

ID-PAS+: Identity-Aware Predict-and-Search for Solving General Mixed-Integer Linear Programs

Junyang Cai¹, El Mehdi Er Raqabi^{2,3}, Pascal Van Hentenryck², Bistra Dilkina¹

¹University of Southern California, Los Angeles, CA, USA

²Georgia Institute of Technology, Atlanta, GA, USA

³Université Laval, Québec, Canada

{caijunya, dilkina}@usc.edu, {eraqabi6, pvh}@gatech.edu

Abstract

Mixed-Integer Linear Programs (MIPs) are powerful and flexible tools for modeling a wide range of real-world combinatorial optimization problems. Predict-and-Search methods operate by using a predictive model to estimate promising variable assignments and then guiding a search procedure toward high-quality solutions. Recent research has demonstrated that incorporating machine learning (ML) into the Predict-and-Search framework significantly enhances its performance. Still, it is restricted to binary-only problems and overlooks the presence of fixed variable structures that commonly arise in real-world settings. This work extends the current Predict-and-Search (PAS) framework to parametric MIPs and introduces ID-PAS+, an identity-aware learning framework that enables the ML model to handle heterogeneous variable types more effectively. Experiments on several real-world large-scale problems demonstrate that ID-PAS+ consistently achieves superior performance compared to the state-of-the-art solver Gurobi and PAS.

Introduction

This paper considers Parametric Mixed Integer Linear Programs (MIPs) of the form $P_I = (A, b, c)$ where we optimize for

$$\min\{c^\top x \mid Ax \leq b, x \in \mathbb{R}^n, x_j \in \mathbb{Z} \forall j \in I\},$$

where P is the instance data, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $I \subseteq \{1, \dots, n\}$ denotes the index set of general integer variables. The objective is to minimize $c^\top x$ subject to the linear constraints. Many such real-world MIPs contain millions of variables and constraints, making exact optimization challenging in environments where they must be solved quickly. However, in many industrial settings, these MIPs are repeatedly solved for instances drawn from distributions that are learned from historical data and/or forecasts. Machine learning (ML) offers an opportunity to reduce the search space of these MIPs by learning patterns from historical optimal or high-quality solutions.

Recent work in this area has integrated ML with solvers, for example, through large neighborhood search (Song et al. 2020; Huang et al. 2023; Cai, Kadioğlu, and Dilkina 2025),

optimization proxies (Chen, Tanneau, and Van Hentenryck 2023; Tanneau and Van Hentenryck 2024), or predict-then-optimize pipelines (Tang and Khalil 2024; Er Raqabi, Wu et al. 2024; Mohan, Er Raqabi, and Van Hentenryck 2025). A complementary line of research develops ML-based methods that generate high-quality feasible solutions. The Predict-and-Search (PAS) framework (Han, Yang et al. 2023), inspired by trust-region methods, explores a local neighborhood around a predicted partial assignment rather than fixing variable values outright, often yielding better solutions and reducing the risk of infeasible solutions. Recent improvements, such as contrastive learning (Huang et al. 2024a) and multitask learning (Cai, Huang, and Dilkina 2025), further enhance PAS performance. Yet, these methods remain largely limited to binary problems, whereas real-world MIPs frequently involve general integer variables (Greening, Dahan, and Erera 2023; Akhlaghi, Zandehshahvar, and Van Hentenryck 2025; Kim et al. 2025). In addition, prior evaluations focus mainly on synthetic rather than large-scale industrial benchmarks.

Careful analysis of real-world Parametric MIPs—such as those governing power systems (Er Raqabi, Bani et al. 2025), supply chains (Er Raqabi, Beljadid et al. 2025), planning (Cai, Huang et al. 2025), scheduling (Ye et al. 2025), and rail networks (Kim et al. 2025)—reveals two critical, shared properties. First, a substantial fraction of their variables take a value of exactly *zero* in optimal or high-quality solutions (Er Raqabi, Himmich et al. 2023). Second, since they model slowly changing physical infrastructures, the *identity* of the variables remains consistent across instances (e.g., a specific variable always represents the same factory or train route, even as daily demand fluctuates) (Akhlaghi, Zandehshahvar, and Van Hentenryck 2025).

The first observation is the key to extending PAS to general integer optimization. Predicting the exact value of a general integer variable is computationally inefficient and highly susceptible to error. However, by shifting the focus specifically to predicting which variables are *exactly zero*, we can drastically prune the search space with high confidence. A non-zero value in real-world settings typically represents an active, costly decision (e.g., opening a facility or allocating a discrete batch of resources); consequently, the vast majority of available options remain unselected. Focusing on zero-values not only improves the model’s accuracy

by targeting the most common state, but also provides the exact mechanism needed to generalize the PAS framework from strictly binary settings to general integer problems.

Equally critical is the second observation regarding variable identity, which is essential for cross-instance generalization. Standard Graph Neural Networks (GNNs) represent MIPs as bipartite graphs and operate under permutation invariance. Consequently, they fail to recognize that “variable 1” in today’s instance structurally corresponds to “variable 1” in tomorrow’s instance. In real-world parametric MIPs, retaining this identity mapping is highly informative, as certain physical components consistently exhibit similar historical behaviors. By injecting identity-aware positional embeddings into the graph formulation, the model can learn and leverage these variable-specific historical priors, vastly improving its predictive accuracy on unseen instances drawn from the same underlying infrastructure.

Building on these two insights, this paper proposes ID-PAS+, a simple yet effective identity-aware framework that successfully bridges the gap between binary-only problems and general integer MIPs. ID-PAS+ generalizes the PAS approach to general MIPs by smartly predicting zero-valued variables, avoiding the combinatorial burden of predicting all integer values. Compared to standard permutation-invariant graph networks, ID-PAS+ incorporates identity-aware features to improve cross-instance generalization. Evaluated on three large-scale, real-world problems, ID-PAS+ demonstrates consistent and significant performance improvements over both standard Gurobi and the original PAS framework.

