

Supplier Menus for Dynamic Matching in Peer-to-Peer Transportation Platforms

Rosemonde Ausseil, Jennifer A. Pazour

Industrial and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, ausser@rpi.edu, pazouj@rpi.edu

Marlin W. Ulmer

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

Peer-to-peer transportation platforms dynamically match requests (e.g., a ride, a delivery) to independent suppliers who are not employed nor controlled by the platform. Thus, the platform cannot be certain that a supplier will accept an offered request. To mitigate this selection uncertainty, a platform can offer each supplier a menu of requests to choose from. Such menus need to be created carefully because there is a trade-off between selection probability and duplicate selections. In addition to a complex decision space, supplier selection decisions are vast and have systematic implications, impacting the platform’s revenue, other suppliers’ experiences (in the form of duplicate selections) and the request waiting times. Thus, we present a multiple scenario approach, repeatedly sampling potential supplier selections, solving the corresponding two-stage decision problems, and combining the multiple different solutions through a consensus algorithm. Extensive computational results using the Chicago Region as a case study illustrate that our method outperforms a set of benchmark policies. We quantify the value of anticipating supplier selection, offering menus to suppliers, offering requests to multiple suppliers at once, and holistically generating menus with the entire system in mind. Our method leads to more balanced assignments by sacrificing some “easy wins” towards better system performance over time and for all stakeholders involved, including increased revenue for the platform, and decreased match waiting times for suppliers and requests.

Key words: peer-to-peer transportation, dynamic matching, supplier-side choice, multiple scenario approach, ride-sharing

Author copy of <https://doi.org/10.1287/trsc.2022.1133>

1. Introduction

Online platforms are an emerging transportation business model that enables individual customers to request a service (e.g., a ride, a delivery) from independent suppliers (e.g., drivers) who are not employed by the platform. In this work, we focus on transportation platforms wanting to tap into underutilized transportation capacity (e.g., the empty seats and trunk spaces) in the vehicles already on the roads. That is, we focus on outsourcing goods or person transportation to occasional carriers drawn from existing travelers (Punel and Stathopoulos 2017). This is in contrast

to traditional transportation companies which commit to resources in advance (e.g., owning/leasing physical assets like vehicles and employing a staff of full time drivers), as well as platforms like Uber, Lyft, and Grubhub, whose drivers work longer horizons with several transportation assignments not interconnected with their personal travel. In this work we use the term *request* to denote a need for transportation services and *supplier* to denote someone willing to adjust their own planned travel to provide at most one transportation service in return for compensation by the platform.

The main role of these online platforms — which own no physical assets nor employ full time operators — is to facilitate matching of spontaneous transportation requests with independent suppliers dynamically over the course of the day. Because suppliers are not employed by the platform, the platform does not have a perfect understanding of a supplier’s preferences for which requests they would like to service (if any). The platform needs to account for this novel type of uncertainty in their decisions to avoid offering requests a supplier declines to serve, as such would lead to no platform revenue and two unhappy participants. To keep the system simple and attractive for suppliers and requests, we also do not want to burden the suppliers too much by requiring them to sort through all available requests nor require them to provide and update time consuming information (like preferences or bids). Thus, we propose offering a *menu*, a set of carefully selected requests, to each supplier instead.

When creating menus for the suppliers in the system, the platform faces several challenges. First, the set of potential menus is vast, and searching this vast space in real-time is complex. Second, the supplier selections are uncertain and have systematic implications due to the experience of the stakeholders being interdependent. For example, two suppliers selecting the same request results in at least one unhappy supplier. This uncertainty and interdependency lead to a tradeoff when creating menus. Increasing menu sizes can increase the probability that a supplier selects a request. However, increased menu size and overlaps between menus also lead to less systematic control by the platform, which has the danger of duplicate selections, fewer successful matches, and more unhappy participants.

Addressing these challenges, this work contributes by explicitly modeling both the platform decisions and the suppliers’ choices in a dynamic setting. Among the first to consider the interactions of dynamic matching, supplier choices, and menus, we present a comprehensive dynamic decision model for this novel and important problem type, and solve it using a multiple scenario approach (MSA). The general idea of MSA-methods is to reduce the stochastic dynamic problem to a set of deterministic scenarios via sampling, then solving the individual scenarios via mixed-integer programming, and finally deriving a consensus solution from the individual solutions (Bent and Van Hentenryck 2004). For our problem, we sample scenarios of supplier selection preferences, and solve the associated deterministic two-stage programs via CPLEX to optimality. To account for

the tradeoff of selection probability and duplicate selections, we add two additional artificial constraints with respect to maximum menu size and maximum overlap. Based on the menus created in each scenario, we derive a consensus of menus that we offer to the suppliers. We further present an upper bound solution to assess the absolute performance of our policy. Tailored to the new challenge of supplier-side choice, our MSA provides excellent results compared to benchmark policies - with and without menus - evaluated via a comprehensive case study based on data from the city of Chicago. Our method achieves statistically better results than the benchmarked approaches and allows for real-time decision making in our experiments for up to 100 requests and 100 suppliers in the system simultaneously. A comprehensive analysis of our method's components in different transportation settings and supplier behavior quantifies the value of anticipating supplier selection, offering menus to suppliers, offering requests to multiple suppliers at once, holistically generating menus with the entire system in mind, and learning supplier preferences. We further derive the following managerial insights:

- The MSA creates menus that not only increase platform revenue, but also increase the number of successful matches and reduce the waiting time for suppliers and requests. This is important for long-term participation and success of the platform (Ermagun and Stathopoulos 2018).
- Our MSA approach balances revenue more evenly over time and over the set of suppliers compared to menu creation for individual suppliers (that leads to “cherry picking”). Our approach creates menus that are better for the platform, suppliers, and requests as a whole.
- Having menus is particularly valuable when supplier behavior is highly uncertain and/or when suppliers are picky in their selections.
- Menus that are too large can be counterproductive because they lead to less controllable outcomes, e.g., a higher number of duplicate selections and supplier selections that have less benefit to the platform.
- While request overlaps may cause occasional duplicate selections, they are essential when offering menus and eventually lead to better system performance for the suppliers.
- Our method is beneficial in both the case of suppliers indicating their top choice request from their menu as well as the case of suppliers indicating all the requests in their menu they are willing to serve (see Appendix A.3.1).

After conducting a literature review in Section 2, we describe our problem with an illustrative example and a mathematical model in Section 3. Then, we present our multiple scenario approach in Section 4, followed by our experimental set up and the description of our benchmark policies in Section 5. Finally, in Sections 6 and 7, we explore and discuss the results of our computational studies and present future research directions.

2. Literature Review

This research contributes to the problem domain of peer-to-peer logistics services, which include crowdsourced delivery and ride sharing. Three major decisions are available when optimizing such services, and include order dispatching (also known as matching), driver repositioning, and pricing (Qin et al. 2020). We focus on matching occasional drivers (or suppliers) with demand requests, which is part of a fast growing field of research: matching crowdsourced suppliers; see reviews including Agatz et al. (2012), Cleophas et al. (2019), Furuhashi et al. (2013), Mourad, Puchinger, and Chu (2019), Rai et al. (2017), Tafreshian, Masoud, and Yin (2020), Wang and Yang (2019). As illustrated in Figure 1, this work contributes to the problem domain of matching problems in crowdsourced transportation problems through integration: As the majority of the literature either ignores supplier selection behavior or assumes supplier behavior can be determined a priori (e.g., all suppliers will always accept a demand request as long as it is within a given distance), this work contributes by explicitly modeling both the platform decisions and the suppliers' choices in a dynamic setting. Next, we review research at the three intersections, beginning with matching crowdsourced suppliers in a dynamic system, then work explicitly modeling suppliers' decision making, and finally work providing demand-side choice (e.g., offering a passenger a menu of transportation modes to choose from).

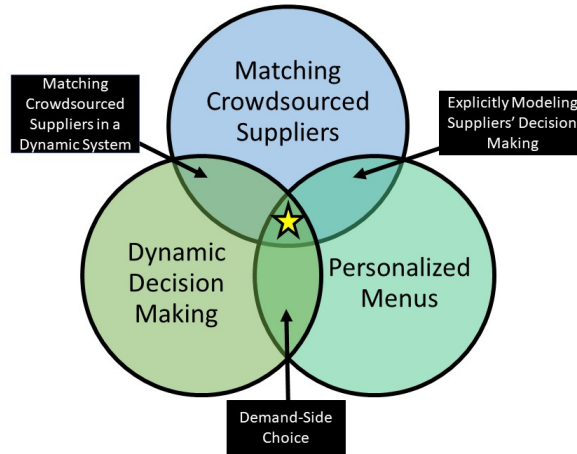


Figure 1 This work's contribution is in explicitly capturing dynamic decisions by both the platform and the suppliers, who get to make selections from personalized menus of transportation requests.

2.1. Matching Crowdsourced Suppliers in a Dynamic System

The vast majority of models matching supply and demand for dynamic peer-to-peer logistics services take a centralized approach, solving a single-level optimization model assuming suppliers comply with assignments made by the platforms, e.g., Arslan et al. (2018), Dayarian and Savelsbergh (2020), Lee and Savelsbergh (2015), Li et al. (2014), Liu and Li (2017), Pelzer et al. (2015),

Reyes et al. (2018), Yu and Shen (2019). Another way is to deploy a given matching policy (e.g., first come first serve, closest supplier, or more sophisticated policies via dynamic matching control problems) to sequentially arriving requests (Özkan and Ward 2020).

While a fundamental property of peer-to-peer logistics systems is that the suppliers are not employees of the platform, only limited research considers decision autonomy in supplier behavior. These include Wang, Agatz, and Erera (2018), which is a centralized optimization approach enforcing dynamic ride-sharing matches to be stable that assumes the platform's and suppliers' preferences are known a priori. This work also assumes all suppliers will accept matches as long as they are stable. Other works enforce different constraints on the types of requests offered to suppliers (e.g., maximum detour or length of trip), but similarly assume as long as these constraints are met, all suppliers will accept all requests offered, e.g., Arslan et al. (2018), Archetti, Savelsbergh, and Speranza (2016). Others require suppliers to provide preferences for all requests in advance, e.g., Kaffle, Zou, and Lin (2017). More recently, a body of literature captures that both supply and demand are stochastic in a dynamic system. For example, Qin et al. (2020) describe methodologies combining combinatorial optimization models, semi-Markov decision processes, and deep reinforcement learning to match trip requests to suppliers at massive scale in dynamic and stochastic environments. Like our approach, their dynamic approach relies on making matching decisions at discrete time intervals. Yet, none of these papers provide suppliers with a selection of requests and then allow the suppliers to choose. And thus, none model supplier selection decisions directly in their optimization methodologies.

2.2. Explicitly Modeling Suppliers' Decision Making

One way to provide suppliers with decision making autonomy is to use a decentralized dispatching approach, in which agents (either demand requests or suppliers) make decisions without coordinating with a centralized decision maker, e.g., Nourinejad and Roorda (2016), Sánchez, Martínez, and Domingo-Ferrer (2016), but those approaches are time-consuming for suppliers. To alleviate the time commitments of suppliers, Di Febbraro, Gattorna, and Sacco (2013), Soto Setzke et al. (2017) and Zhao et al. (2019) explicitly capture a supplier's ability to reject an offered request, yet, they do not consider menus nor do they capture any interdependencies linked to the multiple suppliers' simultaneous decisions. Similarly, Powell, Towns, and Marar (2000) model that a single request is recommended to a single dispatcher to accept or reject. The acceptance probability is represented by an exogenous random variable, and so choice is not modeled explicitly in the optimization model, and because there is only a single dispatcher, performance dependencies due to multiple suppliers' decisions are not considered. Equilibrium models capture supply and demand side behaviors at an aggregate level, e.g., Nourinejad and Ramezani (2020). While related, equilibrium models do

not consider a supplier selection process, only the ability to reject a recommended request. Also, their focus is strategic and not on the operational problem of matching specific requests to specific suppliers by providing personalized menus.

We are aware of only two papers that create optimization frameworks to provide a menu of requests to suppliers to select from, and both consider a single period problem. Mofidi and Pazour (2019) is the first work to study how personalized recommendations should be made to a set of suppliers to coordinate decentralized resource allocation decisions. They develop a bi-level optimization model, capturing the platform first decides the composition of multiple, simultaneous personalized recommendations, and then suppliers select from this set. Assuming deterministic selection behavior by suppliers, an equivalent single-level reformulation technique is developed, which we adopt and embed in our methodology. By simulating different menu sizes and selection behaviors, they computationally illustrate that providing suppliers with choice can be beneficial if a platform has uncertainty about suppliers' selections. Horner, Pazour, and Mitchell (2021) extend the bi-level model to capture stochastic selection behaviors directly in the optimization model and develop a Sample Average Approximation technique to solve a special case where suppliers signal their willingness to serve requests recommended via a menu.

2.3. Demand-Side Choice

Related is work that integrates demand-side choice into modeling of ride-sharing platforms. For example, research has modeled riders' selection from a set of transportation modes (Liu et al. 2019, Atasoy et al. 2015), among different ridesharing drivers (Zhang and Zhao 2018; Lei, Jiang, and Ouyang 2020), and from solo and pooled ride options (Jacob and Roet-Green 2021). In supply-side choice, decisions must take into account that ultimately one (and only one) supplier can fulfill each request (i.e., enforcing a one-to-one matching constraint) and the implications of violating this constraint. This is in contrast to demand-side choice models, in which capacity of the modes are typically ignored and thus, no penalty is concerned with multiple riders choosing the same option. Therefore, the type of decisions vary (for example, Liu et al. (2019) integrate Bayesian Optimization to find fleet size and fares), whereas we are interested in menu composition.

2.4. Summary

In summary, the use of supply-side menus for matching transportation requests is limited. Furthermore, all existing work to explicitly model supply-side choice in an optimization formulation consider a single period problem. In contrast, we focus on supply-side choice in a dynamic setting over multiple periods, capturing how menu decisions over time are influenced with stochastically arriving and leaving suppliers and requests. Modeling the problem as a dynamic problem allows us to capture more broadly the stakeholder experiences, specifically, the average waiting time in the

system for requests and suppliers, the platform’s revenue over time, the percentage of unassigned suppliers and requests over time, the disruptiveness of selections, and the impact of longer-term anticipation. Methodologies to anticipate arrivals of future demand and incorporate this information are common in a variety of problem domains, whereas the need to incorporate uncertainty in the availability of both supply and demand is an emerging field (see Ulmer et al. (2020) for a recent overview). However, none of the dynamic papers explicitly allow for suppliers to have choice, and thus, none of their developed methodologies explicitly anticipate suppliers’ choice to participate.

3. Problem Description

We consider a platform for dynamically matching transportation requests to occasional drivers (termed suppliers), who have their own planned travel. Consequently, we assume that each supplier extends their planned travel to serve at most one request. Both transportation requests and suppliers enter the system over the course of the day spontaneously and are unknown beforehand. Each request consists of a pickup origin, a delivery destination, and a fee to pay if they are matched. The fee is a combination of a fixed fee equal for every request plus an additional travel fee dependent on the distance between origin and destination (which will go directly to the matched supplier). Suppliers have planned transportation of their own, and each arriving supplier provides their current location (i.e., their origin) and their planned destination to the platform. Thus, the platform knows the origin and destination of each request and supplier in the system.

In each equidistant time step (e.g., a minute long), there is a menu offering window, a selection window, and an assignment window. Such discrete decision epochs are common in practice for ridesharing, e.g., Uber (2021) and Didi (Qin et al. 2020), and in the literature, e.g., Di Febbraro, Gattorna, and Sacco (2013) and Lei, Jiang, and Ouyang (2020). During the menu offering window, the platform offers each supplier currently in the system a personalized *menu* (i.e., subset) of requests also currently in the system (we consider an alternative platform protocol in Appendix A.3.3 where suppliers provide to the platform their preferences on all requests in the system; while valuable, that protocol is likely not practical for systems of realistic size). The menus may not cover all requests, and they may overlap (i.e., a request may be offered to multiple suppliers simultaneously). To avoid suppliers’ inconvenience, there is a maximum menu size, which limits the number of requests recommended to each supplier. Once the offers are made, the selection window begins in which each supplier selects the request from their menu (if there are any) they most want to fulfill (i.e., suppliers are assumed rational). We also consider an alternative approach for supplier selections, in which suppliers select any and all requests in their menu they are willing to serve (see Appendix A.3.1 that shows that our MSA method is beneficial for both supplier selection approaches). All suppliers have the ability to not select any requests and instead

can wait for future offers or can take their planned travel without helping to fulfill a request. This selection window is a pre-defined amount of time given to all suppliers, and if a supplier does not respond back within this time, the platform assumes they did not want to participate in this time step. If a supplier is assigned a request to fulfill, the supplier's payment accounts for their trip from their origin to the request's origin, and then from the request's origin to the request's destination. After arriving to the request's destination, payment stops. Thus, the supplier is not paid to drive to their planned personal destination, and as a consequence, suppliers tend to select requests that align well with their remaining personal travel (from the request drop-off destination to their own planned destination). However, these supplier participation decisions and request selections are uncertain to the platform. For example, some suppliers may be more or less flexible in the types of requests or areas in the city they are willing to serve; also, some may prefer serving requests with smaller detour distances, whereas others try avoiding traffic congestion and/or long travel times. While the platform does not have exact information about the suppliers' preferences nor when suppliers and requests will enter or leave the system, the platform does have access to historical data and can make predictions. For example, the platform can estimate the time suppliers and requests are willing to stay in the system before being matched, aggregate supplier behavior (e.g., parametrization of the suppliers' choice models), and probability distributions for future arrivals of requests and suppliers.

After providing each supplier with a menu and observing the suppliers' actual preferences and then selections, the assignment window begins where the platform performs assignments. The overall goal of the platform is to maximize the expected reward of successful matches. Assigned pairs of suppliers and requests leave the system and provide a reward to the platform. Because the travel fee goes directly to the supplier, the platform's reward is the fixed fee minus any additional compensation for the supplier. This compensation depends on the "empty" travel from the supplier's origin to the request's origin. Between the assignment and the next decision time step, new suppliers and requests may enter the system, and others may leave because they are unhappy from failed attempts of being matched (due to menu rejection or selection without assignment). As suppliers get increasingly unhappy with every time step they do not end up getting matched, and because we model a supplier staying in the system only for a limited number of time steps, this results in lower chances a supplier will stay in the system with each offered menu attempt by the platform.

3.1. Illustrative Example

To illustrate our problem statement and model, we provide an example for a single decision state. The example, shown in Figure 2, depicts the sequence of a state, a menu offering, a set of selections by the suppliers, an assignment by the platform, and a transition to the next state.

