Learning to Optimize at Scale: A Benders Decomposition-TransfORmers Framework for Stochastic Combinatorial Optimization

Seung Jin Choi
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061
seungj9@vt.edu

Kimiya Jozani
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061
kimiya@vt.edu

Josh Cooper
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061
joshfcooper@vt.edu

Esra Buyuktahtakin Toy
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061
esratoy@vt.edu

Abstract

We propose a learning-augmented Benders decomposition framework to solve large-scale two-stage stochastic mixed-integer programs. We focus on the two-stage stochastic capacitated lot-sizing problem (TSSCLSP) under demand uncertainty. Our method accelerates the convergence of the decomposition by using a pre-trained TransfORmer model to rapidly generate high-quality approximate solutions for the scenario subproblems. This hybrid strategy uses the TransfORmer's predictions to generate strong optimality and feasibility cuts, effectively guiding the Benders master problem. Our framework includes a novel expandable generation mechanism, allowing a model trained on a fixed horizon to solve instances of arbitrary length. For the test set considered, our method solves instances up to T=270, a scale previously intractable for this approach, while maintaining zero infeasibility in the generated subproblem solutions. This demonstrates the potential of TransfORmers as powerful surrogate solvers embedded within classical decomposition algorithms.

Presented at the NeurIPS 2025 Workshop on Machine Learning and Operations Research (MLxOR). Non-archival workshop paper.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

The stochastic capacitated lot sizing (SCLSP) is a classic NP-hard problem that focuses on optimizing production planning in amidst uncertain demand and capacity limits. Traditional approaches, such as stochastic dynamic programming and decomposition techniques, face difficulties when applied at larger scales [Birge and Louveaux, 2011, Shapiro et al., 2009, Sox, 1997, Askin, 1981]. Classical techniques, including Benders' decomposition [Benders, 1962, Slyke and Wets, 1969], Lagrangian relaxation [Sen, 2005, Wolsey, 2020], hedging algorithms [Rockafellar and Wets, 1991, Watson and Woodruff, 2011, Haugen et al., 2001], Dantzig-Wolfe decomposition [Dantzig and Wolfe, 1960], sample average approximation [Shapiro et al., 2009, Kleywegt et al., 2002], and scenario reduction [Heitsch and Römisch, 2003] have been employed to tackle the complexity of SCLSPs, but still struggle in complex problems as the problem scales.

Recently, the integration of deep learning (DL) models, such as TransfORmers, in optimization problems has gained momentum to approximate feasible regions or optimal policies in combinatorial settings [Bengio et al., 2021, Khalil et al., 2017, Bushaj and Büyüktahtakın, 2024, Bushaj et al., 2023, Cooper et al., 2024]. DL has shown significant potential in large-scale and real-time planning scenarios, such as SCLSP [van Hezewijk et al., 2023, Temizöz et al., 2025, Felizardo et al., 2024, Yilmaz and Büyüktahtakın, 2023, Gong et al., 2024, Kärcher and Meyr, 2025]. Transformers [Vaswani et al., 2017] have recently emerged as a powerful tool for sequential decision-making, showing success in large-scale settings [Yilmaz and Büyüktahtakın, 2023, Lim and Zohren, 2021, Cooper et al., 2024, Yilmaz and Büyüktahtakın, 2024b,a,c]. Their self-attention mechanism enables efficient parallel processing and modeling of long-range dependencies, making them ideal for high-dimensional, dynamic problems, such as the SCLSP.

