A Appendix: Dataset

Figures 6, 7, and 8 offer deeper insights into the structure and content of the *AutoOpt-11k* dataset developed in this study. These visualizations collectively help characterize the dataset along multiple dimensions—expression complexity, comparative scale, and token diversity.

Figure 6 presents a histogram of LaTeX expression lengths, indicating the number of samples corresponding to different expression sizes. This figure is particularly useful for understanding the distribution of expression complexity in our dataset. Unlike many datasets that contain shorter and simpler expressions, our dataset encompasses a broad range of lengths, including a substantial number of longer and more elaborate expressions. This distribution is indicative of real-world mathematical programs. Figure 7 provides a comparative analysis of LaTeX expression lengths across multiple publicly available datasets [59, 82, 24, 32] for mathematical expression recognition. It is evident from the figure that our dataset distinguishes itself by including a significantly higher proportion of longer, multi-line expressions. This unique characteristic enhances its applicability to practical use cases that require parsing long mathematical expressions. Figure 8 illustrates the 100 most frequent LaTeX tokens in our dataset, underscoring its syntactic richness and diversity. The tokens span a wide range of categories, including comparison operators, set theory notations, mathematical operators, syntactic elements, Latin letters, numbers, Blackboard capital letters, Greek symbols, mathematical constructs, modifiers, matrix environments, delimiters, arrows, dots, punctuations and various other symbols. This token diversity confirms the dataset's relevance for training models that must generalize across diverse types of notation.

Furthermore, Table 6 and Table 7 provide concrete examples from the dataset, showcasing images of optimization formulations along with their corresponding LaTeX and PYOMO representations. These examples demonstrate the alignment between visual representations and their semantic counterparts, highlighting the dataset's utility for a variety of tasks. Together, the figures and tables substantiate the comprehensiveness and quality of our dataset, validating its potential to support robust training and evaluation of machine learning models targeting advanced mathematical understanding and code generation tasks.

