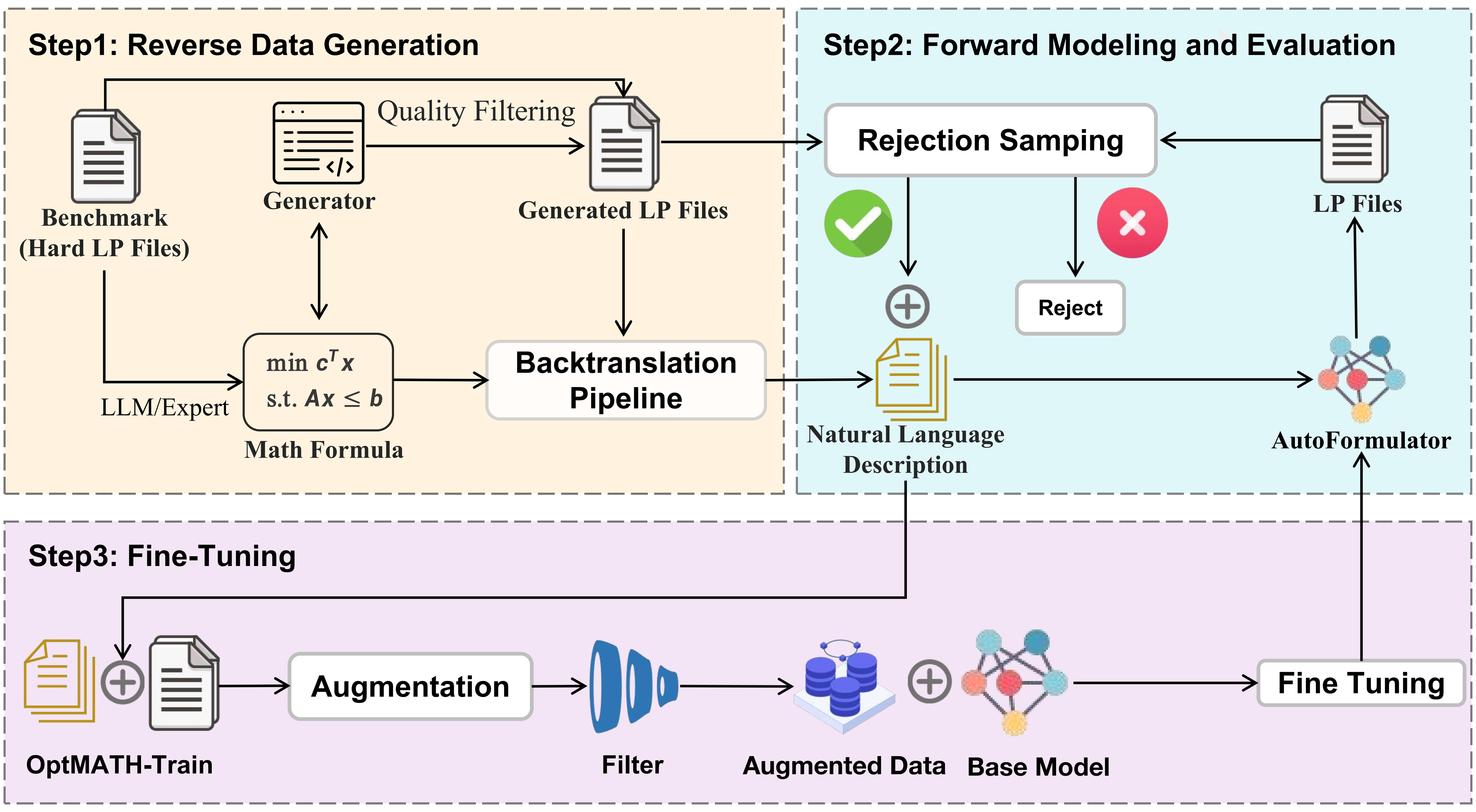
OptMATH: A Scalable Bidirectional Data Synthesis Framework for Optimization Modeling

