

# OptiMUS: Scalable Optimization Modeling with (MI)LP Solvers and LLMs\*

**Presented by** Ravi B. Sojitra @ Stanford's [REFORM Reading Group](#)

**Authors:** Ali AhmadiTeshnizi, Wenzhi Gao, Madeleine Udell

\*Actually, I am presenting the [newer version](#) by Ali AhmadiTeshnizi, Wenzhi Gao, Herman Brunborg, Shayan Talaei, Connor Lawless, and Madeleine Udell.

# Summary: AI + hard coded workflow goes far.

To increase adoption of optimization modeling, they want to remove frictions related to tasks that are cycled through during problem formulation. For example,

1. Identifying parameters, constraints, and objectives from documentation and data;
2. Constructing formal mathematical models;
3. Reformulating the models so they may be solved efficiently;
4. Selecting solvers; and
5. Generating code.

To address these, they contribute a framework for AI agentic systems and evaluation data.

OptiMUS 0.3 is an example instantiation of this framework.

# Real Example: McDonald's China used an MILP.

Tang and Wang et al. (2025) designed a supply chain network to reduce costs and emissions by formulating a Mixed Integer Linear Program (MILP) that commercial solvers can handle.

  
<https://pubsonline.informs.org/journal/ijaa>

**INFORMS JOURNAL ON APPLIED ANALYTICS**  
Vol. 55, No. 1, January–February 2025, pp. 36–47  
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

  
THE FRANZ EDELMAN AWARD  
*Achievement in Operations Research*

## McDonald's China Adopts Operations Research for Network Design

Shouwei Tang,<sup>a,\*</sup> Lei Wang,<sup>a,\*</sup> Yun Shi,<sup>b</sup> Andy Li,<sup>b</sup> Kevin Lin,<sup>c</sup> Chen Xiang,<sup>a</sup> Sophia Niu,<sup>a</sup> Shaofeng Zhou,<sup>a</sup> Ming Liu,<sup>c</sup> Hank Tang<sup>c</sup>

<sup>a</sup>Optimization Analytics Technology Pte. Ltd., Singapore 199591; <sup>b</sup>McDonald's China, Shanghai 201103, China; <sup>c</sup>Xianhui Logistics, Shanghai 201801, China

\*Corresponding author

Contact: shouwei.tang@oatSERVICE.com (ST); wendy.wang@oatSERVICE.com,  <http://orcid.org/0009-0006-3838-4953> (LW); jim.shi@cn.mcd.com (YS); andy.li@cn.mcd.com (AL); kevin.lin@havi-cn.com (KL); chen.xiang@oatSERVICE.com (CX); sophia.niu@oatSERVICE.com (SN); shaofeng.zhou@oatSERVICE.com (SZ); ming.liu@havi-cn.com (ML); hank.tang@havi-cn.com (HT)

# Real Example: McDonald's China used an MILP.

Tang and Wang et al. (2025) designed a supply chain network to reduce costs and emissions by formulating a Mixed Integer Linear Program (MILP) that commercial solvers can handle.

