LISTEN to Your Preferences: An LLM Framework for Multi-Objective Selection

Anonymous[1]

[1]Anonymous Institution

Abstract Multi-objective optimization often produces large sets of Pareto-optimal solutions, creating a bottleneck for human experts who must select the best option. This difficulty is compounded by the fact that expert preferences are often complex and hard to formalize. To address this, we introduce LISTEN, a framework that leverages a Large Language Model as a zero-shot preference oracle. Guided by a high-level description of an expert's priorities in natural language, LISTEN uses an iterative ranking algorithm to perform pairwise comparisons on subsets of items. This approach contends with practical LLM limitations by using these pairwise judgments to build a preference model, which guides the search and informs the final selection. We evaluate our framework on a real-world final exam scheduling problem. Preliminary results suggest this approach is promising, consistently identifying high-quality solutions and showing encouraging fidelity to the stated preferences. This work demonstrates a path toward steering complex multi-objective selection problem directly with natural language, bypassing the need for mathematical utility functions to express preferences.

1 Introduction

Multi-objective optimization (MOO) of time-consuming black-box functions (Gunantara, 2018; Daulton et al., 2022) is central to automated machine learning (AutoML) (Hutter et al., 2019). For example, when tuning the hyperparameters of a machine learning model or pipeline, one must often trade off speed, memory, and test time accuracy. Moreover, machine learning models are typically time-consuming to train and do not provide derivative information describing how their performance characteristics vary with their hyperparameters.

Multi-objective Bayesian optimization (MOBO) and other algorithms for MOO (Knowles, 2006; Daulton et al., 2022; Tu et al., 2022) can generate hundreds of Pareto-optimal solutions. This creates a new bottleneck. **How can an expert decision-maker efficiently sift through a vast set of viable candidates to find the one that best reflects their nuanced, often unstated, priorities?** This is also a challenge when tuning hyperparameters in generative AI pipelines via preference learning with Bayesian optimization (Christiano et al., 2017; Ouyang et al., 2022; Touvron et al., 2023).

Traditional solutions are often inadequate. Manually comparing all options is time-consuming and prone to error. Pairwise preferences (Obayashi et al., 2007; Wang et al., 2022) or faceted search (Ozaki et al., 2024) can help, but still require significant human effort. The core difficulty is that **human experts lack a time-efficient way to accurately articulate their preferences**.

Large Language Models (LLMs) offer a new paradigm for tackling this challenge. With their profound ability to interpret nuanced, hierarchical text, LLMs present an opportunity for zero-shot preference modeling, where a decision-maker's goals can be understood directly from a verbal description. This bypasses the need for rigid, numerical utility functions. While recent research has begun integrating LLMs into preference learning, they are typically used as components within larger systems—for instance, to guide questioning (Lawless et al., 2023; Austin et al., 2024), extract preferences from reviews (Bang and Song, 2025), or simulate user behavior (Okeukwu-Ogbonnaya et al., 2025; Zhang et al., 2025). However, it remains an open question whether an LLM can, on its

Submitted to AutoML 2025 Non-Archival Track

own, effectively navigate the complex trade-offs inherent in a Pareto frontier using only high-level, natural language instructions.

To address this gap, we introduce **LISTEN**: **L**LM-based **I**terative **S**election with **T**rade-off **E**valuation from **N**atural-language, a framework that uses an LLM as a preference oracle for selection of a human expert's most preferred item from a list that is too long for the human to examine directly. In our framework, a human expert first describes their potentially complex priorities in natural language. An iterative search algorithm then uses this utterance in repeated LLM calls to compare subsets of items, summarizing the LLM's responses via a classical utility surrogate. Ultimately, the algorithm selects its estimate of the best item. Our approach must contend with limits on the LLM's context window and the LLM's ability to reason directly over large item lists, which prevent the LLM from directly selecting the best item in a single call. Our approach must also limit the number of calls to the LLM to save computational and financial cost.

In our experiments, we adopt the real-world multi-objective problem of final exam scheduling, leveraging a large-scale codebase used to schedule final exams at a major university (Ye et al., 2024). This shares the characteristics of MOO in AutoML: evaluating the characteristics resulting from a particular decision requires a time-consuming black box optimization step using many hours of server time; and decision-makers (university registrars) must juggle competing priorities (faculty want to avoid writing make-up exams, students need adequate study time between tests, and administrators aim to clear facilities efficiently).