Hongliang Lu*, Zhonglin Xie*, Yaoyu Wu, Can Ren, Yuxuan Chen, Zaiwen Wen

*Equal contribution Peking University



Dataset Statistics and Performance Results

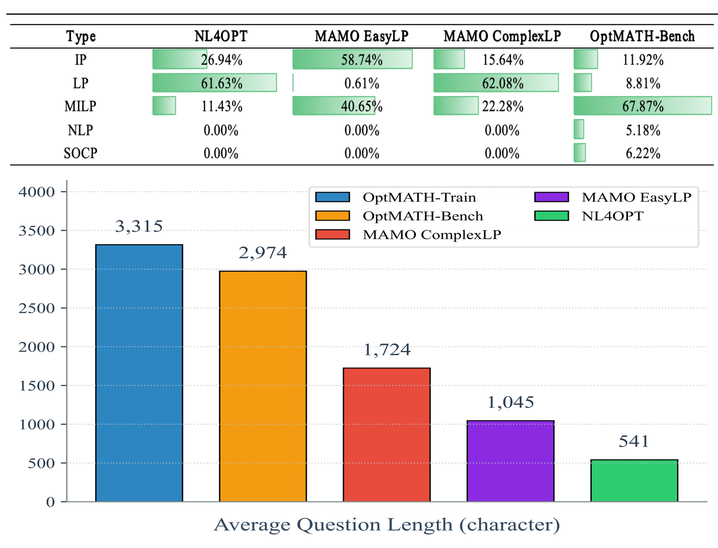


图 1. Problem type and length coverage

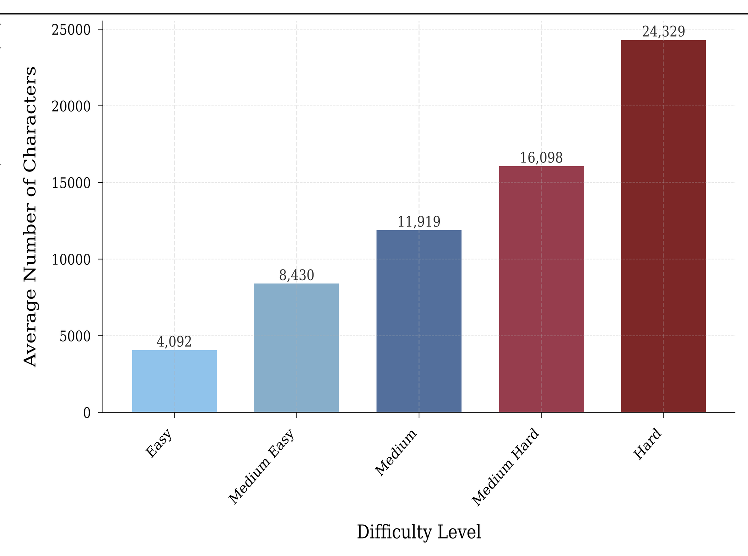


图 2. LP file length by difficulty

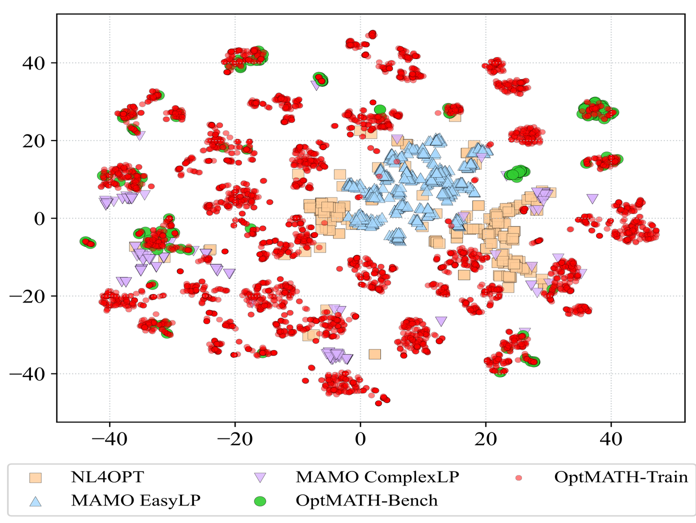


图 3. t-SNE visualization of problem space

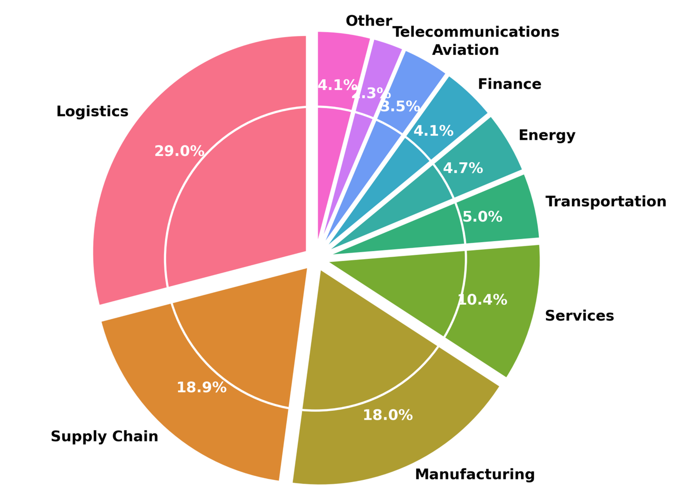


图 4. Distribution of application scenarios

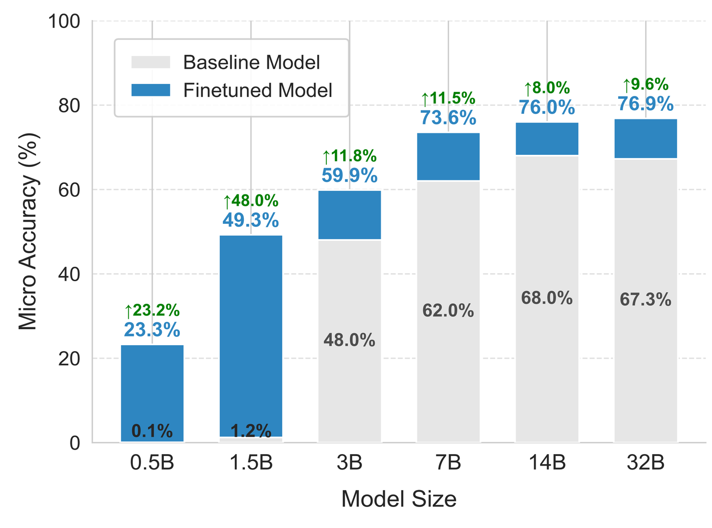


图 5. Model size scaling (0.5B-32B)

Types	Models	NL4OPT	MAMO EasyLP	MAMO ComplexLP	OptMATH Bench	IndustryOR	OptiBench	Macro AVG
Baseline	Llama3.1-8B	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.1%
	Qwen2.5-7B	86.9%	83.6%	21.8%	1.6%	10.0%	36.2%	40.0%
	GPT-3.5-turbo	78.0%	79.3%	33.2%	15.0%	21.0%	47.4%	51.4%