Background

Parametric MIP

Many early works focus on parametric linear programs (Gal 2009), where solution behavior under parameter changes is well understood. Parametric MIP, by contrast, concerns families of mixed-integer problems whose objectives or constraints depend on varying parameters, enabling analysis of how optimal solutions evolve across related instances. Parametric structure also plays a central role in areas such as online learning and control (Russo et al. 2023; Bertsimas and Stellato 2022). Recent progress within mixed-integer optimization has focused mainly on quadratic or nonlinear formulations (Ranjan and Stellato 2025; Pangia and Wiecek 2025), while general parametric linear problems remain far less explored.

Predict-and-Search

Predict-and-Search (Han, Yang et al. 2023) is a primal heuristic that leverages the prediction of the optimal solutions to guide the search process. Given a MIP instance P_I , let $p_\pi(x_i \mid P_I)$ denote the predicted probability for each binary variable x_i . PAS identifies near-optimal solutions by exploring a neighborhood informed by these predictions. Specifically, it selects k_0 binary variables X_0 with the smallest $p_\pi(x_i \mid P_I)$ and k_1 binary variables X_1 with the largest $p_\pi(x_i \mid P_I)$, ensuring X_0 and X_1 are disjoint ($k_0 + k_1 \leq q$). Variables in X_0 are fixed to 0, and those in X_1 are fixed to

1 in a sub-MIP. However, PAS allows up to $\Delta \geq 0$ of these fixed variables to be flipped during solving. Formally, let

$$B(X_0, X_1, \Delta) = \{x : \sum_{x_i \in X_0} x_i + \sum_{x_i \in X_1} (1 - x_i) \leq \Delta\}$$

be the neighborhood defined by X_0 , X_1 , and Δ , and let D represent the feasible region of the original MIP. PAS then solves the following optimization problem:

$$\min c^T x \quad \text{s.t.} \quad x \in D \cap B(X_0, X_1, \Delta).$$

Restricting the solution space to $D \cap B(X_0, X_1, \Delta)$ simplifies the problem, enabling the solver to find high-quality feasible solutions to the original MIP more efficiently.

The Identity-aware General PAS Framework

This paper extends the Predict-and-Search (PAS) framework, originally designed for binary MIPs, to general MIPs containing both binary and general integer variables. Inspired by (Akhlaghi, Zandehshahvar, and Van Hentenryck 2025), the core insight is to learn whether each discrete variable is zero, rather than attempting to predict its exact integer value. This strategy preserves compatibility with the trust-region search of PAS while directly exploiting the sparsity structure prevalent in real-world applications, where a vast majority of active decisions default to zero. Furthermore, this framework aligns with the reality of industrial settings: instances often share the exact same underlying variables and physical constraints, differing only in their parameters. ID-PAS+ leverages this by encoding variable identity, allowing the model to exploit consistent historical semantics across different instances. The entire pipeline is illustrated in Figure 1 and in the following subsections, we introduce these modules in detail.

Data Collection

For a given MIP P_I , the learning task collects a set of optimal or near-optimal solutions as training samples. Up to u_p distinct solutions with best objective values are extracted (in our experiments, $u_p = 50$). Because the primary objective of the model is to identify zero-valued variables, ID-PAS+ binarizes each training solution by keeping the zero entries as 0 and converting all non-zero entries (regardless of their exact integer value) to 1. This binarized representation provides a consistent, simplified learning target across both binary and general integer variables, drastically reducing the combinatorial complexity of the learning task.

Bipartite Graph Representation

Each MIP $P_I = (A, b, c)$ is represented as a bipartite graph following (Gasse et al. 2019; Cai, Huang, and Dilkina 2024). The graph $P' = (G, V, C, E)$ contains variable nodes V and constraint nodes C , with an edge $(i, j) \in E$ existing whenever variable i appears in constraint j with a non-zero coefficient. This representation uses 15 variable features (e.g., variable type, objective coefficient, bounds), 4 constraint

