

A Framework of Multi-stage Dynamic Pricing for Meal-Delivery Platform

Z. Wu^{a,*}, L. Wang^{a,*}, F. Huang^a

^aMeituan Group

Abstract

Online meal delivery is undergoing explosive growth, as this service is becoming increasingly fashionable. A meal delivery platform aims to provide efficient services for customers and restaurants. However, in reality, several hundred thousand orders are canceled per day in the Meituan meal delivery platform since they are not accepted by the crowdsourcing drivers, which is detrimental to the interests of multiple stakeholders: customers, crowdsourcing drivers, restaurants, and the meal delivery platform. Therefore, allocating bonus is an effective means to encourage crowdsourcing drivers to accept more orders. In this study, we propose a framework to deal with the multistage bonus allocation problem for a meal delivery platform. The objective of this framework is to maximize the number of accepted orders within a limited bonus budget. This framework consists of a semi-black-box acceptance probability model, a Lagrangian dual-based dynamic programming algorithm, and an online algorithm. The semi-black-box acceptance probability model is employed to forecast the relationship between the bonus allocated to an order and its acceptance probability, the Lagrangian dual-based dynamic programming algorithm aims to calculate the empirical Lagrangian multiplier for each allocation stage offline based on the historical data set, and the online algorithm uses the results attained in the offline part to calculate a proper delivery bonus for each order. To verify the effectiveness and efficiency of our framework, both offline experiments on a real-world data set and online A/B tests on the Meituan meal delivery platform are conducted. Our results show that using the proposed framework, the total order cancellations can be decreased by more than 30% in reality.

Keywords: multistage bonus allocation, meal delivery platform, Lagrangian dual-based dynamic programming, convex optimization, real-time optimization

1. Introduction

With the explosive growth of online meal delivery, it is becoming an essential service in our daily life. For example, Meituan, the most popular Chinese meal delivery platform, takes 30 million meal orders each day. The platform aims to provide effective and efficient services for restaurants and customers.

^{**}Both authors contributed equally to this research.

Email addresses: wuzhuolin@meituan.com (Z. Wu), mxdsjyd@163.com (L. Wang), huangfangsheng@meituan.com (F. Huang)

A typical process for the meal delivery platform is depicted in Figure 1. After the customer orders a meal through the Meituan application, the corresponding order information is immediately sent to the meal delivery platform. First, the pricing system determines the delivery price for the order based on its properties, such as restaurant and customer locations, customer-side service difficulty, and so on. Second, the order is visible to nearby crowdsourcing drivers. Third, the crowdsourcing driver accepts the order, picks up the meal from the restaurant, and delivers it to the customer. However, if the delivery price is not attractive enough, the order might be unaccepted by the crowdsourcing drivers. Furthermore, the order is canceled by the customer or the platform since no one accepts it for too long. As a result, there are several hundred thousand canceled orders in the Meituan meal delivery platform per day, which is detrimental to the interests of multiple stakeholders: customers, crowdsourcing drivers, restaurants, and the meal delivery platform. Order cancellations mean less income for the crowdsourcing drivers, lower quality of user experience, more food waste for restaurants, and a lower reputation of the meal delivery platform. In addition, the Meituan meal delivery platform pays restaurants about billions of RMB per year to compensate for food waste. Therefore, there is an urgent need for a bonus allocation mechanism to increase total order acceptance.

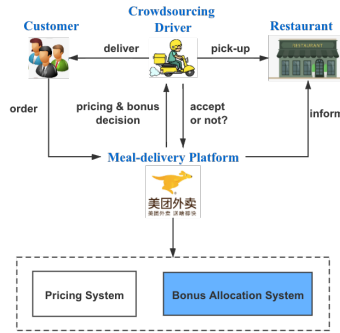


Figure 1: The typical process in the meal delivery platform.

In this paper, we develop a multistage bonus allocation problem (MSBA) for the meal delivery platform. Zhao et al.[23] studied a marketing budget allocation problem similar to ours. However, they solved a single-stage problem, while we deal with a multistage one. According to the historical data analysis, the acceptance probability of an order decreases since its delay possibility increases as time passes. Therefore, to increase the acceptance probability of an order, the delivery bonus needs to be changed multiple times as time passes, i.e., multistage bonus allocation.

Let N be the order set placed during a week or month. The bonus allocation decision could be made $|T|$ times at the most before they are accepted by the crowdsourcing drivers. As orders are accepted, the size of the order set decreases. Otherwise, the unaccepted orders in the former allocation stage are transited to the next allocation stage. As shown in Figure 2, N_t means the remaining order set transited to the allocation stage t with $|N_1| > |N_2| > |N_0| > \dots > |N_{|T|}|$. Normally, the time interval between the two allocation stages

is a constant value ranging from 1 to 10 minutes.

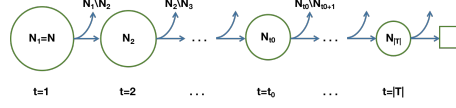


Figure 2: The order transitions.

To the best of our knowledge, this is the first study to discuss MSBA for the meal delivery platform, which is not trivial because of the following challenges:

- Long-term budget constraints need to be addressed. In this problem, the total available bonus is restricted within a predetermined weekly/monthly budget. However, all information regarding orders placed during a week/month cannot be attained, while bonus allocation decisions must be made for each order in real-time.
- This problem involves multistage decision-making. Properly allocating a bonus for each order at each allocation stage is another challenge.
- In reality, the delivery bonus allocation decision for an order must be performed in a few milliseconds. To meet such a stringent computational target, an efficient algorithm is required to generate fast and efficient bonus allocation decisions.

The main contribution of this study is the development of a multistage bonus allocation framework. This framework includes an acceptance probability model, a Lagrangian dual-based dynamic programming (LDDP) algorithm, and an online algorithm. The semi-black-box acceptance probability model is employed to forecast the relationship between the bonus allocated to an order and its acceptance probability. The LDDP algorithm is developed to calculate an effective parameter (Lagrangian multiplier) for each allocation stage offline based on the historical dataset. This algorithm is the most significant novelty of this paper. The resulting parameters are used to make the bonus allocation decisions online. The combination of the offline and online calculations can deal with the long-term budget constraints. Particularly, the proposed framework can make quick decisions in real-time with the computational complexity of $O(1)$ for the online calculation. Moreover, it is easy to implement and can be successfully applied to the Meituan meal delivery platform. Our online A/B tests show that the number of canceled orders is reduced by more than 30% compared to the previous unified bonus mechanism in which the same delivery bonus is imposed on each order.

