Constraint Learning and Schedule Proposal via Transformers for the Nurse Scheduling Problem

Walid Abdelaidoum[1,2], Mohamed A. Madani[2], Larbi Boubchir[1], and Boubaker Daachi[1]

[1] LIASD Laboratory, University of Paris 8, Paris, France
[2] SWAPPY, Paris, France

Abstract. We train a transformer on feasible schedules of a simplified Nurse Scheduling Problem variant generated by a constraint programming solver, and use the trained model as a learnability probe with no constraint fed at inference. We evaluate three constraint configurations spanning 3 to 40 nurses and report per-family satisfaction rates across four constraint families. The results reveal a learnability hierarchy by structural scope: local per-nurse rules are captured reliably, bounded-window rules degrade with sequence length, and global cross-nurse rules remain the hardest to satisfy. Median violation counts stay low (0 to 5.5 per schedule over the full horizon), suggesting that the generated schedules are suitable as warm starts for repair or local-search procedures.

Keywords: Nurse Scheduling Problem · Transformers · Constraint Programming · Learnability

1 Introduction

The Nurse Scheduling Problem (NSP) assigns shifts to nurses over a planning horizon subject to hard and soft constraints. We study a simplified variant with hard constraints only and four shift types Night (N), Day (D), Evening (E), and Rest (R), over a horizon of d days for a team of $|\mathcal{I}|$ nurses. Many practically relevant NSP variants are NP-hard, although some sub-classes are polynomially solvable [3].

Recent NSP work falls in three lines. Exact and decomposition approaches model the problem as mixed-integer programs solved monolithically or by branch-and-price [2, 8]. Metaheuristic and matheuristic methods address larger instances, e.g., simulated annealing on real rosters [4] and hybrid meta-heuristics on extended formulations with reserve nurses [9]. Neural networks have been introduced inside NSP solvers to guide or assist an existing search procedure, through recurrent-network-assisted heuristics [6], deep-network-assisted heuristic selection [5], multi-agent deep Q-network-based neighbourhood selection [11], or federated LSTM-based fatigue prediction [1]. Beyond the NSP, decoder-only transformers have been applied to combinatorial problems such as Sudoku, where pairing a small GPT-2 with depth-first search and a verifier reaches near-perfect accuracy [7].

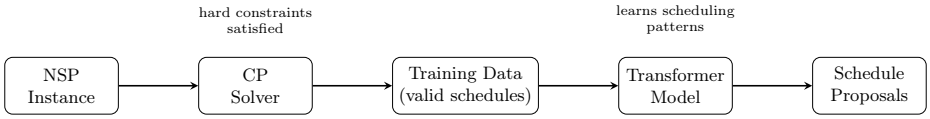

Fig. 1. Pipeline: a CP solver generates feasible schedules; a transformer is trained on them and proposes new ones.

The present work uses a transformer as a learnability probe: no constraint is fed to the model; we measure which constraint families the transformer captures from feasible examples alone. Figure 1 illustrates the proposed pipeline.

2 Problem Formulation and Transformer Architecture

Let $\mathcal{I} = \{0, \dots, n-1\}$ denote the set of nurses, $\mathcal{J} = \{0, \dots, d-1\}$ the set of days, and $\mathcal{S} = \{N, D, E, R\}$ the four shift types (Night, Day, Evening, Rest); the decision variable $x_{i,j} \in \mathcal{S}$ gives the shift of nurse i on day j. We further use $\mathcal{F} \subseteq \mathcal{S} \times \mathcal{S}$ for forbidden consecutive pairs, $\mathcal{T} \subseteq \mathcal{S}$ for shifts subject to consecutive-day limits, $\mathcal{M} \subseteq \mathcal{S}$ (with $\mathcal{M} \not\subseteq \mathcal{T}$) for the mandatory shifts following a maximal run, $k_{\max} \in \mathbb{Z}^+$ for the maximum consecutive shifts from $\mathcal{T}$, $p \in \mathbb{Z}^+$ for the rolling-window period, and triples (s, l, u) for coverage requirements (shift s with bounds $[l, u]$); the sliding-window start day is $j_0 \in \{0, \dots, d-1-k_{\max}\}$ and the indicator $\mathbf{1}[x_{i,j} \in \mathcal{T}]$ marks whether nurse i works a shift in $\mathcal{T}$ on day j. The CP model reads