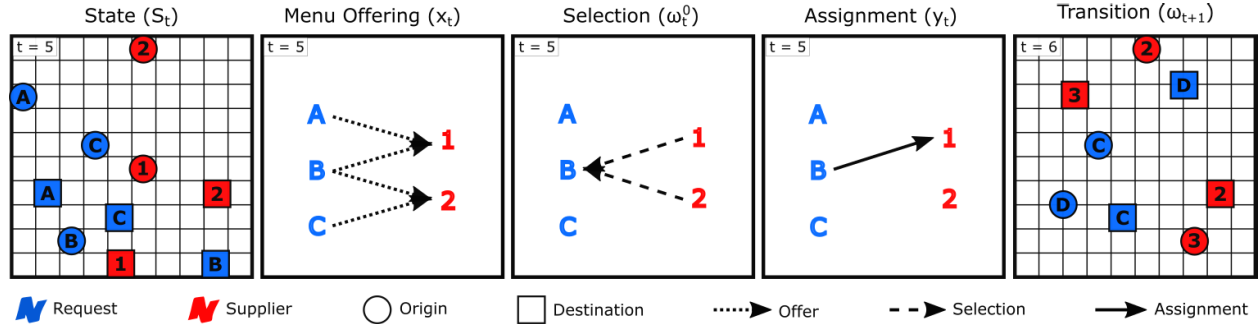


Figure 2 Example for a decision sequence.

We are at time step $t = 5$. The state, depicted on the left, contains a set of two suppliers, 1 and 2, as well as three requests, A, B, and C. The geographical locations of the origins and destinations of requests and suppliers are depicted by circles and squares. In this example, they are connected by a Manhattan-style road network with travel time of 5 minutes per segment. The platform's fixed revenue for matching is 10 revenue units; thus, the platform's reward for a match is 10 minus the supplier's travel from their origin to the matched request's origin, which is paid to the supplier at a rate of 1 revenue unit per 5 minutes travel time.

Based on the current system state, the platform offers menus to the two suppliers as shown in the second box of Figure 2. In this example, the menu for supplier 1 contains requests A and B. The menu for supplier 2 contains requests B and C. Based on the suppliers' preferences, which are not fully known to the platform at the menu creation stage, each of the suppliers decide which preferred request to select (if any) from those offered in their menus. This is shown in the third box, denoted as Selection, and in this example, both suppliers select request B. Based on observed supplier selections, the platform then assigns requests to the suppliers. In this example, request B is assigned to supplier 1, and a reward of $10 - 6 = 4$ is generated for the platform. Both supplier 1 and request B are successfully assigned and leave the system. After the assignment, a transition leads to a new state in time step $t = 6$. In the example, the transition changes the suppliers, by adding a new supplier 3. Requests change by removing request A (the corresponding request decided not to wait any longer) and by adding a new request D.

3.2. Mathematical Model

Here, we present the mathematical model by means of a sequential decision process. The problem is special, because we observe two types of stochastic information at different times: the suppliers' selections and the changes in suppliers and requests. At every time step in the sequential decision problem, the platform faces a two-stage decision problem: first, what menus to offer, and after the suppliers make their selections, what to assign. This is reflected in the model, which has two subsequent decisions per decision point t , an offering decision x_t and an assignment decision y_t .

States. The model comprises a set of subsequent decision points $t = 1, \dots, T$. At each point t , a decision state $S_t = (s_t, r_t)$ is observed. A state consists of the following elements:

- Set of suppliers $s_t = \{s_{1t}, \dots, s_{m_t t}\}$ with m_t denoting the number of suppliers in state S_t . Each supplier s has an entering time τ^s , an origin o^s , and a destination d^s .
- Set of requests $r_t = \{r_{1t}, \dots, r_{n_t t}\}$ with n_t denoting the number of requests in state S_t . Each request r has an entering time τ^r , an origin o^r and a destination d^r .

The initial state is $S_0 = (\emptyset, \emptyset)$. The final state is $S_T = (\emptyset, \emptyset)$.

Decisions. In every state S_t , a first decision x_t is selected, which is offering requests to suppliers. Decision x_t is a binary matrix, with $x_t \in M^{n_t, m_t}$. A decision value is $x_{ijt} = 1$, if request r_{it} is offered to supplier s_{jt} ; else, it is $x_{ijt} = 0$. A decision is feasible, if the menu size for each supplier is smaller than the maximum menu size b :

$$\sum_{i=1}^{n_t} x_{ijt} \leq b \quad \forall j = 1, \dots, m_t. \quad (1)$$

After a decision x_t is selected, the state is transferred to post-decision state $S_t^x = (S_t, x_t)$ and the first type of stochastic information is observed, ω_t^o , reflecting the selection of the suppliers based on the offered requests in S_t^x . Suppliers are rational and select the request in their menu with the highest utility value (which could be the no choice option). The value for supplier s_{jt} of serving request r_{it} is denoted v_{ijt} , with v_{0jt} being the utility of not serving any request (i.e., *no choice threshold*).

The utility values are not revealed to the platform, only the eventual selection of a supplier is. Thus, the stochastic information of the selection is also modeled as a binary matrix, $\omega_t^o \in M^{n_t, m_t}$ with values of 0 and at most one value of 1 in each column. Value $\omega_{ijt}^o = 1$ indicates that the utility value v_{ijt} was highest amongst all offered values and the no choice threshold value. Thus, supplier s_{jt} is willing to serve request r_{it} . If not, $\omega_{ijt}^o = 0$. Because suppliers can only select offered requests, $\omega_{ijt}^o \leq x_{ijt}$ for all r_{it} and s_{jt} . Further, every supplier only selects at most one request, thus:

$$\sum_{i=1}^{n_t} \omega_{ijt}^o \leq 1 \quad \forall j = 1, \dots, m_t. \quad (2)$$

The combination of decisions x_t and stochastic information ω_t^o leads to a state augmented by information ω_t^o . We denote this state $S_t^o = (S_t^x, \omega_t^o)$. For this augmented state, the second platform decisions, i.e., the assignment decisions y_t , assign requests to suppliers. This is done via a binary matrix $y_{ijt} \in M^{n_t, m_t}$. Value $y_{ijt} = 1$ represents an assignment of request r_{it} to supplier s_{jt} , else $y_{ijt} = 0$. An assignment is feasible if the following conditions hold:

$$\sum_{i=1}^{n_t} y_{ijt} \leq 1 \quad \forall j = 1, \dots, m_t$$

$$\begin{aligned} \sum_{j=1}^{m_t} y_{ijt} &\leq 1 & \forall i = 1, \dots, n_t \\ y_{ijt} &\leq \omega_{ijt}^o & \forall i = 1, \dots, n_t, j = 1, \dots, m_t. \end{aligned}$$

That is, every supplier is assigned at most one request; every request is assigned to at most one supplier; and assignments can only be made if a supplier selected the corresponding request. The revenue of assigning request r_{it} to supplier s_{jt} is denoted as c_{ijt} . The platform reward of assignment decision y_t is the sum of these revenues:

$$R(S_t^o, y_t) = \sum_{i=1}^{n_t} \sum_{j=1}^{m_t} c_{ijt} y_{ijt}. \quad (3)$$

A decision changes the sets of requests and suppliers by removing successfully assigned suppliers and requests. The updated sets are denoted r_t^y and s_t^y . We denote the post-decision state after the assignment decision as S_t^y .

Transition. The transition from post-decision state S_t^y depends on the realization of changes in requests and suppliers. These changes are provided by realization $\omega_{t+1} = (\omega_{t+1}^r, \omega_{t+1}^s)$. Realization ω_{t+1} contains the following information:

- New requests $r^+(\omega_{t+1})$, leaving requests $r^-(\omega_{t+1})$
- New suppliers $s^+(\omega_{t+1})$, leaving suppliers $s^-(\omega_{t+1})$

The combination of post-decision state S_t^y and stochastic information ω_{t+1} is transferred to a new state S_{t+1} by the transition function $W(S_t^y, \omega_{t+1})$ as follows:

- New time $t + 1$
- Requests: $r_{t+1} = r_t^y \setminus r^-(\omega_{t+1}) \cup r^+(\omega_{t+1})$
- Suppliers: $s_{t+1} = s_t^y \setminus s^-(\omega_{t+1}) \cup s^+(\omega_{t+1})$

Objective Function. A solution for the problem is a two-part decision policy $\pi = (\pi^x, \pi^y)$. A policy π maps each state S_t to an offering decision $\mathcal{X}^{\pi^x}(S_t)$ and each resulting augmented state S_t^o to an assignment decision $\mathcal{X}^{\pi^y}(S_t^o)$. The objective is to find a policy π^* maximizing the expected revenue:

$$\pi^* = \arg \max_{\pi = (\pi^x, \pi^y)} \mathbb{E} \left[\sum_{t=0}^T R(S_t^o, \mathcal{X}^{\pi^y}(S_t^o)) | \pi^x, S_0 \right]. \quad (4)$$

The optimal policy is the policy that maximizes the expected overall reward when starting in the initial state S_0 and applying the policy (π^x, π^y) throughout the problem horizon.

4. A Multiple Scenario Approach

In this section we present our solution method, a multiple scenario approach. We start with a short motivation and overview, and then describe in detail the individual steps of our approach.

4.1. Motivation

Solving the sequential decision problem presented in Section 3 to optimality is not possible for instances of reasonable size because of the infamous curse of dimensionality. As for many dynamic problems in urban logistics and transportation, the state space is vast because of the manifold potential locations in the city. However, for our problem, we have two additional challenges, the vast decision space and the stochastic supplier selection.

The decision space is vast because it grows substantially with the number of requests n_t and suppliers m_t in the system. For example, if, in state S_t , each of the m_t suppliers are offered menus of maximum size b , the set of potential menus is $\binom{n_t}{b}^{m_t}$. Searching this vast decision space is already challenging. Moreover, finding effective menus is difficult for an additional reason, which is the tradeoff between encouraging supplier participation and duplicate selections. Duplicate selections result from overlapping menus and are undesirable for two reasons. First, they block at least one successful assignment in the current period, which reduces the revenue the platform can obtain. Second, they can lead to dissatisfaction for suppliers, if their selections are frequently disregarded. However, non-overlapping menus may also be counterproductive because they reduce the probability that a supplier selects a request from their menu; thus, if a request is only offered to a single supplier each time period, the waiting time for requests to be matched can increase. Notably, duplicate selections and non-selected requests are a function of multiple supplier (simultaneous) selection behaviors. Therefore, menu decisions cannot be solely based on each individual supplier; instead, menus must be created holistically by considering the entire set of suppliers and requests.

In addition to the complexity in the decision space, our problem shows substantial uncertainty, not only in future requests and suppliers, but especially in the supplier selection behavior. Suppliers can select any request of their menu or none at all, making the holistic set of selections vast. For example, if, in state S_t , each of the m_t suppliers are offered menus of maximum size b , the set of potential selections is $(b + 1)^{m_t}$. The selections depend on manifold factors and are disruptive to the system, because, as aforementioned, they do not only impact the individual supplier, but the entire system as suppliers may select the same requests (see the example in Section 3.1) or a request may not be selected by any supplier, increasing request waiting time.

In essence, a suitable heuristic should be tractable in considering the vast decision space, consider the high and disruptive uncertainty by the suppliers' selections, and create corresponding, holistic menus that balance the tradeoffs among (1) increasing supplier selection, (2) decreasing rejected requests by all suppliers, and (3) decreasing duplicate selections. To address the two novel and complex challenges of interdependent menus and uncertain supplier selections, we present a multiple scenario approach (MSA) (Bent and Van Hentenryck 2004). The concept of MSAs has already shown much promise in different stochastic routing problems (see Song et al. (2020) for a recent

overview). While the proposed MSAs differ as they depend on the corresponding problems, they all follow a similar concept: the idea is to sample a set of scenarios and solve the corresponding static and deterministic problems individually. Then, a “consensus” solution is derived based on a holistic view of the set of individual solutions. In our MSA, we focus on sampling the supplier selections, as preliminary tests indicate that this selection is so disruptive that extending the time horizon of scenarios does not yield much benefit (see Online Appendix B.4). To account for the tradeoff between selection probability and duplicate selection, we limit the menu sizes beyond b and augment the problems solved in the individual scenarios with an artificial constraint, a maximum number of overlaps per request.

The detailed MSA-procedure for our problem is illustrated in Figure 3. At each time period t , our method consists of five steps, starting with scenario sampling of supplier properties; then, each scenario is solved using the augmented integer program (IP). The decision variable values of these separate IP’s are combined via a developed consensus solution. The fourth step (which is out of the control of the platform) is the actual suppliers’ selections. The final step is the assignment algorithm. We provide detail of the five steps in the remainder of this section.

4.2. Step 1 - Scenario Sampling

In our method, we sample multiple scenarios to capture the uncertainty of a supplier’s utility for a request. Multiple methods could be used by a platform to estimate a set of potential utility values (e.g., machine learning approaches, regression functions, multi-attribute utility functions). In our experiments, which focus on occasional suppliers with a planned destination, the platform estimates a supplier’s utility for a request to be based on two attributes and an external factor that captures any deviation not associated with those attributes. The two attributes are the travel distance and the travel time from the request’s destination to the supplier’s destination, which corresponds to the final leg of the supplier’s trip. Those two attributes were selected due to being common attributes found to influence participating in peer-to-peer logistics systems (travel distance e.g., Le et al. (2019); travel time, e.g., Ashkrof et al. (2020) and Ashkrof et al. (2021)) and due to availability of data. Each supplier has a preference for either one of the attributes, with varying weights. Because the suppliers and the requests announce their o-d pairs when they enter the platform, the platform has access to each suppliers’ values for each attribute, but has uncertainty about the suppliers’ preferences, i.e., the attribute weight values. Therefore, in Step 1, for each scenario σ , we sample attribute weight values that are then used by the platform to estimate a supplier’s utility value $\hat{v}_{ijt}^{\sigma}$. Different attribute weight values result in requests being ranked differently for each supplier. For example, in scenario 1 of Figure 3, the platform estimates that supplier 2 puts more weight on the trip’s distance than the trip’s time. This results in supplier 2 ranking request B first, as request

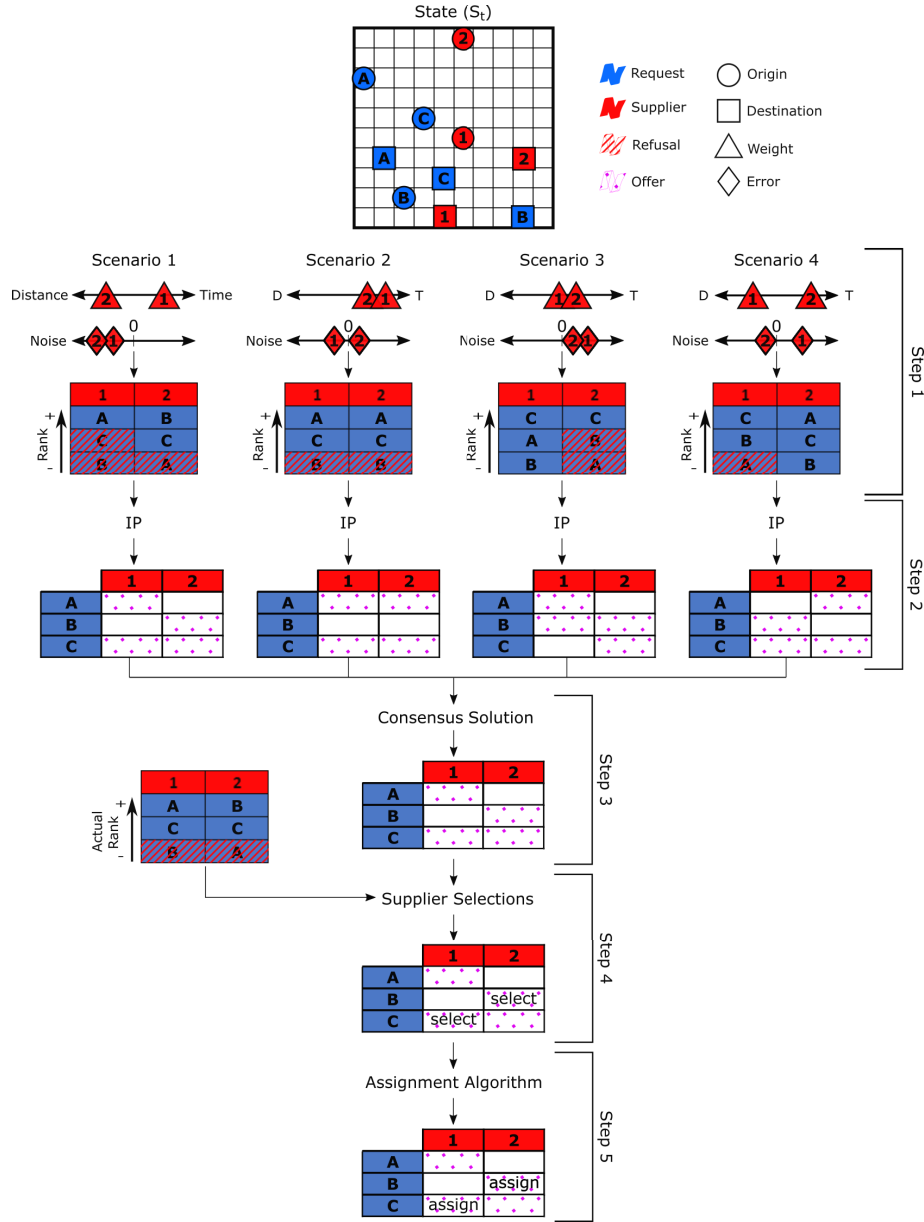


Figure 3 Multiple Scenario Approach Example.

B's destination is closest to request 2's destination, and request A is ranked last, as its destination is the furthest from request 2's destination. In contrast, when supplier 2 puts more weight on the trip's time (e.g., scenario 4), supplier 2's ranking of requests is different than the one in scenario 1.

In addition, a supplier may refuse to service requests below a certain utility value, e.g., requests whose destinations are too far away from the supplier's destination or are in an area with a lot of traffic. Thus, each scenario also samples a refusal or no choice value, below which a supplier will refuse to service any request. For example, in scenario 1, the platform estimates that supplier 2 will refuse requests whose distance is greater than the distance from request C's destination. The output of Step 1 is thus a set of estimated $\hat{v}_{ijt}$ values for all suppliers and requests in state S_t ,

including each supplier's no choice utility value. Each scenario σ represents a separate sampling of the attribute weight values and of the no choice utility value; thus, each scenario σ yields estimated utility values $\hat{v}_{ijt}^\sigma$.

4.3. Step 2 - Integer Program

In Step 2, we solve an integer program (IP) separately for each scenario σ 's generated utility values $\hat{v}_{ijt}^\sigma$ (obtained in Step 1). This IP explicitly models the hierarchical decision making structure of our problem: the platform first determines menus, and then from those menus, suppliers make their selections. For each (deterministic) scenario σ , we assume each supplier selects the request recommended with the largest $\hat{v}_{ijt}^\sigma$ value (which could be the no choice alternative). This premise means the platform knows, for a given scenario, which requests the suppliers will select, which allows us to transform the bi-level optimization model into an equivalent single-level IP. Since we are solving a deterministic problem, the expected value of the stochastic information ω_t^o is input in the IP as $\hat{\omega}_t$. To capture supplier uncertainty, we solve a separate IP for each scenario σ , and we differentiate the parameters and decision variables associated with that scenario with the superscript σ . To address the tradeoff between duplicate selections and selection probability, we add three modifications to the IP. To reduce the risk of duplicate selections, we limit the menu size for every supplier to $\bar{b} \leq b$ and add an artificial constraint limiting the number of overlaps per request to $\bar{a}$. To increase the selection probability while respecting these new constraints, we force the IP to offer as many requests as possible to each supplier. To allow the combination of overlaps and fixed menu sizes, we introduce dummy requests. These steps also avoid one-request menus that result from assuming deterministic information in our scenarios (Mofidi and Pazour 2019).