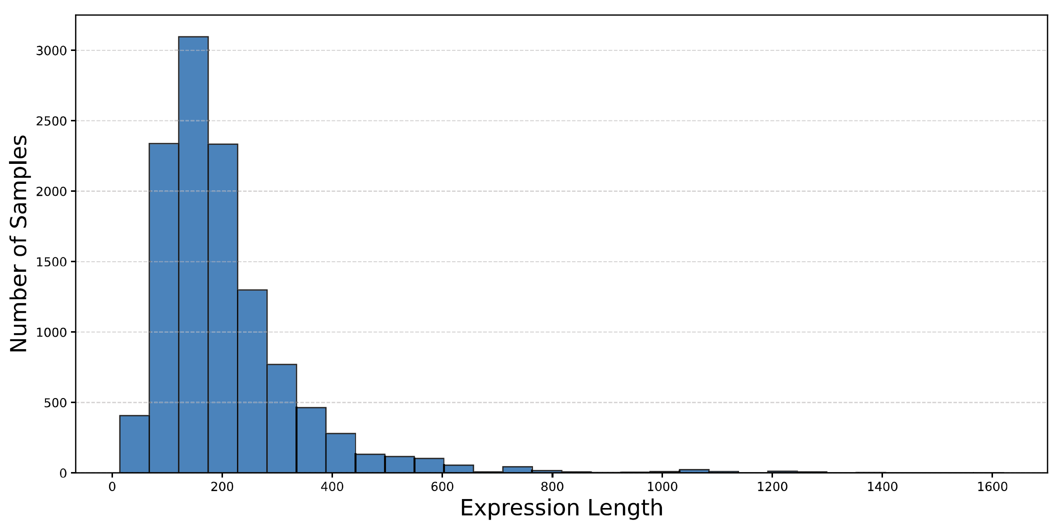


Figure 6: Histogram of LaTeX expression lengths

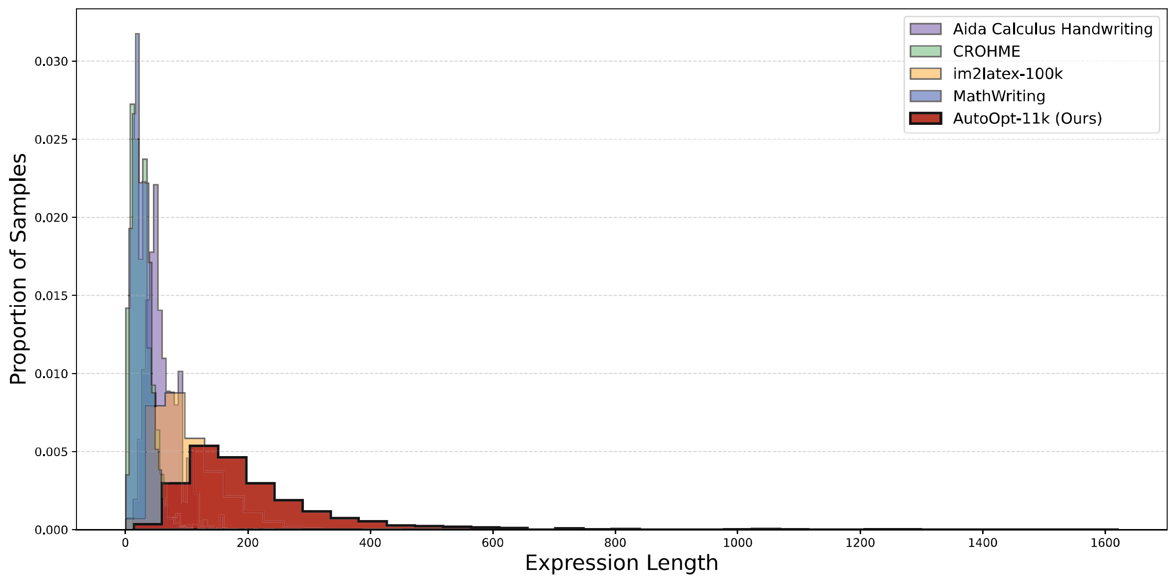


Figure 7: Normalized comparison of LaTeX expression lengths

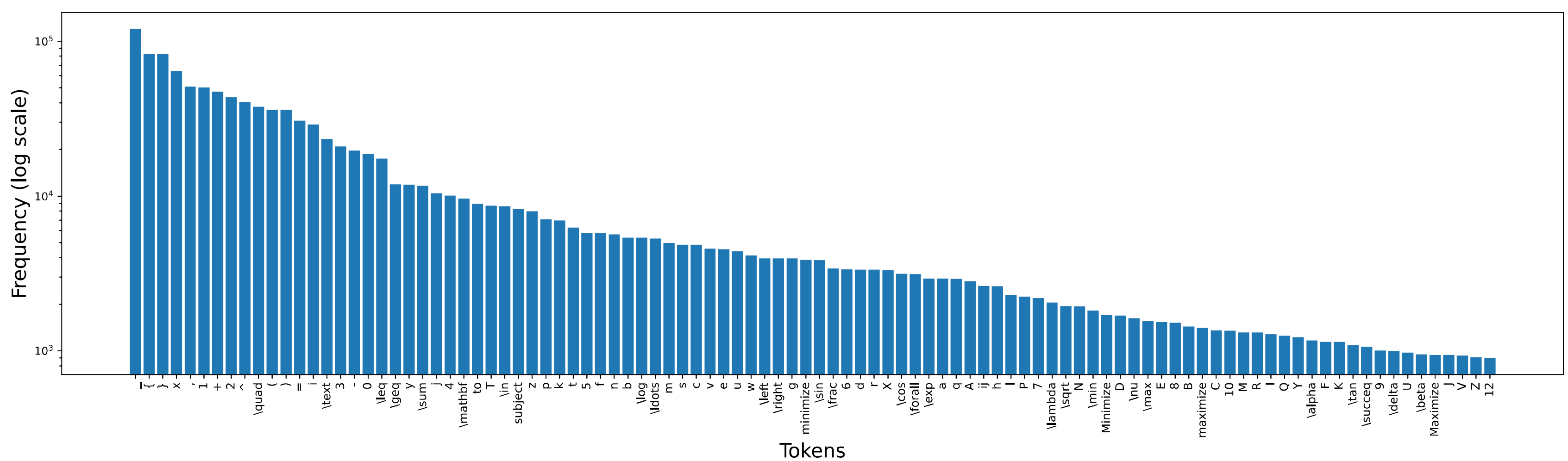


Figure 8: Top 100 most frequent tokens in LaTeX

Table 6: Samples from *AutoOpt-11k* dataset: Images and Labels

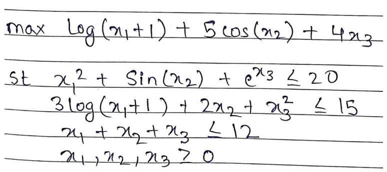
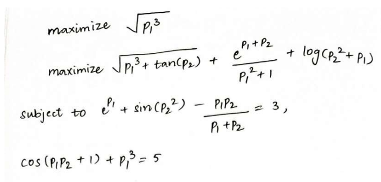
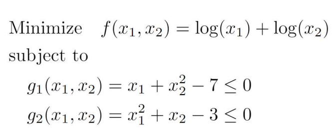
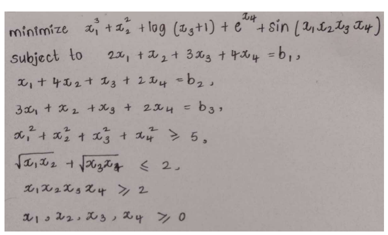
Image	LaTex and PYOMO
	`\text{max} \quad & \log(x_1 + 1) + 5 \cos(x_2) + 4x_3 \\ \text{st} \quad & x_1^2 + \sin(x_2) + e^{x_3} \leq 20 \\ & 3 \log(x_1 + 1) + 2x_2 + x_3^2 \leq 15 \\ & x_1 + x_2 + x_3 \leq 12 \\ & x_1, x_2, x_3 \geq 0.`
	`model.x1 = Var(within=NonNegativeReals)\nmodel.x2 = Var(within=NonNegativeReals)\nmodel.x3 = Var(within=NonNegativeReals)\n def objective_function(model):\n return log(x1 + 1) + 5*cos(x2) + 4*x3\nmodel.OF = Objective(rule=objective_function,sense=maximize) \nmodel.Constraint1 = Constraint(expr = x1**2 + sin(x2) + exp(x3) - 20 <= 0)\nmodel.Constraint2 = Constraint(expr = 3*log(x1 + 1) + 2*x2 + x3**2 - 15 <= 0)\nmodel.Constraint3 = Constraint(expr = x1 + x2 + x3 - 12 <= 0)`
	`\text{maximize} \quad & \sqrt{p_1^3 + \tan(p_2)} + \frac{e^{p_1 + p_2}}{p_1^2 + 1} + \log(p_2^2 + p_1) \\ \text{subject to} \quad & e^{p_1} + \sin(p_2^2) - \frac{p_1 p_2}{p_1 + p_2} = 3, \\ & \cos(p_1 p_2 + 1) + p_1^3 = 5`
	`model.p1 = Var()\nmodel.p2 = Var()\ndef objective_rule(model):\n return sqrt(p1**3 + tan(p2)) + exp(p1 + p2)/(p1**2 + 1) + log(p2**2 + p1)\n model.objective = Objective(rule=objective_rule, sense=maximize)\nmodel. Constraint1 = Constraint(expr = exp(p1) + sin(p2**2) - (p1*p2)/(p1 + p2) == 3)\n model.Constraint2 = Constraint(expr = cos(p1*p2 + 1) + p1**3 == 5)`
	`& \text{Minimize} \quad f(x_1, x_2) = \log(x_1) + \log(x_2) \\ & \text{subject to} \\ & \quad g_1(x_1, x_2) = x_1 + x_2^2 - 7 \leq 0 \\ & \quad g_2(x_1, x_2) = x_1^2 + x_2 - 3 \leq 0`
	`model.x1 = Var(within=PositiveReals)\nmodel.x2 = Var(within=PositiveReals)\ndef objective_function(model):\n return log(x1) + log(x2)\nmodel.obj = Objective(rule=objective_function, sense=minimize)\nmodel.Constraint1 = Constraint(expr = x1 + x2**2 <= 7)\n model.Constraint2 = Constraint(expr = x1**2 + x2 <= 3)`
	`\text{minimize} \quad & x_1^3 + x_2^2 + \log(x_3 + 1) + e^{x_4} + \sin(x_1 x_2 x_3 x_4) \\ \text{subject to} \quad & 2x_1 + x_2 + 3x_3 + 4x_4 = b_1, \\ & x_1 + 4x_2 + x_3 + 2x_4 = b_2, \\ & 3x_1 + x_2 + x_3 + 2x_4 = b_3, \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 \geq 5, \\ & \sqrt{x_1 x_2} + \sqrt{x_3 x_4} \leq 2, \\ & x_1 x_2 x_3 x_4 \geq 2, \\ & x_1, x_2, x_3, x_4 \geq 0`
	`model.x1 = Var(within=NonNegativeReals)\nmodel.x2 = Var(within=NonNegativeReals)\nmodel.x3 = Var(within=NonNegativeReals)\n model.x4 = Var(within=NonNegativeReals)\ndef objective_rule(model):\n return x1**3 + x2**2 + log(x3 + 1) + exp(x4) + sin(x1 * x2 * x3 * x4)\nmodel.objective = Objective(rule=objective_rule, sense=minimize)\nmodel.Constraint1 = Constraint(expr= 2*x1 + x2 + 3*x3 + 4*x4 == b1)\nmodel.Constraint2 = Constraint(expr = x1 + 4*x2 + x3 + 2*x4 == b2)\nmodel.Constraint3 = Constraint(expr= 3*x1 + x2 + x3 + 2*x4 == b3)\nmodel.Constraint4 = Constraint(expr = x1**2 + x2**2 + x3**2 + x4**2 >= 5)\n model.Constraint5 = Constraint(expr= sqrt(x1 * x2) + sqrt(x3 * x4) <= 2)\nmodel.Constraint6 = Constraint(expr= x1 * x2 * x3 * x4 >= 2)`

Table 7: Samples from *AutoOpt-11k* dataset: Images and Labels (Cont.)

