WORKING PAPER SERIES

Dynamic Optimization in Peer-To-Peer Transportation with Acceptance Probability Approximation

Rosemonde Ausseil/Marlin W. Ulmer/Jennifer A. Pazour

Working Paper No. 8/2022

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

FACULTY OF ECONOMICS
AND MANAGEMENT

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
R. Ausseil, M. W. Ulmer and J. A. Pazour
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Dynamic Optimization in Peer-To-Peer Transportation with Acceptance Probability Approximation

Rosemonde Ausseil
David D. Reh School of Business, Clarkson University, Potsdam, NY, rausseil@clarkson.edu

Marlin W. Ulmer
Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany, marlin.ulmer@ovgu.de, Corresponding Author

Jennifer A. Pazour
Industrial and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, pazouj@rpi.edu

Crowdsourced transportation by independent suppliers (or drivers) is central to urban delivery and mobility platforms. While utilizing crowdsourced resources has several advantages, it comes with the challenge that suppliers are not bound to assignments made by the platforms. In practice, suppliers often decline offered service requests, e.g., due to the required travel detour, the expected tip, or the area a request is located. This leads to inconveniences for the platform (ineffective assignments), the corresponding customer (delayed service), and also the suppliers themselves (non-fitting assignment, less revenue). In this work, we show how approximating suppliers' acceptance behavior by analyzing their past decision making can alleviate these inconveniences. To this end, we propose a dynamic matching problem where suppliers' acceptances or rejections of offers are uncertain and depend on a variety of request attributes. Suppliers who accept an offered request from the platform are assigned and reenter the system after service looking for another offer. Suppliers declining an offer stay idle to wait for another offer, but leave after a limited time if no acceptable offer is made. Every supplier decision reveals partial information about the suppliers' acceptance behavior, and in this paper, we present a corresponding mathematical model and a solution approach that translates supplier responses into the probability of a specific supplier to accept a specific future offer and uses this information to optimize subsequent offering decisions. We show that our approach leads to overall more successful assignments, more revenue for the platform and most of the suppliers, and less waiting for the customers to be served. We also show that considering individual supplier behavior can lead to unfair treatment of more agreeable suppliers.

Key words: peer-to-peer transportation, dynamic matching, supplier-side choice, stochastic acceptance behavior, restaurant meal delivery

1. Introduction/Motivation

Crowdsourced transportation providers, such as Uber, Instacart, and Grubhub, have become household names. These peer-to-peer platforms provide critical transportation services, including rides to passengers and delivery of shopping items or restaurant meals for the challenging last mile segment of the supply chain. What all these platforms have in common is that they do not rely on their own drivers, but instead outsource transportation fulfillment to private individuals, who supply their time and vehicle to complete the service (in this work, we denote these individuals as *suppliers*). While such crowdsourcing of transportation services has many advantages for the platform, it comes with significant uncertainties in planning and operations (Savelsbergh and Ulmer 2022). Crowdsourced suppliers can decide when they work (Ulmer and Savelsbergh 2020), where they work (Auad et al. 2021), and what offered jobs they want to fulfill (Ausseil et al. 2022). All these uncertainties can severely reduce service quality and platform revenue, because if no supplier can be found in time, the requesting customer may leave the platform unserved and unhappy. It is therefore essential for a platform to acknowledge the uncertainty in suppliers' behaviors, derive predictions about them, and use these predictions in decision making to improve their operations.

In this paper, we focus on the critical platform decision of how to match suppliers with transportation requests, also known as order dispatching (Qin et al. 2020). One particularly challenging uncertainty to predict is whether a supplier will accept or reject a specific offered request. This decision is outside the control of the platform and depends on the supplier's utility value of the offered request. If the utility of the offered request is lower than what the supplier views as acceptable, the request gets rejected. The request's utility value depends on a variety of observable factors, such as the travel time required for service, the location of the requesting customer, or the expected tipping amount (Castillo et al. 2022), but also additional, unobservable factors hidden to the platform (and often even to the suppliers themselves, Train 2009, p.20 ff.). Furthermore, suppliers are very heterogeneous. Some of them are quite agreeable, accepting most of the offered requests regardless of their utility, e.g., because they are new to the platform, while others are rather selective, and only accept requests with high utility values (Cook et al. 2021). Thus, it may be valuable for a platform to account for the heterogeneity by offering them different requests. This may not only improve the service rate but it could also increase supplier satisfaction as they get more "acceptable" offers. Since supplier behavior is unknown to the platform (and often to the suppliers themselves), it can only be approximated based on the suppliers' reactions to offers in daily operations. This approximation is challenging for three reasons: First of all, an acceptance or a rejection does not allow a direct prediction of the minimum acceptable utility value of a supplier; it can only point towards a general direction. For example, if an offered request is rejected, the request's observable utility is likely below the supplier's minimum acceptance threshold; however,

it is not clear how far below. The same holds for acceptances. Second, the platform only sees if a supplier accepts or rejects a request, but not why. Is the offer rejected because the observable utility value is below the threshold, or is it due to some other unobservable factors? Third, the number of measurements for each supplier is very limited, especially since the suppliers leave the platform after a short time if no acceptable offers are made (Castro et al. 2020). All these challenges come on top of an already complex combinatorial dynamic matching problem where holistic offering decisions for sets of requests and suppliers have to be made in real-time.

Thus, for successful operations, it is critical for the platform to both dynamically approximate supplier behavior based on suppliers' reactions and dynamically optimize offering decisions based on the approximated acceptance probabilities viewed holistically across the entire set of suppliers and requests currently in the system. In this paper, we present a corresponding mathematical model and propose a policy addressing both in an integrated way. For the approximation, we propose carefully "encircling" a supplier's minimum utility acceptance threshold based on a supplier's previous reactions to offered requests. We then translate these reactions into a supplier-specific probability distribution that can be used to map a specific new request to an acceptance probability. For the optimization, we integrate the probabilities into an integer program maximizing the expected number of accepted offers in a state and solve it to optimality. Our approach is iterative, in that the observed reactions of the suppliers to the platform's offering decisions are then used in the next state to update the approximated probabilities, which are then used in the next offering decisions.

We test our policy in a comprehensive experimental study in the case of restaurant meal delivery. For a large set of instances, we compare our policy to a variety of benchmark approximation and optimization policies, as well as the perfect-information case. In addition to quantifying how our policy outperforms the benchmark policies, we also derive six main managerial insights:

1. Acknowledging that suppliers can reject offered requests and modeling this behavior with probabilities is essential for successful peer-to-peer operations.
2. Approximating supplier behavior probabilistically can substantially increase platform revenue, particularly if the suppliers are rather "picky".
3. Integrating and updating supplier behavior approximations into the request offering optimization problem does not only increase revenue for the platform, but it also leads to faster service for the requesting customers.
4. The platform should acknowledge and exploit that suppliers have heterogeneous willingness to participate.
5. Considering individual supplier behaviors when offering requests keeps suppliers happy and retained in the system. The platform needs to make sure there is enough work for all of them though.

6. Considering supplier heterogeneity can lead to unfair treatment of the more agreeable suppliers, as they are more likely to get offered the low-utility requests.

Our work makes a variety of contributions. As we show in the literature review in Section 2, we are among the first to approximate, update, and integrate supplier acceptance behavior in peer-to-peer transportation matching problems and in transportation optimization in general. In Section 3, we present a comprehensive general mathematical model reflecting the interplay of offering and approximating decisions. This model is likely transferable to a variety of application areas from peer-to-peer transportation with uncertain supplier behavior and even to more traditional transportation settings when company drivers may have preferences and behaviors uncertain to more traditional transportation companies. The same is true for the proposed, effective policy introduced in Section 4. We present a carefully crafted set of experiments in Section 5 with a particular focus on supplier behavior. This allows a focused and comprehensive analysis of our policy in Section 6, showing how a better understanding of suppliers' decision making impacts all stakeholders involved. We also use our model to provide a number of new insights. Based on our model, method, and experiments, in Section 7, we present a set of specific promising areas of future research.

2. Literature Review

In this section, we discuss the related literature, starting with an overview on matching in peer-to-peer transportation. We then discuss transportation and logistic problems where information is accumulated during the process (i.e., "learning").

2.1. Peer-to-Peer Transportation

Research on peer-to-peer or crowdsourced transportation is booming, see Agatz et al. (2012), Cleophas et al. (2019), Furuhata et al. (2013), Mourad et al. (2019), Rai et al. (2017), Savelsbergh and Ulmer (2022), Tafreshian et al. (2020), or Wang and Yang (2019) for recent surveys. While in practice supplier availability and behavior are uncertain to the platform, the majority of research assumes deterministic settings with full information to investigate the general potential and changes in optimization when using crowdsourced resources for delivery (see, e.g., Archetti et al. 2016, Behrend and Meisel 2018, Behrend et al. 2019, and Macrina et al. 2020).

There is work on matching and routing in crowdsourced transportation with respect to uncertainty that outsourcing to crowdsourced suppliers brings. Yet, this uncertainty usually manifests in the suppliers' availability, whereas the focus of our work considers uncertainty in suppliers' acceptances of jobs offered by the platform. Work on (anticipating) uncertain availability is, for example, presented by Chen et al. (2020), Skålnes et al. (2020), Dayarian and Savelsbergh (2020), Lei et al. (2020), Ulmer and Savelsbergh (2020), Nieto-Isaza et al. (2022), and Behrendt et al. (2022). All of these works assume that suppliers enter and leave the system when they want. The papers present

methods that anticipate the arrival and departure of suppliers in their assignment and routing decisions, but once arrived to the platform, the methods assume supplier acceptance behavior with certainty. In our problem, the arrival and departure of suppliers are also uncertain to the platform, but so is the additional uncertainty in whether suppliers will accept offered transportation jobs.

As in our work, some work on crowdsourced transportation assume that suppliers do not accept every offered job. However, they assume that the behavior of suppliers is known. For example, Arslan et al. (2019) assume suppliers accept the offer if the detour is below a known, but supplier-specific threshold. Only a small number of papers consider the uncertain acceptance behavior of suppliers. Yildiz and Savelsbergh (2019) assume the probability of acceptance depends on distance and expected compensation and tip. In a stylized setting, they calculate optimal service radii and compensation values. Gdowska et al. (2018) and Cao et al. (2020) assume the probability of acceptance depends on the compensation. Santini et al. (2022) assume acceptance probabilities depend on compensation, travel distance, and some additional attributes hidden from the platform. Mofidi and Pazour (2019), Horner et al. (2021) and Ausseil et al. (2022) assume probabilities depend on the required time to perform the service and the traveled distance. All of these works optimize for expected cost or revenue based on (assumed known) acceptance probabilities, deciding about offering or compensation. In our work, we optimize offering decisions based on the acceptance probabilities, similar to the aforementioned papers. However, in contrast to existing work, we do not assume these probabilities are given, but instead iteratively approximate the probabilities based on observed decisions. To the best of our knowledge, our work is the first to model the problem of sequentially approximating supplier acceptance behavior and using this information to optimize order dispatching in peer-to-peer transportation problems.

2.2. Learning in Transportation and Logistics

Our work uses observed information to approximate supplier behavior for better future decisions, i.e., "learning" from the past. There are different types of learning considered in the transportation and logistics literature. We differentiate between (1) learning by the workforce, (2) learning the value of decisions in the model, and (3) learning about the optimization model's parametrization.

The first set of literature considers problems where the workforce learns based on previous information. The learning process is usually assumed deterministic and follows a given functional form. For example, delivery drivers may get to know their customers better with every visit (see Ulmer et al. 2020 for a recent survey) or technicians may learn to perform certain tasks more effectively via repetition (Valeva et al. 2017, Chen et al. 2017, Jin et al. 2018, Bakker et al. 2021). In all those cases, employee learning leads to faster service, and the decision making models consider the impact of workforce learning when assigning a customer or task. In our research, the workforce does not learn, instead the platform learns about the behaviors of its workforce, i.e., the suppliers.