Formulating our IP requires the following set of notations, parameters, and decision variables. As all suppliers have the autonomy to not participate, i.e., not select any requests, we use the set A_t to denote the union of the requests and the no choice alternative. We use the term *alternative* to include either a request or the no choice option. We use the term *request* to denote only requests.

Notations.

Sets:

$r_t = \{r_{1t}, \dots, r_{n_t t}\}$	set of customer requests, with n_t the number of requests in state S_t
$r'_t = \{r'_{1t}, \dots, r'_{n'_t t}\}$	set of n'_t dummy requests
$N = \{r_0\}$	set of no choice alternative
$A_t = r_t \cup N$	set of alternatives
$s_t = \{s_{1t}, \dots, s_{m_t t}\}$	set of suppliers, with m_t the number of suppliers in state S_t
$t = \{1, \dots, T\}$	set of decision points

Parameters:

$\bar{a}$	maximum number of suppliers who can be recommended the same request
$\bar{b}$	maximum number of requests (excluding no choice alternative) recommended to a supplier
$\hat{v}_{ijt}^\sigma$	platform's estimate of supplier s_{jt} 's utility of fulfilling alternative r_{it} in scenario σ
c_{ijt}	reward if platform assigns request r_{it} to supplier s_{jt}

Decision Variables:

x_{ijt}^σ	1, if the platform recommends alternative r_{it} to supplier s_{jt} in scenario σ ; 0, otherwise
$\dot{x}_{i'jt}^\sigma$	1, if the platform recommends dummy request $r'_{i't}$ to supplier s_{jt} in scenario σ ; 0, otherwise
$\hat{\omega}_{ijt}^\sigma$	1, if supplier s_{jt} selects alternative r_{it} in scenario σ ; 0, otherwise
y_{ijt}^σ	1, if the platform assigns request r_{it} to supplier s_{jt} in scenario σ ; 0, otherwise

IP Formulation for Scenario σ .

$$\max \sum_{i=1}^{n_t} \sum_{j=1}^{m_t} c_{ijt} y_{ijt}^\sigma - M \sum_{i'=1}^{n'_t} \sum_{j=1}^{m_t} \dot{x}_{i'jt}^\sigma \quad (5)$$

subject to:

$$\sum_{j=1}^{m_t} x_{ijt}^\sigma \leq \bar{a} \quad \forall i = 1, \dots, n_t \quad (6)$$

$$\sum_{i=1}^{n_t} x_{ijt}^\sigma + \sum_{i'=1}^{n'_t} \dot{x}_{i'jt}^\sigma = \bar{b} \quad \forall j = 1, \dots, m_t \quad (7)$$

$$x_{0jt}^\sigma = 1 \quad \forall j = 1, \dots, m_t \quad (8)$$

$$\sum_{i=1}^{n_t} y_{ijt}^\sigma \leq 1 \quad \forall j = 1, \dots, m_t \quad (9)$$

$$\sum_{j=1}^{m_t} y_{ijt}^\sigma \leq 1 \quad \forall i = 1, \dots, n_t \quad (10)$$

$$y_{ijt}^\sigma \leq \hat{\omega}_{ijt}^\sigma \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (11)$$

$$\max \sum_{i=0}^{n_t} \sum_{j=1}^{m_t} \hat{v}_{ijt}^\sigma \hat{\omega}_{ijt}^\sigma \quad (12)$$

subject to:

$$\hat{\omega}_{ijt}^\sigma \leq x_{ijt}^\sigma \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (13)$$

$$\sum_{i=0}^{n_t} \hat{\omega}_{ijt}^\sigma \leq 1 \quad \forall j = 1, \dots, m_t \quad (14)$$

$$x_{ijt}^\sigma, \hat{\omega}_{ijt}^\sigma, y_{ijt}^\sigma \in \{0, 1\} \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (15)$$

$$\dot{x}_{i'jt}^\sigma \in \{0, 1\} \quad \forall i' = 1, \dots, n'_t, \forall j = 1, \dots, m_t \quad (16)$$

A bi-level optimization model is presented in (5)-(16), in which the platform serves as a leader and determines menus and assignments by maximizing (5), subject to (6)-(11), (15)-(16). Then the suppliers follow by making selections from the menus by maximizing (12), subject to (13)-(14). In the leader problem, we enforce a maximum overlap of requests among supplier menus with constraints (6). If there are enough requests in the system to do so, we enforce a fixed menu size $\bar{b}$ for all suppliers. However, a fixed menu size constraint becomes infeasible when there is an imbalance in the number of requests to the number of suppliers, so dummy requests are introduced. Thus, equation (1) of our model is replaced with constraints (7). As suppliers are allowed to reject

their recommendations, a no choice option is required to be offered to all suppliers via constraints (8). Constraints (9)-(11) constitute what should be the third stage of the IP, as they provide the request-supplier assignments determined after the supplier selection process, which is the second stage of the IP. However, by assuming a deterministic supplier selection behavior, we can merge this third stage with the first stage of the IP, as currently shown. Constraints (13) and (14) enforce that a supplier can only select a request from their menu and only a single one, respectively. The objective function of the platform (5) includes the reward (3), as well as a penalty with cost $M > \max c_{ijt}$ to ensure the platform includes dummy requests to supplier menus only in cases of a request-supplier imbalance to satisfy menu size constraints (7).

Single Stage IP Conversion. Solving a two-stage problem is computationally difficult, so we convert the two-stage IP into a single stage IP by replacing the second-stage objective function with a set of equivalent constraints. Specifically, the second stage objective (12) is converted into constraints (17) based on Mofidi and Pazour (2019). The second stage's objective function captures each supplier selecting the recommended request with the highest expected utility value, and the equivalent set of constraints mimics that behavior. The constraints represent all possible ways an assortment can be made to each supplier. For a given set of recommendations, as the supplier chooses based on the expected utility value, these constraints enforce that the selection variable (i.e., $\hat{\omega}_t$) is one for the highest recommended request. In the constraints, the subscript k represents the ordinal preference index of an alternative for a given supplier s_{jt} 's expected utility, i.e., the lower the k value, the higher the preference. The constraints designate if-then logical expressions, in that if a supplier is recommended a set of alternatives, then they will select the alternative with the lowest k value. To enforce these if-then constraints requires new input parameters, α_{ijk}^σ and β_{ijk}^σ , which are either a value of 0, 1, or -1, and represent the rank ordering coefficients for supplier s_{jt} of alternative r_{it} with preference index k (see Section 4 in Mofidi and Pazour (2019) for details).

$$\sum_{i=0}^{n_t} \alpha_{ijk}^\sigma x_{ijt}^\sigma - \sum_{i=0}^{n_t} \beta_{ijk}^\sigma \hat{\omega}_{ijt}^\sigma \leq 0 \quad \forall j = 1, \dots, m_t, \forall k = 0, \dots, n_t \quad (17)$$

The output of the IP is a menu of requests for each supplier for each scenario, i.e., x_{ijt}^σ . This output is illustrated in the tables at the bottom of Step 2 in Figure 3, where a purple dotted cell refers to an offer. In scenario 1, for example, the menu that ensues from the IP consists of recommending requests A and C to supplier 1, and requests B and C to supplier 2. Due to solving the IP with different input values, the menus from scenarios 2, 3, and 4 are different. Thus, in the next step, we develop a method to combine these ensuing, but different menus.

4.4. Step 3 - Consensus Solution

Given that multiple scenarios are solved at each decision point t , yielding numerous menus for each supplier, a “consensus” solution is needed to create a single, final menu for each supplier. Such a solution should reflect the structure discovered in the individual solutions. As across the scenarios suppliers’ attribute weights are varied, a recommendation to a supplier appearing in most scenario menus suggests the recommendation is robust to the set of suppliers’ preferences. Thus, such a request has a greater chance that the supplier will not reject this frequent recommendation, no matter the supplier’s actual preference. Furthermore, a request recommended throughout the scenarios also indicates such a recommendation does not interfere (too) much with the preferences of the other suppliers. Including this request in a supplier’s menu is not too disruptive to the system (the platform performance is a function of multiple suppliers’ selection outcomes). Therefore, the consensus solution prioritizes the most frequent recommendations in each supplier’s final menu.

Algorithmically, the consensus solution is derived as follows. Once the IP is solved in Step 2 for a given scenario and decision point, the menus x_{ijt}^σ are recorded, and once all scenarios are solved, the frequency of request r_{it} being recommended to supplier s_{jt} at decision point t across all scenarios is computed, i.e., $Freq_{ijt} = \sum_{\sigma} x_{ijt}^\sigma$. The Hungarian algorithm establishes the final recommendations x_{ijt} by finding the frequency-maximizing menu for a given maximum menu size and maximum overlap. Specifically, we solve an integer program to optimality with input parameters x_{ijt}^σ from Step 2 and decision variables x_{ijt} , in which we maximize $\sum_{i=1}^{n_t} \sum_{j=1}^{m_t} Freq_{ijt} x_{ijt}$, subject to constraints (1) and (6). This approach takes a systematic view of menu creation considering a set of suppliers, which we find works better than maximizing the frequency of the recommendations for each individual supplier (see Online Appendix B.1).

In our example in Figure 3, request A is recommended to supplier 1 in three out of the four scenarios, request B in two of the four scenarios, and request C in three of the four scenarios. With a menu size of $\bar{b} = 2$ requests and an overlap of $\bar{a} = 2$ suppliers, the top two most frequently recommended requests to supplier 1 are A and C, and requests A and C’s most frequently recommended supplier is supplier 1, thus establishing supplier 1’s final menu with requests A and C. For supplier 2, its top two most frequently recommended requests are B and C, and vice versa, thus creating supplier 2’s final menu with requests B and C.

4.5. Step 4 - Supplier Selections

Once suppliers are offered their final menu of requests, they each select their top recommended request based on their actual utility values, i.e., v_{ijt} . Specifically, given a supplier s_{jt} ’s final recommendations x_{ijt} , s_{jt} will select the request $r_{i_{max}t}$, where $i_{max} = \arg \max_{i=0, \dots, n_t} (v_{ijt} \odot x_{ijt})$, and the symbol $\odot$ signifies element-wise multiplication. In our example, supplier 1 selects request C over request A because supplier 1’s utility value for request C is higher than that of request A, and supplier 2 selects request B over request C for similar reasons.

4.6. Step 5 - Assignment Algorithm

Based on the supplier selections, three outcomes are possible: a unique selection of a request, multiple selections of a request, or rejection of a request. In the first case, a request is selected by a single supplier, so that request is assigned to that supplier. If a request is selected by multiple suppliers, the supplier yielding the highest platform reward is assigned. In our example in Figure 3, each request is selected by a single supplier. Thus, those requests are assigned to their corresponding supplier, i.e., request C is assigned to supplier 1, and request B is assigned to supplier 2. Once assignments are made, the actual objective value at decision point t is computed with formula (3).

5. Experimental Set Up

Our computational experiments use Chicago as a case study. In this section, we first describe data generation for requests and suppliers, and then detail how we set platform rewards, supplier utility, and parametrization of the suppliers' choice models. Finally, we present a summary of the exogenous factors varied in experiments, as well as the benchmark policies we compare our MSA method to and how (i.e., stakeholder key performance indicators).

5.1. Transportation Network for Requests and Suppliers

We consider a time horizon of $T = 10$, which results in 9 time periods where decisions are made (we show that the results are similar for longer horizons in Section A.2). We set the expected numbers of suppliers and requests over the entire horizon to 100 each. The period in which a request or a supplier arrives is generated from the uniform distribution $U[1, 10]$. The maximum number of periods a request or a supplier will stay in the system is generated from the uniform distribution $U[3, 5]$, but is limited by the end of the problem horizon (we explore the impact of varying the length of stay (i.e., patience) of suppliers and requests in Section A.2). The menu size limit externally set by the platform is $b = 5$; thus, every supplier can be offered at most 5 requests per time step.

For suppliers and requests, we generate origins and destinations within the Chicago Region and their travel distances and times as follows. We use the Chicago Region's actual road network and historical transportation data collected via the Chicago Area Transportation Study (all network data are obtained from github.com/bstabler/TransportationNetworks/tree/master/chicago-regional). The Chicago Regional Transportation Network data set includes 1778 zones, 12978 nodes, and 39017 arcs. For each road arc, we have its length, its free flow travel time T_{ff} , its vehicle volume V , and its maximum vehicle capacity C . The latter three are needed to calculate the vehicle travel times given different levels of traffic. From this data set, we also derive zone-to-zone origin-destination (o-d) probabilities, and used to generate both request and supplier o-d trips.

Next, utilizing Dijkstra’s Algorithm, we find the shortest travel distance from a zone centroid to every other zone centroid. Similarly, we also find the shortest travel time between two zones (which depends on the traffic levels). The average travel time L between two nodes is computed utilizing the Bureau of Public Roads (BPR) function (of Public Roads 1964):

$$L = T_{ff} \left(1 + p_{BPR} \left(\frac{V}{C} \right)^{q_{BPR}} \right). \quad (18)$$

Standard values for the BPR function parameters are selected, i.e., $p_{BPR} = 0.15$ and $q_{BPR} = 4.0$. We simulate different levels of traffic congestion by multiplying the vehicle volume V by one of three coefficients, 1, 3, and 5, which correspond, respectively, to *low traffic*, *medium traffic*, and *high traffic* in our experiments. As we describe later in this section, these different traffic levels impact the supplier selections and the accuracy of a platform to predict supplier utilities.

5.2. Platform Reward

The parameter c_{ijt} represents the platform’s reward of assigning request r_{it} to supplier s_{jt} , and it is dependent on the shortest travel distance from a supplier’s origin to a request’s origin, $D_{ijt}^{(o-o)}$. As this distance increases, the platform’s reward decreases as the platform has to pay for a supplier further from the supplier’s origin. Thus, we capture that the platform’s reward is highest when a supplier’s origin is closest to the request’s origin in (19).

$$c_{ijt} = D^{max} - D_{ijt}^{(o-o)}. \quad (19)$$

The constant D^{max} is the maximum distance of all shortest distances in the population data, which ensures all rewards are greater than or equal to zero.

5.3. Supplier Choice Model

The choice model introduced in Section 3.2 requires two parameters, the utility value v_{ijt} for supplier s_{jt} and request r_{it} and a “no choice threshold” v_{0jt} , i.e., the utility value below which a supplier will not select a request at all during a given time step.

Supplier Utility Values v_{ijt} . For supplier utility values v_{ijt} , we use a multi-attribute utility model with two attributes, distance and time, as well as a term reflecting additional features difficult for the platform to estimate with the data available (such as preferences of neighborhoods).

Each supplier has personalized attribute weights, which capture different supplier preferences. The time attribute is influenced by the traffic level. In low traffic, the distance and the time attributes are highly correlated, so the ranking of requests based on a supplier’s utility values in low traffic will barely change regardless of the supplier’s attribute weights. This translates to a platform having less uncertainty in supplier selection. However, as traffic increases, the supplier’s

attribute weights will have a greater impact on the supplier's utility values because the attribute values of distance and time are less correlated; thus the platform's uncertainty in supplier selection is higher for higher traffic levels.

$$v_{ijt} = T^{max} - (\delta_{ijt} D_{ijt}^{(d-d)} + (1 - \delta_{ijt}) T_{ijt}^{(d-d)}) + \epsilon_{ijt}. \quad (20)$$

The supplier utility values are generated with formula (20), in which the parameter $D_{ijt}^{(d-d)}$ is the shortest distance from request r_{it} 's destination to supplier s_{jt} 's destination; $T_{ijt}^{(d-d)}$ is the distance with the shortest travel time from request r_{it} 's destination to supplier s_{jt} 's destination; T^{max} is the maximum distance with the shortest travel time within the population data; and δ_{ijt} is supplier s_{jt} 's attribute weight, which is assumed to be a random variable generated from a uniform distribution with a minimum of 0 and a maximum of 1. A high value for δ_{ijt} indicates a stronger preference for a shorter distance from the request's destination to the supplier's destination, as opposed to a shorter travel time between the two destinations. Finally, ϵ_{ijt} is a random variable capturing a supplier's utility from sources other than distance and time, which is generated from a normal distribution with a mean of 0 and a standard deviation as a function of the shortest travel time distance $T^{(d-d)}$ (i.e., $1/3 \times \text{std}(T^{(d-d)}) \times (60\text{th percentile of } T^{(d-d)})$).

We assume the platform has knowledge about the stochastic distribution of the utility weights and ϵ_{ijt} values, for example, by using different approaches to model specification techniques from discrete choice experiments (e.g., Lancsar, Fiebig, and Hole 2017). Equation (20) is used to generate the actual supplier utility values v_{ijt} needed in subsection 4.5, the estimated supplier utility values $\hat{v}_{ijt}^\sigma$ for scenario σ from Section 4.2, and the expected average supplier utility values $\bar{v}_{ijt}$ for the expected average benchmark policy described in Section 5.5.

Supplier Pickiness and the No Choice Threshold v_{0jt} . The no choice threshold v_{0jt} represents the utility value below which a supplier will not select a request and instead will either wait until the next time step or start travel from their planned origin to their planned destination. Based on the Chicago data, approximately 60% of the shortest distances are less than or equal to 30 miles, and approximately 30% are less than or equal to 18 miles. To capture two levels of supplier pickiness, we thus set the no choice thresholds accordingly. A low no choice threshold represents that, on average, a supplier is willing to select about 60% of all requests, and a high no choice threshold indicates that, on average, a supplier is willing to select about 30% of all requests. To model this, we transform the no choice threshold to a no choice rank in regards to a supplier's utility values v_{ijt} over the entire time horizon. The no choice rank is generated from a normal distribution with a mean equal to the no choice mean rank (60 for low threshold and 30 for high threshold) and a standard deviation equal to the product of the mean and a coefficient of variation.

The coefficient of variation captures heterogeneity within supplier selection behavior. We select two levels for the coefficient of variation, a low level in which the coefficient of variation is equal to 10% and a high level equal to 30%. Thus, a supplier's no choice rank per time period is not exactly equal to these average values and is independent of the menus offered to the suppliers.

5.4. Exogenous Factors for our Design of Experiments

In our experiments we vary three exogenous factors summarized in Table 1. Each of these exogenous factors, which we assume are outside of the control of the platform, result in different platform operating environments. Specifically, by varying traffic levels, we capture environments that have different accuracy levels of predicting suppliers' utility values (e.g., high traffic environments make it harder for the platform to estimate than low traffic environments). We also vary the pickiness of the suppliers (via the no choice threshold) and vary supplier heterogeneity, i.e., variation in the suppliers' selection behavior (via coefficients of variation for the no choice threshold). Using a full factorial design, this leads to a set of 12 different instance settings.

Table 1 A full-factorial design with these exogenous factors and levels.

Factor	Levels
Traffic	Low
	Med
	High
Pickiness (No Choice Threshold)	Low
	High
Heterogeneity (Coefficient of Variation)	10%
	30%

5.5. Benchmark Policies

We present four benchmark policies used to compare the performance of our MSA method in Sections 6.1 and 6.2, and to quantify the value of our method's components in Section 6.3.

