A Simulation Approach to Designing Digital Matching Platforms

Andrey Fradkin*
Boston University
fradkin@bu.edu

January 21, 2019

Abstract

Digital matching marketplaces are characterized by user heterogeneity, limited capacity, and dynamic market clearing. These features create spillovers between users. For example, an Airbnb listing booked by one guest cannot be booked by another guest for the same night. Spillovers limit the applicability of many experimental and observational methods for evaluating the effects of marketplace policies. In this paper, I show how to use marketplace simulations as an input into the design of user acquisition strategies and ranking algorithms. I calibrate a marketplace simulation using data on searches and transactions from Airbnb and use it to address three topics: the returns to scale in matching, the heterogeneity in returns to user acquisition, and the size of bias in experimental designs. I find that returns to scale are initially increasing due to market thickness effects and then decreasing due to availability frictions in search. Furthermore, heterogeneity in the value of listings to the platform is large — the effect of acquiring 25% more listings on bookings varies between -4.1% and 5.4% depending on the quartile of listing quality. I then measure the extent of bias in experimental treatment effects due to spillovers. The treatment effect of a better ranking algorithm on conversion rates is overstated by 53% when a quarter of users are randomized into treatment.

*This paper supersedes the second half of "Search Frictions and the Design of Online Marketplaces." I thank Liran Einav, Caroline Hoxby, Jon Levin, Luigi Pistaferri, Susan Athey, Mike Dinerstein, Chiara Farronato, John Horton, Ramesh Johari, Scott Kominers, and numerous seminar participants for advice. I am grateful to Airbnb (especially the analytics and data infrastructure teams) for providing me with the data and for helping me execute every aspect of this project. Financial support was provided by the SIEPR Graduate Fellowship, the B.F. Haley and E.S. Shaw Fellowship for Economics, the NET Institute, and Airbnb. The author was previously employed by Airbnb, Inc. but the paper represents the views of the author and not necessarily Airbnb, Inc.

1 Introduction

Digital matching platforms intermediate markets for jobs, rides, short-term rentals, and dates. Designing these platforms is hard because of user heterogeneity, limited capacity, and a dynamic matching process. In this paper, I propose and demonstrate a data-driven simulation-based approach for informing platform design decisions in the dynamic matching setting. I use this simulation to study the returns to scale in matching, the heterogeneity in the value of users, and the effects of ranking algorithms on Airbnb.

To understand the challenge of evaluating matching platform designs and why marketplace simulations are useful, consider the problem of measuring the returns to the platform of an extra guest on Airbnb. Whenever a guest books a listing, other potential guests can no longer match with the same listing for that night and consequently face fewer choices. In contrast, the remaining listings face less competition. These spillovers create violations of the Stable Unit Treatment Value Assumption (SUTVA), which is needed for experimental treatment effects to be unbiased. Consequently, the benefit of an additional guest cannot be accurately measured using only the revenue generated by that guest's transactions. More generally, the user level experiments ('A/B tests') that many digital platforms conduct result in biased estimates in matching markets.[1]

The bias from these spillovers plays out in two ways. First, conditional on a given treatment probability within a market, the average difference between treated and control units does not correspond to the effect of the experiment on the market. Second, the per unit effect of the treatment depends on the share of the population that is treated.

One potential solution to the problem of spillovers is to run experiments randomized at the group or market level (e.g. Hudgens and Halloran [2008]). While this approach can be fruitful, finding similar but separated markets is often difficult. For example, it is unclear how Airbnb would find a set of treatment and control cities similar to New York City. There can also be logistical challenges to these experiments due to engineering challenges and the difficulty of replicating treatments across markets. Lastly, the number of experiments that can be run is limited by the number of independent markets and the length of time needed to run the experiment.

As a complement to experiments, marketplace simulations give market designers the ability to explore the effects of proposed policies without having to spend engineering and operational resources implementing them. A similar strategy is used in the design of autonomous vehicles.[2] Simulation can be combined with experiments in the following way. A user level experiment of a policy such as a matching algorithm is conducted. The results from this experiment are then used to get a biased measure of a treatment effect and to help in the estimation of the parameters of models of individual behavior. The simulation is then created using models of individuals behavior

[1] See Bakshy et al. [2014] for a description of how user level 'A/B tests' are typically conducted at Facebook.

and used to obtain a measure of the bias in the experimental treatment effect due to spillovers. This informs the benefits from a policy and whether it should be launched. The simulation can also be used to explore policies which are candidates for future experiments. The most promising of these can be launched as experiments in the future.

I design such a simulation and use it to address three important questions in digital marketplace design within the context of Airbnb. The first question concerns the extent of returns to scale in the matching process. I find that, on average, returns to scale are close to one and that returns to scale are increasing when the market size is small and start decreasing as the market size increases. Second, I measure the heterogeneity in the benefits to acquiring more users. Guests are relatively more important in creating matches than hosts at the observed market conditions. However observable characteristics are poor at differentiating between guests who are more or less likely to match. In contrast, there is substantial predictable variation in the returns to additional listings depending on their characteristics. Lastly, I propose and evaluate a new ranking algorithm. I find that this algorithm improves the rate of matches by 34% but that an experiment with 25% of users treated would overstate the effect of the algorithm by 53%. Better ranking algorithms also increase the returns to scale in the market and the relative importance of listings in creating matches.

The simulation consists of three types of user behavior: the browsing behavior of guests, the choices of guests regarding which host(s) to contact, and the host's choice of whether to accept or reject a request to book. This paper builds on Fradkin [2018], which describes the search and matching process on Airbnb but does not model marketplace dynamics. Notably, that paper establishes that guest search is limited, that hosts frequently reject guests, and that rejection causes guests to leave the platform without transacting. These facts provide a motivation for developing market designs that improve matching between guests and hosts.

The first aspect of the simulation is the search process for guests. I observe all searchers looking to stay in Chicago in the month following January 21, 2014 as they submit queries to the Airbnb search engine and attempt to book specific listings. I separate this process into two parts: consideration set formation (browsing) and sending inquiries. Consideration set formation is modeled as a Markov process in user actions. In the data, I observe the next action of a searcher as a function of the number of results, the filters applied, and the order of the search. I estimate transition probabilities as a function of these variables and use these to simulate the search process. Each search action returns a set of listings which fulfill the filtering criteria (e.g. price, room type, and neighborhood filters) and are ranked according to a ranking algorithm. The search process eventually ends and the set of all listings which were shown to a given searcher comprise a consideration set. The searcher

[2] According to Waymo, "the vast majority of work done – new feature work – is motivated by stuff seen in simulations." Waymo's self-driving cars logged 2.5 billion virtual miles in 2016 relative to 3 million real world miles. `https://bit.ly/2vpoOoR`. Farronato et al. [2018] also discuss simulations conducted by Uber to determine the parameters of their Uber Pool Express matching algorithm.

then chooses which of these listings to request to book according to a separately estimated discrete choice model. Importantly, unobserved user heterogeneity in preferences is modeled by drawing searcher preferences for location, price, and room type from the observed distribution of filtering behavior. If a guest is accepted, the listing is removed from subsequent searchers' results after a randomly determined period reflecting the time it takes for both sides to agree to a transaction.

Once searchers send inquiries, hosts make the decision to accept each one sequentially. There are three reasons why hosts reject guests. The first is congestion, which occurs when two searchers try to book the same listing in the same period. Because hosts can only accept one of these inquiries, the non-transacting inquiries must be rejected. The frequency of congested inquiries is determined endogenously in the simulation. Intuitively, the more searchers there are relative to listings, the higher the probability that an inquiry overlaps with another one. Second, listings may have stale vacancies. These occur when a host who is unavailable does not update their calendar accordingly. I assume that the timing at which hosts update their calendars in the simulation is the same as in the data. While the timing of these calendar updates is not modeled, the frequency at which searchers see stale listings is determined endogenously by the search behavior of guests, the ranking algorithm, and by the prior transactions in the market. The last form of rejection occurs when hosts screen searchers based on their characteristics. I model the decision to accept or reject an inquiry which has not been rejected for other reasons as a logistic regression.

The last step of initiating the simulation requires the calibration of parameters relating to the process of searching. These include the probability of searching again after a rejection, the share of searches with a strict geographic filter, the types of inquiries that are sent, the number of inquiries that are sent, and the probability of booking after an acceptance.

The reason that the above procedure can overcome the problem of spillovers is that it models individual behavior relative to the state of the market when the individual is making a decision. For example, an individual searcher makes browsing and booking decisions relative to the listings that are visible and available in the market at the time of the search. In the data, searchers are never shown listings that have been booked prior to the time of the search.[3] Consequently, the estimated parameters that govern an individual's decisions are not affected by cannibalization as long as the state of the market is properly accounted for and the spillovers are asynchronous. There are other types of spillovers which occur in the longer-run as market participants learn about policy changes and adjust their beliefs. I do not model these spillovers.

I evaluate the fit of the simulation to the data by comparing outcomes for four weeks in my sample with actual outcomes for those weeks. Although the parameters of the simulation were not explicitly chosen to match booking or contacting rates, the majority of actual market outcomes

[3] In some cases, a listing may be booked while the searcher is browsing it. If such cases occurred frequently, then they would cause problems for the above strategy.

come within a standard deviation of simulation outcomes for each of the four weeks. This suggests that the simulation is doing a reasonable job of capturing the market clearing process observed on Airbnb.

I then use the simulation to study the marketplace. I first conduct a set of exercises to determine the importance of rejection for matching. In one exercise, I study what would happen if the marketplace removed all hosts who would reject a given guest from that guest's search results. Under this scenario, the booking rate would increase by 15%. I show that most of this improvement is accounted for by stale vacancies and screening hosts rather than by congestion. I then evaluate the effects of an improved ranking algorithm and show that it improves booking rates by over 34%.

Next, I investigate the returns scale to matching in this marketplace. The standard way to measure returns to scale in matching is to estimate a Cobb-Douglass matching function using observational data from across markets and over time.[4] The Cobb-Douglass matching function provides a useful way to summarize matching data, even if it is a simplification of a complex underlying process. A shortcoming of observational methods estimating Cobb-Douglass matching functions that they may suffer from endogeneity issues and that marginal participants that generate the variation on each side of the market may be different from the average participant. A simulation approach can overcome these hurdles because outcomes can be investigated by exogenously re-sampling or removing agents from the market and measuring outcomes.