The remainder of this paper is organized as follows: In Section 2, we first describe our problem and present a formal mathematical formulation. To solve this problem, in Section 3, we introduce an our algorithm based on dynamic programming and Lagrangian dual theory. In Section 4, we report the computational results. Section 5 presents a brief review of the existing work related to our proposed problem. Finally, we conclude the paper in Section 6.

2. Problem Description and Mathematical Formulation

2.1. Acceptance Probability Model Forecasting

As previously mentioned, the aim of MSBA is to maximize the number of accepted orders. To formulate this, our objective function is to maximize the expected value of the number of accepted orders based on the acceptance probability of orders. With multiple allocation stages, an unaccepted order from a former allocation stage is transited to the next allocation stage. The transition process of the order is illustrated in Figure 3. Each node represents an allocation stage. Let p_1 be the acceptance probability in the first allocation stage. Correspondingly, $1 - p_1$ is the probability of transitioning to the second allocation stage. Consequently, the acceptance probability after the allocation stage $|T|$ is $\prod_{t_0=1}^{|T|-1} (1 - p_{t_0}) p_{|T|}$. Otherwise, the order is canceled with the probability of $\prod_{t_0=1}^{|T|} (1 - p_{t_0})$. In total, the probability of an order being accepted is expressed as follows:

$$p = \sum_{t=1}^{|T|} \prod_{t_0=1}^{t-1} (1 - p_{t_0}) p_{|T|} \quad (1)$$



Figure 3: The transition process of an order.

To make optimal delivery bonus allocation decisions, we need to forecast the relationship between the bonus allocated to an order and its acceptance probability. Although black-box forecasting methods, such as neural networks, are widely used in many applications, there are still gaps between the black-box model and bonus allocation decision-making. Instead, in this study, we use a semi-black-box forecasting method. We assume that the acceptance probability model conforms to the following logistic function:

$$p_{i,t}(c_{i,t}) = \frac{1}{1 + e^{\alpha_{i,t} c_{i,t} + \beta_{i,t}}} \quad (2)$$

Meanwhile, $\alpha_{i,t}$ and $\beta_{i,t}$ are attained by the machine learning model such as neural network. The input features of the forecasting model can be split into two different parts: the bonus allocated to each order $c_{i,t}$ and contextual features $\mathbf{x}_{i,t}$. Contextual features are intrinsic attributes of the order, including the geographical locations of the customer and the restaurant, estimated time of arrival (ETA), and others. The training set $\{[\mathbf{x}_{i,t}, c_{i,t}], p_{i,t}^*\}$ is constructed for each order i at each allocation stage t from historical observations. Contrary to the method in the reference[23] where $\alpha_{i,t}$ and $\beta_{i,t}$ are learned separately, in this study, we learn $\alpha_{i,t}$ and $\beta_{i,t}$ simultaneously but with different hidden layers (see Figure4). In practice, the bonus is only allocated to a minority of orders such that the sample distribution of the training set is uneven. Therefore, we divide the training set into two kinds of batches: bonus batches with $c_{i,t} > 0$ and normal batches with $c_{i,t} = 0$. As in reference[23], $\beta_{i,t}$ is determined by contextual features $\mathbf{x}_{i,t}$. Therefore, to improve the performance of the model, the hidden layers of $\alpha_{i,t}$ and $\beta_{i,t}$ are updated using different kinds

of batches. More specifically, the bonus batches are used to update the parameters of hidden layer 0 and layer 1, and the normal batches are used to do the same for hidden layer 0 and layer 2. Note that the attained $\alpha_{i,t}$ is less than 0 which is consistent with common sense of the more the bonus, the larger the acceptance probability.

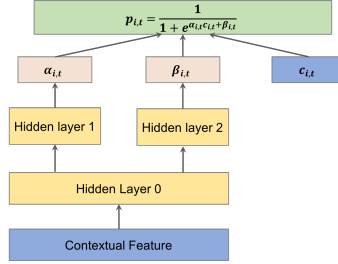


Figure 4: A simple network structure of the acceptance probability model.

The acceptance probability model of orders is illustrated in Figure 5. As shown, the acceptance probabilities of the two orders may differ, even though their delivery bonus are the same. The acceptance probability of order A (over 95%) is much higher than that of order B (approximately 45%) when the bonus is 0. Given the bonus of 2 RMB, the increment of acceptance probability of order A is 0.01 whereas order B is 0.38. Therefore, the motivation is to increase the total acceptance probability by allocating bonus to order B.



Figure 5: The relationship between the bonus and acceptance probability.

2.2. Definition and Mathematical Model

The set N consists of the order placed within a week/month. The set T means the bonus allocation stage set, i.e., the bonus allocated to an order can be changed at most $|T|$ times. The decision variable $c_{i,t}$ is the delivery bonus of the order $i \in N$ at allocation stage $t \in T$. The upper bound of the delivery bonus is C_i^u . The function $p_{i,t}(c_{i,t})$ denotes the acceptance probability model of the order $i \in N$ if it is transited to allocation stage $t \in T$. The total delivery bonus cost should be within the given budget B .

The MSBA is modeled as follows:

$$\max \sum_{i \in N} \sum_{t \in T} \left(\prod_{t_0=1}^{t-1} (1 - p_{i,t_0}(c_{i,t_0})) \right) p_{i,t}(c_{i,t}) \quad (3)$$

$$\text{s.t.} \sum_{i \in N} \sum_{t \in T} \left(\prod_{t_0=1}^{t-1} (1 - p_{i,t_0}(c_{i,t_0})) \right) p_{i,t}(c_{i,t}) c_{i,t} \leq B \quad (4)$$

$$0 \leq c_{i,t} \leq C_i^u \quad \forall i \in N, t \in T \quad (5)$$

The objective function (3) maximizes the total acceptance probability, i.e., the expected value of the accepted order quantity. Constraint (4) indicates that the expected value of the total delivery bonus cost should be within the given budget B . Constraints (5) restrain the delivery bonus $c_{i,t}$ within the given upper bound.

3. Algorithm

In reality, we need to make a bonus allocation decision in real time and control the total bonus within a weekly/monthly budget. However, we cannot obtain all information regarding orders during a week/month. Therefore, controlling the total bonus within a budget and making full use of the budget are the keys to solving this problem. In this study, the offline and online calculation is combined to solve it. First, the LDDP algorithm is developed to solve the model presented in Section 2 offline, based on the historical data set. This algorithm is able to calculate the empirical Lagrangian multiplier λ_t^* for the allocation stage t . Then, λ_t^* is used to make real-time bonus allocation decisions online. The framework of our bonus allocation approach is depicted in Figure 6.