Single Closest Request (SR): In current ride sharing and peer-to-peer logistics practice, no menus are considered; instead, only at most one request is suggested to each supplier in every time step. In practice, requests are often suggested based on proximity to the supplier, for example, by minimizing the average pickup distance of requests (Qin et al. 2020). Following this idea, SR solves an assignment problem based on o-o distances. Specifically, we solve an IP at decision point t that maximizes $\sum_{i=1}^{n_t} \sum_{j=1}^{m_t} c_{ijt} x_{ijt}$ subject to constraints (1) and (6), with, respectively, $\bar{b} = 1$ (i.e., a menu size of 1) and $\bar{a} = 1$ (i.e., each request can only be recommended to a single supplier).

Menus of Closest Request (CR): We explore allowing menus of size $\bar{b}$ and no constraint on the amount of overlap $\bar{a}$. Each supplier is suggested the $\bar{b}$ closest requests based on o-o distances, but ignores systematic impacts of having multiple suppliers. Specifically, this greedy method searches

for the smallest values in each supplier's column in the o-o matrix, adds the corresponding request to the supplier's menu, and replaces the value with $\max(o - o)$ in the supplier's column. This procedure is repeated until there are $\min(\bar{b}, n_t)$ requests in each supplier's menu.

Expected Average Policy (EAA): This policy ignores uncertainty in supplier selections, and instead assumes that all suppliers act deterministically by selecting the request offered with the highest expected preferences (Mofidi and Pazour (2019)), i.e., by assuming expected average values for all parameters within the choice model. To capture this policy, a single IP given in Section 4.3 is solved by the same procedure as a single scenario of our MSA. In generating the expected average supplier utility values $\bar{v}_{ijt}$, the expected values of the parameter δ_{ij} and the error ϵ_{ij} are taken (i.e., $\delta_{ij} = 0.5$ and $\epsilon_{ij} = 0$) in formula (20). As for a supplier's no choice threshold, the standard deviation of the generated no choice rank is set to zero.

Upper Bound (UB): To represent the ideal situation for the platform, we assume all information about request and supplier arrivals and departures, as well as the suppliers' preferences (and thus selections), are known to the platform. We then solve the multi-period and deterministic MIP to optimality (see Online Appendix B.3 for the details). While such an upper bound is not practically achievable, it is helpful for estimation of the optimality gap, and it also helps indicate the severity and cost of the uncertainty of the stochastic and dynamic equivalent.

5.6. Stakeholder Key Performance Indicators

A challenge of platforms is their need to accommodate multiple stakeholders. Therefore, we are interested in understanding how our method impacts not only the platform, but also supplier and request experiences. To measure the impact of each policy on these three stakeholders, we define five key performance indicators (KPIs):

Revenue: The platform revenue per replication is an indicator of the platform's profitability, and a high value is preferred. The platform revenue over the whole time horizon is computed by summing (3) over all decision points t in the time horizon, as shown in (21):

$$\sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{j=1}^{m_t} c_{ijt} y_{ijt}. \quad (21)$$

Number of Assignments: The average number of successful assignments per day. A high value indicates more demand is being met, and more suppliers found requests they are willing to serve.

Request Waiting Time: The average number of periods a request waits in the system to be assigned. Waiting also ends when the request leaves the system unassigned. Requests prefer a small value, i.e., short waits and quick assignments.

Supplier Waiting Time: Similar to "Request Waiting Time", but measured for suppliers. Again, a small value is preferred.

Supplier Unhappiness: The suppliers’ unhappiness is a combination of unacceptable offers and unsuccessful selections. Unacceptable offers occur when a supplier does not select any of the offered requests in a given time step, thus rejecting them all. An unsuccessful selection in a time step occurs when a supplier selects a request, but is not assigned to the supplier (e.g., due to duplicate selections). Supplier unhappiness is computed as the average number of times a supplier either rejects an unacceptable offer or experiences an unsuccessful selection. A small value is preferred.

6. Computational Study

This section presents results from our computational study. We first compare KPIs from our MSA method to the introduced benchmark policies. Then, we quantify the value of our method’s components over a variety of instance characteristics and MSA policy parametrization. Lastly, we analyze the value of holistically generated menus of our MSA. To compare different MSA parametrizations, we denote the MSA-policies “MSA($\bar{b}, \bar{a}$)”. Our default MSA is MSA(3,3), and if we use only MSA, it denotes that it is at this default.

6.1. Performance Experiments

In this section, we compare our MSA method to the benchmark policies introduced in Section 5.5. For all policies, we set the base case inputs as described in Section 5.1. For all policies that use a menu (i.e., all policies but SR), we set the maximum menu size for each supplier to $\bar{b} = 3$ requests. For MSA, EAA, and UB, the maximum overlap for each request is set to $\bar{a} = 3$ suppliers. We run 20 replications for each of the 12 instance settings by varying the three factors described in Table 1, the pickiness threshold, its coefficient of variation (COV), and the traffic level. For each of these 240 runs, we compute the policy’s achieved objective values and conduct paired t-tests between the MSA and the benchmark policies. We then compare each policy’s objective values to the UB’s objective value. The percent differences compared to this UB of the 240 runs are shown as box plots in Figure 4. The MSA, over all runs, achieves an average gap with the UB of just 15.3%. This is a statistically better performance than EAA, CR, and SR, which achieve average gaps of 18.2%, 18.6%, and 27.5%, respectively. This indicates that both components of the MSA, anticipating supplier behavior and offering menus, are valuable. We quantify the value of these components in different environmental situations later in this section. The average gaps for EAA (18.2% gap) and CR (18.6% gap) are similar; however, EAA shows a larger variance. MSA also statistically outperforms the benchmark policies (i.e., the p-values of the paired t-tests were all less than 0.05) for all individual instances with SR and for the large majority of individual instances with EAA and CR (see the detailed results in Appendix A.1). For these experiments, the runtime for the MSA was always less than 5 seconds for one time period (see Appendix A.4 for a more detailed analysis of runtime).

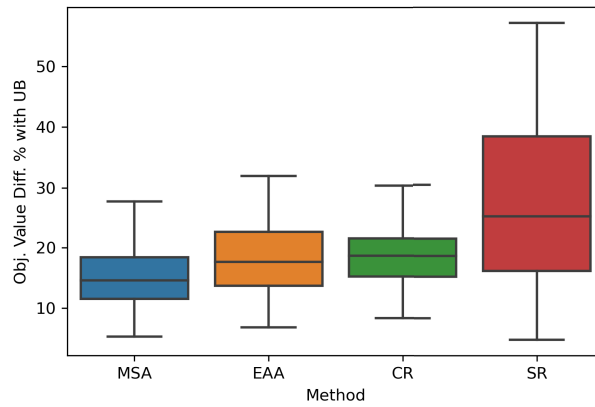


Figure 4 Objective Value Difference Percentages of UB v. MSA, EAA, CR, SR.

6.2. Comparisons to the Single Closest (SR) Request Policy

In the previous section, we have shown that the MSA significantly outperforms the single closest request policy (SR), a policy applied by some companies in practice, in terms of platform revenue. In this section, we analyze how MSA achieves these performance gains by directly comparing the decision variable values from SR and MSA, as well as extending our analysis to stakeholder KPIs beyond platform revenue. To this end, we calculate the differences in KPIs of the two policies, SR (which does not use menus nor overlap) and MSA (with menus of size 3 and overlap of 3). The average results are presented in Table 2. Positive improvement values are preferred for platform revenue and number of assignments, while negative improvement values are preferred for waiting times and supplier unhappiness. The p-values found in this table and all following tables are the result of performing a paired t-test on the two methods being compared in that subsection.

Table 2 Average KPI Changes of MSA(3,3) Compared to SR.

KPI	Avg. Change (%)	p-value
Platform Revenue	+16.7	<<0.05
Number of Assignments	+21.0	<<0.05
Request Waiting Time	-39.0	<<0.05
Supplier Waiting Time	-36.1	<<0.05
Supplier Unhappiness	-35.9	<<0.05

The MSA with menus significantly improves all five performance measures compared to SR. The average platform revenue increases by 16.7%, and the number of assignments increase even further by about 21.0%. This increase is valuable as successful assignments will retain both suppliers and requests and may lead to business growth in the long run. The larger increase in assignments compared to revenue indicates that MSA does not perform “cherry picking”, but instead holistically makes decisions that may give up a little revenue on some assignments to make more assignments and more revenue overall (see further analysis in Online Appendix B.2).

Moreover, Table 2 indicates substantially reduced waiting times for both requests (-39.0%) and suppliers (-36.1%), which improves the experience for both requests and suppliers. Also, supplier unhappiness is lower for MSA compared to SR, even though in contrast to SR, the MSA allows request overlaps and therefore potential duplicate selections. We analyze the two causes of supplier unhappiness (supplier *rejections* and *failed selections*) in SR and MSA in detail, with the results shown in Figures 5, 6, and 7. The average percentage of rejections represents the average percentage of suppliers who rejected their menus in a time period. The average percentage of failed selections represents the average percentage of selections that led to non-assignments in a time period.

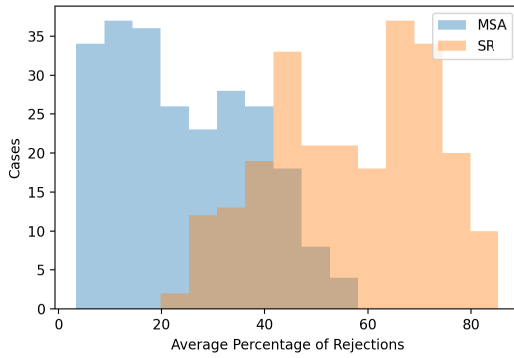


Figure 5 Average Percentage of Rejections for MSA(3,3) and SR.

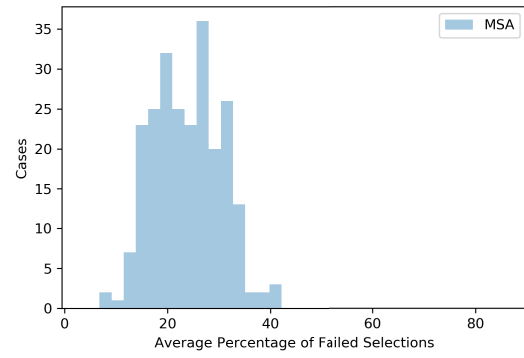


Figure 6 Average Percentage of Failed Selections for MSA(3,3) and SR.

Figure 5 depicts, over all instances, the percentage of suppliers per time step rejecting the offered requests. For example, a value of 50% indicates that for the corresponding instance setting, in a time step, half of the suppliers on average reject their offer. The percentages for SR and MSA differ dramatically. With menus (i.e., for the MSA), the percentages are usually below 50%. However, for SR, the percentages are nearly always above 50%. Thus, menus enable suppliers to find a (supplier) suitable request even though the platform does not have full information about their preferences. In contrast to SR, duplicate selections or failed selections can occur for the MSA, as shown in Figure 6. The percentages in this figure indicate the percentage of suppliers in a time step that selected a request but were ultimately not assigned that request. Dependent on the instance settings, the percentages of failed selections mainly range between 15% and 35%, with some outliers up to 40%. In the majority of cases, the supplier is assigned to their selected request, but there is a chance a selection might fail due to duplicate selections. In Figure 7, we aggregate both rejections and failed selections to calculate the percentage of cases where a supplier is “unhappy”. The MSA can reduce the unhappy supplier percentage substantially because MSA has the ability to capture the trade-off between rejections and duplicate selections when making menus. In an attempt to further reduce supplier unhappiness, we extend the MSA in Appendix A.3.2 to learn supplier preferences

from selections of unhappy suppliers, and we show that learning is especially useful when suppliers are picky and when their behavior is highly uncertain to the platform.

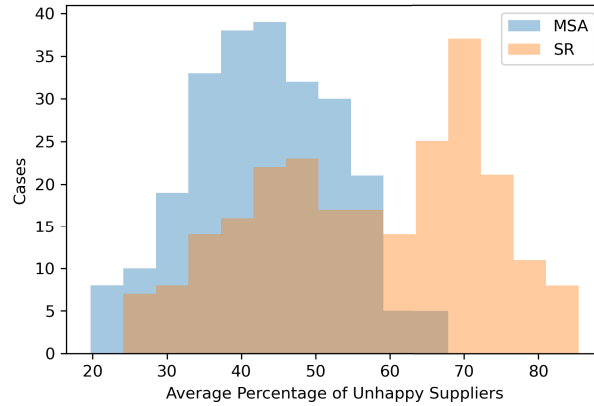


Figure 7 Average Percentage of Unhappy Suppliers for MSA(3,3) and SR.

6.3. Analysis of the MSA

In this section, we analyze how and when the MSA is particularly valuable, and quantify how the MSA can impact the experience for the platform, suppliers, and requests. To do so, we compare different approaches that turn on and off specific aspects of the MSA: anticipation, menus, and overlap. We also run an analysis of variance (ANOVA) on the objective value of the MSA to highlight the significant interaction effects that involve the menu size and the overlap. The detailed results of the ANOVA can be found in the Online Appendix B.7.

Value of Anticipation. Our MSA anticipates supplier selection by sampling a set of scenarios and selecting a solution with respect to the individual scenario solutions. In Table 3 we measure the value of this anticipation by comparing MSA(1,1) to SR. Both of these methods do not have menus nor overlap, but MSA(1,1) does anticipate supplier selection (while SR does not).

Table 3 Average KPI Changes of MSA(1,1) Compared to SR.

KPI	Avg. Change (%)	p-value
Platform Revenue	+6.7	<<0.05
Number of Assignments	+7.8	<<0.05
Request Waiting Time	-15.4	<<0.05
Supplier Waiting Time	-14.3	<<0.05
Supplier Unhappiness	-13.9	<<0.05

The MSA(1,1) without menus significantly improves all five performance measures compared to the SR (however, not as much as the MSA(3,3) with menus). Revenue is increased by 6.7% and assignments by 7.8%. Waiting times for suppliers and requests are reduced, as is supplier unhappiness. As there are no menus nor overlap in either of the compared methods, “supplier unhappiness”

captures the total number of rejections. These results indicate that even when a platform decides against offering menus, our MSA approach that anticipates supplier selections is still valuable for all three stakeholders. To provide additional insights on when the value of anticipation is greatest, we calculate the average objective values for varying supplier pickiness and traffic levels. The traffic level is a proxy for a platform's ability to accurately predict supplier selections, with the platform observing more uncertainty in supplier selection behavior with increased traffic levels. For different traffic levels, the platform revenue (or objective value) achieved using MSA(1,1) and SR are depicted as box plots in Figure 8, with the left side depicting values for low supplier pickiness and the right side for high pickiness.

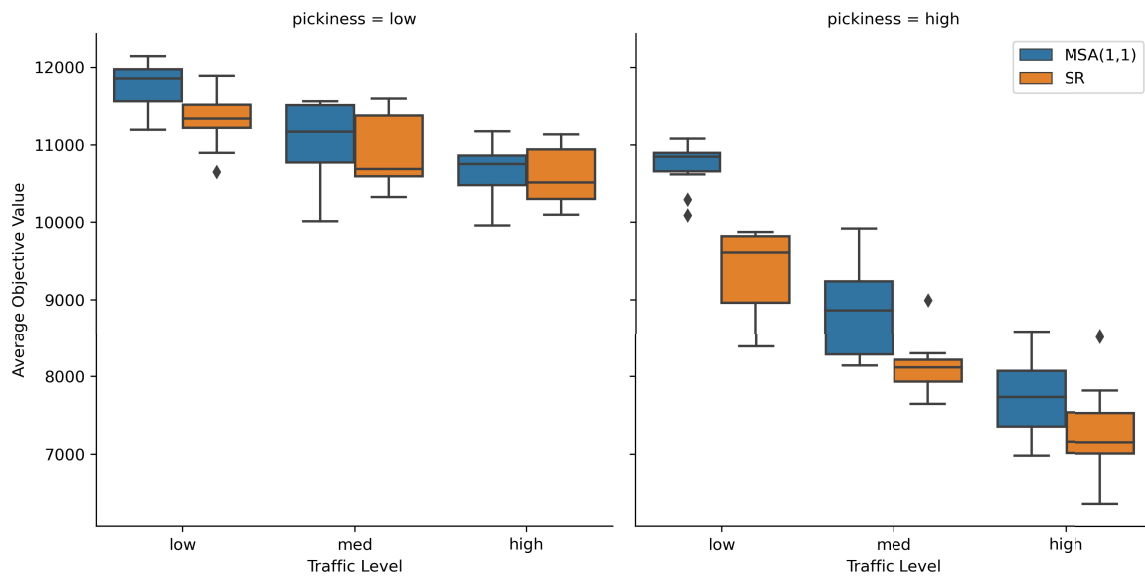


Figure 8 Average Objective Values for Various Traffic and Pickiness Levels - Comparison of MSA(1,1) to SR.

First, as expected, the objective values of both policies depend on both pickiness and traffic levels. With high pickiness, the objective values are smaller because fewer suppliers select requests. With increasing traffic level and therefore increasing uncertainty in supplier selection, the objective values decrease as well. With more uncertainty in supplier selections, the likelihood of the platform recommending suitable requests (to the suppliers) becomes smaller. Thus, in general, being able to predict supplier behavior with high accuracy and having less picky suppliers can benefit a platform. Anticipating supplier behavior provides less value in environments in which the platform has suppliers who are willing to participate, regardless of the impact of fulfilling a request on their planned travel. However, suppliers are likely picky because they have a pre-defined trip, and in environments where pickiness is high and the traffic level is low or medium, we observe that the anticipation of the MSA is particularly valuable (see Figure 8). With high pickiness, offering the “right” requests becomes more challenging. Here, the different scenarios of the MSA are valuable

for anticipating the selection of the suppliers. With a high traffic level, the improvement of the MSA decreases slightly. For these instances, supplier selection becomes more unpredictable and anticipation via scenarios more challenging.

Value of Menus. To measure the value of anticipating supplier selections and having menus, we compare our MSA(3,3) with menu size of 3 and overlap of 3 to the MSA(1,1) without menus nor overlap. As displayed in the second and third column of Table 4, we observe that having menus increases the revenue by 8.8% and the assignments by 11.8%, while waiting times and supplier unhappiness are further reduced. This highlights the value of giving suppliers choice. We further test the improvement when increasing our MSA's menu size to 5. The results comparing MSA(1,1) to MSA(5,3) are displayed in the fourth and fifth columns of Table 4. The improvements in revenue and assignments are smaller than for MSA(3,3). As discussed in the motivation of our method, providing the suppliers with too many choices does not provide additional benefits and may even be counterproductive. Instead, increasing the number of choices for suppliers to select from is useful only up to a point, after which providing more choices leads to higher duplicate selections and supplier selections of requests with less platform benefit.

Table 4 Average KPI Changes of MSA(3,3) and MSA(5,3) Compared to MSA(1,1).

KPI	MSA(3,3) Avg. Change (%)	p-value	MSA(5,3) Avg. Change (%)	p-value
Platform Revenue	+8.8	<<0.05	+7.6	<<0.05
Number of Assignments	+11.8	<<0.05	+11.2	<<0.05
Request Waiting Time	-30.4	<<0.05	-31.0	<<0.05
Supplier Waiting Time	-34.0	<<0.05	-34.0	<<0.05
Supplier Unhappiness	-34.0	<<0.05	-34.0	<<0.05

Next, we analyze when menus are of particular importance. To this end, we repeat the analysis of the previous section, but now we compare MSA(3,3) to MSA(1,1) (rather than SR). The corresponding results are depicted in Figure 9. With low pickiness, menus are not as important, but when pickiness is high, the value of menus is significant. Both observations are confirmed by the ANOVA. In the case of low pickiness, suppliers are more likely to select any of the offered requests; thus, providing them with choice does not add benefit to the platform. However, when suppliers become more picky, selection probabilities decrease, and menus are a valuable remedy to entice participation, which benefits all stakeholders. A similar observation can be made for the traffic level. With low traffic (and therefore more predictable supplier behavior), the differences between MSA(3,3) and MSA(1,1) are rather small, while for medium and high traffic levels (less predictable behavior), the differences increase. Menus are particularly valuable for platforms needing to hedge against less predictable supplier behavior. This observation is again confirmed by the ANOVA.

