State-Dependent Pricing & Matching Policies for optimizing ride-sharing profit

Aparimit Kasliwal[1], Dimitris Triantis[1], and Lejun Zhou[1]

[1]Department of Civil and Environmental Engineering, University of California, Berkeley

May 13, 2024

Abstract

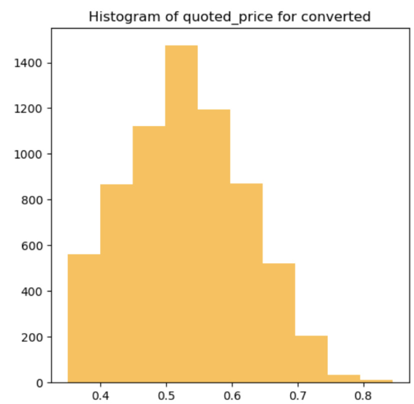
This report examines a strategic approach to pricing and matching in the Calyber Game. The simulation challenges teams to maximize profits through dynamic pricing and efficient matching of riders, who arrive unpredictably and can accept or reject quoted prices. Our strategy employs a sophisticated algorithmic framework to manage real-time pricing and rider pairing, emphasizing the optimization of revenue and cost-effectiveness. This approach leverages extensive data analysis on rider behaviors and geographic pricing patterns, enabling informed decision-making under the constraints of rider patience and availability of matches. The results highlight the effectiveness of our strategies in enhancing profitability while addressing the complexities inherent in a dynamic urban transportation environment. This abstract encapsulates our key methodologies, findings, and the implications of our strategies on the operational success of ride-sharing platforms.

1 Introduction

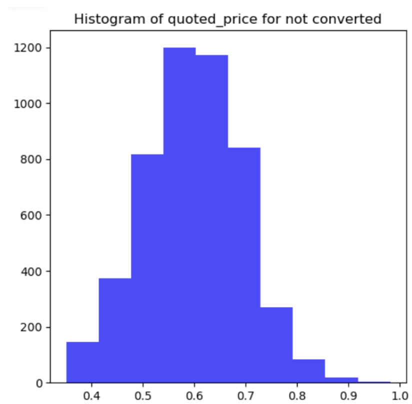
Calyber is a ride-sharing platform in Chicago which offers shared ride services where it matches two riders with similar routes to share a ride, reducing costs and fares. In this game, drivers are assumed to be infinitely available. Thus, teams focus on pricing. Riders arrive randomly, and each is quoted a price which they can accept or decline. Accepted riders are attempted to be paired for shared rides. If a suitable match isn't available immediately, the platform can delay matching, although each rider's limited waiting tolerance might force solo dispatches, increasing costs. The game simulates the platform's role in maximizing profit by managing pricing and pairing decisions. The data that each team is given consists of a set of historical rider arrival and request data as well information about the Chicago Community Areas that are included. Using these data, each team is tasked to develop a matching and pricing policy that aims to maximize their profit. In this report we will discuss our approach to the Calyber Game.

2 Exploratory Data Analysis

2.1 Conversion Rate & Quoted Price



(a) Distribution of quoted price for converted riders



(b) Distribution of quoted price for not-converted riders

Figure 1: Comparison of quoted prices

2.2 Heterogeneity in Quoted Price & Count of Rides Based on Geographical Variation

The primary objective of this phase of analysis is to delve into the intricate and diverse distribution patterns of quoted prices, which exhibit significant variability based on the specific pickup and dropoff locations of the rides. This exploration allows us to uncover nuanced trends and disparities in pricing strategies, revealing how different factors such as distance, demand, and local market conditions impact the prices quoted to the riders. By comprehensively examining these variations, we can better understand the complexities of pricing mechanisms within the transportation network, enabling us to formulate more informed strategies for pricing optimization and service delivery.

The way we construct the weights for different areas, discussed in one of the following sections, also depends on the number of rides that started and ended in a specific area.

