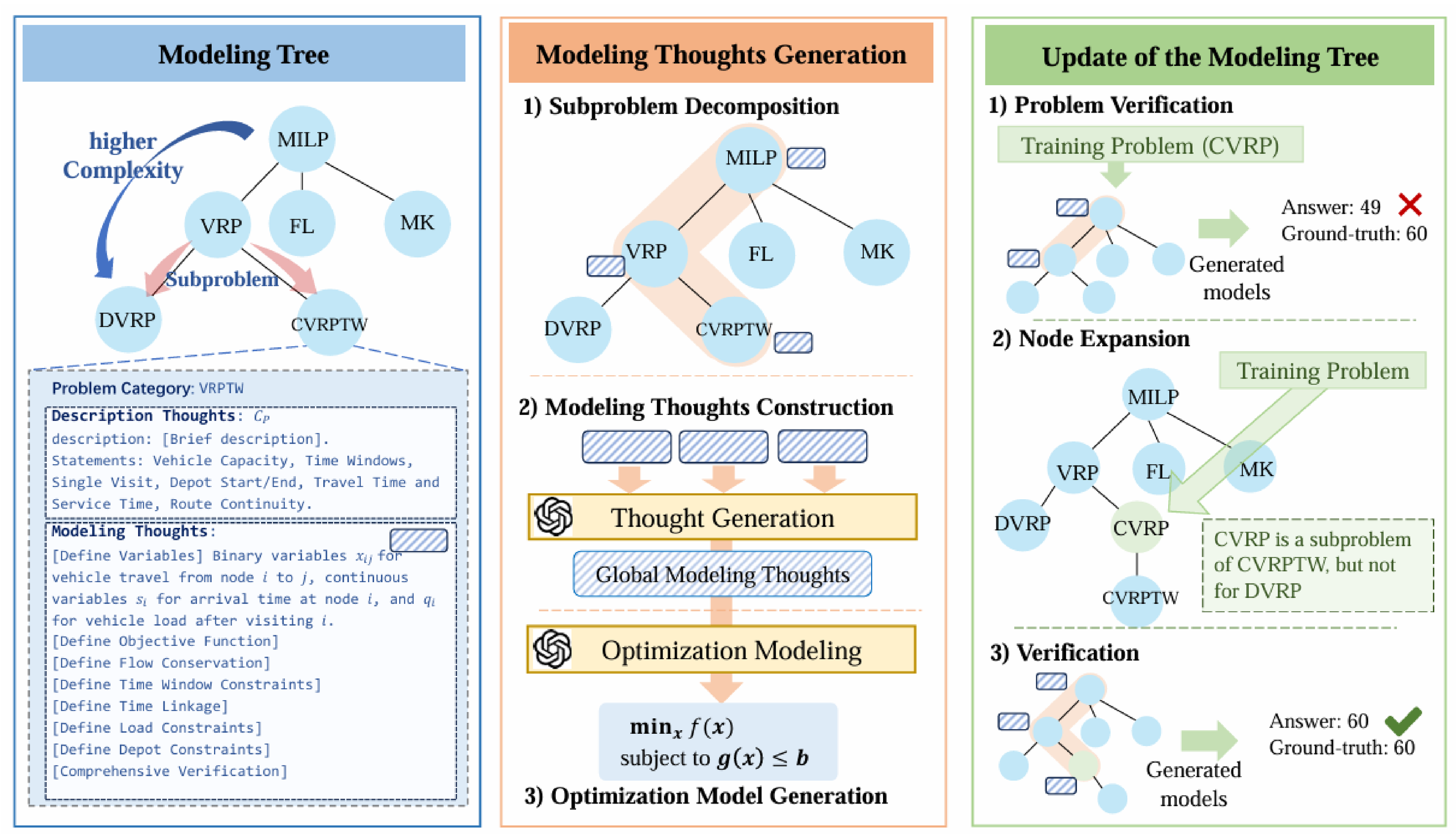


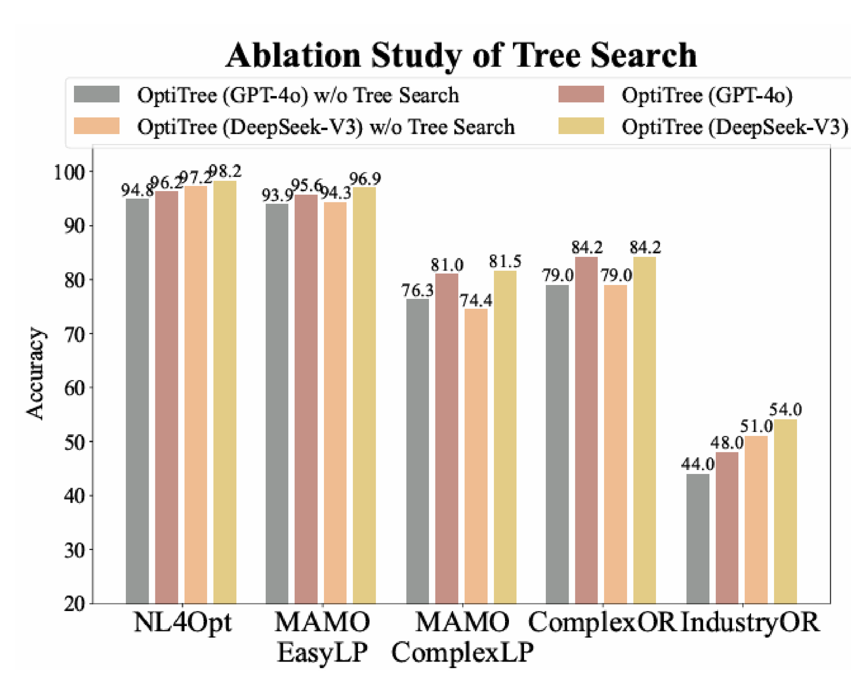


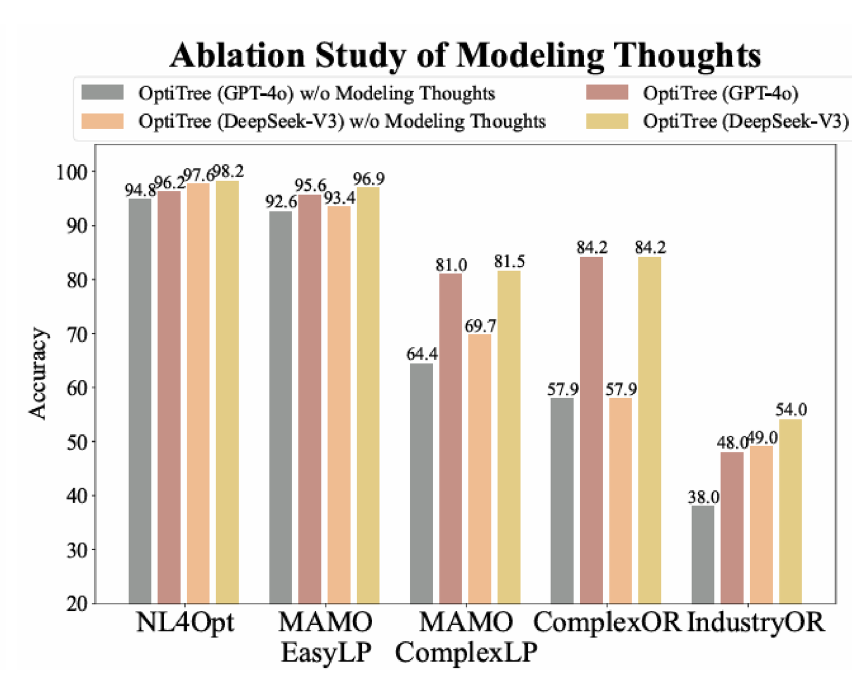


Method

- Construct a modeling tree where each node represents an OR problem, storing its description features and modeling thoughts (variable definition, constraints, objectives, etc.).
- Use tree search to adaptively find matching subproblems for the target problem, starting from the root and iteratively selecting the most similar subproblems.
- Synthesize global modeling thoughts by integrating hierarchical thoughts of subproblems, and dynamically update the modeling tree to ensure scalability and reliability.







Experiments

- Outperforms state-of-the-art baselines on 5 datasets (NL4Opt, MAMO EasyLP/ComplexLP, ComplexOR, IndustryOR), achieving over 10% accuracy improvement on complex datasets.
- Demonstrates strong generalization: trained on only 400 problems, it performs well on both easy and challenging benchmarks.
- Efficient in search and modeling, with lower total time cost compared to other prompt-based methods.
- Adapts to different LLM backbones (GPT-4o, DeepSeek-V3) and consistently maintains superior performance.