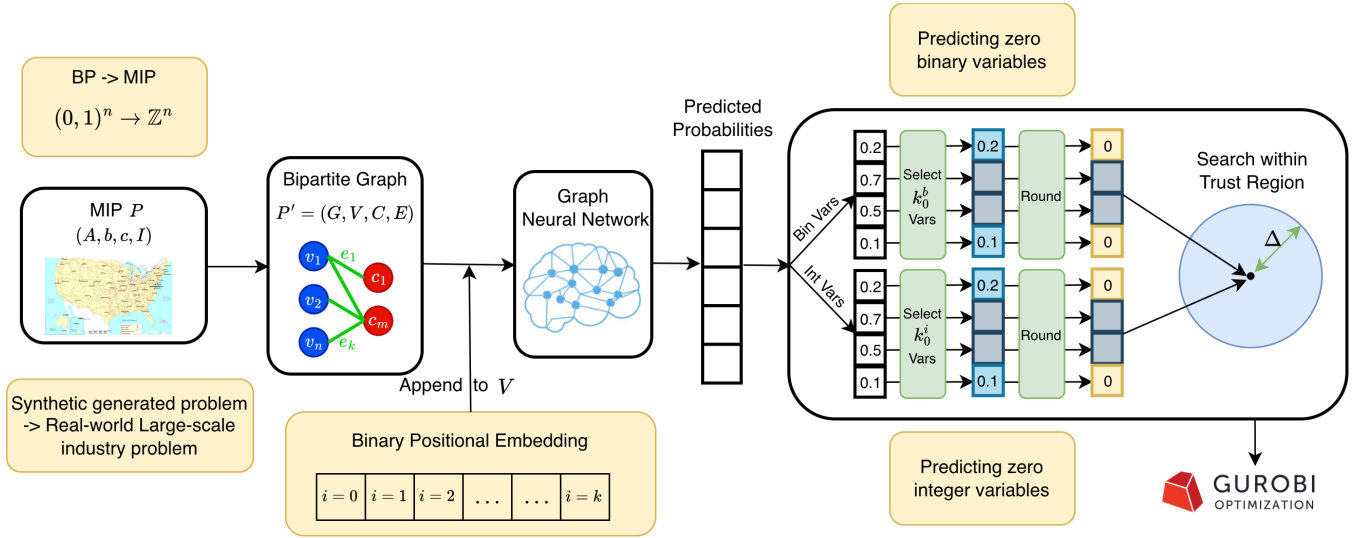


Figure 1. Overview of ID-PAS+ for general MIPs: A MIP instance is represented as a bipartite graph with identity-aware positional embeddings. A GNN predicts the probability that each variable is zero, then the selection process splits into two distinct, parallel tracks for binary and integer variables. High-confidence variables from both the binary (k_0^b) and integer (k_0^i) sets are selected, rounded, and fixed to zero. A subsequent search within a trust region is then performed before passing the reduced problem to a solver. The components highlighted in yellow denote the key extensions over PAS: binary positional embeddings (identity awareness), specific predictions for zero-valued binary and integer variables, and the generalization from synthetic binary problems to large-scale, real-world industry MIPs.

features (e.g., right-hand side and sense), and 1 edge feature (the constraint matrix coefficient).

Overcoming Permutation Invariance with Identity Awareness: While standard bipartite encodings are beneficial for synthetic benchmarks, they are inherently permutation-invariant. Real-world problems, however, are typically instances of a parametric MIP governing a fixed underlying infrastructure (e.g., a specific power grid or railway network). Standard graph networks cannot recognize that a specific node always corresponds to the same physical entity across different daily instances. To address this critical limitation, ID-PAS introduces *identity-aware features*. By appending a binary positional vector that encodes the unique index of each variable to the original feature set, ID-PAS+ breaks permutation invariance. This preserves the structural advantages of GNNs while enabling the model to recognize and learn from the historical behavior of specific variable identities across instances.

Model Architecture and Training Objective

ID-PAS+ learns a policy π parameterized by a Graph Attention Network (GAT) (Brody, Alon, and Yahav 2021). The network takes the bipartite graph P' as input and outputs a probability score for each variable. First, embedding layers map the node and edge features into a shared embedding dimension of size L . The GAT then performs two rounds of message passing: in the first round, each constraint node attends to its neighboring variable nodes using H attention heads; in the second round, variable nodes attend back to constraints with a separate set of attention parameters. The convolution and attention parameters are shared across the

graph, scaling the model size as $\Theta(HL^2)$. A multi-layer perceptron followed by a sigmoid activation then produces a probability score in $[0, 1]$ for each variable. In our experiments, we set the embedding dimension $L = 64$ and the number of attention heads $H = 8$.

ID-PAS+ trains the network using an imitation-learning objective (Han, Yang et al. 2023). Given the binarized target vector $y \in \{0, 1\}^{|I|}$ and the predicted probabilities $\hat{y} \in [0, 1]^{|I|}$, where I is the index set of all discrete (binary and integer) variables, ID-PAS+ optimizes a binary cross-entropy loss (Mao, Mohri, and Zhong 2023) over a batch of MIP instances $\mathcal{B}$:

$$\mathcal{L}(\theta) = - \sum_{P_I \in \mathcal{B}} \sum_{u \in u_p(P_I)} \sum_{i \in I} \left(y_i^u \log \hat{y}_i(\theta, P_I) + (1 - y_i^u) \log(1 - \hat{y}_i(\theta, P_I)) \right).$$

This simple yet effective loss encourages the network to assign the lowest possible probabilities to variables that consistently take the value of zero in high-quality solutions.

Applying the Learned Network

During inference, a new MIP instance is converted into a bipartite graph and evaluated by the GAT to produce continuous probability scores. In contrast to the original PAS framework (Han, Yang et al. 2023), which attempts to predict both zero binaries and one binaries, ID-PAS+ strictly focuses on zero-predictions and splits the outputs into two parallel sets: one for binary variables and one for general integer variables. The framework processes the scores to iden-

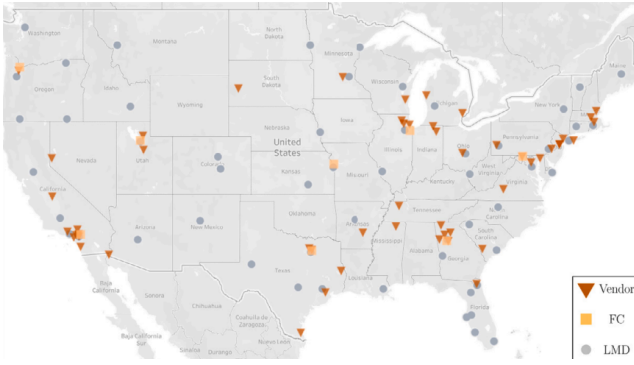


Figure 2. Middle-Mile Consolidation Network Design: example from (Greening, Dahan, and Erera 2023).

tify the variables most likely to be zero. Specifically, it selects the k_0^b binary variables and the k_0^i general integer variables that have the lowest predicted probabilities of being non-zero. These selected variables are rounded and form a combined set X_0 , where $|X_0| = k_0^b + k_0^i$. The variables in X_0 are initially fixed to 0.

To prevent the solver from being trapped by inaccurate predictions, similar to PAS, ID-PAS+ also defines a trust region that allows the solver to flip up to Δ of these fixed variables during optimization. The search is thus restricted to the PAS neighborhood:

$$B(X_0, \Delta) = \left\{ x : \sum_{x_i \in X_0} 1[x_i \neq 0] \leq \Delta \right\}.$$

The reduced MIP is then solved over the feasible region intersected with $B(X_0, \Delta)$. By treating binary and integer zero-predictions in parallel, ID-PAS+ provides highly reliable guidance that drastically prunes the search space while remaining robust to occasional misclassifications.

