# 差分分析に基づくプロンプトの自動最適化手法

古賀光[1] 西島敏文[1] 木村優介[1] 福島真太朗[1]

[3] トヨタ自動車株式会社{ko_koga, toshifumi_nishijima, yusuke_kimura_zm, s_fukushima}@mail.toyota.co.jp

## 概要

本研究では，大規模言語モデル（LLM）によるプロンプト最適化において，直前と現在のプロンプトに基づく推論結果の差分を用いて要因を分析し，誤答に寄与した文を自然言語的に特定・修正する手法を提案する．従来手法が出力の正誤のみをスカラー値で評価し，プロンプト全体の良否をブラックボックス的に扱っていたのに対し，本手法は回答の変化に着目し，プロンプト内の問題箇所を文単位で明示的に反省・改善する点に特徴がある．駐車場価格を推定する独自のデータセットを構築し，提案手法の有効性を検証した結果，Validation のピーク時の正解率は，提案手法 80.7%，GEPA 70.0%，ProTeGi 64.0%となり最も高い性能を示した．

## 1 はじめに

本章では，プロンプト最適化の現状と課題を概観し，提案手法による解決の方向性と貢献を述べる．

### 1.1 プロンプト最適化の動向

近年，LLM によるプロンプト自動最適化の手法が多く提案されている．ProTeGi [1] は推論誤りを自然言語的に反省し，プロンプト修正の勾配として活用する手法である．APE [2] は例示に基づき複数のプロンプト候補を生成し，出力の尤度を用いて最適なものを選択する探索的手法である．OPRO [3] は，メタプロンプトを用い，評価済みのプロンプト群から LLM に新たな候補を生成させ，その実行結果のスコアを再評価して，より良いプロンプトへと更新していく手法である．EvoPrompt [4] は，プロンプト空間を遺伝的アルゴリズムにより探索し，LLM を用いて突然変異・交叉などの操作を行う手法である．GEPA [5] は自然言語的な反省を通じた修正に加え，正解履歴を用いたパレート最適な選択と進化的併合操作によってプロンプトを改善する手法である．

### 1.2 従来手法の課題

このように，従来手法はプロンプト生成の方法に違いはあるものの，共通しているのは推論結果と真の正解の差分のみを用いている点にある．しかしながら，人間の教師で考えれば，誤答が生じた際にはその思考過程の誤りを特定し，さらにその誤りを引き起こした考え方（すなわちプロンプト）の問題点を明確にする．こうした視点から考えた時，従来手法では誤答の原因分析や修正方針の導出が曖昧であると考えられる．

### 1.3 差分分析法

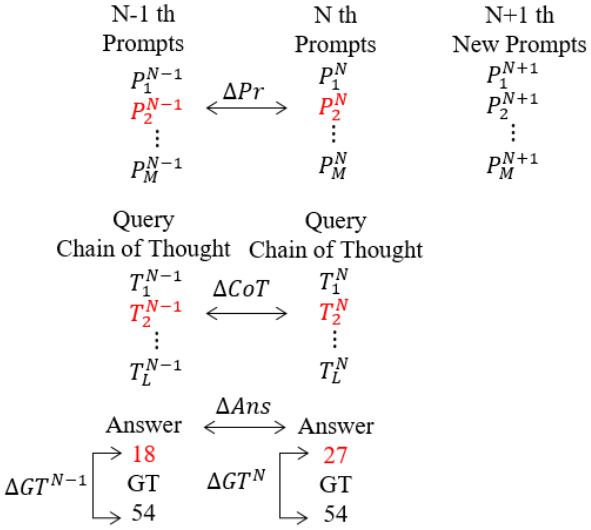


**図 1 差分分析法の概念図**

上述の課題に対し，我々は図 1 に示す差分分析法（**D**ifferential **A**nalysis **M**ethod: 以下，図表中では DAM と称する）を提案する．図は，$N$番目の最適化サイクルにおいて，現在のプロンプト$P_i^N$（合計$M$個），設問（Query），考察過程$T_i^N$（合計$L$個），および回答と正解（$Answer$，$GT$）とその差$\Delta GT^N$を示している．従来手法ではこの$\Delta GT^N$のみから LLM に次のプロンプト$P_i^{N+1}$を生成させるが，その因果は明示されない．差分分析法では，直前のサ