Description	
<b>Sets and Indices</b>	
$I$	Set of all potential sites and existing DC locations with index $i$
$J$	Set of all existing DCs with index $j$ , a subset of $I$
$F$	Set of factories (suppliers) with index $f$
$D$	Set of demand points with index $d$
$SKU$	Set of SKU groups (aggregated SKUs) with index $s$
$T$	Set of time periods (years) with index $t$
$P$	Set of storage areas in a DC with index $p$
$K$	Set of integer numbers starting from one, with index $k$
<b>Parameters<sup>a</sup></b>	
$a_i$	One-time building investment cost for DC $i$ in RMB yuan
$\beta_i$	Fixed operational cost per time period for DC $i$ in RMB yuan
$\gamma_{d,i,s}$	Demand of demand point $d$ per time period $t$ for SKU group $s$ in cases
$\delta_{f,i,s}$	Supply capacity of factory $f$ per time period $t$ for each SKU group $s$ in cases
$\epsilon_s$	Storage area of SKU group $s$ in cases
$\zeta_{i,p}$	DC volume capacity of storage area $p$ in DC $i$ per time period $t$ in cases
$\eta_p$	Unit storage cost per day for storage area $p$ in RMB yuan
$\theta_{i,s}$	Per-unit handling cost for SKU $s$ in DC $i$ in RMB yuan
$u_s$	Current safety stock cost in number of days of demand for SKU $s$
$\kappa_s$	Current cycle stock level in number of days of demand for SKU $s$
$\lambda_{f,i,s}$	Unit inbound transportation cost from factory $f$ to DC $i$ for SKU $s$ in RMB yuan
$\mu_{d,s}$	Unit outbound transportation cost from DC $i$ to demand point $d$ for SKU $s$ in RMB yuan
$v$	Number of days in a time period
$\xi$	Number of DCs in baseline setting (i.e., number of McDonald's China DCs in 2020)
$\pi_{d,i,t}$	The ratio (percentage) between SKU demand volume and total demand volume for each demand point $d$ , time period $t$ , and SKU $s$
<b>Decision variables</b>	
$x_{i,t}$	Binary variable, one if DC $i$ will be opened at beginning of time period $t$ , and zero otherwise
$y_{i,t}$	Binary variable, one if DC $i$ will be closed at beginning of time period $t$ , and zero otherwise
$z_{i,t}$	Binary variable, one if DC $i$ exists during time period $t$ , and zero otherwise
$n_t$	Integer variable, total number of DCs
$m_{k,t}$	Binary variable, one if the number of DCs in time period $t$ is $k$
$if_{f,i,s,t}$	Nonnegative continuous variable, inbound flow volume from factory $f$ to DC $i$ of SKU $s$ in year $t$
$of_{i,d,t}$	Nonnegative continuous variable, outbound flow volume from DC $i$ to demand point $d$ of SKU $s$ in year $t$
$ss_{s,t}$	Nonnegative continuous variable, total safety stock volume for SKU $s$ in year $t$
$cs_{s,t}$	Nonnegative continuous variable, total cycle stock volume for SKU $s$ in year $t$

<sup>a</sup>The exchange rate from USD to RMB yuan is 1:7.12 as of this writing.

## Objective Function

The mathematical model's objective function is to minimize the total cost, which comprises the following components:

$$\begin{aligned}
 z_1 &= \sum_{f \in F, i \in I, s \in SKU, t \in T} if_{f,i,s,t} * \lambda_{f,i,s} \\
 z_2 &= \sum_{i \in I, d \in D, s \in SKU, t \in T} of_{i,d,t} * \mu_{i,d,s} * \pi_{d,t,s} \\
 z_3 &= \sum_{f \in F, i \in I, s \in SKU, t \in T} if_{f,i,s,t} * \theta_{i,s} \\
 z_4 &= \sum_{s \in SKU, t \in T} (ss_{s,t} + cs_{s,t}) * \eta_{\xi_s} * v \\
 z_5 &= \sum_{i \in I, t \in T} \alpha_i * x_{i,t} \\
 z_6 &= \sum_{i \in I, t \in T} \beta_i * z_{i,t}.
 \end{aligned}$$

The objective function is the amalgamation of the components above:

$$z = z_1 + z_2 + z_3 + z_4 + z_5 + z_6.$$

## Constraints

$$\sum_i x_{i,t} \leq 1 \quad \forall i \in I \quad (A.1)$$

$$z_{i,t} = \sum_{n \in T, n \leq t} (x_{i,n} - y_{i,n}) \quad \forall i \in I, t \in T \quad (A.2)$$

$$\sum_{t \in T, t \neq 1} x_{j,t} = 0 \quad \forall j \in J \quad (A.3)$$

$$\sum_{f \in F, s \in SKU} if_{f,i,s,t} \leq \text{Big\_}M * z_{i,t} \quad \forall i \in I, t \in T \quad (A.4)$$

$$\sum_j of_{j,i,s,t} = \sum_d of_{i,d,t} * \pi_{d,t,s} \quad \forall i, s, t, \quad \forall i \in I, t \in T, \quad s \in SKU \quad (A.5)$$