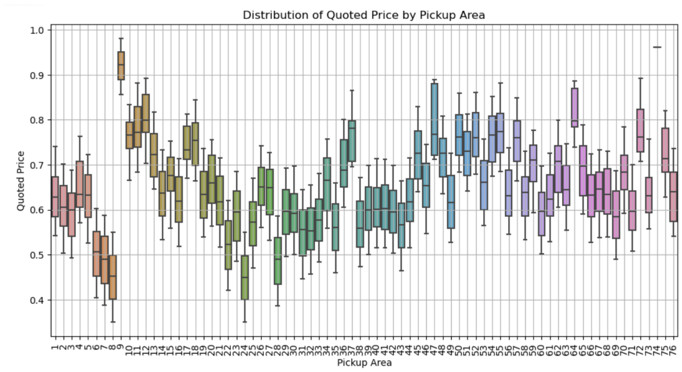


Figure 2: Quoted Price by Pickup-Area

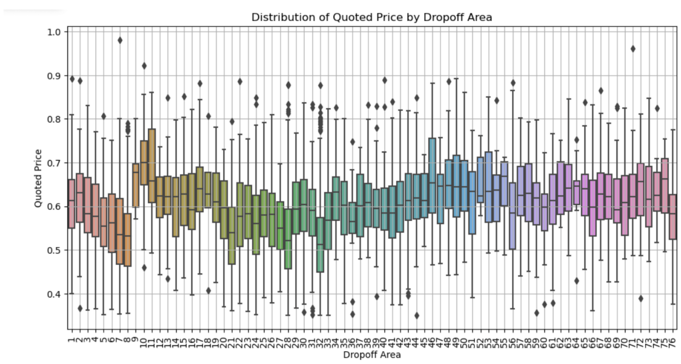


Figure 3: Quoted Price by Dropoff-Area

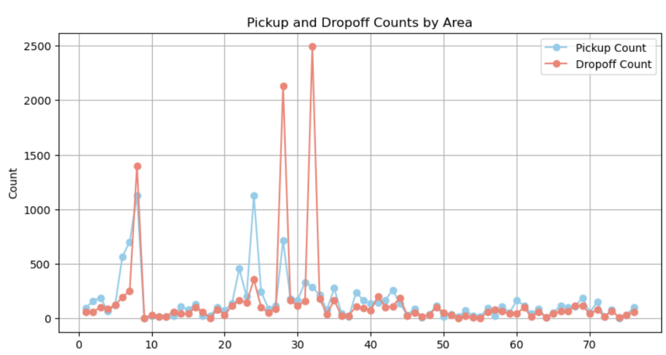


Figure 4: Pickup & Dropoff counts by area

The conclusion drawn from the above three plots is that the quoted price is lower for trips originating from and terminating at one of those regions that has a relatively higher demand.

2.2.1 Aggregating Area IDs

In order to also take into account the relative geographical distance between different pickup & dropoff locations, we aggregated these points into different Uber H3 Hexagons. The intuition for why we did this is as follows: we want to quote a lower price not only for those regions that have a higher pickup / dropoff count, but also for those regions that are **geographcially near** those high demand regions.

The following figure captures the process of aggregating different points based on their geographical closeness into H3 Hexagons at resolution 8.

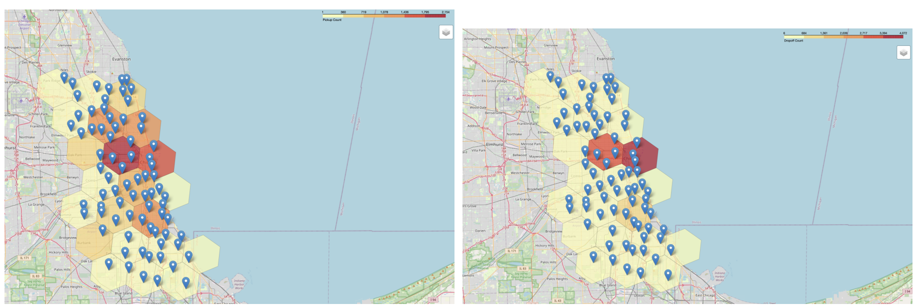


Figure 5: Caption for the two figures side by side

2.3 Understanding Willingness to Pay & Maximum Waiting Time

This part of analysis gives us an insight into two important parameters that can be deduced from the data, but are not explicitly provided. For the quoted price, understandably, the fraction of converted riders decreases as the quoted price is increased. Getting the exact value of this conversion fraction for different ranges of the quoted price was very helpful for us when we coded our own simulation to test our pricing and matching algorithms, as discussed later.