Real-world Benchmarks

Middle-Mile Consolidation Network Design

The first set of experiments considers the middle-mile consolidation network design problem (MMCNP) (Greening, Dahan, and Erera 2023; Huang et al. 2024b). The MMCNP focuses on planning transportation capacity from vendors to fulfillment centers (FCs) and last-mile delivery (LMDs) centers within required lead times (see Figure 2 for an example). Shipments must be routed and consolidated into scheduled loads to ensure efficient delivery. The planner must decide which connections to operate, how much capacity to allocate, and how to group shipments so all demand is delivered on time. The large problem setting includes 50 facilities, 2,000 paths, 400 arcs, and 300 commodities, and the very large (designated as *-Hard*) problem setting includes 200 facilities, 24,000 paths, 1,000 arcs, and 6,000 commodities. In the large setting, instances close the optimality gap in an average of 18 hours. In the very large setting, instances cannot converge and have an average gap of 5.12% after 24 hours, underscoring the difficulty of these models.



Figure 3. Strategic Locomotive Assignment: US railroad map (Maps of World 2025).

We assume that what varies across problem instances is the demand for each commodity. To generate instances, the experiments vary each commodity demand around its reference value using a normal distribution with bounds to keep demands realistic relative to arc capacities. Crucially, while daily demand fluctuates, the underlying physical infrastructure (the specific facilities and available transportation arcs) remains entirely fixed across instances. This static topology means that mathematical variables consistently represent the exact same physical connections, retaining a persistent identity characterized by historical routing behaviors. In this model, the general integer variables capture the number of trucks assigned to each arc. Since real consolidation networks operate sparsely, with only a highly optimized fraction of all possible connections used, the vast majority of these integer variables are exactly zero in high-quality solutions. Focusing on the underlying sparsity of these zero-valued integer variables provides a pathway to bypass the combinatorial complexity of determining exact truck counts.

Strategic Locomotive Assignment

The second set of experiments considers the strategic locomotive assignment problem (SLAP) (Kim et al. 2025). The SLAP determines a weekly, repeatable plan for assigning locomotives to a fixed train schedule. The problem specifies the required inputs, operational constraints, and cost components governing locomotive movements and work events such as pickups and set-outs. Its main outputs are the weekly locomotive flows across the network and the locations where work events occur. These strategic decisions, including fleet sizing and event placement, provide the foundation for subsequent tactical and operational planning. The large problem setting includes 1,000 nodes and 40,000 arcs, and the very large (designated as *-Hard*) problem setting includes 8,000 nodes and 50,000 arcs (see Figure 3 for an overview of the underlying US-wide railway network). In the large setting, instances reach optimality in an average runtime of five hours. In the very large setting, instances remain more demanding, and cannot close the gap even after 24 hours, with an average gap of 0.03%.

In both sets of SLAP instances, the experiment assumes

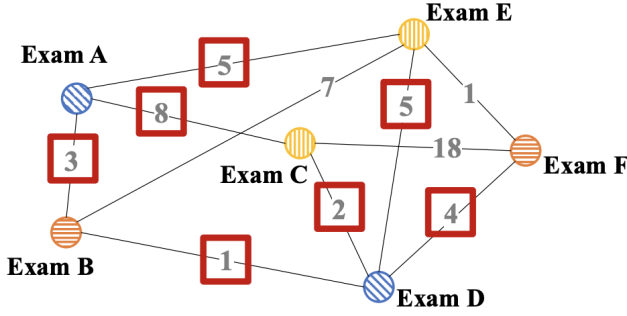


Figure 4. Exam Scheduling: example from (Ye et al. 2025)

that the variation across problem instances is in the lower and upper bounds on the number of locomotives per arc. Because real fleets evolve with maintenance cycles, failures, seasonality, and acquisitions or retirements, the available number of locomotives is inherently time-varying. To generate instances, the experiments vary these bounds around reference values using a uniform distribution, ensuring the resulting limits remain realistic relative to operational constraints. Because the underlying US-wide railway network and the core train schedules remain completely static across instances, specific rail corridors retain a persistent identity. This structural consistency means that specific physical paths exhibit highly predictable, historically grounded utilization patterns. In this model, the general integer variables indicate the number of locomotives on each active arc and the number of light travel trains on each light travel arc. Given the vastness of the network, only a small subset of arcs is actually utilized, meaning the vast majority of these general integer variables are exactly zero. Identifying these zero-valued variables reveals the underlying sparsity of the physical routing, effectively pruning massive, unproductive regions of the decision space.

Exam Scheduling

The third set of experiments considers a final-exam scheduling problem (EXAM) based on the block sequencing formulation introduced in (Ye et al. 2025). The EXAM determines an optimal assignment of university courses’ final exams to discrete time blocks. The problem specifies the required inputs, operational constraints such as block capacities, and penalty components governing scheduling conflicts between co-enrolled courses. Its main outputs are the exact placements of courses into time slots. The large problem setting includes 200 courses and 12 time slots, and the very large (designated as *-Hard*) problem setting includes 400 courses and 14 time slots. As the number of courses increases, the density of conflict constraints grows significantly, making the resulting MIPs substantially more challenging to solve.

We assume that what varies across problem instances is the co-enrollment structure and associated penalty coefficients. Because real student enrollments fluctuate from term to term, the conflict density between courses is inherently time-varying. To generate instances, the experiments vary

Problem	#Const.	#Bin.	#Int.
MMCNP	643	2,540	357
MMCNP-hard	4,361	23,984	1,135
SLAP	80,982	8	77,946
SLAP-hard	82,996	20	77,032
EXAM	30,173	43,488	173
EXAM-hard	53,205	79,968	229

Table 1. Overview of problem instances. Columns: #Const.: number of constraints; #Bin.: number of binary variables; #Int.: number of general integer variables.

these parameters based on an anonymized version of Cornell University’s Spring 2024 exam-scheduling input data using different random seeds, ensuring the resulting conflicts remain realistic relative to university operations. Despite these parameter variations, the physical dimensions of the problem (e.g. available time slots) remain perfectly fixed. This consistency ensures that individual courses maintain a distinct identity, exhibiting persistent historical scheduling characteristics regardless of term-to-term enrollment shifts. In this model, the discrete decisions are represented by binary variables indicating the assignment of each course to a specific time slot. Since each course must be placed into exactly one time slot out of all available options, all remaining assignment variables for that course are strictly zero. This structural sparsity highlights how isolating zero-valued binary variables can massively reduce the combinatorial search space.