$$\sum_{i \in I} of_{i,d,t} = \sum_s \gamma_{d,i,t} \quad \forall d \in D, t \in T \quad (A.6)$$

$$\sum_{i \in I} if_{f,i,s,t} \leq \delta_{f,t,s} \quad \forall f \in F, s \in SKU, t \in T \quad (A.7)$$

$$\sum_{s \in SKU | \epsilon_s = p, i \in F} of_{f,i,s,t} \leq \zeta_{i,p} \quad \forall i \in I, p \in P, t \in T \quad (A.8)$$

$$\sum_{i \in I} z_{i,t} = n_t \quad \forall t \in T \quad (A.9)$$

$$\sum_{k \in K} k * m_{k,t} = n_t \quad \forall t \in T \quad (A.10)$$

$$\sum_{k \in K} m_{k,t} = 1 \quad \forall t \in T \quad (A.11)$$

$$cs_{s,t} = \sum_{d \in D} \kappa_s * \gamma_{d,i,t} / v \quad \forall s \in SKU, t \in t \quad (A.12)$$

$$ss_{s,t} * \text{sqrt}(\xi) = \sum_{k \in K} \text{sqrt}(k) * m_{k,t} \sum_d I_s * \gamma_{d,t,s} / v \quad \forall s \in SKU, t \in T. \quad (A.13)$$

# How do you arrive at good problem formulations?

Assume that this is a good formulation. What did the authors do to arrive at this formulation?

## Why these variables?

Description	
<b>Sets and indices</b>	
$I$	Set of all potential sites and existing DC locations with index $i$
$J$	Set of all existing DCs with index $j$ , a subset of $I$
$F$	Set of factories (suppliers) with index $f$
$D$	Set of demand points with index $d$
$SKU$	Set of SKU groups (aggregated SKUs) with index $s$
$T$	Set of time periods (years) with index $t$
$P$	Set of storage areas in a DC with index $p$
$K$	Set of integer numbers starting from one, with index $k$
<b>Parameters<sup>a</sup></b>	
$a_i$	One-time building investment cost for DC $i$ in RMB yuan
$\beta_i$	Fixed operational cost per time period for DC $i$ in RMB yuan
$\gamma_{d,i,s}$	Demand of demand point $d$ per time period $t$ for SKU group $s$ in cases
$\delta_{f,i,s}$	Supply capacity of factory $f$ per time period $t$ for each SKU group $s$ in cases
$\epsilon_s$	Storage area of SKU group $s$ in cases
$\zeta_{i,p}$	DC volume capacity of storage area $p$ in DC $i$ per time period $t$ in cases
$\eta_p$	Unit storage cost per day for storage area $p$ in RMB yuan
$\theta_{i,s}$	Per-unit handling cost for SKU $s$ in DC $i$ in RMB yuan
$u_s$	Current safety stock cost in number of days of demand for SKU $s$
$\kappa_s$	Current cycle stock level in number of days of demand for SKU $s$
$\lambda_{f,i,s}$	Unit inbound transportation cost from factory $f$ to DC $i$ for SKU $s$ in RMB yuan
$\mu_{d,i,s}$	Unit outbound transportation cost from DC $i$ to demand point $d$ for SKU $s$ in RMB yuan
$v$	Number of days in a time period
$\xi$	Number of DCs in baseline setting (i.e., number of McDonald's China DCs in 2020)
$\pi_{d,i,t}$	The ratio (percentage) between SKU demand volume and total demand volume for each demand point $d$ , time period $t$ , and SKU $s$
<b>Decision variables</b>	
$x_{i,t}$	Binary variable, one if DC $i$ will be opened at beginning of time period $t$ , and zero otherwise
$y_{i,t}$	Binary variable, one if DC $i$ will be closed at beginning of time period $t$ , and zero otherwise
$z_{i,t}$	Binary variable, one if DC $i$ exists during time period $t$ , and zero otherwise
$n_t$	Integer variable, total number of DCs
$m_{k,t}$	Binary variable, one if the number of DCs in time period $t$ is $k$
$if_{f,i,s,t}$	Nonnegative continuous variable, inbound flow volume from factory $f$ to DC $i$ of SKU $s$ in year $t$
$of_{i,d,t}$	Nonnegative continuous variable, outbound flow volume from DC $i$ to demand point $d$ of SKU $s$ in year $t$
$ss_{s,t}$	Nonnegative continuous variable, total safety stock volume for SKU $s$ in year $t$
$cs_{s,t}$	Nonnegative continuous variable, total cycle stock volume for SKU $s$ in year $t$

<sup>a</sup>The exchange rate from USD to RMB yuan is 1:7.12 as of this writing.

