

LERC-COLING 2024

Towards Human-aligned Evaluation for Linear Programming Word Problems

Linzi Xing[●], Xinglu Wang[●], Yuxi Feng[●], Zhenan Fan[●], Jing Xiong[●], Zhijiang Guo[●], Xiaojin Fu[●],
Rindra Ramamonjison[●], Mahdi Mostajabdaveh[●], Xiongwei Han[●], Zirui Zhou[●], Yong Zhang[●]

● Huawei Technologies

● Simon Fraser University

● University of British Columbia

● Sun Yat-sen University

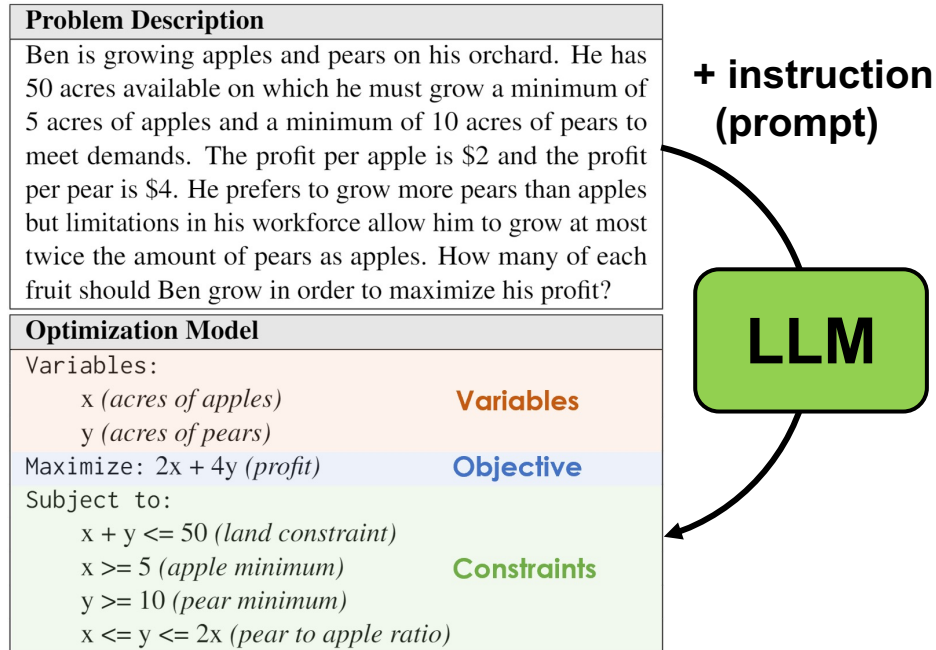


Background and Motivation

- ❑ **Math Word Problem (MWP)** is a fundamental NLP task:
 - Most prior research for MWP has primarily centered on elementary arithmetic problems and algebra problems.
 - **Linear Programming Word Problems (LPWP)**, as another particular category of MWP, remains largely under-explored.
- ❑ **LPWP** can more authentically reflect real-world decision-making processes and thus offer considerable potential to benefit the field of **operations research (OR)**.
- ❑ With the improving **emergent capabilities of LLMs**, building the model for LPWP becomes **end-to-end**, without breaking it into sub-tasks and requiring ground-truth data for these sub-tasks in scale, as:
 - Previously, the reasoning-intensive nature of LPWP mandates the deconstruction of prior neural solutions into sub-steps leading to inevitable error accumulation.
 - The data sparsity issue also introduces extra difficulty for neural approaches to consistently achieve reliable and robust performance.
- ❑ Therefore, how to evaluate the performance of gigantic LLMs on LPWP in an **effective and standardized manner** becomes critical and will attract more attention in the near future.

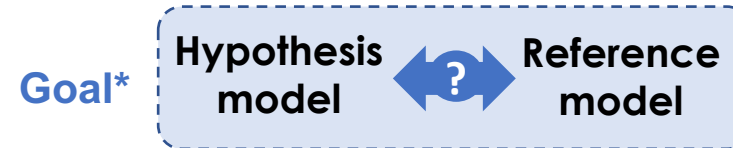
Limitation of Prior Art

- **Data Types:**



- Here we define the equivalence of two optimization models as two models contain exactly the same information covered in the problem description.

- **Prior Evaluation Strategies:**



- Canonical Metric

- Based on the declaration-level matching between hypothesis and reference model.
 - Not robust to the altered order of variables:

$$a \cdot X + b \cdot Y \leq c$$

$$b \cdot Y + a \cdot X \leq c$$

- Executable Metric

- Comparing the optimal solutions between hypothesis and reference models.
 - Problematic to detect the different models with identical optimal solutions (e.g., both infeasible).