To address the scalability challenges of large-scale stochastic optimization and building on the emerging paradigm of ML-OR [Cooper et al., 2024, Yilmaz and Büyüktahtakın, 2024b,c], we propose a novel **TransfORmer-accelerated Benders decomposition** framework to solve the two-stage stochastic capacitated lot-sizing problem (TSSCLSP). Our method addresses the primary bottleneck of Benders decomposition—the iterative solving of numerous scenario subproblems—by replacing the exact subproblem solver with a TransfORmer-based surrogate. This model rapidly predicts near-optimal subproblem solutions, which are then used to generate strong optimality and feasibility cuts to guide the master problem. Critically, we introduce an **expandable, sliding-window generation technique** that enables a Transformer trained on a fixed-length horizon (e.g., T=90) to produce complete solutions for significantly longer, unseen horizons (e.g., T=270). This framework advances the state-of-the-art by coupling exact mathematical decomposition with scalable, learning-based inference, bridging the gap between optimization and machine learning. Although this paper focuses on the TSSCLSP, the approach is general and can be adapted to a wide range of dynamic two-stage stochastic optimization problems.

2 Mathematical Formulation

The mathematical model of TSSCLSP is presented in (1)–(5) [Büyüktahtakın, 2023]. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the set of time periods. Let $\omega \in \Omega$ index the demand scenarios. The deterministic parameters are: f_t (fixed production cost), g_t (unit production cost), and κ_t (production capacity) in period t. The stochastic parameters are: d_t^ω (demand in scenario ω), h_t (inventory holding cost) and p^ω (probability of scenario ω). The decision variables are: $Y_t \in \{0, 1\}$, a first-stage binary variable that indicates whether production is initiated; $X_t^\omega \geq 0$ and $S_t^\omega \geq 0$ are the second-stage continuous variables for quantity produced and inventory in period t under scenario ω.

$$\min \quad \sum_{t \in \mathcal{T}} f_t Y_t + \sum_{\omega \in \Omega} p^\omega \left(\sum_{t \in \mathcal{T}} g_t X_t^\omega + h_t S_t^\omega \right) \tag{1}$$

$$\text{s.t.} \quad S_{t-1}^\omega + X_t^\omega - d_t^\omega = S_t^\omega \qquad \forall \omega \in \Omega, \forall t \in \mathcal{T} \tag{2}$$

$$X_t^\omega \leq \kappa_t Y_t \qquad \forall t \in \mathcal{T}, \forall \omega \in \Omega \tag{3}$$

$$X_t^\omega, S_t^\omega \geq 0 \qquad \forall \omega \in \Omega, \forall t \in \mathcal{T} \tag{4}$$

$$Y_t \in \{0, 1\} \qquad \forall t \in \mathcal{T}. \tag{5}$$

The objective function (1) minimizes the total first-stage fixed setup costs and the expected second-stage production and inventory holding costs. Constraint (2) enforces inventory balance. Constraint (3) links production to the setup decision, subject to capacity κ_t. Constraints (4) and (5) define the variable domains.

3 Methodology

The extensive form in (1)–(5) is NP-hard and intractable for large-scale instances. We therefore employ a Benders decomposition, which we accelerate leveraging a TransfORmer model that solves scenario sub-problems.

3.1 Benders Decomposition Framework

The problem is decomposed into a Master Problem (MP) that determines the first-stage setup decisions (Y_t) and a set of independent, scenario-specific Subproblems (SP) that evaluate the consequences of those decisions. The MP minimizes setup costs plus an approximation of the expected recourse cost, represented by variables η_ω.

$$\textbf{(MP)} \quad \min \quad \sum_{t \in \mathcal{T}} f_t Y_t + \sum_{\omega \in \Omega} p^\omega \eta_\omega \tag{6}$$

$$\text{s.t.} \quad Y_t \in \{0, 1\} \quad \forall t \in \mathcal{T} \tag{7}$$

$$\eta_\omega \geq \theta_\omega^k + (\pi_\omega^k)^\top (Y - \hat{Y}^k), \quad \forall \omega \in \Omega, \ \forall k \in \mathcal{K}_\omega. \tag{8}$$

Constraint (8) represents the family of Benders optimality and feasibility cuts that iteratively approximate the expected recourse function, where $(\pi_\omega^k, \theta_\omega^k)$ are dual solutions obtained from the kth subproblem of scenario ω. For a fixed master solution $\hat{Y}$, each SP reduces to a Linear Program (LP); its dual is solved to generate these cuts, which are then added to the MP to iteratively refine the solution.