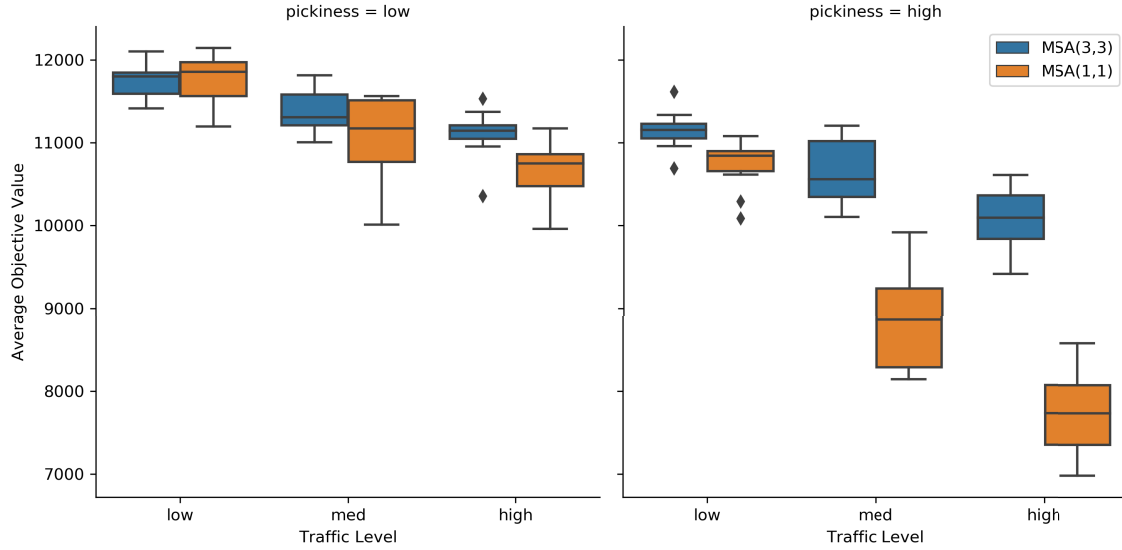


Figure 9 Average Objective Values for Traffic and Pickiness Levels - Comparison of MSA(3,3) and MSA(1,1).

Value of Overlap. The third component of the MSA is the platform’s ability to recommend a request to multiple suppliers simultaneously. Overlap increases selection probability, but also duplicate selections. Therefore, because of this trade-off, we are interested in analyzing when and how overlap is beneficial, and specifically its impact on request and supplier waiting times. To this end, in Table 5 we compare MSA(3,1) without overlap and MSA(3,3) with overlap to MSA(1,1).

Table 5 Average KPI Changes of MSA(3,1) and MSA(3,3) Compared to MSA(1,1).

KPI	MSA(3,1) Avg. Change (%)	p-value	MSA(3,3) Avg. Change (%)	p-value
Platform Revenue	+0.8	<<0.05	+8.8	<<0.05
Number of Assignments	+1.2	<<0.05	+11.8	<<0.05
Request Waiting Time	-2.4	<<0.05	-30.4	<<0.05
Supplier Waiting Time	-1.6	<<0.05	-34.0	<<0.05
Supplier Unhappiness	-1.6	<<0.05	-34.0	<<0.05

The improvements of MSA(3,1) are marginal compared to MSA(1,1). Thus, when overlap is prohibited, menus do not add much value. Indeed, in the “balanced” system of 100 expected suppliers and requests, prohibiting overlap between menus reduces the net menu size per supplier drastically (see Online Appendix B.6). Thus, overlap is required to utilize the power of menus, at least when the system is balanced. In Appendix A.2, we show that overlap can be avoided in systems with a surplus of requests. Also, overlap becomes important in conjunction with offering menus because overlaps mitigate the risks of supplier selection autonomy by offering multiple suppliers the same request. Menus with overlaps can reduce request and supplier waiting times substantially. The value of overlap again depends on the instance characteristics, as depicted in Figure 10 showing the average request wait time by traffic and pickiness levels. When suppliers

are less picky and more predictable (i.e., low traffic level), overlap is less valuable because supplier selection is more likely. Yet, when suppliers are more picky and less predictable (i.e., medium and high traffic levels), the MSA(3,1) without overlap performs substantially worse. Menus with overlaps in uncertain environments can find acceptable matches quicker, which has benefits to all three stakeholders.

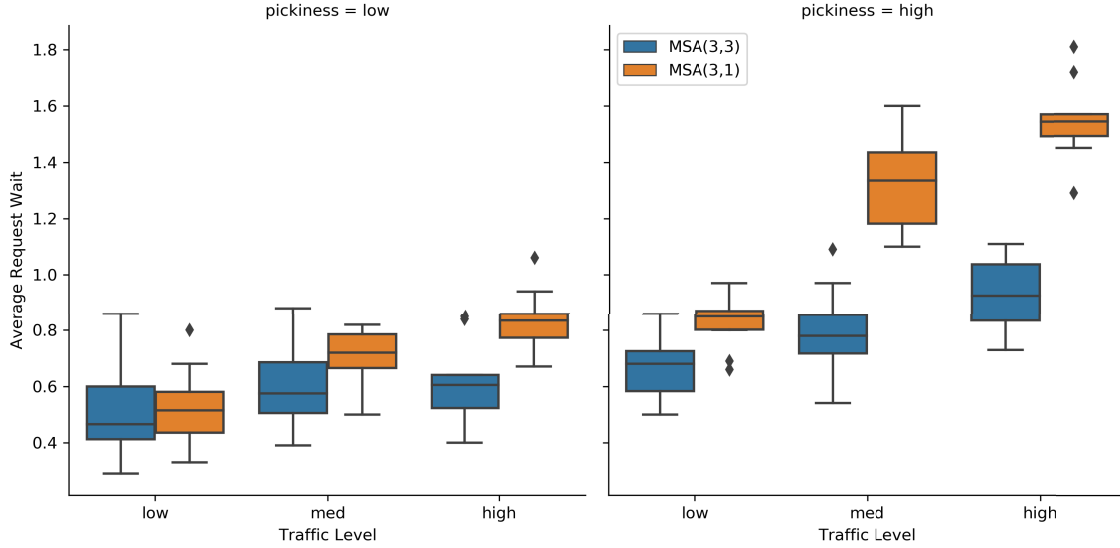


Figure 10 Average Request Wait for Various Traffic and Pickiness Levels - Comparing MSA(3,3) and MSA(3,1).

6.4. Value of Holistically Generating Menus

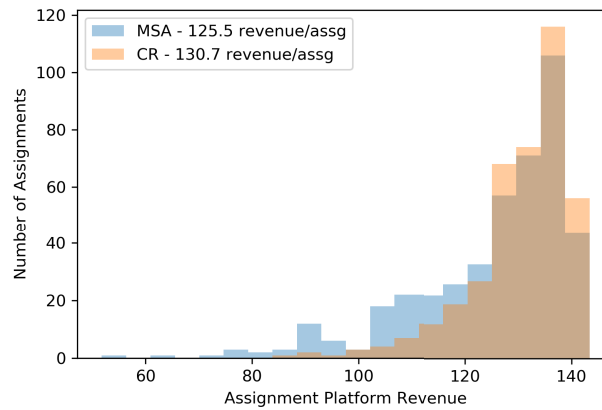
Our MSA considers suppliers and requests holistically when generating menus. Because suppliers often have similar preferences, a holistic view can increase performance over approaches that individually consider how to make menus for suppliers in two ways: one is reducing duplicate selections, and the second is avoiding “cherry picking”. To quantify the value of holistically generating menus, we compare the MSA(3,3) to the CR-policy that offers suppliers a menu of their 3 closest requests. As presented in Table 6, the MSA with menus significantly improves all five performance measures compared to the CR. While the revenue increases by 4.4%, the increase in the number of assignments is twice as high at 8.7%. The waiting times for requests and suppliers, as well as supplier unhappiness, are substantially reduced when applying the MSA(3,3). This is due to the high number of failed (duplicate) selections caused by the CR-policy, which does “cherry picking”. The “cherries” are the same for many suppliers, as shown in Online Appendix B.5.

To illustrate that CR indeed leads to cherry picking, we plot in Figure 11 the distribution of the per assignment platform revenue for MSA(3,3) and CR, for all assignments made in the instance setting with medium traffic, high pickiness threshold, and a 10% coefficient of variation. For CR, most assignments achieve 120 revenue units or more, while, for the MSA, the revenue per assignment is less on average, but the MSA’s more balanced approach leads to greater number of

Table 6 Average KPI Changes of MSA(3,3) Compared to CR.

KPI	Avg. Change (%)	p-value
Platform Revenue	+4.4	$<<0.05$
Number of Assignments	+8.7	$<<0.05$
Request Waiting Time	-23.8	$<<0.05$
Supplier Waiting Time	-30.5	$<<0.05$
Supplier Unhappiness	-30.9	$<<0.05$

assignments. Offering every supplier their closest requests results in the revenue per assignment for the platform to be high. However, given only one supplier can fulfill each request, CR's greedy approach results in many more duplicate selections. In contrast, the MSA, which considers the system holistically and captures supplier selection behavior, may offer requests to a variety of suppliers whose origins may be near or far from the requests' origins. This approach can lead to less revenue per assignment, but the increase in overall assignments surpasses the decreased revenue per assignment many times over. This phenomena can be observed when considering the performance over time. While in the first period the simpler CR policy does slightly better, the MSA is able to outperform it in the second and following periods likely because the "cherries" are gone for CR after the first period while the MSA does not rely on cherry picking over time.

**Figure 11** Histograms of Platform Revenue Per Assignment for MSA(3,3) and CR.

We now compare the performance of MSA and CR for different pickiness and traffic levels in Figure 12. Regardless of the pickiness level, the relative improvement of the MSA is particularly large for low and medium traffic levels, i.e., in presence of more predictable supplier selection behavior. When the traffic level is high, the predictability decreases and a "good guess" of the CR works relatively well sometimes. In fact, as detailed in Appendix A.1, compared to CR, the MSA does not always perform statistically better (i.e., five out of twelve paired t-tests are not significant), specifically when the no choice threshold is high and the traffic level is medium or high, regardless of supplier heterogeneity. This suggests that, as uncertainty increases, the need to holistically consider the whole set of suppliers and requests when generating personalized menus to suppliers

is not as valuable, and instead, simpler, more greedy approaches prioritizing platform revenues may be warranted for menu generation. Interestingly, while the objective value for instances with low pickiness and low traffic levels is larger than for low pickiness and medium traffic levels for all previous policies, it is not the case for CR. This is likely because, with more random supplier selections, “cherry picking” of requests close to many suppliers is occasionally avoided.

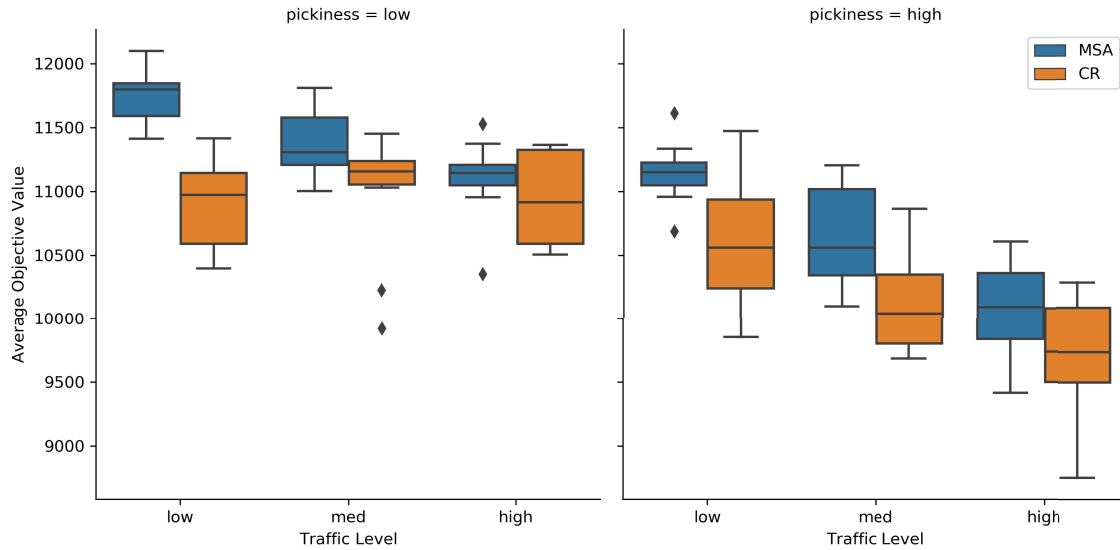


Figure 12 Average Objective Values for Various Traffic and Pickiness Levels - Comparison of MSA(3,3) and CR.

7. Discussions and Future Research

This work is among the first to match crowdsourced occasional drivers (i.e., suppliers) by jointly (1) using a dynamic model capturing stochastic supply capacity and stochastic demand; and (2) optimizing composition of personalized menus of requests for suppliers to select from. As this work integrates two previously disjoint fields of literature, a number of areas for future research exist. While this work provides more autonomy for suppliers, the platform has to deal with rejected requests and duplicated selections. Future work could explore additional incentive structures to try to decrease these outcomes, for example, by considering pricing and compensation in addition to menu composition decisions to strategically incentivize supplier request selection or to encourage suppliers and requests to wait longer in the system. Further future work could decide if a platform should always proactively offer suppliers requests (to avoid annoying the suppliers) and if not, the decision of when to offer a menu could be modeled as a game structure. As this work focused on occasional suppliers with planned personal trips, we matched a supplier to a single transportation request and then the supplier left the system and continued to their planned destination. Future research could investigate and create methodologies for situations in which suppliers complete more than one request (e.g., full-time drivers). One challenge (which our approach does not address)

with full-time drivers, is that this means they could become available again after some time once the current assignment is finished, and the platform might use that information. Furthermore, the platform may create menus in a way that leads to an effective distribution of drivers becoming available again over time and space, similar to “conventional” dynamic pickup and delivery (e.g., Kullman et al. 2021). If suppliers are allowed to select bundles of requests, both in the current time step and in future time steps, this could improve stakeholder performance, yet requires more complex decision making by both the platform and the suppliers (Mancini and Gansterer 2022). The platform could also consider longer-term effects of menus, e.g., the impact of current offerings to future supplier availability. Our experiments showed that with the right menus the suppliers’ waiting times for being matched were reduced and the number of matches increased. Thus, our method may be a good starting point for increasing participation in the long run. In this work, we set the maximum menu size as a parameter, yet, the exploration of the optimal menu size for a given state space is an interesting open research topic. Specifically, state-dependent methods could recommend personalized menu sizes by exploiting spatial information that is good for all stakeholders. This is especially difficult due to the complex state space, and also would require research into perceived fairness concerns by suppliers. Supplier selections will be uncertain to the platform, and as we explore in this work, the platform can benefit from having better predictions of supplier preference and selection behaviors (especially when suppliers are either picky or supplier selections are highly uncertain). Therefore, future research can develop integrated learning methodologies for this problem context. Specifically, a multi-arm bandit trade-off between offering menus that help with learning (exploration) versus offering requests in a menu that have high value to the platform (exploitation) could be pursued. Further, suppliers’ selection behaviors could change from one time step to another, and future research is needed to capture such complex selection behaviors. Resource matching via menus is a promising solution beyond peer-to-peer logistics, and thus future work can adopt our methodologies to other applications, e.g., time slotting for attended home delivery has recently explored the challenging situation when multiple customer selections can occur simultaneously (Visser, Agatz, and Spliet 2019). As our method is limited by the size of the network it can handle, approaches could be explored for large-scale networks, specifically during peak-hours of crowdsourced transportation platforms.

Acknowledgements

We thank the editorial team and the three reviewers for their thorough remarks and great support during the entire process. We also thank Warren Powell for his advice on sequential decision problems with two types of uncertainty and decisions. 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

- Agatz N, Erera A, Savelsbergh M, Wang X, 2012 *Optimization for dynamic ride-sharing: a review*. *European Journal of Operational Research* 223(2):295–303.
- Archetti C, Savelsbergh M, Speranza MG, 2016 *The vehicle routing problem with occasional drivers*. *European Journal of Operational Research* 254(2):472–480.
- Arslan AM, Agatz N, Kroon L, Zuidwijk R, 2018 *Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers*. *Transportation Science* 53(1):222–235.
- Ashkrof P, Correia GHdA, Cats O, van Arem B, 2021 *Ride acceptance behaviour of ride-sourcing drivers*. *arXiv preprint arXiv:2107.07864* .
- Ashkrof P, de Almeida Correia GH, Cats O, van Arem B, 2020 *Understanding ride-sourcing drivers’ behaviour and preferences: Insights from focus groups analysis*. *Research in Transportation Business & Management* 37:100516.
- Atasoy B, Ikeda T, Song X, Ben-Akiva ME, 2015 *The concept and impact analysis of a flexible mobility on demand system*. *Transportation Research Part C: Emerging Technologies* 56:373–392.
- Bent RW, Van Hentenryck P, 2004 *Scenario-based planning for partially dynamic vehicle routing with stochastic customers*. *Operations Research* 52(6):977–987.
- Cleophas C, Cottrill C, Ehmke JF, Tierney K, 2019 *Collaborative urban transportation: Recent advances in theory and practice*. *European Journal of Operational Research* 273(3):801–816.
- Dayarian I, Savelsbergh M, 2020 *Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders*. *Production and Operations Management* 29(9):2153–2174.
- Di Febbraro A, Gattorna E, Sacco N, 2013 *Optimization of dynamic ridesharing systems*. *Transportation Research Record* 2359(1):44–50.
- Ermagun A, Stathopoulos A, 2018 *To bid or not to bid: An empirical study of the supply determinants of crowd-shipping*. *Transportation Research Part A: Policy and Practice* 116:468–483.
- Furuhata M, Dessouky M, Ordóñez F, Brunet ME, Wang X, Koenig S, 2013 *Ridesharing: the state-of-the-art and future directions*. *Transportation Research Part B: Methodological* 57:28–46.
- Horner H, Pazour JA, Mitchell JE, 2021 *Optimizing driver menus under stochastic selection behavior for ridesharing and crowdsourced delivery*. *Transportation Research Part E: Logistics and Transportation Review* 153:102419.
- Jacob J, Roet-Green R, 2021 *Ride solo or pool: Designing price-service menus for a ride-sharing platform*. *European Journal of Operational Research* .
- Kaffe N, Zou B, Lin J, 2017 *Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery*. *Transportation Research Part B: Methodological* 99:62–82.
- Kullman ND, Cousineau M, Goodson JC, Mendoza JE, 2021 *Dynamic ride-hailing with electric vehicles*. *Transportation Science* .