Proposed Evaluation Method

- ❑ Our proposed evaluation method for LPWP of LLMs is based on **graph edit distance**, consisting of three steps:
 - **Step 1:** LP General Form Conversion
 - **Step 2:** Graph Representation Conversion
 - **Step 3:** Graph Edit Distance Calculation

- ❑ Our proposal targets for permutation invariance and better effectiveness in model exact match identification.

- ❑ Our proposal can be potentially extended to other types of optimization problems if they can be represented as graphs, such as Mixed Integer Linear Programming (MILP), Quadratic Programming (QP) and Quadratically Constrained Quadratic Programming (QCQP).

Proposed Evaluation Method

□ Our proposed evaluation method for LPWP of LLMs is based on **graph edit distance**, consisting of three steps:

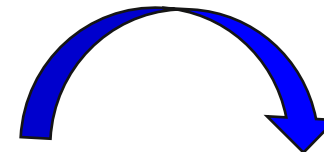
- **Step 1: LP General Form Conversion**
- **Step 2: Graph Representation Conversion**
- **Step 3: Graph Edit Distance Calculation**

For generality, we consider the following form of LP:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \ell^s \leq \mathbf{A}\mathbf{x} \leq \mathbf{u}^s \\ & \ell^v \leq \mathbf{x} \leq \mathbf{u}^v \end{aligned}$$

A: constraint matrix
c: cost vector
x: decision variable
 ℓ^x / u^x : lower/upper bound of variables
 ℓ^s / u^s : lower/upper bound of constraints

An example



Problem Description
Ben is growing apples and pears on his orchard. He has 50 acres available on which he must grow a minimum of 5 acres of apples and a minimum of 10 acres of pears to meet demands. The profit per apple is \$2 and the profit per pear is \$4. He prefers to grow more pears than apples but limitations in his workforce allow him to grow at most twice the amount of pears as apples. How many of each fruit should Ben grow in order to maximize his profit?

Variables: x, y
Maximize: 2x + 4y
Subject to:
 x + y ≤ 50
 y ≤ 2x
 x, y ≥ 0

$$\begin{aligned} \min \quad & -2x - 4y \\ \text{s.t.} \quad & \begin{bmatrix} 1 & 1 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 50 \\ 0 \end{bmatrix} \\ & \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned}$$

Proposed Evaluation Method

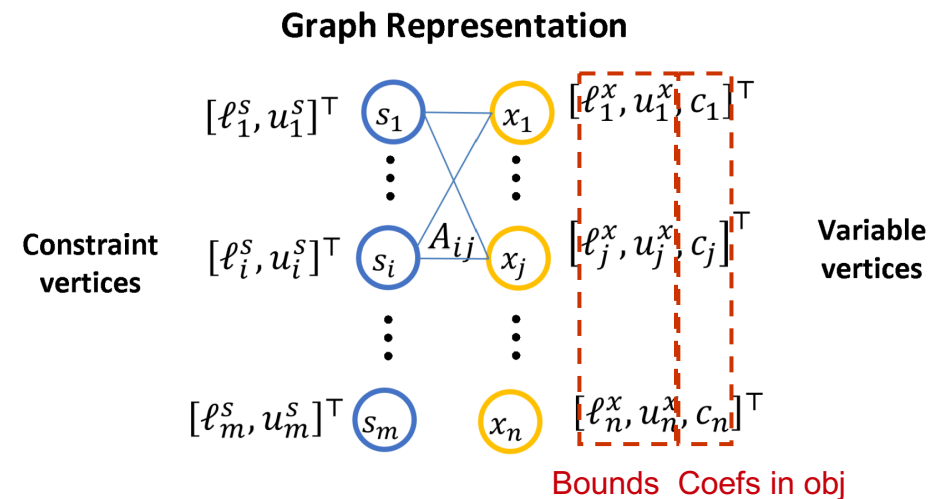
□ Our proposed evaluation method for LPWP of LLMs is based on **graph edit distance**, consisting of three steps:

- **Step 1:** LP General Form Conversion
- **Step 2:** Graph Representation Conversion
- **Step 3:** Graph Edit Distance Calculation

We choose to represent the LP problem as **an attributed bipartite graph**:

* Attributes include: variable coefficients, bounds of constraints.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \ell^s \leq \mathbf{A}\mathbf{x} \leq \mathbf{u}^s \\ & \ell^x \leq \mathbf{x} \leq \mathbf{u}^x \end{aligned}$$



