Large Language Model-Based Explainable Hyper-heuristics for Examination Timetabling

Nilgün Şengöz[1,2⋆], Ender Özcan[2], and Jeremie Clos[2]

[1] Burdur Mehmet Akif Ersoy University, Burdur, Türkiye
nilgunsengoz@mehmetakif.edu.tr
[2] University of Nottingham, Nottingham, UK
{ender.ozcan,jeremie.clos}@nottingham.ac.uk

Abstract. We introduce LLM-STAR (Structure-Aware Selector), a generation hyper-heuristic that uses GPT-4 as an offline knowledge source to produce a transparent, deterministic rule engine for algorithm selection in examination timetabling. The approach is motivated by the observation that no single graph colouring heuristic dominates across the structural diversity of realistic timetabling conflict graphs, and that human-readable selection rules are preferable to opaque learned models in scheduling practice. Using GPT-4 in an offline distillation phase, processing selected academic papers on examination timetabling and graph colouring, we convert chain-of-thought reasoning into a seven-rule engine that maps five structural instance features to candidate heuristic sets. Experiments on the Toronto benchmark instances demonstrate a perfect success rate (13/13). Independent validation on 18 randomly generated instances confirms out-of-sample generalisation (18/18), demonstrating the viability of offline LLM distillation for explainable combinatorial optimisation.

Keywords: Timetabling · Graph colouring · Hyper-heuristics · Large language models · Algorithm selection

1 Introduction

The algorithm selection problem [9] remains a central challenge in combinatorial optimisation, requiring prediction of the best-performing algorithm for a given instance based on measurable features. Examination timetabling is a widely studied NP-hard problem [8], reducible to the Minimum Graph Colouring Problem [6] in its simplest form, where each vertex represents an examination and edges indicate conflicts arising from shared student enrolments. The objective of assigning the minimum number of time-slots to a set of mutually non-clashing exams is therefore equivalent to determining the chromatic number of the corresponding graph. Well-known constructive heuristics such as Welsh-Powell, DSATUR, and

⋆ Corresponding author.

Smallest-Last build feasible but near-optimal solutions efficiently [2]; however, no single heuristic consistently dominates across diverse graph structures.

A key aspect of LLM-STAR is that it targets the *structural subspace* of graph colouring that corresponds to realistic examination timetabling instances, rather than the full space of all graphs. Examination conflict graphs are not arbitrary: student enrolment patterns are shaped by curriculum structure and departmental boundaries, producing local clustering that is absent in random graphs of comparable density. This yields conflict graphs with elevated clustering coefficients (κ) and degree variance profiles (σ^2_δ) reflecting programme-level modular structure, while edge density ρ is bounded by realistic enrolment distributions. The five structural features $(\rho, \kappa, \sigma^2_\delta, \delta_{\max}, n)$ capture the axes along which realistic timetabling instances vary [3] and along which heuristic performance differences are most pronounced. This work addresses algorithm selection within this timetabling-relevant subspace and does not claim generality across all graph colouring instances.

Recent studies explore large language models (LLMs) as online optimisation engines [12] and evolutionary code generators [14]. In contrast, this study investigates LLMs as an *offline* knowledge source for generation hyper-heuristics [5]. We introduce LLM-STAR, which uses GPT-4 to distil academic literature into a stand-alone, explainable rule engine; once generated, the engine operates entirely without the LLM as a deterministic, auditable decision procedure.

Relation to Instance Space Analysis. Instance Space Analysis (ISA) [11] provides a systematic methodology for understanding algorithm performance across a feature-defined instance space, and has been applied to graph colouring [10] and curriculum-based course timetabling [4]. LLM-STAR shares the goal of mapping instance structure to algorithm performance but differs from ISA in three respects: (i) features are derived from LLM-distilled domain expertise in the literature rather than by automated statistical selection; (ii) decision boundaries are produced by chain-of-thought reasoning and maximum-margin thresholding rather than geometric footprint analysis; and (iii) the output is a set of natural-language-annotated deterministic rules deployable without visualisation infrastructure, rather than a footprint map. The two approaches are complementary: ISA methodology could validate LLM-STAR's feature space and thresholds, which we identify as future work.