Our approach combines principles from preference-based optimization (Chu and Ghahramani, 2005; Brochu et al., 2010) and active ranking (Jamieson and Nowak, 2011; Yue et al., 2012), situating these classical methods within the emerging field of LLMs for decision-making (Hao et al., 2023; Xi et al., 2025). While prior work uses LLMs to assist in planning or preference elicitation (Valmeekam et al., 2022; Zhang et al., 2023; Lawless et al., 2023, 2025), we specifically investigate if an LLM can serve as the primary, zero-shot oracle for selecting from a pre-computed Pareto frontier, guided only by high-level natural language goals.

This paper's primary contribution is the exploration of a framework, LISTEN, that uses an LLM as a preference oracle for multi-objective decision-making. Our preliminary evaluation on a real-world final exam scheduling dataset suggests this approach is promising, identifying higher-quality solutions than baselines and showing encouraging fidelity to the stated preferences. We present these initial findings as part of our ongoing work to understand the potential and limitations of using LLMs for multi-objective decision-making.

2 Problem Description

We are given a collection of items $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$. Each item s_i is a d-dimensional vector giving the values for each of d attributes. We are focused on settings where $N > 1000$ is too large for the user to directly examine all items. Such collections are often produced in MOO when a MOO algorithm identifies a large collection of items on the Pareto frontier. We are also given a natural language utterance describing a human decision-maker's preferences, the name of each attribute, and a pre-trained LLM. An example utterance and items are given in the appendix. In our experiments, items are final exam schedules considered by a university registrar. Our goal is to use a limited number of calls to the LLM to select the item that the human decision maker most prefers.

3 Methodology

To address the selection problem, we propose *LLM-based Iterative Selection with Trade-off Evaluation from Natural-language (LISTEN)*, a framework that uses a Large Language Model (LLM) as a zero-shot preference oracle within an iterative ranking procedure.

The LLM as a Zero-Shot Preference Oracle. We leverage an LLM as a surrogate for the human decision-maker. An advantage of our approach is its zero-shot nature; the LLM is not fine-tuned on

any preference data. Instead, we provide it with a carefully crafted natural language prompt that establishes a persona, outlines a hierarchical set of preferences, and presents two or more candidate solutions for evaluation, each represented by its vector of objective values. The LLM's task is to choose the superior solution in the prompt based only on the provided context.

One naive approach for using the LLM to select the item's most preferred item is to directly compare all items in the list in a single prompt. Unfortunately, when N is large, the prompt may not fit in the LLM's context window. We have also found in our experiments that the LLMs we evaluate have a strong bias against items in the middle of the list presented in the prompt, making the preference oracle unreliable for long lists. Another naive approach is to compare pairs of items against each other using the preference oracle and remove those that the LLM judges to be less preferred, only keeping the winner in each comparison. Evaluating all items, however, requires $N - 1$ calls to the LLM, which may be prohibitively expensive when N is large.

Iterative Ranking via Pairwise Comparisons. Our framework uses an iterative process to efficiently explore the solution pool, as detailed in Algorithm 1 in the appendix. The process begins by querying the LLM on a small number of randomly selected item pairs to gather an initial set of preferences. Then, in each iteration, a flexible **selection strategy** is used to choose a new batch of pairs for the LLM to evaluate. This strategy is a modular component of our framework. For instance, an active learning strategy can be employed, where a probabilistic surrogate model (e.g., a Gaussian Process) is fit to the collected preferences, and an acquisition function (e.g., Expected Improvement) is used to select the most informative pairs (Chu and Ghahramani, 2005; Astudillo and Frazier, 2020). Alternatively, a simpler strategy like uniform random sampling can be used. In either case, the LLM's choice for each pair provides a new preference label, which is used to inform subsequent selections.

When we prompt the preference oracle, we include the result of previous comparisons in the prompt. We find that this in-context learning mechanism helps the model establish a more stable internal representation of the trade-offs.

4 Experiments

To evaluate the LISTEN framework, we apply it to the real-world problem of final exam scheduling.

The Final Exam Scheduling Problem. We conduct our experiments using a MOO benchmark introduced in Ye et al. (2024). This benchmark focuses on final exam scheduling using mixed integer programming (MIP). The MIP solver and formulation have hyperparemeters and decision-makers care about a variety of solution attributes (see the appendix for full list). For each attribute, smaller is better. Using ParEGO (Knowles, 2006), we generate a diverse set $\mathcal{S}$ of 5,000 unique candidate solutions on the Pareto frontier.