Proposed Evaluation Method

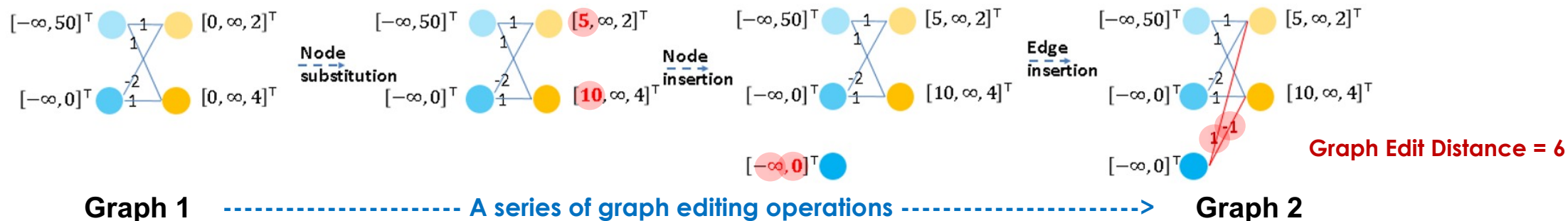
□ Our proposed evaluation method for LPWP of LLMs is based on **graph edit distance**, consisting of three steps:

- **Step 1:** LP General Form Conversion
- **Step 2:** Graph Representation Conversion
- **Step 3:** Graph Edit Distance Calculation

Graph Edit Distance (GED) is defined by the minimum-cost sequence of basic edit operations to transform one graph into another by means of insertion, deletion and substitution of vertices and/or edges.

In practice, we adopt well-established algorithm for GED computing. For editing cost, the operation of each number in the hypothesis graph requires **1 unit cost**.

- Node insertion and deletion can be broken down into a set of smaller substitution steps.
- The cost of edge insertion/deletion = the cost of edge substitution, as each edge has only one attribute.



Experiments

Dataset:

- **NL4OPT** – the first-ever LPWP dataset (713 training, 99 validation, and 289 testing).

Language Models:

- Llama-2-Chat (13B)
- Code-Llama-Instruct (34B)
- Llama-2-Chat (70B)
- Llama-2-SFT (13B)

Llama is open-sourced, enabling us to compare the effectiveness of supervised tuning for LPWP.

Llama comes in different sizes, which allows us to compare across the dimension of model scale.

Human Evaluation:



3 OR experts



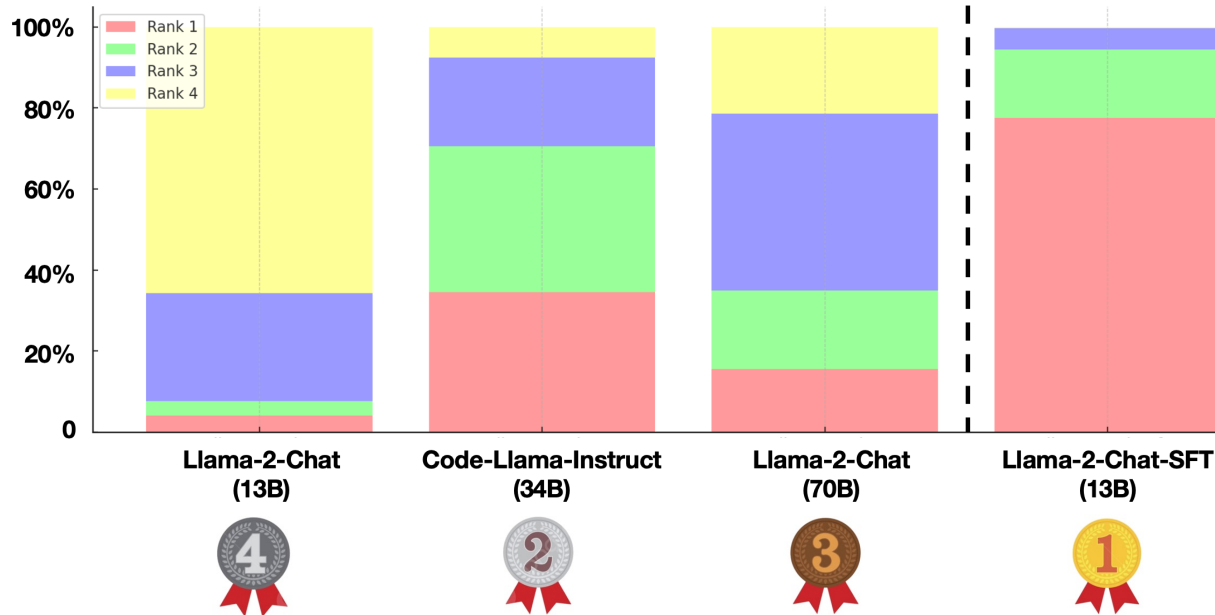
Order 4 LLM's prediction by quality

Example

llama-13b-sft > code-llama-34b > llama-70b > llama-13b

Experiments

Human judgements on the NL4OPT test set



Correlation with human judgements

Metrics	C-Match	F-Match
Execution	9 / 289	716 / 1734
Canonical	64 / 289	1336 / 1734
Ours	178 / 289	1641 / 1734

- **C-Match:** coarse-grained ranking match.
- **F-Match:** fine-grained ranking match.

Average scores of auto evaluation metrics on the NL4OPT test set

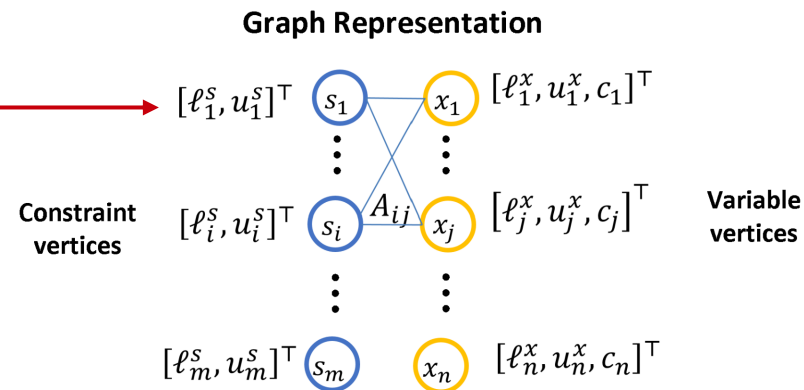
Language Models	Execution(↑)	Canonical(↑)	Ours(↓)
Llama-2-Chat (13B)	0.07 (4)	0.24 (4)	0.52 (4)
Code-Llama-Instruct (34B)	0.35 (2)	0.54 (2)	0.25 (2)
Llama-2-Chat (70B)	0.21 (3)	0.31 (3)	0.41 (3)
Llama-2-Chat-SFT (13B)	0.53 (1)	0.64 (1)	0.14 (1)



Distinguishing Features

- A more accurate and robust evaluation method for auto-formulating Optimization Modeling with Large Language Models (LLMs), addressing pitfalls of prior approaches through:
 - Permutation invariance.
 - Better identification of model exact match.
- Support additional integration of available information in other modalities (e.g., textual explanations)

Optimization Model	
Variables:	
x (acres of apples)	Variables
y (acres of pears)	
Maximize: $2x + 4y$ (profit)	Objective
Subject to:	
$x + y \leq 50$ (land constraint)	
$x \geq 5$ (apple minimum)	Constraints
$y \geq 10$ (pear minimum)	
$x \leq y \leq 2x$ (pear to apple ratio)	



Cost function can be defined as conventional text matching metric (e.g., BERT-Score)

- Support error traceback.
 - Instead of merely obtaining the score indicating the degree of difference between optimization models, we can also list where are the mismatches based on the graph edit operations throughout the GED computing.

Thank You!

Canonical Evaluation

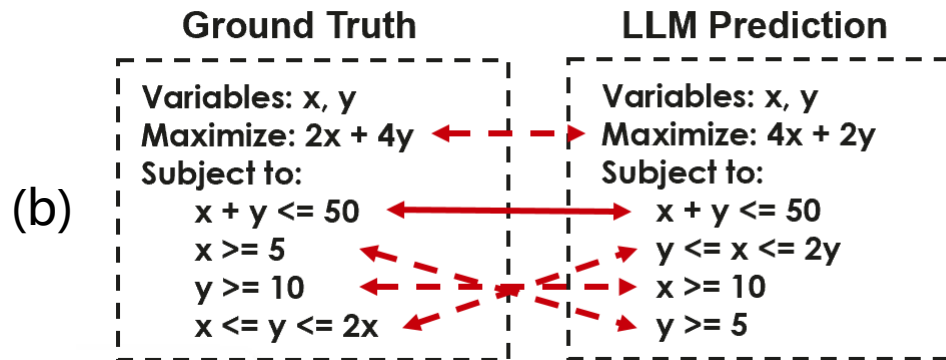
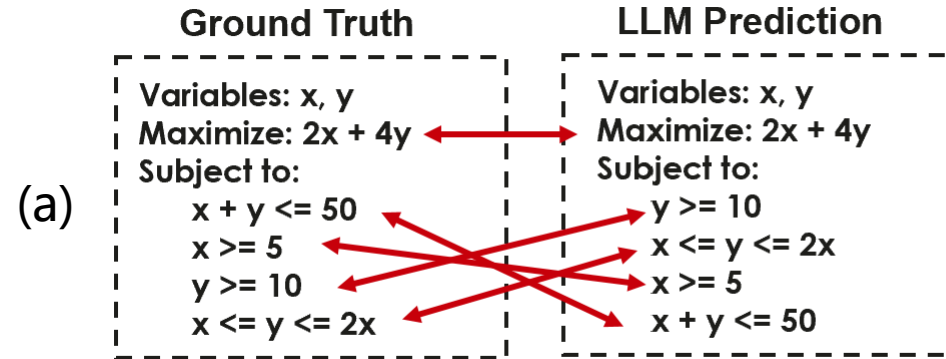
- **Features:**