	GPT-4	89.0%	87.3%	49.3%	16.6%	33.3%	68.6%	57.4%
	Deepseek-V3	95.9%	88.3%	51.1%	32.6%	37.0%	71.6%	62.8%
	OptiMUS (GPT-4o-2024-05-13)	78.8%	77.0%	43.6%	20.2%	31.0%	45.8%	49.4%
	Qwen2.5-32B	92.7%	82.2%	44.6%	9.3%	16.0%	47.6%	48.7%
Fine-tuning	ORLM-Llama-3-8B (reported)	85.7%	82.3%	37.4%	*	38.0%	*	60.9%
	ORLM-Llama-3-8B (reproduced)	84.5%	74.9%	34.1%	2.6%	24.0%	51.1%	45.2%
	OptMATH-Llama3.1-8B (pass@1)	55.5%	73.9%	40.8%	24.4%	18.0%	55.5%	44.7%
	OptMATH-Qwen2.5-7B (pass@1)	94.7%	86.5%	51.2%	24.4%	20.0%	57.9%	55.8%
	OptMATH-Qwen2.5-32B (pass@1)	**95.9%**	**89.9%**	**54.1%**	**34.7%**	**31.0%**	**66.1%**	**62.0%**
Pass@8	OptMATH-Llama3.1-8B	97.6%	94.2%	71.6%	51.6%	37.0%	66.6%	69.8%
	OptMATH-Qwen2.5-7B	98.4%	94.5%	72.5%	56.0%	38.0%	68.1%	71.3%
	OptMATH-Qwen2.5-32B	97.9%	93.9%	75.4%	67.4%	47.0%	76.8%	76.4%