- Lancsar E, Fiebig DG, Hole AR, 2017 *Discrete choice experiments: a guide to model specification, estimation and software*. *Pharmacoeconomics* 35:697–716.
- Le TV, Stathopoulos A, Van Woensel T, Ukkusuri SV, 2019 *Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence*. *Transportation Research Part C: Emerging Technologies* 103:83–103.
- Lee A, Savelsbergh M, 2015 *Dynamic ridesharing: is there a role for dedicated drivers?* *Transportation Research Part B: Methodological* 81:483–497.
- Lei C, Jiang Z, Ouyang Y, 2020 *Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers*. *Transportation Research Part B: Methodological* 132:60–75.
- Li B, Krushinsky D, Reijers HA, Van Woensel T, 2014 *The share-a-ride problem: People and parcels sharing taxis*. *European Journal of Operational Research* 238(1):31–40.
- Liu Y, Bansal P, Daziano R, Samaranayake S, 2019 *A framework to integrate mode choice in the design of mobility-on-demand systems*. *Transportation Research Part C: Emerging Technologies* 105:648–665.
- Liu Y, Li Y, 2017 *Pricing scheme design of ridesharing program in morning commute problem*. *Transportation Research Part C: Emerging Technologies* 79:156–177.
- Mancini S, Gansterer M, 2022 *Bundle generation for last-mile delivery with occasional drivers*. *Omega* 108:102582.
- Mofidi SS, Pazour JA, 2019 *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.
- Mourad A, Puchinger J, Chu C, 2019 *A survey of models and algorithms for optimizing shared mobility*. *Transportation Research Part B: Methodological* 123:323–346.
- Nourinejad M, Ramezani M, 2020 *Ride-sourcing modeling and pricing in non-equilibrium two-sided markets*. *Transportation Research Part B: Methodological* 132:340–357.
- Nourinejad M, Roorda MJ, 2016 *Agent based model for dynamic ridesharing*. *Transportation Research Part C: Emerging Technologies* 64:117–132.
- of Public Roads UB, 1964 *Traffic Assignment Manual for Application with a Large, High Speed Computer* (US Department of Commerce).
- Özkan E, Ward AR, 2020 *Dynamic matching for real-time ride sharing*. *Stochastic Systems* 10(1):29–70.
- Pelzer D, Xiao J, Zehe D, Lees MH, Knoll AC, Aydt H, 2015 *A partition-based match making algorithm for dynamic ridesharing*. *IEEE Transactions on Intelligent Transportation Systems* 16(5):2587–2598.
- Powell WB, Towns MT, Marar A, 2000 *On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of user noncompliance*. *Transportation Science* 34(1):67–85.
- Punel A, Stathopoulos A, 2017 *Modeling the acceptability of crowdsourced goods deliveries: Role of context and experience effects*. *Transportation Research Part E: Logistics and Transportation Review* 105:18–38.

- Qin Z, Tang X, Jiao Y, Zhang F, Xu Z, Zhu H, Ye J, 2020 *Ride-hailing order dispatching at DiDi via reinforcement learning*. *INFORMS Journal on Applied Analytics* 50(5):272–286.
- Rai HB, Verlinde S, Merckx J, Macharis C, 2017 *Crowd logistics: an opportunity for more sustainable urban freight transport?* *European Transport Research Review* 9(3):39.
- Reyes D, Erera A, Savelsbergh M, Sahasrabudhe S, O’Neil R, 2018 *The meal delivery routing problem*. *Optimization Online* .
- Sánchez D, Martínez S, Domingo-Ferrer J, 2016 *Co-utile P2P ridesharing via decentralization and reputation management*. *Transportation Research Part C: Emerging Technologies* 73:147–166.
- Song Y, Ulmer MW, Thomas BW, Wallace SW, 2020 *Building trust in home services—stochastic team-orienting with consistency constraints*. *Transportation Science* 54(3):823–838.
- Soto Setzke D, Pflügler C, Schreieck M, Fröhlich S, Wiesche M, Krcmar H, 2017 *Matching drivers and transportation requests in crowdsourced delivery systems*. *Twenty-Third Americas Conference on Information Systems*, 1–10.
- Tafreshian A, Masoud N, Yin Y, 2020 *Frontiers in service science: Ride matching for peer-to-peer ride sharing: A review and future directions*. *Service Science* 12(2-3):44–60.
- Uber, 2021 *How does uber match riders with drivers?* <https://www.uber.com/us/en/marketplace/matching/>. (Retrieved on April 27, 2021).
- Ulmer MW, Goodson JC, Mattfeld DC, Thomas BW, 2020 *On modeling stochastic dynamic vehicle routing problems*. *EURO Journal on Transportation and Logistics* 9(2):100008.
- Visser T, Agatz N, Spliet R, 2019 *Simultaneous customer interaction in online booking systems for attended home delivery*. Technical Report ERS-2019-011-LIS, ERIM report series research in management Erasmus Research Institute of Management.
- Wang H, Yang H, 2019 *Ridesourcing systems: A framework and review*. *Transportation Research Part B: Methodological* 129:122–155.
- Wang X, Agatz N, Erera A, 2018 *Stable matching for dynamic ride-sharing systems*. *Transportation Science* 52(4):850–867.
- Yu X, Shen S, 2019 *An integrated decomposition and approximate dynamic programming approach for on-demand ride pooling*. *IEEE Transactions on Intelligent Transportation Systems* 21(9):3811–3820.
- Zhang H, Zhao J, 2018 *Mobility sharing as a preference matching problem*. *IEEE Transactions on Intelligent Transportation Systems* 20(7):2584–2592.
- Zhao B, Xu P, Shi Y, Tong Y, Zhou Z, Zeng Y, 2019 *Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach*. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2245–2252.
- Zuo H, Cao B, Zhao Y, Shen B, Zheng W, Huang Y, 2021 *High-capacity ride-sharing via shortest path clustering on large road networks*. *The Journal of Supercomputing* 77(4):4081–4106.

Appendix

A. Appendix

We present the detailed results of our performance experiments, provide the results for varying problem dimensions, explore the value of supplier selection information via different platform protocol settings, and finally analyze the scalability of the MSA.

A.1. Performance Experiments Details

Table 7 presents the MSA’s objective value for each combination of factors of Section 5.4, as well as the objective value percentage differences with the benchmark policies. To further compare the methods, we perform paired t-tests and their significances are provided in Table 7 with the p-values.

Table 7 Average Objective Value Differences and p-values of Various Methods Across Various Conditions.

No Choice Threshold	COV	Traffic Level	MSA Obj.	EAA		CR		SR		PSS		UB	
				Diff.	p-value	Diff.	p-value	Diff.	p-value	Diff.	p-value	Diff.	p-value
low	10%	low	9388	-2.0%	3.5E-02	-8.3%	4.1E-09	-3.4%	4.7E-04	4.1%	1.8E-05	9.9%	2.5E-10
		med	9124	-2.9%	7.9E-04	-4.4%	1.2E-04	-5.4%	2.4E-05	7.0%	4.7E-09	13.2%	1.8E-16
		high	8987	-3.8%	1.5E-04	-2.7%	8.1E-03	-7.7%	3.9E-08	8.2%	3.0E-11	14.7%	7.2E-15
high		low	8747	-3.7%	6.4E-05	-7.3%	2.7E-07	-18.0%	3.8E-13	8.1%	1.7E-12	15.9%	5.8E-16
		med	8174	-3.4%	2.4E-03	-0.4%	0.71	-24.1%	3.6E-12	15.1%	7.6E-13	23.8%	2.3E-16
		high	7947	-4.5%	5.9E-05	-0.7%	0.56	-30.5%	8.3E-14	17.7%	3.0E-11	26.6%	2.1E-14
low	30%	low	9167	-0.8%	0.10	-6.6%	2.3E-08	-3.1%	1.4E-03	6.4%	1.1E-09	12.4%	2.8E-15
		med	8998	-2.7%	2.8E-04	-4.3%	1.4E-03	-5.4%	6.7E-06	8.1%	5.8E-11	14.6%	1.5E-14
		high	8813	-3.8%	1.8E-03	-1.9%	0.06	-8.3%	1.2E-06	9.6%	3.4E-11	16.6%	8.4E-16
high		low	8609	-5.2%	9.3E-06	-7.6%	6.8E-06	-17.2%	3.1E-11	9.9%	3.1E-10	17.6%	2.1E-15
		med	8012	-3.4%	9.7E-03	-0.4%	0.75	-24.6%	6.3E-13	16.9%	2.4E-11	26.0%	3.5E-14
		high	7708	-4.4%	3.2E-03	-0.1%	0.93	-30.9%	6.3E-14	20.7%	1.4E-12	30.2%	1.3E-14
AVERAGE			8639	-3.4%		-3.7%		-14.9%		11.0%		18.5%	

A.2. Impact of Varying Problem Dimensions

In this section, we briefly summarize further observations when problem dimensions vary.

Supplier and Request Patience. In our main experiments, we assume that suppliers and requests stay in the system for 4 time steps on average (or until the problem horizon is reached). However, suppliers and requests may be more or less patient. To test this impact, we perform additional experiments with very impatient suppliers and requests that only stay in the system for 1 time step and with more patient suppliers and requests that stay in the system 6 time steps on average. Our results show that, with 4 expected time steps, our policy achieves 54.4% more assignments and 56.1% more revenue than for the case where suppliers and requests are very impatient. The difference compared to more supplier and request patience is less distinct with 7.4% more assignments and 7.8% more revenue. The relatively small increase may be explained partially by the end-of-horizon effects. For suppliers and requests entering late in the horizon, the difference of 4 or 6 expected periods does not lead to any different behavior. In summary, these results indicate that it is very valuable to keep both suppliers and requests longer in the system to have the opportunity to offer several different menus.

Further, we test the performance gains of MSA to the SR-policy (a non-menu benchmark policy) for requests and suppliers who may be quite impatient if they are not matched initially. With very impatient requests and suppliers, MSA achieves 17.3% more revenue and 21.7% more assignments than SR. Those

improvements match those with more patient requests and suppliers (see Table 2), which suggests that menus are valuable, even if suppliers stay longer or shorter in the system. Specifically, when suppliers only stay for one period, menus provide an advantage because there is a higher chance of participation by the suppliers (compared to a platform only offering them a single request).

Problem Horizon. In our main experiments, we limited the problem horizon to 10 periods to keep the computational burden manageable for our extensive analyses and the UB policy. In an additional experiment, we increase the time horizon to 20 periods for the instance setting with medium traffic, high pickiness threshold, and a 10% coefficient of variation. Simultaneously, we increase the expected number of suppliers and requests to 200 each. The results are shown in Table 8 presenting the improvement of our policy compared to the single closest request (SR) policy.

Table 8 Average KPI Changes of MSA(3,3) Compared to SR with a horizon of 10 periods and 20 periods.

KPI	10 periods Avg. Change (%)	p-value	20 periods Avg. Change (%)	p-value
Platform Revenue	+31.7	$<<0.05$	+19.7	$<<0.05$
Number of Assignments	+37.7	$<<0.05$	+23.8	$<<0.05$
Request Waiting Time	-51.4	$<<0.05$	-44.5	$<<0.05$
Supplier Waiting Time	-53.8	$<<0.05$	-47.4	$<<0.05$

We observe that the improvement for 20 periods is still substantial in all categories, but it decreases compared to 10 periods. One likely reason is that, for 20 periods, the previously discussed end-of-horizon effects are less important. Thus, the requests and suppliers stay on average longer in the system, which leaves the SR more time for a “lucky strike”. This explanation is supported by the observation that the differences in waiting time improvement for suppliers and requests is comparably small between 20 and 10 periods.

Suppliers-Requests Ratio. In our main experiments, we assumed a balanced system with expected numbers of 100 requests and 100 suppliers. However, in some systems, there may be an overflow of either suppliers or requests. Thus, we repeated our experiments, but increased either the number of suppliers or the number of requests to 120. We compared the three policies MSA(1,1), MSA(3,1) and MSA(3,3) for the two new instances, one with 120 requests and 100 suppliers and one with 100 requests and 120 suppliers to the original instances in Figure 13. The x-axis shows the different ratios; the y-axis shows the performance of the three policies. We observe that when applying our MSA(3,3), having either more requests or suppliers in the system increases the objective value significantly compared to the balanced system. Thus, it is not mandatory to have a balanced system to generate revenue. By comparing MSA(1,1) without menus and MSA(3,1) with menus, we further observe that menus are particularly valuable when a surplus of requests is given. In that case, the menus increase the probability of selections without the danger of duplicates. If there are more suppliers than requests in the system, menus are less important because the likelihood to find a match for each request is relatively high even without menus. Finally, when comparing MSA(3,1) without overlap and MSA(3,3) with overlap, we see that overlap becomes essential when more suppliers are in the system. The overlap further increases the probability of requests being selected by at least one supplier.

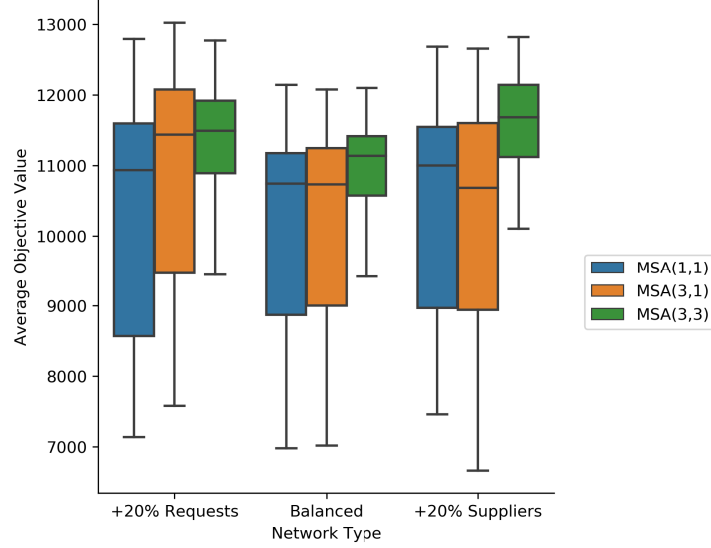


Figure 13 Network Type v. Objective Value - Comparison of MSA(1,1), MSA(3,1), MSA(3,3).

A.3. Value of Supplier Selection Information

In this section we explore the value of different platform protocol settings. The section's tables and graphs present average results from rerunning all experiments described in Section 5.4.

A.3.1. Value of Supplier Willingness Selection

Instead of asking suppliers to respond back to the platform with only their top-choice request in a menu, an alternative protocol is to have suppliers respond back to the platform with any requests in their menu they are willing to serve. We capture this variant within our MSA method, which we label *MSA with willingness*, by modifying the IP given in Section 4.3 by replacing the rejection constraints (8) and the request ranking constraints (17) with supplier willingness constraints (22), in which the parameter q_{ijt}^σ represents supplier s_{jt} 's willingness to service request r_{it} (i.e., if $v_{ijt}^\sigma \geq v_{0jt}^\sigma$, then $q_{ijt}^\sigma = 1$; else $q_{ijt}^\sigma = 0$). To ensure suppliers' menus are filled with as many available requests as possible (instead of with dummy requests $\hat{x}_{i'jt}^\sigma$), we add constraints (23) with the binary decision variable $\ddot{x}_{ijt}^\sigma$ which equals 1 if the platform offers request r_{it} to non-willing supplier s_{jt} and 0 otherwise. As a result of this additional decision variable, overlap constraints (6) and menu size constraints (7) are changed to constraints (24) and (25), respectively. With the willingness variant, we can remove the supplier decision variables $\hat{\omega}_{ijt}^\sigma$; as such, constraints (13) and (14) are removed, and the assignment constraints (11) are replaced with constraints (26).

$$x_{ijt}^\sigma \leq q_{ijt}^\sigma \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (22)$$

$$\ddot{x}_{ijt}^\sigma \leq 1 - q_{ijt}^\sigma \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (23)$$

$$\sum_{j=1}^{m_t} (x_{ijt}^\sigma + \ddot{x}_{ijt}^\sigma) \leq \bar{a} \quad \forall i = 1, \dots, n_t \quad (24)$$

$$\sum_{i=1}^{n_t} (x_{ijt}^\sigma + \ddot{x}_{ijt}^\sigma) + \sum_{i'=1}^{n'_t} \dot{x}_{i'jt}^\sigma = \bar{b} \quad \forall j = 1, \dots, m_t \quad (25)$$

$$y_{ijt}^\sigma \leq x_{ijt}^\sigma \quad \forall i = 1, \dots, n_t, \forall j = 1, \dots, m_t \quad (26)$$

Further, we modify how suppliers select in Step 4 to be for each request r_{it} in supplier s_{jt} 's menu, if $v_{ijt} \geq v_{0jt}$, then supplier s_{jt} selects that request; otherwise, supplier s_{jt} does not select that request. For the remaining requests in each supplier's menu, we then solve a simple assignment problem in Step 5, maximizing platform revenue (i.e., formula (3)).

To quantify the value of our method within the willingness variant, we compare the MSA(3,3) with willingness to the CR-policy with willingness. The CR-policy with willingness generates each supplier's menu the same way as the CR-policy. In Step 4, instead of suppliers selecting their top choice, they select all the requests within their menus they are willing to serve. A breakdown of the performances of MSA with willingness and CR with willingness for different pickiness and traffic levels is described in Figure 14, which illustrates that MSA with willingness yields significantly higher objective values than CR with willingness when pickiness is high.

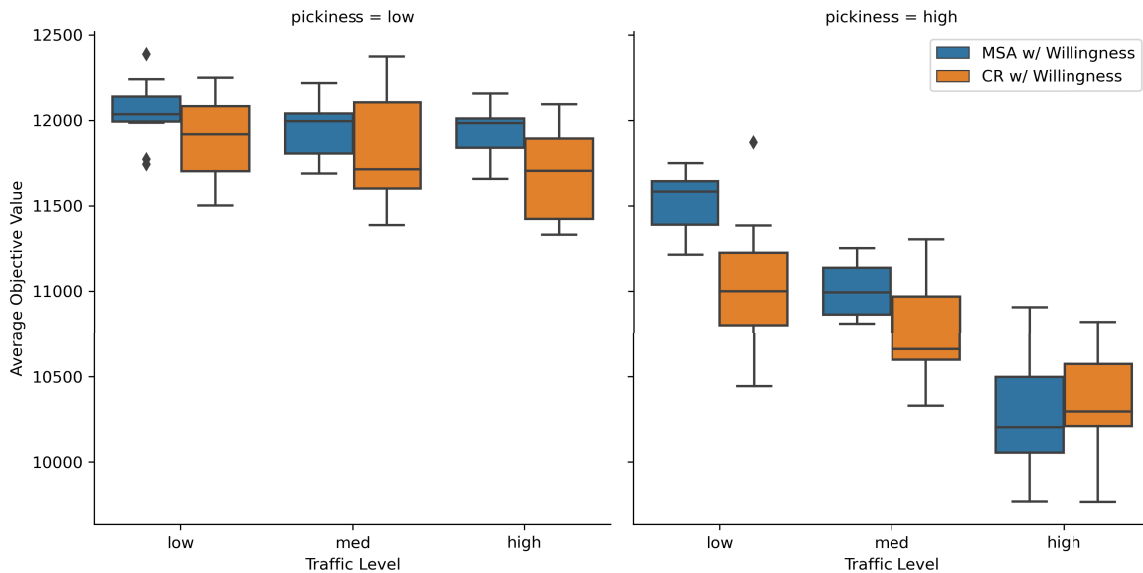


Figure 14 Average Objective Values for Various Traffic and Pickiness Levels for the Willingness Variant: Comparison of MSA with Willingness and CR with Willingness.

