AutoOpt: A Dataset and a Unified Framework for Automating Optimization Problem Solving

Authored by: Ankur Sinha, Shobhit Arora, Pujara Dhaval

Presented by: Dhaval Pujara

Affiliation:
Brij Disa Centre for Data Science and Artificial Intelligence,
Indian Institute of Management Ahmedabad (IIM A), India

Flow of Presentation

Contribution 1

▪ AutoOpt-11k: Dataset of Optimization Formulations

- Composition of *AutoOpt-11k* dataset
- Diversity in *AutoOpt-11k* dataset based on the characteristics of optimization problems
- Labels for *AutoOpt-11k* dataset: LaTeX and PYOMO scripts for optimization problems
- Annotation procedure for AutoOpt-11k dataset

Contribution 2

• AutoOpt: Framework for Automating Optimization Problem-Solving Task

- Overview of AutoOpt Framework (Series of Modules: M1- M2- M3)
- Details of Module M1(*Image_to_Text*): Generates a LaTeX code corresponding to given optimization problem
- Details of Module M2(*Text_to_Text*): Generates a PYOMO script from the LaTeX code

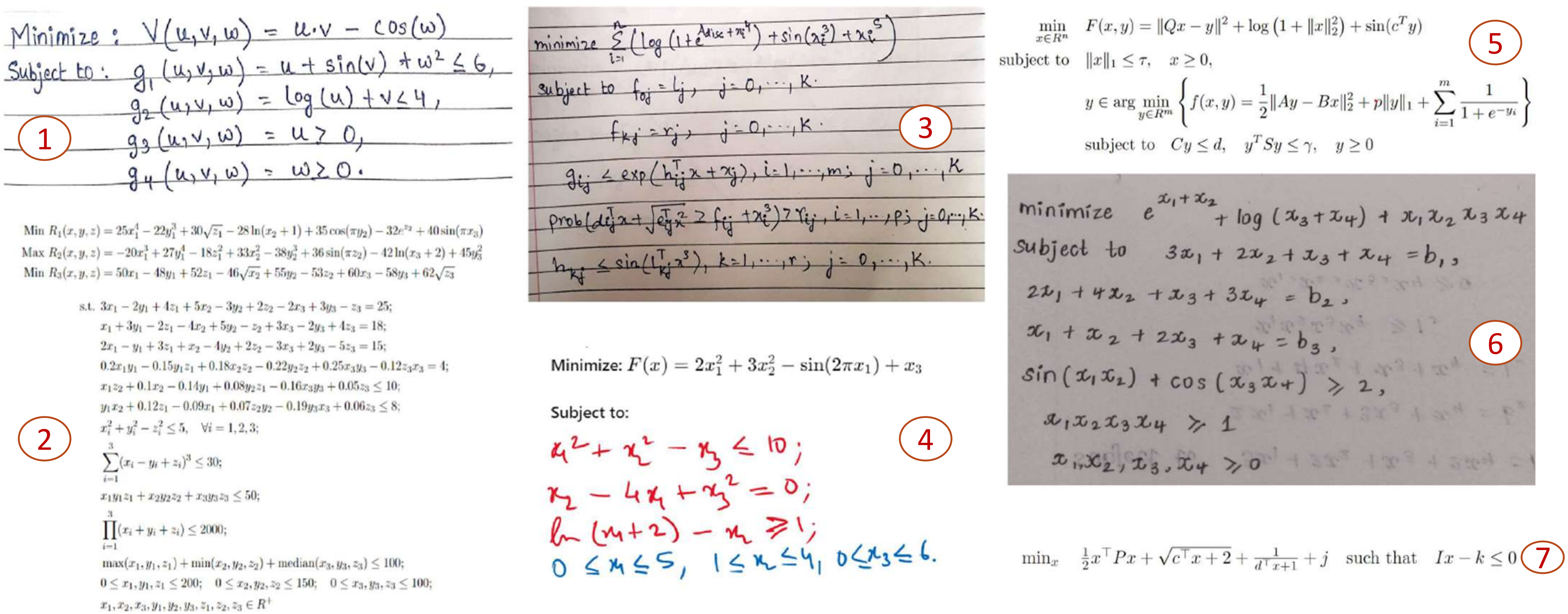
Deep Learning Models

- Details of Module M3(*Optimization*): Solves the optimization problem from the PYOMO script (Optimization Solver)
- Performance of AutoOpt framework

AutoOpt-11k: Dataset of Optimization Formulations

What is *AutoOpt-11k* composed of?

- It contains the images of ***11,554*** mathematical models written by hand or typed in the computer system.