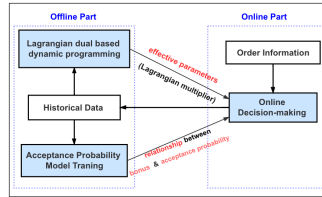


Figure 6: Our multistage bonus allocation framework.

3.1. Larangian dual-based Dynamic Programming

Since the mathematical model in Section 2.2 is non-linear and non-convex optimization model, it is intractable in the industry with huge-scale data in practice. Therefore, we propose a dynamic programming algorithm based on Lagrangian dual theory to solve it. In addition, to find out the recursion of the dynamic programming process, a dimensionality reduction technique is employed.

Let $N_{\tilde{t}}$ be the remaining order set transited to allocation stage \tilde{t} , if there is no budget posed on stage 1 to $\tilde{t} - 1$. Thinking of the right-hand side of constraint (4) \tilde{B} taking value 0, 1, 2,...,B as "state", and the

allocation stage subset $\{\tilde{t}, \tilde{t} + 1, \dots, |T|\}$ represented by \tilde{t} as the "stage" (Note that this "stage" is used to define a sub-problem of dynamic programming, which is different from the allocation stage mentioned before), leads us to define the subproblem $P_{\tilde{t}}(\tilde{B})$ and the optimal value function $G_{\tilde{t}}(\tilde{B})$ as follows:

$$G_{\tilde{t}}(\tilde{B}) = \max \sum_{i \in N_{\tilde{t}}} \sum_{t=\tilde{t}}^{|T|} \left(\prod_{t_0=\tilde{t}}^{t-1} (1 - p_{i,t_0}(c_{i,t_0})) \right) p_{i,t}(c_{i,t}) \quad (6)$$

$$\text{s.t. } \sum_{i \in N_{\tilde{t}}} \sum_{t=\tilde{t}}^{|T|} \left(\prod_{t_0=\tilde{t}}^{t-1} (1 - p_{i,t_0}(c_{i,t_0})) \right) p_{i,t}(c_{i,t}) c_{i,t} \leq \tilde{B} \quad (7)$$

$$0 \leq c_{i,t} \leq C_i^u, \forall i \in N_{\tilde{t}}, t = \tilde{t}, \tilde{t} + 1, \dots, |T| \quad (8)$$

Then, $z = G_1(B)$ gives us the optimal value of the MSBA[18].

To define a recursion that allows us to calculate $G_{\tilde{t}}(\tilde{B})$ in terms of $G_s(\tilde{B}_0)$ for $s > \tilde{t}$ and $\tilde{B}_0 \leq \tilde{B}$, we need to define a single-stage bonus allocation problem with the objective value function $g_{\tilde{t}}(\tilde{B})$, where budget \tilde{B} is only spent on a single stage \tilde{t} . It indicates the optimal expected value of the accepted order quantity if budget \tilde{B} is spent on a single stage \tilde{t} . Function $g_{\tilde{t}}(\tilde{B})$ is expressed as follows:

$$g_{\tilde{t}}(\tilde{B}) = \max \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}}(c_{i,\tilde{t}}) \quad (9)$$

$$\text{s.t. } \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}}(c_{i,\tilde{t}}) c_{i,\tilde{t}} \leq \tilde{B} \quad (10)$$

$$0 \leq c_{i,\tilde{t}} \leq C_i^u \quad \forall i \in N_{\tilde{t}} \quad (11)$$

By solving the single-stage bonus allocation problem, we can attain the accepted probability vector $\mathbf{p}_{\tilde{t}}(k) = [p_{1,\tilde{t}}(k), p_{2,\tilde{t}}(k), \dots, p_{|N_{\tilde{t}},\tilde{t}}(k)]$.

Then, $G_{\tilde{t}}(\tilde{B})$ can be expressed as follows:

$$G_{\tilde{t}}(\tilde{B}) = \max_{k=0,1,\dots,\tilde{B}} \{g_{\tilde{t}}(k) + G'_{\tilde{t}+1}(\tilde{B} - k)\} \quad (12)$$

$G'_{\tilde{t}+1}(\tilde{B} - k)$ indicates the optimal value function with budget of $\tilde{B} - k$ spent on allocation stage set $\{\tilde{t} + 1, \dots, |T|\}$, under the condition that budget of k is spent on single allocation stage \tilde{t} . $G'_{\tilde{t}+1}(\tilde{B} - k)$ can be formulated as follows (Let x_i be $\sum_{t=\tilde{t}+1}^{|T|} \left(\prod_{t_0=\tilde{t}}^{t-1} (1 - p_{i,t_0}(c_{i,t_0})) \right) p_{i,t}(c_{i,t})$ in the following expressions for short.):

$$G'_{\tilde{t}+1}(\tilde{B} - k) = \max \sum_{i \in N_{\tilde{t}+1}} (1 - p_{i,\tilde{t}}(k)) x_i \quad (13)$$

$$\text{s.t. } \sum_{i \in N_{\tilde{t}+1}} (1 - p_{i,\tilde{t}}(k)) x_i c_{i,t} \leq \tilde{B} - k \quad (14)$$

$$0 \leq c_{i,t} \leq C_i^u, \forall i \in N_{\tilde{t}+1}, t = \tilde{t} + 1, \dots, |T| \quad (15)$$

3.1.1. Dimensionality Reduction

As we can see, both objective (13) and constraint (14) are the inner product of vector $\mathbf{1} - \mathbf{p}_i(k)$ and the original objective and constraint of sub-problem $P_{\tilde{t}+1}(\tilde{B})$ with optimal value $G_{\tilde{t}+1}(\tilde{B})$. So, there is no recursive relationship between $G_{\tilde{t}}(\tilde{B})$ and $G_{\tilde{t}+1}(\tilde{B})$. To make the model tractable, the recursion can be attained by reducing the number of dimension of the domain of acceptance probability vector $\mathbf{p}_i(k)$. Suppose we reduce the number dimension of the domain of $\mathbf{p}_i(k)$ to m-dimensional and the basis of the new m-dimensional domain is $\{\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^m\}$, then $G'_{\tilde{t}+1}(\tilde{B} - k)$ can be formulated as follows:

$$G'_{\tilde{t}+1}(\tilde{B} - k) = \max \sum_{m_0=1}^m \sum_{i \in N_{\tilde{t}+1}} w^{m_0} (1 - q_i^{m_0}) x_i \quad (16)$$

$$\text{s.t.} \quad \sum_{m_0=1}^m \sum_{i \in N_{\tilde{t}+1}} w^{m_0} (1 - q_i^{m_0}) x_i c_{i,t} \leq \tilde{B} - k \quad (17)$$

$$0 \leq c_{i,t} \leq C_i^u, \forall i \in N_{\tilde{t}+1}, t = \tilde{t} + 1, \dots, |T| \quad (18)$$