The second set of literature uses (reinforcement) learning to identify high quality decisions, e.g., via value function approximation (e.g., Ulmer et al. 2018, van Heeswijk et al. 2019, Kullman et al. 2022, see Soeffker et al. 2022 for a recent survey). In these papers, there is no learning within the model. Instead, offline approaches, typically repeated simulations, are performed to evaluate decisions with respect to their future value. Thus, the optimization model characteristics are fully known, and the learning is part of an offline heuristic method that feeds later into the online optimization model. Notably, in our case, offline learning cannot be applied as learning is based on the outcome of *online* decisions.

The third set of literature is closest to our research, in that methodologies are created and employed to identify specific, unknown parameterizations of the optimization model. This falls into the field of Optimal Learning (OL) (Powell and Ryzhov 2012). In OL, decisions carefully balance the exploitation of parameterizations-knowledge and the exploration of parameterizations for future use. To the best of our knowledge, there are only two papers related to OL in the transportation literature. Al-Kanj et al. (2016) face a problem of routing to repair a larger-scale power outage. Several areas of the city are without energy and vehicles are routed to identify (and repair) broken connections in the system. Besides optimizing the routing of the vehicles, additionally, there is the identification of the broken parts of the system ("information collection"). The latter can be seen as learning the model's parameterizations. As in our paper, the optimization decision is based on the current *belief* of the parameterizations; thus, no "suboptimal" decisions are selected to enforce explicit exploration of the model's parametrization. The only work explicitly exploring the parameterizations is provided by Huang et al. (2019) for an urban delivery problem. Initially, the routing cost of a specific number of vehicles in a city district is unknown. Over the decision periods, the costs are updated based on observations and used for optimization in the subsequent states. In that paper, the authors propose to explicitly test more costly setups to allow exploration. For a small instance with four districts, they show that this active OL reduces cost in the long run.

Similar to Al-Kanj et al. (2016) and Huang et al. (2019), we face uncertainty in the model's parameterizations. In our case, we are uncertain about the behavior of the suppliers. Like Al-Kanj et al. (2016), we subsequently update the belief about the parameterizations and use it for optimization. However, in contrast to the work by Al-Kanj et al. (2016), we do not observe the specific parameter information, but only a surrogate about the parameterizations of the suppliers' preferences via their acceptance/rejection decisions. In addition to focusing on a different type of parametization than Huang et al. (2019), we also differ by focusing on pure exploitation. Our focus on pure exploitation is motivated by our problem setting. First, in our problem, we only have a very limited number of trials to find a successful matching between suppliers and requests, since both leave the platform if they do not receive an acceptable offer. Research on dynamic matching with

"disengagement" of participants has shown that OL-exploration can actually be counterproductive (Bastani et al. 2021). Further, our problem requires complex and fast offering decisions for several suppliers and requests. In contrast to Huang et al. (2019), our decisions are interrelated (across a set of suppliers with uncertain behavior) and impact future states.

We finally note that our work is also related to the general field of preference learning outside of logistical settings, often applied in many areas where companies try to identify the participants' preferences, e.g., in pricing (Nambiar et al. 2019), personalized revenue management (Chen et al. 2022), or dating platforms (Cao and Zhang 2021). While our work shares the aspect of approximating preferences for more successful decisions, it differs substantially in the setting and complexity of decision making. In our work, we aim for an orchestrated combinatorial decision of matching several suppliers and customers at once. Further, the consequences of an unsuccessful offer are more severe as it impacts the entire operations, as well as suppliers' earnings and future participation.

3. Problem Definition

In this section, we begin with a problem description and an illustrative example, and then define the components of the mathematical model.

3.1. Problem Description

We consider a peer-to-peer platform dynamically matching customer requests for transportation services with individual crowdsourced drivers (from now on called "suppliers") over the course of the day. Spontaneous service requests arrive to the platform during a service period, and the requesting customers expect fulfillment of service a short time later. The fulfillment usually comprises a timely pickup of either a passenger, goods, or food, at one location and a dropoff at another location nearby. Simultaneously, crowdsourced suppliers spontaneously log into the app of the platform, indicating their willingness to receive offers of requests from the platform to provide transportation services with their own vehicles. As the suppliers are self-employed, they are not bound to accept the offers made by the platform. Suppliers are free to reject an "unacceptable" offer because their utility for the request is too small. For example, the utility of an offered request decreases when serving it requires long travel, leads to operating in an area difficult to drive and park in, or - as often observed in meal delivery - the expected tipping amount is small. While these attributes can be measured relatively well by the platform (Castillo et al. 2022), there may also be additional factors impacting a supplier's utility that are hidden to the platform. We assume that each supplier has a specific minimum utility acceptance threshold (the utility of the no-choice option). This threshold is unknown to the platform, and the supplier rejects all offered requests with utilities below it. If a request's utility is higher than the threshold, the supplier accepts it, starts working, and then usually reappears on the platform once the service is finished.

The role of the platform is to successfully match requests and suppliers. In equidistant time steps (e.g., every five minutes), the platform makes an offering decision based on the requests and suppliers currently in the system. After the response of the suppliers, the accepted offered requests are served by the corresponding suppliers, and those requests leave the system. A matched supplier leaves the system for the time duration of the service request, but then reappears after completion. All unassigned suppliers and requests stay in the system, however, only for a limited overall time. This results in the overall goal of the platform to maximize the expected number of successful assignments over the course of the day, especially because the platform must perform recourse actions for requests that have been in the system for "too long". For example, these requests may be served by an expensive taxi or are rejected without service, leading to customer inconvenience and lost revenue. If a supplier gets an unacceptable offer (or no offers at all), they stay in the system and wait for the next offer. However, if they do not receive an acceptable offer within a certain amount of time, they leave the platform for that day, which leads to lower service capacity options for the platform. As the platform does not know the suppliers' preferences (i.e., the minimum utility acceptance thresholds), it cannot be sure if a supplier will accept or reject an offer. However, as suppliers stay in the system for a while and reappear after a service, the platform can use their previous response behaviors to approximate their preferences and use the approximations for a more fitting offer later.

3.2. Example

To illustrate the problem's components and in preparation for the mathematical model, we describe a simplified example of our problem in Figure 1. The first box of the figure depicts an example state of the problem. The state shown here is in time step 8, in which three requests, A, B, and C, as well as two suppliers, 1 and 2, are currently in the system. For the purpose of presentation, in this example, we assume service is performed at the requests' locations (in our mathematical model, we consider pickup and delivery). The requests and the suppliers (depicted by red squares and blue circles, respectively) display heterogeneity in their geographical locations. Supplier 1 is currently located in the Southwest of the service area and supplier 2 in the Northeast. Specifically for this simplified example, we assume a request's utility only depends on the (Euclidean) travel distance from the respective supplier to the request. Thus, in this example, a supplier's minimum acceptance threshold limits how far the supplier is willing to go to service a request. A green shade surrounding a supplier's origin depicts the platform's current assumption of a supplier's acceptance region based on their approximated minimum utility acceptance threshold. The larger the approximated acceptance threshold is, the smaller is the respective maximum travel distance and consequently, the acceptance region. In this example, we model this via a point estimate

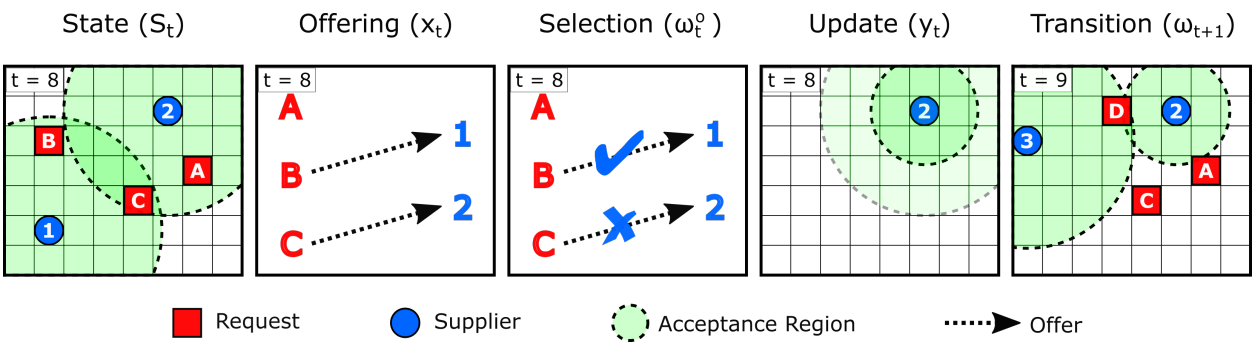

Figure 1 **Example for a state, offering decision and first stochastic information, update decision and stochastic information, and transition.**

for reasons of presentation (in model and method, we will rely on probability distributions). For example, requests A and C are within supplier 2's approximated acceptance region, so the platform assumes supplier 2 would accept either of those requests and would reject request B. Based on the (approximated) supplier thresholds and on the platform's objective to maximize the expected number of assigned requests, the platform offers request B to supplier 1 and request C to supplier 2, as depicted in the second box of Figure 1. Request A is not offered to any supplier, as a supplier can only be offered at most one request in a given time step. Supplier 1 accepts offered request B, i.e., the platform's assumption that supplier 1's utility for serving request B is above their acceptance threshold was correct. As request B is successfully assigned, request B leaves the platform. Supplier 1 starts serving request B and will reappear in the system after service. However, supplier 2 rejects their offer of request C, i.e, the actual minimum utility acceptance threshold of supplier 2 is either higher than assumed or potentially the unobservable attributes resulted in a lower than approximated utility for request B. Based on the rejection observation, the platform updates its knowledge on supplier 2's threshold in the next time step (which leads to a smaller acceptance region for supplier 2). In the next time step, the updated thresholds are considered, along with updated supplier and request sets; in the example, a new order D and a new or returning supplier 3 appear.

3.3. Mathematical Model

In this section, we formally define the mathematical model and its supporting notation. As the problem is relatively "rich", we start with a preparation of the model; then, we describe the components of the sequential decision process following the framework from Powell (2022). Notably, our model has two types of decisions, one for the offer and one for the approximation.

Preparation. In preparation for the model, we first provide global notation. Then, we discuss the particular characteristics of our problem – the suppliers, the uncertain utility values and (approximated) acceptance probabilities – and how they will be modeled.

Global Notation: We consider a time horizon with equidistant time steps $t = \{1, \ldots, T\}$. Each supplier s_j entering the system has a starting location l_{j0}. The maximum time without an assignment for suppliers is $t^s_{\max}$; afterwards, they leave the system. As a request r_i can represent a variety of services, in time t, we define $\delta_t(s_j, r_i)$ as the overall required time for s_j to serve request r_i. Further, l^r_i represents the location supplier s_j reappears in the system in case the supplier is successfully assigned to request r_i.

Suppliers: The set of suppliers differ from state to state as new suppliers enter the system, assigned suppliers reappear later, and other suppliers leave for good. To keep track of all the suppliers and to keep the indices for each supplier the same, we assume that, at each time step t, we have an ordered set of suppliers $s_t = (s_1, \ldots, s_j, \ldots, s_{J_t})$ with (random number) J_t, denoting the overall number of suppliers observed up to that point. Over the time steps, the set of suppliers interacting with the platform is subsequently revealed, with new suppliers appended to this set. For every supplier s_j, we carry information on their status in time t. The status is represented by two values: the availability time $\tau^s_{jt} \in T$ and the location l_{jt}. Time τ^s_{jt} can be in the past, i.e., the supplier idles, or in the future, i.e., the supplier is currently busy. For suppliers who left the system, $\tau^s_{jt} = \infty$ and $l_{jt} = -$. Furthermore, for every supplier, we carry information about their threshold approximation as discussed later in this section. For the ease of presentation, we do the same for the (random) number of overall requests at time t, $r_t = (r_1, \ldots, r_i, \ldots, r_{I_t})$ with I_t being a random variable. Like suppliers, requests also have availability times $\tau^r_{it} \leq t$ indicating when the request was issued. Value $\tau^r_{it} = \infty$ indicates that request r_i already left the system.