Image	LaTex and PYOMO
$\begin{array}{ll} \text{maximize} & 160E + 250C + 210P_1 + 230P_2 + 300B \\ \text{subject to} & 12E + 20C + 15P_1 + 25P_2 + 30B \leq 260 \\ & \log(E+1) + 3C + 5P_1 + 7P_2 + 6B \leq 45 \\ & \exp(C) + 2E + \cos(P_1) + 4P_2 + 5B \leq 38 \\ & 6E + 7C + 8P_1 + 9P_2 + 10B \leq 50 \\ & 4E + 6C + 2P_1 - 3P_2 + 4B \leq 18 \\ & P_1 - P_2 = -6 \\ & E, C, P_1, P_2, B \geq 0 \end{array}$	`\text{maximize} & 160E + 250C + 210P_1 + 230P_2 + 300B \\ \text{subject to} & 12E + 20C + 15P_1 + 25P_2 + 30B \leq 260 \\ & \log(E + 1) + 3C + 5P_1 + 7P_2 + 6B \leq 45 \\ & \exp(C) + 2E + \cos(P_1) + 4P_2 + 5B \leq 38 \\ & 6E + 7C + 8P_1 + 9P_2 + 10B \leq 50 \\ & 4E + 6C + 2P_1 - 3P_2 + 4B \leq 18 \\ & P_1 - P_2 = -6 \\ & E, C, P_1, P_2, B \geq 0.`
	`model.E = Var(within=NonNegativeReals)\nmodel.C = Var(within=NonNegativeReals)\nmodel.P1 = Var(within=NonNegativeReals)\nmodel.P2 = Var(within=NonNegativeReals)\nmodel.B = Var(within=NonNegativeReals)\ndef objective_function(model):\n return 160*E + 250*C + 210*P1 + 230*P2 + 300*B\nmodel.obj = Objective(rule=objective_function, sense=maximize)\nmodel.Constraint1 = Constraint(expr = 12*E + 20*C + 15*P1 + 25*P2 + 30*B <= 260)\nmodel.Constraint2 = Constraint(expr = log(E + 1) + 3*C + 5*P1 + 7*P2 + 6*B <= 45)\n model.Constraint3 = Constraint(expr = exp(C) + 2*E + cos(P1) + 4*P2 + 5*B <= 38)\nmodel.Constraint4 = Constraint(expr = 6*E + 7*C + 8*P1 + 9*P2 + 10*B <= 50)\nmodel.Constraint5 = Constraint(expr = 4*E + 6*C + 2*P1 - 3*P2 + 4*B <= 18)\nmodel.Constraint6 = Constraint(expr = P1 - P2 == -6)`
$\begin{array}{ll} \max & 45x_1 + 60x_2 + 30x_3 + 65x_4 + 10x_5 + 35x_6 \\ \text{subject to} & 25x_1 + 50x_2 + 35x_3 + 45x_4 + 30x_5 + 20x_6 \leq 8500 \\ & x_i \leq 14 \ (i = 1, \ldots, 6) \\ & x_4 \leq 3x_2 \\ & x_1 + x_3 \geq 4y \\ & x_5 + x_6 \geq 6(1-y) \\ & x_i \geq 0 \ (i = 1, \ldots, 6) \\ & x_i \in \mathbb{Z} \ (i = 1, \ldots, 6), \ y \in \{0, 1\} \end{array}$	`\text{max} \quad & 45x_1 + 60x_2 + 30x_3 + 65x_4 + 10x_5 + 35x_6 \\ \text{subject to} \quad & 25x_1 + 50x_2 + 35x_3 + 45x_4 + 30x_5 + 20x_6 \leq 8500 \\ & x_i \leq 14 \quad (i = 1, \ldots, 6) \\ & x_4 \leq 3x_2 \\ & x_1 + x_3 \geq 4y \\ & x_5 + x_6 \geq 6(1 - y) \\ & x_i \geq 0 \quad (i = 1, \ldots, 6) \\ & x_i \quad \text{integer} \quad (i = 1, \ldots, 6) \\ & y \in \{0, 1\}.`
	`model.x1 = Var(within=NonNegativeReals, domain=Integers)\\nmodel.x2 = Var(within=NonNegativeReals, domain=Integers)\\nmodel.x3 = Var(within=NonNegativeReals, domain=Integers)\\nmodel.x4 = Var(within=NonNegativeReals, domain=Integers)\\nmodel.x5 = Var(within= NonNegativeReals, domain=Integers)\\nmodel.x6 = Var(within=NonNegativeReals, domain=Integers)\\nmodel.y = Var(domain=Binary)\\n def objective_function(model):\\n return 45*x1 + 60*x2 + 30*x3 + 65*x4 + 10*x5 + 35*x6\\nmodel.OF = Objective(rule= objective_function, sense=maximize)\\nmodel.Constraint1 = Constraint(expr = 25*x1 + 50*x2 + 35*x3 + 45*x4 + 30*x5 + 20*x6 <= 8500) \\nmodel.Constraint2 = Constraint(expr = x1 <= 14)\\nmodel.Constraint3 = Constraint(expr = x2 <= 14)\\nmodel.Constraint4 = Constraint(expr = x3 <= 14)\\nmodel.Constraint5 = Constraint(expr = x4 <= 14)\\nmodel.Constraint6 = Constraint(expr = x5 <= 14)\\n model.Constraint7 = Constraint(expr = x6 <= 14)\\nmodel.Constraint8 = Constraint(expr = x4 <= 3*x2)\\nmodel.Constraint9 = Constraint(expr = x1 + x3 >= 4*y)\\nmodel.Constraint10 = Constraint(expr = x5 + x6 >= 6*(1 - y))`
$\begin{array}{l} \text{Minimize} \quad f(x_1, x_2) = \sqrt{x_1} + \sqrt{x_2} \\ \text{subject to} \\ \quad x_1 + 2x_2 \leq 13 \\ \quad 1 \leq x_i \leq 9, \quad i = 1, 2 \end{array}$	`\text{Minimize} \quad f(x_1, x_2) = \sqrt{x_1} + \sqrt{x_2} \\ & \text{subject to} \\ & \quad x_1 + 2x_2 \leq 13 \\ & \quad 1 \leq x_i \leq 9, \quad i = 1, 2`
	`model.x1 = Var(bounds=(1,9))\nmodel.x2 = Var(bounds=(1,9))\ndef objective_function(model):\n    return sqrt(x1) + sqrt(x2)\n model.obj = Objective(rule=objective_function, sense=minimize)\nmodel.Constraint1 = Constraint(expr = x1 + 2*x2 <= 13)`