MSA with willingness performs better than CR with willingness across all KPIs, as shown in Table 9. Thus, our MSA method also works well with this willingness variant of the problem.

Table 9 Average KPI Changes of MSA(3,3) with Willingness Compared to CR with Willingness.

KPI	Avg. Change (%)	p-value
Platform Revenue	+1.7	$<<0.05$
Number of Assignments	+6.7	$<<0.05$
Request Waiting Time	-27.3	$<<0.05$
Supplier Waiting Time	-37.2	$<<0.05$

To measure the value of using a supplier willingness protocol, we compare MSA(3,3) to MSA(3,3) with willingness, see results in Table 10. Overall, MSA(3,3) (which asks suppliers to respond back to the platform with their top-choice in the menu) performs worse across all KPIs compared to MSA(3,3) with willingness.

This is because in Step 5 the platform now has more flexibility in what requests can be matched. A minor trade-off with the willingness variant is that this can lead to suppliers not receiving their top choice. On average, only 66.1% of *assigned* suppliers get their top choice with the MSA with willingness, compared to 100% of assigned suppliers with MSA. Yet, in both variants, some suppliers are never assigned. Exploring all suppliers (not just the ones who get assigned), we show in Figure 15 the percentage of suppliers assigned their top choice is influenced by supplier pickiness and traffic levels. When supplier pickiness is low, the willingness method yields fewer suppliers being assigned their top choice, compared to when supplier pickiness is high.

Table 10 Average KPI Changes of MSA(3,3) Compared to MSA(3,3) with Willingness.

KPI	Avg. Change (%)	p-value
Platform Revenue	-3.9	$<<0.05$
Number of Assignments	-4.6	$<<0.05$
Request Waiting Time	+43.4	$<<0.05$
Supplier Waiting Time	+62.4	$<<0.05$

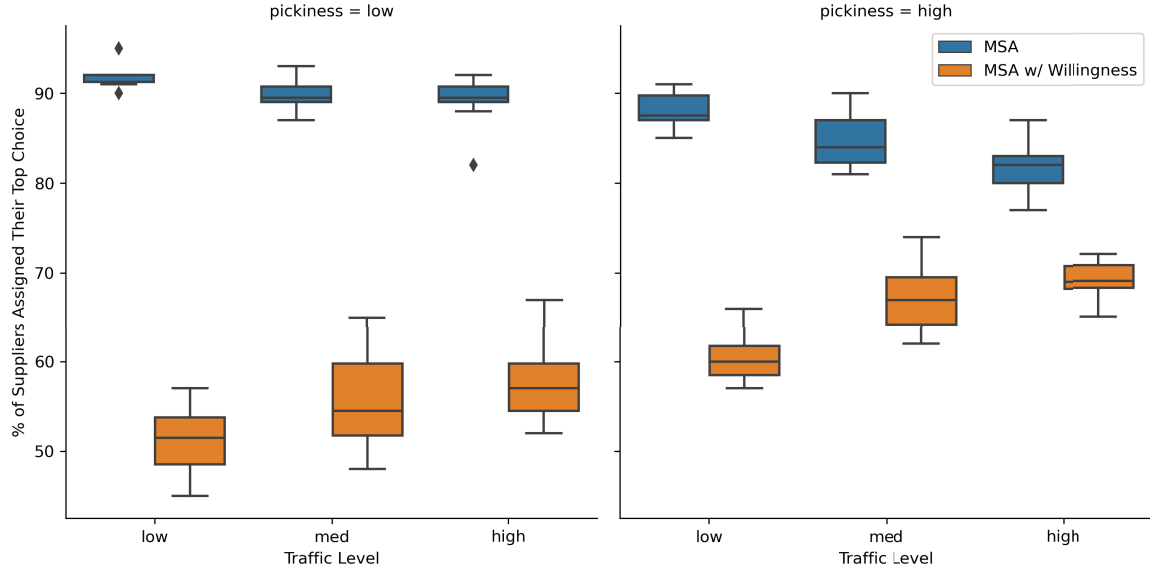


Figure 15 Percentage of Suppliers Assigned Their Top Choice for Various Traffic and Pickiness Levels.

A.3.2. Value of Learning Supplier Selection After Failed Menu Attempts

While the platform does not have full information about supplier preferences, the platform may be able to derive them (at least theoretically) based on the suppliers' previous selections. In the following, we analyze the potential of learning supplier selection. To do so, we present a (potentially unrealistic) "best-case" learning approach, that is we assume the platform learns the selection behavior perfectly for the suppliers who either reject all requests in their menus or select but are not assigned a request. We denote this as the *perfect learning protocol*. To implement this protocol into our MSA, in the IP formulation, we modify constraints (17) for the suppliers with now known selection behaviors, by replacing the estimated parameters α_{ijk}^σ and β_{ijk}^σ with those suppliers' actual parameters α_{ijk} and β_{ijk} .

We measure the value of learning supplier selections after a failed menu attempt by comparing MSA to MSA with perfect learning. Although the MSA with perfect learning outperforms the MSA across all KPIs as shown in Table 11, on average, the MSA performs only 2.4% worse in platform revenue and number of assignments. As our experiment relies on an optimistic learning approach, the real value of learning is likely smaller. We break down these results based on traffic levels and supplier pickiness in Figure 16. While perfect learning after a failed menu attempt does not statistically improve upon the MSA when traffic and pickiness levels are low, when uncertainty is high (i.e., high pickiness and/or high/med traffic levels), statistical improvements of MSA with perfect learning exist. Thus, the value of learning increases with the number of unsuccessful suggestions, i.e., when supplier behavior is highly uncertain to the platform or when suppliers are picky.

Table 11 Average KPI Changes of MSA(3,3) Compared to MSA with Perfect Learning.

KPI	Avg. Change (%)	p-value
Platform Revenue	-2.4	$<<0.05$
Number of Assignments	-2.4	$<<0.05$
Request Waiting Time	+23.4	$<<0.05$
Supplier Waiting Time	+21.0	$<<0.05$
Supplier Unhappiness	+15.4	$<<0.05$

A.3.3. Value of Perfect Supplier Preference Information

An alternative of offering menus to suppliers is to show the suppliers all requests in the system and then ask them to provide their utility values for all the possible requests in the system. Such an approach can be tedious and becomes increasingly time-consuming when the number of requests in the system increases, and it is likely infeasible if requests demand quick matches. Knowing the suppliers' preference information for all requests has value, which we quantify by comparing MSA(3,3) to a policy of perfect supplier preference information (PSP). For this policy, we solve the IP given in Section 4.3 with a supplier's actual utility values v_{ijt} (rather than a prediction). As presented in Table 12, the MSA performs less than 10% worse than the PSP in terms of platform revenue and number of assignments. However, the MSA averages about twice and four times the waiting time for requests and suppliers, respectively, compared to the PSP. This is expected as both rejections and duplicate selections can be avoided by PSP. In essence, there is value in asking every supplier to indicate their preference for every request in the system. However, it is unlikely suppliers will participate in doing so repeatedly over the time horizon and at least not in a timely manner required by requests.

Table 12 Average KPI Changes of MSA(3,3) Compared to PSP.

KPI	Avg. Change (%)	p-value
Platform Revenue	-8.6	$<<0.05$
Number of Assignments	-9.7	$<<0.05$
Request Waiting Time	+146.3	$<<0.05$
Supplier Waiting Time	+287.5	$<<0.05$
Supplier Unhappiness	+935.3	$<<0.05$

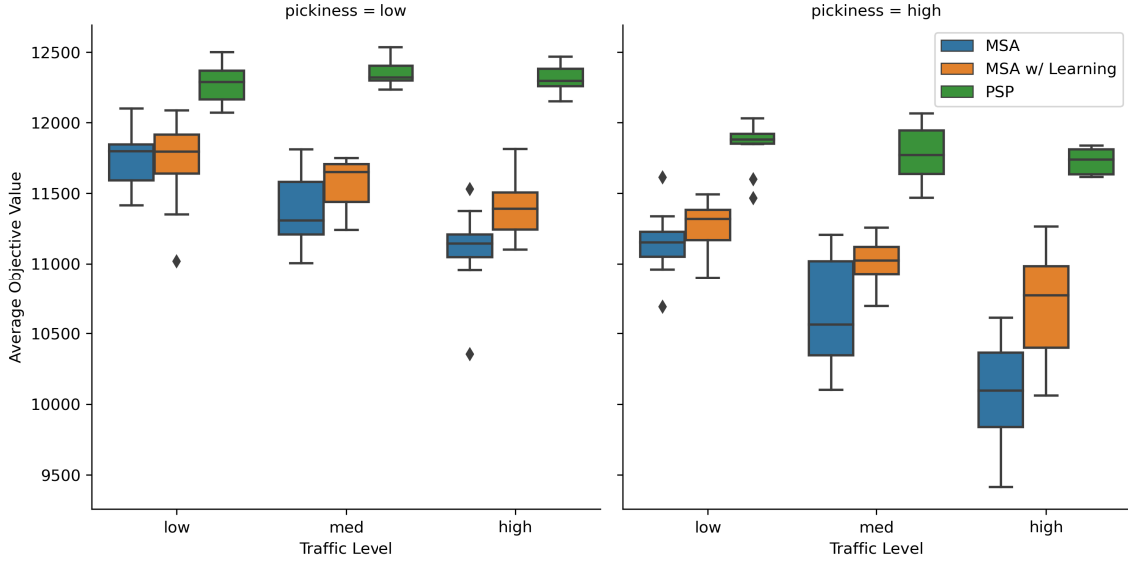


Figure 16 Average Objective Values for Various Traffic and Pickiness Levels: Comparison of MSA, MSA with Learning, and PSP

A.4. Scalability of the MSA

To assess the scalability of the MSA, we first measure the runtime of solving one time period with the IP formulation in Section 4.3. The runtimes for a variety of balanced network problem sizes are explored, i.e., 10 requests and 10 suppliers (10/10), 50 requests and 50 suppliers (50/50), 100 requests and 100 suppliers (100/100), 150 requests and 150 suppliers (150/150), and 200 requests and 200 suppliers (200/200). Figure 17 shows the average runtime (in seconds) across 20 replications for each problem size. Each instance has a high pickiness level with a heterogeneity of 30%, a medium traffic level, a menu size of 3 requests (i.e., $\bar{b} = 3$), and an overlap of 3 suppliers (i.e., $\bar{a} = 3$). Problems in which up to 100 requests and 100 suppliers simultaneously need to be matched can be solved in less than 2 seconds (on average), after which the runtime grows exponentially. Hence, the MSA (especially with parallelization) can support real-time decision making in our experiments for up to 100 requests and 100 suppliers in the system simultaneously. Yet for situations in which greater than 100 requests and 100 suppliers simultaneously need to be matched, additional considerations may be required. For example, a platform could divide the suppliers and request sets into separate subsets to be solved independently using clustering methods (e.g., Zuo et al. 2021) or could use the proposed approach for only a subset of all the requests and suppliers in the system by creating a filtering preprocessing approach, e.g., platform lucrative request-supplier pairs with high likelihood of being selected could be instantly matched. Further, requests with small likelihood of selection or providing very low benefit to the platform could be removed from consideration. Both would lead to a reduced decision space.

To better understand what influences runtimes, we conduct a deeper analysis of the scalability of the MSA with each run being a single time period with a total of 200 simultaneous participants. We run a full factorial analysis considering each of the factors and levels denoted in Section 5.4, plus additional factors of the menu size ($\bar{b}$, with levels of 1, 3, 5, and 10), the overlap ($\bar{a}$, with levels of 1, 3, and 5), and different ratios of numbers of requests and suppliers (i) a balanced ratio (i.e., 100 requests and 100 suppliers), (ii)

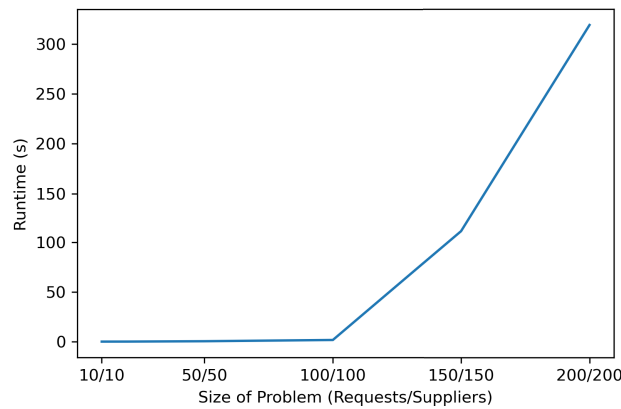


Figure 17 Average Runtime (in Seconds) for One Time Period for Various Problem Sizes (Requests/Suppliers)

a 2:1 request-to-supplier ratio (i.e., 133 requests and 67 suppliers), and (iii) a 1:2 request-to-supplier ratio (i.e., 67 requests and 133 suppliers). This resulted in 432 different instances. For each of these instances we ran 20 replications.

For each ratio of suppliers and requests, Table 13 presents the runtime statistics (minimum, mean, maximum, and standard deviation (std)) across each factor and their levels. For all ratio types, mean runtime increases with increased menu size and overlap, as the platform has more options to consider when creating the suppliers' menus. Similarly, mean runtime is higher when pickiness is low and occasionally when supplier heterogeneity is high (i.e., 30%). With low pickiness, the number of requests a supplier is willing to serve in the scenarios is higher which leads to an increase in potential menu decisions. Traffic level does not have a significant impact on runtime; although for unbalanced networks, systems with a low traffic level may take longer to solve, as supplier preferences are less uncertain in low traffic. Regardless of the factor, all instances with a balanced ratio having 100 requests and 100 suppliers simultaneously to match was solved in under one minute, and took on average 4.6 seconds to solve. Having more suppliers than requests increases the runtime compared to a balanced ratio. Yet, the unbalanced ratio with more suppliers than requests results in lower runtime than an equal number of suppliers and requests. While some instances with unbalanced ratios have quite high runtimes, on average, the runtimes were around 22 seconds for the 133 requests and 67 supplier cases, and around 3.5 seconds for the 67 requests and 133 suppliers cases. This indicates that the number of requests in the system is the primary driver of runtime as it increases the number of potential menus the most.

Table 13 Runtime Statistics (in Seconds) for 1 time period with 100 Requests and 100 Suppliers, 133 Requests and 67 Suppliers, and with 67 Requests and 133 Suppliers.

Factor	Level	100 Requests and 100 Suppliers				133 Requests and 67 Suppliers				67 Requests and 133 Suppliers			
		Min	Mean	Max	Std	Min	Mean	Max	Std	Min	Mean	Max	Std
Pickiness	Low	1.3	6.4	48.6	5.2	1.1	35.7	3661.7	110.5	1.1	5.1	125.2	7.0
	High	1.5	2.8	8.1	1.2	4.9	7.4	28.1	1.8	0.8	1.9	5.3	1.0
Traffic Level	Low	1.4	4.5	28.3	3.8	1.1	29.0	3661.7	135.9	0.8	4.2	125.2	7.4
	Med	1.3	4.7	36.0	4.4	4.9	18.0	129.6	14.3	0.8	3.3	61.4	3.6
	High	1.3	4.6	48.6	4.3	5.2	17.6	60.0	13.1	0.8	2.9	55.7	3.8
Heterogeneity	10	1.3	4.6	31.0	4.2	1.1	20.5	1626.9	53.2	0.8	3.4	61.4	4.0
	30	1.3	4.6	48.6	4.2	5.2	22.6	3661.7	99.0	0.8	3.6	135.2	6.3
Menu Size	1	1.3	2.7	6.7	1.1	1.1	8.1	19.1	3.2	1.1	2.4	5.8	1.1
	3	1.5	4.4	23.1	3.0	4.9	20.8	198.9	17.5	1.0	6.3	135.2	9.6
	5	1.5	6.2	48.6	6.2	5.4	22.0	155.3	17.3	0.8	3.5	14.7	2.2
	10	1.6	5.1	22.2	3.8	5.4	35.3	3661.7	155.8	0.8	1.8	5.9	1.0
Overlap	1	1.4	2.5	5.7	0.9	2.3	15.7	198.9	16.9	1.1	1.9	5.0	0.5
	3	1.3	4.2	23.1	3.1	2.4	17.4	84.7	13.9	0.8	3.0	19.6	2.6
	5	2.4	7.2	48.6	5.5	1.1	31.7	3661.7	135.3	0.8	5.5	135.2	8.3
Overall		1.3	4.6	48.6	3.6	1.1	21.6	3661.7	56.3	0.8	3.5	135.2	4.2

B. ONLINE Appendix

We first present the greedy consensus solution, followed by additional details to the changes to the Single Closest Request policy, and the Multi-Period IP. We then provide the results of the MSA with forecast, give more insight in the supplier unhappiness measure in the case of the CR-policy, investigate the actual menu sizes in case of overlap and no overlap, and finally present the ANOVA results in detail.

B.1. Greedy Consensus Solution

Algorithmically, the greedy consensus solution starts the same way as the consensus solution described in Section 4.4 (i.e., the non-greedy consensus solution), in that the frequency of the request r_{it} recommended to supplier s_{jt} at decision point t across all scenarios is computed, i.e., $Freq_{ijt} = \sum_{\sigma} x_{ijt}^{\sigma}$. The greedy consensus solution then consists of three steps. In the first step, for each supplier, their top $\bar{b}$ most frequent requests are added to their first tentative menu. In the second step, each request is added to their top $\bar{a}$ most frequent suppliers' second tentative menus. These first two steps reflect the maximum menu size constraint and the maximum overlap constraint, respectively. In the final step, for each supplier, requests that are in both the supplier's first tentative menu and second tentative menu are added to the supplier's final menu.

We run the MSA with the greedy consensus solution with the same set of experiments as Section 6.1. We then calculate the differences in KPIs of the MSA with the non-greedy consensus solution and the MSA with the greedy consensus solution. The results are presented in Table 14. The p-values found in the table are the results of performing a paired t-test on the two methods being compared.

Table 14 Average KPI Changes of MSA with Non-Greedy Consensus Solution Compared to MSA with Greedy Consensus Solution.

KPI	Avg. Change (%)	p-value
Platform Revenue	+0.6	0.07
Number of Assignments	+1.0	<<0.05
Supplier Unhappiness	-3.6	<<0.05

Although the non-greedy consensus solution does not statistically improve on the quality of the assignments (i.e., the p-value is greater than 0.05 for the platform revenue), it does statistically decrease the number of unhappy suppliers and increase the number of assignments. The non-greedy consensus solution statistically reduces the number of rejections by about 14.1%, while statistically increasing the number of failed selections by about 14.6%. Creating the suppliers' menus in a more holistic manner yields an increase in supplier participation, which accounts for both the increase in the number failed selections and the decrease in the number of unhappy suppliers.