To satisfy the computational requirement of industry in practice, we reduce the number of dimension the domain of $\mathbf{p}_i(k)$ to 1-dimensional in our work. In the simplest way, we choose the average value of $\mathbf{1} - \mathbf{p}_i(k)$ multiplied with unit vector, i.e., $\frac{\sum_{i \in N_{\tilde{t}+1}} (1 - p_{i,i}(k))}{|N_{\tilde{t}+1}|} [1, 1, \dots, 1]^T$. Meanwhile, $\sum_{i \in N_{\tilde{t}}} p_{i,i}(k) = g_t(k)$ holds according to the definition of $\mathbf{p}_i(k)$. Therefore, $G'_{\tilde{t}+1}(\tilde{B} - k)$ is formulated as follows:

$$G'_{\tilde{t}+1}(\tilde{B} - k) = \max \frac{|N_{\tilde{t}+1}| - g_t(k)}{|N_{\tilde{t}+1}|} x_i \quad (19)$$

$$\text{s.t.} \quad \sum_{i \in N_{\tilde{t}+1}} x_i c_{i,t} \leq \frac{|N_{\tilde{t}+1}|}{|N_{\tilde{t}+1}| - g_t(k)} \tilde{B} - k \quad (20)$$

$$0 \leq c_{i,t} \leq C_i^u, \forall i \in N_{\tilde{t}+1}, t = \tilde{t} + 1, \dots, |T| \quad (21)$$

3.1.2. Recursion to Dynamic Programming

In summary, the expression (12) can be reformulated as the recursion as follows:

$$G_{\tilde{t}}(\tilde{B}) = \max_{k=0,1,\dots,\tilde{B}} \left\{ g_{\tilde{t}}(k) + \frac{|N'_{\tilde{t}+1}|}{|N_{\tilde{t}+1}|} * G_{\tilde{t}+1}[(\tilde{B} - k) \frac{|N_{\tilde{t}+1}|}{|N'_{\tilde{t}+1}|}] \right\} \quad (22)$$

where $|N'_{\tilde{t}+1}| = |N_{\tilde{t}+1}| - g_{\tilde{t}}(k)$. Starting the recursion with $G_{|T|}(\tilde{B}) = g_{|T|}(\tilde{B})$ for $\tilde{B} \geq 0$, we use the recursion (22) to successively calculate $G_{|T|-1}, G_{|T|-2}, \dots, G_1$ for all integral values of \tilde{B} from 0 to B. An overview of the dynamic programming algorithm is presented in Algorithm 1. In this process, we must solve a single-stage bonus allocation problem with objective value of $g_{\tilde{t}}(\tilde{B})$ for all $\tilde{t} \in T$ and $\tilde{B} \in \{0, 1, 2, \dots, B\}$.

3.1.3. Solve Single-stage Bonus Allocation Problem Based on Lagrangian Dual Theory

As defined by functions (9)-(11), both the objective function and constraint functions are non-convex with respect to \mathbf{c} . Therefore, solving the problem directly is very difficult. Inspired by [23, 5, 11], we

reformulate the problem into an equivalent convex optimization problem. Because $p_{i,t}(c_{i,t}) = \frac{1}{1+e^{\alpha_{i,t}c_{i,t}+\beta_{i,t}}}$, we have

$$\begin{aligned} c_{i,t}(p_{i,t}) &= -\frac{\beta_{i,t}}{\alpha_{i,t}} + \frac{1}{\alpha_{i,t}}(\ln(1-p_{i,t}) - \ln(p_{i,t})) \\ p_{i,t} &\in (0, 1), \forall i \in N, t \in T \end{aligned} \quad (23)$$

Consequently, the constraint function (10) is rewritten as a function of $p_{i,\tilde{t}}$:

$$f_{\tilde{t}}(\mathbf{p}) = \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}} c_{i,\tilde{t}}(p_{i,\tilde{t}}) \quad (24)$$

Referring to [23, 5, 11], $f_{\tilde{t}}(\mathbf{p})$ is a convex function in \mathbf{p} for $\forall \tilde{t} \in T$.

Therefore, the original problem (9)-(11) is reformulated as follows:

$$-g_{\tilde{t}}(\tilde{B}) = \min - \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}} \quad (25)$$

$$\text{s.t. } \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}} c_{i,\tilde{t}}(p_{i,\tilde{t}}) \leq \tilde{B} \quad (26)$$

$$P_i^l \leq p_{i,\tilde{t}} \leq P_i^u \quad \forall i \in N_{\tilde{t}} \quad (27)$$

Clearly, this is a differentiable convex optimization problem with respect to \mathbf{p} and Slater's condition holds easily. Therefore, the duality gap between the primal and dual objective value is equal to 0[1].

By introducing a dual variable λ , the Lagrangian relaxation function of the single-stage bonus allocation problem is represented as follows:

$$L(\mathbf{p}, \lambda) = - \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}} + \lambda \left[\sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}} c_{i,\tilde{t}}(p_{i,\tilde{t}}) - \tilde{B} \right] \quad (28)$$

where $P_i^l \leq p_{i,\tilde{t}} \leq P_i^u, \forall i \in N, \lambda \in \mathbb{R}_+$. Correspondingly, the Lagrangian dual problem is formulated as:

$$\max_{\lambda \in \mathbb{R}_+} \min_{P_i^l \leq p_{i,\tilde{t}} \leq P_i^u} L(\mathbf{p}, \lambda) \quad (29)$$

As mentioned before, the primal optimal value $-g_{\tilde{t}}(\tilde{B})$ is equal to the dual optimal value owing to the zero duality gap. By introducing a bisection algorithm represented in Algorithm 2, we can obtain the optimal solution $\lambda_{\tilde{t}}(\tilde{B})$ and the optimal value $g_{\tilde{t}}(\tilde{B})$.