Utility: For a specific supplier s_j and a specific request r_i in a specific time step t, v_{ijt} denotes the utility value. This is unknown to the platform because only a part of the request's utility can be quantified by the function $f(a_{ijt})$ via a set of c attributes $a_{ijt} = \{a_{1ijt}, \ldots, a^r_{cijt}\}$. Some attributes may be independent of the supplier (e.g., the expected tipping amount), whereas others may be a function of the supplier and the request (e.g., the time to complete the service request). Another part of the utility is hidden from the platform, represented by the (potentially negative) value $\epsilon_{ij} \in \mathbb{R}$. The overall utility v_{ijt} is then the combination of both known and unknown attributes: $v_{ijt} = f(a_{ijt}) + \epsilon_{ij}$.

Furthermore, each supplier s_j has a stationary utility acceptance threshold v_{0j}, i.e., the minimum utility a supplier will accept a request, which is unknown to the platform, but stays constant over the time periods. A supplier s_j rejects a request r_i if $v_{ijt} < v_{0j}$. Even if the platform knew the exact utility of a request for a supplier in time t, the supplier's threshold does not become known when the supplier accepts or rejects the request. Thus, at every time step t, the platform operates with approximated acceptance probability functions $\hat{p}_{jt}$ for each supplier s_j. These probability functions allow us to map a request r_i to value $\hat{p}_{jt}(r_i) \in [0, 1]$, which denotes the approximated probability

that request r_i is accepted by supplier s_j in time t. These probabilities are captured in the *belief state* and are updated over time based on supplier s_j's rejection/acceptance behavior. How these probabilities are updated is modeled as a decision in our mathematical model.

States. In each time step t, state S_t is defined based on the set of present requests and suppliers and their associated attributes and parameters. For ease of presentation, we denote the sets of available suppliers as $s_t^a = \{s_j \in s_t : \tau_{jt}^s \leq t\}$ and open requests as $r_t^o = \{r_i \in r_t : \tau_{it}^r \leq t\}$. The suppliers $s_j \in s_t, j \leq J_t$ are represented by the time and location they became or will become available again, $\tau_t^s = (\tau_{1t}^s, \ldots, \tau_{J_t t}^s)$ and $l_t^s = (l_{1t}^s, \ldots, l_{J_t t}^s)$. The requests are represented by their time of request vector τ_t^r. Further, a state contains the attribute value matrix $a_t = (a_{ijt})_{i \leq I_t, j \leq J_t}$, and location vector, $l_i^r, i \leq I_t$ which denotes the geographical location a supplier will reappear at when serving request r_i. A state also contains the belief state, e.g., the current approximated probability functions $\hat{p}_{jt}(r_i)$ that supplier s_j will accept offered request r_i.

First Decision. Each time step t begins with the platform making decisions x_t, which represent the offers made to the suppliers. A decision value is $x_{ijt} = 1$ if platform offers request r_i to supplier s_j; $x_{ijt} = 0$, otherwise. The platform can only offer open requests to suppliers currently available in the system; it can offer an open request to at most one available supplier, and at most one open request can be offered to each available supplier:

$$\sum_{s_j \in s_t^a} x_{ijt} \leq 1 \qquad \forall r_i \in r_t^o \tag{1}$$

$$\sum_{r_i \in r_t^o} x_{ijt} \leq 1 \qquad \forall s_j \in s_t^a \tag{2}$$

The reward $R(S_t, x_t)$ of a decision x_t in state S_t is the expected number of accepted offers. In the next step, the first stochastic transition occurs.

First Stochastic Transition. After the platform decides the request offers to the suppliers with decision x_t, the platform observes the suppliers' selections ω_t^x with $\omega_{ijt}^x \leq x_{ijt}$. A selection value is $\omega_{ijt}^x = 1$ if supplier s_j accepts offered request r_i; otherwise, $\omega_{ijt}^x = 0$ if supplier s_j rejects offered request r_i. The observed reward is

$$\sum_{s_j \in s_t^a, r_i \in r_t^o} \omega_{ijt}^x x_{ijt}.$$

The availability times $\tau_t^{s,x}$ and locations $l_t^{s,x}$ of suppliers are updated in set s_t^x as follows. If a supplier s_j accepts an offered request r_i, the availability time and location of the supplier are set to $\tau_{jt}^{s,x} = t + \delta_t(r_i, s_j)$ and $l_{jt}^{s,x} = l_i^r$. For an unassigned supplier s_j, if the maximum waiting time is reached, i.e., $t = \tau_{jt}^s + t_{\max}^s$, this supplier leaves the system and $\tau_{jt}^{s,x} = \infty$, and $l_{jt}^{s,x} = -$. The set r_t^x is

updated similarly. For a request r_i, if the maximum waiting time is reached ($t = \tau^r_{it} + t^r_{\max}$) or the request is assigned ($\omega^x_{ijt} = 1$ for one supplier s_j), the request leaves the system and $\tau^{r,x}_{it} = \infty$. Else, it stays in the system for the next state in $t+1$.

Second State and Decision. After the observation of the suppliers' acceptance/rejection behavior, the platform observes a second state and has a second decision to make. The second state, $S^{x,\omega}_t$, contains the now updated information about suppliers s^x_t and requests r^x_t, the previous approximated probability function $\hat{p}_t$, and the observed selections of the suppliers, ω^x_t. The second decision y_t updates the belief state, i.e., the approximated probability functions $\hat{p}_{jt}$ are updated to $\hat{p}^y_{jt}$ for every supplier s_j based on the observed behavior of the suppliers. There is no reward associated with this update decision.

Second Stochastic Transition. The transition $\omega_{t+1} = ((s_{J_t+1}, \dots s_{J_{t+1}}), (r_{I_t+1}, \dots r_{I_{t+1}}))$ accounts for the arrival of $I_{t+1} - I_t$ new requests and $J_{t+1} - J_t$ new suppliers in the system between the end of state S_t and the start of the next state S_{t+1}. The corresponding values are set to $\tau^s_{jt+1} = t+1, l^s_{jt+1} = l_{j0} \forall J_t < j \leq J_{t+1}$ and $\tau^r_{it+1} = t+1 \forall I_t < i \leq I_{t+1}$. The new sets s_{t+1} and r_{t+1} are created by augmenting the sets s^x_t and r^x_t by the new $J_{t+1} - J_t$ suppliers and $I_{t+1} - I_t$ new requests.

Objective Function. A solution to the problem is a two-part decision policy $\pi = (\pi^x, \pi^y)$ with one decision policy for the offer, π^x, and one for the update, π^y. A policy π maps each state S_t to an offering decision $\mathcal{X}^{\pi^x}(S_t)$ and each state $S^{x,\omega}_t$ to an approximation decision $\mathcal{X}^{\pi^y}(S^{x,\omega}_t)$. The objective is to find a policy π^* maximizing the expected number of successful assignments:

$$\pi^* = \underset{\pi=(\pi^x,\pi^y)\in\Pi}{\arg\max}\ \mathbb{E}\left[\sum_{t=0}^{T} R(S_t, \mathcal{X}^{\pi^x}(S_t))|\pi^y, S_0\right]. \tag{3}$$

The optimal policy is the policy that maximizes the expected overall reward, the number of successful assignments, when starting in the initial state S_0 (when there are no requests or suppliers yet, i.e., $I_0 = J_0 = 0$), and applying the offering decisions $\mathcal{X}^{\pi^x}$ of policy π^x and approximation decisions of the update policy π^y throughout the entire problem horizon.

4. Solution Methods

In this section we present our solution method to solve the mathematical model of Section 3.3. We first give a general overview and motivation to highlight how we address the two steps of optimization and approximation. We then present our method in detail. We further describe a selection of benchmark policies, differing in one or two of the steps.

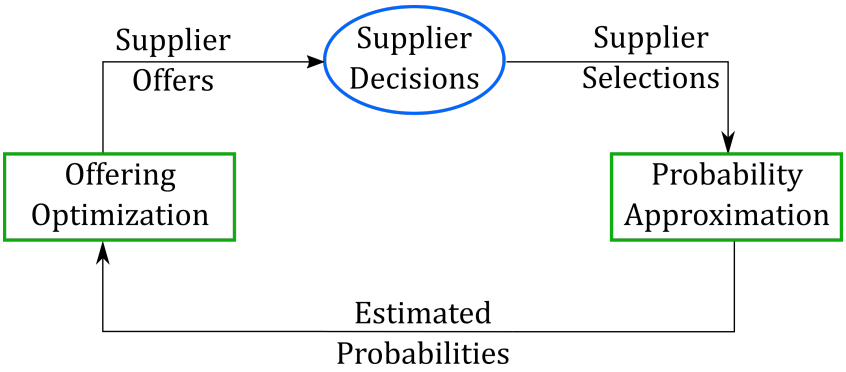

Figure 2 **A Process Diagram of the Iterative Solution Method. A green box means it is a platform decision, and a blue oval means it is a supplier decision.**

4.1. Overview

As depicted in Figure 2, decision making has two main components: offering optimization and probability approximation. Given the current time step's probability approximations, the platform performs an optimization to output suppliers' offers. Each supplier then makes the decision of either accepting or rejecting their offered request, and this process of supplier decisions is outside the control of the platform. After the platform receives the suppliers' selections, the platform performs a probability approximation step to update suppliers' estimated acceptance thresholds, i.e., acceptance probability functions. This process continues in an iterative fashion, with these functions being then used as inputs to the subsequent optimization step in the next time step, in which new suppliers and requests may appear.

Designing effective methodologies for both steps is challenging. In the offering optimization, instant and holistic considerations of suppliers and requests are required, while taking into account that a one-to-one match is needed, as well as the state details and the individual attributes of each request and individual (approximated) preferences of each supplier. In the probability approximation, the platform observes preferences only indirectly via a supplier's rejection or acceptance decision. Furthermore, the approximation is complicated by the unobservable utility component. We address these challenges as follows:

Optimization: We focus on how to make offering decisions that optimizes the expected number of successful matches, i.e., the expected reward of the decision. This rather short-term objective was selected for two reasons. First, as suppliers and requests only stay in the system for a very limited time, exploratory learning reduces the small number of offering opportunities even further, a phenomenon also observed in other domains with participant disengagement (Bastani et al. 2021). Because of supplier and request impatience that leads to disengagement, not offering anything to a supplier now in the hope for better, future assignment opportunities is also not warranted. Second, longer-term anticipation is difficult in crowdsourced transportation problems due to the manifold

and disruptive uncertainties of the problem. As shown in Ausseil et al. (2022), the uncertainty in supplier selection is very disruptive, and, for a setting similar to ours, anticipating beyond the current time period did not yield any advantage.

Approximation: We approximate each supplier's acceptance probabilities via two threshold values: an upper border value and a lower border value. These upper and lower border values are used to create the platform's assumption about acceptance boundaries (to be fed into the optimization step), which assumes that all requests are accepted with certainty if their observable utility lies above the upper border value, and that all requests are rejected with certainty if their observable utility lies below the lower border value. Everything in between is assumed with a certain probability that monotonically decreases with decreasing observable utility value. Then based on observed acceptance and rejection decisions for each supplier, we encircle their minimum acceptance threshold by narrowing the gap between the upper and lower border values. A rejection of a request may increase the lower border value, and an acceptance may decrease the upper border value. As suppliers have an unobservable utility component, a hard update may not be advisable. Therefore, we carefully balance the new observations with the previous approximation.

4.2. Main Method: Objective-Based Optimization with Parameter-Based Approximation

Our main method (called *OB-RA*) consists of an objective-based (OB) optimization step with a parameter-based probability approximation (RA) step based on supplier rejections (R) and acceptances (A). The whole process is described in Algorithm 1, in which we start by initializing, for each supplier, their estimated threshold values; then, in the optimization step we solve an integer program, as described in Section 4.2.1, given those estimated thresholds and state information. Next, the suppliers make their selections of the offers resulting from the optimization step. Finally, the supplier thresholds are updated via the RA approximation, as described in Section 4.2.2, based on those supplier decisions. At which point, the process repeats itself in the next state in the following time step, with the arrivals and departures of requests and suppliers.