A sample of 7 Optimization Problems Images from AutoOpt-11k dataset

- Mathematical models in *AutoOpt-11k* are corresponding to optimization problems observed in various domains:
  - ❑ Business ❑ Engineering ❑ Science

Diversity in AutoOpt-11k Dataset

Characteristics of optimization problems

	Type	Count	Description
AutoOpt-11k	Handwritten	5,070	Written by hand on paper, tablet, electronic book, etc.
	Typeset	6,484	Printed format extracted from books, articles, etc.
	Total	11,554	
Types of Problems	Single-objective	10,838	Only one objective function is defined
	Multi-objective	159	Contains multiple objective functions
	Multi-level	399	Contains two or more levels of optimization
	Uncertainty	158	Contains some form of parameter or variable uncertainty
Constraint Availability	Unconstrained	155	Constraints are absent
	Constrained	11,399	One or more constraints are present
Model Form	General Form	7,349	Contains undefined parameters, functions, etc.
	Fully Defined	4,205	Completely defined with all necessary parameters
Presentation Form	Vector Form	608	Defined in a form containing vector and matrix operations
	Scaler Form	10,246	Defined in a form containing only scalar operations
	Scalable Form	804	Problem is scalable in terms of variables, objectives, etc.
Other (Complexities)	Linear	2,130	All objectives and constraints are linear
	Non-linear	9,122	One or more objectives or constraints are non-linear
	Continuous	10,806	All variables and functions are continuous
	Discontinuous	424	Involves integer variables or contains discontinuities
	Convex	2,580	Belongs to the class of convex optimization
	Non-convex	3,574	Belongs to the class of non-convex optimization
	Differentiable	9,502	All the functions defined are differentiable
	Non-differentiable	502	Some functions defined are not differentiable

There are formulations that belong to multiple categories and also formulations that cannot be classified appropriately.

Diversity in AutoOpt-11k Dataset (Cont.)

Variation due to Optimization Problem Declaration Style

▪ *Orientation*	Horizontal or Vertical (objective function and constraints written in single or multi line)
▪ *Objective function mentioning style*	*min/max* OR *minimize/maximize*
▪ *Objective function & constraint separators*	*s.t.* , *w.r.t.*, *subject to*, etc.
▪ *Constraint indexing style*	$i \in [1, K]$ or $i = 1, \ldots, K$
▪ *Math expression writing*	$x^{0.5}$, $x^{1/2}$, $\sqrt{x}$
▪ *Variable name style*	$p/q/r$, $a/b/c$, $x_1/x_2/x_3$, $p/a/x_1$

Variation due to Different Hand-writing Styles

▪ Writing style	▪ Font style and font size
▪ Paper type	▪ Plain or ruled
▪ Ink color	▪ Black, red, blue