イクル $N-1$ のプロンプト・考察・回答と現在のそれとを比較し，プロンプトの差分 $\Delta Pr$，考察の差分 $\Delta CoT$，回答の差分 $\Delta Ans$を明示的に抽出する．これにより，プロンプト内のどの文が考察と回答に影響したのかを推定し，自然言語により具体的な反省と修正を導くことが可能となる．

### 1.4 独自データセットでの検証

GSM8K [6] や HotpotQA [7] に代表される従来のデータセットは，事前学習に含まれている可能性があると Zhang 等 [8] により指摘されており，プロンプト自動最適化の性能評価には不適切と考えられる．そこで本研究では，駐車場の価格推定を主題とする独自タスクを設計し，未知のデータセットとして検証に用いた．

## 2 方法とデータセット

本章では，提案手法である差分分析法の処理フローと，検証に使用した独自の駐車場データセットについて述べる．

### 2.1 差分分析法

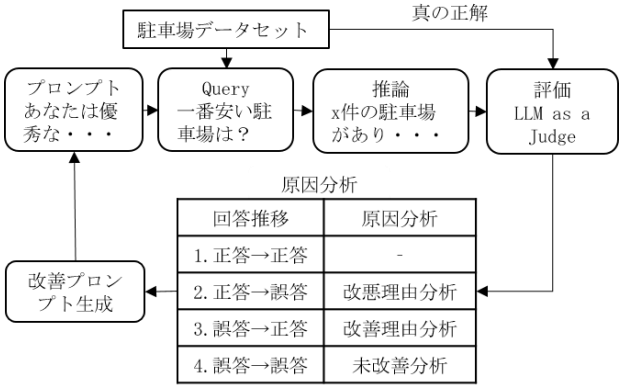


**図 2　差分分析法の流れ**

差分分析法の流れを図 2 に示す．まず，タスク実行時に CoT（Chain-of-Thought）による推論を強制し，出力と真の正解を LLM as a Judge で自動評価する．誤答と判定された結果から任意の 4 件をサンプリングし，現在とひとつ前のプロンプト・推論過程・回答を比較して差分を抽出する．分析対象は現在とひとつ前の回答から 4 事象（正答→正答，正答→誤答，誤答→正答，誤答→誤答）に分類し，それぞれについて推論の変化に寄与したプロンプトの箇所を特定する．最後に，当該箇所の修正案を自然言語で生成することで，新たなプロンプトとする．

### 2.2 検証モデルの設定

検証の公平性を期すために，すべての手法において，原因分析やプロンプトを生成するモデルには GPT-5.2，推論するモデルには GPT-4o-mini を使用した．

### 2.3 駐車場データセット

1.4 で述べた通り，従来の汎用データセットには事前学習の影響がある可能性があるため，駐車場料金に関する質問応答タスクからなる独自の駐車場データセットを構築した．本データセットは，参照・条件分岐・計算・逆引き推論を含む問題を体系的に網羅しており，条件数（日時，駐車時間，車種など），対象施設数（単施設か複数施設か），計算の複雑さ（最大料金の有無，逆算処理など）の 3 要素により難易度をポイントで定量化している．合計ポイントにより，設問を難易度別に低（5 点以下），中（6〜7 点），高（8 点以上）の 3 段階に分類した．Train と Validation セットは同難易度となるように低〜中から選んだ 30 問，Test セットには高難易度となるように中〜高難易度を多く含めた 30 問とすることで，適合と汎化の性能を明確に評価可能な構成とした（表 1）．

**表 1　駐車場データセットの難易度構成**

| Split | Total | Easy | Moderate | Hard |
|---|---|---|---|---|
| Training | 30 | 18 | 9 | 3 |
| Validation | 30 | 18 | 9 | 3 |
| Test | 30 | 6 | 13 | 11 |

## 3 結果

本章では，差分分析法（DAM）と従来手法（ProTeGi, GEPA）を駐車場データセット上で比較した結果を示す．比較の公平性を保つため，各手法とも駐車場データセットの全 30 件に対して評価を行い正解率（Accuracy）を算出した．安定化のため Round ごとの推論評価は 5 回繰り返した平均値で比較した．なお，各手法の最初のプロンプトはすべて同一のものを用いた．

### 3.1 差分分析法と他手法の比較結果

各手法において，Train データでプロンプト最適化を行い，各 Round で得られたプロンプトを用いて Train，Validation および Test の正解率を測定し