I first simulate outcomes by randomly re-sampling or removing buyers and searchers from the market at ratios ranging from -75% to 100%. I then estimate a Cobb-Douglass matching function using a linear regression on the simulated outcomes, treating each simulation run as an observation. I find that returns to scale are slightly increasing with a scale coefficient of 1.029 and that searchers account for .937 of that coefficient. These returns to scale are not constant with marketplace policy, market size, or the types of agents in the market. For example, returns to scale increase to 1.086 and the relative importance of searchers decreases when a better ranking algorithm is introduced into the simulation. Intuitively, the ranking algorithm does a better job of showing relevant listings to searchers, which means that searchers are more likely to see a good match and that fewer listings are required to generate the same number of bookings. I also find that returns to scale are relatively large when the overall market size is small, but become decreasing as the overall market size grows. Lastly, the effect on bookings of drawing listings from the bottom quartile is -4% while the effect on bookings of drawing listings from the top quartile is 5.4%. The negative effect of worse listings comes from the fact that listings which are either undesirable or selective replace better listings in the consideration set and cause searchers to transact at lower rates.

Next, I consider the spillovers caused by violations of SUTVA in user level experiments of ranking algorithms. I use the framework of Hudgens and Halloran [2008] to guide the calculation

[4] See Petrongolo and Pissarides [2001] for an early overview of matching function estimation in the labor market.

of effects for each ranking and treatment probability. I find that the direct effect (measured by the difference between treated and control groups) of a better algorithm on booking rates is 9.2 percentage points when the ranking algorithm treats 25% of users. However, the overall effect of a fully launched algorithm on the booking rate is 6.0 percentage points. The overstatement of effects can be consequential for choosing which policies to launch and for evaluating the benefits of better marketplace design.

This paper directly contributes to a growing literature regarding spillovers and matching in digital platforms. Blake and Coey [2014] show that interference occurs in auctions experiments on Ebay. Holtz [2018] conducts simulations to ascertain the statistical properties of cluster randomized experiments in marketplaces and Saveski et al. [2017] study cluster-randomized experiments on LinkedIn. Horton [2018] empirically shows that the availability friction is important on digital labor markets and studies how availability signaling can help. Cullen and Farronato [2018] estimate a Cobb-Douglass matching function using data from a digital market for local services. They find returns to scale that are just below 1 but are unable to reject constant returns to scale. An important difference is that on Airbnb, sellers have unit capacity while in the above case, sellers' capacity constraints are not important. Relatedly, Roth and Xing [1997] use a similar simulation strategy to study the market for clinical psychologists.

Hitsch et al. [2010] study how closely decentralized matching in the dating market comes to achieving a stable match. Cheron and Decreuse [2017] and Albrecht et al. [2017] study 'phantom' vacancies and show that they have a large effect on matching both theoretically and in a model calibrated to the US labor market. Farronato and Fradkin [2018] estimate a model of the equilibrium effects of Airbnb's entry but abstract away from the search and matching process.

With regards to theoretical contributions, Arnosti et al. [2014] present a theoretical model of matching in a peer-to-peer market use it to show that the equilibrium in the model is inefficient due to uncertain availability. They show that limiting applications can improve market efficiency. Kanoria and Saban [2017] show that restricting actions by allowing only the more selective side to contact the other, as with guests on Airbnb, can improve social welfare. In the current paper, overlapping inquiries account for a small share of rejections and consequently limiting applications is unlikely to improve market outcomes. Ashlagi et al. [2018] study communication protocols for achieving stable matchings in decentralized matching markets. These contributions relate to the canonical models of platform design (e.g. Rochet and Tirole [2003]) by richly modeling direct and indirect network effects.

There is also a literature on search in digital settings. Dinerstein et al. [2018] study a redesign of the search engine on eBay and how it affects matching and pricing and Lewis and Wang [2013] theoretically study how lowering search costs affects the size and split of surplus across markets. Bronnenberg et al. [2016] study search in the market for digital cameras and find that consumers consider products in a limited attribute space and narrow their search criteria over the course of a search. Chen and Yao [2017], los Santos and Koulayev [Forthcoming], Ghose et al. [2014], and Ursu [2018] study the effects of search rankings in the hotel industry. In contrast to some of the above papers, my paper takes a statistical approach to modeling consideration set formation. This modeling decision allows me to use rich browsing and filtering data in the model. Incorporating this data into a rational expectations search model would be difficult. It is also not clear how searchers could form rational expectations over treatments in the short-run experiments conducted by digital matching platforms. A final factor affecting my choice of model is scalability. A platform like Airbnb must design policies at a global level, and we don't yet know how to scale the estimation of structural, rational expectations search models.

The rest of the paper proceeds as follows. Section 2 gives additional details regarding Airbnb's platform. Section 3 describes how the simulation was designed. Section 4 describes the results derived from simulating market outcomes. Section 6 discusses the results pertaining to spillovers in experiments and their causes. Section 7 discusses how these simulation results can inform marketplace design and directions for future work.

2 Setting

Airbnb describes itself as a trusted community marketplace for people to list, discover, and book unique accommodations around the world – online or from a mobile phone. In addition to a search engine, Airbnb operates a reputation and fraud detection system, customer service, a communications platform, a mobile application, an insurance policy for hosts, and a transaction processing platform. As of 2014, a typical Airbnb transaction consists of the following steps:[5]

1. Using the Search Engine (Figure 1) - Searchers enter the travel dates, number of guests and location into a search engine and receive a list of results. The search can then be refined using filters. Only listings that have an 'open' calendar for the trip dates are potentially loaded in the search results. These listings are displayed according to a ranking algorithm. Calendar dates become unavailable either when a listing is booked or when a host updates the calendar to be unavailable. Importantly, calendars are frequently not an accurate representation of true availability because hosts do not always attend to their calendar or because hosts may be in conversation with other potential guests (either on or off of Airbnb the platform).

2. Investigation (Figure 2) - The searcher clicks on a listing in search. The subsequent page displays additional photos, amenities, reviews, house rules and information about the host. This paper does not model the process of investigation.

[5] Note, that the booking flow has evolved over time and looks differently as of 2018.

3. Communication and Booking (Figure 3) - The searcher sends a message to hosts inquiring about room details and availability. This can be done in one of two ways, either by sending an inquiry or by clicking the "Book It" button. In the case of an inquiry, a host will typically reply with an acceptance or a rejection. If accepted, the guest can then click the "Book It" button to go through with the booking. A host who has received either type of request has the right to make a final decision of whether to accept or reject. There are two exceptions. First, some hosts are available to be "Instant Booked" by some guests, in which a transaction is confirmed as soon as guests click "Instant Book". Second, a host can "Pre-approve" a guest after an inquiry, which subsequently allows the guest to book without further communication.

4. Stay - There is frequently communication regarding the key exchange and details of the trip. Either party can cancel a booking with a pre-specified cancellation penalty (a monetary amount for a guest and an Airbnb specific punishment for the host). Following the transaction, guests and hosts can review each other.

3 Simulation Setup

This section displays the data and models which are used to construct the simulation. Before delving into the model details, I describe why each part of the model is necessary and how these parts fit together. Consider the problem of modeling the effect of an additional listing on market outcomes. The listing only affects the choices of searchers if that listing is shown to the searcher. Since a majority of listings are not seen by searchers (Fradkin [2018]), a model of exposure to listings is needed in addition to a model of choices. This model needs to capture the fact that searchers submit multiple, potentially redundant search queries to Airbnb and that the listings returned by Airbnb for those queries are determined by an algorithm. A model of that algorithm also needs to be part of the simulation.

Next, conditional on a set of listings to which searchers are exposed, the searcher needs to make a choice of which listings to contact. The effect of an additional listing will be determined by the propensity of searchers to contact that listing conditional on exposure. I model the searcher's decision as a discrete choice.

For each inquiry and listing pair, the simulation needs to have a procedure for keeping track of the decision of hosts to accept or reject. At the beginning of each simulation, I assign each listing an availability status based on the observed availability in the data for a given week. A listing is available if it has not yet been booked, and if the host never updates the calendar to be unavailable in the data. A listing is unavailable and visible in search results if the host updated the listing's calendar to be unavailable after the time at which they searcher entered the market. This type of listing can be seen in the search results up to the point at which the host updated it to be unavailable.

Conditional on being available in general, a host may still reject an inquiry under two conditions. First, if the host has recently accepted another searcher then the host must reject subsequent inquiries. Second, the host may reject if she does not like the characteristics of the searcher or trip. This is modeled as a logistic regression. If a host accepts an inquiry, the listing is removed from subsequent search results after a randomly determined period. Lastly, even if accepted, some guests still don't book. I set the probability that this happen to be equal to 32%, as observed in the data.

Figure 4 displays the set of possible simulation outcomes for each searcher. To simulate the market, I take all of the observed searchers in the month leading up to a check-in-week and simulate their decisions. These searchers can potentially see and choose the listings which were present and potentially visible on the site at the time of search. The simulation constitutes a dynamic process by which searchers book listings, which are then removed from the search results of subsequent searchers. Once the last searcher prior to the check-in date conducts their search, the simulation ends, and the market level outcomes are finalized.

3.1 Browsing and Ranking Models

In this section, I describe the process by which searchers form a consideration set. The consideration set consists of all listings that were displayed to a searcher in the two days leading up to that searcher either sending an inquiry or stopping searching. This consideration set is conditional on a city and check-in week pair. Up to 18 listings are shown for each search throughout most of my estimation sample. The ordering of these listings is determined by a ranking algorithm.

The searcher has three types of search actions to choose from. The first action type is to input a search with new query parameters such as price, neighborhood, or room type filters. The second action type is to select the next page (e.g. page 2) of results for a given search query. The last action type is to stop browsing and to either send inquiries to hosts or leave the site without transacting. I model this process as a multinomial logistic regression where the probability of each action is a function of the characteristics of the prior search, which include the log of the number of prior queries the searcher conducted, whether the searcher applied one of several filters, and whether the number of results returned on the prior search was less than 10. These explanatory variables capture aspects of state dependence in search, such as the fact that search behavior changes with the length of the search and with the amount of results displayed in each search. Intuitively, seeing few results should encourage searchers to change the query parameters or to stop searching.

Conditional on choosing new query parameters, the searcher must choose which query to conduct. In order to model this, I assume that each searcher has latent preferences for price, location, and room type. These preferences are drawn from the observed frequencies of filtering in the data. The query parameters for a search are drawn in two steps. First, a combination of filter types to apply is drawn according to the probability those filters are applied at a given search number. Second, conditional on a set of filter types to apply, the specific filter to apply (e.g. price $<$ \$200) is determined according to the latent preferences which are randomly drawn at the beginning of the simulation for each searcher. The probability that the next action is a search with filter types FT_{it} consisting of unique combinations of private room, price, and location is expressed in the equation below:

$$\begin{aligned} Pr(FT_{it}|t=n) = & \frac{\exp^{\beta_n x_{it}}}{1+\exp^{\beta_n x_{it}}+\exp^{\beta_f x_{it}}} * \rho(FT|t=n) \\ + & \frac{1}{1+\exp^{\beta_n x_{it}}+\exp^{\beta_f x_{it}}} 1(FT_{it}=FT_{i,t-1}) \end{aligned} \tag{1}$$

The above equation contains two terms. The first relates to cases where the searcher enters a new set of search parameters. The first part of the product is the probability that the next search action is a new search, conditional on the search related observables, x_{it}. β_n are the coefficients of the multinomial browsing model corresponding to a new query and β_f are the coefficients corresponding to finishing the search process. The second part of the product, ρ, is the probability that a set of filters is used conditional on the search number, n. The final term is the probability that a search will use the same filter combination as the previous one because the searcher went to the next page of results.