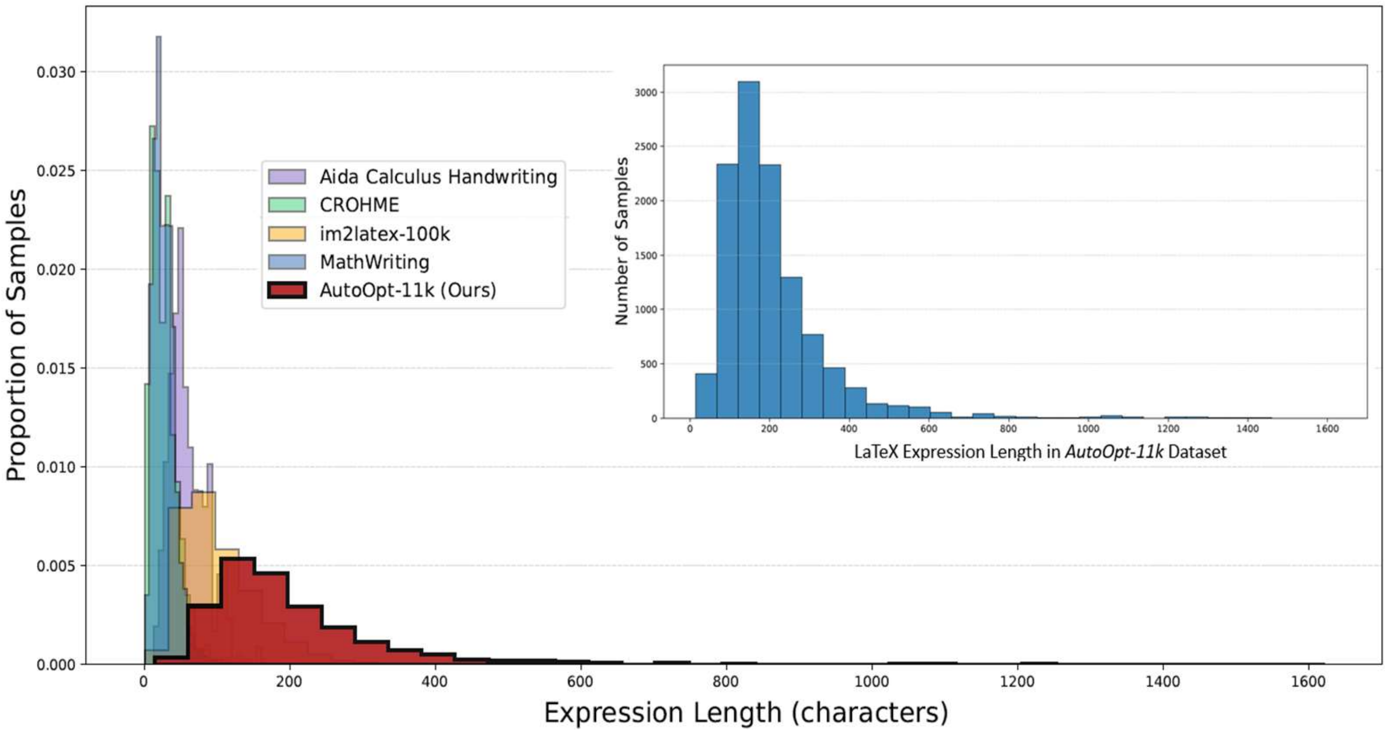
Variation due to Image Capturing Style

▪ Camera angle	▪ Light intensity
▪ Distance	▪ Camera specification
▪ Orientation	

Optimization models with all mentioned variations are included in *AutoOpt-11k* dataset.

AutoOpt-11k Dataset: Labels and Annotation

AutoOpt-11k also contains the **LaTeX code for all mathematical formulations.**



❖ *AutoOpt-11k* also contains the **PYOMO script for a subset of 11k problems** (PYOMO scripts for 1018 optimization problems).

Annotation

❖ **25 experts** participated in *AutoOpt-11k* dataset formation.

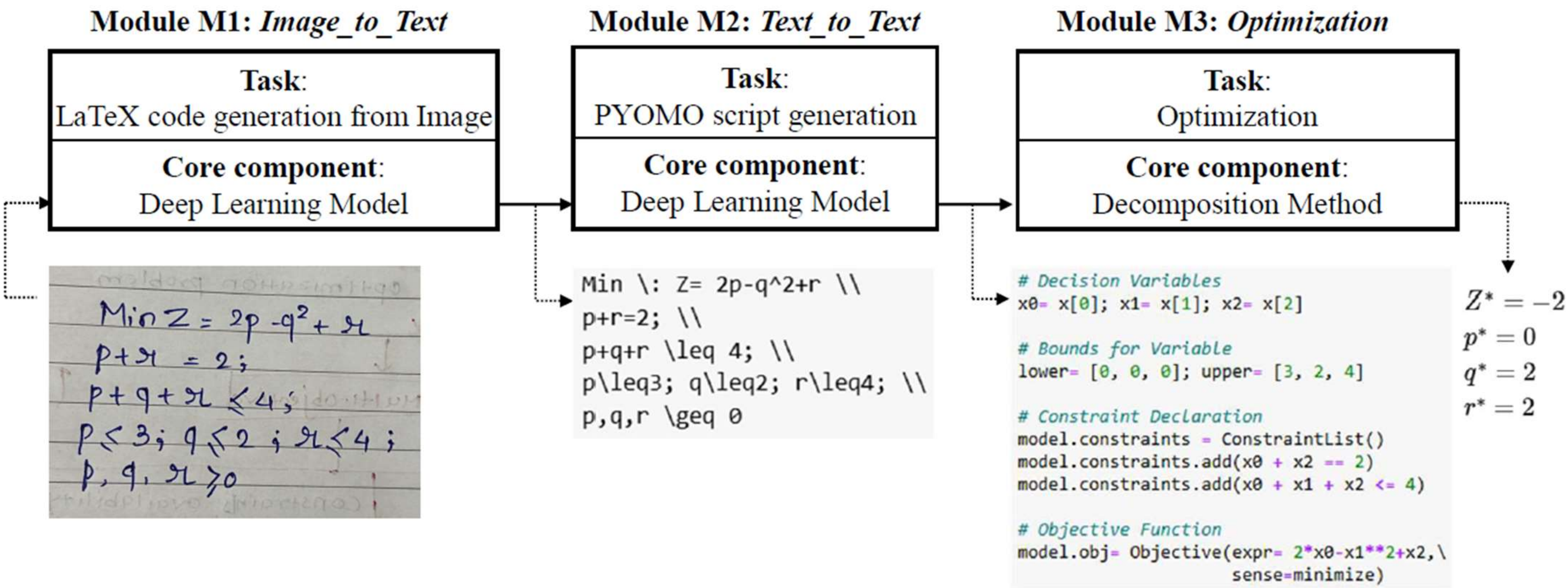
❖ **5 annotators** (A1, A2, A3, A4, A5) performed **cross-verification**.