3.2 Learning-Augmented Subproblem Solving with TransfORmers

The bottleneck in Benders decomposition is the repeated solving of SPs. We accelerate this by using a TransfORmer model as a surrogate learning-to-optimize solver. The process is illustrated in Figure 1. Within each Benders worker: (i) A partial solution from the MP is encoded into a masked input tensor. (ii) The TransfORmer performs a single forward pass to complete the solution, predicting the binary Y_t variables for the full horizon of a specific scenario. (iii) With all Y_t variables fixed, the SP reduces to a fast LP, which is solved to obtain exact costs and duals. (iv) These duals are used to construct valid Benders cuts, which are sent back to the master.

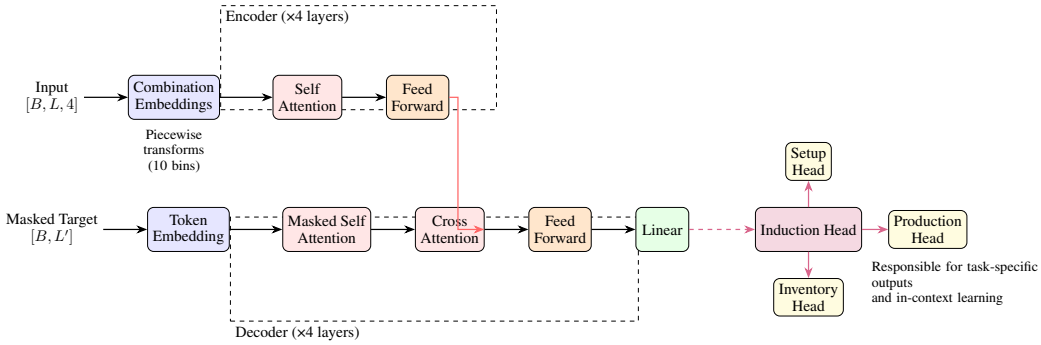

Figure 1: TransfORmer architecture used to solve subproblems in the decomposition framework.

3.3 Expandable Generation for Long-Horizon Problems

A key contribution is an expandable generation mechanism that allows a TransfORmer trained on a fixed horizon (e.g., $T = 90$) to solve instances of arbitrary length (e.g., $T = 270$). This is achieved via a sliding window method: (i) *Initial Generation*: The model generates a solution for the first window (e.g., periods 0-89). (ii) *Slide and Predict*: To generate the next chunk, the model's input uses

Table 1: TransfORmer hyperparameter configuration.

Parameter	Value
Parameter Count	6.8M
Source Vocabulary Size	16,000
Target Vocabulary Size	5
Embedding Dimensions	256
Training GPU Hours	<1

the last portion of the previous output as historical context, concatenated with any known future master variables and '[MASK]' tokens. (iii) *Iterative Completion*: This process repeats, sliding the window until a complete solution for the full length is constructed.

The model configuration is summarized in Table 1. To increase robustness, we apply 5-shot sampling per subproblem and select the best feasible prediction to construct cuts.

4 Results of Numerical Experiments

We evaluated the proposed decomposition-TransfORmer framework (ML-Benders) on two-stage TSSCLSP instances, using the benchmark style of Atamtürk and Munoz [2004] and Yilmaz and Büyüktahtakın [2023]. The TransfORmer model was trained exclusively on deterministic capacitated lot-sizing problem instances with a fixed horizon of $T = 90$. Our experiments are designed to test two key hypotheses: first, the viability and robustness of our expandable generation mechanism on long-horizon problems and second, the quality of the TransfORmer-based cuts at the model's native training horizon.