$$\min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{s \in \mathcal{S}} w_{i,j,s}\, \mathbf{1}[x_{i,j} = s], \tag{1}$$

subject to

$$(x_{i,j},\, x_{i,j+1}) \notin \mathcal{F},\ \forall\, i,\ j \in \{0, \dots, d-2\}, \tag{2}$$

$$\sum_{\delta=0}^{k_{\max}} \mathbf{1}[x_{i,\, j_0+\delta} \in \mathcal{T}] \le k_{\max},\ \forall\, i, j_0, \tag{3}$$

$$\bigwedge_{\delta=0}^{k_{\max}-1} \left(x_{i,\, j_0+\delta} \in \mathcal{T}\right) \ \Rightarrow\ x_{i,\, j_0+k_{\max}} \in \mathcal{M}, \tag{4}$$

$$l \le \sum_{i \in \mathcal{I}} \mathbf{1}[x_{i,j} = s] \le u, \tag{5}$$

$$l_W \le \sum_{j \in W} \mathbf{1}[x_{i,j} \in \mathcal{A}] \le u_W,\ \forall\, i, W. \tag{6}$$

The objective (1) carries no preference: weights $w_{i,j,s} \sim \mathcal{U}[0,1]$ are redrawn at each solver call so that independent runs return distinct feasible schedules used as training data. A schedule is feasible if it satisfies the five constraint families (consecutive limits and the mandatory-shift rule are grouped into one evaluation family in Section 3): forbidden transitions on adjacent days, Eq. (2); maximum consecutive shifts from $\mathcal{T}$, Eq. (3); mandatory shift in $\mathcal{M}$ after a maximal run, Eq. (4); daily coverage bounds (applied uniformly to all days, weekends included), Eq. (5); shift-limit bounds over rolling windows of p days, Eq. (6). Table 1 shows the active constraints per data configuration.

Table 1. Active constraints per configuration. Dash: inactive. Parameter values $k_{\max}$, coverage bounds, and $\mathcal{A}_k$ subsets in the text.

	Forb. (2)	Consecutive (3)			Cov. (5)	Per 7 d (6)
	N → D	nts → R	wrk → R	rst → wrk (4)		
S_1	✓	3	5	—	✓	✓
S_2	✓	3	4	2	✓	—
S_3	✓	3	—	—	✓	✓

Transformer architecture: Each schedule is serialised row-by-row into a token sequence over a vocabulary of size $|\mathcal{V}|$=136: the four shift tokens from $\mathcal{S}$, four special tokens BOS/PAD/SEP/EOS, and condition tokens encoding $|\mathcal{I}|$ and $|\mathcal{J}|$. Three learned positional embeddings (sequence, nurse, day) are summed to recover the 2D coordinate that flattening loses; about 65% of model parameters live in these tables. The model is a decoder-only transformer [10] (4 layers × 4 heads, d_{model}=128, ∼837K parameters), Pre-LayerNorm, masked self-attention, GELU (Gaussian Error Linear Unit) feed-forward, output projection weight-tied to the token embedding. Training minimises masked cross-entropy with AdamW, learning rate 3×10^{-4}, linear warmup then cosine decay, gradient clipping at norm 1, and early stopping after 5 epochs without validation improvement. At inference, the trained transformer generates a complete assignment $x_{i,j}$ autoregressively by greedy left-to-right decoding; no CP model is used.

3 Experimental Evaluation

Each trained model generates 150 schedules on held-out instance dimensions; the constraint checker tests each of the four families independently (consecutive limits and the mandatory-shift rule count as one family). The main metric is the satisfaction rate, the fraction of schedules that fully respect a given constraint.