3.2. Online Algorithm

Given the multiplier $\lambda_{\tilde{t}}^*$ attained by the LDDP algorithm for stage $\tilde{t} \in T$, the online problem is written as:

$$\min_{0 \leq c_{i,\tilde{t}} \leq C_i^u} \left\{ - \sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}}(c_{i,\tilde{t}}) + \lambda_{\tilde{t}}^* \left[\sum_{i \in N_{\tilde{t}}} p_{i,\tilde{t}}(c_{i,\tilde{t}}) c_{i,\tilde{t}} \right] \right\} \quad (30)$$

which is equivalent to:

$$\min_{0 \leq c_{i,\tilde{t}} \leq C_i^u} \left\{ \sum_{i \in N_{\tilde{t}}} (\lambda_{\tilde{t}}^* p_{i,\tilde{t}}(c_{i,\tilde{t}}) c_{i,\tilde{t}} - p_{i,\tilde{t}}(c_{i,\tilde{t}})) \right\} \quad (31)$$

The above problem is reformulated as the summation of $|N_{\tilde{t}}|$ separable minimizing problems as follows:

$$\sum_{i \in N_{\tilde{t}}} \min_{0 \leq c_{i,\tilde{t}} \leq C_i^u} \{ (\lambda_{\tilde{t}}^* p_{i,\tilde{t}}(c_{i,\tilde{t}}) c_{i,\tilde{t}} - p_{i,\tilde{t}}(c_{i,\tilde{t}})) \} \quad (32)$$

Therefore, to determine the delivery bonus for a specific order, only a one-dimensional problem needs to be solved. An optimal solution for each separated problem can easily be obtained online. The optimal delivery bonus of a specific order $i \in N$ is expressed as follows:

$$c_{i,\tilde{t}}^* = \arg \min_{0 \leq c_{i,\tilde{t}} \leq C_i^u} \lambda_{\tilde{t}}^* p_{i,\tilde{t}}(c_{i,\tilde{t}}) c_{i,\tilde{t}} - p_{i,\tilde{t}}(c_{i,\tilde{t}}) \quad (33)$$

Because the bonus of the order i is ranging from 0 to C_i^u , and the number of potential bonus is limited, the optimization problem (33) is easily solved by enumerating the potential bonus. Other numerical optimization methods are applicable, but we are not going to further investigate them here. Note that multiple problems can be solved in parallel here because the primal problem is separated into multiple independent problems.

3.3. Computational Complexity

In this section, we discuss the computational complexity of our LDDP algorithm and online algorithm. For the offline part, we analyze the computational complexity of the bisection algorithm to solve the single-stage bonus allocation problem and the dynamic programming algorithm to solve the multistage bonus allocation problem. First, there are $\log_2(\frac{1}{\epsilon})$ iterations for the bisection search algorithm and the complexity of each iteration is $O(N)$. Therefore, the complexity of solving the single-stage bonus allocation problem is $O(\log_2(\frac{1}{\epsilon})N)$. Second, there are three steps for the dynamic programming algorithm. At lines 1-4 of Algorithm 1, we solve the single-stage bonus allocation problem for $|T|B$ times. For each iteration, we solve a single-stage bonus allocation problem with a complexity of $O(\log_2(\frac{1}{\epsilon})N)$. Therefore, the complexity of lines 1-4 is $O(\log_2(\frac{1}{\epsilon})N|T|B)$. At lines 8-21, we solve the subproblem $P_{\tilde{t}}(\tilde{B})$ recursively, and the complexity is $O(|T|B^2)$. At lines 23-26, the complexity of the backtracking process is $O(|T|)$. Overall, the total complexity of the LDDP algorithm is $O(\log_2(\frac{1}{\epsilon})N|T|B + |T|(B^2 + 1))$.

As for the online part, the potential delivery bonus number is $\frac{C_i^u}{\delta}$ for order i , where δ is the bonus allocation interval for enumeration. Because $\frac{C_i^u}{\delta}$ is a constant, the complexity of determining the delivery bonus of an order online is $O(1)$.

4. Experiments

To verify the effectiveness of this method, we present the results of both offline experiments and online A/B tests.

4.1. Performance of offline experiments

In this section, we conducted several offline experiments. The instances are generated on real-world data derived from the Meituan meal delivery platform. Without the loss of generality, we chose the instances of four cities named Lanzhou, Nanchang, Weihai, and Chengdu with different typical order sizes over one week. Specifically, the data sets contain 46,978, 71,427, 120,017, and 258,612 orders, respectively. The delivery bonus of each order is calculated within the upper bound which is set by the business manager. It is worth mentioning that the *budget* in the figures below indicates the *average budget* for each order, which makes the results comparable among instances with different numbers of orders. A typical order life cycle starts from the order placed time to order delivered time or order canceled time, and the number of allocation stages indicates the dynamic bonus allocation frequency. For example, if the order life cycle is 50 minutes and the number of allocation stages is 10, then the bonus allocation decision is determined every 5 minutes.

4.1.1. Performance of the multistage bonus allocation

In this section, we examine the performance of the multistage bonus allocation method considering eight allocation stages and about a six-minute time interval between two allocation stages. A comparison of the results with the unified bonus mechanism in which the same delivery bonus is allocated to each order, single-stage bonus allocation[23] and multistage bonus allocation is presented in Figure 7. As shown in the figure, compared to the cases without bonus allocation, using a multistage bonus allocation reduces the number of canceled orders by more than 60%. Moreover, it also shows that the multistage bonus allocation outperforms the other two approaches. More specifically, compared to single-stage bonus allocation and the unified bonus mechanism, the number of canceled orders derived from the multistage bonus allocation is around 20% and 40% lower, respectively. More importantly, for the instances with a larger number of orders, the multistage bonus allocation performs better whereas the unified bonus mechanism and single-stage bonus allocation perform worse.

Figure 8 illustrates the total bonus allocation for each allocation stage. We observe that the total bonus allocation increases at first and then decreases as the allocation stage extends. To determine the potential reasons, we investigate the distribution of the accepted order numbers on these allocation stages. As shown in Figure 9, most orders are accepted immediately at the first allocation stage, and the number of accepted orders decreases as the allocation stage extends. Recall from Section 2.1 that our motivation was to increase the total acceptance probabilities of all orders by allocating delivery bonus to "order B" instead of "order A" (Figure 5). Correspondingly, orders with high acceptance probability play the role of "order A".

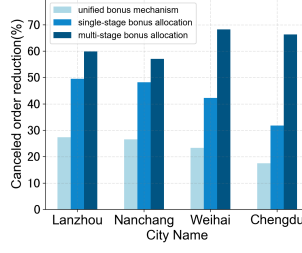


Figure 7: Comparison of the canceled order reduction with different approaches.