## Why this sum of costs?

### Objective Function

The mathematical model's objective function is to minimize the total cost, which comprises the following components:

$$\begin{aligned}
 z_1 &= \sum_{f \in F, i \in I, s \in SKU, t \in T} if_{f,i,s,t} * \lambda_{f,i,s} \\
 z_2 &= \sum_{i \in I, d \in D, s \in SKU, t \in T} of_{i,d,t} * \mu_{i,d,s} * \pi_{d,t,s} \\
 z_3 &= \sum_{f \in F, i \in I, s \in SKU, t \in T} if_{f,i,s,t} * \theta_{i,s} \\
 z_4 &= \sum_{s \in SKU, t \in T} (ss_{s,t} + cs_{s,t}) * \eta_{\xi_s} * v \\
 z_5 &= \sum_{i \in I, t \in T} \alpha_i * x_{i,t} \\
 z_6 &= \sum_{i \in I, t \in T} \beta_i * z_{i,t}
 \end{aligned}$$

The objective function is the amalgamation of the components above:

$$Z = z_1 + z_2 + z_3 + z_4 + z_5 + z_6.$$

## Why these constraints?

### Constraints

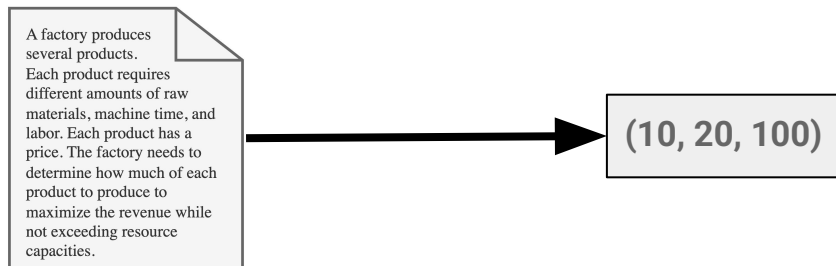
$$\begin{aligned}
 \sum_i x_{i,t} &\leq 1 \quad \forall i \in I & (A.1) \\
 z_{i,t} &= \sum_{n \in T, n \leq t} (x_{i,n} - y_{i,n}) \quad \forall i \in I, t \in T & (A.2) \\
 \sum_{t \in T, t \neq 1} x_{j,t} &= 0 \quad \forall j \in J & (A.3) \\
 \sum_{f \in F, s \in SKU} if_{f,i,s,t} &\leq \text{Big\_}M * z_{i,t} \quad \forall i \in I, t \in T & (A.4) \\
 \sum_j if_{f,i,s,t} &= \sum_d of_{i,d,t} * \pi_{d,t,s} \quad \forall i, s, t, \forall i \in I, t \in T, s \in SKU & (A.5) \\
 \sum_{i \in I} of_{i,d,t} &= \sum_s \gamma_{d,i,s} \quad \forall d \in D, t \in T & (A.6) \\
 \sum_{i \in I} if_{f,i,s,t} &\leq \delta_{f,i,s} \quad \forall f \in F, s \in SKU, t \in T & (A.7) \\
 \sum_{s \in SKU | \epsilon_s = p, i \in F} of_{f,i,s,t} &\leq \zeta_{i,p} \quad \forall i \in I, p \in P, t \in T & (A.8) \\
 \sum_{i \in I} z_{i,t} &= n_t \quad \forall t \in T & (A.9) \\
 \sum_{k \in K} k * m_{k,t} &= n_t \quad \forall t \in T & (A.10) \\
 \sum_{k \in K} m_{k,t} &= 1 \quad \forall t \in T & (A.11) \\
 cs_{s,t} &= \sum_{d \in D} \kappa_s * \gamma_{d,i,s} / v \quad \forall s \in SKU, t \in t & (A.12) \\
 ss_{s,t} * \text{sqrt}(\xi) &= \sum_{k \in K} \text{sqrt}(k) * m_{k,t} \sum_d I_s * \gamma_{d,i,s} / v \quad \forall s \in SKU, t \in T. & (A.13)
 \end{aligned}$$