表 1. Performance Comparison of Models on Different Benchmarks

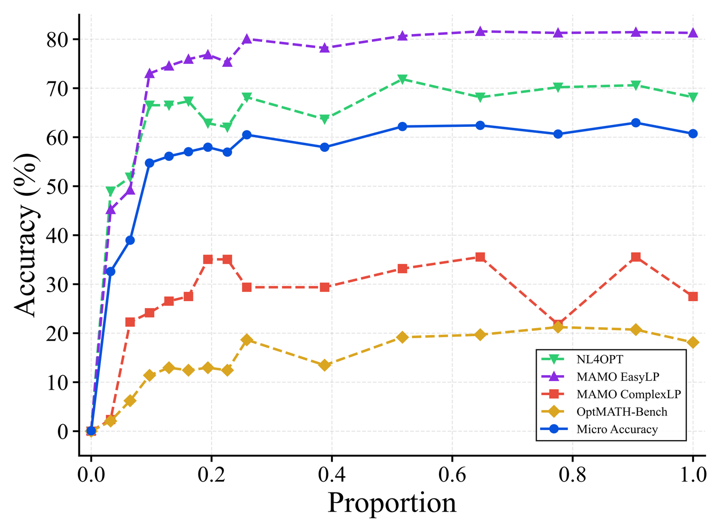


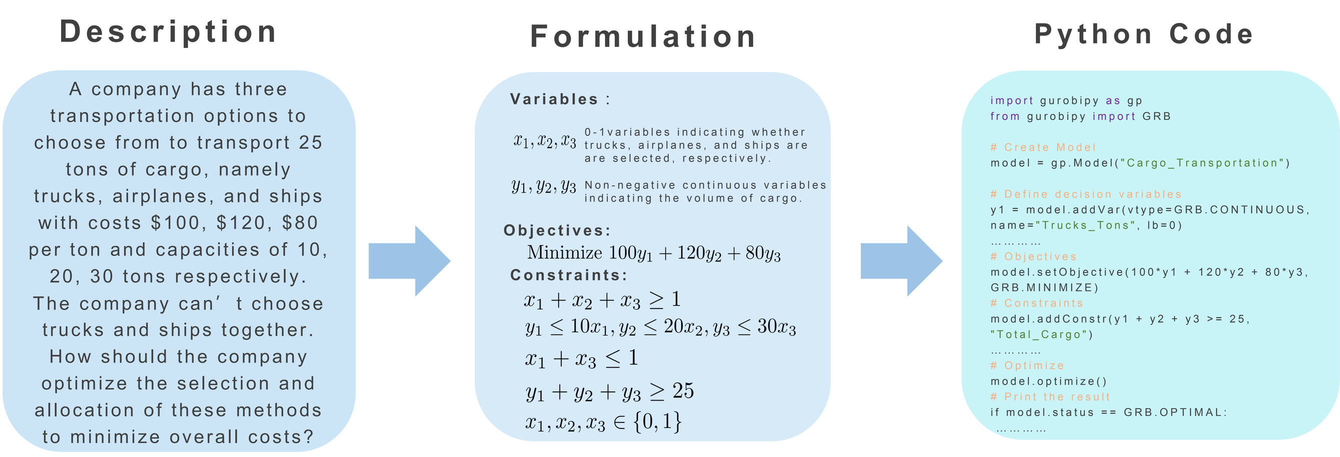
图 6. Training data scaling (Qwen2.5-1.5B)

Problem Formulation

The formulation for increasing the modeling capability of LLMs:

$$\max_{\theta} \quad \mathbb{E}_{(\mathrm{NL},\mathrm{MF},\mathrm{PD})\sim\mathcal{D}}[Q_{(\mathrm{NL},\mathrm{MF},\mathrm{PD})}(\mathrm{MF}',\mathrm{PD}')]$$
$$\text{s.t.} \quad (\mathrm{MF}',\mathrm{PD}') = \mathcal{A}_\theta(\mathtt{prompt}_{\mathrm{M}}(\mathrm{NL}))$$

- $\mathcal{A}_\theta$: Large Language Model with parameters θ
- Q: Quality metric for evaluation
- $\mathcal{D}$: Distribution of problem instances
- $\mathtt{prompt}_{\mathrm{M}}$: Modeling prompt template
- NL: Natural Language Description
- MF: Mathematical Formulation (abstract)
- PD: Problem Data (concrete, solver-ready)



Feedback-Driven PD Generation

Algorithm 1 Feedback-Driven Problem Data Generation

Require: Target complexity range $[S_{\min}, S_{\max}]$, time limits $[T_{\min}, T_{\max}]$, instance generator G, feasibility threshold $\mathcal{F}_{\text{target}}$, max iterations T

Ensure: Configuration Θ such that for $\mathrm{PD}_i \sim G(\Theta)$: $S(\mathrm{PD}_i) \in [S_{\min}, S_{\max}]$ (complexity), $\tau_i \le T_{\max}$ (solving time), $\Pr(f_i = \mathtt{feasible}) \ge \mathcal{F}_{\text{target}}$