Inter Annotator Agreement (IAA) scores

Pair	BLEU score		CER score	
	Mean	Std	Mean	Std
A1 vs A2	0.8187	0.1066	0.1784	0.1154
A1 vs A3	0.8185	0.1065	0.1788	0.1153
A1 vs A4	0.8195	0.1071	0.1776	0.1167
A1 vs A5	0.8201	0.1068	0.1782	0.1155
A2 vs A3	0.8588	0.1031	0.1267	0.1020
A2 vs A4	0.8581	0.1042	0.1273	0.1040
A2 vs A5	0.8189	0.1062	0.1791	0.1148
A3 vs A4	0.8574	0.1044	0.1286	0.1050
A3 vs A5	0.8192	0.1064	0.1787	0.1150
A4 vs A5	0.8197	0.1070	0.1779	0.1159

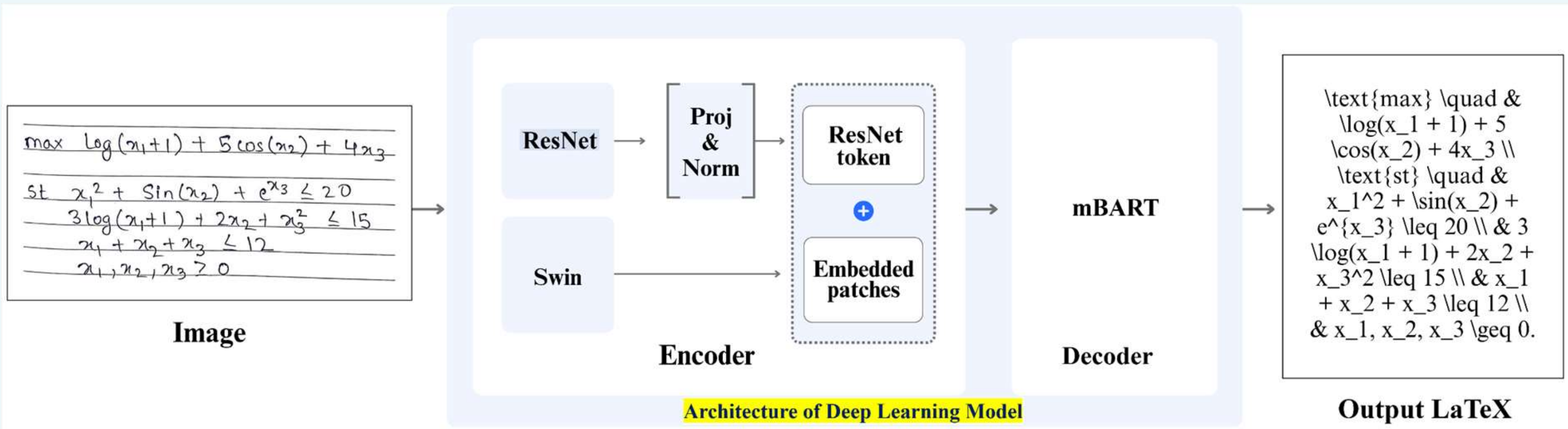
CER= Character Error Rate

AutoOpt: Framework for Automating Optimization-Problem-Solving Task



Module M1: Deep Learning Model for Image_to_Text

Generates a LaTeX code corresponding to the optimization formulation in image



Characteristics of Deep Learning Model

- **Task:** Mathematical Expression Recognition (MER)
- **Encoder**: ResNet (CNN) + Swin Transformer
- **Inspiration:** NOUGAT
- **Decoder:** mBART

Encoder

- **ResNet**: Extracts local features (e.g., symbols, strokes, shapes)
- **Swin Transformer:** Identifies global structures (e.g., (sub/super)scripts, matrices, & fractions)
- **Result:** Integrate the outputs of ResNet (feature vector) & Swin transformer (patch embeddings)

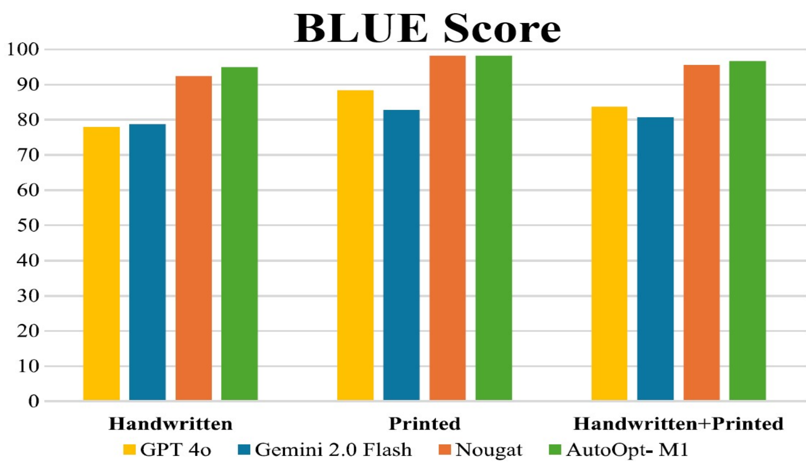
Decoder

- **mBART**: A pre-trained transformer-based autoregressive decoder; generates LaTeX code token wise