# There are many pain points in MILP modeling.

- Identifying decision variables and parameters in documentation, data (e.g. dimensions)
- Mathematical problem formulation (e.g. constraints, objective function)
- Performance/efficiency (e.g. adding slack, relaxations)
- Algorithms (e.g. column or row generation)
- Programming (the actual code)
- Sensitivity analysis, visualization (to understand the solution, robustness)
- Debugging (e.g. Identify, assess, diagnose infeasibility or redundant constraints)
- Iteration: What if I reroute...what if I buy more.. What if I use supplier X instead?
- (i) verifying the irreducible infeasible subset, (ii) activating/deactivating constraints, and (iii) relaxing constraints by adding slack variables.

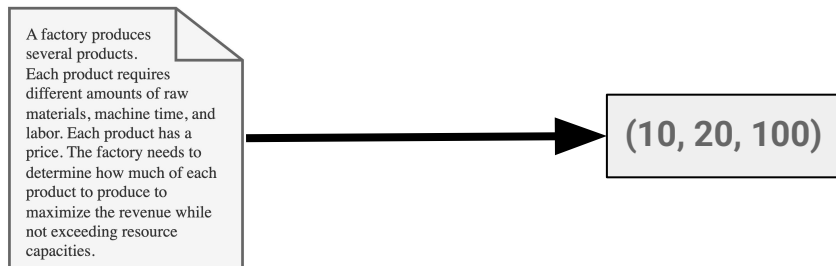
# OptiMUS maps NLP goals to quantitative choices.

**Vision:** Provide a business problem in natural language and receive optimal solutions.



# OptiMUS maps NLP goals to quantitative choices.

**Vision:** Provide a business problem in natural language and receive optimal solutions.



**The OptiMUS architecture has four ingredients.**

- States
- Components
- Error Correction
- Structure Detection



# 1. OptiMUS constructs states with useful details.

A factory produces several products. Each product requires different amounts of raw materials, machine time, and labor. Each product has a price. The factory needs to determine how much of each product to produce to maximize the revenue while not exceeding resource capacities.

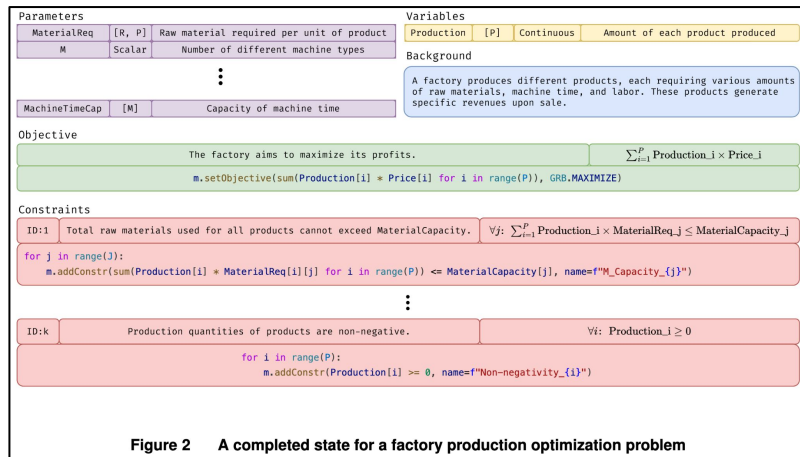
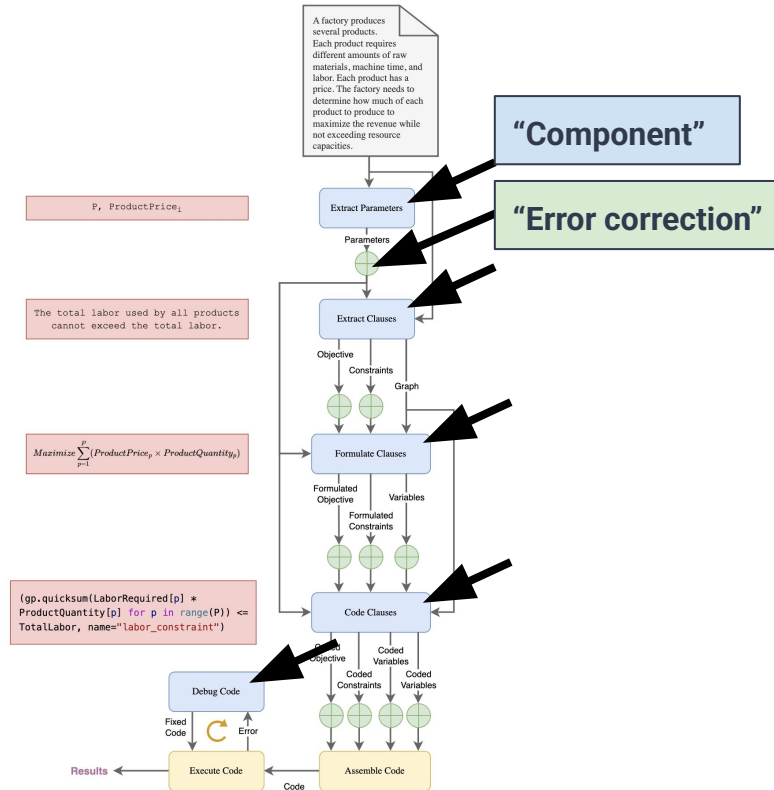