InstSpecHH [15] generates instance-specific heuristics via LLM meta-feature interpretation. HeurAgenix [13] introduces a two-phase framework with offline heuristic evolution followed by online LLM-guided selection; unlike LLM-STAR, it still invokes the LLM at runtime, incurring latency and non-determinism. Atias [1] applied LLMs to multi-objective timetabling, confirming their promise in scheduling domains.

The remainder of this paper is structured as follows: Section 2 details the LLM-STAR architecture and the LLM knowledge distillation protocol. Section 3 presents the experimental results, statistical analysis, and independent validation. Section 4 concludes the study.

2 Rule Engine Generation for Heuristic Selection

GPT-4 (gpt-4-0613 API) is initially provided with selected academic papers on examination timetabling and graph colouring to identify important features for algorithm selection. This offline training phase is followed by a four-stage process building a rule-based engine.

Stage 1. Instances are characterised using a five-dimensional structural feature vector $\mathbf{f} = (\rho, \kappa, \sigma_\delta^2, \delta_{\max}, n)$:

- **Edge density** ρ: the proportion of realised edges relative to the maximum possible:
$$\rho = \frac{|E|}{\binom{n}{2}} = \frac{2|E|}{n(n-1)} \tag{1}$$
- **Global clustering coefficient** κ (transitivity [7]): the ratio of closed triplets (triangles) to *all* connected triplets, measuring the tendency of conflict neighbourhoods to form cliques:
$$\kappa = \frac{3 \times |\triangle(G)|}{\sum_{v \in V} \binom{\delta_v}{2}} \tag{2}$$
where $|\triangle(G)|$ is the number of triangles, δ_v is the degree of vertex v, and $\binom{\delta_v}{2}$ counts the connected triples centred at v. This is the *global* (transitivity) form of [7], distinct from the average local clustering coefficient.
- **Degree variance** σ_δ^2 (population variance):
$$\sigma_\delta^2 = \frac{1}{n} \sum_{v \in V} (\delta_v - \bar{\delta})^2, \quad \bar{\delta} = \frac{2|E|}{n} \tag{3}$$
- **Maximum degree** $\delta_{\max} = \max_{v \in V} \delta_v$.
- **Instance size** $n = |V|$: the number of examinations.

Figure 1 illustrates these features on a small example conflict graph.

These features capture computational difficulty [3] in graph colouring and timetabling. Features are extracted from the Toronto benchmark instances [8]; these instances are used throughout all stages of our workflow and are treated as the effective training set.

Stage 2. GPT-4 is prompted five times per instance, using structural features and observed algorithm performance from academic studies on the Toronto training instances, to produce chain-of-thought responses identifying promising heuristic subsets from a pool of five algorithms: $\{\text{WP}, \text{DSA}, \text{SL}, \text{BFS}, \text{DFS}\}$. Prompts follow the structure: *"Given feature vector* $\mathbf{f}$ *and performance data showing [heuristic rankings], which 1–2 heuristics are most likely to succeed? Explain your structural reasoning."* All 65 raw chain-of-thought responses will be made publicly available, enabling independent verification of per-instance

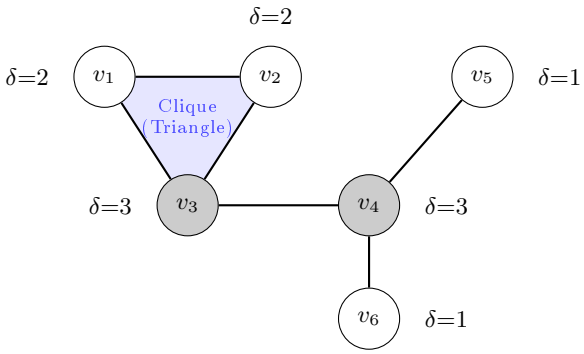

Fig. 1. Five structural features illustrated on a toy conflict graph with n=6 examinations and $|E|$=6 conflict edges. The shaded triangle (v_1, v_2, v_3) is the graph's only triangle and the sole contributor to the clustering coefficient, while the grey vertices v_3 and v_4 attain the maximum degree. For this graph the features evaluate to an edge density ρ=$2|E|/[n(n-1)]$=0.40 (Eq. 1), a global clustering coefficient κ=3/8=0.375 (Eq. 2, transitivity [7]), a degree variance σ^2_δ=0.67 about a mean degree $\bar{\delta}$=2.0 (Eq. 3), and a maximum degree $\delta_{\max}$=3. We emphasise that κ denotes the *global* (transitivity) coefficient of [7]; the average local clustering coefficient yields a distinct value (0.39) that is not employed by the rule engine.