4.1 Scalability Demonstration on Long-Horizon Instances (T=270)

The primary test of our framework is its ability to scale to problem sizes far beyond its training data. We tested the model, trained only on $T = 90$ instances, on a set of TSSCLSP instances with a horizon of $T = 270$. This represents a challenging zero-shot generalization task. The key result is the successful application of the expandable generation mechanism. The framework was able to consistently generate complete, feasible solutions for all scenarios in these long-horizon problems. As shown in Table 2, this was achieved with only a modest increase in the number of Benders iterations compared to the baseline task.

Table 2: Performance of ML-Benders on the expandable long-horizon task. The model was trained only on $T = 90$ instances.

Test Horizon	**Avg. Iterations**	**Final Opt. Gap (%)**
$T = 90$ (Baseline Training Horizon)	51.6	3.55%
$\mathbf{T = 270}$ (Expandable Test)	**66.9**	**19.60%**

The final average optimality gap of 19.60% indicates a trade-off in solution quality when generalizing to a horizon three times the training length. However, the ability to generate valid, structured solutions for a problem of this complexity and scale—without any direct training and without causing subproblem infeasibility—is a significant demonstration of the model's robust generalization. This validates our expandable framework as a viable method for tackling large-scale instances that would otherwise be intractable.

4.2 Analysis of Cut Quality at the Training Horizon (T=90)

To understand the foundational performance that enables this scalability, we performed a detailed analysis comparing ML-Benders with classical Benders in 50 instances at the native training horizon of $T = 90$. These results highlight the superior quality of the cuts generated by solving sub-problems with TransfORmer.

4.2.1 Performance under a 300-Second Time Limit

In a time-constrained setting, the ML-Benders framework demonstrates clear superiority across all metrics, as summarized in Table 3.

The key findings are: (i) *Superior Solution Quality:* At its native horizon, ML-Benders achieves an average gap of only **3.55%**, a **91.5% reduction** compared to classical Benders (41.81%). (ii) *Enhanced Efficiency:* The method requires **29.4% fewer iterations** and **19.9% less time**, confirming that TransfORmer-generated cuts are highly effective in guiding the master problem.

Table 3: Average performance comparison under a 300-second time limit ($T = 90$, 10 scenarios, 50 instances).

Method	**Avg. Iterations**	**Avg. Time (s)**	**Final Opt. Gap (%)**
Classical Benders	73.1	308.2	41.81%
ML-Benders	**51.6**	**246.8**	**3.55%**

4.2.2 Cut-Budget Sensitivity (50-Cut Limit)

To directly test the quality of the generated cuts, we performed a "stress test" by imposing a strict budget of only 50 cuts. The computational results are presented in Table 4.

Table 4: Average optimality gap under a strict 50-cut limit.

Method	**Final Opt. Gap (%)**
Classical Benders	77.03%*
ML-Benders	**2.80%**

*Average calculated over 9 instances; one instance failed to find a feasible solution.

With a severely restricted budget, classical Benders fails to converge, whereas ML-Benders remains robust, achieving a final gap of just **2.80%**. This provides conclusive evidence that the TransfORmer-based surrogate generates dense, highly informative constraints. It is this foundational strength in generating high-quality cuts that allows the expandable mechanism to function effectively on much larger problems.

5 Discussion and Future Directions

This work integrates a TransfORmer-based predictor with a Benders-style decomposition, yielding a hybrid that couples the efficiency of deep learning with the rigor of mathematical optimization. The success of our framework, particularly its ability to solve $T = 270$ instances without generating infeasible subproblems, is a significant step towards building trust in ML-augmented optimization solvers. The *expandable generation mechanism* is a key contribution, proving that models trained on smaller, manageable problem sizes can generalize their learned policies to solve much larger, industrially-relevant instances. The framework is problem-agnostic and extensible to other two-stage settings. Future work will focus on completing the large-scale stochastic experiments, refining architectures, and exploring adaptive retraining to further improve cut quality and robustness.

References

Ronald G Askin. A procedure for production lot sizing with probabilistic dynamic demand. *AIIE Transactions*, 13(2):132–137, 1981.