1: Initialize parameters via LLM:
  $\Theta_0 \leftarrow \mathcal{L}(\mathtt{prompt}_{\mathrm{IC}}(S_{\min}, S_{\max}, T_{\min}, T_{\max}))$
2: **for** $t = 1$ **to** T **do**
3: Generate N PDs: $\{\mathrm{PD}_i\}_{i=1}^N \leftarrow G(\Theta_{t-1})$
4: Compute metrics: $S(\mathrm{PD}_i)$ (Eq. 4), τ_i (solving time), f_i (feasibility)
5: Aggregate statistics: $\bar{S}_t = \frac{1}{N}\sum S(\mathrm{PD}_i)$, $\bar{\tau}_t = \frac{1}{N}\sum \tau_i$, $\mathcal{F}_t = \frac{1}{N}\sum \mathbb{I}(f_i = \mathtt{feasible})$
6: **if** $\bar{S}_t \in [S_{\min}, S_{\max}]$ **and** $\bar{\tau}_t \le T_{\max}$ **and** $\mathcal{F}_t \ge \mathcal{F}_{\text{target}}$ **then**
7:   **return** Θ_{t-1}
8: **else**
9:   Refine parameters via feedback:
   $\Theta_t \leftarrow \mathcal{L}(\mathtt{prompt}_{\mathrm{RC}}(\bar{S}_t, \bar{\tau}_t, \mathcal{F}_t; \Theta_{t-1}))$
10: **end if**
11: **end for**
12: **return** $\emptyset$ (no valid Θ found)

Complexity score function:

$$S(\mathrm{PD}) = \alpha_{\text{bin}}N_{\text{bin}} + \alpha_{\text{int}}N_{\text{int}} + \alpha_{\text{cont}}N_{\text{cont}} + \beta_{\text{lin}}N_{\text{lin}} + \beta_{\text{indic}}N_{\text{indic}} + \beta_{\text{quad}}N_{\text{quad}} + \beta_{\text{gen}}N_{\text{gen}} + \gamma_{\text{BigM}}f_{\text{BigM}} + \delta_{\text{expr}}\overline{L_{\text{expr}}}$$