た．正解率の推移を表 2 に示した．表において，平均とは各 Round 正解率の平均値，ピーク値はその際の最大値を意味し，改善率はピーク値と Round 0 正解率の差を意味する．また，正解率（Train） - 正解率（Validation）のギャップを ΔT-V，正解率（Train） - 正解率（Test）のギャップを ΔT-T，正解率（Validation） - 正解率（Test）のギャップを ΔV-T と定義する．表 2 を使用して，図 3 に差分分析法（DAM），図 4 に ProTeGi，図 5 に GEPA のグラフを示した．

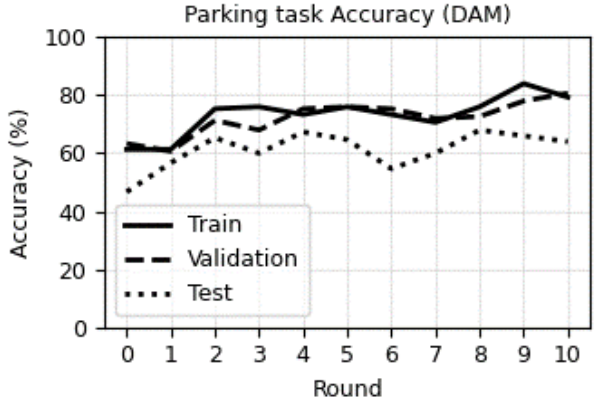


**図 3 差分分析法による駐車場タスクの最適化結果**

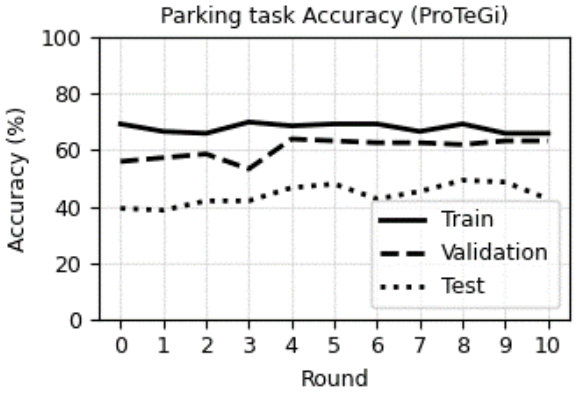


**図 4　ProTeGi による駐車場タスクの最適化結果**

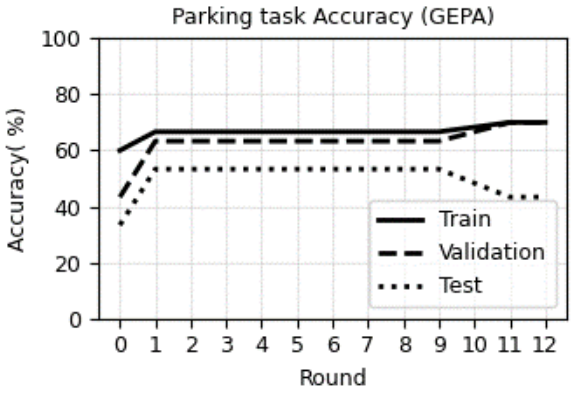


**図 5　GEPA による駐車場タスクの最適化結果**

**表 2　ProTeGi, GEPA, 差分分析法（DAM）による自動最適化による正解率の推移**

| | | Round 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 平均 | ピーク | 改善率 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ProTeGi | Train | 69.3% | 66.7% | 66.0% | *70.0%* | 68.7% | 69.3% | 69.3% | 66.7% | 69.3% | 66.0% | 66.0% | 67.9% | 70.0% | 0.7% |
| | Validation | 56.0% | 57.3% | 58.7% | 53.3% | *64.0%* | 63.3% | 62.7% | 62.7% | 62.0% | 63.3% | 63.3% | 60.6% | *64.0%* | 8.0% |
| | Test | 39.3% | 38.7% | 42.0% | 42.0% | 46.7% | 48.0% | 42.7% | 45.3% | *49.3%* | 48.7% | 42.7% | 44.1% | *49.3%* | 10.0% |
| | ΔT-V | 13.3% | 9.3% | 7.3% | *16.7%* | 4.7% | 6.0% | 6.7% | 4.0% | 7.3% | 2.7% | 2.7% | 7.3% | *16.7%* | - |
| | ΔT−T | 30.0% | 28.0% | 24.0% | 28.0% | 22.0% | 21.3% | 26.7% | 21.3% | 20.0% | 17.3% | 23.3% | 23.8% | 30.0% | - |
| | ΔV−T | 16.7% | 18.7% | 16.7% | 11.3% | 17.3% | 15.3% | 20.0% | 17.3% | 12.7% | 14.7% | 20.7% | 16.5% | 20.7% | - |
| GEPA | Train | 60.0% | 66.7% | 66.7% | 66.7% | 66.7% | 66.7% | 66.7% | 66.7% | 70.0% | 70.0% | 70.0% | 67.0% | *70.0%* | 10.0% |
| | Validation | 43.3% | 63.3% | 63.3% | 63.3% | 63.3% | 63.3% | 63.3% | 63.3% | *70.0%* | 70.0% | 70.0% | 63.3% | *70.0%* | **26.7%** |
| | Test | 33.3% | *53.3%* | 53.3% | 53.3% | 53.3% | 53.3% | 53.3% | 53.3% | 43.3% | 43.3% | 43.3% | 48.8% | *53.3%* | 20.0% |