B Appendix: Module M1

We did 5 runs for AutoOpt-M1, Nougat, ChatGPT, and Gemini by randomly splitting the data with a training, validation, and test split of 80%, 10% and 10%, respectively. The models corresponding to the median BLUE score for AutoOpt-M1, Nougat, ChatGPT and Gemini are reported in Section 3.1.1. The standard deviations in BLUE score and Character Error Rate are reported in Table 8 and Table 9, respectively.

Table 8: Standard deviation of BLUE score from 5 runs

Model	HW	PR	HW+PR
GPT-4o	0.76	0.48	0.52
Gemini 2.0 Flash	0.88	0.51	1.18
Nougat	1.16	0.8	1.07
AutoOpt-M1	1.14	0.87	1.04

Table 9: Standard deviation of Character Error Rate from 5 runs

Model	HW	PR	HW+PR
GPT-4o	0.0068	0.0084	0.0032
Gemini 2.0 Flash	0.0224	0.0054	0.0113
Nougat	0.0098	0.0087	0.0058
AutoOpt-M1	0.0122	0.0086	0.0072

All experiments were conducted on Google Colab Pro using NVIDIA A100 GPU. The AutoOpt-M1 model was trained for 180 epochs. We used AdamW optimizer with learning rate $2e^{-5}$, weight decay 0.02, and with a cosine scheduler. The batch size was set to 8 with gradient accumulation of 2. Each epoch took approximately 15-20 minutes. Figure 9 shows the convergence plot for the model for a particular run.

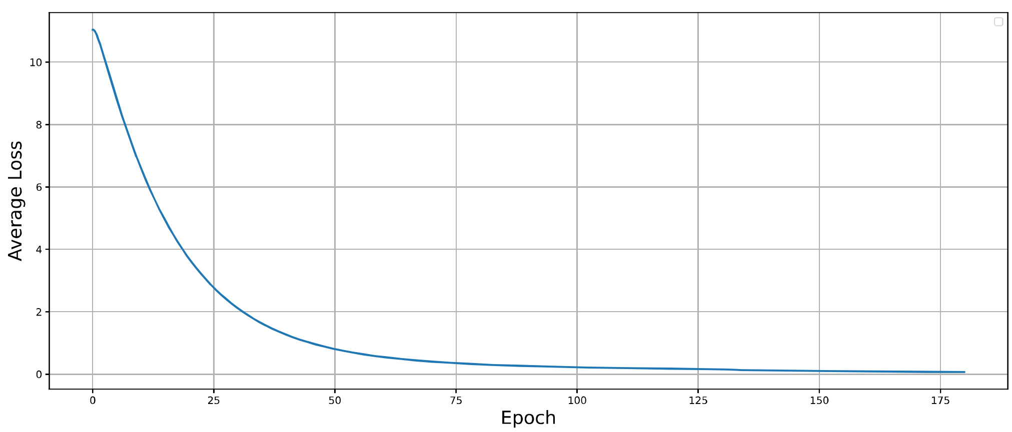


Figure 9: Convergence plot for AutoOpt-M1

C Appendix: Module M2

The median BLEU score and median Character Error Rate (CER) correspond to the same run among the five runs we performed. The median scores are reported in Section 3.2. The standard deviations for BLUE score (in percent) and Character Error Rate over 5 runs are 1.08 and 0.0051, respectively.

We fine-tuned the model for 15 epochs on Google Colab Pro using an NVIDIA A100 GPU with mixed-precision (fp16). For training, the batch size was set to 2 with gradient accumulation of 4 (effective batch size 8). The learning rate was set to $5e^{-5}$ with a weight decay of 0.01. Each run took approximately 30–35 minutes.

D Appendix: Module M3

In this study, we consider a Bilevel Optimization based Decomposition (BOBD) method [72] for optimization task in module M3 (Section 3.3). BOBD method offers the advantages of both exact and approximation methods, by providing efficient solutions within a reasonable time frame. BOBD method attains this advantage by using a bilevel optimization structure that allows it to simultaneously use exact and approximation methods for problem solving. To explain the working procedure of the BOBD method, we first review the structures of general and bilevel optimization problems based on their formal definitions.

Definition 1. *A general optimization problem can be represented using its basic elements, decision variables $x = (x_1, \ldots, x_n)$, objective function $F(x)$, and constraints $G(x)$ and $H(x)$, as follows:*

$$\min_{x} \quad F(x) \tag{1}$$

$$\textit{subject to} \quad G_i(x) \leq 0, \quad i = 1, \ldots, I \tag{2}$$

$$H_j(x) = 0, \quad j = 1, \ldots, J \tag{3}$$

Bilevel optimization problem is characterized by a unique structure in which the primary or upper level optimization problem contains an additional optimization problem, lower level optimization problem, nested within it as a constraint [78, 70, 71].

Definition 2. *A bilevel optimization problem, with upper level and lower level decision variables (u and l), objective functions ($F(u,l)$ and $f(u,l)$), and constraints ($G(u,l)$ & $H(u,l)$ and $g(u,l)$ & $h(u,l)$), can be represented as follows:*

$$\min_{u,l} \quad F(u,l) \tag{4}$$

$$\textit{subject to}$$

$$l \in \underset{l}{\operatorname{argmin}}\{f(u,l) : g_p(u,l) \leq 0, \quad p = 1, \ldots, P, \quad h_q(u,l) = 0, \quad q = 1, \ldots, Q\} \tag{5}$$

$$G_i(u,l) \leq 0, \quad i = 1, \ldots, I \tag{6}$$

$$H_j(u,l) = 0, \quad j = 1, \ldots, J \tag{7}$$

To solve a bilevel problem (Definition 2) in a nested manner, the values of upper level variables u are fixed, and the lower level problem is solved with respect to l. By intelligently sampling u and solving the lower level problem repeatedly, it is possible to converge to the bilevel optimum.