Model Training

- **Approach**: Transfer learning
- **ResNet**: ImageNet weights
- **Swin & mBART**: NOUGAT weights

Performance of Module M1



Character Error Rate (CER)			
Model (Model Size)	Handwritten (HW)	Printed (PR)	HW+PR
	Character Error Rate		
GPT 4o (Large)	0.1465	0.0664	0.1017
Gemini 2.0 Flash (Large)	0.1607	0.1047	0.1338
Nougat (348.7M)	0.0752	**0.0168**	0.0440
AutoOpt-M1 (393.3M)	**0.0412**	0.0176	**0.0286**

No.	Architecture	BLEU Score	CER
1.	with CNN + without Transformer	16.10	0.8812
2.	without CNN + with Transformer	95.51	0.0440
3*.	with CNN + with Transformer	96.70	0.0286

Ablation Study:
Trial of Multiple Architectures for Deep Learning Model

* Current Study

Model Training Setup

- **Epochs:** 180
- **GPU:** NVIDIA A100 (Google Colab Pro)
- **Learning Rate**: 2e^{-5}
- **Optimizer:** AdamW
- **Scheduler**: Cosine
- **Weight Decay:** 0.02
- **Batch Size**: 8
- **Gradient Accumulation**: 2
- **Effective Batch Size:** 16

Module M2: Deep Learning Model for Text_to_Text

Generates a problem specific PYOMO script from the LaTeX code

Deep Learning Model Development AutoOpt-M2

- **Approach:** Transfer Learning
- **Parent Model**: DeepSeek-Coder 1.3B
- **Training**: Fine-tuned on 1018 models
- **BLEU Score**: 88.25
- **CER**: 0.0825

Computational Resources and Model Training Setup

- **Epochs:** 15
- **GPU:** NVIDIA A100 (Google Colab Pro)
- **Precision:** Mixed-precision (fp16)
- **Learning Rate**: $5e^{-5}$
- **Batch Size**: 2
- **Gradient Accumulation**: 4
- **Effective Batch Size:** 8
- **Weight Decay:** 0.01

Module M3: Optimization Method

Solves the optimization problem contained in PYOMO script using a developed Bilevel Optimization based Decomposition (BOBD) method

Bilevel Decomposition- *Transform a general optimization problem into bilevel optimization problem* as follows:

1). ***Classify each decision variable*** x_i $(i = 1, \ldots, n)$ into upper-level or lower-level categories:

upper-level variables (u): causing complexities such as *non-convexity, discontinuity, non-differentiability*, etc.

lower-level variables (l): maintaining *linearity, convexity, and causing high-dimensionality*.

This study develops a *Logistic Regression based Variable Classification Model* to *automate the variable classification task*.

2). For general optimization problem, ***write its objective function*** $\boldsymbol{F(x)}$, ***constraints*** $\boldsymbol{G(x)}$ & $\boldsymbol{H(x)}$ ***in terms of upper-level and lower-level variables*** $(u$ & $l)$. That provides the bilevel optimization problem elements:

- upper-level objective function: $F(u, l)$
- upper-level constraints: $G(u.l)/H(u, l)$
- lower-level objective functions: $f(u, l)$
- lower-level constraints: $g(u, l)/h(u, l)$

$$\begin{aligned} \min_{x} \quad & F(x) \\ \textit{subject to} \quad & \\ & G_i(x) \le 0, \quad i = 1, \ldots, I \\ & H_j(x) = 0, \quad j = 1, \ldots, J \end{aligned}$$

Bilevel Decomposition $\Longrightarrow$

$$\begin{aligned} \min_{u,l} \quad & F(u, l) \\ \textit{subject to} \quad & \\ & l \in \arg\min_{l} \{ f(u, l) : g_p(u, l) \le 0, \ p = 1, \ldots, P, \\ & \qquad\qquad\qquad h_q(u, l) = 0, \ q = 1, \ldots, Q \} \\ & G_i(u, l) \le 0, \quad i = 1, \ldots, I \\ & H_j(u, l) = 0, \quad j = 1, \ldots, J \end{aligned}$$

Module M3: Optimization Method (Cont.)