4.2.1. Objective-Based Optimization. The objective-based (OB) optimization aims to maximize the expected number of accepted offers based on the platform's approximated preferences for the suppliers currently in the system. The preferences in state S_t are represented by two parameters. Parameter $\underline{v}_{0jt}$ indicates supplier s_j's approximated lower border threshold in time step t. Parameter $\overline{v}_{0jt}$ represents supplier s_j's approximated upper border threshold in time step t. By definition, $\underline{v}_{0jt} \leq \overline{v}_{0jt}$. The OB approach translates these parameters into probabilities of acceptances for all suppliers and requests currently in the system.

Because a platform has uncertainty in the belief state of each supplier s_j's acceptance threshold, in this approach we consider an interval, in which for requests with utility values below the lower

Algorithm 1 Pseudo-code for OB-RA Process

$t \leftarrow 0$
for all $s_j \in s$ **do**
  $\underline{v}_{0j0} \leftarrow 0$ ▷ Initialize lower border threshold
  $\overline{v}_{0j0} \leftarrow v_{\max}$ ▷ Initialize upper border threshold
end for
$t \leftarrow 1$ ▷ Initialize time horizon
while $t \leq T$ **do**
  $x_t \leftarrow OB(S_t, \underline{v}_{0t}, \overline{v}_{0t}, \hat{v}_t)$ ▷ Solve optimization OB
  $\omega_t^x \leftarrow SupplierDecisions(x_t)$ ▷ Suppliers reject or accept their respective offer
  for all $s_j \in s_t^a$ **do**
    for all $r_i \in r_t^o$ **do**
      for all $\omega_{ijt}^x \in \omega_t^x$ **do**
        if $\omega_{ijt}^x = 0$ && $x_{ijt} = 1$ **then**
          $\underline{v}_{0jt+1} \leftarrow UpdateLowerThresh(\hat{v}_{ijt}, \underline{v}_{0jt})$ ▷ Apply R approximation
        else if $\omega_{ijt}^x = 1$ **then**
          $\overline{v}_{0jt+1} \leftarrow UpdateUpperThresh(\hat{v}_{ijt}, \overline{v}_{0jt})$ ▷ Apply A approximation
        end if
      end for
    end for
  end for
  $S_{t+1} \leftarrow Transition(S_t^{x,\omega})$ ▷ New suppliers and requests enter the system
  $t \leftarrow t + 1$
end while

border threshold $\underline{v}_{0jt}$, the platform assumes the supplier will reject, and above the upper border threshold $\overline{v}_{0jt}$, accept. In between those upper and lower bound thresholds, the platform assigns a probability of acceptance of supplier s_j accepting request r_i, $\hat{p}_{jt}(r_i)$, between 0 and 1. In our method, we use a piece-wise linear function:

$$\hat{p}_{jt}(r_i) = \begin{cases} 1 & \text{if } \hat{v}_{ijt} \geq \overline{v}_{0jt} \\ \frac{\hat{v}_{ijt} - \underline{v}_{0jt}}{\underline{v}_{0jt}} & \text{if } \overline{v}_{0jt} > \hat{v}_{ijt} \geq \underline{v}_{0jt} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The initial lower border threshold, $\underline{v}_{0j0}$, is given a value of zero, indicating that, initially, the platform assumes with a positive probability $\hat{p}_{jt}(r_i)$ that supplier s_j will accept requests with any utility value. The initial upper border threshold, $\overline{v}_{0j0}$, is set at a value that is at least as high as

the highest possible value of a utility, $v_{\max}$, which implies that the platform initially assumes there is no guarantee that supplier s_j will accept any request.

Then, in every state S_t, a integer linear program is solved via function $OB(S_t, \underline{v}_{0t}, \overline{v}_{0t}, \hat{v}_t)$ in Algorithm 1. Inputs for the function are the current approximated thresholds for all suppliers via vectors $\underline{v}_{0t}$ and $\overline{v}_{0t}$, and the matrix of observable utilities for all pairs of suppliers and requests, $\hat{v}_t$. This program maximizes objective (5), subject to constraints (1) and (2). Specifically, this optimization method maximizes the (approximated) expected number of accepted offers, while enforcing that the platform can only offer at most a single request to a supplier, and each request can only be offered to at most a single supplier.

$$\max \quad \sum_{r_i \in r_t^o} \sum_{s_j \in s_t^a} \hat{p}_{jt}(r_i) x_{ijt} \tag{5}$$

4.2.2. Parameter-Based Probability Approximation. The Parameter-Based approximation updates the platform's estimates for the set of suppliers that make a selection in a given time step. Specifically, a supplier s_j's lower border threshold $\underline{v}_{0jt}$ or upper border threshold $\overline{v}_{0jt}$ is updated based on supplier s_j's selection ω^x_{ijt} of offered request r_i. For each selection a supplier s_j makes in t, only one of two outcomes is possible: a rejection ($\omega^x_{ijt} = 0$) or an acceptance ($\omega^x_{ijt} = 1$) of offered request r_i. Thus, the parameter-based approximation updates only one of the two border thresholds in each time step supplier s_j makes a selection.

Due to the unobservable attribute ϵ_{ij}, the update of the thresholds needs to be carefully designed. On the one hand, the larger the threshold step size updates are, the faster the approximation can be achieved. This is important since suppliers do not tolerate many unacceptable offers. On the other hand, thresholds can become too strict, e.g., if a supplier rejects an offer with an objectively high utility, it indicates a relatively high pickiness, but the rejection may be based solely on the unobservable attribute ϵ_{ij} not known to the platform. A more aggressive update of the thresholds could reduce the number of offers made to this supplier in the future, and may result in the platform assuming a supplier would not accept a request, even though a supplier would actually be happy to serve it. To strike the right balance between fast approximation and not preventing future offers, we update the thresholds as a combination of previous thresholds and new observation data.

The general procedure is shown in Figure 3. Let $\underline{v}_{0jt}$ and $\overline{v}_{0jt}$ be the approximated thresholds of supplier s_j in time t. Let $\hat{v}_{ijt}$ be the (observable) utility of offered request r_i with $\underline{v}_{0jt} \leq \hat{v}_{ijt} \leq \overline{v}_{0jt}$. If the supplier rejects offer x_{ijt} (i.e., $\omega^x_{ijt} = 0$), as shown in the upper right of Figure 3, the platform updates the estimate of that supplier's lower border threshold $\underline{v}_{0jt+1}$ based on the previous estimate and the rejected request's estimated utility value:

$$\underline{v}_{0jt+1} = (1 - \rho)\underline{v}_{0jt} + \rho \hat{v}_{ijt}.$$

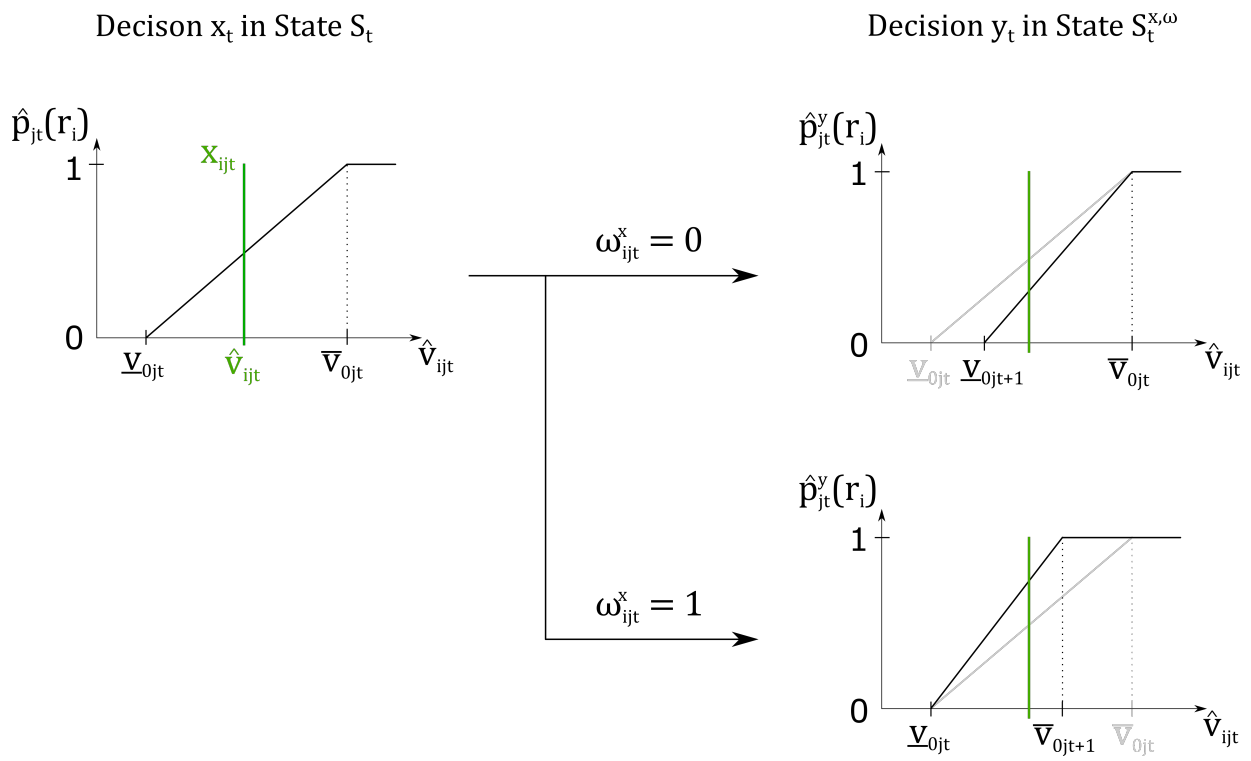

Figure 3 Update of Probability Function For Supplier s_j Via Parameter-Based Approximation

This is done via function $\mathit{UpdateLowerThresh}(\hat{v}_{ijt}, \underline{v}_{0jt})$ in Algorithm 1. Parameter $\rho \in [0,1]$ denotes the step size of the update. The extreme value of $\rho = 0$ would not change the thresholds at all (no approximation), while a value of $\rho = 1$ would solely rely on the new observation (likely overly aggressive).

Similarly, if the supplier accepts the offer (i.e., $\omega_{ijt}^x = 1$), as show in the lower right of Figure 3, the platform updates the estimate of that supplier's upper border threshold $\overline{v}_{0jt+1}$ based on the previous estimate and the accepted request's estimated utility value:

$$\overline{v}_{0jt+1} = (1-\rho)\overline{v}_{0jt} + \rho\hat{v}_{ijt}.$$

This is done via function $\mathit{UpdateUpperThresh}(\hat{v}_{ijt}, \overline{v}_{0jt})$ in Algorithm 1. We note that because $\hat{v}_{ijt} \geq \underline{v}_{0jt}$ and $\hat{v}_{ijt} \leq \overline{v}_{0jt}$, it is easy to show that there is monotonicity in values $\underline{v}_{0jt}$ and $\overline{v}_{0jt}$ over the time steps t. Thus, over time, the difference between $\overline{v}_{0jt}$ and $\underline{v}_{0jt}$ decreases, and the approximation becomes tighter. Further, in the special case that the exact utility values are known (thus, there is no unobservable attribute ϵ_{ij}), we can fully trust our observations and update aggressively by setting $\rho = 1$. Else, we set $\rho = 0.5$ based on preliminary tests. In the very rare case when the estimated lower border threshold is updated and becomes greater than the estimated upper border threshold, the estimated upper border threshold is reset to its initial value.

4.3. Benchmark Approaches

To compare our main approach to existing methods, we explore alternative, benchmark approaches that combine different optimizations and probability approximations, and a perfect information policy.

4.3.1. Constraint-Based Optimization. A benchmark optimization step is motivated by the large body of existing literature (e.g., Archetti et al. (2016), Arslan et al. (2019)), which assumes that only requests above a certain threshold value (often in terms of detour distance or extra time) can be offered to suppliers. This is similar to the procedure in our example with the point-based estimation where it is assumed that every request with utility higher than the acceptance threshold is accepted and every request with utility below is rejected. Thus, instead of probability values, the point estimate threshold can be seen rather as a constraint reducing the set of requests that can be offered to a supplier. Consequently, we label this approach *constraint-based* (CB) optimization. Similar to our objective-based method, CB can also integrate approximations of the threshold, however, only from rejections since acceptances do not change the constraint.