The set of listings shown to the searcher for a given queries is determined as follows. A potential set of listings is constructed by intersecting the set of listings that fulfills each of the query parameters and filters (if any). These listings are ordered according to a ranking algorithm and the top 18 of these are displayed. If there are fewer than 18 listings, one of two things can happen. With a probability of 35.2% (as observed in the data), a location filter is strict and consequently only those listings that fulfill the location filter are displayed. In the rest of the search results, the location filter is not strict, and listings not fulfilling the location filter but fulfilling the others are concatenated to the results.

I now describe the estimation of these models. The dataset used to estimate the model starts with all searches for stays in Chicago with a search starting on November 26, 2013 and with a check-in date before February 25, 2014. These comprise 335 thousand distinct search actions. The search data I observe includes each query by a user fulfilling the above criteria, the filters applied, the page number of the search results, the dates of the stay, the number of results, the position the map was centered on and the zoom level (when the map was used), the time of the search, the filters loaded, and the listings shown (as well as their ranks on the page). I do not observe whether and when the searcher clicked on a particular listing. Below is a list of additional criteria for queries to be in the sample.

- Queries are conducted by individuals who have an Airbnb ID.
- Queries take place for check-in dates within 60 days of the search.
- Queries occur up to two days before the first contact or booking request the searcher makes for a listing.
- The number of nights of the trip is fewer than 8 and the number of guests is fewer than 5.
- There is more than 1 search and at least one search returns 18 listings. This is meant to eliminate anomalous searches.
- The searcher conducts fewer than 100 total searchers and the depth of search given a query is less than 8 (the 90^{th} percentile). This is meant to eliminate bots scraping the website.
- The number of nights is greater than 0 and the check-in date is after the search date.

These conditions narrow the number of search actions to 126 thousand and the number of searchers to 7711. In this dataset, 83% of search actions involve a new query, 12% involve going to the next page of results for a given query, and 5% result in the end of search. Table 1 displays the results of a multinomial logistic regression predicting the next search choice of searchers. This regression shows that as searchers advance in their search actions, they are less likely to finish search. If searchers are shown fewer than 10 results, they become more likely to finishing searching and this effect becomes larger further into the search process. This reflects that if searchers cannot find listings that fulfill their query parameters, they are likely to leave the site without attempting to book.

Next, I discuss the underlying heterogeneous preferences drawn from filtering frequencies. Turning first to room-type preferences, 66% of searchers filter only for entire properties, 22% filter only for private rooms, and 12% filter for both. With regards to price, Figure 5 displays the distribution of the highest of the maximum prices that each searcher queries. Similarly, there is substantial heterogeneity with regards to which locations are used in filters, as shown in Figure 6. The last part of the above equation concerns the probability that a filter is applied at each state of search. Figure 7 displays the evolution of filtering over the course of a search. The figure shows that at the beginning of a search, one or no filters are likely to be used. As the search continues, more filters are applied so that the most common search type uses location, price, and room type filters.

Conditional on a set of search query, the simulation needs a ranking used to determine which listings are shown and in what order. I do not have access to the exact ranking formula used by Airbnb, but I can approximate it using the data on which listings are shown. One concern with this strategy is that the estimated ranking algorithm could be a poor approximation of the true ranking algorithm. However, the platform did not use a sophisticated machine learning algorithm at the time of my study and relied on few signals, which I include in the regression.

The sample to estimate a ranking algorithm is drawn as follows. I take a 50% sample of searchers to Chicago for the dates of the study and find both the set of listings which they saw on their first search, and the set of potential listings that they could have seen in search conditional on availability and capacity. I then estimate a conditional logistic regression where the outcome is whether a listing was seen in the first set of results by the searcher, and the strata are determined by searcher.

The estimation sample for the ranking function consists of 204333 searcher by listing pairs, of which 13% are seen by searchers. Table 2 displays the results from the estimation procedure. It shows that the ranking algorithm positively weights whether a listing has many five star reviews and is instant bookable. In contrast, the ranking algorithm penalizes non-five star reviews and entire homes. Price is not an important factor in determining the ranking. This is a good placebo test because Airbnb did not use price in the ranking algorithm at the time.

3.2 Choice Model

In this section, I describe how a searcher chooses which listing(s) to contact conditional on a consideration set. I model this choice using a random utility discrete choice model. The searcher's contact decision is a function of the property characteristics, searcher and search characteristics, and filtering choices. Conditional on these observables, the searcher chooses listings to contact. The most important difference between this choice model and standard discrete choice models is that I use the realized filter choices as proxies for otherwise unobservable idiosyncratic preferences for neighborhood, room type, and price. For example, if a searcher uses the map to filter for a particular neighborhood, I allow the searcher's choice probability to differ for listings in that neighborhood.

Denote each guest-trip (a combination of unique searcher, city, and trip dates) as g. Each g receives utility from property, h, according to a linear combination of property characteristics, interactions with idiosyncratic preferences, and a guest specific error term according to the equation below:

$$
\begin{aligned}
u_{gh} = \alpha_0 + p_{gh} * (FP_g' \alpha_1 + NFP_g' \alpha_1 + Z_g' \alpha_2) + f(X_{ht}, Z_g)' \lambda \\
+ \kappa_N + \phi_{FN} FN_{gh} + \phi_{FR} FR_{gh} + \phi_R R_{gh} + \varepsilon_{gh}
\end{aligned}
\tag{2}
$$

where X_{ht} is a vector of property characteristics including review quality, property type and whether the host controls multiple listings. Z_g is a vector of trip and guest characteristics (Nights, Age, Guests, and a constant), p_{ght} is the nightly price of the property for the trip inclusive of platform fees and cleaning costs, FP_g is the maximum price filter used by the searcher (set to 0 if no price filter used), NFP_g is an indicator that takes the value of 1 if a price filter is used, $f(X_{ht}, Z_g)$ is a set of interactions between guest and host characteristics. κ_N is a neighborhood fixed effect, FN_{gh} is an indicator variable for whether a listing's neighborhood was specified by the searcher's filtering action, FR_{gh} is an indicator variable for whether a listing's room type was specified by the searcher's filter, R_{gh} is the lowest search rank a listing was shown at to the guest, and ε_{ght} is an unobserved component of the utility which is distributed according to the type 1 Extreme Value (EV) distribution.

The searcher can also choose the outside option and leave the platform without sending a contact. The searcher's value of the outside option is determined by the following equation:

$$
u_{go} = T_g' \mu + \gamma_{FP} + \phi_H H_g + \varepsilon_{go} \tag{3}
$$

where T_{go} are guest and trip characteristics, γ_{FP} is a set of fixed effects for filters used by the searcher, H_g is the number of unique listings shown by the search engine to the guest[6] and ε_{go} is a type 1 EV error term.

Before moving to discussing the estimation procedure, it is important to clarify the interpretation of this model. The main purpose of this model in this paper is for prediction. The estimated parameters only have a structural interpretation if additional and highly restrictive assumptions are made.

3.3 Estimation of Choice Model

The data use for the choice model estimation consists of the set of users who searched for short-term rentals in Chicago between September 2013 and September 2014. I limit the analysis to a 10% sample of these users because the quantity of data would otherwise be even more difficult to

[6]This term can be interpreted in two ways. First, it controls for unobserved heterogeneity in a searcher's returns from searching. Second, it serves as an analogue to the procedure in Ackerberg and Rysman [2005], which mitigates the tendency of discrete choice models to overstate the benefits of variety.

work with. Otherwise, the data are cleaned in the same way as described in subsection 3.1. This sample is used for estimation because it provides more data for estimating the search model both in terms of observations and variation in observables such as prices.

I group a sequence of search queries into distinct search spells so that searches by the same searcher which differ in destinations and trip date are kept separate. To create a search spell, I first link the searches to a contact (an inquiry or booking) conducted by the searcher. For those searches that can be linked to a contact, I only keep the searches which occurred within two days preceding the contact. Furthermore, I use only the searches related to the first contact by a user in the city during the sample period. For those searches that cannot be linked to a contact, I keep only the searches conducted within the last two days of search activity. This selection criteria ensures that the search results in the data reflect perceived availability of the searcher before an inquiry. The final set of searches contains 236 thousand observations. For the purposes of this estimation, I include the chosen option as well as random sample of up to 20 other options in the estimation procedure. This sampling procedure reduces the computational time of the estimation procedure and retains statistical consistency (see Train [2009] and Wasi and Keane [2012] for details).

The problem of predicting which option a searcher chooses is difficult. The searcher has many options that were typically selected into the consideration set due to having desirable characteristics. Furthermore, there is relatively little information in the data on searcher preferences other than the query parameters. Nonetheless, the predicted probability that a searcher chooses the listing they were observed to have chosen is 10%, conditional on not choosing the outside option. The chosen listing is in the top 5 options ranked by predicted choice probability 60% of the time. This demonstrates that the choice model has some predictive power. In principle, a more sophisticated machine learning model such as a neural network could be used, but it would be difficult to interpret and the simulation already does a good job of fitting the aggregate moments of the data.

The results of the choice model estimation are displayed in Table 3. The estimates are for the most part consistent with prior intuition regarding listing quality. First, with regard to reviews, the average rating and total number of five star reviews are predictive of choice. Second, entire properties and listings with lower search ranks are more likely to be chosen. Third, the outside option is more likely to be chosen when searches have fewer guests, when searchers don't filter for price, and when the search is further away from the check-in date. Interestingly, listings that allow instant booking are less likely to be chosen. This likely reflects the fact that, at least during the sample period, listings that allowed instant book were of a lower quality than those that did not. Furthermore, it suggests that searchers do not strongly respond to the probability of rejection when choosing listings to book. Otherwise, we would expect the coefficient on an instant bookable listing to be positive and large. The filtering behavior of searchers is predictive of choice. Listings which are in a neighborhood that the searcher filtered for and are of the filtered property type are more likely to be chosen. The price filtering behavior is also predictive of choice. The higher a searcher's maximum price filter, the less sensitive they are to the prices of listings and searchers who use a price filter are less likely to pick the outside option.

There are several additional aspects of searcher choice. A searcher can send either an inquiry or a booking request to a host. A booking request commits the searcher to book a listing if the host accepts. Consequently, it precludes sending more than one inquiry. I set the probability of sending a booking request at 31%, equal to the rate at which booking requests are sent in the data.