Stepwise Procedure of BOBD method

Algorithm	**Bilevel Optimization Based Decomposition (BOBD)**
Input:	$F(x)$, $G(x)$, $H(x)$- single level optimization problem (i.e., original problem)
Output:	x^*- the best solution obtained 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 and lower levels using LR-VCM.
Step 4:	*for* $g = 1$ to $number\_of\_generations$:
Step 5:	If g is divisible by *variable_classification_alternation_number*:
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 (GA).
Step 9:	For given u, obtain l by solving a corresponding lower level problem using the interior point or linear programming methods.
Step 10:	Update population $\mathcal{P}$ with new solutions if they are better than the worst solutions in $\mathcal{P}$.

Module M3: Optimization Method (Cont.)

Numerical Example to Demonstrate the Procedure of BOBD method

$$
\begin{aligned}
& Min\ Z = 2p - q^2 + r \\
& s.t. \quad p + r = 2; \\
& \qquad p + q + r \leq 4; \\
& \qquad p \leq 3;\ q \leq 2;\ r \leq 4; \\
& \qquad p,\ q,\ r \geq 0
\end{aligned}
$$

(A). General Problem

$$q = u_1 \qquad p = l_1 \qquad r = l_2$$

$$
\begin{aligned}
& \min_{u,l} \quad 2l_1 - u_1^2 + l_2 \\
& \text{subject to} \\
& l \in \operatorname{argmin} \left\{ \begin{array}{l} 2l_1 - u_1^2 + l_2 \\ \text{subject to} \\ l_1 + l_2 = 2; \\ l_1 + u_1 + l_2 \leq 4; \\ l_1 \leq 3; \quad l_2 \leq 4; \\ l_1, l_2 \geq 0 \end{array} \right\} \\
& 0 \leq u_1 \leq 2
\end{aligned}
$$

(B). Bilevel Decomposition

$$u_1 = 2 \qquad l_1 = 0 \qquad l_2 = 2$$

$$Z^* = -2 \qquad p^* = 0 \qquad q^* = 2 \qquad r^* = 2$$

(C). Solution

Performance of AutoOpt Framework

Module-level Evaluation

Modules	Performance Metrics	Reliability
module M1	CER: 0.0286	(1-CER)×100= 97.14%
module M2	CER: 0.0825	(1-CER)×100= 91.75%
module M3	There is no prediction task or error-prone task associated with module M3 as it contains an optimization solver that solves exactly what is provided in PYOMO script.	

Reliability of entire AutoOpt framework: (0.9714× 0.9175) × 100 = 89.12%

Note: 89.12% is actually a lower bound, as in many cases where the LaTeX or PYOMO is syntactically different from the expected output, the CER metric incorrectly counts such differences as character errors.

Framework-level Evaluation

- We *measure the performance of complete pipeline (M1–M2–M3)* on 500 sample problems outside the *AutoOpt-11k* dataset.
- The *overall success rate* (i.e., ability to correctly read the problem in LaTeX and PYOMO and subsequently deploy the solver successfully) was observed to be *94.20%*.

Conclusion

- Proposed *AutoOpt-11k*, a curated dataset comprising over 11,554 images of handwritten and typeset mathematical programs, labeled with corresponding LaTeX code for all images and modeling language script for a subset of images.

- This study introduced *AutoOpt*, an end-to-end automated framework that enables optimization problem-solving directly from images of mathematical formulations, thereby significantly reducing human intervention.

- Core component of each *AutoOpt* module, deep learning models for modules M1 & M2 and Bilevel Optimization based Decomposition (BOBD) method for module M3, is discussed along with their performance evaluation.

- The public release of the dataset and the framework is expected to encourage future research at the intersection of computer vision, natural language processing, and mathematical optimization.

Appendices: More Details About This Study

▪ Appendix A: AutoOpt-11k Dataset

- Deeper insights into the structure and content of *AutoOpt-11k* dataset
- Top 100 most frequent tokens in LaTeX
- Examples of LaTeX and PYOMO labels for optimization problems in *AutoOpt-11k* dataset

▪ Appendix B & C: Module M1 & M2

- Standard Deviation of BLUE score and CER from 5 runs of various models for modules M1 & M2:
  - ❑ GPT-4o ❑ Gemini 2.0 Flash ❑ Nougat ❑ AutoOpt M1/M2
- Convergence plot for AutoOpt-M1

▪ Appendix D: Module M3 (BOBD method)

- Deeper insights into BOBD method (with mathematical definition of Bilevle)
- Logistic Regression based Variable Classification Method (LR-VCM) to automate the variable classification task in BOBD
- Test suite of 10 test problems (TP1-TP10) to evaluate the performance of BOBD method.
- Compare the performance of BOBD method with metaheuristic (genetic algorithm) and classical (interior point) methods.