To implement this strategy, we define an additional parameter, q_{ijt}, which is 1 if the platform can offer an request to a supplier based on the estimated lower border threshold (i.e., $\hat{v}_{ijt} > \underline{v}_{0jt}$); else, 0. The constraint-based (CB) approach maximizes function (6), which seeks to maximize the number of assignments (assuming that all suppliers will accept any request offered as long as the request is above their known threshold utility), subject to constraints (7) that ensure that the platform only offers a request with an acceptable utility value to a supplier, and constraints (1) and (2) that enforce that the platform can only offer a single request to a supplier, and each request can only be offered to a single supplier, respectively.

$$\max \quad \sum_{r_i \in r_t^o} \sum_{s_j \in s_t^a} x_{ijt} \tag{6}$$

$$x_{ijt} \leq q_{ijt} \qquad \forall r_i \in r_t^o, \forall s_j \in s_t^a \tag{7}$$

4.3.2. Benchmark Probability Approximations. We present five different benchmark probability approximations, in which the first two do not update the platform's estimates of supplier thresholds, whereas the last three do update estimates, but in a different way than our proposed parameter-based probability approximation in Section 4.2.2.

None (N): The None approximation does not update the initial values of the estimated supplier thresholds, meaning the estimated lower border threshold and the estimated upper border threshold of every supplier retain their respective initial value throughout the time horizon. This can be seen as setting $\rho = 0$ in the OB-RA case. Combined with the CB optimization, this approximation does not consider supplier preferences but assumes every request will be accepted by every supplier.

Mean (M): The Mean approximation assumes all suppliers will reject requests with utility below the (global) average threshold value. To do this, we set the initial estimated lower border threshold to its expected value, and the upper border threshold is set to its initial value. Neither of the estimated thresholds are updated throughout the time horizon. Combined with CB, this approximation does not consider supplier preference heterogeneity, as it assumes the same supplier preference behavior across all suppliers, and also does not update its assumptions based on any supplier observations.

Exclude (E): The Exclude approximation prevents a rejected request to be re-offered to the same supplier in future time steps. Combined with CB, this approximation is implemented by adding a new binary parameter m_{ijt} that is 0 if supplier s_j rejected request r_i in a previous time step and 1 otherwise. The new constraint (8) then prevents the platform from offering a previously-rejected request to the same supplier in the next time step and all future time steps.

$$x_{ijt} \leq m_{ijt} \quad \forall r_i \in r_t^o, \forall s_j \in s_t^a \tag{8}$$

Combined with OB, this approximation is implemented by updating the probability of acceptance of the rejected request for that supplier to zero, i.e., if supplier s_j rejected request r_i in a previous time step, then for all subsequent time steps, $\hat{p}_{jt}(r_i) = 0$.

Rejections Only (R): The Rejections Only approximation applies our proposed approximation as described in Section 4.2.2, but only in case of supplier rejections. Thus, it only updates the estimate supplier lower border thresholds.

Acceptances Only (A): The Acceptances Only approximation applies our proposed approximation as described in Section 4.2.2, only in case of supplier acceptances, so it only updates the estimate supplier upper border thresholds.

4.3.3. Perfect Information Optimization. Finally, we apply a strategy that can be seen as an "upper bound" of our approximations. In this method, we assume that the platform knows the exact utilities and acceptance thresholds for every supplier, which would not be realistic in practice, but serves as a useful benchmark for comparison purposes. The perfect information optimization solves the same integer program as the CB optimization, but instead of estimates as inputs, it uses

the suppliers' actual acceptance thresholds v_{0j} and their actual utility values v_{ijt} (including actual ϵ_{ij} values). We note that while this policy is an upper bound on the expected reward of a decision, it is theoretically not an upper bound on our policy because it does not capture the dynamics over multiple periods.

5. Experimental Set Up

We design a set of computational experiments to analyze the performance of our solution approach. While our model captures a variety of crowdsourced delivery applications, we base our experiments on restaurant meal delivery where the issue of suppliers rejecting offered jobs is especially severe (Savelsbergh and Ulmer 2022). When interested in driving and delivering food for a third-party platform, independent suppliers log in to the platform's app. Simultaneously and throughout the day, customers place orders for delivery of food from local restaurants that have also partnered with the platform. The platform facilitates the matching of suppliers who are tasked with picking up a specific order from a restaurant and then dropping the ordered food off at the customer's requested destination location. As such, each order is linked to a restaurant.

Next, we describe the problem parameters in our experiments and then the supplier choice model in detail. This section finishes with an overview of the design of experiments.

5.1. Problem Parameters

We consider the meal delivery service area to be a 10km times 10km square, and contains 50 potential restaurants from which customers can request food for delivery to their locations (which is in the service area). We model travel based on rectilinear distances. We consider two spatial distributions of these restaurants: spread out and centrally clustered. In the spread out restaurant case, the restaurants are equally likely to span any part of the service area, and we generate the restaurant locations so that they are uniformly distributed over the service area. In the centrally clustered restaurant case, all the restaurants are in the city-center, and we implement this case by generating restaurant locations to be uniformly located within a square-kilometer in the center of the service area.

For our experiments, decisions are made every five minutes. We set the overall daily service horizon to 100 time steps (500 minutes). We further consider 1000 expected requests per day that appear uniformly over the time horizon. The probability that a request is for a given restaurant is equally likely across the 50 restaurants. The delivery location of the requests are uniformly distributed across the service area. We assume that, on average, 100 suppliers participate every day. They are not all available in the beginning but appear randomly at a rate that is uniformly distributed across the first 25 time steps of the time horizon, i.e., 4 new suppliers arrive per time

step on average. The coordinates of the initial locations of the suppliers are drawn randomly from a uniform distribution.

If a supplier is matched with a request, they perform the delivery, which means they travel from their current location to the restaurant and then travel to the delivery location. During this time, they are not available to be matched in the system. We model the supplier's reappearance time δ_t as the travel time from the supplier's current location to the restaurant, plus the travel time from the restaurant to the request. We assume suppliers travel with an average speed of 36 kilometers per hour. Suppliers then re-appear in the system at the destination of their most recently served order.

Requests are assumed to wait a maximum of 5 time steps (20-25 minutes) to get assigned; otherwise, they leave the system. Suppliers may have varying levels of patience of how long they are willing to wait to be matched with an order until they decide to exit the system, so we set supplier patience as a parameter in our experiments, testing low (5 time steps), medium (10 time steps), and high patience (30 time steps).

5.2. Supplier Choice Model

In this section we describe how suppliers evaluate offered requests. We assume that all requests have the same fixed maximum utility value, e.g., the fixed compensation by the platform. From this value, we deduct the "hassle" a supplier has to endure to satisfy the order. In our experiments, we measure this by four attributes motivated by Castillo et al. (2022): time of travel, navigation difficulties, tipping amount, and the unobservable attribute. The first three attributes are assumed to be equally important to all suppliers, and the deduction is relative to the maximum hassle per attribute, e.g., when travel is exceptionally long, navigation at the restaurant is very difficult, or no tip can be expected. The first three attributes are observable to the platform. In addition to these observable values, each supplier's utility value includes an unobservable value capturing attributes hidden to the platform.

Mathematically, we rely on an additive utility model with bounded rationality (Train 2009). The maximum utility value is $v_{\max}$ from which are deducted the three known attributes: (a^1) restaurant characteristic (e.g., parking availability, neighborhood, etc.), (a^2) the monetary value of the order or the tipping amount, and (a^3) the distance from the supplier's origin to the restaurant's location and then to the request's destination. Further, we deduct the unobservable supplier and request specific attribute ϵ_{ij}. Overall, the utility for supplier s_j and request r_i in time t is then calculated as:

$$v_{ijt} = v_{\max} - (a_i^1 + a_i^2 + a_{ijt}^3) - \epsilon_{ij}.$$

We assume that all three attributes can be normalized into values between 0 and 1, i.e., $a_i^1, a_i^2, a_{ijt}^3 \in [0,1]$, for all suppliers and requests, with 1 being the maximum "hassle" possible. The first two components, a_i^1 and a_i^2, are specific to a request r_i and are categorized into three utility levels: low, medium, and high. The low category is assigned the deduction value of 0, the medium category the value of 0.5, and the high category the value of 1. The third component a^3, which varies for each supplier, is the relative travel time and is calculated as the travel time from the supplier's origin to the order's restaurant location, and then to the order's destination. This travel time is then normalized by dividing it by the maximum travel time possible for the restaurant, i.e., twice the maximum travel time from a restaurant's location to the location furthest away from the restaurant.

Following standard bounded rationality models, the unobservable attribute value ϵ_{ij} for each request-supplier combination is normally distributed with a mean of zero and a standard deviation of σ^ϵ (i.e., $\epsilon_{ij} \sim \mathrm{N}(0, \sigma^\epsilon)$). In our experiments, we test three different values for σ^ϵ, specified in the next subsection (see Table 1). We note that the unobservable attribute can also increase the utility of a request. Given the four attributes, we set $v_{\max} = 5$ to ensure that the utility of a request is generally positive, even though it might be below a supplier's acceptance threshold.

A supplier's fixed minimum acceptance threshold v_{0j} is generated from a normal distribution with a mean of μ^{v_0} and a standard deviation of σ^{v_0}, i.e., $v_{0j} = N(\mu^{v_0}, \sigma^{v_0})$. Both values will be varied in our experiments as described in the next subsection. The mean μ^{v_0} represents the collective level of pickiness of suppliers in the system: a high value for μ^{v_0} represents the platform has picky suppliers, while a low value for μ^{v_0} represents the platform has agreeable suppliers. The standard deviation σ^{v_0} establishes a heterogeneity level among the suppliers: a low standard deviation value produces suppliers with similar acceptance thresholds, while a higher standard deviation value yields greater disparity in acceptance thresholds among the suppliers. Model-wise, the minimum acceptance threshold v_{0j} represents the utility of the "no-choice" option. Thus, in time t, supplier s_j accepts offered request r_i in case of $v_{ijt} \geq v_{0j}$. Else, the offer is rejected.

5.3. Design of Experiments

For our experiments, we capture a variety of exogenous factors summarized in Table 1, as well as various probability approximation policies for the benchmark methods and our method, summarized in Table 2. For each method-combination, we conduct a full factorial experiment on the exogenous levels, resulting in 162 experiments with different parameters. We run 20 replications, for a total of 3,240 instance realizations (days).

Factor	Levels
unobservable attribute std dev σ^{ϵ}	1.5x0% (none)
	1.5x10% (low)
	1.5x20% (high)
supplier patience $t^{s}_{\max}$ (time steps)	5 (low)
	10 (med)
	30 (high)
supplier heterogeneity σ^{v_0}	0.1 (low)
	0.3 (med)
	0.5 (high)
supplier pickiness μ^{v_0}	2.5 (low)
	3.25 (med)
	4 (high)
restaurant layout	centrally clustered
	spread out

Table 1 Exogenous Factors and Levels for Design of Experiments

Factor	Level
optimization	CB
	OB
probability approximation	N
	M
	E
	R
	A (only OB)
	RA (only OB)
	PI (only CB)

Table 2 Methods for Design of Experiments

6. Computational Study

In this section, we compare the performance of our method OB-RA against the set of benchmark methods introduced in Section 4.3. We first compare the main goal of the platform, i.e., the objective function, which is to maximize the number of successful matches over the day, and how this objective function value is influenced with different exogenous factors. The section concludes with an analysis of the experiences of the other two stakeholders, i.e., the requesting customers and the suppliers.

6.1. Changes to the Objective Function Values

First, we compare the average objective values of all policies over all instances in our design of experiments. We calculate the average improvement in assignments of all the other policies against the policy CB-N (which has the platform make offering decisions assuming all offers are accepted), which is used as the baseline benchmark. The results are shown in Figure 4: the lighter the color, the smaller the percentage difference, and conversely, the darker the color, the greater the percentage difference. Our proposed method OB-RA is highlighted in red. First, we observe that

explicitly incorporating acceptance probabilities (OB) is superior to the respective constraint-based methods, regardless of the approximation. Even the OB-method without any approximation, OB-N, improves upon the baseline benchmark by 13%, which is better than any of the improvements using the constraint-based policies. The superior performance of the objective-based approaches is independent of the underlying distribution of restaurants as shown in Figure 5. This leads to our first main insight:

Insight 1. Acknowledging that suppliers can reject offered requests and considering acceptance probabilities are valuable for peer-to-peer transportation platforms. Providers should therefore analyze supplier behavior carefully and develop assignment mechanisms that can integrate acceptance probabilities in their offering decision making process.