| | ΔT-V | 16.7% | 3.3% | 3.3% | 3.3% | 3.3% | 3.3% | 3.3% | 3.3% | 0.0% | 0.0% | 0.0% | 3.6% | 16.7% | - |
| | ΔT−T | 26.7% | 13.3% | 13.3% | 13.3% | 13.3% | 13.3% | 13.3% | 13.3% | 26.7% | 26.7% | 26.7% | 18.2% | 26.7% | - |
| | ΔV−T | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 26.7% | 26.7% | 26.7% | 14.5% | 26.7% | - |
| 差分分析法 (DAM) | Train | 61.3% | 61.3% | 75.3% | 76.0% | 73.3% | 76.0% | 73.3% | 70.7% | 76.0% | ***84.0%*** | 79.3% | **73.3%** | **84.0%** | **22.7%** |
| | Validation | 63.3% | 60.7% | 71.3% | 68.0% | 75.3% | 76.0% | 75.3% | 72.0% | 72.7% | 78.0% | ***80.7%*** | **72.1%** | **80.7%** | 17.3% |
| | Test | 46.7% | 56.7% | 65.3% | 60.0% | 67.3% | 64.7% | 54.7% | 60.0% | ***68.0%*** | 66.0% | 64.0% | **61.2%** | **68.0%** | **21.3%** |
| | ΔT-V | -2.0% | 0.7% | 4.0% | 8.0% | -2.0% | 0.0% | -2.0% | -1.3% | 3.3% | 6.0% | -1.3% | **1.2%** | 8.0% | - |
| | ΔT−T | 14.7% | 4.7% | 10.0% | 16.0% | 6.0% | 11.3% | 18.7% | 10.7% | 8.0% | 18.0% | 15.3% | **12.1%** | 18.7% | - |
| | ΔV−T | 16.7% | 4.0% | 6.0% | 8.0% | 8.0% | 11.3% | 20.7% | 12.0% | 4.7% | 12.0% | 16.7% | **10.9%** | 20.7% | - |

表 2 と図 3, 4, 5 の結果から，差分分析法（DAM）は他手法と比較して，Train の平均正解率（73.3%）と改善率（22.7%）が最も高かった．Validation の平均正解率（72.1%）も最も高く，ΔT-V ギャップの平均値が 1.2%と最も小さかった．また，Test の平均正解率（61.2%）と改善率（21.3%)も最も高く，ΔV-T ギャップの平均値が 10.9%と最も小さかった．差分分析法では，Train，Validation，Test の Round が進むにつれて正解率が段階的に向上したが，ProTeGi および GEPA では，正解率の推移は比較的横ばいであり改善は限定的であった．

## 3.2 差分分析法の有効性の検証

2.1 で定義した 4 事象について，図 6 で各 Round における Validation 上の 4 事象の割合変化を示した．改善（誤答→正答）と改悪（正答→誤答）は Round が進むごとに減少しており，特に Round 7 では両者とも 0 件になった．

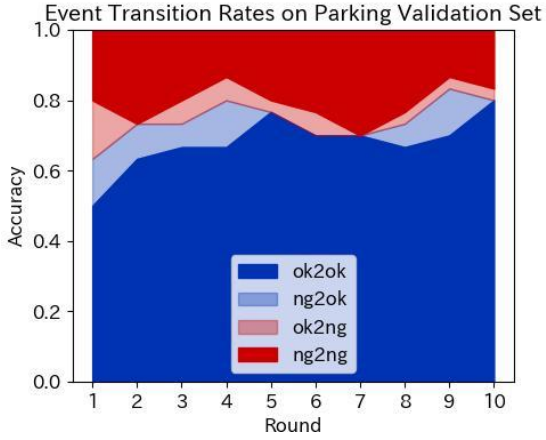


**図 6　Round ごとの 4 事象の割合**

## 3.3 最適化されたプロンプトの比較

各手法において Test 正解率が最も高かった Round のプロンプト中の各章のタイトルを原文のままで抽出し，章内の文字数をそれぞれカウントした結果を表 3 に示す．作業手順のルール，出力の際のフォーマット，トータルの文字数のすべてにおいて，差分分析法が最も多く，次いで GEPA ，ProTeGi の順であった．