Alper Atamtürk and Juan Carlos Munoz. A study of the lot-sizing polytope. *Mathematical Programming*, 99(3): 443–465, 2004.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer, 2nd edition, 2011.

Sabah Bushaj and İ Esra Büyüktahtakın. A k-means supported reinforcement learning framework to multi-dimensional knapsack. *Journal of Global Optimization*, 89(3):655–685, 2024.

Sabah Bushaj, Xuecheng Yin, Arjeta Beqiri, Donald Andrews, and İ Esra Büyüktahtakın. A simulation-deep reinforcement learning (sirl) approach for epidemic control optimization. *Annals of Operations Research*, 328(1):245–277, 2023.

İ Esra Büyüktahtakın. Scenario-dominance to multi-stage stochastic lot-sizing and knapsack problems. *Computers & Operations Research*, 153:106149, 2023.

Joshua F Cooper, Seung Jin Choi, and İ Esra Büyüktahtakın. Toward transformers: Revolutionizing the solution of mixed integer programs with transformers. In *IISE Annual Conference. Proceedings*, pages 1–6. Institute of Industrial and Systems Engineers (IISE), 2024. URL `https://arxiv.org/abs/2402.13380`.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960. doi: 10.1287/opre.8.1.101.

Leonardo Kanashiro Felizardo, Edoardo Fadda, Emilio Del-Moral-Hernandez, and Paolo Brandimarte. Reinforcement learning approaches for the stochastic discrete lot-sizing problem on parallel machines. *Expert Systems with Applications*, 246:1–13, 2024. doi: 10.1016/j.eswa.2023.123036.

H. Gong, Y. Zhang, and Z.-H. Zhang. Training demand prediction models by decision error for two-stage lot-sizing problems. *IEEE Transactions on Automation Science and Engineering (submitted)*, 2024.

K. K. Haugen, A. Løkketangen, and D. L. Woodruff. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research*, 132(1):116–122, 2001.

H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24(2-3):187–206, 2003.

J. Kärcher and H. Meyr. A machine learning approach for predicting the best heuristic for a large scaled capacitated lot sizing problem. *OR Spectrum*, 2025. Accepted Jan 2025.

Elias B Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

A. J. Kleywegt, A. Shapiro, and T. Homem de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

Suvrajeet Sen. Algorithms for stochastic mixed-integer programming models. *Handbooks in operations research and management science*, 12:515–558, 2005.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2009.

R. M. Van Slyke and R. J.-B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

Charles R Sox. Dynamic lot sizing with random demand and nonstationary costs. *Operations Research*, 45(6): 878–892, 1997.

Tarkan Temizöz, Christina Imdahl, Remco Dijkman, Douniel Lamghari-Idrissi, and Willem van Jaarsveld. Deep controlled learning for inventory control. *European Journal of Operational Research*, 324(1):104–117, 2025. doi: 10.1016/j.ejor.2025.01.026.

Lotte van Hezewijk, Nico P. Dellaert, Tom van Woensel, and Noud Gademann. Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. *International Journal of Production Research*, 61(6):1955–1978, 2023. doi: 10.1080/00207543.2022.2056540.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jean-Paul Watson and David L. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, 2011.

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

Dogacan Yilmaz and İ Esra Büyüktahtakın. Learning optimal solutions via an lstm-optimization framework. In *Operations Research Forum*, volume 4, page 48. Springer, 2023.

Dogacan Yilmaz and İ Esra Büyüktahtakın. A deep reinforcement learning framework for solving two-stage stochastic programs. *Optimization Letters*, 18(9):1993–2020, 2024a.

Dogacan Yilmaz and İ Esra Büyüktahtakın. An expandable machine learning-optimization framework to sequential decision-making. *European Journal of Operational Research*, 314(1):280–296, 2024b.

Dogacan Yilmaz and İ Esra Büyüktahtakın. A non-anticipative learning-optimization framework for solving multi-stage stochastic programs. *Annals of Operations Research*, pages 1–41, 2024c.