We further observe that updating acceptance probabilities based on suppliers' interactions with the platform increases the average improvement in number of accepted offers even more, from 13% (for OB-N) to 19% (for OB-RA) when both acceptance and rejection decisions (RA) are used in the update process. Further, our method (OB-RA) is between an objective-based method not updating probabilities (13%) and one assuming perfect information (27%). This observation holds for different levels of supplier patience, as shown in Figure 6. However, updating approximations is generally more valuable if suppliers are more patient, because the number of observations per supplier tends to increase, leading to an even better understanding of the individual suppliers. Updating approximations is particularly important when suppliers are very picky, as detailed in Figure 7. In the case where every supplier mostly accepts every request (low pickiness), the value of approximating is rather small with only a 3% increase over a policy that ignores supplier behavior when making offering decisions. Yet, in situations when suppliers are not willing to accept all requests, even at medium pickiness levels, the average improvement is 21%. If suppliers are very picky, the average improvement goes up to 60%.

Another aspect of supplier behavior is their predictability by the platform, which we model via the two types of attributes in the utility model (i.e., observable and unobservable). With increasing impact of the unobservable attribute, the platform has a more challenging time understanding the suppliers. This is illustrated in Figure 8, which shows the value of our method for different levels of the unobservable attribute. In the case of no unobservable attributes ("none"), the platform can approximate suppliers' preferences relatively easily based on their previous interactions with the platform, and our method performs very well. It nearly matches the performance of the perfect information policy which has full knowledge of actual utility values and thresholds. In contrast, when the unobservable attribute has a high impact, our method performs only as well as method OB-E, which only excludes unsuccessful offers. In this very unpredictable setting, understanding

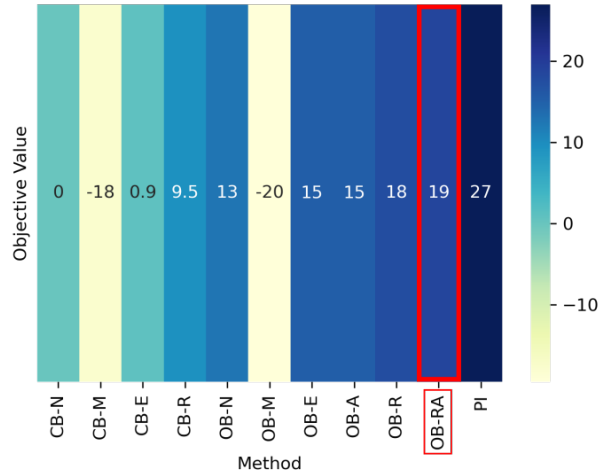

Figure 4 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N**

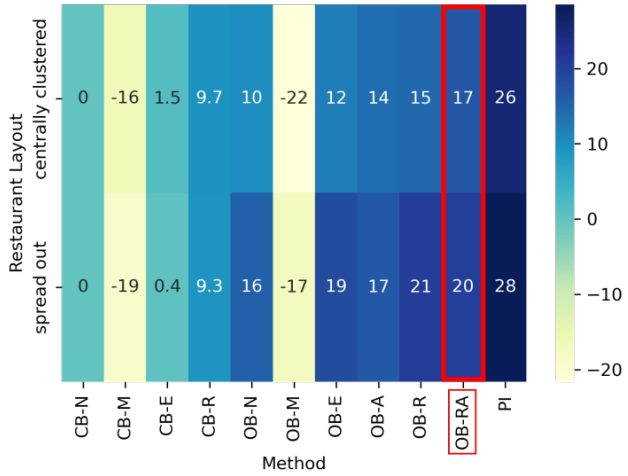

Figure 5 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N - Restaurant Breakdown**

the suppliers' decision with respect to the observable attributes becomes very difficult. Notably, even in this very unpredictable setting, considering probabilities (OB) is still better than a platform using constraints (CB).

In all the results, the value of approximating from rejections only (OB-R) is relatively close to that of our main method (OB-RA), while approximating from acceptances only (OB-A) provides less improvement compared to not approximating at all (OB-N). This can be explained by the suppliers' impatience. In the case of a rejection, the supplier is dissatisfied with the platform's offer and waits for an acceptable offer, but only for a limited time until the supplier abandons the system. Thus, updating the supplier's acceptance threshold to be higher than currently thought is substantially more important compared to the case where the supplier accepted the offer and therefore is satisfied with the platform's service. This leads to our second main insight:

Insight 2. Understanding suppliers better and using this information to offer requests that better balance the needs of the platform with suppliers' preferences is very important, especially when suppliers are relatively picky or when they are more patient, even when they get offered unacceptable requests. Thus, a platform may consider ways to increase the suppliers' patience, e.g., via a bonus program or by rewarding guaranteed availability during the day. While, ideally, the platform should use both acceptance and rejection decisions for a better approximation, it is especially important to understand why a supplier was unhappy about an offered request and therefore rejected it. This avoids adding to the corresponding supplier's dissatisfaction with more unacceptable offers and, in the worst case, the supplier leaving the platform entirely.

The importance of fast and accurate approximation updates differs, depending on the suppliers' general behavior. As previously discussed, the results in Figure 4 indicate that considering and

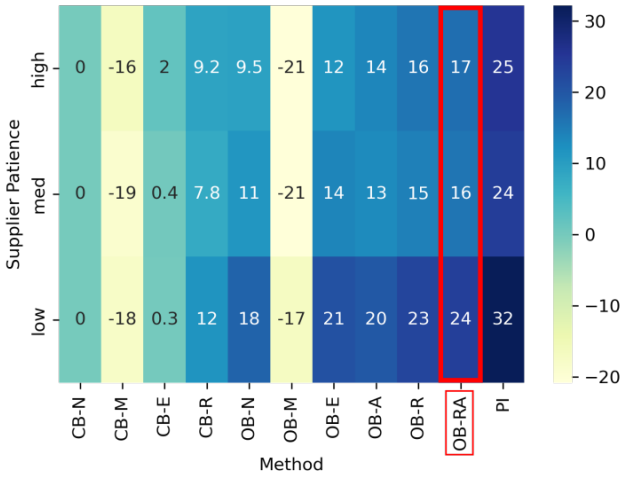

Figure 6 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N - Supplier Patience Breakdown**

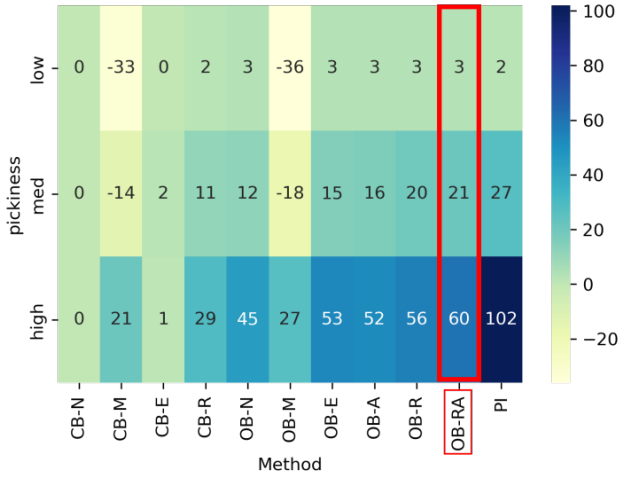

Figure 7 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N - Pickiness Breakdown**

approximating acceptance probabilities is valuable compared to assuming every supplier accepts every request. Interestingly, method CB-M, which assumes suppliers reject requests but every supplier has the same threshold, performs substantially worse than CB-N. This is especially noteworthy as this assumption is one of the most prominent approaches in the literature (e.g., Archetti et al. 2016, Arslan et al. 2019). One reason for this poor performance is assuming that all suppliers are homogeneous, which leads to two problems. First, and most obvious, pickier suppliers may be offered unacceptable requests. Second, and maybe more subtle, is that some suppliers are willing to serve less attractive requests, but because the platform does not offer such requests to them, the matches are not made. For both cases, acknowledging and understanding heterogeneity within the set of suppliers is necessary. To further analyze this phenomenon, we differentiate the improvements with respect to the suppliers' heterogeneity, shown in Figure 9. We observe that with increasing heterogeneity, CB-M performs even worse, while the improvement of OB-RA increases even further from 17% (low heterogeneity) to 21% (high heterogeneity). This leads us to our third main insight:

Insight 3. Every supplier is different. Treating them equally not only leads to unhappy suppliers leaving the system due to unacceptable offers, but also to missed assignment opportunities for more agreeable suppliers who are willing to serve less attractive requests. Platforms should therefore consider their suppliers individually, understanding and utilizing their heterogeneous preferences. This is especially important in cases where the supplier base is quite heterogeneous, e.g., if they consist of rather "occasional" drivers only serving very convenient requests versus gig-workers who rely on this for their main source of income and may have to serve a higher number of requests offered to them.

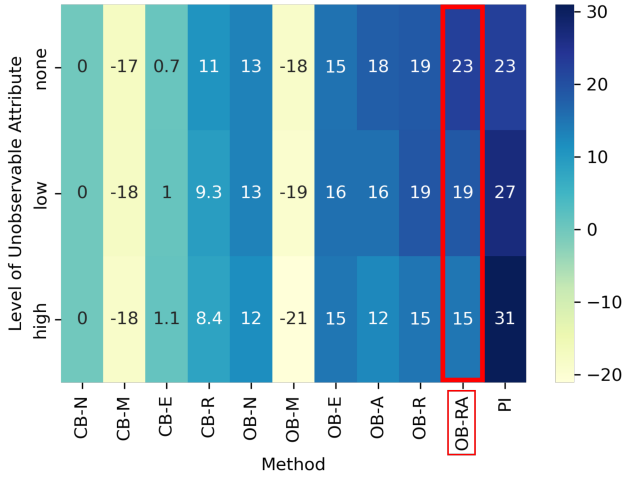

Figure 8 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N - Unobservable Attribute Breakdown**

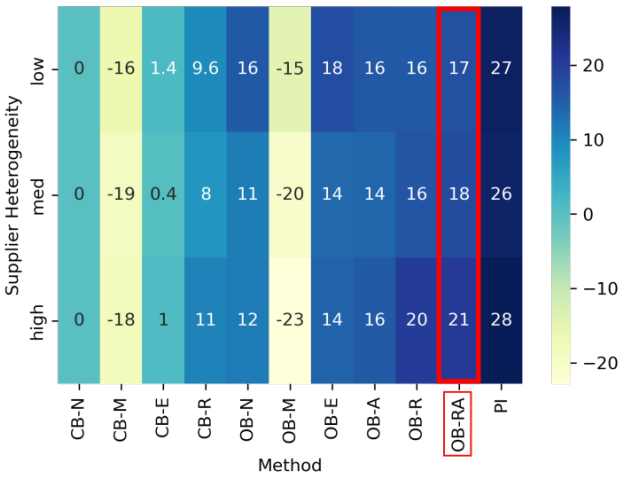

Figure 9 **Heat Map of % Increase of Number of Assignments for Methods Compared to CB-N - Supplier Heterogeneity**

6.2. Changes for the Requesting Customers

While our method increases the platform's objective significantly, it also changes the experiences for the requesting customers. First, with the increase in successful assignments, the number of requests experiencing recourse decreases, as does the average time a request waits to be matched. Figure 10 shows the distribution of the average request match time, i.e., the number of time steps a request stays in the system until getting assigned or leaving, across supplier pickiness levels, for the benchmark CB-N, the perfect information case PI, and our method OB-RA. When suppliers are very agreeable (i.e., low pickiness), our method OB-RA matches requests to suppliers, on average, as quickly as the perfect information case and slightly faster than the CB-N case. With low pickiness, offers are usually accepted, and no request has to wait longer than one time step, regardless of method. As suppliers become pickier, requests take longer to get matched, regardless of whether a platform adopts approximating or not. Yet, when suppliers are pickier, approximating substantially improves the time to match a request against not approximating by one time step or more. As each time step reflects five minutes in the process, this means that the requests are served significantly faster. This leads to our fourth insight:

Insight 4. Platforms that understand suppliers better and consider uncertain and heterogeneous supplier behavior in their offer decisions make more successful matches (and thus more revenue), but also operate systems with better and faster service for the requesting customers. This is critical for on-demand transportation services with instant gratification and especially for meal delivery, where only minutes can lie between fresh and soggy food.