BOBD method involves the variable classification task, in which we classify each decision variable ($x_i \in x$; $i = 1...n$) into upper level (u) or lower level (l) variables category. It allows to express the general optimization problem (Definition 1) in the form of bilevel optimization problem (Definition 2), which is called a bilevel decomposition process. In this study, we develop a Logistic Regression based Variable Classification Model (LR-VCM), a method to perform the variable classification task. To begin with, a population is generated and all the variables are initialized randomly. Thereafter, we start with a random approach of variable classification, where each decision variable is classified randomly into upper level (tag 0) or lower level (tag 1). Every single trial of level selection for all variables is referred to as Level Configuration (LC), and the collection of corresponding tag values constitutes a single observation for logistic regression. For given LC, we evaluate if it leads to an improvement in the objective function, when the lower level problem is solved. The improvement results are recorded on a binary scale: 0 indicates little improvement, and 1 indicates notable improvement. This binary score (0 or 1) acts as a label for a given LC. A pair of LC (observation) and corresponding improvement score (label) creates a single training sample for LR-VCM. We construct a training dataset by repeating the same procedure for the entire population. We perform a logistic regression using the developed training set and classify variables with statistically significant p-values into lower level and the remaining variables into upper level. After variable classification (u, l), we intelligently sample the values of upper level variables u using a genetic algorithm, and for each sample, the corresponding lower level problem is solved using a suitable classical optimization method, which provides the values of lower level variables l. The obtained values of upper level and lower level variables yield a particular solution for a given bilevel problem. This procedure is repeated several times to generate multiple solutions and an efficient solution (u^*, l^*) is then identified from the generated solutions, as outlined in the pseudocode provided in Algorithm 1.

Algorithm 1 Bilevel Optimization-based Decomposition (BOBD)

Input:	$F(x)$, $G(x)$, $H(x)$: single level optimization problem (i.e., original problem)
Output:	$x^* = (u^*, l^*)$: the best solution found for single level optimization problem
Step 1:	Generate a population ($\mathcal{P}$) of random initial solutions.
Step 2:	Develop a Logistic Regression based Variable Classification Model (LR-VCM).
Step 3:	Perform a bilevel decomposition of original problem into upper level and lower level using LR-VCM.
Step 4:	*for* $g = 1$ to *number_of_generations*:
Step 5:	If g is divisible by *variable_classification_alternation_number* (C=10):
Step 6:	Develop a new LR-VCM using updated dataset.
Step 7:	Perform a new bilevel decomposition of original problem.
Step 8:	Sample the values of upper level variables u using genetic algorithm.
Step 9:	For a given u, obtain l by solving the corresponding lower level problem using the interior point or the linear programming methods.
Step 10:	Update population $\mathcal{P}$ with new solutions if they are better than the worst solutions in the population.

We bring novelty to BOBD method by incorporating LR-VCM for variable classification task. To evaluate the performance of the BOBD method, we consider a test suite of 10 optimization Test Problems (TP), TP1-TP10 [72]. These test problems are derived from the real-world applications and exhibit various types of complexities such as non-convexity, non-linearity, non-differentiability, discreteness, high-dimensionality, etc. The description of test problems (TP1-TP10) and computational experiments are discussed next.

TP1 (Structural Sensitivity Problem in a Chemical System [76]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6 \\
\text{s.t.} \quad & x_2 - 3x_1 - 3x_4 = 0; \\
& x_3 - 2x_2 - 2x_5 = 0; \\
& 4x_4 - x_6 = 0; \\
& x_1 + 2x_4 \leq 4; \\
& x_2 + x_5 \leq 4; \\
& x_3 + x_6 \leq 6; \\
& x_1 \leq 3;\ x_3 \leq 4;\ x_5 \leq 2;\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

TP2 (Heat Exchanger Design Problem [46]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = x_1 \\
\text{s.t.} \quad & 35x_2^{0.6} + 35x_3^{0.6} - x_1 \leq 0; \\
& -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0; \\
& 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0; \\
& -x_5 + \ln(-x_4 + 900) = 0; \\
& -x_6 + \ln(x_4 + 300) = 0; \\
& -x_7 + \ln(-2x_4 + 700) = 0; \\
& 0 \leq x_1 \leq 1000; \quad 0 \leq x_2, x_3 \leq 40; \quad 100 \leq x_4 \leq 300; \\
& 6.3 \leq x_5 \leq 6.7; \quad 5.9 \leq x_6 \leq 6.4; \quad 4.5 \leq x_7 \leq 6.25
\end{aligned}
$$

TP3 (More complexities added to TP1 [72]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6 + \frac{x_3^2}{16} - 2\cos(2\pi x_2) \\
\text{s.t.} \quad & x_2 - 3x_1 - 3x_4 = 0; \\
& x_3 - 2x_2 - 2x_5 = 0; \\
& 4x_4 - x_6 = 0; \\
& x_1 + 2x_4 \leq 4; \\
& x_2 + x_5 \leq 4; \\
& x_3 + x_6 \leq 6; \\
& x_1 \leq 3; \quad x_5 \leq 2; \quad x_3 \leq 4; \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

TP4 (Scalable variables y and constraints added to TP3 [72]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6 + \frac{x_3^2}{16} - \frac{x_2^2}{16} \\
& \qquad - 2\cos(2\pi x_3) - 2\cos(2\pi x_2) + \sum_{p=1}^{P} y_p \cdot x_1^{0.6} \\
\text{s.t.} \quad & x_2 - 3x_1 - 3x_4 = 0; \\
& x_3 - 2x_2 - 2x_5 = 0; \\
& x_1 + 2x_4 \leq 4; \\
& x_2 + x_5 \leq 4; \\
& x_3 + x_6 \leq 6; \\
& \sqrt{x_1 + x_2 + x_3} - y_p \leq 0, \quad \forall p; \\
& x_1 \leq 3; \quad x_5 \leq 2; \quad x_3 \leq 4; \\
& 1 \leq y_p \leq 5, \ \forall p; \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

TP5 (Scalable variables y and constraints added to TP2 [72]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = x_1 - 50\cos(2\pi x_4) + \sum_{p=1}^{P} \frac{{y_p}^2}{x_4} \\
\text{s.t.} \quad & 35x_2^{0.6} + 35x_3^{0.6} - x_1 \leq 0; \\
& -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0; \\
& 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0; \\
& -x_5 + \ln(-x_4 + 900) = 0; \\
& -x_6 + \ln(x_4 + 300) = 0; \\
& -x_7 + \ln(-2x_4 + 700) = 0; \\
& x_4^{0.2} + x_5 + x_6 - y_p \leq 0, \ \forall p; \\
& 0 \leq x_1 \leq 1000; \quad 0 \leq x_2, x_3 \leq 40; \quad 100 \leq x_4 \leq 300; \\
& 6.3 \leq x_5 \leq 6.7; \quad 5.9 \leq x_6 \leq 6.4; \quad 4.5 \leq x_7 \leq 6.25; \quad 10 \leq y_p \leq 30, \forall p
\end{aligned}
$$