Next, some searchers send multiple inquiries to hosts simultaneously. To model this, I assume that each searcher draws a potential number of inquiries to send from a Poisson distribution. The searcher then sends the amount of inquiries that is the minimum of the random Poisson draw plus one and the number of listings whose utility is greater than the outside option. The mean of this Poisson distribution is set to 1.199, which is equal to the mean number of contacts minus one divided by the probability of sending at least one contact.

Lastly, after being rejected, guests may search again and send a new set of inquiries, which I call sequential. I set the probability of returning to search again to 64.7%, its empirical probability. Conditional on returning, searchers draw a new potential number of inquiries from a Poisson distribution. I set the mean of this Poisson equal to 3.596, which is the mean number of sequential inquiries conditional on sending at least one divided by the probability of sending at least one contact minus one. The higher mean reflects the fact that conditional on returning after a rejection and sending inquiries, searchers send more inquiries than they did initially. This would be a reasonable response to learning that the probability of rejection is high.

3.4 Host Rejections

In this section, I describe the rejection behavior of hosts. Hosts can reject guests for three reasons: stale vacancies, congestion, and screening.

Stale vacancies occur when searchers contact a host who's listing is unavailable but who has not yet updated the calendar to unavailable status. These rejections are detectable in the data because the data records whether a host later sets a set of dates to be unavailable on the calendar. When those dates correspond to the dates of a rejected contact, then the rejection is classified as due to a 'stale vacancy'. Fradkin [2018] documents that these rejections occur for approximately 15% of US contacts between September 2013 and September 2014. For each unavailable listing included in the simulation, I observe the exact time at which that listing becomes unavailable in the data. Searchers who search before the time at which the listing becomes unavailable and happen to contact that listing are consequently rejected.

Congestion vacancies occur when a searcher contacts a host after another searcher who is not rejected. Because a listing can only host on trip at a time, the second searcher must be rejected. Fradkin [2018] documents that these rejections occur for approximately 8% of US contacts between September 2013 and September 2014. In section 4, I discuss the incidence of rejections for the time period studied in this paper. The time between an inquiry and the confirmation of booking governs the likelihood of congestions. I set this time to be a Poisson random variable with a mean of 1 day.

Lastly, even if a listing is available and not congested, the host may reject a particular searcher. This occurs for 12% of US inquiries. These screening rejections occur because hosts have preferences over when and whom they host. For example, a host might reject a contact because the guest is not reviewed, has a vague inquiry, or does not have enough information in his profile. Hosts also reject guests because the check-in dates of the inquiry can break up a bigger, uninterrupted time of availability for the host, preventing future inquiries. Lastly, hosts may be waiting for a better guest/trip combination or might not be willing to take a particular guest for idiosyncratic reasons.[7]

I model screening rejections by using a logistic regression where the decision to reject is a function of guest, trip, and listing characteristics. The estimating equation for the screening model is:

$$Pr(R_{gh}) = Pr(\alpha_0 + Z_h'\delta + f(X_g, Z_h)'\beta + \eta_{gh} > 0) \tag{4}$$

where η_{gh} is a logit error term, R_{gh} is an indicator for whether the response is a rejection, X_g are the number of guests, guest reviews, guest gender, weekly demand, days in advance of the trip nights, guest age, searcher census tract demographics and month of check-in. Z_h are property type, multi-listing host indicator, host age, the number of reviews and price. $f(X_g, Z_h)$ are interactions between guest and listing characteristics.

I account for the dynamic aspects of the host decision by controlling for the time in advance of the trip of inquiry and for the overall demand for each week of check-in. A more complete model of a host's decision to accept or reject would require the host to have expectations over the flow and quality of potential future searchers in the market. I choose not to add this additional complexity to the already complicated simulation. Further, inquiries with longer lead times are actually less likely to be rejected. This suggests that option value concerns are not the primary driver of rejection behavior.

The model is estimated from a set of 100,617 contacts occurring in Chicago between September 2013 and September 2014 that were not rejected due to congestion or stale vacancies. Table 4 displays the estimates from the above specification. Guests who send an inquiry are more likely to

[7] Edelman et al. [2017] use an audit study to show that some hosts discriminate against non-reviewed guests with African-American names. Cui et al. [2016] use an audit study to show that most of the discimination by race of hosts is eliminated once a guest is reviewed (even if the review is not positive). I do not observe race in my sample and cannot consequently control for it in this regression.

be rejected. This is a function of two factors. First, guests who send 'book it' requests are more committed to booking and are likely to be accepted by hosts. Second, some 'book it' requests are instant bookings, which means that they are guaranteed to be accepted.

Guest reputation affects host decisions. Guests with a prior review are less likely to be rejected.[8] The extent to which reviews are valued by hosts varies across host types. Hosts who have multiple-listings and are less likely to value the social aspect of the Airbnb transaction show no statistically significant differences in rejection behavior between reviewed and non-reviewed guests. Other types of guest information including Airbnb verification of identity and profile descriptions are also associated with lower guest rejection rates.

Trip characteristics also affect the decisions of hosts. Trips with more guests are less likely to be rejected by bigger listings. Furthermore, when the number of guests equals the capacity of the host, the host is more likely to reject. The number of nights is negatively correlated with rejections (conditional on not being rejected due to congestion or a stale vacancy). All else equal, hosts prefer longer trips to shorter trips and longer lead times.

On the listing side, hosts with entire properties are more likely to reject and hosts that have enabled instant booking are less likely to reject. For qualifying guests and trips, instant booking guarantees that a booking request will be accepted. The rejection rate for instant-book enabled listings is not exactly zero for two reasons. First, not all guests are eligible for instant booking. Second, guests may choose to send an inquiry to the host first and the host may respond with a rejection.

4 Simulation Results

In this section, I describe the simulation, its fit to the data, and counterfactual scenarios. I begin by conducting the simulation with the observed market participants and policies for the four weeks in my sample. To elaborate, these simulations take each searcher observed in the data looking for a stay for the focal week, and sequentially run the searchers through the consideration set formation, choice, rejection, and booking processes. The searcher can potentially match with any of the listings visible and available at the time of the search and the simulation keeps track of the visibility and availability of listings.

There are two complications to the above procedure. First, not all bookings observed in the data can be linked to a search. Second, some searches with query parameters for a given week end up contacting hosts regarding a different week. In order to get around this issue, I compute the ratio of observed searchers who can be linked with a contact in any week to the share of observed

[8] I exclude variables related to guest ratings because these are not seen by the host when a guest submits an inquiry and because hosts recommend guests 99% of the time (Fradkin et al. [2018]).

contacters who can be linked to the searcher for the focal week. This ratio, whose average is 1.297 across the four weeks considered, captures the amount of 'missing' searchers in the data for a given focal week. To account for these missing searchers, I resample from the set of searchers so that the total number of searchers is the integer closest to 1.297 times the observed number of searchers. To break ties for the time of search, I perturb the entry time of resampled searches by a random uniform that takes a value between 0 and .1 days. The final set of calibrated parameters is displayed in Table 5.

4.1 Simulation Fit

Table 6 displays empirical outcomes as well as the mean and standard deviation of outcomes for 100 simulation runs in each of the four weeks considered. Turning first to searchers and bookings, the numbers of searchers ranges from 551 to 1169 across the four weeks and the number of bookings ranges from 105 to 236. The mean of simulated bookings ranges between 112 and 226. For all weeks, the actual booking rate is within a standard deviation of the mean booking simulated booking rate. The search to contact rates in this sample range from 33% to 42% and the contact to book rates range from 57% to 51%. For each of these metrics, the observed moment in the data is within a standard deviation of the mean simulated outcome. I take this as evidence that the simulation fit is good. It is noteworthy that no part of the above model was explicitly targeted to match the above moments.

For other outcomes such as rejection rates, and sequential contacts, most of the observed moment and week combinations are close to the simulated means. There are weeks when the outcome is outside of a standard deviation in the simulation but this is unsurprising given the variance across simulations within a given week and given the fact that the simulation did not directly target these moments. The rejection rates in this sample are lower than the rejection rates reported in Fradkin [2018] due to the fact that this sample is from a winter period where travel demand is relatively low. The lower rate of searchers per host should result in less congestion, more options to choose from for guests, and fewer incentives for hosts to be selective. All of these factors would result in fewer overall rejections and higher conversion rates.

4.2 Simulations of Alternative Policies

In the section, I use the simulations to explore the mechanisms that affect matching outcomes. Table 7 displays the results for 48 simulations of week 4.[9] Column (2) shows the mean outcomes when searchers are no longer shown the listings that would reject their inquiries in search. Note, this policy would require an impossibly omniscient platform designer and therefore represents the limit of what the platform could achieve in reducing rejections. In this counterfactual, the number of bookings increases by 15% while the number of contacts decreases by 12%. Column (3) demonstrates the simulation outcomes if all of congestion was removed. In this case, the gains are small with just a 1% increase in the booking rate.

Column 4 considers the market level effects of a better ranking algorithm. This ranking algorithm will be used later in the paper to document bias from experimental treatment effects. I construct a ranking using the formula below and simulate outcomes when this algorithm is used instead of the one that was actually used:

$$w_{gh,t} = \bar{\mu}_{gh} * (1 - \widehat{Pr(R_{gh})}) \tag{5}$$

This algorithm is based on the product of two components: the expected utility for the searcher from a listing ($\bar{\mu}_{gh}$) and the predicted acceptance probability of a listing for that searcher (1 - $\widehat{Pr(R_{gh})}$). Listings which are more appealing to searchers are ranked higher while listings with a higher probability of rejecting a searcher are ranked lower. Column (4) displays the results for this counterfactual. The booking rate is 34% higher than the baseline while the contacter rate is 33% higher. Consequently, this ranking algorithm has a large positive effect on matching outcomes.

5 Returns to Scale and User Acquisition

In this section, I explore the returns to scale in matching and the returns to user acquisition on the platform. The most common way to estimate the returns to scale in matching is to use the Cobb-Douglass matching function, displayed below. This function makes the assumption that the total number of matches created in a matching market is determined solely by the total number of guests (buyers) and hosts (sellers).

$$log(M_{it}) = log(A_{it}) + \alpha_M log(G_{it}) + \beta_M log(H_{it}) \tag{6}$$

The simple form of the above equation makes it easy to fit. However, it's interpretation is complicated by the fact that the composition of guests and hosts varies over time. Therefore, observed variations in guests, hosts, and bookings can be driven not by the shape of the matching function but by differences between marginal market entrants and average market entrants. My simulation allows me to vary the number and types of guests and hosts in the market, and to evaluate the effects on matching rates.