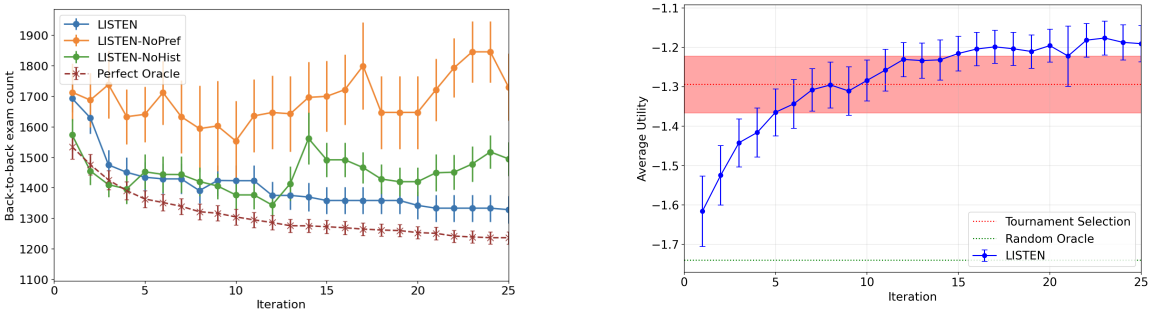
Experimental Setup and Results. To measure performance, we simulate a decision-maker with a known ground-truth utility function. For each experimental run, we define a ground-truth utility as a weighted linear combination of the scheduling conflict metrics. The weights are hidden from all algorithms. The primary evaluation metric is the value of the hidden ground-truth utility function for the final schedule selected by an algorithm. The default LLM is Gemini 2.5 Flash's Reasoning model (Comanici et al., 2025). We run 10 replicates for each algorithm and report the average performance +/- two times the standard error.

We explore several variants of LISTEN. Our main algorithm, **LISTEN**, uses the complete prompt with persona, hierarchical preferences, and conversational history. We test two ablations: **LISTEN-NoPref**, which removes the explicit natural language preferences to test reasoning from context alone, and **LISTEN-NoHist**, which omits the history of previous comparisons to test the impact of in-context learning by making each comparison independent.

To benchmark performance, we use a **Perfect Oracle** as a practical upper bound, which replaces the LLM and makes pairwise comparisons using the true hidden utility function. To isolate the LLM's contribution, a **Random Oracle** baseline uses active search, but the winner of each pair is determined by a coin flip. Finally, as a non-Bayesian alternative, we use a **Tournament Selection** approach, where the LLM selects the best schedule from 50 random batches of 100, and then selects the top 5 from the resulting 50 winners.

Figure 1 compares our approach against baselines and ablations across two ground-truth utility functions. The first setting (Fig. 1a) involves a simple utility over back-to-back conflicts, where we employ an active learning selection strategy using a GP surrogate and the EI-UU acquisition function. The second setting (Fig. 1b) uses a complex, lexicographical utility function that prioritizes simultaneous conflicts, then three-in-a-row, and finally back-to-back conflicts (see appendix for weights); in this more complex case, we use a uniform random search selection strategy.

These results suggest that LISTEN can identify high-quality solutions within a limited budget of LLM calls. Compared to the baselines, LISTEN consistently found higher-utility solutions. It significantly outperformed Random Selection, which confirms that the LLM's reasoning is superior to chance. It also surpassed the non-Bayesian Tournament Selection, highlighting the benefit of iterative surrogate modeling. We observed that providing the utility-driven prompt significantly improved performance over the baseline prompt. Furthermore, including conversational history as context consistently accelerated convergence, suggesting that in-context learning provides value.



(a) Avg. # of Back-to-backs vs. Iterations

(b) Avg. Utility vs. Iterations

Figure 1: Comparison of our proposed LISTEN approach against benchmarks and ablations on two example settings: minimizing the number of back-to-back exams (left) and prioritizing a more complex hierarchical goal (right).

5 Discussion & Future Work

Our initial experiments suggest that LLMs can interpret structured, hierarchical preferences in a zero-shot prompt to make consistent lexicographic choices. This points toward a promising new avenue for preference elicitation that could avoid explicit utility function design.

This capability, however, is not without its limitations. A core challenge is evaluation: without a ground-truth utility function, assessing the "correctness" of the LLM's choices remains an open question. The current framework serves as a snapshot interaction, and future work must explore how to manage evolving preferences and learn from feedback over repeated interactions with a human decision-maker. Moreover, it is worthwhile to explore more sophisticated selection strategies within the iterative ranking procedure for better exploration-exploitation trade-off.

References

Astudillo, R. and Frazier, P. (2020). Multi-attribute bayesian optimization with interactive preference learning. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4496–4507. PMLR.

Austin, D. E., Korikov, A., Toroghi, A., and Sanner, S. (2024). Bayesian optimization with llm-based acquisition functions for natural language preference elicitation. *arXiv preprint arXiv:2405.00981*.