Figure 2 A completed state for a factory production optimization problem

1. **Background/Context**
2. **Parameters:** Symbol, shape, and definition.
3. **(Decision) variables:** Symbol, shape, definition, and type (binary, continuous, or integer)
4. **Clauses (objective function, constraints)** in written in English, LaTeX, AND code.
5. **Connection graph:** Matrix with rows representing clauses and columns variables.

## 2. OptiMUS has 5 components (workflow steps).



# 3. Its Error Correction includes reformulation.

They use three tricks.

**1. Reflective prompts:** Prompt with prior LLM output asking LLM to correct it if necessary.

**2. Confidence based feedback:** LLMs rates its confidence (1-5); if <5, request feedback.

**3. Code debugging:** LLM is allowed to make changes it sees fit to make the code run.

# 3. Its Error Correction includes reformulation.

They use three tricks.

**1. Reflective prompts:** Prompt with prior LLM output asking LLM to correct it if necessary.

**2. Confidence based feedback:** LLMs rates its confidence (1-5); if <5, request feedback.

**3. Code debugging:** LLM is allowed to make changes it sees fit to make the code run.

Are units the same for both sides of this constraint?

$$(p_a + x_a) \cdot d_a \cdot (1 + e_a \cdot \frac{x_a}{p_a}) \leq m_a, \forall a \in A?$$

... Left-hand side (LHS):

- $(p_a + x_a)$  represents the new price for article  $a$ , which is in euros (€).
- $d_a$  represents the sales forecast (demand) for article  $a$  for the next twelve months at the current price, which is in units of the article.
- $(1 + e_a \cdot \frac{x_a}{p_a})$  is a unitless factor ...

Therefore, the unit of the left-hand side is: **euros (€) × units of the article**

Right-hand side (RHS):

- $m_a$  represents the maximum production volume for article  $a$ , which is in units of the article (e.g., number of shirts, pants, etc.).

The unit of the right-hand side is: **units of the article**

... this inconsistency suggests an error in the formulation of Constraint 5. To correct this, we should ... here is the corrected constraint:

$$d_a \cdot (1 + e_a \cdot \frac{x_a}{p_a}) \leq m_a, \forall a \in A$$

Figure 3 OptiMUS-0.3 can fix its constraint modeling errors when prompted “Are units the same for both sides of C?”

# 3. Its Error Correction includes reformulation.

They use three tricks.