To investigate returns to scale, I simulate market outcomes while varying the number of par-

[9]Note that the results for the baseline specification differ from the results above due to the lower number of draws used for the simulations. The reduced number of simulation draws in the evaluation of counterfactuals was done to reduce simulation time. In order to be comparable, the same random seeds need to be used across counterfactuals.

ticipating guests and hosts. For example, to simulate a market where there are 1.5 times as many guests, I randomly resample searchers from the data until the total number of searchers is 1.5 times the number of observed searchers. To simulate a market with .75 times as many guests, I randomly remove guests from the simulation. A similar procedure is used for hosts. This procedure insures that there are no systematic changes in the composition of market participants.

I simulate outcomes for all combinations of guest and host amounts in the set $\{.25 : .25 : 2\}$ relative to the observed data. Each combination is simulated 48 times. Table 8 displays the estimates of Cobb-Douglass matching functions, where each observation is a simulation run. Column (1) displays the outcome for unique searchers who send at least one contact to a host. Most of the weight in the matching function is placed on guests and the sum of $\alpha_M + \beta_M$ is 1.036, implying increasing returns to contacting. Column (2) shows the result for total contacts sent. The major difference between (1) and (2) is that the weight on hosts increases because having more hosts also encourages searchers to send more contacts. Lastly, Column (3) displays the results for bookings. Guests remain more important than hosts for bookings and returns to scale remain increasing (1.029).

The matching of guests to host is also affected by marketplace design decisions such as ranking algorithms. To see how this affects returns to scale, I redo the exercise above but simulate the marketplace with the better ranking algorithm. Table 9 displays the results from these estimates. Across all three outcomes, returns to scale increase relative to the baseline. Furthermore, the relative importance of guests to matching decreases while the importance hosts increases. Intuitively, the better ranking algorithm shows guests better matches and consequently increases the number of matches for a given set of guests and hosts. Furthermore, the ranking algorithm surfaces better hosts, which are then booked and become more scarce relative to the baseline.

Next, I investigate the possibility that returns to scale vary with the size of the market. When the market is small, there may be certain types of guests and listings which are not present in the market. The absence of these types on one side of the market, may reduce conversion on the other side of the market. As the market size grows, there is a lower probability that particular types are missing, and consequently a larger share of users can find a good match. Furthermore, because search is limited, after some point, the number of hosts in the market becomes less important since searchers will not see the majority of them. Lastly, the probability of congestion may be higher as market size grows.

Table 10 displays estimates of the returns to scale for subsamples where the amount of guests and hosts is either smaller or larger than the amount observed in the data. Columns (1) - (3) display the estimates from the smaller market. Relative to the estimates in Table 8, the overall returns to scale are larger in this sample. In contrast, the returns to scale in the larger markets shown in columns (4) - (6) are much smaller. In fact, returns to scale become decreasing for bookings

(.974) and the relative importance of hosts decreases. These results suggest that there are increasing returns to scale in matching but that they eventually fade out unless there is complimentary marketplace policy.

5.1 User Acquisition

The above results suggest that additional guests are more important for generating bookings than additional hosts when the ratio of guests to hosts is approximately equal. For the time-period in my sample, there are fewer guests than hosts, and this suggests that adding additional guests is more important than adding additional hosts.[10] However, the above regression is based on data in which marginal guests and hosts are drawn from the same distribution as average guests and hosts. It may be the case that marginal users may be systematically different from existing users.

For example, searchers acquired using advertising may be less familiar with Airbnb and may convert at worse rates. Suppose that the platform had many similar markets and randomized ad purchases across those markets. The platform could then evaluate the market level effects of those ad purchases and calculate an ROI. In practice, the platform does not have many similar markets and it's hard to standardize ad purchases across markets. For example, the users acquired by advertising in one market may be different from the users acquired by advertising in a different market. What the platform can see is whether a user came to the platform through an advertising campaign and that user's characteristics. If the platform had a measure of advertising cannibalization, it could then use a simulations like the one in this paper in order to measure the market level effects of additional users acquired through advertising.

My data does not contain information on the user's acquisition channel, but it does contain observable characteristics that can be used to segment users. I create an index of predicted guest and host matching utility as a function of these characteristics. For guests, I calculate the average of utilities they receive across the listings in the market. For hosts, I calculate the mean utility across guests of that host. I then simulate market outcomes if an additional 25% of users, drawn without replacement, were taken either from the top or bottom quartile of the index.

Table 11 displays the results from these simulations. Columns (2) and (3) display the results for the bottom and top 25% of listings according to the above index. Adding lower quality listings actually reduces the number of bookings and contacts. This results form the fact that worse listings are shown in search and searchers are consequently less likely to send contacts and send fewer contacts conditional on sending one. The rate of rejections also falls slightly. When adding the higher quality listings, the booking and contact rates increase by 5% and 7% respectively.

[10] A proper calculation of the benefit of additional listings must also account for the fact that listings can stay in the market for a long time and improve in their quality as they accumulate experience and reviews.

Columns (4) and (5) display the results from adding searchers from the bottom and top quartiles of the index distribution. The overall increase in the number of bookings from adding searchers in the bottom quartile is 17%, larger than from having additional listings. This is consistent with the results from the matching function, which show that searchers are relatively more important in creating matches than listings. Furthermore, there is little difference in outcomes between column (4) and (5), suggesting that the index of utility does not have much predictive power for the booking rates of individual searchers.

6 Experimentation and Spillovers

In this section, I consider the estimation biases that occur when evaluating ranking experiments with spillovers. These spillovers can occur when treated searchers book listings that would have otherwise been booked by other searchers or when the treatment affects the probability of congestion.

I use the terminology of Hudgens and Halloran [2008] to frame this discussion. Consider a policy ($\psi_i(x)$) which treats x% of searchers (g) in a market with a new ranking algorithm in a market, $i \in 1, \ldots, I$. There are several causal effects that could be of interest under this policy. First, there is the group *direct* causal effect, which is the average difference between treated and control individuals under $\psi_i(x)$. This corresponds to the typically studied treatment effect in experiments without interference.

Second, there is the group *indirect* causal effect of the policy, which is the average effect on individuals of the treatments received by other individuals. This can be measured by the average difference between outcomes for the control group under policy $\psi_i(x)$ with the average outcomes for individuals under a baseline policy, $\psi_i(0)$ in the same market. The indirect effect is a measure of interference in the control group.

Lastly, there is the group *overall* causal effect, which is the difference between average outcomes when a market is treated with $\psi_i(x)$ and when it is treated with $\psi_i(0)$. This measures the effect of the policy on the market and is a function of both the direct and indirect effects. From the firm's perspective, the decision of a policy to use should be based on this overall effect, as it measures the effect of $\psi_i(x)$ on revenues and other aggregate outcomes. Outside of the simulation, it is impossible to observe the same market in multiple treatment states. Consequently, these effects must be estimated by averaging across markets. Hudgens and Halloran [2008] call these population level effects.

I characterize each of these causal effects where the policy is the previously studied ranking algorithm assigned to 25%, 50%, 75%, or 100% of users. In these simulations, each simulation run can be thought of as one market. There are 48 simulation runs for each policy treatment.

Table 12a displays the direct effects of each treatment on conversion and rejection rates.[11] Turning first to the case when 25% of users are treated, there are is 9.2 percentage point increase in the conversion rate from searching to booking. There is no effect on congestion rejections. As the share of treated units increases, the treatment effects decrease on conversion rates decrease in magnitude. For example, the effect on search to book rates falls to 6.9 percentage points. The rate of congested inquiries increases from .003 when 25% are treated to .042 when 75% are treated. This increase in congestion suggests that the new algorithm is causing a high correlation between the choices of treated searchers.

Table 12b displays the indirect effects of these treatments. Relative to the direct effects, the magnitudes in this table are small, suggesting that there is little spillover from the treatment group to the control group. The most notable effect is on congestion when 75% of searchers are treated. In this case, the congestion rejection rate for the control group falls by 1 percentage point. There are fewer individuals seeing the control ranking algorithm and consequently control searchers are less likely to send congested inquiries. The small indirect effects also suggest that the decreasing direct treatment effects are caused by spillovers within the treatment group.

Table 12c displays the overall effect of the policies on market outcomes. Across all conversion rates, the overall effect is smaller than the direct effect. For cases when x, the share treated, is less than 100%, this occurs in part because the control group is not receiving the beneficial treatment. Nonetheless, even when everyone is treated, the overall effect on search to book (.06) is smaller than the direct effect on search to book from a 25% experiment (.092). This shows that a naive experimental analysis would overstate the true effect of the algorithm by 53%. Policies that assign a larger share of searchers to be treated result in direct effects closer to the overall effect. This would not necessarily be true if there were large indirect effects.

The treatment effect tables can also be used to diagnose mechanisms by which interference occurs. One hypothesis is that treated searchers who booked options diminished the supply of listings for control searchers, reducing their booking rates. To check whether this is true, we can look at the indirect effects table, which shows tiny effects on search to contact rates across treatment shares. This implies that such an effect is not happening. In contrast, the direct effects table shows that as the treatment share increases, the effect difference in search to contact rates decreases. This demonstrates that there is interference due to competition for listings *within* the treated group. Second, the treatment may create spillovers by changing the level of congestion if multiple treated searchers try to book the same listings. Indeed, we observe that the direct effect on contact to book falls with the share of individuals treated. At the same time, the direct effect on congested rejections increases with the share treated. This implies that the treatment creates

[11] Note that direct and indirect effects are undefined when the treatment is assigned to 100% of users because there is no control group.

congestion among the treated searchers and that this reduces the benefit of the improved ranking algorithm.

7 Discussion

This paper has presented an argument for the value and practically of simulations as a complement to experiments in marketplace design. Marketplace simulations can quantify returns to scale, the returns to acquiring users, and the size and mechanisms of spillovers between users. In turn, these simulated experiments can be used to guide the decisions of marketplace designers regarding user acquisition and experimentation strategies.

My simulation of Airbnb was composed of three main components: a model of how searchers navigate the site, a model of how searchers choose among listings, and a model of how hosts choose whether to accept or reject guest contacts. I applied my model to a month long time period in Chicago in 2014. The simulations demonstrate that returns to scale in matching are increasing when the market is small but then become decreasing as the market grows. The shape and scale of the matching function is also sensitive to the ranking policy used by the platform. This result suggests that returns to scale in matching are not enough, on their own, to generate a winner take all effect in this digital market. However, technological or market design advances can be used to create increasing returns to scale.

Next, I show that the effects of additional listings on the number of matches vary greatly based on observables. This simulation approach can inform a marketplace's strategy regarding the types of users that should receive marketing. The static effect of a listing is however insufficient to make this determination because listing type may change with reputation and experience. Market designers should also take into account the trajectory of listings over time when making this calculation.