consensus levels and LLM consistency across repeated queries. All source papers processed in this offline phase were accessed lawfully through institutional subscriptions or open-access venues and were used solely for non-consumptive text-and-data-mining for scientific research; no verbatim text from these sources is reproduced in the rule engine or in this paper.

Stage 3. A recommendation is accepted if at least four of five responses agree. The LLM provided qualitative rationales (e.g., *"sparse and highly clustered, favouring degeneracy ordering"*) based on continuous feature values. To convert instance-specific rationales into generalisable rules, we project training instances onto the five-dimensional feature space, identifying three structural classes through manual inspection: heavy-tailed degree distributions, sparse clustered, and medium-density unstructured.

Stage 4. We formalise quantitative decision boundaries through comparative threshold analysis between instances where the LLM preferred different heuristic classes. The quantitative boundaries (τ) dividing the five-dimensional feature space $\mathcal{F}$ into the seven operational rules (Table 1) are defined as follows.

Density boundaries (τ_ρ). The transition from sparse to medium-density to dense graphs is partitioned at τ_{ρ_1}=0.12 and τ_{ρ_2}=0.35, calculated via the maximum-margin separation:

$$\tau_\rho = \frac{\max_{x \in C_{\text{sparse}}} \rho(x) + \min_{x \in C_{\text{dense}}} \rho(x)}{2} \tag{4}$$

where C_{sparse} and C_{dense} are instance sets identified by Stage 3 consensus.

Table 1. LLM-generated Rule Engine: seven structure-conditioned selection rules (priority order). DSA=DSATUR; SL=Smallest-Last; WP=Welsh-Powell; BFS=BFS-Sequential.

#	Structural Condition	Candidate Set	LLM-Distilled Rationale
1	$n<300$ & $\rho>0.35$	{DSA, SL}	Dense small graph; saturation and degeneracy competitive.
2	$n\geq400$ & $\sigma_\delta^2>80$ & $\delta_{\max}>3\bar{\delta}$	{DSA, WP}	Heavy-tailed topology; degree-based methods effective.
3	$\rho\leq0.12$ & $\kappa>0.40$	{SL, DSA}	Sparse clustered; degeneracy exploits dense subgraphs.
4	$0.12<\rho\leq0.35$ & $\kappa>0.30$	{DSA, SL}	Medium-density clustered; saturation dominant.
5	$n\geq300$ & $\rho\leq0.10$ & $\sigma_\delta^2<20$	{SL, BFS}	Large sparse uniform; traversal beats saturation.
6	$100\leq n<400$ & $0.10<\rho\leq0.20$	{BFS, DSA}	Medium-scale unstructured; BFS captures neighbourhood structure.
7	(default)	{DSA}	Default fallback for ambiguous signatures.

Clustering boundaries (τ_κ). To distinguish random edge distributions from clustered enrolment patterns, the clustering margin is empirically set at $\tau_\kappa \in \{0.30, 0.40\}$ using the same maximum-margin equation.

Heavy-tailed detection. An instance triggers Rule 2 if:

$$\delta_{\max} > 3\bar{\delta} \quad \wedge \quad \sigma_\delta^2 > 80 \tag{5}$$

The threshold $\sigma_\delta^2>80$, together with the $\delta_{\max}>3\bar{\delta}$ condition, isolates the single extreme heavy-tailed instance (pur-s-93, σ_δ^2=8527.5); other high-variance instances (e.g. car-s-91, σ_δ^2=3843.7) do not satisfy the joint condition. We explicitly acknowledge that this single-instance calibration presents an overfitting risk; validation on additional heavy-tailed topologies is essential to confirm generalisation.