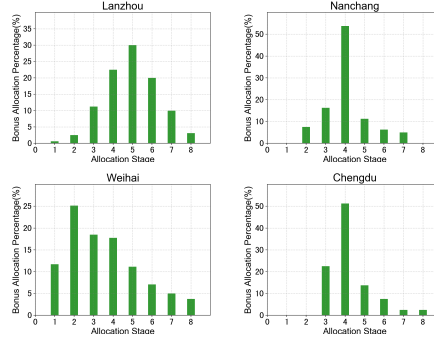


Figure 8: Total bonus allocation among all allocation stages.

Consequently, the total bonus allocated to the first stage is small because most instances of "order A" are accepted at the first stage, and the total allocated bonus increases at the early stages because the number of "order B" increases. Then, as the allocation stage extends, the total bonus decreases since the number of remaining orders decreases.

4.1.2. Performance of bonus allocation methods with different budget

In this section, we analyze the performance of bonus allocation methods with different budget. We compare the results derived by the multistage bonus allocation and the unified bonus mechanism when the budget changes. Figure 10 reflects the relationship between the budget and the number of accepted orders. Obviously, the multistage bonus allocation method outperforms the unified bonus mechanism. Particularly, with a small budget, the multistage bonus allocation method performs much better than the unified bonus mechanism.

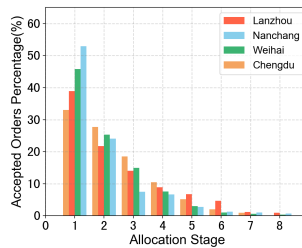


Figure 9: The number of accepted orders on multiple allocation stages.

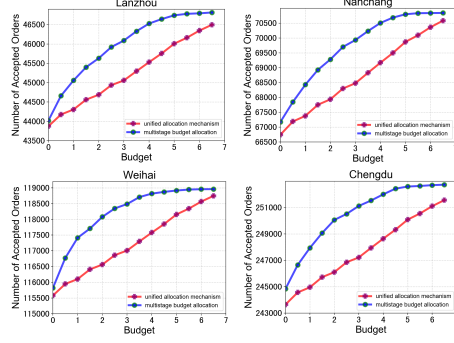


Figure 10: Comparison between multistage bonus allocation and unified bonus mechanism.

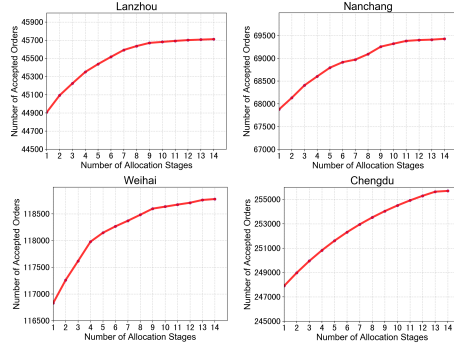


Figure 11: Impact of the number of the allocation stages on the number of accepted orders.

4.1.3. Impact of the number of the allocation stages

In this section, we investigate the impact of the number of the allocation stages on the accepted order quantity. As shown in Figure 11, the accepted order quantity increases when the number of allocation stages increases. However, the slope of the curve becomes smaller when the number of allocation stages is more than 10. Besides, it is detrimental to the crowdsourcing driver experiences that the delivery bonus of an order changes too frequently. Therefore, to balance the accepted order quantity and the crowdsourcing driver experience, it is reasonable to set the number of allocation stages no more than 10.

4.1.4. Performance of the acceptance probability model

As stated earlier, the acceptance probability model follows a logistic function (2). In this paper, $\alpha_{i,t}$ and $\beta_{i,t}$ are trained by DeepFM. We divide the training set into bonus batches with $c_{i,t} > 0$ and normal batches with $c_{i,t} = 0$. The hidden layers of $\alpha_{i,t}$ and $\beta_{i,t}$ are updated using bonus batches and normal batches, respectively. We test our method against two baselines:

- Two models: $\alpha_{i,t}$ and $\beta_{i,t}$ are trained by two models separately[23].
- One model: $\alpha_{i,t}$ and $\beta_{i,t}$ are trained simultaneously and share a common hidden layer, but the training set is not divided into two kinds batches.

In Table 1, we list the performance of two baseline models and the proposed model for acceptance probability forecasting. We test these models on orders across the country for 30 days. The result shows that our method outperforms the other two models in AUC for the entire test set, the bonus test set with $c_{i,t} > 0$, and the normal test set with $c_{i,t} = 0$.

4.2. Performance of online A/B tests

In this section, we introduce our online A/B tests on the Meituan meal delivery platform.

4.2.1. Experiment Settings

We implemented our A/B tests on three cities in China, covering 40 areas and 1,740,000 orders per day. The areas in each city were randomly categorized into experimental and control groups. We only used our method on the experimental group. The budget for both the experimental and control group is set to 0.2 RMB per order. To verify the effectiveness of our algorithm, we recorded the order information data of two weeks before and after the start of the experiment for both groups.

4.2.2. Implementation of online A/B tests

We employ the LDDP algorithm to calculate the Lagrangian multiplier for each allocation stage by the historical data set for 30 days before the experiment date. The empirical Lagrangian multipliers are updated every day. The online algorithm is triggered when an order is presented on the screen of the crowdsourcing driver. The delivery bonus of an order is calculated in milliseconds using the empirical Lagrangian multiplier.

4.2.3. Results

In reality, the order would be canceled by the platform if nobody accepts it for more than 50 minutes, which is detrimental to the interests of multiple stakeholders. Therefore, the most important KPI is the number of canceled orders. As shown in Table 2, using the proposed method, the number of canceled orders reduced by more than 30%. Moreover, the utilization of the bonus budget is another KPI that needs to be evaluated. Table 2 also measures bonus budget utilization, which is computed as the monthly budget divided by the given budget at the beginning of a month. It shows that using our approach, we can make the best use of budget with the budget utilization nearby 1.0. Furthermore, we can save more than 35% of the compensation paid to restaurants for food waste.