Lastly, I show that there are spillovers between users which bias experimental treatment effects. The treatment effect of a better ranking algorithm on conversion rates is overstated by 53% when a quarter of users are randomized into treatment. This overstatement occurs because treated users compete for similar listings, which causes a dwindling in supply and an increase in congestion. Market designers can use this style of simulation to adjust experimental treatment effects from simple experiment to reflect their market-level effects. They can also use these simulations to prototype ranking algorithms before running complicated experimental designs.

This paper has two main limitations. First, I was not able to obtain data to validate the simulation results with an experiment. It would be comforting if the direct effects of a treatment in simulated and realized settings were similar. Conducting such an exercise should be simple for individuals working at digital marketplaces. An iterative process of simulation development and experiments has the potential to converge to a model that captures the key properties of matching in a digital marketplace. Second, as participants learn about the effects of policy changes they may adjust choices regarding prices or other behavioral variables. These types of responses, also not captured in many user level experiments, may substantially change the evaluation of a policy. I hope that this paper encourages the development of new methods to understand these longer-run adjustments.

References

Daniel A. Ackerberg and Marc Rysman. Unobserved Product Differentiation in Discrete-Choice Models: Estimating Price Elasticities and Welfare Effects. *RAND Journal of Economics*, 36(4): 771–788, 2005.

James Albrecht, Bruno Decreuse, and Susan Vroman. Directed search with phantom vacancies. 2017.

Nick Arnosti, Ramesh Johari, and Yash Kanoria. Managing Congestion in Decentralized Matching Markets. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 451–451, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2565-3. doi: 10.1145/2600057.2602893.

Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Clearing Matching Markets Efficiently: Informative Signals and Match Recommendations. SSRN Scholarly Paper ID 2965135, Social Science Research Network, Rochester, NY, March 2018.

Eytan Bakshy, Dean Eckles, and Michael S. Bernstein. Designing and Deploying Online Field Experiments. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 283–292, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2567967.

Thomas Blake and Dominic Coey. Why Marketplace Experimentation is Harder Than It Seems: The Role of Test-control Interference. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 567–582, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2565-3. doi: 10.1145/2600057.2602837.

Bart Bronnenberg, Jun B. Kim, and Carl Mela. Zooming in on choice. *Marketing Science*, 2016.

Yuxin Chen and Song Yao. Sequential Search with Refinement: Model and Application with Click-Stream Data. *Management Science*, 2017.

Arnaud Cheron and Bruno Decreuse. Matching with Phantoms*. *The Review of Economic Studies*, 2017. doi: 10.1093/restud/rdw032.

Ruomeng Cui, Jun Li, and Dennis Zhang. Reducing Discrimination with Reviews in the Sharing Economy: Evidence from Field Experiments on Airbnb. SSRN Scholarly Paper ID 2882982, Social Science Research Network, Rochester, NY, December 2016.

Zoë Cullen and Chiara Farronato. Outsourcing tasks online: Matching supply and demand on peer-to-peer internet platforms. *Manuscript*, 2018.

Michael Dinerstein, Liran Einav, Jonathan Levin, and Neel Sundaresan. Consumer Price Search and Platform Design in Internet Commerce. *American Economic Review*, 108(7):1820 – 1859, July 2018.

Benjamin G. Edelman, Michael Luca, and Dan Svirsky. Racial Discrimination in the Sharing Economy: Evidence from a Field Experiment. *American Economic Journal: Applied Economics*, 9 (2):1–22, 2017.

Chiara Farronato and Andrey Fradkin. The Welfare Effects of Peer Entry in the Accommodations Market: The Case of Airbnb. *NBER Working Paper*, 2018.

Chiara Farronato, Alan Maccormack, and Sarah Mehta. Innovation at Uber: The Launch of Express POOL, 2018.

Andrey Fradkin. Search, Matching, and the Role of Digital Marketplace Design in Enabling Trade: Evidence from Airbnb. SSRN Scholarly Paper, Social Science Research Network, Rochester, NY, 2018.

Andrey Fradkin, Elena Grewal, and David Holtz. The Determinants of Online Review Informativeness: Evidence from Field Experiments on Airbnb. 2018.

Anindya Ghose, Panagiotis G. Ipeirotis, and Beibei Li. Examining the Impact of Ranking on Consumer Behavior and Search Engine Revenue. *Management Science*, 60(7):1632–1654, February 2014. ISSN 0025-1909. doi: 10.1287/mnsc.2013.1828.

Gunter J. Hitsch, Ali Hortaçsu, and Dan Ariely. Matching and Sorting in Online Dating. *American Economic Review*, 100(1):130–163, March 2010. ISSN 0002-8282. doi: 10.1257/aer.100.1.130.

David Michael Holtz. *Limiting Bias from Test-Control Interference in Online Marketplace Experiments*. Thesis, Massachusetts Institute of Technology, 2018.

John J. Horton. Misdirected Search Effort in a Matching Market: Causes, Consequences and a Partial Solution. *Working Paper*, 2018.

Michael G Hudgens and M. Elizabeth Halloran. Toward Causal Inference With Interference. *Journal of the American Statistical Association*, 103(482):832–842, June 2008. ISSN 0162-1459. doi: 10.1198/016214508000000292.

Yash Kanoria and Daniela Saban. Facilitating the Search for Partners on Matching Platforms: Restricting Agents' Actions. SSRN Scholarly Paper ID 3004814, Social Science Research Network, Rochester, NY, July 2017.

Gregory Lewis and Albert Wang. Who benefits from improved search in platform markets? *Working Paper*, 2013.

Babur De los Santos and Sergei Koulayev. Optimizing Click-through in Online Rankings for Partially Anonymous Consumers. *Marketing Science*, Forthcoming.

Barbara Petrongolo and Christopher A Pissarides. Looking into the Black Box: A Survey of the Matching Function. *Journal of Economic Literature*, 39(2):390–431, June 2001. ISSN 0022-0515. doi: 10.1257/jel.39.2.390.

Jean-Charles Rochet and Jean Tirole. Platform Competition in Two-Sided Markets. *Journal of the European Economic Association*, 1(4):990–1029, June 2003. ISSN 1542-4774. doi: 10.1162/154247603322493212.

Alvin E. Roth and Xiaolin Xing. Turnaround Time and Bottlenecks in Market Clearing: Decentralized Matching in the Market for Clinical Psychologists. *Journal of Political Economy*, 105 (2):284–329, 1997.

Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M. Airoldi. Detecting Network Effects: Randomizing Over Randomized Experiments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1027–1035, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098192.

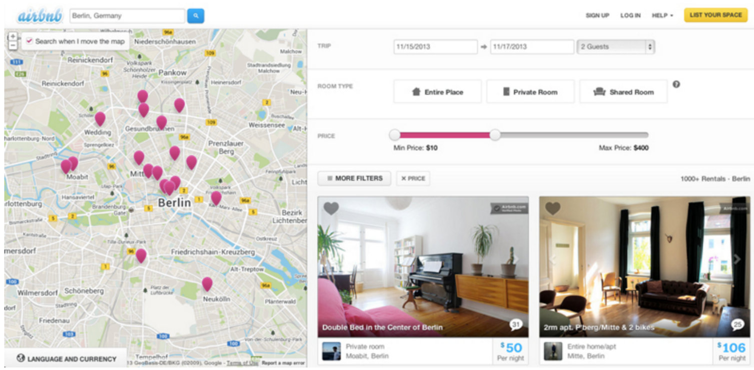
Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge; New York, June 2009. ISBN 978-0-521-74738-7.

Raluca Mihaela Ursu. The power of rankings: Quantifying the effects of rankings on online consumer search and choice. *Marketing Science*, Forthcoming, 2018.

Nada Wasi and Michael P. Keane. Estimation of Discrete Choice Models with Many Alternatives Using Random Subsets of the Full Choice Set: With an Application to Demand for Frozen Pizza. Economics Paper 2012-W13, Economics Group, Nuffield College, University of Oxford, 2012.

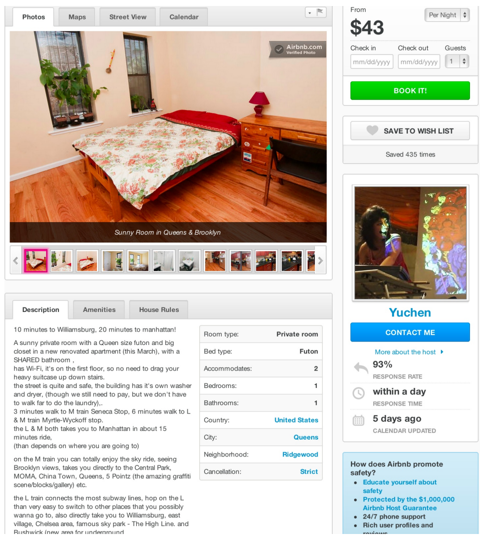
Figures

Figure 1: Search View



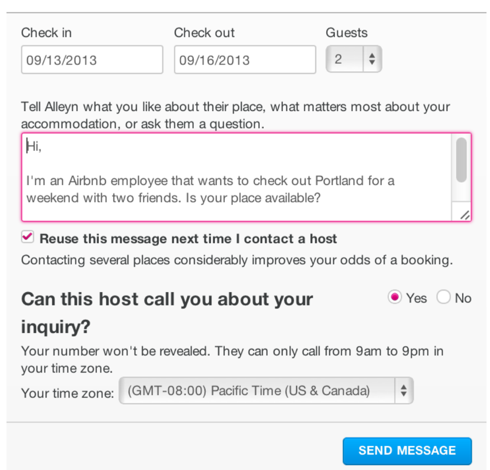
This figure displays the results of a search in Berlin for November 15, 2013 to November 17, 2013. Selecting filters or moving the map changes the set of displayed results. The searcher can scroll the page to see 21 listings before she is prompted to go to the next page of results. The results are displayed according to a ranking algorithm which is common across all searchers who input those search parameters.

Figure 2: Listing View



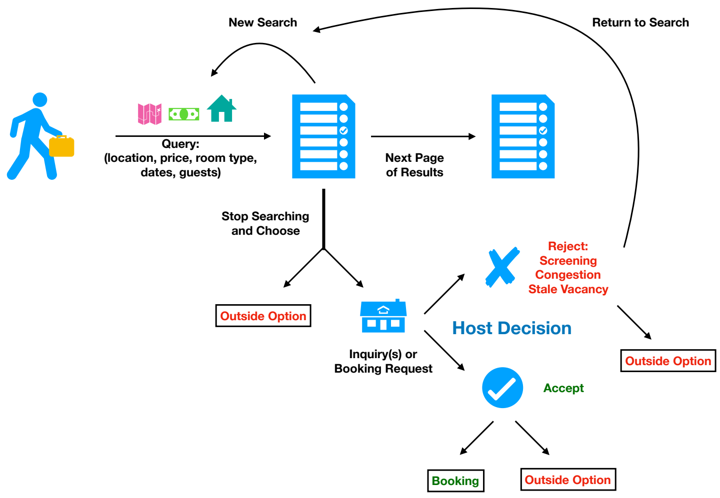
In 2014, a searcher who clicked on a listing in the search results saw the following view. The ratings and text of reviews for the listing are shown lower on the page.