Scale boundaries (τ_n). Instance size thresholds ($\tau_n \in \{300, 400\}$) delineate scale regimes where the asymptotic behaviour of specific heuristics becomes relevant.

This formalisation ensures that the transition from five continuous features to seven discrete rules is both structurally grounded in LLM reasoning and quantitatively calibrated to the benchmark distribution, replacing ad-hoc parameter tuning with benchmark-calibrated thresholds.

The rule engine operates in two steps: (i) evaluate the feature vector against rules in priority order; the first match returns the candidate set. (ii) execute only the candidate heuristics and select the minimum-timeslot solution.

3 Experimental Results

Table 2 summarises LLM-STAR performance on 13 Toronto benchmark instances [8] (training set) and 18 randomly generated instances (validation set). The validation set was produced using Qu et al.'s generator [8], with a stratified design of 9 small (n<100) and 9 large (n≥500) instances, with densities spanning 6%–48% to broadly mirror the Toronto distribution. This split ensures that rules gated on scale boundaries (Rules 1, 2, 5, 6) are tested in both size regimes. We acknowledge that the density range does not cover extreme sparsity (ρ<5%) or very high density (ρ>60%); these represent directions for future stress-testing.

LLM-STAR achieves a success rate—defined as the proportion of instances matching empirical pool-best performance (best among the five constructive heuristics)—of 100% on Toronto (13/13). Statistical analysis using the Wilcoxon signed-rank test (two-tailed) confirms that LLM-STAR selections significantly outperform a static Smallest-Last baseline: W=0.00, p=0.002, Cohen's d=−1.05 [95% CI: $-1.70, -0.71$], indicating a large effect size. Smallest-Last serves as baseline following prior comparative work [2]; note that Rule 7 designates DSATUR as the default fallback, and comparison against static DSATUR is reserved for the extended version where per-heuristic timeslot data will be reported in full.

To address overfitting concerns, LLM-STAR was validated on 18 randomly generated out-of-sample instances [8]: 9 small (n<100), 9 large (n≥500), densities spanning 6%–48%. LLM-STAR achieved full success across all instances, matching pool-best on every validation instance. Four rules activated {R1, R4, R6, R7}, demonstrating generalisation across unseen instances.

Rule 3 (sparse clustered: $\rho \leq 0.12 \wedge \kappa > 0.40$) successfully identifies structural anomalies like `lse-f-91`, where Smallest-Last outperforms DSATUR (18 vs. 19 timeslots). Rule 4 (medium-density clustered) is the most frequently triggered rule (7/13 Toronto, 6/18 generated), reflecting the prevalence of this structural class. Rule 5 (large sparse uniform) never activates on either benchmark, indicating that neither Toronto nor generated instances sufficiently represent this structural regime; this defines a validation target for future work. The perfect success rate across both training and validation sets confirms that LLM-distilled structural conditions generalise beyond the training distribution—consistent with the generalisation patterns observed in ISA studies on adjacent domains [4,10].

4 Conclusion

This study demonstrates that the reasoning capabilities of large language models can be systematically distilled into deterministic and interpretable rules for guiding heuristic selection. Using examination timetabling as a case study, we showed that the proposed LLM-STAR framework reliably identifies the best-performing graph colouring heuristic within the structural subspace corresponding to realistic timetabling conflict graphs, achieving full success on both the Toronto benchmark and randomly generated validation instances, while providing transparent and auditable explanations for each decision.

Table 2. Performance summary of LLM-STAR. **R** = rule triggered; **PB** = pool-best timeslots (best among five heuristics). **Bold** entries: LLM-STAR matches Pool-Best. Conflict densities ρ are computed via Eq. (1) (Carter et al.); rules are evaluated in priority order (Table 1).