5. Related Work

Some studies have studied dynamic pricing for attended home delivery [8, 9, 19] where time slot pricing was implemented to dynamically affect customers' bookings using approximate dynamic programming. Reference [19] proposed approximating the opportunity cost by calculating the insertion cost of an incoming request based on the insertion heuristic algorithm. Klein et al.[8] improved the method of Yang et al.

by considering expected future demand, making delivery cost approximation more accurate and linking the latest accepted customer information. Koch et al. [9] combine and extend the methods in the previous paper by considering limited fleet size, more flexible customer choice, and dynamic vehicle routing with time windows. Ulmer [15] established a Markov decision process model for dynamic routing and same-day delivery pricing and presented an anticipatory pricing and routing policy method to solve it. Based on this work of Ulmer [15], Prokhorchuk et al. [13] considered a stochastic travel time to make the model more applicable. Some references discussed the researches related to the meal delivery problem [20, 16, 4], but few of them involved with the pricing and bonus allocation problem.

There are some studies on dynamic pricing for ride-hailing platforms. Some ride-hailing platforms use a dynamic pricing strategy called surge/prime pricing, in which the base fare is multiplied by a multiplier that is greater than one when the demand is high relative to supply [6]. Castillo et al. [3] proposed a steady-state model for dynamic pricing in ride-hailing applications and validated that dynamic pricing is particularly important for ride-hailing owing to the so-called wild "goose chase phenomenon". Zha et al. [21] built a spatial pricing model based on a discrete-time geometric matching framework in which a customer is matched to their closest available vehicle within a set radius. Tong et al. [14] proposed the global dynamic pricing (GDP) in spatial crowdsourcing in a ride-hailing platform context. It aims to propose dynamic pricing in multiple local markets with unknown demand, limited supply, or dependent supply. However, there are two differences in pricing between ride-hailing and meal delivery problems. First, dynamic pricing for ride-hailing determines the price presented to passengers spatio-temporally, but for meal delivery, delivery bonus is presented to drivers based on the order features. Second, the aim of dynamic pricing for ride-hailing is to maximize the platform revenue, whereas maximization of the total acceptance probability of orders within a budget is the aim of meal delivery platform.

Another related topic to dynamic bonus allocation for the meal delivery platform, which is called the real-time bidding (RTB) for advertisement displays. The RTB determines the bidding price for advertisers to display advertisement in real time. The aim of RTB is to maximize the KPIs (such as click-through rate, click-conversion rate (CTR/CVR) and so on) within a budget. Perlich et al. [12] first proposed a linear bidding function based on impression evaluation, which has many real-world applications. Later on, Zhang et al. [22] derive the non-linear relationship between the impression level evaluation and the optimal bid. Lee et al. [10] formulate the RTB as an online linear programming that optimizes the performance metrics while satisfying the so-called smooth delivery constraint. Cai et al. [2] formulate the bid decision process as a reinforcement learning problem and solves it by dynamic programming for a small-scale problem, and a neural network is used to generalize the solution to the large-scale problem. Wang et al. [17] optimized RTB without a budget using deep reinforcement learning, specifically DQN. Jin et al. [7] proposed a multi-agent reinforcement learning framework that treats all the advertisers equally. The main difference between RTB problem and our problem is that the meal delivery platform does not have competitors when they present the delivery bonus to crowdsourcing drivers that the advertiser has in the RTB problem.

6. Conclusion

In this paper, we study a multistage bonus allocation problem for a meal delivery platform. To solve this problem, we propose a framework including a semi-black-box acceptance probability model, an LDDP algorithm, and an online algorithm. The relationship between the bonus allocated to an order and its acceptance probability is forecast by the acceptance probability model. The offline results of the empirical Lagrangian multiplier for each stage calculated by the LDDP algorithm are used for online bonus allocation decision-making. Using this framework, an online decision can be made within milliseconds, because the computation complexity of the online algorithm is $O(1)$.

Both offline experiments and online A/B tests are implemented to verify the effectiveness of our method. The results of offline experiments show that compared to single-stage bonus allocation and unified bonus mechanism, the number of canceled orders derived from multistage bonus allocation is around 20% and 40% less, respectively. More importantly, for instances of a larger number of orders, our method performs better whereas the unified bonus mechanism and single-stage bonus allocation perform worse. Furthermore, this framework is easy to implement and is successfully applied to the Meituan meal delivery platform. The online A/B tests show that the number of canceled orders reduced by more than 30%.

For future work, we are interested in analyzing the approximation error brought by the dimensionality reduction technique in Section 3. In this paper, to ensure the efficiency of the algorithm, we reduce the dimension of the domain of acceptance probability vector to 1-dimensional. Furthermore, how to solve the problem efficiently if the domain of acceptance probability vector is reduced to k -dimensional ($k > 1$) is another topic worth exploring.

References

- [1] Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.
- [2] Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., Guo, D., 2017. Real-time bidding by reinforcement learning in display advertising, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 661–670.
- [3] Castillo, J.C., Knoepfle, D., Weyl, G., 2017. Surge pricing solves the wild goose chase, in: Proceedings of the 2017 ACM Conference on Economics and Computation, pp. 241–242.
- [4] Ding, X., Zhang, R., Mao, Z., Xing, K., Du, F., Liu, X., Wei, G., Yin, F., He, R., Sun, Z., 2020. Delivery scope: A new way of restaurant retrieval for on-demand food delivery service, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3026–3034.
- [5] Dong, L., Kouvelis, P., Tian, Z., 2009. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management* 11, 317–339.

- [6] Hall, J., Kendrick, C., Nosko, C., 2015. The effects of uber’s surge pricing: A case study. The University of Chicago Booth School of Business .
- [7] Jin, J., Song, C., Li, H., Gai, K., Wang, J., Zhang, W., 2018. Real-time bidding with multi-agent reinforcement learning in display advertising, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 2193–2201.
- [8] Klein, R., Mackert, J., Neugebauer, M., Steinhardt, C., 2018. A model-based approximation of opportunity cost for dynamic pricing in attended home delivery. *OR Spectrum* 40, 969–996.
- [9] Koch, S., Klein, R., 2020. Route-based approximate dynamic programming for dynamic pricing in attended home delivery. *European Journal of Operational Research* .
- [10] Lee, K.C., Jalali, A., Dasdan, A., 2013. Real time bid optimization with smooth budget delivery in online advertising, in: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, pp. 1–9.
- [11] Li, H., Huh, W.T., 2011. Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management* 13, 549–563.
- [12] Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., Provost, F., 2012. Bid optimizing and inventory scoring in targeted online advertising, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 804–812.
- [13] Prokhorchuk, A., Dauwels, J., Jaillet, P., 2019. Stochastic dynamic pricing for same-day delivery routing. *arXiv preprint arXiv:1912.02946* .
- [14] Tong, Y., Wang, L., Zhou, Z., Chen, L., Du, B., Ye, J., 2018. Dynamic pricing in spatial crowdsourcing: A matching-based approach, in: Proceedings of the 2018 International Conference on Management of Data, pp. 773–788.
- [15] Ulmer, M.W., 2020. Dynamic pricing and routing for same-day delivery. *Transportation Science* 54, 1016–1033.
- [16] Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N., 2021. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science* 55, 75–100.
- [17] Wang, Y., Liu, J., Liu, Y., Hao, J., He, Y., Hu, J., Yan, W.P., Li, M., 2017. Ladder: A human-level bidding agent for large-scale real-time online auctions. *arXiv preprint arXiv:1708.05565* .
- [18] Wolsey, L.A., 1998. Integer programming. volume 52. John Wiley & Sons.