Figure 3: Inquiry Submission Form



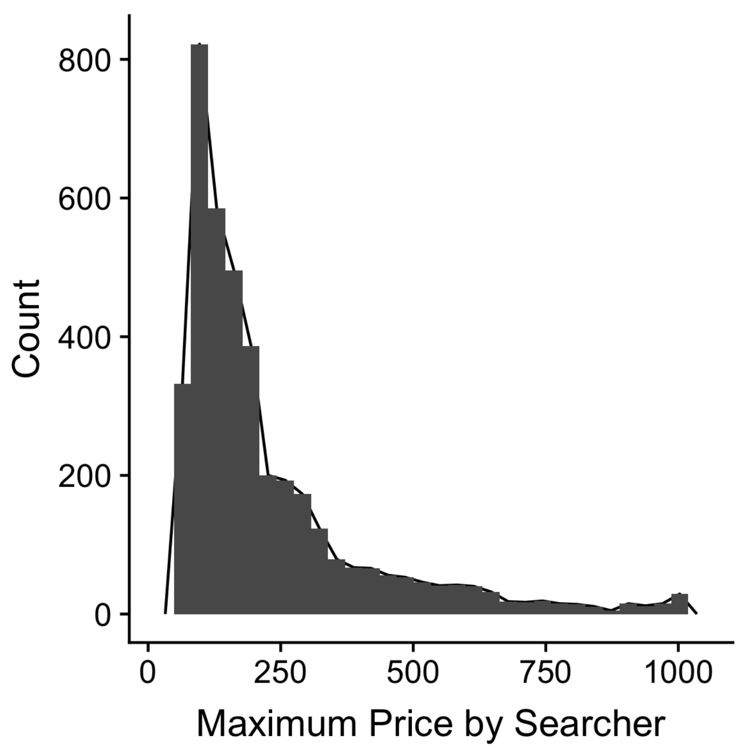
The figure above displays the prompt that searchers saw in 2014 when they clicked the “Contact Me” button.

Figure 4: Model Diagram



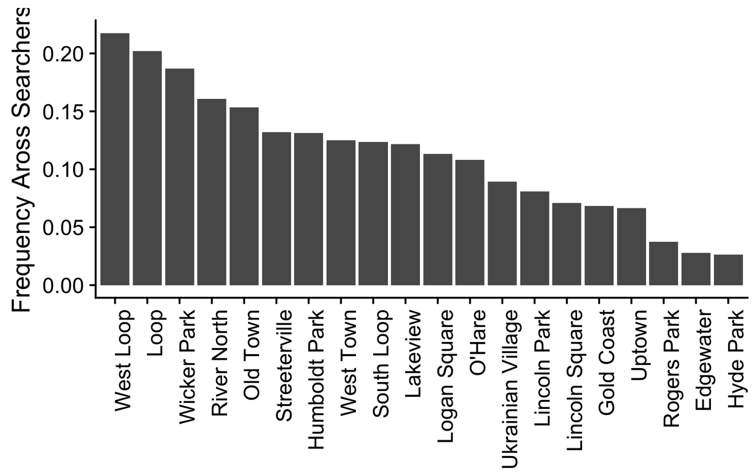
This figure displays the set of actions and outcomes being simulated for each searcher.

Figure 5: Maximum Price Filtering Frequencies



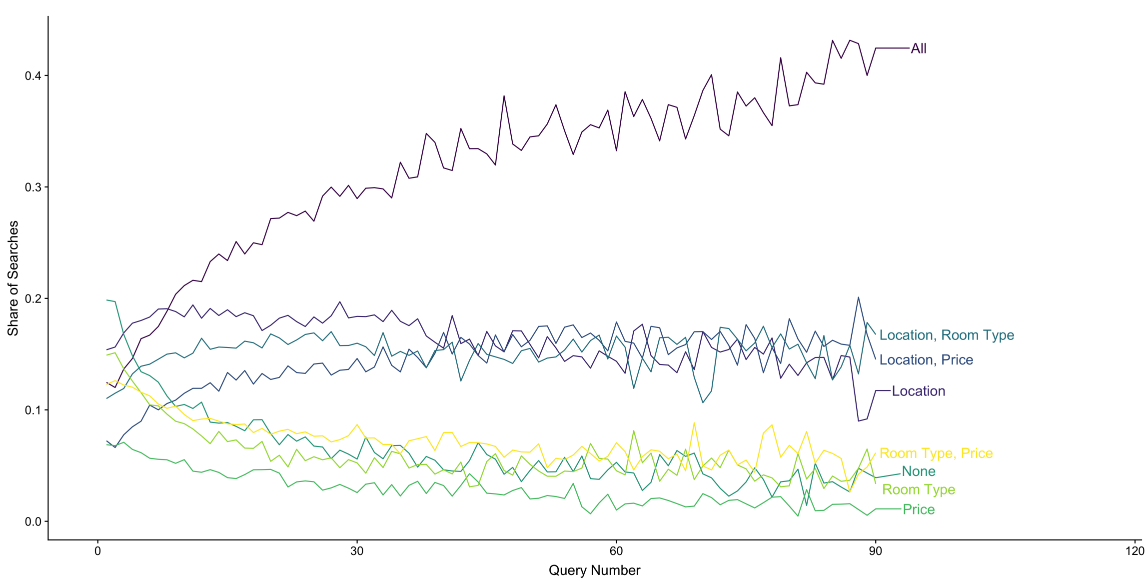
This figure displays a histogram of the maximum price filters applied by each searcher in the sample.

Figure 6: Neighborhood Filtering Frequencies



This figure displays the frequency, at a searcher level, that each neighborhood is either filtered for or selected using the search map.

Figure 7: Filter Types as a Function of Query Number



This figure displays the frequency of each filtering combination as a function of the query order.

8 Tables

Table 1: Browsing Model

	Dependent variable:	
	New Search	Finish Search
	(1)	(2)
Constant	2.405^{***}	1.079^{***}
	(0.035)	(0.049)
log(Search Number)	0.010	-0.835^{***}
	(0.012)	(0.019)
$<$ 10 Results	1.727^{***}	1.601^{***}
	(0.122)	(0.135)
Price Filter	-0.097^{***}	0.168^{***}
	(0.023)	(0.033)
Room Type Filter	-0.551^{***}	-0.182^{***}
	(0.024)	(0.034)
Neighborhood Filter	0.938^{***}	0.860^{***}
	(0.024)	(0.035)
Not First Page	-3.240^{***}	-1.480^{***}
	(0.025)	(0.037)
$<$ 10 Results * log(Search Number)	-0.080^{**}	0.011
	(0.041)	(0.049)
Note:	*p<0.1; **p<0.05; ***p<0.01	

This figure displays the coefficients in a multinomial logistic regression of user actions while browsing. There are three actions, going to a new search, finish the search and making the choice to send a contact, and going to the next page of results (e.g. page 3). Search number refers to the order of the query. '$<$ 10 Results' is an indicator that takes the value of 1 when there were fewer than 10 listings returned by the search. The filter variables refer to whether the filter were used in the prior search. 'Not First Page' is an indicator for whether the prior search page was not the first page of results for a given query.

Table 2: Ranking Model

	Dependent variable:
	Listing Shown
Reviews	-0.008^{***}
	(0.001)
Five Star Reviews	0.046^{***}
	(0.002)
Capacity	-0.129^{***}
	(0.005)
Instant Bookable	1.130^{***}
	(0.017)
Num. Photos	0.017^{***}
	(0.001)
Private Room	0.494^{***}
	(0.020)
New Listing	-1.457^{***}
	(0.069)
Days Active	-0.002^{***}
	(0.00005)
Log(Price)	-0.0003
	(0.020)
Neighborhood FE	Yes
Observations	204,333
Note:	*p<0.1; **p<0.05; ***p<0.01

This table displays the coefficients of a logisic regression for whether a given listing was shown in the first search conducted by a searcher. It is meant to be a proxy for the ranking algorithm used by the platform. 'Reviews' and 'Five Star Reviews' refers to the number of reviews and five star reviews respectively. 'Capacity' is the maximum number of guests a listing can host. 'Instant Bookable' is an indicator for whether a listing can be instant booked.

Table 3: Demand Model

Variable	Estimate	Std. Error
Price	-0.006	0.0004
Avg. Rating	0.044	0.034
Total Reviews	-0.014	0.004
Total Five Star Reviews	0.023	0.006
Has Prof. Photo	-0.031	0.036
Weird Property Type	-0.493	0.202
Is Instant Bookable	-0.403	0.050
No Reviews	0.422	0.162
In Neigbhorhood Filter	0.957	0.047
Entire Prop.	0.354	0.055
In Room Type Filter	0.711	0.057
Minimum Search Rank	-0.106	0.004
OO x Guests	-0.184	0.033
OO x Num. Listings Seen	-0.001	0.001
OO x Lead Time	0.007	0.002
OO x Has Max Price	-0.713	0.099
Price x Max Price Filter	0.00001	0.00000
Price x Has Max Price	-0.008	0.001

The above table displays results from a conditional logistic regression of choice probabilities among searchers. “Weird Property Type” refers to properties that are not apartments, houses, or condos. “Is Instant Bookable” refers to whether a listing is open to being instant booked by at least some sample of users. “In Neighborhood Filter” equals one when the listing’s neighborhood is filtered for by the searcher in a least one search. “In Room Type Filter” equals one when the listing’s room type (private or entire home) is filtered for by the searcher in a least one search. “Entire Prop.” equals one when the listing is for the rental of an entire property rather than a room within a property. “OO” equals one when the option is the outside option. “Max Price Filter” is the maximum price filtered for by the searcher across all searches. “Minimum Search Rank” is the minimum rank at which the listing appeared in a searchers results. “Num. Listings Seen” is the number of unique listings loaded by Airbnb for the search pages browsed by the searcher. “Lead Time” is the time (in days) between the search and the check-in date. “Hotel Price” is the average hotel price in Chicago for the days of the stay.