All three models converge with training-loss plateaus of 0.78 (S_1), 0.69 (S_2), and 0.94 (S_3), reflecting the entropy of the feasible-solution space.

Constraint satisfaction rates are: forbidden transitions 80.8–99.3%, consecutive limits 30.7–100%, coverage 14.0–86.7%, and period limits 7.3–62.0%, exhibiting a clear hierarchy local > bounded-window > global cross-nurse.

Median total violations per schedule are 0 (S_2), 4 (S_3), and 5.5 (S_1). Even when satisfaction is low, the generated schedules carry few violations and could plausibly seed a repair heuristic or local-search phase rather than being discarded.

A finer breakdown shows where these violations concentrate: on S_1, in night/day coverage (medians ∼0.8–0.9), consecutive work limits (median 1), and period limits (nights/rests per 7 d, medians 2–3); on S_3, in day-coverage shortfalls (median 2). The contrast between S_2 (86.7%) and S_3 (33.3%) on coverage, at identical team sizes, suggests that the added period limits in S_3 compete with coverage targets rather than reinforce them. These failure modes match the warm-start scenario: residual errors lie in a few cross-nurse cells that a repair or local-search step can resolve cheaply.

4 Conclusion

The results reveal a learnability hierarchy by structural scope: local per-nurse rules are captured reliably, bounded-window rules degrade with sequence length, and global

cross-nurse rules remain the hardest. Low median violation counts (0 to 5.5 per schedule over the full horizon) suggest that the model can be used as a warm-start generator for downstream repair or local-search procedures. The study is limited to hard constraints on synthetic instances; per-nurse contracts and soft preferences are left for future work. Logit masking at inference could enforce hard constraints directly.

References

[1] Aibin, M., Chen, F., Menzi, M., Scholte, L., Jiang, Z., Chen, Y., Jin, X., Liu, C., Aibin, A.: Federated learning framework for nurse scheduling to lower fatigue levels of nurses. In: 2025 International Conference on Computing, Networking and Communications (ICNC). pp. 340–344. IEEE (2025)

[2] Böðvarsdóttir, E.B., Bagger, N.C.F., Høffner, L.E., Stidsen, T.J.: A flexible mixed integer programming-based system for real-world nurse rostering. Journal of Scheduling **25**(1), 59–88 (2022)

[3] Burke, E.K., De Causmaecker, P., Berghe, G.V., Van Landeghem, H.: The state of the art of nurse rostering. Journal of scheduling **7**(6), 441–499 (2004)

[4] Ceschia, S., Di Gaspero, L., Mazzaracchio, V., Policante, G., Schaerf, A.: Solving a real-world nurse rostering problem by simulated annealing. Operations Research for Health Care **36**, 100379 (2023)

[5] Chen, Z., De Causmaecker, P., Dou, Y.: A combined mixed integer programming and deep neural network-assisted heuristics algorithm for the nurse rostering problem. Applied Soft Computing **136**, 109919 (2023)

[6] Chen, Z., Dou, Y., De Causmaecker, P.: Neural networked-assisted method for the nurse rostering problem. Computers & Industrial Engineering **171**, 108430 (2022)

[7] Giannoulis, P., Pantis, Y., Tzamos, C.: Teaching transformers to solve combinatorial problems through efficient trial & error. Advances in Neural Information Processing Systems **38**, 133548–133580 (2026)

[8] Legrain, A., Omer, J.: A dedicated pricing algorithm to solve a large family of nurse scheduling problems with branch-and-price. INFORMS Journal on Computing **36**(4), 1108–1128 (2024)

[9] Saemi, S.: Nurse scheduling problem by considering reserve nurses: A mathematical modeling and hybrid meta-heuristic algorithm. Operational Research **25**(4), 103 (2025)

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[11] Zhang, X., Yang, Y., Zhu, Q., Lin, Q., Chen, W., Li, J., Coello, C.A.C.: Multiagent deep q-network-based metaheuristic algorithm for nurse rostering problem. Swarm and Evolutionary Computation **87**, 101547 (2024)