- > It is based on the declaration-level* matching between hypothesis and reference model:

$$Acc = 1 - \frac{\min(FP_i + FN_i, D_i)}{D_i}$$

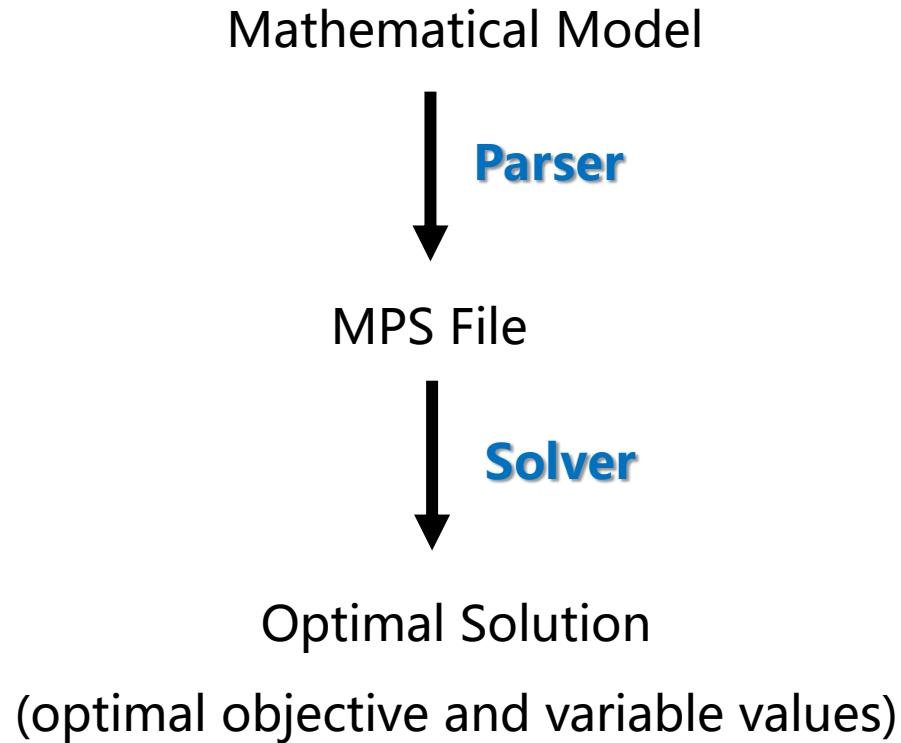
For a given problem i :

- D_i : the number of actual declarations in the ground-truth model.
- FP_i : the number of declarations in prediction not matching with any of the actual declarations.
- FN_i : the number of excess actual declarations not matching with any of the predicted declarations.



* A declaration is, by definition, the representation of either an optimization objective or a constraint.

Executable Evaluation



Scenario 1:

Ground Truth	LLM Prediction
<p>Variables: x, y Maximize: $2x + 4y$ Subject to: $x + y \leq 50$ $x \geq 5$ $y \geq 10$ $x \leq y \leq 2x$</p>	<p>Variables: x, y Maximize: $2x + 4y$ Subject to: $x + y \leq 50$ $y \leq 2x$ $x, y \geq 0$</p>
<p>Variables: $x = 16.6, y = 33.3$ Objective: $2x + 4y = 50$</p>	<p>Variables: $x = 16.6, y = 33.3$ Objective: $2x + 4y = 50$</p>



Scenario 2:

<p>Objective: Infeasible</p>	<p>Objective: Infeasible</p>
----------------------------------	----------------------------------