- [19] Yang, X., Strauss, A.K., Currie, C.S., Eglese, R., 2016. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation science* 50, 473–488.
- [20] Yildiz, B., Savelsbergh, M., 2019. Provably high-quality solutions for the meal delivery routing problem. *Transportation Science* 53, 1372–1388.
- [21] Zha, L., Yin, Y., Xu, Z., 2018. Geometric matching and spatial pricing in ride-sourcing markets. *Transportation Research Part C: Emerging Technologies* 92, 58–75.
- [22] Zhang, W., Yuan, S., Wang, J., 2014. Optimal real-time bidding for display advertising, in: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1077–1086.
- [23] Zhao, K., Hua, J., Yan, L., Zhang, Q., Xu, H., Yang, C., 2019. A unified framework for marketing budget allocation, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1820–1830.

Algorithm 1: Lagrangian dual-based Dynamic Programming

Input: Allocation stage set T , order set $N_t, \forall t \in T$, total budget B

Output: empirical Lagrangian Multipliers $\lambda^*[[T]]$

```

    ▶ Solve the single-stage bonus allocation problems repeatedly and record
    the solutions in arrays  $g[[T]][B]$  and  $\lambda[[T]][B]$ 
1  forall  $\tilde{t} \in T$  do
2      while  $\tilde{B} \leq B$  do
3           $g[\tilde{t}][\tilde{B}], \lambda[\tilde{t}][\tilde{B}] \leftarrow BA(N_{\tilde{t}}, \tilde{B})$ 
4           $\tilde{B} \leftarrow \tilde{B} + 1$ 

    ▶ Solve the sub-problem  $P_{\tilde{t}}(\tilde{B})$  recursively and record the optimal solution
    in the array  $G[[T]][B]$ 
5   $G[[T]] \leftarrow g[[T]]$ 
6   $\tilde{t} \leftarrow |T| - 1$ 
7   $\tilde{B} \leftarrow 0$ 
8  while  $\tilde{t} \geq 1$  do
9      while  $\tilde{B} \leq B$  do
10          $G[\tilde{t}][\tilde{B}] \leftarrow G[\tilde{t} + 1][\tilde{B}]$ 
11          $k^* \leftarrow 0$ 
12         for  $k \leftarrow 0$  to  $\tilde{B}$  by 1 do
13              $N' \leftarrow |N_{\tilde{t}+1}| - g[\tilde{t}][k]$ 
14              $temp \leftarrow g[\tilde{t}][k] + \frac{N'}{|N_{\tilde{t}+1}|} G[\tilde{t} + 1][\lfloor \frac{|N_{\tilde{t}+1}|}{N'} (\tilde{B} - k) \rfloor]$ 
15             if  $G[\tilde{t}][\tilde{B}] < temp$  then
16                  $G[\tilde{t}][\tilde{B}] \leftarrow temp$ 
17                  $k^* \leftarrow k$ 
18          $a[\tilde{t}][\tilde{B}] \leftarrow k^*$  ▶  $a[] []$  is used to backtrack the optimal Lagrangian
        multiplier for each allocation stage
19          $\tilde{B} \leftarrow \tilde{B} + 1$ 
20      $\tilde{t} \leftarrow \tilde{t} - 1$ 

    ▶ Backtrack the optimal solution and record the optimal empirical
    Lagrangian multiplier for each stage into arrays  $\lambda^*[[T]]$ 
21  $B_0 \leftarrow B$ 
22 for  $t = 1$  to  $|T|$  by 1 do
23      $B^* \leftarrow a[t][B_0]$ 
24      $\lambda^*[t] \leftarrow \lambda[t][B^*]$ 
25      $B_0 \leftarrow B_0 - B^*$ 
26 return  $\lambda^*[[T]]$ 
```

Algorithm 2: Bisection Algorithm for a single-stage problem(BA($N_{\tilde{t}}, \tilde{B}$))

Input: Order set $N_{\tilde{t}}$, budget \tilde{B}
Output: $\lambda_t(\tilde{B})$ and $g_{\tilde{t}}(\tilde{B})$

```

1  $low \leftarrow 0$ 
2  $high \leftarrow M$   $\triangleright$   $M$  is a big number
3 while  $high - low > \epsilon$  do
4    $s \leftarrow 0$ 
5    $opt \leftarrow 0$ 
6    $mid \leftarrow \frac{high+low}{2}$ 
7   forall  $i \in N_{\tilde{t}}$  do
8      $p_{i,\tilde{t}}^* \leftarrow \arg \min_{p_i^l \leq p_{i,\tilde{t}} \leq p_i^u} -p_{i,\tilde{t}} + mid * p_{i,\tilde{t}} c_{i,\tilde{t}}(p_{i,\tilde{t}})$ 
9      $s \leftarrow s + p_{i,\tilde{t}}^* c_{i,\tilde{t}}(p_{i,\tilde{t}}^*)$ 
10     $opt \leftarrow opt - p_{i,\tilde{t}}^*$ 
11   if  $s - \tilde{B} \geq 0$  then
12      $low \leftarrow mid$ 
13   else
14      $high \leftarrow mid$ 
15 return  $\lambda_t(\tilde{B}) \leftarrow high, g_{\tilde{t}}(\tilde{B}) \leftarrow -opt$ 

```

Table 1: Performance of ML Methods on the acceptance probability model.

ML method	AUC	AUC (normal set)	AUC (bonus set)
Two models	0.792	0.816	0.784
One model	0.823	0.827	0.812
Our method*	0.828	0.842	0.812

Table 2: Performance of online A/B tests.

City Name	Canceled Order Reduction	Compensation Saving	Budget Utilization
Nanchang	60.82%	42.69%	0.97
Weihai	40.00%	43.12%	0.93
Zhuhai	29.83%	35.09%	1.09