TP6 (Scalable variables (y, z) and constraints added to existing problem [28]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = -25\left(x_1-2\right)^2-\left(x_2-2\right)^2-\left(x_3-1\right)^2-\left(x_4-4\right)^2-\left(x_5-1\right)^2-\left(x_6-4\right)^2 \\
& \qquad + \sum_{p=1}^{P}\left(x_3-y_p\right)^2-\sum_{\mathrm{q}=1}^{Q}\left(x_5-z_q\right)^2 \\
\text{s.t.} \quad & -\left(x_3-3\right)^2-x_4+4 \leq 0; \\
& -\left(x_5-3\right)^2-x_6+4 \leq 0; \\
& -x_1-x_2+2 \leq 0; \\
& \quad x_1-3x_2 \leq 2; \\
& \quad x_2-x_1 \leq 2; \\
& \quad x_1+x_2 \leq 6; \\
& \quad y_p-x_3+1 \leq 0,\ \forall p; \\
& \quad z_q^2-x_3^2-x_5^2 \leq 0,\ \forall q; \\
& \quad 0 \leq x_1; \quad 0 \leq x_2; \quad 1 \leq x_3 \leq 5; \quad 0 \leq x_4 \leq 6; \\
& \quad 1 \leq x_5 \leq 5; \quad 0 \leq x_6 \leq 10; \quad 0 \leq y_p \leq 5, \forall p; \quad 0 \leq z_q \leq 5, \forall q
\end{aligned}
$$

TP7 (Pool blending Problem with additional complexities [46]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = 6x_1+16x_2-9x_5+10\left(x_6+x_7\right)-15x_8+x_9^2+50\cos\left(\pi x_9\right)-25\cos\left(\pi x_8\right) \\
& \qquad -\ln\left(x_8-x_9\right)-\sum_{p=1}^{P}\left(y_p-x_9\right)^2+\sum_{q=1}^{Q}\left(z_q-x_8\right)^2 \\
\text{s.t.} \quad & x_1+x_2-x_3-x_4=0; \\
& x_3+x_6-x_5=0; \\
& x_4+x_7-x_8=0; \\
& 0.03x_1+0.01x_2-x_3x_9-x_4x_9=0; \\
& x_3x_9+0.02x_6-0.025x_5 \leq 0; \\
& x_4x_9+0.02x_7-0.015x_8 \leq 0; \\
& x_9^2-y_p^2 \leq 0,\ \forall p; \\
& x_8^2-z_q^2 \leq 0,\ \forall q; \\
& 0 \leq x_1, x_2, x_6 \leq 300; \quad 0 \leq x_3, x_5, x_7 \leq 100; \quad 0 \leq x_4, x_8 \leq 200; \\
& 0.01 \leq x_9 \leq 0.03; \quad 0 \leq y_p \leq 1, \forall p; \quad 1 \leq z_q \leq 200, \forall q
\end{aligned}
$$

TP8 (Existing problem with additional complexities [72]):

$$
\begin{aligned}
\min_{x} \quad & F(x) = 5\sum_{i=1}^{4} x_i-5\sum_{i=1}^{4} x_i^2-\sum_{i=5}^{13} x_i-20e^{-0.1\sqrt{\sum_{i=1}^{4} x_i^2}}-e^{0.25\sum_{\mathrm{i}=1}^{4}\cos(2\pi x_i)} \\
& \qquad +\sum_{p=1}^{P}\left(y_p^2+\sum_{\mathrm{i}=1}^{4}\cos\left(2\pi x_i\right)\right) \\
\text{s.t.} \quad & 2x_1+2x_2+x_{10}+x_{11} \leq 10; \\
& 2x_1+2x_3+x_{10}+x_{12} \leq 10; \\
& 2x_2+2x_3+x_{11}+x_{12} \leq 10;
\end{aligned}
$$

$$
\begin{aligned}
& x_{10} - 8x_1 \le 0; \\
& x_{11} - 8x_2 \le 0; \\
& x_{12} - 8x_3 \le 0; \\
& x_{10} - x_5 - 2x_4 \le 0; \\
& x_{11} - x_7 - 2x_6 \le 0; \\
& x_{12} - x_9 - 2x_8 \le 0; \\
& \sum_{\mathrm{i}=1}^{4} \cos(2\pi x_i) - y_p \le 0,\ \forall p; \\
& 0 \le x_i \le 3\ \ (i = 1, \ldots, 4); \quad 0 \le x_i \le 1\ \ (i = 5, \ldots, 9); \\
& 0 \le x_i \le 100\ \ (i = 10, 11, 12); \quad 0 \le x_{13} \le 1; \quad -5 \le y_p \le 5, \forall p
\end{aligned}
$$

TP9 (Heat Exchanger Design Problem with additional complexities [46]):

$$
\begin{aligned}
\min_x \quad & F(x) = x_1 + x_2 + x_3 + \sum_{p=1}^{P} \left(\tan(y_p) - 15\cos 2\pi\,(x_1 + x_2 + x_3)\right)^2 \\
\text{s.t.} \quad & 0.0025x_4 + 0.0025x_6 \le 1; \\
& 0.0025x_5 + 0.0025x_7 - 0.0025x_4 \le 1; \\
& 0.01x_8 - 0.01x_5 \le 1; \\
& -x_1x_6 + 100x_1 + 833.33x_4 \le 83333.33; \\
& -x_2x_7 + 1250x_5 - 1250x_4 + x_2x_4 \le 0; \\
& -x_3x_8 - 2500x_5 + x_3x_5 + 1250000 \le 0; \\
& \tan(y_p) - \ln(x_1 + x_2 + x_3) \le 0 \quad \forall p; \\
& -\tan(y_p) - \ln(x_1 + x_2 + x_3) \le 0 \quad \forall p; \\
& 100 \le x_1 \le 10000; \quad 1000 \le x_i \le 10000\ \ (i = 2, 3); \\
& 10 \le x_i \le 1000\ \ (i = 4, \ldots, 8); \quad -\pi/2 \le y_p \le \pi/2, \forall p
\end{aligned}
$$