B.2. Changes to Single Closest Request Policy Details

Figure 18 plots the distribution of the per assignment platform revenue for MSA and SR, for all assignments made in the instance setting with medium traffic, high pickiness threshold, and a 10% coefficient of variation (i.e., the same instance setting as in Figure 11). The revenue per assignment is depicted on the x-axis, and the corresponding number of assignments over all time steps and replications is shown on the y-axis. For this instance setting, the average revenue per assignment for SR with 131.2 revenue units is higher than for the

MSA with 125.5. This phenomenon of SR achieving a higher average revenue per assignment than MSA but a lower overall revenue than MSA, is statistically significant over all instances. While both policies target assignments with high revenues and have a large number of assignments between the values of 120 and 140, we observe two distinct differences between the two policies. First, the number of cases with assignments less than 120 in revenue is substantially higher for the MSA. Second, the number of cases with highest revenue (around 140) for SR is slightly higher compared to the MSA. This indicates that the holistic view of the MSA may occasionally sacrifice a high reward assignment for the benefit of more assignments with less average reward. By refraining from “cherry picking”, MSA achieves more assignments and eventually, higher overall revenue.

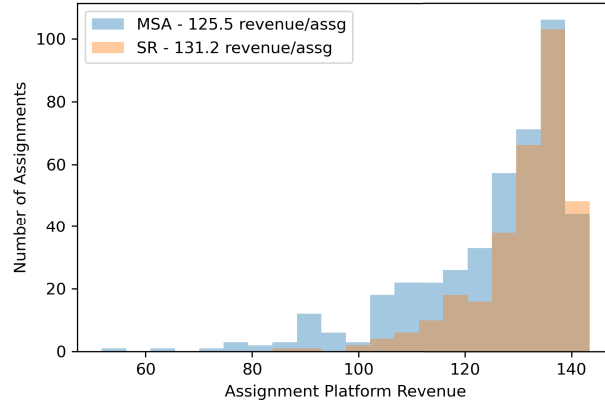


Figure 18 Histograms of Platform Revenue Per Assignment for MSA(3,3) and SR.

B.3. Multi-Period IP

In this section, we present the multi-period integer program required to calculate the upper bound solutions.

Additional/New Notations.

Sets:

- $R = \{1, \dots, m\}$ set of requests
- $R' = \{1, \dots, m'\}$ set of dummy requests
- $N = \{0\}$ set of no-choice alternative
- $A = R \cup N$ set of alternatives
- $S = \{1, \dots, n\}$ set of suppliers

Parameters:

- δ_{it} 1, if request $i \in R$ is present (or forecast to be present) in the system in time period t ; 0, otherwise
- p_{jt} 1, if supplier $j \in S$ is present (or forecast to be present) in the system in time period t ; 0, otherwise

Decision Variables:

- ψ_{it} 1, if request $i \in R$ is in the system and has yet to be assigned in time period t ; 0, otherwise
- ρ_{jt} 1, if supplier $j \in S$ is in the system and has yet to be assigned in time period t ; 0, otherwise

Multi-Period IP for UB.

$$\max \sum_{i \in R} \sum_{j \in S} \sum_{t=1}^T c_{ijt} y_{ijt} \quad (27)$$

subject to:

$$\sum_{j \in S} x_{ijt} \leq \bar{a} \psi_{it} \quad \forall i \in R, \forall t = 1, \dots, T \quad (28)$$

$$\sum_{i \in R} x_{ijt} \leq \bar{b} \rho_{jt} \quad \forall j \in S, \forall t = 1, \dots, T \quad (29)$$

$$x_{ijt} = \rho_{jt} \quad \forall i \in N, \forall j \in S, \forall t = 1, \dots, T \quad (30)$$

$$y_{ijt} \leq \omega_{ijt} \quad \forall i \in R, \forall j \in S, \forall t = 1, \dots, T \quad (31)$$

$$\sum_{i \in R} \sum_{t=1}^T y_{ijt} \leq 1 \quad \forall j \in S \quad (32)$$

$$\sum_{j \in S} \sum_{t=1}^T y_{ijt} \leq 1 \quad \forall i \in R \quad (33)$$

$$\psi_{i1} = \delta_{i1} \quad \forall i \in R \quad (34)$$

$$\psi_{it} \leq \delta_{it} \quad \forall i \in R, \forall t = 2, \dots, T \quad (35)$$

$$\delta_{i,t+1} - \delta_{it} \leq \psi_{i,t+1} \quad \forall i \in R, \forall t = 1, \dots, T-1 \quad (36)$$

$$\rho_{j1} = p_{j1} \quad \forall j \in S \quad (37)$$

$$\rho_{jt} \leq p_{jt} \quad \forall j \in S, \forall t = 2, \dots, T \quad (38)$$

$$p_{j,t+1} - p_{jt} \leq \rho_{j,t+1} \quad \forall j \in S, \forall t = 1, \dots, T-1 \quad (39)$$

$$\psi_{i,t+1} \leq 1 - \sum_{j \in S} y_{ijt} \quad \forall i \in R, \forall t = 1, \dots, T-1 \quad (40)$$

$$\psi_{i,t+1} \leq 1 - \delta_{it} + \psi_{it} \quad \forall i \in R, \forall t = 1, \dots, T-1 \quad (41)$$

$$\rho_{j,t+1} \leq 1 - \sum_{i \in R} y_{ijt} \quad \forall j \in S, \forall t = 1, \dots, T-1 \quad (42)$$

$$\rho_{j,t+1} \leq 1 - p_{jt} + \rho_{jt} \quad \forall j \in S, \forall t = 1, \dots, T-1 \quad (43)$$

$$\delta_{i,t+1} - \psi_{i,t+1} \leq 1 - \psi_{it} + \sum_{j \in S} y_{ijt} \quad \forall i \in R, \forall t = 1, \dots, T-1 \quad (44)$$

$$p_{j,t+1} - \rho_{j,t+1} \leq 1 - \rho_{jt} + \sum_{i \in R} y_{ijt} \quad \forall j \in S, \forall t = 1, \dots, T-1 \quad (45)$$

$$\sum_{i \in A} \alpha_{ijk} x_{ijt} - \sum_{i \in A} \beta_{ijk} \omega_{ijt} \leq 0 \quad \forall j \in S, \forall k \in A, \forall t = 1, \dots, T \quad (46)$$

$$\omega_{ijt} \leq x_{ijt} \quad \forall i \in R, \forall j \in S, \forall t = 1, \dots, T \quad (47)$$

$$\sum_{i \in A} \omega_{ijt} \leq 1 \quad \forall j \in S, \forall t = 1, \dots, T \quad (48)$$

Constraints (28)-(33) and (46)-(48) are the same as in Section 4.3. The other constraints are new. Specifically, constraints (34)-(39) represent the arrivals of current and forecasted suppliers and requests. Constraints (40)-(43) describe the departures of suppliers and requests in case of either an assignment, i.e., if a supplier or a request is assigned in a time period, then they depart and are no longer present in the next time period

and in the time periods thereafter. Constraints (44) and (45) represent the no assignment constraints, i.e., if a supplier or a request is not assigned in a time period, then they may be present in the next time period, unless they no longer want to wait to be assigned.

B.4. MSA with Forecast

Our method focuses on the novel and disruptive uncertainty of supplier selection behavior in the current period. However, there may be value in extending the scenarios beyond the current period by sampling departures and arrivals of suppliers and requests, as well as future suppliers' utility values in future periods. To this end, we augment the scenarios and IP of the MSA by one period. The scenarios now also capture the potential departure of unassigned requests and suppliers and the arrival of new requests and suppliers. The IP not only determines the assignments of the current period but also the assignments of the next period. The IP is therefore a special case of the IP presented in Section B.3 used to derive the upper bound. The consensus solution remains the same. We call this extended method "MSA with forecast".

Several modifications are made to the multi-period IP of Section B.3 to solve the MSA with forecast. Firstly, the current time period is considered as $t = 1$ and the forecasted periods as $t > 1$. Secondly, the MSA with forecast utilizes the same expected values of stochastic parameters as the MSA described in Section 4.3, as well as the same decision variables associated with scenario σ . A scenario σ samples the current suppliers' estimated utility values, the forecasted suppliers' estimated utility values, as well as the departures and arrivals of suppliers and requests in the forecasted periods. If a supplier is present in the current time period, its estimated utility values from the current time period do not change in the forecasted periods. This assumption is applied by replacing constraints (46) with constraints (50)-(57) and introducing the decision variables q_{ijkt}^α , an integer in $[-1, 1]$, and q_{ijkt}^β which is binary. The parameters α_{ijk} and β_{ijk} of the first time period are denoted with the superscript (1) . Finally, the menu constraints in the current time period include the dummy requests. Accordingly, the penalty term $-M \sum_{i' \in R'} \sum_{j \in S} \dot{x}_{i'j}^\sigma$ is added to objective function (27), and constraints (29) are replaced with constraints (7) for the current time period (i.e., $t = 1$) and constraints (49) for the forecasted periods.

$$\sum_{i \in R} x_{ijkt}^\sigma \leq \bar{b} \rho_{jt}^\sigma \quad \forall j \in S, \forall t = 2, \dots, T \quad (49)$$

$$q_{ijkt}^\alpha \leq x_{ijkt}^\sigma \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (50)$$

$$-x_{ijkt}^\sigma \leq q_{ijkt}^\alpha \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (51)$$

$$q_{ijkt}^\alpha \leq \alpha_{ijk}^{(1)} \rho_{ij1} + \alpha_{ijk} (1 - \rho_{ij1}) + (1 - x_{ijkt}^\sigma) \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (52)$$

$$\alpha_{ijk}^{(1)} \rho_{ij1} + \alpha_{ijk} (1 - \rho_{ij1}) - (1 - x_{ijkt}^\sigma) \leq q_{ijkt}^\alpha \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (53)$$

$$q_{ijkt}^\beta \leq \hat{\omega}_{ijkt}^\sigma \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (54)$$

$$q_{ijkt}^\beta \leq \beta_{ijk}^{(1)} \rho_{ij1} + \beta_{ijk} (1 - \rho_{ij1}) \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (55)$$

$$\beta_{ijk}^{(1)} \rho_{ij1} + \beta_{ijk} (1 - \rho_{ij1}) - (1 - y_{ijkt}) \leq q_{ijkt}^\beta \quad \forall i \in R, \forall j \in S, \forall k \in A, \forall t \in T \quad (56)$$

$$\sum_{i \in A} q_{ijkt}^\alpha - \sum_{i \in A} q_{ijkt}^\beta \leq 0 \quad \forall j \in S, \forall k \in A, \forall t \in T \quad (57)$$

We compare this MSA with forecast to our original MSA. Due to the computational burden of solving the IPs with forecast, we focus on the less disruptive instance settings where supplier behavior can be anticipated relatively well (COV 10%, low/medium traffic levels). If the MSA with forecast does not show any benefit here, it is likely that it also will not show benefit for the more uncertain, more disruptive instances. Table 15 presents the MSA’s objective value for each combination of factors, as well as the objective value percentage difference with the MSA with forecast and its significance (i.e., p-value). The MSA does not always perform statistically better than the MSA with forecast; however, the MSA with forecast never performs better than the MSA. Thus, for the problem at hand, anticipation beyond the current time period is very challenging most likely because of the manifold uncertainties in suppliers and demand.

Table 15 Average Objectives and p-values of MSA w/o Forecast v. MSA w/ Forecast Across Various Conditions.

No Choice Threshold	COV	Traffic Level	MSA	MSA w/ Forecast	
			Obj.	Diff.	p-value
low	10%	low	9388	-3.5%	5.8E-04
		med	9124	-1.4%	0.05
high		low	8747	-1.6%	0.11
		med	8174	-0.5%	0.55
low	30%	med	8998	-1.5%	2.5E-02
high		med	8012	-0.6%	0.65
AVERAGE			8741	-1.5%	

B.5. Supplier Unhappiness for the Closest-Request Policy

In this section, we compare the supplier unhappiness of the MSA with the unhappiness of the CR policy. The following figures are similar to the unhappiness figures in the main body of the paper (e.g., see Figure 7). The MSA has slightly more rejections as shown in Figure 19 and significantly less failed selections as shown in Figure 20 than the CR. Because the holistic menu of the MSA refrains from too many duplicate offers, a supplier might not always be offered the “obvious” choice, hence, the increase in rejections. However, the holistic menu avoids too many suppliers selecting the same requests, and as a result, the MSA significantly decreases supplier unhappiness, as shown in Figure 21.

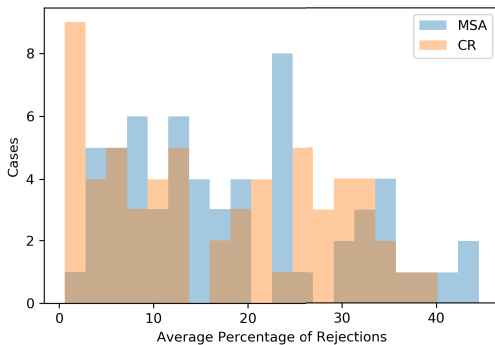


Figure 19 Average Percentage of Rejections for MSA(3,3) and CR.

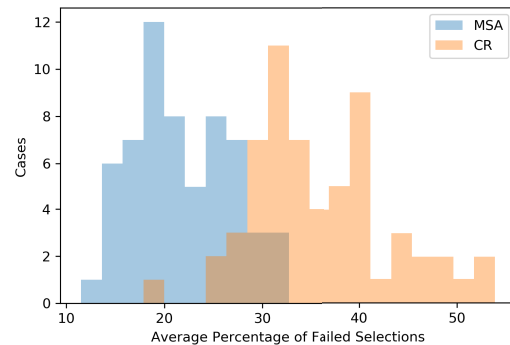


Figure 20 Average Percentage of Failed Selections for MSA(3,3) and CR.

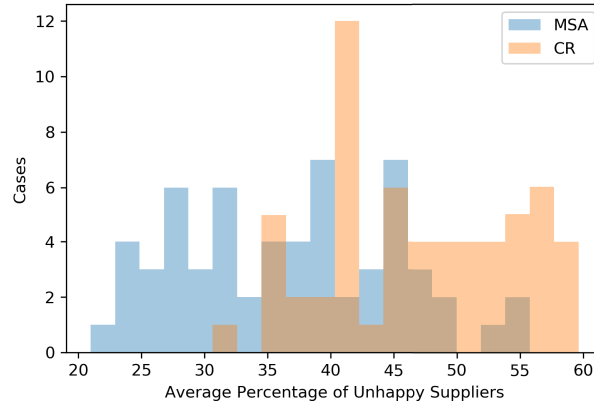


Figure 21 Average Percentage of Unhappy Suppliers for MSA(3,3) and CR.

B.6. Actual Menu Sizes

In this section, we present the actual menu sizes for policies MSA(5,3), MSA(5,1), MSA(3,3), MSA(3,1), MSA(1,3), and MSA(1,1) for the balanced and unbalanced networks. The results are depicted in Figure 22. Each plot shows the menu sizes for one of the three network types. In each plot, six MSAs are compared, differing in menu size and overlap. The x-axis lists the varying menu sizes $\bar{b}$. As a reminder, our methodology sets a maximum menu size; however, individually, each supplier can get a menu of size less than the maximum. The red boxplots indicate MSAs with no overlap allowed ($\bar{a} = 1$), and the purple boxplots indicate MSAs with overlap $\bar{a} = 3$. Generally, we observe that when there is no overlap among the menus, an input menu size greater than 1 provides menus that list usually between 1 and 1.5 requests on average. However, when overlap is allowed, an input menu size greater than 1 results in menus with sizes significantly higher than 1. Thus, a lack of overlap is a limiting factor in the number of requests we can offer to a supplier. This explains the inferior performance of MSAs without overlap observed previously in this paper. Notably, for the network type with a surplus of requests, even without overlap, larger menus of about 1.5 requests on average can be achieved. With more requests than suppliers in the system, the necessity for duplicate recommendations decreases.

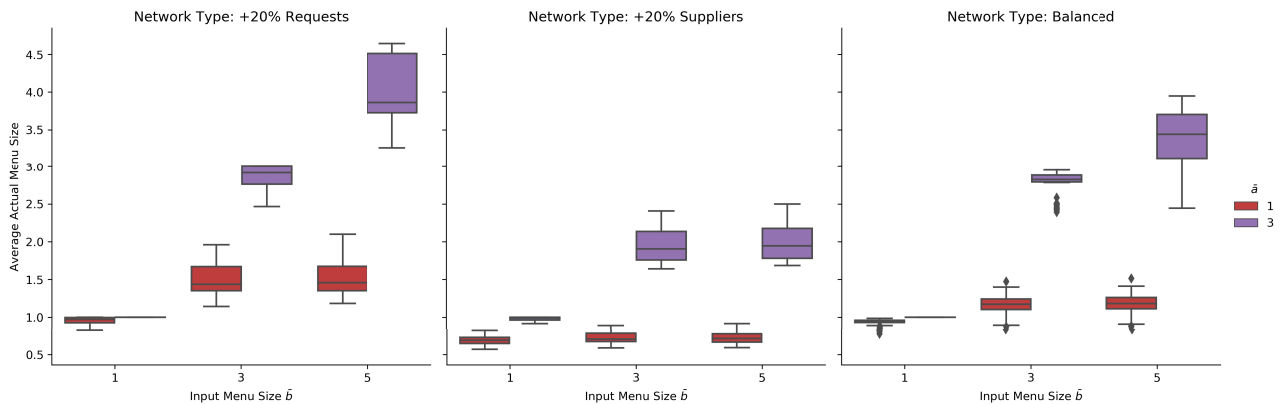


Figure 22 Input Menu Size v. Average Actual Menu Size.

B.7. ANOVA Results

We ran a two-way ANOVA with the objective value of the MSA as the response variable. The independent variables considered were the menu size (b), the overlap (a), the pickiness level, its coefficient of variation, the traffic level, and the network type. To meet the assumptions of the ANOVA, we normalized the objective value. For the ANOVA results found in Table 16, we have all the main factors, as well as the second and third order interactions that only include the menu size (i.e., b) and/or the overlap (i.e., a). For the ANOVA, we use a significance level of 0.05, and the factors denoted by a star (“*”) have a p-value less than 0.05. For example, there are significant differences between the levels of the factor ”b:a:pickiness”, which represents the interaction between the menu size, the overlap, and the pickiness level. Most of the variability in the objective values stems from the suppliers’ pickiness level and the traffic level, as these factors have the largest mean squares values.

Table 16 Normalized Objective Function Value ANOVA Results.

Main and Interaction Factors	Degrees of Freedom	Sum of Squares	Mean Squares	F value	p-value	Significance
b	2	27	13	97.4	2.0E-16	***
a	1	29	29	211	2.0E-16	***
pickiness level	1	423	423	3112	2.0E-16	***
traffic level	2	321	160	1179	2.0E-16	***
network type	2	49	25	182	2.0E-16	***
coefficient of variation	1	9.4	9.4	69.3	3.2E-16	***
b:a	2	9.4	4.7	34.5	4.0E-15	***
b:pickiness	2	4.3	2.1	15.7	2.1E-07	***
a:pickiness	1	9.7	9.7	71.3	2.0E-16	***
b:traffic level	4	7	1.8	12.9	3.3E-10	***
a:traffic level	2	9.9	5	36.4	6.8E-16	***
b:network type	4	6.4	1.6	11.7	3.1E-09	***
a:network type	2	16	8	58.9	2.0E-16	***
b:a:pickiness	2	4	2	14.8	5.0E-07	***
b:a:traffic level	4	2.7	0.7	4.94	6.1E-4	***
b:coefficient of variation	2	0.2	0.1	0.781	0.46	
a:coefficient of variation	1	0	0	0.287	0.59	