Experiments

This section introduces the setup for empirical evaluation and presents the results. The code is available on <https://github.com/caidog1129/ID-PaS> and the instance files will be shared on the Distributional MIPLIB at <https://sites.google.com/usc.edu/distributional-miplib/>.

Experiment Setup

Baselines: We evaluate the performance of ID-PAS+ against the following three baselines:

1. **Gurobi (v12.0.3)** (Gurobi Optimization, LLC 2025): A state-of-the-art commercial MIP solver, run with default parameter settings and all built-in primal heuristics enabled. This serves as a strong industrial reference, reflecting standard solver performance without learning-based guidance.
2. **PAS** (Han, Yang et al. 2023): The original Predict-and-Search framework, which attempts to predict both zero and one assignments exclusively for binary variables.
3. **PAS+**: An extended version of PAS adapted to support general MIP domains by predicting solely zero-valued variables. Because this baseline lacks identity-aware positional embeddings, it serves as a crucial ablation to isolate the performance gains of our zero-prediction mechanism from the benefits of identity awareness.

Benchmark	Approach	PG (%) ↓			PI ↓		
		Mean	Std Dev	Wins	Mean	Std Dev	Wins
MMCNP	Gurobi	0.18	0.25	9	3.50	3.28	9
	PAS	0.16 (11.1%)	0.25	15	3.49 (0.29%)	2.71	7
	PAS+	0.14 (22.2%)	0.25	23	2.42 (30.9%)	2.53	19
	ID-PAS+	0.09 (50.0%)	0.18	53	2.13 (39.1%)	2.06	65
MMCNP-Hard	Gurobi	1.07	0.49	2	35.37	5.83	0
	PAS	0.58 (45.8%)	0.40	10	18.28 (48.3%)	4.82	9
	PAS+	0.40 (62.6%)	0.39	21	16.10 (54.5%)	4.00	15
	ID-PAS+	0.15 (86.0%)	0.25	67	11.53 (67.4%)	2.67	76
SLAP	Gurobi	0.049	0.054	15	1.36	0.59	13
	PAS+	0.025 (49.0%)	0.036	31	1.30 (4.4%)	0.50	20
	ID-PAS+	0.024 (51.0%)	0.032	54	1.16 (14.7%)	0.43	67
SLAP-Hard	Gurobi	0.18	0.21	18	7.85	2.12	13
	PAS+	0.16 (11.1%)	0.19	21	6.36 (19.0%)	1.90	14
	ID-PAS+	0.13 (27.8%)	0.15	61	5.36 (31.7%)	1.55	73
EXAM	Gurobi	0.023	0.010	2	1.97	1.78	21
	PAS	0.013 (43.5%)	0.008	18	1.90 (3.6%)	1.59	25
	PAS+	0.024 (-4.4%)	0.012	1	2.08 (-5.6%)	1.82	13
	ID-PAS+	0.004 (82.6%)	0.003	79	1.75 (11.2%)	1.71	41
EXAM-Hard	Gurobi	0.26	0.44	3	9.87	6.35	11
	PAS	0.13 (50.0%)	0.31	6	9.14 (7.4%)	5.03	9
	PAS+	0.10 (61.5%)	0.29	17	8.38 (15.1%)	5.05	15
	ID-PAS+	0.07 (73.1%)	0.23	74	7.61 (22.9%)	4.56	65

Table 2. Primal Gap (PG) and Primal Integral (PI) averaged over 100 test instances for each benchmark. We compare the performance of Gurobi, PAS, PAS+, and ID-PAS+. Results include the mean, standard deviation, and the number of instances each approach wins. We report the Primal Gap and Primal Integral improvement as a percentage inside the parentheses for the mean. The best-performing entries are highlighted in bold for clarity. We omit PAS for SLAP and SLAP-Hard benchmark because PAS is not applicable to instances with few binary variables.

For all ML-based approaches (PAS, PAS+, and ID-PAS+), a separate model is trained for each benchmark domain and difficulty setting. Each model is trained on 480 instances, validated on 120 instances, and evaluated on a disjoint set of 100 test instances. Summary statistics for the reference instances are reported in Table 1.

Evaluation Metrics: We evaluate solver performance on the test instances using two primary metrics:

1. **Primal Gap (PG)** (Berthold 2006): Defined as the normalized difference between the primal bound v and the best-known objective value v^* : $PG = \frac{|v-v^*|}{\max(|v^*|, \epsilon)}$ where $\epsilon = 10^{-8}$ is used to avoid division by zero. This metric is computed when a feasible solution v exists and $v \cdot v^* \geq 0$.
2. **Primal Integral (PI)** (Achterberg, Berthold, and Hendel 2012): The integral of the primal gap over the runtime interval $[0, t]$. The PI provides a comprehensive measure of both the quality of the solutions found and the speed at which they are discovered.

Hyperparameters: All experiments are conducted on 2.4 GHz Intel Xeon-2640v3 CPUs with 64 GB of memory.

Gurobi v12.0.3 is used as the underlying MIP solver for data collection and the evaluation of all approaches. All solver runtimes during evaluation are strictly capped at 1,000 seconds. Neural network training is accelerated using an NVIDIA V100 GPU with 112 GB of memory. Models are trained using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 1×10^{-5} and a batch size of 16 for a maximum of 48 hours. The model checkpoint achieving the lowest validation loss is selected for testing.