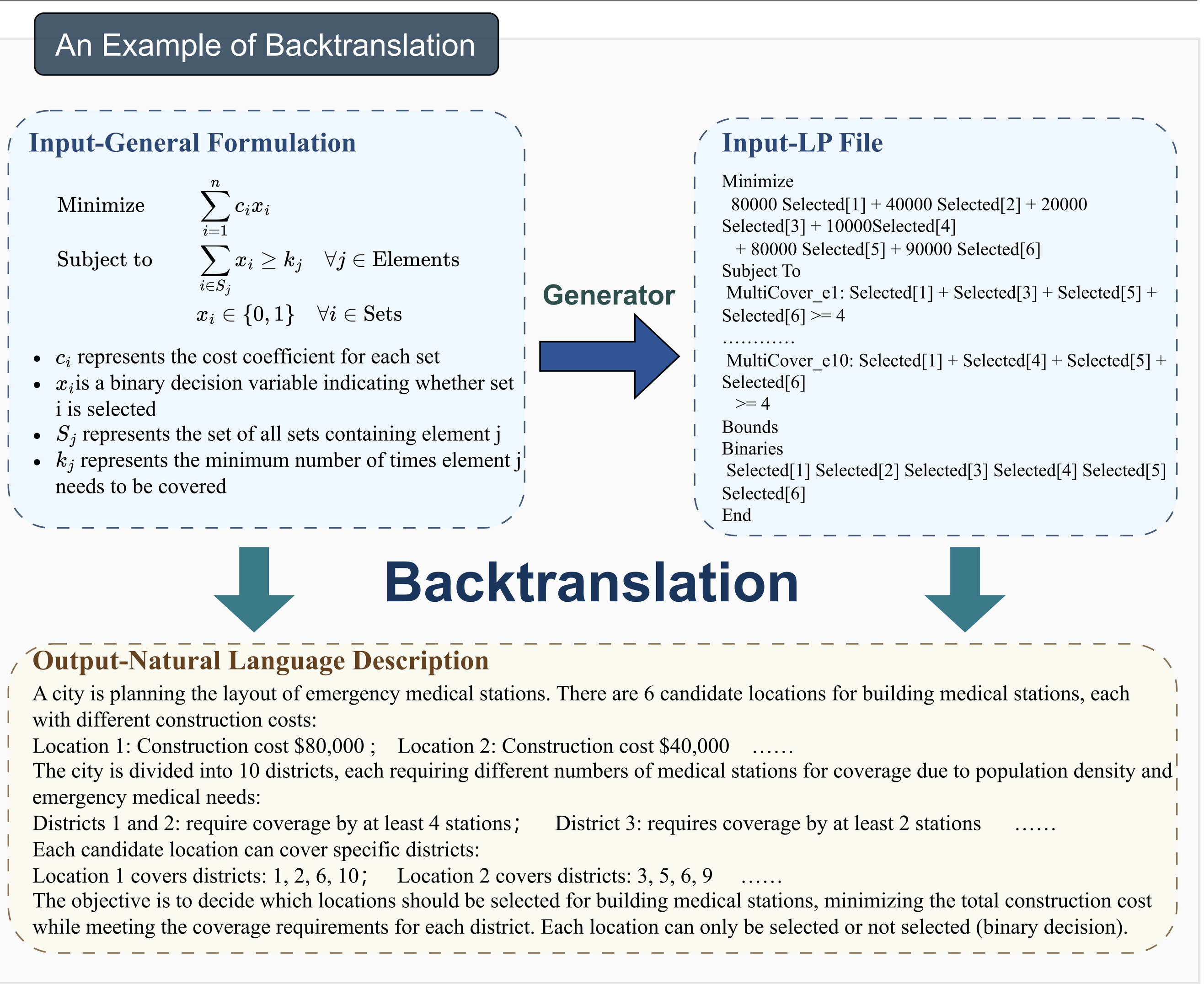
Bidirectional Data Synthesis Algorithm

Algorithm 2 Bidirectional Data Synthesis Algorithm

Require: Instance pair $(\mathrm{MF}_i, \mathrm{PD}_{i,j})$, Max Iteration T

Ensure: $(\mathrm{NL}_{i,j}, \mathrm{MF}'_{i,j}, \mathrm{PD}'_{i,j}, \mathrm{OV}_{i,j})$

1: Initial generation: $\mathrm{NL} \leftarrow \mathcal{L}(\mathtt{prompt}_{\mathrm{I}}(\mathrm{MF}_i, \mathrm{PD}_{i,j}))$
2: Initialize: $\mathrm{SC} = \mathrm{SR} = \mathrm{Null}$
3: **for** $k = 1, \ldots, T-1$ **do**
4: Self-Criticize:
  $\mathrm{SC} \leftarrow \mathcal{L}(\mathtt{prompt}_{\mathrm{C}}(\mathrm{MF}_i, \mathrm{PD}_{i,j}, \mathrm{NL}))$
5: Self-Refine:
  $\mathrm{SR} \leftarrow \mathcal{L}(\mathtt{prompt}_{\mathrm{R}}(\mathrm{MF}_i, \mathrm{PD}_{i,j}, \mathrm{NL}, \mathrm{SC}, \mathrm{SR}))$
6: **if** SR is good enough **then**
7:   **break**
8: **end if**
9: **end for**
10: $\mathrm{NL}_{i,j} \leftarrow \mathrm{SR}$
11: AutoFormulation:
  $(\mathrm{MF}'_{i,j}, \mathrm{PD}'_{i,j}) \leftarrow \mathcal{A}_\theta(\mathtt{prompt}_{\mathrm{M}}(\mathrm{NL}_{i,j}))$
12: $\mathrm{OV}_{i,j} \leftarrow$ Solve $\mathrm{PD}_{i,j}$ by Gurobi
13: $\mathrm{OV}'_{i,j} \leftarrow$ Solve $\mathrm{PD}'_{i,j}$ by Gurobi
14: **if** $\mathrm{OV}_{i,j} = \mathrm{OV}'_{i,j}$ **then**
15: **return** $(\mathrm{NL}_{i,j}, \mathrm{MF}'_{i,j}, \mathrm{PD}'_{i,j}, \mathrm{OV}_{i,j})$
16: **else**
17: **return** Null
18: **end if**



Training Strategy

Parameter-efficient fine-tuning with LoRA:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(p,y)\sim\mathcal{D}_{\text{SFT}}}\left[\sum_{t=1}^{|y|}\log P_\theta(y_t|y_{<t}, p)\right]$$