Table 1: Comparison of modeling accuracy between our method and baselines across the benchmarks. We mark the best results in **bold** the underline the second-best results.

Model	Method	NL4Opt	MAMO EasyLP	MAMO ComplexLP	ComplexOR	IndustryOR	OptiBench	OptMATH
Fine-tuned Method								
	ORLM	85.7	82.3	37.4	63.2	38.0	51.1	2.6
	Evo-Step	84.5	85.3	61.6	-	36.4	-	-
	OptMATH	95.9	89.9	54.1	-	31.0	66.1	34.7
	LLMOPT	93.0	**97.0**	68.0	72.7	46.0	66.4	40.0
Prompt-based Methods								
Reasoning LLMs	DeepSeek-R1	86.1	79.5	57.3	68.4	38.0	70.2	33.1
	OpenAI-o1	87.1	87.6	54.5	73.6	40.0	71.5	34.9
GPT-4o	Standard	70.3	84.3	41.2	57.8	27.0	42.3	17.5
	CoT	71.6	84.8	42.3	57.8	29.0	42.0	20.5
	CoE	76.4	85.7	46.4	68.4	34.0	43.2	18.6
	OptiMUS	82.0	85.1	47.3	79.0	34.0	45.8	20.2
	MCTS	90.3	87.4	56.8	68.4	42.0	64.0	37.3
	OptiTree	<u>96.2</u>	95.6	<u>81.0</u>	**84.2**	<u>48.0</u>	<u>71.9</u>	<u>45.8</u>
DeepSeek-V3	Standard	70.5	84.3	39.8	52.6	29.0	52.4	16.2
	CoT	74.0	82.9	40.7	52.6	35.0	53.1	21.1
	CoE	79.2	85.9	43.1	63.2	33.0	55.2	24.1
	OptiMUS	80.6	87.1	45.2	79.0	36.0	58.8	32.5
	MCTS	89.6	88.0	51.6	79.0	46.0	67.9	38.6
	OptiTree	**98.3**	<u>96.9</u>	**81.5**	**84.2**	**54.0**	**74.7**	**52.4**