Table 4: Screening Model

	Dependent variable:
	Rejected
Inquiry First	0.839***
	(0.018)
Guest Reviewed	−0.078***
	(0.024)
Guest Has About Description	−0.040**
	(0.019)
Guest Verified	−0.134***
	(0.019)
Guest Has Profile Photo	−0.024
	(0.019)
New Guest	−0.024
	(0.020)
Num. Guests	0.017
	(0.018)
Num. Nights	−0.141***
	(0.006)
Entire Property	0.420***
	(0.052)
Multi-listing Host	−0.068
	(0.044)
Instant Book - Experienced	−0.565***
	(0.047)
Instant Book - All	−0.836***
	(0.061)
Instant Book - Social	0.131
	(0.355)
Full Guest Capacity	0.110***
	(0.022)
Reviewed Guest * Multi-Listing Host	0.105**
	(0.041)
Num Guests * Entire Property	−0.041**
	(0.019)
Check-in Month and Lead Time Fixed Effects	Yes
Demographic Controls	Yes
Listing RE	Yes
Observations	100,617

This table displays results from a logistic regression predicting whether a contact was rejected. "Inquiry First" is an indicator for whether an inquiry was sent rather than a booking request. "Foreign Guest" refers to guests outside of the United States, "Guest Reviewed" refers to whether the guest had at least one review prior to inquiry, "Guest Has About Description" refers to whether the guest had a profile description, "Guest Verified" refers to whether the guest's identity was verified by Airbnb, "New Guest" refers to guests who signed up within 31 days of the inquiry. "Multi-listing Host" refers to a host who has more than 2 active listings. "Instant Book" refers to hosts who allow guests to book without the possibility of rejection. "Experienced" requires that guests have had a prior stay, "All" is open to all potential guests, and "Social" is open only to guests with a social connection to the host. "Full Guest Capacity" refers to inquiries in which the number of guests equals the capacity of the listing.

Table 5: Calibrated Parameters

Parameter	Value
Probability of Return	0.647
Share Limited Geo	0.352
Max Results Per Page	18
Share Inquiry	0.690
Share Missing Searchers	1.297
Simultaneous Search Intensity	1.199
Sequential Search Intensity	3.596
Probability of Booking After Acceptance	0.680
Mean of Poisson Time to Book Distribution	1

This table displays the calibrated parameters in the simulation. 'Probability of Return' is the probability that a searcher whose initial contacts were rejected returns to search again. 'Share Limited Geo' is the share of searches with a geographic filter that is strict si that listings in adjacent locations are not shown. 'Share Inquiry' is the share of contacts that were submitted using the 'Contact Me' button and were consequently non-binding. 'Share Missing Searchers' is the share of searchers that can make bookings but are not observed in my data. 'Simultaneous' and 'Sequential' search intensities refer to parameters governing the amount of contacts sent by searchers.

Table 6: Simulation Fit

	Week 1			Week 2			Week 3			Week 4		
	Data	*Mean in Sim.*	*SD in Sim.*	*Data*	*Mean in Sim.*	*SD in Sim.*	*Data*	*Mean in Sim.*	*SD in Sim.*	*Data*	*Mean in Sim.*	*SD in Sim.*
Bookings	105.00	111.81	17.06	105.00	119.13	16.92	146.00	133.60	21.17	236.00	226.20	32.31
Contacts	381.00	381.71	74.05	328.00	390.08	66.15	430.00	463.85	97.86	765.00	781.52	165.47
Contacters	182.00	192.34	25.39	185.00	198.70	23.21	249.00	228.11	29.24	389.00	391.41	52.97
Sequential Contacts	93.00	105.69	31.42	58.00	104.06	25.48	108.00	128.79	41.62	234.00	228.30	67.25
Share Rejected	0.24	0.31	0.03	0.28	0.27	0.03	0.24	0.31	0.02	0.30	0.32	0.02
Share Rejected Screening	0.17	0.20	0.02	0.17	0.20	0.02	0.17	0.19	0.02	0.16	0.20	0.02
Share Rejected Congestion	0.02	0.04	0.02	0.06	0.04	0.01	0.03	0.05	0.02	0.05	0.07	0.02
Contact / Book	0.58	0.58	0.04	0.57	0.60	0.04	0.59	0.58	0.03	0.61	0.58	0.02
Search / Contact	0.33	0.35	0.05	0.33	0.36	0.04	0.42	0.39	0.05	0.33	0.33	0.05
Guests	551.00	551.00	0.00	553.00	553.00	0.00	592.00	592.00	0.00	1169.00	1169.00	0.00
Listings	1808.00	1808.00	0.00	1976.00	1976.00	0.00	1953.00	1953.00	0.00	1958.00	1958.00	0.00

This table displays the actual and simulated average outcomes for the four weeks in my sample. ‘Contacters’ refers to the number of searchers who sent at least one contact. The rejected shares are computed as a fraction of contacts.

Table 7: Counterfactual Policies

	Baseline (1)	No Rejection (2)	No Congestion (3)	Match Based Ranking (4)
Booked	203.62	233.38	205.62	273.46
Contacts	677.48	597.38	668.29	1,017.44
Contacters	351.56	344.19	349.98	468.90
Sequential Contacts	191.71	121.23	181.35	334.06
Rejections	203.77	0	185.92	322.38
Share Rejected	0.30	0	0.28	0.32
Share Stale Rej.	0.05	0	0.05	0.06
Share Screen Rej.	0.22	0	0.23	0.18
Num Guests	1,169	1,169	1,169	1,169
Num Hosts	1,958	1,958	1,958	1,958

This table displays simulated outcomes for the baseline scenario and three counterfactuals. 'No rejection' makes it so that searchers cannot be shown any listing who would reject them. 'No congestion' makes it so that searchers cannot be shown congested listings. 'Match Based Ranking' refers to the outcomes with the ranking algorithm proposed in the paper.

Table 8: Returns to Scale

	Dependent variable:		
	Log(Contacters)	Log(Contacts)	Log(Bookings)
	(1)	(2)	(3)
Log(Guests)	0.966^{***}	0.969^{***}	0.937^{***}
	(0.002)	(0.002)	(0.002)
log(Hosts)	0.070^{***}	0.100^{***}	0.092^{***}
	(0.002)	(0.002)	(0.002)
Constant	-1.512^{***}	-1.105^{***}	-2.024^{***}
	(0.016)	(0.022)	(0.022)
Observations	3,072	3,072	3,072
R^2	0.992	0.985	0.985
Adjusted R^2	0.992	0.985	0.985
Residual Std. Error (df = 3069)	0.056	0.078	0.076
Note:			*p<0.1; **p<0.05; ***p<0.01

This table displays results from three OLS regressions estimated on data from 48 simulation runs across market conditions that vary the amount of guests and hosts in the market.

Table 9: Returns to Scale: Better Ranking

	Dependent variable:		
	Log(Contacters)	Log(Contacts)	Log(Bookings)
	(1)	(2)	(3)
Log(Guests)	0.935***	0.939***	0.879***
	(0.001)	(0.002)	(0.002)
log(Hosts)	0.164***	0.230***	0.207***
	(0.001)	(0.002)	(0.002)
Constant	−1.720***	−1.489***	−2.199***
	(0.014)	(0.020)	(0.019)
Observations	3,072	3,072	3,072
R^2	0.994	0.988	0.987
Adjusted R^2	0.994	0.988	0.987
Residual Std. Error (df = 3069)	0.049	0.071	0.068
Note:			*p<0.1; **p<0.05; ***p<0.01

This table displays results from three OLS regressions estimated on data from 48 simulation runs with a counterfactual ranking algorithm across market conditions that vary the amount of guests and hosts in the market.

Table 10: Non-linear Returns to Scale

	Dependent variable:					
	Log(Contacters)	Log(Contacts)	Log(Bookings)	Log(Contacters)	Log(Contacts)	Log(Bookings)
	(1)	(2)	(3)	(4)	(5)	(6)
Log(Guests)	0.962***	0.958***	0.944***	0.982***	0.992***	0.939***
	(0.008)	(0.010)	(0.010)	(0.014)	(0.019)	(0.019)
log(Hosts)	0.075***	0.112***	0.090***	0.031**	0.047**	0.035*
	(0.008)	(0.010)	(0.010)	(0.014)	(0.019)	(0.020)
Constant	−1.506***	−1.103***	−2.036***	−1.315***	−0.861***	−1.595***
	(0.073)	(0.095)	(0.096)	(0.154)	(0.213)	(0.217)
Market Size	Low	Low	Low	High	High	High
Observations	432	432	432	432	432	432
R^2	0.972	0.954	0.951	0.922	0.863	0.844
Note:						*p<0.1; **p<0.05; ***p<0.01

This table displays results from six OLS regressions estimated on data from 48 simulation runs across market conditions that vary the amount of guests and hosts in the market. Columns (1) - (3) display results when the number of both guests and hosts is less than the number observed in the data. Columns (4) - (6) display results when the number of both guests and hosts is greater than the number observed in the data.

Table 11: Heterogenuous Returns to User Acquisition

	Baseline (1)	Bottom 25% Listings (2)	Top 25% Listings (3)	Bottom 25% Searchers (4)	Top 25% Searchers (5)
Booked	203.62	195.25	214.71	237.52	241.69
Contacts	677.48	634.17	741.38	797.52	799.04
Contacters	351.56	338.42	375.10	416.73	418.12
Sequential Contacts	191.71	170.90	206.79	217.54	217.17
Rejections	203.77	183.92	224.38	247.31	242.77
Share Rejected	0.30	0.29	0.30	0.31	0.30
Share Stale Rej.	0.05	0.05	0.04	0.05	0.05
Share Screen Rej.	0.22	0.21	0.22	0.22	0.21
Num Guests	1,169	1,169	1,169	1,394	1,394
Num Hosts	1,958	2,448	2,448	1,958	1,958

This table displays results from countefactuals where the number of guests or listings is increased by 25%. The guests and hosts are drawn either from the bottom or top quartile in an ordering based on a ranking intended to capture predicted quality as a function of observables.

Table 12: Causal Effects

Treatment Share	Diff. Contact to Book	Diff. Search to Contact	Diff. Search to Book	Diff. Congested Share
0.250	0.045	0.127	0.092	0.003
0.500	0.021	0.113	0.074	0.027
0.750	0.017	0.107	0.069	0.042

(a) Direct Effects

Treatment Share	Diff. Contact to Book	Diff. Search to Contact	Diff. Search to Book	Diff. Congested Share
0.250	-0.006	-0.001	-0.002	-0.001
0.500	-0.001	0.0003	-0.00002	-0.006
0.750	0.001	0.002	0.002	-0.010

(b) Indirect Effects

Treatment Share	Diff. Contact to Book	Diff. Search to Contact	Diff. Search to Book	Diff. Congested Share
0.250	0.008	0.031	0.021	0.0001
0.500	0.006	0.052	0.032	0.018
0.750	0.014	0.082	0.053	0.025
1	0.004	0.100	0.060	0.043

(c) Overall Effects

These tables display three types of causal effects across policies that vary the share of searchers treated with a better ranking algorithm. Direct effects are the averge differences between treated and control units under a given policy. Indirect effects are the average differences between control units under a policy and control units when no searchers are treated. Overall effects are the average market-level differences the policies against a policy in which no searchers are treated.