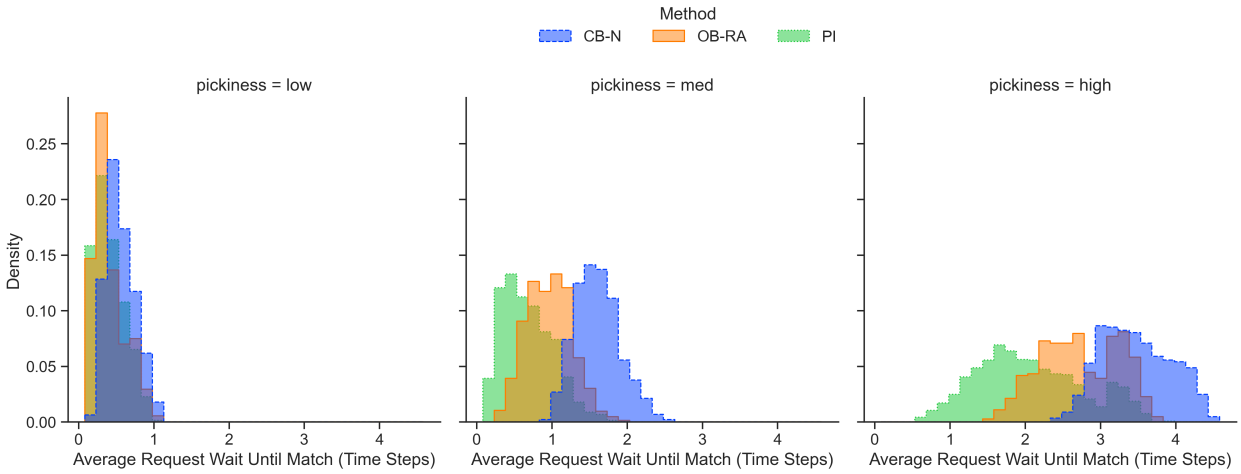

Figure 10 Distribution of Request Match Time in Time Steps

6.3. Changes for the Suppliers

Similarly to the requesting customers, the increase in successful offers also increases the average number of jobs completed per supplier. Furthermore, our method decreases the time a supplier waits for an acceptable offer, however, not as significantly as for the requests. Figure 11 shows the distribution of the average supplier match time in time steps, similarly to Figure 10. The two peaks reflect the two different restaurant location layouts. Interestingly, when supplier pickiness is low, suppliers wait slightly longer with our method compared to CB-N. The reason is that with our method, fewer suppliers abandon the system due to unacceptable offers. Thus, with more suppliers in the system and suppliers accepting most of the offers, the relative assignment opportunities decrease. For cases with higher supplier pickiness, the number of requests per supplier is still sufficient, and our method leads to faster assignments for the suppliers. We summarize these observations in our fifth insight:

Insight 5. Understanding suppliers better and making better offers *can* reduce the time required to match a supplier successfully with an acceptable request. However, if suppliers are willing to participate, a higher number of satisfied suppliers stay in the system looking for the next offer. Thus, the platform should expect and plan to match the increase in suppliers by making sure that enough requests are also generated to increase revenue even more while keeping the participating suppliers happy.

Finally, we analyze how understanding suppliers impacts the number of assignments and the utility across the set of different suppliers. Figure 12 depicts the relationship between a supplier's acceptance threshold and the average number of requests serviced per supplier, across the three pickiness levels and for three methods: the benchmark CB-N, our method OB-RA, and the upper bound PI. As expected, for methods CB-N and OB-RA, very picky suppliers usually service much

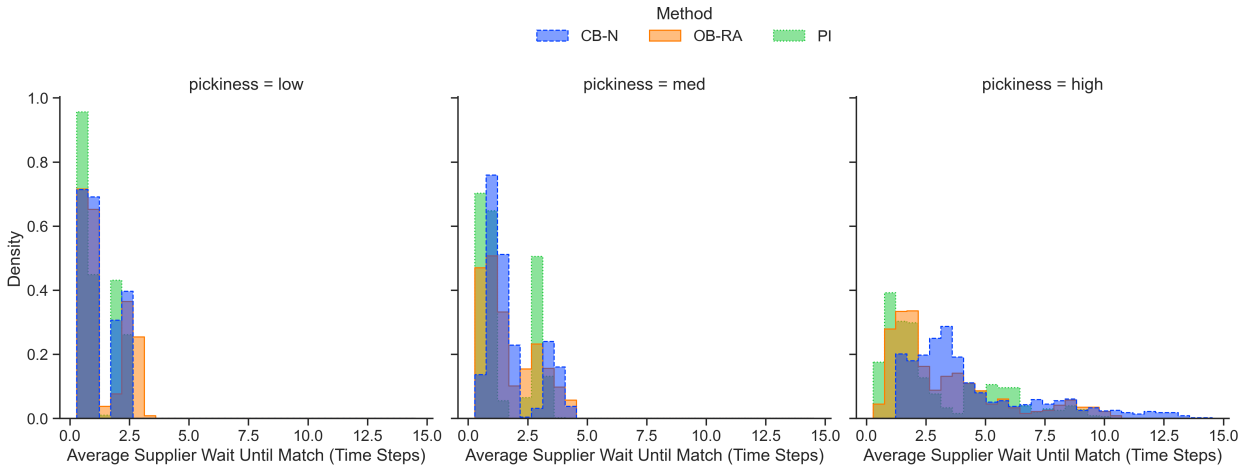

Figure 11 **Distribution of Supplier Match Time in Time Steps**

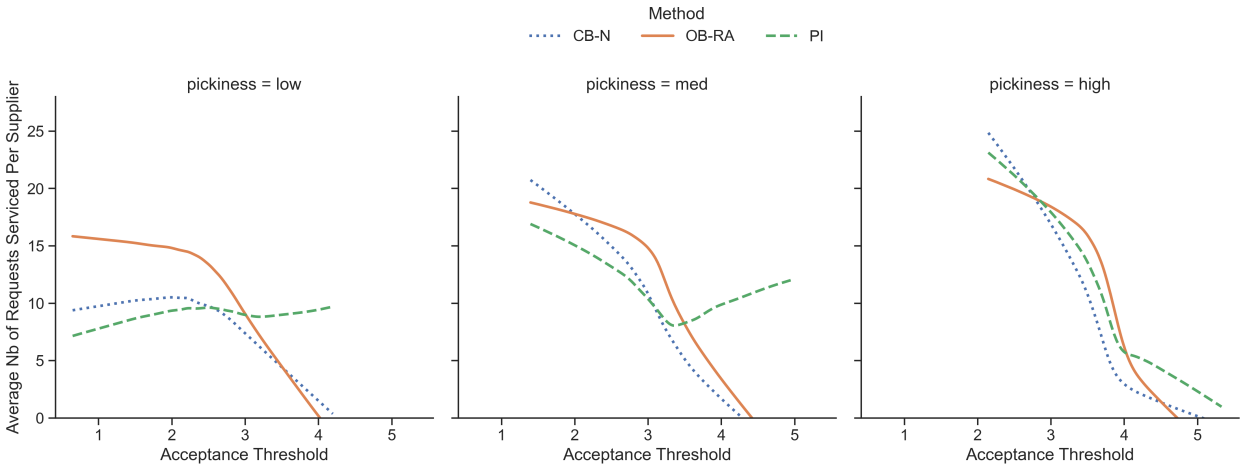

Figure 12 **Relationship Between the Average Number Of Requests Serviced Per Supplier and Supplier Acceptance Thresholds**

fewer requests (over 10 fewer requests) than more agreeable suppliers, especially if all suppliers are relatively picky. However, in the perfect information case, we see that more picky suppliers actually serve more requests than less picky ones. This (at first glance) counterintuitive phenomenon can be explained by the relation between utility values and travel time required for service. With increasing travel time, the utility of a request decreases. Thus, such a request is likely rejected by the pickiest suppliers. Given perfect information, the platform offers the high utility requests with short travel times to the very picky suppliers, while the less picky ones are assigned the requests requiring longer travel. This leads to picky suppliers reappearing in the system earlier to get their next offer, while less picky suppliers travel longer and therefore serve fewer requests per day, even though (or because) they accept all requests offered.

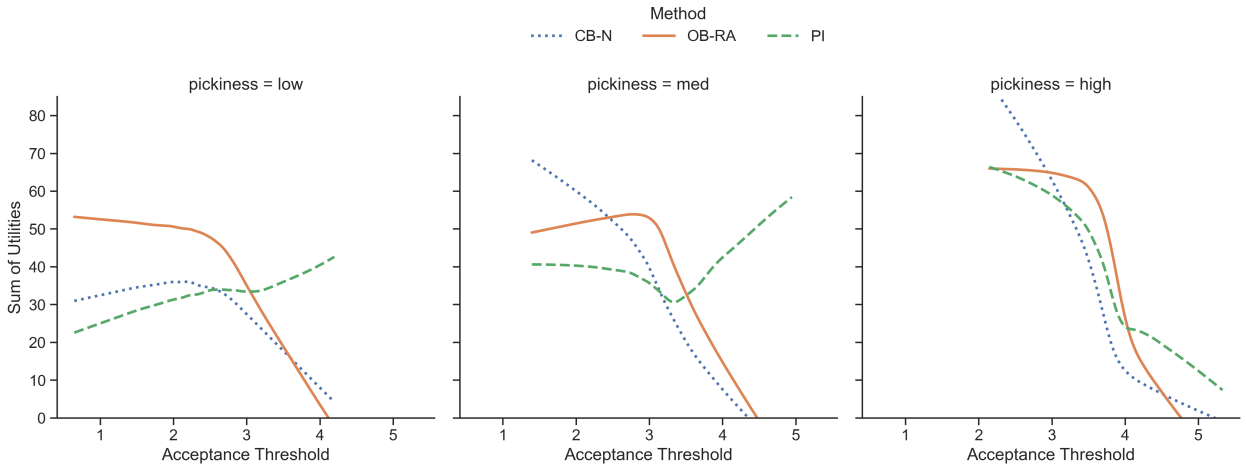

Figure 13 Relationship Between the Average Total Utility of Assignments and Supplier Acceptance Thresholds

While a similar tendency can be observed with our method, in general, the more agreeable a supplier is, the more requests they will be assigned, which one would consider as fair. However, even with our method, the offered requests have relatively lower average utility values, as our method can differentiate between less and more picky suppliers. Thus, although less picky suppliers serve more requests, the total utility per day (i.e., the sum of the utilities of all the orders serviced by a supplier in a day) may actually be smaller for them. Indeed, as Figure 13 shows, our method does not always improve the sum of utility across the day for all suppliers. In the low pickiness case, all suppliers achieve higher utility values regardless of their thresholds. But with medium and high average pickiness, the average utility for rather agreeable suppliers decreases compared to method CB-N. Furthermore, at least for the instances with medium average pickiness, there is a sweet spot of pickiness where suppliers actually gain more utility when accepting fewer offers (again, in the perfect information case, this phenomenon is even more distinct). This observation is in line with work on ride-sharing where more experienced and selective suppliers benefit, and new and agreeable suppliers work more while earning less (Cook et al. 2021). This leads us to our final major insight:

Insight 6. Understanding suppliers and considering their individual preferences may increase platform revenue and service level for requests. However, it also comes with the risk of systematic unfairness to the suppliers. Platforms may identify and exploit agreeable suppliers while rewarding very selective ones. At the same time, it might be of interest for suppliers to "game" the system by not accepting all offered requests initially, even though they might be acceptable. Such strategic behavior by suppliers could pay off in higher total utility values in the long run.