**表 3　Test 正解率最大時のプロンプト中の章立てと文字数（原文から章タイトルを抜粋）**

| ProTeGi Accuracy 49.3% | | GEPA Accuracy 53.3% | | 差分分析法(DAM) Accuracy 68.0% | |
|---|---|---|---|---|---|
| 章タイトル | 文字数 | 章タイトル | 文字数 | 章タイトル | 文字数 |
| 指示 | 129 | 目的 | 256 | 指示 | 243 |
| 必須ルール（解釈基準） | 1224 | 入力の前提・形式 | 298 | 情報の優先順位（必ず守る） | 143 |
| 作業手順（内部処理） | 180 | 日付・曜日・期別の判断 | 155 | 料金・適用判定の手順（料金質問では必須） | 5170 |
| 出力スタイル | 193 | 料金計算ルール | 510 | 「最大料金が適用される」の判定ルール | 114 |
| 遵守事項 | 91 | 制限条件の解釈 | 248 | 「最大料金が適用されない」の判定ルール | 605 |
| | | 比較・選定ロジック | 205 | 欠損値（不明情報）の扱いルール | 1010 |
| | | 出力方針 | 204 | 出力ルール | 1266 |
| | | ドメイン固有の注意 | 683 | | |
| | | 不足情報への対処 | 109 | | |
| | | 品質保証 | 145 | | |
| **Total: 1817** | | **Total: 2813** | | **Total: 8551** | |

# 4 考察

3 章より，差分分析法は，Train において平均正解率と改善率が他手法よりも高かったため，訓練データへの適合能力が他手法よりも高い．Validation において平均正解率が最も高く，ΔT-V の平均値は最も小さく 1.2%であったため，同難易度条件において安定した汎化性能を有する．Test においても平均正解率および改善率が最も高く，ΔV-T の平均値が最も小さく 10.9%であった．ΔV-T が小さいことは，難易度変化に対する汎化耐性が高いことを示している．また図 6 では，差分分析法がプロンプトの改善を促進しつつ，改悪を減らすような修正が行われていることが確認され，その有効性が支持された．差分分析法における文単位での明示的修正は，特定タスクの解法や条件に過度に適合する懸念があった．実際，ピーク時の差分分析法のプロンプトは作業手順や出力形式の記述量が最も多かった（表 3）．しかしながら，Test においても最高正解率を示したことから，こうした詳細化は過剰適合に直結せず，むしろ汎化性能の向上に寄与したと考えられる．

本研究では駐車場データセット単一での検証に留まっており，多様な推論構造を持つタスクへの適用可能性については今後の検証が必要である．

## 参考文献


[1] Pryzant, R., Iter, D., Li, J., Lee, Y.T., Zhu, C. and Zeng, M., 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

[2] Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022, November). Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

[3] Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2023, September). Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

[4] Guo, Q., Wang, R., Guo, J., Li, B., Song, K., Tan, X., ... & Yang, Y. (2023). EvoPrompt: Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

[5] Agrawal, L. A., Tan, S., Soylu, D., Ziems, N., Khare, R., Opsahl-Ong, K., ... & Khattab, O. (2025). Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.

[6] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021). Training Verifiers to Solve Math Word Problems. *ArXiv, abs/2110.14168*.

[7] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2369-2380).

[8] Zhang, H., Da, J., Lee, D., Robinson, V., Wu, C., Song, W., ... & Yue, S. (2024). A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, *37*, 46819-46836.