TP10 (Existing problem with additional complexities [72]):

$$
\begin{aligned}
\min_x \quad & F(x) = 37.293239x_1 + 0.8356891x_1x_5 + 5.3578547x_3^2 - 40792.14 \\
& \qquad + \sum_{p=1}^{P} (y_p - x_1 - x_3)^2 - 150 \sum_{q=1}^{Q} \cos(2\pi z_q) \\
\text{s.t.} \quad & 0.0056858x_2x_5 - 0.0022053x_3x_5 + 0.0006262x_1x_4 \le 6.665593; \\
& -0.0056858x_2x_5 + 0.0022053x_3x_5 - 0.0006262x_1x_4 \le 85.334407; \\
& 0.0071317x_2x_5 + 0.0021813x_3^2 + 0.0029955x_1x_2 \le 29.48751; \\
& -0.0071317x_2x_5 - 0.0021813x_3^2 - 0.0029955x_1x_2 + 9.48751 \le 0; \\
& 0.0047026x_3x_5 + 0.0019085x_3x_4 + 0.0012547x_1x_3 \le 15.699039; \\
& -0.0047026x_3x_5 - 0.0019085x_3x_4 - 0.0012547x_1x_3 + 10.699039 \le 0; \\
& y_p - \ln(x_1 + x_3 + 1) \le 0,\ \forall p; \\
& z_q^3 - x_1^3 - x_3^3 - x_5^3 \le 0,\ \forall q; \\
& 78 \le x_1 \le 102; \quad 33 \le x_2 \le 45; \quad 27 \le x_3, x_4, x_5 \le 45; \\
& 0 \le y_p \le 5, \forall p; \quad -5 \le z_q \le 5, \forall q
\end{aligned}
$$

For computational experiments, we consider the following three scenarios: small-scale ($|y|+|z|=0$), medium-scale ($|y|+|z|=20$), and large-scale ($|y|+|z|=50$) (here, $|y|+|z|$ represents the number of scalable variables and constraints in TP). All test problems are solved in small to large scale scenarios using the Interior Point (IP), Genetic Algorithm (GA), and BOBD methods. For all TP, constraint tolerance is set to 10^{-4}. For genetic algorithm, the details on GA operators and parameters value are provided in Table 10. For GA implemented in BOBD method, all parameters are kept same as mentioned in Table 10, and an improvement based-termination criterion is used. We consider a steady state genetic algorithm [77] in both explicit GA and GA used in BOBD.

Table 10: Genetic algorithm operators and parameter values for computational experiments

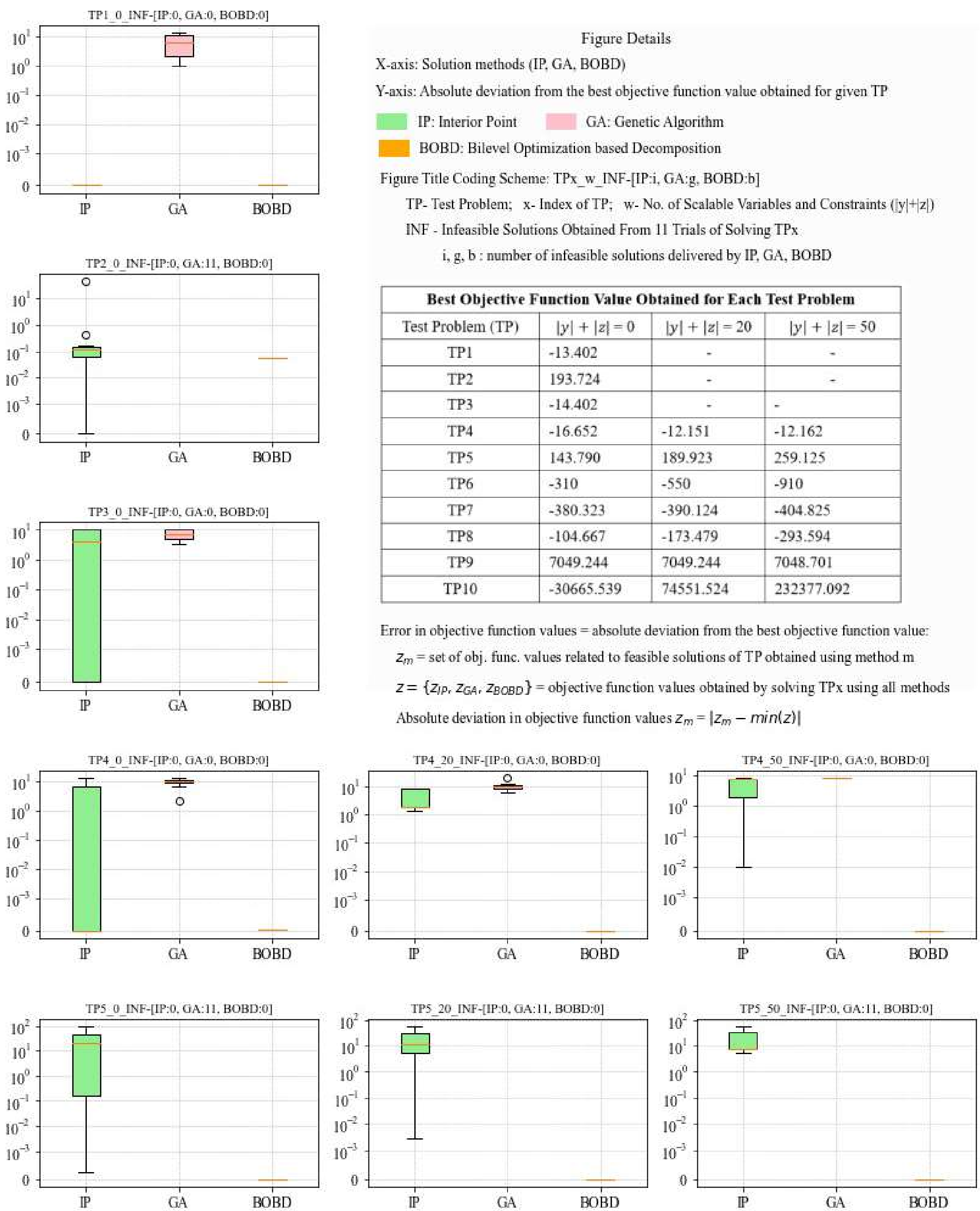
GA operators	**Mechanisms**	**Parameters value**
Crossover	simulated binary crossover (SBX) [20]	0.90
Mutation	polynomial mutation [21]	0.10
Selection	tournament selection [54]	–
Population size	—	200
No. of offsprings	—	2

Each test problem (TP1-TP10) is solved 11 times using the IP, GA, and BOBD methods and the corresponding objective function values are recorded. For each TP, the best feasible objective function value (obtained or known from the literature) is recorded. Every time a method is executed, we evaluate the quality of solution in terms of absolute deviation, which is the absolute difference in the solution obtained from the method and the best known solution. These deviations are illustrated using box-plots in Figure 10 and Figure 11. The figures indicate that BOBD method consistently yields the best solutions across all 11 runs for every test problem. In contrast, IP and GA frequently converge to local optima and, in several cases, even provide infeasible solutions. The BOBD method either matches or outperforms the solutions obtained by IP and GA in all instances. In the case of BOBD, the solutions are at least the same or better compared to the results obtained from IP and GA. For certain problems in figures, BOBD performance appears to be slightly worse than IP. This is because, in such cases, both BOBD and IP have converged close to the optimum and the difference in solution quality is marginal.