7. Discussions and Future Research

In this work, we have shown that approximating, updating, and integrating supplier acceptance behavior into a platform's offering decisions is valuable for peer-to-peer transportation platforms but also for customers and most suppliers. This work is the foundation of several avenues for future work. Our work has shown that approximating supplier behavior via probability functions on the operational level and updating the approximation over the course of the day, is already very valuable. Future work may investigate the impact of supplier-specific approximations over more periods, not only for understanding their acceptance behavior, but also other information, such as the times and areas a specific supplier likes to work. This becomes particularly interesting when considering that suppliers may act strategically (Wang et al. 2022). Our work has also shown that the approximation has to be made fast as suppliers may leave the system. Here, future work may create methods that increase the chance of supplier acceptance and provide new opportunities for approximations, for example, by offering menus to the suppliers to choose from (Ausseil et al. 2022). Such menus should then carefully balance requests that are likely to be accepted with more "risky" requests to explore the supplier behavior in more detail.

We have shown that approximation is particularly valuable when the utility values of suppliers are relatively clear to the platform. However, as the unobservable attributes contribute more to the supplier's utility values, the more difficult the approximation becomes. Thus, another interesting area of future research is to better understand and model the utility attributes of suppliers. Our research has focused on the three main attributes suggested in Castillo et al. (2022): travel time, location of the restaurant, and tipping amount. However, there may be several additional attributes adding value to the approximation, for example, the area a supplier is most familiar with (Auad et al. 2021) or how long the supplier has interacted with the platform already. Future research could create approximation approaches that explicitly capture such temporally and/or spatially varying supplier acceptance behaviors. We have also seen that approximating supplier behavior may lead to unfair distribution of offered requests. This may also affect the longer-term participation of new or agreeable suppliers. Therefore, companies may consider developing methods that increase equitability across offers made to suppliers, or alternatively, bonus programs for "committed" suppliers, who accept all or at least most of the offered requests (Behrendt et al. 2022). Our work may be the starting point for considering pricing schemes, e.g., by increasing compensation for an offer if the observable utility is below a supplier's threshold (or by decreasing the compensation if the utility is way above the threshold). Our research has shown that with better approximation of supplier behavior, more suppliers stay in the system, waiting for more offers. Pricing may therefore not only be a tool for increasing supplier participation but also for increasing the number of requests (e.g. via vouchers) and, consequently, keeping the platform's two sides more balanced,

and the suppliers happy. Finally, while we focused on crowdsourced peer-to-peer transportation, uncertainty in supplier (or company driver) behavior is also present in other areas, e.g., when delivery drivers take their break or where they route and park their vehicles, how comfortable they are when routing certain areas, etc. Future work could also extend to other workforce scheduling problems in which even company employees may choose between different workshifts or tasks and this decision making behavior is uncertain to the scheduler, e.g., traveling nurses. Future work addressing salient features of these kinds of problems may build on our model and methodology, as well as the insights derived from the experiments.

Acknowledgements

We would like to thank Guido Voigt and Sören Köcher for their help on utility modeling with bounded rationality. We also thank Lei Zhao for his advice on learning in transportation. This work was partially funded by the US National Science Foundation CAREER award number 1751801. Marlin Ulmer's work is funded by the DFG Emmy Noether Programme, project 444657906. We gratefully acknowledge their support.

References

Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: a review. *European Journal of Operational Research*, 223(2):295–303, 2012.

Lina Al-Kanj, Warren B Powell, and Belgacem Bouzaiene-Ayari. The information-collecting vehicle routing problem: Stochastic optimization for emergency storm response. *arXiv preprint arXiv:1605.05711*, 2016.

Claudia Archetti, Martin Savelsbergh, and M Grazia Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254:472–480, 2016.

Alp M Arslan, Niels Agatz, Leo Kroon, and Rob Zuidwijk. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53:222–235, 2019.

Ramon Auad, Alan Erera, and Martin Savelsbergh. Courier satisfaction in rapid delivery systems using dynamic operating regions. *Available at SSRN 4089529*, 2021.

Rosemonde Ausseil, Jennifer A Pazour, and Marlin W Ulmer. Supplier menus for dynamic matching in peer-to-peer transportation platforms. *Transportation Science*, 2022.

Steffen J Bakker, Akang Wang, and Chrysanthos E Gounaris. Vehicle routing with endogenous learning: Application to offshore plug and abandonment campaign planning. *European Journal of Operational Research*, 289(1):93–106, 2021.

Hamsa Bastani, Pavithra Harsha, Georgia Perakis, and Divya Singhvi. Learning personalized product recommendations with customer disengagement. *Manufacturing & Service Operations Management*, 2021.

Moritz Behrend and Frank Meisel. The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? *Transportation Research Part B: Methodological*, 111:227 – 243, 2018.

Moritz Behrend, Frank Meisel, Kjetil Fagerholt, and Henrik Andersson. An exact solution method for the capacitated item-sharing and crowdshipping problem. *European Journal of Operational Research*, 279: 589–604, 2019.

Adam Behrendt, Martin Savelsbergh, and He Wang. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Transportation Science*, 2022.

Junyu Cao, Mariana Olvera-Cravioto, and Zuo-Jun Shen. Last-mile shared delivery: A discrete sequential packing approach. *Mathematics of Operations Research*, 45:1466–1497, 2020.

Xinyu Cao and Juanjuan Zhang. Preference learning and demand forecast. *Marketing Science*, 40(1):62–79, 2021.

Vincent E Castillo, Diane A Mollenkopf, John E Bell, and Terry L Esper. Designing technology for on-demand delivery: The effect of customer tipping on crowdsourced driver behavior and last mile performance. *Journal of Operations Management*, 2022.

Francisco Castro, Hamid Nazerzadeh, and Chiwei Yan. Matching queues with reneging: a product form solution. *Queueing Systems*, 96(3):359–385, 2020.

Chao Chen, Sen Yang, Yasha Wang, Bin Guo, and Daqing Zhang. Crowdexpress: a probabilistic framework for on-time crowdsourced package deliveries. *IEEE Transactions on Big Data*, 2020.

Xi Chen, Barrett W Thomas, and Mike Hewitt. Multi-period technician scheduling with experience-based service times and stochastic customers. *Computers & Operations Research*, 82:1–14, 2017.

Xi Chen, Jianjun Gao, Dongdong Ge, and Zizhuo Wang. Bayesian dynamic learning and pricing with strategic customers. *Production and Operations Management*, 2022.

Catherine Cleophas, Caitlin Cottrill, Jan Fabian Ehmke, and Kevin Tierney. Collaborative urban transportation: Recent advances in theory and practice. *European Journal of Operational Research*, 273: 801–816, 2019.

Cody Cook, Rebecca Diamond, Jonathan V Hall, John A List, and Paul Oyer. The gender earnings gap in the gig economy: Evidence from over a million rideshare drivers. *The Review of Economic Studies*, 88 (5):2210–2238, 2021.

Iman Dayarian and Martin Savelsbergh. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29:2153–2174, 2020.

Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: the state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013.

Katarzyna Gdowska, Ana Viana, and João Pedro Pedroso. Stochastic last-mile delivery with crowdshipping. *Transportation Research Procedia*, 30:90–100, 2018.

Hannah Horner, Jennifer A Pazour, and John E Mitchell. Optimizing driver menus under stochastic selection behavior for ridesharing and crowdsourced delivery. *Transportation Research Part E: Logistics and Transportation Review*, 153:102419, 2021.

Yixiao Huang, Lei Zhao, Warren B Powell, Yue Tong, and Ilya O Ryzhov. Optimal learning for urban delivery fleet allocation. *Transportation Science*, 53(3):623–641, 2019.

Huan Jin, Mike Hewitt, and Barrett W Thomas. Workforce grouping and assignment with learning-by-doing and knowledge transfer. *International Journal of Production Research*, 56(14):4968–4982, 2018.

Nicholas D Kullman, Martin Cousineau, Justin C Goodson, and Jorge E Mendoza. Dynamic ride-hailing with electric vehicles. *Transportation Science*, 56(3):775–794, 2022.

Yanzhe Murray Lei, Stefanus Jasin, Jingyi Wang, Houtao Deng, and Jagannath Putrevu. Dynamic workforce acquisition for crowdsourced last-mile delivery platforms. *SSRN*, 2020.

Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, 113:104806, 2020.

Seyed Shahab Mofidi and Jennifer A Pazour. When is it beneficial to provide freelance suppliers with choice? A hierarchical approach for peer-to-peer logistics platforms. *Transportation Research Part B: Methodological*, 126:1–23, 2019.

Abood Mourad, Jakob Puchinger, and Chengbin Chu. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, 123:323–346, 2019.

Mila Nambiar, David Simchi-Levi, and He Wang. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11):4980–5000, 2019.

Santiago Nieto-Isaza, Pirmin Fontaine, and Stefan Minner. The value of stochastic crowd resources and strategic location of mini-depots for last-mile delivery: A benders decomposition approach. *Transportation Research Part B: Methodological*, 157:62–79, 2022.

W. Powell. *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. John Wiley and Sons, Hoboken, NJ, USA, 2022.

Warren B Powell and Ilya O Ryzhov. *Optimal learning*, volume 841. John Wiley & Sons, 2012.

Zhiwei Qin, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, and Jieping Ye. Ride-hailing order dispatching at DiDi via reinforcement learning. *INFORMS Journal on Applied Analytics*, 50(5): 272–286, 2020.

Heleen Buldeo Rai, Sara Verlinde, Jan Merckx, and Cathy Macharis. Crowd logistics: an opportunity for more sustainable urban freight transport? *European Transport Research Review*, 9(3):39, 2017.

Alberto Santini, Ana Viana, Xenia Klimentova, and João Pedro Pedroso. The probabilistic travelling salesman problem with crowdsourcing. *Computers & Operations Research*, 142:105722, 2022.

Martin WP Savelsbergh and Marlin W Ulmer. Challenges and opportunities in crowdsourced delivery planning and operations. *4OR*, 20(1):1–21, 2022.

Jørgen Skålnes, Lars Dahle, Henrik Andersson, Marielle Christiansen, and Lars Magnus Hvattum. The multistage stochastic vehicle routing problem with dynamic occasional drivers. In *International Conference on Computational Logistics*, pages 261–276. Springer, 2020.

Ninja Soeffker, Marlin W. Ulmer, and Dirk C. Mattfeld. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, 298(3):801–820, 2022.

Amirmahdi Tafreshian, Neda Masoud, and Yafeng Yin. Frontiers in service science: Ride matching for peer-to-peer ride sharing: A review and future directions. *Service Science*, 12(2-3):44–60, 2020.

Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

Marlin Ulmer and Martin Savelsbergh. Workforce scheduling in the era of crowdsourced delivery. *Transportation Science*, 54:1113–1133, 2020.

Marlin Ulmer, Maciek Nowak, Dirk Mattfeld, and Bogumił Kaminski. Binary driver-customer familiarity in service routing. *European Journal of Operational Research*, 286(2):477–493, 2020.

Marlin W. Ulmer, Dirk C. Mattfeld, and Felix Köster. Budgeting time for dynamic vehicle routing with stochastic customer requests. *Transportation Science*, 52(1):20–37, 2018.

Silviya Valeva, Mike Hewitt, and Barrett W Thomas. A matheuristic for workforce planning with employee learning and stochastic demand. *International Journal of Production Research*, 55(24):7380–7397, 2017.

Wouter JA van Heeswijk, Martijn RK Mes, and Johannes MJ Schutten. The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science*, 53(1):203–221, 2019.

Hai Wang and Hai Yang. Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological*, 129:122–155, 2019.

Qiaochu Wang, Yan Huang, Stefanus Jasin, and Param Vir Singh. Algorithmic transparency with strategic users. *Management Science*, 2022.

Baris Yildiz and Martin Savelsbergh. Service and capacity planning in crowd-sourced delivery. *Transportation Research Part C: Emerging Technologies*, 100:177 – 199, 2019.

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274