After training, we perform hyperparameter tuning for the search phase on a subset of 30 validation instances via a grid search. For the PAS baseline, the neighborhood parameters (k_0, k_1, Δ) are tuned, with $\Delta \in \{1\%, 3\%, 5\%\}$ and the target binary variable prediction parameters $k_0, k_1 \in \{0\%, 10\%, \dots, 50\%\}$. For both PAS+ and ID-PAS+, the parameters (k_0^b, k_0^i, Δ) are tuned on the same 30 validation instances. Specifically, we tune with $\Delta \in \{1\%, 3\%, 5\%\}$ and the target prediction percentages k_0^b and k_0^i across $\{0\%, 10\%, 30\%, \dots, 90\%\}$ of their respective variable counts. This adaptive tuning strategy ensures that the framework can robustly handle general MIPs, as well as problems containing exclusively binary or integer variables.

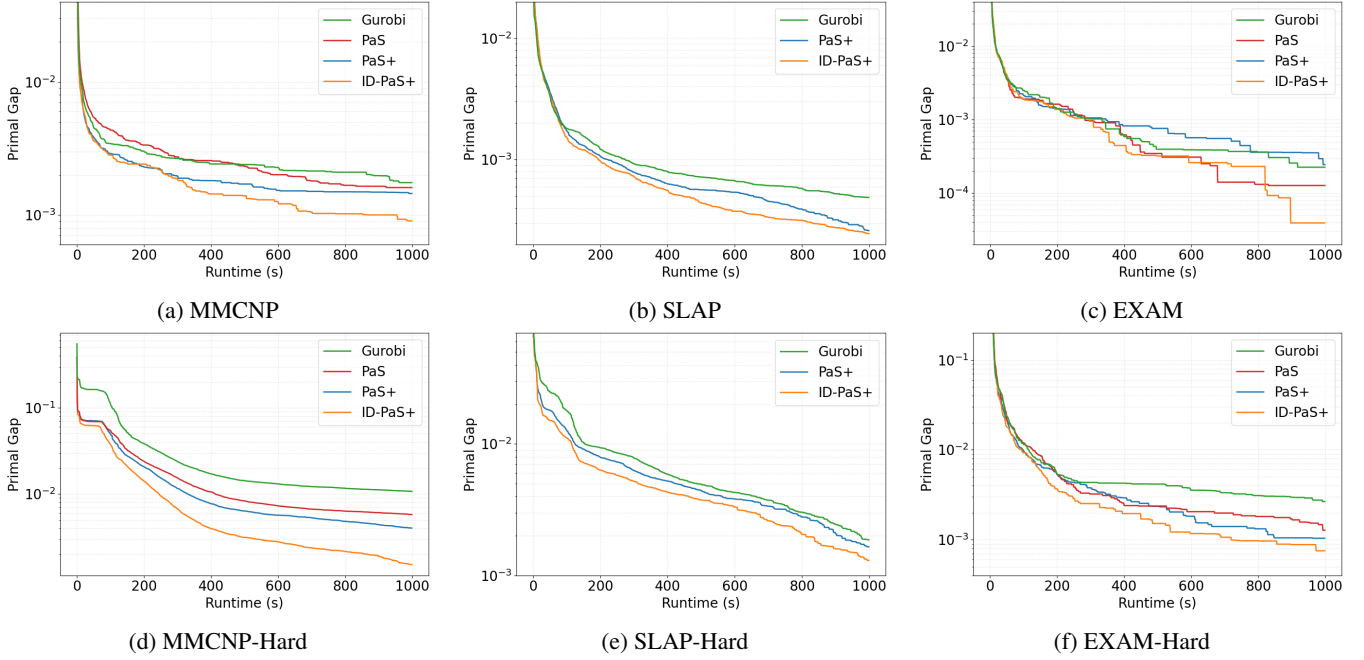


Figure 5. The Primal Gap (the lower, the better) over time, averaged over 100 test instances on each benchmark. We compare Gurobi (green line), PaS (red line), PaS+ (blue line), and ID-PaS+ (orange line). We omit PaS for SLAP and SLAP-Hard benchmark because PaS is not applicable to instances with few binary variables.

Results

Table 2 and Figure 5 demonstrate that ID-PaS+ achieves substantial, state-of-the-art improvements over default Gurobi, the original PaS framework, and the PaS+ ablation baseline across 100 previously unseen test instances for each benchmark. A Wilcoxon signed-rank test ($p < 0.01$) (Woolson 2007) confirms that all reported performance differences between ID-PaS+ and the baselines are statistically significant, validating the robustness of the observed improvements across varying difficulty levels.

The Impact of Zero-Prediction (PaS+ vs. PaS): Table 2 demonstrates the need to extend the Predict-and-Search (PaS) framework beyond binary classifications. While the original PaS improves upon Gurobi on MMCNP and EXAM, it fails on SLAP benchmarks due to its inability to handle general integers. By restricting the learning target strictly to zero-valued variables, PaS+ bridges this gap. Furthermore, PaS+ outperforms the original PaS on the MMCNP, MMCNP-Hard, and EXAM-Hard benchmarks. Ultimately, the improvements on MMCNP demonstrate that predicting general integer variables is crucial for performance. Furthermore, while there is a slight performance dip on the EXAM, the overall results indicate that focusing the solver’s trust region exclusively around highly confident zero-predictions does not broadly harm performance, and in more complex scenarios, significantly improves it.

The Necessity of Identity-Awareness (ID-PaS+ vs. PaS+): While PaS+ establishes a strong baseline, ID-

PaS+ consistently and significantly outperforms it across every single benchmark and difficulty setting. This performance gap isolates the immense value of identity-aware positional embeddings. By allowing the network to recognize specific physical infrastructure (e.g., persistent rail corridors or fixed highway arcs) rather than treating nodes as anonymous entities, ID-PaS+ provides far more accurate zero-predictions. Overall, ID-PaS+ achieves the lowest Primal Gap (PG) and Primal Integral (PI) across the board, reducing PG by 27.8% to 86.0% and PI by 11.2% to 67.4% relative to Gurobi, while winning the absolute majority of individual test instances.