Bang, S. and Song, H. (2025). Llm-based user profile management for recommender system. *arXiv preprint arXiv:2502.14541*.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Chu, W. and Ghahramani, Z. (2005). Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2022). Multi-objective bayesian optimization over high-dimensional search spaces. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR.

Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242.

Hao, S., Gu, Y., Ma, H., Hong, J. J., Wang, Z., Wang, D. Z., and Hu, Z. (2023). Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature.

Jamieson, K. G. and Nowak, R. (2011). Active ranking using pairwise comparisons. *Advances in neural information processing systems*, 24.

Knowles, J. D. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Lawless, C., Li, Y., Wikum, A., Udell, M., and Vitercik, E. (2025). Llms for cold-start cutting plane separator configuration. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 51–69. Springer.

Lawless, C., Schoeffer, J., Le, L., Rowan, K., Sen, S., Hill, C. S., Suh, J., and Sarrafzadeh, B. (2023). "I Want It That Way": Enabling Interactive Decision Support Using Large Language Models and Constraint Programming. *arXiv preprint arXiv:2312.06908*. Submitted December 12, 2023; revised October 1, 2024.

Obayashi, S., Jeong, S., Chiba, K., and Morino, H. (2007). Multi-objective design exploration and its application to regional-jet wing design. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 50(167):1–8.

Okeukwu-Ogbonnaya, A., Amatapu, R., Bergtold, J., and Amariucai, G. (2025). Llm-based community surveys for operational decision making in interconnected utility infrastructures. *arXiv preprint arXiv:2507.13577*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ozaki, R., Ishikawa, K., Kanzaki, Y., Suzuki, S., Takeno, S., Takeuchi, I., and Karasuyama, M. (2024). Multi-objective bayesian optimization with active preference learning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 14490–14498, Vancouver, Canada. AAAI Press.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tu, B., Gandy, A., Kantas, N., and Shafei, B. (2022). Joint entropy search for multi-objective bayesian optimization. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 9922–9938. Curran Associates, Inc.

Valmeekam, K., Olmo, A., Sreedharan, S., and Kambhampati, S. (2022). Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. (2022). Recent advances in bayesian optimization. *arXiv preprint arXiv:2206.03301*.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. (2025). The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Ye, T., Jovine, A., van Osselaer, W., Zhu, Q., and Shmoys, D. B. (2024). Cornell university uses integer programming to optimize final exam scheduling. *arXiv preprint arXiv:2409.04959*.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. (2012). The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556.

Zhang, H., Zhu, Q., and Dou, Z. (2025). Enhancing reranking for recommendation with llms through user preference retrieval. *Proceedings of the 31st International Conference on Computational Linguistics*, pages 658–671.

Zhang, M. R., Desai, N., Bae, J., Lorraine, J., and Ba, J. (2023). Using large language models for hyperparameter optimization. *arXiv preprint arXiv:2312.04528*.

A Algorithm Pseudocode

Algorithm 1 LISTEN: LLM-based Iterative Selection w/ Trade-off Evaluation from Natural-language

Input: Solution pool $\mathcal{S}$, selection strategy π, number of iterations T, batch size K, initial sample size N_{init}.
Select N_{init} random pairs from $\mathcal{S}$; let $\mathcal{S}_{eval}$ be the set of these solutions.
Let $\mathcal{D}$ be the set of preferences obtained by querying the LLM on the initial pairs.
for $t = 1, \ldots, T$ **do**
    Use selection strategy π and preferences $\mathcal{D}$ to select a new batch of K pairs from $\mathcal{S} \setminus \mathcal{S}_{eval}$.
    for each new pair (s_i, s_j) in the batch **do**
        Construct prompt P with persona and solution data.
        Query LLM with prompt P to get preferred solution s_{pref}.
        Add the preference $(s_{pref} \succ s_{other})$ to $\mathcal{D}$.
        Add s_i and s_j to $\mathcal{S}_{eval}$.
    end for
end for
Determine the best solution s^* from $\mathcal{S}_{eval}$ based on the final preferences in $\mathcal{D}$.
Return: s^*.

B Metrics Used in Final Exam Scheduling

Table 1 provides a comprehensive documentation of the metrics used in the final exam scheduling problem.

