Milestone 1: Project Inception

AI-Assisted Symbolic Optimization for Strategic Facility Network Design

CSC 5382 – AI for Digital Transformation

By: Ayoub Akarkouz <78987>

Table of contents

01
Context & Framing

Strategic Facility Location

Why Facility Location Matters

- Warehouse placement
- Data center deployment
- EV charging planning
- Telecom infrastructure
- Healthcare network design

Decisions impact:

- Capital expenditure
- Service latency
- Operational cost

UFLP is NP-hard

Core Business Pain Point

Modern solvers (CPLEX, Gurobi, OR-Tools) are powerful.

‼ But the bottleneck is:

- Manual mathematical modeling
- OR expertise dependency
- High formulation cost
- Slow iteration cycles
- Risk of modeling errors

In enterprise contexts: modeling time > solving time.

Proposed Solution: LLM-Assisted Symbolic Modeling

Instead of:

Human writes MILP model manually

We use:

LLM generates solver-compatible optimization code

Pipeline: (inspired by LLMFP)

1. Structured Input
2. LLM generates MILP code
3. Classical solver executes
4. Solver verifies optimality

ML Framing & Archetype

Project Archetype:

- Neuro-Symbolic Optimization
- LLM-as-Modeler
- Verification-Guided AI
- Code Generation for Optimization

> Correctness guaranteed by solver.

02

Problem & Data

UFLP Formulation

Minimize:

$$min \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij}$$

Subject to:

$$min \sum_{i \in F} x_{ij} = 1 \,, \forall j \epsilon \; C$$

$$x_{ij} \; < \; y_i \,, \forall i \epsilon \; F \,, j \epsilon \; C$$

UFLP Formulation

OR-Library: Curated by John E. Beasley

15 benchmark instances:

- cap71–cap74
- cap101–cap104
- cap131–cap134
- capa, capb, capc

Includes known optimal values (uncapopt.txt)

Research-grade benchmark used in:

- ABPEA
- Hybrid GA-SA

Input / Output Structure

Input:

- m facilities
- n customers
- Fixed cost vector f_i
- Allocation cost matrix c_{ij}

Output:

- Binary facility decisions y_i
- Assignment matrix x_{ij}
- Objective value

Ground truth:

- Exact optimal objective available -> enables 0% gap evaluation.

03

Literature

Classical UFLP Research

Approaches:
- Exact MILP
- Lagrangian relaxation
- Approximation algorithms
- Metaheuristics (ABPEA, GA-SA)

Assumption:
- Model already exists.

Focus: Solution algorithms (not model generation)

ML for Combinatorial Optimization

Methods:

- Neural combinatorial optimization
- Learned branching
- Policy gradient search
- Parameter tuning

Limitation:

- Improve solving
- Do not automate model formalization.

LLM-Based Planning

Key Work: Planning Anything with Rigor

LLMFP paradigm:

- LLM generates symbolic program
- Solver executes
- Solver verifies
- Iterative refinement

Other works:

- LEO
- LLM + Optimization studies

Mostly evaluated on toy planning tasks.

Identified Research Gap

No prior work:

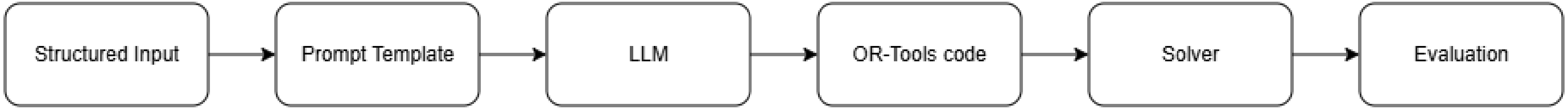
- Evaluates LLM-generated MILP models
- On OR-Library UFLP
- With exact optimal objective benchmarking

Contribution:

First systematic evaluation of LLM-based symbolic optimization generation on classical OR benchmarks.

04
Architecture Overview

Architecture Overview



Pipeline:

- OR-Library parser
- Prompt template
- LLM generates OR-Tools MILP
- Automated execution
- CBC solver
- Gap evaluation vs ground truth

(Verification guided design)

Deterministic Baseline

Canonical MILP in OR-Tools (CBC backend)

Components:

- OR-Library parser
- uncapopt.txt lookup
- MILP formulation
- Gap computation

Baseline Validation Results

Instance	m	n	Solver objective	Best known	Gap (%)	Runtime (s)
cap71	16	50	932615.750	932615.750	0.0	0.057
cap101	25	50	796648.438	796648.437	0.0	0.084
cap131	50	50	793439.563	793439.562	0.0	0.196

Confirms:

- Correct parsing
- Correct formulation
- Reliable benchmark

05

Evaluation

Metrics

Technical Metrics

- Optimality Gap (%)
- Feasibility Rate
- Code Execution Success Rate
- Constraint Satisfaction
- Solver Runtime

Business Metrics

- Modeling time reduction
- Manual intervention reduction
- Scenario iteration speed
- Scalability across instances

➢ Exact ground truth enables strict benchmarking.

➢ Focus: modeling automation impact.

Research Validity

(Minimalistic Research-grade protocol)

- Reproducible notebook
- Raw OR-Library data included
- Deterministic solver baseline
- Exact optimal value comparison
- Prompt stability testing

06

Contribution

Contribution

Scientific Contribution

- Systematic evaluation of LLM-generated MILP models
- Verification against exact OR benchmarks
- Clear separation of generation vs optimization
- Business-oriented modeling automation framework

Intersection of:

Neuro-symbolic AI
Optimization modeling
LLM code generation
Classical OR benchmarking

Limitations

- Prompt sensitivity
- Code formatting errors
- Large-instance scalability

Future

- Iterative repair loop
- Cross-dataset evaluation
- Extension to capacitated FLP
- Multi-objective optimization

Conclusion

Modeling is the real bottleneck in enterprise optimization.

We demonstrate that LLMs can automate formal optimization modeling while preserving solver-grade correctness.

Validated on an OR-Library UFLP with exact optimal benchmarking.

Thanks!

Do you have any questions?

a.akarkouz@aui.ma