**1. Reflective prompts:** Prompt with prior LLM output asking LLM to correct it if necessary.

**2. Confidence based feedback:** LLMs rates its confidence (1-5); if <5, request feedback.

**3. Code debugging:** LLM is allowed to make changes it sees fit to make the code run.

Consider this potential constraint: No more than MaxPriceChange fraction of prices can change

Is it actually a constraint? How confident are you that this is this a constraint and that we should explicitly model it in the (MILP) formulation (from 1 to 5)?

Llama-3-70B

The parameter MaxPriceChange is defined as the "maximum fraction of prices that can be changed." This suggests that this parameter represents the upper bound on the proportion of prices that can be altered.

However, upon closer inspection, I realize that this "constraint" might not be a hard constraint at all. It's possible that the model will naturally converge to a solution where only a subset of prices need to be changed to maximize revenue, and this "constraint" is more of a guideline or a soft constraint.

Furthermore, if we were to explicitly model this as a constraint, it would require introducing binary variables to indicate whether a price is changed or not, and then counting the number of binary variables that are 1 to ensure that it doesn't exceed the MaxPriceChange fraction. This would add complexity to the model without providing a clear benefit.

Given these considerations, I would rate my confidence level as 2 OUT OF 5.

Your colleague is suggesting that the following constraint should be added to the list:  
No more than MaxPriceChange fraction of prices can change  
Here is its explanation:  
The parameter MaxPriceChange ... Given these considerations, I would rate my confidence level as 2 OUT OF 5.

Do you want to keep this constraint?

GPT-4o

The provided parameter, MaxPriceChange, is supposed to represent the fraction of prices that can change. Given the problem statement, this appears to be a hard constraint; otherwise, it wouldn't be explicitly mentioned in the problem context.

Here are the key points that align with this being a hard constraint:

- The problem statement clearly mentions that no more than a certain fraction of prices can be changed.
- The parameter MaxPriceChange is defined explicitly for this purpose.
- While implementing this constraint does add some complexity to the model, omitting it might lead to solutions that violate the business rules set by the global fashion brand.

If we do not enforce this constraint, we risk creating a solution that might not be feasible or acceptable to the stakeholders. Therefore, the constraint **should indeed be added to the model.**

# 3. Its Error Correction includes reformulation.

They use three tricks.

**1. Reflective prompts:** Prompt with prior LLM output asking LLM to correct it if necessary.

**2. Confidence based feedback:** LLMs rates its confidence (1-5); if <5, request feedback.

**3. Code debugging:** LLM is allowed to make changes it sees fit to make the code run.

### 3. Its Error Correction includes reformulation.

They use three tricks.

**1. Reflective prompts:** Prompt with prior LLM output asking LLM to correct it if necessary.

**2. Confidence based feedback:** LLMs rates its confidence (1-5); if <5, request feedback.

**3. Code debugging:** LLM is allowed to make changes it sees fit to make the code run.

**Table 3 Ablation studies on OptiMUS-0.3**

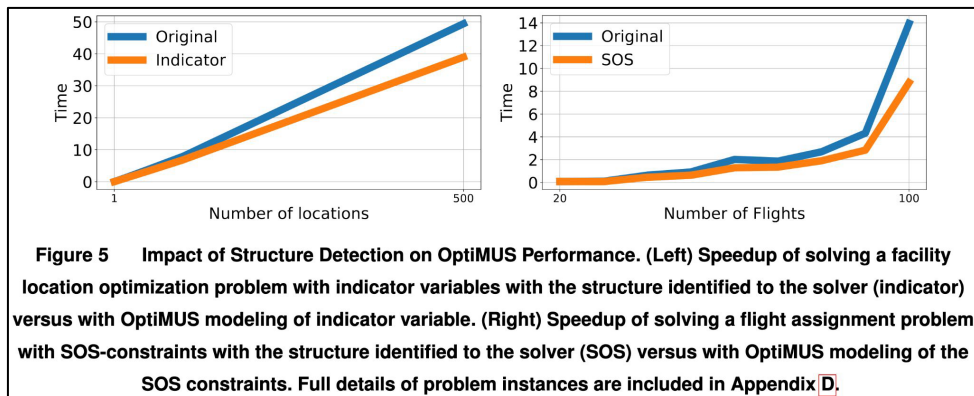
	Easy	Hard
<b>Importance of Different Components</b>		
w/o Debugging	73.2%	26.7%
w/o Extraction EC	86.7%	60.5%
w/o Modeling EC	83.8%	65.7%
w/o LLM Feedback	86.6%	68.4%
<b>OptiMUS-0.3</b>	<b>86.6%</b>	<b>73.7%</b>
<b>Performance with Different LLMs</b>		
LLaMa3.1-70B-Instruct	70.4%	31.5%
<b>GPT4-o</b>	<b>86.6%</b>	<b>73.7%</b>