Runtime Trajectory and Anytime Performance: Figure 5 illustrates the anytime performance improvements of our framework over the 1000-second evaluation horizon. On the easier benchmarks, while the advantages of ID-PaS+ are less pronounced during the initial 100 to 200 seconds, ID-PaS+ continues to aggressively reduce the primal gap at later runtimes, even after PaS and PaS+ have plateaued. On the harder benchmarks, the performance gap is even more pronounced: ID-PaS+ identifies significantly higher-quality solutions almost immediately upon initialization and consistently widens its lead over PaS, PaS+, and Gurobi throughout the entire time limit. These anytime performance plots clearly demonstrate that identity-aware zero-predictions provide highly reliable guidance and large improvement for large-scale, real-world MIPs.

Hyperparameters for search: Across all benchmarks, the best-performing hyperparameter configurations fix approximately 70% or 90% of the variables, utilizing a trust

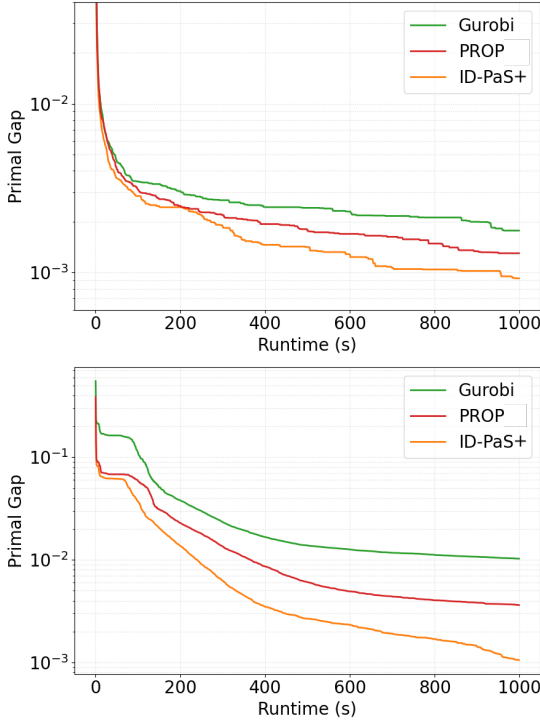


Figure 6. Primal gap (lower is better) over time, averaged over 100 test instances on MMCNP (top) and MMCNP-Hard (bottom). We compare Gurobi (green), PROP (red), and ID-PAS+ (orange).

region $\Delta = 1\%$ of $|X_0|$. We find that for benchmarks, where a single variable type heavily dominates (Binary for EXAM and Integer for SLAP, see Table 1), focusing exclusively on predicting that specific type yields the best performance during hyperparameter tuning. In the MMCNP domain, we surprisingly find that the best-performing configuration only predicts integer variables, although they are much fewer than the binary variables, indicating their outsized importance in this specific problem structure. Ultimately, the adaptive hyperparameter search across both variable types equips the framework to handle diverse problem scenarios robustly.

Comparison Against Static Heuristics (Zero-Frequency): To further validate the need for graph-learning methods, we compare ID-PAS+ against a baseline that fixes the same number of $|X_0|$ variables based solely on their empirical probability of being exactly zero in the training data. As shown in Table 3, ID-PAS+ dominates this baseline across all datasets. By leveraging contextual graph features rather than historical averages, ID-PAS+ wins the vast majority of the 100 test instances.

Comparison Against Independent Predictions (PROP): Finally, we implement the PROP framework from PROPEL (Akhlaghi, Zandehshahvar, and Van Hentenryck 2025), a strong learning baseline that is inherently identity-aware. PROP trains a dedicated neural network for each individual variable to predict its assignment independently. Because

Benchmark	ID-PAS+ Wins (PG)	ID-PAS+ Wins (PI)
MMCNP	83	71
SLAP	76	85
EXAM	82	76

Table 3. Comparison of ID-PAS+ against the Zero-Frequency Baseline. Values represent the number of instances, out of 100, where ID-PAS+ achieved a strictly better Primal Gap (PG) or Primal Integral (PI).

Benchmark	Approach	Wins (PG)	Wins (PI)
MMCNP	PROP	16	18
	ID-PAS+	59	62
MMCNP-Hard	PROP	21	7
	ID-PAS+	63	78

Table 4. Instance-wise win statistics (out of 100) comparing Gurobi, PROP, and ID-PAS+ on the MMCNP benchmarks. Remaining instances are won by Gurobi.

training tens of thousands of individual models is computationally demanding for the massive variable counts in SLAP and EXAM, we evaluate PROP exclusively on the MMCNP and MMCNP-Hard benchmarks.

Figure 6 and Table 4 demonstrate that while PROP improves upon default Gurobi by using a variable-specific approach, ID-PAS+ consistently secures the majority of wins on both Primal Gap and Primal Integral. Although PROP successfully captures distinct variable identities by dedicating a model to each variable, the performance gap indicates that identity awareness alone is insufficient. ID-PAS+, which leverages a shared graph-based representation augmented with identity-aware embeddings, is necessary to achieve the strongest overall performance.

Conclusion

This work introduces ID-PAS+ to extend the Predict-and-Search framework to general Mixed-Integer Programs (MIPs). ID-PAS+ leverages a zero-prediction mechanism to exploit structural sparsity and incorporates identity-aware embeddings to capture persistent physical topologies across varying instances. Extensive experiments on large-scale, real-world benchmarks demonstrate that ID-PAS+ consistently outperforms default Gurobi, the original binary-only PAS, and independent per-variable baselines like PROP. By effectively modeling global structural dependencies, ID-PAS+ achieves up to an 86.0% reduction in Primal Gap and a 67.4% reduction in Primal Integral relative to the state-of-the-art solver, offering a highly reliable and general approach for accelerating complex optimization tasks.

Acknowledgments

The National Science Foundation (NSF) partially supported the research under grant #2112533: AI4OPT and grant #2346058: NRT-AI.