For the waiting time, we leveraged the fact that maximum waiting time of riders would be drawn from / can be well-approximated by an exponential distribution. The following figure shows the distribution of waiting times of those riders who were matched with other riders.

We found the parameter of the exponential distribution to be 69 if we considered only the training data, and 95 if we considered both the training & the validation data.

Our pricing and matching policies are dynamic, tailored to adapt to the evolving state of our system. These policies are outlined in detail in subsequent

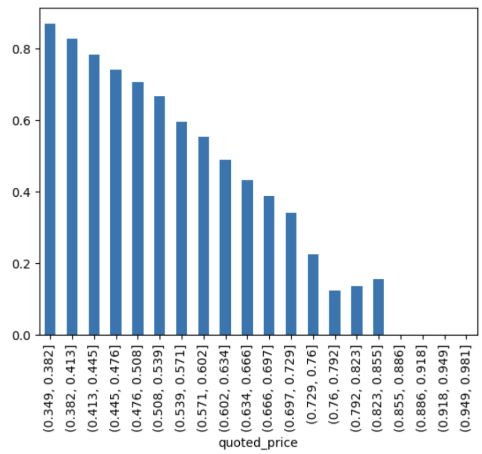


Figure 6: Fraction of converted riders Vs Quoted Price

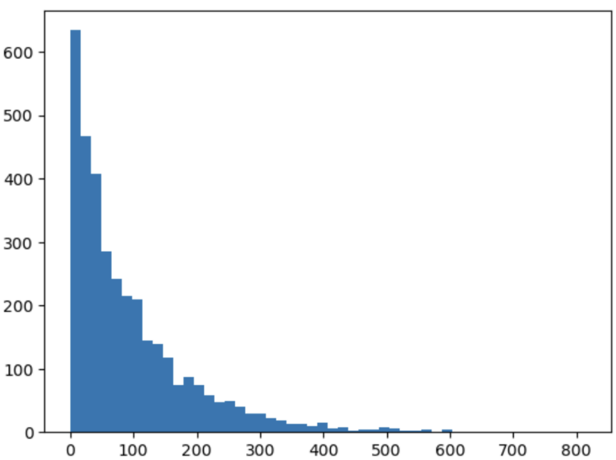


Figure 7: Histogram of Waiting Times

sections. By considering factors such as proximity to other riders, demand levels in specific regions, and minimum pricing thresholds, we aim to ensure efficient matching and fair pricing while maintaining profitability. This dynamic approach enables us to respond effectively to changing market conditions and rider demands, ultimately enhancing the overall profit.

3 Methodology

3.1 Pricing Algorithm

For the pricing policy, if riders in the system are within the top 5 nearest neighbors of a new rider's pickup or dropoff location, we offer lower prices to encourage easier matching. Additionally, pricing adjustments are made based on the demand level of the pickup and dropoff regions, with lower prices for high-demand areas and potentially higher prices for low-demand areas. Lastly, to maintain profitability, we enforce a minimum price threshold of 0.65, ensuring that prices below this level are not quoted to avoid financial loss.