References

[1] Ali AhmadiTeshnizi, Wenzhi Gao, and Madeleine Udell. Optimus: Scalable optimization modeling with (mi) lp solvers and large language models. *arXiv preprint arXiv:2402.10172*, 2024.

[2] Shabbir Ahmed and Alexander Shapiro. The sample average approximation method for stochastic programs with integer recourse. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[3] AMPL. *AMPL: A Modeling Language for Mathematical Programming.* AMPL Optimization Inc., 2023. Version 2023.1.

[4] Neculai Andrei. An unconstrained optimization test functions collection. *Adv. Model. Optim*, 10(1):147–161, 2008.

[5] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms.* John wiley & sons, 2006.

[6] John E. Beasley. Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[7] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. Robust optimization. *Mathematics of Operations Research*, 34(2):1–38, 2009.

[8] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.

[9] John R Birge and Francois Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[10] Robert E. Bixby. The zib challenge and the state of mixed-integer programming. *Annals of Operations Research*, 149(1):37–41, 2007.

References

[11] Jacek Blazewicz, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.

[12] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023.

[13] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[14] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000.

[15] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–35, 2021.

[16] COIN-OR Foundation. *CBC User Guide*, 2023. Version 2.10.10, Accessed: 2025-05-02.

[17] COIN-OR Foundation. COIN-OR: Computational infrastructure for operations research, 2023. Accessed: 2025-05-02.

[18] George B Dantzig. *Linear programming and extensions.* Princeton university press, 2016.

[19] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Chichester, UK, 2001.

[20] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[21] Kalyanmoy Deb and Debayan Deb. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, 4(1):1–28, 2014.

[22] Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1141–1148, 2006.

References

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[24] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989. PMLR, 2017.

[25] Sneha Dhyani Bhatt, Sachin Jayaswal, Ankur Sinha, and Navneet Vidyarthi. Alternate second order conic program reformulations for hub location under stochastic demand and congestion. *Annals of Operations Research*, 304:481–527, 2021.

[26] Max Fehr. Optimization methods in finance by gerard cornuejols, reha tutuncu, 2007.

[27] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.

[28] Christodoulos A Floudas and Panos M Pardalos. *A collection of test problems for constrained global optimization algorithms*. Springer, 1990.

[29] Christodoulos A Floudas, Panos M Pardalos, Claire Adjiman, William R Esposito, Zeynep H Gümüs, Stephen T Harding, John L Klepeis, Clifford A Meyer, and Carl A Schweiger. *Handbook of test problems in local and global optimization*, volume 33. Springer Science & Business Media, 2013.

[30] John J. Forrest. Clp user guide, 2005. COIN-OR Linear Programming (Clp) Solver.

[31] GAMS Development Corporation. *General Algebraic Modeling System (GAMS) Documentation*. GAMS Development Corporation, 2023. GAMS Documentation, Version 41.

[32] Philippe Gervais, Asya Fadeeva, and Andrii Maksai. Mathwriting: A dataset for handwritten mathematical expression recognition. *arXiv preprint arXiv:2404.10690*, 2024.

References

[33] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[34] Andrea Grosso, A. Reza Jamali, and Marco Locatelli. Finding multiple local minima in chemical and biochemical engineering problems. *Computers & Chemical Engineering*, 33(7):1133–1142, 2009.

[35] Bertrand Guenin, Jochen Könemann, and Levent Tuncel. *A gentle introduction to optimization*. Cambridge University Press, 2014.

[36] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y.K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

[37] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Gurobi Optimization, LLC, 2025. https://docs.gurobi.com/projects/optimizer/en/current/index.html.

[38] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. *Pyomo – Optimization Modeling in Python*. Sandia National Laboratories, 2023. Pyomo Documentation, Version 6.6.2.

[39] David M. Himmelblau. Application of nonlinear programming in chemical engineering. *Chemical Engineering Science*, 41(8):1973–1987, 1986.

[40] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

References

[41] IBM Corporation. *User's Manual for CPLEX*. IBM, 2025. https://www.ibm.com/docs/en/icos/22.1.2?topic=optimizers-users-manual-cplex.

[42] Eitan Israeli and R. Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[43] Stanislav Hristov Ivanov. Automated decision-making. *foresight*, 25(1):4–19, 2023.

[44] JuMP Developers. *JuMP Documentation*. JuMP Community, 2023. JuMP.jl, Version 1.9.

[45] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[46] Jing J Liang, Thomas Philip Runarsson, Efren Mezura-Montes, Maurice Clerc, Ponnuthurai Nagaratnam Suganthan, CA Coello Coello, and Kalyanmoy Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics*, 41(8):8–31, 2006.

[47] LINDO Systems Inc. *LINDO API User Manual*. LINDO Systems Inc., 2023. Version 14.0.

[48] Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. *arXiv preprint arXiv:2401.02051*, 2024.