References

- Achterberg, T.; Berthold, T.; and Hendel, G. 2012. Rounding and propagation heuristics for mixed integer programming. In *OR*, 71–76. Springer.
- Akhlaghi, V. E.; Zandehshahvar, R.; and Van Hentenryck, P. 2025. Propel: Supervised and reinforcement learning for large-scale supply chain planning. *arXiv preprint arXiv:2504.07383*.
- Berthold, T. 2006. *Primal heuristics for mixed integer programs*. Ph.D. thesis, Zuse Institute Berlin (ZIB).
- Bertsimas, D.; and Stellato, B. 2022. Online mixed-integer optimization in milliseconds. *INFORMS Journal on Computing*, 34(4): 2229–2248.
- Brody, S.; Alon, U.; and Yahav, E. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.
- Cai, J.; Huang, T.; and Dilkina, B. 2024. Learning Backdoors for Mixed Integer Linear Programs with Contrastive Learning. In *Proceedings of the ECAI 2024*. IOS Press.
- Cai, J.; Huang, T.; and Dilkina, B. 2025. Multi-task Representation Learning for Mixed Integer Linear Programming. In *CPAIOR*, 134–151. Springer.
- Cai, J.; Huang, W.; et al. 2025. Neuro-Symbolic Acceleration of MILP Motion Planning with Temporal Logic and Chance Constraints. *arXiv preprint arXiv:2508.07515*.
- Cai, J.; Kadioğlu, S.; and Dilkina, B. 2025. Balans: Multi-Armed Bandits-based Adaptive Large Neighborhood Search for Mixed-Integer Programming Problems. In Kwok, J., ed., *IJCAI*, 2566–2574.
- Chen, W.; Tanneau, M.; and Van Hentenryck, P. 2023. End-to-end feasible optimization proxies for large-scale economic dispatch. *IEEE Transactions on Power Systems*, 39(2): 4723–4734.
- Er Raqabi, E. M.; Bani, A.; et al. 2025. A Decomposition Matheuristic for the Transient Stability Constrained Unit Commitment At Hydro-Quebec. *IEEE Transactions on Power Systems*.
- Er Raqabi, E. M.; Beljadid, A.; et al. 2025. OCP optimizes its supply chain for Africa. *INFORMS Journal on Applied Analytics*, 55(6): 437–456.
- Er Raqabi, E. M.; Himmich, I.; et al. 2023. Incremental LNS Framework for Integrated Production, Inventory, and Vessel Scheduling: Application to a Global Supply Chain. *Omega*, 116: 102821.
- Er Raqabi, E. M.; Wu, Y.; et al. 2024. Towards resilience: Primal large-scale re-optimization. *Transportation Research Part E: Logistics and Transportation Review*, 192: 103819.
- Gal, T. 2009. Linear parametric programming—a brief survey. *Sensitivity, stability and parametric analysis*, 43–68.
- Gasse, M.; Chételat, D.; Ferroni, N.; Charlin, L.; and Lodi, A. 2019. Exact combinatorial optimization with graph convolutional neural networks. *Advances in neural information processing systems*, 32.
- Greening, L. M.; Dahan, M.; and Erera, A. L. 2023. Lead-time-constrained middle-mile consolidation network design with fixed origins and destinations. *Transportation Research Part B: Methodological*, 174: 102782.
- Gurobi Optimization, LLC. 2025. Gurobi Optimizer Reference Manual.
- Han, Q.; Yang, L.; et al. 2023. A gnn-guided predict-and-search framework for mixed-integer linear programming. *arXiv preprint arXiv:2302.05636*.
- Huang, T.; Ferber, A. M.; Tian, Y.; Dilkina, B.; and Steiner, B. 2023. Searching large neighborhoods for integer linear programs with contrastive learning. In *ICML*, 13869–13890. PMLR.
- Huang, T.; Ferber, A. M.; Zharmagambetov, A.; Tian, Y.; and Dilkina, B. 2024a. Contrastive Predict-and-Search for Mixed Integer Linear Programs. In *ICML*. PMLR.
- Huang, W.; Huang, T.; Ferber, A. M.; and Dilkina, B. 2024b. Distributional MIPLIB: a Multi-Domain Library for Advancing ML-Guided MILP Methods. *arXiv preprint arXiv:2406.06954*.
- Kim, Y.; Hijazi, A.; Dalmeijer, K.; and Van Hentenryck, P. 2025. Practice-Based Optimization for the Strategic Locomotive Assignment Problem. *arXiv preprint arXiv:2507.22235*.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Mao, A.; Mohri, M.; and Zhong, Y. 2023. Cross-entropy loss functions: Theoretical analysis and applications. In *ICML*, 23803–23828. pmlr.
- Maps of World. 2025. USA Rail Map. <https://www.mapsofworld.com/usa/usa-rail-map.html>.
- Mohan, V.; Er Raqabi, E. M.; and Van Hentenryck, P. 2025. A Fair OR-ML Framework for Resource Substitution in Large-Scale Networks. *arXiv:2511.18269*.
- Pangia, A. C.; and Wiecek, M. M. 2025. A branch-and-bound algorithm for parametric mixed-binary nonlinear programs. *Journal of Global Optimization*, 91(3): 457–481.
- Ranjan, V.; and Stellato, B. 2025. Verification of first-order methods for parametric quadratic optimization: V. Ranjan, B. Stellato. *Mathematical Programming*, 1–57.
- Russo, L.; Nair, S. H.; Glielmo, L.; and Borrelli, F. 2023. Learning for online mixed-integer model predictive control with parametric optimality certificates. *IEEE Control Systems Letters*, 7: 2215–2220.
- Song, J.; Lanka, R.; Yue, Y.; and Dilkina, B. 2020. A general large neighborhood search framework for solving integer programs. In *NeurIPS*.
- Tang, B.; and Khalil, E. B. 2024. PyEPO: a PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 16(3): 297–335.
- Tanneau, M.; and Van Hentenryck, P. 2024. Dual lagrangian learning for conic optimization. *Advances in Neural Information Processing Systems*, 37: 55538–55561.
- Woolson, R. F. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, 1–3.
- Ye, T.; Jovine, A. S.; Van Osselaer, W.; Zhu, Q.; and Shmoys, D. B. 2025. Cornell university uses integer programming to optimize final exam scheduling. *INFORMS Journal on Applied Analytics*.