The parameters we consider are:

- Neighbours_origin: List containing top-5 neighbour area_IDs of the origin
- Neighbours_destination: List containing top-5 neighbour area_IDs of the destination
- Hexagon_weights: The ratio of trips within a particular area_ID

Algorithm 1: Pricing Function Algorithm

Input: state s, rider instance i;
Output: quoted price p;
Parameter: neighbours_origin and neighbours_destination;

Initialization: hexagon_weights for areas, $\gamma \leftarrow 0$;

Step 1: Calculate Gamma;
foreach *rider_ in state* **do**
    if *rider_.pickup_area* $\in$ *neighbours_origin* $\vee$ *rider_.dropoff_area* $\in$ *neighbours_destination* **then**
        $\gamma \leftarrow \gamma + 1$;
    end
end
if $\gamma \geq 3$ **then**
    $\gamma \leftarrow 3$;
end

Step 2: Determine Pickup and Dropoff Variables;
pickup_variable $\leftarrow$ Get weight from hexagon_weights for i.pickup_area;
dropoff_variable $\leftarrow$ Get weight from hexagon_weights for i.dropoff_area;

Step 3: Compute Price;
price $\leftarrow 0.65 + \alpha_{\text{pickup}} \times (0.182728 -$ pickup_variable$) + \alpha_{\text{dropoff}} \times (0.345436 -$ dropoff_variable$) - \alpha_{\text{state}} \times \gamma$;

Step 4: Enforce Minimum Price;
if *price* ≤ 0.65 **then**
    price $\leftarrow$ price $+0.1 \times (0.2 -$ pickup_variable$) + 0.1 \times (0.4 -$ dropoff_variable$)$;
end

Return price

3.2 Matching Algorithm

In the matching, we evaluate three key factors and employ them to compute matching scores for all waiting riders. These scores determine whether an existing rider in the state will be matched or if the new arrival will be added to the state. Here, we delve into these four factors and elucidate why they are pivotal in our matching process.

Our first consideration is the reduction in trip length achieved through matching compared to dispatching the two riders individually. This factor directly impacts the final cost. Secondly, we assess the waiting time of the rider until the current timestamp, ensuring it remains within acceptable limits to avoid exceeding maximum waiting times. The last factor is the demand at pickup and drop-off locations. In areas with high demand, finding a better matched rider becomes more feasible, thus elevating their value within the state. After we get all the scores, we will take the waiting rider with highest score as the best match. In addition, there is a standard for matching, if the highest score is still lower than the standard, it will not be matched even though this pair gets the highest score. Below is the pseudo-code for the algorithm:

Algorithm 2: Matching Function Algorithm

```
Input: state s, arriving rider instance r;
Output: best waiting rider instance m, or None if no match;
Parameters: hexagon_weights for areas;
Initialization: score_list ← empty list;
Set origin and destination coordinates for rider r as origin_0_coordinate,
  destination_0_coordinate;
foreach waiting rider w in state do
    for hexagon weight h in hexagon_weights do
        if r.pickup_area is in h then
            pickup_variable_0 ← corresponding pickup weight from h;
        end
        if r.dropoff_area is in h then
            dropoff_variable_0 ← corresponding dropoff weight from h;
        end
        if w.pickup_area is in h then
            pickup_variable_1 ← corresponding pickup weight from h;
        end
        if w.dropoff_area is in h then
            dropoff_variable_1 ← corresponding dropoff weight from h;
        end
    end
    L_shared ← calculate shared ride length between origin_0_coordinate,
      destination_0_coordinate, and w's coordinates;
    t_j ← max(r.arrival_time − (w.arrival_time + θ/2), 0);
    score_w ← (r.solo_length + w.solo_length − L_shared) + β_1 · t_j − β_2 ·
      (pickup_variable_0 + pickup_variable_1) − β_3 ·
      (dropoff_variable_0 + dropoff_variable_1);
    Append score_w to score_list;
end
if score_list is not empty then
    m ← waiting rider with the highest score in score_list;
    if maximum score in score_list < threshold β then
        m ← None ; // No suitable match found within threshold
    end
end
else
    m ← None ;                    // No waiting riders to match
end
Return m
```

3.3 Simulation and Policy Ascent

To determine the optimal parameters, we conduct simulations on both training and validation data. These simulations involve calculating the expected total profit across all instances and employing policy ascent methodology.