Toronto Benchmark (13/13)				**Generated Instances (18/18)**			
Inst.	ρ	**R**	**PB**	**Inst.**	ρ	**R**	**PB**
car-f-92	0.138	R4	**30**	S-p6	0.062	R7	**8**
car-s-91	0.128	R4	**35**	S-p12	0.125	R4	**10**
ear-f-83	0.267	R4	**24**	S-p18	0.184	R6	**12**
hec-s-92	0.421	R1	**19**	S-p25	0.267	R4	**12**
kfu-s-93	0.056	R3	**19**	S-p32	0.335	R4	**15**
lse-f-91	0.063	R3	**18**	S-p38	0.398	R1	**16**
pur-s-93	0.029	R2	**35**	S-p42	0.435	R1	**17**
rye-s-93	0.075	R3	**23**	S-p45	0.475	R1	**17**
sta-f-83	0.144	R4	**13**	S-p48	0.502	R1	**18**
tre-s-92	0.181	R4	**23**	L-p5	0.052	R7	**12**
uta-s-92	0.126	R4	**35**	L-p10	0.102	R6	**22**
ute-s-92	0.085	R3	**10**	L-p15	0.155	R4	**28**
yor-f-83	0.289	R4	**21**	L-p20	0.208	R6	**32**
				L-p25	0.264	R4	**35**
				L-p30	0.315	R4	**38**
				L-p35	0.362	R1	**42**
				L-p40	0.418	R1	**45**
				L-p45	0.468	R1	**48**

It is essential to assess the broader generalisability of the proposed generation hyper-heuristic approach by training and testing on structurally dissimilar benchmark sets (e.g., ITC 2007, Carter benchmarks). As future work, we also plan to expand the candidate set to include metaheuristics and local search operators, to investigate combined ISA–LLM distillation pipelines [11], and to apply the methodology to wider timetabling and scheduling domains without abstracting away their real-world complexities.

References

1. Atias, V., Doychev, E.: LLM implementation for multi-objective university timetabling: technical description. In: Proceedings of ICAI 2025 (2025)
2. Burke, E.K., Qu, R., Soghier, A.: Adaptive selection of heuristics for improving exam timetables. Ann. Oper. Res. **218**(1), 129–145 (2012)

3. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the really hard problems are. In: IJCAI-91, pp. 331–337 (1991)
4. De Coster, A., Musliu, N., Schaerf, A., Schoisswohl, J., Smith-Miles, K.: Algorithm selection and instance space analysis for curriculum-based course timetabling. J. Scheduling **25**(1), 35–58 (2022)
5. Drake, J.H., Kheiri, A., Özcan, E., Burke, E.K.: Recent advances in selection hyper-heuristics. Eur. J. Oper. Res. **285**(2), 405–428 (2020)
6. Lewis, R.M.R.: A Guide to Graph Colouring: Algorithms and Applications. Springer (2016)
7. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**(2), 167–256 (2003)
8. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T.G., Lee, S.Y.: A survey of search methodologies and automated system development for examination timetabling. J. Scheduling **12**(1), 55–89 (2009)
9. Rice, J.R.: The algorithm selection problem. Adv. Comput. **15**, 65–118 (1976)
10. Smith-Miles, K., Baatar, D., Wreford, B., Lewis, R.: Towards objective measures of algorithm performance across instance space. Comput. Oper. Res. **45**, 12–24 (2014)
11. Smith-Miles, K., Muñoz, M.A.: Instance space analysis for algorithm testing: methodology and software tools. ACM Comput. Surv. **55**(12), 1–38 (2023)
12. Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q.V., Zhou, D.: Large language models as optimisers. arXiv:2309.03409 (2023)
13. Yang, X., Zhang, L., Qian, H., Song, L., Bian, J.: HeurAgenix: Leveraging LLMs for solving complex combinatorial optimization challenges. arXiv:2506.15196 (2025)
14. Ye, H., Wang, J., Cao, Z., Berto, F., Hua, C., Kim, H., Park, J., Song, G.: ReEvo: Large language models as hyper-heuristics with reflective evolution. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
15. Zhang, S., Liu, S., Lu, N., Wu, J., Liu, J., Ong, Y.S., Tang, K.: LLM-driven instance-specific heuristic generation and selection. IEEE Trans. Emerg. Top. Comput. Intell. (2026). arXiv:2506.00490