We also record the average computational time for solving all instances using BOBD. The IP method typically terminates within 1–10 seconds for most of the test problems. Computational times for BOBD method are provided in Table 11. For a fair comparison, GA is allowed to run for twice the time taken by the BOBD method for each instance. Overall, the data in Figure 10, Figure 11, and Table 11 convey that IP method provides the solutions quickly but it often converges to suboptimal solutions. GA frequently struggles to find even a feasible solution, particularly in medium and large scale scenarios. BOBD method took more computational time than IP, but it consistently delivers high-quality solutions during all 11 runs, which demonstrates better accuracy and repeatability of BOBD compared to IP and GA methods.

Table 11: Average computational time for BOBD method (in seconds)

Test Problem	$\lvert y\rvert+\lvert z\rvert=0$	$\lvert y\rvert+\lvert z\rvert=20$	$\lvert y\rvert+\lvert z\rvert=50$
TP1	1.90	-	-
TP2	4.00	-	-
TP3	3.34	-	-
TP4	3.86	28.62	32.49
TP5	4.34	8.77	12.33
TP6	7.72	8.26	21.00
TP7	8.33	11.26	51.21
TP8	16.16	4.83	5.79
TP9	14.83	291.26	496.70
TP10	27.23	28.67	32.42



Best Objective Function Value Obtained for Each Test Problem

Test Problem (TP)	$\lvert y\rvert + \lvert z\rvert = 0$	$\lvert y\rvert + \lvert z\rvert = 20$	$\lvert y\rvert + \lvert z\rvert = 50$
TP1	-13.402	-	-
TP2	193.724	-	-
TP3	-14.402	-	-
TP4	-16.652	-12.151	-12.162
TP5	143.790	189.923	259.125
TP6	-310	-550	-910
TP7	-380.323	-390.124	-404.825
TP8	-104.667	-173.479	-293.594
TP9	7049.244	7049.244	7048.701
TP10	-30665.539	74551.524	232377.092

Error in objective function values = absolute deviation from the best objective function value:

z_m = set of obj. func. values related to feasible solutions of TP obtained using method m

$z = \{z_{IP}, z_{GA}, z_{BOBD}\}$ = objective function values obtained by solving TPx using all methods

Absolute deviation in objective function values $z_m = |z_m - min(z)|$

Figure 10: Absolute deviation in objective function values from 11 runs: TP1-TP5

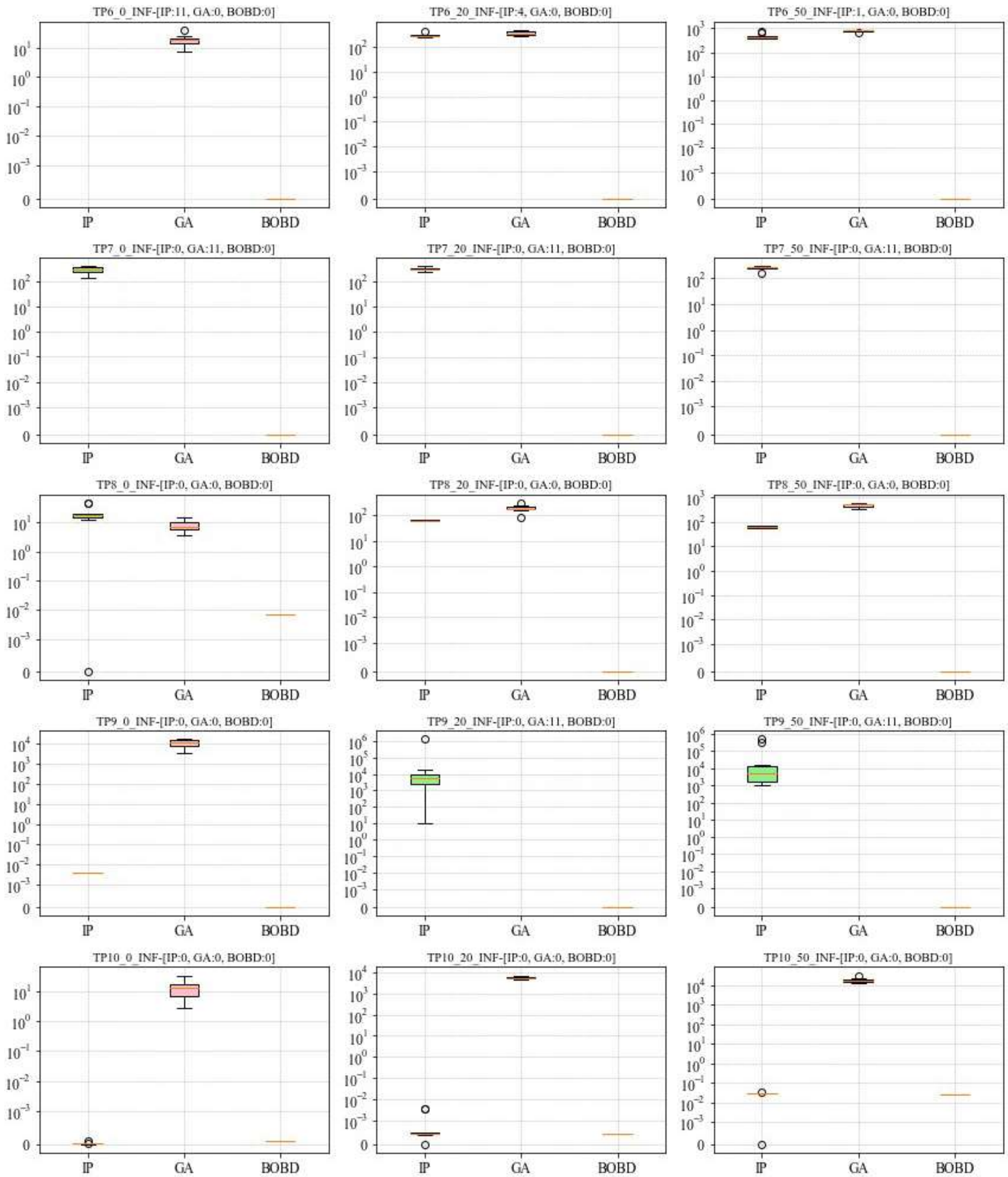


Figure 11: Absolute deviation in objective function values from 11 runs: TP6-TP10