Policy ascent is an iterative optimization technique used to improve decision-making policies. It involves adjusting parameters incrementally in the direction that maximizes the objective function—in this case, total profit.During each iteration of policy ascent, we evaluate the performance of the current policy and adjust the parameters accordingly to improve its effectiveness. This iterative process continues until a satisfactory level of performance is achieved or until convergence is reached, indicating that further adjustments yield marginal improvements. Ultimately, policy ascent enables us to refine our decision-making strategy, leading to improved outcomes and enhanced profitability in our operations.

4 Results

4.1 Simulation Result

In a single instance of our simulation, we evaluated the performance of our pricing and matching algorithms using specific parameter settings. The pricing parameters were set at $\alpha_{\text{pickup}} = 0.25$, $\alpha_{\text{dropoff}} = 0.25$, and $\alpha_{\text{state}} = 0.01$, aimed at reflecting the significance of pickup and dropoff locations as well as the state of the service environment. The matching algorithm was configured with parameters $\beta_1 = 0.002$, $\beta_2 = 0.5$, $\beta_3 = 0.5$, $\beta = 0.09$, and $\theta = 95$, designed to optimize the service matching based on predefined criteria.

The simulation revealed that out of the potential user interactions, 683 riders converted by accepting the quoted prices and completing their trips, while 650 riders were successfully matched with services. This underscores the efficacy of the matching algorithm. However, there were 33 unhappy riders, indicating potential discrepancies in pricing or service expectations that could be addressed in future iterations. The economic analysis of the simulation demonstrated a high level of efficiency with a total profit of 115.743 units. This profitability reflects the financial viability of the model under the current parameter settings.

4.2 Testing Results

When it came to the final simulation, where all the teams competed on the test data, our team ranked 4th in total, not including the benchmark policies. The profit per minute for all teams can be seen in Figure 8. It should be noted that the test data contained three different weeks of data collected from the same time window as the training data. Our team performed decently, achieving a profit per minute that is very close to the one of the winning, especially comparatively to all teams, since as seen in Figure 8 some teams were not able to achieve profitability.

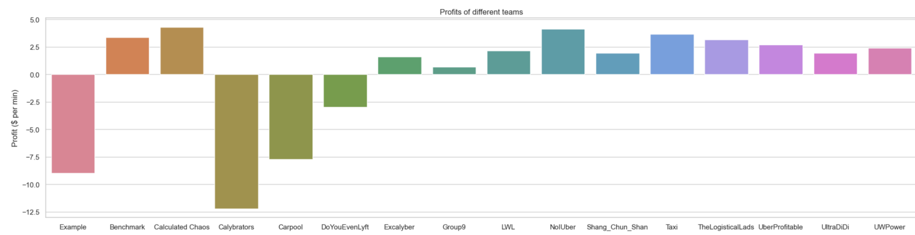


Figure 8: Profit ($/min) by team

When it comes to other relevant metrics, such as the ones seen in Table 1, our team was able to rank around the middle for most of them. This shows that our problem formulation and policies are intuitively sound but there is potential for further improvement.

Table 1: Performance Metrics for TheLogisticalLads

Metric	**Value**	**Rank**	**Metric**	**Value**	**Rank**
Profit ($/min)	3.18	4th	Cost ($/min)	21.31	6th
Cost Efficiency	0.14	8th	Throughput (#/min)	10.11	6th
Match Rate	0.73	5th	Conversion Rate	0.36	7th
Revenue ($/min)	24.49	9th	Ave Payment ($/mile)	0.69	5th
Ave Quoted Price ($/mile)	0.70	7th	Ave Waiting Time (sec)	34.04	7th

Despite securing the 4th rank among competitive teams and demonstrating substantial profitability, our approach indicated potential for further enhancement, particularly if more time were available for refining our model's parameters.

Our strategy fundamentally depends on accurately estimating coefficients that control pricing adjustments and rider matching decisions based on real-time data. These coefficients $(\alpha_{\text{pickup}}, \alpha_{\text{dropoff}}, \alpha_{\text{state}}, \beta_1, \beta_2, \beta_3, \theta)$ are pivotal in determining the optimal pricing and the effectiveness of our matching algorithms. With the limited time frame of the simulation, we established preliminary values that proved effective yet could be further improved.