[49] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

References

[50] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[51] Ladislav Lukšan and Jan Vlcek. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report, Technical report, 2000.

[52] Zeyuan Ma, Hongshu Guo, Jiacheng Chen, Guojun Peng, Zhiguang Cao, Yining Ma, and Yue-Jiao Gong. Llamoco: Instruction tuning of large language models for optimization code generation. *arXiv preprint arXiv:2403.01131*, 2024.

[53] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. *Optimization with constraints*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.

[54] Brad L Miller, David E Goldberg, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995.

[55] MIPLIB. Miplib: Mixed integer programming library, 2023. Accessed: 2025-05-02.

[56] Netlib. Netlib linear programming library, 2023. Accessed: 2025-05-02.

[57] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[58] OR-Library. Or-library: Operations research test problems, 2023. Accessed: 2025-05-02.

[59] Aida Pearson. Aida calculus math handwriting recognition dataset. `https://www.kaggle.com/datasets/aidapearson/ocr-data`, 2020. Synthetic handwritten calculus math expressions for recognition and OCR tasks.

References

[60] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 367–376, 2021.

[61] Michael Pinedo and Xiuli Chao. Operations scheduling with applications in manufacturing and services. *International Transactions in Operational Research*, 6(5):441–453, 1999.

[62] Prasanna Ramamoorthy, Sachin Jayaswal, Ankur Sinha, and Navneet Vidyarthi. Multiple allocation hub interdiction and protection problems: Model formulations and solution approaches. *European Journal of Operational Research*, 270(1):230–245, 2018.

[63] Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Hoboken, NJ, 4 edition, 2009.

[64] UCI Machine Learning Repository. Uci machine learning repository, 2023. Accessed: 2025-05-02.

[65] R Clark Robinson. *Introduction to mathematical optimization*. Department of Mathematics, Northwestern University, Illinois US, 2013.

[66] Felix M Schmitt-Koopmann, Elaine M Huang, and Alireza Darvishy. Accessible pdfs: applying artificial intelligence for automated remediation of stem pdfs. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–6, 2022.

[67] Felix M Schmitt-Koopmann, Elaine M Huang, Hans-Peter Hutter, Thilo Stadelmann, and Alireza Darvishy. Mathnet: a data-centric approach for printed mathematical expression recognition. *IEEE Access*, 2024.

[68] NEOS Server. Neos server: The optimization server, 2023. Accessed: 2025-05-02.

[69] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Test problem construction for single-objective bilevel optimization. *Evolutionary computation*, 22(3):439–477, 2014.

References

[70] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Towards understanding bilevel multi-objective optimization with deterministic lower level decisions. In *Proceedings of the Eighth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2015)*. Berlin, Germany: Springer-Verlag, 2015.

[71] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22(2):276–295, 2017.

[72] Ankur Sinha, Dhaval Pujara, and Hemant Kumar Singh. Decomposition of difficulties in complex optimization problems using a bilevel approach. *arXiv preprint arXiv:2407.03454*, 2024.

[73] James C. Smith and Yinyu Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 201(3):1–14, 2008.

[74] Ray Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, 2007.

[75] James C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* Wiley-Interscience, Hoboken, NJ, 2003.

[76] George Stephanopoulos and Arthur W Westerberg. The use of hestenes' method of multipliers to resolve dual gaps in engineering system optimization. *Journal of Optimization Theory and Applications*, 15:285–309, 1975.

[77] Gilbert Syswerda. A study of reproduction in generational and steady-state genetic algorithms. In *Foundations of genetic algorithms*, volume 1, pages 94–101. Elsevier, 1991.

References

[78] Luis N. Vicente and Paul H. Calamai. Bilevel and multilevel programming: a bibliography review. *Journal of Global Optimization*, 5:291–306, 1994.

[79] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

[80] Xingyu Wu, Sheng-hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*, 2024.

[81] Andreas Wächter and Lorenz T Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[82] Yejing Xie, Harold Mouchère, Foteini Simistira Liwicki, Sumit Rakesh, Rajkumar Saini, Masaki Nakagawa, Cuong Tuan Nguyen, and Thanh-Nghia Truong. Icdar 2023 crohme: Competition on recognition of handwritten mathematical expressions. In *Document Analysis and Recognition - ICDAR 2023*, volume 14234 of *Lecture Notes in Computer Science*, pages 541–551. Springer Nature Switzerland, 2023.

[83] Norm Xu. Nougat-latex-ocr: Fine-tuning and evaluation of nougat-based image-to-latex models, 2025. Accessed: 2025-05-08.

[84] Shunyu Yao, Fei Liu, Xi Lin, Zhichao Lu, Zhenkun Wang, and Qingfu Zhang. Multi-objective evolution of heuristic using large language model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):27144–27152, 2025.

[85] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.