## 4. OptiMUS maintains a list of structures detected.

It maintains a pool of problem structures that can be solved efficiently with existing solvers.

E.g. For the traveling salesman problem, use Concorde instead of MLP.

Notifies user when their problem structure matches one in the pool & recommends a solver.





# For evaluation, they created the NLP4LP dataset.

Table 1 A comparison on different aspects of complexity for various datasets. The unit for description length is characters			
Dataset	Description Length	Instances (#MILP)	Multi-dimensional Parameters
NL4Opt	$518.0 \pm 110.7$	1101 (0)	×
ComplexOR	$497.1 \pm 247.5$	37 (12)	✓
NLP4LP Easy (Ours)	$507.2 \pm 102.6$	287 (0)	✓
NLP4LP Hard (Ours)	$912.3 \pm 498.2$	68 (18)	✓

355 problems: 22 Dev set problems (12 easy, 10 hard), 332 test set problems (227 easy, 55 hard)

- “Easy” means short problem descriptions, scalar parameters, LPs only.
- “Hard” means long problem descriptions, multidimensional parameters, and both LPs and MILPs.
- Application settings: scheduling, cutting, routing, blending, and packing, and a list of common application domains such as sports, government, retail, agriculture, and energy.

Each problem is accompanied by Code to run the instance and a ground-truth intermediary representation:

- Extracted parameters and targets
- List of clauses of the problem represented in natural language
- LATEX problem statement
- Solution and optimal value

# The current results focus on accuracy, efficiency.

	Easy	Hard
Standard	47.3%	33.2%
Reflexion	53.0%	42.6%
CoE	64.2%	49.2%
OptiMUS-0.2	78.8%	68%
<b>OptiMUS-0.3</b>	<b>86.6%</b>	<b>73.7%</b>

**Three Benchmarks:** Standard Prompting, Chain of Experts (CoE), Reflexion.

- CoE: A Conductor agent selects a sequence of agents to perform tasks (each is given a prompt and appends its reasoning trace); A “reflection” phase runs and fixes issues.
- Reflexion: “General Purpose framework” for “verbal reinforcement”

**Performance Metrics:** Accuracy, compilation error (CE) rate, and runtime error (RE) rate.

- Accurate/correct means that (1) the code executes, (2) the optimal value is correct, AND (3) the optimal solution is correct.

# Questions

**If the goal is to reduce frictions, what are the right outcomes to measure for evaluation?**

- Time an individual needs to arrive at a working, efficient, or correct solution?
- Impact of decisions actually made using the solution?

**Who is the target user?**

- Experts would probably want highly AI to perform highly specific tasks (e.g. parse documents, adding/removing constraints, sensitivity analysis).
- A novice who is considering doing optimization is probably an expert in another area. Does it make more sense to have them use AI to do optimization or to just pay for cloud software tools built for non-technical users?

# Summary: GenAI + hardcoded workflow goes far.

To increase adoption of optimization modeling, they want to remove frictions related to tasks that are cycled through during problem formulation. For example,

1. Identifying parameters, constraints, and objectives from documentation and data;
2. Constructing formal mathematical models;
3. Reformulating the models so they may be solved efficiently;
4. Selecting solvers; and
5. Generating code.

To address these, they contribute a framework for AI agentic systems and evaluation data.

OptiMUS 0.3 is an example instantiation of this framework.

# Other potentially interesting prior/related work

[Diagnosing Infeasible Optimization Problems Using Large Language Models](#)

[Large Language Models for Supply Chain Optimization](#)