# A 付録（駐車場データセット）

1．データセット概要（抜粋）

本データセットは，以下の駐車場の料金体系を表す施設データと設問データから構成される．

**表 1　施設データ例**

| 名称 | 駐車料金 | 営業時間 | 収容数 | 方式 | 制限 |
|---|---|---|---|---|---|
| 駐車場B | 【繁忙期 7/1-8/31】<br>普通車　06:00～21:00　260円/30分　21:00～翌06:00　1泊/1040円<br>大型車　06:00～21:00　790円/30分　21:00～翌06:00　1泊/3140円<br>【通常期 9/1-6/30】<br>普通車　09:00～18:00　210円/30分　18:00～翌09:00　1泊/1040円<br>大型車　09:00～18:00　790円/30分　18:00～翌09:00　1泊/3140円 | 9:00～18:00(通常期間)<br>6:00～21:00(7/1～9/8)<br>9:00～24:00(12/31)<br>0:00～18:00(1/1)<br>7:00～18:00(1/2～1/3)<br>定休日:無休 | 200台<br>身障者専用6台 | 地下(自走式) | 高:2.10m まで<br>長:5.60m まで |
| 駐車場C | 【時間料金】(平日)8:00-20:00 ￥100/20分 20:00-翌8:00 ￥100/60分<br>(土日祝)8:00-20:00 ￥300/30分 20:00-翌8:00 ￥100/60分<br>【最大料金】(平日)8:00-20:00 ￥700 (繰返し可)<br>(全日)20:00-翌8:00 ￥150 (繰返し可) | 24時間 | 20台 | 平地(自走式) | 高さ:2.10mまで<br>幅:1.90mまで<br>長さ:5.00mまで<br>重量:2.50tまで |
| 駐車場D | 【通常料金】<br>（月～金）00:00-00:00 20分 330円、当日１日最大料金1700円(24時迄)<br>（土日祝）00:00-00:00 20分 330円、当日１日最大料金2500円(24時迄) | 24時間入出庫可 | 8台 | 平地(自走式) | 全高2.1m<br>全長5m<br>全幅1.9m |

**表 2　設問データ例（Train 3/30）**

| No | 難易度 | 日付 | 時刻 | 質問 | 期待回答 |
|---|---|---|---|---|---|
| 1000 | 低 | 2025/12/9(火) | 8:00 | 駐車場Bに普通車で9:00から15:00まで駐車したときの料金を教えて。 | 駐車料金は2520円です。 |
| 1100 | 中 | 2025/12/12(金) | 9:00 | 普通車で10:00から15:00まで駐車した時の駐車場Cより安い駐車場は？ | 駐車場Cより安い駐車場はありません。 |
| 1200 | 高 | 2025/08/02(土) | 9:00 | 駐車場Cに10:00から駐車する場合、700円でいつまで停められますか？ | 11:00まで駐車できます。 |

**表 3　設問データ例（Test 3/30）**

| No | 難易度 | 日付 | 時刻 | 質問 | 期待回答 |
|---|---|---|---|---|---|
| 3000 | 低 | 2025/12/13(土) | 18:30 | 駐車場Cに19:00から21:00まで駐車した時の料金は？ | 駐車料金は700円です。 |
| 3100 | 中 | 2025/12/10(水) | 12:00 | 普通車で13:00から14:00まで駐車した時、420円未満のところは？ | 駐車場Cです。 |
| 3200 | 高 | 2025/12/13(土) | 15:00 | 16:00から45分間だけ駐車した時に、駐車料金が600円未満の駐車場は？ | 駐車場Bです。 |

2．難易度のポイント

本データセットの設問の難易度は、以下の 3 要素のポイントの合計により算出する．

**表 4　難易度算出のポイント**

| 条件 | | 対象施設数 | | 計算の複雑さ | |
|---|---|---|---|---|---|
| 内容 | ポイント数 | 内容 | ポイント数 | 内容 | ポイント数 |
| 条件ひとつにつき | 1 | 対象施設数が1つ | 1 | 計算不要、参照のみ | 0 |
| | | 対象施設数が2つ以上 | 2 | 1つの料金体系の計算が必要 | 1 |
| | | | | 2つの料金体系をまたぐ計算が必要 | 2 |
| | | | | 3つ以上の料金体系をまたぐ計算が必要 | 3 |
| | | | | 質問に最大料金を考慮した計算が必要 | 1 |
| | | | | 質問に境界値が含まれる | 1 |
| | | | | 質問に端数処理の計算が必要 | 3 |
| | | | | 逆引き計算が必要 | 3 |

【難易度算出例】

質問：[2025/12/12(金)]　駐車場 C に 10:00 から駐車する場合、700 円でいつまで停められますか？

ポイント：条件[4 ポイント]＋施設[1 ポイント]＋計算[4 ポイント]＝合計[9 ポイント]　（難易度　高）