Table 1: Final Exam Scheduling Conflict Metrics

Metric	Description
Conflicts	Instances of a student having two or more exams in the same time slot.
Quints	Instances of a student having five exams in consecutive time slots.
Quads	Instances of a student having four exams in consecutive time slots.
Four in Five Slots	Instances of a student having four exams within five consecutive time slots.
Triple in 24h (no gaps)	Instances of a student having three back-to-back exams in a 24-hour period.
Triple in Same Day (no gaps)	Instances of a student having three back-to-back exams on the same day.
Three in Four Slots	Instances of a student having three exams within four consecutive time slots.
Evening/Morning B2Bs	Instances of a student having an exam in the last slot of one day and the first slot of the next day.
Other B2Bs	All other instances of a student having exams in adjacent time slots.
Two in Three Slots	Instances of a student having two exams within three consecutive time slots.

C Weights for the Complex Utility Function

The specific (normalized) weights used to define the complex, lexicographical utility function are detailed in Table 2.

Table 2: Normalized Weights for the Complex Utility Function

Conflict Type	Weight
Simultaneous Conflicts	-3
All Triple Conflicts	-2
All Back-to-Back Conflicts	-1

D Prompts

Schedule A: conflicts=1.0, quints=0.0, quads=5.0, four in five slots=3.0, triple in 24h (no gaps)=53.0, triple in same day (no gaps)=31.0, three in four slots=441.0, evening/morning b2b=586.0, other b2b=1303.0, two in three slots=3100.0

Schedule B: conflicts=18.0, quints=0.0, quads=5.0, four in five slots=15.0, triple in 24h (no gaps)=75.0, triple in same day (no gaps)=46.0, three in four slots=456.0, evening/morning b2b=838.0, other b2b=982.0, two in three slots=3163.0

You are an expert in final exam schedule optimization. All metrics represent student exam schedules, they all should be minimized . When comparing Schedule A and Schedule B, provide a few sentence analysis that highlights key trade-offs between their metrics. Conclude with your final choice formatted exactly in curly braces, e.g. {A} or {B}. Do not output just 'A' or 'B'; include the reasoning and marker. Make sure to end with your choice either {A} or {B}

Figure 2: An example of the prompt for the **LISTEN-NoPref** ablation, which omits the explicit preference hierarchy.

You are an experienced University Registrar. Your absolute top priority is to ensure that no student has a simultaneous conflict. After that, your next most critical goal is to minimize the number of students facing three exams in a 24-hour period, as this causes the most stress. Finally, use the number of back-to-back exams as a tie-breaker to choose between otherwise equal schedules. Your goal is to find the schedule that best reflects these priorities. Here are the two schedules:

Schedule A: conflicts=0.0, quints=1.0, quads=3.0, four in five slots=3.0, triple in 24h (no gaps)=46.0, triple in same day (no gaps)=29.0, three in four slots=318.0, evening/morning b2b=420.0, other b2b=1230.0, two in three slots=3086.0

Schedule B: conflicts=1.0, quints=0.0, quads=4.0, four in five slots=7.0, triple in 24h (no gaps)=56.0, triple in same day (no gaps)=34.0, three in four slots=333.0, evening/morning b2b=600.0, other b2b=1198.0, two in three slots=3269.0

Which schedule do you choose?

Figure 3: An example of the **Preference-Guided Prompt Component**, which provides the explicit hierarchical utility information.

[YOUR PREVIOUS REASONING HISTORY]

Here are your previous decisions and reasoning:

Iteration 1:

Schedule A: conflicts: 0, quints: 1.0, quads: 3.0, four in five slots: 4.0, triple in 24h (no gaps): 62.0, triple in same day (no gaps): 20.0, three in four slots: 372.0, evening/morning b2b: 482.0, other b2b: 1106.0, two in three slots: 3481.0

Schedule B: conflicts: 1, quints: 0.0, quads: 5.0, four in five slots: 3.0, triple in 24h (no gaps): 53.0, triple in same day (no gaps): 31.0, three in four slots: 441.0, evening/morning b2b: 586.0, other b2b: 1303.0, two in three slots: 3100.0

Your choice: {B}

Your reasoning: Both schedules are conflict-free or nearly so (A: 0, B: 1), and neither has runaway quints or quads, so attention falls squarely on "two in three slots." Here, Schedule B posts 3100, which is 381 fewer instances than Schedule A's 3481—an 11% improvement on the single metric that matters. Schedule B does incur small increases in four-of-five (3 vs 4), three-in-four (441 vs 372), evening/morning back-to-back (586 vs 482) and other back-to-back exams (1303 vs 1106), but these are secondary. The decisive reduction in two-in-three slots outweighs those trade-offs under the specified priority.

Figure 4: An example of the **Conversational History Component**, which is prepended to the prompt to enable in-context learning based on previous comparisons.